Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Επικοινωνιων, Ηλεκτρονικης και Συστηματων Πληροφορικης

Παραγωγή Σημασιολογικού Περιεχομένου από Δομημένες και Ημιδομημένες Πηγές Δεδομένων

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Δημήτριος-Εμμανουήλ Α. Σπανός

Αθήνα, Μάρτιος 2013

Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Επικοινωνιων, Ηλεκτρονικης και Συστηματων Πληροφορικης

Παραγωγή Σημασιολογικού Περιεχομένου από Δομημένες και Ημιδομημένες Πηγές Δεδομένων

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Δημήτριος-Εμμανουήλ Α. Σπανός

Συμβουλευτική Επιτροπή: Νικόλαος Μ. Μήτρου
Ευστάθιος Δ. Συκάς
Ιωάννης Βασιλείου

Εγκρίθηκε από την επταμελή εξεταστική επιτροπή την 11η Μαρτίου 2013.

.....	.....	.....
Ν. Μήτρου	Ε. Συκάς	Ι. Βασιλείου
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.
.....	.....	.....
Μ. Θεολόγου	Κ. Κοντογιάννης	Ν. Παπασπύρου
Καθηγητής Ε.Μ.Π.	Αν. Καθηγητής Ε.Μ.Π.	Επ. Καθηγητής Ε.Μ.Π.

.....
Μ. Βαζιργιάννης
Καθηγητής Οικονομικού
Πανεπιστημίου Αθηνών

Αθήνα, Μάρτιος 2013

.....
Δημήτριος-Εμμανουήλ Α. Σπανός

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

Περιεχόμενα

Κατάλογος σχημάτων

Κατάλογος πινάκων

Αφιερωμένη
στους γονείς μου
Βάσω και Θανάση
και στη γιαγιά Μαρίκα!

Περίληψη

Ο Παγκόσμιος Ιστός αποτελεί πλέον αναπόσπαστο κομμάτι της καθημερινότητας, έχοντας αλλάξει τον τρόπο με τον οποίο επικοινωνούμε με τους συνανθρώπους μας, δημιουργούμε, μοιραζόμαστε και αναζητούμε πληροφορία. Ο Σημασιολογικός Ιστός φιλοδοξεί να φέρει μια αντίστοιχη επανάσταση στη χρήση αυτού του τεράστιου όγκου διαθέσιμης πληροφορίας, επιτρέποντας σε προγραμματιστικές διαδικασίες να αξιοποιήσουν τη σημασία της, να εξάγουν συμπεράσματα από αυτήν και να τη συνδυάσουν με άλλη πληροφορία με τρόπο ωφέλιμο για τον ανθρώπινο χρήστη. Ωστόσο, το όραμα του Σημασιολογικού Ιστού δεν έχει ακόμα υλοποιηθεί στον επιθυμητό βαθμό και μια αιτία για αυτό αποτελεί η έλλειψη ικανής ποσότητας δεδομένων άμεσα αξιοποιήσιμων από σημασιολογικές εφαρμογές. Στην παρούσα διατριβή, εξετάζεται το πρόβλημα της παραγωγής σημασιολογικού περιεχομένου από υπάρχουσες δομημένες και ημιδομημένες πηγές δεδομένων, με απώτερο στόχο τη μεταφορά του πλούτου της πληροφορίας που ενυπάρχει σε αυτές στο Σημασιολογικό Ιστό. Αρχικά, εξετάζεται το πρόβλημα της συμμετοχής και αξιοποίησης σχεσιακών ΒΔ στο πλαίσιο του Σημασιολογικού Ιστού και πραγματοποιείται μια εκτεταμένη βιβλιογραφική επισκόπηση, η οποία μεταξύ άλλων περιλαμβάνει τα προβλήματα της παραγωγής οντολογίας από μια σχεσιακή ΒΔ, της εξαγωγής των περιεχομένων μιας σχεσιακής ΒΔ σε έναν RDF γράφο, καθώς και το πρόβλημα της ανακάλυψης αντιστοιχιών μεταξύ σχεσιακής ΒΔ και οντολογίας. Στη συνέχεια, περιγράφεται ένα απλό σύστημα αντιστοιχίας μιας σχεσιακής βάσης δεδομένων με μια οντολογία, το οποίο προτείνει τη χρήση της SQL για τον ορισμό της αντιστοιχίας, και τονίζονται οι θεωρητικές αδυναμίες και ελλείψεις μιας τέτοιας προσέγγισης. Επίσης, αναλύεται ένας αλγόριθμος για τη μετεγγραφή SPARQL ερωτημάτων σε σημασιολογικά ισοδύναμα SQL υπό την παρουσία μιας R2RML αντιστοιχίας, δυνατότητα που επιτρέπει τη δυναμική πρόσβαση στα περιεχόμενα μιας βάσης δεδομένων μέσω γλωσσών σημασιολογικών ερωτημάτων. Τέλος, εξετάζεται το πρόβλημα της σημασιολογικής επισημείωσης και επεξεργασίας ημιδομημένων δεδομένων από δυναμικές πηγές όπως δίκτυα αισθητήρων και προτείνεται κατάλληλη επέκταση σε ένα υλοποιημένο σύστημα επίγνωσης περιβάλλοντος, η οποία εφαρμόζει παραθυρικές τεχνικές προκειμένου να διατηρήσει το χρόνο απόκρισής του εντός αποδεκτών ορίων.

Λέξεις-κλειδιά: Σημασιολογικός Ιστός, Σχεσιακή βάση δεδομένων, Οντολογία, RDF, Αντιστοιχία, Εικονική βάση γνώσης, SPARQL, R2RML, Ροή δεδομένων, Δίκτυα αισθητήρων, Παράθυρο οντολογικών ατόμων

Abstract

The World Wide Web has become a part of everyday life, having changed the way people communicate, as well as the way we create, share and search for information. The Semantic Web seeks to revolutionize the way this huge amount of available information is used, allowing automated procedures to make use of its meaning, infer new facts and integrate it with other information in a way that is meaningful to the end user. However, the Semantic Web vision has not been fully materialized yet and one of many possible reasons is the lack of an adequate critical mass of data that can be readily used by semantic applications. Therefore, this thesis investigates the issue of semantic content generation from existing structured and semistructured data sources, with the ultimate goal of bringing this entire wealth of information in the Semantic Web. First of all, an extended literature survey is carried out in order to clarify the various aspects of relational database usage in the Semantic Web context and identify all relevant issues, such as ontology generation from a relational schema, the export of relational database contents in the form of an RDF graph and the discovery of mappings between a relational database and an ontology. A simplified mapping system that merely uses SQL queries for the mapping definition is then proposed and the theoretical and practical shortcomings of this approach are pointed out. Furthermore, an algorithm for the rewriting of SPARQL queries to semantically equivalent SQL ones in the presence of an R2RML mapping is analysed, allowing for dynamic access of relational database contents via semantic queries. Finally, the problem of semantic annotation and processing of semistructured data from dynamic sources is investigated and an appropriate extension to an already implemented context-aware system is proposed. This extension applies windowing techniques in the incoming data stream in order to keep the response time of the system under acceptable levels.

Keywords: Semantic Web, Relational database, Ontology, RDF, Mapping, Virtual knowledge base, SPARQL, R2RML, Data stream, Sensor networks, Individual window

Ευχαριστίες

Το κείμενο αυτό αποτελεί το επιστέγασμα της ερευνητικής προσπάθειας που άρχισε πριν από 5 περίπου χρόνια και ουσιαστικά σηματοδοτεί το τέλος μιας πορείας. Μιας πορείας δύσκολης και απαιτητικής, γεμάτης εμπόδια και προκλήσεις, αλλά και κρυμμένους θησαυρούς ανεκτίμητης αξίας. Η διαδικασία εκπόνησης μιας διδακτορικής διατριβής αποτελεί ένα συχνά μοναχικό ταξίδι αυτογνωσίας, διαμόρφωσης χαρακτήρα και αντιμετώπισης προσωπικών αδυναμιών, το οποίο όμως δύσκολα μπορεί να ολοκληρωθεί χωρίς την παρουσία και τη συνδρομή ανθρώπων που λειτουργούν ως αρωγοί αλλά και συνταξιδιώτες σε αυτό.

Έτσι λοιπόν, αρχικά, θα ήθελα να ευχαριστήσω θερμά τον Καθηγητή κ. Νικόλαο Μήτρου, ο οποίος είχε την επίβλεψη της συγκεκριμένης διδακτορικής διατριβής και μου έδωσε τη δυνατότητα να έρθω σε επαφή και να ασχοληθώ με ένα ευρύ φάσμα ερευνητικών θεμάτων. Θα ήθελα ακόμα να τον ευχαριστήσω για την ευκαιρία που μου παρείχε να συμμετέχω σε ερευνητικά προγράμματα, αποκομίζοντας πολύτιμες εμπειρίες, καθώς και για τη συνεχή και με κάθε μέσο υποστήριξή του. Επίσης, ευχαριστώ τους Καθηγητές κ.κ. Ι. Βασιλείου, Ε. Συκά, Μ. Θεολόγου και Μ. Βαζιργιάννη για τις ιδέες και τα πολύτιμα σχόλια που μου παρείχαν σε ένα κρίσιμο στάδιο της εκπόνησης της συγκεκριμένης διατριβής, όπως επίσης και τον Αν. Καθηγητή κ. Κ. Κοντογιάννη καθώς και τον Επ. Καθηγητή κ. Ν. Παπασπύρου, οι οποίοι μου έκαναν την τιμή να είναι μέλη της επταμελούς εξεταστικής επιτροπής αυτής της διατριβής.

Θέλω επίσης να ευχαριστήσω το Κοινωφελές Ίδρυμα Αλέξανδρος Σ. Ωνάσης για την τιμή που μου έκανε επιλέγοντάς με ως υπότροφο και για την τόσο αναγκαία υποστήριξή του στο μεγαλύτερο μέρος εκπόνησης της παρούσας διατριβής.

Ιδιαίτερη μνεία πρέπει επίσης να γίνει σε όλους τους συνοδοιπόρους σε αυτό το ταξίδι της γνώσης, οι οποίοι το έκαναν να φαίνεται ξεχωριστό. Συγκεκριμένα, ευχαριστώ τους Νίκο Κωνσταντίνου, Τάσο Ζαφειρόπουλο, Γιάννη Παπαϊωάννου και Μανώλη Σολιδάκη, οι οποίοι ως εμπειρότεροι μου έδωσαν πολύτιμες συμβουλές και με βοήθησαν στα πρώτα βήματα της διατριβής μου, καθώς επίσης και τους Σταμάτη Αρκουλή, Περικλή Σταύρου, Ελένη Γιαννοπούλου, Άγγελο Αναγνωστόπουλο και Πάρη Χαραλάμπου για τις ατελείωτες συζητήσεις και το χαμόγελο που μου προσέφεραν απλόχερα ακόμα και σε δύσκολες ή απαιτητικές καταστάσεις. Θέλω ακόμα να ευχαριστήσω όλους όσους κατά καιρούς μοιραστήκαμε έστω και για λίγο τον ίδιο χώρο εργασίας καθώς επίσης και όλους όσους με βοήθησαν και μου έδωσαν κίνητρο, εκούσια ή ακούσια, ώστε να ολοκληρώσω την παρούσα διατριβή.

Οφείλω ένα ευχαριστώ στους φίλους Μιχάλη Χαλά, Δημήτρη Τραϊφόρο, Νίκο Πέππα και Βαγγέλη Στραβουδάκη για όλες εκείνες τις χαλαρές και δια-

σκεδαστικές στιγμές που μου έδιναν συχνά τη δύναμη να συνεχίσω. Δεν μπορώ επίσης να μην ευχαριστήσω άτομα του στενού οικογενειακού και κοινωνικού κύκλου για την ηθική υποστήριξη που μου παρείχαν.

Τέλος, ίσως το μεγαλύτερο ευχαριστώ απευθύνεται στους γονείς μου, Θανάση και Βάσω, στη γιαγιά Μαρίκα και στο θείο Νίκο για την αγάπη, την υπομονή, την αφοσίωση και συμπαράστασή τους όλα αυτά τα χρόνια.

Δημήτριος-Εμμανουήλ Α. Σπανός

Κεφάλαιο 1

Εισαγωγή

Ο *Παγκόσμιος Ιστός (World Wide Web)* έχει αλλάξει τον τρόπο με τον οποίο οι άνθρωποι επικοινωνούν μεταξύ τους, δημιουργούν, διαδίδουν, μοιράζονται, διαχειρίζονται και ανακτούν πληροφορία, αποτελώντας ένα σημαντικό συστατικό της ευρύτερης επανάστασης που συντελείται τις τελευταίες δεκαετίες και στοχεύει στη μετατροπή του ανεπτυγμένου κόσμου σε μια κοινωνία γνώσης. Η αρχική ιδέα πάνω στην οποία βασίστηκε η ανάπτυξη του Παγκόσμιου Ιστού ήταν η δημιουργία ενός δικτύου κόμβων πληροφορίας, στο οποίο ο χρήστης θα μπορεί να πλοηγείται κατά βούληση ακολουθώντας συνδέσμους που ενώνουν τους κόμβους του δικτύου μεταξύ τους. Δηλαδή, στα πρώτα βήματα του Παγκόσμιου Ιστού, η πληροφορία προοριζόταν αποκλειστικά για ανθρώπινη κατανάλωση και αυτό, δυστυχώς, ισχύει σε μεγάλο βαθμό μέχρι σήμερα. Αυτό συνεπάγεται μικρότερη υποστήριξη του χρήστη του Παγκόσμιου Ιστού από εργαλεία λογισμικού σε διαδικασίες όπως η απάντηση σε ένα συγκεκριμένο ερώτημα, ο συνδυασμός και η επεξεργασία πληροφορίας από δύο ή περισσότερες πηγές και η αναζήτηση πληροφοριών σχετικών με μια έννοια.

Η ιδέα ενός *Σημασιολογικού Ιστού (Semantic Web)* επινοήθηκε από τον ίδιο τον ιδρυτή του Παγκόσμιου Ιστού, Sir Tim Berners-Lee, ως μία επέκταση του Web που θα καταστήσει την υπάρχουσα γνώση διαθέσιμη σε ευρεία κλίμακα και κυρίως, θα αυξήσει τη χρησιμότητά της επιτρέποντας σε προηγμένες εφαρμογές την αναζήτηση, πλοήγηση και επεξεργασία αυτής [38]. Κατά μία έννοια, ο Σημασιολογικός Ιστός φιλοδοξεί να μετατρέψει σταδιακά τον υπάρχοντα Παγκόσμιο Ιστό από έναν *ιστό εγγράφων (Web of Documents)* σε έναν *ιστό δεδομένων (Web of Data)*, όπου οντότητες του πραγματικού κόσμου θα διαθέτουν ένα μοναδικό αναγνωριστικό, το οποίο θα αντιστοιχεί σε μια διεύθυνση Ιστού που θα παρέχει περισσότερες πληροφορίες για το περιγραφόμενο αντικείμενο ή έννοια. Αυτή η τακτική δημοσίευσης σημασιολογικού περιεχομένου είναι γνωστή ως *Συνδεδεμένα Δεδομένα* (Linked Data) και κερδίζει συνεχώς σε δημοτικότητα, παράλληλα με το ομώνυμο κίνημα που προωθεί τη συγκεκριμένη φιλοσοφία.

Για την επίτευξη του στόχου της μετατροπής του Παγκόσμιου Ιστού σε έναν ιστό δεδομένων, έχουν προταθεί μια σειρά σχετικών τεχνολογιών – πρωτόκολλα και μορφότυπα – που μπορούν να χρησιμοποιηθούν για την κωδικοποίηση γνώσης έτσι ώστε αυτή να καταστεί επεξεργάσιμη από υπολογιστικά συστήματα. Αξίζει, βέβαια, να σημειωθεί ότι ο Παγκόσμιος Ιστός δεν αποτέλεσε τη μοναδική επιρροή για την ανάπτυξη αυτών των τεχνολογιών, καθώς

και ότι τα κίνητρα που συνετέλεσαν στην τελική διαμόρφωση των τεχνολογιών Σημασιολογικού Ιστού ήταν ευρύτερα από την ανάγκη για αποτελεσματικότερη χρήση του Παγκόσμιου Ιστού. Πιο συγκεκριμένα, κάποια από τα βασικά θέματα στα οποία στηρίζεται το όραμα του Σημασιολογικού Ιστού και οι τεχνολογίες του είναι:

- η κατασκευή αφηρημένων μοντέλων που αναπαριστούν πτυχές του πραγματικού κόσμου,
- η ανάπτυξη μηχανών συλλογισμού για την εξαγωγή χρήσιμων συμπερασμάτων από κωδικοποιημένη γνώση και
- η δυνατότητα ανταλλαγής ετερογενούς πληροφορίας σε ευρεία κλίμακα.

Ως εκ τούτου, σήμερα, πάνω από μία δεκαετία μετά την αρχική επινόησή του, ο Σημασιολογικός Ιστός αποτελεί ένα ξεχωριστό, πολυσυλλεκτικό πεδίο της Επιστήμης των Υπολογιστών, που αντλεί και συνδυάζει γνώση προερχόμενη από άλλα περισσότερο ώριμα πεδία, όπως η τεχνητή νοημοσύνη, η θεωρία πολυπλοκότητας, οι βάσεις δεδομένων και τα δίκτυα υπολογιστών, και του οποίου οι βασικές τεχνολογίες χρησιμοποιούνται και σε εφαρμογές και περιβάλλοντα εκτός του Παγκόσμιου Ιστού, από ενδοεταιρικά πληροφοριακά συστήματα και βιβλιοθήκες μέχρι βιοϊατρικά συστήματα και συστήματα δημόσιας διοίκησης.

Ωστόσο, το αρχικό όραμα δημιουργίας ενός Ιστού Δεδομένων με τυπικά ορισμένη σημασία σε μορφή επεξεργάσιμη από μηχανές παραμένει σε μεγάλο βαθμό ανεκπλήρωτο. Οι λόγοι είναι αρκετοί και έχουν αποτελέσει αντικείμενο συζήτησης μεταξύ των υποστηρικτών του Σημασιολογικού Ιστού και αυτών που αντιμετωπίζουν το συνολικό εγχείρημα ως κάτι ουτοπικό ή υπερτιμημένο. Ένας από τους λόγους αυτούς είναι η μειωμένη υιοθέτηση προτύπων Σημασιολογικού Ιστού και τεχνολογιών αναπαράστασης γνώσης από εταιρείες-κολοσσούς που δραστηριοποιούνται στον Παγκόσμιο Ιστό – μια στάση πάντως που σταδιακά τα τελευταία χρόνια αρχίζει να αλλάζει. Η αρχική διστακτικότητα των εν λόγω εταιρειών να επενδύσουν στις νέες αυτές τεχνολογίες οφειλόταν στην απουσία ζήτησης και μιας σχηματισμένης αγοράς για προϊόντα και εφαρμογές που χρησιμοποιούν αυτές τις τεχνολογίες. Η απουσία αυτή σχετίζεται με την έλλειψη επιτυχημένων παραδειγμάτων σχετικών εφαρμογών, οι οποίες με τη σειρά τους χρειάζονται σημαντικό όγκο σημασιολογικού περιεχομένου προκειμένου να γίνουν φανερά τα οφέλη τους. Αυτή η κατάσταση οδηγούσε και συνεχίζει ακόμα και σήμερα να τροφοδοτεί ένα φαύλο κύκλο, καθώς οι υπεύθυνοι για την παραγωγή σημασιολογικού περιεχομένου, είτε αυτοί είναι οργανισμοί είτε απλοί χρήστες, δεν έχουν κίνητρο για την μετατροπή των δεδομένων τους σε μορφή αξιοποιήσιμη από σημασιολογικές εφαρμογές, εφόσον δεν μπορούν να πειστούν για τα χειροπιαστά οφέλη που θα αποκομίσουν από την κίνησή τους αυτή.

Σε πλήρη αντιστοιχία με το όραμα του Σημασιολογικού Ιστού που έχει θέσει ως στόχο την προσθήκη νοήματος στα περιεχόμενα του Παγκόσμιου Ιστού, τα τελευταία χρόνια προωθείται παράλληλα η ιδέα ενός Σημασιολογικού Ιστού Αισθητήρων (Semantic Sensor Web) [176], ο οποίος θα προσδώσει σαφώς ορισμένο νόημα σε δίκτυα αισθητήρων, των οποίων ο αριθμός και το μέγεθος ολοένα αυξάνεται. Η τεχνική και συντακτική διαλειτουργικότητα αισθητήριων

κόμβων εξασφαλίζεται μέσω καθιερωμένων πρωτοκόλλων επικοινωνίας (π.χ. IEEE 802.15.4, IEEE 802.15.1), καθώς και προτύπων ανταλλαγής μηνυμάτων (π.χ. IEEE 1451). Ο στόχος της σημασιολογικής διαλειτουργικότητας, ο οποίος είναι σαφώς πιο απαιτητικός, θα συνεισφέρει στην αυτοματοποίηση διαδικασιών αίσθησης και παρατήρησης, επιτρέποντας την αυτοματοποιημένη ανακάλυψη και συνεργασία μεταξύ αισθητήριων συσκευών αλλά και την αυτόματη λήψη αποφάσεων μέσω εφαρμογής διαδικασιών συλλογισμού. Για την υλοποίηση αυτού του στόχου, ο Σημασιολογικός Ιστός Αισθητήρων χρησιμοποιεί τεχνολογίες Σημασιολογικού Ιστού, προκειμένου να εμπλουτίσει τις αισθητήριες παρατηρήσεις και μετρήσεις με χωρικά, χρονικά και θεματικά μεταδεδομένα.

Κίνητρο αυτής της διατριβής, η οποία εντάσσεται στο γενικότερο ερευνητικό πεδίο του Σημασιολογικού Ιστού, αποτελεί η γενικότερη ανάγκη για μεθόδους μαζικής δημιουργίας σημασιολογικού περιεχομένου από υπάρχοντα δεδομένα, η οποία, όπως προαναφέρθηκε, εκτιμάται ότι θα συνεισφέρει στην υλοποίηση των οραμάτων του Ιστού Δεδομένων και του Σημασιολογικού Ιστού Αισθητήρων και στη γενίκευση των ωφελειών που αυτά ευαγγελίζονται. Οι δύο σημαντικότερες εξ αυτών είναι, αφενός, η δυνατότητα ανάκτησης περιεχομένου από πηγές δεδομένων με χρήση ορολογίας υψηλού επιπέδου, κατανοητής από τον άνθρωπο σε μεγαλύτερο βαθμό σε σύγκριση με την ορολογία εσωτερικής αποθήκευσης των δεδομένων και, αφετέρου, η δυνατότητα ολοκλήρωσης και συνδυασμού δεδομένων από διαφορετικές, ετερογενείς πηγές δεδομένων, δεδομένου ότι το περιεχόμενο των πηγών αυτών θα είναι εκφρασμένο ή θα μπορεί να ιδωθεί σε όρους ενός κοινού μοντέλου, με σαφώς ορισμένη σημασία.

Στην κατεύθυνση αυτή, η συγκεκριμένη διατριβή προτείνει και εξετάζει μεθόδους παραγωγής σημασιολογικού περιεχομένου από πρωτογενή δεδομένα που δεν βρίσκονται σε μορφή άμεσα αξιοποιήσιμη από εφαρμογές Σημασιολογικού Ιστού. Τα δεδομένα αυτά μπορεί να είναι αποθηκευμένα σε σχεσιακές βάσεις δεδομένων ή να προέρχονται από δυναμικές πηγές όπως δίκτυα αισθητήρων, σχηματίζοντας ροές δεδομένων. Στην πρώτη περίπτωση, τα δεδομένα ακολουθούν μια αυστηρή μορφή οργάνωσης και είναι γνωστά ως *δομημένα*, ενώ στη δεύτερη περίπτωση, η δομή των δεδομένων είναι ελλιπώς προδιαγεγραμμένη ή υπονοούμενη με αποτέλεσμα να αναφερόμαστε πλέον σε *ημιδομημένα* δεδομένα.

Το υπόλοιπο κείμενο διαρθρώνεται σε κεφάλαια ως εξής: το κεφάλαιο 2 παρέχει την απαραίτητη προκαταρκτική γνώση για την κατανόηση των θεμάτων που πραγματεύεται η παρούσα διατριβή, ενώ το κεφάλαιο 3 αποτελεί μια επισταμένη πρωτότυπη βιβλιογραφική επισκόπηση του ερευνητικού χώρου που εξετάζει τη σύμπραξη σχεσιακών βάσεων δεδομένων και Σημασιολογικού Ιστού, θέμα που αποτελεί και το κύριο αντικείμενο της διατριβής αυτής. Το κεφάλαιο 4 περιγράφει ένα απλό σύστημα αντιστοιχίας μιας σχεσιακής βάσης δεδομένων με μια οντολογία, το οποίο προτείνει τη χρήση της SQL για τον ορισμό της αντιστοιχίας, και τονίζει τις θεωρητικές αδυναμίες και ελλείψεις μιας τέτοιας προσέγγισης. Στο κεφάλαιο 5, παρουσιάζεται ένα σύστημα αντιστοιχίας που χρησιμοποιεί τη γλώσσα R2RML και τεκμηριώνεται ένας αλγόριθμος για τη μετεγγραφή SPARQL ερωτημάτων σε ισοδύναμα SQL, δυνατότητα που επιτρέπει τη δυναμική πρόσβαση στα περιεχόμενα μιας βάσης δεδομένων μέσω γλωσσών σημασιολογικών ερωτημάτων. Το κεφάλαιο 6 πραγματεύεται

το ζήτημα της σημασιολογικής επισημείωσης και επεξεργασίας ημιδομημένων δεδομένων από δίκτυα αισθητήρων και επεκτείνει ένα υπάρχον σύστημα επίγνωσης περιβάλλοντος με τη δυνατότητα εφαρμογής παραθύρων το μέγεθος των οποίων εκφράζεται σε όρους οντοτήτων. Το κείμενο της διατριβής ολοκληρώνεται με το κεφάλαιο 7, όπου συγκεντρώνονται τα συμπεράσματα της διατριβής και οι κατευθύνσεις μελλοντικής σχετικής έρευνας.

Κεφάλαιο 2

Θεωρητικό Υπόβαθρο

Περιεχόμενα

Σε αυτό το κεφάλαιο, επιχειρείται μια συνοπτική εισαγωγή στο γνωστικό πεδίο της παρούσας πρότασης, ορίζοντας τις κυριότερες έννοιες και τεχνολογίες από τα πεδία των βάσεων δεδομένων, του Σημασιολογικού Ιστού, αλλά και των ασύρματων δικτύων αισθητήρων, στις οποίες βασίζεται το υπόλοιπο του κειμένου. Στόχος είναι η χρήση του συγκεκριμένου κεφαλαίου ως σημείου αναφοράς για βασικές έννοιες που χρησιμοποιούνται στην παρούσα διατριβή και για αυτό το σκοπό, στο υπόλοιπο του κειμένου, γίνεται εκτενής χρήσης παραπομπών προς το τρέχον κεφάλαιο. Αξίζει να τονιστεί ότι στόχος είναι η διευκόλυνση του αναγνώστη για την καλύτερη κατανόηση του κειμένου και συνεπώς, μια ενδελεχής και διεξοδική παρουσίαση των παραπάνω πεδίων ξεφεύγει από το συγκεκριμένο πλαίσιο. Σε περιπτώσεις όπου απαιτείται μεγαλύτερη ανάλυση μιας τεχνολογίας ή έννοιας, αυτή πραγματοποιείται στο αντίστοιχο κεφάλαιο.

Πιο συγκεκριμένα, στην ενότητα 2.1, αναφέρονται οι βασικές αρχές εννοιολογικού και λογικού σχεδιασμού μιας βάσης δεδομένων, με ιδιαίτερη έμφαση

στην περιγραφή του μοντέλου οντοτήτων-συσχετίσεων και του σχεσιακού μοντέλου, αλλά και στη μετάβαση από το πρώτο στο δεύτερο, ενώ γίνεται και μια σύντομη αναφορά στη σχεσιακή άλγεβρα. Η ενότητα 2.2 εισάγει τα θεμελιώδη μοντέλα και τεχνολογίες του Σημασιολογικού Ιστού, παρέχοντας πληροφορίες μεταξύ άλλων για το RDF μοντέλο, τη γλώσσα ορισμού οντολογιών OWL και πρότυπα όπως η SPARQL και η R2RML. Τέλος, η ενότητα 2.3 εισάγει την έννοια του Σημασιολογικού Ιστού Αισθητήρων και των προτύπων στα οποία αυτός στηρίζεται, ενώ ορίζει και βασικές έννοιες από το πεδίο της διαχείρισης ροών δεδομένων.

2.1 Βάσεις Δεδομένων

Οι βάσεις δεδομένων και η επιστήμη των βάσεων δεδομένων επηρεάζουν σε μεγάλο βαθμό την αυξανόμενη χρήση υπολογιστών και έχουν σημαντικό αντίκτυπο σε περιοχές όπου χρησιμοποιούνται υπολογιστές, όπως είναι η ιατρική, οι επιχειρήσεις, η ιατρική, οι διάφορες επιστήμες μηχανικών και η βιβλιοθηκονομία, μεταξύ άλλων. Μια *βάση δεδομένων* (database) ορίζεται ως μια οργανωμένη συλλογή συσχετιζόμενων δεδομένων, η οποία διατηρείται για ένα συγκεκριμένο σκοπό και αναπαριστά μια άποψη του πραγματικού κόσμου [78]. Ως *δεδομένα* (data) θεωρούνται γεγονότα που μπορούν να καταγραφούν και που έχουν κάποιο εγγενές νόημα. Η δημιουργία, επεξεργασία και συντήρηση μιας βάσης δεδομένων (ΒΔ) είναι δυνατή μέσα από μια συλλογή τμημάτων λογισμικού που αποτελούν αυτό που είναι γνωστό ως *σύστημα διαχείρισης βάσεων δεδομένων* (database management system ή DBMS). Μια βάση δεδομένων μαζί με το απαραίτητο λογισμικό για το χειρισμό της αποτελούν ένα *σύστημα βάσης δεδομένων* (database system).

Η ενδεδειγμένη διαδικασία σχεδιασμού μιας βάσης δεδομένων περιλαμβάνει μια σειρά βημάτων, ξεκινώντας από την ανάλυση των απαιτήσεων των τελικών χρηστών και τον εννοιολογικό σχεδιασμό και καταλήγοντας στον λογικό και φυσικό σχεδιασμό της βάσης δεδομένων. Ο *εννοιολογικός σχεδιασμός* (conceptual design) στοχεύει στην ανάπτυξη ενός μοντέλου στο οποίο συγκεντρώνονται οι έννοιες υψηλού επιπέδου και οι αντίστοιχες συσχετίσεις που χρειάζονται για την περιγραφή των δεδομένων που θα αποθηκευτούν στη βάση. Το επόμενο βήμα, αυτό του *λογικού σχεδιασμού* (logical design), αναλαμβάνει να μετατρέψει το προηγούμενο εννοιολογικό μοντέλο σε ένα σχήμα που ακολουθεί το μοντέλο δεδομένων με βάση το οποίο θα υλοποιηθεί η βάση δεδομένων. Τέλος, το στάδιο του *φυσικού σχεδιασμού* (physical design) καθορίζει τις εσωτερικές δομές αποθήκευσης και την οργάνωση των αρχείων της βάσης δεδομένων, με στόχο την βέλτιστη απόδοση του συστήματος. Στις επόμενες παραγράφους, παρουσιάζουμε επιγραμματικά τα δύο δημοφιλέστερα μοντέλα που χρησιμοποιούνται στα στάδια του εννοιολογικού και λογικού σχεδιασμού μιας βάσης δεδομένων: το μοντέλο οντοτήτων-συσχετίσεων (παράγραφος 2.1.1) και το σχεσιακό μοντέλο (παράγραφος 2.1.2) αντίστοιχα, καθώς και την πλέον καθιερωμένη διαδικασία μετάβασης από το πρώτο στο δεύτερο (παράγραφος 2.1.3), ενώ κάνουμε μια σύντομη αναφορά στη σχεσιακή άλγεβρα (παράγραφος 2.1.4).

2.1.1 Μοντέλο Οντοτήτων-Συσχετίσεων

Το μοντέλο οντοτήτων-συσχετίσεων (μοντέλο ΟΣ) [61] είναι ένα διαγραμματικό μοντέλο που απεικονίζει τις οντότητες του πραγματικού κόσμου και τις μεταξύ τους συσχετίσεις, παρέχοντας μια αφαιρετική περιγραφή της πληροφορίας που αποθηκεύεται σε μια βάση δεδομένων. Προτάθηκε στα μέσα της δεκαετίας του 1970 και αποτελεί μέχρι σήμερα το πιο δημοφιλές εννοιολογικό μοντέλο που χρησιμοποιείται στη σχεδίαση μιας βάσης δεδομένων. Τα κύρια δομικά συστατικά του μοντέλου ΟΣ είναι οι *οντότητες*, οι *συσχετίσεις* και τα *γνωρίσματα*. Πιο συγκεκριμένα:

- μια **οντότητα** (entity) αποτελεί ένα βασικό αντικείμενο με ανεξάρτητη ύπαρξη που μπορεί να προσδιοριστεί μονοσήμαντα και να διακριθεί από άλλες οντότητες.

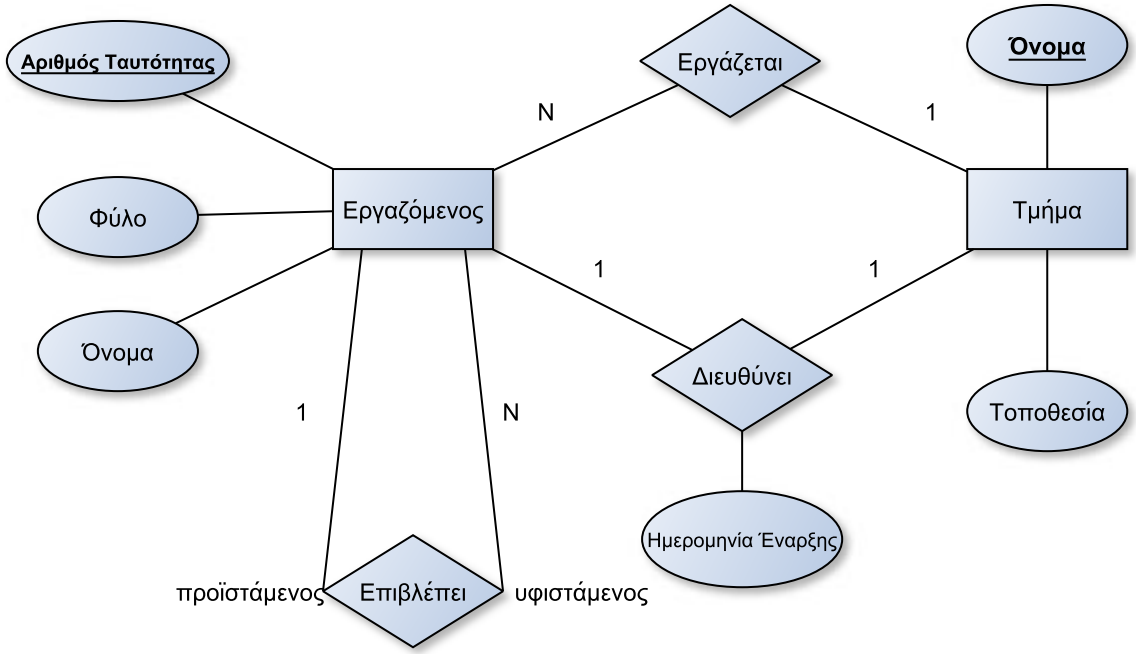
- ένα **γνώρισμα** (attribute) είναι μια ιδιότητα που χαρακτηρίζει και περιγράφει μια συγκεκριμένη οντότητα. Κάθε οντότητα μπορεί να έχει οσαδήποτε γνωρίσματα, καθένα εκ των οποίων μπορεί να έχει μία (*μονότιμο γνώρισμα*) ή περισσότερες τιμές (*πλειότιμο γνώρισμα*). Ένα γνώρισμα που μπορεί να χωριστεί σε μικρότερα επιμέρους τμήματα με ανεξάρτητη σημασία το καθένα ονομάζεται *σύνθετο* (composite), ενώ ένα γνώρισμα που δεν μπορεί να διαιρεθεί ονομάζεται *απλό* (single). Κάθε γνώρισμα συνδέεται με ένα *πεδίο ορισμού* (domain), που περιορίζει το σύνολο τιμών που μπορεί να λάβει το γνώρισμα για μια συγκεκριμένη οντότητα.

- μια **συσχέτιση** (relationship) είναι μια σύνδεση μεταξύ δύο ή περισσότερων οντοτήτων. Ο αριθμός των οντοτήτων που συμμετέχουν σε μια συσχέτιση ονομάζεται και *βαθμός* (degree) της συσχέτισης. Μια συσχέτιση μπορεί να συνοδεύεται και από έναν αριθμό γνωρισμάτων, όπως ακριβώς και μια οντότητα.

- ένας **τύπος οντοτήτων** (entity type) είναι μια ομάδα οντοτήτων που έχουν τα ίδια γνωρίσματα. Κάθε τύπος οντοτήτων έχει ένα γνώρισμα, η τιμή του οποίου προσδιορίζει μοναδικά μια συγκεκριμένη οντότητα. Το γνώρισμα αυτό ονομάζεται **κλειδί** (key). Το κλειδί ενός τύπου οντοτήτων μπορεί να είναι απλό ή σύνθετο, αν αποτελείται από περισσότερα του ενός γνωρίσματα. Στην τελευταία περίπτωση, ο συνδυασμός των τιμών των γνωρισμάτων του κλειδιού θα πρέπει να είναι διαφορετικός για κάθε οντότητα. Ένας τύπος οντοτήτων μπορεί επίσης να έχει περισσότερα από ένα κλειδιά ή και κανένα. Αν ισχύει το δεύτερο, ο τύπος οντοτήτων ονομάζεται **ασθενής** (weak) και οντότητες που ανήκουν σε αυτόν τον τύπο προσδιορίζονται μόνο μέσω της συσχέτισής τους με κάποια άλλη οντότητα που ονομάζεται *ιδιοκτήτρια οντότητα* (owner entity). Η συσχέτιση με την ιδιοκτήτρια οντότητα ονομάζεται *προσδιορίζουσα συσχέτιση* (identifying relationship). Ένας ασθενής τύπος οντοτήτων έχει ένα *μερικό κλειδί* (partial key), το οποίο διακρίνει μεταξύ τους οντότητες αυτού του τύπου που έχουν την ίδια ιδιοκτήτρια οντότητα.

- ένας **τύπος συσχέτισης** (relationship type) ορίζει ένα σύνολο ομοειδών συσχετίσεων μεταξύ οντοτήτων συγκεκριμένων τύπων. Κάθε τύπος συσχέτισης συνοδεύεται και από ένα **λόγο πληθικότητας** (cardinality ratio)

που καθορίζει τον αριθμό των συσχετίσεων αυτού του τύπου στις οποίες μπορεί να συμμετέχει μια οντότητα. Για την περίπτωση ενός δυαδικού τύπου συσχέτισης (δηλαδή, με βαθμό 2), ο λόγος πληθικότητας μπορεί να είναι 1:1, 1:N, N:1 ή M:N. Παραδείγματος χάριν, για έναν τύπο συσχέτισης μεταξύ των τύπων οντοτήτων Α και Β με λόγο πληθικότητας 1:N, ισχύει ότι κάθε οντότητα τύπου Β μπορεί να συμμετέχει σε μόνο μία συσχέτιση αυτού του τύπου, ενώ οι οντότητες του τύπου Α μπορεί να συμμετέχουν σε N συσχετίσεις αυτού του τύπου. Επίσης, ένας τύπος συσχέτισης συνοδεύεται και από **περιορισμούς συμμετοχής** (participation constraints) για τους τύπους οντοτήτων που συμμετέχουν σε αυτόν. Αν η συμμετοχή ενός τύπου οντοτήτων σε έναν τύπο συσχέτισης είναι *ολική* (total), τότε κάθε οντότητα αυτού του τύπου πρέπει υποχρεωτικά να συμμετέχει σε κάποια συσχέτιση αυτού του τύπου. Αντίθετα, αν η συμμετοχή είναι *μερική* (partial), τότε η συμμετοχή μιας οντότητας αυτού του τύπου είναι προαιρετική.



Σχήμα 2.1: *Παράδειγμα μοντέλου ΟΣ που ακολουθεί το συμβολισμό του Chen [61]*

Ένα παράδειγμα απλού μοντέλου ΟΣ με τομέα ενδιαφέροντος το εταιρικό περιβάλλον παρουσιάζεται στο σχήμα 2.1, όπου με παραλληλόγραμμα απεικονίζονται οι τύποι οντοτήτων, με ρόμβους οι τύποι συσχετίσεων και με ελλείψεις τα γνωρίσματα. Το γνώρισμα-κλειδί ενός τύπου οντοτήτων δηλώνεται με υπογράμμιση, ενώ στο σχήμα παρατίθενται επίσης και οι λόγοι πληθικότητας κάθε τύπου συσχέτισης. Αξίζει να σημειωθεί ότι ένας τύπος συσχέτισης μπορεί να συνδέει οντότητες του ίδιου τύπου, όπως φαίνεται και από τον τύπο συσχέτισης «Επιβλέπει». Σε αυτή την περίπτωση, η συσχέτιση ονομάζεται *αναδρομική* (recursive) και είναι απαραίτητη η ανάθεση ρόλων σε κάθε συμμετέχουσα οντότητα, προκειμένου να είναι σαφές το νόημα κάθε συμμετοχής: στο εν λόγω παράδειγμα, ένας εργαζόμενος που συμμετέχει σε μια συσχέτιση «Επιβλέπει» θα έχει το ρόλο του προϊσταμένου και ο άλλος το ρόλο του υφισταμένου.

Από τον αρχικό ορισμό του μοντέλου ΟΣ, προτάθηκαν διάφορες επεκτάσεις του που εισήγαγαν περισσότερα χαρακτηριστικά στοιχεία, αυξάνοντας την εκφραστικότητά του. Η ενισχυμένη έκδοση του μοντέλου ΟΣ είναι γνωστή ως **εκτεταμένο μοντέλο οντοτήτων-συσχετίσεων** (μοντέλο ΕΟΣ). Το σημαντικότερο χαρακτηριστικό που προσθέτει το μοντέλο ΕΟΣ στο αρχικό μοντέλο ΟΣ είναι η δυνατότητα ορισμού *συσχετίσεων κλάσης/υποκλάσης* μεταξύ τύπων οντοτήτων. Ένας τύπος οντοτήτων Α είναι υποκλάση ενός τύπου οντοτήτων Β, αν κάθε οντότητα τύπου Α είναι υποχρεωτικά και οντότητα τύπου Β. Παράλληλα, επιτρέπεται και ο ορισμός περιορισμών που ισχύουν για τις σχηματιζόμενες ιεραρχίες και πλέγματα τύπων οντοτήτων, όπως οι περιορισμοί *επικάλυψης* ή μη μεταξύ υποκλάσεων της ίδιας κλάσης και οι περιορισμοί συμμετοχής μιας κλάσης στις υποκλάσεις της (*μερική ή ολική*). Επίσης, το μοντέλο ΕΟΣ εισάγει την έννοια της *συνάθροισης* (aggregation), η οποία επιτρέπει τον ορισμό συσχετίσεων μεταξύ τύπων οντοτήτων και τύπων συσχετίσεων, διευκολύνοντας τη μοντελοποίηση πολύπλοκων συσχετίσεων του πραγματικού κόσμου.

2.1.2 Σχεσιακό μοντέλο

Ενώ στόχος του σταδίου του εννοιολογικού σχεδιασμού είναι η ακριβής απεικόνιση των απαιτήσεων των χρηστών σε ένα εκφραστικό μοντέλο υψηλού επιπέδου (όπως είναι το μοντέλο ΟΣ), στόχος του λογικού σχεδιασμού είναι η δημιουργία του σχήματος της βάσης δεδομένων εκφρασμένου στο μοντέλο δεδομένων που χρησιμοποιεί το εκάστοτε σύστημα διαχείρισης βάσης δεδομένων (ΣΔΒΔ). Παραδείγματα λογικών μοντέλων δεδομένων είναι το *ιεραρχικό* (hierarchical), το *δικτυωτό* (network), το *αντικειμενοστρεφές* (object-oriented) και το *σχεσιακό* (relational), με το τελευταίο να είναι το πιο δημοφιλές και την αντίστοιχη κατηγορία των σχεσιακών βάσεων δεδομένων να παρουσιάζει πλήθος εμπορικών και ελεύθερων ΣΔΒΔ. Το σχεσιακό μοντέλο προτάθηκε από τον Edgar Codd το 1970 [63] και τα κυριότερα συστατικά στοιχεία του είναι τα εξής:

- η **σχέση** (relation). Η σχέση αποτελεί το βασικό δομικό συστατικό του σχεσιακού μοντέλου και μπορεί να παρομοιαστεί με έναν πίνακα τιμών. Μια σχέση είναι ένα σύνολο από *πλειάδες* (tuples), οι οποίες μπορούν να ιδωθούν και ως οι γραμμές ενός πίνακα.

- το **γνώρισμα** (attribute). Μια σχέση μπορεί να έχει ένα ή περισσότερα γνωρίσματα, τα οποία μπορούν να παρομοιαστούν με τις στήλες ενός πίνακα τιμών. Ο αριθμός των γνωρισμάτων που περιέχει μια σχέση ονομάζεται και *βαθμός* (degree ή arity) της σχέσης. Ένα σύνολο γνωρισμάτων μιας σχέσης που προσδιορίζει μοναδικά κάθε πλειάδα της και είναι ελάχιστο (με την έννοια ότι, αν αφαιρεθεί ένα οποιοδήποτε γνώρισμα, το σύνολο των υπόλοιπων γνωρισμάτων δεν έχει πλέον αυτή την ιδιότητα) ονομάζεται *υποψήφιο κλειδί* (candidate key). Αν μια σχέση έχει περισσότερα από ένα υποψήφια κλειδιά, ένα από αυτά επιλέγεται για το μοναδικό προσδιορισμό των πλειάδων της σχέσης και ονομάζεται *πρωτεύον κλειδί* (primary key). Ένα σύνολο γνωρισμάτων μιας σχέσης R, των

οποίων οι τιμές είναι ίδιες με αυτές ενός υποψήφιου κλειδιού μιας σχέσης S ονομάζεται *ξένο κλειδί* (foreign key). Η σχέση R είναι γνωστή ως αναφέρουσα σχέση (referencing relation) και η σχέση S ως αναφερόμενη σχέση (referenced relation). Αυτό σημαίνει ότι κάθε πλειάδα της R (αν έχει μη κενές τιμές για τα γνωρίσματα του ξένου κλειδιού) πρέπει να αναφέρεται σε μια υπάρχουσα πλειάδα της S, περιορισμός γνωστός και ως *περιορισμός αναφορικής ακεραιότητας* (referential integrity constraint).

- το **πεδίο ορισμού** (domain), που αποτελεί ένα σύνολο από ατομικές σταθερές τιμές. Κάθε γνώρισμα έχει ένα πεδίο ορισμού και όλες οι τιμές του γνωρίσματος πρέπει να ανήκουν σε αυτό.

- το **σχήμα της σχέσης** (relation schema), το οποίο αποτελείται από το όνομα της σχέσης και μια σειρά από γνωρίσματα, καθένα εκ των οποίων χαρακτηρίζεται από ένα πεδίο ορισμού. Επίσης, το σχήμα της σχέσης συνδέεται και με ένα πλήθος περιορισμών: περιορισμών που σχετίζονται με πεδία ορισμού αλλά και δομικών περιορισμών που σχετίζονται με την ύπαρξη πρωτευόντων και ξένων κλειδιών. Το *στιγμιότυπο της σχέσης* (relation instance) είναι το σύνολο των πλειάδων που έχουν τον ίδιο αριθμό γνωρισμάτων με το σχήμα της σχέσης και ταυτόχρονα ικανοποιούν όλους τους περιορισμούς του.

- το **σχήμα της βάσης δεδομένων** (database schema), το οποίο αποτελείται από το σύνολο των σχημάτων των σχέσεων που περιλαμβάνονται στη βάση δεδομένων. Το *στιγμιότυπο της βάσης δεδομένων* (database instance) είναι το σύνολο των στιγμιοτύπων των σχέσεων που περιλαμβάνονται σε αυτή.

2.1.3 Μετάβαση από το μοντέλο ΟΣ στο σχεσιακό

Η παραγωγή ενός σχεσιακού σχήματος από ένα διαθέσιμο μοντέλο ΟΣ αποτελεί, όπως αναφέρθηκε, τον πυρήνα του λογικού σχεδιασμού μιας σχεσιακής βάσης δεδομένων. Αν και έχουν προταθεί αρκετοί αλγόριθμοι για τη μετάβαση από το μοντέλο ΟΣ στο σχεσιακό [81], μπορούν να συνοψιστούν σε μια βασική, αυτοματοποιημένη διαδικασία, η οποία ακολουθείται από σημαντικό αριθμό εργαλείων που προσφέρουν τη σχετική λειτουργικότητα. Στην παρούσα παράγραφο περιγράφουμε σε αδρές γραμμές τη διαδικασία αυτή, η οποία αποτελεί τη βάση και για πλήθος αλγορίθμων που έχουν προταθεί για το αντίστροφο πρόβλημα: την ανάκτηση ενός μοντέλου ΟΣ από το οποίο προήλθε ένα σχεσιακό σχήμα (database reverse engineering). Τα καθιερωμένα βήματα για τη δημιουργία ενός σχεσιακού σχήματος από ένα μοντέλο ΟΣ, σύμφωνα με το [78], τα οποία συνοψίζονται και στον πίνακα 2.1 είναι τα ακόλουθα:

Βήμα 1. Ένας *τύπος οντοτήτων* E οδηγεί στη δημιουργία μιας σχέσης R. Όλα τα γνωρίσματα του τύπου E μεταφράζονται ως αντίστοιχα γνωρίσματα της σχέσης R. Για κάθε σύνθετο γνώρισμα *att* του τύπου E, δημιουργούνται απλά γνωρίσματα στη σχέση R για καθένα από τα συστατικά απλά γνωρίσματα του *att*. Ως πρωτεύον κλειδί της R επιλέγεται κάποιο από τα γνωρίσματα-κλειδιά του τύπου E. Αν αυτό είναι σύνθετο, τότε ως πρωτεύον κλειδί ορίζεται ο συνδυασμός των απλών γνωρισμάτων της R που το αποτελούν.

Βήμα 2. Ένας *ασθενής τύπος οντοτήτων* W οδηγεί επίσης στη δημιουργία μιας σχέσης S με γνωρίσματα που παράγονται όπως στο Βήμα 1. Επιπλέον, για όλα τα γνωρίσματα που ανήκουν στο πρωτεύον κλειδί του ιδιοκτήτη τύπου οντοτήτων E, δημιουργούνται αντίστοιχα ξένα κλειδιά που αναφέρονται στη σχέση R που έχει προκύψει από τον τύπο E. Το πρωτεύον κλειδί της S είναι ο συνδυασμός αυτών των ξένων κλειδιών με το μερικό κλειδί του ασθενούς τύπου W.

Βήμα 3. Ένας *δυαδικός τύπος συσχέτισης* r μεταξύ των τύπων οντοτήτων E_1 και E_2 με *λόγο πληθικότητας 1:1*, ενσωματώνεται σε μια από τις σχέσεις R_1 και R_2 που έχουν παραχθεί αντίστοιχα από τους δύο τύπους. Ο τύπος r απεικονίζεται ως ξένο κλειδί, που ανήκει συνήθως στη σχέση που αντιστοιχεί στον τύπο οντοτήτων (έστω ο E_1) με ολική συμμετοχή στη συσχέτιση, και αναφέρεται στο πρωτεύον κλειδί της δεύτερης σχέσης (σε αυτή την περίπτωση, η R_2). Πιθανά γνωρίσματα του τύπου r μεταφράζονται σε γνωρίσματα της R_1. Μια εναλλακτική προσέγγιση είναι η ενσωμάτωση των γνωρισμάτων των E_1 και E_2 καθώς και του τύπου συσχέτισης r σε μία σχέση R.

Βήμα 4. Παρομοίως, ένας *δυαδικός τύπος συσχέτισης* r μεταξύ των τύπων οντοτήτων E_1 και E_2 με *λόγο πληθικότητας 1:N* ενσωματώνεται σε μία από τις δύο σχέσεις που έχουν παραχθεί από τους E_1 και E_2, με τη μόνη διαφορά ότι το ξένο κλειδί τοποθετείται πλέον στη σχέση που αντιστοιχεί στον τύπο οντοτήτων που βρίσκεται από την «N πλευρά» της συσχέτισης, δηλ. στον E_2 στη δεδομένη περίπτωση. Πιθανά γνωρίσματα του τύπου r μεταφράζονται σε γνωρίσματα της E_2.

Βήμα 5. Ένας *δυαδικός τύπος συσχέτισης* r μεταξύ των τύπων οντοτήτων E_1 και E_2 με *λόγο πληθικότητας M:N* αντιστοιχεί σε μια νέα σχέση R η οποία περιέχει δύο ξένα κλειδιά προς τις σχέσεις R_1 και R_2 που έχουν παραχθεί αντίστοιχα από τους δύο τύπους οντοτήτων, καθώς και τα γνωρίσματα του τύπου συσχέτισης r, αν υπάρχουν. Ως πρωτεύον κλειδί της σχέσης R, ορίζεται ο συνδυασμός των γνωρισμάτων των δύο ξένων κλειδιών που αναφέρονται στις σχέσεις R_1 και R_2. Η συγκεκριμένη αντιστοιχία τύπου συσχέτισης με σχέση μπορεί να χρησιμοποιηθεί και για λόγους πληθικότητας 1:1 και 1:N (βήματα 3 και 4 αντίστοιχα) με τη δημιουργία αντίστοιχων ξένων κλειδιών που αναφέρονται στις σχέσεις που έχουν προκύψει από τους συμμετέχοντες τύπους οντοτήτων και πρωτεύον κλειδί να αποτελεί το ένα από τα δύο ξένα κλειδιά: αυτό του τύπου οντοτήτων με ολική συμμετοχή στην 1:1 συσχέτιση ή αυτό του τύπου οντοτήτων που βρίσκεται από την «N πλευρά» της συσχέτισης.

Βήμα 6. Ένας *τύπος συσχέτισης r βαθμού n ($n > 2$)* μεταξύ των τύπων οντοτήτων E_1, E_2, ..., E_n απεικονίζεται ως μια νέα σχέση R με n ξένα κλειδιά προς τις σχέσεις R_1, R_2, ..., R_n που έχουν προκύψει από τους n τύπους οντοτήτων. Όπως στο βήμα 5, τα γνωρίσματα του τύπου r προστίθενται στη σχέση R, ενώ πρωτεύον κλειδί της R είναι ο συνδυασμός των n ξένων κλειδιών.

Βήμα 7. Ένα *πλειότιμο γνώρισμα att* ενός τύπου οντοτήτων E αντιστοιχεί σε μια νέα σχέση S, η οποία εκτός από το πλειότιμο γνώρισμα περιέχει και ένα ξένο κλειδί προς τη σχέση R που έχει προέλθει από τον E. Το πρωτεύον κλειδί

της S είναι ο συνδυασμός αυτού του ξένου κλειδιού και του πλειότιμου γνωρίσματος *att*.

Πίνακας 2.1: *Αντιστοιχία μοντέλου ΟΣ με σχεσιακό*

Μοντέλο ΟΣ	*Σχεσιακό μοντέλο*
Τύπος οντοτήτων E	Σχέση R
Ασθενής τύπος οντοτήτων W με ιδιοκτήτη τύπο E	Σχέση S με ξένα κλειδιά που αναφέρονται στη σχέση R
Απλό γνώρισμα *att*	Γνώρισμα *col*
Σύνθετο γνώρισμα X με συστατικά απλά γνωρίσματα $attrs(X)$	Γνωρίσματα $attrs(X)$
Πλειότιμο γνώρισμα *att* στον τύπο οντοτήτων E	Σχέση S με πρωτεύον κλειδί το συνδυασμό του κλειδιού της R και του *att*
Δυαδικός τύπος συσχέτισης r μεταξύ τύπων οντοτήτων E_1 και E_2:	Ξένο κλειδί που ορίζεται:
με λόγο πληθικότητας 1:1	στη σχέση με ολική συμμετοχή στον τύπο r
με λόγο πληθικότητας 1:N	στη σχέση R_2 που βρίσκεται από την «N πλευρά» της συσχέτισης
Δυαδικός τύπος συσχέτισης r μεταξύ τύπων οντοτήτων E_1 και E_2 με λόγο πληθικότητας M:N	Σχέση R με ξένα κλειδιά σε R_1 και R_2
Τύπος συσχέτισης βαθμού n ($n > 2$) μεταξύ E_1, E_2, ..., E_n	Σχέση R με n ξένα κλειδιά σε R_1, R_2, ..., R_n

Ακολουθώντας τα παραπάνω βήματα για το απλό μοντέλο ΟΣ του σχήματος 2.1, καταλήγουμε στο επόμενο σχεσιακό σχήμα, όπου με έντονη γραφή δηλώνονται τα πρωτεύοντα κλειδιά των σχέσεων, ενώ οι περιορισμοί ξένων κλειδιών δηλώνονται ως ζεύγη γνωρισμάτων με το πρώτο να αποτελεί το ξένο κλειδί και το δεύτερο το πρωτεύον κλειδί στο οποίο αναφέρεται.

ΤΜΗΜΑ(**Όνομα**, Τοποθεσία)
ΕΡΓΑΖΟΜΕΝΟΣ(**Αρ_Ταυτ**, Όνομα, Φύλο, Προϊστάμενος, Τμήμα, Διευθύνει)
ξένο_κλειδί(ΕΡΓΑΖΟΜΕΝΟΣ.Προϊστάμενος, ΕΡΓΑΖΟΜΕΝΟΣ.Αρ_Ταυτ)
ξένο_κλειδί(ΕΡΓΑΖΟΜΕΝΟΣ.Τμήμα, ΤΜΗΜΑ.Όνομα)
ξένο_κλειδί(ΕΡΓΑΖΟΜΕΝΟΣ.Διευθύνει, ΤΜΗΜΑ.Όνομα)

Η παραπάνω διαδικασία μπορεί να εμπλουτιστεί με περισσότερα βήματα προκειμένου να συμπεριλάβει και επιπλέον χαρακτηριστικά που συναντώνται σε επεκτάσεις του μοντέλου ΟΣ, όπως η συσχέτιση κλάσης/υποκλάσης και η συνάθροιση, που αποτελούν μέρη του μοντέλου ΕΟΣ. Βέβαια, η απεικόνιση δομών όπως οι παραπάνω, οι οποίες δεν περιέχονται στο απλό μοντέλο ΟΣ, χαρακτηρίζεται από περισσότερες εναλλακτικές επιλογές μοντελοποίησης, γεγονός που φυσιολογικά δυσκολεύει και την ανάστροφη διαδικασία ανάκτησης του εννοιολογικού μοντέλου από ένα υπάρχον σχεσιακό. Ενδεικτικά, παρουσιάζουμε μερικές από τις προσεγγίσεις που έχουν προταθεί [119] για την απεικόνιση μιας συσχέτισης κλάσης/υποκλάσης μεταξύ των τύπων οντοτήτων E_1 και E_2, με τον E_1 να αποτελεί υπερκλάση του E_2 ($E_1 \sqsupseteq E_2$) στο σχεσιακό μοντέλο.

Επιλογή 1. Δημιουργία δύο σχέσεων R_1 και R_2 για τους τύπους E_1 και E_2 αντίστοιχα με κοινό πρωτεύον κλειδί το γνώρισμα-κλειδί του E_1. Η σχέση R_2 περιέχει τόσο τα γνωρίσματα του τύπου-υπερκλάση E_1 όσο και τα *ιδιαίτερα γνωρίσματα* (specific attributes) του τύπου E_2. Όταν ακολουθείται η συγκεκριμένη επιλογή, συνηθίζεται η σχέση R_1 να περιέχει μόνο εκείνες τις οντότητες που είναι τύπου E_1 αλλά όχι E_2, για λόγους αποφυγής πλεονασματικής αποθήκευσης δεδομένων.

Επιλογή 2. Δημιουργία δύο σχέσεων R_1 και R_2 για τους τύπους E_1 και E_2 αντίστοιχα, με το πρωτεύον κλειδί της R_2 να είναι ένα ξένο κλειδί που αναφέρεται στο πρωτεύον κλειδί της R_1. Η διαφορά με τον προηγούμενο σχεδιασμό έγκειται στο γεγονός ότι η R_2 περιέχει ως γνωρίσματα μόνο τα ιδιαίτερα γνωρίσματα του τύπου E_2, με τα γνωρίσματα της υπερκλάσης να αποθηκεύονται στην R_1.

Επιλογή 3. Δημιουργία μιας σχέσης R, η οποία περιέχει το σύνολο των γνωρισμάτων των τύπων E_1 και E_2 και, ως εκ τούτου, χρησιμεύει για την αποθήκευση όλων των οντοτήτων τύπου E_1. Ο σχεδιασμός αυτός μπορεί να οδηγήσει σε μεγάλο πλήθος κενών (null) τιμών, καθώς τα ιδιαίτερα γνωρίσματα της υποκλάσης E_2 δεν έχουν νόημα για τις οντότητες τύπου E_1 που *δεν* ανήκουν στην E_2.

Παραδείγματος χάριν, η απεικόνιση της συσχέτισης κλάσης/υποκλάσης μεταξύ του τύπου οντοτήτων *Άτομο* με γνωρίσματα *Αρ_Ταυτ*, *Όνομα* και *Ηλικία* και του τύπου *Φοιτητής* με ιδιαίτερα γνωρίσματα *Σχολή* και *Επίπεδο* μπορεί να είναι μία από τις ακόλουθες.

Επιλογή 1:
ΑΤΟΜΟ(**Αρ_Ταυτ**, Όνομα, Ηλικία)
ΦΟΙΤΗΤΗΣ(**Αρ_Ταυτ**, Όνομα, Ηλικία, Σχολή, Επίπεδο)

Επιλογή 2:
ΑΤΟΜΟ(**Αρ_Ταυτ**, Όνομα, Ηλικία)
ΦΟΙΤΗΤΗΣ(**Αρ_Ταυτ**, Σχολή, Επίπεδο)
ξένο_κλειδί(ΦΟΙΤΗΤΗΣ.Αρ_Ταυτ, ΑΤΟΜΟ.Αρ_Ταυτ)

Επιλογή 3:
ΑΤΟΜΟ(**Αρ_Ταυτ**, Όνομα, Ηλικία, Σχολή, Επίπεδο)

Βέβαια, το σχεσιακό σχήμα που προκύπτει από τη διαδικασία λογικού σχεδιασμού δεν είναι απαραίτητα και αυτό που χρησιμοποιείται τελικά για την αποθήκευση των δεδομένων. Συχνά ακολουθεί και μια διαδικασία, γνωστή ως *κανονικοποίηση* (normalization), η οποία διασπά τις αρχικες σχέσεις σε μικρότερες, έτσι ώστε να αποφεύγονται πλεονασμοί στην αποθήκευση των δεδομένων που με τη σειρά τους μπορεί να οδηγήσουν σε ανωμαλίες κατά τις διαδικασίες της εισαγωγής, της διαγραφής ή της τροποποίησης δεδομένων. Το αποτέλεσμα είναι ένα σχεσιακό σχήμα όπου κάθε νέα αποσυντεθειμένη (decomposed) σχέση ικανοποιεί μια σειρά κριτηρίων, ανάλογα με την αυστηρότητα των οποίων το τελικό σχήμα είναι λιγότερο ή περισσότερο ευάλωτο στις προαναφερθείσες ανωμαλίες. Κάθε σύνολο τέτοιων κριτηρίων ορίζει και μια

λεγόμενη *κανονική μορφή* (normal form). Οι πιο γνωστές κανονικές μορφές, κατά σειρά αυστηρότητας, είναι η πρώτη (1NF), δεύτερη (2NF), τρίτη (3NF), Boyce-Codd κανονική μορφή (BCNF) και η τέταρτη κανονική μορφή (4NF).

2.1.4 Σχεσιακή Άλγεβρα

Εκτός από τον ορισμό του σχεσιακού μοντέλου, ο Codd [63] πρότεινε και μια σειρά βασικών πράξεων για τη διαχείριση σχεσιακών δομών. Το σύνολο των πράξεων αυτών αποτελούν τη **σχεσιακή άλγεβρα** (relational algebra), η οποία έχει την ίδια εκφραστικότητα με τη λογική πρώτης τάξης (first-order logic), χωρίς την παρουσία συναρτήσεων. Η σχεσιακή άλγεβρα αποτέλεσε τη βάση για τη δημιουργία της SQL (Structured Query Language), μια δηλωτική γλώσσα όχι μόνο για χειρισμό δεδομένων (Data Manipulation Language ή DML) αλλά και για ορισμό δεδομένων (Data Definition Language ή DDL). Ο ρόλος της SQL και η καθιέρωσή της ως πρότυπη γλώσσα ερωτημάτων αποτέλεσε καθοριστικό παράγοντα για την επιτυχία και τη διάδοση των σχεσιακών ΣΔΒΔ, διευκολύνοντας μεταξύ άλλων τη μετάβαση από ένα ΣΔΒΔ σε ένα άλλο.

Οι πράξεις της σχεσιακής άλγεβρας διακρίνονται σε δύο κατηγορίες: τις τυπικές συνολοθεωρητικές πράξεις οι οποίες εφαρμόζονται και σε σχέσεις, δεδομένου του ότι μια σχέση είναι ένα σύνολο πλειάδων και τις ειδικές για σχεσιακές ΒΔ πράξεις. Στην πρώτη κατηγορία ανήκουν οι γνωστές πράξεις της *ένωσης* (union), *τομής* (intersection) και *διαφοράς* (difference), ενώ στη δεύτερη ανήκουν οι πράξεις της *επιλογής* (selection), *προβολής* (projection), *καρτεσιανού γινομένου* (cartesian product), *συνένωσης* (join) και *μετονομασίας* (renaming). Πιο συγκεκριμένα, οι πράξεις της δεύτερης κατηγορίας ορίζοντας ως εξής:

1. **Επιλογή**: Η επιλογή συμβολίζεται ως $\sigma_{\langle\text{συνθήκη επιλογής}\rangle}(R)$, όπου η συνθήκη επιλογής είναι μια λογική έκφραση των γνωρισμάτων της σχέσης R. Το αποτέλεσμα της επιλογής είναι μια νέα σχέση βαθμού ίσου με αυτόν της R που περιέχει τις πλειάδες της R που ικανοποιούν τη συνθήκη επιλογής.

2. **Προβολή**: Η προβολή συμβολίζεται ως $\pi_{\langle\text{λίστα γνωρισμάτων της R}\rangle}(R)$, και το αποτέλεσμά της είναι μια σχέση με τον ίδιο αριθμό πλειάδων με αυτόν της R, η οποία όμως διαθέτει μόνο τα γνωρίσματα της λίστας, απορρίπτοντας τα υπόλοιπα.

3. **Καρτεσιανό γινόμενο**: Το καρτεσιανό γινόμενο δύο σχέσεων συμβολίζεται ως $R \times S$ και το αποτέλεσμά του είναι ο συνδυασμός των σχέσεων R και S, διαθέτει δηλαδή το σύνολο των γνωρισμάτων των R και S και $|R| \cdot |S|$ πλειάδες όπου $|R|$, $|S|$ ο αριθμός πλειάδων των R και S αντίστοιχα.

4. **Συνένωση**: Η συνένωση δύο σχέσεων $R \bowtie_{\langle\text{συνθήκη συνένωσης}\rangle} S$ αποτελεί συνδυασμό των πράξεων του καρτεσιανού γινομένου και της επιλογής. Το αποτέλεσμα είναι το σύνολο των πλειάδων του $R \times S$ που ικανοποιούν τη συνθήκη συνένωσης.

5. **Μετονομασία**: Η πράξη της μετονομασίας γνωρισμάτων $\rho_{B_1,\ldots,B_n}(R)$ μιας σχέσης R βαθμού n παράγει μια νέα σχέση όπου τα γνωρίσματα της R έχουν μετονομαστεί σε $B_1, \ldots, B_n$.

Όπως φαίνεται από τα παραπάνω, το αποτέλεσμα κάθε πράξης της σχεσιακής άλγεβρας είναι μια σχέση, γεγονός που καθιστά την σχεσιακή άλγεβρα *κλειστή* (closed) για όλους τους παραπάνω τελεστές.

2.2 Τεχνολογίες Σημασιολογικού Ιστού

Όπως αναφέρθηκε και στο κεφάλαιο 1, ο Σημασιολογικός Ιστός αποτελεί την ιδανική εξέλιξη του τρέχοντος Παγκόσμιου Ιστού, όπου οι υπολογιστές θα είναι σε θέση να αναζητούν με έξυπνο τρόπο, να επεξεργάζονται και να συνδυάζουν περιεχόμενο με βάση το νόημα που έχει αυτό το περιεχόμενο για τους ανθρώπους. Αυτό μπορεί να επιτευχθεί μόνο αν το περιεχόμενο που υπάρχει στον Ιστό αποκτήσει τυπικό αυστηρό νόημα, κωδικοποιημένο σε κάποιο μορφότυπο ικανό για επεξεργασία από υπολογιστές [103]. Καθώς για τη διαδικασία αυτή δεν επαρκούν οι κλασικές τεχνολογίες Ιστού (όπως η HTML και η XML), έχει αναπτυχθεί μια σειρά κατάλληλων τεχνολογιών και προτύπων, γνωστών και ως τεχνολογιών Σημασιολογικού Ιστού, από το W3C (World Wide Web Consortium), τον οργανισμό που είναι υπεύθυνος για την προώθηση ανοικτών προτύπων για τον Παγκόσμιο Ιστό. Οι τεχνολογίες αυτές περιλαμβάνουν, μεταξύ άλλων, γλώσσες αναπαράστασης γνώσης, γλώσσες ερωτημάτων, πρότυπα για την εξαγωγή συμπερασμάτων από μοντέλα αναπαράστασης γνώσης και πρότυπα σημασιολογικής επισημείωσης.

Στην παρούσα ενότητα, θα περιγράψουμε συνοπτικά τις σχετικές με την παρούσα διατριβή τεχνολογίες Σημασιολογικού Ιστού, ξεκινώντας από το βασικό μοντέλο περιγραφής RDF (παράγραφος 2.2.1), συνεχίζοντας με διαδεδομένες γλώσσες αναπαράστασης γνώσης, όπως η RDF Schema και η OWL (παράγραφος 2.2.2), τη γλώσσα ερωτημάτων SPARQL (παράγραφος 2.2.3) και την, πιο πρόσφατη, γλώσσα αντιστοιχίας σχεσιακών βάσεων δεδομένων με RDF γράφους, R2RML (παράγραφος 2.2.4), ενώ γίνεται και μια σύντομη αναφορά στην έννοια των Συνδεδεμένων Δεδομένων (παράγραφος 2.2.5).

2.2.1 Το μοντέλο RDF

Το RDF (Resource Description Framework) είναι ένα μοντέλο περιγραφής δομημένης πληροφορίας και θεωρείται το βασικό μορφότυπο αναπαράστασης στο πλαίσιο του Σημασιολογικού Ιστού. Ο κύριος στόχος του RDF είναι να επιτρέψει σε εφαρμογές την ανταλλαγή δεδομένων και πληροφορίας που διατηρούν την αρχική τους σημασία. Συνεπώς, σε αντίθεση με την HTML, όπου σκοπός είναι η ορθή παρουσίαση εγγράφων του Ιστού σε ανθρώπους, το RDF μοντέλο στοχεύει στη διευκόλυνση και αυτοματοποίηση της επεξεργασίας και συνδυασμού της πληροφορίας που περιέχεται σε αυτά τα έγγραφα.

Μερικές από τις θεμελιώδεις έννοιες στις οποίες βασίζεται το RDF μοντέλο[1] είναι οι ακόλουθες:

[1] Το W3C έχει δημοσιεύσει μια σειρά εγγράφων για τον ορισμό του RDF μοντέλου, της σημασιολογίας του (semantics), δηλαδή της ερμηνείας που πρέπει να αποδοθεί σε κάθε συστατικό στοιχείο του, και των πιθανών συντακτικών μορφοτύπων στα οποία αυτό μπορεί να κωδικοποιηθεί. Το έγγραφο που προσφέρει μια σφαιρική επισκόπηση του RDF μοντέλου είναι το RDF Primer: `www.w3.org/TR/rdf-primer`.

Δομή γράφου. Το RDF μοντέλο είναι ένα μοντέλο βασισμένο σε γράφο και συγκεκριμένα, σε *κατευθυνόμενο γράφο*, δηλαδή αποτελεί ένα σύνολο *κόμβων* που συνδέονται μεταξύ τους με *κατευθυνόμενες ακμές*. Το πλεονέκτημα της υιοθέτησης δομής γράφου σε σύγκριση με τη δενδρική δομή της XML έγκειται στην ευκολότερη ενοποίηση των πρώτων, επιτρέποντας με αυτόν τον τρόπο το συνδυασμό RDF εγγράφων από πολλαπλές πηγές προέλευσης. Ένας RDF γράφος, όπως άλλωστε κάθε γράφος, μπορεί να εκφραστεί με διάφορους τρόπους (π.χ. με τη βοήθεια ενός πίνακα γειτνίασης), όμως όλα τα καθιερωμένα συντακτικά μορφότυπα RDF κωδικοποιούν έναν RDF γράφο ως το σύνολο των ακμών που τον απαρτίζουν. Οι ακμές του RDF γράφου εκφράζονται ως *τριάδες* (triples) ή αλλιώς *προτάσεις* (statements), καθώς αποτελούνται από ένα *υποκείμενο* (subject), ένα *κατηγόρημα* (predicate) και ένα *αντικείμενο* (object), με το υποκείμενο και το αντικείμενο να δηλώνουν κόμβους, ενώ η κατεύθυνση της ακμής είναι από το υποκείμενο προς το αντικείμενο της τριάδας. Μια RDF τριάδα ουσιαστικά δηλώνει μια συσχέτιση, η οποία υποδεικνύεται από το κατηγόρημα, μεταξύ των αντικειμένων ή εννοιών που υποδηλώνονται από το υποκείμενο και το αντικείμενο της τριάδας. Συχνά, το κατηγόρημα μιας RDF πρότασης αναφέρεται και ως *ιδιότητα* (property).

Ενιαία Αναγνωριστικά Πόρων (URIs). Μια ακόμα διαφοροποίηση μεταξύ του RDF μοντέλου και της XML, που αποτελεί και σημαντικό προσόν του πρώτου και καθιστά εφικτή την εύκολη σύνθεση διακριτών εγγράφων είναι το γεγονός ότι όλοι οι πόροι (δηλαδή οι κόμβοι και οι ιδιότητες) στο RDF μοντέλο έχουν ένα αναγνωριστικό, γνωστό και ως Ενιαίο Αναγνωριστικό Πόρου (Uniform Resource Identifier ή URI). Το URI αποτελεί γενίκευση του Ενιαίου Εντοπιστή Πόρων (Uniform Resource Locator ή URL), το οποίο αποτελεί μια διεύθυνση στον Ιστό για την πρόσβαση σε κάποιο έγγραφο, και χρησιμοποιείται για το σαφή διαχωρισμό μεταξύ πόρων. Εξαίρεση στον παραπάνω κανόνα αποτελούν κόμβοι οι οποίοι δεν έχουν κάποιο παγκόσμιο αναγνωριστικό, και οι οποίοι είναι γνωστοί ως *κενοί κόμβοι* (blank nodes), καθώς και κόμβοι που δεν αναπαριστούν κάποιο αντικείμενο αλλά μια δεδομένη τιμή. Ένας κενός κόμβος έχει ένα τοπικό αναγνωριστικό που είναι μοναδικό στα πλαίσια του RDF γράφου στον οποίο ανήκει. Συνεπώς, κατά τη συγχώνευση RDF γράφων, προκειμένου να διατηρηθεί το αρχικό νόημά τους, θα πρέπει να εξασφαλιστεί ότι οι κενοί κόμβοι κάθε γράφου θα έχουν διαφορετικά αναγνωριστικά.

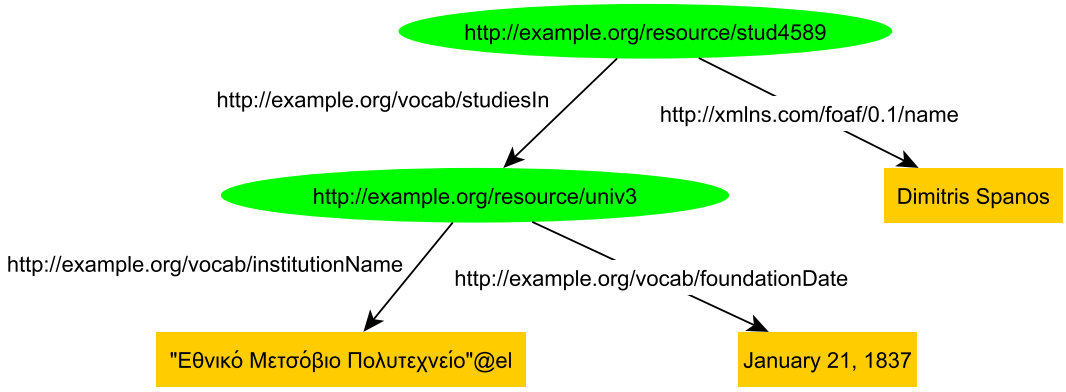
Τύποι Δεδομένων. Οι τύποι δεδομένων (datatypes) χρησιμοποιούνται στο RDF μοντέλο για την αναπαράσταση τιμών, όπως ακεραίων, αριθμών κινητής υποδιαστολής και ημερομηνιών. Ουσιαστικά, αποτελούν μια συνάρτηση απεικόνισης (lexical-to-value mapping) μεταξύ της λεκτικής αναπαράστασης μιας τιμής (π.χ. «T», «True» ή «1») και της ίδιας της τιμής (π.χ. η αληθής boolean τιμή). Το RDF μοντέλο δεν έχει κάποιον εσωτερικό μηχανισμό ορισμού νέων τύπων δεδομένων και στην πράξη, γίνεται αναφορά συνήθως σε εξωτερικά ορισμένους τύπους δεδομένων που προσδιορίζονται από ένα URI. Επικρατέστεροι για αυτόν το σκοπό είναι οι τύποι δεδομένων του XML Schema.

Λεκτικά. Οι τιμές δεδομένων αναπαριστώνται στο RDF μοντέλο μέσω ειδικών κόμβων που ονομάζονται λεκτικά (literals). Τα λεκτικά αποτελούνται από μια σειρά χαρακτήρων που δημιουργούν την λεκτική αναπαράσταση της τι-

μής, η οποία ερμηνεύεται με βάση έναν τύπο δεδομένων και της συνάρτησης απεικόνισης που αυτός ορίζει. Ένα λεκτικό που δεν συνοδεύεται από κάποιον τύπο δεδομένων ονομάζεται *απλό* (plain), σε αντίθεση με ένα *λεκτικό με τύπο δεδομένων* (typed). Ένα απλό λεκτικό μπορεί να συνοδεύεται και από μια προαιρετική γλωσσική ετικέτα (language tag) που δηλώνει τη γλώσσα στην οποία αυτό είναι εκφρασμένο. Ένα λεκτικό δεν μπορεί να χρησιμοποιηθεί στη θέση του υποκειμένου ή του κατηγορήματος μιας RDF πρότασης.

Συνεπαγωγή. Η δυνατότητα του RDF μοντέλου να αναπαριστά τη σημασία της κωδικοποιημένης σε αυτό πληροφορίας οφείλεται στη δυνατότητα συνεπαγωγής (entailment) που είναι συνυφασμένη με αυτό. Μιλώντας χωρίς αυστηρότητα, μια RDF πρόταση ή γενικότερα ένας RDF γράφος A συνεπάγεται μια RDF πρόταση ή RDF γράφο B, όταν τα αντικείμενα ή έννοιες που ικανοποιούν το A ικανοποιούν και το B. Η σχέση συνεπαγωγής μας επιτρέπει να εξάγουμε περισσότερη γνώση (τον γράφο B στη συγκεκριμένη περίπτωση) από την σαφώς δηλωμένη γνώση που κωδικοποιείται στο αρχικό RDF μοντέλο.

Ένα απλό παράδειγμα RDF γράφου απεικονίζεται στο σχήμα 2.2, όπου φαίνονται 2 κόμβοι, που αναπαριστώνται με ελλείψεις, 3 λεκτικά που αναπαριστώνται με παραλληλόγραμμα, εκ των οποίων το ένα διαθέτει γλωσσική ετικέτα.



Σχήμα 2.2: *Παράδειγμα RDF γράφου*

Καθώς η διαγραμματική αυτή αναπαράσταση του RDF μοντέλου δεν ενδείκνυται για επεξεργασία από υπολογιστές, έχουν προταθεί διάφορα μορφότυπα σειριακοποίησης (serialization formats) για την κωδικοποίηση RDF γράφων με τη χρήση απλών συμβολοσειρών χαρακτήρων. Άξια αναφοράς είναι η XML κωδικοποίηση του RDF, γνωστή ως RDF/XML, καθώς και οι συναφείς, βασισμένες σε απλό κείμενο, σειριακοποιήσεις σε Turtle, N-Triples και N3[2], οι οποίες είναι και πιο αναγνώσιμες από τον άνθρωπο.

[2]Οι Turtle (http://www.w3.org/TR/turtle/) και N-Triples (http://www.w3.org/TR/rdf-testcases/#ntriples) αποτελούν υποσύνολα της πιο εκφραστικής N3 (http://www.w3.org/TeamSubmission/n3/) που επιτρέπει την περιγραφή μονοπατιών και τον ορισμό κανόνων. Η Turtle, από την πλευρά της, προσφέρει μια πιο λιτή αναπαράσταση ενός RDF γράφου σε σχέση την N-Triples, επιτρέποντας την ομαδοποίηση τριάδων με ίδιο υποκείμενο ή και ιδιότητα.

Ο RDF γράφος του σχήματος 2.2 κωδικοποιείται σε μορφή N-Triples ως εξής (όπου για λόγους απεικόνισης έχουν συντομευθεί τα URIs των πόρων του μοντέλου ορίζοντας αντίστοιχους χώρους ονομάτων):

```
@prefix ex: <http://example.org/resource/>.
@prefix vocab: <http:///example.org/vocab/>.
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
ex:stud4589 vocab:studiesIn ex:univ3.
ex:stud4589 foaf:name ‘‘Dimitris Spanos’’.
ex:univ3 vocab:institutionName ‘‘Εθνικό Μετσόβιο Πολυτεχνείο’’@el.
ex:univ3 vocab:foundationDate ‘‘January 21, 1837’’.
```

Ενώ οι ίδιες 4 τριάδες μπορούν να αναπαρασταθούν πιο περιεκτικά σε μορφή Turtle:

```
@prefix ex: <http://example.org/resource/>.
@prefix vocab: <http:///example.org/vocab/>.
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
ex:stud4589 vocab:studiesIn ex:univ3;
            foaf:name ‘‘Dimitris Spanos’’.
ex:univ3 vocab:institutionName ‘‘Εθνικό Μετσόβιο Πολυτεχνείο’’@el;
         vocab:foundationDate ‘‘January 21, 1837’’.
```

Μια επέκταση του RDF μοντέλου που έχει ως στόχο την ομαδοποίηση και καλύτερη οργάνωση των RDF τριάδων είναι αυτή των *ονομαστικών γράφων* (named graphs) [52]. Ένας ονομαστικός γράφος είναι ένα ζεύγος (u,g), όπου u ένα URI και g ένα σύνολο τριάδων. Με άλλα λόγια, ένας ονομαστικός γράφος αναθέτει ένα μοναδικό αναγνωριστικό σε ένα σύνολο RDF τριάδων, επιτρέποντας μεταξύ άλλων και την καταγραφή της προέλευσής τους. Για την αναπαράσταση ονομαστικών γράφων, έχουν προταθεί διάφορες μορφές σειριακοποίησης, όπως οι TriX [53], N-Quads[3] και TriG[4], οι οποίες επεκτείνουν προαναφερθείσες μορφές αναπαράστασης τριάδων με ένα τέταρτο στοιχείο που αναφέρεται στο URI του ονομαστικού γράφου στον οποίο ανήκει μια τριάδα. Ενδεικτικά, η N-Quads σύνταξη προσθέτει στο τέλος κάθε τριάδας το URI ενός γράφου, δημιουργώντας έτσι μια *τετράδα* (quad):

```
@prefix ex: <http://example.org/resource/>.
@prefix vocab: <http:///example.org/vocab/>.
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
ex:stud4589 vocab:studiesIn ex:univ3 ex:graph1.
ex:stud4589 foaf:name ‘‘Dimitris Spanos’’ ex:graph1.
```

2.2.2 Γλώσσες αναπαράστασης γνώσης

Το RDF μοντέλο παρέχει τη δυνατότητα έκφρασης απλών δηλώσεων για κάποιο αντικείμενο ή έννοια, με τη χρήση προσδιορισμένων ιδιοτήτων και τι-

[3]Η N-Quads σύνταξη ορίζεται στο: http://sw.deri.org/2008/07/n-quads/.

[4]Η TriG σύνταξη ορίζεται στο: http://wifo5-03.informatik.uni-mannheim.de/bizer/trig/.

μών δεδομένων. Αυτό από μόνο του βέβαια δεν αρκεί για την περιγραφή και μοντελοποίηση ενός γνωστικού πεδίου, καθώς είναι απαραίτητος όχι μόνο ο ορισμός μεμονωμένων αντικειμένων, αλλά και κλάσεων οντοτήτων με κοινά χαρακτηριστικά. Το **RDF Schema (RDFS)** συμπληρώνει κατ' αυτή την έννοια το RDF μοντέλο παρέχοντας ένα λεξιλόγιο που επιτρέπει τον ορισμό και την περιγραφή τέτοιων κλάσεων, καθώς και την περιγραφή των ιδιοτήτων που συνδέουν RDF πόρους. Με άλλα λόγια, το RDF Schema παρέχει τους απαραίτητους μηχανισμούς για τον ορισμό ορολογικής γνώσης (terminological knowledge) και λεξιλογίων συγκεκριμένης θεματολογίας.

Οι πλέον σημαντικοί από τους μηχανισμούς αυτούς, οι οποίοι δηλώνονται μέσω αντίστοιχων ειδικών RDF κόμβων και ιδιοτήτων, είναι:

- **Ορισμός κλάσεων.** Οι πόροι ενός RDF μοντέλου μπορούν να ομαδοποιηθούν σε κλάσεις μέσω της ιδιότητας `rdf:type`. Οι πόροι που ανήκουν σε μια κλάση ονομάζονται και *στιγμιότυπα* (instances) αυτής της κλάσης. Μια RDFS κλάση αποτελεί έναν ειδικό πόρο, ο οποίος με τη σειρά του ανήκει στη (μετα-)κλάση `rdfs:Class`.

- **Ορισμός ιεραρχίας κλάσεων.** Μια RDFS κλάση C_1 μπορεί να οριστεί ως υποκλάση μιας άλλης κλάσης C_2 μέσω της ιδιότητας `rdfs:subClassOf`. Αυτό σημαίνει ότι όλα τα στιγμιότυπα της C_1 θα αποτελούν και στιγμιότυπα της κλάσης C_2, χωρίς να χρειάζεται να δηλωθεί κατηγορηματικά η συμμετοχή τους σε αυτή μέσω της ιδιότητας `rdf:type`.

- **Ορισμός ιδιοτήτων.** Όπως αναφέρθηκε και στην περιγραφή του RDF μοντέλου, οι ιδιότητες αποτελούν ειδικούς πόρους που εμφανίζονται στη θέση του κατηγορήματος σε RDF τριάδες και συνδέουν δύο πόρους μεταξύ τους ή έναν πόρο με μια τιμή δεδομένων. Κάθε RDF ιδιότητα αποτελεί στιγμιότυπο της κλάσης `rdf:Property`.

- **Ορισμός ιεραρχίας ιδιοτήτων.** Με τρόπο αντίστοιχο με αυτόν των RDFS κλάσεων, το RDFS επιτρέπει τον ορισμό ιεραρχιών ιδιοτήτων, μέσω της ειδικής ιδιότητας `rdfs:subPropertyOf`. Αν μια ιδιότητα P_1 οριστεί ως υπο-ιδιότητα μιας ιδιότητας P_2, αυτό σημαίνει ότι όλα τα ζεύγη πόρων που συνδέονται μέσω της P_1 θα συσχετίζονται και μέσω της P_2, χωρίς να χρειάζεται να δηλωθούν κατηγορηματικά οι αντίστοιχες RDF προτάσεις.

- **Ορισμός περιορισμών ιδιοτήτων.** Το πεδίο ορισμού και πεδίο τιμών μιας ιδιότητας μπορεί να οριστεί στο RDFS μέσω των ιδιοτήτων `rdfs:domain` και `rdfs:range` αντίστοιχα. Αυτό σημαίνει ότι, αν μια κλάση C δηλωθεί ως πεδίο ορισμού μιας ιδιότητας P, κάθε πόρος που βρίσκεται στη θέση του *υποκειμένου* σε μια RDF πρόταση με κατηγόρημα την P θα είναι στιγμιότυπο της C. Αντίστοιχα, αν μια κλάση D δηλωθεί ως πεδίο τιμών της P, αυτό συνεπάγεται ότι κάθε πόρος που συμμετέχει ως *αντικείμενο* σε μια τριάδα με κατηγόρημα την P θα είναι στιγμιότυπο της D.

Οι προαναφερθέντες μηχανισμοί σε συνδυασμό και με τις υπόλοιπες δυνατότητες του RDF Schema που αναφέρονται αναλυτικά στην επίσημη προδιαγραφή του [46], το καθιστούν μια βασική και πολύ απλή *γλώσσα αναπαράστασης γνώσης* ή ισοδύναμα, μια *γλώσσα ορισμού οντολογιών*.

Ο όρος **οντολογία** (ontology) συνιστά μια από τις βασικές έννοιες όχι μόνο του Σημασιολογικού Ιστού αλλά και της επιστήμης της πληροφορίας γενικότερα. Η προέλευση του όρου βρίσκεται στη φιλοσοφία, όπου δηλώνει το πεδίο που ασχολείται με τη μελέτη του *είναι* και των διαφόρων όντων, αισθητών και μη, νοητών και φανταστικών. Στο πλαίσιο της επιστήμης της πληροφορίας, έχει δοθεί πληθώρα ορισμών του όρου οντολογία: από τον πλέον γνωστό και με πλήθος αναφορών «Οντολογία είναι μια ρητή προδιαγραφή μιας εννοιολογικής σύλληψης» του Tom Gruber [91] και τον ορισμό του Nicola Guarino «Οντολογία είναι μια λογική θεωρία που θεμελιώνει το νόημα ενός τυπικού λεξιλογίου» [92] έως πιο αυστηρούς μαθηματικούς ορισμούς [70, 110]. Πιο απλά, θα μπορούσαμε να πούμε ότι μια οντολογία απλά ορίζει τους όρους που χρησιμοποιούνται για τον ορισμό και την περιγραφή μιας γνωστικής περιοχής, ή ακόμα πιο συγκεκριμένα, δίνουμε μια προσαρμογή του αρκετά γενικού ορισμού των Maedche και Staab [136].

Ορισμός 2.2.1. (Οντολογία) Μια οντολογία είναι ένα σύνολο δομικών στοιχείων που περιλαμβάνει:

α) ένα σύνολο *εννοιών*,
β) ένα σύνολο *συσχετίσεων* μεταξύ εννοιών, συνοδευόμενο από περιορισμούς για το πεδίο ορισμού και το πεδίο τιμών τους,
γ) μια *ταξινομία εννοιών* (ισοδύναμα, ιεραρχία) με δυνατότητα πολλαπλής κληρονομικότητας,
δ) μια *ταξινομία συσχετίσεων* με δυνατότητα πολλαπλής κληρονομικότητας,
ε) ένα *σύνολο αξιωμάτων* που θέτει επιπρόσθετους περιορισμούς, οι οποίοι επιτρέπουν τη συνεπαγωγή νέων γεγονότων από γεγονότα που έχουν δηλωθεί κατηγορηματικά στην οντολογία.

Οι οντολογίες μπορούν να διακριθούν μεταξύ τους με βάση το αντικείμενο της εννοιολογικής σύλληψης που περιγράφουν. Μεταξύ άλλων, συναντώνται οντολογίες [89]:

- *αναπαράστασης γνώσης* (knowledge representation), οι οποίες συγκεντρώνουν τους βασικούς μηχανισμούς που χρησιμοποιούνται σε μια γλώσσα αναπαράστασης γνώσης, όπως το RDF Schema,
- *ανώτερου επιπέδου* (top-level ή upper-level), που περιγράφουν πολύ γενικές έννοιες, κοινές για τις περισσότερες θεματικές περιοχές,
- *πεδίου* (domain), οι οποίες είναι οι πλέον συνηθισμένες και αποτελούν λεξιλόγια όρων και συσχετίσεων για μια συγκεκριμένη θεματική περιοχή,
- *εφαρμογής*, οι οποίες δημιουργούνται στα πλαίσια συγκεκριμένων εφαρμογών ή πειραματικών μελετών περιπτώσεων.

Με βάση τα παραπάνω, το RDF Schema δίνει τη δυνατότητα μοντελοποίησης οντολογιών που ικανοποιούν τις ελάχιστες προϋποθέσεις του ορισμού 2.2.1. Εντούτοις, η εκφραστικότητα που παρέχει το RDF Schema είναι αρκετά μικρή και ανεπαρκής για την αναπαράσταση σύνθετης γνώσης, όπως συνήθως απαιτείται στην πράξη. Παραδείγματος χάριν, οι προτάσεις *«Ένα τμήμα έχει τουλάχιστον 4 εργαζομένους»* και *«Ένας εργαζόμενος μερικής*

απασχόλησης δεν μπορεί να γίνει διευθυντής τμήματος» δεν μπορούν να μοντελοποιηθούν σε μια RDFS οντολογία. Για την αναπαράσταση σύνθετης γνώσης, χρησιμοποιούνται εκφραστικές γλώσσες αναπαράστασης βασισμένες σε τυπική λογική, που επιτρέπουν διεργασίες συλλογισμού (reasoning) και συνεπώς, την πρόσβαση σε γνώση που υπονοείται από τα κατηγορηματικά εκφρασμένα γεγονότα.

Η **OWL** (Web Ontology Language) είναι μια τέτοια γλώσσα, έχει προταθεί από το W3C και αποτελεί την πλέον δημοφιλή επιλογή για μοντελοποίηση οντολογιών σε διάφορους τομείς εφαρμογών. Η πιο πρόσφατη έκδοσή της, γνωστή ως OWL 2, προτάθηκε το 2009 [143], διαθέτει επιπρόσθετες δυνατότητες μοντελοποίησης σε σχέση με την προκάτοχό της και ορίζει 3 νέα συντακτικά υποσύνολά της (profiles) με υπολογιστικά πλεονεκτήματα για ξεχωριστά είδη εφαρμογών: την OWL EL για οντολογίες με μακρές ιεραρχίες κλάσεων και ιδιοτήτων, την OWL QL για την υλοποίηση βάσεων γνώσης με μεγάλο αριθμό οντολογικών ατόμων χρησιμοποιώντας σχεσιακά ΣΔΒΔ και την OWL RL για την υλοποίηση βάσεων γνώσης και πραγματοποίηση συλλογισμού με συστήματα κανόνων. Τα εν λόγω υποσύνολα αντικαθιστούν τα αντίστοιχα γλωσσικά υποσύνολα OWL Lite, OWL DL και OWL Full που όριζε η πρώτη έκδοση της OWL.

Τα βασικά δομικά στοιχεία που χρησιμοποιεί η OWL για την αναπαράσταση γνώσης είναι τα εξής:

α) *ατομικές οντότητες* (entities), δηλαδή κλάσεις, ιδιότητες και στιγμιότυπα κλάσεων (τα οποία είναι γνωστά και ως *άτομα* στην ορολογία της OWL). Αυτές οι οντότητες προσδιορίζονται από κάποιο IRI (Internationalized Resource Identifier)[5] και αποτελούν τους πρωταρχικούς όρους μιας οντολογίας, στους οποίους βασίζονται σύνθετες εκφράσεις και αξιώματα.
β) *σύνθετες εκφράσεις* (expressions) που χρησιμοποιούν ατομικούς όρους και μπορούν με τη σειρά τους να χρησιμοποιηθούν σε άλλες σύνθετες εκφράσεις ή αξιώματα. Ένα παράδειγμα σύνθετης έκφρασης είναι ο ορισμός μιας νέας κλάσης ως τομή δύο ατομικών κλάσεων.
γ) *αξιώματα* (axioms), δηλαδή προτάσεις ή δηλώσεις που είναι πάντα αληθείς για τον τομέα γνώσης που μοντελοποιεί μια οντολογία.

Κάθε OWL οντολογία μπορεί να εκφραστεί και ως RDF γράφος, οπότε για την αναπαράστασή της μπορούν να χρησιμοποιηθούν οι συντακτικές μορφές που αναφέρθηκαν στην παράγραφο 2.2.1, ενώ έχουν οριστεί και τρεις επιπλέον συντακτικές μορφές ειδικά για τη γλώσσα OWL. Ενδεικτικά, μερικές μόνο από τις δυνατότητες της OWL είναι:

- ορισμός κλάσεων, ιδιοτήτων και ατόμων κλάσεων
- ορισμός ιεραρχίας κλάσεων και ιδιοτήτων
- καθορισμός πεδίου ορισμού και πεδίου τιμών ιδιοτήτων
- διάκριση μεταξύ ιδιοτήτων που συσχετίζουν άτομα (object properties ή ιδιότητες αντικειμένου), ιδιοτήτων που συσχετίζουν ένα άτομο με μια

[5]Τα IRI αναγνωριστικά αποτελούν γενίκευση των URIs, επιτρέποντας και Unicode χαρακτήρες.

τιμή (datatype properties ή ιδιότητες τύπου δεδομένων) και ιδιοτήτων που τεκμηριώνουν κάποιο στοιχείο της οντολογίας (annotation properties ή ιδιότητες επισημείωσης)

- ορισμός αξιωμάτων μη επικάλυψης κλάσεων (class disjointness)
- ορισμός αξιωμάτων ισότητας και ανισότητας ατόμων
- ορισμός σύνθετων εκφράσεων που περιλαμβάνουν τις λογικές πράξεις της ένωσης (union), τομής (intersection) και συμπληρώματος (complement) μεταξύ κλάσεων
- ορισμός σύνθετων εκφράσεων με εφαρμογή περιορισμών συμμετοχής σε δεδομένες ιδιότητες, χρησιμοποιώντας υπαρξιακή (existential quantification) και καθολική ποσοτικοποίηση (universal quantification), καθώς και συγκεκριμένη πληθικότητα συμμετοχής (cardinality restriction)
- ορισμός κλάσης μέσω απαρίθμησης των ατόμων της
- προσδιορισμός χαρακτηριστικών μιας ιδιότητας, η οποία μπορεί να είναι μεταβατική (transitive), συμμετρική (symmetric), ασύμμετρη (asymmetric), κλειδί (key), συναρτησιακή (functional), ανάστροφη (inverse) μιας άλλης, αυτοπαθής (reflexive) ή μη (irreflexive)
- ορισμός αλυσίδων ιδιοτήτων (property chains)

Η OWL βασίζεται σε μια οικογένεια φορμαλισμών αναπαράστασης γνώσης, γνωστών ως **Περιγραφικές Λογικές** (Description Logics), κάθε μία εκ των οποίων έχει και διαφορετικό βαθμό εκφραστικότητας. Οι Περιγραφικές Λογικές επιτρέπουν την περιγραφή ενός γνωσιακού τομέα, προσδιορίζοντας πρώτα τις έννοιες που ενυπάρχουν σε αυτόν και στη συνέχεια, χρησιμοποιώντας αυτές τις έννοιες για να οριστούν συγκεκριμένα αντικείμενα και ιδιότητές τους [19]. Σε συστήματα αναπαράστασης γνώσης όπου χρησιμοποιούνται Περιγραφικές Λογικές, είναι συνηθισμένη η διάκριση μεταξύ αξιωμάτων που ορίζουν την ορολογία του τομέα εφαρμογής και αποτελούν το λεγόμενο *σώμα ορολογίας* (Terminology Box ή TBox) και αξιωμάτων που αναφέρονται σε συγκεκριμένα άτομα ή στιγμιότυπα, αποτελώντας το *σώμα ισχυρισμών* (Assertional Box ή ABox). Ο συνδυασμός των δύο αυτών σωμάτων με διαδικασίες συλλογισμού δημιουργούν μια *βάση γνώσης* (knowledge base).

Δύο χαρακτηριστικά ως προς τα οποία διαφέρουν οι Περιγραφικές Λογικές με τις περισσότερες μοντέλα αναπαράστασης δεδομένων (μεταξύ αυτών και με το σχεσιακό και το μοντέλο ΟΣ) είναι η έλλειψη της *Υπόθεσης Μοναδικών Ονομάτων* (Unique Name Assumption ή UNA) και της *Υπόθεσης Κλειστού Κόσμου* (Closed World Assumption ή CWA). Η έλλειψη της πρώτη υπόθεσης στις Περιγραφικές Λογικές συνεπάγεται ότι άτομα με διαφορετικό όνομα δεν είναι απαραίτητα διαφορετικά άτομα, καθώς μπορεί να αποδειχθεί ότι πρόκειται για το ίδιο άτομο. Όσον αφορά στην έλλειψη της δεύτερης υπόθεσης ή ισοδύναμα, στην υιοθέτηση της Υπόθεσης Ανοικτού Κόσμου (Open World Assumption ή OWA), αυτή δηλώνει ότι μια βάση γνώσης δεν μπορεί να περιέχει πλήρη γνώση και συνεπώς, η απουσία ενός γεγονότος από αυτή δεν ισοδυναμεί με τη μη ισχύ του γεγονότος αυτού (καθώς αυτό μπορεί να ισχύει, αλλά να μην είναι γνωστό στη βάση γνώσης). Αυτή είναι και μια σημαντική διαφορά με

το σχεσιακό μοντέλο και στα συστήματα ΒΔ γενικότερα, τα οποία υιοθετούν την Υπόθεση Κλειστού Κόσμου, σύμφωνα με την οποία αν ένα γεγονός δεν περιέχεται στη βάση, τότε αυτό δεν είναι αληθές.

Ο **συλλογισμός** (reasoning) σε μια βάση γνώσης είναι μια διαδικασία που περιλαμβάνει ένα σύνολο αλγορίθμων, οι οποίοι εξάγουν νέα γνώση από κατηγορηματικά δηλωθέντα αξιώματα. Οι βασικές υπηρεσίες συλλογισμού που μπορούν να εφαρμοστούν σε μια βάση γνώσης είναι οι ακόλουθες:

- έλεγχος **ικανοποιησιμότητας** (satisfiability) μιας έννοιας. Μια έννοια είναι ικανοποιήσιμη, αν μπορεί να έχει στιγμιότυπα που ικανοποιούν το σώμα ορολογίας της βάσης γνώσης.
- έλεγχος **υπαγωγής** (subsumption). Μια έννοια C υπάγεται στην έννοια D, αν κάθε στιγμιότυπο της C είναι υποχρεωτικά και στιγμιότυπο της D.
- έλεγχος **ισοδυναμίας** (equivalence) δύο εννοιών
- έλεγχος **μη επικάλυψης** (disjointness) δύο εννοιών
- έλεγχος **συνέπειας** (consistency) ενός σώματος ισχυρισμών με βάση ένα σώμα ορολογίας
- έλεγχος **συνεπαγωγής** (entailment) ενός ισχυρισμού από ένα σώμα ισχυρισμών
- **ανάκτηση** (retrieval) των ατόμων ενός σώματος ισχυρισμών που αποτελούν στιγμιότυπα μιας δεδομένης έννοιας

Το στοιχείο που κυρίως διαμορφώνει το βαθμό υιοθέτησης μιας γλώσσας Περιγραφικής Λογικής στην πράξη, αλλά και οδηγεί στην ανάπτυξη νέων γλωσσών είναι η πολυπλοκότητα των αλγορίθμων συλλογισμού για τη συγκεκριμένη γλώσσα. Δεδομένης της σχέσης ανταλλαγής που υπάρχει μεταξύ της εκφραστικότητας μιας γλώσσας Περιγραφικής Λογικής και της υπολογιστικής πολυπλοκότητας υπηρεσιών συλλογισμού, όπως αυτών που προαναφέρθηκαν, ένα σημαντικό κομμάτι έρευνας σε αυτό το πεδίο αποσκοπεί στη δημιουργία εκφραστικών γλωσσών που κρατούν την πολυπλοκότητα των αλγορίθμων συλλογισμού εντός αποδεκτών ορίων.

Η σημασία μιας OWL οντολογίας μπορεί να οριστεί με δύο τρόπους: είτε ως ένας RDF γράφος, όπου δεν ισχύει ο διαχωρισμός μεταξύ κλάσεων και ατόμων, είτε ακολουθώντας τη θεωρία των Περιγραφικών Λογικών. Αυτές οι δύο οπτικές ορίζουν δύο διαφορετικές μεταξύ τους *σημασιολογίες* για μια OWL οντολογία: την RDF σημασιολογία, γνωστή και ως OWL Full και την άμεση μοντελο-θεωρητική σημασιολογία, γνωστή και ως OWL DL.

2.2.3 Η γλώσσα ερωτημάτων SPARQL

Εξίσου σημαντική με την περιγραφή δομημένης πληροφορίας, είναι και η δυνατότητα αναζήτησης σε αυτή, γεγονός που καθιστά την SPARQL (SPARQL Protocol and RDF Query Language) [93] μια από τις θεμελιώδεις τεχνολογίες του Σημασιολογικού Ιστού. Η SPARQL είναι μια εκφραστική γλώσσα ερωτημάτων για RDF γράφους, με αρκετές από τις δυνατότητες που διαθέτει και η

SQL, αποτελεί πρότυπο του W3C από το 2008 και στην παρούσα φάση, βρίσκεται σε εξέλιξη η προτυποποίηση της δεύτερης έκδοσής της, γνωστής και ως SPARQL 1.1. Η οικογένεια εγγράφων που απαρτίζουν το πρότυπο SPARQL περιλαμβάνουν επίσης μια γλώσσα για την τροποποίηση RDF γράφων (SPARQL Update), ένα πρωτόκολλο για τη μετάδοση ερωτημάτων και των αποκρίσεων σε αυτά που βασίζεται στο HTTP, καθώς και μια σειρά μορφοτύπων για την αναπαράσταση των αποτελεσμάτων ενός SPARQL ερωτήματος.

Η SPARQL χρησιμοποιεί σύνολα από *πρότυπα τριάδων* (triple patterns), τα οποία δημιουργούν *πρότυπα βασικών γράφων* (basic graph patterns). Ένα πρότυπο τριάδας είναι ουσιαστικά μια RDF τριάδα, όπου οποιοδήποτε από τα συστατικά της μπορεί να είναι μια μεταβλητή. Ένα πρότυπο βασικού γράφου *ταιριάζει* με έναν υπογράφο G του συνολικού RDF γράφου, αν η αντικατάσταση των μεταβλητών του πρώτου από όρους του δεύτερου οδηγούν σε RDF γράφο ισοδύναμο του G. Στην πιο απλή μορφή του, ένα SPARQL ερώτημα αποτελείται από μια `WHERE` πρόταση, η οποία περιέχει ένα πρότυπο βασικού γράφου και μια `SELECT` πρόταση που περιέχει τις μεταβλητές που θα περιλαμβάνονται στο αποτέλεσμα. Το αποτέλεσμα ενός SPARQL ερωτήματος περιέχει τους RDF όρους από όλους τους υπογράφους του RDF συνόλου δεδομένων, οι οποίοι ταιριάζουν με το πρότυπο βασικού γράφου της `WHERE` πρότασης. Αν, για παράδειγμα, θεωρήσουμε το παρακάτω σύνολο RDF προτάσεων:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
_:a foaf:name ‘‘Alice’’ .
_:a foaf:mbox <mailto:alice@example.com> .
_:b foaf:name ‘‘Bob’’ .
_:b foaf:mbox <mailto:bob@example.org> .
_:c foaf:mbox <mailto:carol@example.org> .
```

και θέσουμε το επόμενο απλό SPARQL ερώτημα το οποίο αναζητά όλους τους RDF όρους που έχουν τιμές για τις ιδιότητες `foaf:name` και `foaf:mbox`:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
SELECT ?name ?mbox
WHERE
    { ?x foaf:name ?name.
      ?x foaf:mbox ?mbox}
```

θα λάβουμε το ακόλουθο αποτέλεσμα:

name	mbox
‘‘Alice’’	<mailto:alice@example.com>
‘‘Bob’’	<mailto:bob@example.org>

Μερικές από τις δυνατότητες που προσφέρει η SPARQL περιλαμβάνουν προαιρετικό ταίριασμα προτύπων, ταίριασμα εναλλακτικών προτύπων, συνθήκες φιλτραρίσματος, άρνηση, εμφωλευμένα ερωτήματα, συναθροιστικές συναρτήσεις και τελεστές διάταξης και σύγκρισης. Παράλληλα, ένα SPARQL ερώτημα, εκτός από έναν πίνακα με τιμές για κάθε μεταβλητή, μπορεί να επιστρέψει και έναν RDF γράφο (`CONSTRUCT` και `DESCRIBE` ερωτήματα) ή ακόμα και

μια boolean τιμή (ASK ερωτήματα). Μια υπηρεσία ιστού η οποία ακολουθεί το SPARQL πρωτόκολλο, επιτρέποντας σε χρήστες να θέσουν ερωτήματα στην υποκείμενη βάση γνώσης, είναι γνωστή και ως *τελικό σημείο SPARQL* (SPARQL endpoint).

Κάθε SPARQL ερώτημα μπορεί να εκφραστεί σε μια ισοδύναμη μορφή, η οποία χρησιμοποιεί τελεστές της SPARQL άλγεβρας [93]. Η SPARQL άλγεβρα αποτελεί τη βάση της γλώσσας SPARQL, καθώς βάσει αυτής ορίζεται η σημασιολογία ενός SPARQL ερωτήματος. Οι απαραίτητοι μετασχηματισμοί που πραγματοποιεί κάθε μηχανή εκτέλεσης SPARQL ερωτημάτων εφαρμόζονται στις ισοδύναμες αλγεβρικές εκφράσεις των εισερχόμενων ερωτημάτων. Ενδεικτικά, το προηγούμενο SPARQL ερώτημα εκφράζεται στην ακόλουθη αλγεβρική μορφή:

```
(project(?name ?mbox)
(bgp(triple(?x foaf:name ?name), triple(?x foaf:mbox ?mbox))))
```

όπου project ο τελεστής της προβολής μεταβλητών και bgp ο τελεστής του προτύπου βασικού γράφου.

2.2.4 R2RML: Γλώσσα αντιστοιχίας σχεσιακών βάσεων δεδομένων με RDF γράφους

Η δυνατότητα συνεργασίας του σχεσιακού με το RDF μοντέλο έχει εξεταστεί τα τελευταία χρόνια από πλήθος ερευνητών, στο πλαίσιο διαφόρων εφαρμογών. Όπως θα δούμε αναλυτικά και στο κεφάλαιο 3, οι περισσότερες από αυτές τις προσπάθειες χρειάζονται να αναπαριστούν σε κάποια μορφή συσχετίσεις μεταξύ του σχεσιακού και του RDF μοντέλου. Η υιοθέτηση μιας κοινής γλώσσας αναπαράστασης αυτών των αντιστοιχιών αποτελεί τη λύση σε προβλήματα διαλειτουργικότητας, ενώ διευκολύνει και τους χρήστες τέτοιων αντιστοιχιών. Η R2RML (RDB to RDF Mapping Language) [69] προτυποποιήθηκε πρόσφατα (Σεπτέμβριος 2012) από το W3C και φιλοδοξεί να καλύψει το κενό που υπήρχε μέχρι σήμερα στις σχετικές υλοποιήσεις. Η R2RML, ως μια πρότυπη γλώσσα αναπαράστασης που είναι ανεξάρτητη του εκάστοτε χρησιμοποιούμενου ΣΔΒΔ, διευκολύνει τη μετάβαση από ένα ΣΔΒΔ σε ένα άλλο χωρίς να χρειάζεται η επανεγγραφή των ήδη ορισμένων αντιστοιχιών και επιτρέπει την εφαρμογή μιας αντιστοιχίας σε περισσότερες ΒΔ διαφορετικών κατασκευαστών που χρησιμοποιούν το ίδιο σχήμα. Παράλληλα, αποτρέπει τον εγκλωβισμό των χρηστών στο λογισμικό το οποίο χρησιμοποίησαν για τον ορισμό της αντιστοιχίας, αφού πλέον αντιστοιχίες μεταξύ σχεσιακών ΒΔ και RDF γράφων θα μπορούν να μεταφέρονται και να επαναχρησιμοποιούνται από εργαλεία που την υποστηρίζουν. Εν ολίγοις, η R2RML αναμένεται να αναλάβει το ρόλο που διαδραμάτισε και η SQL, κατ' αναλογία, στο χώρο των ΣΔΒΔ και ο οποίος ήταν καθοριστικός στη διάδοση και επιτυχία των τελευταίων.

Εν συντομία, η R2RML θεωρεί *λογικούς πίνακες* (logical tables), οι οποίοι μπορεί να είναι πίνακες ή όψεις μιας σχεσιακής ΒΔ ή ακόμα και αυθαίρετα SQL ερωτήματα που επίσης ορίζουν μια όψη αλλά δεν αποθηκεύονται στο σχήμα της ΒΔ. Κάθε γραμμή αυτών των λογικών πινάκων αντιστοιχεί σε μία η περισσότερες RDF προτάσεις, μέσω της βασικής δομής της R2RML, της *αντιστοιχίας τριάδων* (triples map). Κάθε αντιστοιχία τριάδων αποτελείται από

μια *αντιστοιχία υποκειμένου* (subject map) και μία ή περισσότερες *αντιστοιχίες κατηγορήματος-αντικειμένου* (predicate-object map). Οι RDF τριάδες δημιουργούνται συνδυάζοντας, για κάθε γραμμή ενός λογικού πίνακα, το υποκείμενο που προκύπτει από την αντιστοιχία υποκειμένου με κάθε δυνατό ζεύγος κατηγορήματος-αντικειμένου που προκύπτει από τις αντιστοιχίες κατηγορήματος και αντικειμένου. Παράλληλα, δίνεται η δυνατότητα τοποθέτησης των παραγόμενων RDF τριάδων σε έναν ή περισσότερους ονομαστικούς γράφους (named graphs) μέσω του μηχανισμού *αντιστοιχίας γράφου* (graph map).

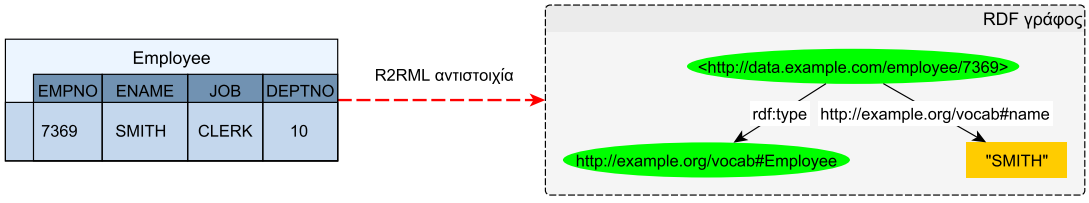
Όλοι οι μηχανισμοί αντιστοιχίας που αναφέρθηκαν μπορούν να προσδιορίσουν ένα IRI ή λεκτικό, που μπορεί να: α) είναι σταθερό, β) παίρνει την τιμή από τη στήλη ενός λογικού πίνακα ή γ) προέρχεται από ένα πρότυπο (template) που συνδυάζει σταθερές τιμές με τιμές από μία ή περισσότερες στήλες ενός λογικού πίνακα. Παράλληλα, η R2RML επιτρέπει:

- ορισμό κενών κόμβων
- ορισμό γλώσσας και τύπου δεδομένων για παραγόμενα λεκτικά
- ορισμό αντιστοιχίας για ξένα κλειδιά
- ορισμό κλάσης της οποίας αποτελεί στιγμιότυπο ένα παραγόμενο IRI

Η R2RML χρησιμοποιεί το RDF μοντέλο για την αναπαράσταση μιας αντιστοιχίας, υιοθετώντας ως προτιμητέα σύνταξη την Turtle. Ένα απλό παράδειγμα ενός R2RML εγγράφου είναι το παρακάτω:

```
@prefix rr: <http://www.w3.org/ns/r2rml#>.
@prefix ex: <http://example.org/vocab#>.
<#TriplesMap1>
     rr:logicalTable [rr:tableName ‘‘Employee’’];
     rr:subjectMap[
          rr:template ‘‘http://data.example.com/employee/EMPNO’’;
          rr:class ex:Employee;
          ];
     rr:predicateObjectMap[
          rr:predicate ex:name;
          rr:objectMap [rr:column ‘‘ENAME’’];
     ].
```

το οποίο, όταν εφαρμοστεί στον πίνακα του σχήματος 2.3, οδηγεί στη δημιουργία του RDF γράφου που απεικονίζεται στο ίδιο σχήμα.



Σχήμα 2.3: *Παράδειγμα εφαρμογής R2RML αντιστοιχίας*

Παράλληλα, ως συνοδευτικό έγγραφο της R2RML, προτάθηκε από το W3C και μια Άμεση Αντιστοιχία (Direct Mapping) μεταξύ σχεσιακών ΒΔ και RDF γράφων [14], η οποία ορίζει έναν απλό, προκαθορισμένο μετασχηματισμό που μπορεί να χρησιμεύσει ως σημείο έναρξης για τον ορισμό μιας πιο πολύπλοκης R2RML αντιστοιχίας. Η ιδέα αυτού του μετασχηματισμού ταυτίζεται με τη φιλοσοφία της βασικής προσέγγισης αντιστοιχίας ΒΔ με RDF, η οποία περιγράφεται στην ενότητα 3.2.

2.2.5 Συνδεδεμένα Δεδομένα

Ο Παγκόσμιος Ιστός είναι ακόμα και σήμερα, σε μεγάλο βαθμό, ένας ιστός εγγράφων που οφείλει την επιτυχία και ανάπτυξή του στην ευκολία δημοσίευσης εγγράφων και στην ανακάλυψη αυτών μέσω υπερσυνδέσμων που συνδέουν το ένα με το άλλο. Μέχρι πρόσφατα, αυτές οι βασικές αρχές δεν εφαρμόζονταν και σε δεδομένα, τα οποία δημοσιεύονταν στον Ιστό κυρίως ως μαζικές συλλογές σε ανεπεξέργαστη CSV ή XML μορφή. Ο όρος Συνδεδεμένα Δεδομένα (Linked Data) αναφέρεται σε ένα σύνολο βέλτιστων πρακτικών για τη δημοσίευση και σύνδεση δομημένων δεδομένων στον Παγκόσμιο Ιστό [95]. Οι πρακτικές αυτές συνεισφέρουν στην προσπάθεια μετατροπής του ιστού εγγράφων σε έναν ιστό δεδομένων, όπου μεμονωμένες οντότητες και δεδομένα συνδέονται μεταξύ τους, βασιζόμενες στην υπάρχουσα αρχιτεκτονική Ιστού και στις τεχνολογίες αυτού. Η ύπαρξη συνδέσμων μεταξύ δεδομένων που αναφέρονται σε ποικίλους τομείς ενδιαφέροντος επιτρέπει όχι μόνο την πλοήγηση του τελικού χρήστη σε σύνολα δεδομένων διαφόρων θεματολογιών, αλλά καθιστά δυνατή την επερώτηση, ανακάλυψη και αναζήτηση αυτών από ειδικό λογισμικό.

Οι βασικές αρχές δημοσίευσης Συνδεδεμένων Δεδομένων αναφέρονται από τον Tim Berners-Lee στο [37] και είναι οι ακόλουθες:

1. Χρήση URIs (ή IRIs) για την ονομασία οντοτήτων
2. Πρόσβαση σε αυτά τα URIs μέσω HTTP
3. Παροχή χρήσιμης πληροφορίας μέσω κατάλληλων προτύπων (RDF, SPARQL) όταν κάποιος προσπελαύνει ένα URI
4. Ορισμός συνδέσμων προς άλλα URIs, ώστε ο χρήστης να μπορεί να ανακαλύψει περισσότερες οντότητες

Βασικά συστατικά υλοποίησης των παραπάνω αρχών είναι το πρωτόκολλο HTTP ως ένας ενιαίος παγκόσμιος μηχανισμός πρόσβασης και το μοντέλο RDF (παράγραφος 2.2.1) ως ενιαίο μοντέλο για τη δόμηση και σύνδεση δεδομένων. Οι συγκεκριμένες αρχές αποτελούν τη βάση για την ανάπτυξη ενός παγκόσμιου χώρου δεδομένων και ανοίγουν το δρόμο για την ανάπτυξη γενικών εφαρμογών που θα λειτουργούν εντός αυτού του χώρου, δίχως να χρειάζεται να λάβουν υπόψη τους ετερογενείς μεθόδους πρόσβασης, ξεχωριστές για κάθε πηγή δεδομένων.

Βέβαια, κατά αναλογία με την περίπτωση του Παγκόσμιου Ιστού, στην ανάπτυξη και διάδοση του οποίου συνετέλεσαν διάφορες κατηγορίες εργαλείων και εφαρμογών, όπως εργαλεία ανάπτυξης ιστοσελίδων, εφαρμογές πλοήγησης

και μηχανές αναζήτησης, προκειμένου το κίνημα των Συνδεδεμένων Δεδομένων να πετύχει το στόχο του, θα χρειαστεί η επινόηση μεθόδων και η ανάπτυξη των εφαρμογών εκείνων που θα απλοποιήσουν για τους τελικούς χρήστες κατ' αρχήν τη δημοσίευση και κατά δεύτερο λόγο, την αναζήτηση και αξιοποίηση Συνδεδεμένων Δεδομένων. Μια σημαντική υποκατηγορία εργαλείων δημοσίευσης Συνδεδεμένων Δεδομένων είναι αυτά που αντλούν πληροφορία από σχεσιακές ΒΔ και τα οποία παρουσιάζονται μεταξύ άλλων στο κεφάλαιο 3.

2.3 Δίκτυα αισθητήρων και ροές δεδομένων

Στις μέρες μας, τα *ασύρματα δίκτυα αισθητήρων* (wireless sensor networks) παρουσιάζουν έναν ολοένα αυξανόμενο αριθμό εγκαταστάσεων παγκοσμίως και χρησιμοποιούνται σε πλήθος εφαρμογών που περιλαμβάνουν από στρατιωτικές εφαρμογές και εφαρμογές υγείας μέχρι οικιακές και περιβαλλοντικές εφαρμογές. Αυτή η εξάπλωση οφείλεται στην τεχνολογική πρόοδο στο χώρο της ηλεκτρονικής και των ασύρματων επικοινωνιών που οδήγησαν στην αυξημένη διαθεσιμότητα φθηνών, μικρών σε μέγεθος και ενεργειακά αποδοτικών αισθητήριων συσκευών με δυνατότητες ασύρματης επικοινωνίας. Στο υπόλοιπο κείμενο, θα αναφερόμαστε σε τέτοιου είδους συσκευές με τον όρο *αισθητήρες*. Ένας αισθητήρας είναι μια μικροσκοπική ηλεκτρονική συσκευή που διαθέτει υποσυστήματα επεξεργασίας, επικοινωνίας, παροχής ενέργειας και αίσθησης και η οποία μπορεί να μετρά φυσικά μεγέθη, όπως θερμοκρασία, πίεση, ταχύτητα, επίπεδο φωτισμού, θόρυβο και υγρασία.

Τα ασύρματα δίκτυα αισθητήρων αποτελούνται από αυτόνομους αισθητήρες που παρατηρούν το περιβάλλον τους και συνεργάζονται ώστε να μεταφέρουν τα δεδομένα των μετρήσεών τους σε ένα κεντρικό σημείο με μεγαλύτερη επεξεργαστική ισχύ και περισσότερα ενεργειακά αποθέματα. Το κεντρικό αυτό σημείο είναι γνωστό και ως *πύλη* (gateway). Μερικά από τα χαρακτηριστικά που συχνά διαφοροποιούν τα ασύρματα δίκτυα αισθητήρων από άλλα επικοινωνιακά δίκτυα είναι η περιορισμένη ενέργεια των αισθητήρων, η οποία μπορεί να οδηγήσει σε αποτυχία κόμβων του δικτύου, η τεχνική ετερογένεια των κόμβων και ο μεγάλος αριθμός των κόμβων του δικτύου (ο οποίος μερικές φορές μπορεί να είναι της τάξης των μερικών χιλιάδων αισθητήρων). Αυτά τα χαρακτηριστικά θέτουν και αντίστοιχες προκλήσεις όσον αφορά στην εξοικονόμηση ενέργειας του δικτύου, τη διαχείριση των δεδομένων του και την κλιμακωσιμότητά (scalability) του.

Τα δεδομένα που προκύπτουν από ένα δίκτυο αισθητήρων, είτε αυτά είναι παρατηρήσεις είτε περιγραφές του δικτύου, δε χαρακτηρίζονται από την αυστηρή, γνωστή εκ των προτέρων δομή των σχεσιακών δεδομένων, τα οποία είναι γνωστά και με τον όρο *δομημένα* δεδομένα (structured data). Στην περίπτωση των αισθητήριων δεδομένων, δεν υπάρχει ανάλογη ομοιομορφία, καθώς ακόμα και δεδομένα του ίδιου είδους μπορεί να μην έχουν τα ίδια γνωρίσματα, η διάταξη των τελευταίων δεν έχει σημασία, ενώ συχνά κάποιο γνώρισμα μπορεί να λείπει. Τέτοιου είδους δεδομένα ονομάζονται και *ημιδομημένα* (semistructured data) [5], καθώς χαρακτηρίζονται από μια δομή που συχνά υπονοείται και είναι περισσότερο ευέλικτη από ένα τυπικό σχεσιακό σχήμα.

Δύο από τα ζητήματα που σχετίζονται άμεσα με τη διαχείριση και αξιοποίηση των παρατηρήσεων ενός δικτύου αισθητήρων είναι η διαλειτουργικότητα

των κόμβων του καθώς και ο δυναμικός χαρακτήρας των αισθητήριων παρατηρήσεων, οι οποίες σχηματίζουν χρονικές αλληλουχίες ή ροές μετρήσεων. Το θέμα της διαλειτουργικότητας είναι το κύριο αντικείμενο των πρωτοβουλιών του Ιστού Αισθητήρων (Sensor Web) [44] και Σημασιολογικού Ιστού Αισθητήρων (Semantic Sensor Web) [176], οι οποίες παρουσιάζονται στην παράγραφο 2.3.1, ενώ κάποιες εισαγωγικές έννοιες πάνω στο θέμα της διαχείρισης ροών δεδομένων αναφέρονται στην παράγραφο 2.3.2.

2.3.1 Σημασιολογικός Ιστός Αισθητήρων

Η ανάγκη για την εξασφάλιση - κατ' αρχήν συντακτικής - διαλειτουργικότητας διαφορετικών δικτύων αισθητήρων οδήγησε το Open Geospatial Consortium (OGC)[6], στο πλαίσιο της πρωτοβουλίας Sensor Web Enablement (SWE), να προτείνει μια σειρά από πρότυπα για την ανακάλυψη, τον προγραμματισμό και την πρόσβαση σε αισθητήρες καθώς και στις μετρήσεις αυτών. Τα συγκεκριμένα πρότυπα, που περιλαμβάνουν κωδικοποιήσεις ανταλλαγής μηνυμάτων και διεπαφές υπηρεσιών ιστού, φιλοδοξούν να διαδραματίσουν ένα ρόλο ανάλογο με αυτόν της HTML και του HTTP πρωτοκόλλου στα πρώτα βήματα του Παγκόσμιου Ιστού, αποτελώντας με τη σειρά τους τη βάση του λεγόμενου *Ιστού Αισθητήρων* (Sensor Web). Στο συγκεκριμένο πλαίσιο, ο Ιστός Αισθητήρων ορίζεται ως «ένα σύνολο δικτύων αισθητήρων που είναι προσβάσιμα μέσω Ιστού και επιτρέπουν την ανάκτηση μετρήσεων μέσω πρότυπων πρωτοκόλλων και προγραμματιστικών διεπαφών» [44].

Τα πρότυπα στα οποία βασίζεται ο Ιστός Αισθητήρων είναι τα λεξιλόγια:

- Observations & Measurements Schema (O&M) για την κωδικοποίηση παρατηρήσεων και μετρήσεων από έναν αισθητήρα,
- Sensor Model Language (SensorML) για τη λειτουργική περιγραφή ενός αισθητήρα ως συνόλου διαδικασιών (με αναφορά στις εισόδους, εξόδους και παραμέτρους αυτών),
- Transducer Markup Language (TransducerML) για την λεπτομερή περιγραφή των εσωτερικών συστημάτων ενός αισθητήρα σε επίπεδο υλικού

καθώς και οι υπηρεσίες ιστού:

- Sensor Observations Service (SOS) για την αίτηση και ανάκτηση παρατηρήσεων και πληροφοριών από αισθητήρες,
- Sensor Planning Service (SPS) για την αίτηση και ανάκτηση συλλογών από παρατηρήσεις,
- Sensor Alert Service (SAS) για τη δημοσίευση και ενημέρωση για ειδοποιήσεις από αισθητήρες και
- Web Notification Service (WNS) για την παράδοση των μηνυμάτων που προέρχονται από τις υπηρεσίες SPS και SAS,

[6] Το OGC είναι μια διεθνής συνεργασία εταιρειών, κυβερνητικών οργανισμών και πανεπιστημίων με στόχο την καθιέρωση ανοικτών προτύπων για γεωχωρικό περιεχόμενο και υπηρεσίες.

που στοχεύουν κυρίως στο πρόβλημα της συντακτικής διαλειτουργικότητας αισθητήρων διαφορετικών ειδών και κατασκευαστών.

Εντούτοις, η λειτουργικότητα που προσδίδουν τα SWE πρότυπα σε ένα δίκτυο αισθητήρων συχνά δεν καλύπτει τις ανάγκες πραγματικών εφαρμογών, οι οποίες εκτείνονται πέραν απλών λειτουργιών αναζήτησης αισθητήρων και ανάγνωσης μετρήσεων. Σε τέτοιες περιπτώσεις, ανακύπτει η ανάγκη τυπικού ορισμού της σημασίας των αισθητήριων παρατηρήσεων, όπως και των ίδιων των δυνατοτήτων και παραμέτρων των αισθητήριων συσκευών. Η επίτευξη της σημασιολογικής διαλειτουργικότητας αισθητήρων είναι ένας απαιτητικός και φιλόδοξος στόχος που αποτελεί αντικείμενο του οράματος του *Σημασιολογικού Ιστού Αισθητήρων* (Semantic Sensor Web) [176], ο οποίος θα συνεισφέρει στην αυτοματοποιημένη ανακάλυψη και συνεργασία μεταξύ αισθητήριων συσκευών αλλά και στην αυτόματη λήψη αποφάσεων σε επίπεδο δικτύου μέσω εφαρμογής διαδικασιών συλλογισμού. Στη συγκεκριμένη προσπάθεια, οι τεχνολογίες και τα πρότυπα του Σημασιολογικού Ιστού έχουν πρωτεύοντα ρόλο, με την εφαρμογή τους να προσδίδει στον Ιστό Αισθητήρων δυνατότητες επίγνωσης κατάστασης (situation awareness). Πιο συγκεκριμένα, τα βήματα και συστατικά που κρίνονται απαραίτητα για την υλοποίηση του Σημασιολογικού Ιστού Αισθητήρων είναι:

α) η σημασιολογική επισημείωση των πρότυπων αναπαραστάσεων και υπηρεσιών του SWE,
β) η ανάπτυξη κατάλληλων χωρικών, χρονικών και θεματικών οντολογιών για την αξιωματική θεμελίωση των αντίστοιχων περιγραφόμενων τομέων γνώσης και
γ) η πραγματοποίηση συλλογισμού στα επισημειωμένα δεδομένα, λαμβάνοντας υπόψη αξιώματα οντολογιών και κανόνες.

Η ταχεία μετάβαση στο Σημασιολογικό Ιστό Αισθητήρων ήταν και το ζητούμενο για την ομάδα εργασίας Semantic Sensor Network (SSN) Incubator Group, η οποία συστάθηκε από το W3C προκειμένου να δημιουργήσει μια οντολογία που θα ορίζει τυπικά τις δυνατότητες συστημάτων αισθητήρων και να προτείνει μηχανισμούς για τη σημασιολογική επισημείωση των SWE προτύπων. Το αποτέλεσμα των εργασιών της συγκεκριμένης ομάδας [26] ήταν ο ορισμός της οντολογίας Σημασιολογικού Δικτύου Αισθητήρων (Semantic Sensor Network Ontology) [64] και η πρόταση της χρήσης της τεχνολογίας XLink[7] για την επισημείωση αναπαραστάσεων και υπηρεσιών όπως οι O&M και SOS.

Μια άλλη παρεμφερής, αλλά ευρύτερη έννοια είναι αυτή του *σημασιολογικού δικτύου αισθητήρων* (semantic sensor network), η οποία συμπεριλαμβάνει κάθε δίκτυο αισθητήρων, το οποίο χρησιμοποιεί σημασιολογία για την αποτελεσματικότερη διαχείρισή του ή για την επεξεργασία των δεδομένων που παράγονται από αυτό [65]. Ένα σημασιολογικό δίκτυο αισθητήρων δε βασίζεται απαραίτητα στα SWE πρότυπα, γεγονός που ισχύει για το Σημασιολογικό Ιστό Αισθητήρων, και μπορεί να διαθέτει μία ή περισσότερες από τις παρακάτω λειτουργικότητες:

– ταξινόμηση των αισθητήρων με βάση το παρατηρούμενο φυσικό μέγεθος ή τη μεθοδολογία λήψης μέτρησης,

[7] Η XLink (XML Linking Language) είναι μια γλώσσα για τον ορισμό συνδέσμων μέσα σε XML έγγραφα: http://www.w3.org/TR/xlink11/.

– εύρεση αισθητήρων που μπορούν να λάβουν μια απαιτούμενη μέτρηση σε συγκεκριμένο μορφότυπο, καθώς και συνδυασμός αισθητήρων για τη δημιουργία σύνθετων νοητών αισθητήριων οργάνων,
– εξαγωγή γνώσης από δεδομένα χαμηλού επιπέδου μέσω διαδικασιών συλλογισμού,
– παραγωγή συμβάντος όταν ικανοποιείται μια συγκεκριμένη συνθήκη εντός ενός δεδομένου χρονικού διαστήματος.

Στο κεφάλαιο 6, παρουσιάζεται μια αρχιτεκτονική για ένα σημασιολογικό δίκτυο αισθητήρων, που χρησιμοποιεί συστατικά και τεχνολογίες του Σημασιολογικού Ιστού, όπως οντολογίες, κανόνες και συλλογισμό, για την εξαγωγή γνώσης από δεδομένα χαμηλού επιπέδου και την διευκόλυνση ανάπτυξης εφαρμογών που εκμεταλλεύονται αυτά τα δεδομένα. Επίσης, αναλύεται ένα σύστημα σημασιολογικής επεξεργασίας αισθητήριων μετρήσεων, το οποίο ακολουθεί μια γενικότερη μέθοδο σημασιολογικής επισημείωσης σε σχέση με αυτήν που προτείνεται από το SSN Incubator Group, παράγοντας νέες RDF προτάσεις μέσω της εφαρμογής ειδικών κανόνων αντιστοιχίας σε παρατηρήσεις αισθητήρων.

2.3.2 Διαχείριση ροών δεδομένων

Τα δίκτυα αισθητήρων αποτελούν ένα μόνο παράδειγμα συστήματος, για τη διαχείριση των δεδομένων του οποίου τα παραδοσιακά ΣΔΒΔ δεν επαρκούν. Το μοντέλο δεδομένων και η στρατηγική απάντησης ερωτημάτων για την περίπτωση παραδοσιακών βάσεων δεδομένων ταιριάζουν περισσότερο σε εφαρμογές όπου απαιτείται η μόνιμη αποθήκευση όλων των παραγόμενων δεδομένων, η ενημέρωση των δεδομένων είναι σπανιότερη από την υποβολή ερωτημάτων, ενώ τα ερωτήματα εκτελούνται όταν υποβάλλονται και οι απαντήσεις αντανακλούν τα τρέχοντα δεδομένα της βάσης [88]. Αντίθετα, εφαρμογές όπου η παραγωγή νέων δεδομένων είναι συνεχής, τα πιο πρόσφατα δεδομένα έχουν μεγαλύτερη αξία σε σχέση με παλαιότερα και η επεξεργασία των δεδομένων πρέπει να ολοκληρωθεί εντός συγκεκριμένου χρονικού ορίου, χρειάζονται να διαχειριστούν μια εισερχόμενη *ροή* πληροφορίας. Πεδία όπου το μοντέλο επεξεργασίας ροών δεδομένων βρίσκει εφαρμογή περιλαμβάνουν, εκτός από τα δίκτυα αισθητήρων, την ανάλυση κίνησης δικτύων, την ανάλυση τιμών μετοχών και οικονομικών δεικτών καθώς και την ανάλυση αρχείων καταγραφής χρήσης διαδικτυακών εφαρμογών ή αρχείων τηλεφωνικών κλήσεων.

Σε γενικές γραμμές, μια ροή δεδομένων είναι μια συνεχής, διατεταγμένη με βάση κάποια χρονική μεταβλητή, ακολουθία αντικειμένων. Η χρονική αυτή μεταβλητή μπορεί να δηλώνει είτε το χρόνο άφιξης του αντικειμένου στο σύστημα είτε κάποιο άλλο χρονικό ορόσημο (timestamp) που αναφέρεται σε εγγενείς ιδιότητες του αντικειμένου, όπως π.χ. η χρονική στιγμή που έλαβε χώρα το γεγονός που περιγράφεται από το αντικείμενο. Στην πρώτη περίπτωση, το χρονικό ορόσημο ονομάζεται *υπονοούμενο* (implicit), ενώ στη δεύτερη περίπτωση *κατηγορηματικό* (explicit). Γενικά, μια ροή δεδομένων $\mathbb{S}$ μπορεί να απεικονιστεί ως ένα σύνολο αντικειμένων $\mathbb{S}(t)$, καθένα εκ των οποίων χαρακτηρίζεται από ένα χρονικό ορόσημο. Οι βασικές διαφορές που παρουσιάζει το μοντέλο διαχείρισης μιας ροής δεδομένων σε σχέση με μοντέλα διαχείρισης στατικών δεδομένων είναι [20]:

α) Τα στοιχεία της ροής γίνονται διαθέσιμα σταδιακά κατά τη λειτουργία της εφαρμογής.

β) Το σύστημα που διαχειρίζεται τη ροή δεν έχει δυνατότητα ελέγχου της σειράς με την οποία γίνονται διαθέσιμα τα στοιχεία της ροής.

γ) Μια ροή είναι συνήθως άπειρη σε μέγεθος.

δ) Μόλις το σύστημα διαχείρισης επεξεργαστεί ένα στοιχείο της ροής, αυτό στη συνέχεια απορρίπτεται ή καταχωρείται σε δευτερεύον σύστημα αποθήκευσης και δεν μπορεί πλέον να ανακτηθεί εκτός και αν διατηρείται στην προσωρινή μνήμη του συστήματος, η οποία όμως είναι κατά πολύ μικρότερη του συνολικού μεγέθους της ροής.

ε) Τα ερωτήματα που τίθενται στο σύστημα διαχείρισης είναι συνεχή και αποτιμώνται συνεχώς καθώς τα δεδομένα της ροής γίνονται διαθέσιμα, σε αντίθεση με τα συνηθισμένα στατικά ερωτήματα που εκτελούνται μια φορά πάνω στο τρέχον στιγμιότυπο μιας βάσης δεδομένων.

Οι παραπάνω ιδιαιτερότητες οδήγησαν στην ανάπτυξη μιας ειδικής κατηγορίας συστημάτων, γνωστών ως *συστήματα διαχείρισης ροών δεδομένων* (data stream management systems ή DSMS), με πιο αξιοσημείωτα παραδείγματα τα STREAM [13], TelegraphCQ [56], Borealis [4], NiagaraCQ [60], OpenCQ [131], καθώς και τα εμπορικά StreamBase[8] και Sybase ESP[9]. Τα συγκεκριμένα συστήματα διαφοροποιούνται σε σχέση με παραδοσιακά συστήματα διαχείρισης βάσεων δεδομένων, προσαρμόζοντας τη λειτουργία τους στις παραπάνω ιδιαιτερότητες και κυρίως, στην απαίτηση εκτέλεσης συνεχών ερωτημάτων. Η συγκεκριμένη απαίτηση οδηγεί τα DSMS συστήματα στην υιοθέτηση μηχανισμών εκτέλεσης που συνεχώς τροφοδοτούνται με νέα δεδομένα (push-based στρατηγική), σε αντίθεση με συνηθισμένες μηχανές ερωταποκρίσεων που ανακτούν τα δεδομένα που είναι μόνιμα αποθηκευμένα σε κάποιο DBMS (pull-based στρατηγική). Επίσης, τα περισσότερα DSMS συστήματα είναι σε θέση να αντιμετωπίσουν πιθανές εκρήξεις στο ρυθμό με τον οποίο καθίστανται διαθέσιμα τα δεδομένα, προσαρμόζοντας τη συμπεριφορά τους και μειώνοντας την ακρίβεια των υπολογισμών τους καθώς πραγματοποιούν περισσότερες προσεγγίσεις μέσω τεχνικών αποβολής φόρτου (load shedding).

Μια άλλη έννοια στενά συνυφασμένη με το μοντέλο επεξεργασίας μιας ροής δεδομένων είναι αυτή του *παραθύρου*. Πέρα από το γεγονός ότι μια ροή δεδομένων είναι άπειρη, για την πλειονότητα των εφαρμογών που σχετίζονται με ροές δεδομένων τα πιο πρόσφατα δεδομένα είναι και τα πλέον κατάλληλα για τη λήψη αποφάσεων, οπότε η ανάλυση και επεξεργασία δεν έχει νόημα να πραγματοποιηθεί για το σύνολο της ροής, αλλά για ένα πρόσφατο υποσύνολο αυτής. Η διαδικασία απάντησης των συνεχών ερωτημάτων που έχουν καταχωρηθεί στο σύστημα διαχείρισης λαμβάνει συνήθως υπόψη της μονάχα αυτό το πεπερασμένο υποσύνολο πρόσφατων στοιχείων της ροής, που ονομάζεται παράθυρο. Επίσης, η εφαρμογή παραθύρων σε μια ροή δεδομένων επιτρέπει

[8] StreamBase: www.streambase.com/

[9] Sybase Event Stream Processor: www.sybase.com/products/financialservicessolutions/complex-event-processing

την αποτίμηση ερωτημάτων που χρησιμοποιούν ανασχετικούς τελεστές (blocking operators), όπως π.χ. τελεστές ομαδοποίησης ή διάταξης, καθώς οι συγκεκριμένοι τελεστές μπορούν μόνο να εφαρμοστούν σε πεπερασμένο αριθμό στοιχείων. Ένα άλλο πλεονέκτημα της εφαρμογής παραθύρων είναι η μείωση του υπολογιστικού φόρτου μέσω της μείωσης του όγκου των δεδομένων που μια πράξη (π.χ. συνένωση) χρειάζεται να λάβει υπόψη της [137].

Στη βιβλιογραφία, έχουν προταθεί διάφορα είδη παραθύρων ανάλογα με τον τρόπο σχηματισμού τους και εξέλιξής τους κατά την άφιξη νέων στοιχείων της ροής. Όταν το μήκος ενός παραθύρου ορίζεται με βάση ένα χρονικό διάστημα, τότε το παράθυρο ονομάζεται *χρονικό* ή *λογικό*. Αντίθετα, όταν το μήκος παραθύρου ορίζεται με βάση έναν αριθμό στοιχείων, τότε το παράθυρο ονομάζεται *φυσικό*. Ένα χρονικό παράθυρο με χρονικά όρια τ_1, τ_2 ορίζεται ως το σύνολο στοιχείων:

$$w_l(\mathbb{S}, \tau_1, \tau_2) = \{\mathbb{S}(t), \tau_1 \leq t \leq \tau_2\} \tag{2.1}$$

ενώ ένα φυσικό παράθυρο μήκους N στοιχείων περιέχει τα N πιο πρόσφατα στοιχεία της ροής, δεδομένης της τρέχουσας χρονικής στιγμής τ:

$$w_p(\mathbb{S}, N, \tau) = \{w_l(\mathbb{S}, \tau_1, \tau) : \tau_1 \leq \tau \wedge |w_l(\mathbb{S}, \tau_1, \tau)| \leq N \wedge \forall \tau_2 < \tau_1 : |w_l(\mathbb{S}, \tau_2, \tau)| > N\} \tag{2.2}$$

όπου $|w_l(\mathbb{S}, \tau_1, \tau_2)|$ ο αριθμός των στοιχείων που περιέχει το χρονικό παράθυρο w_l. Η εξίσωση 2.2 ορίζει το φυσικό παράθυρο ως ένα χρονικό παράθυρο εύρους $[\tau_1, \tau]$, με τη χρονική στιγμή τ_1 να επιλέγεται με τέτοιο τρόπο, ώστε ο συγκεντρωτικός αριθμός στοιχείων της ροής μετρώντας αντίστροφα από την τρέχουσα χρονική στιγμή τ μέχρι την τ_1 να είναι ακριβώς N. Ο συγκεκριμένος ορισμός υποθέτει ότι σε περιπτώσεις όπου στοιχεία με το ίδιο χρονικό ορόσημο τ_1 οδηγούν σε πλήθος στοιχείων του παραθύρου μεγαλύτερο του N, γίνεται αυθαίρετη επιλογή στοιχείων ώστε το μέγεθος του παραθύρου να είναι ακριβώς N [151].

Μια άλλη διάκριση χρονικών παραθύρων που παρουσιάζεται στη βιβλιογραφία βασίζεται στον τρόπο με τον οποίο ενημερώνεται ένα παράθυρο, καθώς η ροή δεδομένων εξελίσσεται. Οι πιο διαδεδομένες, από αυτή την άποψη, κατηγορίες παραθύρων είναι:

1. *ολισθαίνοντα* (sliding) παράθυρα
2. *επάλληλα* (tumbling) παράθυρα
3. παράθυρα *οροσήμου* (landmark)

Οι δύο παράμετροι που καθορίζουν το είδος του παραθύρου είναι το *μήκος* του και το *βήμα προόδου*, και τα δύο εκφρασμένα ως χρονικά διαστήματα. Το πιο συνηθισμένο μεταξύ των παραπάνω τριών παραθύρων είναι το ολισθαίνον, το οποίο ανανεώνεται ανά τακτά χρονικά διαστήματα ίσα με το βήμα προόδου, αντικαθιστώντας παλαιότερα στοιχεία της ροής με πιο πρόσφατα. Ένα ολισθαίνον παράθυρο μήκους w και βήματος προόδου d ορίζεται ως εξής:

$$w_{sl}(\mathbb{S}, w, d, \tau_0, \tau) = \begin{cases} \emptyset, & \tau_0 \leq \tau < \tau_0 + w \\ w_l(\mathbb{S}, \tau - w, \tau), & \tau_0 + w \leq \tau \wedge mod(\tau - \tau_0, d) = 0 \\ w_{sl}(\mathbb{S}, w, d, \tau_0, \tau - dt), & mod(\tau - \tau_0, d) \neq 0 \end{cases} \tag{2.3}$$

όπου τ_0 είναι η χρονική στιγμή κατά την οποία αρχίζει η εφαρμογή παραθύρου στη ροή και τ η τρέχουσα χρονική στιγμή. Ένα ολισθαίνον παράθυρο έχει σταθερό μήκος w και συνεπώς, παραμένει κενό τουλάχιστον[10] μέχρι τη χρονική στιγμή $\tau_0 + w$, ενώ όπως φαίνεται και από την εξίσωση 2.3, ανανεώνεται στις χρονικές στιγμές $\tau_0 + kd, k = 1, 2, \ldots$. Κατά τις ενδιάμεσες χρονικές στιγμές, το παράθυρο παραμένει σταθερό και ίδιο με αυτό που υπολογίστηκε την προηγούμενη χρονική στιγμή (ο όρος dt δηλώνει την χρονική ανάλυση του συστήματος, δηλαδή το χρονικό διάστημα που μεσολαβεί μεταξύ δύο διαδοχικών υπολογισμών του παραθύρου). Συνήθως, το βήμα προόδου d είναι μικρότερο από το μήκος w του παραθύρου, με αποτέλεσμα να υπάρχει επικάλυψη στοιχείων μεταξύ δύο διαδοχικών ολισθαίνοντων παραθύρων.

Ένα επάλληλο παράθυρο αποτελεί μια ειδική περίπτωση ολισθαίνοντος παραθύρου με το βήμα προόδου να ισούται με το μήκος του ($d = w$). Αυτό σημαίνει ότι δεν υπάρχουν επικαλύψεις στοιχείων μεταξύ διαδοχικών επάλληλων παραθύρων:

$$w_{tum}(\mathbb{S}, w, \tau_0, \tau) = w_{sl}(\mathbb{S}, w, w, \tau_0, \tau) \tag{2.4}$$

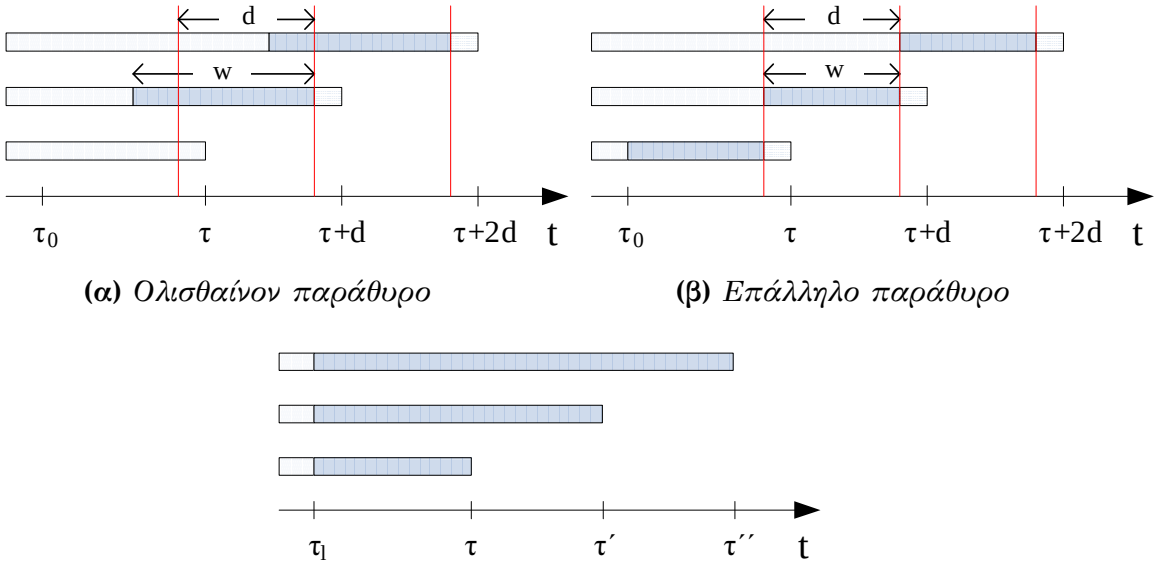
Τέλος, ένα παράθυρο οροσήμου έχει σταθερό ένα από τα δύο άκρα του, με αποτέλεσμα να υπάρχουν δύο είδη τέτοιων παραθύρων: αυτά με σταθερή χρονική έναρξη και αυτά με σταθερή χρονική λήξη [151], με τα πρώτα να είναι πιο συνηθισμένα σε σχέση με τα δεύτερα. Στην περίπτωση ενός παραθύρου οροσήμου με σταθερή χρονική έναρξη τ_l, το μήκος του είναι μεταβλητό, καθώς όσα νέα στοιχεία της ροής γίνονται διαθέσιμα το επαυξάνουν συνεχώς:

$$w_{land}(\mathbb{S}, \tau_l, \tau) = \begin{cases} w_l(\mathbb{S}, \tau_l, \tau), & \tau \geq \tau_l \\ \varnothing, & \tau < \tau_l \end{cases} \tag{2.5}$$

Οι τρεις κατηγορίες χρονικών παραθύρων απεικονίζονται διαγραμματικά στο σχήμα 2.4. Για κάθε κατηγορία, απεικονίζονται τρία διαδοχικά παράθυρα επί μιας ροής δεδομένων. Για το ολισθαίνον και το επάλληλο παράθυρο στα σχήματα 2.4α και 2.4β αντίστοιχα (μήκος παραθύρου w και βήμα προόδου d), τα παράθυρα λαμβάνονται τις χρονικές στιγμές τ, $\tau + d$ και $\tau + 2d$ και απεικονίζονται ως σκιασμένα υποσύνολα της ροής. Οι κάθετες συνεχείς γραμμές στα ίδια σχήματα σηματοδοτούν τις χρονικές στιγμές $t_0 + d$, $t_0 + 2d$ και $t_0 + 3d$, κατά τις οποίες ανανεώνεται το παράθυρο. Παρατηρούμε ότι στο σχήμα 2.4α, ισχύει $\tau < \tau_0 + w$, οπότε το παράθυρο είναι κενό τη χρονική στιγμή τ. Στο σχήμα 2.4γ απεικονίζονται παράθυρα οροσήμου με σταθερή έναρξη t_l και στιγμές λήξης τ, τ' και τ''.

Σε πλήρη αντιστοιχία με τις τρεις κατηγορίες χρονικών παραθύρων που αναφέρθηκαν πιο πάνω, μπορούν να οριστούν και φυσικά παράθυρα που ενημερώνονται με ανάλογο τρόπο. Αυτά τα φυσικά παράθυρα μπορούν να εκφραστούν με κατάλληλη τροποποίηση των εξισώσεων 2.3, 2.4 και 2.5, ώστε να λαμβάνουν υπόψη τους τον ορισμό του φυσικού παραθύρου (εξίσωση 2.2). Εναλλακτικά και προς αποφυγή της πολυπλοκότητας που εισάγει η εξίσωση 2.2, τα διάφορα είδη φυσικών παραθύρων μπορούν να εκφραστούν ευκολότερα, θεωρώντας μια ροή δεδομένων ως ένα σύνολο αντικειμένων $\mathbb{S}(n)$, όπου n

[10] Αν $w > d$, το πρώτο στιγμιότυπο του παραθύρου υπολογίζεται τη χρονική στιγμή $\tau_0 + 2d$, ενώ αν $w \leq d$, υπολογίζεται τη χρονική στιγμή $\tau_0 + d$.



(α) *Ολισθαίνον παράθυρο* **(β)** *Επάλληλο παράθυρο*

(γ) *Παράθυρο οροσήμου*

Σχήμα 2.4: *Κατηγορίες χρονικών παραθύρων*

ο αύξων αριθμός του αντικειμένου ο οποίος υπολογίζεται με βάση κάποιο χρονικό ορόσημο του εν λόγω αντικειμένου. Για λόγους διευκόλυνσης της παρουσίασης, υιοθετούμε και εδώ την παραδοχή ότι σε αντικείμενα με ίδιο χρονικό ορόσημο ανατίθενται διαδοχικοί αύξοντες αριθμοί με τυχαίο τρόπο. Σύμφωνα με τα προηγούμενα, ένα φυσικό παράθυρο που περιέχει τα στοιχεία με αύξοντες αριθμούς n_1 ως n_2 έχει μήκος $n_2 - n_1 + 1$ και ορίζεται ως εξής:

$$w_{ph}(\mathbb{S}, n_1, n_2) = \{\mathbb{S}(n), n_1 \leq n \leq n_2\} \tag{2.6}$$

ενώ τα αντίστοιχα φυσικά ολισθαίνοντα, επάλληλα και παράθυρα οροσήμου ορίζονται αντίστοιχα ως εξής:

$$w_{sl,ph}(\mathbb{S}, W, D, \nu_0, \nu) = \begin{cases} \varnothing, & \nu_0 \leq \nu < \nu_0 + W - 1 \\ w_{ph}(\mathbb{S}, \nu - W + 1, \nu), & \nu_0 + W - 1 \leq \nu \wedge mod(\nu - \nu_0, D) = 0 \\ w_{sl,ph}(\mathbb{S}, W, D, \nu_0, \nu - 1), & mod(\nu - \nu_0, D) \neq 0 \end{cases} \tag{2.7}$$

$$w_{tum,ph}(\mathbb{S}, W, \nu_0, \nu) = w_{sl,ph}(\mathbb{S}, W, W, \nu_0, \nu) \tag{2.8}$$

$$w_{land,ph}(\mathbb{S}, \nu_0, \nu) = \begin{cases} w_{ph}(\mathbb{S}, \nu_0, \nu), & \nu \geq \nu_0 \\ \varnothing, & \nu < \nu_0 \end{cases} \tag{2.9}$$

όπου W και D το μήκος του παραθύρου και το βήμα προόδου αντίστοιχα εκφρασμένα σε πλήθος στοιχείων, ν_0 ο αύξων αριθμός στοιχείου πέραν του οποίου αρχίζει η εφαρμογή του παραθύρου και ν ο αύξων αριθμός του τρέχοντος στοιχείου της ροής.

Κεφάλαιο 3

Σχεσιακές Βάσεις Δεδομένων στο Σημασιολογικό Ιστό: Επισκόπηση

Περιεχόμενα

Ο ρόλος των σχεσιακών βάσεων δεδομένων (ΒΔ) στο γρήγορα αναπτυσσόμενο περιβάλλον του Παγκόσμιου Ιστού εξετάστηκε από τα πρώτα στάδια σύλληψης του οράματος του Σημασιολογικού Ιστού, όχι μόνο επειδή ο εμπνευστής του τελευταίου, Tim Berners-Lee, το έχει χαρακτηρίσει ως μία "παγκόσμια βάση δεδομένων" [36], αλλά - κυρίως - επειδή αυτό το νέο, την εποχή εκείνη, ερευνητικό πεδίο θα μπορούσε να εκμεταλλευτεί την τεχνογνωσία και ωριμότητα του πεδίου των βάσεων δεδομένων, βασιζόμενο στα θεωρητικά και πρακτικά πορίσματά του. Εντούτοις, η συνεργασία και ανταλλαγή ιδεών μεταξύ των δύο αυτών πεδίων δεν ήταν μονόπλευρη: η ερευνητική κοινότητα των βάσεων δεδομένων σύντομα αναγνώρισε τις ευκαιρίες που προσέφερε το πεδίο του Σημασιολογικού Ιστού και πώς αυτό θα μπορούσε να δώσει λύσεις και να παρέχει έμπνευση για την αντιμετώπιση γνωστών θεμάτων, όπως η ολοκλήρωση ετερογενών βάσεων δεδομένων, η διαλειτουργικότητα, ο εννοιολογικός σχεδιασμός και οι επαγωγικές βάσεις δεδομένων [177].

Οι πρώτες προσπάθειες συνδυασμού των δύο αυτών ερευνητικών πεδίων αρχικά εστίασαν στην εναρμόνιση των διαφορών μεταξύ των δύο κυριότερων και πλέον αντιπροσωπευτικών μοντέλων κάθε πεδίου: του **σχεσιακού** και του **RDF** μοντέλου. Το πρόβλημα της εναρμόνισης του σχεσιακού με το οντολογικό μοντέλο είναι γνωστό και ως *πρόβλημα αντιστοιχίας σχεσιακής βάσης δεδομένων με οντολογία*, το οποίο πηγάζει από την ασυμφωνία και τις δομικές διαφορές μεταξύ σχεσιακών και αντικειμενοστρεφών μοντέλων (object-relational

impedance mismatch). Εντούτοις, το εν λόγω πρόβλημα της αντιστοιχίας μεταξύ σχεσιακών ΒΔ και RDF γράφων ή οντολογιών δεν έχει μόνο αξία ως ένα καθαρά θεωρητικό πρόβλημα αντιστοιχίας μεταξύ δύο διαφορετικών φορμαλισμών, αλλά και ποικιλόμορφη πρακτική αξία. Αυτό έχει ως συνέπεια, το συγκεκριμένο πρόβλημα να εξειδικεύεται σε επιμέρους κατηγορίες, σε κάθε μία από τις οποίες οι αντιστοιχίες ανακαλύπτονται, ορίζονται και χρησιμοποιούνται με διαφορετικό τρόπο.

Μία από τις πρώτες, χρονικά, κατηγορίες εφαρμογών όπου εξετάστηκε το ζήτημα της αντιστοιχίας μεταξύ σχεσιακών ΒΔ και RDF ήταν η ανάπτυξη συστημάτων αποθήκευσης, διαχείρισης και επερώτησης RDF γράφων, γνωστών και ως triple stores [71]. Οι επιδόσεις και η κλιμάκωση που επιτυγχάνουν τα σύγχρονα συστήματα διαχείρισης σχεσιακών ΒΔ τα καθιστούν εξαιρετική λύση για τις λειτουργίες που καλείται να φέρει σε πέρας ένα triple store. Αυτός εξάλλου ήταν και ο λόγος που αρκετά από τα πρώτα triple stores που υλοποιήθηκαν βασίζονταν στην τεχνολογία σχεσιακών ΒΔ. Στην περίπτωση σχεδιασμού ενός triple store, ουσιαστικά υπάρχει μια ροή δεδομένων και πληροφορίας *από* τον RDF γράφο ή οντολογία *προς* τη σχεσιακή ΒΔ, η οποία έχει ένα κατάλληλα βελτιστοποιημένο σχήμα που θα φιλοξενήσει υπάρχοντα RDF δεδομένα [30]. Σήμερα, το ερευνητικό ενδιαφέρον για το συγκεκριμένο πρόβλημα έχει ατονήσει, δεδομένου του ότι πλέον έχουν αναπτυχθεί πολλά ώριμα και σταθερά συστήματα διαχείρισης RDF γράφων, τα οποία βασίζονται όχι μόνο σε σχεσιακές ΒΔ, αλλά και σε βάσεις δεδομένων που χρησιμοποιούν δομές γράφων (graph databases).

Μια σαφώς διακριτή και ίσως πιο ενδιαφέρουσα κατηγορία εφαρμογών είναι αυτή που θεωρεί μια υπάρχουσα και πλήρως λειτουργική σχεσιακή ΒΔ και προσπαθεί να καταστήσει την αποθηκευμένη σε αυτή πληροφορία αξιοποιήσιμη από εφαρμογές Σημασιολογικού Ιστού. Σε αυτή την περίπτωση, το κίνητρο δεν είναι πλέον η αποδοτική αποθήκευση και επερώτηση οντολογικών δομών, αλλά η εύκολη ολοκλήρωση σχεσιακών ΒΔ, ο συνδυασμός των περιεχομένων τους με περιεχόμενο από άλλες πηγές, η μάθηση μιας οντολογίας από μια σχεσιακή ΒΔ, η μαζική παραγωγή δεδομένων Σημασιολογικού Ιστού, η επερώτηση μιας σχεσιακής ΒΔ μέσω οντολογικών ερωτημάτων, ακόμα και η σημασιολογική επισημείωση δυναμικών ιστοσελίδων. Σε καθένα από τα παραπάνω θέματα, η έννοια της αντιστοιχίας μεταξύ μιας σχεσιακής ΒΔ και τεχνολογιών Σημασιολογικού Ιστού ορίζεται διαφορετικά, αν και στις αντίστοιχες ερευνητικές εργασίες χρησιμοποιείται ο ίδιος όρος για να περιγράψει διαφορετικά προβλήματα, το καθένα εκ των οποίων θέτει τις δικές του προκλήσεις και χρήζει ξεχωριστής αντιμετώπισης. Στο παρόν κεφάλαιο, προκειμένου να τεθεί μια τάξη στην πλούσια σχετική βιβλιογραφία και να αποφευχθεί η σύγχυση μεταξύ διαφορετικών προβλημάτων, εξετάζουμε και κάνουμε σαφή διάκριση μεταξύ προσεγγίσεων που πραγματοποιούν κάποια από τα παρακάτω:

α) δημιουργούν εκ του μηδενός μια νέα οντολογία, βασιζόμενες σε πληροφορία που εξάγεται από μια σχεσιακή ΒΔ

β) παράγουν RDF προτάσεις που χρησιμοποιούν όρους από μία ή περισσότερες υπάρχουσες οντολογίες και που αντικατοπτρίζουν τα περιεχόμενα μιας σχεσιακής ΒΔ

γ) θέτουν σημασιολογικά ερωτήματα σε μια σχεσιακή ΒΔ

δ) ανακαλύπτουν συσχετίσεις και αντιστοιχίες μεταξύ του σχήματος μιας σχεσιακής ΒΔ και μιας δοσμένης οντολογίας.

3.1 Κίνητρα και οφέλη

Η σημασία των βάσεων δεδομένων για το Σημασιολογικό Ιστό φαίνεται από τα πολλαπλά οφέλη και χρήσεις μιας αντιστοιχίας ΒΔ σε RDF ή οντολογία. Όπως αναφέρθηκε και στην εισαγωγή του παρόντος Κεφαλαίου, το εν λόγω πρόβλημα αντιστοιχίας δεν προέκυψε ως μια αποκλειστικά θεωρητική άσκηση μετατροπής ενός μοντέλου σε ένα άλλο. Είναι σημαντικό να αναγνωριστούν και να γίνουν κατανοητά τα διαφορετικά κίνητρα ερευνητικών προσεγγίσεων που συνδυάζουν τις σχεσιακές ΒΔ με το Σημασιολογικό Ιστό, ώστε να υπάρξει ένας σαφής διαχωρισμός στόχων και προκλήσεων, αλλά παράλληλα και να καταδειχθεί η σημασία του όλου προβλήματος. Στην τρέχουσα ενότητα, προχωρούμε σε μια αναφορά και σύντομη περιγραφή των κινήτρων και ωφελειών του υπό εξέταση προβλήματος. Αξίζει βέβαια να τονιστεί ότι τα παρακάτω οφέλη δεν είναι αμοιβαία αποκλειόμενα, αφού υπάρχουν ερευνητικές εργασίες που συνδυάζουν περισσότερα του ενός.

Σημασιολογική επισημείωση δυναμικών ιστοσελίδων. Ένας από τους στόχους του οράματος του Σημασιολογικού Ιστού, όπως αναφέρθηκε και στο Κεφάλαιο 1, είναι η μετατροπή του Παγκόσμιου Ιστού, από την τρέχουσα μορφή του που αποτελεί έναν Ιστό εγγράφων (Web of documents) σε έναν Ιστό δεδομένων (Web of Data). Ένας προφανής τρόπος για να πλησιάσουμε στην επίτευξη αυτού του στόχου είναι η σημασιολογική επισημείωση (semantic annotation) των HTML σελίδων του Ιστού, οι οποίες καθορίζουν τον τρόπο παρουσίασης του περιεχομένου και προορίζονται αποκλειστικά για ανθρώπινη κατανάλωση. Προκειμένου το περιεχόμενο μιας HTML σελίδας να καταστεί κατάλληλο για επεξεργασία από πράκτορες λογισμικού και υπηρεσίες Ιστού, θα πρέπει αυτό να επισημειωθεί με όρους από οντολογίες ή άλλα καθιερωμένα λεξιλόγια. Η διαδικασία της επισημείωσης έχει διευκολυνθεί σε μεγάλο βαθμό από την εμφάνιση τεχνολογιών, όπως η RDFa και τα μικροπρότυπα (microformats), οι οποίες ενσωματώνουν στις οδηγίες (tags) της XHTML οντολογικούς όρους. Παρ' όλα αυτά, η παραπάνω διαδικασία δεν είναι η πλέον κατάλληλη για την περίπτωση δυναμικών ιστοσελίδων που αντλούν το περιεχόμενό τους κατευθείαν από υποκείμενες ΒΔ, γεγονός που ισχύει για συστήματα διαχείρισης περιεχομένου (content management systems ή CMS) και Web 2.0 εφαρμογές (π.χ. fora, wikis) [18]. Οι δυναμικές ιστοσελίδες αποτελούν το μεγαλύτερο μέρος του Παγκόσμιου Ιστού, γνωστό και ως Βαθύς Ιστός (Deep Web) [85], τα περιεχόμενα του οποίου δεν είναι πάντα διαθέσιμα μέσω συμβατικών μηχανών αναζήτησης, αφού παρουσιάζονται δυσκολίες στη δεικτοδότησή του (indexing). Λαμβάνοντας υπόψη την πρακτική αδυναμία χειροκίνητης επισημείωσης όλων των δυναμικών ιστοσελίδων ενός ιστότοπου, μία λύση θα ήταν η άμεση «επισημείωση» της υποκείμενης ΒΔ (ή τουλάχιστον, εκείνου του τμήματος της ΒΔ που ο υπεύθυνος του ιστότοπου είναι διατεθειμένος να αποκαλύψει). Αυτή η «επισημείωση» θα ήταν απλά ένα σύνολο αντιστοιχιών μεταξύ των στοιχείων του σχήματος της ΒΔ και μιας οντολογίας που ταιριάζει θεματικά με το περιεχόμενο της δυναμικής ιστοσελίδας [197]. Δεδομένου ενός τέτοιου συνόλου

αντιστοιχιών, η αυτόματη παραγωγή σημασιολογικά επισημειωμένων δυναμικών ιστοσελίδων θα μπορούσε να προκύψει ως αποτέλεσμα μιας τετριμμένης προγραμματιστικής διαδικασίας.

Ολοκλήρωση ετερογενών βάσεων δεδομένων. Η επίλυση της ετερογένειας είναι ένα από τα μακροβιότερα και δημοφιλή προβλήματα στο ευρύτερο ερευνητικό πεδίο των βάσεων δεδομένων, το οποίο παραμένει σε μεγάλο βαθμό, άλυτο. Ετερογένεια εμφανίζεται μεταξύ δύο η περισσότερων ΒΔ, όταν αυτές χρησιμοποιούν διαφορετική υλική υποδομή (hardware) ή λογισμικό (software), ακολουθούν διαφορετικές συντακτικές συμβάσεις ή μοντέλα αναπαράστασης ή ακόμα όταν ερμηνεύουν διαφορετικά ίδια ή παρεμφερή δεδομένα. Με την επικράτηση του σχεσιακού μοντέλου και την καθιέρωση προτύπων όπως η SQL, το είδος της ετερογένειας το οποίο θέτει ακόμα ερευνητικές προκλήσεις είναι η *σημασιολογική ετερογένεια* [178], δηλαδή η πιθανή διαφορά στο νόημα συντακτικά πανομοιότυπων δεδομένων. Η επίλυση όλων των μορφών ετερογένειας επιτρέπει την ολοκλήρωση δύο ή περισσότερων ΒΔ και την ενιαία επερώτηση των περιεχομένων τους. Σε τυπικές αρχιτεκτονικές συστημάτων ολοκλήρωσης ΒΔ, ένα ή περισσότερα εννοιολογικά μοντέλα χρησιμοποιούνται για την περιγραφή των περιεχομένων κάθε τοπικής υπό ολοκλήρωση ΒΔ, ερωτήματα τίθενται με βάση όρους ενός καθολικού (global) εννοιολογικού μοντέλου και ένα ειδικό υπο-σύστημα (wrapper) αναλαμβάνει, για κάθε συμμετέχουσα ΒΔ, να ανασχηματίσει τα εισερχόμενα ερωτήματα και να ανακτήσει τα κατάλληλα δεδομένα. Σε συστήματα ολοκλήρωσης με χρήση οντολογιών (ontology-based integration), όπου το ρόλο των εννοιολογικών μοντέλων αναλαμβάνουν οντολογίες, είναι απαραίτητος ο ορισμός ενός συνόλου αντιστοιχιών μεταξύ κάθε τοπικής ΒΔ και μίας ή περισσότερων οντολογιών [198]. Κάθε τέτοιο σύνολο αντιστοιχιών αποτελείται από λογικές φόρμουλες (formulas) που:

α) εκφράζουν τα στοιχεία του σχήματος μιας ΒΔ (π.χ. σχέσεις, γνωρίσματα) ως συζευκτικά ερωτήματα (conjunctive queries) που χρησιμοποιούν όρους της οντολογίας (αντιστοιχίες γνωστές και ως Local as View ή LAV),

β) εκφράζουν τους όρους της οντολογίας ως συζευκτικά ερωτήματα που χρησιμοποιούν στοιχεία του σχήματος μιας ΒΔ (Global as View ή GAV αντιστοιχίες) ή

γ) εκφράζουν την ισοδυναμία δύο συζευκτικών ερωτημάτων, με το ένα να χρησιμοποιεί όρους της τοπικής ΒΔ και το άλλο όρους της οντολογίας (Global Local as View ή GLAV αντιστοιχίες) [125].

Ο τύπος των αντιστοιχιών που χρησιμοποιούνται σε ένα σύστημα ολοκλήρωσης ΒΔ επηρεάζει την πολυπλοκότητα της διαδικασίας απάντησης ερωτημάτων, αλλά και την ευκολία επεκτασιμότητάς του. Για παράδειγμα, για την περίπτωση GAV αντιστοιχιών, η διαδικασία επανεγγραφής (rewriting) και απάντησης ερωτημάτων είναι πολύ απλή, αλλά η προσθήκη μιας νέας τοπικής ΒΔ απαιτεί επαναπροσδιορισμό όλων των αντιστοιχιών. Το αντίστροφο ισχύει για την περίπτωση LAV αντιστοιχιών. Σε κάθε περίπτωση, γίνεται φανερό ότι η ανακάλυψη και ο ορισμός αντιστοιχιών μεταξύ ενός σχήματος σχεσιακής ΒΔ και μιας οντολογίας αποτελεί ένα από τα απαραίτητα βήματα σε μια διαδικασία ολοκλήρωσης ΒΔ με χρήση οντολογιών.

Πρόσβαση στα περιεχόμενα μιας ΒΔ με χρήση οντολογιών. Με τρόπο ανάλογο αυτού μιας αρχιτεκτονικής ολοκλήρωσης ΒΔ, προσεγγίσεις πρόσβασης σε δεδομένα με χρήση οντολογιών (ontology-based data access ή OBDA εν συντομία) υποθέτουν ότι μια οντολογία είναι συσχετισμένη με μια τοπική ΒΔ, λειτουργώντας ως ένα ενδιάμεσο επίπεδο μεταξύ του χρήστη και των δεδομένων. Ο στόχος ενός OBDA συστήματος είναι να δώσει στο χρήστη ενός πληροφοριακού συστήματος μια υψηλού επιπέδου άποψη των δεδομένων, αποκρύπτοντας λεπτομέρειες αποθήκευσής τους στην υποκείμενη ΒΔ [158]. Η οντολογία παρέχει μια αφαιρετική περιγραφή των περιεχομένων της ΒΔ, επιτρέποντας στον τελικό χρήστη αλλά και σε μια εξωτερική εφαρμογή να θέτει ερωτήματα σε όρους υψηλού επιπέδου από ένα συγκεκριμένο γνωστικό αντικείμενο. Κατά κάποιον τρόπο, ένα OBDA σύστημα μοιάζει με το υπο-σύστημα-κέλυφος (wrapper) μιας αρχιτεκτονικής ολοκλήρωσης, καθώς λειτουργεί ως περίβλημα που αποκρύπτει τις χαμηλού επιπέδου λεπτομέρειες μιας τοπικής ΒΔ ή και πιθανές αλλαγές στο λογικό επίπεδο που δεν επηρεάζουν την έννοια των δεδομένων (όπως επιφέρει λόγου χάριν, η διαδικασία της κανονικοποίησης). Το OBDA σύστημα αναλαμβάνει να μετατρέψει ερωτήματα που χρησιμοποιούν όρους ενός εννοιολογικού σχήματος (μιας οντολογίας, στη συγκεκριμένη περίπτωση) σε ερωτήματα που τίθενται κατευθείαν στην τοπική ΒΔ, λαμβάνοντας υπόψη του αντιστοιχίες μεταξύ μιας ΒΔ και μιας σχετικής οντολογίας που περιγράφει την ίδια θεματική περιοχή, διαδικασία γνωστή και ως επανεγγραφή ερωτημάτων (query rewriting). Το βασικό πλεονέκτημα μιας OBDA αρχιτεκτονικής είναι το γεγονός ότι παρέχεται η δυνατότητα άμεσης σημασιολογικής επερώτησης μιας ΒΔ, αποφεύγοντας τη φυσική αναπαραγωγή όλου του περιεχομένου της σε μορφή RDF.

Σημασιολογικός εμπλουτισμός SQL ερωτημάτων. Ένα άλλο πρόβλημα στο οποίο βρίσκει εφαρμογή μια αντιστοιχία μεταξύ οντολογίας και ΒΔ είναι η σημασιολογική επανεγγραφή SQL ερωτημάτων, διαδικασία η οποία έχει ως αποτέλεσμα ένα εμπλουτισμένο SQL ερώτημα που αντιπροσωπεύει καλύτερα την πρόθεση και τις απαιτήσεις του χρήστη [34]. Αυτή η επανεγγραφή επιτυγχάνεται αντικαθιστώντας όρους του SQL ερωτήματος με σχετικούς και συνώνυμους όρους από την οντολογία. Μια παραπλήσια άξια αναφοράς εφαρμογή είναι η δυνατότητα επερώτησης σχεσιακών δεδομένων με χρήση μιας οντολογίας ως πλαίσιο αναφοράς [68]. Η συγκεκριμένη δυνατότητα έχει υλοποιηθεί σε κάποια συστήματα διαχείρισης ΒΔ[1], επιτρέποντας την ενσωμάτωση σε ένα SQL ερώτημα συνθηκών που εκφράζονται σε όρους μιας οντολογίας.

Μαζική παραγωγή περιεχομένου για το Σημασιολογικό Ιστό. Ένας από τους λόγους καθυστέρησης της πλήρους υλοποίησης του οράματος του Σημασιολογικού Ιστού είναι η έλλειψη επιτυχημένων εργαλείων και εφαρμογών που θα αναδείξουν τα πλεονεκτήματα των τεχνολογιών Σημασιολογικού Ιστού (μια ολοκληρωμένη ανάλυση του οικοσυστήματος του Παγκόσμιου Ιστού, των κατηγοριών χρηστών που δραστηριοποιούνται σε αυτόν και των λόγων της καθυστερημένης ανάπτυξης του Σημασιολογικού Ιστού παρουσιάζεται στη δημοσίευση [116]). Ο βαθμός επιτυχίας και αποδοχής σχετικών εργαλείων συσχετίζεται όμως άμεσα με τη διαθεσιμότητα μεγάλου όγκου σημασιολογικών

[1] Χαρακτηριστικά παραδείγματα αποτελούν οι βάσεις δεδομένων Oracle και OpenLink Virtuoso.

δεδομένων, γεγονός που οδηγεί σε ένα φαύλο κύκλο αιτίας και αποτελέσματος [98]. Καθώς οι σχεσιακές ΒΔ αποτελούν ένα από τα πλέον δημοφιλή αποθηκευτικά μέσα, στα οποία φιλοξενείται η πλειοψηφία του περιεχομένου του Παγκόσμιου Ιστού, μία πιθανή λύση για την μαζική παραγωγή σημασιολογικού περιεχομένου θα ήταν η, κατά προτίμηση αυτόματη, εξαγωγή των περιεχομένων αυτών των ΒΔ σε RDF. Η εφαρμογή μιας τέτοιας διαδικασίας θα δημιουργούσε μεγάλο όγκο RDF δεδομένων, γεγονός που με τη σειρά του θα έδινε την απαραίτητη ώθηση σε προγραμματιστές και εταιρείες να επενδύσουν περισσότερο σε τεχνολογίες Σημασιολογικού Ιστού, με αναμενόμενο αποτέλεσμα την αυξημένη παραγωγή σχετικών εφαρμογών. Σημειώνουμε ότι ο όρος «αντιστοιχία ΒΔ με οντολογία» έχει χρησιμοποιηθεί για να περιγράψει, μεταξύ άλλων, και την παραπάνω διαδικασία εξαγωγής RDF δεδομένων.

Οντολογική μάθηση. Η διαδικασία ανάπτυξης μιας οντολογίας εκ του μηδενός για κάποια θεματική περιοχή είναι μια διαδικασία επίπονη, χρονοβόρα και αρκετά επιρρεπής σε λάθη. Προκειμένου να μειωθεί ο βαθμός εξάρτησης από τον ανθρώπινο παράγοντα, αρκετές ημι-αυτόματες μέθοδοι για την δημιουργία μιας οντολογίας έχουν προταθεί, οι οποίες εξάγουν γνώση από έγγραφα ελεύθερου κειμένου, ημιδομημένα έγγραφα, λεξιλόγια, θησαυρούς και από άλλες πηγές, δομημένες ή μη [89]. Το πρόβλημα αυτό είναι γνωστό και ως οντολογική μάθηση (ontology learning) και αποτελεί ειδική περίπτωση του προβλήματος της εξαγωγής πληροφορίας (information extraction). Οι σχεσιακές βάσεις δεδομένων αποτελούν δομημένες πηγές πληροφορίας και, στην περίπτωση που ο σχεδιασμός του σχήματός τους ακολουθεί κάποιες καθιερωμένες πρακτικές [78] (δηλαδή έχει προηγηθεί η δημιουργία ενός εννοιολογικού σχήματος με βάση το εκτεταμένο μοντέλο οντοτήτων-συσχετίσεων ή σύμφωνα με κάποια μεθοδολογία αντικειμενοστρεφούς μοντελοποίησης, όπως η UML), αποτελούν σημαντικές και αξιόπιστες πηγές γνώσης για μια δεδομένη θεματική περιοχή. Αυτό ισχύει ιδιαίτερα για την περίπτωση εταιρικών ΒΔ, των οποίων το σχήμα και τα δεδομένα συντηρούνται και ανανεώνονται συχνά, ώστε να διατηρούν πλήρη και ενημερωμένη πληροφορία για κρίσιμες επιχειρησιακές διεργασίες [207]. Συνεπώς, οντολογίες πλούσιας εκφραστικότητας μπορούν να εξαχθούν από συστήματα σχεσιακών ΒΔ, αντλώντας πληροφορία από τα σχήματα, περιεχόμενα, ερωτήματα και αποθηκευμένες διεργασίες (stored procedures) τους, αρκεί βέβαια το τελικό αποτέλεσμα να ελεγχθεί και πιθανώς εμπλουτιστεί από ένα γνώστη της υπό περιγραφή θεματικής περιοχής. Η μάθηση μιας οντολογίας είναι ένα αρκετά συνηθισμένο κίνητρο για την πραγματοποίηση αντιστοιχίας μεταξύ βάσεων δεδομένων και οντολογιών, ειδικά στην περίπτωση που δεν υπάρχει διαθέσιμη οντολογία για ένα συγκεκριμένο γνωστικό αντικείμενο, φαινόμενο συνηθισμένο ιδιαίτερα τα προηγούμενα χρόνια, όταν ο αριθμός των διαθέσιμων οντολογιών ήταν πολύ μικρότερος από τα σημερινά επίπεδα. Τα τελευταία χρόνια, τεχνικές οντολογικής μάθησης από σχεσιακές ΒΔ χρησιμοποιούνται κυρίως για να δημιουργήσουν μια οντολογία που λειτουργεί ως μονάδα-κέλυφος σε συστήματα πρόσβασης σε δεδομένα με χρήση οντολογιών [172] και συστήματα ολοκλήρωσης ΒΔ [47].

Ορισμός του νοήματος του σχήματος μιας σχεσιακής ΒΔ. Όπως αναφέρθηκε και προηγουμένως, οι καθιερωμένες πρακτικές δημιουργίας του σχήματος μιας σχεσιακής ΒΔ προϋποθέτουν την αρχική σχεδίαση ενός εννοιολογικού μοντέ-

λου, το οποίο στη συνέχεια μετατρέπεται στο τελικό σχεσιακό σχήμα, σε ένα βήμα γνωστό και ως λογική σχεδίαση. Το αρχικό εννοιολογικό μοντέλο συνήθως δεν διατηρείται μαζί με το σχεσιακό σχήμα της ΒΔ, με αποτέλεσμα μεταγενέστερες αλλαγές στο δεύτερο να μην μεταφέρονται και στο πρώτο, ενώ συχνά οι αλλαγές αυτές δεν τεκμηριώνονται καν. Αυτή η κατάσταση οδηγεί σε σχήματα ΒΔ όπου έχει χαθεί η αρχική πρόθεση του σχεδιαστή και, ως εκ τούτου, δύσκολα επεκτείνονται ή μετατρέπονται σε κάποιο άλλο μοντέλο (π.χ. σε κάποιο αντικειμενοστρεφές) με ταυτόχρονη διατήρηση της αρχικής τους σημασίας. Η διαδικασία καθορισμού συσχετίσεων μεταξύ μιας σχεσιακής ΒΔ και μιας οντολογίας συμβάλλει στη θεμελίωση του νοήματος της πρώτης σε όρους ενός πολύ εκφραστικού εννοιολογικού μοντέλου, όπως είναι μια οντολογία, γεγονός καίριας σημασίας όχι μόνο για λόγους συντήρησης της ΒΔ, αλλά επίσης για την ολοκλήρωση της ΒΔ με άλλες πηγές δεδομένων [59], καθώς και για την ανακάλυψη αντιστοιχιών μεταξύ δύο ή περισσότερων σχημάτων ΒΔ [9, 75]. Στην τελευταία περίπτωση, οι αντιστοιχίες μεταξύ ΒΔ και οντολογίας χρησιμοποιούνται ως σημείο αναφοράς για τον ορισμό συσχετίσεων μεταξύ σχημάτων διαφορετικών ΒΔ.

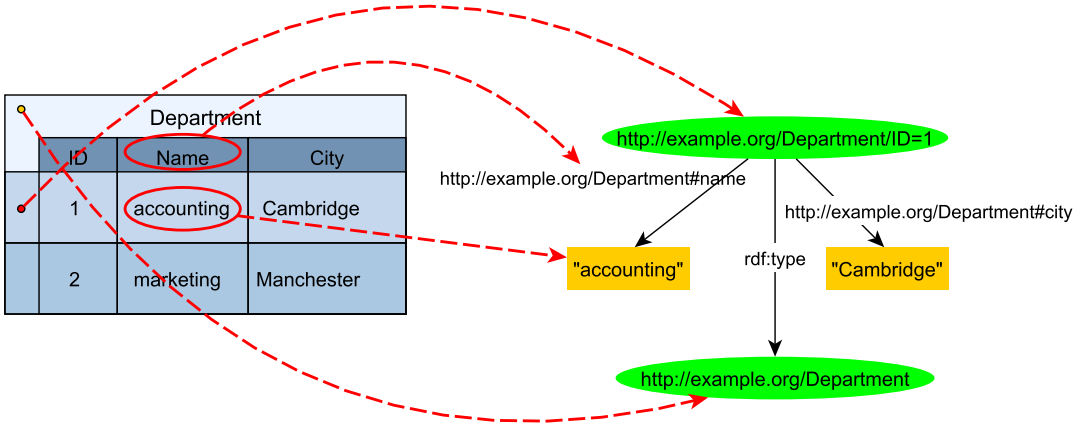
Συνδυασμός περιεχομένων ΒΔ με δεδομένα από άλλες πηγές. Η εξαγωγή των περιεχομένων των σχεσιακών ΒΔ σε ένα καθολικά αποδεκτό μοντέλο περιγραφής, όπως φιλοδοξεί να γίνει το RDF, θα επιτρέψει την ολοκλήρωση και το συνδυασμό τους με πληροφορία που είναι ήδη εκφρασμένη σε RDF. Αυτή η πληροφορία μπορεί να προέρχεται από δομημένες και αδόμητες πηγές των οποίων τα περιεχόμενα έχουν ήδη εξαχθεί σε RDF, γεγονός που παρακάμπτει τις όποιες δυσκολίες ολοκλήρωσης προέκυπταν μέχρι τώρα από συντακτικές διαφορές των πηγών αυτών. Η πρακτική των Συνδεδεμένων Δεδομένων (βλέπε και παράγραφο 2.2.5), σύμφωνα με την οποία οι δημιουργοί RDF δεδομένων ενθαρρύνονται να χρησιμοποιούν ήδη υπάρχουσες δημοφιλείς οντολογίες, να ορίζουν συνδέσμους του συνόλου δεδομένων τους με άλλα δημοσιευμένα σύνολα δεδομένων καθώς και να επαναχρησιμοποιούν αναγνωριστικά που αναφέρονται στην ίδια οντότητα του πραγματικού κόσμου, διευκολύνει ακόμα περισσότερο την καθολική ολοκλήρωση δεδομένων, ανεξαρτήτως της προέλευσης των τελευταίων. Δεδομένης της ραγδαίας αποδοχής και ανάπτυξης του κινήματος των Συνδεδεμένων Δεδομένων και των πρακτικών του, που έχει οδηγήσει στη δημοσίευση μεγάλου όγκου RDF δεδομένων (της τάξης των δεκάδων δισεκατομμυρίων RDF προτάσεων) σε διάφορες θεματικές περιοχές τα τελευταία χρόνια, τα αναμενόμενα οφέλη από τον συνδυασμό όλων αυτών των δεδομένων με δεδομένα που βρίσκονται αποθηκευμένα σε σχεσιακές ΒΔ είναι ανυπολόγιστα, όπως εξάλλου και ο αριθμός των πιθανών εφαρμογών που μπορούν να εκμεταλλευτούν αυτή τη συνδυασμένη γνώση.

3.2 Βασική προσέγγιση

Όλες οι μέθοδοι που εξετάζουν τη σύμπραξη σχεσιακών βάσεων δεδομένων με μοντέλα αναπαράστασης γνώσης του Σημασιολογικού Ιστού και οι οποίες θα αναλυθούν διεξοδικά στην ενότητα 3.3, ορίζουν ή υποθέτουν μια συνάρτηση αντιστοιχίας μεταξύ των δομικών στοιχείων του *σχεσιακού μοντέλου* (παράγραφος 2.1.2) ή του *μοντέλου ΟΣ* (παράγραφος 2.1.1) και μηχανισμών

του *RDF μοντέλου* (παράγραφος 2.2.1) ή κάποιας γλώσσας αναπαράστασης γνώσης (παράγραφος 2.2.2). Στην τρέχουσα ενότητα, παρουσιάζουμε την πιο απλή προσέγγιση αντιστοιχίας μεταξύ του σχεσιακού και του RDF μοντέλου, η οποία ήταν και η πρώτη που εμφανίστηκε στη βιβλιογραφία [35], λειτουργώντας ως βάση για τις σχετικές μεθοδολογίες και προσεγγίσεις που ακολούθησαν. Η συγκεκριμένη μέθοδος επεκτάθηκε στη συνέχεια ώστε να συμπεριλάβει και RDFS οντολογίες [30] και είναι ανεπίσημα γνωστή ως προσέγγιση αντιστοιχίας «πίνακα με κλάση και στήλης με κατηγόρημα», αλλά για λόγους συντομίας θα αναφερόμαστε σε αυτή ως *βασική προσέγγιση*. Σύμφωνα με αυτή:

1. Μία σχέση R αντιστοιχεί σε μια (RDFS ή OWL) κλάση $C(R)$.
2. Μια πλειάδα μιας σχέσης R αντιστοιχεί σε έναν RDF κόμβο που αποτελεί στιγμιότυπο της κλάσης $C(R)$.
3. Ένα γνώρισμα *att* μιας σχέσης R αντιστοιχεί σε μια RDF ιδιότητα $P(att)$.
4. Για μια πλειάδα $R[t]$ μιας σχέσης R, η τιμή ενός γνωρίσματος *att* αντιστοιχεί στην τιμή της ιδιότητας $P(att)$ για τον RDF κόμβο που αντιστοιχεί στην πλειάδα $R[t]$.



Σχήμα 3.1: *Παράδειγμα βασικής προσέγγισης αντιστοιχίας σχέσης σε RDF γράφο*

Ένα απλό παράδειγμα εφαρμογής της βασικής προσέγγισης σε μια πλειάδα μιας σχέσης απεικονίζεται στο σχήμα 3.1.

Τα 4 παραπάνω βασικά σημεία έχουν αποτελέσει τον πυρήνα αρκετών μεθόδων που αντιστοιχούν το σχεσιακό ή το μοντέλο ΟΣ με εκφραστικές οντολογικές γλώσσες, όπως η OWL. Τέτοιες μέθοδοι θα περιγραφούν στην παράγραφο 3.4.2.2. Βέβαια, ακόμα και για την περίπτωση αντιστοιχίας μιας σχεσιακής ΒΔ με έναν RDF γράφο, οι γενικοί αυτοί κανόνες δεν επαρκούν, καθώς χρειάζεται παράλληλα να οριστούν κανόνες για την αντιστοιχία μεταξύ στοιχείων της ΒΔ και των IRI αναγνωριστικών των στοιχείων του RDF γράφου. Μια τέτοια αντιστοιχία έχει υποτεθεί και στο σχήμα 3.1. Η συγκεκριμένη αντιστοιχία θα πρέπει να είναι μια 1-1 συνάρτηση, έτσι ώστε να εξασφαλίζεται ότι για κάθε στοιχείο της ΒΔ παράγεται ένα μοναδικό IRI αλλά και το αντίστροφο, δηλαδή κάθε IRI να περιέχει την απαραίτητη πληροφορία για την

αναγνώριση του στοιχείου ή της τιμής της ΒΔ που αυτό προσδιορίζει. Ο καθορισμός κανόνων παραγωγής IRI είναι ιδιαίτερα σημαντικός για συστήματα που δημιουργούν έναν RDF γράφο που περιγράφει τα περιεχόμενα μιας ΒΔ (παράγραφος 3.4.2.1).

Μέχρι τον πρόσφατο ορισμό της Άμεσης Αντιστοιχίας από το W3C (βλέπε και παράγραφο 2.2.4), δεν υπήρχε ένα κοινά αποδεκτό σύνολο κανόνων παραγωγής IRI από μια σχεσιακή ΒΔ παρά μόνο μεμονωμένες προτάσεις που όλες ακολουθούσαν κάποια παραλλαγή του ιεραρχικού προτύπου που παρουσιάζεται στο [35]. Οι κανόνες παραγωγής IRI της Άμεσης Αντιστοιχίας, οι οποίοι παρουσιάζονται στον πίνακα 3.1 αποκλίνουν ελάχιστα από αυτό το πρότυπο.

Πίνακας 3.1: *Κανόνες παραγωγής IRI Άμεσης Αντιστοιχίας*

Στοιχείο ΒΔ	*Υπόδειγμα IRI*	*Παράδειγμα*
Σχέση *rel*	$\{base\_IRI\}/\{rel\}$	http://example.org/emp
Γνώρισμα *att*	$\{base\_IRI\}/\{rel\}\#\{attr\}$	http://example.org/emp#name
Σύνθετο ξένο κλειδί με γνωρίσματα $att_1, \ldots, att_n$	$\{base\_IRI\}/\{rel\}\#ref\text{-}\{att_1\};\ldots;\{att_n\}$	http://example.org/emp#ref-deptName;deptCity
Πλειάδα με γνωρίσματα-κλειδιά $pk_1, \ldots, pk_n$ και τιμές $pk_1\_val, \ldots, pk_n\_val$	$\{base\_IRI\}/\{rel\}/\{pk_1\}=\{pk_1\_val\};\ldots;\{pk_n\}=\{pk_n\_val\}$	http://example.org/department/name=accounting;city=Oxford

3.3 Μια ταξινόμηση των προσεγγίσεων

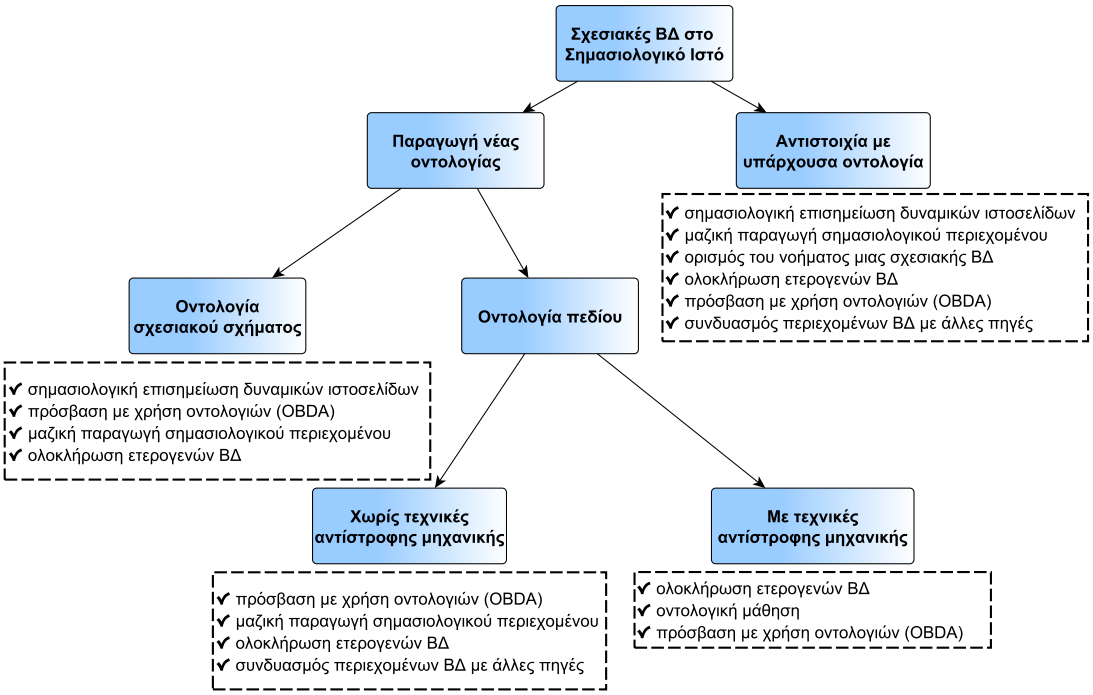
Όπως αναφέρθηκε και στην αρχή αυτού του κεφαλαίου, ο όρος *«αντιστοιχία βάσης δεδομένων με οντολογία»* αναφέρεται χωρίς αυστηρότητα στη σχετική βιβλιογραφία, περικλείοντας ποικίλες προσεγγίσεις και λύσεις σε διακριτά προβλήματα. Στην τρέχουσα ενότητα, παρέχουμε μια ταξινόμηση, απαραίτητη για την ορθή κατηγοριοποίηση και περαιτέρω ανάλυση των μεθόδων που έχουν προταθεί μέχρι σήμερα. Εκτός από τα κριτήρια που λαμβάνουμε υπόψη για την ταξινόμηση, παρουσιάζουμε επίσης και κάποιες περιγραφικές παραμέτρους που χαρακτηρίζουν κάθε μέθοδο.

Στη σχετική βιβλιογραφία, έχουν ήδη προταθεί διάφορα σχήματα ταξινόμησης και περιγραφικές παράμετροι για μεθοδολογίες αντιστοιχίας ΒΔ σε οντολογία. Η διάκριση μεταξύ *κριτηρίων ταξινόμησης* και *περιγραφικών παραμέτρων* συχνά δεν είναι σαφής. Μεγέθη τα οποία λειτουργούν ως κριτήρια ταξινόμησης πρέπει να λαμβάνουν έναν, ιδανικά, μικρό αριθμό τιμών, διαχωρίζοντας τις προσεγγίσεις σε μη επικαλυπτόμενες κατηγορίες, περιορισμός που δεν είναι απαραίτητο να ισχύει για τις περιγραφικές παραμέτρους, η φύση των οποίων μπορεί να είναι και ποιοτική. Οι σχετικές μελέτες που έχουν πραγματοποιηθεί συμφωνούν σε μεγάλο βαθμό ως προς τις σημαντικότερες παραμέτρους και κριτήρια που χαρακτηρίζουν τις μεθόδους αντιστοιχίας ΒΔ με οντολογία. Αναλυτικός πίνακας με τα κριτήρια που θεωρεί κάθε μία από αυτές τις μελέτες υπάρχει στη δημοσίευση [182].

Η ταξινομία που προτείνουμε και την οποία θα ακολουθήσουμε στην επισκόπησή μας στο παρόν κεφάλαιο χρησιμοποιεί αρκετά από τα κριτήρια που

έχουν αναφερθεί σε σχετικές εργασίες [18, 28, 87, 97, 167, 174, 207] αλλά και στη δημοσίευση [115], με τέτοιο τρόπο ώστε αυτή να είναι *ολική* και οι κλάσεις που περιέχει *αμοιβαία αποκλειόμενες*. Με άλλα λόγια, ο χώρος των προβλημάτων που εμπίπτουν στο γενικότερο ζήτημα της αντιστοιχίας ΒΔ με οντολογία χωρίζεται εδώ σε σαφώς διακριτές υποκατηγορίες που περιέχουν ομοειδείς προσεγγίσεις. Οι λίγες εξαιρέσεις προσεγγίσεων που συναντώνται αφορούν σε σχετικά εργαλεία λογισμικού που ενσωματώνουν περισσότερους από έναν τρόπους λειτουργίας, με καθέναν από αυτούς να εμπίπτει σε διαφορετική υποκατηγορία.

Η κατηγοριοποίηση των μεθοδολογιών που ασχολούνται με το πρόβλημα της αντιστοιχίας ΒΔ με οντολογίας φαίνεται στο σχήμα 3.2. Δίπλα σε κάθε κλάση προσεγγίσεων, σημειώνουμε τους στόχους και τα σημαντικότερα πιθανά οφέλη που προκύπτουν από την εφαρμογή της, όπως αυτά παρουσιάστηκαν στην ενότητα 3.1.



Σχήμα 3.2: *Ταξινομία μεθόδων αντιστοιχίας σχεσιακής ΒΔ με οντολογία*

Η πρώτη διάκριση μεταξύ των μεθόδων αφορά στο αν μια υπάρχουσα εξωτερική οντολογία *απαιτείται* για την εφαρμογή της μεθόδου. Επομένως, το πρώτο κριτήριο ταξινόμησης που εφαρμόζουμε είναι αυτό της **ύπαρξης μιας οντολογίας** ως προαπαιτούμενο για την διαδικασία αντιστοιχίας, διακρίνοντας μεταξύ μεθόδων που δημιουργούν μια νέα οντολογία από μια σχεσιακή ΒΔ (ενότητα 3.4) και μεθόδων που ορίζουν αντιστοιχίες μεταξύ μιας σχεσιακής ΒΔ και μιας δεδομένης υπάρχουσας οντολογίας (ενότητα 3.5). Στη δεύτερη περίπτωση, η υπάρχουσα οντολογία θα πρέπει να περιγράφει μια θεματική περιοχή συμβατή με τα περιεχόμενα της ΒΔ, έτσι ώστε οι αντιστοιχίες που θα οριστούν να έχουν νόημα. Αυτός είναι και ο λόγος που, σε αυτή την κατηγορία μεθόδων, η χρησιμοποιούμενη οντολογία επιλέγεται από έναν ανθρώπινο χρήστη με γνώση της σημασίας των δεδομένων που είναι αποθηκευμένα στη ΒΔ. Αντίθετα, στην περίπτωση τεχνικών που δημιουργούν μια νέα οντολογία,

η ύπαρξη μιας εξωτερικής οντολογίας πεδίου δεν είναι απαραίτητη. Τέτοιες μέθοδοι χρησιμοποιούνται σε περιπτώσεις όπου δεν υπάρχει διαθέσιμη οντολογία για τη θεματική περιοχή που καλύπτεται από τη ΒΔ ή ακόμα και σε περιπτώσεις όπου ο χρήστης βασίζεται στην εφαρμογή της μεθόδου προκειμένου να ανακαλύψει την, άγνωστη σε αυτόν, σημασία των περιεχομένων της ΒΔ.

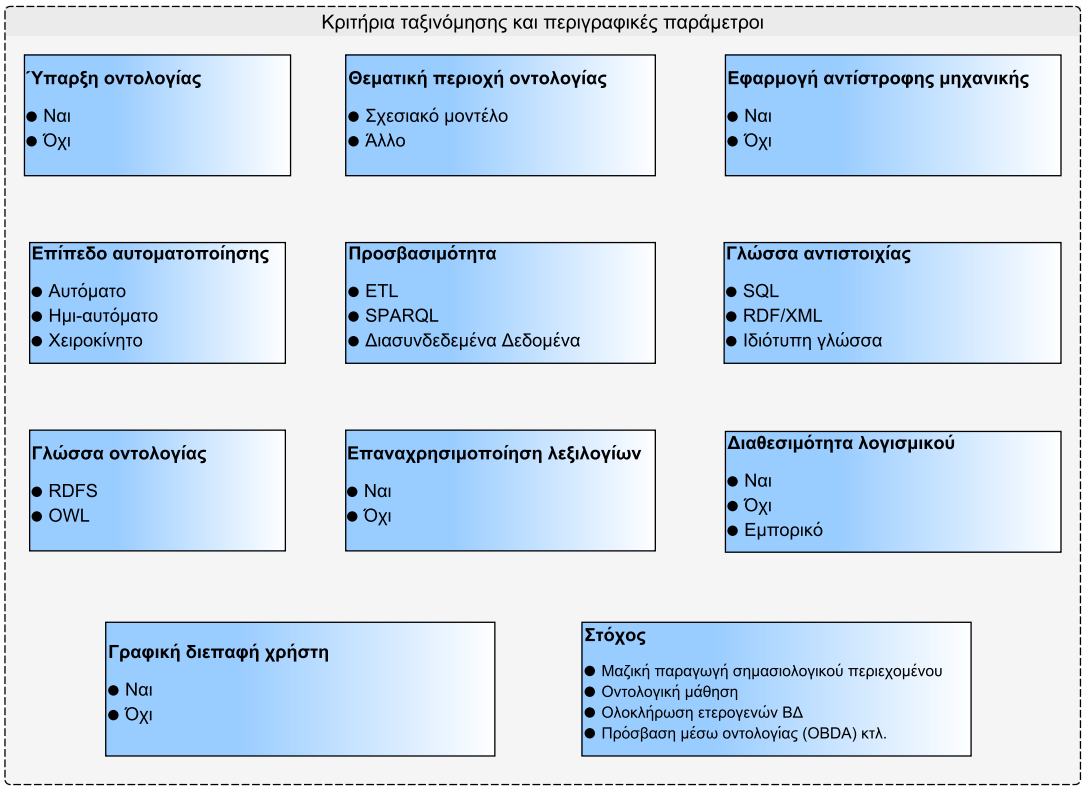
Η κλάση των μεθόδων που παράγουν μια νέα οντολογία διαχωρίζεται περαιτέρω σε δύο υποκατηγορίες, ανάλογα με τη **θεματική περιοχή της παραγόμενης οντολογίας**. Αφενός, υπάρχουν μέθοδοι που κατασκευάζουν οντολογίες που περιγράφουν το σχεσιακό μοντέλο (οι οποίες αναλύονται στην παράγραφο 3.4.1). Σε αυτή την περίπτωση, η παραγόμενη οντολογία αποτελείται από έννοιες και συσχετίσεις που περιγράφουν τους μηχανισμούς του σχεσιακού μοντέλου, αντικατοπτρίζοντας πλήρως τη δομή της σχεσιακής ΒΔ. Θα αναφερόμαστε σε μια οντολογία αυτού του είδους με τον όρο *«οντολογία σχεσιακού σχήματος»* (database schema ontology). Δεδομένου του ότι μια οντολογία σχεσιακού σχήματος δεν περιέχει γνώση από κάποιο άλλο γνωστικό πεδίο, οι μέθοδοι παραγωγής τέτοιων οντολογιών είναι σε μεγάλο βαθμό αυτοματοποιημένες, καθώς βασίζονται μόνο στη γνώση του σχήματος μιας ΒΔ. Αφετέρου, υπάρχουν μέθοδοι που παράγουν μια οντολογία πεδίου (domain ontology), η οποία αναφέρεται στον τομέα γνώσης που περιγράφεται από τα περιεχόμενα της ΒΔ (βλέπε παράγραφο 3.4.2).

Το τελευταίο κριτήριο ταξινόμησης που θεωρούμε είναι η **εφαρμογή τεχνικών αντίστροφης μηχανικής** του εννοιολογικού σχήματος μιας ΒΔ (database reverse engineering) κατά τη διαδικασία παραγωγής μιας νέας οντολογίας πεδίου. Προσεγγίσεις που χρησιμοποιούν αντίστροφη μηχανική προσπαθούν να ανακτήσουν το αρχικό εννοιολογικό σχήμα της ΒΔ από το σχεσιακό σχήμα και να το μετατρέψουν σε μια οντολογία εκφρασμένη σε κάποια γλώσσα αναπαράστασης γνώσης. Προσεγγίσεις αυτής της μορφής αναλύονται στην παράγραφο 3.4.2.2. Αντίθετα, μια άλλη κατηγορία μεθόδων χρησιμοποιεί, αντί για τεχνικές αντίστροφης μηχανικής, κάποιους απλούς κανόνες μετάφρασης από το σχεσιακό στο RDF μοντέλο (παραλλαγές της βασικής προσέγγισης που αναφέρθηκε στην ενότητα 3.2) και επαφίεται σε κάποιον ανθρώπινο χρήστη-ειδικό του χώρου για τον ορισμό πιο σύνθετων αντιστοιχιών και τον εμπλουτισμό της τελικής οντολογίας. Μέθοδοι αυτής της κατηγορίας εξετάζονται στην παράγραφο 3.4.2.1.

Παρατηρώντας το σχήμα 3.2, μπορούμε να δούμε ότι δεν υπάρχει μεγάλος βαθμός συσχέτισης μεταξύ των κλάσεων της ταξινομίας και των κινήτρων και στόχων της ενότητας 3.1. Αυτό ισχύει, επειδή η ταξινομία που αναλύουμε κατηγοριοποιεί τις μεθόδους με βάση τη *μορφή* της αντιστοιχίας ΒΔ και οντολογίας και τις *τεχνικές* που εφαρμόζονται προκειμένου να παραχθεί μια αντιστοιχία. Αντίθετα, τα περισσότερα οφέλη αναφέρονται στις *εφαρμογές* όπου χρησιμοποιείται μια αντιστοιχία (π.χ. ολοκλήρωση ετερογενών ΒΔ, επισημείωση ιστοσελίδων), οι οποίες δεν εξαρτώνται από τις λεπτομέρειες της διαδικασίας ορισμού της αντιστοιχίας. Εξαιρέσεις στην παραπάνω παρατήρηση αποτελούν οι στόχοι της οντολογικής μάθησης και του ορισμού του νοήματος ενός σχεσιακού σχήματος, οι οποίοι σαφώς και εξ ορισμού συσχετίζονται με συγκεκριμένες κλάσεις προσεγγίσεων, ο πρώτος με την κατηγορία μεθόδων που παράγουν μια νέα οντολογία αναλύοντας τη σχεσιακή ΒΔ και ο δεύτε-

ρος με την κατηγορία μεθόδων που αντιστοιχεί δομές της ΒΔ με έννοιες και συσχετίσεις μιας υπάρχουσας οντολογίας.

Δεδομένης της ταξινομίας του σχήματος 3.2 και των τριών κριτηρίων που χρησιμοποιούνται σε αυτή, διαλέγουμε τις σημαντικότερες εκ των περιγραφικών παραμέτρων που αναφέρονται στη σχετική βιβλιογραφία για τον χαρακτηρισμό των μεθόδων που αναφέρονται σε αυτό το κεφάλαιο. Τα κριτήρια ταξινόμησης και οι περιγραφικές παράμετροι που χρησιμοποιούμε απεικονίζονται συγκεντρωτικά στο σχημα 3.3 και αναλύονται στη συνέχεια:



Σχήμα 3.3: *Κριτήρια ταξινόμησης και περιγραφικές παράμετροι για μεθόδους αντιστοιχίας ΒΔ με οντολογία*

Επίπεδο αυτοματοποίησης. Η συγκεκριμένη παράμετρος εκφράζει το βαθμό συμμετοχής του ανθρώπινου χρήστη στη διαδικασία αντιστοιχίας. Οι πιθανές τιμές που μπορεί αυτή να λάβει είναι *αυτόματη*, *ημι-αυτόματη* και *χειροκίνητη*. Μέθοδοι που είναι αυτόματες δεν εμπλέκουν ανθρώπινους χρήστες, ενώ ημι-αυτόματες μέθοδοι απαιτούν κάποια είσοδο από το χρήστη. Αυτή η είσοδος μπορεί να είναι απολύτως απαραίτητη για την λειτουργία της μεθόδου ή και προαιρετική, χρήσιμη για την επαλήθευση και εμπλουτισμό του τελικού αποτελέσματος της διαδικασίας. Στην περίπτωση χειροκίνητων προσεγγίσεων, η αντιστοιχία ορίζεται εξ ολοκλήρου από τον ανθρώπινο χρήστη, χωρίς κάποια βοήθεια ή πρόταση από την εφαρμογή. Σε αρκετές περιπτώσεις, το επίπεδο αυτοματοποίησης είναι μια μεταβλητή που χαρακτηρίζει μια ολόκληρη κλάση μεθόδων (ή τουλάχιστον την πλειοψηφία των μεθόδων που ανήκουν σε αυτές). Για παράδειγμα, μέθοδοι που παράγουν μια «οντολογία σχεσιακού σχήματος» είναι πλήρως αυτόματες, ενώ αντίθετα η πλειοψηφία των προσεγγίσεων που

αντιστοιχούν μια σχεσιακή ΒΔ με μια υπάρχουσα οντολογία είναι χειροκίνητες, δεδομένης της εγγενούς δυσκολίας του προβλήματος.

Προσβασιμότητα του αποτελέσματος της αντιστοιχίας. Αυτή η παράμετρος εξετάζει τον τρόπο με τον οποίο προσπελαύνεται το αποτέλεσμα της αντιστοιχίας (data accessibility). Οι πιθανές τιμές αυτής της παραμέτρου είναι *ETL* (Extract, Transform, Load), *SPARQL* ή άλλη γλώσσα ερωτημάτων για RDF γράφους και *Συνδεδεμένα Δεδομένα*. Ο όρος ETL είναι γνωστός από το χώρο των αποθηκών δεδομένων (data warehouses) και αναφέρεται σε μια διαδικασία, όπου κατά σειρά πραγματοποιείται εξαγωγή δεδομένων από εξωτερικές πηγές, επεξεργασία τους ώστε να πληρούν ορισμένες προϋποθέσεις και τελική τους φόρτωση στην αποθήκη δεδομένων. Στο πλαίσιο ενός συστήματος αντιστοιχίας ΒΔ με οντολογία, ο όρος αυτός περιγράφει μια αντίστοιχη διαδικασία όπου τα δεδομένα εξάγονται από τη σχεσιακή ΒΔ, μετατρέπονται σε RDF γράφο ή οντολογία ακολουθώντας μια ορισμένη αντιστοιχία και εν τέλει, φορτώνονται σε ένα μέσο αποθήκευσης για RDF (το οποίο συνήθως είναι ένα triple store σύστημα, αλλά μπορεί να είναι και ένα απλό RDF αρχείο). Στην περίπτωση αυτή, λέμε ότι το αποτέλεσμα της αντιστοιχίας είναι *υλοποιημένο* (materialized). Συνώνυμος του όρου ETL στο συγκεκριμένο νοηματικό πλαίσιο θεωρείται και ο όρος *μαζική εξαγωγή* (massive dump ή batch transformation).

Ο δεύτερος τρόπος πρόσβασης στο αποτέλεσμα της αντιστοιχίας είναι μέσω SPARQL ή κάποιας άλλης γλώσσας σημασιολογικών ερωτημάτων. Σε αυτή την περίπτωση, μόνο ένα μέρος του συνολικού αποτελέσματος της αντιστοιχίας προσπελαύνεται (συγκεκριμένα, μόνο η απάντηση στο τιθέμενο ερώτημα), έτσι ώστε να μη χρειάζεται κάποιο εξωτερικό μέσο αποθήκευσης, καθώς τα περιεχόμενα της ΒΔ δεν επαναλαμβάνονται υπό τη μορφή ενός υλοποιημένου RDF γράφου. Αντίθετα, το σημασιολογικό ερώτημα μετατρέπεται σε ένα SQL ερώτημα, το οποίο εκτελείται στη ΒΔ και τα αποτελέσματα του οποίου μετατρέπονται στη μορφή που επιβάλλεται να έχουν τα αποτελέσματα της σημασιολογικής γλώσσας ερωτημάτων. Ο συγκεκριμένος τρόπος πρόσβασης είναι γνωστός και ως «βασισμένος σε ερωτήματα» (query driven) ή και «κατά παραγγελία» (access on demand) και συναντάται σε OBDA (ontology-based data access) συστήματα.

Ένας τρίτος τρόπος πρόσβασης στο αποτέλεσμα μιας αντιστοιχίας είναι αυτός που ακολουθεί τις πρακτικές δημοσίευσης των Συνδεδεμένων Δεδομένων, παρέχοντας *πρόσβαση σε επίπεδο οντότητας* (entity-level access). Σύμφωνα με αυτές, όλα τα IRIs ενός RDF γράφου ακολουθούν τους κανόνες ονοματοδοσίας του HTTP πρωτοκόλλου και ένα HTTP GET αίτημα για κάποιο από αυτά τα IRIs θα επιστρέψει κάποιου είδους πληροφορία για τον πόρο στον οποίο αναφέρεται[2].

Η παράμετρος της προσβασιμότητας συνδέεται άμεσα και με το μέγεθος του *συγχρονισμού των δεδομένων* (data synchronization), σύμφωνα με το οποίο οι προσεγγίσεις διακρίνονται σε *στατικές* και *δυναμικές*, ανάλογα με το αν η

[2]Η χρήση του όρου Συνδεδεμένα Δεδομένα μπορεί να είναι ελαφρώς παραπλανητική, καθώς τόσο η μαζική ETL εξαγωγή όσο και η πρόσβαση μέσω SPARQL θεωρούνται εναλλακτικοί τρόποι δημοσίευσης Συνδεδεμένων Δεδομένων [95]. Εντούτοις, ακολουθούμε το παράδειγμα των [18, 97, 167] όπου με τον όρο Συνδεδεμένα Δεδομένα περιγράφεται ένα σύστημα RDF πρόσβασης βασισμένο σε ανακτήσιμα IRIs μέσω HTTP.

απεικόνιση των περιεχομένων της ΒΔ σε RDF γράφο ή οντολογία εκτελείται μόνο μία φορά ή για κάθε εισερχόμενο ερώτημα, αντίστοιχα. Τα δύο σημαντικότερα πλεονεκτήματα που προσφέρουν οι δυναμικές προσεγγίσεις έναντι των στατικών είναι: α) το γεγονός ότι το λαμβανόμενο αποτέλεσμα της αντιστοιχίας συμφωνεί πάντα με το τρέχον περιεχόμενο της ΒΔ και β) οι μειωμένες απαιτήσεις των πρώτων σε αποθήκευση καθώς δεν απαιτείται η φυσική επανάληψη των περιεχομένων της ΒΔ. Αυτά τα πλεονεκτήματα καθιστούν τις δυναμικές προσεγγίσεις προτιμητέες ιδιαίτερα όταν ο ρυθμός μεταβολής των περιεχομένων της ΒΔ είναι υψηλός ή/και όταν ο όγκος τους είναι απαγορευτικός για να υλοποιηθούν σε RDF μορφή. Εντούτοις, η χρήση ενός συστήματος στατικής εφαρμογής μιας αντιστοιχίας μπορεί να είναι περισσότερο συμφέρουσα σε περιπτώσεις όπου το παραγόμενο SQL ερώτημα περιέχει μεγάλο αριθμό ενώσεων ή σε περιπτώσεις όπου η διαδικασία επανεγγραφής σε SQL αποδεικνύεται αρκετά χρονοβόρα, όπως όταν απαιτείται η εφαρμογή συλλογισμού στο αποτέλεσμα μιας αντιστοιχίας.

Γλώσσα αντιστοιχίας. Ένα άλλο χαρακτηριστικό των μεθόδων που εξετάζονται είναι η γλώσσα στην οποία εκφράζεται η αντιστοιχία μεταξύ μιας σχεσιακής ΒΔ και μιας οντολογίας ή RDF γράφου. Η συγκεκριμένη παράμετρος παρουσιάζει μεγάλη μεταβλητότητα στις τιμές τις οποίες λαμβάνει, δεδομένου του ότι, μέχρι πρόσφατα, δεν υπήρχε κάποια καθιερωμένη γλώσσα για την περιγραφή τέτοιων αντιστοιχιών με αποτέλεσμα η πλειοψηφία των συστημάτων να χρησιμοποιούν μια δική τους ιδιότυπη γλώσσα. Μόλις ολοκληρωθεί η προτυποποίηση της R2RML (παράγραφος 2.2.4), αναμένεται να υιοθετηθεί από την πλειοψηφία των ήδη ανεπτυγμένων αλλά και νέων συστημάτων αντιστοιχίας[3]. Η παράμετρος της γλώσσας αντιστοιχίας είναι εφαρμόσιμη σε μεθόδους που χρειάζεται να επαναχρησιμοποιήσουν μια ορισμένη αντιστοιχία. Ένα αντι-παράδειγμα αποτελούν οι προσεγγίσεις που παράγουν μια νέα οντολογία πεδίου. Προσεγγίσεις αυτού του είδους συνήθως δεν αναπαριστούν την αντιστοιχία σε κάποια μορφή, καθώς στόχος τους είναι η παραγωγή μιας οντολογίας και ως εκ τούτου, η αποθήκευση των συσχετίσεων με στοιχεία της ΒΔ έχει δευτερεύουσα σημασία για αυτές.

Γλώσσα οντολογίας. Αυτή η παράμετρος αναφέρει τη γλώσσα αναπαράστασης γνώσης που αποτελεί το στόχο του συστήματος αντιστοιχίας. Ανάλογα με το είδος της μεθόδου, μπορεί να αναφέρεται στη γλώσσα της οντολογίας που θα παραχθεί ή στη γλώσσα της οντολογίας που απαιτείται ως είσοδος για τη λειτουργία της μεθόδου. Καθώς η πλειοψηφία των μεθόδων που εξετάστηκαν εντάσσονται στο πλαίσιο του Σημασιολογικού Ιστού, οι οντολογικές γλώσσες που συναντώνται είναι η RDFS και η OWL.

Επαναχρησιμοποίηση λεξιλογίων. Η συγκεκριμένη μεταβλητή δηλώνει την ικανότητα της μεθόδου να αντιστοιχεί τα περιεχόμενα μιας ΒΔ σε περισσότερες της μίας υπάρχουσες οντολογίες. Αυτό ισχύει κυρίως σε περιπτώσεις χειροκίνητων μεθόδων όπου ο χρήστης έχει την ελευθερία να χρησιμοποιεί όρους από υπάρχουσες οντολογίες κατά βούληση. Η επαναχρησιμοποίηση όρων από

[3] Υπάρχουν ήδη ενθαρρυντικές ενδείξεις για το βαθμό υιοθέτησης της R2RML από την κοινότητα του Σημασιολογικού Ιστού, όπως φαίνεται και από τη λίστα συστημάτων που υλοποιούν τις προδιαγραφές της: http://www.w3.org/TR/rdb2rdf-implementations/.

δημοφιλή λεξιλόγια του είναι μία από τις προϋποθέσεις επιτυχίας του Σημασιολογικού Ιστού και συνεπώς, ένα σημαντικό χαρακτηριστικό προσεγγίσεων αντιστοιχίας σχεσιακών ΒΔ σε οντολογία. Είναι σημαντικό να τονιστεί ότι οι μέθοδοι που επιτρέπουν την επαναχρησιμοποίηση λεξιλογίων δεν *υποχρεώνουν* το χρήστη να ακολουθήσει αυτή την τακτική και επομένως, η συγκεκριμένη μεταβλητή δεν πρέπει να συγχέεται με την *απαίτηση ύπαρξης εξωτερικής οντολογίας* που χρησιμοποιείται ως κριτήριο ταξινόμησης στο σχήμα 3.2. Συχνά, η τιμή της παραμέτρου μπορεί να είναι κοινή μεταξύ όλων των μεθόδων που ανήκουν σε μια κλάση: για παράδειγμα, οι προσεγγίσεις που παράγουν μια νέα οντολογία σχεσιακού σχήματος δεν επιτρέπουν άμεσα την επαναχρησιμοποίηση εξωτερικών λεξιλογίων (αυτό βέβαια, μπορεί να γίνει σε μεταγενέστερο στάδιο), καθώς οι χρησιμοποιούμενοι όροι είναι προκαθορισμένοι από την εκάστοτε μέθοδο και δεν υπόκεινται σε αλλαγή.

Διαθεσιμότητα λογισμικού. Αυτή η παράμετρος δηλώνει αν μια προσέγγιση συνοδεύεται και από αντίστοιχο ελεύθερα διαθέσιμο λογισμικό, επιτρέποντας με αυτόν τον τρόπο τη διάκριση μεταξύ καθαρά θεωρητικών μεθοδολογιών που (μπορεί να) συνοδεύονται από κάποια μη διαθέσιμη ερευνητική υλοποίηση και ελεύθερων εργαλείων που προσφέρουν πρακτικές λύσεις στο πρόβλημα της αντιστοιχίας σχεσιακής ΒΔ με οντολογία. Περιπτώσεις εμπορικού λογισμικού, του οποίου η πλήρης λειτουργικότητα δεν είναι ελεύθερα διαθέσιμη στο ευρύ κοινό, σημειώνονται με αντίστοιχη ένδειξη.

Γραφική διεπαφή χρήστη. Το χαρακτηριστικό αυτό είναι εφαρμόσιμο μόνο σε μεθόδους που συνοδεύονται από αντίστοιχες υλοποιήσεις λογισμικού και δηλώνει τη δυνατότητα του χρήστη να αλληλεπιδράσει με το σύστημα μέσω μιας γραφικής διεπαφής. Η παράμετρος αυτή έχει ιδιαίτερη σημασία για χρήστες με ελάχιστη εμπειρία και εξοικείωση με τεχνολογίες Σημασιολογικού Ιστού, καθώς μια γραφική διεπαφή μπορεί να καθοδηγήσει τον τελικό χρήστη στα διάφορα βήματα της διαδικασίας αντιστοιχίας, καθώς και να παρέχει συμβουλευτικές προτάσεις.

Στόχος. Η τελευταία περιγραφική παράμετρος αναφέρει τον κύριο στόχο της μεθόδου, όπως αυτός αναφέρεται από τους συγγραφείς της. Αυτό, βέβαια, δε σημαίνει ότι ο συγκεκριμένος στόχος είναι ο μοναδικός και ότι δεν είναι δυνατή η εφαρμογή της συγκεκριμένης μεθόδου σε κάποιο άλλο πλαίσιο. Συνήθως, οι ωφέλειες και τα κίνητρα που αναφέρονται στο σχήμα 3.2 ισχύουν για όλα τα εργαλεία και τις προσεγγίσεις που ανήκουν στην ίδια κλάση.

Σε αυτό το σημείο, αξίζει να τονιστεί ότι καθένα από τα παραπάνω χαρακτηριστικά θα μπορούσε να είχε χρησιμοποιηθεί ως κριτήριο ταξινόμησης για τον ορισμό μιας εναλλακτικής ταξινομίας σε σχέση με αυτή του σχήματος 3.2. Παρ' όλα αυτά, υποστηρίζουμε ότι η επιλογή των συγκεκριμένων κριτηρίων οδηγεί σε μια διαμέριση του χώρου του δεδομένου προβλήματος αντιστοιχίας σε επιμέρους ομοειδείς κλάσεις προσεγγίσεων που επιτρέπουν την μεθοδική ανάλυσή τους. Παραδείγματος χάριν, αρκετοί συγγραφείς [18, 28, 87, 167] κατηγοριοποιούν μεθόδους αντιστοιχίας, με βάση το κριτήριο του συγχρονισμού δεδομένων, σε στατικές και δυναμικές. Αυτός ο διαχωρισμός όμως δεν είναι πλήρης, καθώς αγνοεί μεθόδους που ανακαλύπτουν αντιστοιχίες ανάμεσα σε μια σχεσιακή ΒΔ και μια οντολογία, χωρίς ούτε να φορτώνουν δεδομένα από

τη ΒΔ σε μια νέα οντολογία ούτε να προσφέρουν πρόσβαση στα δεδομένα της ΒΔ μέσω ενός οντολογικού ερωτήματος. Αυτός είναι ο κυριότερος λόγος που δεν συμπεριλαμβάνουμε τη συγκεκριμένη διάκριση στην ταξινόμηση που θεωρούμε στο παρόν κεφάλαιο. Η συγκεντρωτική κατάσταση με τις τιμές των περιγραφικών παραμέτρων για όλες τις μεθόδους που εξετάστηκαν παρατίθεται στον πίνακα 3.6.

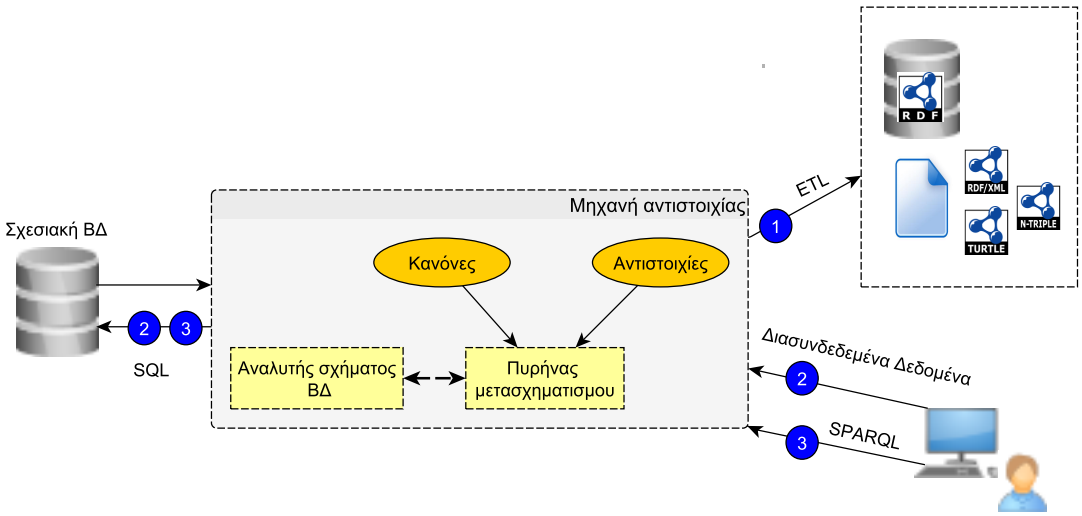
Στις ενότητες που ακολουθούν, εξετάζουμε ξεχωριστά κάθε κατηγορία μεθόδων της ταξινόμησης του σχήματος 3.2. Για κάθε τέτοια κατηγορία, συνοψίζουμε τα χαρακτηριστικά της με αναφορά στις περιγραφικές παραμέτρους της τρέχουσας ενότητας και περιγράφουμε σύντομα την κατεύθυνση που ακολουθείται από την πλειοψηφία των εργαλείων και μεθόδων που ανήκουν σε αυτή. Παράλληλα, παραθέτουμε συγκεντρωτικούς πίνακες όλων των μεθόδων, στους οποίους αναφέρονται χαρακτηριστικά που έχουν σημασία για την εκάστοτε κατηγορία, ενώ παραλείπονται για οικονομία χώρου οι αναλυτικές περιγραφές όλων των μεθόδων, οι οποίες υπάρχουν στη δημοσίευση [182].

3.4 Παραγωγή οντολογίας από σχεσιακή βάση δεδομένων

Σε αυτή την ενότητα, εξετάζουμε το πρόβλημα της παραγωγής μιας νέας οντολογίας από μια σχεσιακή βάση δεδομένων και του προαιρετικού εμπλουτισμού της πρώτης με άτομα που αντανακλούν τα περιεχόμενα της δεύτερης. Ισοδύναμα, σε όρους Περιγραφικής Λογικής, θα λέγαμε ότι οι αυτές οι μέθοδοι κατασκευάζουν το σώμα ορολογίας (TBox) μιας βάσης γνώσης και κάποιες εξ αυτών προχωρούν και στην κατασκευή του σώματος ισχυρισμών (ABox) με δεδομένα που προέρχονται από τη ΒΔ. Διαγραμματικά, αυτή η διαδικασία απεικονίζεται στο σχήμα 3.4. Ο πυρήνας του συστήματος αντιστοιχίας εξάγει από τη ΒΔ τις αναγκαίες πληροφορίες, πληροφορίες σχετικές με το σχήμα ή/και με τα περιεχόμενά της, και λαμβάνοντας υπόψη χειροκίνητα ορισμένες αντιστοιχίες ή εσωτερικούς ευρετικούς κανόνες, παράγει μια οντολογία η οποία (τουλάχιστον για την περίπτωση των RDFS και OWL) μπορεί να θεωρηθεί ότι είναι συντακτικά ισοδύναμη με έναν RDF γράφο.

Ο παραγόμενος RDF γράφος μπορεί να προσπελαστεί με τρεις τρόπους, όπως είδαμε και στην ενότητα 3.3: μέσω μαζικής εξαγωγής (ETL), SPARQL ή μέσω πρακτικών Συνδεδεμένων Δεδομένων (τρόποι λειτουργίας 1, 3 και 2 αντίστοιχα στο σχήμα 3.4). Στην περίπτωση της μαζικής εξαγωγής, ο RDF γράφος μπορεί να αποθηκευτεί σε ένα αρχείο ή σε ένα triple store σύστημα για τη διατήρηση και περαιτέρω διαχείρισή του, οπότε η πρόσβαση γίνεται κατευθείαν από το επιλεγμένο μέσο αποθήκευσης. Αντίθετα, στις περιπτώσεις πρόσβασης μέσω SPARQL ή Συνδεδεμένων Δεδομένων, ο RDF γράφος παράγεται δυναμικά από τα περιεχόμενα της ΒΔ, μετά από αντίστοιχο SPARQL ή HTTP αίτημα, το οποίο μεταφράζεται σε ένα SQL ερώτημα. Ο πυρήνας του συστήματος αντιστοιχίας διοχετεύει το παραγόμενο SQL ερώτημα στη ΒΔ, όπου εκτελείται, λαμβάνει τα αποτελέσματά του και τα επεξεργάζεται ώστε να δημιουργήσει μια κατάλληλη - SPARQL ή HTTP - απόκριση.

Σε ό,τι αφορά στις αντιστοιχίες και ευρετικούς κανόνες που χρησιμοποιούνται από ένα σύστημα παραγωγής οντολογίας, αυτές στηρίζονται σε μεγάλο



Σχήμα 3.4: *Παραγωγή οντολογίας από σχεσιακή ΒΔ*

βαθμό σε προσαρμογές και επεκτάσεις της βασικής προσέγγισης που σκιαγραφήθηκε στην ενότητα 3.2. Η γενικότητα της βασικής προσέγγισης επιτρέπει την εφαρμογή της σε ένα οποιοδήποτε στιγμιότυπο σχεσιακής ΒΔ, ενώ ένα επιπλέον πλεονέκτημά της είναι το γεγονός ότι συνοδεύεται από μεγάλο βαθμό αυτοματοποίησης, καθώς η μόνη είσοδος που απαιτεί είναι ένα βασικό IRI (*base_IRI* στον πίνακα 3.1) για τη δημιουργία των IRI αναγνωριστικών του RDF γράφου. Παρ' όλα αυτά, η βασική προσέγγιση οδηγεί σε μια σχετικά ακατέργαστη εξαγωγή των περιεχομένων της ΒΔ σε μια απλή οντολογία, η οποία μοιάζει περισσότερο με μια συλλογή όρων παρά με σώμα ορολογίας, δεδομένου του ότι η βασική προσέγγιση δεν οδηγεί στην παραγωγή σύνθετων οντολογικών αξιωμάτων. Επιπλέον, η παραγόμενη οντολογία μοιάζει ουσιαστικά με ένα αντίγραφο του σχεσιακού σχήματος, καθώς όλες οι σχέσεις του τελευταίου έχουν μετατραπεί σε - RDFS ή OWL - κλάσεις, ανεξαρτήτως του αν μοντελοποιούν έννοιες, συσχετίσεις ή άλλες εννοιολογικές δομές. Χαρακτηριστικό παράδειγμα αποτελεί η κλασική περίπτωση N:M συσχετίσεων οι οποίες μοντελοποιούνται ως σχέσεις, όπως είδαμε και στην παράγραφο 2.1.3.

Άλλα αξιοσημείωτα μειονεκτήματα της βασικής προσέγγισης αποτελούν τόσο ο μηχανισμός παραγωγής IRI, σύμφωνα με τον οποίο για κάθε πλειάδα μιας σχέσης παράγεται υποχρεωτικά ένα νέο IRI αναγνωριστικό ακόμα και αν κάποιο ήδη υπάρχον IRI ταιριάζει στη συγκεκριμένη οντότητα, όσο και η υπερβολική χρήση λεκτικών ως τιμών ιδιοτήτων, γεγονός που υποβαθμίζει την ποιότητα του RDF γράφου και δυσχεραίνει τη σύνδεσή του με άλλους RDF γράφους [49]. Παρά τα μειονεκτήματα αυτά, αρκετές μέθοδοι εξειδικεύουν τους κανόνες της βασικής προσέγγισης προσπαθώντας να ανακαλύψουν τη σημασία των δομών ενός σχεσιακού σχήματος και ορίζοντας συνθετότερες αντιστοιχίες μεταξύ του πρώτου και μιας οντολογίας. Τέτοιες μέθοδοι αποτελούν το αντικείμενο των επόμενων παραγράφων.

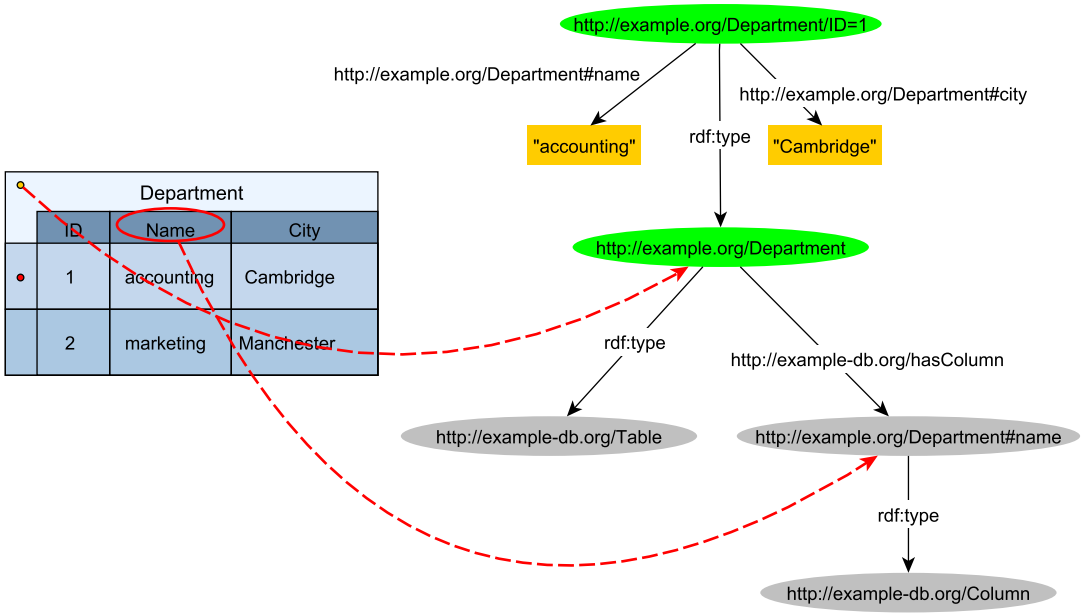
3.4.1 Παραγωγή μιας οντολογίας σχεσιακού σχήματος

Η πρώτη υποκατηγορία μεθόδων αφορά σε αυτές που δημιουργούν μια νέα οντολογία σχεσιακού σχήματος, όπως αυτή ορίστηκε στην ενότητα 3.3. Μια τέτοια οντολογία έχει ως θεματικό αντικείμενο το σχεσιακό μοντέλο, με αποτέλεσμα να μην εξετάζει τη θεματική περιοχή που περιγράφουν τα περιεχόμενα της ΒΔ. Με άλλα λόγια, μια οντολογία σχεσιακού σχήματος περιέχει έννοιες και ιδιότητες που περιγράφουν τα ίδια τα συστατικά του σχεσιακού μοντέλου, όπως σχέση, γνώρισμα, πρωτεύον και ξένο κλειδί, αντανακλώντας πλήρως το σχήμα μιας σχεσιακής ΒΔ.

Η διαδικασία παραγωγής μιας οντολογίας σχεσιακού σχήματος, όπως είναι λογικό, είναι εντελώς *αυτοματοποιημένη*, καθώς όλη η αναγκαία πληροφορία περιέχεται στο σχήμα της ΒΔ και δεν χρειάζεται επιπλέον γνώση από κάποιον ανθρώπινο χρήστη ή άλλη εξωτερική πηγή. Όλοι οι πιθανοί *τρόποι πρόσβασης* στο αποτέλεσμα της αντιστοιχίας συναντώνται σε μεθόδους αυτής της κατηγορίας, ενώ σπάνια η αντιστοιχία εκφράζεται σε κάποια *μορφή*. Αυτό ισχύει επειδή οι αντιστοιχίες μεταξύ του σχεσιακού σχήματος και της οντολογίας είναι προφανείς, δεδομένου του ότι κάθε στοιχείο του πρώτου αντιστοιχεί σε ένα στιγμιότυπο της σχετικής κλάσης της δεύτερης. Παραδείγματος χάριν, μια σχέση και καθένα από τα γνωρίσματά της αντιστοιχούν σε στιγμιότυπα των κλάσεων «Σχέση» και «Γνώρισμα» ή παρομοίων τους αντίστοιχα. Επίσης, ένα στιγμιότυπο της κλάσης «Σχέση» συσχετίζεται με ένα στιγμιότυπο της κλάσης «Γνώρισμα» μέσω μιας κατάλληλης ιδιότητας «έχειΓνώρισμα» ή κάποιας παρόμοιας αυτής. Τα παραπάνω ονόματα κλάσεων και ιδιοτήτων είναι ενδεικτικά, καθώς κάθε διαφορετική προσέγγιση ορίζει τα δικά της οντολογικά στοιχεία.

Ένα ζήτημα που προκύπτει συχνά κατά τη μοντελοποίηση μιας οντολογίας σχεσιακού σχήματος είναι η ανάγκη να αναπαρασταθεί, εκτός από το σχεσιακό σχήμα, και οι περιορισμοί αυτού που εφαρμόζονται στα περιεχόμενα της ΒΔ. Στην πλειοψηφία των περιπτώσεων, κίνητρο αποτελεί ο έλεγχος των δομικών περιορισμών του σχεσιακού μοντέλου μέσω υπηρεσιών συλλογισμού. Αυτό βέβαια, απαιτεί όχι μόνο αναπαράσταση του σχήματος μιας ΒΔ αλλά και την περιγραφή των δεδομένων της. Η πλέον προφανής και φυσική επιλογή μοντελοποίησης που πληροί τις δύο αυτές προϋποθέσεις αναπαριστά στοιχεία του σχήματος μιας ΒΔ (π.χ. σχέσεις, γνωρίσματα) ως στιγμιότυπα πρωταρχικών κλάσεων της οντολογίας σχεσιακού σχήματος (π.χ. κλάσεις «Σχέση», «Γνώρισμα») και τα πραγματικά δεδομένα ως στιγμιότυπα των στοιχείων της οντολογίας που αναπαριστούν το σχήμα της ΒΔ (π.χ. κλάσεις «Εργαζόμενος» και «Τμήμα»). Ένα παράδειγμα αυτού του τρόπου μοντελοποίησης απεικονίζεται στο σχήμα 3.5, όπου έχει εμπλουτιστεί η βασική προσέγγιση και ο προκύπτων RDF γράφος του σχήματος 3.1 με ιδιότητες και κλάσεις μιας ενδεικτικής οντολογίας σχεσιακού σχήματος.

Η παραπάνω τεχνική μοντελοποίησης είναι γενικότερα γνωστή ως *μεταμοντελοποίηση* (metamodeling), και υποστηρίζεται από το RDF μοντέλο καθώς και από OWL οντολογίες που ερμηνεύονται με βάση αυτό. Διαισθητικά, η τεχνική της μεταμοντελοποίησης επιτρέπει σε έναν οντολογικό πόρο να λειτουργεί ταυτόχρονα ως κλάση, ιδιότητα ή και ως άτομο, γεγονός που καθιστά δυσδιάκριτο το διαχωρισμό μεταξύ του σώματος ορολογίας και του σώματος ισχυρισμών μιας βάσης γνώσης. Αυτό είναι φανερό και στο σχήμα 3.5, όπου



Σχήμα 3.5: *Παράδειγμα συνδυασμού οντολογίας σχεσιακού σχήματος και βασικής προσέγγισης*

ο πόρος που αντιστοιχεί στο γνώρισμα "Name" λειτουργεί ως ιδιότητα αλλά και ως άτομο, στιγμιότυπο της κλάσης "Column". Ως εκ τούτου, η μεταμοντελοποίηση παραβιάζει την αρχή διαχωρισμού μεταξύ κλάσεων, ιδιοτήτων και ατόμων που ισχύει στις Περιγραφικές Λογικές, χωρίς την οποία ο συλλογισμός καθίσταται μια μη αποκρίσιμη (undecidable) διαδικασία [141]. Θα πρέπει επίσης να τονιστεί ότι το χαρακτηριστικό της *παρονομασίας* (punning) της OWL μπορεί μεν να επιτρέπει χρήση του ίδιου ονόματος για την περιγραφή μιας κλάσης και μιας ιδιότητας, αλλά τις αντιμετωπίζει ως ξεχωριστά οντολογικά στοιχεία. Η ίδια αντιμετώπιση συναντάται και στην πλειοψηφία εργαλείων συλλογισμού που ερμηνεύουν OWL οντολογίες ως βάσεις γνώσης εκφρασμένες στην αντίστοιχη Περιγραφική Λογική (μοντελο-θεωρητική σημασιολογία), με αποτέλεσμα να μην εξάγουν τα συμπεράσματα που θα αναμένονταν από τη χρήση της μεταμοντελοποίησης. Ίσως αυτός είναι και ένας από τους λόγους για τους οποίους μεθοδολογίες που παράγουν μια οντολογία σχεσιακού σχήματος κάνοντας χρήση μεταμοντελοποίησης [153, 189] τελικά δεν εκμεταλλεύονται την περιγραφή του σχήματος της ΒΔ χρησιμοποιώντας υπηρεσίες συλλογισμού για τον έλεγχο δομικών περιορισμών της ΒΔ.

Εντούτοις, για την εξαγωγή συμπερασμάτων όπως των παραπάνω, είναι πιθανό να αρκούν εργαλεία συλλογισμού που υλοποιούν τους κανόνες συλλογισμού OWL 2 RL/RDF [4] ή κάποια μηχανή απόδειξης θεωρημάτων λογικής πρώτης τάξης (first-order logic theorem prover), παρά το γεγονός ότι και στις δύο περιπτώσεις, ο συλλογισμός δεν μπορεί να είναι *πλήρης* [170]. Αυτό στην πράξη σημαίνει ότι δεν μπορεί να εξαχθεί το σύνολο των συμπερασμάτων που συνεπάγεται μια OWL οντολογία που ερμηνεύεται ως RDF γράφος (OWL Full

[4] Οι κανόνες συλλογισμού OWL 2 RL/RDF (http://www.w3.org/TR/owl2-profiles/#Reasoning_in_OWL_2_RL_and_RDF_Graphs_using_Rules) μπορούν να εφαρμοστούν σε οποιοδήποτε RDF γράφο ή OWL 2 RL οντολογία, περιέχοντας κανόνες που αφορούν σε ένα συντακτικό υποσύνολο της OWL.

οντολογία), γεγονός όμως που δεν αναιρεί τη χρησιμότητα αυτών των εργαλείων συλλογισμού για τις ανάγκες της συγκεκριμένης εφαρμογής.

Εξ ορισμού, η πλειοψηφία των μεθόδων αυτής της κατηγορίας αντιστοιχούν μια ΒΔ σε μία μόνο οντολογία, χωρίς να *επαναχρησιμοποιούν όρους* από εξωτερικά λεξιλόγια, εφόσον το σχεσιακό μοντέλο αποτελεί τον τομέα ενδιαφέροντος. Το προφανές μειονέκτημα αυτών των προσεγγίσεων είναι ότι δεν είναι σαφής η σημασία των περιεχομένων της ΒΔ, γεγονός που τις καθιστά λιγότερο χρήσιμες στο πλαίσιο εφαρμογών ολοκλήρωσης με άλλες πηγές δεδομένων. Ως εκ τούτου, κάποιες μέθοδοι αυτής της κατηγορίας συμπληρώνουν και εμπλουτίζουν την οντολογία σχεσιακού σχήματος με χειροκίνητα ορισμένες αντιστοιχίες προς εξωτερικές οντολογίες πεδίου. Αυτές οι αντιστοιχίες μεταξύ οντολογιών μπορεί να χρησιμοποιούν τους μηχανισμούς που προβλέπουν η RDFS και η OWL για αυτόν το σκοπό (αξιώματα υποκλάσης `rdfs:subClassOf` ή ισοδύναμης κλάσης `owl:equivalentClass`) [117, 118], SPARQL CONSTRUCT ερωτήματα που δημιουργούν έναν καινούριο RDF γράφο που χρησιμοποιεί όρους από γνωστές οντολογίες [154] ή SWRL κανόνες με όρους της οντολογίας σχεσιακού σχήματος στο σώμα τους και όρους από εξωτερικές οντολογίες στην κεφαλή τους [83]. Μέθοδοι σαν και αυτές, έστω και έμμεσα, καταφέρνουν να περιγράψουν τη σημασία των περιεχομένων της ΒΔ και συνεπώς, πλησιάζουν τον στόχο της σημασιολογικής διαλειτουργικότητας μεταξύ ανεξάρτητα ανεπτυγμένων ΒΔ.

Στον πίνακα 3.2, παρουσιάζουμε μια επισκόπηση των μηχανισμών που χρησιμοποιούν οι μέθοδοι αυτής της κατηγορίας για να απεικονίσουν τα σημαντικότερα στοιχεία μιας σχεσιακής ΒΔ σε μια οντολογία σχεσιακού σχήματος, ενώ για κάθε μέθοδο επίσης αναφέρεται αν προβλέπει σύνδεση των όρων της οντολογίας σχεσιακού σχήματος με μια οντολογία πεδίου. Σημειώνουμε και πάλι ότι, επειδή κάθε μέθοδος ορίζει τη δική της οντολογία σχεσιακού σχήματος, η αναφορά σε κλάσεις και ιδιότητες αυτής στον πίνακα 3.2 χρησιμοποιεί την ονομασία που έχει επιλεγεί στην εκάστοτε εργασία και η οποία δίνεται εντός εισαγωγικών, προκειμένου να διακρίνεται από κλάσεις και ιδιότητες των γλωσσών RDFS και OWL. Λεπτομερής περιγραφή όλων των μεθόδων που αναφέρονται στον πίνακα 3.2 υπάρχει στη δημοσίευση [182].

Πίνακας 3.2: *Μέθοδοι παραγωγής οντολογίας σχεσιακού σχήματος*

Εργασία	Στοιχεία σχεσιακής ΒΔ						Σύνδεση με οντολογία πεδίου
	Σχέση	Γνώρισμα	Πλειάδα	Πρωτεύον κλειδί	Ξένο κλειδί	Περιορισμοί γνωρισμάτων	
Automapper [83]	Κλάση	Ιδιότητα τύπου δεδομένων	Στιγμιότυπο της κλάσης μιας σχέσης	SWRL κανόνας	–	`owl:allValuesFrom` και `owl:cardinality = 1` αξιώματα για πεδίο τιμών και NOT NULL περιορισμούς αντίστοιχα	Ναι
CROSS [55]	Υποκλάση της κλάσης *«Γραμμή»*	Ιδιότητα τύπου δεδομένων	Στιγμιότυπο της κλάσης *«Γραμμή»*	Ιδιότητα αντικειμένου	Ιδιότητα αντικειμένου	Περιορισμός `rdfs:range` στην ιδιότητα του γνωρίσματος για περιορισμό πεδίου τιμών	Όχι
DataMaster [147]	Κλάση	Ιδιότητα τύπου δεδομένων	Στιγμιότυπο της κλάσης μιας σχέσης	–	Στιγμιότυπο της κλάσης *«Ξένο κλειδί»*	–	Όχι
	Στιγμιότυπο της κλάσης *«Πίνακας»*	Στιγμιότυπο της κλάσης *«Στήλη»*	–	Στιγμιότυπο της κλάσης *«Πρωτεύον κλειδί»*	Ιδιότητα *«αναφέρεται σε»*	Περιορισμός `rdfs:range` στην ιδιότητα *«έχει τύπο δεδομένων»* για περιορισμούς πεδίου τιμών	Όχι
FDR2 [117]	Λίστα (`rdf:List`) με τις κλάσεις γνωρισμάτων	Κλάση	Λίστα (`rdf:List`) τιμών	–	–	–	Ναι
Kupfer *και συνεργάτες* (2006) [118]	Στιγμιότυπο της κλάσης *«Σχέση»*	Στιγμιότυπο της κλάσης *«Γνώρισμα»*	–	–	Ιδιότητα *«συσχετίζεται με»*	Ιδιότητα *«αποτελείται από»* για περιορισμούς πεδίου τιμών	Ναι
Lausen (2007) [120]	Κλάση	Ιδιότητα	Κενός κόμβος	Στιγμιότυπο της κλάσης *«Κλειδί»*	Στιγμιότυπο της κλάσης *«Ξένο Κλειδί»*	–	Όχι
Levshin (2009) [126]	Υποκλάση της κλάσης *«Πλειάδα»*	Ιδιότητα τύπου δεδομένων	Στιγμιότυπο της κλάσης μιας σχέσης	SWRL κανόνας	SWRL κανόνες	Περιορισμός `rdfs:range` στην ιδιότητα του γνωρίσματος για περιορισμό πεδίου τιμών	Όχι

Συνέχεια στην επόμενη σελίδα

Εργασία	Στοιχεία σχεσιακής ΒΔ						Σύνδεση με οντολογία πεδίου
	Σχέση	Γνώρισμα	Πλειάδα	Πρωτεύον κλειδί	Ξένο κλειδί	Περιορισμοί γνωρισμάτων	
OntoMat-Reverse [197]	Στιγμιότυπο της κλάσης *«Πίνακας»*	Στιγμιότυπο της κλάσης *«Στήλη»*	–	Στιγμιότυπο της κλάσης *«Πρωτεύον κλειδί»*	–	Ιδιότητα *«τύπος»* για περιορισμούς πεδίου τιμών	Ναι
RDB2ONT [189]	Στιγμιότυπο της κλάσης *«Σχέση»*	Ιδιότητα + στιγμιότυπο της κλάσης *«Γνώρισμα»*	Στιγμιότυπο της κλάσης μιας σχέσης	Στιγμιότυπο της κλάσης *«Πρωτεύον γνώρισμα»*	Ιδιότητα *«αναφερόμενο γνώρισμα»*	Ιδιότητες *«is nullable»* και *«τύπος»* για NOT NULL και περιορισμούς πεδίου τιμών αντίστοιχα	Ναι
Relational.OWL [153]	Κλάση + στιγμιότυπο της κλάσης *«Πίνακας»*	Ιδιότητα τύπου δεδομένων + στιγμιότυπο της κλάσης *«Στήλη»*	Στιγμιότυπο της κλάσης μιας σχέσης	Στιγμιότυπο της κλάσης *«Πρωτεύον κλειδί»*	Ιδιότητα *«αναφέρεται σε»*	Περιορισμός `rdfs:range` στην ιδιότητα ενός γνωρίσματος για περιορισμούς πεδίου τιμών	Ναι [154]
ROSEX [66]	Στιγμιότυπο της κλάσης *«Πίνακας»*	Στιγμιότυπο της κλάσης *«Στήλη»*	–	Στιγμιότυπο της κλάσης *«Πρωτεύον κλειδί»*	Στιγμιότυπο της κλάσης *«Ξένο κλειδί»*	–	Ναι
Spyder [139]	Στιγμιότυπο της κλάσης *«Πίνακας»*	Στιγμιότυπο της κλάσης *«Στήλη»*	–	Στιγμιότυπο της κλάσης *«Πρωτεύον κλειδί»*	Στιγμιότυπο της κλάσης *«Ξένο κλειδί»*	Ιδιότητες *«τύπος δεδομένων»*, *«προκαθορισμένη τιμή»* και *«nullable»* για περιορισμούς πεδίου τιμών, προκαθορισμένης τιμής και NOT NULL περιορισμούς αντίστοιχα	Όχι

3.4.2 Παραγωγή μιας οντολογίας πεδίου

Αντί της δημιουργίας μιας οντολογίας που αντικατοπτρίζει πιστά το στιγμιότυπο μιας ΒΔ και του μετέπειτα εμπλουτισμού της με σημασιολογία του αντίστοιχου θεματικού πεδίου, είναι προτιμητέα η άμεση παραγωγή μιας οντολογίας που αναφέρεται στο θεματικό αντικείμενο των περιεχομένων της ΒΔ. Στην παρούσα παράγραφο, αναφερόμαστε σε προσεγγίσεις και εργαλεία που παράγουν απευθείας οντολογίες πεδίου, χωρίς την ενδιάμεση παραγωγή οντολογιών σχεσιακού σχήματος, όπως αυτών της παραγράφου 3.4.1. Αυτές οι οντολογίες πεδίου δεν περιέχουν έννοιες ή συσχετίσεις που αναφέρονται στο σχεσιακό ή στο μοντέλο ΟΣ, αλλά αντίθετα έννοιες και συσχετίσεις που προσιδιάζουν στον τομέα ενδιαφέροντος ενός στιγμιοτύπου ΒΔ. Η εκφραστικότητα και ο πλούτος της παραγόμενης οντολογίας φυσιολογικά εξαρτάται από την ποσότητα της - σχετικής με το συγκεκριμένο θεματικό αντικείμενο - γνώσης που ενσωματώνεται στη διαδικασία. Οι δύο κυριότερες πηγές τέτοιας γνώσης είναι ο ανθρώπινος χρήστης και το ίδιο το στιγμιότυπο της σχεσιακής ΒΔ. Αυτός είναι και ο λόγος για τον οποίο, στην ταξινομία του σχήματος 3.2 αλλά και στην τρέχουσα παράγραφο, κάνουμε διάκριση μεταξύ μεθόδων που στηρίζονται κυρίως στο σχήμα της σχεσιακής ΒΔ και σε *τεχνικές αντίστροφης μηχανικής* για την παραγωγή της οντολογίας πεδίου (αλλά μπορούν να δεχτούν είσοδο και από έναν ανθρώπινο γνώστη) και μεθόδων που *δεν* αναλύουν επισταμένα το σχήμα της ΒΔ, αλλά στηρίζονται σε μεγάλο βαθμό στη *βασική προσέγγιση* και, προαιρετικά, σε είσοδο από κάποιον ανθρώπινο χρήστη.

3.4.2.1 Προσεγγίσεις που δεν χρησιμοποιούν αντίστροφη μηχανική

Πρώτα, εξετάζουμε μεθόδους που δεν στηρίζονται στην ανάλυση του σχήματος μιας σχεσιακής ΒΔ για την εξαγωγή της σημασίας των περιεχομένων της. Τέτοιες μέθοδοι χρησιμοποιούν κυρίως τη βασική προσέγγιση της ενότητας 3.2 για την εξαγωγή του στιγμιοτύπου μιας ΒΔ σε μια RDFS οντολογία και, τις περισσότερες φορές, επιτρέπουν σε ένα γνώστη της σημασίας των περιεχομένων της ΒΔ να ορίσει σημασιολογικά ορθές αντιστοιχίες, εκφρασμένες σε κάποια ιδιότυπη μορφή αναπαράστασης. Το σώμα ορολογίας της παραγόμενης RDFS οντολογίας είναι συνήθως πολύ απλό, περιέχοντας μονάχα ορισμούς κλάσεων και ιδιοτήτων. Προσεγγίσεις αυτού του είδους συνήθως συνοδεύονται από υλοποιήσεις λογισμικού και αυτός είναι και ένας από τους λόγους που η συγκεκριμένη κατηγορία είναι μακράν η πιο δημοφιλής, ενώ παράλληλα αναπτύσσονται και αρκετά σχετικά εμπορικά εργαλεία.

Ο *βαθμός αυτοματοποίησης* για τη συγκεκριμένη κατηγορία εργαλείων ποικίλλει και εξαρτάται από το βαθμό εμπλοκής του ανθρώπινου παράγοντα στη διαδικασία. Τα περισσότερα από τα εργαλεία της κατηγορίας υποστηρίζουν πολλαπλούς τρόπους λειτουργίας: από την αυτοματοποιημένη βασική προσέγγιση με ή χωρίς επίβλεψη και προσαρμογή του τελικού αποτελέσματος από το χρήστη μέχρι τον πλήρως χειροκίνητο ορισμό αντιστοιχιών. Σε ό,τι αφορά στον *τρόπο πρόσβασης* του αποτελέσματος της αντιστοιχίας, και οι 3 επιλογές που αναφέρθηκαν στην ενότητα 3.3 υποστηρίζονται από εργαλεία αυτής της κατηγορίας, με μια ισχυρή προτίμηση σε πρόσβαση βασισμένη στη γλώσσα ερωτημάτων SPARQL.

Η γλώσσα που χρησιμοποιείται για την *αναπαράσταση της αντιστοιχίας*

είναι ένα χαρακτηριστικό ιδιαίτερα κρίσιμο για τις συγκεκριμένες μεθόδους. Σε πλήρη αντίθεση με εργαλεία που παράγουν μια οντολογία σχεσιακού σχήματος, όπου οι αντιστοιχίες με στοιχεία της ΒΔ είναι προφανείς από την επισκόπηση της οντολογίας, οι έννοιες και συσχετίσεις μιας νέας οντολογίας *πεδίου* μπορεί να προκύπτουν από οσοδήποτε σύνθετες εκφράσεις σχεσιακής άλγεβρας. Προκειμένου να εκφραστούν τέτοιες σύνθετες αντιστοιχίες, χρειάζεται μια εκφραστική γλώσσα για την αναπαράσταση της αντιστοιχίας που θα περιλαμβάνει τις αναγκαίες δυνατότητες για την κάλυψη περιπτώσεων που συναντώνται στην πράξη. Οι περισσότερες εκ των γλωσσών αναπαράστασης αντιστοιχιών που ορίζονται από τις προσεγγίσεις αυτής της κατηγορίας προσφέρουν τη δυνατότητα προσδιορισμού συνόλων δεδομένων από τη ΒΔ και μετασχηματισμού αυτών των συνόλων, ώστε να προκύψει εξ αυτών ο τελικός RDF γράφος. Δεδομένου του ότι, μόλις πρόσφατα η πρότυπη γλώσσα αναπαράστασης αντιστοιχιών R2RML πήρε την τελική της μορφή, δεν προξενεί έκπληξη το γεγονός ότι μέχρι τώρα κάθε εργαλείο χρησιμοποιούσε τη δική του γλώσσα, η οποία διέφερε ως προς τη σύνταξη και τις δυνατότητές της από τις υπόλοιπες. Αυτή η κατάσταση αναπόφευκτα οδήγησε στον εγκλωβισμό των χρηστών στο εργαλείο με το οποίο όρισαν μια αντιστοιχία, η οποία δεν μπορούσε να διαμοιραστεί και να επαναχρησιμοποιηθεί από χρήστες άλλων συστημάτων αντιστοιχίας χωρίς να οριστεί εκ νέου. Αυτή η ποικιλία γλωσσών ήταν και ο κυριότερος λόγος για την ανάπτυξη της R2RML από το W3C και αναμένεται ότι στο εγγύς μέλλον ο αριθμός των εργαλείων που θα την υιοθετήσουν θα αυξηθεί.

Όπως αναφέρθηκε και προηγουμένως, καθώς η πλειοψηφία των εργαλείων ακολουθούν τη βασική προσέγγιση, η *γλώσσα της παραγόμενης οντολογίας* είναι συνήθως RDFS. Εξάλλου, οι συγκεκριμένες μέθοδοι δίνουν έμφαση στην κατασκευή απλών οντολογιών που επαναχρησιμοποιούν όρους άλλων οντολογιών με απώτερο σκοπό την σημασιολογική διαλειτουργικότητα, και όχι τόσο στην κατασκευή σύνθετων οντολογικών δομών. Σε αυτό το πλαίσιο, η *επαναχρησιμοποίηση λεξιλογίων* υποστηρίζεται από το σύνολο των μεθόδων αυτής της κατηγορίας για την περίπτωση χειροκίνητα ορισμένων αντιστοιχιών. Προϋπόθεση, βέβαια, για την επιτυχή επαναχρησιμοποίηση όρων αποτελεί η εξοικείωση του χρήστη με δημοφιλείς οντολογίες του Σημασιολογικού Ιστού, αλλά και η γνώση της σημασίας των περιεχομένων της ΒΔ. Το κυρίαρχο σενάριο χρήσης στο πλαίσιο του οποίου εφαρμόζονται τέτοιες μέθοδοι είναι η δημιουργία σημαντικού όγκου RDF δεδομένων από σχεσιακές ΒΔ, γεγονός που με τη σειρά του επιτρέπει την ευκολότερη ολοκλήρωση αυτών των δεδομένων με δεδομένα από άλλες ετερογενείς πηγές.

Μια πρόχειρη επισκόπηση των δυνατοτήτων γλωσσών αναπαράστασης αντιστοιχιών που χρησιμοποιούνται από εργαλεία αυτής της κατηγορίας δίνεται στον πίνακα 3.3, με μια πιο ενδελεχή ανάλυση να δίνεται στο [101]. Οι κυριότερες δυνατότητες που προσφέρονται από γλώσσες του είδους είναι:

- *Υποστήριξη για αντιστοιχίες υπό συνθήκη*, δυνατότητα που επιτρέπει τον ορισμό πιο σύνθετων αντιστοιχιών σε σχέση με την τετριμμένη περίπτωση της απεικόνισης «σχέσης σε κλάση». Γλώσσες με αυτό το χαρακτηριστικό επιτρέπουν τον προσδιορισμό ενός υποσυνόλου των πλειάδων μιας σχέσης, οι οποίες θα απεικονιστούν σε μια οντολογική κλάση, με βάση κάποια δεδομένη συνθήκη.

- *Προσαρμογή του μηχανισμού παραγωγής IRI για τα άτομα μιας κλάσης*. Η συγκεκριμένη δυνατότητα επιτρέπει τον ορισμό προτύπων IRI (που δεν διαφέρουν και πολύ από αυτά που παρουσιάζονται στη μεσαία στήλη του πίνακα 3.1) τα οποία πιθανώς χρησιμοποιούν τιμές από γνωρίσματα πέραν του πρωτεύοντος κλειδιού μιας σχέσης, γενικεύοντας με αυτόν τον τρόπο τον τυπικό μηχανισμό παραγωγής IRI της βασικής προσέγγισης ή της Άμεσης Αντιστοιχίας.
- *Εφαρμογή συναρτήσεων μετασχηματισμού* στις τιμές γνωρισμάτων, δυνατότητα που προσφέρει μεγαλύτερη ελευθερία στη δημιουργία του RDF γράφου. Οι συναρτήσεις μετασχηματισμού μπορεί να είναι απλές διαδικασίες συνδυασμού συμβολοσειρών ή συνθετότερες αποθηκευμένες διαδικασίες (stored procedures) της ΒΔ.
- *Υποστήριξη για δημιουργία οντολογικών κλάσεων από τιμές γνωρισμάτων μιας σχέσης*.
- *Υποστήριξη ξένων κλειδιών*, δυνατότητα που επιτρέπει την ερμηνεία ενός ξένου κλειδιού ως ιδιότητα που συνδέει δύο οντολογικά άτομα, αντί για ιδιότητα που δέχεται ως αντικείμενο ένα απλό λεκτικό.

Πίνακας 3.3: *Σύγκριση γλωσσών αναπαράστασης αντιστοιχιών*

Γλώσσα	Αντιστοιχίες υπό συνθήκη	Μηχανισμός παραγωγής IRI για οντολογικά άτομα	Συναρτήσεις μετασχηματισμού	Δημιουργία κλάσεων από τιμές γνωρισμάτων	Υποστήριξη ξένων κλειδιών
D2RQ [41]	✓	✓	✓	✓	✓
METAmorphoses [185]	✓	–	–	–	✓
R2RML [69]	✓	✓	✓	✓	✓
R3M [100]	–	✓	–	–	✓
Spyder [139]	✓	✓	✓	✓	✓
SquirrelRDF [171]	–	–	–	–	–
Virtuoso Meta-Schema Language [42]	✓	✓	✓	✓	✓

Λεπτομερής ανάλυση εργαλείων που εμπίπτουν σε αυτή την κατηγορία υπάρχει στη δημοσίευση [182].

3.4.2.2 Προσεγγίσεις που χρησιμοποιούν αντίστροφη μηχανική

Τα εργαλεία της παραγράφου 3.4.2.1 χρησιμοποιούν κυρίως τον ανθρώπινο χρηστή ως πηγή άντλησης γνώσης για μια θεματική περιοχή, υποστηρίζοντάς τον κατά τη διαδικασία ορισμού της αντιστοιχίας μέσω της παροχής μιας αρχικής βασικής αντιστοιχίας, την οποία μπορεί στη συνέχεια να προσαρμόσει. Αντίθετα, οι προσεγγίσεις που θα αναφερθούν σε αυτή την παράγραφο θεωρούν ως κύρια πηγή γνώσης τη σχεσιακή ΒΔ, την οποία συμπληρώνουν με γνώση από εξωτερικές πηγές και, προαιρετικά, από κάποιον ανθρώπινο ειδικό. Παραδείγματα τέτοιων εξωτερικών πηγών αποτελούν τα ερωτήματα και εντολές χειρισμού δεδομένων που τίθενται στη συγκεκριμένη ΒΔ, προγράμματα εφαρμογών που επικοινωνούν με τη ΒΔ, θησαυροί, λεξιλόγια καθιερωμένων όρων και εξωτερικές οντολογίες.

Η πλειοψηφία των μεθόδων αυτής της κατηγορίας βασίζεται σε ερευνητικές εργασίες αντίστροφης μηχανικής σχεσιακών ΒΔ, στόχος των οποίων ήταν η ανάκτηση του εννοιολογικού σχήματος μιας ΒΔ από το σχεσιακό της σχήμα. Το εν λόγω πρόβλημα αποτέλεσε ένα από τα κυρίαρχα προβλήματα στο χώρο των ΒΔ από τις αρχές της δεκαετίας του 1990, προσελκύοντας το ενδιαφέρον μεγάλου αριθμού ερευνητών. Αρκετές μεθοδολογίες ανάστροφης μηχανικής ενός σχεσιακού σχήματος στο μοντέλο ΟΣ ή σε κάποιο αντικειμενοστρεφές μοντέλο προτάθηκαν και αρκετές από αυτές προσαρμόστηκαν κατάλληλα για να αντιμετωπίσουν το πρόβλημα της αντιστοιχίας μιας σχεσιακής ΒΔ σε μια οντολογία. Εντούτοις, μέχρι σήμερα, δεν υπάρχει κάποια μέθοδος η οποία να εξασφαλίζει απόλυτη επιτυχία στην ανάκτηση του εννοιολογικού σχήματος μιας ΒΔ, γεγονός που οφείλεται στην αναπόφευκτη απώλεια νοήματος που συνοδεύει την διαδικασία του λογικού σχεδιασμού μιας ΒΔ, δεδομένου του ότι διαφορετικά στοιχεία του μοντέλου ΟΣ μετατρέπονται σε παρόμοιες δομές στο σχεσιακό μοντέλο, όπως είδαμε και στην παράγραφο 2.1.3. Εκτός αυτού,η ανάκτηση του αρχικού νοήματος δυσχεραίνεται και από το γεγονός ότι ένα σχεσιακό σχήμα συχνά δεν προκύπτει από την εφαρμογή των ορθών και καθιερωμένων πρακτικών σχεδιασμού, καθώς και από την αυθαίρετη ονομασία σχέσεων και γνωρισμάτων, η οποία συχνά έχει μικρή σχέση με τη σημασία των τιμών τους [161]. Κατά συνέπεια, η αντίστροφη μηχανική ενός σχεσιακού σχήματος μπορεί να θεωρηθεί περισσότερο ως τέχνη, προσαρμοζόμενη κατά περίπτωση και λιγότερο ως αυστηρή εφαρμογή ενός καθορισμένου αλγορίθμου με εξασφαλισμένα αποτελέσματα.

Οι μεθοδολογίες αντίστροφης μηχανικής σχεσιακών ΒΔ παρουσιάζουν μεγάλη ποικιλομορφία, που οφείλεται στις διαφορετικές υποθέσεις που υιοθετεί η καθεμία. Αυτές οι υποθέσεις αναφέρονται στην κανονική μορφή του σχεσιακού σχήματος, στα στοιχεία του σχεσιακού σχήματος που θεωρούνται γνωστά (π.χ. υποψήφια κλειδιά, ξένα κλειδιά κλπ), στο συγκεκριμένο εννοιολογικό μοντέλο το οποίο πρόκειται να εξαχθεί, καθώς και στις εξωτερικές πηγές πληροφορίας που είναι διαθέσιμες. Οι περισσότερες μέθοδοι υποθέτουν ότι το σχεσιακό σχήμα είναι εκφρασμένο σε τρίτη κανονική μορφή (3NF), λόγω των επιθυμητών ιδιοτήτων που αυτή εξασφαλίζει (διατήρηση συνενώσεων, διατήρηση εξαρτήσεων) [163]. Εντούτοις, υπάρχουν και μέθοδοι λιγότερο περιοριστικές, οι οποίες δέχονται και σχεσιακά σχήματα σε δεύτερη κανονική μορφή (2NF) [164].

Επίσης, κάποιες μέθοδοι δεν θεωρούν ως δεδομένη τη γνώση υποψήφιων, πρωτευόντων και ξένων κλειδιών για κάθε σχέση, με αποτέλεσμα να προσπαθούν να ανακαλύψουν συναρτησιακές εξαρτήσεις (functional dependencies) και εξαρτήσεις εγκλεισμού (inclusion dependencies) με ανάλυση των δεδομένων της ΒΔ [8, 62]. Η γνώση περιορισμών κλειδιών είναι ένα καίριο ζήτημα, καθώς διευκολύνει σε μεγάλο βαθμό τη διαδικασία αντίστροφης μηχανικής. Ως εκ τούτου, θεωρείται απαραίτητος για τις περισσότερες προσεγγίσεις (π.χ. [108, 138]), αν και υπάρχουν και εξαιρέσεις (π.χ. [161, 164]). Κάποιες μεθοδολογίες εστιάζουν σε συγκεκριμένα δομικά στοιχεία του εννοιολογικού μοντέλου που εξετάζουν, όπως για παράδειγμα οι πληθικότητες των περιορισμών συμμετοχής σε τύπους συσχέτισης [8] ή ιεραρχίες γενίκευσης [119] και προσπαθούν να τα εντοπίσουν. Τέλος, μια άλλη διαφοροποίηση μεταξύ των μεθόδων αντίστροφης μηχανικής αναφέρεται στο εννοιολογικό μοντέλο που εξετάζουν:

κάποιες θεωρούν το μοντέλο ΕΟΣ (π.χ. [62]), ενώ άλλες θεωρούν αντικειμενοστρεφή μοντέλα (π.χ. [31, 161]). Αναμενόμενα, οι περισσότερες μέθοδοι αντίστροφης μηχανικής είναι ημι-αυτόματες, καθώς χρειάζονται επαλήθευση του αποτελέσματος της από ειδικό, ενώ μπορούν να εκμεταλλευτούν και άλλες πηγές πληροφορίας, όπως συνηθισμένα ερωτήματα που τίθενται στη ΒΔ ή σχετικά θεματικά λεξιλόγια όρων [111].

Μέθοδοι σαν τις προηγούμενες μπορεί να αποδειχθούν χρήσιμες για την παραγωγή μιας οντολογίας πεδίου από μια σχεσιακή ΒΔ, δεδομένου του ότι οι Περιγραφικές Λογικές παρουσιάζουν αρκετές ομοιότητες με αντικειμενοστρεφή μοντέλα, τα οποία αποτελούν το στόχο μερικών από τις προηγούμενες μεθόδους αντίστροφης μηχανικής. Οι προσεγγίσεις παραγωγής οντολογίας πεδίου που εξετάζουμε σε αυτή την παράγραφο δέχονται ως είσοδο ένα σχεσιακό σχήμα, ορισμένο ως ένα σύνολο SQL εντολών ορισμού δεδομένων, που περιέχει πληροφορία για τα πρωτεύοντα και ξένα κλειδιά των σχέσεων και για τα πεδία τιμών των γνωρισμάτων του σχήματος και εξάγουν μια οντολογία, αντλώντας γνώση όχι μόνο από το στιγμιότυπο (σχήμα και δεδομένα) της ΒΔ αλλά και από άλλες πηγές γνώσης. Η παραγωγή της οντολογίας βασίζεται σε ένα σύνολο ευρετικών κανόνων που ουσιαστικά προσπαθούν να υλοποιήσουν τον αντίστροφο μετασχηματισμό της διαδικασίας λογικού σχεδιασμού, όπως αυτή φαίνεται σε αδρές γραμμές στον πίνακα 2.1. Η ποικιλία που παρατηρείται στις παραδοσιακές μεθόδους αντίστροφης μηχανικής δεν χαρακτηρίζει στον ίδιο βαθμό τις μεθόδους της τρέχουσας ενότητας, καθώς η πλειοψηφία αυτών ξεκινούν από τη θεώρηση ενός σχήματος σε 3NF με πλήρη γνώση κλειδιών και πεδίων τιμών, ενώ η οντολογική γλώσσα-στόχος είναι συνήθως ή κάποια άλλη γλώσσα Περιγραφικής Λογικής.

Εκτός από τις μεθόδους που ακολουθούν την προαναφερθείσα φιλοσοφία, υπάρχουν και δύο άλλες ομάδες σχετικών μεθόδων. Η πρώτη από αυτές θεωρεί ως είσοδο της διαδικασίας παραγωγής οντολογίας ένα μοντέλο ΟΣ, ενώ η δεύτερη χρησιμοποιεί ενδιάμεσα εννοιολογικά μοντέλα για τη μετάβαση από το σχεσιακό στο οντολογικό μοντέλο. Στην πρώτη περίπτωση, η πρακτική χρησιμότητα των μεθόδων που χρησιμοποιούν ένα μοντέλο ΟΣ για την παραγωγή μιας οντολογίας είναι αμφισβητήσιμη, καθώς, όπως αναφέραμε, σπάνια ένα μοντέλο ΟΣ είναι διαθέσιμο για μια δεδομένη σχεσιακή ΒΔ. Εντούτοις, όπως είναι λογικό και αναμενόμενο, αυτές οι μέθοδοι επιτυγχάνουν την κατασκευή πιο εκφραστικών οντολογιών σε σχέση με αυτές που ξεκινούν από ένα σχεσιακό σχήμα, αφού το μοντέλο ΟΣ είναι ένα εννοιολογικό μοντέλο και τα περισσότερα, αν όχι όλα, δομικά στοιχεία του μπορούν να αναπαρασταθούν σε μια OWL οντολογία. Προσεγγίσεις, όπως οι [80, 145, 191, 202], παράγουν OWL οντολογίες κυμαινόμενης εκφραστικότητας - από OWL Lite μέχρι OWL Full - εφαρμόζοντας ένα σύνολο κανόνων μετάφρασης στοιχείων του μοντέλου ΟΣ σε OWL μηχανισμούς. Συνεπώς, δεν εφαρμόζουν αντίστροφη μηχανική στο σχεσιακό σχήμα, αλλά απλά πραγματοποιούν μετατροπή από ένα εννοιολογικό μοντέλο σε άλλο. Αυτή την πορεία ακολουθούν και μέθοδοι, όπως οι [17, 102] με τη μόνη διαφορά ότι σε αυτές χρησιμοποιείται ένα εννοιολογικό μοντέλο ως ενδιάμεσο βήμα για τη μετατροπή από το σχεσιακό στο οντολογικό μοντέλο.

Επιχειρώντας μια σύνοψη των μεθόδων παραγωγής οντολογίας πεδίου αυτής της κατηγορίας, θα λέγαμε ότι αυτές είναι κυρίως *αυτόματες*, καθώς στηρίζονται σε μεγάλο βαθμό σε σύνολα γενικών κανόνων που μπορούν να εφαρ-

μοστούν σε ένα οποιοδήποτε σχεσιακό σχήμα. Βεβαίως, όπως φάνηκε και από τα προηγούμενα, το αποτέλεσμα της εφαρμογής αυτών των κανόνων δεν οδηγεί πάντα σε μια σημασιολογικά ακριβή απεικόνιση του σχεσιακού σχήματος με αποτέλεσμα αρκετές εκ των μεθόδων αυτών να προβλέπουν και τη συμμετοχή ενός ανθρώπινου ειδικού ο οποίος θα επαληθεύσει την τελική οντολογία και, προαιρετικά, θα την εμπλουτίσει με συνδέσμους προς άλλες γνωστές οντολογίες πεδίου, είτε τελείως χειροκίνητα είτε ημι-αυτόματα, αξιοποιώντας προτάσεις που προκύπτουν από κάποιο σχετικό αλγόριθμο [129]. Σε μερικές περιπτώσεις, η ανακάλυψη συσχετίσεων μεταξύ της παραχθείσας οντολογίας και άλλων οντολογιών πεδίου [21] ή λεξικολογικών βάσεων δεδομένων [109], όπως το WordNet, πραγματοποιείται αυτόματα. Σε άλλες περιπτώσεις, ένα μοντέλο ΟΣ που περιγράφει την υπό εξέταση ΒΔ μπορεί να χρησιμοποιηθεί για την επαλήθευση του αποτελέσματος της διαδικασίας παραγωγής οντολογίας [6]. Σε αντίθεση με παλαιότερες μεθόδους αντίστροφης μηχανικής, οι προσεγγίσεις αυτής της παραγράφου δεν χρησιμοποιούν SQL ερωτήματα και εντολές χειρισμού δεδομένων προκειμένου να βελτιώσουν το παραγόμενο τελικό αποτέλεσμα. Ο κατάλογος των πηγών γνώσης που λαμβάνονται υπόψη από τις προσεγγίσεις αυτής της κατηγορίας παρουσιάζεται στον πίνακα 3.4.

Πίνακας 3.4: *Πηγές πληροφορίας για μεθόδους παραγωγής οντολογίας πεδίου*

Εργασία	Πηγές πληροφορίας			
	Σχήμα ΒΔ	Δεδομένα ΒΔ	Άλλες πηγές	Χρήστης
Astrova (2004) [15]	✓	✓		✓
Astrova (2009) [16]	✓			
Buccella *και συνεργάτες* (2004) [47]	✓			✓
DB2OWL [87]	✓			
DM-2-OWL [7]	✓			
Jurić, Banek, Skočir [109]	✓		WordNet	
Lubyte, Tessaris (2009) [133]	✓			✓
R2O [180]	✓		Μοντέλο ΟΣ για επαλήθευση	
RDBToOnto [54]	✓	✓		✓
ROSEX [66]	✓			
Shen *και συνεργάτες* (2006) [175]	✓			
SOAM [129]	✓		Λεξικολογικά αποθετήρια	✓
SQL2OWL [6]	✓	✓	Μοντέλο ΟΣ για επαλήθευση	✓
L. Stojanovic, N. Stojanovic, Volz (2002) [184]	✓			✓
Tirmizi, Sequeda, Miranker (2008) [188]	✓			

Παρατηρώντας τον πίνακα 3.4, γίνεται φανερό ότι ελάχιστες προσεγγίσεις εκμεταλλεύονται το περιεχόμενο της ΒΔ για την κατασκευή μιας πληρέστερης και σημασιολογικά ορθής οντολογίας. Ανάμεσά τους, η πιο αξιόλογη προσπάθεια είναι ίσως το RDBToOnto [54], το οποίο προτείνει έναν αλγόριθμο για την αναγνώριση των γνωρισμάτων μιας σχέσης που πιθανώς δρουν ως κατηγορικά γνωρίσματα και συνεπώς, μπορούν να οδηγήσουν στην κατασκευή μιας ιεραρχίας κλάσεων.

Όσον αφορά στον *τρόπο πρόσβασης* του αποτελέσματος, όλες οι μέθοδοι εξάγουν μια υλοποιημένη (materialized) οντολογία. Αυτό είναι λογικό, εφόσον το βασικό κίνητρο αυτής της κατηγορίας μεθόδων είναι η δημιουργία μιας νέας οντολογίας η οποία θα μπορεί να χρησιμοποιηθεί και να είναι διαθέσιμη σε τρίτες εφαρμογές και συνεπώς, θα πρέπει να είναι υλοποιημένη. Λογική

συνέπεια αυτής της παρατήρησης είναι και η έλλειψη *αναπαράστασης της αντιστοιχίας* από την πλειονότητα των μεθόδων, δεδομένου ότι η ΒΔ στο συγκεκριμένο πλαίσιο λειτουργεί απλά ως πηγή πληροφορίας και η προέλευση των όρων της οντολογίας από αντίστοιχα στοιχεία της ΒΔ έχει δευτερεύουσα σημασία, οπότε και η αντιστοιχία σπάνια χρειάζεται να εκφραστεί ή αποθηκευθεί με κάποια μορφή.

Η *οντολογική γλώσσα*-στόχος των περισσότερων μεθόδων είναι η OWL, με ελάχιστες εξαιρέσεις [15, 133, 184]. Η *επαναχρησιμοποίηση* όρων από άλλες οντολογίες δεν είναι δυνατή σε αυτή την κατηγορία μεθόδων, αν και μερικές από αυτές, όπως αναφέρθηκε και προηγουμένων, προσπαθούν να δημιουργήσουν συνδέσμους με υπάρχουσες οντολογίες πεδίου, προκειμένου να θεμελιώσουν σημασιολογικά τους όρους της δημιουργηθείσας οντολογίας. Ένα εμφανές μειονέκτημα που μοιράζονται οι περισσότερες προσεγγίσεις που εφαρμόζουν τεχνικές αντίστροφης μηχανικής είναι η μεγάλη δομική ομοιότητα της νέας οντολογίας, η οποία συχνά αντιγράφει πιστά το σχεσιακό σχήμα της ΒΔ.

Σε αντίθεση με τα εργαλεία της παραγράφου 3.4.2.1, τα οποία εξάγουν τα περιεχόμενα της ΒΔ σε RDF γράφους, δημιουργώντας αντίστοιχα στιγμιότυπα οντολογικών κλάσεων, οι μεθοδολογίες αυτής της παραγράφου δεν εμπλουτίζουν πάντα τη νέα οντολογία με δεδομένα της βάσης και συχνά περιορίζονται στη δημιουργία του σώματος ορολογίας (TBox) μιας οντολογίας. Προσεγγίσεις που εμπλουτίζουν τη νέα οντολογία με οντολογικά άτομα - διαδικασία γνωστή και ως ontology population - συνήθως υλοποιούν τις αντίστοιχες προτάσεις που αποτελούν το σώμα ισχυρισμών (ABox), με την εξαίρεση των εργαλείων DB2OWL [87] και Ultrawrap [172], τα οποία διατηρούν το σώμα ισχυρισμών στη ΒΔ, προσφέροντας δυναμική πρόσβαση σε αυτό μέσω SPARQL.

Οι ευρετικοί κανόνες στους οποίους στηρίζεται η παραγωγή της οντολογίας συνήθως ομαδοποιούν τις σχέσεις μιας ΒΔ ανάλογα με τον αριθμό των γνωρισμάτων του πρωτεύοντος κλειδιού, τον αριθμό των ξένων κλειδιών και τις επικαλύψεις μεταξύ του πρωτεύοντος και των ξένων κλειδιών, επιχειρώντας να αντιστρέψουν το μετασχηματισμό που ορίζει ο πίνακας 2.1. Στις περισσότερες περιπτώσεις, αυτοί οι κανόνες εκφράζονται σε άτυπη μορφή [47, 87, 129, 175, 180, 184] ή ακόμα και μέσω βολικών παραδειγμάτων που δεν μπορούν να γενικευτούν [15, 16]. Επιπλέον, μερικές μέθοδοι παράγουν ακόμα και μη ορθά αποτελέσματα, καθώς χρησιμοποιούν κανόνες που ερμηνεύουν με λανθασμένο τρόπο το νόημα ενός σχεσιακού σχήματος ή δεν ακολουθούν την καθιερωμένη σημασιολογία των στοιχείων της OWL. Δύο από τις πλέον πλήρεις και τυπικά εκφρασμένες μεθοδολογίες είναι η SQL2OWL [6] και η δουλειά των Tirmizi και συναδέλφων [188], στις οποίες προτείνονται σύνολα Horn κανόνων που καλύπτουν το σύνολο των δυνατών μορφών που μπορεί να λάβει το σχήμα μιας σχέσης.

Καθώς τα σύνολα κανόνων που εφαρμόζουν οι διάφορες μεθοδολογίες είναι σε μεγάλο βαθμό επικαλυπτόμενα, αναφέρουμε άτυπα τα κυριότερα είδη κανόνων που συναντώνται σε αυτές:

1. **Βασικοί κανόνες.** Αυτοί είναι οι κανόνες οι οποίοι επεκτείνουν τη βασική προσέγγιση για την περίπτωση του OWL μοντέλου: μια σχέση αντιστοιχεί σε μια OWL κλάση, ένα γνώρισμα που δεν είναι μέρος ενός ξένου κλειδιού αντιστοιχεί σε μια ιδιότητα τύπου δεδομένων (datatype property) ενώ ένα ξένο κλειδί αντιστοιχεί σε μια ιδιότητα αντικειμένου

(object property) με πεδίο ορισμού και πεδίο τιμών τις OWL κλάσεις που αντιστοιχούν στην αναφέρουσα και στην αναφερόμενη σχέση του ξένου κλειδιού αντίστοιχα. Επιπλέον, μια πλειάδα μιας σχέσης αντιστοιχεί σε ένα άτομο της αντίστοιχης OWL κλάσης.

2. **Κανόνες ιεραρχίας.** Τέτοιοι κανόνες αναγνωρίζουν ιεραρχίες κλάσεων στο σχεσιακό σχήμα. Σύμφωνα με τον πιο διαδεδομένο σχετικό κανόνα, όταν τα πρωτεύοντα κλειδιά δύο σχέσεων συνδέονται μεταξύ τους με σχέση ξένου κλειδιού, τότε αντιστοιχούν σε δύο OWL κλάσεις που συνδέονται με σχέση κλάσης/υποκλάσης. Πιο συγκεκριμένα, η αναφερόμενη σχέση του ξένου κλειδιού αντιστοιχεί στην υπερκλάση της κλάσης που αντιστοιχεί στην αναφέρουσα σχέση. Αυτοί οι κανόνες συχνά έρχονται σε σύγκρουση με τους *κανόνες κατακερμάτισης* που θα δούμε στη συνέχεια.

3. **Κανόνες δυαδικής συσχέτισης.** Αυτοί οι κανόνες αναγνωρίζουν μια σχέση που αναπαριστά μια δυαδική συσχέτιση στο μοντέλο ΟΣ και συνεπώς, πρέπει να αντιστοιχηθεί σε μια OWL ιδιότητα αντικειμένου p. Το πρωτεύον κλειδί μιας τέτοιας σχέσης R αποτελείται από ξένα κλειδιά προς δύο άλλες σχέσεις S και T που αντιστοιχούν σε OWL κλάσεις, ενώ η εν λόγω σχέση R δεν έχει επιπλέον γνωρίσματα. Μία από τις κλάσεις που αντιστοιχεί στις S και T επιλέγεται ως το πεδίο ορισμού της νέας ιδιότητας αντικειμένου p και η άλλη αντιστοιχεί στο πεδίο τιμών της p. Η επιλογή αυτή συνήθως γίνεται από τον ανθρώπινο χρήστη που επιβλέπει τη διαδικασία. Όταν η μέθοδος είναι πλήρως αυτόματη, συνήθως δημιουργούνται δύο ιδιότητες αντικειμένου, ανάστροφες η μία της άλλης. Στην περίπτωση που η σχέση R έχει επιπλέον γνωρίσματα, αυτή αντιστοιχεί σε μια δυαδική συσχέτιση με περιγραφικά γνωρίσματα στο μοντέλο ΟΣ. Παρ' όλα αυτά, καθώς η OWL δεν υποστηρίζει με άμεσο τρόπο το συγκεκριμένο χαρακτηριστικό, η δυαδική συσχέτιση αντιστοιχεί σε μια OWL κλάση και τα περιγραφικά γνωρίσματα αυτής σε κατάλληλες OWL ιδιότητες τύπου δεδομένων, όπως ακριβώς επιτάσσουν και οι *βασικοί κανόνες*.

4. **Κανόνες ασθενούς τύπου οντοτήτων**. Αυτοί οι κανόνες αναγνωρίζουν ασθενείς τύπους οντοτήτων (weak entity types) και τους ιδιοκτήτες τύπους οντοτήτων (owner entity types) αυτών. Ένας ασθενής τύπος οντοτήτων συνήθως αναπαρίσταται με μια σχέση με σύνθετο πρωτεύον κλειδί που περιέχει ένα ξένο κλειδί προς μια άλλη σχέση, η οποία αναπαριστά τον ιδιοκτήτη τύπο οντοτήτων. Αυτές οι σχέσεις αντιστοιχούν μεν σε OWL κλάσεις, όπως και οι συνηθισμένοι τύποι οντοτήτων, αλλά η σημασία της προσδιορίζουσας συσχέτισης (identifying relationship) που συνδέει μια ασθενή οντότητα με την ιδιοκτήτριά της είναι συνήθως δύσκολο να εξακριβωθεί με επιτυχία. Μερικοί κανόνες επιλέγουν να ερμηνεύσουν αυτή τη συσχέτιση ως ένα ζεύγος αντίστροφων OWL ιδιοτήτων αντικειμένων *«είναι μέρος»* και *«έχει μέρος»*, που αναπαριστούν μια σχέση όλου-μέρους μεταξύ των δύο οντοτήτων.

5. **Κανόνες n-αδικής συσχέτισης.** Κανόνες αυτού του είδους αναγνωρίζουν συσχετίσεις βαθμού $n, n > 2$, όταν υπάρχουν σχέσεις των οποίων το πρωτεύον κλειδί αποτελείται εξ ολοκλήρου από n ξένα κλειδιά προς σχέσεις

που αντιστοιχούν σε OWL κλάσεις. Καθώς δεν υπάρχει ειδικός OWL μηχανισμός για την αναπαράσταση n-αδικών συσχετίσεων, αυτές αναπαριστώνται ως OWL κλάσεις και συνεπώς, οι κανόνες αυτοί ουσιαστικά ακολουθούν τη λογική των *βασικών κανόνων*.

6. **Κανόνες κατακερματισμού.** Αυτοί οι κανόνες αναγνωρίζουν σχέσεις που έχουν υποστεί κάθετη διαμέριση (vertical partitioning) και κατά συνέπεια, αναπαριστούν την ίδια οντότητα. Αυτές οι σχέσεις αντιστοιχούν στην ίδια OWL κλάση και η αναγνώρισή τους βασίζεται στο γεγονός ότι μοιράζονται το ίδιο πρωτεύον κλειδί, όπως ισχύει και στους *κανόνες ιεραρχίας*.

7. **Κανόνες πλειότιμου γνωρίσματος.** Αυτοί οι κανόνες αναγνωρίζουν σχέσεις που αναπαριστούν πλειότιμα γνωρίσματα. Καθώς υπάρχουν αρκετοί τρόποι για να μοντελοποιηθεί ένα πλειότιμο γνώρισμα στο σχεσιακό μοντέλο, η αναγνώρισή του είναι σχετικά δύσκολη υπόθεση. Εντούτοις, οι περισσότεροι από τους κανόνες αυτής της κατηγορίας υποθέτουν ότι μια σχέση με σύνθετο πρωτεύον κλειδί που περιέχει ένα ξένο κλειδί προς μια σχέση R μπορεί να αντιστοιχηθεί σε μια OWL ιδιότητα τύπου δεδομένων με πεδίο ορισμού την κλάση που αντιστοιχεί στη σχέση R. Οι κανόνες αυτοί αναγνωρίζουν τον ίδιο τύπο σχέσεων με τους *κανόνες ασθενούς τύπου οντοτήτων*, με τους οποίους μπορεί να έρθουν σε σύγκρουση.

8. **Κανόνες περιορισμών.** Αυτοί οι κανόνες εκμεταλλεύονται επιπρόσθετους κανόνες του σχεσιακού σχήματος, που μπορεί να υπάρχουν σε SQL εντολές ορισμού δεδομένων. Τέτοιοι περιορισμοί περιλαμβάνουν NOT NULL περιορισμούς, περιορισμούς μοναδικότητας, αλλά και περιορισμούς πεδίου τιμών για τα γνωρίσματα μιας σχέσης. Αυτοί συνήθως μεταφράζονται, αντίστοιχα, σε κατάλληλα αξιώματα πληθικότητας, OWL ιδιότητες των οποίων η ανάστροφη ιδιότητα είναι συναρτησιακή (inverse functional property) και σε καθολικά (`rdfs:range`) ή τοπικά αξιώματα καθολικής ποσοτικοποίησης (π.χ. `owl:allValuesFrom` περιορισμοί) που εφαρμόζονται σε μια συγκεκριμένη OWL κλάση.

9. **Κανόνες τύπου δεδομένων.** Κανόνες αυτού του είδους αποτελούνται ουσιαστικά από πίνακες μετασχηματισμού που ορίζουν αντιστοιχίες μεταξύ SQL τύπων δεδομένων και τύπων δεδομένων του XML Schema, οι οποίοι είναι αυτοί που χρησιμοποιούνται κυρίως από RDF γράφους και OWL οντολογίες. Τέτοιες αντιστοιχίες ορίζονται αναλυτικά στο πρότυπο της SQL [2].

Οι συγκεκριμένες κατηγορίες κανόνων που εφαρμόζονται από καθεμία από τις μεθόδους αυτής της παραγράφου φαίνονται στον πίνακα 3.5, μαζί με σχετική ένδειξη για το αν μια μέθοδος εμπλουτίζει την παραχθείσα οντολογία με άτομα από τη σχεσιακή ΒΔ. Είναι εμφανές ότι ο πίνακας 3.5 προσφέρει μια αρκετά αδρή σύγκριση αυτών των μεθοδολογιών, χωρίς να εμφανίζει τις επιμέρους διαφορές τους, εστιάζοντας για παράδειγμα στο πώς αυτές χειρίζονται τους διάφορους SQL περιορισμούς. Μια αναλυτικότερη επισκόπηση μερικών μόνο από τις προσεγγίσεις που αναφέρουμε σε αυτή την παράγραφο, μαζί με μια λεπτομερή καταγραφή των συγκεκριμένων RDFS και OWL μηχανισμών που χρησιμοποιούνται σε αυτές για την αναπαράσταση στοιχείων

του σχεσιακού σχήματος υπάρχει στο [174]. Προσεγγίσεις που εφαρμόζουν αντικρουόμενους κανόνες, όπως αυτούς που σημειώσαμε παραπάνω, συχνά επαφίενται στην κρίση του ανθρώπινου ειδικού ή εισάγουν και επιπλέον υποθέσεις που διακρίνουν τους επίμαχους κανόνες μεταξύ τους. Παρ' όλα αυτά, σε αυτή την τελευταία περίπτωση, η ορθότητα των συγκεκριμένων κανόνων είναι αμφισβητήσιμη, αφού μπορούν εύκολα να βρεθούν αντι-παραδείγματα σχεσιακών σχημάτων όπου δεν παράγεται το αναμενόμενο αποτέλεσμα.

Πίνακας 3.5: *Κατηγορίες κανόνων που χρησιμοποιούνται από προσεγγίσεις αντίστροφης μηχανικής*

Εργασία	Κατηγορίες κανόνων									Εμπλουτισμός με οντολογικά άτομα
	1	2	3	4	5	6	7	8	9	
Astrova (2004) [15]	✓	✓	✓	✓	✓	✓	–	✓	–	Ναι
Astrova (2009) [16]	✓	✓	✓	✓	✓	–	–	✓	✓	Ναι
Buccella *et al.* (2004) [47]	✓	–	✓	✓	✓	–	–	✓	✓	Όχι
DB2OWL [87]	✓	✓	✓	–	–	–	–	✓	✓	Ναι
DM-2-OWL [7]	✓	✓	✓	–	–	–	–	✓	✓	Όχι
Jurić, Banek, Skočir [109]	✓	–	✓	–	–	–	–	✓	✓	Όχι
Lubyte, Tessaris (2009) [133]	✓	–	✓	–	✓	–	–	✓	–	Όχι
R2O [180]	✓	✓	✓	✓	✓	–	✓	✓	–	Όχι
RDBToOnto [54]	✓	✓	✓	–	–	–	–	–	–	Ναι
ROSEX [66]	✓	–	✓	✓	–	–	–	–	–	Όχι
Shen *και συνεργάτες* (2006) [175]	✓	✓	✓	–	–	✓	–	✓	✓	Ναι
SOAM [129]	✓	✓	✓	✓	✓	✓	–	✓	✓	Ναι
SQL2OWL [6]	✓	✓	✓	–	✓	✓	✓	✓	✓	Προαιρετικά
L. Stojanovic, N. Stojanovic, Volz (2002) [184]	✓	✓	✓	–	✓	✓	–	✓	–	Ναι
Tirmizi, Sequeda, Miranker (2008) [188]	✓	✓	✓	–	✓	–	–	✓	✓	Ναι

Λαμβάνοντας υπόψη ότι το σύνολο των προσεγγίσεων αυτής της παραγράφου βασίζεται σε ευρετικές μεθόδους, η ικανότητά τους να ερμηνεύουν σωστά τη σημασία ενός *οποιουδήποτε* σχεσιακού σχήματος είναι αμφίβολη, καθώς δεν μπορούν να ενσωματώσουν όλες τις πιθανές πρακτικές λογικού σχεδιασμού και να μαντέψουν την αρχική πρόθεση του σχεδιαστή μιας ΒΔ. Τα πιο σημαντικά προβλήματα, για τα οποία κάναμε προηγουμένως νύξη, είναι τα ακόλουθα:

Αναγνώριση ιεραρχίας κλάσεων. Συνήθως, οι ιεραρχίες τύπων οντοτήτων μοντελοποιούνται σε ένα σχεσιακό σχήμα με την παρουσία ενός πρωτεύοντος κλειδιού που ταυτόχρονα αποτελεί και ξένο κλειδί που αναφέρεται στο πρωτεύον κλειδί μιας άλλης σχέσης. Οι *κανόνες ιεραρχίας* που αναφέραμε προηγουμένως στηρίζονται σε αυτή την πρακτική σχεδιασμού. Δυστυχώς, μια παρόμοια συστοιχία σχέσεων μπορεί να αναπαριστά μια κάθετη διαμέριση μιας σχέσης, οπότε η πληροφορία σχετικά με μια οντότητα είναι διασπαρμένη σε περισσότερες από μία σχέσεις. Για τη μοντελοποίηση μιας ιεραρχίας κλάσεων, όμως, υπάρχουν και άλλες εναλλακτικές πρακτικές σχεδίασης που δυσκολεύουν περισσότερο την αναγνώρισή τους [119]. Όπως είδαμε, η εύρεση κατηγορικών γνωρισμάτων μέσω ανάλυσης των περιεχομένων μιας σχέσης μπορεί να αυξήσει τις πιθανότητες επιτυχούς αναγνώρισης μιας ιεραρχίας κλάσεων.

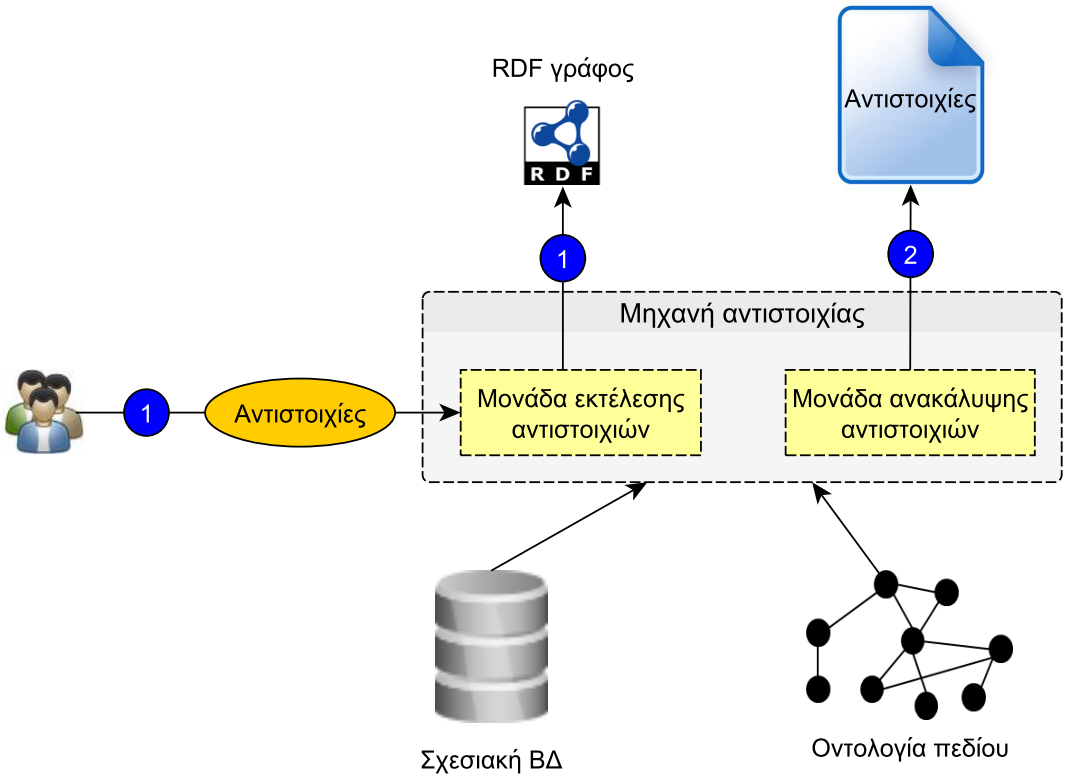
Έκφραση πρωτευόντων κλειδιών. Πρωτεύοντα κλειδιά και γενικότερα, γνωρίσματα που έχουν μοναδικές τιμές συνήθως ερμηνεύονται ως OWL ιδιότητες

τύπων δεδομένων, των οποίων η αντίστροφη ιδιότητα είναι συναρτησιακή. Η εισαγωγή τέτοιων αξιωμάτων οδηγεί σε OWL Full οντολογίες, για τις οποίες φαίνεται να επικρατεί η λανθασμένη αντίληψη ότι η διαδικασία συλλογισμού είναι αδύνατη, δεδομένου του ότι πρόκειται για μια μη αποκρίσιμη διαδικασία. Παρά το γεγονός ότι αρκετά εργαλεία πραγματοποίησης συλλογισμού, όπως παραδείγματος χάριν το Pellet [179], εξάγουν τα αναμενόμενα συμπεράσματα από τέτοια αξιώματα, αρκετές προσεγγίσεις, προκειμένου να αποφύγουν μια OWL Full οντολογία, μοντελοποιούν το πρωτεύον κλειδί μιας σχέσης ως ξεχωριστή κλάση με αποτέλεσμα ο περιορισμός πρωτεύοντος κλειδιού να ερμηνεύεται ως μια αντίστροφα συναρτησιακή *ιδιότητα αντικειμένου*, αξίωμα που επιτρέπεται στα αποκρίσιμα υποσύνολα της OWL. Εναλλακτικά, θα μπορούσε να χρησιμοποιηθεί η ιδιότητα της OWL `hasKey`, η οποία δηλώνει ότι ένα σύνολο ιδιοτήτων αντικειμένου ή τύπου δεδομένων αποτελούν «κλειδί» για μια κλάση, λύση που δεν προτείνεται από κάποια μέθοδο, ακόμα και από αυτές που προτάθηκαν μετά τον ορισμό της OWL 2. Η `owl:hasKey` παρέχει και μια εύκολη λύση για τη μοντελοποίηση σύνθετων πρωτευόντων κλειδιών, χαρακτηριστικό που επίσης δεν έχει ληφθεί υπόψη από κάποια από τις αναφερθείσες μεθοδολογίες.

Εξαγωγή σύνθετων οντολογικών αξιωμάτων. Μέχρι στιγμής, δεν έχουν προταθεί μέθοδοι για την επιτυχημένη εξαγωγή OWL αξιωμάτων που χρησιμοποιούν κάποια από τα πιο εκφραστικά στοιχεία της γλώσσας, όπως π.χ. συμμετρικές και μεταβατικές ιδιότητες ή μονοπάτια ιδιοτήτων, καθώς αυτή η πληροφορία δεν μπορεί να εξαχθεί αποκλειστικά από το σχεσιακό σχήμα. Επιπλέον, φαίνεται ότι, μεταξύ των διαφόρων προσεγγίσεων, υπάρχει διάσταση απόψεων σε ό,τι αφορά στην έκφραση περιορισμών πεδίου τιμών για γνωρίσματα. Κάποιες μέθοδοι προτείνουν την έκφρασή τους ως `rdfs:range` αξιώματα που εφαρμόζονται στην αντίστοιχη OWL ιδιότητα τύπου δεδομένων, ενώ κάποιες άλλες συνιστούν τη χρήση αξιωμάτων καθολικής ποσοτικοποίησης για μια συγκεκριμένη OWL κλάση μέσω του στοιχείου `owl:allValuesFrom`.

3.5 Αντιστοιχία σχεσιακής ΒΔ με υπάρχουσα οντολογία

Στην ενότητα 3.4, εστιάσαμε την προσοχή μας σε εργαλεία και μεθοδολογίες που δεν απαιτούν μια ήδη υπάρχουσα οντολογία για να λειτουργήσουν, αφού έχουν τη δυνατότητα να ορίσουν νέες οντολογικές κλάσεις και ιδιότητες. Στην τρέχουσα ενότητα, εξετάζουμε προσεγγίσεις που θεωρούν δεδομένη την ύπαρξη ενός ή περισσότερων οντολογιών και είτε α) προσπαθούν να ορίσουν αντιστοιχίες μεταξύ του σχήματος μιας σχεσιακής ΒΔ και αυτών των οντολογιών ή β) βασίζονται σε χειροκίνητα ορισμένες αντιστοιχίες προκειμένου να δημιουργήσουν έναν RDF γράφο που χρησιμοποιεί όρους από αυτές τις οντολογίες. Αυτοί οι δύο εναλλακτικοί τρόποι λειτουργίας απεικονίζονται στο σχήμα 3.6 (τρόποι λειτουργίας 2 και 1 αντίστοιχα). Η βασική υπόθεση που γίνεται από όλες τις προσεγγίσεις αυτής της κατηγορίας είναι ότι οι επιλεγείσες οντολογίες μοντελοποιούν τον ίδιο τομέα ενδιαφέροντος με αυτόν που περιγράφει η σχεσιακή ΒΔ.



Σχήμα 3.6: *Αντιστοιχία σχεσιακής ΒΔ με υπάρχουσα οντολογία*

Αφενός λοιπόν, υπάρχουν προσεγγίσεις που εφαρμόζουν αλγορίθμους *αντιστοίχησης σχημάτων* (schema matching) ή εμπλουτίζουν αλγορίθμους αντίστροφης μηχανικής σαν αυτούς της παραγράφου 3.4.2.2 με *δείκτες λεξικολογικής ομοιότητας*, με απώτερο στόχο την ανακάλυψη αντιστοιχιών μεταξύ μιας σχεσιακής ΒΔ και μιας κατάλληλα επιλεγμένης οντολογίας. Αυτές οι αντιστοιχίες, σε δεύτερο στάδιο, μπορούν να χρησιμοποιηθούν σε διάφορα σενάρια και εφαρμογές, στα πλαίσια ολοκλήρωσης ετερογενών ΒΔ ή ακόμα και εύρεσης αντιστοιχιών μεταξύ σχεσιακών σχημάτων [9, 75]. Αφετέρου, υπάρχουν εργαλεία που μοιάζουν αρκετά με τα εργαλεία της παραγράφου 3.4.2.1, καθώς δέχονται χειροκίνητα ορισμένες αντιστοιχίες μεταξύ του σχήματος ΒΔ και μιας ή περισσότερων υπαρχουσών οντολογιών, προκειμένου να παράξουν έναν RDF γράφο που ουσιαστικά εμπλουτίζει το σώμα ισχυρισμών (ABox) αυτών των οντολογιών.

Στη γενικότερη περίπτωση, το σχήμα της ΒΔ και η οντολογία με την οποία αυτό θα αντιστοιχηθεί έχουν αναπτυχθεί ανεξάρτητα, αναμένεται να διαφέρουν σημαντικά και συχνά, οι θεματικές περιοχές που εξετάζουν δεν ταυτίζονται απόλυτα, αλλά επικαλύπτονται. Συνεπώς, οι αντιστοιχίες δεν είναι πάντα τετριμμένες και προφανείς και γενικά, θα εκφράζονται ως σύνθετες εκφράσεις και λογικές φόρμουλες. Εργαλεία αυτής της κατηγορίας πρέπει να είναι σε θέση να *ανακαλύπτουν* και να *εκφράζουν* σύνθετες αντιστοιχίες, με τη διαδικασία της ανακάλυψης να είναι αυτή που θέτει τις περισσότερες προκλήσεις και να είναι δύσκολο να αυτοματοποιηθεί πλήρως. Κατά συνέπεια, η συντριπτική πλειοψηφία των προσεγγίσεων αυτής της κατηγορίας είναι χειροκίνητες ή ημι-αυτόματες. Στην τελευταία περίπτωση, ο αλγόριθμος εύρεσης αντιστοι-

χιών απαιτεί κάποια είσοδο από το χρήστη, συνήθως υπό τη μορφή απλών αντιστοιχών μεταξύ ατομικών στοιχείων του σχεσιακού σχήματος και όρων της οντολογίας. Μια πλήρως αυτοματοποιημένη διαδικασία θα χρειαζόταν να υιοθετήσει υπεραπλουστευτικές υποθέσεις που αφορούν στην λεξικολογική εγγύτητα των ονομάτων που χρησιμοποιούν η ΒΔ και η οντολογία, υποθέσεις που δεν ισχύουν πάντα στη γενική περίπτωση. Τέτοιες υποθέσεις φαίνεται να οδηγούν στην υπερεκτίμηση της αποδοτικότητας μεθόδων ανακάλυψης αντιστοιχιών, όπως το MARSON [105] και το OntoMat-Reverse [197].

Όσον αφορά στον *τρόπο πρόσβασης* στο αποτέλεσμα της αντιστοιχίας, η συγκεκριμένη παράμετρος, όπως ορίστηκε στην ενότητα 3.3, δε βρίσκει εφαρμογή σε μεθόδους αυτής της κατηγορίας που απλά εξάγουν ένα σύνολο λογικών εκφράσεων - αντιστοιχιών (ακολουθώντας τον τρόπο λειτουργίας 2 στο σχήμα 3.6), χωρίς να προχωρούν στον εμπλουτισμό του σώματος ισχυρισμών της οντολογίας με δεδομένα της ΒΔ. Τέτοιες μέθοδοι ούτε εξάγουν τα περιεχόμενα της ΒΔ σε μορφή RDF ούτε προσφέρουν δυναμική πρόσβαση σε αυτά μέσω SPARQL. Αυτός ήταν και ο κυριότερος λόγος για τον οποίο δε χρησιμοποιήθηκε αυτή η παράμετρος ως κριτήριο ταξινόμησης στην ταξινομία του σχήματος 3.2. Εξαιρούμενων των συγκεκριμένων μεθόδων, οι υπόλοιπες αυτής της κατηγορίας υποστηρίζουν και τους τρεις τρόπους πρόσβασης που έχουν αναφερθεί μέχρι τώρα.

Αν και η *γλώσσα αναπαράστασης των αντιστοιχιών* είναι μια πολύ σημαντική παράμετρος για την επαναχρησιμοποίησή τους από τρίτες εφαρμογές, επικρατεί και εδώ μια κατάσταση ανάλογη με αυτή που περιγράφηκε στην παράγραφο 3.4.2.1, δηλαδή η έλλειψη μιας κοινής γλώσσας, αφού κάθε εργαλείο χρησιμοποιεί τον δικό του τρόπο αναπαράστασης. Βέβαια, και σε αυτή την περίπτωση, αναμένεται ότι το επίπεδο υιοθέτησης της R2RML θα ανέβει στο προσεχές μέλλον, αποτελώντας την προφανή λύση για την αναπαράσταση αντιστοιχιών. Όλες οι προσεγγίσεις υποστηρίζουν OWL οντολογίες, ενώ η *επαναχρησιμοποίηση λεξιλογίων* δεν υποστηρίζεται από μερίδα εργαλείων που προβλέπουν την αντιστοιχία με αυστηρά μία οντολογία. Πρέπει επίσης να ληφθεί υπόψη ότι, για τους ίδιους λόγους με παραπάνω, η παράμετρος της επαναχρησιμοποίησης λεξιλογίων όπως ορίστηκε στην ενότητα 3.3 δεν είναι εφαρμόσιμη για μεθόδους που μόνο ανακαλύπτουν αντιστοιχίες, χωρίς να τις εκμεταλλεύονται.

Ένα χαρακτηριστικό γνώρισμα των περισσότερων εργαλείων αυτής της κατηγορίας είναι η παρουσία μιας *γραφικής διεπαφής χρήστη*, ο ρόλος της οποίας είναι, αφενός ο γραφικός προσδιορισμός των απλών αντιστοιχιών μεταξύ ΒΔ και οντολογίας, που αρκετές μέθοδοι χρειάζονται ως είσοδο και ο οποίος διευκολύνεται σημαντικά από τη γραφική απεικόνιση των δομών των δύο μοντέλων και αφετέρου, η επαλήθευση και προσαρμογή των αντιστοιχιών που προτείνονται αυτόματα από τον εκάστοτε αλγόριθμο. Αναλυτικά, οι προσεγγίσεις της κατηγορίας εξετάζονται μία προς μία στη δημοσίευση [182].

Πίνακας 3.6: *Περιγραφικές παράμετροι για τις μεθόδους αντιστοιχίας ΒΔ με οντολογία*

Εργασία	Επίπεδο αυτομα-τοποίησης	Προσβασιμότητα	Γλώσσα αντιστοι-χίας	Γλώσσα οντολο-γίας	Επαναχρησιμοποίηση λεξιλογίων	Διαθεσιμότητα λογισμικού	Γραφική διεπαφή χρήστη	Στόχος
Astrova (2004, 2009) [15, 16]	Ημι-αυτόματος / Αυτόματος	ETL	–	F-Logic / OWL DL	Όχι	Όχι	Όχι	Οντολογική μάθηση
AuReLi [160]	Ημι-αυτόματος	SPARQL / Συνδεδεμένα Δεδομένα	Γλώσσα D2RQ (βασισμένη σε RDF)	OWL	Ναι	Όχι	Ναι	Πρόσβαση με χρήση οντολογιών
Automapper (Asio SBRD) [83]	Ημι-αυτόματος	SPARQL	Ιδιότυπη (βασισμένη σε RDF)	OWL+SWRL	Όχι	Εμπορικό	Ναι	Πρόσβαση με χρήση οντολογιών
Buccella *και συνεργάτες* (2004) [47]	Ημι-αυτόματος	ETL	–	OWL DL	Όχι	Όχι	Όχι	Ολοκλήρωση ΒΔ
CROSS [55]	Ημι-αυτόματος	Ιδιότυπο πρωτόκολλο πρόσβασης	–	OWL	Ναι	Ναι	Όχι	Ολοκλήρωση ΒΔ
D2OMapper [203]	Ημι-αυτόματος	–	Ιδιότυπη αναπαράσταση (XML αρχείο)	OWL DL	Όχι	Όχι	Όχι	Σημασιολογική επισημείωση ιστοσελίδων
D2RQ [41] / D2R Server [39]	Αυτόματος / Χειροκίνητος	ETL / SPARQL / Συνδεδεμένα Δεδομένα	Γλώσσα D2RQ (βασισμένη σε RDF)	RDFS	Ναι	Ναι	Όχι	Πρόσβαση με χρήση οντολογιών
DartGrid [59]	Χειροκίνητος	SPARQL	Ιδιότυπη αναπαράσταση (RDF/XML αρχείο)	OWL DL	Όχι	Όχι	Ναι	Ολοκλήρωση ΒΔ
DataMaster [147]	Αυτόματος	ETL	–	OWL DL/Full	Όχι	Ναι	Ναι	Παραγωγή σημασιολογικού περιεχομένου
DB2OWL [87]	Αυτόματος	ETL / SPARQL	R_2O γλώσσα (βασισμένη σε χείμενο)	OWL DL	Όχι	Όχι	Όχι	Ολοκλήρωση ΒΔ
DM-2-OWL [7]	Αυτόματος	ETL	–	OWL Full	Όχι	Όχι	Όχι	Οντολογική μάθηση
Dolbear, Hart (2008) [74]	Ημι-αυτόματος	SPARQL	–	OWL DL	Όχι	Όχι	Όχι	Ολοκλήρωση χωρικών ΒΔ
FDR2 [117]	Ημι-αυτόματος	ETL	–	RDFS	Ναι	Όχι	Όχι	Σημασιολογική διαλειτουργικότητα ΒΔ

Συνέχεια στην επόμενη σελίδα

Εργασία	Επίπεδο αυτοματοποίησης	Προσβασιμότητα	Γλώσσα αντιστοιχίας	Γλώσσα οντολογίας	Επαναχρησιμοποίηση λεξιλογίων	Διαθεσιμότητα λογισμικού	Γραφική διεπαφή χρήστη	Στόχος
Jurić, Banek, Skočir [109]	Αυτόματος	ETL	–	OWL Full	Όχι	Όχι	Όχι	Οντολογική μάθηση
Kupfer *και συνεργάτες* (2006) [118]	Ημι-αυτόματος	ETL	–	OWL-Lite	Ναι	Όχι	Όχι	Ολοκλήρωση ΒΔ
Lausen (2007) [120]	Αυτόματος	ETL	–	RDFS	Όχι	Όχι	Όχι	Ολοκλήρωση ΒΔ
Levshin (2009) [126]	Αυτόματος	ETL	–	OWL+SWRL	Όχι	Όχι	Όχι	Έλεγχος περιορισμών
Linked Data Mapper [208]	Χειροκίνητος	SPARQL	Virtuoso Meta-Schema Language	OWL	Όχι	Ναι	Ναι	Ολοκλήρωση ΒΔ
Lubyte, Tessaris (2009) [133]	Ημι-αυτόματος	SPARQL	–	*DLR-DB*	Όχι	Όχι	Όχι	Πρόσβαση με χρήση οντολογιών
MAPONTO [10]	Ημι-αυτόματος	–	Ιδιότυπη αναπαράσταση (XML αρχείο)	OWL DL	Όχι	Ναι	Ναι	Ορισμός νοήματος ΒΔ
MARSON [105]	Αυτόματος	–	Ιδιότυπη αναπαράσταση (XML αρχείο)	OWL DL	Όχι	Όχι	Όχι	Σημασιολογική διαλειτουργικότητα ΒΔ
MASTRO [51] / OBDA plugin για Protégé [159]	Χειροκίνητος	SPARQL	Ιδιότυπη αναπαράσταση	*DL-Lite*$_A$ / OWL 2 QL	Όχι	Ναι	Ναι	Πρόσβαση με χρήση οντολογιών
METAmorphoses [185]	Χειροκίνητος	ETL	Ιδιότυπη (βασισμένη σε XML)	RDF	Ναι	Ναι	Ναι	Παραγωγή σημασιολογικού περιεχομένου
OntoAccess [100]	Χειροκίνητος	SPARQL Update	Ιδιότυπη (βασισμένη σε XML)	RDFS	Ναι	Ναι	Όχι	Ενημέρωση με χρήση οντολογιών
OntoMat-Reverse [197] (OntoMat-Annotizer)	Ημι-αυτόματος	ETL/F-Logic	F-Logic	F-Logic	Όχι	Ναι	Ναι	Σημασιολογική επισημείωση ιστοσελίδων
R2O [180]	Ημι-αυτόματος	ETL	–	OWL DL	Όχι	Όχι	Όχι	Ολοκλήρωση ΒΔ
R$_2$O / ODEMapster [28]	Χειροκίνητος	ETL / Ιδιότυπη γλώσσα ερωτημάτων (ODEMQL)	R$_2$O γλώσσα (βασισμένη σε κείμενο)	OWL	Ναι	Ναι	Όχι	Πρόσβαση με χρήση οντολογιών
RDB2ONT [189]	Αυτόματος	ETL	–	OWL Full	Όχι	Όχι	Όχι	Ολοκλήρωση ΒΔ
RDB2OWL [48]	Χειροκίνητος	ETL	SQL	OWL	Ναι	Όχι	Όχι	Παραγωγή σημασιολογικού περιεχομένου

Συνέχεια στην επόμενη σελίδα

Εργασία	Επίπεδο αυτοματοποίησης	Προσβασιμότητα	Γλώσσα αντιστοίχισης	Γλώσσα οντολογίας	Επανεχρησιμοποίηση λεξιλογίων	Διαθεσιμότητα λογισμικού	Γραφική διεπαφή χρήστη	Στόχος
RDBToOnto [54]	Ημι-αυτόματος	ETL	–	OWL	Όχι	Ναι	Ναι	Οντολογική μάθηση
RDOTE [193]	Χειροκίνητος	ETL	SQL	OWL	Ναι	Ναι	Ναι	Παραγωγή σημασιολογικού περιεχομένου
Relational.OWL [153] / RDQuery [155]	Αυτόματος	ETL / SPARQL	–	OWL Full	Όχι	Ναι	Ναι	Ανταλλαγή δεδομένων σε δίκτυα ομοτίμων
RONTO [149]	Ημι-αυτόματος	–	Ιδιότυπη αναπαράσταση	RDFS/OWL	Όχι	Όχι	Ναι	Παραγωγή σημασιολογικού περιεχομένου
ROSEX [66]	Αυτόματος	SPARQL	Ιδιότυπη οντολογία	OWL DL	Όχι	Όχι	Όχι	Πρόσβαση με χρήση οντολογιών
Shen και συνεργάτες (2006) [175]	Αυτόματος	ETL	–	OWL DL	Όχι	Όχι	Όχι	Ολοκλήρωση ΒΔ
SOAM [129]	Ημι-αυτόματος	ETL	–	OWL DL	Όχι	Όχι	Όχι	Οντολογική μάθηση
SQL2OWL [6]	Ημι-αυτόματος	ETL	–	OWL DL	Όχι	Όχι	Ναι	Οντολογική μάθηση
Spyder	Αυτόματος / Χειροκίνητος	ETL / SPARQL	Ιδιότυπη (βασισμένη σε RDF) / R2RML	RDFS	Ναι	Ναι	Όχι	Πρόσβαση με χρήση οντολογιών
SquirrelRDF [171]	Αυτόματος / Χειροκίνητος	SPARQL	Ιδιότυπη (βασισμένη σε RDF)	RDFS	Ναι	Ναι	Όχι	Πρόσβαση με χρήση οντολογιών
StdTrip [168]	Ημι-αυτόματος	ETL	Ιδιότυπη αναπαράσταση / SQL	OWL	Ναι	Όχι	Ναι	Παραγωγή σημασιολογικού περιεχομένου
L. Stojanovic, N. Stojanovic, Volz (2002) [184]	Ημι-αυτόματος	ETL	–	F-Logic / RDFS	Όχι	Όχι	Όχι	Σημασιολογική επισημείωση ιστοσελίδων
Tether [49]	Ημι-αυτόματος	ETL	–	RDFS	Ναι	Όχι	Όχι	Παραγωγή σημασιολογικού περιεχομένου
Tirmizi, Sequeda, Miranker (2008) [188]	Αυτόματος	ETL	–	OWL DL	Όχι	Όχι	Όχι	Παραγωγή σημασιολογικού περιεχομένου
Triplify [18]	Χειροκίνητος	ETL / Συνδεδεμένα Δεδομένα	SQL	RDFS	Ναι	Ναι	Όχι	Παραγωγή σημασιολογικού περιεχομένου

Συνέχεια στην επόμενη σελίδα

Εργασία	*Επίπεδο αυτοματοποίησης*	*Προσβασιμότητα*	*Γλώσσα αντιστοιχίας*	*Γλώσσα οντολογίας*	*Επανεχρησιμοποίηση λεξιλογίων*	*Διαθεσιμότητα λογισμικού*	*Γραφική διεπαφή χρήστη*	*Στόχος*
Ultrawrap [172]	Αυτόματος	SPARQL	–	OWL DL	Όχι	Όχι	Όχι	Πρόσβαση με χρήση οντολογιών
Virtuoso RDF Views [42]	Αυτόματος / Χειροκίνητος	ETL / SPARQL / Συνδεδεμένα Δεδομένα	Virtuoso Meta-Schema Language	RDFS	Ναι	Ναι	Ναι	Πρόσβαση με χρήση οντολογιών
VisAVis [114]	Χειροκίνητος	RDQL	SQL, OWL	OWL DL	Όχι	Ναι	Ναι	Ορισμός νοήματος ΒΔ

3.6 Σύνοψη και συμπεράσματα

Έχοντας εξετάσει τα διαφορετικά προβλήματα που εντάσσονται στο γενικότερο πρόβλημα της αντιστοιχίας σχεσιακής ΒΔ με οντολογίας και αναφερθεί στις κυριότερες προτεινόμενες λύσεις, σε αυτή την ενότητα επιχειρούμε να συνοψίσουμε μερικά από τα σημαντικότερα σημεία αυτού του κεφαλαίου και να σταθούμε στις *προκλήσεις* που αντιμετωπίζει κάθε κατηγορία προσεγγίσεων. Επίσης, εξετάζουμε το θεμελιώδες ζήτημα της *αξιολόγησης* μιας μεθόδου, τονίζοντας ότι είναι πολύ δύσκολη, αν όχι αδύνατη, η ανάπτυξη μιας κατάλληλης μεθοδολογίας που θα είναι εφαρμόσιμη για όλες τις μεθόδους που αναφέρθηκαν στο τρέχον κεφάλαιο. Αυτό ισχύει, διότι, όπως έχει ήδη γίνει φανερό από τις προηγούμενες παραγράφους, ορισμένες κατηγορίες μεθόδων μπορούν να αξιολογηθούν μόνο σε ποιοτική βάση και πολύ δύσκολα, σε όρους ενός αριθμητικού μεγέθους.

Παρατηρώντας τον πίνακα 3.6, γίνεται φανερή η μειωμένη διαθεσιμότητα λογισμικού των σχετικών μεθοδολογιών, καθώς λιγότερες από τις μισές εξ αυτών συνοδεύονται από ελεύθερα διαθέσιμο λογισμικό ενώ ακόμα λιγότερες συντηρούνται ενεργά έχοντας φτάσει σε σχετική ωριμότητα. Μπορούμε ακόμα να παρατηρήσουμε την ύπαρξη *αρνητικής συσχέτισης μεταξύ του βαθμού αυτοματοποίησης μιας μεθόδου και της «γνωσιακής ικανότητας» της μεθόδου*, με άλλα λόγια της ικανότητας της μεθόδου να εξάγει τη σωστή σημασία των περιεχομένων μιας ΒΔ. Αυτό ισχύει κυρίως επειδή ο ανθρώπινος ειδικός είναι συνήθως η πιο αξιόλογη πηγή πληροφορίας για τη σημασία ενός σχεσιακού σχήματος. Ως εκ τούτου, πλήρως χειροκίνητες μέθοδοι, όπου οι αντιστοιχίες προσδιορίζονται από το χρήστη, εξ ορισμού ερμηνεύουν ορθά το σχήμα μιας σχεσιακής ΒΔ, ενώ σπανίως πλήρως αυτόματες μέθοδοι έχουν την ίδια αποτελεσματικότητα.

Προχωρώντας σε μια απαρίθμηση των προκλήσεων και σε παρατηρήσεις που αφορούν στην αξιολόγηση κάθε κατηγορίας μεθόδων που αναφέρθηκε σε αυτό το κεφάλαιο και αρχίζοντας από μεθόδους που **δημιουργούν μια οντολογία σχεσιακού σχήματος**, θα μπορούσαμε να πούμε ότι η δημιουργία μιας οντολογίας σχεσιακού σχήματος σπάνια αποτελεί από μόνη της στόχο, άποψη που ενισχύεται και από την πλειονότητα των σχετικών μεθόδων που εξετάστηκαν. Όπως αναφέρθηκε και στην παράγραφο 3.4.1, μια οντολογία σχεσιακού σχήματος απλά αντικατοπτρίζει πλήρως το σχήμα και, σε μερικές περιπτώσεις, τα δεδομένα μιας σχεσιακής ΒΔ, χωρίς να περιέχει επιπρόσθετη γνώση από κάποια άλλη θεματική περιοχή. Συνήθως βέβαια, εξωτερικές εφαρμογές αλλά και ανθρώπινοι χρήστες προτιμούν να αλληλεπιδρούν με μια ΒΔ χρησιμοποιώντας έννοιες υψηλού επιπέδου, αγνοώντας τις λεπτομέρειες αποθήκευσης των δεδομένων της. Αυτός είναι και ο λόγος που οι περισσότερες προσεγγίσεις συμπληρώνουν την οντολογία σχεσιακού σχήματος με επιπλέον αξιώματα που συσχετίζουν τους όρους της με όρους από μια εξωτερική οντολογία πεδίου, ενώ παράλληλα προσφέρουν δυναμική πρόσβαση στα δεδομένα της ΒΔ επιτρέποντας την απάντηση ερωτημάτων που χρησιμοποιούν όρους αυτής της οντολογίας πεδίου. Η μετατροπή αυτών των σημασιολογικών ερωτημάτων σε SQL ερωτήματα πραγματοποιείται λαμβάνοντας υπόψη τα αξιώματα αντιστοιχίας μεταξύ της οντολογίας σχεσιακού σχήματος και της οντολογίας πεδίου. Εντούτοις μέχρι σήμερα, δεν έχει αναφερθεί κάποιο πρακτικό πλεονέκτημα που να

δικαιολογεί αυτή τη χρήση της οντολογίας σχεσιακού σχήματος ως ενός ενδιάμεσου επιπέδου στη διαδικασία επανεγγραφής ερωτημάτων, σε σχέση με τον ορισμό απευθείας αντιστοιχίας μεταξύ μιας ΒΔ και μιας οντολογίας πεδίου.

Αντίθετα, μέθοδοι που εκμεταλλεύονται την απεικόνιση του σχεσιακού σχήματος σε μια οντολογική δομή, είτε εφαρμόζοντας διαδικασίες συλλογισμού σε αυτή είτε εκτελώντας κατάλληλα SPARQL ερωτήματα, είναι αυτές που πραγματοποιούν *έλεγχο των περιορισμών ακεραιότητας του σχεσιακού σχήματος* (π.χ. [121, 126]). Παράλληλα, η απεικόνιση των περιορισμών του σχεσιακού σχήματος σε μορφή RDF μπορεί να χρησιμοποιηθεί για την απλοποίηση SPARQL ερωτημάτων [121]. Η πρόκληση στις μεθόδους αυτού του είδους είναι, αφενός, η συμφιλίωση της σημασιολογίας κλειστού κόσμου των συστημάτων ΒΔ με τη σημασιολογία ανοικτού κόσμου που υιοθετούν τα οντολογικά μοντέλα και, αφετέρου, η εύρεση μιας κατάλληλης οντολογικής αναπαράστασης που διατηρεί την πολυπλοκότητα των υπηρεσιών συλλογισμού εντός αποδεκτών ορίων. Οι μέθοδοι που ασχολούνται με αυτό το πρόβλημα χρησιμοποιούν μια σειρά τεχνολογιών, όπως κανόνες, SPARQL ερωτήματα, καθώς και τους μηχανισμούς μεταμοντελοποίησης και ανάστροφα συναρτησιακών ιδιοτήτων τύπου δεδομένων της OWL. Εντούτοις, υπάρχουν σοβαρές αμφιβολίες σχετικά με την πρακτική χρησιμότητα αυτής της μικρής ομάδας μεθόδων, δεδομένου του ότι ο έλεγχος των περιορισμών ακεραιότητας σε ένα σχεσιακό σχήμα θεωρείται αναπόσπαστο κομμάτι της λειτουργικότητας ενός ΣΔΒΔ, χωρίς να αποτελεί παράγοντα που επηρεάζει σημαντικά την επίδοσή του, ενώ είναι μάλλον απίθανο ο έλεγχος μέσω διαδικασιών συλλογισμού - οι οποίες σε γενικές γραμμές είναι διαδικασίες υψηλής υπολογιστικής πολυπλοκότητας - να αποδειχθεί συμφερότερος σε σχέση με αυτόν που πραγματοποιείται από ένα ΣΔΒΔ. Μάλιστα, καμία από τις εν λόγω μελέτες δεν προσφέρει μια πρακτική αξιολόγηση των αλγορίθμων που προτείνει. Μια αντικειμενική αξιολόγηση αυτών των μεθόδων θα περιλάμβανε τη μέτρηση του συνολικού χρόνου για τον έλεγχο των περιορισμών ακεραιότητας, σε σχεσιακά σχήματα με κλιμακούμενο αριθμό σχέσεων αλλά και περιορισμών πρωτευόντων και ξένων κλειδιών, και σύγκρισή του με τον αντίστοιχο χρόνο που επιτυγχάνεται από ένα ΣΔΒΔ.

Όσον αφορά σε προσεγγίσεις που **παράγουν μια νέα οντολογία πεδίου** (παράγραφος 3.4.2), οι κύριες προκλήσεις που αυτές αντιμετωπίζουν περιλαμβάνουν: α) τον ορισμό εκφραστικών γλωσσών αντιστοιχιών που επιτρέπουν την πλήρως προσαρμοζόμενη εξαγωγή των περιεχομένων μιας ΒΔ σε έναν RDF γράφο, αλλά και την αναπαράσταση της απαραίτητης πληροφορίας για αλγορίθμους επανεγγραφής ερωτημάτων, β) η αναζήτηση αποδοτικών αλγορίθμων επανεγγραφής ερωτημάτων, γ) η εξαγωγή γνώσης από ένα σχεσιακό σχήμα και δ) ο, όσο το δυνατόν περισσότερο αυτοματοποιημένος, εμπλουτισμός της παραχθείσας οντολογίας με γνώση από άλλες πηγές. Κάθε μέθοδος της κατηγορίας ασχολείται με μερικούς μόνο από τους παραπάνω γενικούς στόχους, αναλόγως του κινήτρου που υιοθετεί. Έτσι, οι 2 πρώτες προκλήσεις αναφέρονται κατά κύριο λόγο σε εργαλεία της παραγράφου 3.4.2.1, η πλειοψηφία των οποίων προσφέρει δυναμική πρόσβαση στο αποτέλεσμα της αντιστοιχίας μέσω ερωτημάτων, ενώ οι επόμενες 2 προκλήσεις απασχολούν τις προσεγγίσεις της παραγράφου 3.4.2.2.

Ο ορισμός νέων εκφραστικών και φιλικών προς το χρήστη γλωσσών αναπαράστασης αντιστοιχιών ΒΔ με οντολογίες υπήρξε ένα ανοικτό θέμα που

απασχόλησε αρκετούς ερευνητές στο, όχι και τόσο μακρινό, παρελθόν. Στην παρούσα φάση βέβαια, το θέμα αυτό έχει φύγει από το επίκεντρο του ενδιαφέροντος, καθώς η διαδικασία προτυποποίησης της R2RML βρίσκεται σε εξέλιξη. Αντίθετα, η έρευνα που αφορά σε αποδοτικούς αλγορίθμους επανεγγραφής ερωτημάτων είναι συνεχιζόμενη και ιδιαίτερα κρίσιμη για εργαλεία που παρέχουν **πρόσβαση στα περιεχόμενα μιας ΒΔ μέσω οντολογιών**. Για εργαλεία αυτού του είδους, ο αλγόριθμος επανεγγραφής ερωτημάτων αποτελεί τον σημαντικότερο παράγοντα επιτυχίας. Επίσης, η εμφάνιση αποδοτικών αλγορίθμων για το μετασχηματισμό ενός SPARQL ερωτήματος σε ένα σημασιολογικά ισοδύναμο SQL ερώτημα (π.χ. [57, 67, 77, 132, 162]) οδήγησε στη θεώρηση τέτοιων εργαλείων ως μια υποκατάστατη λύση αντί των γνωστών triple store συστημάτων, τουλάχιστον για την περίπτωση όπου τα πρωτογενή δεδομένα δεν είναι σε RDF μορφή, αλλά αποθηκευμένα σε κάποιο σχεσιακό ΣΔΒΔ. Δεδομένου του ότι τα σχεσιακά ΣΔΒΔ, χάρη στις ώριμες στρατηγικές βελτιστοποίησης που διαθέτουν, πετυχαίνουν έως και 100 φορές καλύτερους χρόνους απόκρισης ερωτημάτων από triple store συστήματα [40], είναι λογικό ότι, αν ένας αλγόριθμος επανεγγραφής SPARQL σε SQL δεν εισάγει μεγάλη καθυστέρηση, ο συνδυασμός ενός ΣΔΒΔ και ενός εργαλείου αντιστοιχίας ΒΔ με οντολογία θα επιτυγχάνει καλύτερες επιδόσεις από ένα triple store σύστημα. Πράγματι, αυτό ισχύει για δύο τέτοια εργαλεία, το D2R Server [39] και τη λειτουργικότητα RDF όψεων του Virtuoso Universal Server [42], τα οποία, σύμφωνα με μετρήσεις, πετυχαίνουν ταχύτερους χρόνους απόκρισης από triple store συστήματα, για ογκώδη σύνολα δεδομένων [40]. Επιπλέον, είναι αναμενόμενο ότι, και για την περίπτωση R2RML αντιστοιχιών, θα προσαρμοστούν οι ήδη υπάρχοντες ή θα αναπτυχθούν νέοι αλγόριθμοι επανεγγραφής SPARQL σε SQL.

Παρά το γεγονός ότι η αποδοτικότητα συστημάτων που παρέχουν πρόσβαση στα περιεχόμενα μιας σχεσιακής ΒΔ μέσω SPARQL μπορεί να ποσοτικοποιηθεί μετρώντας το συνολικό χρόνο απόκρισης, δεν είναι εύκολη υπόθεση ο καθορισμός μιας *αντικειμενικής μεθοδολογίας μέτρησης επιδόσεων* (benchmark) για την επισταμένη και αμερόληπτη σύγκριση τέτοιων συστημάτων. Είναι χαρακτηριστικό το γεγονός ότι ορισμένες από τις σχετικές μεθοδολογίες αξιολόγησης που έχουν προταθεί εμφανίζουν αντιφατικά αποτελέσματα, σε ό,τι αφορά στις επιδόσεις triple store συστημάτων και συστημάτων αντιστοιχίας στην εκτέλεση SPARQL ερωτημάτων [40, 90]. Μια συνηθισμένη στρατηγική που ακολουθείται από μεθοδολογίες μέτρησης επιδόσεων είναι η κατασκευή αλγορίθμων που παράγουν συνθετικά σύνολα δεδομένων με κλιμακούμενο μέγεθος και διάφορες επιθυμητές ιδιότητες (π.χ. αναλογία ιδιοτήτων ανά άτομο, ύψος και εύρος ιεραρχίας κλάσεων) και η πρόταση σειράς SPARQL ερωτημάτων κλιμακούμενης πολυπλοκότητας, τα οποία εκτελούνται στα παραγόμενα σύνολα δεδομένων. Η αποτελεσματικότητα ενός συστήματος αξιολογείται με βάση το χρόνο απόκρισης για ένα μείγμα συγκεκριμένων SPARQL ερωτημάτων ή, ισοδύναμα, με βάση τον αριθμό των ερωτημάτων που προλαβαίνουν να εκτελεστούν σε ένα δεδομένο χρονικό διάστημα. Μια τέτοια διαδικασία προτείνεται από το Berlin SPARQL Benchmark [40], μια από τις ευρύτερα αναγνωρισμένες μεθοδολογίες μέτρησης επιδόσεων για συστήματα αποθήκευσης που παρέχουν ένα τελικό σημείο SPARQL (SPARQL endpoint) και στα οποία περιλαμβάνονται και εργαλεία αντιστοιχίας ΒΔ με RDF που παρέ-

χουν δυναμική SPARQL πρόσβαση. Μια άλλη παρόμοια μεθοδολογία είναι η SP^{2}Bench [169], η οποία μπορεί να εφαρμοστεί τόσο σε triple store συστήματα όσο και σε συστήματα επανεγγραφής SPARQL σε SQL, έστω και αν κανένα από τα τελευταία δεν έχει περιληφθεί στη σχετική μελέτη.

Ένα ποσοτικό μέγεθος που βρίσκει εφαρμογή κυρίως σε προσεγγίσεις που *υλοποιούν* τον παραγόμενο RDF γράφο είναι ο χρόνος που χρειάζεται για την παραγωγή μιας RDF πρότασης. Δύο από τους παράγοντες που επηρεάζουν αυτή τη μεταβλητή είναι το συνολικό μέγεθος και η πολυπλοκότητα του τελικού RDF γράφου [186]. Συνεπώς, και σε αυτή την περίπτωση, μια κατάλληλη μεθοδολογία μέτρησης επιδόσεων θα πρέπει να θεωρήσει RDF γράφους διαφόρων μεγεθών και πολυπλοκότητας. Παρ' όλα αυτά, η συγκεκριμένη παράμετρος συνήθως δεν είναι τόσο σημαντική, ιδιαίτερα για μεθόδους όπου κυριαρχεί το κίνητρο της οντολογικής μάθησης. Σε τέτοιες περιπτώσεις, ο χρόνος παραγωγής της οντολογίας δεν αποτελεί τον καθοριστικότερο παράγοντα επιτυχίας, αφού η συγκεκριμένη διαδικασία προβλέπεται να πραγματοποιηθεί μόνο μία φορά. Αντίθετα, αυτό που ενδιαφέρει περισσότερο τον τελικό χρήστη και, ταυτόχρονα, αποτελεί πρόκληση για αυτή την κατηγορία μεθόδων είναι η ικανότητα εξαγωγής θεματικής γνώσης από το στιγμιότυπο μιας σχεσιακής ΒΔ ή ενδεχομένως και από επιπρόσθετες εξωτερικές πηγές. Με βάση τους παραπάνω στόχους, γίνεται κατανοητό ότι η μέτρηση της αποδοτικότητας **εργαλείων που «μαθαίνουν» μια οντολογία από μια σχεσιακή ΒΔ** δεν είναι εύκολο να ποσοτικοποιηθεί και να αξιολογηθεί αντικειμενικά. Κάποια από αυτά τα συστήματα αξιολογούν την αποδοτικότητά τους, συγκρίνοντας την παραγόμενη οντολογία με ένα γνωστό εκ των προτέρων μοντέλο ΟΣ που αντιστοιχεί στη σχεσιακή ΒΔ στην οποία εφαρμόζεται ο αλγόριθμός τους. Με βάση αυτό το σκεπτικό, θα μπορούσαν να οριστούν ζεύγη σχεσιακών σχημάτων και αντίστοιχων οντολογιών που αντικατοπτρίζουν πλήρως το νόημα των πρώτων. Ιδανικά, τα ζεύγη αυτά θα πρέπει να περιλαμβάνουν σχεσιακά σχήματα που ακολουθούν ποικιλία, συνηθισμένων και μη, πρακτικών σχεδιασμού, έτσι ώστε να καλύπτεται ένα σημαντικό εύρος περιπτώσεων ελέγχου. Αυτά τα ζεύγη σχεσιακών σχημάτων - οντολογιών θα μπορούσαν να αποτελέσουν τη βάση μιας μεθοδολογίας μέτρησης επιδόσεων, όπου η αποδοτικότητα ενός συστήματος οντολογικής μάθησης θα εξαρτάται από το ποσοστό των οντολογικών δομών που αναγνωρίστηκαν σωστά σε κάθε περίπτωση ελέγχου.

Παρόμοιες συλλογές από ζεύγη σχεσιακών σχημάτων και οντολογιών θα ήταν ιδιαίτερα χρήσιμες και για την αξιολόγηση εργαλείων που **ανακαλύπτουν αντιστοιχίες μεταξύ μιας σχεσιακής ΒΔ και μιας υπάρχουσας οντολογίας** (βλέπε ενότητα 3.5). Μάλιστα, μια τέτοια μικρή συλλογή δημιουργήθηκε για την αξιολόγηση του MAPONTO [10] και χρησιμοποιήθηκε και κατά την αξιολόγηση του MARSON [105]. Η συγκεκριμένη συλλογή σχεσιακών σχημάτων και οντολογιών σε συνδυασμό με τη σχετική συλλογή της μεθοδολογίας μέτρησης επιδόσεων συστημάτων αντιστοιχίας Thalia[5] θα μπορούσε να αποτελέσει τη βάση μιας πλήρους συλλογής περιπτώσεων που θα περιλαμβάνει περισσότερο

[5]Η εν λόγω συλλογή ζευγών σχεσιακών σχημάτων και οντολογιών βασίστηκε στο σύνολο σχεσιακών σχημάτων που προβλέπονται από την ομώνυμη αυθεντική διαδικασία μέτρησης επιδόσεων συστημάτων ολοκλήρωσης ΒΔ. Η παραλλαγή για συστήματα αντιστοιχίας είναι διαθέσιμη στο http://www.w3.org/wiki/TaskForces/CommunityProjects/LinkingOpenData/THALIATestbed.

σύνθετα σχεσιακά σχήματα που χρησιμοποιούνται σε πραγματικές εφαρμογές και οντολογίες από διάφορες θεματικές περιοχές.

Καθώς ο στόχος **εργαλείων ανακάλυψης αντιστοιχιών** είναι ο όσο το δυνατόν μεγαλύτερος βαθμός αυτοματοποίησης, μια πρόκληση που αυτά αντιμετωπίζουν είναι, αφενός, η μείωση της εξάρτησης από τεχνικές υπολογισμού λεξικολογικής ομοιότητας οι οποίες μειώνουν την αποτελεσματικότητά τους και αφετέρου, η χρήση κατάλληλων λεξικολογικών πηγών για την αντιμετώπιση του προβλήματος ομωνυμίας και συνωνυμίας, καθώς και η χρήση τεχνικών που συγκρίνουν τη δομή των δύο μοντέλων. Σε ό,τι αφορά σε εργαλεία που επιτρέπουν το χειροκίνητο ορισμό αντιστοιχιών μεταξύ μιας σχεσιακής ΒΔ και μιας υπάρχουσας οντολογίας - τα οποία επίσης αναφέρονται στην ενότητα 3.5 - οι προκλήσεις είναι λίγο ως πολύ οι ίδιες με αυτές των εργαλείων της παραγράφου 3.4.2.1, δηλαδή η έρευνα εκφραστικών γλωσσών αναπαράστασης αντιστοιχιών και η αποδοτική επανεγγραφή ερωτημάτων για όσα εργαλεία υποστηρίζουν δυναμική πρόσβαση μέσω οντολογίας. Εκτός από αυτούς τους στόχους, εργαλεία αυτής της κατηγορίας αντιμετωπίζουν την πρόκληση παροχής ενός διαισθητικού γραφικού περιβάλλοντος που θα επιτρέψει, ακόμα και στον αδαή χρήστη, τον ορισμό πολύπλοκων αντιστοιχιών, χωρίς να είναι απαραίτητο αυτός να γνωρίζει τις λεπτομέρειες της εκάστοτε γλώσσας αναπαράστασης. Αυτός ο παράγοντας αποτελεί σημαντική προϋπόθεση για την ευρεία χρήση και αποδοχή τέτοιων εργαλείων, αλλά συνήθως δε δίνεται τόσο βάρος σε αυτόν όσο σε παράγοντες σχετικούς με την αποδοτικότητα του συστήματος.

Τέλος, εκτός από τις ήδη αναφερθείσες δυσκολίες στη συγκρότηση μιας αντικειμενικής μεθοδολογίας αξιολόγησης, σημειώνουμε και την έλλειψη μιας κοινής διεπαφής για τις υλοποιήσεις των μεθόδων αντιστοιχίας ΒΔ με οντολογία. Με εξαίρεση εργαλεία τα οποία ακολουθούν το SPARQL πρωτόκολλο και για τα οποία είναι εύκολο να υλοποιηθεί μια εφαρμογή μέτρησης επιδόσεων που να αλληλεπιδρά με αυτά, οι υλοποιήσεις των υπόλοιπων κατηγοριών που εξετάσαμε παρουσιάζουν ποικιλία μορφών εισόδου και εξόδου, δηλαδή διαφορετικές οντολογικές γλώσσες ή/και διαφορετικές γλώσσες αναπαράστασης αντιστοιχιών, γεγονός που κάνει τη σύγκρισή τους ακόμα πιο δύσκολη.

3.7 Μελλοντικές κατευθύνσεις

Το τρέχον κεφάλαιο ολοκληρώνεται με την παρουσίαση μερικών προβλημάτων στα οποία ακόμα δεν έχει δοθεί ιδιαίτερο βάρος στη σχετική βιβλιογραφία της θεώρησης σχεσιακών ΒΔ στο πλαίσιο του Σημασιολογικού Ιστού.

1. **Ενημέρωση δεδομένων με χρήση οντολογιών.** Ένας μεγάλος αριθμός προσεγγίσεων που αναφέρθηκαν παρέχουν πρόσβαση στα περιεχόμενα μιας ΒΔ μέσω SPARQL. Εντούτοις, αυτή η πρόσβαση είναι μιας κατεύθυνσης. Με την προτυποποίηση της SPARQL Update, που επιτρέπει την ενημέρωση και τροποποίηση RDF γράφων, η ιδέα επανεγγραφής SPARQL Update λειτουργιών σε αντίστοιχες SQL εντολές που θα εκτελούνται στην υποκείμενη ΒΔ γίνεται ολοένα και πιο δημοφιλής. Κάποιες προκαταρκτικές διερευνήσεις του θέματος και αντίστοιχες υλοποιήσεις έχουν προταθεί, όπως το clj-r2rml [84], το OntoAccess [100] και οι επεκτάσεις του

D2RQ εργαλείου, D2RQ/Update [76] και D2RQ++ [165], με διάφορα μειονεκτήματα το καθένα. Για παράδειγμα, το D2RQ++ χρησιμοποιεί κατά τη λειτουργία του και ένα βοηθητικό triple store σύστημα, γεγονός που εισάγει το πρόβλημα του συγχρονισμού των δεδομένων μεταξύ triple store και ΒΔ, ενώ το D2RQ/Update δεν ακολουθεί πλήρως τη σημασιολογία της SPARQL Update κατά τη λειτουργία της εισαγωγής. Επίσης, οι περισσότερες προσεγγίσεις θεωρούν μόνο απλές αντιστοιχίες «σχέσης σε κλάση και γνωρίσματος σε ιδιότητα». Το πρόβλημα της ενημέρωσης σχεσιακών δεδομένων μέσω SPARQL Update είναι παρόμοιο με το κλασικό πρόβλημα της *ενημέρωσης όψεων σε βάσεις δεδομένων*, οπότε η εκμετάλλευση και προσαρμογή των σχετικών ερευνητικών προσπαθειών στο καινούριο αυτό πλαίσιο αναμένεται να βοηθήσει αρκετά στην επίλυση αυτού του προβλήματος.

2. **Ενημέρωση αντιστοιχίας.** Στη γενική περίπτωση, τόσο το σχήμα μιας σχεσιακής ΒΔ όσο και μια οντολογία δεν μπορούν να θεωρηθούν αμετάβλητα, αφού οι απαιτήσεις μιας εφαρμογής μπορεί να αλλάξουν απαιτώντας αντίστοιχες μεταβολές σε κάποιο από τα δύο μοντέλα. Αυτό σημαίνει ότι αντιστοιχίες που έχουν ήδη οριστεί μεταξύ των δύο θα πρέπει να προσαρμοστούν κατάλληλα και όχι να οριστούν ή ανακαλυφθούν εκ νέου. Για μεθόδους που παράγουν μια νέα βάση γνώσης από μια σχεσιακή ΒΔ, το ζήτημα της προσαρμογής του σώματος ορολογίας (TBox) σε πιθανές μεταβολές του σχήματος της ΒΔ έχει αγνοηθεί πλήρως. Όσον αφορά σε μεθόδους που ανακαλύπτουν αντιστοιχίες μεταξύ μιας σχεσιακής ΒΔ και μιας οντολογίας, μόλις μια εργασία [11] εξετάζει το ζήτημα της προσαρμογής της αντιστοιχίας ή/και της οντολογίας, όταν μεταβάλλεται το σχήμα της ΒΔ. Η αντίθετη κατεύθυνση μετατροπής του σχεσιακού σχήματος όταν μεταβάλλεται η οντολογία δεν έχει προς το παρόν εξεταστεί. Το εν λόγω πρόβλημα είναι στενά συνδεδεμένο με το *πρόβλημα της προσαρμογής αντιστοιχιών* μεταξύ σχημάτων [194, 204] και οι προταθέντες αλγόριθμοι μπορούν να αποτελέσουν τη βάση για τη θεώρηση και λογικών φορμαλισμών, όπως είναι οι γλώσσες Περιγραφικής Λογικής. Από πρακτική σκοπιά, μηχανισμοί σκανδάλης (triggers) σε γεγονότα μεταβολής του σχήματος της ΒΔ - οι οποίοι όμως δεν υλοποιούνται από όλους τους κατασκευαστές σχεσιακών ΣΔΒΔ - καθώς και το Πρωτόκολλο Διατήρησης Συνδέσμων (Link Maintenance Protocol) του πλαισίου Silk [196] θα μπορούσαν να αποτελέσουν συστατικά μιας πιθανής λύσης, ανιχνεύοντας μεταβολές στο σχήμα μιας ΒΔ και σε έναν RDF γράφο αντίστοιχα.

3. **Παραγωγή Συνδεδεμένων Δεδομένων.** Ένας ικανοποιητικός αριθμός μεθόδων υποστηρίζει την επαναχρησιμοποίηση όρων από εξωτερικές οντολογίες, παράγοντα κρίσιμο για την πρόοδο του Σημασιολογικού Ιστού, ενώ κάποιες άλλες μέθοδοι ανακαλύπτουν αυτόματα τις κλάσεις και ιδιότητες δημοφιλών λεξιλογίων που περιγράφουν ακριβέστερα τα περιεχόμενα μιας σχεσιακής ΒΔ. Δυστυχώς όμως, αυτές οι προσπάθειες δεν αρκούν για την παραγωγή RDF γράφων που μπορούν άμεσα να ενσωματωθούν στο Σύννεφο Συνδεδεμένων Δεδομένων (Linking Open Data Cloud)[6].

[6]Το πολύ γνωστό διάγραμμα που δείχνει τους συνδέσμους μεταξύ ανοικτών RDF συνόλων δεδομένων υπάρχει στο http://richard.cyganiak.de/2007/10/lod/.

Για την παραγωγή πραγματικά Συνδεδεμένων Δεδομένων, οι οντότητες του πραγματικού κόσμου που περιγράφονται σε σχέσεις μιας ΒΔ πρέπει να αναγνωριστούν και ταυτόχρονα, πρέπει να οριστούν σύνδεσμοι με το κατάλληλο IRI αναγνωριστικό. Αυτό έρχεται σε αντίθεση με την τρέχουσα συνηθισμένη πρακτική, που απλά ερμηνεύει τις περισσότερες τιμές μιας ΒΔ ως λεκτικά, γεγονός που δυσχεραίνει τη συγχώνευση δύο RDF γράφων. Μερικές ενδιαφέρουσες προσπάθειες που ασχολούνται με το μετασχηματισμό λογιστικών φύλλων (spreadsheets) σε RDF γράφους έχουν προταθεί, με τα πιο αξιοσημείωτα παραδείγματα να είναι η RDF επέκταση του Google Refine [135] και το T2LD [144]. Τεχνικές σαν και αυτές που χρησιμοποιούνται στα εν λόγω εργαλεία σίγουρα μπορούν να προσαρμοστούν κατάλληλα και να εφαρμοστούν στο παρεμφερές σχεσιακό μοντέλο.

Τα παραπάνω προβλήματα σε συνδυασμό με τις προκλήσεις που αναφέρθηκαν στην ενότητα 3.6 οριοθετούν τις κατευθύνσεις προς τις οποίες θα κινηθεί στο άμεσο μέλλον η έρευνα σε αυτό το συναρπαστικό και άκρως ενδιαφέρον πεδίο. Δύο ερευνητικές προσπάθειες που εντάσσονται στο συγκεκριμένο πεδιο παρουσιάζονται στα κεφάλαια 4 και 5 της παρούσας διατριβής.

Κεφάλαιο 4

VisAVis: Μια απλή προσέγγιση αντιστοιχίας σχεσιακής ΒΔ με υπάρχουσα οντολογία

Περιεχόμενα

Στο παρόν κεφάλαιο, περιγράφουμε μια απλή προσέγγιση ορισμού αντιστοιχίας μιας σχεσιακής ΒΔ σε μια υπάρχουσα OWL οντολογία πεδίου. Η αντιστοιχία μεταξύ των δύο μοντέλων μπορεί να χρησιμοποιηθεί προκειμένου να προσδοθεί στη ΒΔ η δυνατότητα επερώτησής της σε όρους της οντολογίας πεδίου ή να αποτελέσει μέρος μιας ευρύτερης αρχιτεκτονικής ολοκλήρωσης ΒΔ βασισμένης σε οντολογίες. Ακόμα, η σύνδεση που επιτυγχάνεται μεταξύ της οντολογίας πεδίου και της ΒΔ επιτρέπει την ερμηνεία του περιεχομένου της τελευταίας ως ενός συνόλου οντολογικών ατόμων.

Εν συντομία, η προτεινόμενη μέθοδος προβλέπει τη χειροκίνητη συσχέτιση OWL κλάσεων με όψεις της ΒΔ, οι οποίες περιγράφονται από αντίστοιχα SQL ερωτήματα. Οι συσχετίσεις αυτές αποθηκεύονται εντός της οντολογίας λειτουργώντας ως επισημειώσεις αυτής. Η υλοποίηση της μεθόδου στο σύστημα VisAVis επιτρέπει τον ορισμό τέτοιων αντιστοιχιών, τις οποίες αυτό αξιοποιεί παρέχοντας τη δυνατότητα επερώτησης της ΒΔ μέσω σημασιολογικών ερωτημάτων. Το σύστημα VisAVis περιγράφεται τόσο στη δημοσίευση [114] όσο και στη διπλωματική εργασία [1], όπου δίνεται έμφαση στη λειτουργικότητά του. Αντίθετα, στο τρέχον κεφάλαιο, εστιάζουμε σε μια πιο τυπική θεώρηση της κεντρικής ιδέας του συστήματος (ενότητα 4.1) - το οποίο περιγράφουμε μόνο συνοπτικά - και εξετάζουμε σενάρια αξιοποίησής της (ενότητα 4.3).

4.1 Η κεντρική ιδέα

Στο κεφάλαιο 3 δόθηκε το γενικό περίγραμμα που ακολουθείται από την πλειοψηφία των προσεγγίσεων και εργαλείων που ορίζουν αντιστοιχίες μεταξύ μιας σχεσιακής ΒΔ και μιας OWL οντολογίας. Η πλέον συνηθισμένη τακτική επιβάλλει την έκφραση αυτών των αντιστοιχιών σε μια ιδιότυπη, συχνά σύνθετη, γλώσσα αντιστοιχιών, η οποία παρέχει τους απαραίτητους μηχανισμούς για τον ορισμό συσχετίσεων μεταξύ στοιχείων της οντολογίας και εκφράσεων σχεσιακής άλγεβρας. Παραδείγματα τέτοιων μηχανισμών αποτελούν ο ορισμός συνθηκών κάτω από τις οποίες ισχύει μια αντιστοιχία, ο ορισμός συναρτήσεων μετασχηματισμού στις τιμές της ΒΔ, ο καθορισμός πιθανών συνενώσεων μεταξύ σχέσεων της ΒΔ και ο ορισμός των σχεσιακών γνωρισμάτων που προσδιορίζουν μοναδικά τα στιγμιότυπα μιας OWL κλάσης.

Το προφανές μειονέκτημα στην προηγούμενη τακτική αποτελεί η ανάγκη εκμάθησης μιας νέας γλώσσας για τον ορισμό συσχετίσεων μεταξύ των δύο μοντέλων. Παράλληλα, η διαδικασία ορισμού αντιστοιχιών γίνεται ακόμα πιο δύσκολη όταν ο χρήστης που καλείται να φέρει σε πέρας αυτό το καθήκον (παραδείγματος χάριν, ο διαχειριστής της ΒΔ) δεν είναι εξοικειωμένος με οντολογικά μοντέλα. Είναι σημαντικό, λοιπόν, η αντιστοιχία να εκφράζεται σε μια μορφή που δεν επιβάλλει την εκμάθηση μιας νέας γλώσσας, αλλά να βασίζεται σε διαδεδομένες τεχνολογίες που είναι ευρύτερα γνωστές, ιδιαίτερα σε χρήστες με γνώσεις πληροφορικής. Μια τέτοια τεχνολογία είναι η SQL, η οποία στην παρούσα μέθοδο χρησιμοποιείται για τον προσδιορισμό του συνόλου δεδομένων της ΒΔ που θα συσχετιστεί με μια OWL κλάση.

Η απλότητα αυτής της προσέγγισης αποτελεί και το κύριο προσόν της, ένα προσόν ιδιαίτερα επιθυμητό σε εφαρμογές Σημασιολογικού Ιστού, όπου συνήθως απαιτείται εξοικείωση με τις αντίστοιχες τεχνολογίες, με αποτέλεσμα η καμπύλη εκμάθησής τους να είναι αρκετά απότομη. Παράλληλα, εκμεταλλεύεται την εκφραστικότητα της SQL για την περιγραφή ενός συνόλου δεδομένων, προσφέροντας μάλιστα και δυνατότητες που δε διαθέτουν κάποιες γλώσσες αντιστοιχίας (π.χ. R_2O [29]). Αναπόφευκτα βέβαια, υστερεί σε άλλα σημεία και δυνατότητες έναντι μεθόδων που χρησιμοποιούν μια εξειδικευμένη γλώσσα αντιστοιχίας, όπως π.χ. τη δυνατότητα αντιστοιχίας OWL ιδιοτήτων, την ανάθεση IRIs σε οντότητες της ΒΔ και την πραγματοποίηση μετασχηματισμών που δεν υποστηρίζονται από συναρτήσεις του SQL προτύπου.

4.1.1 Ορισμός και σημασιολογία αντιστοιχίας

Στη συγκεκριμένη ενότητα, εκφράζουμε τυπικά την έννοια της αντιστοιχίας ως μια συνάρτηση που συσχετίζει μια OWL κλάση με ένα SQL ερώτημα ή ισοδύναμα, με μια όψη (view) της σχεσιακής ΒΔ. Πρώτα παραθέτουμε έναν απλοποιημένο τυπικό ορισμό μιας σχεσιακής ΒΔ:

Ορισμός 4.1.1. (Σχεσιακό σχήμα) Το *σχήμα μιας σχεσιακής ΒΔ* ορίζεται από τα στοιχεία $\langle R_S, att, D_S, sort, f_{col}, f_{ref}, f_{dom} \rangle$, όπου:

- R_S είναι ένα σύνολο σχέσεων
- att είναι ένα σύνολο γνωρισμάτων

- D_S είναι ένα σύνολο τύπων δεδομένων
- $sort : R_S \rightarrow att^n$ είναι μια συνάρτηση η οποία αντιστοιχεί σε κάθε σχέση ένα σύνολο γνωρισμάτων
- $f_{col} : att \rightarrow R_S$ είναι μια συνάρτηση που δηλώνει τη σχέση στην οποία ανήκει ένα γνώρισμα
- $f_{ref} : att^n \rightarrow att^n$ είναι μια συνάρτηση που περιγράφει σχέσεις ξένου κλειδιού μεταξύ συνόλων γνωρισμάτων
- $f_{dom} : att \rightarrow D_S$ είναι μια συνάρτηση που συσχετίζει ένα γνώρισμα με έναν συγκεκριμένο τύπο δεδομένων

Ο ορισμός 4.1.1 σκόπιμα αγνοεί στοιχεία του σχεσιακού μοντέλου, όπως πρωτεύοντα κλειδιά, NOT NULL περιορισμούς και περιορισμούς μοναδικότητας, τα οποία δεν είναι απαραίτητα για την παρουσίαση της προσέγγισής μας.

Ορισμός 4.1.2. (Σχεσιακό στιγμιότυπο) Μια πλειάδα μιας σχέσης R αποτελείται από ζεύγη $\langle attr, val\rangle$, $\forall attr \in sort(R)$, όπου $val \in f_{dom}(attr) \cup null$. Ένα ζεύγος $\langle attr, val\rangle$ ονομάζεται και κελί. Το *στιγμιότυπο μιας σχέσης* R_I ορίζεται ως ένα πολυσύνολο (multiset) πλειάδων της σχέσης.

Το *στιγμιότυπο μιας σχεσιακής ΒΔ* DB_I ορίζεται ως η ένωση των στιγμιοτύπων των σχέσεων που αυτή περιέχει, δηλαδή $DB_I = \bigcup_{R \in R_S} R_I$.

Ο ορισμός 4.1.2 για το στιγμιότυπο μιας σχέσης είναι αρκετά γενικός και είναι σε θέση να περιγράψει και το σύνολο αποτελεσμάτων της εκτέλεσης ενός SQL ερωτήματος, εφόσον το αποτέλεσμα της εφαρμογής μιας ακολουθίας σχεσιακών τελεστών σε μια σχέση αποτελεί επίσης σχέση. Σύμφωνα με την παραπάνω παρατήρηση, ένα SQL ερώτημα q είναι ισοδύναμο με μια συνάρτηση q, η οποία αντιστοιχεί ένα πολυσύνολο πλειάδων (το αρχικό στιγμιότυπο της σχεσιακής ΒΔ) σε ένα άλλο (αποτέλεσμα ερωτήματος). Δηλαδή, για ένα συγκεκριμένο ερώτημα q, το αποτέλεσμα της εκτέλεσής του στο DB_I συμβολίζεται με $q(DB_I)$.

Έχοντας ορίσει μια χαλαρή εκδοχή μιας σχεσιακής ΒΔ, προχωρούμε στον ορισμό μιας OWL οντολογίας, προσαρμόζοντας ελαφρώς τον ορισμό 2.2.1:

Ορισμός 4.1.3. (OWL οντολογία) Μια *OWL οντολογία* ορίζεται από μια πλειάδα της μορφής $\langle V_O, T_O, A_O, Annot_O\rangle$, όπου:

- V_O είναι το λεξιλόγιο της OWL οντολογίας, το οποίο περιλαμβάνει ένα σύνολο κλάσεων C_O, ένα σύνολο ιδιοτήτων αντικειμένου OP_O, ένα σύνολο ιδιοτήτων τύπου δεδομένων DP_O, ένα σύνολο ιδιοτήτων επισημείωσης AP_O και ένα σύνολο ατόμων I_O
- T_O είναι το σύνολο των ορολογικών αξιωμάτων, αντίστοιχο του TBox σε μια βάση γνώσης
- A_O είναι το σύνολο των ισχυρισμών της οντολογίας, δηλαδή των προτάσεων που αναφέρονται σε άτομα, αντίστοιχο του ABox σε μια βάση γνώσης
- $Annot_O$ είναι το σύνολο των αξιωμάτων επισημείωσης

Με βάση τα παραπάνω, ορίζουμε την αντιστοιχία μιας σχεσιακής ΒΔ με μια OWL οντολογία ως εξής:

Ορισμός 4.1.4. (Αντιστοιχία ΒΔ με οντολογία) Μια *αντιστοιχία* μεταξύ ενός στιγμιοτύπου σχεσιακής ΒΔ DB_I με μια OWL οντολογία O ορίζεται ως μια συνάρτηση map η οποία συσχετίζει μια OWL κλάση $C \in C_O$ με ένα SQL ερώτημα $q = map(C)$. Το αποτέλεσμα της εκτέλεσης αυτού του SQL ερωτήματος στο στιγμιότυπο μιας ΒΔ DB_I θα είναι ένα σύνολο πλειάδων που συμβολίζεται με $map(C)(DB_I)$.

Σύμφωνα με την ορολογία που χρησιμοποιείται σε συστήματα ολοκλήρωσης δεδομένων, η αντιστοιχία αυτή μπορεί να ιδωθεί και ως μια Global as View (GAV) αντιστοιχία της μορφής: $C \rightsquigarrow q_S$, όπου C μια κλάση μιας οντολογίας O που λειτουργεί ως καθολικό σχήμα και q_S ένα ερώτημα ή όψη που αναφέρεται σε μία ή περισσότερες ΒΔ (ή πηγές δεδομένων στη γενικότερη περίπτωση) [125].

Ένα κρίσιμο ζήτημα που προκύπτει ως συνέπεια του προβλήματος της ασυμφωνίας μεταξύ του σχεσιακού και αντικειμενοστρεφών μοντέλων (object-relational impedance mismatch) - παράδειγμα των τελευταίων μπορεί να θεωρηθεί και το OWL μοντέλο - είναι το γεγονός ότι το πρώτο αποθηκεύει σταθερές τιμές δεδομένων, ενώ τα δεύτερα αναφέρονται σε αντικείμενα που χαρακτηρίζονται από ένα μοναδικό αναγνωριστικό. Αυτό το κενό μεταξύ των δύο μοντέλων αντιμετωπίζεται από μεθόδους αντιστοιχίας σχεσιακής ΒΔ σε οντολογία, μέσω του ορισμού μηχανισμών παραγωγής IRI αναγνωριστικών, οι οποίοι χρησιμοποιούν τιμές γνωρισμάτων της ΒΔ εντός κατάλληλων προτύπων συμβολοσειρών. Δυστυχώς, η χρήση της SQL ως μέσου έκφρασης μιας αντιστοιχίας, παρά την απλότητά της, αποκλείει δυνατότητες όπως την προηγούμενη ή την αντιστοιχία OWL ιδιοτήτων με στοιχεία της ΒΔ.

Η υπόθεση που υιοθετείται από την παρούσα προσέγγιση είναι αυτή που υπαγορεύει και η βασική προσέγγιση αντιστοιχίας σχεσιακής ΒΔ με οντολογία (ενότητα 3.2), ήτοι κάθε πλειάδα του αποτελέσματος της εκτέλεσης του SQL ερωτήματος $map(C_i)$ αντιστοιχεί σε ένα άτομο της κλάσης C_i. Επίσης, καθώς δεν είναι δυνατός ο ορισμός ενός μηχανισμού παραγωγής IRI για κάθε πλειάδα ούτε τίθεται κάποιος περιορισμός στην μορφή του SQL ερωτήματος q που να επιβάλλει την παρουσία κάποιου πρωτεύοντος κλειδιού μεταξύ των γνωρισμάτων του $q(DB_I)$, γίνεται η παραδοχή ότι η αναγνώριση ενός ατόμου και η διάκρισή του από άλλα άτομα βασίζεται στο σύνολο των τιμών της πλειάδας με την οποία αυτό συσχετίζεται. Υπονοείται δηλαδή, η ύπαρξη μιας συνάρτησης τ, ανάλογης με αυτής στο [158], η οποία αντιστοιχεί πλειάδες τιμών σε αναγνωριστικά οντολογικών ατόμων. Πιο συγκεκριμένα, μια πλειάδα t αποτελούμενη από τα *μη κενά* κελιά $cell_1, cell_2, \ldots, cell_n$ αντιστοιχεί σε ένα OWL άτομο με αναγνωριστικό $\tau(t) = \tau(cell_1, cell_2, \ldots, cell_n)$. Η παραμέληση των κενών κελιών μιας πλειάδας στη δημιουργία ενός αναγνωριστικού, πέρα από το ότι είναι συνηθισμένη στην πράξη και υιοθετείται και από τη γλώσσα αντιστοιχίας R2RML (παράγραφος 2.2.4), εξασφαλίζει την ανάθεση του ίδιου αναγνωριστικού σε πλειάδες με το ίδιο περιεχόμενο, ακόμα και αν αυτές αποτελούν μέρος του αποτελέσματος διαφορετικών SQL ερωτημάτων.

Παράδειγμα 4.1.1. Έστω οι σχέσεις EMP_1 (EmpID, EmpName) και EMP_2 (EmpID, EmpName, Email). Παρά τη διαφορά στο σχήμα των δύο σχέσεων, η

συνάρτηση τ θα αναθέσει το ίδιο αναγνωριστικό στις παρακάτω πλειάδες των σχέσεων EMP_1 και EMP_2 εφόσον τα μη κενά κελιά τους συμφωνούν ως προς τα γνωρίσματα και τις τιμές τους.

EMP_1	
EmpID	EmpName
7369	John Doe

EMP_2		
EmpID	EmpName	Email
7369	John Doe	-

Υπενθυμίζουμε ότι, σύμφωνα με το χαλαρό ορισμό 4.1.1 για το στιγμιότυπο μιας σχέσης, οι δύο πλειάδες των σχέσεων EMP_1, EMP_2 μπορεί να είναι τα αποτελέσματα της εκτέλεσης δύο διαφορετικών SQL ερωτημάτων q_1, q_2.

Αυτή η έννοια της «ισοδυναμίας» πλειάδων έρχεται σε αντίθεση με τον παραδοσιακό ορισμό του σχεσιακού μοντέλου και πλησιάζει περισσότερο προς την έννοια της *RDF πλειάδας* στο [67] ή ισοδύναμα, την έννοια της *SPARQL λύσης* (SPARQL solution), οι οποίες ορίζονται ως μερικές συναρτήσεις (partial functions) που συσχετίζουν ένα σύνολο μεταβλητών με αντίστοιχες σταθερές τιμές. Και στις δύο περιπτώσεις, η συσχέτιση της κενής (null) τιμής με μια μεταβλητή ισοδυναμεί με την απουσία της μεταβλητής αυτής από την πλειάδα, θεώρηση που συμβαδίζει με την σημασιολογία ανοικτού κόσμου των Περιγραφικών Λογικών.

Μια άλλη διαφορά μεταξύ του σχεσιακού μοντέλου και των Περιγραφικών Λογικών, την οποία σημειώσαμε και στην παράγραφο 2.2.2, αφορά στην Υπόθεση Μοναδικών Ονομάτων (UNA), η οποία ισχύει στο πρώτο αλλά συνήθως όχι στην περίπτωση των δεύτερων. Όπως και στο [158], υιοθετούμε την Υπόθεση Μοναδικών Ονομάτων για τα OWL άτομα που αντιστοιχούν σε πλειάδες της σχεσιακής ΒΔ. Αυτό σημαίνει ότι διαφορετικά αναγνωριστικά - που παράγονται από τη συνάρτηση τ - σηματοδοτούν διαφορετικά OWL άτομα ή, με άλλα λόγια, πλειάδες με διαφορετικό περιεχόμενο αντιστοιχούν σε διακριτά στιγμιότυπα μιας OWL κλάσης.

Σε αυτό το σημείο, χρειάζεται να τονιστεί ότι, σύμφωνα με τις υποθέσεις στις οποίες βασίζεται η προτεινόμενη προσέγγιση, η συνάρτηση τ *υπονοείται*, δηλαδή δεν υπάρχει κάποιος απτός μηχανισμός παραγωγής IRI από τα σχεσιακά δεδομένα. Συνεπώς, η προηγούμενη συζήτηση για την παραγωγή αναγνωριστικών και την υιοθέτηση της Υπόθεσης Μοναδικών Ονομάτων ανάγεται στο επίπεδο των σχεσιακών πλειάδων και στο πότε αυτές μπορούν να θεωρηθούν ισοδύναμες. Ο ορισμός 4.1.5 εξετάζει το θέμα αυτό.

Ορισμός 4.1.5. (Ισοδύναμες ως προς το περιεχόμενο πλειάδες) Δύο γνωρίσματα att_1, att_2 θεωρούνται ίσα, αν έχουν το ίδιο όνομα και ανήκουν στην ίδια σχέση:

$$att_1 = att_2 \Leftrightarrow name(att_1) = name(att_2) \wedge f_{col}(att_1) = f_{col}(att_2)^{1}$$

όπου $name(att)$ το όνομα του γνωρίσματος att.

Δυο κελιά $\langle att_1, val_1 \rangle, \langle att_2, val_2 \rangle$ θεωρούνται *ίσα* όταν έχουν την ίδια τιμή και είτε ανήκουν στο ίδιο γνώρισμα είτε ανήκουν σε γνωρίσματα που συνδέονται

[1]Προϋπόθεση για την επιτυχή αναγνώριση δύο ίσων γνωρισμάτων είναι η απουσία μετονομασμένων γνωρισμάτων στα SQL ερωτήματα που χρησιμοποιούνται για την αντιστοιχία.

με σχέση ξένου κλειδιού. Ισοδύναμα:

$$\langle att_1, val_1\rangle = \langle att_2, val_2\rangle \Leftrightarrow val_1 = val_2 \wedge (att_1 = att_2 \vee f_{ref}(att_1) = att_2 \vee f_{ref}(att_2) = att_1)$$

Δυο πλειάδες t_1, t_2 με μη κενά κελιά $\langle cell_{11}, cell_{12}, \ldots, cell_{1n}\rangle$ και $\langle cell_{21}, cell_{22}, \ldots, cell_{2n}\rangle$ αντίστοιχα ονομάζονται *ισοδύναμες ως προς το περιεχόμενό τους* όταν τα μη κενά κελιά τους είναι ίσα, δηλαδή:

$$t_1 \equiv t_2 \Leftrightarrow \forall cell_{1i}, cell_{1i}.val \neq null : \exists cell_{2j} : cell_{1i} = cell_{2j}$$

Η έννοια των ίσων κελιών, σύμφωνα με τον ορισμό 4.1.5, εξασφαλίζει ότι αυτά αναφέρονται σε μια τιμή με κοινή σημασία σε αντίθεση π.χ. με τα κελιά $\langle$*έτος_γέννησης*, 1982$\rangle$ και $\langle$*κωδ_εργαζομένου*, 1982$\rangle$. Αντίστοιχα, δύο ισοδύναμες πλειάδες αποτελούνται από το ίδιο σύνολο μη κενών κελιών, επομένως μπορεί να υποτεθεί με σχετική βεβαιότητα ότι αναφέρονται στην ίδια οντότητα και το περιεχόμενό τους εκφράζει ιδιότητες αυτής της οντότητας. Δυστυχώς, δεν ισχύει το ίδιο και για την περίπτωση μη ισοδύναμων πλειάδων, όπως φαίνεται και από το παράδειγμα 4.1.2.

Παράδειγμα 4.1.2. Έστω οι σχέσεις EMP_1 και EMP_2 του παραδείγματος 4.1.1, με τη μόνη διαφορά ότι το γνώρισμα EmpName της σχέσης έχει μετονομαστεί σε EName. Πλέον, οι πλειάδες του παραδείγματος 4.1.1, δεν μπορούν να αναγνωριστούν ως ισοδύναμες, παρά το γεγονός ότι πιθανότατα αναφέρονται στην ίδια οντότητα. Επομένως, στη γενική περίπτωση, για δύο μη ισοδύναμες πλειάδες, δεν μπορεί να γίνει κάποια υπόθεση σχετικά με την ισοδυναμία των οντολογικών ατόμων στα οποία αυτές αντιστοιχούν.

Η έλλειψη μιας σαφώς ορισμένης συνάρτησης τ, η οποία εξάλλου αποτελεί βασικό κομμάτι των περισσότερων γλωσσών αντιστοιχίας ΒΔ με οντολογία, δεν επιτρέπει την εξαγωγή βέβαιου συμπεράσματος για μη ισοδύναμες πλειάδες και αποτελεί συνέπεια της επιλογής της SQL ως μέσου έκφρασης των αντιστοιχιών. Επειδή όμως η γνώση των γενικών χαρακτηριστικών της συνάρτησης τ (π.χ. το αν αποτελεί συνάρτηση 1:1 ή όχι) είναι απαραίτητα για τον τυπικό ορισμό της σημασιολογίας μιας αντιστοιχίας αλλά και για την πραγματοποίηση συλλογισμού (παράγραφος 4.1.2), υποθέτουμε, για λόγους ευκολίας, ότι μη ισοδύναμες πλειάδες οδηγούν στην παραγωγή διαφορετικών IRI αναγνωριστικών, και κατ' επέκταση, λόγω και της υιοθέτησης της Υπόθεσης Μοναδικών Ονομάτων, σε διαφορετικά οντολογικά άτομα.

Λαμβάνοντας υπόψη όλα τα προηγούμενα, δοκιμάζουμε να ορίσουμε τη σημασιολογία των OWL κλάσεων που έχουν αντιστοιχηθεί σε κάποιο SQL ερώτημα, μέσω της *συνάρτησης ερμηνείας* (interpretation function) αυτών, μια τακτική που είναι συνηθισμένη στις Περιγραφικές Λογικές. Στην περίπτωση μιας κλάσης, η συνάρτηση ερμηνείας είναι μια συνάρτηση η οποία αντιστοιχεί την κλάση σε ένα σύνολο αντικειμένων που προέρχονται από ένα χώρο ερμηνείας (interpretation domain) Δ^I. Σύμφωνα με τις υποθέσεις της προσέγγισής μας, η αντιστοιχία μιας OWL κλάσης C με ένα SQL ερώτημα $map(C)$ δημιουργεί τόσα νέα στιγμιότυπα της C όσα και οι πλειάδες του αποτελέσματος $map(C)(DB_I)$. Αυτά τα νέα άτομα έρχονται να προστεθούν σε ήδη υπάρχοντα άτομα της C

που πιθανώς υπάρχουν στο ABox A_O της οντολογίας O. Συνεπώς, μπορούμε να ορίσουμε την ερμηνεία της κλάσης C ως:

$$C^I = \{\alpha \in \Delta^I : C(\alpha) \in A_O\} \cup \{\tau(t) : t \in map(C_i)(DB_I)\} \tag{4.1}$$

Η εξίσωση (4.1) οδηγεί σε μια αντίστοιχη επέκταση A_V του A_O με ισχυρισμούς $C_i(\tau(t))$ για τις κλάσεις C_i που βρίσκονται στο πεδίο ορισμού της συνάρτησης *map*, δηλαδή αυτές τις κλάσεις που έχουν αντιστοιχηθεί σε κάποιο SQL ερώτημα. Επίσης, η υιοθέτηση της Υπόθεσης Μοναδικών Ονομάτων επιβάλλει την προσθήκη ισχυρισμών της μορφής $\tau(t_i) \neq \tau(t_j)$ για όλες τις πλειάδες των αποτελεσμάτων των SQL ερωτημάτων που έχουν αντιστοιχηθεί στις κλάσεις C_i. Αξίζει να τονιστεί ότι, καθώς η συνάρτηση τ είναι υπονοούμενη, το ίδιο ισχύει και για τους ισχυρισμούς του A_V, οι οποίοι είναι *εικονικοί* (virtual assertions), με τα αντίστοιχα δεδομένα να παραμένουν στη ΒΔ και να μην υλοποιούνται σε φυσική μορφή. Το A_V ονομάζεται **εικονικό σώμα ισχυρισμών** και δίνεται από την εξίσωση (4.2):

$$\begin{aligned} A_V = \{C_i(\tau(t)) : C_i \in dom(map), t \in map(C_i)(DB_I)\} \cup \{\tau(t_i) \neq \tau(t_j) : \\ C_i \in dom(map), t_i, t_j \in map(C_i)(DB_I), t_i \not\equiv t_j\} \end{aligned} \tag{4.2}$$

Σε αντίθεση με την εικονική επέκταση του σώματος ισχυρισμών της οντολογίας, οι αντιστοιχίες του ορισμού 4.1.4 αποθηκεύονται σε φυσική μορφή στην OWL οντολογία O, με τη μορφή αξιωμάτων επισημείωσης, εμπλουτίζοντάς την.

Ορισμός 4.1.6. (Επισημειωμένη οντολογία) Έστω $O = \langle V_O, T_O, A_O, Annot_O \rangle$ μια OWL οντολογία, κλάσεις της οποίας αντιστοιχούνται μέσω της συνάρτησης *map* σε SQL ερωτήματα επί ενός στιγμιοτύπου ΒΔ DB_I. Η συνάρτηση *map* οδηγεί στη δημιουργία μιας εμπλουτισμένης έκδοσης της O, την $O_{annot} = \langle V'_O, T_O, A'_O, Annot'_O \rangle$, όπου $V'_O = V_O \cup \{p_{map}\}$, $Annot'_O = Annot_O \cup \{p_{map}(C_i, q_i) : C_i \in C_O, q_i = map(C_i)\}$ και $A'_O = A_O \cup A_V$, με το A_V να αποτελεί το εικονικό σώμα ισχυρισμών της εξίσωσης (4.2). Η O_{annot} ονομάζεται *επισημειωμένη οντολογία*.

Όπως φαίνεται από τον ορισμό 4.1.6, η εφαρμογή μιας αντιστοιχίας *map* σε κλάσεις μιας οντολογίας O εμπλουτίζει το λεξιλόγιο της O με μια νέα ιδιότητα επισημείωσης p_{map} και με ισχυρισμούς ιδιοτήτων $p_{map}(C_i, q_i)$ για κάθε επισημειωμένη κλάση C_i. Η επιλογή αυτή αφήνει άθικτη τη ΒΔ και, παράλληλα, δεν επηρεάζει εξωτερικές εφαρμογές που πιθανώς στηρίζονται σε αυτήν. Ταυτόχρονα, η αποθήκευση των αντιστοιχιών με τη μορφή αξιωμάτων επισημείωσης δεν αλλοιώνει το ορολογικό περιεχόμενο της οντολογίας, ενώ οι επισημειώσεις αυτές μπορούν να είναι αξιοποιήσιμες στο πλαίσιο διαφόρων σεναρίων χρήσης από εξωτερικές εφαρμογές (ενότητα 4.3).

4.1.2 Συλλογισμός με αντιστοιχίες

Η προσθήκη των αξιωμάτων επισημείωσης του ορισμού 4.1.6 επηρεάζει έμμεσα την *ικανοποιησιμότητα* της επισημειωμένης οντολογίας O_{annot}, μέσω της εικονικής επέκτασης A_V. Καθώς το σώμα ισχυρισμών (ABox) της επισημειωμένης οντολογίας είναι εν μέρει εικονικό, δεν μπορούμε να χρησιμοποιήσουμε υπάρχοντα εργαλεία συλλογισμού για τον έλεγχο της ικανοποιησιμότητάς της.

Είναι απαραίτητη λοιπόν η εισαγωγή μιας σειράς ελέγχων οι οποίοι εξασφαλίζουν ότι ο ορισμός νέων αντιστοιχιών δεν έρχεται σε αντίφαση με τα ορολογικά αξιώματα της οντολογίας και τις ήδη υπάρχουσες αντιστοιχίες. Με βάση τη μορφή του A_V στην εξίσωση (4.2), όπου οι πρόσθετοι εικονικοί ισχυρισμοί δηλώνουν μόνο τη συμμετοχή ατόμων σε κλάσεις, τα μοναδικά OWL αξιώματα που μπορούν να οδηγήσουν σε μια ασυνεπή οντολογία είναι τα **αξιώματα ξένων κλάσεων** (ιδιότητα `owl:disjointWith`).

Ένα αξίωμα ξένων κλάσεων $C \sqcap D \sqsubseteq \bot$[2] θα ισχύει αν $C^I \cap D^I = \emptyset$. Υποθέτοντας ότι το αρχικό ABox της οντολογίας A_O δεν περιέχει κάποια ασυνέπεια και δεδομένου του ότι η συνάρτηση παραγωγής IRI είναι υπονοούμενη και όχι απτή, μπορούμε να υποθέσουμε ότι το εικονικό τμήμα του επεκταμένου A'_O δεν έχει κοινά άτομα με το αρχικό A_O, δηλαδή $A_O \cap A_V = \emptyset$. Η συγκεκριμένη υπόθεση ισχύει συνήθως και στην πράξη, καθώς το σχήμα παραγωγής IRI ορίζεται έτσι ώστε να χρησιμοποιεί ξεχωριστό χώρο ονομάτων για άτομα που προέρχονται από τη ΒΔ, εξασφαλίζοντας την απουσία επικάλυψης με υπάρχοντες όρους. Με βάση την παραπάνω υπόθεση, ο έλεγχος του αξιώματος ξένων κλάσεων θα περιοριστεί μονάχα σε *εικονικά* άτομα του A_V. Επομένως, σύμφωνα και με την εξίσωση (4.1), το αρχικό αξίωμα ξένων κλάσεων θα ισχύει, αν τα αποτελέσματα των SQL ερωτημάτων $map(C_i)$ και $map(D_i)$ δεν περιέχουν πλειάδες ισοδύναμες ως προς το περιεχόμενό τους (οι οποίες θα αντιστοιχούσαν στο ίδιο IRI αναγνωριστικό μέσω της συνάρτησης τ, οδηγώντας σε ασυνέπεια).

Η διαδικασία ελέγχου της συνέπειας μιας επισημειωμένης κλάσης C, δεδομένου ενός συνόλου αξιωμάτων ξένων κλάσεων στα οποία αυτή συμμετέχει και ενός συνόλου πλειάδων στα οποία αυτή έχει αντιστοιχηθεί, απεικονίζεται στον αλγόριθμο 1. Ο αλγόριθμος 1 εξετάζει μόνο κλάσεις που έχουν αντιστοιχηθεί, καθώς, όπως υποθέσαμε προηγουμένως, τα A_V και A_O είναι ξένα μεταξύ τους και δε χρειάζεται να γίνει σύγκριση μεταξύ των ατόμων που ανήκουν σε αυτά. Η εύρεση όλων των συνεπαγόμενων από την O ξένων κλάσεων της κλάσης C επιβάλλει αρχικά την πραγματοποίηση συλλογισμού στο T_O, ώστε να βρεθούν και να ελεγχθούν όλα τα πιθανά ζεύγη. Παραδείγματος χάριν, το σύνολο αξιωμάτων:

$$\begin{aligned} Man \sqcap Woman &\sqsubseteq \bot \\ Father &\sqsubseteq Man \\ Mother &\sqsubseteq Woman \end{aligned}$$

οδηγεί στα επόμενα τρία ζεύγη ξένων κλάσεων, τα οποία πρέπει να ληφθούν υπόψη από τον αλγόριθμο 1:

$$\begin{aligned} Father \sqcap Woman &\sqsubseteq \bot \\ Mother \sqcap Man &\sqsubseteq \bot \\ Father \sqcap Mother &\sqsubseteq \bot \end{aligned}$$

Σημαντικό τμήμα του αλγορίθμου 1 αποτελεί η σύγκριση των αποτελεσμάτων της εκτέλεσης του $map(C)$ με τα αποτελέσματα καθενός από τα ερωτήματα

[2]Το σύμβολο $\bot$ δηλώνει την κενή έννοια στις Περιγραφικές Λογικές, δηλαδή μια μη ικανοποιήσιμη έννοια.

Αλγόριθμος 1 *Έλεγχος ξένων κλάσεων*

Είσοδος: OWL κλάση C, σύνολο πλειάδων *tuples*, επισημειωμένη οντολογία O_{annot}
Έξοδος: True / False, ανάλογα με το αν το ζεύγος $(C, tuples)$ είναι σύμφωνο με την O_{annot} και τα σύνολα πλειάδων στα οποία έχουν αντιστοιχηθεί οι ξένες κλάσεις της C

```
1: function ΕΛΕΓΧΟΣ_ΞΕΝΩΝ_ΚΛΑΣΕΩΝ(C, tuples, O_annot)
2:     C_disjoint ← σύνολο συνεπαγόμενων ξένων κλάσεων της C
3:     for all D στο C_disjoint do
4:         if D έχει τιμή για ιδιότητα επισημείωσης p_map στην O_annot then
5:             D_query ← τιμή της ιδιότητας επισημείωσης p_map για D
6:             D_tuples ← σύνολο πλειάδων του αποτελέσματος της εκτέλεσης του D_query
7:             if D_tuples ∩ tuples ≠ ∅ then return false          # βλέπε και αλγόριθμο 2
8:             end if
9:         end if
10:    end for
11:    return true
12: end function
```

map(D) για κάθε ξένη κλάση D (γραμμή 7), προκειμένου να διαπιστωθεί αν τα δύο σύνολα διαθέτουν ισοδύναμες ως προς το περιεχόμενο πλειάδες. Η σύγκριση αυτή μπορεί να γίνει αποδοτικά με τη βοήθεια ενός SQL ερωτήματος, όπως περιγράφεται και από τον αλγόριθμο 2. Ο αλγόριθμος 2 πρώτα εξετάζει τα γνωρίσματα εξόδου δύο SQL ερωτημάτων που έχουν αντιστοιχηθεί σε δύο ξένες OWL κλάσεις. Αν τα δύο ερωτήματα δεν έχουν κοινά γνωρίσματα εξόδου, αυτό αυτομάτως σημαίνει ότι δεν θα έχουν και ισοδύναμες ως προς το περιεχόμενο πλειάδες. Διαφορετικά, λαμβάνεται η τομή των δύο ερωτημάτων, αφού πρώτα επαυξηθεί κατάλληλα το σύνολο των γνωρισμάτων εξόδου για κάθε ερώτημα, έτσι ώστε να είναι συμβατά μεταξύ τους, καθιστώντας επιτρεπτή την πράξη της τομής (γραμμές 4-6). Υπενθυμίζουμε ότι η προσθήκη επιπλέον κενών γνωρισμάτων εξόδου σε κάθε ερώτημα (δηλαδή με προσθήκη όρων της μορφής `NULL AS attr` στο `SELECT` τμήμα του ερωτήματος) δεν αλλοιώνει το περιεχόμενο μιας πλειάδας σύμφωνα με τον ορισμό 4.1.5. Αν το αποτέλεσμα της τομής των δύο ερωτημάτων είναι κενό, τότε οι δύο ξένες κλάσεις δεν έχουν ισοδύναμες ως προς το περιεχόμενο πλειάδες και το αντίστοιχο OWL αξίωμα ικανοποιείται.

Εκτός από την εξασφάλιση της συνέπειας της επισημειωμένης οντολογίας, χρειάζεται να εξεταστεί και η αλληλεπίδραση του A_V με άλλα OWL αξιώματα που οδηγούν στην εξαγωγή νέων συμπερασμάτων. Τα OWL αξιώματα που επηρεάζουν το συλλογισμό παρουσία των εικονικών ισχυρισμών του A_V είναι τα εξής:

1. αξιώματα υπαγωγής κλάσεων `rdfs:subClassOf`
2. αξιώματα ισοδύναμων κλάσεων `owl:equivalentClass`
3. σύνθετες εκφράσεις κλάσεων που περιλαμβάνουν τις πράξεις της τομής (`owl:intersectionOf`), ένωσης (`owl:unionOf`) ή συμπληρώματος κλάσεων (`owl:complementOf`)

Με τρόπο ανάλογο με αυτόν με τον οποίο αντιμετωπίστηκαν τα αξιώματα ξένων κλάσεων, εξετάζουμε και εδώ μόνο την παρουσία επισημειωμένων κλάσεων, εφόσον έχουμε υποθέσει ότι τα εικονικά άτομα που προέρχονται από τη

Αλγόριθμος 2 *Σύγκριση ξένων συνόλων πλειάδων*

Είσοδος: SQL ερώτημα q_1, SQL ερώτημα q_2
Έξοδος: True / False, ανάλογα με το αν τα αποτελέσματα της εκτέλεσης των q_1, q_2 περιέχουν ισοδύναμες ως προς το περιεχόμενο πλειάδες

```
1: function ΣΥΓΚΡΙΣΗ_ΞΕΝΩΝ_ΣΥΝΟΛΩΝ_ΠΛΕΙΑΔΩΝ(q1, q2)
2:     if sort(q1) ∩ sort(q2) = ∅ then return false;
3:     else
4:         q1' ← q1 με επιπλέον προβαλλόμενα γνωρίσματα sort(q2) − sort(q1)
5:         q2' ← q2 με επιπλέον προβαλλόμενα γνωρίσματα sort(q1) − sort(q2)
6:         q ← q1' INTERSECT q2'
7:         q_tuples ← σύνολο πλειάδων του αποτελέσματος της εκτέλεσης του q
8:         if q_tuples = ∅ then return false
9:         else
10:            return true
11:        end if
12:    end if
13: end function
```

ΒΔ δεν ταυτίζονται με κάποια από τα άτομα του αρχικού ABox A_O. Σύμφωνα λοιπόν με τη σημασιολογία της OWL, ένα **αξίωμα υπαγωγής κλάσεων** της μορφής $C \sqsubseteq D$ θα οδηγήσει στην επέκταση του D^I κατά τα άτομα $\tau(t)$ που βρίσκονται στο C^I αλλά όχι και στο D^I. Με άλλα λόγια, η διαδικασία του συλλογισμού θα οδηγήσει σε μια νέα ερμηνεία της κλάσης D:

$$D^{I'} = D^I \cup (C^I - D^I) = D^I \cup C^I$$

Στην πράξη, αυτό σημαίνει ότι θα πρέπει να αλλάξει η ήδη ορισμένη αντιστοιχία της κλάσης D, ώστε να συμπεριλάβει το σύνολο πλειάδων που έχει αντιστοιχηθεί στην κλάση C, κάτι που επιτυγχάνεται λαμβάνοντας την ένωση (union) των δύο SQL ερωτημάτων $map(C)$ και $map(D)$:

$$map(D) \leftarrow map(C) \cup map(D)$$

Η συγκεκριμένη ενημέρωση θα πρέπει να πραγματοποιηθεί για κάθε ζεύγος επισημειωμένων OWL κλάσεων που συνδέεται με μια συνεπαγόμενη σχέση κλάσης - υποκλάσης. Η εύρεση της συνεπαγόμενης ιεραρχίας κλάσεων μπορεί να πραγματοποιηθεί με εφαρμογή διαδεδομένων διαδικασιών συλλογισμού στο T_O.

Ανάλογα αποτελέσματα αναμένονται και από τις υπόλοιπες μορφές αξιωμάτων. Ένα **αξίωμα ισοδύναμων κλάσεων** $C \equiv D$ ισοδυναμεί με δύο αξιώματα $C \sqsubseteq D$ και $D \sqsubseteq C$, οπότε οι αντιστοιχίες των κλάσεων C και D θα πρέπει να προσαρμοστούν ώστε αυτές να περιέχουν τις ίδιες πλειάδες (οπότε και τα ίδια άτομα):

$$map(C) \leftarrow map(C) \cup map(D)$$
$$map(D) \leftarrow map(C) \cup map(D)$$

Όσον αφορά σε αξιώματα τομής και ένωσης κλάσεων, αυτά αντιμετωπίζονται παρόμοια, καθώς συνεπάγονται αντίστοιχα αξιώματα υπαγωγής, τα οποία θεωρούμε ότι έχουν ήδη εξαχθεί κατά τη διαδικασία συλλογισμού του T_O. Πιο συγκεκριμένα, η έκφραση μιας κλάσης ως **τομής** δύο άλλων κλάσεων $C \equiv D \sqcap E$

συνεπάγεται τα αξιώματα $C \sqsubseteq D$ και $C \sqsubseteq E$, με αποτέλεσμα να πρέπει να ενημερωθούν κατάλληλα οι αντιστοιχίες των κλάσεων D και E:

$$map(D) \leftarrow map(C) \cup map(D)$$
$$map(E) \leftarrow map(C) \cup map(E)$$

Το ίδιο ισχύει και για την περίπτωση της **ένωσης** δύο κλάσεων $C \equiv D \sqcup E$ η οποία συνεπάγεται τα αξιώματα $D \sqsubseteq C$ και $E \sqsubseteq C$, με αποτέλεσμα να χρειάζεται να μεταβληθεί η αντιστοιχία της C ώστε να συμπεριλάβει τις πλειάδες που έχουν αντιστοιχηθεί τόσο στην D όσο και στην E:

$$map(C) \leftarrow map(C) \cup map(D) \cup map(E)$$

Η περίπτωση του **συμπληρώματος** μιας κλάσης $C \equiv \neg D$ διαφέρει από τις προηγούμενες και χρήζει ιδιαίτερης προσοχής. Κάποιος θα μπορούσε να ισχυριστεί ότι επακόλουθο του παραπάνω αξιώματος είναι η ενημέρωση της αντιστοιχίας της C με τέτοιο τρόπο ώστε αυτή να περιλαμβάνει όλες τις πλειάδες που δεν έχουν αντιστοιχηθεί στην D, οπότε θα αρκούσε η αντιστοιχία με την άρνηση του $map(D)$. Εντούτοις, σύμφωνα με τη σημασιολογία της OWL και την υπόθεση ανοικτού κόσμου, δε γνωρίζουμε τι ισχύει για τις πλειάδες της ΒΔ που δεν έχουν αντιστοιχηθεί στην D: μπορεί να αναπαριστούν άτομα τύπου D ή όχι. Αν όμως, για παράδειγμα, γνωρίζαμε ότι επιπλέον ισχύει $D \sqcap E \sqsubseteq \bot$ τότε θα μπορούσαμε να συμπεράνουμε ότι η C θα περιλαμβάνει (τουλάχιστον) τα άτομα της E, αφού αυτά ανήκουν στο συμπλήρωμα της D, λόγω του αξιώματος ξένων κλάσεων. Σε μια τέτοια περίπτωση και μόνο, θα έπρεπε να επεκταθεί η αντιστοιχία της C κατά τα γνωστά ώστε να περιλάβει και τις πλειάδες που έχουν αντιστοιχηθεί στην E:

$$map(C) \leftarrow map(C) \cup map(E)$$

Παράδειγμα 4.1.3. Έστω η σχέση EMP (EmpID, EmpName, Email, ReportsTo, IsTemp), ο οποίος καταγράφει κάποια βασικά στοιχεία για τους εργαζόμενους μιας εταιρείας, με τη στήλη ReportsTo να περιέχει το αναγνωριστικό του προϊσταμένου ενός εργαζομένου, αν υπάρχει τέτοιος, και τη στήλη IsTemp να δηλώνει αν ένας εργαζόμενος είναι προσωρινός. Ένα ενδεικτικό στιγμιότυπο της σχέσης EMP απεικονίζεται στον επόμενο πίνακα:

EMP

EmpID	EmpName	Email	ReportsTo	IsTemp
1843	Daffy Duck	daffy@acme.com	4839	F
4839	Elmer Fudd	elmer@acme.com	-	F
3911	Wile E. Coyote	wile@acme.com	-	T

Έστω επίσης μια οντολογία που μοντελοποιεί τον ίδιο γνωστικό τομέα και περιέχει τις σχετικές κλάσεις *Employee*, *Manager* και *Temp*, καθώς και τα αξιώματα:

$$Manager \sqsubseteq Employee$$
$$Temp \sqsubseteq Employee$$
$$Manager \sqcap Temp \sqsubseteq \bot$$

Αν υποτεθεί ότι κάποιος χρήστης ορίσει την αντιστοιχία για την κλάση *Manager*:

$$Manager \rightsquigarrow \text{“ SELECT * FROM EMP WHERE ReportsTo = null ”} = q_1 \quad (4.3)$$

τότε λόγω του πρώτου αξιώματος υπαγωγής και των όσων αναλύθηκαν προηγουμένως, θα πρέπει να προστεθεί και η ακόλουθη αντιστοιχία για την κλάση *Employee*:

$$Employee \rightsquigarrow \text{“ SELECT * FROM EMP WHERE ReportsTo = null ”} = q_1 \quad (4.4)$$

Αν στη συνέχεια, ο χρήστης ορίσει την επόμενη αντιστοιχία για την κλάση *Temp*:

$$Temp \rightsquigarrow \text{“ SELECT * FROM EMP WHERE isTemp = ‘T’ ”} = q_2 \quad (4.5)$$

η διαδικασία συλλογισμού θα εκτελέσει τον αλγόριθμο 1 για το ζεύγος ξένων κλάσεων *Employee* και *Temp*, υπολογίζοντας το ερώτημα q_1 `INTERSECT` q_2, το οποίο επιστρέφει ως αποτέλεσμα την τρίτη πλειάδα της σχέσης EMP: ⟨3911, Wile E. Coyote, wile@acme.com, null, T⟩. Αυτό σημαίνει ότι η συνολική επισημειωμένη οντολογία εμπεριέχει μια αντίφαση, γεγονός που υποδηλώνει ότι τα δεδομένα της σχεσιακής ΒΔ, σε συνδυασμό με τις αντιστοιχίες που όρισε ο χρήστης, δε συμφωνούν με τη σημασιολογία της θεωρούμενης οντολογίας.

Αντίθετα, αν η τρίτη πλειάδα απουσιάζει από το στιγμιότυπο της σχέσης EMP, τότε δεν παρουσιάζεται κάποια ασυνέπεια και η διαδικασία συλλογισμού προχωρά με την ενημέρωση της αντιστοιχίας της κλάσης *Employee*, λόγω του δεύτερου αξιώματος υπαγωγής:

$$Employee \rightsquigarrow q_1 \ \texttt{UNION} \ q_2 \quad (4.6)$$

OWL αξιώματα ως περιορισμοί ακεραιότητας

Οι παραπάνω ενημερώσεις αντιστοιχιών εξασφαλίζουν την εξαγωγή των αναμενόμενων συμπερασμάτων, σύμφωνα με τη σημασιολογία της OWL και τη μορφή του εικονικού A_V της εξίσωσης (4.2). Εντούτοις, αρκετές από τις παραπάνω ενέργειες ίσως να μη συμφωνούν με τη διαίσθηση χρηστών μη εξοικειωμένων με την υπόθεση ανοικτού κόσμου της OWL, η οποία έρχεται σε αντίθεση με την υπόθεση κλειστού κόσμου που ισχύει σε συστήματα ΒΔ. Η διαφορά μεταξύ των δύο υποθέσεων οδηγεί συχνά σε παρεξηγήσεις σχετικά με το ρόλο που διαδραματίζουν τα αξιώματα μιας OWL οντολογίας κατά τη διαδικασία συλλογισμού στο σώμα ισχυρισμών μιας βάσης γνώσης. Σύμφωνα με την OWL σημασιολογία, τα OWL αξιώματα οδηγούν στην *εξαγωγή νέων συμπερασμάτων* και δεν λειτουργούν ως *περιορισμοί ακεραιότητας*, οι οποίοι καθιστούν ένα σώμα ισχυρισμών ασυνεπές αν λείπει από αυτό κάποια πληροφορία. Η διαφορά μεταξύ αυτών των δύο τρόπων αντιμετώπισης των OWL αξιωμάτων έχει μελετηθεί στο [142], όπου μάλιστα προτείνεται η διάκριση των ορολογικών αξιωμάτων μιας οντολογίας ανάλογα με τη χρήση τους: ως συνηθισμένα OWL αξιώματα ή ως περιορισμοί ακεραιότητας.

Στη συνέχεια, εξετάζουμε τα ίδια αξιώματα που είδαμε και προηγουμένως, αλλά αυτή τη φορά ως περιορισμούς ακεραιότητας οι οποίοι μπορεί να απορρίψουν κάποια αντιστοιχία, αν αυτή έρχεται σε σύγκρουση με κάποια ήδη ορισμένη αντιστοιχία και το τρέχον εικονικό σώμα ισχυρισμών A_V. Πιστεύουμε

ότι αυτή η ερμηνεία είναι πιο κοντά στη διαίσθηση απλών χρηστών, οι οποίοι αντιμετωπίζουν μια OWL οντολογία ως ένα απλό αντικειμενοστρεφές μοντέλο. Υπενθυμίζουμε ότι, και σε αυτή τη θεώρηση, ισχύει η υπόθεση μη επικάλυψης μεταξύ των εικονικών ατόμων του A_V και των ατόμων του αρχικού A_O, οπότε οι έλεγχοι περιορίζονται σε OWL κλάσεις που έχουν ήδη αντιστοιχηθεί σε κάποιο SQL ερώτημα. Στους επόμενους ελέγχους, ελέγχεται η αντιστοιχία $map(C)$ μιας κλάσης C ως προς ήδη υπάρχουσες αντιστοιχίες, οι οποίες θεωρούνται νοηματικά ορθές.

Ξεκινώντας από **αξιώματα υπαγωγής κλάσεων**, χρειάζεται να διακρίνουμε δύο περιπτώσεις, ανάλογα με το αν η κλάση C συμμετέχει σε αυτά ως υποκλάση ή υπερκλάση. Στην πρώτη περίπτωση, για κάθε συνεπαγόμενο αξίωμα $C \sqsubseteq D$, όπου D μια ήδη επισημειωμένη OWL κλάση, θα πρέπει να εξασφαλιστεί ότι $C^I \sqsubseteq D^I$, χωρίς κάποια επέκταση της ερμηνείας της D. Αυτό πρακτικά σημαίνει ότι το σύνολο των πλειάδων του αποτελέσματος της εκτέλεσης του $map(C)$ θα πρέπει να είναι υποσύνολο των πλειάδων του αντίστοιχου αποτελέσματος $map(D)(DB_I)$ για την υπερκλάση D. Διαφορετικά, η αντιστοιχία $map(C)$ απορρίπτεται ως μη αποδεκτή με βάση την οντολογία O και το εικονικό σώμα ισχυρισμών A_V. Η διαδικασία αυτή απεικονίζεται στον αλγόριθμο 3.

Αλγόριθμος 3 *Έλεγχος υπερκλάσεων*

Είσοδος: OWL κλάση C, σύνολο πλειάδων *tuples*, επισημειωμένη οντολογία O_{annot}
Έξοδος: True / False, ανάλογα με το αν το ζεύγος (C, *tuples*) είναι σύμφωνο με την O_{annot} και τα σύνολα πλειάδων στα οποία έχουν αντιστοιχηθεί οι υπερκλάσεις της C

```
1: function ΕΛΕΓΧΟΣ_ΥΠΕΡΚΛΑΣΕΩΝ(C, tuples, O_annot)
2:     C_super ← σύνολο συνεπαγόμενων υπερκλάσεων της C
3:     for all D στο C_super do
4:         if D έχει τιμή για ιδιότητα επισημείωσης p_map στην O_annot then
5:             D_query ← τιμή της ιδιότητας επισημείωσης p_map για D
6:             D_tuples ← σύνολο πλειάδων του αποτελέσματος της εκτέλεσης του D_query
7:             if tuples ⊈ D_tuples then return false
8:             end if
9:         end if
10:    end for
11:    return true
12: end function
```

Ανάλογος είναι και ο έλεγχος της αντιστοιχίας $map(C)$ ως προς όλες τις συνεπαγόμενες υποκλάσεις της C. Ο αλγόριθμος 4, για κάθε αξίωμα της μορφής $D \sqsubseteq C$, ελέγχει αν το σύνολο των πλειάδων του $map(C)(DB_I)$ είναι υπερσύνολο των πλειάδων του $map(D)(DB_I)$. Η σύγκριση των συνόλων πλειάδων, που επιτελείται στη γραμμή 7 των αλγορίθμων 3 και 4, μπορεί επίσης να επιτευχθεί με ένα SQL ερώτημα, αντικαθιστώντας τη γραμμή 6 στον αλγόριθμο 2 με:

$$q \leftarrow q'_1 \texttt{ EXCEPT } q'_2$$

το οποίο ισοδυναμεί με τη διαφορά δύο ερωτημάτων. Αν το αποτέλεσμα του q είναι κενό, τότε όλες οι πλειάδες του q_1 περιέχονται σε αυτές του q_2 και αυτό το συμπέρασμα χρησιμοποιείται κατάλληλα από τους αλγορίθμους 3 και 4.

Τα **αξιώματα τομής και ένωσης κλάσεων**, με βάση τη λογική που αναλύθηκε προηγουμένως, ισοδυναμούν με κατάλληλα αξιώματα υπαγωγής, οπότε

Αλγόριθμος 4 *Έλεγχος υποκλάσεων*

Είσοδος: OWL κλάση C, σύνολο πλειάδων *tuples*, επισημειωμένη οντολογία O_{annot}
Έξοδος: True / False, ανάλογα με το αν το ζεύγος $(C, tuples)$ είναι σύμφωνο με την O_{annot} και τα σύνολα πλειάδων στα οποία έχουν αντιστοιχηθεί οι υποκλάσεις της C

```
1: function ΕΛΕΓΧΟΣ_ΥΠΟΚΛΑΣΕΩΝ(C, tuples, O_annot)
2:     C_sub ← σύνολο συνεπαγόμενων υποκλάσεων της C
3:     for all D στο C_sub do
4:         if D έχει τιμή για ιδιότητα επισημείωσης p_map στην O_annot then
5:             D_query ← τιμή της ιδιότητας επισημείωσης p_map για D
6:             D_tuples ← σύνολο πλειάδων του αποτελέσματος της εκτέλεσης του D_query
7:             if D_tuples ⊈ tuples then return false
8:             end if
9:         end if
10:    end for
11:    return true
12: end function
```

αντιμετωπίζονται κατά τον ίδιο τρόπο. Όσον αφορά σε **αξιώματα ισοδύναμων κλάσεων** $C \equiv D$, η αναμενόμενη συμπεριφορά τους ως περιορισμοί ακεραιότητας θα επέβαλλε έλεγχο της ισοδυναμίας των πλειάδων των $map(C)(DB_I)$ και $map(D)(DB_I)$. Ένα μειονέκτημα της επιβολής ενός ελέγχου τέτοιου είδους είναι η αδυναμία μεταγενέστερης ενημέρωσης των αντιστοιχιών $map(C)$, $map(D)$, εφόσον κάθε προσπάθεια αντικατάστασης μίας εκ των δύο θα οδηγεί εκ των πραγμάτων σε σύγκρουση με βάση το αξίωμα ισοδυναμίας. Αυτό μπορεί να επιτευχθεί μόνο αν διαγραφούν και οι δύο αντιστοιχίες και αντικατασταθούν με νέες.

Τέλος, ένα **αξίωμα συμπληρώματος** $C \equiv \neg D$, ερμηνευόμενο ως περιορισμός ακεραιότητας, θα επέβαλλε στο $map(C)(DB_I)$ να περιλαμβάνει όλα τα δεδομένα της ΒΔ, εκτός αυτών που περιέχονται στο $map(D)(DB_I)$. Αυτός ο τελευταίος έλεγχος αποτελεί ένα χαρακτηριστικό παράδειγμα της υπόθεσης κλειστού κόσμου που υιοθετούν τα συστήματα ΒΔ, όπου θεωρείται ότι το σύνολο των ατόμων του κόσμου είναι αυτά που αναπαριστώνται από τις σχέσεις της ΒΔ. Το βασικό μειονέκτημα αυτού του είδους ελέγχου είναι η παρατήρηση ότι, μετά τον ορισμό του $map(C)$, ακόμα και η προσθήκη μιας νέας σχέσης στο σχήμα της ΒΔ καθιστά μη ικανοποιήσιμη την C, αφού πλέον ο χώρος των εικονικών ατόμων που προέρχονται από τη ΒΔ έχει επεκταθεί, με αποτέλεσμα η C να μην περιλαμβάνει *ολόκληρο* το συμπλήρωμα της D. Αυτός είναι και ο κύριος λόγος, σε συνδυασμό με τη χαμηλή συχνότητα εμφάνισης του συγκεκριμένου αξιώματος[3], για τον οποίο αποφεύγουμε ως μη πρακτικό τον έλεγχο αξιωμάτων συμπληρώματος.

Παράδειγμα 4.1.4. Επανερχόμενοι στο παράδειγμα 4.1.3, αντιμετωπίζουμε αυτή τη φορά τα αξιώματα της οντολογίας ως περιορισμούς ακεραιότητας και εξετάζουμε τις διαφορές στην πραγματοποίηση του συλλογισμού. Υπενθυμίζουμε ότι πλέον τα αξιώματα δεν οδηγούν σε ενημέρωση υπαρχουσών αντιστοιχιών, αλλά ως περιορισμοί που πρέπει να ικανοποιούν οι νέες αντιστοιχίες. Σύμφωνα λοιπόν με τη συγκεκριμένη ερμηνεία των OWL αξιωμάτων,

[3]Σύμφωνα με τα στατιστικά της σημασιολογικής μηχανής αναζήτησης Sindice (http://sindice.com/stats/direct/basic-predicate-stats), οι `owl:complementOf` και `owl:oneOf` είναι μακράν οι λιγότερο χρησιμοποιούμενες ιδιότητες της OWL.

η αρχική προσθήκη της αντιστοιχίας (4.3) για την κλάση *Manager* δε θα οδηγήσει στην αυτόματη προσθήκη της αντιστοιχίας (4.4) για την κλάση *Employee*. Απεναντίας, ο ορισμός της ακόλουθης αντιστοιχίας για την κλάση *Employee*:

$$Employee \rightsquigarrow \text{“ SELECT * FROM EMP ”} = q_3 \tag{4.7}$$

θα προκαλέσει την εκτέλεση του αλγορίθμου 4, προκειμένου να ελεγχθεί αν η συγκεκριμένη αντιστοιχία περιέχει όλα τα άτομα που υπονοούνται από την κλάση *Manager*. Πιο συγκεκριμένα, το ερώτημα q_1 EXCEPT q_3 δεν θα επιστρέψει κάποιο αποτέλεσμα και η αντιστοιχία (4.7) γίνεται αποδεκτή.

Ο ορισμός της αντιστοιχίας (4.5) για την κλάση *Temp* θα προκαλέσει την εκτέλεση του αλγορίθμου 3, λόγω του αξιώματος υπαγωγής $Temp \sqsubseteq Employee$ και θα γίνει αρχικά αποδεκτή (καθώς το SQL ερώτημα q_2 EXCEPT q_3 δεν επιστρέφει αποτέλεσμα). Λόγω όμως του αξιώματος ξένων κλάσεων $Manager \sqcap Temp \sqsubseteq \bot$, όπως ακριβώς και στο παράδειγμα 4.1.3, θα προκαλέσει την ασυνέπεια της επισημειωμένης οντολογίας και συνεπώς, θα απορριφθεί.

Εξαιρουμένης της διαδικασίας συλλογισμού στο TBox της οντολογίας για την εξαγωγή νέων αξιωμάτων υπαγωγής και ξένων κλάσεων η οποία μπορεί να υποτεθεί ότι πραγματοποιείται μία φορά στην αρχή της διαδικασίας, η κυριότερη πηγή πολυπλοκότητας στους ελέγχους σημασιολογικής ορθότητας που αναλύθηκαν παραπάνω είναι η σύγκριση των συνόλων πλειάδων που έχουν αντιστοιχηθεί στις ελεγχόμενες OWL κλάσεις. Όπως είδαμε, αυτές οι συγκρίσεις μπορούν να πραγματοποιηθούν με μια σειρά SQL ερωτημάτων, τα οποία υπολογίζονται σε χώρο LOGSPACE ως προς το μέγεθος της ΒΔ [192].

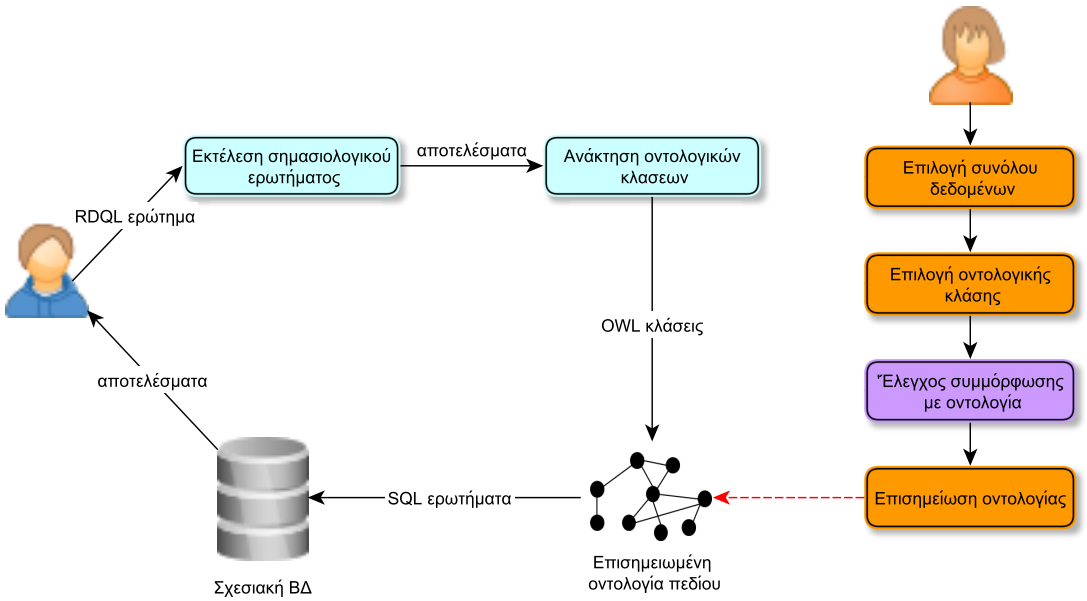
4.2 Αρχιτεκτονική του συστήματος

Οι βασικές αρχές της προσέγγισης της ενότητας 4.1 υλοποιήθηκαν στο σύστημα VisAVis, το οποίο περιγράφουμε συνοπτικά στην παρούσα ενότητα για λόγους πληρότητας της παρουσίασης. Η αρχιτεκτονική υψηλού επιπέδου του VisAVis απεικονίζεται στο σχήμα 4.1. Στο δεξιό τμήμα του εν λόγω σχήματος, φαίνονται τα διακριτά στάδια της διαδικασίας ορισμού μιας αντιστοιχίας, όπως αυτή ορίστηκε και αναλύθηκε στην ενότητα 4.1. Συγκεκριμένα, τα στάδια αυτά είναι:

1. επιλογή συνόλου δεδομένων από ΒΔ
2. επιλογή OWL κλάσης
3. έλεγχος συμμόρφωσης αντιστοιχίας με οντολογία
4. επισημείωση οντολογίας

Το σύστημα υλοποιήθηκε ως επέκταση (plugin) του δημοφιλούς περιβάλλοντος ανάπτυξης οντολογιών ανοικτού κώδικα Protégé[4], επιτρέποντας την εκμετάλλευση των δυνατοτήτων του τελευταίου για επεξεργασία, εφαρμογή συλλογισμού και οπτικοποίηση της οντολογίας πεδίου που χρησιμοποιείται για την αντιστοιχία. Τα λογισμικά ΣΔΒΔ που δοκιμάστηκαν κατά την υλοποί-

[4]Το Protégé είναι διαθέσιμο στο http://protege.stanford.edu/.



Σχήμα 4.1: *Αρχιτεκτονική υψηλού επιπέδου του VisAVis*

ηση του VisAVis ήταν τα δημοφιλή MySQL[5] και PostgreSQL[6] και η υλοποίηση της εφαρμογής πραγματοποιήθηκε σε Java[7].

Το γραφικό περιβάλλον της εφαρμογής υποστηρίζει τη γραφική κατασκευή ενός απλού SQL ερωτήματος και τη συσχέτιση αυτού με μια κλάση της οντολογίας, σύμφωνα με τον ορισμό 4.1.4. Πριν οριστικοποιηθεί η αντιστοιχία μεταξύ των δύο στοιχείων, πραγματοποιούνται οι έλεγχοι συμμόρφωσης αυτής με την επιλεγμένη οντολογία και τις ήδη ορισθείσες αντιστοιχίες. Οι έλεγχοι αυτοί αντιμετωπίζουν τα αξιώματα ορολογίας της οντολογίας ως περιορισμούς ακεραιότητας τους οποίους πρέπει να πληρούν τα *εικονικά* άτομα που προέρχονται από τη ΒΔ, όπως αναλύθηκε στην ενότητα 4.1. Οι έλεγχοι που υλοποιήθηκαν αφορούν μονάχα σε αξιώματα ξένων κλάσεων και υπαγωγής κλάσεων, τα οποία είναι και τα πιο συνηθισμένα μεταξύ των OWL αξιωμάτων που αναφέρονται αποκλειστικά σε κλάσεις. Οι έλεγχοι που συσχετίζονται με αξιώματα ισοδυναμίας και συμπληρώματος κλάσεων αποφεύχθηκαν για τους λόγους που αναφέρθηκαν στην ενότητα 4.1. Στην περίπτωση που οι προηγούμενοι έλεγχοι ικανοποιούνται, η αντιστοιχία αποθηκεύεται στην οντολογία, εμπλουτίζοντάς την με ένα αντίστοιχο αξίωμα επισημείωσης.

Η επιλογή αποθήκευσης της αντιστοιχίας στην οντολογία μέσω αξιωμάτων επισημείωσης δεν αλλοιώνει τη σημασιολογία της οντολογίας κατά OWL, αλλά επιτρέπει σε τρίτες εφαρμογές που αναγνωρίζουν τις εμφωλευμένες αντιστοιχίες να τις εκμεταλλευτούν κατάλληλα. Ένα τέτοιο σενάριο εκμετάλλευσης των αντιστοιχιών που προτείνουμε είναι η εκτέλεση σημασιολογικών ερωτημάτων στο συνδυασμό ΒΔ και επισημειωμένης οντολογίας, επιτρέποντας την ανάκτηση ατόμων του εικονικού ABox που παραμένουν αποθηκευμένα στη ΒΔ. Το σενάριο αυτό υποστηρίζεται από την υλοποίηση του VisAVis, καταδεικνύει στην πράξη τη χρησιμότητα αντιστοιχιών ΒΔ με οντολογία και σκιαγραφείται

[5] http://www.mysql.com/
[6] http://www.postgresql.org/
[7] Ο κώδικας της εφαρμογής είναι διαθέσιμος στο http://code.google.com/p/visavis/.

στο αριστερό μέρος του σχήματος 4.1.

Η γλώσσα σημασιολογικών ερωτημάτων που υποστηρίζει η συγκεκριμένη υλοποίηση είναι η RDQL (RDF Data Query Language), πρόδρομος της SPARQL με παρόμοια σύνταξη. Το αποτέλεσμα ενός RDQL ερωτήματος, ακριβώς όπως και αυτό της SPARQL, είναι ένα σύνολο *λύσεων*, όπου κάθε λύση αντιστοιχεί μία (το πολύ) τιμή σε κάθε μεταβλητή του ερωτήματος. Η προσέγγιση που ακολουθείται από το VisAVis είναι αρκετά χαλαρή, καθώς επιλέγει να επιστρέψει μια λίστα εικονικών ατόμων μιας OWL κλάσης και των δεδομένων που σχετίζονται με αυτά, στην περίπτωση που η συγκεκριμένη κλάση περιέχεται σε κάποια λύση του RDQL ερωτήματος (αλγόριθμος 5). Η υλοποίηση ενός αλγορίθμου επανεγγραφής του RDQL ερωτήματος σε ένα σημασιολογικά ισοδύναμο SQL ερώτημα βρίσκεται εκτός του πλαισίου της συγκεκριμένης προσέγγισης, στόχος της οποίας είναι απλώς η επίδειξη της χρησιμότητας μιας αντιστοιχίας ΒΔ με οντολογία σε ένα αναγνωρισμένο σενάριο χρήσης (τη σημασιολογική επερώτηση σχεσιακών δεδομένων). Εκτός αυτού, ένας αλγόριθμος επανεγγραφής χρειάζεται και έναν απτό μηχανισμό παραγωγής IRI από στοιχεία της ΒΔ, ο οποίος όμως δεν ορίζεται κατηγορηματικά στην προσέγγισή μας, όπως αναφέρθηκε και στην ενότητα 4.1. Ως εκ τούτου, το VisAVis δρα ως ενδιάμεσος, παρεμβαίνοντας στην τυπική διαδικασία εκτέλεσης σημασιολογικών ερωτημάτων μέσω της εκτέλεσης SQL ερωτημάτων και της παρουσίασης των αποτελεσμάτων αυτών στο χρήστη.

Αλγόριθμος 5 *Εκτέλεση σημασιολογικού ερωτήματος*

Είσοδος: RDQL ερώτημα q, επισημειωμένη οντολογία O_{annot}, στιγμιότυπο ΒΔ DB_I
Έξοδος: Σύνολα δεδομένων που αντιστοιχούν στα εικονικά άτομα των OWL κλάσεων που περιέχονται στο αποτέλεσμα της εκτέλεσης του q στην O_{annot}

```
1: function ΕΚΤΕΛΕΣΗ_ΣΗΜΑΣΙΟΛΟΓΙΚΟΥ_ΕΡΩΤΗΜΑΤΟΣ(q, O_annot, DB_I)
2:     results ← ∅
3:     q_res ← αποτελέσματα εκτέλεσης q στην O_annot
4:     for all C_i στο dom(p_map) do                    # για κάθε επισημειωμένη κλάση
5:         if C_i ∈ q_res then
6:             sql_query ← τιμή της ιδιότητας επισημείωσης p_map για C_i
7:             results ← results ∪ αποτέλεσμα εκτέλεσης sql_query στο DB_I
8:         end if
9:     end for
10:    return results
11: end function
```

Συνοψίζοντας, το VisAVis ανήκει στην ομάδα των εργαλείων που απαιτούν την ύπαρξη μιας οντολογίας προκειμένου να αντιστοιχήσουν όρους αυτής με μια σχεσιακή ΒΔ (ενότητα 3.5), ενώ ακολουθεί τον τρόπο λειτουργίας 2 στο σχήμα 3.6, παράγοντας μονάχα ένα σύνολο αντιστοιχιών. Επίσης, σύμφωνα με το συγκριτικό πλαίσιο της ενότητας 3.3, αποτελεί ένα χειροκίνητο εργαλείο, το οποίο χρησιμοποιεί ένα συνδυασμό SQL και OWL για την αναπαράσταση των αντιστοιχιών που ορίζονται μέσω αυτού. Κύριος στόχος του VisAVis και της γενικότερης φιλοσοφίας του, είναι ο εύκολος ορισμός μιας αντιστοιχίας σχεσιακής ΒΔ με οντολογία και η εξέταση των ζητημάτων που προκύπτουν στη συνεργασία των δύο διαφορετικών αυτών φορμαλισμών. Η αντιστοιχία αυτή ουσιαστικά περιγράφει το νόημα του περιεχομένου μιας ΒΔ σε τυπικούς όρους από μια οντολογία. Το σαφώς ορισμένο αυτό νόημα μπορεί να χρησιμοποιηθεί

σε διάφορες εφαρμογές και σενάρια χρήσης, όπως θα δούμε στην ενότητα 4.3.

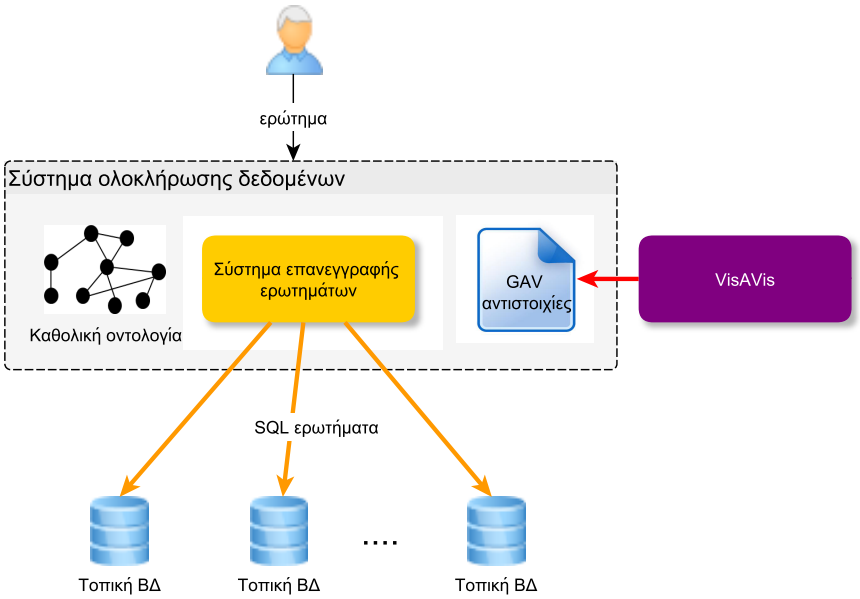
4.3 Σενάρια χρήσης

Οι αντιστοιχίες που ορίζονται μέσω του VisAVis μπορούν να χρησιμοποιηθούν και σε άλλα σενάρια εφαρμογών, πέρα από αυτό της **σημασιολογικής επερώτησης σχεσιακών δεδομένων**. Όπως περιγράφηκε και στην ενότητα 4.2, η προσέγγιση που υιοθετήθηκε στην εκτέλεση σημασιολογικών ερωτημάτων μπορεί να χαρακτηριστεί ως χαλαρή, καθώς τα αποτελέσματα της εκτέλεσης δεν ακολουθούν τη μορφή που επιβάλλει το πρότυπο της RDQL, όμως αυτό δεν παύει να αποτελεί μια εφαρμογή που λειτουργεί ως ένδειξη για τα οφέλη μιας αντιστοιχίας ΒΔ με οντολογία. Παράλληλα, οι εν λόγω αντιστοιχίες μπορούν, σε συνδυασμό με ένα σαφώς ορισμένο μηχανισμό παραγωγής IRI, να χρησιμοποιηθούν από έναν τυπικό αλγόριθμο επανεγγραφής SPARQL ερωτημάτων σε SQL, τα τελικά αποτελέσματα του οποίου θα έχουν την μορφή που επιβάλλεται από το πρότυπο της SPARQL.

Ένα άλλο σενάριο χρήσης αντιστοιχιών που ορίζονται μέσω του VisAVis θα μπορούσε να αφορά σε ένα **σύστημα ολοκλήρωσης δεδομένων**, όπως αυτό του σχήματος 4.2. Όπως αναφέραμε και στην ενότητα 4.1, η μορφή που έχουν οι αντιστοιχίες αυτές μοιάζουν με GAV αντιστοιχίες, δηλαδή εκφράσεις όρων μιας καθολικής οντολογίας ως SQL ερωτήματα σε μία ή περισσότερες ετερογενείς ΒΔ. Σε ένα σύστημα ολοκλήρωσης δεδομένων, ερωτήματα τίθενται σε όρους του καθολικού σχήματος και οι απαντήσεις περιλαμβάνουν δεδομένα από τις τοπικές πηγές δεδομένων. Λόγω της φύσης των GAV αντιστοιχιών, η διαδικασία επανεγγραφής του αρχικού ερωτήματος σε επιμέρους ερωτήματα στις τοπικές ΒΔ θεωρείται τετριμμένη, καθώς προκύπτει με απευθείας αντικατάσταση των όρων της καθολικής οντολογίας με τα SQL ερωτήματα που δηλώνουν οι GAV αντιστοιχίες. Αντίθετα, το μειονέκτημα ενός συστήματος ολοκλήρωσης που βασίζεται σε GAV αντιστοιχίες είναι η δυσκολία ενημέρωσης των τελευταίων όταν εισάγεται στο σύστημα μια νέα πηγή δεδομένων, οπότε και πρέπει να μεταβληθούν κατάλληλα όλες οι GAV αντιστοιχίες που θα αναφέρονται στη νέα πηγή.

Το εν λόγω σενάριο χρήσης αποτελεί ουσιαστικά μια επέκταση της λειτουργίας σημασιολογικών επερωτήσεων που είναι υλοποιημένη στο VisAVis, όπου πλέον κάθε αντιστοιχία θα μπορεί να αναφέρεται σε περισσότερες της μίας σχεσιακές ΒΔ. Με άλλα λόγια, για ένα σύστημα ολοκλήρωσης με n τοπικές ΒΔ $DB_1, DB_2, \ldots, DB_n$, μια OWL κλάση C θα αντιστοιχείται σε μια ένωση ερωτημάτων $q_1 \cup q_2 \cup \ldots \cup q_m, m \leq n$, καθένα εκ των οποίων τίθεται σε μία εκ των $DB_1, DB_2, \ldots, DB_n$.

Μια παραπλήσια κατηγορία συστημάτων όπου μπορούν να βρουν εφαρμογή οι αντιστοιχίες του VisAVis είναι τα **συστήματα ανταλλαγής δεδομένων** (data exchange) [79]. Στόχος αυτών των συστημάτων είναι η μεταφορά δεδομένων από ένα σχήμα-πηγή Σ_S σε ένα σχήμα-στόχο Σ_T. Οι κυριότερες διαφορές αυτών των συστημάτων με συστήματα ολοκλήρωσης δεδομένων είναι: α) το γεγονός ότι στην περίπτωση των πρώτων, το σχήμα-πηγή έχει αναπτυχθεί ανεξάρτητα από το σχήμα-στόχο, ενώ αντίθετα στην περίπτωση των δεύτερων, το καθολικό σχήμα έχει προκύψει ως μια συμφιλιωμένη όψη των τοπικών σχημάτων και β) το γεγονός ότι συνήθως, σε συστήματα ανταλλαγής

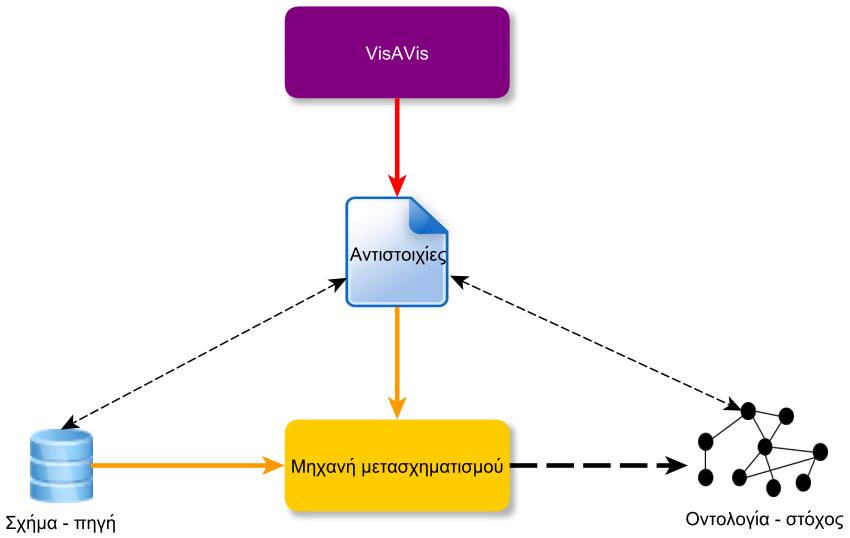


Σχήμα 4.2: *Χρήση του VisAVis σε σύστημα ολοκλήρωσης δεδομένων*

δεδομένων, τα περιεχόμενα του σχήματος-πηγής, αφού υποστούν τους αναγκαίους μετασχηματισμούς, μεταφέρονται σε *φυσική μορφή* στο σχήμα-στόχο. Αν και ένα από τα πλεονεκτήματα της προσέγγισης του VisAVis και γενικότερα των δυναμικών προσεγγίσεων αντιστοιχίας είναι η αποφυγή της επανάληψης των δεδομένων σε περισσότερες της μίας τοποθεσίες, υπάρχουν περιπτώσεις όπου είναι επιθυμητός ο εμπλουτισμός μιας οντολογίας με άτομα σε φυσική μορφή και μια αντιστοιχία ΒΔ με οντολογία μπορεί να χρησιμεύσει για αυτό το σκοπό. Σε αυτή την περίπτωση, το σώμα ισχυρισμών της οντολογίας θα είναι πλέον υλοποιημένο (materialized) και όχι εικονικό, όπως υποτέθηκε στην ενότητα 4.1. Βέβαια, όπως και στα υπόλοιπα σενάρια χρήσης, είναι απαραίτητος ο ορισμός ενός μηχανισμού παραγωγής IRI για τα οντολογικά άτομα που δημιουργούνται από τα περιεχόμενα της ΒΔ. Η συμμετοχή του VisAVis σε μια τυπική αρχιτεκτονική ενός συστήματος ανταλλαγής δεδομένων απεικονίζεται στο σχήμα 4.3.

4.4 Συμπεράσματα

Σε αυτό το κεφάλαιο, παρουσιάστηκε μια απλή προσέγγιση αντιστοιχίας των περιεχομένων μιας σχεσιακής βάσης δεδομένων με κλάσεις μιας συγκεκριμένης OWL οντολογίας. Η απλότητα της προτεινόμενης προσέγγισης οφείλεται στη χρήση της SQL ως γλώσσας ορισμού της αντιστοιχίας, γεγονός που διευκολύνει σημαντικά το χειροκίνητο ορισμό μιας αντιστοιχίας ακόμα και από κάποιο μη εξοικειωμένο με τεχνολογίες Σημασιολογικού Ιστού χρήστη. Ο τυπικός ορισμός της αντιστοιχίας και της σημασιολογίας του αποτελέσματος της εφαρμογής της, δεδομένου ενός στιγμιοτύπου σχεσιακής ΒΔ και μιας οντολογίας, κατέδειξε την αδυναμία της SQL να υποκαταστήσει από μόνη της, χωρίς την επιβολή συγκεκριμένων περιορισμών, μια ολοκληρωμένη γλώσσα ορισμού αντιστοιχίας. Συγκεκριμένα, οι δύο κυριότερες ελλείψεις που αναγνωρίστη-



Σχήμα 4.3: *Χρήση του VisAVis σε σύστημα ανταλλαγής δεδομένων*

καν ήταν: α) η απουσία ενός μηχανισμού παραγωγής IRI για νέα οντολογικά άτομα που δημιουργούνται από τα περιεχόμενα της ΒΔ και β) η έλλειψη ενός μηχανισμού αντιστοιχίας OWL ιδιοτήτων. Τα συγκεκριμένα χαρακτηριστικά μαζί με τη δυνατότητα επιλογής ενός αυθαίρετου συνόλου δεδομένων από τη ΒΔ αποτελούν τους βασικούς άξονες που θα πρέπει να διαθέτει μια ελάχιστη γλώσσα ορισμού αντιστοιχίας ΒΔ σε οντολογία.

Για τον έλεγχο της συνέπειας μιας οντολογίας που συνοδεύεται από ένα σύνολο αντιστοιχιών αλλά και για την εξαγωγή συμπερασμάτων από αυτή, προτάθηκαν δύο είδη αλγορίθμων συλλογισμού: αυτοί που ακολουθούν την OWL σημασιολογία και αυτοί που αντιμετωπίζουν τα OWL αξιώματα ως περιορισμούς ακεραιότητας. Οι τελευταίοι πλησιάζουν περισσότερο τη διαίσθηση ενός μέσου χρήστη, ο οποίος αντιμετωπίζει (λανθασμένα) την οντολογία ως ένα εννοιολογικό μοντέλο που θέτει επιπλέον περιορισμούς στο σχήμα μιας σχεσιακής ΒΔ. Παρουσιάστηκε συνοπτικά το σύστημα VisAVis που υλοποιεί το προτεινόμενο θεωρητικό πλαίσιο αντιστοιχίας και αξιοποιεί αντιστοιχίες προκειμένου να επιτρέψει την επερώτηση μιας σχεσιακής ΒΔ μέσω ενός σημασιολογικού ερωτήματος που χρησιμοποιεί όρους της συνδεδεμένης οντολογίας. Επίσης, υποστηρίχθηκε ότι αντιστοιχίες τέτοιου είδους μπορούν να αξιοποιηθούν και στο πλαίσιο εφαρμογών ολοκλήρωσης και ανταλλαγής δεδομένων.

Ενδεχομένως, η επιβολή συγκεκριμένων περιορισμών στα SQL ερωτήματα που επιτρέπονται να αντιστοιχηθούν στην OWL οντολογία θα μπορούσε να συνεισφέρει στην αντιμετώπιση των ελλείψεων που αναφέρθηκαν προηγουμένως. Συγκεκριμένα, για την υποκατάσταση ενός μηχανισμού παραγωγής IRI, θα μπορούσε να υιοθετηθεί η σύμβαση ότι κάθε SQL ερώτημα που αντιστοιχεί σε μια OWL κλάση θα πρέπει να διαθέτει ένα προβαλλόμενο γνώρισμα, οι τιμές του οποίου θα αποτελούν τα IRIs των νεο-δημιουργηθέντων ατόμων της κλάσης. Αντίστοιχα, για την υποστήριξη αντιστοιχίας OWL ιδιοτήτων, θα μπορούσε να ακολουθηθεί η ίδια λογική επισημείωσης μιας ιδιότητας με ένα SQL ερώτημα, με τον πρόσθετο περιορισμό ότι το ερώτημα αυτό θα πρέπει να διαθέτει δύο προβαλλόμενα γνωρίσματα, τα οποία θα αποτελούν το πεδίο

ορισμού και το πεδίο τιμών της συγκεκριμένης ιδιότητας. Η δυνατότητα αντιστοιχίας ιδιοτήτων θα επέβαλλε την ανάγκη για τον ορισμό επιπρόσθετων αλγορίθμων συλλογισμού, οι οποίοι θα λάμβαναν υπόψη τους και άλλα OWL αξιώματα, πέραν αυτών που εξετάστηκαν στο κεφάλαιο αυτό.

Η λιτότητα της συγκεκριμένης μορφής αντιστοιχίας δεν επιτρέπει την πλήρη συμμόρφωση με γλώσσες σημασιολογικών ερωτημάτων, όταν οι τελευταίες χρησιμοποιούνται για την επερώτηση των σχεσιακών δεδομένων. Για το σκοπό αυτό, είναι απαραίτητη η έκφραση της αντιστοιχίας με τη βοήθεια ολοκληρωμένων γλωσσών, όπως η R2RML. Στο κεφάλαιο 5 προτείνεται μια μέθοδος υποβολής σημασιολογικών ερωτημάτων σε σχεσιακές ΒΔ όταν οι αντιστοιχίες μεταξύ της ΒΔ και ενός RDF γράφου είναι εκφρασμένες σε R2RML.

Κεφάλαιο 5

Δυναμική SPARQL πρόσβαση στα περιεχόμενα σχεσιακής ΒΔ μέσω R2RML αντιστοιχιών

Περιεχόμενα

Όπως αναφέρθηκε στην ενότητα 3.3, οι μέθοδοι αντιστοιχίας ΒΔ με οντολογία μπορούν να διακριθούν σε *στατικές* και *δυναμικές*, ανάλογα με το αν το αποτέλεσμα εφαρμογής της αντιστοιχίας υλοποιείται σε φυσική μορφή ή όχι. Τα δύο βασικά μειονεκτήματα των στατικών μεθόδων εντοπίζονται αφενός στο γεγονός ότι το αποτέλεσμα της αντιστοιχίας μπορεί σύντομα να καταστεί παρωχημένο αν το περιεχόμενο της ΒΔ μεταβληθεί – οπότε χρειάζεται να επαναληφθεί η εκτέλεση της αντιστοιχίας για να εξασφαλιστεί ο συγχρονισμός των δεδομένων – και αφετέρου στις αυξημένες απαιτήσεις τους σε αποθηκευτικό χώρο, καθώς το υλοποιημένο αποτέλεσμα επαναλαμβάνει το συνολικό περιεχόμενο της ΒΔ. Τα συγκεκριμένα μειονεκτήματα δεν υφίστανται σε περιπτώσεις δυναμικών μεθόδων που ανακτούν ολόκληρο ή μέρος του αποτελέσματος της αντιστοιχίας όταν το χρειαστούν, γεγονός που εξασφαλίζει την επικαιρότητα του αποτελέσματος χωρίς να απαιτεί επιπρόσθετο αποθηκευτικό μέσο ή χώρο.

Σε αυτό το κεφάλαιο, παρουσιάζεται μια μέθοδος δυναμικής πρόσβασης σε έναν *εικονικό* RDF γράφο που υπονοείται από το συνδυασμό ενός στιγμιότυπου σχεσιακής ΒΔ και μιας R2RML αντιστοιχίας. Συγκεκριμένα, προτείνε-

ται ένας πρωτότυπος αλγόριθμος για την επανεγγραφή SPARQL ερωτημάτων που αναφέρονται στον εικονικό RDF γράφο σε σημασιολογικά ισοδύναμα SQL ερωτήματα που εκτελούνται στο υποκείμενο στιγμιότυπο σχεσιακής ΒΔ. Η μέθοδος αυτή χαρακτηρίζεται από τα πλεονεκτήματα των δυναμικών μεθόδων και ταυτόχρονα επιτυγχάνει καλύτερη απόδοση σε σύγκριση με αντίστοιχους αλγορίθμους επανεγγραφής, οι οποίοι αναφέρονται στην ενότητα 5.1. Ο προτεινόμενος αλγόριθμος υλοποιήθηκε στο σύστημα αντιστοιχίας RDB4RDF, η αρχιτεκτονική του οποίου παρουσιάζεται στην ενότητα 5.2, ενώ η κεντρική ιδέα και ο αλγόριθμος επανεγγραφής αποτελούν το αντικείμενο της ενότητας 5.3. Μια πρώτη αξιολόγηση του συστήματος πραγματοποιείται στην ενότητα 5.4, ενώ τα συμπεράσματα του κεφαλαίου και η μελλοντική εργασία αποτελούν το αντικείμενο της ενότητας 5.5.

5.1 Σχετικές εργασίες και κίνητρο

Όπως καταγράφηκε και στο κεφάλαιο 3, πλήθος μεθόδων και συστημάτων εξετάζουν το ζήτημα της δυναμικής πρόσβασης στο αποτέλεσμα μιας αντιστοιχίας σχεσιακής ΒΔ και οντολογίας (βλέπε και στήλη «*Προσβασιμότητα*» στον πίνακα 3.6). Στη σχετική βιβλιογραφία μεθόδων δυναμικής σημασιολογικής πρόσβασης σε ΒΔ, παρουσιάζεται αξιοσημείωτη ποικιλομορφία σε χαρακτηριστικά όπως η γλώσσα ερωτημάτων, η μορφή της αντιστοιχίας ή ακόμα και το σχεσιακό σχήμα που υποθέτει κάθε μία από αυτές. Η πλειονότητα των μεθόδων εξετάζει τη SPARQL γλώσσα ερωτημάτων, με ελάχιστες εξαιρέσεις (RDQL στο [156] και RQL στο [187]), ενώ σχεδόν όλοι οι προταθέντες αλγόριθμοι είναι γενικοί και μπορούν να εφαρμοστούν σε ένα οποιοδήποτε σχεσιακό σχήμα (μια εξαίρεση αποτελεί το [94]). Όσον αφορά στη μορφή της αντιστοιχίας που εξετάζει κάθε μέθοδος, κάποιες μέθοδοι υποθέτουν ότι οι αντιστοιχίες έχουν τη μορφή Horn προτάσεων (όπως στα [59, 156]), ενώ άλλες υποθέτουν 1:N αντιστοιχίες μεταξύ στοιχείων του σχεσιακού σχήματος και RDF όρων (όπως π.χ. στα [57, 77]).

Απεναντίας, υπάρχει έλλειψη ολοκληρωμένων αλγορίθμων επανεγγραφής SPARQL ερωτημάτων σε σημασιολογικά ισοδύναμα SQL ερωτήματα, όταν η αντιστοιχία μεταξύ σχεσιακής ΒΔ και RDF εκφράζεται σε R2RML [84]. Όσον αφορά στα λογισμικά που υλοποιούν το πρότυπο της R2RML[1] παρέχοντας δυναμική πρόσβαση σε μια σχεσιακή ΒΔ μέσω SPARQL, κανένα από αυτά δεν έχει τεκμηριώσει επαρκώς τον αλγόριθμο επανεγγραφής του, ενώ παράλληλα αρκετά (όπως π.χ. το D2RQ [41]) είναι ακόμα σε δοκιμαστικό στάδιο υποστήριξης της R2RML.

Επιπλέον, οι περισσότεροι αλγόριθμοι επανεγγραφής SPARQL σε SQL [57, 67, 84, 112, 134] παράγουν SQL ερωτήματα που χαρακτηρίζονται από: α) *υψηλό βαθμό εμφώλευσης* (nesting) και β) *περισσότερες συνενώσεις* (joins) από αυτές που είναι απολύτως απαραίτητες. SQL ερωτήματα με αυτά τα χαρακτηριστικά είναι συνήθως μη αποδοτικά, με αποτέλεσμα να επηρεάζεται η συνολική αποδοτικότητα ενός συστήματος δυναμικής πρόσβασης σε μια ΒΔ μέσω SPARQL.

[1] Μια ενδεικτική λίστα υπάρχει στο http://www.w3.org/TR/rdb2rdf-implementations/.

Ελαφρώς διαφορετική φαίνεται να είναι η στρατηγική του δημοφιλούς συστήματος αντιστοιχίας ΒΔ με RDF γράφους, D2RQ[2], το οποίο αποτιμά ξεχωριστά κάθε τελεστή ενός SPARQL ερωτήματος και συνδυάζει τα αποτελέσματά τους σύμφωνα με τη σημασιολογία της SPARQL. Αυτή η στρατηγική έχει ως αποτέλεσμα τη συχνή αποστολή SQL ερωτημάτων προς εκτέλεση στη ΒΔ, με τον αριθμό των ερωτημάτων να αυξάνεται όσο μεγαλύτερος είναι ο όγκος των περιεχομένων της ΒΔ, ενώ επιπλέον, ένα μεγάλο μέρος της επεξεργασίας των αποτελεσμάτων (π.χ. συνάθροιση, διάταξη) γίνεται από την ίδια την εφαρμογή. Συνέπεια των παραπάνω είναι η μειωμένη επίδοση του συστήματος σε ΒΔ με μεγάλο όγκο δεδομένων.

Η προσέγγιση του Ultrawrap [173] κινείται στο αντίθετο άκρο, αναθέτοντας όλη την επεξεργασία και τις πιθανές βελτιστοποιήσεις του SQL ερωτήματος στη μηχανή ερωτημάτων της ΒΔ. Η βασική ιδέα του Ultrawrap είναι ο ορισμός κατάλληλων όψεων στη σχεσιακή ΒΔ, οι οποίες αντανακλούν τα RDF δεδομένα και το σχήμα τους μοιάζει με το σχήμα ενός συστήματος αποθήκευσης RDF δεδομένων. Δεδομένων αυτών των όψεων, η διαδικασία επανεγγραφής των SPARQL ερωτημάτων σε SQL είναι τετριμμένη και η αποδοτικότητα του συστήματος επαφίεται πλήρως στην ικανότητα βελτιστοποίησης της μηχανής εκτέλεσης ερωτημάτων της ΒΔ, γεγονός που δεν εγγυάται βέλτιστα αποτελέσματα για κάθε ΣΔΒΔ. Επίσης, η προσέγγιση αυτή προϋποθέτει τη δυνατότητα ορισμού νέων όψεων στη ΒΔ, υπόθεση που μπορεί να μην ισχύει πάντα λόγω έλλειψης των απαραίτητων δικαιωμάτων χρήστη, ενώ παράλληλα αλλοιώνει το σχήμα της ΒΔ με στοιχεία που δεν έχουν σχέση με το γνωστικό τομέα των περιεχομένων της ΒΔ.

Τα παραπάνω γίνονται περισσότερο κατανοητά στο παράδειγμα 5.1.1, όπου δίνεται μια πολύ αδρή σκιαγράφηση των συγκεκριμένων αλγορίθμων και συγκρίνονται τα παραγόμενα SQL ερωτήματα για καθέναν από αυτούς.

Παράδειγμα 5.1.1. Έστω μια σχέση Review(nr, product, rating, date) και μια συνάρτηση αντιστοιχίας *map*, η οποία καθορίζει τις RDF τριάδες που θα παραχθούν από το στιγμιότυπο της Review. Έστω ότι η *map* αντιστοιχεί καθένα από τα γνωρίσματα της Review σε ισάριθμα RDF κατηγορήματα, οπότε *map*(nr) = ex:id, *map*(product)=ex:isReviewFor, *map*(rating) = ex:hasRating, *map*(date) = ex:reviewDate και επίσης, έστω ότι το γνώρισμα nr χρησιμεύει για την παραγωγή ενός IRI για κάθε πλειάδα της Review. Θεωρούμε ότι το ακόλουθο SPARQL ερώτημα, το οποίο ανακτά όλα τα διαθέσιμα προϊόντα που έχουν βαθμολογηθεί, τίθεται στον – υλοποιημένο (materialized) ή εικονικό (virtual) – RDF γράφο που ορίζεται από το συνδυασμό του επόμενου στιγμιοτύπου της σχέσης Review και της συνάρτησης αντιστοιχίας *map*:

Review

nr	product	rating	date
245	Twix	7	2011-07-23
682	Snickers	8	-
784	Bounty	3	2012-09-14
815	Mars	-	-

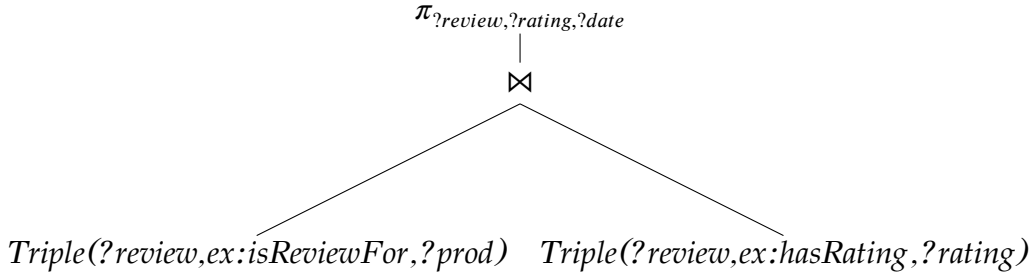
```
PREFIX ex: <http://example.org>
SELECT ?review ?prod ?rating
WHERE {
(tp1):   ?review ex:isReviewFor ?prod.
(tp2):   ?review ex:hasRating ?rating
      }
```

[2]Η τεκμηρίωση του ακριβούς αλγορίθμου επανεγγραφής του D2RQ δεν είναι διαθέσιμη κάπου και τα συμπεράσματα για τον τρόπο λειτουργίας του προέρχονται από μελέτη του κώδικα της εφαρμογής.

Δεδομένης της αντιστοιχίας *map* και του στιγμιοτύπου της Review, η αναμενόμενη SPARQL λύση θα αποτελείται από τρεις αντιστοιχίες των μεταβλητών ?review, ?prod και ?rating μ_1, μ_2 και μ_3[3], όπου έχει υποτεθεί ότι το IRI κάθε πλειάδας της Review προκύπτει μετά από συνένωση ενός βασικού IRI με την τιμή του γνωρίσματος nr. Σημειώνεται ότι η τέταρτη πλειάδα του στιγμιοτύπου της Review δεν συμμετέχει στη SPARQL λύση, εφόσον δεν παράγεται κάποια τιμή που να μπορεί να ανατεθεί στη μεταβλητή ?rating.

	review	prod	rating
μ_1:	http://example.org/Review245	‘‘Twix’’	‘‘7’’
μ_2:	http://example.org/Review682	‘‘Snickers’’	‘‘8’’
μ_3:	http://example.org/Review784	‘‘Bounty’’	‘‘3’’

Μια δυναμική προσέγγιση αντιστοιχίας θα μετατρέψει το προηγούμενο SPARQL ερώτημα σε ένα ισοδύναμο SQL που αναφέρεται στη σχέση Review. Η πλειοψηφία των μεθόδων επανεγγραφής εκφράζει το εισερχόμενο SPARQL ερώτημα σε όρους της SPARQL άλγεβρας, διασχίζει το προκύπτον δέντρο SPARQL τελεστών με συγκεκριμένη κατεύθυνση – συνήθως bottom-up και σπανιότερα top-down [58] – και παράγει ένα κατάλληλο SQL ερώτημα, τα αποτελέσματα του οποίου θα χρησιμοποιηθούν για την κατασκευή της SPARQL λύσης. Για το συγκεκριμένο SPARQL ερώτημα, θα δημιουργηθεί το ακόλουθο απλό SPARQL δέντρο τελεστών:



όπου π ο τελεστής της προβολής μεταβλητών και $\bowtie$ ο τελεστής της συνένωσης. Το αποτέλεσμα της επανεγγραφής του συγκεκριμένου ερωτήματος σύμφωνα με τα [57, 67, 84, 134] θα είναι το εξής SQL ερώτημα:

```
Q1: SELECT COALESCE(R1.nr, R2.nr), R1.product, R2.rating
    FROM (SELECT nr, product FROM Review) AS R1
    INNER JOIN (SELECT nr, rating FROM Review) AS R2
    ON (R1.nr = R2.nr OR R1.nr IS NULL OR R2.nr IS NULL)
```

Το Q_1 περιέχει δύο SQL υποερωτήματα και μια πράξη συνένωσης αυτών, οπότε γίνεται σαφές ότι όσο μεγαλύτερο το μέγεθος της σχέσης Review τόσο πιο αργή θα είναι η εκτέλεση του Q_1. Τα δύο υποερωτήματα αντιστοιχούν στα δύο πρότυπα τριάδων του SPARQL ερωτήματος, ενώ η συνθήκη συνένωσης μεταξύ των δύο προσομοιώνει τη σημασιολογία της συνένωσης SPARQL αντιστοιχιών (περισσότερες λεπτομέρειες θα αναφερθούν στην ενότητα 5.3). Η προσθήκη ενός τρίτου προτύπου τριάδας tp_3 θα προσέθετε ένα επιπλέον

[3]Περισσότερες λεπτομέρειες σχετικά με τη SPARQL ορολογία υπάρχουν στην ενότητα 5.3.

επίπεδο εμφώλευσης με το Q_1 να συνενώνεται με το SQL ερώτημα που θα αντιστοιχούσε στο tp_3. Το κοινό σημείο αυτών των αλγορίθμων είναι το γεγονός ότι θεωρούν ένα πρότυπο τριάδας ως ελάχιστη μονάδα και δημιουργούν ένα SQL ερώτημα για αυτό. Αντίθετα, ο αλγόριθμος που παρουσιάζεται σε αυτό το κεφάλαιο προτείνει την ομαδοποίηση των προτύπων τριάδων με βάση τις μεταβλητές που αυτά περιέχουν και τη δημιουργία ενός SQL ερωτήματος για κάθε ομάδα.

Το D2RQ, από την πλευρά του, ακολουθεί διαφορετική πορεία για το συγκεκριμένο SPARQL ερώτημα. Αρχικά, θα εκτελέσει το υποερώτημα R1: `SELECT nr, product FROM Review` που αντιστοιχεί στο πρώτο πρότυπο τριάδας και θα χρησιμοποιήσει τα αποτελέσματα για να θέσει συνθήκες στο υποερώτημα R2. Έτσι, για το δεδομένο στιγμιότυπο της σχέσης Review, εκτός του R1 θα εκτελεστούν και τα ακόλουθα SQL ερωτήματα:

$Q_{2\alpha}$:
```
SELECT nr, rating FROM Review
WHERE nr=245 AND product='Twix'
```
$Q_{2\beta}$:
```
SELECT nr, rating FROM Review
WHERE nr=682 AND product='Snickers'
```
$Q_{2\gamma}$:
```
SELECT nr, rating FROM Review
WHERE nr=784 AND product='Bounty'
```
$Q_{2\delta}$:
```
SELECT nr, rating FROM Review
WHERE nr=815 AND product='Mars'
```

Το D2RQ πραγματοποιεί με έμμεσο τρόπο συνένωση μεταξύ των υποερωτημάτων R1 και R2, εκτελώντας τόσα SQL ερωτήματα όσα και το μέγεθος των αποτελεσμάτων του πρώτου υποερωτήματος. Και σε αυτή την περίπτωση, γίνεται φανερό ότι η επίδοση του συστήματος χειροτερεύει όταν το μέγεθος της σχέσης Review είναι μεγάλο.

Το Ultrawrap αρχικά ορίζει έναν αριθμό όψεων στο σχήμα της ΒΔ, η οποία αντικατοπτρίζει την πλήρη αντιστοιχία μεταξύ ΒΔ και RDF γράφου. Για την τρέχουσα απλή αντιστοιχία, θα δημιουργηθεί η παρακάτω όψη $View_1$:

$View_1$:
```
SELECT ''http://example.org/Review''+nr AS s, nr AS s_id,
''http://example.org/isReviewFor'' AS p,product AS o,NULL AS o_id
FROM Review
UNION ALL
SELECT ''http://example.org/Review''+nr AS s, nr AS s_id,
''http://example.org/hasRating'' AS p, rating AS o, NULL AS o_id
FROM Review
UNION ALL
SELECT ''http://example.org/Review''+nr AS s, nr AS s_id,
''http://example.org/reviewDate'' AS p, date AS o, NULL AS o_id
FROM Review
```

Εκτός από τους όρους s, p και o, που αντιστοιχούν στους όρους μιας RDF τριάδας, η $View_1$ προβάλλει και τα γνωρίσματα s_id και o_id, τα οποία περιέχουν τις τιμές του πρωτεύοντος κλειδιού που συμμετέχει στη δημιουργία των s και o αντίστοιχα. Τα συγκεκριμένα γνωρίσματα χρησιμοποιούνται για τη συνένωση αντιγράφων της $View_1$ και διευκολύνουν τη βελτιστοποίηση του SQL ερωτήματος από τη μηχανή ερωτημάτων της ΒΔ. Η επανεγγραφή του τρέ-

χοντος SPARQL ερωτήματος είναι τετριμμένη και θα οδηγήσει στο ακόλουθο SQL ερώτημα που αναφέρεται στην όψη View_1:

```
Q3: SELECT v1.s, v1.o, v2.o
    FROM View1 AS v1 INNER JOIN View1 AS v2 ON v1.s_id = v2.s_id
    WHERE v1.p=‘‘http://example.org/isReviewFor’’ AND
    v2.p=‘‘http://example.org/hasRating’’
```

Παρατηρούμε ότι το Q_3, όπως και το Q_1, χρησιμοποιεί τόσες συνενώσεις όσα και τα πρότυπα τριάδων του SQL ερωτήματος, γεγονός που μπορεί να έχει αρνητικές επιπτώσεις στο χρόνο εκτέλεσής του, ιδιαίτερα αν η μηχανή ερωτημάτων της ΒΔ δεν καταφέρει να βρει το βέλτιστο σχέδιο αποτίμησης. Επιπρόσθετα, στο [173] δεν αναφέρεται κάποιος αλγόριθμος για την επανεγγραφή πιο σύνθετων SPARQL ερωτημάτων.

Σε αντίθεση με τις παραπάνω προσεγγίσεις, το τρέχον SPARQL ερώτημα θα μπορούσε να επανεγγραφεί στο επόμενο απλό SQL ερώτημα:

```
Q4: SELECT nr, product, rating
    FROM Review
    WHERE nr IS NOT NULL AND product IS NOT NULL AND
          rating IS NOT NULL
```

το οποίο αποφεύγει τη συνένωση της σχέσης Review με τον εαυτό της και επιτυγχάνει το ίδιο αποτέλεσμα με καθεμία από τις προαναφερθείσες προσεγγίσεις.

Παράδειγμα 5.1.2. Έστω η σχέση Review και το στιγμιότυπο αυτής από το παράδειγμα 5.1.1 και έστω ότι θεωρούμε το ακόλουθο απλό SPARQL ερώτημα:

```
PREFIX ex: <http://example.org>
SELECT ?review ?prod ?rating
WHERE {
        ?review ex:isReviewFor ?prod.
        OPTIONAL {?review ex:hasRating ?rating}
      }
```

το οποίο περιέχει ένα πρότυπο προαιρετικού γράφου και επιστρέφει το όνομα του προϊόντος και προαιρετικά, τη βαθμολογία του. Δεδομένης της αντιστοιχίας του παραδείγματος 5.1.1, η αναμενόμενη λύση αυτού του SPARQL ερωτήματος θα περιέχει 4 αντιστοιχίες μ_1, μ_2, μ_3 και μ_4 μία για κάθε πλειάδα του στιγμιοτύπου της Review:

	review	prod	rating
μ_1:	http://example.org/Review245	‘‘Twix’’	‘‘7’’
μ_2:	http://example.org/Review682	‘‘Snickers’’	‘‘8’’
μ_3:	http://example.org/Review784	‘‘Bounty’’	‘‘3’’
μ_3:	http://example.org/Review815	‘‘Mars’’	-

Το δέντρο SPARQL τελεστών θα έχει την ίδια μορφή με αυτό του παραδείγματος 5.1.1 με μόνη διαφορά την αντικατάσταση της πράξης της συνένωσης ($\bowtie$) με την πράξη της αριστερής εξωτερικής συνένωσης ($⟕$). Αντίστοιχη θα είναι

και η διαφορά στα SQL ερωτήματα που παράγονται π.χ. από τους αλγορίθμους [57, 67, 84, 134]:

```
Q5: SELECT COALESCE(R1.nr, R2.nr), R1.product, R2.rating
    FROM (SELECT nr, product FROM Review) AS R1
    LEFT OUTER JOIN (SELECT nr, rating FROM Review) AS R2
    ON (R1.nr = R2.nr OR R1.nr IS NULL OR R2.nr IS NULL)
```

Αντίθετα, ένα ισοδύναμο, περισσότερο αποδοτικό SQL ερώτημα θα ήταν το ακόλουθο:

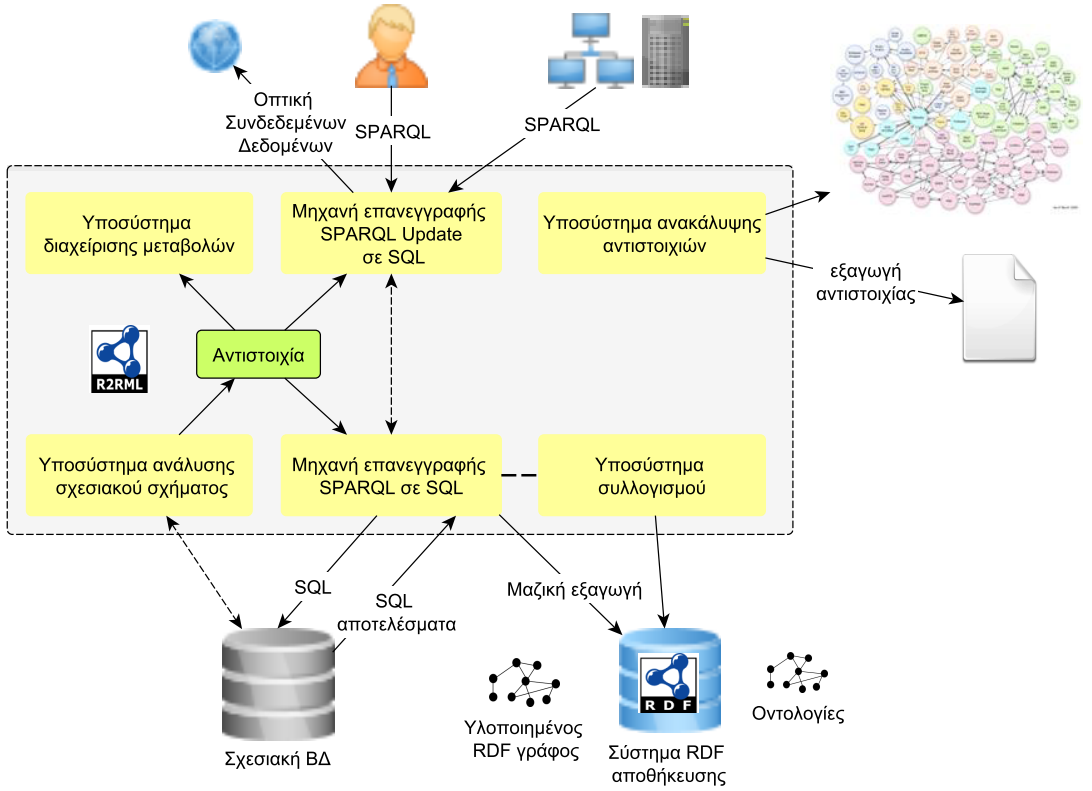
```
Q6: SELECT nr, product, rating
    FROM Review
    WHERE nr IS NOT NULL AND product IS NOT NULL
```

Στόχος του προτεινόμενου αλγορίθμου είναι η εύρεση ενός όσο το δυνατόν πιο επίπεδου SQL ερωτήματος (όπως τα Q_4, Q_6 των παραδειγμάτων 5.1.1 και 5.1.2 αντίστοιχα) που είναι σημασιολογικά ισοδύναμο με το αρχικό SPARQL ερώτημα, λαμβάνοντας υπόψη μια R2RML αντιστοιχία μεταξύ του σχήματος μιας σχεσιακής ΒΔ και ενός RDF γράφου. Αξίζει να σημειωθεί ότι όλοι οι αλγόριθμοι που αναφέρθηκαν στο εν λόγω παράδειγμα θεωρούν αντιστοιχίες εκφρασμένες σε διαφορετική μορφή, με την πλειονότητα αυτών να υποθέτει μια συνάρτηση αντιστοιχίας β που αντιστοιχεί κάθε όρο ενός προτύπου τριάδας σε ένα μοναδικό σχεσιακό γνώρισμα (N:1 αντιστοιχία), υπόθεση που δεν μπορεί να γενικευθεί για κάθε R2RML αντιστοιχία. Μάλιστα, σε αρκετές περιπτώσεις, δεν είναι προφανής η επέκταση ενός αλγορίθμου ώστε να λειτουργεί εξίσου αποδοτικά και για R2RML αντιστοιχίες. Παραδείγματος χάριν, όπως φάνηκε και στο παράδειγμα 5.1.1, το Ultrawrap προτείνει το σχηματισμό όψεων που μεταξύ άλλων προβάλλουν και το πρωτεύον κλειδί που χρησιμοποιείται στην παραγωγή IRIs από μια σχέση. Καθώς ο μηχανισμός παραγωγής IRI της R2RML δε θέτει κάποιον περιορισμό στο πλήθος των μοναδικών κλειδιών που μπορούν να χρησιμοποιηθούν σε ένα IRI, ο ορισμός των εν λόγω όψεων θα χρειαστεί να μεταβληθεί ώστε να προβάλλει περισσότερα γνωρίσματα, χωρίς όμως να μπορεί να καλύψει όλες τις περιπτώσεις και να εγγυηθεί αποδοτική λειτουργία για κάθε πιθανή R2RML αντιστοιχία.

Ο συγκεκριμένος αλγόριθμος έχει αρκετά κοινά στοιχεία με τις μεθόδους των [77, 162], οι οποίες επίσης προσπαθούν να παράγουν ένα επίπεδο SQL ερώτημα με χαμηλό βαθμό εμφώλευσης. Συγκεκριμένα, στο [77] υιοθετείται η χρήση ενός ενδιάμεσου μοντέλου που αναπαριστά ένα SQL ερώτημα και το οποίο εμπλουτίζεται κατά τη διάσχιση του δέντρου των SPARQL τελεστών, ώστε να αποφευχθεί η υπέρμετρη χρήση εμφωλευμένων υποερωτημάτων. Στο [162] αντίθετα, προτείνεται η ομαδοποίηση των προτύπων τριάδων με βάση το υποκείμενό τους, με στόχο τη μείωση των μη απαραίτητων συνενώσεων. Στον προτεινόμενο αλγόριθμο, προσπαθούμε να συνδυάσουμε τις ιδέες αυτές στο πλαίσιο R2RML αντιστοιχιών.

5.2 Μια γενική αρχιτεκτονική για την αντιστοιχία ΒΔ σε RDF γράφους

Στην παρούσα ενότητα, παρουσιάζουμε μια γενική αρχιτεκτονική, βασισμένη σε πρότυπες τεχνολογίες του W3C, για ένα ολοκληρωμένο σύστημα αντιστοιχίας μιας ΒΔ με έναν RDF γράφο, μέρος της οποίας έχει υλοποιηθεί στο σύστημα RDB4RDF. Ένα σύστημα που ακολουθεί την προτεινόμενη αρχιτεκτονική λειτουργεί ως διεπαφή μιας σχεσιακής ΒΔ με το Σημασιολογικό Ιστό και το Σύννεφο Συνδεδεμένων Δεδομένων, απελευθερώνοντας το περιεχόμενο της πρώτης σε μορφή κατάλληλη για επεξεργασία από εξωτερικές εφαρμογές, οι οποίες θα είναι σε θέση να συνδυάσουν αυτό το περιεχόμενο με άλλα δεδομένα και να πραγματοποιήσουν συλλογισμό, δημιουργώντας νέα γνώση.



Σχήμα 5.1: *Αρχιτεκτονική ολοκληρωμένου συστήματος αντιστοιχίας ΒΔ με RDF γράφους*

Τα βασικά συστατικά της προτεινόμενης αρχιτεκτονικής, η οποία απεικονίζεται στο σχήμα 5.1 και συνδυάζει κατά κάποιο τρόπο τα σχήματα 3.4 και 3.6 είναι τα ακόλουθα:

Υποσύστημα ανάλυσης σχεσιακού σχήματος. Το υποσύστημα ανάλυσης σχεσιακού σχήματος επικοινωνεί με τη σχεσιακή ΒΔ και ανακτά το σχήμα της, το οποίο καθιστά διαθέσιμο στο υπόλοιπο σύστημα και στις διεργασίες που χρειάζονται να έχουν γνώση αυτού. Η γνώση του σχεσιακού σχήματος κατ' αρχήν είναι απαραίτητη για τη διαδικασία SPARQL-σε-SQL επανεγγραφής, όπως αυτή θα περιγραφεί στην παράγραφο 5.3, για την εξαγωγή πληροφοριών όπως π.χ. ο τύπος δεδομένων μιας στήλης. Επίσης, το σχεσιακό σχήμα

είναι απαραίτητο για την κατασκευή της Άμεσης Αντιστοιχίας, μιας προκαθορισμένης αντιστοιχίας η οποία εξάγεται αυτόματα χωρίς να χρειάζεται είσοδο από κάποιο χρήστη.

Εκτός βέβαια από την Άμεση Αντιστοιχία, ένα ολοκληρωμένο σύστημα αντιστοιχίας πρέπει να είναι σε θέση να εξάγει αυτόματα μια περισσότερο σύνθετη οντολογική δομή αναλύοντας το σχήμα – και σπανιότερα τα δεδομένα – της ΒΔ, βασιζόμενο σε ένα σύνολο κανόνων αντίστροφης μηχανικής, παρόμοιων με αυτών που παρουσιάστηκαν στην παράγραφο 3.4.2.2. Ιδανικά, το σύνολο των κανόνων αυτών θα πρέπει να είναι παραμετροποιήσιμο και όχι απευθείας ενσωματωμένο στον κώδικα του συστήματος, προσφέροντας ευελιξία στον υπεύθυνο για την πραγματοποίηση της αντιστοιχίας. Οι κανόνες παραγωγής οντολογίας θα πρέπει να είναι εκφρασμένοι σε κάποιο καθιερωμένο πρότυπο, το οποίο θα διευκολύνει το διαμοιρασμό και την επαναχρησιμοποίησή τους. Ενδεικτικές τεχνολογίες που μπορούν να χρησιμοποιηθούν για αυτό το σκοπό είναι οι RIF[4] και SPIN[5]. Η έκφραση του σχεσιακού σχήματος ως οντολογία (παράγραφος 3.4.1), διαδικασία η οποία επίσης χρειάζεται γνώση του σχεσιακού σχήματος της ΒΔ, διευκολύνει σημαντικά την εφαρμογή των συγκεκριμένων τεχνολογιών και συνεπώς, θα πρέπει να υποστηρίζεται από το σύστημα.

Το υποσύστημα ανάλυσης σχεσιακού σχήματος είναι υπεύθυνο για την κατασκευή μιας R2RML αντιστοιχίας και για τις δύο προαναφερθείσες μεθόδους λειτουργίας, ήτοι την Άμεση Αντιστοιχία και την εφαρμογή κανόνων αντίστροφης μηχανικής.

Μηχανή επανεγγραφής SPARQL-σε-SQL. Η μηχανή επανεγγραφής SPARQL-σε-SQL είναι ο πυρήνας ενός συστήματος αντιστοιχίας, καθώς αναλαμβάνει την εφαρμογή της αντιστοιχίας στο δεδομένο στιγμιότυπο της σχεσιακής ΒΔ. Το συγκεκριμένο υποσύστημα είναι υπεύθυνο για τη λειτουργία δυναμικής πρόσβασης στα δεδομένα της ΒΔ, μεταφράζοντας ένα εισερχόμενο SPARQL ερώτημα σε ένα ισοδύναμο SQL και κατασκευάζοντας την SPARQL λύση από το αποτέλεσμα της εκτέλεσης του τελευταίου. Ο αλγόριθμος επανεγγραφής χρησιμοποιεί μια R2RML αντιστοιχία, η οποία μπορεί να έχει κατασκευαστεί χειροκίνητα από το χρήστη ή να έχει παραχθεί από το υποσύστημα ανάλυσης σχεσιακού σχήματος. Η αντιστοιχία που χρησιμοποιείται για την SPARQL-σε-SQL επανεγγραφή για μια συγκεκριμένη χρονική περίοδο λειτουργίας του συστήματος αναφέρεται ως *ενεργή αντιστοιχία*.

Ωστόσο, η μηχανή επανεγγραφής SPARQL-σε-SQL μπορεί να χρησιμοποιείται από το σύστημα και για τη στατική εφαρμογή μιας R2RML αντιστοιχίας, εξάγοντας το περιεχόμενο της ΒΔ σε μορφή RDF. Αυτό ισχύει αν αναλογιστούμε ότι η ανάκτηση του συνολικού RDF γράφου μπορεί να θεωρηθεί ως το αποτέλεσμα της εκτέλεσης ενός γενικού SPARQL ερωτήματος. Ο υλοποιημένος RDF γράφος μπορεί να αποθηκεύεται σε μορφή αρχείου ή, σε περίπτωση μεγάλου

[4] Η RIF (Rule Interchange Format) είναι πρότυπο του W3C (http://www.w3.org/TR/rif-overview/) και αποτελεί μια οικογένεια γλωσσών για την έκφραση και ανταλλαγή διαφόρων ειδών κανόνων.

[5] Η SPIN (SPARQL Inferencing Notation) είναι μια διαδεδομένη γλώσσα για την έκφραση περιορισμών και κανόνων που αναφέρονται σε RDF μοντέλα και οντολογίες (http://spinrdf.org/).

όγκου δεδομένων, σε ένα ξεχωριστό RDF σύστημα αποθήκευσης. Η αποδοτικότητα του αλγορίθμου επανεγγραφής καθορίζει την ταχύτητα και τις επιδόσεις του συνολικού συστήματος αντιστοιχίας και, ως εκ τούτου, χρειάζεται να κλιμακώνει με το μέγεθος της υποκείμενης ΒΔ.

Μηχανή επανεγγραφής SPARQL Update-σε-SQL. Η μηχανή επανεγγραφής SPARQL Update αιτημάτων σε SQL DML προτάσεις αναλαμβάνει την ενημέρωση του εικονικού RDF γράφου που ορίζεται από την ενεργή αντιστοιχία του συστήματος. Το συγκεκριμένο υποσύστημα αποτελεί απαραίτητο συστατικό ενός συστήματος αντιστοιχίας που φιλοδοξεί να λειτουργεί ως ολοκληρωμένη διεπαφή μεταξύ μιας σχεσιακής ΒΔ και εφαρμογών Σημασιολογικού Ιστού. Είναι λογικό ότι η μηχανή επανεγγραφής SPARQL Update αιτημάτων θα μοιράζεται αρκετά χαρακτηριστικά και αλγορίθμους με τη μηχανή SPARQL επανεγγραφής, οπότε είναι πιθανό τα δύο αυτά υποσυστήματα να είναι ενσωματωμένα σε έναν κεντρικό πυρήνα επεξεργασίας ερωτημάτων.

Υποσύστημα συλλογισμού. Ένα σύστημα αντιστοιχίας πρακτικά ορίζει μια εικονική βάση γνώσης, στην οποία τα δεδομένα μιας σχεσιακής ΒΔ ερμηνεύονται με βάση γνωστές εξωτερικές οντολογίες ή μια οντολογία που έχει εξαχθεί από το σχήμα της ΒΔ (βλέπε υποσύστημα ανάλυσης σχεσιακού σχήματος). Όπως κάθε βάση γνώσης χρειάζεται διαδικασίες συλλογισμού προκειμένου να συμπεράνει νέα γεγονότα και γνώση από κατηγορηματικά δηλωθέντα γεγονότα, με τον ίδιο τρόπο και ένα σύστημα αντιστοιχίας πρέπει να διαθέτει ένα υποσύστημα συλλογισμού για την εξαγωγή νέας γνώσης.

Στην περίπτωση που το σύστημα αντιστοιχίας λειτουργεί στατικά, υλοποιώντας τον RDF γράφο και αποθηκεύοντάς τον σε ένα εξωτερικό σύστημα RDF αποθήκευσης, ο συλλογισμός αποτελεί μια διαδικασία που μπορεί να πραγματοποιηθεί από ώριμα εξωτερικά εργαλεία, αλλά συχνά και από το ίδιο το σύστημα αποθήκευσης που μπορεί να έχει τέτοιες δυνατότητες.

Αντίθετα, στην περίπτωση που το σύστημα αντιστοιχίας λειτουργεί δυναμικά και ο RDF γράφος που ορίζεται από την ενεργή αντιστοιχία είναι εικονικός, το πρόβλημα της πραγματοποίησης συλλογισμού θέτει σημαντικές προκλήσεις. Σε μια τέτοια περίπτωση, τα αξιώματα κάθε RDFS ή OWL οντολογίας, της οποίας οι όροι συμμετέχουν στον εικονικό RDF γράφο θα πρέπει να λαμβάνονται υπόψη κατά τη διαδικασία επανεγγραφής ενός SPARQL ερωτήματος, με αποτέλεσμα το τελικό SQL ερώτημα να είναι τέτοιο ώστε να ανακτά και υπονοούμενα γεγονότα. Με άλλα λόγια, στην περίπτωση δυναμικής λειτουργίας του συστήματος, το υποσύστημα συλλογισμού θα πρέπει να αποτελεί μέρος της μηχανής SPARQL-σε-SQL επανεγγραφής, επεκτείνοντας τη λειτουργικότητα της τελευταίας. Αξίζει να σημειωθεί ότι ο αλγόριθμος συλλογισμού θα πρέπει να είναι προσαρμοζόμενος στην εκφραστικότητα της αναφερόμενης οντολογίας, καθώς όσο πιο εκφραστική είναι η τελευταία, τόσο πιο σύνθετος θα πρέπει να είναι ο αλγόριθμος επανεγγραφής.

Υποσύστημα ανακάλυψης αντιστοιχιών. Εκτός από τη δυνατότητα ενός συστήματος αντιστοιχίας να παράγει αυτόματα μια νέα οντολογία αναλύοντας το σχήμα μιας σχεσιακής ΒΔ (και συνεπώς, και μια αντιστοιχία της πρώτης με όρους της δεύτερης), ακόμα μεγαλύτερη αξία έχει η ανακάλυψη αντιστοιχιών με ήδη υπάρχουσες εξωτερικές οντολογίες. Η ερμηνεία των περιεχομένων μιας

ΒΔ σε όρους γνωστών και ευρέως χρησιμοποιούμενων λεξιλογίων οδηγεί στην παραγωγή πραγματικά Συνδεδεμένων Δεδομένων και ενισχύει τη σημασιολογική διαλειτουργικότητα διακριτών ΒΔ και συστημάτων γενικότερα. Το υποσύστημα ανακάλυψης αντιστοιχιών θα πρέπει να επικοινωνεί με σημασιολογικές μηχανές αναζήτησης για την ανάκτηση σχετικών οντολογιών ή εναλλακτικά να διατηρεί κάποιο μητρώο δημοφιλών οντολογιών, από τις οποίες θα εντοπίζει τις κλάσεις, ιδιότητες και οντότητες που ταιριάζουν καλύτερα στα στοιχεία της ΒΔ, βασιζόμενο σε τεχνικές λεξικολογικής και δομικής ομοιότητας. Η τελική απόφαση για την ορθότητα των προτεινόμενων αντιστοιχιών θα λαμβάνεται από το χρήστη που είναι υπεύθυνος για τον ορισμό της ενεργής αντιστοιχίας, οδηγώντας σε ένα μοντέλο ημι-αυτόματης λειτουργίας, το οποίο είναι συνήθως και το πιο αποδοτικό, συνδυάζοντας τη διευκόλυνση του ανθρώπινου χρήστη και τη σημασιολογική ορθότητα, η οποία δύσκολα επιτυγχάνεται από αυστηρά αυτόματες διαδικασίες.

Υποσύστημα διαχείρισης μεταβολών. Το υποσύστημα διαχείρισης μεταβολών συνεισφέρει στην αποφυγή επανάληψης υπολογισμού του αποτελέσματος μιας αντιστοιχίας όταν παρατηρηθεί κάποια μεταβολή είτε σε επίπεδο δεδομένων ή σχήματος ΒΔ είτε σε επίπεδο αντιστοιχίας, αυξάνοντας έτσι την απόδοσή του συνολικού συστήματος. Μεταβολές στο επίπεδο δεδομένων της ΒΔ καθώς και στην ενεργή αντιστοιχία του συστήματος επηρεάζουν μονάχα τη στατική λειτουργία ενός συστήματος αντιστοιχίας, καθώς εξ ορισμού η δυναμική λειτουργία λαμβάνει πάντα υπόψη της το τρέχον στιγμιότυπο της σχεσιακής ΒΔ. Στην περίπτωση στατικής εφαρμογής της αντιστοιχίας, το υποσύστημα διαχείρισης μεταβολών πρέπει να είναι σε θέση να ανιχνεύει τις διαφορές μεταξύ δύο στιγμιοτύπων της ΒΔ καθώς και πιθανές διαφορές στην ενεργή αντιστοιχία του συστήματος και να υπολογίζει τη διαφορική επίδραση που έχουν αυτές στον παραγόμενο RDF γράφο, αποφεύγοντας τον επανυπολογισμό του εκ του μηδενός.

Η διαχείριση και αντιμετώπιση αλλαγών στο σχήμα της ΒΔ είναι ένα πιο σύνθετο πρόβλημα που επηρεάζει και τους δύο τρόπους λειτουργίας, δεδομένου του γεγονότος ότι τέτοιες αλλαγές μπορούν να καταστήσουν μια αντιστοιχία μη έγκυρη. Σε μια τέτοια περίπτωση υπάρχουν αρκετές στρατηγικές αντιμετώπισης, καθώς το υποσύστημα διαχείρισης μεταβολών μπορεί να αγνοήσει τα μη έγκυρα τμήματα της αντιστοιχίας, να τα προσαρμόσει κατάλληλα στο νέο σχεσιακό σχήμα ή ακόμα και να ανακαλύψει νέες αντιστοιχίες για τα νέα στοιχεία του σχήματος.

Γραφικές και προγραμματιστικές διεπαφές. Ένα σύστημα αντιστοιχίας μπορεί να προορίζεται για χρήση τόσο από ανθρώπινους χρήστες όσο και από εφαρμογές, δημιουργώντας την απαίτηση ύπαρξης γραφικών αλλά και προγραμματιστικών διεπαφών. Ένα σύστημα αντιστοιχίας μπορεί να παρέχει γραφικές διεπαφές στους χρήστες του για τον ορισμό μιας αντιστοιχίας και την επιλογή διαφορετικών μορφών λειτουργίας του συστήματος, μεταξύ των οποίων η εκτέλεση SPARQL ερωτημάτων ή η πλοήγηση στα περιεχόμενα μιας σχεσιακής ΒΔ μέσω μιας μιας οπτικής Συνδεδεμένων Δεδομένων. Παράλληλα, το σύστημα αντιστοιχίας θα πρέπει να παρέχει τη δυνατότητα εκτέλεσης SPARQL ερωτημάτων μέσω HTTP αιτημάτων, υλοποιώντας το SPARQL πρωτόκολλο. Η συγκεκριμένη δυνατότητα επιτρέπει σε απομακρυσμένες εφαρμογές την

ανάκτηση δεδομένων από τη σχεσιακή ΒΔ και τη χρήση αυτών με διάφορους τρόπους, συνδυάζοντάς τα με άλλα δεδομένα ή παρέχοντας διάφορες μορφές απεικόνισής τους.

Βοηθητικά συστήματα RDF αποθήκευσης. Κατά τη λειτουργία του, ένα σύστημα αντιστοιχίας μπορεί να δημιουργήσει RDF δεδομένα και οντολογίες ή να χρησιμοποιήσει όρους από εξωτερικές οντολογίες κατά τον ορισμό μιας αντιστοιχίας. Συνεπώς, το σύστημα αντιστοιχίας χρειάζεται να επικοινωνεί με ένα ή περισσότερα συστήματα RDF αποθήκευσης, τα οποία θα διαχειρίζονται και, προαιρετικά, θα πραγματοποιούν συλλογισμό στον παραγόμενο RDF γράφο σε περιπτώσεις στατικής εφαρμογής της αντιστοιχίας. Επίσης, τα βοηθητικά αυτά συστήματα μπορούν να αποθηκεύουν εξωτερικές οντολογίες-στόχους της αντιστοιχίας και πιθανώς, να τηρούν και έναν κατάλογο R2RML αντιστοιχιών, επίσης εκφρασμένο σε μορφή RDF.

Μια βασική υλοποίηση ενός μέρους της προτεινόμενης αρχιτεκτονικής πραγματοποιήθηκε στο σύστημα RDB4RDF, το οποίο είναι υλοποιημένο σε Java. Η τρέχουσα πρώιμη έκδοση του RDB4RDF[6] περιλαμβάνει μονάχα μια μηχανή επανεγγραφής SPARQL-σε-SQL, η οποία υποστηρίζει R2RML αντιστοιχίες καθώς και ένα τελικό σημείο SPARQL για την υποβολή SPARQL ερωτημάτων σε μια σχεσιακή ΒΔ μέσω HTTP. Ο κεντρικός αλγόριθμος της μηχανής επανεγγραφής περιγράφεται αναλυτικά στην ενότητα 5.3.

5.3 Αλγόριθμος επανεγγραφής

Στην τρέχουσα ενότητα, παρουσιάζεται ο αλγόριθμος επανεγγραφής SPARQL ερωτημάτων, ο οποίος αποτελεί τον πυρήνα του RDB4RDF συστήματος. Τα βασικά στάδια του αλγορίθμου απεικονίζονται στο σχήμα 5.2 και είναι τα εξής:

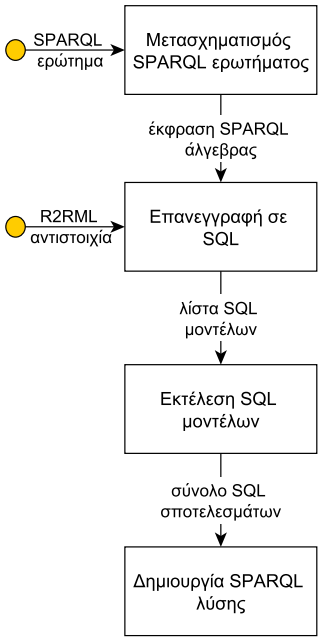
1. Μετασχηματισμός εισερχόμενου SPARQL ερωτήματος σε όρους μιας εκτεταμένης SPARQL άλγεβρας (παράγραφος 5.3.2)
2. Επανεγγραφή SPARQL ερωτήματος σε λίστα ισοδύναμων SQL ερωτημάτων (παράγραφος 5.3.3)
3. Εκτέλεση SQL ερωτήματος στη συνδεδεμένη ΒΔ
4. Μετατροπή SQL αποτελεσμάτων σε SPARQL λύση (παράγραφος 5.3.4)

Πριν δούμε αναλυτικά καθένα από αυτά τα βήματα, προχωρούμε σε μια περισσότερο εκτεταμένη επισκόπηση των SPARQL και R2RML προτύπων σε σχέση με τις αντίστοιχες εισαγωγικές παραγράφους του κεφαλαίου 2.

5.3.1 Προκαταρκτικά

Αρχικά, αναφερόμαστε στη σημασιολογία της SPARQL, δανειζόμενοι σχετικούς ορισμούς από την επίσημη προδιαγραφή της SPARQL [93] και το [152]. Έστω I, B, L και V τα αμοιβαίως αποκλειόμενα σύνολα των IRIs, κενών κόμβων, RDF λεκτικών και μεταβλητών αντίστοιχα.

[6] https://www.assembla.com/code/rdb4rdf/subversion/nodes



Σχήμα 5.2: *Αλγόριθμος εκτέλεσης SPARQL ερωτημάτων*

Ορισμός 5.3.1. (RDF τριάδα και RDF γράφος) Μια *RDF τριάδα* (RDF triple) είναι μια πλειάδα $\langle s, p, o\rangle \in (I \cup B) \times I \times (I \cup B \cup L)$. Ένα σύνολο RDF τριάδων ονομάζεται *RDF γράφος* (RDF graph).

Ορισμός 5.3.2. (Σύνολο δεδομένων) Ένα *σύνολο δεδομένων* (dataset) είναι μια δομή που αποτελείται από:

1. έναν RDF γράφο που ονομάζεται *προκαθορισμένος* (default graph) και
2. ένα σύνολο ζευγών $(name, graph)$, όπου *name* ένα IRI και *graph* ένας RDF γράφος. Καθένα από αυτά τα ζεύγη είναι γνωστό και ως *ονομαστικός γράφος* (named graph).

Ο προκαθορισμένος γράφος δε χαρακτηρίζεται από κάποιο IRI και αποτελεί το γράφο πάνω στον οποίο εκτελείται ένα SPARQL ερώτημα όταν δεν έχει καθοριστεί κατηγορηματικά ο RDF γράφος που θα χρησιμοποιηθεί. Η έννοια του συνόλου δεδομένων είναι στενά συνυφασμένη με την έννοια της RDF τετράδας.

Ορισμός 5.3.3. (RDF τετράδα) Μια *RDF τετράδα* (RDF quad) είναι μια πλειάδα $\langle s, p, o, g\rangle \in (I \cup B) \times I \times (I \cup B \cup L) \times (I \cup \{DG\})$, όπου DG ο προκαθορισμένος γράφος.

Ορισμός 5.3.4. (Πρότυπο τριάδας και πρότυπο τετράδας) Ένα *πρότυπο τριάδας* (triple pattern) tp είναι μια πλειάδα $\langle sp, pp, op\rangle \in (I \cup B \cup L \cup V) \times (I \cup V) \times (I \cup B \cup L \cup V)$. Αντίστοιχα, ένα *πρότυπο τετράδας* (quad pattern) qp είναι μια πλειάδα $\langle sp, pp, op, gp\rangle \in (I \cup B \cup L \cup V) \times (I \cup V) \times (I \cup B \cup L \cup V) \times (I \cup \{DG\} \cup V)$. Ένα σύνολο προτύπων τριάδας ονομάζεται *πρότυπο βασικού γράφου* (basic graph pattern ή BGP). Ονομάζουμε ένα σύνολο προτύπων τετράδας *πρότυπο βασικού συνόλου δεδομένων* (basic dataset pattern).

Οι έννοιες του προτύπου τριάδας και του προτύπου τετράδας επεκτείνουν τις αντίστοιχες έννοιες της τριάδας και της τετράδας και αποτελούν βασικά στοιχεία της SPARQL. Αξίζει να σημειωθεί ότι ο ορισμός 5.3.4 επιτρέπει την παρουσία λεκτικών ως υποκειμένων σε πρότυπα τριάδων και τετράδων, σε αντίθεση με τον τρέχοντα ορισμό του RDF μοντέλου. Με βάση το πρότυπο τριάδας και το πρότυπο βασικού γράφου, η SPARQL ορίζει σύνθετα πρότυπα γράφου (αντίστοιχα, σύνθετα πρότυπα συνόλου δεδομένων, αν ακολουθείται η θεώρηση τετράδων) χρησιμοποιώντας μεταξύ άλλων τους SPARQL τελεστές `AND` (συνένωση), `OPT` (προαιρετική συνένωση), `UNION` (ένωση) και `FILTER` (φιλτράρισμα).

Ορισμός 5.3.5. (Αντιστοιχία και SPARQL λύση) Έστω $\mu : V \rightarrow I \cup B \cup L$ μια μερική συνάρτηση η οποία αναθέτει RDF όρους σε μεταβλητές. Η συνάρτηση μ ονομάζεται *αντιστοιχία* (mapping) και το πεδίο ορισμού της $dom(\mu)$ είναι το υποσύνολο του V στο οποίο αυτή έχει οριστεί. Ένα σύνολο αντιστοιχιών Ω ονομάζεται *SPARQL λύση* (SPARQL solution).

Επίσης, για ένα πρότυπο τετράδας qp, συμβολίζουμε με $\mu(qp)$ την RDF τετράδα που λαμβάνεται με εφαρμογή της μ στις μεταβλητές του qp.

Ορισμός 5.3.6. (Συμβατές αντιστοιχίες) Δύο αντιστοιχίες μ_1, μ_2 ονομάζονται *συμβατές* (compatible), όταν $\mu_1(x) = \mu_2(x), \forall x \in dom(\mu_1) \cap dom(\mu_2)$.

Με άλλα λόγια, δύο αντιστοιχίες είναι συμβατές όταν η μ_1 μπορεί να «επεκταθεί» με τη μ_2 δίνοντας μία καινούρια αντιστοιχία και το αντίστροφο. Εξ ορισμού, δύο αντιστοιχίες με ξένα πεδία ορισμού θα είναι συμβατές. Οι γνωστές συνολοθεωρητικές πράξεις μεταξύ δύο συνόλων αντιστοιχιών Ω_1 και Ω_2 μπορούν να οριστούν ως εξής:

$\Omega_1 \bowtie \Omega_2 = \{\mu_1 \cup \mu_2 | \mu_1 \in \Omega_1, \mu_2 \in \Omega_2 \text{ και } \mu_1, \mu_2 \text{ είναι συμβατές}\}$ **(Συνένωση)**
$\Omega_1 \cup \Omega_2 = \{\mu | \mu_1 \in \Omega_1 \text{ ή } \mu_2 \in \Omega_2\}$ **(Ένωση)**
$\Omega_1 \setminus \Omega_2 = \{\mu \in \Omega_1 | \forall \mu' \in \Omega_2, \mu \text{ και } \mu' \text{ μη συμβατές ή } dom(\mu_1) \cap dom(\mu_2) = \emptyset\}$ **(Διαφορά)**
$\Omega_1 \bowtie\!\!\!\!- \Omega_2 = (\Omega_1 \bowtie \Omega_2) \cup (\Omega_1 \setminus \Omega_2)$ **(Αριστερή εξωτερική συνένωση)**

Η *αποτίμηση* (evaluation) ενός SPARQL ερωτήματος σε ένα σύνολο δεδομένων D ορίζεται ως μια συνάρτηση $[\![\cdot]\!]_D$, η οποία δέχεται ως είσοδο ένα πρότυπο τετράδας ή, στη γενική περίπτωση, ένα SPARQL ερώτημα και επιστρέφει ένα σύνολο αντιστοιχιών. Σύμφωνα με το [152], η συνάρτηση αποτίμησης για ένα πρότυπο γράφου gp ορίζεται αναδρομικά και ισχύουν τα παρακάτω για τα πιο διαδεδομένα πρότυπα γράφου:

$[\![gp]\!]_D = \{\mu | dom(\mu) = var(gp) \text{ και } \mu(gp) \in D\}$, όταν gp είναι πρότυπο τετράδας
$[\![gp_1 \text{ AND } gp_2]\!]_D = [\![gp_1]\!]_D \bowtie [\![gp_2]\!]_D$
$[\![gp_1 \text{ OPT } gp_2]\!]_D = [\![gp_1]\!]_D \bowtie\!\!\!\!- [\![gp_2]\!]_D$
$[\![gp_1 \text{ UNION } gp_2]\!]_D = [\![gp_1]\!]_D \cup [\![gp_2]\!]_D$
$[\![gp \text{ FILTER } expr]\!]_D = \{\mu \in [\![gp]\!]_D \wedge \mu \vDash expr\}$

όπου $var(qp)$ το σύνολο των μεταβλητών του προτύπου τετράδας qp και ο συμβολισμός $\mu \vDash expr$ δηλώνει ότι η αντιστοιχία μ ικανοποιεί τη λογική έκφραση

expr. Το σύνολο των SPARQL τελεστών και η σημασιολογία τους ορίζεται στην επίσημη προδιαγραφή της SPARQL [93].

Όσον αφορά στην R2RML, επιχειρούμε μια περισσότερο τυπική παρουσίασή της συγκριτικά με αυτήν που πραγματοποιήθηκε στην παράγραφο 2.2.4. Όπως αναφέρθηκε και εκεί, η R2RML ορίζει μια αντιστοιχία μεταξύ του σχήματος μιας σχεσιακής ΒΔ και ενός RDF συνόλου δεδομένων. Η ίδια η αντιστοιχία εκφράζεται σε μορφή RDF και αποτελεί το *γράφο αντιστοιχίας* (mapping graph). Ένας γράφος αντιστοιχίας αποτελείται από μία ή περισσότερες *αντιστοιχίες τριάδων* (triples maps), οι οποίες αποτελούν τη βασική μονάδα της R2RML. Μια αντιστοιχία τριάδων[7] καθορίζει το σύνολο των RDF τετράδων που θα παραχθεί από ένα μέρος του στιγμιοτύπου ενός σχεσιακού σχήματος. Συνεπώς, μια αντιστοιχία τριάδων μπορεί να ιδωθεί ως μια συνάρτηση $triplesMap : DB_I \rightarrow 2^{RDF\text{-}Q}$, όπου DB_I το στιγμιότυπο ενός σχεσιακού σχήματος, $RDF\text{-}Q = (I \cup B) \times I \times (I \cup B \cup L) \times (I \cup \{DG\})$ το σύνολο όλων των RDF τετράδων και $2^{RDF\text{-}Q}$ το δυναμοσύνολο αυτού. Εντούτοις, μια τέτοια θεώρηση δεν εξυπηρετεί τους σκοπούς της συγκεκριμένης παρουσίασης, καθώς επιθυμούμε να αναφερθούμε και στα επιμέρους συστατικά στοιχεία μιας αντιστοιχίας τριάδων. Συνεπώς, υιοθετούμε μια συνολοθεωρητική προσέγγιση για τον ορισμό ενός R2RML γράφου αντιστοιχίας και των συστατικών αυτού, αρχίζοντας από τα βασικά συστατικά του. Σημειώνουμε ότι η R2RML χρησιμοποιεί την ορολογία του SQL προτύπου, χρησιμοποιώντας τους όρους πίνακας και στήλη για να αναφερθεί στις έννοιες της σχέσης και του γνωρίσματος του σχεσιακού μοντέλου. Στο τρέχον κεφάλαιο, κάνουμε εναλλακτική χρήση και των δύο ορολογιών.

Ορισμός 5.3.7. (Λογικός πίνακας) Λογικός πίνακας (logical table) ονομάζεται:

α) ένας πίνακας ή όψη της θεωρούμενης ΒΔ, οπότε ο λογικός πίνακας ονομάζεται *βασικός πίνακας ή όψη* (base table or view) ή

β) ένα SQL ερώτημα επί της θεωρούμενης ΒΔ, οπότε ο λογικός πίνακας ονομάζεται R2RML όψη (R2RML view).

Ο λογικός πίνακας μιας αντιστοιχίας τριάδας καθορίζει το υποσύνολο της ΒΔ από το οποίο θα προκύψουν οι RDF τετράδες. Η συγκεκριμένη μορφή που θα έχουν αυτές οι τετράδες καθορίζεται από τις *αντιστοιχίες όρου* (term maps) που περιέχει η αντιστοιχία τριάδων.

Ορισμός 5.3.8. (Αντιστοιχία όρου) Μια *αντιστοιχία όρου* (term map) είναι μια πλειάδα της μορφής $\langle termGen, type, invExpr \rangle$, όπου $termGen : DB_I \rightarrow I \cup B \cup L \cup \{DG\}$ μια συνάρτηση παραγωγής RDF όρων από το στιγμιότυπο της ΒΔ, $type = \{\text{IRI, BlankNode, Literal}\}$ το είδος του RDF όρου που παράγεται από την αντιστοιχία όρου και $invExpr$ μια έκφραση που αναφέρει τη σχέση που συνδέει μια στήλη μιας R2RML όψης με μια στήλη ενός βασικού πίνακα της ΒΔ. Η συνάρτηση $termGen$ μπορεί να έχει μία από τις παρακάτω 3 μορφές:

α) $termGen : DB_I \rightarrow \{t\}$ όπου t ένας σταθερός RDF όρος, οπότε η αντιστοιχία όρων ονομάζεται *σταθερής τιμής* (constant-valued),

[7] Η ονομασία είναι παραπλανητική!

β) $termGen : att \rightarrow I \cup B \cup L$, όπου att το σύνολο των γνωρισμάτων του σχεσιακού σχήματος, οπότε η αντιστοιχία όρων ονομάζεται *αντιστοιχία τιμής από στήλη* (column-valued) και

γ) $termGen : att^n \rightarrow I \cup B \cup L$, όπου n θετικός ακέραιος, οπότε έχουμε την περίπτωση *αντιστοιχίας τιμής από πρότυπο* (template-valued).

Για απλοποίηση της παρουσίασης, θεωρούμε ότι κάθε μία από τις παραπάνω μορφές της *termGen* μπορεί να υποκατασταθεί με τις συμβολοσειρές *constant*, *column* και *template* αντίστοιχα. Οι συμβολοσειρές αυτές καθορίζουν πλήρως το μηχανισμό παραγωγής ενός όρου, σύμφωνα με τη σημασιολογία της R2RML. Πιο συγκεκριμένα, η *constant* δηλώνει το σταθερό παραγόμενο όρο, η *column* δηλώνει τη στήλη από την οποία θα προέλθει ο όρος και η *template* ένα οσοδήποτε σύνθετο πρότυπο το οποίο μπορεί να συνδυάζει τιμές μίας ή περισσότερων στηλών.

Κάθε αντιστοιχία τριάδων πρέπει να διαθέτει τουλάχιστον 3 αντιστοιχίες όρων, για καθεμία από τις 3 πρώτες θέσεις μιας RDF τετράδας (απουσία αντιστοιχίας όρου για το γράφο της τετράδας υπονοεί τον προκαθορισμένο γράφο). Οι αντιστοιχίες όρων διακρίνονται, ανάλογα με τη θέση του όρου που παράγουν σε μια RDF τετράδα, σε *αντιστοιχίες υποκειμένου* (subject maps), *αντιστοιχίες κατηγορήματος* (predicate maps), *αντιστοιχίες αντικειμένου* (object maps) και *αντιστοιχίες γράφου* (graph maps).

Ορισμός 5.3.9. (Αντιστοιχία υποκειμένου) Μια *αντιστοιχία υποκειμένου* (subject map) αποτελεί την εξειδίκευση $\langle subjGen, type_s, invExpr, graphMaps, classIRI \rangle$ μιας αντιστοιχίας όρων με $subjGen : DB_I \rightarrow I \cup B$, $type_s = \{\text{IRI, BlankNode}\}$, $classIRI$ το IRI μιας οντολογικής κλάσης και $graphMaps$ ένα σύνολο αντιστοιχιών γράφου (βλέπε ορισμό 5.3.12).

Ορισμός 5.3.10. (Αντιστοιχία κατηγορήματος) Μια *αντιστοιχία κατηγορήματος* (predicate map) αποτελεί την εξειδίκευση $\langle predGen, type_p, invExpr \rangle$ μιας αντιστοιχίας όρων με $predGen : DB_I \rightarrow I$ και $type_p = \{\text{IRI}\}$.

Ορισμός 5.3.11. (Αντιστοιχία αντικειμένου) Μια *αντιστοιχία αντικειμένου* (object map) αποτελεί την εξειδίκευση $\langle objGen, type, invExpr, language, datatype \rangle$ μιας αντιστοιχίας όρων με $objGen : DB_I \rightarrow I \cup B \cup L$, $language$ η γλώσσα και $datatype$ ο RDF τύπος δεδομένων του παραγόμενου λεκτικού. Τα δύο τελευταία στοιχεία υπάρχουν μόνο αν ο παραγόμενος όρος είναι ένα RDF λεκτικό.

Ορισμός 5.3.12. (Αντιστοιχία γράφου) Μια *αντιστοιχία γράφου* (graph map) αποτελεί την εξειδίκευση $\langle graphGen, type_g, invExpr \rangle$ μιας αντιστοιχίας όρων με $graphGen : DB_I \rightarrow I \cup \{DG\}$ και $type_g = \{\text{IRI}\}$. Η αντιστοιχία γράφου $\langle DG, IRI, true \rangle$ ονομάζεται *αντιστοιχία προκαθορισμένου γράφου*.

Εκτός των παραπάνω, η R2RML ορίζει και ένα άλλο είδος αντιστοιχίας, την αναφέρουσα αντιστοιχία αντικειμένου, η οποία επαναχρησιμοποιεί μια αντιστοιχία υποκειμένου για την παραγωγή ενός RDF όρου.

Ορισμός 5.3.13. (Αναφέρουσα αντιστοιχία αντικειμένου) Μια *αναφέρουσα αντιστοιχία αντικειμένου* (referencing object map) είναι μια πλειάδα της μορφής $\langle parentMap, joins \rangle$, όπου $parentMap$ μια αντιστοιχία τριάδων-γονέας και $joins$ ένα σύνολο *συνενώσεων*. Μια συνένωση ορίζεται ως ένα ζεύγος στηλών $(child, parent)$.

Ορισμός 5.3.14. (Αντιστοιχία κατηγορήματος-αντικειμένου) Μια *αντιστοιχία κατηγορήματος-αντικειμένου* (predicate-object map) είναι μια πλειάδα της μορφής $\langle predMaps, objMaps, graphMaps \rangle$, όπου $predMaps \neq \emptyset$ ένα σύνολο αντιστοιχιών κατηγορήματος, $objMaps \neq \emptyset$ ένα σύνολο αντιστοιχιών αντικειμένου ή αναφερουσών αντιστοιχιών αντικειμένου και $graphMaps$ ένα σύνολο αντιστοιχιών γράφου.

Ορισμός 5.3.15. (Αντιστοιχία τριάδων) Μια *αντιστοιχία τριάδων* (triples map) είναι μια πλειάδα της μορφής $\langle table, subjMap, predObjMaps \rangle$, όπου $table$ ένας λογικός πίνακας, $subjMap$ μια αντιστοιχία υποκειμένου και $predObjMaps \neq \emptyset$ ένα σύνολο αντιστοιχιών κατηγορήματος-αντικειμένου.

Ορισμός 5.3.16. (R2RML γράφος αντιστοιχίας) Ένα σύνολο αντιστοιχιών τριάδων ονομάζεται *R2RML γράφος αντιστοιχίας*.

Ένα απλό παράδειγμα ενός R2RML γράφου αντιστοιχίας σε Turtle σύνταξη δόθηκε στην παράγραφο 2.2.4, ενώ στο σχήμα 2.3 φαίνεται το αποτέλεσμα της εφαρμογής του στο στιγμιότυπο μιας ΒΔ.

Στη συνέχεια, ορίζουμε την έννοια της *συμβατότητας* μεταξύ μιας αντιστοιχίας τριάδων και ενός προτύπου τετράδας. Η έννοια της συμβατότητας βασίζεται στους ορισμούς των R2RML αντιστοιχιών που δόθηκαν προηγουμένως και παίζει κεντρικό ρόλο στη διαδικασία επανεγγραφής του SPARQL ερωτήματος, όπως αυτή παρουσιάζεται στην παράγραφο 5.3.3.

Προκειμένου να απλοποιήσουμε κάπως τον ορισμό της συμβατότητας αντιστοιχίας τριάδων και προτύπων τετράδας, ορίζουμε τις έννοιες της *κανονικοποιημένης αντιστοιχίας τριάδων* και του *κανονικοποιημένου R2RML γράφου αντιστοιχίας*, οι οποίες αποτελούν συντακτικά ισοδύναμες εκφράσεις μιας αντιστοιχίας τριάδων και ενός R2RML γράφου αντιστοιχίας.

Ορισμός 5.3.17. (Κανονικοποιημένη αντιστοιχία τριάδων) Μια αντιστοιχία τριάδων $triplesMap = \langle table, subjMap, predObjMaps \rangle$ *χρήζει κανονικοποίησης*, αν η αντιστοιχία υποκειμένου της διαθέτει IRI οντολογικής κλάσης, δηλαδή αν $subjMap.classIRI \neq null$, όπου με το συμβολισμό '.' αναφερόμαστε σε ένα συστατικό στοιχείο μιας σύνθετης δομής.
Έστω $classPredObj = \langle classPred, classObj, true \rangle$ μια νέα αντιστοιχία κατηγορήματος-αντικειμένου με $classPred = \{\langle \texttt{rdf:type}, IRI, true \rangle\}$ και $classObj = \{\langle subjMap.classIRI, IRI, true, null, null \rangle\}$. Η αντιστοιχία τριάδων $triplesMap' = \langle table, subjMap, predObjMaps \cup classPredObj \rangle$ ορίζεται ως η κανονικοποιημένη εκδοχή της $triplesMap$.

Ορισμός 5.3.18. (Κανονικοποιημένος R2RML γράφος αντιστοιχίας) Ένας R2RML γράφος αντιστοιχίας που περιλαμβάνει αποκλειστικά κανονικοποιημένες αντιστοιχίες τριάδων ή αντιστοιχίες τριάδων που δε χρήζουν κανονικοποίησης ονομάζεται *κανονικοποιημένος R2RML γράφος αντιστοιχίας*.

Παράδειγμα 5.3.1. Έστω η σχέση Review και η απλή αντιστοιχία που ορίστηκε στο παράδειγμα 5.1.1. Η αντιστοιχία αυτή είναι ισοδύναμη με την αντιστοιχία τριάδων $triplesMap = \langle table, subjMap, predObjMaps \rangle$, με $table$ =Review, $subjMap.subjGen$ μια συνάρτηση παραγωγής IRI από πρότυπο, η οποία χρησιμοποιεί το γνώρισμα nr και $predObjMap = \{poMap_1, poMap_2, poMap_3, poMap_4\}$, όπου $poMap_i$ αντιστοιχίες κατηγορήματος-αντικειμένου για καθένα από τα 4

γνωρίσματα της σχέσης. Παραδείγματος χάριν, για το γνώρισμα product η σχετική αντιστοιχία θα είναι η:

$$poMap_2 = \langle \{pMap_2\}, \{oMap_2\}, \{gMap_2\} \rangle$$
$$pMap_2 = \langle \texttt{ex:isReviewFor}, IRI, true \rangle$$
$$oMap_2 = \langle product, Literal, true, null, \texttt{xsd:string} \rangle$$
$$gMap_2 = \langle \texttt{DG}, IRI, true \rangle \text{ η αντιστοιχία προκαθορισμένου γράφου}$$

Έστω επίσης ότι θέτουμε $subjMap.classIRI = \texttt{ex:Review}$ προκειμένου να παράγεται μια τριάδα της μορφής $\langle subjMap.subjGen(nr), \texttt{rdf:type}, \texttt{ex:Review} \rangle$ για κάθε πλειάδα της σχέσης Review. Πλέον, η αντιστοιχία τριάδων $triplesMap$ χρήζει κανονικοποίησης. Η κανονικοποιημένη εκδοχή της θα περιέχει επιπλέον την ακόλουθη αντιστοιχία κατηγορήματος-αντικειμένου:

$$poMap_5 = \langle \{pMap_5\}, \{oMap_5\}, \{gMap_5\} \rangle$$
$$pMap_5 = \langle \texttt{rdf:type}, IRI, true \rangle$$
$$oMap_5 = \langle \texttt{ex:Review}, IRI, true, null, null \rangle$$
$$gMap_5 = gMap_2 \text{ η αντιστοιχία προκαθορισμένου γράφου}$$

Σύμφωνα με τον ορισμό 5.3.17, η διαδικασία της κανονικοποίησης ερμηνεύει την παρουσία ενός $classIRI$ στοιχείου σε μια αντιστοιχία υποκειμένου (ιδιότητα class της R2RML) ως μια επιπρόσθετη αντιστοιχία κατηγορήματος-αντικειμένου στην περιβάλλουσα αντιστοιχία τριάδων, σύμφωνα με τη σημασιολογία της R2RML. Αυτό έχει ως αποτέλεσμα ο αλγόριθμος επανεγγραφής να μπορεί να αγνοήσει τιμές της ιδιότητας class και να αρκεί η εξέταση των αντιστοιχιών κατηγορήματος-αντικειμένου μιας αντιστοιχίας τριάδων. Παράλληλα, απλοποιείται και ο ακόλουθος ορισμός της συμβατότητας μιας αντιστοιχίας τριάδων και ενός προτύπου τετράδας.

Διαισθητικά, μια αντιστοιχία τριάδων είναι *συμβατή* με ένα πρότυπο τετράδας, όταν *μπορεί* να οδηγήσει στη δημιουργία τουλάχιστον μίας τετράδας που συμφωνεί με το δοθέν πρότυπο. Με άλλα λόγια, αν υποθέσουμε ότι μια αντιστοιχία τριάδων μπορεί να παραγάγει ένα RDF σύνολο δεδομένων D για κάποιο στιγμιότυπο ΒΔ, αυτή είναι συμβατή με το πρότυπο τετράδας qp αν υπάρχει SPARQL αντιστοιχία μ, τέτοια ώστε $\mu(qp) \in D$. Ο έλεγχος της συμβατότητας μιας αντιστοιχίας τριάδων με ένα πρότυπο τετράδας ανάγεται με τη σειρά του στον έλεγχο συμβατότητας των επιμέρους συστατικών της πρώτης με τα στοιχεία του δεύτερου, ήτοι στη συμβατότητα της αντιστοιχίας υποκειμένου με το υποκείμενο του προτύπου κ.ο.κ.. Συνεπώς, για να είναι συμβατή μια αντιστοιχία τριάδας με το πρότυπο qp, θα πρέπει να περιλαμβάνει τουλάχιστον μία συμβατή αντιστοιχία όρου για κάθε όρο του qp. Τα παραπάνω αναλύονται στους ορισμούς 5.3.19 – 5.3.21:

Ορισμός 5.3.19. (Συμβατότητα αντιστοιχιών όρου με RDF όρο) Έστω $pos : (I \cup B \cup L \cup V) \times RDF - Q \rightarrow \{sub, pred, obj, graph\}$ η *συνάρτηση θέσης*, η οποία επιστρέφει τη θέση ενός RDF όρου σε ένα πρότυπο τετράδας. Επίσης, ορίζουμε τις βοηθητικές συναρτήσεις $getAbsoluteIRI : (I \cup L) \times I \rightarrow I$ η οποία επιστρέφει ένα IRI δεδομένης μιας συμβολοσειράς και ενός βασικού IRI, $isCompatibleTemplate : L \times L \rightarrow \{true, false\}$, η οποία επιστρέφει true αν μια

συμβολοσειρά είναι συμβατή με ένα πρότυπο συμβολοσειράς, $datatype : L \rightarrow I$, η οποία επιστρέφει τον τύπο δεδομένων ενός λεκτικού και $language : L \rightarrow L$, η οποία επιστρέφει τη γλωσσική ετικέτα ενός λεκτικού.
Ένας RDF όρος *term* ενός προτύπου τετράδας *qp* είναι *συμβατός* με μια αντιστοιχία όρου *tmap*, δεδομένου ενός βασικού IRI $baseIRI$, όταν η *tmap* αναφέρεται στη θέση του *term* και ισχύει μία από τις επόμενες συνθήκες:

1. $term \in V$
2. $term \in B \wedge tmap.type = \text{BlankNode}$
3. $term \in I, tmap.type = \text{IRI}$ και ισχύει μία από τις ακόλουθες συνθήκες:
   α. η *tmap* είναι σταθερής τιμής και $getAbsoluteIRI(tmap.constant, baseIRI) = term$
   β. η *tmap* είναι αντιστοιχία τιμής από πρότυπο και $isCompatibleTemplate(getAbsoluteIRI(tmap.template, baseIRI), term) = true$
   γ. $pos(term) = obj$, η *tmap* είναι αναφέρουσα αντιστοιχία αντικειμένου και ο *term* είναι συμβατός με το $tmap.parentMap.subjMap$
4. $term \in L, tmap.type = \text{Literal}$ και ισχύει μία από τις ακόλουθες συνθήκες:
   α. η *tmap* είναι σταθερής τιμής και $tmap.constant = term$
   β. η *tmap* είναι τιμής από στήλη, $tmap.datatype = datatype(term)$ και $tmap.language = language(term)$
   γ. η *tmap* είναι τιμής από πρότυπο, $isCompatibleTemplate(tmap.template, term) = true$, $tmap.datatype = datatype(term)$ και $tmap.language = language(term)$

Αξίζει να παρατηρηθεί ότι ο ορισμός 5.3.19 θεωρεί τις αντιστοιχίες τιμής από στήλη a priori συμβατές με οποιοδήποτε RDF όρο (εκτός και αν η ασυμβατότητα οφείλεται στα στοιχεία *type*, *language* ή *datatype* της αντιστοιχίας όρου). Αυτή η παραδοχή διευκολύνει την πραγματοποίηση του ελέγχου συμβατότητας, καθώς αποφεύγει την εξέταση της στήλης ενός πίνακα προκειμένου να βρεθεί μια συγκεκριμένη τιμή. Ο ορισμός 5.3.20 χρησιμοποιεί τον ορισμό 5.3.19 για να ορίσει τη συμβατότητα μιας ολόκληρης αντιστοιχίας τριάδων με τον όρο γράφου ενός προτύπου τριάδων.

Ορισμός 5.3.20. (Συμβατότητα αντιστοιχίας τριάδων με όρο γράφου ενός προτύπου τετράδας) Έστω μια αντιστοιχία τριάδων $triplesMap = \langle table, subjMap, predObjMaps \rangle$, ένα πρότυπο τετράδας *qp* με όρο γράφου *graph*, ένα σύνολο ονομαστικών γράφων *namedGraphs* και ένα σύνολο γράφων *defGraphs*, η ένωση των οποίων αποτελεί τον προκαθορισμένο γράφο.
Η αντιστοιχία τριάδων *triplesMap* είναι συμβατή με τον όρο γράφου *graph*, όταν ισχύει μία από τις παρακάτω συνθήκες:

1. $graph \in V$ και $\exists graphMap \in subjMap.graphMaps \cup predObjMaps.graphMaps$ και $\exists namedGraph \in namedGraphs$, έτσι ώστε *graphMap* και *namedGraph* συμβατά σύμφωνα με τον ορισμό 5.3.19

2. $graph \in I$ και $\exists graphMap \in subjMap.graphMaps \cup predObjMaps.graphMaps$, έτσι ώστε $graphMap$ συμβατή με $graph$

3. $graph = DG$ και ισχύει μία από τις ακόλουθες συνθήκες:

   α. $\exists graphMap \in subjMap.graphMaps \cup predObjMaps.graphMaps$, έτσι ώστε $graphMap$ είναι σταθερής τιμής και $graphMap.constant = DG$

   β. $\exists graphMap \in subjMap.graphMaps \cup predObjMaps.graphMaps$ και $\exists defaultGraph \in defGraphs$, έτσι ώστε $graphMap$ και $defaultGraph$ συμβατά σύμφωνα με τον ορισμό 5.3.19

Σημειώνεται ότι τα σύνολα $namedGraphs$ και $defGraphs$ που αναφέρονται στον ορισμό 5.3.20 δηλώνονται μέσω των προτάσεων `FROM NAMED` και `FROM` αντίστοιχα σε ένα SPARQL ερώτημα. Πλέον, είμαστε σε θέση να ορίσουμε τη συμβατότητα μιας αντιστοιχίας τριάδων με ένα πρότυπο τετράδας.

Ορισμός 5.3.21. (Συμβατότητα αντιστοιχίας τριάδων με πρότυπο τετράδας) Μια αντιστοιχία τριάδων $triplesMap = \langle table, subjMap, predObjMaps \rangle$ είναι συμβατή με ένα πρότυπο τετράδας $qp = \langle s, p, o, g \rangle$, όταν ισχύουν *όλες* οι παρακάτω συνθήκες:

1. η $subjMap$ είναι συμβατή με το s

2. $\exists predObjMap \in predObjMaps$, έτσι ώστε $\exists predMap \in predObjMap.predMaps$: $predMap$ συμβατή με p και $\exists objMap \in predObjMap.objMaps$: $objMap$ συμβατή με o

3. η $triplesMap$ είναι συμβατή με το g

Παράδειγμα 5.3.2. Έστω η αντιστοιχία τριάδων $triplesMap$ του παραδείγματος 5.3.1 και τα ακόλουθα πρότυπα τριάδων:

$$qp_1 = \langle ?s, \texttt{ex:isReviewFor}, \text{“3Bit”}, DG \rangle$$
$$qp_2 = \langle ?s, ?p, ?o, ?g \rangle$$
$$qp_3 = \langle ?s, \texttt{ex:hasRating}, ?s, DG \rangle$$
$$qp_4 = \langle ?s, \texttt{dc:title}, ?o, ?g \rangle$$
$$qp_5 = \langle ?s, \texttt{ex:reviewDate}, \text{“2012-09-14”}, \texttt{ex:graph1} \rangle$$

Η $triplesMap$ είναι συμβατή με τα qp_1, qp_2, qp_3 και μη συμβατή με τα qp_4 (μη συμβατό κατηγόρημα), qp_5 (μη συμβατός γράφος). Τονίζουμε ότι η συμβατότητα με το qp_1 δεν επηρεάζεται από το γεγονός ότι η συγκεκριμένη τιμή του αντικειμένου του δεν υπάρχει στη ΒΔ, ενώ με αφορμή τη συμβατότητα με το qp_3, σημειώνουμε ότι οι ορισμοί 5.3.19 – 5.3.21 δε λαμβάνουν υπόψη τους την παρουσία της ίδιας μεταβλητής σε παραπάνω από μία θέσεις.

Συχνά οι αντιστοιχίες τριάδων αποτελούν ένα αρκετά μεγάλο υποσύνολο του R2RML γράφου αντιστοιχίας, με αποτέλεσμα η χρήση τους για την καταγραφή της προέλευσης ενός προτύπου τετράδας κατά τη διαδικασία επανεγγραφής να μην είναι απόλυτα ικανοποιητική. Αυτός είναι και ο λόγος που εισάγουμε επίσης τις έννοιες του παραγωγού τετράδας και του παραγωγού κόμβου.

Ορισμός 5.3.22. (Παραγωγός τετράδας) Μια πλειάδα της μορφής $\langle table, subj\ Map, predMap, objMap, graphMap\rangle$, όπου *table* ένας λογικός πίνακας ο οποίος μπορεί να δηλώνεται με κάποιο ψευδώνυμο, *subjMap* μια αντιστοιχία υποκειμένου, *predMap* μια αντιστοιχία κατηγορήματος, *objMap* μια αντιστοιχία αντικειμένου ή αναφέρουσα αντιστοιχία αντικειμένου και *graphMap* μια αντιστοιχία γράφου, ονομάζεται *παραγωγός τετράδας*.

Ορισμός 5.3.23. (Παραγωγός κόμβου) Μια πλειάδα της μορφής $\langle table, term\ Map\ \rangle$, όπου *table* ένας λογικός πίνακας ο οποίος μπορεί να δηλώνεται με κάποιο ψευδώνυμο και *termMap* μια αντιστοιχία όρου ή μια αναφέρουσα αντιστοιχία αντικειμένου, ονομάζεται *παραγωγός κόμβου*.

Με άλλα λόγια, ένας παραγωγός τετράδας είναι το ελάχιστο εκείνο υποσύνολο μιας αντιστοιχίας τριάδων που υπονοεί την παραγωγή μιας τετράδας για κάθε πλειάδα του λογικού πίνακα *table* με αναγνωριστικό *alias*, ενώ το ίδιο ισχύει και για τον παραγωγό κόμβου, αλλά σε επίπεδο RDF όρου. Στο στάδιο της επανεγγραφής του SPARQL ερωτήματος, χρησιμοποιούνται *ελάχιστοι* λογικοί πίνακες για το στοιχείο *table* των παραγωγών τετράδας και κόμβου. Ένας **ελάχιστος λογικός πίνακας** ορίζεται ως ένας λογικός πίνακας μιας αντιστοιχίας τριάδων ή μια συνένωση μεταξύ δύο λογικών πινάκων αντιστοιχιών τριάδων. Όπως θα δούμε και στην παράγραφο 5.3.3, η τελευταία περίπτωση συναντάται όταν ένας παραγωγός τετράδων περιέχει μια αναφέρουσα αντιστοιχία αντικειμένου με μη κενό σύνολο *joins*.

Επίσης, ορίζουμε τις έννοιες της συμβατότητας ενός παραγωγού τετράδας με ένα πρότυπο τετράδας, καθώς και της ασυμβατότητας αντιστοιχιών όρου και παραγωγών κόμβου.

Ορισμός 5.3.24. (Συμβατότητα παραγωγού τετράδας με πρότυπο τετράδας) Ένας παραγωγός τετράδας $qgen = \langle table, subjMap, predMap, objMap, graphMap\rangle$ είναι συμβατός με ένα πρότυπο τετράδας $qp = \langle s, p, o, g\rangle$, όταν ισχύουν όλες οι παρακάτω συνθήκες:

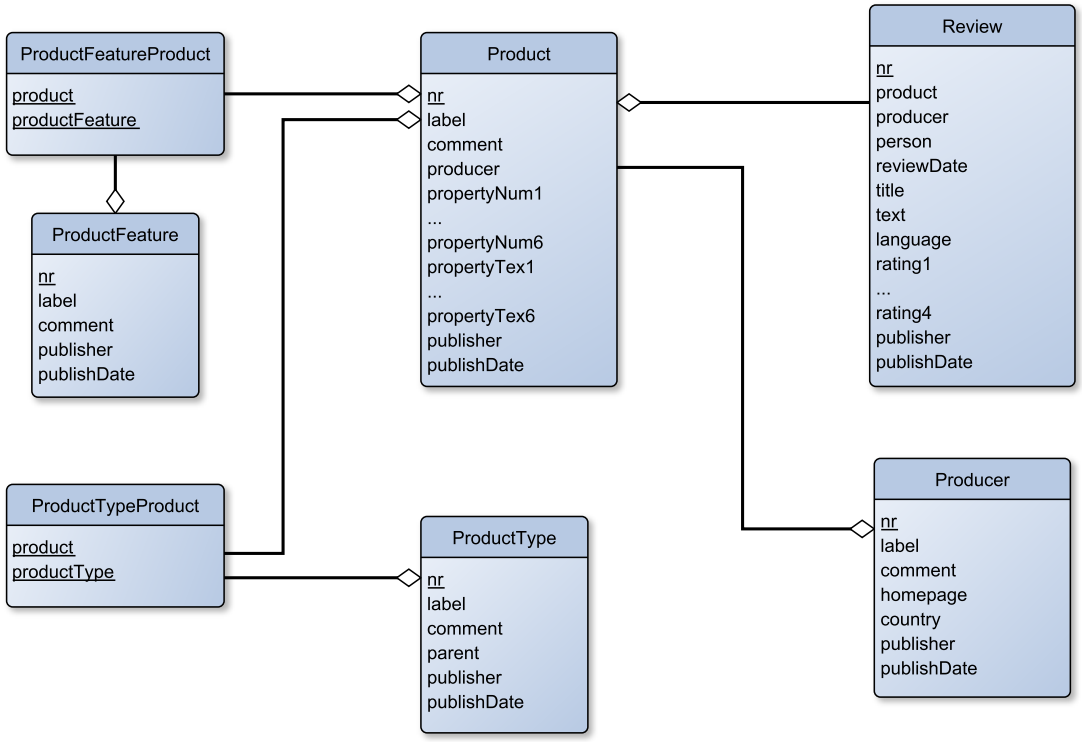
1. η *subjMap* είναι συμβατή με το s
2. η *predMap* είναι συμβατή με το p
3. η *objMap* είναι συμβατή με το o
4. η *graphMap* είναι συμβατή με το g

Ορισμός 5.3.25. (Ασυμβατότητα αντιστοιχιών όρου) Δύο αντιστοιχίες όρου $tmap_1$, $tmap_2$ ονομάζονται *ασύμβατες*, όταν δεν μπορούν να οδηγήσουν στην παραγωγή του ίδιου RDF όρου. Θεωρούμε τη συνάρτηση *isCompatibleTemplate* από τον ορισμό 5.3.19. Οι $tmap_1, tmap_2$ είναι ασύμβατες, όταν ισχύει μία από τις παρακάτω συνθήκες:

1. $tmap_1.type \neq tmap_2.type$
2. $tmap_1$ και $tmap_2$ είναι σταθερής τιμής και $tmap_1.constant \neq tmap_2.constant$
3. $tmap_1$ και $tmap_2$ είναι τιμής από πρότυπο και $isCompatibleTemplate\ (tmap_1.template, tmap_2.template) = false$
4. $tmap_1$ (αντ. $tmap_2$) είναι σταθερής τιμής και $tmap_2$ (αντ. $tmap_1$) είναι τιμής από πρότυπο και $isCompatibleTemplate\ (tmap_2.template, tmap_1.constant) = false$ (αντ. $isCompatibleTemplate(tmap_1.template, tmap_2.constant) = false$)

Ορισμός 5.3.26. (Ασυμβατότητα παραγωγών κόμβου) Δύο παραγωγοί κόμβου $ngen_1 = \langle table_1, tmap_1 \rangle$, $ngen_2 = \langle table_2, tmap_2 \rangle$ ονομάζονται *ασύμβατοι*, όταν δεν μπορούν να οδηγήσουν στην παραγωγή του ίδιου RDF όρου, δηλαδή όταν η $tmap_1$ είναι ασύμβατη με την $tmap_2$ σύμφωνα με τον ορισμό 5.3.25.

Για την καλύτερη κατανόηση της συνολικής διαδικασίας επανεγγραφής, στην τρέχουσα ενότητα θα αναφερόμαστε κατά κύριο λόγο σε παραδείγματα SPARQL ερωτημάτων που τίθενται στο σχεσιακό σχήμα της μεθοδολογίας αξιολόγησης μηχανών εκτέλεσης SPARQL ερωτημάτων Berlin SPARQL Benchmark (BSBM) [40], υπό την παρουσία κατάλληλων R2RML αντιστοιχιών. Το συγκεκριμένο σχήμα αναφέρεται σε μια περίπτωση χρήσης ηλεκτρονικού εμπορίου, όπου ένα σύνολο προϊόντων είναι διαθέσιμο από διάφορους προμηθευτές, ενώ καταναλωτές γράφουν κριτικές για τα προϊόντα αυτά σε διάφορους σχετικούς ιστοτόπους. Ένα μέρος της σχεσιακής αναπαράστασης του BSBM, το οποίο και χρησιμοποιούμε στα παραδείγματα αυτής της ενότητας απεικονίζεται στο σχήμα 5.3.



Σχήμα 5.3: *Μέρος της σχεσιακής αναπαράστασης του Berlin SPARQL Benchmark*

5.3.2 Μετασχηματισμός SPARQL ερωτήματος

Ένας τυπικός αλγόριθμος επανεγγραφής SPARQL ερωτημάτων δέχεται ως είσοδο ένα SPARQL ερώτημα εκφρασμένο σε μορφή κατάλληλη προς επεξεργασία. Η πλέον βολική μορφή είναι αυτή που χρησιμοποιεί τελεστές της SPARQL άλγεβρας, καθώς η σημασιολογία και αποτίμησή τους είναι ορισμένη με τυπικό τρόπο στην επίσημη προδιαγραφή της SPARQL, γεγονός που επιτρέπει το μετασχηματισμό μιας αλγεβρικής SPARQL έκφρασης σε μια ισοδύναμή

της. Συνεπώς, κάθε μηχανή SPARQL ερωτημάτων πραγματοποιεί τους κατάλληλους μετασχηματισμούς που φέρνουν το SPARQL ερώτημα στη μορφή εκείνη η οποία διευκολύνει περισσότερο την αποτίμησή του.

Ακολουθώντας την ίδια λογική και έχοντας πάντα ως κίνητρο την παραγωγή ενός SQL ερωτήματος χωρίς περιττές συνενώσεις όπως στα εισαγωγικά παραδείγματα 5.1.1 και 5.1.2, επεκτείνουμε τη SPARQL άλγεβρα με δύο νέους τελεστές, οι οποίοι διευκολύνουν τη διαδικασία της επανεγγραφής (παράγραφος 5.3.3). Στην τρέχουσα παράγραφο, εισάγουμε τη SPARQL-$\mathbb{DB}$ άλγεβρα, η οποία αποτελεί επέκταση της κλασικής SPARQL άλγεβρας [93] και ορίζουμε μια σειρά μετασχηματισμών με τη βοήθεια των οποίων ένα SPARQL ερώτημα εκφράζεται σε όρους αυτής της εκτεταμένης άλγεβρας.

Όπως έγινε φανερό στην ενότητα 5.1, η πλειονότητα των SPARQL-σε-SQL μεθόδων μετατρέπει ένα απλό SPARQL ερώτημα που περιέχει n πρότυπα τριάδων σε ένα SQL ερώτημα με τουλάχιστον $n-1$ συνενώσεις. Αυτό συμβαίνει επειδή οι μέθοδοι αυτές αντιμετωπίζουν ως ελάχιστη υπομονάδα ενός SPARQL ερωτήματος ένα πρότυπο τριάδας, με αποτέλεσμα να ξεκινούν με την κατασκευή ενός βασικού SQL ερωτήματος που αντιστοιχεί σε ένα πρότυπο τριάδας και να το χρησιμοποιούν ως εμφωλευμένο υποερώτημα καθώς προχωρούν στην επεξεργασία του υπόλοιπου SPARQL προτύπου γράφου. Αντίθετα, στόχος του προτεινόμενου αλγορίθμου είναι να ξεκινήσει την κατασκευή του SQL ερωτήματος από μια *ομάδα προτύπων* και να το *εμπλουτίσει* σταδιακά.

Επίσης, οι αλγόριθμοι που έχουν μέχρι προταθεί μέχρι σήμερα δε δίνουν ιδιαίτερη έμφαση στη θεώρηση του RDF γράφου ως μέρους ενός συνόλου δεδομένων, σε αντίθεση με τις προδιαγραφές των SPARQL και R2RML όπου η έννοια του ονομαστικού γράφου κατέχει εξέχουσα θέση. Όπως λοιπόν οι αλγόριθμοι της ενότητας 5.1 δεν αποφεύγουν τη συνένωση για το απλά SPARQL ερωτήματα των παραδειγμάτων 5.1.1 και 5.1.2, το ίδιο συμβαίνει και με το ακόλουθο ερώτημα:

```
PREFIX ex: <http://example.org>
SELECT ?review ?rating ?date
WHERE {
        GRAPH ex:graph1 {?review ex:hasRating ?rating}.
        GRAPH ex:graph2 {?review ex:reviewDate ?date}
      }
```

όπου εδώ τα 2 πρότυπα τριάδων ανήκουν σε 2 διαφορετικούς ονομαστικούς γράφους. Στο συγκεκριμένο ερώτημα, παρά το γεγονός ότι η μεταβλητή `?review` χρειάζεται να αντιστοιχηθεί στην ίδια τιμή και για τα 2 πρότυπα τριάδων, όλοι οι μέχρι σήμερα προταθέντες αλγόριθμοι αδυνατούν να αποφύγουν τη συνένωση πινάκων στο παραγόμενο SQL ερώτημα. Αυτό ισχύει ακόμα και για τις εργασίες [77, 162], οι οποίες παράγουν επίπεδα SQL ερωτήματα, και οφείλεται στη θεώρηση προτύπων τριάδων την οποία ακολουθούν.

Όλα τα παραπάνω μας οδηγούν στη *θεώρηση προτύπων τετράδων* κατά την επεξεργασία ενός SPARQL ερωτήματος καθώς και στον ορισμό κατάλληλων SPARQL τελεστών που εφαρμόζονται σε ομάδες τέτοιων προτύπων τετράδων, ή ισοδύναμα σε *πρότυπα συνόλου δεδομένων*. Στην επόμενη παράγραφο, ορίζουμε τη SPARQL-$\mathbb{DB}$ άλγεβρα και τη σημασιολογία των τελεστών της.

5.3.2.1 Η SPARQL-𝔻𝔹 άλγεβρα

Η SPARQL-𝔻𝔹 άλγεβρα χρησιμοποιεί ως βάση την κλασική SPARQL άλγεβρα και προσθέτει σε αυτήν δύο νέους τελεστές, οι οποίοι αποτελούν συνδυασμό υπαρχόντων SPARQL τελεστών. Οι δύο νέοι SPARQL-𝔻𝔹 τελεστές ορίζονται στη συνέχεια.

Ορισμός 5.3.27. (Πρότυπο βασικού συνόλου δεδομένων) Έστω $bdp = \{qp_i, 1 \leq i \leq n\}$ ένα σύνολο n προτύπων τετράδων. Το bdp ονομάζεται *πρότυπο βασικού συνόλου δεδομένων* και ο αντίστοιχος τελεστής BDP αποτελεί επέκταση του τελεστή προτύπου βασικού γράφου BGP.
Έστω μια διαμέριση του συνόλου bdp σε επιμέρους σύνολα που περιέχουν πρότυπα τετράδων που ανήκουν στον ίδιο γράφο. Υποθέτουμε ότι στο bdp εμφανίζονται k $(0 \leq k \leq n)$ IRIs στη θέση του γράφου των προτύπων τετράδας, l $(0 \leq l \leq n)$ μεταβλητές και ο προκαθορισμένος γράφος DG. Αυτό σημαίνει ότι τα πρότυπα τετράδων του bdp θα ομαδοποιηθούν σε $k + l + 1$ πρότυπα γράφου για καθένα από τα $IRI_1, \ldots, IRI_k, var_1, \ldots, var_l$ και DG. Ακολουθώντας το συμβολισμό της SPARQL, μπορούμε να συμβολίσουμε ένα από τα παραπάνω πρότυπα γράφου ως $Graph(IRI_1, gp_1)$.
Σύμφωνα με τα παραπάνω, ο τελεστής BDP μπορεί να οριστεί με βάση τους τελεστές $Join$ και $Graph$ της κλασικής SPARQL άλγεβρας ως εξής:

$$\begin{aligned} BDP(bdp) = & Join\left(Join\left(Join\left(\ldots\left(Join\left(Graph\left(IRI_1, gp_1\right), Graph\left(IRI_2, gp_2\right)\right), \ldots\right.\right.\right. \\ & Graph\left(IRI_n, gp_n\right)\left.\left.\left.\right)\ldots\right), Join\left(\ldots\left(Join\left(Graph\left(var_1, gp_{v1}\right),\right.\right.\right. \\ & \left.\left.\left.Graph\left(var_2, gp_{v2}\right), \ldots Graph\left(var_n, gp_{vn}\right)\right)\ldots\right)\right), gp_{DG}\right) \end{aligned} \tag{5.1}$$

Η εξίσωση (5.1) υποδεικνύει και τη σημασιολογία της αποτίμησης του τελεστή BDP, η οποία βασίζεται στις συναρτήσεις αποτίμησης των πρωταρχικών SPARQL τελεστών, όπως αυτές έχουν οριστεί στην προδιαγραφή της SPARQL.

Ορισμός 5.3.28. (Πρότυπο προαιρετικού συνόλου δεδομένων) Έστω bdp ένα πρότυπο βασικού συνόλου δεδομένων και $optExprs : RDF\text{-}Q \rightarrow SPARQL\text{-}Expr$ μια συνάρτηση που αντιστοιχεί ένα πρότυπο τετράδας με μια SPARQL λογική έκφραση (συμβολίζουμε το σύνολο όλων των SPARQL λογικών εκφράσεων με $SPARQL\text{-}Expr$). Η συνάρτηση $optExprs$ ονομάζεται *συνάρτηση συνθηκών αριστερής συνένωσης*.
Ορίζουμε τον *τελεστή προτύπου προαιρετικού συνόλου δεδομένων* BDPLeftJoin $(bdp_1, bdp_2, optExprs)$, όπου $dom(optExprs) = bdp_2$ και $bdp_1 \cap bdp_2 = \emptyset$, ως μια συντόμευση για το συμβολισμό διαδοχικών αριστερών εξωτερικών συνενώσεων του προτύπου βασικού συνόλου δεδομένων bdp_1 με καθένα από τα πρότυπα τετράδων qp_{2i} που ανήκουν στο (διατεταγμένο) bdp_2 και αντίστοιχες συνθήκες συνένωσης $optExprs(qp_{2i})$. Δεδομένου λοιπόν του ότι $bdp_2 = \{qp_{2i}, 1 \leq i \leq n\}$[8] και με βάση τα παραπάνω, ο τελεστής BDPLeftJoin μπορεί να εκφραστεί χρησιμοποιώντας τον πρωταρχικό SPARQL τελεστή $LeftJoin$ ως εξής:

[8] Υπενθυμίζουμε ότι η σειρά των qp_{2i} εντός του bdp_2 έχει σημασία.

$$BDPLeftJoin\left(bdp_1, bdp_2, optExprs\right) = \\ LeftJoin\left(\ldots\left(LeftJoin\left(bdp_1, qp_{21}, optExprs(qp_{21})\right), \ldots, qp_{2n}, optExprs(qp_{2n})\right)\right) \tag{5.2}$$

Η αποτίμηση του τελεστή BDPLeftJoin βασίζεται στην εξίσωση (5.2) και στη συνάρτηση αποτίμησης του τελεστή *LeftJoin*, όπως αυτή ορίζεται στην προδιαγραφή της SPARQL.

Ορισμός 5.3.29. (SPARQL-𝔻𝔹 άλγεβρα) Η SPARQL-𝔻𝔹 άλγεβρα ορίζεται ως η επέκταση της SPARQL άλγεβρας που περιλαμβάνει τους τελεστές BDP και BDPLeftJoin.

Είναι προφανές ότι οι δύο νέοι τελεστές της SPARQL-𝔻𝔹 άλγεβρας αποτελούν συντακτικές συντομεύσεις αλληλουχιών πρωταρχικών SPARQL τελεστών, οι οποίες βοηθούν στην αναγνώριση προτύπων σαν και αυτών που παρουσιάστηκαν στα παραδείγματα 5.1.1 και 5.1.2, προκειμένου να αντιμετωπιστούν ξεχωριστά και να εξεταστεί η πιθανότητα αποφυγής περιττών συνενώσεων. Στην επόμενη παράγραφο, παρουσιάζεται η διαδικασία μετασχηματισμού ενός SPARQL ερωτήματος σε μια έκφραση που χρησιμοποιεί τελεστές της SPARQL-𝔻𝔹 άλγεβρας.

5.3.2.2 Μετασχηματισμός σε SPARQL-𝔻𝔹

Σε αυτή την παράγραφο, περιγράφουμε το μετασχηματισμό ενός SPARQL ερωτήματος σε μια SPARQL-𝔻𝔹 αλγεβρική έκφραση. Συνοπτικά, τα βήματα της διαδικασίας είναι τα ακόλουθα:

0. Μετασχηματισμός σε SPARQL άλγεβρα σύμφωνα με τον αλγόριθμο της SPARQL προδιαγραφής [93]
1. Μετατροπή της αλγεβρικής έκφρασης σε τελεστές τετράδων
2. Ομαδοποίηση προτύπων τετράδων σε BDP τελεστές
3. Αναδιάταξη Join και LeftJoin τελεστών
4. Ομαδοποίηση προαιρετικών προτύπων γράφων σε BDPLeftJoin τελεστές

Τα βήματα 0 και 1 είναι τετριμμένα και μετατρέπουν ένα SPARQL ερώτημα σε αλγεβρική μορφή τετράδων, κατάλληλο για περαιτέρω επεξεργασία. **Το βήμα 2 ομαδοποιεί πρότυπα τετράδων**, προκειμένου να δημιουργήσει ένα όσο το δυνατόν μεγαλύτερο πρότυπο βασικού συνόλου δεδομένων, σύμφωνα πάντα με τη σημασιολογία των SPARQL τελεστών. Η αναγνώριση μέγιστων προτύπων συνόλου δεδομένων εξαντλεί τις πιθανότητες για εύρεση συσχετίσεων μεταξύ προτύπων τετράδων και, κατ' επέκταση, αποφυγή περιττών συνενώσεων.

Η διαδικασία του μετασχηματισμού του βήματος 2 βασίζεται στις ακόλουθες αλγεβρικές ισοδυναμίες, όπου qp_i ένα πρότυπο τετράδας, bdp_i είναι ένα

πρότυπο βασικού συνόλου δεδομένων, *dp* ένα σύνθετο πρότυπο συνόλου δεδομένων, *var* το σύνολο των μεταβλητών ενός προτύπου ή μιας SPARQL λογικής έκφρασης και *expr* η σύζευξη ενός συνόλου SPARQL λογικών εκφράσεων:

$$QuadPattern\left(\{qp_i, 1 \leq i \leq n\}\right) = BDP\left(\{qp_i, 1 \leq i \leq n\}\right) \tag{5.3}$$

$$Join\left(BDP\left(bdp_1\right), BDP\left(bdp_2\right)\right) = BDP\left(bdp_1 \cup bdp_2\right) \tag{5.4}$$

$$Join\left(BDP\left(bdp_1\right), Filter\left(BDP\left(bdp_2\right), expr\right)\right) = Filter\left(BDP\left(bdp_1 \cup bdp_2\right), expr\right) \tag{5.5}$$

$$Project\left(BDP\left(bdp_1\right), var_1\right) = BDP\left(bdp_1\right), \text{ ανν } var_1 = var(bdp_1) \tag{5.6}$$

$$Project\left(Filter\left(BDP\left(bdp_1\right), expr\right), var_1\right) = Filter\left(BDP\left(bdp_1\right), expr\right), \text{ ανν } var_1 = var(bdp_1) \tag{5.7}$$

$$Join\left(BDP\left(bdp_1\right), ToMultiSet\left(Project\left(BDP\left(bdp_2\right), var_2\right)\right)\right) = BDP\left(bdp_1 \cup bdp_2\right), \text{ ανν } var_2 = var(bdp_2) \tag{5.8}$$

$$Join\left(BDP\left(bdp_1\right), ToMultiSet\left(Project\left(Filter\left(BDP\left(bdp_2\right), expr\right), var_2\right)\right)\right) = Filter\left(BDP\left(bdp_1 \cup bdp_2\right), expr\right), \text{ ανν } var_2 = var(bdp_2) \tag{5.9}$$

$$Filter\left(Filter\left(dp, expr_1\right), expr_2\right) = Filter\left(dp, expr_1 \cap expr_2\right) \tag{5.10}$$

Στις εξισώσεις (5.3) – (5.10), ο τελεστής QuadPattern είναι ο SPARQL τελεστής που ομαδοποιεί πρότυπα τετράδων που ανήκουν στον ίδιο γράφο, ο τελεστής Filter συμβολίζει τον τελεστή φιλτραρίσματος (αντίστοιχο της σχεσιακής πράξης της επιλογής σ), ο τελεστής Project την προβολή μεταβλητών και ο τελεστής ToMultiset μετατρέπει μια SPARQL λύση σε πολυσύνολο αντιστοιχιών και συμβολίζει την παρουσία ενός υποερωτήματος.

Η επαλήθευση της ισχύος των παραπάνω εξισώσεων προκύπτει άμεσα από την αντικατάσταση του BDP τελεστή με την ισοδύναμη έκφραση του από την εξίσωση (5.1) και τη χρήση της σημασιολογίας των πρωταρχικών SPARQL τελεστών που αναφέρονται σε αυτές. Το δέντρο SPARQL τελεστών διασχίζεται bottom-up (ή ισοδύναμα, η αλγεβρική έκφραση θεωρείται από τις εσωτερικές παρενθέσεις προς τις έξω) και οι εκφράσεις των αριστερών μελών των εξισώσεων (5.3) – (5.10) αντικαθίστανται από τα αντίστοιχα δεξιά μέλη. Αυτό έχει ως αποτέλεσμα τη μετατροπή της αρχικής αλγεβρικής έκφρασης σε μια ισοδύναμη που περιέχει BDP τελεστές.

Συνοπτικά, οι παραπάνω ισοδυναμίες ομαδοποιούν σύνολα προτύπων τετράδων που συνενώνονται μεταξύ τους με ή χωρίς την παρουσία φίλτρου (εξισώσεις (5.4) και (5.5)), απλοποιούν υποερωτήματα που προβάλλουν όλες τις μεταβλητές που αναφέρονται σε αυτά (εξισώσεις (5.6) και (5.7)) και ομαδοποιούν τα πρότυπα βασικού συνόλου δεδομένων αυτών των υποερωτημάτων με άλλα βασικά πρότυπα με τα οποία συνενώνονται (εξισώσεις (5.8) και (5.9)). Οι μετατροπές που ορίζονται από αυτές τις ισοδυναμίες μετατοπίζουν τελεστές φίλτρου προς τη ρίζα του δέντρου τελεστών, γεγονός που μπορεί να έχει ως αποτέλεσμα την εμφώλευση φίλτρων. Τέτοιες εμφωλεύσεις απλοποιούνται σε έναν τελεστή φίλτρου σύμφωνα με την εξίσωση (5.10).

Το επόμενο βήμα αναφέρεται στην **αναδιάταξη των SPARQL Join και LeftJoin τελεστών**. Το συγκεκριμένο βήμα έχει τον ίδιο στόχο με αυτόν του προηγούμενου: τη δημιουργία όσο το δυνατόν μεγαλύτερων ομάδων από πρότυπα τετράδων. Για παράδειγμα, τα δύο επόμενα SPARQL ερωτήματα είναι

ισοδύναμα, το δεύτερο όμως είναι σε τέτοια μορφή που καθιστά δυνατή την ομαδοποίηση των προτύπων τετράδων t_1 και t_5 σε έναν BDP τελεστή και συνεπώς, σε δεύτερο στάδιο (παράγραφος 5.3.3), την αποφυγή περιττών συνενώσεων κατά την παραγωγή του SQL ερωτήματος:

```
SELECT *                                SELECT *
WHERE {                                 WHERE{
(t1):    ?x :p1 ?value1.                (t1):   ?x :p1 ?value1.
(t2):    OPTIONAL{ ?x :p2 ?value2.      (t5):   ?x :p5 ?value5.
(t3):              ?y :p3 ?value3 }.    (t2):   OPTIONAL{ ?x :p2 ?value2.
(t4):    ?y :p4 ?value4.                (t3):             ?y :p3 ?value3 }.
(t5):    ?x :p5 ?value5.                (t4):   ?y :p4 ?value4.
}                                       }
```

Η ισοδυναμία των δύο ερωτημάτων φαίνεται διαισθητικά από τη σύγκριση των δύο ερωτημάτων και την παρατήρηση πως, για να ανήκει μια αντιστοιχία στη SPARQL λύση και των δυο ερωτημάτων, θα πρέπει υποχρεωτικά να ικανοποιεί τα πρότυπα τριάδων t_1, t_4, t_5 και προαιρετικά, το συνδυασμό των t_2 και t_3. Η σειρά με την οποία πραγματοποιείται η αποτίμηση των συγκεκριμένων προτύπων δεν επηρεάζει την τελική SPARQL λύση. Αυτό σημαίνει ότι στο προηγούμενο ερώτημα, εκτός του t_5, και το πρότυπο t_4 είναι δυνατό να μετατοπιστεί πριν τον OPTIONAL τελεστή χωρίς να αλλάξει η σημασιολογία του ερωτήματος. Εντούτοις, όπως θα φανεί και στην παράγραφο 5.3.3, δεν υπάρχει κάποιο όφελος από μια τέτοια μετατόπιση, καθώς το t_4 δεν έχει κάποια κοινή μεταβλητή με τα υποχρεωτικά πρότυπα t_1 και t_5 και συνεπώς, δεν μπορούμε να αποφύγουμε τη συνένωση πινάκων για το t_4. Έτσι λοιπόν, το τρέχον βήμα του μετασχηματισμού μετατοπίζει πρότυπα τριάδων πριν από μια OPTIONAL έκφραση μόνο όταν αυτά έχουν τουλάχιστον μια κοινή μεταβλητή με το υποχρεωτικό τμήμα του προτύπου γράφου.

Τυπικά, το συγκεκριμένο βήμα του μετασχηματισμού βασίζεται στις εξισώσεις (5.11) – (5.13):

$$\begin{aligned}&Join\left(LeftJoin\left(BDP\left(bdp_1\right), BDP\left(bdp_2\right), true\right), BDP\left(bdp_3\right)\right) = \\ &Join\left(LeftJoin\left(BDP\left(bdp_1 \cup bdp_{3a}\right), BDP\left(bdp_2\right), true\right), BDP\left(bdp_{3b}\right)\right), \\ &bdp_{3a} \text{ το μέγιστο υποσύνολο του } bdp_3 : var(bdp_{3a}) \cap (var(bdp_2) - var(bdp_1)) = \varnothing \\ &\text{και } bdp_{3b} = bdp_3 - bdp_{3a}\end{aligned} \tag{5.11}$$

$$\begin{aligned}&Join\left(LeftJoin\left(Filter\left(BDP\left(bdp_1\right), expr\right), BDP\left(bdp_2\right), true\right), BDP\left(bdp_3\right)\right) = \\ &Join\left(LeftJoin\left(Filter\left(BDP\left(bdp_1 \cup bdp_{3a}\right), expr\right), BDP\left(bdp_2\right), true\right), BDP\left(bdp_{3b}\right)\right), \\ &bdp_{3a} \text{ το μέγιστο υποσύνολο του } bdp_3 : var(bdp_{3a}) \cap (var(bdp_2) - var(bdp_1)) = \varnothing \\ &\text{και } bdp_{3b} = bdp_3 - bdp_{3a}\end{aligned} \tag{5.12}$$

$$
\begin{aligned}
&Join\left(LeftJoin\left(BDP\left(bdp_1\right), BDP\left(bdp_2\right), true\right), Filter\left(BDP\left(bdp_3\right), expr\right)\right) = \\
&Join\left(LeftJoin\left(Filter\left(BDP\left(bdp_1 \cup bdp_{3a}\right), expr_1\right), BDP\left(bdp_2\right), true\right),\right. \\
&\left.Filter\left(BDP\left(bdp_{3b}\right), expr_2\right)\right), \\
&bdp_{3a} \text{ το μέγιστο υποσύνολο του } bdp_3 : var(bdp_{3a}) \cap (var(bdp_2) - var(bdp_1)) = \varnothing \\
&\text{και } bdp_{3b} = bdp_3 - bdp_{3a}, \\
&expr_1 \text{ το μέγιστο υποσύνολο του } expr : var(expr_1) \cap (var(bdp_2) - var(bdp_1)) = \varnothing \\
&\text{και } expr_2 = expr - expr_1
\end{aligned}
\tag{5.13}
$$

Οι εξισώσεις (5.12) και (5.13) αναφέρονται στην περίπτωση που κάποιο από τα υποχρεωτικά πρότυπα περιέχει κάποιον τελεστή φίλτρου, ο οποίος μπορεί να διασπαστεί σε δύο τελεστές, ακολουθώντας την ίδια λογική με αυτή της διάσπασης του υποχρεωτικού προτύπου bdp_{3a}, το οποίο έπεται της OPTIONAL πρότασης. Σημειώνεται ότι οι εξισώσεις (5.11) – (5.13) περιλαμβάνουν και την περίπτωση να μετατοπιστεί ολόκληρο το πρότυπο bdp_3 πριν τον OPTIONAL τελεστή, οπότε ισχύει $bdp_{3a} = bdp_3$ και $bdp_{3b} = \varnothing$.

Όπως και στο προηγούμενο βήμα του μετασχηματισμού, το δέντρο τελεστών που έχει προκύψει από το βήμα 2 διασχίζεται με κατεύθυνση bottom-up και αν βρεθεί κάποια από τις εκφράσεις των αριστερών μελών των εξισώσεων (5.11) – (5.13), αντικαθίσταται από το αντίστοιχο δεξιό μέλος.

Το τελευταίο βήμα του μετασχηματισμού στη SPARQL-$\mathbb{DB}$ άλγεβρα **αναγνωρίζει διαδοχικά πρότυπα προαιρετικών γράφων μοναδιαίου μεγέθους** και κατασκευάζει αντίστοιχους BDPLeftJoin τελεστές. Κίνητρο για την αναγνώριση τέτοιων προτύπων αποτελεί η αποφυγή εισαγωγής μιας περιττής συνένωσης για ερωτήματα όπως αυτό του παραδείγματος 5.1.2. Ο περιορισμός του μοναδιαίου μεγέθους για τα προαιρετικά πρότυπα γράφων οφείλεται στην παρατήρηση ότι για μεγαλύτερα μεγέθη προτύπων γράφων, η αριστερή εξωτερική συνένωση μεταξύ πινάκων δεν μπορεί να αποφευχθεί. Αυτό φαίνεται αν παρατηρήσουμε το πρώτο από τα ακόλουθα SPARQL ερωτήματα που επεκτείνουν το παράδειγμα 5.1.2:

```
SELECT ?product ?rating ?date
WHERE {
        ?review ex:isReviewFor ?product.
        OPTIONAL{
        ?review ex:hasRating ?rating.
        ?review ex:reviewDate ?date
        }
}
```

```
SELECT ?product ?rating ?date
WHERE {
        ?review ex:isReviewFor ?product.
        OPTIONAL{
        ?review ex:hasRating ?rating}.
        OPTIONAL{
        ?review ex:reviewDate ?date}
}
```

Το ερώτημα αυτό ανακτά όλα τα διαθέσιμα προϊόντα και προαιρετικά, τη βαθμολογία και ημερομηνία κριτικής, αλλά μόνο αν είναι και οι δύο διαθέσιμες. Αντίθετα, το δεύτερο ερώτημα ανακτά όλα τα προϊόντα και προαιρετικά, τη βαθμολογία και ημερομηνία κριτικής, ακόμα και αν ένα από τα δύο στοιχεία είναι μη διαθέσιμο. Για το στιγμιότυπο της σχέσης Review που ορίστηκε στο παράδειγμα 5.1.1, οι SPARQL λύσεις των δύο ερωτημάτων θα είναι αντίστοιχα οι εξής:

product	rating	date
''Twix''	''7''	''2011-07-23''
''Snickers''	-	-
''Bounty''	''3''	''2012-09-14''
''Mars''	-	-

product	rating	date
''Twix''	''7''	''2011-07-23''
''Snickers''	''8''	-
''Bounty''	''3''	''2012-09-14''
''Mars''	-	-

Γίνεται φανερό ότι για την περίπτωση του πρώτου SPARQL ερωτήματος, δεν υπάρχει ισοδύναμο SQL ερώτημα που να αποφεύγει την αριστερή εξωτερική συνένωση του πίνακα Review με τον εαυτό του, ενώ αντίθετα αυτό είναι δυνατό για το δεύτερο SPARQL ερώτημα και μάλιστα, το αποτέλεσμα είναι το ίδιο απλό SQL ερώτημα που παρουσιάστηκε στο παράδειγμα 5.1.2.

Οι αλγεβρικές ισοδυναμίες που ορίζουν το συγκεκριμένο βήμα του μετασχηματισμού είναι οι ακόλουθες, όπου *size* συνάρτηση που επιστρέφει το πλήθος των προτύπων τετράδων ενός προτύπου βασικού συνόλου δεδομένων:

$$LeftJoin\left(BDP\left(bdp_1\right), BDP\left(bdp_2\right), true\right) = \begin{cases} BDPLeftJoin\left(bdp_1, bdp_2, optExprs\right), & \text{με } optExprs(bdp_2) = true, \text{ ανν } size(bdp_2) = 1 \wedge \\ & (bdp_1 \cap bdp_2 = \varnothing) \\ BDP\left(bdp_1\right), \text{ ανν } bdp_1 \cap bdp_2 \neq \varnothing \end{cases} \tag{5.14}$$

$$LeftJoin\left(BDP\left(bdp_1\right), BDP\left(bdp_2\right), expr\right) = BDPLeftJoin\left(bdp_1, bdp_2, optExprs\right), \\ \text{με } optExprs(bdp_2) = expr, \text{ ανν } size(bdp_2) = 1 \wedge (bdp_1 \cap bdp_2 = \varnothing) \tag{5.15}$$

$$LeftJoin\left(Filter\left(BDP\left(bdp_1\right), expr\right), BDP\left(bdp_2\right), true\right) = \\ Filter\left(BDPLeftJoin\left(bdp_1, bdp_2, optExprs\right), expr\right), \text{ με } optExprs(bdp_2) = true, \\ \text{ανν } size(bdp_2) = 1 \wedge (bdp_1 \cap bdp_2 = \varnothing) \tag{5.16}$$

$$LeftJoin\left(Filter\left(BDP\left(bdp_1\right), expr_1\right), BDP\left(bdp_2\right), expr_2\right) = \\ Filter\left(BDPLeftJoin\left(bdp_1, bdp_2, optExprs\right), expr_1\right), \text{ με } optExprs(bdp_2) = expr_2, \\ \text{ανν } size(bdp_2) = 1 \wedge (bdp_1 \cap bdp_2 = \varnothing) \tag{5.17}$$

$$LeftJoin\left(BDPLeftJoin\left(bdp_1, bdp_2, optExprs\right), BDP\left(bdp_3\right), true\right) = \\ BDPLeftJoin\left(bdp_1, bdp_2 \cup bdp_3, optExprs'\right), \text{ με } optExprs' = optExprs \cup \\ \{optExprs'(bdp_3) = true \wedge optExprs(bdp_2)\}, \text{ ανν } size(bdp_3) = 1 \wedge (bdp_1 \cap bdp_3 = \varnothing) \tag{5.18}$$

$$LeftJoin\left(BDPLeftJoin\left(bdp_1, bdp_2, optExprs\right), BDP\left(bdp_3\right), expr\right) = \\ BDPLeftJoin\left(bdp_1, bdp_2 \cup bdp_3, optExprs'\right), \text{ με } optExprs' = optExprs \cup \\ \{optExprs'(bdp_3) = expr \wedge optExprs(bdp_2)\}, \text{ ανν } size(bdp_3) = 1 \wedge (bdp_1 \cap bdp_3 = \varnothing) \tag{5.19}$$

$$Join\left(BDP\left(bdp_1\right), BDPLeftJoin\left(bdp_2, bdp_3, optExprs\right)\right) = \begin{cases} BDP\left(bdp_1 \cup bdp_2\right), \text{ανν } bdp_3 \subseteq bdp_1 \\ BDPLeftJoin\left(bdp_1 \cup bdp_2, bdp_3, optExprs'\right), & \text{με } optExprs' = optExprs - \{optExprs(qp_i) : \\ & qp_i \in bdp_1\} \text{ ανν } bdp_3 \nsubseteq bdp_1 \end{cases} \tag{5.20}$$

$$Join\left(Filter\left(BDP\left(bdp_1\right),expr\right)BDPLeftJoin\left(bdp_2,bdp_3,optExprs\right)\right)=$$
$$\begin{cases} Filter\left(BDP\left(bdp_1\cup bdp_2\right),expr\right), \text{αν} \ bdp_3\subseteq bdp_1 \\ Filter\left(BDPLeftJoin\left(bdp_1\cup bdp_2,bdp_3,optExprs'\right),expr\right), \text{με} \ optExprs'=optExprs- \\ \{optExprs(qp_i):qp_i\in bdp_1\} \ \text{αν} \ bdp_3\nsubseteq bdp_1 \end{cases} \tag{5.21}$$

$$Join\left(BDP\left(bdp_1\right),Filter\left(BDPLeftJoin\left(bdp_2,bdp_3,optExprs\right),expr\right)\right)=$$
$$\begin{cases} Filter\left(BDP\left(bdp_1\cup bdp_2\right),expr\right), \text{αν} \ bdp_3\subseteq bdp_1 \\ Filter\left(BDPLeftJoin\left(bdp_1\cup bdp_2,bdp_3,optExprs'\right),expr\right), \text{με} \ optExprs'=optExprs- \\ \{optExprs(qp_i):qp_i\in bdp_1\} \ \text{αν} \ bdp_3\nsubseteq bdp_1 \end{cases} \tag{5.22}$$

Συνοπτικά, οι εξισώσεις (5.14) – (5.17) δημιουργούν ένα BDPLeftJoin τελεστή που προκύπτει από την εφαρμογή αριστερής συνένωσης μεταξύ BDP τελεστών, με ή χωρίς την παρουσία φίλτρων. Οι εξισώσεις (5.18) και (5.19) επεκτείνουν έναν ήδη υπάρχοντα BDPLeftJoin τελεστή που συνενώνεται εξωτερικά με έναν (φιλτραρισμένο ή μη) BDP τελεστή. Τέλος, οι εξισώσεις (5.20) – (5.22) εξετάζουν τις περιπτώσεις συνένωσης ενός BDPLeftJoin τελεστή και ενός BDP τελεστή, με ή χωρίς την παρουσία φίλτρων.

Όμοια με πριν, το συγκεκριμένο βήμα του μετασχηματισμού εφαρμόζεται μετά από διάσχιση του SPARQL δέντρου με κατεύθυνση bottom-up και αντικατάσταση των αριστερών μελών των εξισώσεων (5.14) – (5.22) με τα αντίστοιχα δεξιά μέλη.

Παράδειγμα 5.3.3. Έστω το ακόλουθο SPARQL ερώτημα που χρησιμοποιεί όρους από την BSBM οντολογία:

```
SELECT DISTINCT ?product ?label ?value1
WHERE {
(t1):     ?product rdfs:label ?label.
(t2):     ?product rdf:type bsbm-inst:ProductType1848.
(t3):     OPTIONAL{?product bsbm:productPropertyNumeric1 ?value1.}
(t4):     OPTIONAL{ ?product bsbm:productFeature bsbm-inst:ProductFeature43883.
(t5):     ?product bsbm:productFeature bsbm-inst:ProductFeature7746
          }
}
ORDER BY ?label
LIMIT 10
```

Με εφαρμογή του μετασχηματισμού που παρουσιάστηκε σε αυτή την παράγραφο, το παραπάνω SPARQL ερώτημα θα εκφραστεί σε όρους της SPARQL-$\mathbb{DB}$ άλγεβρας ως εξής:

```
Slice 0 10
  (Distinct
    (Project(?product ?label ?value1)
      (Order(?label)
        (Filter (> ?value1 136)
          (LeftJoin
            (BDPLeftJoin(bdp1,bdp2,optExprs),
            BDP(bdp3),true
          )))))
```

όπου `Slice` είναι ο SPARQL τελεστής που συμβολίζει το συνδυασμό των προτάσεων `LIMIT` και `OFFSET`, `Order` ο τελεστής διάταξης μιας SPARQL λύσης, το πρότυπο `bdp1` αποτελείται από τα πρότυπα τριάδων t_1 και t_2, το `bdp2` περιέχει μόνο το πρότυπο t_3 και `optExprs(bdp2)`=true, ενώ το `bdp3` αποτελείται από τα πρότυπα t_4 και t_5.

Ένα SPARQL ερώτημα εκφρασμένο σε όρους της SPARQL-$\mathbb{DB}$ άλγεβρας αποτελεί την είσοδο για το κύριο στάδιο του αλγορίθμου επανεγγραφής, το οποίο αναλύεται στην επόμενη παράγραφο.

5.3.3 Επανεγγραφή SPARQL ερωτήματος

Το δεύτερο και κύριο στάδιο του συνολικού αλγορίθμου αποτίμησης ενός SPARQL ερωτήματος είναι η επανεγγραφή αυτού σε ένα SQL ερώτημα το οποίο θα εκτελεστεί σε επόμενο στάδιο στη ΒΔ. Το συγκεκριμένο στάδιο λαμβάνει ως είσοδο ένα SPARQL ερώτημα εκφρασμένο σε όρους της SPARQL-$\mathbb{DB}$ άλγεβρας και έναν κανονικοποιημένο (βλέπε ορισμό 5.3.18) R2RML γράφο αντιστοιχίας. Ο αλγόριθμος διασχίζει το ισοδύναμο δέντρο SPARQL-$\mathbb{DB}$ τελεστών του ερωτήματος με κατεύθυνση bottom-up και δημιουργεί ένα μοντέλο που αναπαριστά ένα SQL ερώτημα, το οποίο και εμπλουτίζεται κατά τη διάσχιση του δέντρου. Το μοντέλο αυτό ονομάζεται SQL μοντέλο και μοιράζεται αρκετά κοινά στοιχεία με το αντίστοιχο μοντέλο του [77].

Η διαδικασία διάσχισης του δέντρου τελεστών και ο εμπλουτισμός του SQL μοντέλου βασίζεται εν μέρει στον αλγόριθμο του [77], ο οποίος έχει προσαρμοστεί κατάλληλα για την αντιμετώπιση των νέων SPARQL-$\mathbb{DB}$ τελεστών, υπό την παρουσία μιας R2RML αντιστοιχίας.

Στο [77], κεντρικό ρόλο κατά τη διάσχιση του SPARQL δέντρου παίζει η διατήρηση των **διαθέσιμων** και **υποχρεωτικών** μεταβλητών κάθε τελεστή. Το σύνολο των *διαθέσιμων* μεταβλητών ενός τελεστή είναι το σύνολο των μεταβλητών εκείνων που έχουν δεσμευτεί σε κάποια τιμή κατά την αποτίμηση του εν λόγω τελεστή ή κατά την αποτίμηση ενός τελεστή-απογόνου του συγκεκριμένου τελεστή. Αντίθετα, το σύνολο των μεταβλητών που χρειάζονται για την αποτίμηση ενός τελεστή ονομάζεται σύνολο *υποχρεωτικών* μεταβλητών του τελεστή.

Στον πίνακα 5.1, αναφέρονται συνοπτικά οι ορισμοί των συνόλων διαθέσιμων και υποχρεωτικών μεταβλητών για τους SPARQL τελεστές[9]. Για κάθε τελεστή q, ορίζονται οι αντίστοιχες συναρτήσεις *available* και *required*, οι συναρτήσεις *children* και *parent* επιστρέφουν τα παιδιά και το γονέα του τελεστή αντίστοιχα, ενώ η συνάρτηση *var* επιστρέφει τις μεταβλητές του τελεστή. Επίσης, ειδικά για τους τελεστές Join, LeftJoin και Minus (πράξη διαφοράς SPARQL αντιστοιχιών), ορίζονται οι συναρτήσεις *left* και *right*, οι οποίες επιστρέφουν το αριστερό και το δεξί παιδί του τελεστή αντίστοιχα.

Γενικά, όπως φαίνεται από τον πίνακα 5.1, οι διαθέσιμες μεταβλητές ενός τελεστή καθορίζονται από τα παιδιά του στο δέντρο τελεστών, ενώ αντίθετα οι υποχρεωτικές μεταβλητές καθορίζονται κατά βάση από το γονέα του. Εξαίρεση αποτελεί ο τελεστής της προβολής Project, για τον οποίο δεν μπορεί

[9]Ο πίνακας 5.1 αποτελεί προσαρμογή των συνόλων διαθέσιμων και υποχρεωτικών μεταβλητών για SPARQL τελεστές, τα οποία αναφέρονται στο [77].

Πίνακας 5.1: *Διαθέσιμες και υποχρεωτικές μεταβλητές για τους SPARQL τελεστές*

SPARQL τελεστής	Διαθέσιμες μεταβλητές	Υποχρεωτικές μεταβλητές
Project	$available(children(q))$	$var(q)$
Join/LeftJoin/Minus	$available(children(q))$	$(available(left(q)) \cap available(right(q)) \cup required(parent(q))$
Τελεστής-φύλλο	$var(q)$	$var(q) \cup required(parent(q))$
Άλλος τελεστής	$available(children(q))$	$var(q) \cup required(parent(q))$

κάποιος τελεστής - γονέας να θέσει επιπλέον περιορισμούς που αφορούν στις προβαλλόμενες μεταβλητές. Επίσης, καθώς η σημασιολογία των SPARQL τελεστών Join, LeftJoin και Minus, βασίζεται στην έννοια των συμβατών SPARQL αντιστοιχιών (ορισμός 5.3.6), για την αποτίμησή τους είναι απαραίτητη η αποτίμηση των κοινών μεταβλητών των παιδιών τους.

Οι υποχρεωτικές μεταβλητές ενός τελεστή χρησιμοποιούνται κατά την επεξεργασία αυτού από τον αλγόριθμο επανεγγραφής, προκειμένου να προβληθούν μόνο οι απαραίτητες – και όχι όλες οι διαθέσιμες – μεταβλητές στο κατασκευαζόμενο SQL μοντέλο. Οι τελικές απαραίτητες μεταβλητές είναι οι προβαλλόμενες μεταβλητές του SPARQL ερωτήματος. Οι διαθέσιμες μεταβλητές ενός τελεστή χρησιμοποιούνται κατά την επανεγγραφή, μόνο όταν δεν υπάρχει κατηγορηματικός τελεστής προβολής, όπως π.χ. στις περιπτώσεις SELECT * ερωτημάτων, οπότε υπονοείται η προβολή όλων των διαθέσιμων μεταβλητών.

Στη συνέχεια, ορίζουμε το **SQL μοντέλο**, το οποίο δημιουργείται και εμπλουτίζεται από τον αλγόριθμο επανεγγραφής, συγκεντρώνει όλη την απαραίτητη πληροφορία για την κατασκευή ενός SQL ερωτήματος και αποτελεί παραλλαγή του αντίστοιχου μοντέλου του [77]. Η προσέγγιση αυτή προσφέρει σαφή πλεονεκτήματα στο επίπεδο της υλοποίησης του αλγορίθμου σε αντίθεση με τη διατήρηση και επεξεργασία ενός κεντρικού SQL ερωτήματος. Τα συστατικά του SQL μοντέλου περιγράφονται στον πίνακα 5.2.

Το SQL μοντέλο, πέρα από τα συστατικά στοιχεία ενός SQL ερωτήματος, περιέχει και πληροφορίες που προκύπτουν κατά τη διάσχιση του SPARQL δέντρου και τις οποίες χρειάζεται ο αλγόριθμος επανεγγραφής. Μεταξύ αυτών είναι και το στοιχείο VarSources που περιέχει, για κάθε μεταβλητή του SPARQL ερωτήματος, μια λίστα με τους παραγωγούς κόμβου, οι οποίοι, εφαρμοζόμενοι στο τρέχον στιγμιότυπο της σχεσιακής ΒΔ, οδηγούν στην παραγωγή ενός RDF όρου που έχει συνδεθεί με τη συγκεκριμένη μεταβλητή.

Η ανάγκη να περιληφθούν πληροφορίες του αλγορίθμου στο SQL μοντέλο οφείλεται στην ιδιαιτερότητα της R2RML αντιστοιχίας, η οποία δεν αποτελεί 1:N αντιστοιχία μεταξύ στοιχείων του σχεσιακού σχήματος και προτύπων τετράδων, όπως υποθέτουν οι περισσότεροι αλγόριθμοι επανεγγραφής και μεταξύ αυτών και ο [77]. Αυτό οδηγεί αναπόφευκτα στην πιθανότητα παράλληλης δημιουργίας περισσότερων του ενός SQL μοντέλων κατά τη διάσχιση του SPARQL δέντρου, καθένα εκ των οποίων οφείλεται σε ένα μοναδικό συνδυασμό παραγωγών τετράδας. Αντίθετα, στο [77] η μορφή της αντιστοιχίας που υιοθετείται εξασφαλίζει την ύπαρξη ενός μόνο συνδυασμού παραγωγών τετράδας και συνεπώς μοναδικού SQL μοντέλου, οπότε οι βοηθητικές πληροφορίες μπορούν να δηλωθούν ως παγκόσμιες μεταβλητές. Στην περίπτωσή μας, το

Πίνακας 5.2: *Δομή του SQL μοντέλου*

Στοιχείο	**Περιγραφή**
Projections	Λίστα με προβαλλόμενες στήλες του SQL ερωτήματος, οι οποίες μπορεί να περιέχει και ψευδώνυμα που ορίζονται με τη λέξη `AS` (στοιχείο `SELECT`)
Tables	Λίστα με ελάχιστους λογικούς πίνακες που σχετίζονται με την παραγωγή υποχρεωτικών τετράδων (οι λογικοί πίνακες είναι πιθανό να δηλώνονται και με ψευδώνυμα)
OptTables	Σύνολο ζευγών {πρότυπο τετράδας, ελάχιστος λογικός πίνακας} για προαιρετικές τετράδες (ομοίως, οι λογικοί πίνακες μπορούν να δηλώνονται με ψευδώνυμα)
OptConditions	Σύνολο ζευγών {πρότυπο τετράδας, SQL συνθήκη συνένωσης} για προαιρετικές τετράδες
Where	Λίστα SQL συνθηκών επιλογής (πρόταση `WHERE`)
Distinct	Λογική τιμή που δηλώνει την απαλοιφή ίδιων αποτελεσμάτων (στοιχείο `DISTINCT`)
Order	Λίστα SQL συνθηκών διάταξης (στοιχείο `ORDER BY`)
Limit	Στοιχεία ορίου αποτελεσμάτων και μετατόπισης (`LIMIT` και `OFFSET`)
PostWhere	Λίστα συνθηκών επιλογής που χρησιμοποιούν συναρτήσεις εκτός του SQL προτύπου
PostOrder	Λίστα συνθηκών διάταξης που χρησιμοποιούν συναρτήσεις εκτός του SQL προτύπου
VarSources	Σύνολο {Μεταβλητή, Σύνολο παραγωγών κόμβου} με τις προελεύσεις των όρων που συνδέονται με μεταβλητές του SPARQL ερωτήματος

SQL μοντέλο μπορεί να θεωρηθεί ως *σύνοψη μιας πιθανής επανεγγραφής του SPARQL ερωτήματος σε SQL*.

Ο αλγόριθμος 6 δίνει σε αδρές γραμμές την παραγωγή ενός SQL ερωτήματος από ένα SQL μοντέλο. Οι συναρτήσεις Συνενωση_πινακων, Συζευξη_συνθηκων και Παραγωγη_προτασης_LimitOffset είναι βοηθητικές συναρτήσεις που παραλείπονται για οικονομία χώρου. Σε ό,τι αφορά στη συνάρτηση συνένωσης πινάκων, αυτή αρχικά συνενώνει εσωτερικά τους πίνακες που παράγουν υποχρεωτικές τετράδες (λίστα Tables του SQL μοντέλου) και στη συνέχεια, πραγματοποιεί αριστερή εξωτερική συνένωση με τους πίνακες που παράγουν προαιρετικές τετράδες (στοιχείο OptTables) με συνθήκες συνένωσης τις συνθήκες που αναφέρονται στο στοιχείο OptConditions του SQL μοντέλου.

Στη συνέχεια, ορίζουμε τις πράξεις της συνένωσης και της αριστερής συνένωσης δύο SQL μοντέλων, οι οποίες χρησιμοποιούνται από τον αλγόριθμο επανεγγραφής, ακολουθώντας την προσέγγιση του [77]. Η λογική των πράξεων αυτών θα γίνει φανερή κατά την περιγραφή του αλγορίθμου επεξεργασίας κάθε SPARQL τελεστή. Προς το παρόν, περιοριζόμαστε στην αναφορά ότι οι προβαλλόμενοι όροι του SQL ερωτήματος που περιγράφεται από ένα SQL μοντέλο συνδέονται άμεσα με SPARQL μεταβλητές, ενώ το αποτέλεσμα της εκτέλεσης του SQL ερωτήματος αποτελεί ένα σύνολο SPARQL αντιστοιχιών.

Η πράξη της **συνένωσης** περιγράφεται στον αλγόριθμο 7. Ο αλγόριθμος 7 δέχεται ως είσοδο, εκτός των δύο SQL μοντέλων, και τα σύνολα των προ-

Αλγόριθμος 6 *Κατασκευή SQL ερωτήματος*

Είσοδος: SQL μοντέλο *model*
Έξοδος: το SQL ερώτημα που αντιστοιχεί στο *model*

```
1: function Παραγωγη_SQL_ερωτηματος(model)
2:     if model.Distinct = true then
3:         sqlQuery ← "SELECT DISTINCT "+ model.Projections
4:     else
5:         sqlQuery ← "SELECT "+ model.Projections
6:     end if
7:     sqlQuery+ = " FROM "+Συνενωση_πινακων(model.Tables, model.OptTables, model.OptConditions)
8:     sqlQuery+ = " WHERE "+Συζευξη_συνθηκων(model.Where)
9:     sqlQuery+ = " ORDER BY "+model.Order
10:    sqlQuery+ = Παραγωγη_προτασης_LimitOffset(model.Limit)
11:    return sqlQuery
12: end function
```

αιρετικών μεταβλητών των δύο μοντέλων, δηλαδή τις μεταβλητές εκείνες για τις οποίες μπορεί να μην υπάρχει αντιστοιχία με κάποιον RDF όρο. Όπως θα δούμε και στα επόμενα, η πληροφορία αυτή αποθηκεύεται σε ένα SQL μοντέλο κατά τη διάσχιση του SPARQL δέντρου και ουσιαστικά πρόκειται για τις μεταβλητές εκείνες που εμφανίζονται *μόνο* στο δεξιό υποδέντρο μιας αριστερής συνένωσης. Η γνώση των προαιρετικών μεταβλητών για κάθε μοντέλο οδηγεί στην απλοποίηση του τελικού SQL ερωτήματος, οδηγώντας σε αποτελέσματα παρόμοια με αυτά των βελτιστοποιήσεων του [57].

Η συνένωση δύο SQL μοντέλων δεν αποφεύγει τη δημιουργία εμφωλευμένων ερωτημάτων, καθώς κατασκευάζεται ένα νέο SQL μοντέλο, το στοιχείο `FROM` του οποίου είναι μια συνένωση των υποερωτημάτων που αντιστοιχούν στα δύο μοντέλα. Οι συνθήκες συνένωσης βασίζονται στις κοινές προβαλλόμενες μεταβλητές των δύο όρων και στην έννοια των συμβατών SPARQL αντιστοιχιών (ορισμός 5.3.6). Στην πράξη, η συνένωση των δύο ερωτημάτων θα πρέπει να εξομοιώνει την πράξη της συνένωσης SPARQL αντιστοιχιών, όπως αυτή παρουσιάστηκε στην παράγραφο 5.3.1. Αυτό επιβάλλει την προσθήκη συνθηκών συνένωσης της μορφής "OR $alias_1.var$ IS NULL OR $alias_2.var$ IS NULL" για κάθε κοινή μεταβλητή *var* των δύο μοντέλων, εφόσον μια κενή μεταβλητή υπονοεί απουσία αυτής από μια SPARQL αντιστοιχία, γεγονός που δεν επηρεάζει τη συμβατότητα αυτής με κάποια άλλη. Εντούτοις, όπως προτείνεται και στο [57], οι παραπάνω διαζεύξεις μπορούν να αποφευχθούν αν είναι γνωστό ότι η *var* είναι σίγουρα δεσμευμένη σε κάποιο από τα δύο μοντέλα. Αυτός ο έλεγχος πραγματοποιείται στις γραμμές 11-16 του αλγορίθμου 7. Αντίστοιχα, όσον αφορά στις προβαλλόμενες μεταβλητές, η πλειονότητα των αλγορίθμων επανεγγραφής προβάλλουν εκφράσεις της μορφής "COALESCE($alias_1.var, alias_2.var$)", οι οποίες εξασφαλίζουν ότι θα προβληθεί ο όρος που δεν είναι κενός. Αν και πάλι όμως, είναι γνωστό ότι μια προβαλλόμενη μεταβλητή θα είναι πάντα δεσμευμένη («υποχρεωτική») σε ένα SQL μοντέλο, τότε επιλέγεται να προβληθεί αυτή. Αυτή η απλοποίηση πραγματοποιείται στις γραμμές 21-24. Τέλος, συνδυάζονται τα στοιχεία VarSources των δύο μοντέλων, έτσι ώστε το προκύπτον SQL μοντέλο να έχει ενημερωμένες προελεύσεις μεταβλητών.

Η πράξη της **αριστερής συνένωσης** δύο SQL μοντέλων (αλγόριθμος 8) είναι κατ' ουσία ίδια με την πράξη της συνένωσης, μόνο που σε αυτή αποφεύγεται η

Αλγόριθμος 7 *Συνένωση SQL μοντέλων (προσαρμογή από [77])*

Είσοδος: SQL μοντέλα $model_1, model_2$, σύνολα προαιρετικών μεταβλητών $optVars_1, optVars_2$
Έξοδος: ένα νέο SQL μοντέλο που αναπαριστά τη συνένωση των $model_1, model_2$

```
1: function ΣΥΝΕΝΩΣΗ(model_1, model_2, optVars_1, optVars_2)
2:     subquery_1 ← ΠΑΡΑΓΩΓΗ_SQL_ΕΡΩΤΗΜΑΤΟΣ(model_1)          # βλέπε αλγόριθμο 6
3:     subquery_2 ← ΠΑΡΑΓΩΓΗ_SQL_ΕΡΩΤΗΜΑΤΟΣ(model_2)
4:     alias_1 ← ψευδώνυμο για subquery_1
5:     alias_2 ← ψευδώνυμο για subquery_2
6:     projVars_1 ← προβαλλόμενοι όροι model_1
7:     projVars_2 ← προβαλλόμενοι όροι model_2
8:     commonVars ← projVars_1 ∩ projVars_2
9:     joinCond =" "
10:    if commonVars ≠ ∅ then
11:        for all var ∈ commonVars do
12:            joinCond += "alias_1.var = alias_2.var"
13:            if var ∈ optVars_1 then
14:                joinCond+=" OR alias_1.var IS NULL"
15:            end if
16:            if var ∈ optVars_2 then
17:                joinCond+=" OR alias_2.var IS NULL"
18:            end if
19:        end for
20:        table = subquery_1 AS alias_1 INNER JOIN subquery_2 AS alias_2 ON joinCond
21:    else
22:        table = subquery_1 AS alias_1 CROSS JOIN subquery_2 AS alias_2
23:    end if
24:    projVars = ∅
25:    for all var ∈ projVars_1 ∪ projVars_2 do
26:        if var ∉ optVars_1 then
27:            projVars+=alias_1.var
28:        else if var ∉ optVars_2 then
29:            projVars+=alias_2.var
30:        else
31:            projVars+=COALESCE(alias_1.var, alias_2.var)
32:        end if
33:    end for
34:    varSources ← Συνδύασε τα VarSources των model_1, model_2
35:    return SQLModel(table, projVars, varSources)
36: end function
```

εμφώλευση ερωτημάτων. Πιο συγκεκριμένα, οι βασικές διαφορές με την πράξη της συνένωσης είναι οι εξής:

α) το στοιχείο `FROM` του SQL ερωτήματος που περιγράφεται από το νέο SQL μοντέλο αποτελείται από τη συνένωση των στοιχείων Tables των δύο μοντέλων, αντί για τη συνένωση ολόκληρων ερωτημάτων (γραμμή 18)

β) το είδος της συνένωσης είναι αριστερή εξωτερική συνένωση

γ) η συνθήκη συνένωσης περιλαμβάνει επιπλέον το στοιχείο Where του δεύτερου μοντέλου και την SQL συνθήκη συνένωσης που (προαιρετικά) δέχεται ως είσοδο ο αλγόριθμος 8 (γραμμή 6)

δ) το στοιχείο Where του πρώτου μοντέλου ανατίθεται αυτούσιο στο νέο SQL μοντέλο (γραμμή 31)

Αλγόριθμος 8 *Αριστερή συνένωση SQL μοντέλων (με βάση το [77])*

Είσοδος: SQL μοντέλα $model_1, model_2$, σύνολα προαιρετικών μεταβλητών $optVars_1, optVars_2$, SQL έκφραση συνένωσης $expr$

Έξοδος: ένα νέο SQL μοντέλο που αναπαριστά την αριστερή συνένωση των $model_1, model_2$ με συνθήκη συνένωσης $expr$

```
1: function ΑΡΙΣΤΕΡΗ_ΣΥΝΕΝΩΣΗ(model1, model2, optVars1, optVars2, expr)
2:     projVars1 ← προβαλλόμενοι όροι model1
3:     projVars2 ← προβαλλόμενοι όροι model2
4:     commonVars ← projVars1 ∩ projVars2
5:     where2 ← model2.Where
6:     joinCond = where2 AND expr
7:     if commonVars ≠ ∅ then
8:         for all var ∈ commonVars do
9:             joinCond += "alias1.var = alias2.var"
10:            if var ∈ optVars1 then
11:                joinCond+=" OR alias1.var IS NULL"
12:            end if
13:            if var ∈ optVars2 then
14:                joinCond+=" OR alias2.var IS NULL"
15:            end if
16:        end for
17:    end if
18:    table = model1.Tables LEFT OUTER JOIN model2.Tables ON joinCond
19:    projVars = ∅
20:    for all var ∈ projVars1 ∪ projVars2 do
21:        if var ∉ optVars1 then
22:            projVars+=alias1.var
23:        else if var ∉ optVars2 then
24:            projVars+=alias2.var
25:        else
26:            projVars+=COALESCE(alias1.var, alias2.var)
27:        end if
28:    end for
29:    varSources ← Συνδύασε τα VarSources των model1, model2
30:    postWhere ← model1.PostWhere ∪ model2.PostWhere
31:    return SQLModel(table, projVars, model1.Where, postWhere, varSources)
32: end function
```

Έχοντας ορίσει τις έννοιες των υποχρεωτικών και διαθέσιμων μεταβλητών, καθώς και το SQL μοντέλο, προχωρούμε στην περιγραφή του αλγορίθμου διάσχισης ενός SPARQL-$\mathbb{DB}$ δέντρου τελεστών και επανεγγραφής του σε μια λίστα SQL ερωτημάτων, μέσω της δημιουργίας αντίστοιχων SQL μοντέλων. Στα επόμενα, εξετάζουμε κάθε τελεστή ξεχωριστά, ξεκινώντας από τους νέους τελεστές που εισάγει η SPARQL-$\mathbb{DB}$ άλγεβρα. Η αντιμετώπιση των υπόλοιπων SPARQL τελεστών βασίζεται σε μεγάλο βαθμό στο [77] και ως εκ τούτου αφιερώνεται μικρότερη έκταση σε αυτή.

5.3.3.1 BDP

Ο τελεστής BDP αποτελεί ένα πρότυπο βασικού συνόλου δεδομένων ή, ισοδύναμα, ένα σύνολο προτύπων τετράδας. Σύμφωνα με τον ορισμό 5.3.27, οι μέχρι σήμερα προταθέντες αλγόριθμοι επανεγγραφής αντιμετωπίζουν ένα τέτοιο σύνολο ως ένα συνδυασμό των πρωταρχικών SPARQL τελεστών TriplePattern,

Graph και Join, χωρίς να εξετάζουν προοπτικές απλοποίησης των παραγόμενων SQL ερωτημάτων.

Ο τρέχων αλγόριθμος αξιοποιεί την παρουσία κοινών μεταβλητών μεταξύ προτύπων τετράδας ενός προτύπου συνόλου δεδομένων, προκειμένου να εντοπιστούν *παραγωγοί τετράδας με κοινούς ελάχιστους πίνακες*. Η φιλοσοφία των απλοποιήσεων που επιτυγχάνει ο αλγόριθμος σκιαγραφήθηκε συνοπτικά στα εισαγωγικά παραδείγματα 5.1.1 και 5.1.2 και αναλύεται διεξοδικά στα επόμενα. Πιο συγκεκριμένα, αναζητούνται και αξιοποιούνται ομάδες προτύπων τετράδας με κοινό υποκείμενο, οι οποίες περιγράφουν ένα *πρότυπο γράφου με μορφή αστέρα* καθώς και *ομάδες προτύπων με συμφωνία αντικειμένου-υποκειμένου*.

Το πρώτο βήμα στη διαδικασία επανεγγραφής ενός προτύπου βασικού συνόλου δεδομένων *bdp* αποτελεί η εύρεση όλων των πιθανών προελεύσεων των προτύπων τετράδων qp_i που το απαρτίζουν ή με άλλα λόγια, των πινάκων και στηλών της ΒΔ από τις οποίες μπορεί να παραχθεί μια τετράδα συμβατή με τα πρότυπα qp_i. Η προέλευση ενός προτύπου τετράδας κωδικοποιείται σε έναν *παραγωγό τετράδας* (ορισμός 5.3.22), ο οποίος περιέχει τον ελάχιστο λογικό πίνακα από τον οποίο θα προέλθει μια τετράδα και 4 αντιστοιχίες όρου, οι οποίες καθορίζουν το μηχανισμό παραγωγής καθενός από τους όρους της τετράδας. Όπως αναφέρθηκε και προηγουμένως, ένα πρότυπο τετράδας διαθέτει περισσότερους του ενός παραγωγούς τετράδας, οι οποίοι πρέπει να ληφθούν υπόψη στην αποτίμηση των μεταβλητών του[10]. Αυτό σημαίνει ότι για την επανεγγραφή του *bdp*, θα πρέπει να θεωρηθούν *όλοι οι πιθανοί συνδυασμοί παραγωγών τετράδας για καθένα από τα πρότυπα τετράδας* qp_i. Ονομάζουμε έναν τέτοιο συνδυασμό παραγωγών τετράδας **πλάνο αποτίμησης**. Κάθε πλάνο αποτίμησης θα οδηγήσει σε μια διαφορετική επανεγγραφή του *bdp* ή ισοδύναμα, σε ένα διαφορετικό SQL μοντέλο. Αν π.χ. το *bdp* αποτελείται από τρία πρότυπα τετράδας, καθένα εκ των οποίων διαθέτει 2 συμβατούς παραγωγούς τετράδας, ο αλγόριθμος θα θεωρήσει 8 διαφορετικούς συνδυασμούς και θα κατασκευάσει το πολύ 8 SQL ερωτήματα[11], η ένωση των οποίων θα είναι σημασιολογικά ισοδύναμη με ένα SPARQL ερώτημα που περιέχει το *bdp*. Η γενική εικόνα της διαδικασίας απεικονίζεται στον αλγόριθμο 9.

Η συνάρτηση Βρες_συμβατους_παραγωγους_τετραδας (γραμμή 4) αρχικά χρησιμοποιεί τους ορισμούς 5.3.19 – 5.3.21 για να βρει αντιστοιχίες τριάδων συμβατές με ένα πρότυπο τετράδας και από τις συμβατές αντιστοιχίες όρου που περιέχονται σε αυτές, κατασκευάζονται αντίστοιχοι παραγωγοί τετράδας. Επίσης, χρησιμοποιεί τον ορισμό 5.3.25 προκειμένου να απορρίψει παραγωγούς τετράδας αν αυτοί περιέχουν ασύμβατες αντιστοιχίες όρων οι οποίες αναφέρονται στην ίδια μεταβλητή εντός ενός προτύπου τετράδας. Η συνάρτηση Συνδυασε_παραγωγους_τετραδων (γραμμή 9) από την πλευρά της επιστρέφει όλους τους δυνατούς συνδυασμούς των ευρεθέντων παραγωγών τετράδας, αναθέτοντας έναν σε κάθε πρότυπο τετράδας. Τέλος, η συνάρτηση Αποτιμηση_συνολου_τετραδων (γραμμή 11), η οποία είναι και η πιο ενδιαφέρουσα

[10] Αρκεί να φανταστούμε το γενικό πρότυπο $\langle ?s, ?p, ?o, ?g \rangle$, το οποίο περιλαμβάνει όλες τις τετράδες ενός συνόλου δεδομένων, μπορεί να προέλθει από οποιαδήποτε αντιστοιχία τριάδων ενός R2RML γράφου αντιστοιχίας και συνεπώς, για την αποτίμηση των μεταβλητών του χρειάζεται η εφαρμογή όλων των αντιστοιχιών τριάδων στο στιγμιότυπο της ΒΔ.

[11] Όπως θα φανεί στη συνέχεια, κάποιο πλάνο αποτίμησης μπορεί να περιέχει ασύμβατους παραγωγούς τετράδας, με συνέπεια να μην οδηγεί στην κατασκευή ενός SQL ερωτήματος.

Αλγόριθμος 9 *Αλγόριθμος επανεγγραφής BDP τελεστή*

Είσοδος: BDP τελεστής *bdp*, R2RML γράφος αντιστοιχίας *r2rmlMap*
Έξοδος: μια λίστα SQL μοντέλων *sqlModels*

```
1: function ΕΠΑΝΕΓΓΡΑΦΗ_BDP(bdp, r2rmlMap)
2:     sqlModels ← ∅
3:     for all qp_i ∈ bdp do
4:         qgen(qp_i) ← ΒΡΕΣ_ΣΥΜΒΑΤΟΥΣ_ΠΑΡΑΓΩΓΟΥΣ_ΤΕΤΡΑΔΑΣ(qp_i, r2rmlMap)
5:     end for
6:     if ∃qp_i : qgen(qp_i) = ∅ then
7:         return sqlModels
8:     else
9:         evaluationPlans ← ΣΥΝΔΥΑΣΕ_ΠΑΡΑΓΩΓΟΥΣ_ΤΕΤΡΑΔΩΝ(qgen)  # σύνολο ζευγών {qp_i, qgen_i}
10:        for all plan ∈ evaluationPlans do
11:            sqlModels+ = ΑΠΟΤΙΜΗΣΗ_ΣΥΝΟΛΟΥ_ΤΕΤΡΑΔΩΝ(bdp, plan)      # βλέπε αλγόριθμο 10
12:        end for
13:    end if
14:    return sqlModels
15: end function
```

και αναλύεται στη συνέχεια, επιστρέφει ένα μοναδικό SQL μοντέλο για ένα δεδομένο συνδυασμό παραγωγών τετράδας. Σημειώνουμε ότι, προκειμένου να υπάρξει SPARQL αντιστοιχία για τις μεταβλητές του *bdp*, θα πρέπει να βρεθεί τουλάχιστον ένας παραγωγός τετράδας για κάθε πρότυπο τετράδας, διαφορετικά θεωρείται ότι δεν υπάρχει ένα βασικό σύνολο δεδομένων που να ταιριάζει με το *bdp*, με αποτέλεσμα να επιστρέφεται μια κενή λίστα (γραμμές 6-7).

Η διαδικασία αποτίμησης ενός συνόλου προτύπων τετράδων δεν αρκείται στη μετάφραση των προτύπων σε ισοδύναμες SQL εκφράσεις, αλλά προσπαθεί να εκμεταλλευτεί την παρουσία κοινών μεταβλητών μεταξύ αυτών των προτύπων τετράδων. Το περίγραμμα της συγκεκριμένης διαδικασίας, η οποία δέχεται ως ορίσματα ένα σύνολο προτύπων τετράδων και έναν πίνακα συσχέτισης των προτύπων αυτών με αντίστοιχο παραγωγό τετράδας, φαίνεται στον αλγόριθμο 10.

Η συνάρτηση ΑΠΟΤΙΜΗΣΗ_ΠΡΟΤΥΠΟΥ_ΤΕΤΡΑΔΑΣ (γραμμή 7) αποτελεί την καρδιά του αλγορίθμου, καθώς είναι αυτή που δημιουργεί και ανανεώνει τα συστατικά στοιχεία ενός SQL μοντέλου, επεξεργαζόμενη ένα πρότυπο τετράδας τη φορά. Η συγκεκριμένη συνάρτηση απεικονίζεται στον αλγόριθμο 12 και θα αναλυθεί στη συνέχεια. Η επεξεργασία όλων των προτύπων τετράδας ακολουθείται από έναν έλεγχο ασυμβατότητας των προελεύσεων των όρων που έχουν ανατεθεί στις αναφερόμενες μεταβλητές (γραμμή 9). Η συνάρτηση ΥΠΑΡΧΟΥΝ_ΑΣΥΜΒΑΤΟΤΗΤΕΣ, βασιζόμενη στον ορισμό 5.3.26, ελέγχει αν υπάρχει κάποια μεταβλητή που υπάρχει σε κάποιο πρότυπο τετράδων και έχει δεσμευτεί σε ασύμβατους παραγωγούς κόμβου. Αν ισχύει κάτι τέτοιο, αυτό σημαίνει ότι το τρέχον πλάνο αποτίμησης δεν μπορεί να οδηγήσει σε ένα σημασιολογικά έγκυρο SQL μοντέλο και η συνολική διαδικασία θα συνεχίσει με το επόμενο πλάνο αποτίμησης στον αλγόριθμο 9.

Η αποτίμηση του συνόλου προτύπων τετράδων QP ολοκληρώνεται με την προσθήκη απαραίτητων IS NOT NULL συνθηκών στο στοιχείο Where του SQL μοντέλου (γραμμή 12) και στη δημιουργία της λίστας προβολών Projections του SQL μοντέλου (γραμμή 13). Η συνάρτηση ΠΡΟΣΘΗΚΗ_NOT_NULL_ΣΥΝΘΗΚΩΝ εξετά-

Αλγόριθμος 10 *Αποτίμηση συνόλου προτύπων τετράδων BDP τελεστή*

Είσοδος: Σύνολο προτύπων τετράδων QP, πλάνο αποτίμησης *evaluationPlan* # σύνολο ζευγών $\{qp_i, qgen_i\}$
Έξοδος: ένα SQL μοντέλο

```
1: function ΑΠΟΤΙΜΗΣΗ_ΣΥΝΟΛΟΥ_ΤΕΤΡΑΔΩΝ(QP, evaluationPlan)
2:     varSources ← ∅                                  # αρχικοποίηση μεταβλητών
3:     where ← ∅
4:     postWhere ← ∅
5:     from ← ∅
6:     for all qp_i ∈ QP do
7:         ΑΠΟΤΙΜΗΣΗ_ΠΡΟΤΥΠΟΥ_ΤΕΤΡΑΔΑΣ(qp_i, evaluationPlan, varSources, where, postWhere, from)
                                  # ενημέρωση μεταβλητών, βλέπε αλγόριθμο 12
8:     end for
9:     if ΥΠΑΡΧΟΥΝ_ΑΣΥΜΒΑΤΟΤΗΤΕΣ(varSources) then
10:        return null
11:    end if
12:    ΠΡΟΣΘΗΚΗ_NOT_NULL_ΣΥΝΘΗΚΩΝ(where, varSources, QP)
13:    projections ← ΔΗΜΙΟΥΡΓΙΑ_ΠΡΟΒΟΛΩΝ(varSources)   # βλέπε αλγόριθμο 11
14:    return SQLModel(from, projections, where, postWhere, varSources)
15: end function
```

ζει την κατασκευασμένη λίστα συνθηκών επιλογής, καθώς και το στοιχείο προέλευσης μεταβλητών *varSources* και προσθέτει μια συνθήκη IS NOT NULL για κάθε στήλη πίνακα που αναφέρεται στην αντιστοιχία όρου ενός παραγωγού κόμβου και η οποία δε συμμετέχει στη συνθήκη επιλογής. Η φιλοσοφία της συγκεκριμένης συνάρτησης περιγράφεται στο επόμενο παράδειγμα.

Παράδειγμα 5.3.4. Έστω η σχέση Review και το στιγμιότυπό της, όπως αυτά ορίζονται στο παράδειγμα 5.1.1, καθώς και η αντιστοιχία τριάδων που ορίστηκε στο παράδειγμα 5.3.1. Όπως είδαμε και στο παράδειγμα 5.1.1, το SPARQL ερώτημα:

```
SELECT ?review ?prod ?rating
WHERE {
        ?review ex:isReviewFor ?prod.
        ?review ex:hasRating ?rating
      }
```

το οποίο εκφράζεται σε όρους της SPARQL-$\mathbb{DB}$ άλγεβρας ως: `BDP({ ⟨?review, ex:isReviewFor, ?prod ⟩, ⟨?review, ex:hasRating, ?rating⟩})`, είναι σημασιολογικά ισοδύναμο με το SQL ερώτημα:

```
SELECT nr, product, rating
FROM Review
WHERE nr IS NOT NULL AND product IS NOT NULL AND
      rating IS NOT NULL
```

Ο λόγος για την προσθήκη των IS NOT NULL περιορισμών στη συνθήκη επιλογής του ερωτήματος οφείλεται στις εξής παρατηρήσεις:

α) σύμφωνα με την προδιαγραφή της R2RML, ένας RDF όρος μπορεί να παραχθεί μονάχα από μη κενές τιμές της ΒΔ και

β) η σημασιολογία της αποτίμησης προτύπων βασικού γράφου της SPARQL επιβάλλει το ταίριασμα όλων των προτύπων τριάδας που αποτελούν ένα

βασικό πρότυπο γράφου και κατά συνέπεια, επιβάλλει τη δέσμευση όλων των μεταβλητών του σε RDF όρους.

Με άλλα λόγια, προκειμένου να μπορεί κάθε αποτέλεσμα του παραπάνω SQL ερωτήματος να ερμηνευτεί (υιοθετώντας μια χαλαρή θεώρηση) ως μια SPARQL αντιστοιχία, θα πρέπει να εξασφαλιστεί ότι η «προέλευση» (ισοδύναμα, ο παραγωγός κόμβου) κάθε μεταβλητής θα είναι μη κενή. Υπενθυμίζουμε ότι οι παραγωγοί κόμβου των μεταβλητών `?prod` και `?rating` περιέχουν column-valued αντιστοιχίες όρου (τιμής από στήλη) που αναφέρονται στις στήλες product και rating αντίστοιχα, ενώ ο παραγωγός κόμβου της `?review` περιέχει μια template-valued αντιστοιχία όρου (τιμής από πρότυπο), η οποία αναφέρεται στη στήλη nr. Αυτό δικαιολογεί και το σκεπτικό της προσθήκης των τριών περιορισμών στη συνθήκη επιλογής του SQL ερωτήματος.

Αξίζει να τονιστεί ότι οι IS NOT NULL περιορισμοί εισάγονται μόνο σε περίπτωση στηλών που δεν αναφέρονται στη συνθήκη επιλογής του ήδη κατασκευασμένου SQL ερωτήματος. Αν για παράδειγμα, το παραπάνω SQL ερώτημα περιείχε τη συνθήκη rating = 6, δεν θα ήταν απαραίτητη η προσθήκη ενός IS NOT NULL περιορισμού για τη στήλη rating. Η συνάρτηση Προσθηκη_not_null_συνθηκων, η οποία αναφέρεται στον αλγόριθμο 10, αναλαμβάνει την προσθήκη τέτοιων περιορισμών στη συνθήκη επιλογής του παραχθέντος SQL ερωτήματος.

Σε ό,τι αφορά στην κατασκευή της λίστας προβολών του SQL ερωτήματος, ακολουθούμε την προσέγγιση του [77], στο οποίο για κάθε SPARQL μεταβλητή, προβάλλονται, εκτός από τη στήλη με τις τιμές της μεταβλητής, και στήλες με επιπλέον πληροφορίες που απαιτούνται για την κατασκευή των RDF όρων της SPARQL λύσης (παράγραφος 5.3.4). Οι πληροφορίες αυτές είναι το είδος του RDF όρου (IRI, κενός κόμβος ή λεκτικό), ο τύπος δεδομένων και η γλωσσική ετικέτα. Οι δύο τελευταίοι όροι αναφέρονται μονάχα σε λεκτικά, αλλά είναι απαραίτητο να περιληφθούν στη λίστα προβολών, προκειμένου να εξασφαλιστεί η συμβατότητα των παραγόμενων SQL ερωτημάτων ως προς τη σχεσιακή πράξη της ένωσης. Η συνάρτηση Δημιουργια_προβολων απεικονίζεται στον αλγόριθμο 11.

Αρχικά, για κάθε μεταβλητή, επιλέγεται τυχαία ένας παραγωγός κόμβου για αυτή από τον πίνακα προέλευσης που έχει συσταθεί σε προηγούμενο βήμα, με προτεραιότητα σε παραγωγούς κόμβου σταθερής τιμής και τιμής από στήλη, έτσι ώστε το παραγόμενο SQL ερώτημα να έχει πιο απλή μορφή (γραμμή 3). Αξίζει να σημειωθεί ότι η επιλογή παραγωγού κόμβου δεν έχει επίδραση στο τελικό αποτέλεσμα εφόσον μια μεταβλητή ενός προτύπου βασικού συνόλου δεδομένων έχει δεσμευτεί σε συγκεκριμένο RDF όρο και αρκεί ο υπολογισμός του από έναν εκ των παραγωγών κόμβου που έχουν εντοπιστεί. Υπενθυμίζουμε ότι, σε αυτό το σημείο του αλγορίθμου, έχει εξασφαλιστεί ότι οι παραγωγοί κόμβου για κάθε μεταβλητή δεν είναι ασύμβατοι μεταξύ τους (συνάρτηση Υπαρχουν_ασυμβατοτητες).

Ένα σημείο που επίσης χρήζει διευκρίνισης είναι η περίπτωση της προβολής μιας έκφρασης προτύπου (γραμμή 10). Σύμφωνα με την R2RML, η τιμή ενός RDF όρου που προέρχεται από μια έκφραση προτύπου θεωρεί ότι οι τιμές που προέρχονται από τη ΒΔ υφίστανται πρώτα «εκατοστιαία κωδικοποίηση»[12].

[12]Η διαδικασία της εκατοστιαίας κωδικοποίησης (percent encoding) δίνεται στο http://tools.ietf.org/html/rfc3986.

Αλγόριθμος 11 *Δημιουργία λίστας προβολών SQL ερωτήματος*

Είσοδος: Πίνακας προέλευσης μεταβλητών *varSources* # σύνολο ζευγών $\{var_i, NGen\}$
Έξοδος: Λίστα προβαλλόμενων όρων *projections*

```
1: function ΔΗΜΙΟΥΡΓΙΑ_ΠΡΟΒΟΛΩΝ(varSources) projections ← ∅
2:     for all var ∈ required(BDP) do          # υποχρεωτικές μεταβλητές BDP τελεστή
3:         ngen ← ΔΙΑΛΕΞΕ_ΠΑΡΑΓΩΓΟ_ΚΟΜΒΟΥ(var, varSources)
4:         tmap ← ngen.termMap
5:         if tmap είναι σταθερής τιμής then
6:             projections+ = "tmap.constant AS varName, tmap.type AS varName_termType"
7:         else if tmap είναι τιμής από στήλη then
8:             projections+ = "tmap.column AS varName, tmap.type AS varName_termType"
9:         else if tmap είναι τιμής από πρότυπο then
10:            projections+ = "tmap.template AS varName, tmap.type AS varName_termType"
11:        end if
12:        if tmap.type είναι Literal then
13:            projections+ = "tmap.language AS varName_language, tmap.datatype AS var-
    Name_datatype"
14:        else
15:            projections+ = "NULL AS varName_language, NULL AS varName_datatype"
16:        end if
17:    end for
18:    return projections
19: end function
```

Καθώς δεν μπορεί να γίνει η υπόθεση ότι στη γενική περίπτωση, θα υπάρχει στη ΒΔ μια αποθηκευμένη διαδικασία που θα πραγματοποιεί εκατοστιαία κωδικοποίηση, σε τέτοιες περιπτώσεις προβάλλεται κατάλληλη συμβολοσειρά που σηματοδοτεί τη χρήση έκφρασης προτύπου, έτσι ώστε η κωδικοποίηση να γίνει κατά το στάδιο της κατασκευής της SPARQL λύσης (παράγραφος 5.3.4).

Ο αλγόριθμος 10 ολοκληρώνεται με την κατασκευή ενός νέου SQL μοντέλου από τα συστατικά στοιχεία που δημιουργήθηκαν κατά την εκτέλεσή του. Μέχρι τώρα, αναφερθήκαμε μονάχα στη δημιουργία της λίστας προβολών (στοιχείο Projections του SQL μοντέλου) καθώς και στον εμπλουτισμό των συνθηκών επιλογής (στοιχείο Where). Μένει να δούμε πώς δημιουργούνται τα υπόλοιπα στοιχεία του SQL μοντέλου, διαδικασία που πραγματοποιείται στη συνάρτηση ΑΠΟΤΙΜΗΣΗ_ΠΡΟΤΥΠΟΥ_ΤΕΤΡΑΔΑΣ και η οποία παρουσιάζεται στον αλγόριθμο 12.

Η αποτίμηση ενός προτύπου τετράδας, όπως φαίνεται από τον αλγόριθμο 12 συνίσταται στην προσθήκη συνθηκών στη λίστα Where και στην προσθήκη ελάχιστων λογικών πινάκων στη λίστα Tables ενός SQL μοντέλου. Συνοπτικά, οι συνθήκες που προστίθενται μπορούν να οφείλονται είτε σε κάποιο σταθερό όρο του προτύπου τετράδας (γραμμές 7-18) είτε στο γεγονός ότι μια μεταβλητή εμφανίζεται σε περισσότερες από μία θέσεις του προτύπου (γραμμή 19). Η εμφάνιση μιας μεταβλητής σε μοναδική θέση δεν επηρεάζει τη λίστα συνθηκών. Οι σταθεροί όροι του προτύπου τετράδας αντιμετωπίζονται από τον αλγόριθμο 13, ενώ ο αλγόριθμος 14 χειρίζεται πολλαπλές εμφανίσεις μεταβλητών.

Η συνάρτηση ΠΡΟΣΘΗΚΗ_ΣΥΝΘΗΚΩΝ_ΣΤΑΘΕΡΟΥ_ΟΡΟΥ χρησιμοποιεί βοηθητικές συναρτήσεις για την εύρεση ενός σχετικού IRI (γραμμή 7), την εκατοστιαία αποκωδικοποίηση μιας τιμής (γραμμή 12), την αναγνώριση των στηλών που συμμετέχουν σε μια έκφραση προτύπου (γραμμές 10, 25) καθώς και τον υπολογισμό των τιμών που χρειάζεται να αντικατασταθούν σε ένα πρότυπο προκειμένου

Αλγόριθμος 12 *Αποτίμηση προτύπου τετράδας*

Είσοδος: Πρότυπο τετράδας *qp*, πλάνο αποτίμησης *evaluationPlan*, προέλευση μεταβλητών *varSources*, λίστα συνθηκών επιλογής *where*, λίστα εκτός-SQL συνθηκών επιλογής *postWhere*, λίστα ελάχιστων λογικών πινάκων *from*
Έξοδος: Ενημερωμένα *evaluationPlan*, *varSources*, *where*, *postWhere*, *from*

```
1: function ΑΠΟΤΙΜΗΣΗ_ΠΡΟΤΥΠΟΥ_ΤΕΤΡΑΔΑΣ(qp, evaluationPlan, varSources, where, postWhere, from)
2:     qgen = evaluationPlan(qp)                          # ανάκτηση παραγωγού τετράδας
3:     if var(qp) = ∅ then
4:         ΠΡΟΣΘΗΚΗ_EXISTS_ΥΠΟΕΡΩΤΗΜΑΤΟΣ(where)
5:     end if
6:     ΑΝΑΘΕΣΗ_ΠΙΝΑΚΑ(qp, evaluationPlan, from)
7:     if qp.s ∉ V then                                   # V το σύνολο των μεταβλητών
8:         ΠΡΟΣΘΗΚΗ_ΣΥΝΘΗΚΩΝ_ΣΤΑΘΕΡΟΥ_ΟΡΟΥ(qp.s, qgen.subjMap, where)   # βλέπε αλγόριθμο 13
9:     end if
10:    if qp.p ∉ V then
11:        ΠΡΟΣΘΗΚΗ_ΣΥΝΘΗΚΩΝ_ΣΤΑΘΕΡΟΥ_ΟΡΟΥ(qp.p, qgen.predMap, where)
12:    end if
13:    if qp.o ∉ V then
14:        ΠΡΟΣΘΗΚΗ_ΣΥΝΘΗΚΩΝ_ΣΤΑΘΕΡΟΥ_ΟΡΟΥ(qp.o, qgen.objMap, where)
15:    end if
16:    if qp.g ∉ V then
17:        ΠΡΟΣΘΗΚΗ_ΣΥΝΘΗΚΩΝ_ΣΤΑΘΕΡΟΥ_ΟΡΟΥ(qp.g, qgen.graphMap, where)
18:    end if
19:    ΠΡΟΣΘΗΚΗ_ΣΥΝΘΗΚΩΝ_ΜΕΤΑΒΛΗΤΩΝ(qp, qgen, where, postWhere)   # βλέπε αλγόριθμο 14
20: end function
```

να ληφθεί μια συγκεκριμένη τιμή (γραμμές 11, 26). Χρήζει αναφοράς η περίπτωση του προκαθορισμένου γράφου, που συνεπάγεται την προσθήκη μιας λογικής *διάζευξης* συνθηκών (γραμμή 3) για καθέναν από τους προκαθορισμένους γράφους του SPARQL ερωτήματος – εφόσον αρκεί η ζητούμενη τετράδα να ανήκει σε έναν από αυτούς – καθώς και η αντιμετώπιση ενός παραγωγού κόμβου με αναφέρουσα αντιστοιχία αντικειμένου (γραμμή 18), η οποία ανάγεται στην εξέταση της αντιστοιχίας υποκειμένου της αντιστοιχίας-γονέα της τελευταίας. Επίσης, ο αλγόριθμος 13 προσθέτει στη λίστα συνθηκών και πιθανές αντίστροφες εκφράσεις που έχουν οριστεί στην αντιστοιχία όρου του παραγωγού κόμβου (γραμμή 32). Η ανάλυση της τελευταίας διαδικασίας παραλείπεται για λόγους οικονομίας της τρέχουσας παρουσίασης.

Αξίζει επίσης να σημειωθεί η σκόπιμη παράλειψη της περίπτωσης κενών κόμβων. Σύμφωνα με τη θεώρηση της SPARQL, η παρουσία κενών κόμβων σε ένα πρότυπο γράφου ισοδυναμεί με την παρουσία μεταβλητών, ενώ και η σημασιολογία της αποτίμησης προτύπων γράφου υποθέτει ότι το όνομα ενός κενού κόμβου διαφέρει μεταξύ του πραγματικού γράφου και του γράφου στον οποίο η SPARQL προσπαθεί να ταιριάξει ένα πρότυπο [93]. Συνεπώς, θα ήταν εσφαλμένη η χρησιμοποίηση του αναγνωριστικού ενός κενού κόμβου για την δημιουργία μιας συνθήκης επιλογής και ως εκ τούτου, παραλείπεται από τον αλγόριθμο 13.

Παράδειγμα 5.3.5. Έστω οι πίνακες Product και Producer του BSBM σχήματος (σχήμα 5.3) με τη στήλη product του πρώτου να αναφέρεται στο πρωτεύον κλειδί nr του δεύτερου και με ενδεικτικά στιγμιότυπα τα παρακάτω:

Αλγόριθμος 13 *Προσθήκη συνθηκών για σταθερό όρο προτύπου τετράδας*

Είσοδος: RDF όρος *node*, αντιστοιχία όρου *tmap*, λίστα συνθηκών επιλογής *where*
Έξοδος: Ενημερωμένη λίστα *where*

```
1: function ΠΡΟΣΘΗΚΗ_ΣΥΝΘΗΚΩΝ_ΣΤΑΘΕΡΟΥ_ΟΡΟΥ(node, tmap, where)
2:     if node = DG then                                  # DG ο προκαθορισμένος γράφος
3:         where+ = ⋁_{i=1}^{n} ΠΡΟΣΘΗΚΗ_ΣΥΝΘΗΚΩΝ_ΣΤΑΘΕΡΟΥ_ΟΡΟΥ(defGraphs_i, tmap, where)
4:     end if
5:     if (node ∈ I ∩ tmap δεν είναι αναφέρουσα αντιστοιχία αντικειμένου) then
6:         if tmap είναι τιμής από στήλη then
7:             where+ = (tmap.column = node ∨ tmap.column = getRelative(node, baseIRI))
8:         end if
9:         if tmap είναι τιμής από πρότυπο then
10:            for all column ∈ getColumns(tmap.template) do
11:                for all value ∈ getPlaceholderValues(node, tmap.template) do
12:                    where+ = (column = percentDecode(value))
13:                end for
14:            end for
15:        end if
16:    end if
17:    if (node ∈ I ∩ tmap είναι αναφέρουσα αντιστοιχία αντικειμένου) then
18:        ΠΡΟΣΘΗΚΗ_ΣΥΝΘΗΚΩΝ_ΣΤΑΘΕΡΟΥ_ΟΡΟΥ(node, tmap.parentMap.subjMap)
19:    end if
20:    if node ∈ L then                                   # αν ο RDF όρος είναι λεκτικό
21:        if tmap είναι τιμής από στήλη then
22:            where+ = (tmap.column = node)
23:        end if
24:        if tmap είναι τιμής από πρότυπο then
25:            for all column ∈ getColumns(tmap.template) do
26:                for all value ∈ getPlaceholderValues(node, tmap.template) do
27:                    where+ = (column = value)
28:                end for
29:            end for
30:        end if
31:    end if
32:    where+ = ΠΡΟΣΘΗΚΗ_ΑΝΤΙΣΤΡΟΦΗΣ_ΕΚΦΡΑΣΗΣ(tmap, node)
33: end function
```

Product

nr	label	producer	propertyNum1	...
1	ProductA	2	134	...
2	ProductB	3	108	...
3	ProductC	2	145	...
4	ProductD	1	96	...

Producer

nr	label	country	...
1	Producer1	DE	...
2	Producer2	US	...
3	Producer3	GR	...
4	Producer4	RU	...

Έστω επίσης δύο αντιστοιχίες τριάδων $productMap = \langle$ Product, $subjMap_1$, $\{poMap_1, poMap_2, poMap_3\} \rangle$ και $producerMap = \langle$ Producer, $subjMap_2$, $\{poMap_4\}\rangle$, με την αντιστοιχία $poMap_1$ να περιέχει μια αναφέρουσα αντιστοιχία αντικειμένου προς την *producerMap*. Πιο συγκεκριμένα:

$subjMap_1.template$ = http://example.org/Product{nr}
$poMap_1.predMap.constant$ = bsbm:producer
$poMap_1.objMap.parentMap = producerMap$
$poMap_1.objMap.joins$ = {($Product.producer$, $Producer.nr$)}
$poMap_2.predMap.constant$ = rdfs:label
$poMap_2.objMap.column = label$
$poMap_3.predMap.constant$ = bsbm:propertyNumeric1
$poMap_3.objMap.column = propertyNum1$
$subjMap_2.template$ = http://example.org/Producer{nr}
$poMap_4.predMap.constant$ = bsbm:country
$poMap_4.objMap.column = country$

Θεωρούμε το πρότυπο τετράδας qp = ⟨?s, bsbm:producer, ex:Producer4⟩, DG⟩. Το πλάνο αποτίμησης που έχει εντοπιστεί για το συγκεκριμένο πρότυπο τετράδας (αλγόριθμος 9, γραμμή 4) περιλαμβάνει μονάχα τον παραγωγό τετράδας $qgen$ = ⟨$table$, $subjMap_1$, $poMap_1.predMap$, $poMap_1.objMap$, $gMap$⟩ όπου $gMap$ η αντιστοιχία προκαθορισμένου γράφου και $table$ ο ελάχιστος λογικός πίνακας ο οποίος έχει ανατεθεί στο συγκεκριμένο πρότυπο (αλγόριθμος 12, γραμμή 6), σύμφωνα με τη διαδικασία που θα περιγραφεί στην επόμενη παράγραφο.

Ο αλγόριθμος 13 προσθέτει μία ή περισσότερες συνθήκες επιλογής για κάθε σταθερό όρο του qp, λαμβάνοντας τις σχετικές αντιστοιχίες όρου του $qgen$ (αλγόριθμος 12, γραμμές 7-18). Αναλυτικότερα, για το κατηγόρημα bsbm:producer δεν θα προστεθεί κάποια συνθήκη, εφόσον η $poMap_1.predMap$ είναι σταθερής τιμής, ενώ για το αντικείμενο http://example.org/Producer4 η συνάρτηση Προσθηκη_συνθηκων_σταθερου_οροϒ θα κληθεί με όρισμα τη $subjMap_2$ της $producer Map$, αφού η τελευταία αποτελεί την αντιστοιχία-γονέα της $poMap_1$ (αλγόριθμος 13, γραμμή 18). Η συνθήκη που τελικά θα προστεθεί στη λίστα συνθηκών επιλογής λόγω της template-valued αντιστοιχίας $subjMap_2$ θα είναι η "producer.nr=4", με την τιμή 4 να προκύπτει από το ταίριασμα του IRI αντικειμένου με την έκφραση προτύπου της $subjMap_2$.

Αλγόριθμος 14 *Προσθήκη συνθηκών για μεταβλητές προτύπου τετράδας*

Είσοδος: Πρότυπο τετράδας qp, παραγωγός τετράδας $qgen$, λίστα συνθηκών επιλογής $where$, λίστα εκτός-SQL συνθηκών επιλογής postWhere
Έξοδος: Ενημερωμένες λίστες $where$, $postWhere$

```
1: function Προσθηκη_συνθηκων_μεταβλητων(qp, qgen, where, postWhere)
2:     for all variable ∈ var(qp) do
3:         if variable εμφανίζεται σε ≥ 1 θέσεις pos then
4:             NGen ← Φτιαξε_παραγωγουσ_κομβου(qgen, pos)
5:             {where, postWhere}+ = Συνθηκεσ_ισοτητασ_παραγωγων_κομβου(NGen)
6:         end if
7:     end for
8: end function
```

Ο αλγόριθμος 14 προσθέτει επιπλέον συνθήκες επιλογής όταν ένα πρότυπο τετράδας εμφανίζει την ίδια μεταβλητή σε παραπάνω της μίας θέσης. Οι συν-

θήκες αυτές περιγράφουν την ισότητα των RDF όρων που προκύπτουν από τις σχετικές αντιστοιχίες όρου, με τη συνάρτηση στη γραμμή 5 να εξετάζει όλες τις πιθανές περιπτώσεις ισοδυναμίας των τελευταίων. Για παράδειγμα, δεδομένου ενός κατάλληλου παραγωγού τετράδας *qgen*, ένα πρότυπο τετράδας της μορφής $\langle ?s, ?p, ?s, \boldsymbol{DG} \rangle$ θα προκαλέσει την προσθήκη συνθηκών που εξασφαλίζουν την ισοδυναμία της αντιστοιχίας υποκειμένου *qgen.subjMap* με την αντιστοιχία αντικειμένου *qgen.objMap*. Έτσι, αν π.χ. η *qgen.subjMap* είναι αντιστοιχία τιμής από πρότυπο και η *qgen.objMap* αντιστοιχία τιμής από στήλη, θα προστεθεί η συνθήκη *qgen.subjMap.template* = *qgen.objMap.column*. Πρέπει να σημειωθεί ότι συνθήκες που εμπλέκουν αντιστοιχίες τιμής από πρότυπο αναγκαστικά περιέχουν μια συνάρτηση που εκτελεί εκατοστιαία κωδικοποίηση, με αποτέλεσμα αυτές να προστίθενται στη λίστα με τις εκτός-SQL συνθήκες.

Επιστρέφοντας στον αλγόριθμο 12 και στην αποτίμηση ενός προτύπου τετράδας, σημειώνουμε ότι, ένα πρότυπο τετράδας χωρίς μεταβλητές θα οδηγήσει στην προσθήκη ενός EXISTS υποερωτήματος στη λίστα συνθηκών επιλογής (γραμμή 4), καθώς χρειάζεται να εξασφαλιστεί η παρουσία της συγκεκριμένης τετράδας στο (νοητό) σύνολο δεδομένων. Το συγκεκριμένο υποερώτημα δημιουργείται ακολουθώντας τη λογική του αλγορίθμου 13, με την προσθήκη κατάλληλων συνθηκών για κάθε όρο της τετράδας. Με αυτόν τον τρόπο, ο αλγόριθμος επιτυγχάνει το ίδιο αποτέλεσμα με άλλους αλγορίθμους, αποφεύγοντας όμως τη συνένωση με το λογικό πίνακα του παραγωγού της συγκεκριμένης τετράδας.

Μια από τις πιο σημαντικές διαδικασίες ολόκληρου του αλγορίθμου είναι η ανάθεση ενός λογικού πίνακα στον παραγωγό ενός προτύπου τετράδας (γραμμή 6), η οποία γίνεται με τέτοιο τρόπο ώστε να μεγιστοποιήσει την επαναχρησιμοποίηση λογικών πινάκων, αποφεύγοντας όπου είναι δυνατόν τις συνενώσεις αυτών. Η κεντρική ιδέα της διαδικασίας αυτής αναλύεται στην επόμενη παράγραφο.

Ανάθεση λογικού πίνακα σε πρότυπο τετράδας Ανατρέχοντας στο εισαγωγικό παράδειγμα 5.1.1, παρατηρήσαμε ότι ένα SQL ερώτημα το οποίο είναι ισοδύναμο με τα SQL ερωτήματα που παράγουν άλλοι αλγόριθμοι επανεγγραφής ήταν το Q_4:

```
PREFIX ex: <http://example.org>
SELECT ?review ?prod ?rating
WHERE {
(tp1):   ?review ex:isReviewFor ?prod.
(tp2):   ?review ex:hasRating ?rating
      }
```

```
Q4: SELECT nr, product, rating
    FROM Review
    WHERE nr IS NOT NULL AND
          product IS NOT NULL AND
          rating IS NOT NULL
```

Η αποφυγή της συνένωσης του πίνακα Review με τον εαυτό του είναι δυνατή λόγω: α) του γεγονότος ότι τα πρότυπα τριάδας tp_1, tp_2 προέρχονται από τον ίδιο λογικό πίνακα και β) της παρουσίας της μεταβλητής `?review` στη θέση υποκειμένου των tp_1 και tp_2. Γενικεύοντας τη συγκεκριμένη παρατήρηση, ακολουθούμε την εξής στρατηγική για τον καθορισμό του συγκεκριμένου ελάχιστου λογικού πίνακα (με ή χωρίς ψευδώνυμο) από τον οποίο προέρχεται ένα πρότυπο τετράδας qp_i που ανήκει σε ένα πρότυπο βασικού συνόλου δεδομένων *bdp*:

1. βρίσκουμε τα πρότυπα qp_j του bdp, των οποίων ο παραγωγός τετράδας – για το τρέχον πλάνο αποτίμησης – έχει τον ίδιο λογικό πίνακα με αυτόν του παραγωγού του qp_i και

2. αν υπάρχει ένα qp_j με κοινές μεταβλητές με το qp_i σε συγκεκριμένες θέσεις, τότε ο ελάχιστος λογικός πίνακας του παραγωγού του qp_j ανατίθεται και στον παραγωγό του qp_i. Διαφορετικά, ένας λογικός πίνακας με νέο ψευδώνυμο ανατίθεται στον παραγωγό του qp_i.

Όπως είδαμε και παραπάνω, ένας τρόπος συσχέτισης προτύπων που μπορεί να οδηγήσει στην αποφυγή συνένωσης είναι η ύπαρξη κοινής μεταβλητής-υποκειμένου ή, με άλλα λόγια, η ύπαρξη μιας $s \bowtie s$ συνένωσης. Ένας άλλος τρόπος συσχέτισης που χρειάζεται να ληφθεί υπόψη είναι αυτός μεταξύ δύο προτύπων qp_1, qp_2 για τα οποία $qp_1.o = qp_2.s$ ($s \bowtie o$ συνένωση) και ο ελάχιστος λογικός πίνακας του qp_1 αποτελεί συνένωση δύο βασικών πινάκων[13]. Στα επόμενα, χρησιμοποιούμε τους πίνακες Product και Producer του BSBM σχήματος καθώς και τις αντιστοιχίες τριάδων $productMap$ και $producerMap$ που ορίστηκαν στο παράδειγμα 5.3.5, προκειμένου αν εξηγήσουμε τις διαφορετικές περιπτώσεις αλληλεπίδρασης προτύπων τετράδας και των λογικών πινάκων από τα οποία μπορούν αυτά να προέλθουν.

Περίπτωση 1. Συμφωνία υποκειμένων. Έστω δύο πρότυπα τετράδων $qp_1 = \langle ?s, ?p_1, ?o_1, ?g_1 \rangle$, $qp_2 = \langle ?s, ?p_2, ?o_2, ?g_2 \rangle$, τα οποία αποτελούν μέρος ενός προτύπου βασικού συνόλου δεδομένων bdp. Τότε, διακρίνουμε τις εξής συμμετρικές περιπτώσεις για τους παραγωγούς τετράδας $qgen_1$ και $qgen_2$ των qp_1, qp_2 αντίστοιχα:

1. Ο $qgen_1$ περιέχει το λογικό πίνακα $table_1$ με ψευδώνυμο $alias_1$. Τότε, επαναχρησιμοποιούμε τον πίνακα $alias_1$ στον $qgen_2$ αν και μόνο αν ο λογικός πίνακας του $qgen_2$ είναι τύπου $table_1$. Αναλυτικότερα:
   α. αν ο $qgen_2$ δεν περιέχει μια αναφέρουσα αντιστοιχία αντικειμένου, ο λογικός πίνακας του $qgen_2$ τίθεται ίσος με $alias_1$.
   β. αν ο $qgen_2$ περιέχει μια αναφέρουσα αντιστοιχία αντικειμένου $roMap$ με μη κενή λίστα συνενώσεων $joins$, ο λογικός πίνακας του $qgen_2$ τίθεται ίσος με $alias_1$ INNER JOIN $table_2$ ON $joins$, όπου $table_2$ ο λογικός πίνακας της αντιστοιχίας-γονέα της $roMap$.

2. Ο $qgen_1$ περιέχει έναν ελάχιστο λογικό πίνακα της μορφής $table_1$ AS $alias_1$ INNER JOIN $table_2$ AS $alias_2$ ON {συνθήκες συνένωσης}. Αυτό σημαίνει ότι ο $qgen_1$ περιέχει μια αναφέρουσα αντιστοιχία αντικειμένου $roMap_1$ με μη κενή λίστα συνενώσεων και υποθέτουμε ότι $table_2$ είναι ο λογικός πίνακας της αντιστοιχίας-γονέα της $roMap_1$[14]. Επαναχρησιμοποιούμε τον πίνακα $alias_1$ στον $qgen_2$ αν και μόνο αν ο λογικός πίνακας του $qgen_2$ είναι τύπου $table_1$. Αναλυτικότερα, όπως και πριν:

[13] Υπενθυμίζουμε ότι αυτό συμβαίνει μόνο όταν ο παραγωγός του qp_1 περιέχει μια αναφέρουσα αντιστοιχία αντικειμένου με μη κενή λίστα συνενώσεων.

[14] Στα επόμενα, γενικεύουμε την υιοθέτηση αυτής της σύμβασης για σύνθετους ελάχιστους λογικούς πίνακες. Αυτό σημαίνει ότι ο δεξιός πίνακας θα είναι ο λογικός πίνακας της αντιστοιχίας-γονέα μιας αναφέρουσας αντιστοιχίας αντικειμένου, εκτός και αν αναφέρεται κατηγορηματικά κάτι διαφορετικό.

α. αν ο $qgen_2$ δεν περιέχει μια αναφέρουσα αντιστοιχία αντικειμένου, ο λογικός πίνακας του $qgen_2$ τίθεται ίσος με $alias_1$.
β. αν ο $qgen_2$ περιέχει μια αναφέρουσα αντιστοιχία αντικειμένου $roMap_2$ με μη κενή λίστα συνενώσεων *joins*, ο λογικός πίνακας του $qgen_2$ τίθεται ίσος με $alias_1$ INNER JOIN $table_3$ ON *joins*, όπου $table_3$ ο λογικός πίνακας της αντιστοιχίας-γονέα της $roMap_2$.

Παράδειγμα 5.3.6. Θεωρούμε τις αντιστοιχίες του παραδείγματος 5.3.5 καθώς και τα ακόλουθα τμήματα δυο προτύπων βασικού συνόλου δεδομένων:

bdp₁: { ⋯
qp_{1a}: `?prod rdfs:label ?label DG.`
qp_{1b}: `?prod bsbm:propertyNum1 ?value DG.`
⋯ }

bdp₂: { ⋯
qp_{2a}: `?prod bsbm:producer ?producer DG.`
qp_{2b}: `?prod rdfs:label ?label DG.`
⋯ }

Στην περίπτωση του bdp_1, οι αρχικοί παραγωγοί $qgen_{1a}$, $qgen_{1b}$ των προτύπων qp_{1a}, qp_{1b} αντίστοιχα διαθέτουν τον ίδιο λογικό πίνακα Product. Αν υποτεθεί ότι κατά τη διάρκεια της διαδικασίας αποτίμησης, έχει ανατεθεί στον $qgen_{1a}$ ένας λογικός πίνακας με ψευδώνυμο Product3, τότε ο ίδιος πίνακας θα ανατεθεί και στον παραγωγό $qgen_{1b}$, σύμφωνα με το σημείο 1α παραπάνω.

Στην περίπτωση του bdp_2, λόγω της αναφέρουσας αντιστοιχίας αντικειμένου $poMap_1.objMap$ και των συνθηκών συνένωσης που αυτή περιέχει, υποθέτουμε ότι στον παραγωγό $qgen_{2a}$ έχει ανατεθεί ένας ελάχιστος λογικός πίνακας της μορφής Product AS Product5 INNER JOIN Producer AS Producer2 ON Product5.producer = Producer2.nr. Η συμφωνία υποκειμένου των qp_{2a}, qp_{2b} θα οδηγήσει στην ανάθεση του λογικού πίνακα Product5 στον παραγωγο $qgen_{2b}$ σύμφωνα με το σημείο 2α.

Η αναγνώριση και αξιοποίηση $s \bowtie s$ συνενώσεων μεταξύ προτύπων τετράδων ουσιαστικά είναι ισοδύναμη με την αναγνώριση προτύπων γράφου σε μορφή αστέρα. Οι συγκεκριμένες μορφές προτύπων συναντώνται αρκετά συχνά σε πρακτικά SPARQL ερωτήματα, καθώς επιτρέπουν την ανάκτηση πολλαπλών πληροφοριών για την ίδια οντότητα και αυτός είναι και ένας από τους λόγους που έχουν εξεταστεί αρκετές φορές στη βιβλιογραφία στο πλαίσιο βελτιστοποίησης της εκτέλεσης SPARQL ερωτημάτων (π.χ. [104, 190, 195]).

Περίπτωση 2. Συμφωνία υποκειμένου-αντικειμένου. Έστω δύο πρότυπα τετράδων $qp_1 = \langle ?s_1, ?p_1, ?s, ?g_1 \rangle$, $qp_2 = \langle ?s, ?p_2, ?o_2, ?g_2 \rangle$, τα οποία αποτελούν μέρος ενός προτύπου βασικού συνόλου δεδομένων *bdp* και έστω $qgen_1$, $qgen_2$ οι παραγωγοί τους για ένα δεδομένο πλάνο αποτίμησης. Μπορούμε να αξιοποιήσουμε την $s \bowtie o$ συνένωση μεταξύ του αντικειμένου του qp_1 και του υποκειμένου του qp_2 για να αποφύγουμε μια περιττή συνένωση πινάκων, αν και μόνο αν ο $qgen_1$ περιέχει μια αναφέρουσα αντιστοιχία αντικειμένου *roMap* με μη κενή λίστα συνενώσεων *joins* και η αντιστοιχία υποκειμένου του $qgen_2$ είναι ίδια με την αντιστοιχία υποκειμένου της αντιστοιχίας-γονέα της *roMap*.

Σε αυτή την περίπτωση, αν έχει ανατεθεί στον $qgen_1$ ένας λογικός πίνακας της μορφής $table_1$ AS $alias_1$ INNER JOIN $table_2$ AS $alias_2$ ON *joins*, όπου *table*2 ο λογικός πίνακας της αντιστοιχίας-γονέα της *roMap*, τότε θα ανατεθεί ο λογικός πίνακας $table_2$ AS $alias_2$ στον παραγωγό $qgen_2$.

Ανάλογη θα είναι και η συμμετρική περίπτωση, όπου η $qgen_2$ διαθέτει μια αναφέρουσα αντιστοιχία αντικειμένου που είναι σύμφωνη με την αντιστοιχία υποκειμένου του παραγωγού $qgen_1$.

Παράδειγμα 5.3.7. Θεωρούμε και πάλι τις αντιστοιχίες του παραδείγματος 5.3.5 και το ακόλουθο τμήμα του προτύπου βασικού συνόλου δεδομένων:

bdp$_3$: `{ ···`
qp_{3a}: `?prod bsbm:producer ?producer DG.`
qp_{3b}: `?producer bsbm:country ?country DG.`
`··· }`

Σύμφωνα με τα προηγούμενα, αν υποθέσουμε ότι στον παραγωγό $qgen_{3a}$ του qp_{3a} έχει ανατεθεί ο ελάχιστος λογικός πίνακας Product AS Product3 INNER JOIN Producer AS Producer1 ON Product3.producer = Producer1.nr, λόγω της αναφέρουσας αντιστοιχίας αντικειμένου $poMap_1.objMap$, τότε στον παραγωγό $qgen_{3b}$ του qp_{3b} θα ανατεθεί ο λογικός πίνακας Producer1.

Σε αυτό το σημείο, οφείλουμε να σημειώσουμε ότι οι συγκεκριμένες βελτιστοποιήσεις αποφυγής συνενώσεων θα οδηγούν σε ορθά, σύμφωνα με τη σημασιολογία της SPARQL, αποτελέσματα όταν η αντιστοιχία υποκειμένου (της τρέχουσας αντιστοιχίας τριάδων, στην περίπτωση μιας $s \bowtie s$ συνένωσης, ή της αντιστοιχίας-γονέα στην περίπτωση μιας $s \bowtie o$ συνένωσης) εγγυάται την παραγωγή διαφορετικού RDF όρου για κάθε πλειάδα του λογικού πίνακα ή ισοδύναμα, όταν ο συνδυασμός των στηλών που συμμετέχουν στην αντιστοιχία υποκειμένου αποτελεί κλειδί του λογικού πίνακα.

Παραδείγματος χάριν, θεωρούμε τον πίνακα ProductFeatureProduct του BSBM σχήματος και τη σχετική αντιστοιχία τριάδων $pfpmap = \langle$ Productfeatureproduct, $subjMap_3$, $\{poMap_5\}\rangle$, όπου:

$subjMap_3.template$ = http://example.org/Product{*product*}
$poMap_5.predMap.constant$ = bsbm:productFeature
$poMap_5.objMap.template$ = http://example.org/ProductFeature{*productfeature*}

Ο πίνακας Productfeatureproduct είναι ένας τυπικός πίνακας M:N συσχέτισης μεταξύ των πινάκων Product και Productfeature και το πρωτεύον κλειδί του περιλαμβάνει και τις δύο στήλες του. Γίνεται φανερό ότι η προτεινόμενη βελτιστοποίηση δεν μπορεί να εφαρμοστεί π.χ. στα ακόλουθα πρότυπα τετράδων, παρά την $s \bowtie s$ συνένωση που παρατηρείται:

```
?product bsbm:productFeature bsbm-inst:ProductFeature43883.
?product bsbm:productFeature bsbm-inst:ProductFeature7746
```

Δοκιμάζοντας την εφαρμογή των όσων περιγράφηκαν προηγουμένως, θα καταλήγαμε σε ένα SQL ερώτημα που θα είχε περίπου ως εξής:

```
SELECT product FROM productfeatureproduct
WHERE productfeature = 43883 AND productfeature = 7746
```

το οποίο φυσικά δεν είναι ορθό και δεν επιστρέφει κάποιο αποτέλεσμα. Αυτό ισχύει επειδή η στήλη product, η οποία είναι η μόνη που συμμετέχει στην αντιστοιχία υποκειμένου $subjMap_3$, δεν είναι κλειδί του πίνακα Productfeatureproduct, με αποτέλεσμα ένα δεδομένο IRI να μπορεί να παραχθεί από περισσότερες της μίας πλειάδες του εν λόγω πίνακα. Σε τέτοιες περιπτώσεις, *η συνένωση του λογικού πίνακα με τον εαυτό του δεν μπορεί να αποφευχθεί*, όπως φαίνεται και από το επόμενο παράδειγμα.

Παράδειγμα 5.3.8. Έστω το SPARQL ερώτημα του παραδείγματος 5.3.3 και η ισοδύναμη SPARQL-$\mathbb{DB}$ έκφραση που εμφανίζεται στο ίδιο παράδειγμα. Σύμφωνα με τα παραπάνω, το πρότυπο βασικού συνόλου δεδομένων bdp_3 που αναφέρεται στο συγκεκριμένο SPARQL ερώτημα:

```
bdp3: {
      ?product bsbm:productFeature bsbm-inst:ProductFeature43883.
      ?product bsbm:productFeature bsbm-inst:ProductFeature7746
      }
```

θα επανεγγραφεί στο ακόλουθο SQL ερώτημα:

```
SELECT 'http://example.org/Product$PERC_ENCODE('||pfp3.product||')$' AS product,
'IRI' AS product_termType, NULL AS product_language, NULL AS product_datatype
FROM ProductFeatureProduct AS pfp3 INNER JOIN ProductFeatureProduct AS pfp4 ON
pfp3.product = pfp4.product
WHERE pfp3.productFeature = '43883' AND pfp4.productFeature = '7746' AND
pfp3.product IS NOT NULL
```

όπου PERC_ENCODE μια συνάρτηση εκατοστιαίας κωδικοποίησης. Σημειώνουμε την προβολή 4 όρων με τις απαραίτητες πληροφορίες για τη δημιουργία των RDF όρων που θα δεσμευτούν στη μεταβλητή ?product, ενώ τονίζουμε ότι η συνθήκη συνένωσης του πίνακα ProductFeatureProduct με τον εαυτό του κανονικά αποτελεί μέρος της λίστας PostWhere του SQL μοντέλου, καθώς περιλαμβάνει την εκτός-SQL συνάρτηση PERC_ENCODE.

5.3.3.2 BDPLeftJoin

Ο BDPLeftJoin τελεστής (ορισμός 5.3.28) ισοδυναμεί με διαδοχικές αριστερές συνενώσεις ενός προτύπου βασικού συνόλου δεδομένων με – φιλτραρισμένα ή μη – πρότυπα τετράδων. Όπως αναφέρθηκε και στην παράγραφο 5.3.2, αριστερές συνενώσεις με ένα πρότυπο τετράδας προσφέρονται για την απαλοιφή συνενώσεων πινάκων στο τελικό SQL ερώτημα, σε αντίθεση με την περίπτωση αριστερών συνενώσεων με πρότυπα βασικού συνόλου δεδομένων, όπου δεν μπορεί να συμβεί κάτι τέτοιο. Ο BDPLeftJoin τελεστής μπορεί να θεωρηθεί μια επέκταση του BDP τελεστή με μια λίστα προτύπων τετράδων bdp_2και μια αντίστοιχη λίστα SPARQL λογικών εκφράσεων $optExprs$, γεγονός που ευνοεί την επαναχρησιμοποίηση των αλγορίθμων 9 – 14 με τις κατάλληλες προσαρμογές για το χειρισμό των προαιρετικών προτύπων τετράδων.

Στη συνέχεια, περιγράφουμε συνοπτικά τις προσθήκες που είναι απαραίτητες στους προαναφερθέντες αλγορίθμους για την επεξεργασία ενός BDPLeftJoin τελεστή. Αποφεύγουμε να δώσουμε τη νέα μορφή των αλγορίθμων προκειμένου να μην αυξήσουμε άσκοπα την πολυπλοκότητα της παρουσίασης.

Αρχικά, ο αλγόριθμος 9 μεταβάλλεται ώστε, εκτός από τους παραγωγούς τετράδας του υποχρεωτικού προτύπου συνόλου δεδομένων, να υπολογίσει κατά τον ίδιο τρόπο και τους αντίστοιχους παραγωγούς των προτύπων του bdp_2. Ο αλγόριθμος 10, αμέσως μετά την επεξεργασία των υποχρεωτικών προτύπων τετράδων (γραμμή 12), επεξεργάζεται τα πρότυπα τετράδας του bdp_2 με μια τροποποιημένη εκδοχή του αλγορίθμου 12, η οποία ενημερώνει τα ήδη δημιουργημένα συστατικά στοιχεία του SQL μοντέλου και δημιουργεί επιπλέον τα στοιχεία OptTables και OptConditions. Η μόνη ουσιαστική διαφορά του αλγορίθμου αποτίμησης προτύπου τετράδας (αλγόριθμος 12) για προαιρετικά πρότυπα είναι η αντιμετώπιση των συνθηκών που προκύπτουν ως συνθηκών αριστερής συνένωσης αντί για συνθηκών επιλογής, με αποτέλεσμα να εμπλουτίζουν το στοιχείο OptConditions του SQL μοντέλου, αντί το στοιχείο Where. Αυτή η διαφοροποίηση είναι σύμφωνη με την προσέγγιση του [77] και την πράξη της αριστερής συνένωσης SQL μοντέλων (αλγόριθμος 8) που παρουσιάστηκε προηγουμένως. Το ίδιο ισχύει και για τις SPARQL λογικές εκφράσεις $optExprs$, οι οποίες επίσης μεταφράζονται σε ισοδύναμες SQL συνθήκες, με λογική παρόμοια με αυτή του τελεστή Filter (παράγραφος 5.3.3.7).

Μια σημαντική διαφοροποίηση αφορά στον υπολογισμό των συνθηκών που αναφέρονται στο σταθερό όρο ενός προτύπου τετράδας, αλλά και σε αυτές που αναφέρονται στην παρουσία της ίδιας μεταβλητής σε παραπάνω της μίας θέσεις. Δεδομένου του ότι τα πρότυπα τετράδας που εξετάζονται είναι προαιρετικά, οι μεταβλητές που περιέχονται σε αυτά δεν είναι απαραίτητο να δεσμεύονται σε μια τιμή, αλλά μπορεί να μένουν αδέσμευτες. Αυτό οδηγεί στην επαύξηση των παραγόμενων συνθηκών με διαζεύξεις της μορφής "OR column IS NULL", για κάθε στήλη που συμμετέχει σε μια σχετική αντιστοιχία όρου.

Παράδειγμα 5.3.9. Αν στο παράδειγμα 5.3.5, υποθέσουμε ότι το απλό πρότυπο τετράδας $qp_2 = \langle ?s, \text{bsbm:producer}, \text{ex:Producer4}\rangle, DG\rangle$ είναι προαιρετικό, τότε θα προστεθεί η συνθήκη "producer.nr=4 OR producer.nr IS NULL" ως συνθήκη συνένωσης του πίνακα producer με τον ελάχιστο λογικό πίνακα που αντιστοιχεί στον παραγωγό ενός υποχρεωτικού προτύπου qp_1.

Αυτό θα ισχύσει βέβαια μόνο στην περίπτωση που οι παραγωγοί των δύο προτύπων έχουν διαφορετικούς λογικούς πίνακες. Διαφορετικά, όπως θα δούμε στη συνέχεια, η εν λόγω συνθήκη αποτελεί συνθήκη επιλογής (στοιχείο Where) στο παραγόμενο SQL μοντέλο.

Η σημαντικότερη διαφορά στην επεξεργασία ενός BDPLeftJoin τελεστή σε σύγκριση με έναν BDP τελεστή αφορά στη λογική της ανάθεσης λογικών πινάκων στους παραγωγούς των προαιρετικών προτύπων τετράδας. Στην περίπτωση του BDPLeftJoin τελεστή, εκτός από το είδος της συνένωσης που παρατηρείται μεταξύ των μεταβλητών δύο προτύπων qp_1, qp_2 ($s \bowtie s$ ή $s \bowtie o$ συνένωση), σημαντικό ρόλο παίζει και το σύνολο (υποχρεωτικών ή προαιρετικών) προτύπων στο οποίο ανήκουν τα qp_1, qp_2. Επιλέγουμε και εδώ τη διάκριση σε δύο περιπτώσεις, ανάλογα με το είδος της συνένωσης μεταξύ των qp_1 και qp_2.

Περίπτωση 1. Συμφωνία υποκειμένων. Έστω δύο πρότυπα τετράδων $qp_1 = \langle ?s, ?p_1, ?o_1, ?g_1\rangle$, $qp_2 = \langle ?s, ?p_2, ?o_2, ?g_2\rangle$, οι οποίες αποτελούν μέρος ενός

τελεστή BDPLeftJoin(bdp_1, bdp_2, $optExprs$) με $qp_2 \in bdp_2$. Τότε διακρίνουμε τις εξής περιπτώσεις για τους παραγωγούς $qgen_1$ και $qgen_2$ των προτύπων qp_1, qp_2 και το σύνολο στο οποίο ανήκει η qp_1:

1. Αν $qp_1 \in bdp_1$, η ανάθεση ενός ελάχιστου λογικού πίνακα στον $qgen_2$ θα ακολουθήσει τις 4 εναλλακτικές της περίπτωσης 1 της παραγράφου 5.3.3.1, με μόνη διαφορά την αντικατάσταση της εσωτερικής συνένωσης με αριστερή εξωτερική συνένωση.

2. Αν $qp_1 \in bdp_2$ και δεν υπάρχει κάποιος $qgen_i \in bdp_1$ που να μοιράζεται το λογικό του πίνακα με αυτόν του qp_1 (περιπτώσεις 1 και 2, σημεία 1), τότε η ανάθεση ενός ελάχιστου λογικού πίνακα στον $qgen_2$ θα ακολουθήσει αυτούσιες τις 4 εναλλακτικές της περίπτωσης 1 της παραγράφου 5.3.3.1.

3. Αν $qp_1 \in bdp_2$ και υπάρχει κάποιος $qgen_i \in bdp_1$ που να μοιράζεται το λογικό του πίνακα με αυτόν του qp_1 (περιπτώσεις 1 και 2, σημεία 1), τότε διακρίνουμε τις εξής περιπτώσεις:

   α. Ο $qgen_1$ περιέχει το λογικό πίνακα $table_1$ με ψευδώνυμο $alias_1$. Τότε, μπορούμε να επαναχρησιμοποιήσουμε τον πίνακα $alias_1$ στον $qgen_2$, αν και μόνο αν ο λογικός πίνακας του $qgen_2$ είναι τύπου $table_1$. Η ανάθεση λογικού πίνακα στον $qgen_2$ θα ακολουθήσει τις εναλλακτικές της περίπτωσης 1, σημείο 1 της παραγράφου 5.3.3.1.

   β. Ο $qgen_1$ περιέχει έναν ελάχιστο λογικό πίνακα της μορφής $table_1$ AS $alias_1$ LEFT OUTER JOIN $table_2$ AS $alias_2$ ON {συνθήκες συνένωσης}, μετά από εφαρμογή του σημείου 1. Τότε, μπορούμε να επαναχρησιμοποιήσουμε το λογικό πίνακα $alias_2$ στον $qgen_2$ αν και μόνο αν ο λογικός πίνακας του $qgen_2$ είναι τύπου $table_2$. Σε αυτή την τελευταία περίπτωση και υποθέτοντας ότι ο $qgen_1$ περιέχει μια αναφέρουσα αντιστοιχία αντικειμένου $roMap_1$, με $table_1$ την αντιστοιχία-γονέα της $roMap_1$:

      i. αν ο $qgen_2$ δεν περιέχει αναφέρουσα αντιστοιχία αντικειμένου, ο λογικός πίνακας του $qgen_2$ τίθεται ίσος με $alias_2$.

      ii. αν ο $qgen_2$ περιέχει μια αναφέρουσα αντιστοιχία αντικειμένου $roMap_2$ με μη κενή λίστα συνενώσεων $joins$, ο λογικός πίνακας του $qgen_2$ τίθεται ίσος με $alias_2$ INNER JOIN $table3$ ON $joins$, όπου $table_3$ ο λογικός πίνακας της αντιστοιχίας-γονέα της $roMap_2$.

Οι λογικοί πίνακες που ανατίθενται στους παραγωγούς προαιρετικών προτύπων συνδυάζονται από τη συνάρτηση ΣΥΝΕΝΩΣΗ_ΠΙΝΑΚΩΝ στον αλγόριθμο 6, προκειμένου να δημιουργηθεί το FROM τμήμα ενός SQL ερωτήματος. Εν συντομία, η συγκεκριμένη συνάρτηση πραγματοποιεί αριστερή εξωτερική συνένωση του λογικού πίνακα που θα προκύψει από το συνδυασμό της λίστας Tables με καθένα από τα στοιχεία της λίστας OptTables του SQL μοντέλου. Σημειώνεται ότι οι εσωτερικές συνενώσεις που έχουν ανατεθεί σε παραγωγούς προαιρετικών προτύπων (π.χ. σημεία 2, 3α παραπάνω) θα μετατραπούν σε αριστερές εξωτερικές συνενώσεις, καθώς η συγκεκριμένη ανάθεση χρησιμοποιείται αποκλειστικά για τη διάκριση των περιπτώσεων εκείνων όπου επαναχρησιμοποιείται ο λογικός πίνακας ενός παραγωγού υποχρεωτικού προτύπου.

Παράδειγμα 5.3.10. Θεωρούμε τις αντιστοιχίες του παραδείγματος 5.3.5 καθώς και τα ακόλουθα τμήματα προτύπων συνόλου δεδομένων:

```
dp1: { ···
qp1a:   ?prod rdfs:label ?label DG.
qp1b:   OPTIONAL {?prod bsbm:propertyNum1 ?value1 DG }.
        ··· }

dp2: { ···
qp2a:   OPTIONAL{ ?prod rdfs:label ?label DG }.
qp2b:   OPTIONAL{ ?prod bsbm:propertyNum1 ?value1 DG }.
        ··· }

dp3: { ···
qp3a:   ?producer bsbm:country ?country DG.
        ···
qp3b:   OPTIONAL{ ?prod bsbm:producer ?producer DG }.
qp3c:   OPTIONAL{ ?prod rdfs:label ?label DG }.
        ··· }
```

Στην περίπτωση του dp_1, οι αρχικοί παραγωγοί $qgen_{1a}$, $qgen_{1b}$ των qp_{1a}, qp_{1b} περιέχουν τον ίδιο λογικό πίνακα Product. Αν υποτεθεί ότι κατά τη διάρκεια της διαδικασίας αποτίμησης, έχει ανατεθεί στον $qgen_{1a}$ ένας λογικός πίνακας με ψευδώνυμο Product3, τότε ο ίδιος πίνακας θα ανατεθεί και στον $qgen_{1b}$, σύμφωνα με το σημείο 1 παραπάνω.

Στην περίπτωση του dp_2, το qp_{2a} αποτελεί πλέον προαιρετικό πρότυπο. Αν υποθέσουμε ότι δεν υπάρχει κάποιος παραγωγός υποχρεωτικού προτύπου με ίδιο λογικό πίνακα (δηλαδή τον Product) με τον $qgen_{2a}$, τότε ακολουθούμε το σημείο 2 παραπάνω και, αν ο λογικός πίνακας του $qgen_{2a}$ είναι ο Product3, ο ίδιος λογικός πίνακας ανατίθεται και στον $qgen_{2b}$.

Τέλος, το dp_3 αναφέρεται στην περίπτωση που περιγράφεται από το σημείο 3β, εφόσον, λόγω μιας $s \bowtie o$ συνένωσης μεταξύ των προτύπων qp_{3a}, qp_{3b} και όπως θα δούμε στη συνέχεια, στον παραγωγό $qgen_{3b}$ ανατίθεται ένας λογικός πίνακας της μορφής Producer1 LEFT OUTER JOIN Product5 ON Producer1.nr = Product5.producer. Λόγω της $s \bowtie s$ συνένωσης μεταξύ των qp_{3b}, qp_{3c}, στον $qgen_{3c}$ θα ανατεθεί ο λογικός πίνακας Product5.

Περίπτωση 2. Συμφωνία υποκειμένου-αντικειμένου. Έστω δύο πρότυπα τετράδων $qp_1 = \langle ?s_1, ?p_1, ?s, ?g_1 \rangle$, $qp_2 = \langle ?s, ?p_2, ?o_2, ?g_2 \rangle$, οι οποίες αποτελούν μέρος ενός τελεστή BDPLeftJoin(bdp_1, bdp_2, $optExprs$) με $qp_2 \in bdp_2$. Τότε διακρίνουμε τις εξής περιπτώσεις για τους παραγωγούς $qgen_1$ και $qgen_2$ των προτύπων qp_1, qp_2 και το σύνολο στο οποίο ανήκει η qp_1:

1. Αν $qp_1 \in bdp_1$, η ανάθεση ενός ελάχιστου λογικού πίνακα στον $qgen_2$ θα ακολουθήσει την περίπτωση 2 της παραγράφου 5.3.3.1, με μόνη διαφορά την αντικατάσταση της εσωτερικής συνένωσης με αριστερή εξωτερική συνένωση.

2. Αν $qp_1 \in bdp_2$ και δεν υπάρχει κάποιος $qgen_i \in bdp_1$ που να μοιράζεται το λογικό του πίνακα με αυτόν του qp_1 (περιπτώσεις 1 και 2, σημεία 1), τότε η ανάθεση ενός ελάχιστου λογικού πίνακα στον $qgen_2$ θα ακολουθήσει τη λογική της περίπτωσης 2 της παραγράφου 5.3.3.1.

3. Αν $qp_1 \in bdp_2$ και υπάρχει κάποιος $qgen_i \in bdp_1$ που να μοιράζεται το λογικό του πίνακα με αυτόν του qp_1 (περιπτώσεις 1 και 2, σημεία 1), τότε μπορούμε να εκμεταλλευτούμε την $s \bowtie o$ συνένωση, αν και μόνο ο $qgen_1$ περιέχει μια αναφέρουσα αντιστοιχία αντικειμένου $roMap_1$ με μη κενή λίστα συνενώσεων *joins* και η αντιστοιχία υποκειμένου του $qgen_2$ είναι ίδια με την αντιστοιχία υποκειμένου της αντιστοιχίας-γονέα της $roMap_1$. Σε αυτή την περίπτωση, αν έχει ανατεθεί στον $qgen_1$ ένας λογικός πίνακας της μορφής $table_1$ AS $alias_1$ LEFT OUTER JOIN $table_2$ AS $alias_2$ ON *joins*, όπου $table_2$ ο λογικός πίνακας της αντιστοιχίας-γονέα της $roMap_1$, ο $qgen_2$ θα επαναχρησιμοποιήσει το λογικό πίνακα $alias_2$.

Παράδειγμα 5.3.11. Θεωρούμε τις αντιστοιχίες του παραδείγματος 5.3.5 καθώς και τα ακόλουθα τμήματα προτύπων συνόλου δεδομένων:

dp₄: { ⋯
qp_{4a}: ?producer bsbm:country ?country DG.
qp_{4b}: OPTIONAL {?prod bsbm:producer ?producer DG }.
⋯ }

dp₅: { ⋯
qp_{5a}: OPTIONAL{ ?producer bsbm:country ?country DG }.
qp_{5b}: OPTIONAL{ ?prod bsbm:producer ?producer DG }.
⋯ }

dp₆: { ⋯
qp_{6a}: ?prod rdfs:label ?label DG.
⋯
qp_{6b}: OPTIONAL{ ?prod bsbm:producer ?producer DG }.
qp_{6c}: OPTIONAL{ ?producer bsbm:country ?country DG }.
⋯ }

Στο dp_4, ο παραγωγός του qp_{4b} περιέχει μια αναφέρουσα αντιστοιχία αντικειμένου με αντιστοιχία-γονέα που έχει ίδια αντιστοιχία υποκειμένου με αυτήν του $qgen_{4a}$. Αυτό σημαίνει ότι αν στον $qgen_{4a}$ έχει ανατεθεί ο λογικός πίνακας Producer2, στον $qgen_{4b}$ θα ανατεθεί ένας λογικός πίνακας τη μορφής Producer2 LEFT OUTER JOIN Product AS Product1 ON Producer2.nr = Product1.producer.

Στην περίπτωση του dp_5, η διαφορά έγκειται στο ότι το qp_{5a} είναι προαιρετικό πρότυπο. Αν υποθέσουμε ότι το qp_{5a} δεν εμφανίζει κάποια συνένωση με κάποιο υποχρεωτικό πρότυπο έτσι ώστε να αποφευχθεί κάποια συνένωση καθώς και ότι έχει ανατεθεί στον αντίστοιχο παραγωγό $qgen_{5a}$ ο λογικός πίνακας Producer2, στον $qgen_{5b}$ θα ανατεθεί ο λογικός πίνακας Producer2 INNER JOIN Product AS Product1 ON Producer2.nr = Product1.producer.

Όσον αφορά στον dp_6, λόγω της $s \bowtie s$ συνένωσης μεταξύ των qp_{6a}, qp_{6b}, έστω ότι έχει ανατεθεί (περίπτωση 1, σημείο 1) ένας λογικός πίνακας της μορφής Product3 LEFT OUTER JOIN Producer2 ON Producer2.nr = Product3.Producer. Τότε, στον $qgen_{6c}$ θα ανατεθεί ο λογικός πίνακας Producer2.

Παράδειγμα 5.3.12. Αναφερόμενοι και πάλι στο παράδειγμα 5.3.3, το ακόλουθο τμήμα του SPARQL ερωτήματος του παραδείγματος:

(t_1): `?product rdfs:label ?label.`
(t_2): `?product rdf:type bsbm-inst:ProductType1848.`
(t_3): `OPTIONAL{?product bsbm:productPropertyNumeric1 ?value1.}`

ισοδυναμεί με έναν BDPLeftJoin τελεστή, με πρώτο όρισμα το πρότυπο βασικού συνόλου δεδομένων που αποτελείται από τα πρότυπα t_1 και t_2, δεύτερο όρισμα το πρότυπο t_3 και τρίτο όρισμα μια συνάρτηση $optExprs$ με $optExprs(t_3) = true$. Υποθέτουμε επίσης την ύπαρξη μιας αντιστοιχίας τριάδων $ptpMap = \langle \text{RView}_1, subjMap_4, \{poMap_6\}\rangle$, με:

$RView_1 = \text{Product} \bowtie_{\text{Product.nr = ProductTypeProduct.product}} \text{ProductTypeProduct}$
$subjMap_4.template = \text{http://example.org/Product\{nr\}}$
$poMap_6.predMap.constant = \text{rdf:type}$
$poMap_6.objMap.template = \text{bsbm-inst:ProductType\{nr\}}$

Ο συγκεκριμένος BDPLeftJoin τελεστής θα επανεγγραφεί στο ακόλουθο SQL ερώτημα:

```
SELECT 'http://www.example.org/Producer$PERC_ENCODE('||R2RMLView5.producer||')$/
Product$PERC_ENCODE('||R2RMLView5.nr||')$' AS product, 'IRI' AS product_termType,
NULL AS product_language, NULL AS product_datatype, Product6.label AS label,
'Literal' AS label_termType, NULL AS label_language, 'xsd:string' AS label_datatype,
Product6.propertyNum1 AS value1, 'Literal' AS value1_termType, NULL AS value1_language,
'xsd:int' AS value1_datatype
FROM RView1 AS R2RMLView5 INNER JOIN Product AS Product6 ON R2RMLView5.nr = Product6.nr
WHERE R2RMLView5.productType = '16' AND R2RMLView5.nr IS NOT NULL AND
Product6.label IS NOT NULL
```

όπου έχει αποφευχθεί η αριστερή εξωτερική συνένωση του πίνακα Product με τον εαυτό του, λόγω της συμφωνίας υποκειμένων των t_1, t_3 και του ίδιου λογικού πίνακα που έχουν οι παραγωγοί τους.

5.3.3.3 Join

Η πράξη της συνένωσης είναι η πιο συνηθισμένη σε ένα SPARQL ερώτημα και συνήθως, εφαρμόζεται σε πρότυπα τριάδων που αποτελούν μέρη ενός προτύπου γράφου. Στην προτεινόμενη μέθοδο, τέτοιου είδους συνενώσεις αντιμετωπίζονται κατά την επεξεργασία των BDP και BDPLeftJoin τελεστών. Οι υπόλοιπες συνενώσεις που μπορεί να υπάρχουν σε ένα SPARQL ερώτημα – όπως π.χ. συνενώσεις υποερωτημάτων με έναν BDP τελεστή – πραγματοποιούνται με τη βοήθεια της πράξης της συνένωσης δύο SQL μοντέλων, όπως αυτή ορίστηκε στον αλγόριθμο 7.

Πιο συγκεκριμένα, τα SQL μοντέλα που έχουν υπολογιστεί κατά τη διάσχιση του αριστερού υποδέντρου του Join τελεστή συνενώνονται *ένα προς ένα* με τα SQL μοντέλα που έχουν υπολογιστεί κατά τη διάσχιση του αντίστοιχου δεξιού υποδέντρου και κάθε συνδυασμός οδηγεί στη δημιουργία ενός νέου SQL μοντέλου. Συνεπώς, αν το αριστερό υποδέντρο επανεγγράφεται σε m SQL μοντέλα και το δεξιό σε n, η αποτίμηση του Join τελεστή θα οδηγήσει στη δημιουργία $m \times n$ μοντέλων. Συνοπτικά, η διαδικασία φαίνεται στον αλγόριθμο 15.

Αλγόριθμος 15 *Επανεγγραφή Join τελεστή*

Είσοδος: Τελεστής Join *op*
Έξοδος: Λίστα SQL μοντέλων *sqlModels*

```
1: function ΕΠΑΝΕΓΓΡΑΦΗ_JOIN(op)
2:     leftModels ← ΕΠΑΝΕΓΓΡΑΦΗ(left(op))          # left : ανάκτηση αριστερού υποδέντρου
3:     rightModels ← ΕΠΑΝΕΓΓΡΑΦΗ(right(op))        # right : ανάκτηση δεξιού υποδέντρου
4:     leftOptVars ← ΕΥΡΕΣΗ_ΠΡΟΑΙΡΕΤΙΚΩΝ_ΜΕΤΑΒΛΗΤΩΝ(left(op))
5:     rightOptVars ← ΕΥΡΕΣΗ_ΠΡΟΑΙΡΕΤΙΚΩΝ_ΜΕΤΑΒΛΗΤΩΝ(right(op))
6:     for all model_1 ∈ leftModels do
7:         for all model_2 ∈ rightModels do
8:             sqlModels+ = ΣΥΝΕΝΩΣΗ(model_1, model_2, leftOptVars, rightOptVars)
9:         end for
10:    end for
11:    return sqlModels
12: end function
```

Υπενθυμίζουμε ότι η πράξη της συνένωσης δύο SQL μοντέλων (βλέπε και αλγόριθμο 7) οδηγεί σε ένα νέο SQL μοντέλο που εμφωλεύει τα SQL ερωτήματα που αντιστοιχούν στα δύο μοντέλα. Η συγκεκριμένη μέθοδος για την επανεγγραφή του Join τελεστή είναι ίδια με αυτήν του [77], όπου όμως δεν υπάρχει κάποια πρόβλεψη για την υποστήριξη εκτός-SQL συναρτήσεων. Η χρήση συναρτήσεων και μεθόδων που δεν υπάρχουν στο SQL μοντέλο είναι αναπόφευκτη, όπως είδαμε, λόγω της εκατοστιαίας κωδικοποίησης που συνοδεύει τις R2RML εκφράσεις προτύπου, αλλά και λόγω SPARQL συναρτήσεων που μπορεί να περιέχονται σε μια SPARQL λογική έκφραση. Αυτός είναι και ο λόγος για τον οποίο περιλαμβάνουμε στον ορισμό του SQL μοντέλου τα στοιχεία PostWhere και PostOrder, τα οποία περιέχουν συνθήκες επιλογής και διάταξης με συναρτήσεις που δεν ανήκουν στο SQL πρότυπο. Τα στοιχεία αυτά δε χρησιμοποιούνται στην κατασκευή του SQL ερωτήματος (όπως φαίνεται και από τον αλγόριθμο 6), αλλά λαμβάνονται υπόψη στο τελικό στάδιο κατασκευής της SPARQL λύσης (παράγραφος 5.3.4).

5.3.3.4 LeftJoin

Ο SPARQL τελεστής της αριστερής συνένωσης συναντάται επίσης πολύ συχνά σε SPARQL ερωτήματα και χρησιμοποιείται για την προσθήκη προαιρετικής πληροφορίας στα SPARQL αποτελέσματα. Μια υποπερίπτωση εφαρμογής του LeftJoin τελεστή (ισοδύναμα, OPTIONAL πρότασης) αποτελεί ο τελεστής BDPLeftJoin που εξετάστηκε στην παράγραφο 5.3.3.2, ο οποίος εξετάζει μονάχα περιπτώσεις αριστερής συνένωσης με μεμονωμένα πρότυπα τετράδας. Για τις υπόλοιπες περιπτώσεις αριστερής συνένωσης που εμφανίζονται σε ένα SPARQL ερώτημα, χρησιμοποιούμε την πράξη της αριστερής συνένωσης μεταξύ δύο SQL μοντέλων, όπως αυτή παρουσιάστηκε στον αλγόριθμο 8. Υπενθυμίζουμε ότι η συνάρτηση αριστερής συνένωσης SQL μοντέλων δέχεται ως τρίτο όρισμα και μια SQL συνθήκη συνένωσης κατ' αναλογία με τον ορισμό του LeftJoin τελεστή στην προδιαγραφή της SPARQL, όπου ως τρίτο όρισμα θεωρείται μια SPARQL λογική έκφραση.

Ένας LeftJoin τελεστής με συνθήκη συνένωσης μπορεί να προκύψει λόγω της ύπαρξης ενός φίλτρου μέσα σε ένα προαιρετικό πρότυπο γράφου, δηλαδή

μια έκφραση της μορφής:

$$bgp_1 \texttt{ OPTIONAL\{ } bgp_2 \texttt{ FILTER}(expr)\texttt{\}}$$

είναι ισοδύναμη με την αλγεβρική έκφραση LeftJoin$(bgp_1, bgp_2, expr)$. Επομένως, κατά την επεξεργασία ενός LeftJoin τελεστή, χρειάζεται να ελέγξουμε αν υπάρχει κάποιο φίλτρο στο δεξιό υποδέντρο, έτσι ώστε η έκφραση του φίλτρου να μετατραπεί σε μια ισοδύναμη SQL συνθήκη και να ληφθεί υπόψη στην αριστερή συνένωση των SQL μοντέλων. Κατά τα άλλα, η μέθοδος επεξεργασίας του LeftJoin τελεστή μοιάζει με αυτή του τελεστή Join, ήτοι υπολογίζονται τα SQL μοντέλα του αριστερού υποδέντρου του τελεστή και συνενώνονται εξ αριστερών με *καθένα* από τα SQL μοντέλα που προκύπτουν από το δεξιό υποδέντρο. Η διαδικασία συνοψίζεται στον αλγόριθμο 16.

Αλγόριθμος 16 *Επανεγγραφή LeftJoin τελεστή*

Είσοδος: Τελεστής LeftJoin *op*
Έξοδος: Λίστα SQL μοντέλων *sqlModels*

```
1: function ΕΠΑΝΕΓΓΡΑΦΗ_LEFTJOIN(op)
2:     cond = []
3:     leftModels ← ΕΠΑΝΕΓΓΡΑΦΗ(left(op))
4:     if right(op) είναι τύπου Filter then
5:         cond ← ΜΕΤΑΦΡΑΣΗ_ΣΕ_SQL(right(op).expr)
6:         rightModels ← ΕΠΑΝΕΓΓΡΑΦΗ(child(right(op)))
7:     else
8:         rightModels ← ΕΠΑΝΕΓΓΡΑΦΗ(right(op))
9:     end if
10:    leftOptVars ← ΕΥΡΕΣΗ_ΠΡΟΑΙΡΕΤΙΚΩΝ_ΜΕΤΑΒΛΗΤΩΝ(left(op))
11:    rightOptVars ← ΕΥΡΕΣΗ_ΠΡΟΑΙΡΕΤΙΚΩΝ_ΜΕΤΑΒΛΗΤΩΝ(right(op))
12:    for all model_1 ∈ leftModels do
13:        for all model_2 ∈ rightModels do
14:            sqlModels+ = ΑΡΙΣΤΕΡΗ_ΣΥΝΕΝΩΣΗ(model_1, model_2, leftOptVars, rightOptVars, cond)
15:        end for
16:    end for
17:    return sqlModels
18: end function
```

5.3.3.5 Union

Η ένωση SPARQL τελεστών αποτιμάται, όπως είδαμε και στη παράγραφο 5.3.1, μέσω της ένωσης των συνόλων των SPARQL αντιστοιχιών που προκύπτουν από την αποτίμηση κάθε τελεστή. Η διαδικασία αυτή πραγματοποιείται στο τελικό στάδιο του αλγορίθμου, όπου δημιουργείται ένα SQL ερώτημα ένωσης όλων των SQL ερωτημάτων που προκύπτουν από κάθε κατασκευασμένο SQL μοντέλο. Δεδομένου λοιπόν ότι η λίστα των SQL μοντέλων που δημιουργούνται κατά τη διάσχιση του SPARQL δέντρου *υπονοεί* τη λογική διάζευξή τους, ο τελεστής Union αποτιμάται μέσω της ενοποίησης σε μία λίστα όλων των SQL μοντέλων που προκύπτουν από τα υποδέντρα του Union τελεστή.

5.3.3.6 Minus

Η πράξη της διαφοράς δύο SPARQL τελεστών έχει ως αποτέλεσμα εκείνες τις SPARQL αντιστοιχίες μ_1 που προκύπτουν από την αποτίμηση του πρώτου

τελεστή και οι οποίες είναι ασύμβατες ή δεν έχουν κοινό πεδίο ορισμού με κάθε SPARQL αντιστοιχία μ_2 που προκύπτει από την αποτίμηση του δεύτερου τελεστή. Με τρόπο ανάλογο των τελεστών Join και LeftJoin, κατασκευάζονται δύο λίστες SQL μοντέλων, μία για κάθε υποδέντρο του τελεστή Minus και για κάθε πιθανό ζεύγος SQL μοντέλων, λαμβάνεται η διαφορά τους.

Η πράξη της διαφοράς δύο SQL μοντέλων μπορεί να εκφραστεί ως μια αριστερή εξωτερική συνένωση ακολουθούμενη από μια συνθήκη επιλογής που εξασφαλίζει ότι οι τιμές των κοινών μεταβλητών για το δεύτερο μοντέλο είναι όλες κενές (null), καθώς αν οι κοινές μεταβλητές ήταν μη κενές, αυτό θα σήμαινε ότι το αντίστοιχο ζεύγος αντιστοιχιών μ_1, μ_2 θα ήταν συμβατό και δε θα έπρεπε να περιλαμβάνεται στο τελικό αποτέλεσμα. Συνεπώς, η πράξη της διαφοράς δύο SQL μοντέλων $model_1, model_2$ ακολουθεί τον αλγόριθμο 8 της αριστερής συνένωσης, με τη μόνη διαφορά να υπάρχει στην κατασκευή του νέου SQL μοντέλου στη γραμμή 32, όπου το στοιχείο Where του νέου μοντέλου θα περιέχει επιπλέον και μια έκφραση της μορφής AND *var* IS NULL, για κάθε $var \in commonVars$.

5.3.3.7 Filter

Μια πρόταση FILTER περιέχει μια SPARQL λογική έκφραση την οποία πρέπει να ικανοποιούν όλες οι SPARQL αντιστοιχίες μιας λύσης. Συνεπώς, ένας Filter τελεστής λειτουργεί ως συνθήκη επιλογής και εμπλουτίζει το στοιχείο Where ενός SQL μοντέλου. Σε γενικές γραμμές, η SPARQL έκφραση μεταφράζεται σε μια ισοδύναμη SQL συνθήκη, αντικαθιστώντας τις μεταβλητές που αναφέρονται στην πρώτη με μια SQL έκφραση που υποδεικνύεται από το στοιχείο VarSources του SQL μοντέλου και τους παραγωγούς κόμβου που αντιστοιχούν σε κάθε μεταβλητή. Σημειώνουμε επίσης ότι αρκετές από τις SPARQL συναρτήσεις που μπορούν να χρησιμοποιηθούν σε μια SPARQL λογική έκφραση δεν έχουν αντίστοιχή τους στην SQL, με αποτέλεσμα κάποιες παραγόμενες συνθήκες να προστίθενται στη λίστα PostWhere με τις εκτός-SQL συνθήκες. Η αναλυτική παρουσίαση του αλγορίθμου μετάφρασης του τελεστή Filter για το σύνολο των δυνατών SPARQL λογικών εκφράσεων παραλείπεται για λόγους συντομίας από το παρόν κεφάλαιο.

5.3.3.8 Τροποποιητές λύσης

Οι SPARQL τροποποιητές λύσης αποτελούν τελεστές οι οποίοι εφαρμόζονται σε ένα σύνολο SPARQL αντιστοιχιών, αλλάζοντας τη διάταξή τους ή απορρίπτοντας κάποιες από αυτές. Οι τροποποιητές που εξετάζουμε είναι οι τελεστές Project, Order, Distinct και Slice (συνδυασμός LIMIT και OFFSET τροποποιητών).

Project Ο τελεστής της προβολής μεταβλητών Project διατηρεί στην τελική SPARQL λύση μονάχα τις μεταβλητές που περιέχονται σε αυτόν. Συνεπώς, η επεξεργασία αυτού του τελεστή συνίσταται στην απαλοιφή μεταβλητών που δεν ανήκουν στη λίστα προβολών του SPARQL ερωτήματος. Υπενθυμίζουμε ότι για κάθε μεταβλητή, προβάλλουμε 4 SQL όρους (αλγόριθμος 11): την τιμή στην οποία δεσμεύεται η μεταβλητή, το είδος του παραγόμενου RDF όρου,

τη γλώσσα και τον τύπο δεδομένων του RDF όρου, αν πρόκειται για λεκτικό. Οι όροι αυτοί είναι απαραίτητοι για την ορθή παραγωγή ενός RDF όρου στο τελικό στάδιο του αλγορίθμου, ενώ αν ένας όρος δεν είναι λεκτικό προβάλλεται η τιμή NULL στου αντίστοιχους όρους της γλώσσας και του τύπου δεδομένων. Σε κάθε περίπτωση, χρειάζεται η προβολή και των 4 όρων προκειμένου να εξασφαλιστεί η συμβατότητα των παραγόμενων SQL ερωτημάτων ως προς τη σχεσιακή πράξη της ένωσης.

Order Ο τελεστής διάταξης SPARQL αντιστοιχιών Order περιέχει στη γενική του μορφή μια SPARQL έκφραση, σύμφωνα με την οποία διατάσσονται οι SPARQL αντιστοιχίες. Όπως και στην περίπτωση του τελεστή Filter, η έκφραση του τελεστή Order μεταφράζεται σε μια ισοδύναμη SQL συνθήκη σύμφωνα με το στοιχείο VarSources του SQL μοντέλου και ανατίθεται στο στοιχείο Order του μοντέλου. Αν η μετάφραση της SPARQL έκφρασης απαιτεί τη χρήση συναρτήσεων εκτός του SQL προτύπου, αυτές ανατίθενται στο στοιχείο PostOrder και η εφαρμογή της συνθήκης διάταξης πραγματοποιείται στο τελικό βήμα του αλγορίθμου. Ειδική μέριμνα χρειάζεται να ληφθεί για τους περιορισμούς που θέτει η SPARQL στη διάταξη όρων, καθώς δίνει προτεραιότητα στις κενές τιμές, κατόπιν στους κενούς κόμβους και τα IRIs και τέλος, στα λεκτικά. Αυτή η απαίτηση οδηγεί στην προσθήκη δύο ακόμα συνθηκών στη λίστα Order του SQL μοντέλου: η πρώτη είναι της μορφής "CASE WHEN *expr* IS NULL THEN 0 ELSE 1" για την ανάθεση προτεραιότητας σε κενές τιμές, όπου *expr* μια SQL έκφραση και η δεύτερη διατάσσει τα αποτελέσματα με βάση το είδος του RDF όρου "*varName*_termType", όπου *varName* το όνομα μιας SPARQL μεταβλητής.

Distinct Ο τροποποιητής Distinct απορρίπτει διπλές SPARQL αντιστοιχίες στην τελική SPARQL λύση, ισοδυναμώντας με τη χρήση του ομώνυμου τροποποιητή στην SQL. Αν ο συγκεκριμένος τροποποιητής είναι παρών σε ένα SPARQL ερώτημα, η λογική τιμή Distinct για όλα τα SQL μοντέλα που έχει κατασκευάσει ο αλγόριθμος επανεγγραφής τίθεται ίση με true.

Slice Ο SPARQL τελεστής Slice συνδυάζει τους τροποποιητές LIMIT και OFFSET, οι οποίοι καθορίζουν αντίστοιχα τον αριθμό των SPARQL αντιστοιχιών που αποτελούν τη SPARQL λύση και τον αύξοντα αριθμό πέραν του οποίου αρχίζουν να περιλαμβάνονται οι SPARQL αντιστοιχίες στην τελική λύση. Οι δύο ακέραιοι αριθμοί του τελεστή Slice ανατίθενται ως έχουν στο στοιχείο Slice κάθε παραγόμενου SQL μοντέλου και λαμβάνονται υπόψη κατά τη δημιουργία του αντίστοιχου SQL ερωτήματος.

Παράδειγμα 5.3.13. Με βάση τα όσα αναφέρθηκαν μέχρι τώρα και τα παραδείγματα 5.3.8 και 5.3.12, όπου παρουσιάστηκαν τα SQL ερωτήματα που αντιστοιχούν σε επιμέρους τμήματα του ερωτήματος του παραδείγματος 5.3.3, είμαστε πλέον σε θέση να επανεγγράψουμε το πλήρες ερώτημα. Το ισοδύναμο SQL ερώτημα θα έχει ως εξής:

```
SELECT DISTINCT ON (product, product_termType, label, label_termType, value1,
value1_termType)
'http://example.org/Product$PERC_ENCODE('||R2RMLView1.nr||')$' AS product,
'IRI' AS product_termType, NULL AS product_language, NULL AS product_datatype,
```

```
Product6.label AS label, 'Literal' AS label_termType, NULL AS label_language,
'xsd:string' AS label_datatype, Product6.propertyNum1 AS value1,
'Literal' AS value1_termType, NULL AS value1_language, 'xsd:nt' AS value1_datatype
FROM
(Product AS Product6 INNER JOIN RView1 AS R2RMLView5 ON Product6.nr = R2RMLView5.nr)
LEFT OUTER JOIN
(ProductFeatureProduct AS pfp3 INNER JOIN ProductFeatureProduct AS pfp4 ON
pfp3.product = pfp4.product) ON R2RMLView5.nr = pfp3.nr
AND pfp3.productFeature = '43883' AND pfp4.productFeature = '7746'
AND pfp3.product IS NOT NULL
WHERE R2RMLView5.productType = '1848' AND R2RMLView5.nr IS NOT NULL
AND Product6.label IS NOT NULL AND Product6.propertyNum1 > 136.0
ORDER BY CASE WHEN Product6.label IS NULL THEN 0 ELSE 1 END, label_termType DESC,
Product6.label DESC
LIMIT 10
```

5.3.4 Κατασκευή SPARQL λύσης

Όπως είδαμε στην παράγραφο 5.3.3, η επανεγγραφή του SPARQL ερωτήματος οδηγεί σε μια λίστα SQL μοντέλων, η ένωση των οποίων ισοδυναμεί με το αρχικό SPARQL ερώτημα. Μετά την εκτέλεση της ένωσης των ερωτημάτων που αντιστοιχούν στα SQL μοντέλα, τα SQL σύνολα αποτελεσμάτων μετατρέπονται σε SPARQL αντιστοιχίες στο τελευταίο βήμα του αλγορίθμου. Σε αυτό το τελευταίο βήμα πραγματοποιείται και η εφαρμογή πιθανών εκτός-SQL συνθηκών επιλογής και διάταξης στη λίστα αποτελεσμάτων. Κάθε SQL μοντέλο που διαθέτει τουλάχιστον μία τέτοια εκτός-SQL συνθήκη εκτελείται ξεχωριστά από τα υπόλοιπα, ώστε να μπορούν να εφαρμοστούν οι σχετικές συνθήκες. Η κατασκευή μιας SPARQL αντιστοιχίας από ένα στοιχείο ενός SQL συνόλου αποτελεσμάτων απεικονίζεται στον αλγόριθμο 17 και έπεται της εφαρμογής των εκτός-SQL συνθηκών επιλογής.

5.3.5 Επισκόπηση και παρατηρήσεις επί του αλγορίθμου

Στην παρούσα ενότητα, παρουσιάσαμε έναν αλγόριθμο SPARQL-σε-SQL επανεγγραφής, υπό την παρουσία R2RML αντιστοιχιών, ο οποίος χρησιμοποιεί την ιδέα διατήρησης και εμπλουτισμού ενός SQL μοντέλου που παρουσιάστηκε στο [77]. Εν συντομία, οι σημαντικότερες διαφορές με το συγκεκριμένο αλγόριθμο συνοψίζονται στα επόμενα:

α. Η αξιοποίηση $s \bowtie s$ και $s \bowtie o$ συνενώσεων μεταξύ προτύπων τετράδων για την αποφυγή άσκοπων συνενώσεων πινάκων στο παραγόμενο SQL ερώτημα.

β. Η εισαγωγή της SPARQL-$\mathbb{DB}$ άλγεβρας η οποία λειτουργεί σε επίπεδο τετράδων και διευκολύνει την αναγνώριση προτύπων που προσφέρονται για την αποφυγή συνενώσεων πινάκων.

γ. Η επέκταση του ορισμού του SQL μοντέλου ώστε να περιλαμβάνει τα απαραίτητα στοιχεία για τη δημιουργία της τελικής πρότασης FROM ενός SQL ερωτήματος, καθώς και συνθήκες επιλογής και διάταξης που περιέχουν εκτός-SQL συνθήκες. Η παρουσία των τελευταίων είναι συχνή στην

Αλγόριθμος 17 *Κατασκευή SPARQL αντιστοιχίας*

Είσοδος: SQL σύνολο αποτελεσμάτων *resultSet*, σύνολο SPARQL μεταβλητών *vars*
Έξοδος: Σύνολο SPARQL αντιστοιχιών *bindings*

```
1: function ΚΑΤΑΣΚΕΥΗ_SPARQL_ΑΝΤΙΣΤΟΙΧΙΑΣ(resultSet)
2:     bindings = ∅
3:     for all result ∈ resultSet do
4:         for all var ∈ vars do
5:             RDFnode, binding =[]
6:             value ← result.var
7:             if value περιέχει ένδειξη για εκατοστιαία κωδικοποίηση then
8:                 value ← percentEncode(result.var)
9:             end if
10:            if result.var_termType = BlankNode then
11:                RDFNode ← ΔΗΜΙΟΥΡΓΙΑ_ΚΕΝΟΥ_ΚΟΜΒΟΥ
12:            else if result.var_termType =IRI then
13:                RDFNode ← ΔΗΜΙΟΥΡΓΙΑ_IRI(value)
14:            else if result.var_termType =Literal then
15:                RDFNode ← ΔΗΜΙΟΥΡΓΙΑ_ΛΕΚΤΙΚΟΥ(value, result.var_language, result.var_datatype)
16:            end if
17:            binding+ = (var, RDFnode)
18:        end for
19:        bindings+ = binding
20:    end for
21:    return bindings
22: end function
```

περίπτωση της R2RML και SPARQL λογικών εκφράσεων (εντός φίλτρων ή ORDER προτάσεων).

δ. Η προσαρμογή του αλγορίθμου ώστε να παράγει περισσότερα του ενός SQL μοντέλα, κατάσταση η οποία είναι σχεδόν αναπόφευκτη στην R2RML, δεδομένου του ότι οι αντιστοιχίες μεταξύ προτύπων τριάδων/πινάκων και RDF όρων/στηλών δεν είναι πλέον 1:N.

ε. Υποστήριξη του Minus τελεστή.

στ. Προσθήκη IS NOT NULL συνθηκών για την περίπτωση μεταβλητών που είναι σίγουρα δεσμευμένες σε ένα πρότυπο βασικού γράφου. Επίσης, προσθήκη EXISTS υποερωτήματος στην περίπτωση παρουσίας προτύπων χωρίς μεταβλητές.

ζ. Υποστήριξη του R2RML χαρακτηριστικού της αντίστροφης έκφρασης.

η. Απλοποίηση των συνθηκών (αριστερής εξωτερικής ή εσωτερικής) συνένωσης, όταν είναι γνωστό ότι μια μεταβλητή συνένωσης είναι σίγουρα δεσμευμένη σε κάποιο από τα δύο υποδέντρα του τελεστή (αριστερής εξωτερικής ή εσωτερικής) συνένωσης.

Αν και στόχος του συγκεκριμένου αλγορίθμου, όπως άλλωστε και του [77], είναι η παραγωγή ενός μοναδικού SQL ερωτήματος, στην πράξη οι απαιτήσεις που θέτουν οι R2RML και SPARQL για χρήση μεθόδων εκτός του SQL προτύπου καθιστούν ανέφικτο αυτό το στόχο και επιβάλλουν την προσθήκη ενός επιπρόσθετου τελευταίου βήματος επιλογής και διάταξης αποτελεσμάτων, για

τις περιπτώσεις που αυτές δεν μπορούν να εκφραστούν στο SQL ερώτημα. Το συγκεκριμένο εμπόδιο βέβαια μπορεί εύκολα να ξεπεραστεί υποθέτοντας ότι υπάρχουν αντίστοιχες αποθηκευμένες διαδικασίες στη ΒΔ που υλοποιούν τις αναγκαίες συναρτήσεις.

Εκτός από την εκτέλεση SPARQL SELECT ερωτημάτων, ο συγκεκριμένος αλγόριθμος μπορεί να χρησιμοποιηθεί και για τα υπόλοιπα είδη SPARQL ερωτημάτων. Ένα ASK ερώτημα χρειάζεται απλά να ελέγξει αν υπάρχει μία μη κενή SPARQL λύση, επομένως αρκεί ένας προγραμματιστικός έλεγχος που θα έπεται της εκτέλεσης των SQL ερωτημάτων ή ισοδύναμα, η προσθήκη των SQL ερωτημάτων σε ισάριθμες εκφράσεις EXISTS και η εκτέλεσή τους. Ένα CONSTRUCT ερώτημα δημιουργεί έναν RDF γράφο χρησιμοποιώντας τις SPARQL αντιστοιχίες του αποτελέσματος, διαδικασία που επίσης μπορεί να πραγματοποιηθεί σε ένα τελικό προγραμματιστικό βήμα, ενώ ένα DESCRIBE ερώτημα επίσης θα επιστρέψει έναν RDF γράφο με πληροφορία για ένα συγκεκριμένο RDF πόρο, οπότε μπορεί να ακολουθηθεί η ίδια διαδικασία. Η ακριβής μορφή του γράφου που θα επιστραφεί εξαρτάται από το εκάστοτε σύνολο δεδομένων και την παραμετροποίηση της μηχανής επανεγγραφής. Σε κάθε περίπτωση, το αρχικό DESCRIBE ερώτημα θα μετατραπεί σε ένα ισοδύναμο CONSTRUCT, το οποίο θα εκτελεστεί σύμφωνα με τα όσα αναφέρθηκαν προηγουμένως.

5.4 Αξιολόγηση συστήματος

Οι κυριότεροι παράγοντες που επηρεάζουν την απόδοση ενός συστήματος SPARQL-σε-SQL επανεγγραφής είναι το υποβληθέν SPARQL ερώτημα και το μέγεθος της υποκείμενης ΒΔ. Σε αυτή την ενότητα, αξιολογούμε τη συμπεριφορά του συστήματος καθώς μεταβάλλονται οι προηγούμενοι παράγοντες, ακολουθώντας την πρότυπη μεθοδολογία BSBM [40]. Η BSBM αποτελεί μια μεθοδολογία μέτρησης επιδόσεων συστημάτων αποθήκευσης RDF δεδομένων, η οποία θεωρεί SPARQL ερωτήματα με διαφορετικά χαρακτηριστικά και πολυπλοκότητα το καθένα και μετρά το χρόνο απόκρισης για κλιμακούμενα μεγέθη RDF γράφων που όλοι τους ακολουθούν μια συγκεκριμένη δομή. Επίσης, η BSBM ορίζει και την αντίστοιχη σχεσιακή αναπαράσταση του εξεταζόμενου RDF γράφου (ένα μέρος της οποίας φαίνεται στο σχήμα 5.3), οπότε προσφέρεται και για την αξιολόγηση συστημάτων SPARQL-σε-SQL επανεγγραφής, σε αντίθεση με τη μεθοδολογία SP^{2}Bench [169]. Η αντιστοιχία μεταξύ της σχεσιακής και της RDF αναπαράστασης δίνεται υπό τη μορφή μιας D2R αντιστοιχίας και για τις ανάγκες της παρούσας αξιολόγησης, μεταφράστηκε στη γλώσσα R2RML.

Η BSBM ακολουθεί ένα σενάριο ηλεκτρονικού εμπορίου στο οποίο ένας καταναλωτής θέτει ερωτήματα σε μια RDF βάση γνώσης (ισοδύναμα, σε μια ΒΔ), στην προσπάθειά του να εντοπίσει το κατάλληλο για αυτόν προϊόν. Στο πλαίσιο αυτό, το BSBM σχήμα περιλαμβάνει τύπους οντοτήτων, όπως «Πρόσωπο», «Προϊόν», «Προσφορά», «Κριτική» και «Παραγωγός» και παράλληλα, ορίζεται ένα μείγμα 25 SPARQL ερωτημάτων, αποτελούμενο από 12 διαφορετικά πρότυπα ερωτημάτων που αντιπροσωπεύουν τις τυπικές ενέργειες που θα πραγματοποιήσει ο καταναλωτής κατά την παραμονή του στον ιστότοπο του καταστήματος. Τα SPARQL ερωτήματα αναφέρονται στο [40] και παρατίθενται στον πίνακα 5.3 προς διευκόλυνση του αναγνώστη. Καθένα από τα

πρότυπα SPARQL ερωτημάτων του πίνακα 5.3 δέχεται παραμέτρους (δηλώνονται με το σύμβολο %) οι οποίες υπολογίζονται με ψευδοτυχαίο τρόπο από το πρόγραμμα εκτέλεσης της BSBM μεθοδολογίας.

Πίνακας 5.3: *SPARQL ερωτήματα της BSBM μεθολογίας*

Q_1:

```
SELECT DISTINCT ?product ?label
WHERE {
?product rdfs:label ?label .
?product rdf:type %ProductType% .
?product bsbm:productFeature %ProductFeature1% .
?product bsbm:productFeature %ProductFeature2% .
?product bsbm:productPropertyNumeric1 ?value1 .
FILTER (?value1 > %x%)}
ORDER BY ?label
LIMIT 10
```

Q_2:

```
SELECT ?label ?comment ?producer ?productFeature ?propertyTextual1
?propertyTextual2 ?propertyTextual3 ?propertyNumeric1
?propertyNumeric2 ?propertyTextual4 ?propertyTextual5
?propertyNumeric4
WHERE {
%ProductXYZ% rdfs:label ?label .
%ProductXYZ% rdfs:comment ?comment .
%ProductXYZ% bsbm:producer ?p .
?p rdfs:label ?producer .
%ProductXYZ% dc:publisher ?p .
%ProductXYZ% bsbm:productFeature ?f .
?f rdfs:label ?productFeature .
%ProductXYZ% bsbm:productPropertyTextual1 ?propertyTextual1 .
%ProductXYZ% bsbm:productPropertyTextual2 ?propertyTextual2 .
%ProductXYZ% bsbm:productPropertyTextual3 ?propertyTextual3 .
%ProductXYZ% bsbm:productPropertyNumeric1 ?propertyNumeric1 .
%ProductXYZ% bsbm:productPropertyNumeric2 ?propertyNumeric2 .
OPTIONAL { %ProductXYZ% bsbm:productPropertyTextual4 ?propertyTextual4 }
OPTIONAL { %ProductXYZ% bsbm:productPropertyTextual5 ?propertyTextual5 }
OPTIONAL { %ProductXYZ% bsbm:productPropertyNumeric4 ?propertyNumeric4 }}
```

Q_3:

```
SELECT ?product ?label
WHERE {
?product rdfs:label ?label .
?product rdf:type %ProductType% .
?product bsbm:productFeature %ProductFeature1% .
?product bsbm:productPropertyNumeric1 ?p1 .
FILTER ( ?p1 > %x% )
?product bsbm:productPropertyNumeric3 ?p3 .
FILTER (?p3 < %y% )
OPTIONAL {
?product bsbm:productFeature %ProductFeature2% .
?product rdfs:label ?testVar }
FILTER (!bound(?testVar)) }
ORDER BY ?label
LIMIT 10
```

Q_4:

```
SELECT ?product ?label
WHERE {
{ ?product rdfs:label ?label .
?product rdf:type %ProductType% .
?product bsbm:productFeature %ProductFeature1% .
?product bsbm:productFeature %ProductFeature2% .
?product bsbm:productPropertyNumeric1 ?p1 .
FILTER ( ?p1 > %x% )
} UNION {
?product rdfs:label ?label .
?product rdf:type %ProductType% .
?product bsbm:productFeature %ProductFeature1% .
?product bsbm:productFeature %ProductFeature3% .
?product bsbm:productPropertyNumeric2 ?p2 .
FILTER ( ?p2> %y% ) }}
ORDER BY ?label
LIMIT 10 OFFSET 10
```

Q_5:

```
SELECT DISTINCT ?product ?productLabel
WHERE {
?product rdfs:label ?productLabel .
FILTER (%ProductXYZ% != ?product)
%ProductXYZ% bsbm:productFeature ?prodFeature .
?product bsbm:productFeature ?prodFeature .
%ProductXYZ% bsbm:productPropertyNumeric1 ?origProperty1 .
?product bsbm:productPropertyNumeric1 ?simProperty1 .
FILTER (?simProperty1 < (?origProperty1 + 120) && ?simProperty1 >
(?origProperty1 - 120))
%ProductXYZ% bsbm:productPropertyNumeric2 ?origProperty2 .
?product bsbm:productPropertyNumeric2 ?simProperty2 .
FILTER (?simProperty2 < (?origProperty2 + 170) && ?simProperty2 >
(?origProperty2 - 170)) }
ORDER BY ?productLabel
LIMIT 5
```

Q_7:

```
SELECT ?productLabel ?offer ?price ?vendor ?vendorTitle ?review
?revTitle ?reviewer ?revName ?rating1 ?rating2
WHERE {
%ProductXYZ% rdfs:label ?productLabel .
OPTIONAL {
?offer bsbm:product %ProductXYZ% .
?offer bsbm:price ?price .
?offer bsbm:vendor ?vendor .
?vendor rdfs:label ?vendorTitle .
?vendor bsbm:country <http://downlode.org/rdf/iso-3166/countries#DE>.
?offer dc:publisher ?vendor .
?offer bsbm:validTo ?date .
FILTER (?date > %currentDate% ) }
OPTIONAL {
?review bsbm:reviewFor %ProductXYZ% .
?review rev:reviewer ?reviewer .
?reviewer foaf:name ?revName .
?review dc:title ?revTitle .
OPTIONAL { ?review bsbm:rating1 ?rating1 . }
OPTIONAL { ?review bsbm:rating2 ?rating2 . } } }
```

Q_8:

```
SELECT ?title ?text ?reviewDate ?reviewer ?reviewerName ?rating1
?rating2 ?rating3 ?rating4
WHERE {
?review bsbm:reviewFor %ProductXYZ% .
?review dc:title ?title .
?review rev:text ?text .
FILTER langMatches( lang(?text), ''EN'' )
?review bsbm:reviewDate ?reviewDate .
?review rev:reviewer ?reviewer .
?reviewer foaf:name ?reviewerName .
OPTIONAL { ?review bsbm:rating1 ?rating1 . }
OPTIONAL { ?review bsbm:rating2 ?rating2 . }
OPTIONAL { ?review bsbm:rating3 ?rating3 . }
OPTIONAL { ?review bsbm:rating4 ?rating4 . } }
ORDER BY DESC(?reviewDate) LIMIT 20
```

Q_{10}:

```
SELECT DISTINCT ?offer ?price
WHERE {
?offer bsbm:product %ProductXYZ% .
?offer bsbm:vendor ?vendor .
?offer dc:publisher ?vendor .
?vendor bsbm:country %CountryXYZ% .
?offer bsbm:deliveryDays ?deliveryDays .
FILTER (?deliveryDays <= 3)
?offer bsbm:price ?price .
?offer bsbm:validTo ?date .
FILTER (?date > %currentDate% ) }
ORDER BY xsd:double(str(?price))
LIMIT 10
```

Q_{11}:

```
SELECT ?property ?hasValue ?isValueOf
WHERE {
{ %OfferXYZ% ?property ?hasValue }
UNION
{ ?isValueOf ?property %OfferXYZ% } }
```

Σε ό,τι αφορά στο μέγεθος της ΒΔ, η BSBM παρέχει κατάλληλη γεννήτρια δεδομένων που, με βάση έναν παράγοντα κλιμάκωσης, παράγει τυχαία δεδομένα που ακολουθούν το BSBM σχήμα. Για τις ανάγκες της τρέχουσας αξιολόγησης, παράχθηκαν σχεσιακά δεδομένα που αντιστοιχούν σε 50.000, 250.000, 1.000.000, 25.000.000 και 100.000.000 RDF τριάδες[15], προκειμένου να μετρηθεί η επίδοση του συστήματος για ένα ευρύ φάσμα μεγεθών ΒΔ.

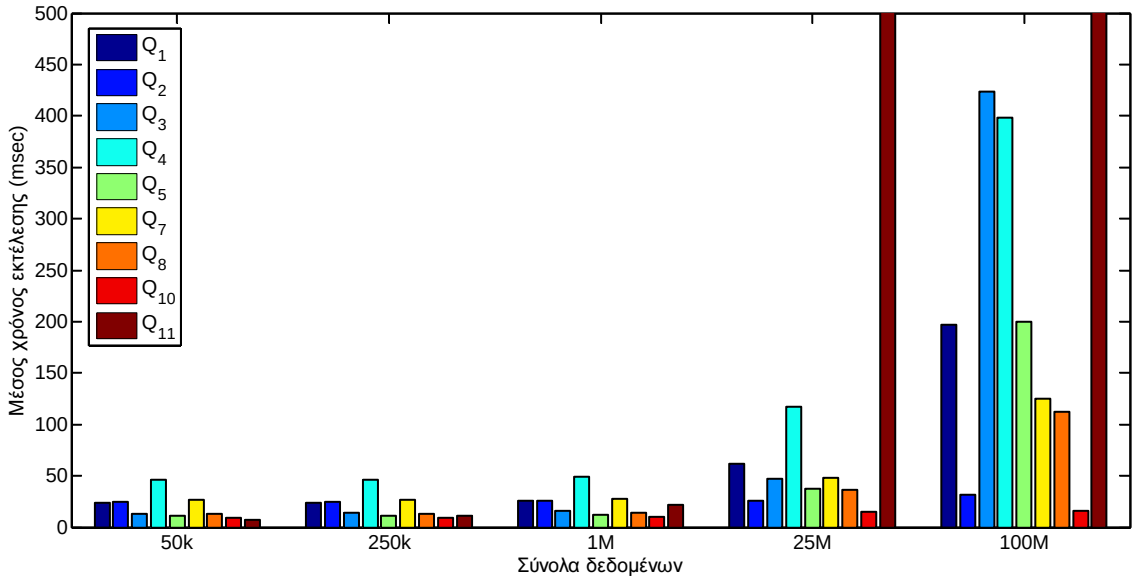
Στην αξιολόγηση που πραγματοποιείται σε αυτή την ενότητα, θεωρούνται μόνο SELECT ερωτήματα και συνεπώς αγνοούνται τα ερωτήματα Q_9 και Q_{12} του [40]. Επίσης, η προκαθορισμένη μελέτη περίπτωσης της BSBM (Explore use case) αγνοεί το ερώτημα Q_6, το οποίο πραγματοποιεί αναζήτηση σε προϊόντα μέσω μιας κανονικής έκφρασης.

Σε γενικές γραμμές, υιοθετήθηκε η πειραματική μεθοδολογία που παρουσιάζεται στο [40] και η οποία ακολουθείται στις διάφορες εκδόσεις των BSBM πειραμάτων[16]. Για καθένα από τα θεωρούμενα σύνολα δεδομένων, εκτελέστη-

[15] Τα ποσοτικά χαρακτηριστικά των συγκεκριμένων συνόλων δεδομένων αναφέρονται στο [40] και στο http://wifo5-03.informatik.uni-mannheim.de/bizer/berlinsparqlbenchmark/spec/Dataset/index.html.

[16] http://wifo5-03.informatik.uni-mannheim.de/bizer/berlinsparqlbenchmark/spec/BenchmarkRules/index.html

καν 500 BSBM μειωμένα (δηλ. χωρίς τα Q_6, Q_9, Q_{12}) μείγματα ερωτημάτων και μετρήθηκε ο χρόνος εκτέλεσής τους. Κάθε φορά, ένα σύνολο 50 μειγμάτων εκτελείται πριν από το κυρίως σύνολο των 500 μειγμάτων, προκειμένου να ενεργοποιηθεί το σύστημα και η μέτρηση της επίδοσής του να γίνει σε συνθήκες που πλησιάζουν όσο το δυνατόν περισσότερο τις κανονικές συνθήκες λειτουργίας. Οι μετρήσεις πραγματοποιήθηκαν σε ένα συμβατικό σύστημα με επεξεργαστή Intel® Core™i5-2450M @ 2.5GHz, συνολική μνήμη 4GB σε περιβάλλον 64-bit Windows 7, ενώ στα πειράματα χρησιμοποιήθηκε το ΣΔΒΔ PostgreSQL. Ο μέσος χρόνος εκτέλεσης των Q_1 – Q_{12} απεικονίζεται στο σχήμα 5.4, όπου για λόγους ευκρίνειας ο κάθετος άξονας εκτείνεται μέχρι τα 500 msec.



Σχήμα 5.4: *Μέσοι χρόνοι εκτέλεσης για RDB4RDF*

Παρατηρούμε ότι το ερώτημα που συνεισφέρει το μεγαλύτερο ποσοστό του συνολικού χρόνου εκτέλεσης του μείγματος ερωτημάτων για μεγαλύτερες τάξεις μεγέθους είναι το Q_{11} (με μέσους χρόνους 540 και 2130 msec για τα σύνολα 25M και 100M αντίστοιχα). Το Q_{11} περιλαμβάνει μεταβλητή στη θέση του κατηγορήματος και η εκτέλεση ερωτημάτων αυτού του είδους θεωρείται γενικά πολύ ακριβή ακόμα και για μηχανές που λειτουργούν πάνω από φυσικά συστήματα RDF αποθήκευσης. Στην περίπτωση μας, η παρουσία μεταβλητής στη θέση του κατηγορήματος οδηγεί στην αναγνώριση περισσότερων του ενός συμβατών παραγωγών για τα πρότυπα τριάδας του Q_{11}, καθένας εκ των οποίων οδηγεί σε ξεχωριστό πλάνο αποτίμησης, με αποτέλεσμα το τελικό SQL ερώτημα να αποτελείται από την ένωση αρκετών υποερωτημάτων.

Τα επόμενα πιο απαιτητικά ερωτήματα είναι τα Q_3 και Q_4, με το πρώτο να διαθέτει μια αριστερή συνένωση και το δεύτερο να αποτελείται από την ένωση δύο προτύπων γράφου. Το Q_3 δεν είναι ισοδύναμο με έναν BDPLeftJoin τελεστή, καθώς το προαιρετικό πρότυπο γράφου του αποτελείται από δύο πρότυπα τριάδας, με αποτέλεσμα να μην μπορεί να αποφευχθεί η αριστερή εξωτερική συνένωση πινάκων. Επίσης, αν και τα δύο πρότυπα γράφου του Q_3 έχουν τη μορφή αστέρα, γεγονός που τα καθιστά υποψήφια για την εφαρμογή

των προτεινόμενων βελτιστοποιήσεων, η παρουσία των ιδιοτήτων `rdf:type` και `bsbm:productFeature` των οποίων η προέλευση είναι αντίστοιχοι M:N πίνακες συσχέτισης δεν επιτρέπουν την αποφυγή αντίστοιχων εσωτερικών συνενώσεων. Το ίδιο ισχύει και για τα δύο πρότυπα γράφου του ερωτήματος Q_4, όπως και για τα πρότυπα γράφου των Q_1, Q_5. Αν οι συγκεκριμένες ιδιότητες έλειπαν από τα εν λόγω πρότυπα γράφου, ο χρόνος εκτέλεσης των αντίστοιχων ερωτημάτων θα ήταν σαφώς μικρότερος.

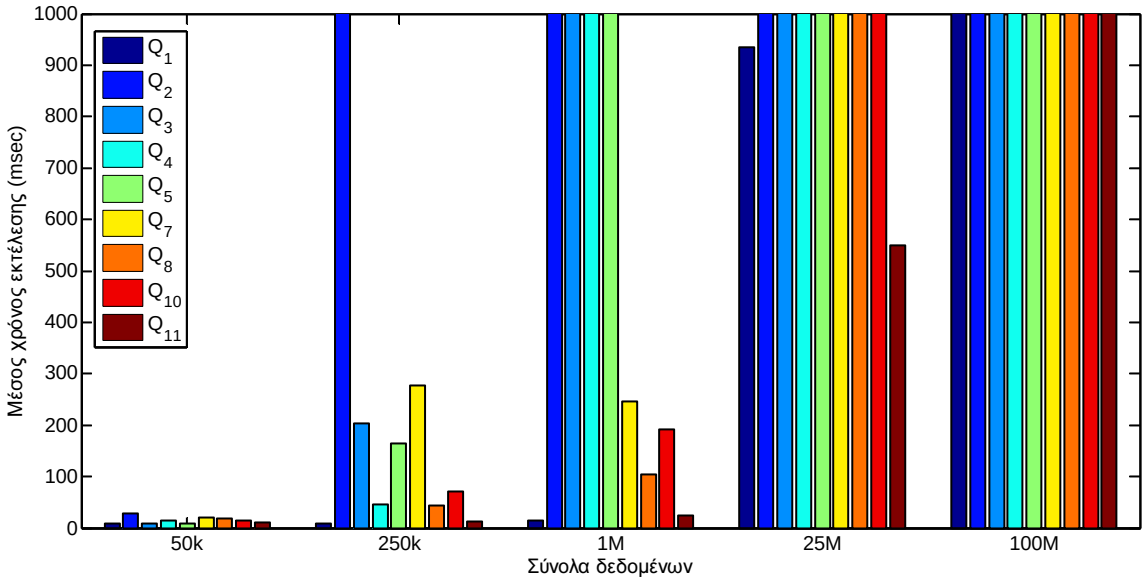
Το Q_7 είναι ένα από τα πιο σύνθετα ερωτήματα, καθώς περιέχει διπλή εμφώλευση του τελεστή OPTIONAL. Εντούτοις, ο χρόνος εκτέλεσής του διατηρείται σε ικανοποιητικά επίπεδα δεδομένου του ότι το εξωτερικό προαιρετικό πρότυπο γράφου αναγνωρίζεται ως ένας BDPLeftJoin τελεστής. γεγονός που οδηγεί στην αποφυγή δύο αριστερών εξωτερικών συνενώσεων. Βέβαια, όπως και στην περίπτωση του Q_3, ο εξωτερικός OPTIONAL τελεστής θα οδηγήσει αναπόφευκτα σε μια αριστερή εξωτερική συνένωση πινάκων, ενώ και τα δύο πρότυπα γράφου του Q_7 δεν έχουν τη μορφή αστέρα, οπότε κάποιες συνενώσεις πινάκων είναι απαραίτητες. Παρομοίως, το Q_8 αναγνωρίζεται ως ένας BDPLeftJoin τελεστής, γεγονός που οδηγεί στην αποφυγή 4 αριστερών εξωτερικών συνενώσεων, όμως και πάλι κάποιες εσωτερικές συνενώσεις πινάκων είναι απαραίτητες λόγω της μορφής του προτύπου γράφου και της χρήσης της ιδιότητας `bsbm:reviewFor`.

Ο χαμηλότερος χρόνος εκτέλεσης επιτυγχάνεται για τα Q_2, Q_{10}, ο χρόνος εκτέλεσης των οποίων παρουσιάζει ελάχιστη αύξηση καθώς αυξάνεται το μέγεθος της ΒΔ. Τα συγκεκριμένα ερωτήματα αναδεικνύουν τα οφέλη που συνδέονται με την αξιοποίηση κοινών μεταβλητών μεταξύ προτύπων τριάδας. Αυτό ισχύει ιδιαίτερα για την περίπτωση του Q_2, το οποίο, αν και περιέχει το μεγαλύτερο πρότυπο γράφου από όλα τα SPARQL ερωτήματα καθώς και 3 διαδοχικούς OPTIONAL τελεστές, επανεγγράφεται σε ένα επίπεδο SQL ερώτημα με μόλις 3 εσωτερικές συνενώσεις.

Για να επιβεβαιώσουμε τα οφέλη που αποκομίζουμε από την αποφυγή συνενώσεων, εκτελούμε το ίδιο μείγμα ερωτημάτων απενεργοποιώντας το κομμάτι της διαδικασίας που αξιοποιεί τις $s \bowtie s$ και $s \bowtie o$ συνενώσεις μεταξύ προτύπων τριάδων για αυτό το σκοπό. Ακόμα και χωρίς τις συγκεκριμένες βελτιστοποιήσεις, ο αλγόριθμος αναμένεται να αποδίδει καλύτερα από αυτόν του [77], όπου, όπως είδαμε και στην ενότητα 5.1, η συνένωση προτύπων τριάδας οδηγεί σε ένα εμφωλευμένο ερώτημα αντί να εμπλουτίσει το FROM τμήμα του SQL ερωτήματος. Η ίδια διαπίστωση ισχύει και για όλους τους υπόλοιπους αλγορίθμους που ακολουθούν την ίδια στρατηγική [57, 67, 84, 112, 134].

Οι μέσοι χρόνοι εκτέλεσης των 9 θεωρούμενων SPARQL ερωτημάτων όταν δεν αξιοποιούνται $s \bowtie s$ και $s \bowtie o$ συνενώσεις μεταξύ προτύπων τριάδων απεικονίζονται στο σχήμα 5.5. Για κάθε εκτελούμενο ερώτημα, τίθεται ένα μέγιστο χρονικό περιθώριο απόκρισης (timeout) 30 sec, ενώ για λόγους απεικόνισης στο σχήμα 5.5 οι χρόνοι έχουν ψαλιδιστεί στο 1 sec.

Ήδη από το σύνολο των 250.000 τριάδων, είναι εμφανής η ραγδαία αύξηση του μέσου χρόνου εκτέλεσης για τα περισσότερα ερωτήματα, ενώ για το Q_2 υπάρχουν συνεχόμενες παρελεύσεις του μέγιστου χρονικού περιθωρίου. Όπως αναφέρθηκε και προηγουμένως, το Q_2 περιέχει το μεγαλύτερο πρότυπο γράφου από όλα τα ερωτήματα, καθώς και 3 προαιρετικά πρότυπα γράφου, με αποτέλεσμα οι παραδοσιακοί αλγόριθμοι επανεγγραφής να παράγουν ένα



Σχήμα 5.5: *Μέσοι χρόνοι εκτέλεσης χωρίς βελτιστοποίηση αποφυγής συνενώσεων*

SQL ερώτημα με 11 εσωτερικές και 3 αριστερές εξωτερικές συνενώσεις. Αυτό έρχεται σε πλήρη αντίθεση με τους αντίστοιχους χρόνους εκτέλεσης που επιτυγχάνονται με την προσθήκη των βελτιστοποιήσεων, όπου το Q_2 έχει συνολικά το δεύτερο μικρότερο χρόνο εκτέλεσης από όλα τα υπόλοιπα.

Η κατάσταση γίνεται ακόμα χειρότερη στο σύνολο 1M, όπου παρατηρούνται παρελεύσεις του χρονικού ορίου και για τα Q_3, Q_4 και Q_5, ενώ στα σύνολα 25M και 100M πρακτικά ο αλγόριθμος δεν είναι εφαρμόσιμος, με εξαίρεση το Q_{11} για το 25M, το οποίο εξάλλου περιλαμβάνει και το μικρότερο αριθμό προτύπων τριάδας από όλα τα ερωτήματα. Ο αριθμός των προτύπων τριάδας n ενός ερωτήματος αποτελεί έναν από τους καθοριστικότερους παράγοντες που επηρεάζουν το χρόνο εκτέλεσης στην περίπτωση αυτή, δεδομένου ότι η επανεγγραφή θα οδηγήσει σε ένα ή περισσότερα SQL ερωτήματα με $n - 1$ συνενώσεις πινάκων.

Στη συνέχεια, συγκρίνουμε το σύστημά μας με το D2RQ, ένα από τα πλέον διαδεδομένα συστήματα αντιστοιχίας ΒΔ με RDF γράφους και το μόνο που στην παρούσα φάση υποστηρίζει R2RML αντιστοιχίες για τη δυναμική πρόσβαση στα περιεχόμενα μιας ΒΔ μέσω SPARQL[17]. Όπως αναφέρθηκε και στην ενότητα 5.1, το D2RQ ακολουθεί ελαφρώς διαφορετική στρατηγική σε σχέση με τους υπόλοιπους προταθέντες αλγορίθμους, καθώς υπολογίζει ενδιάμεσα αποτελέσματα χωρίς να μεταφράζει το συνολικό SPARQL ερώτημα σε ένα ισοδύναμο SQL.

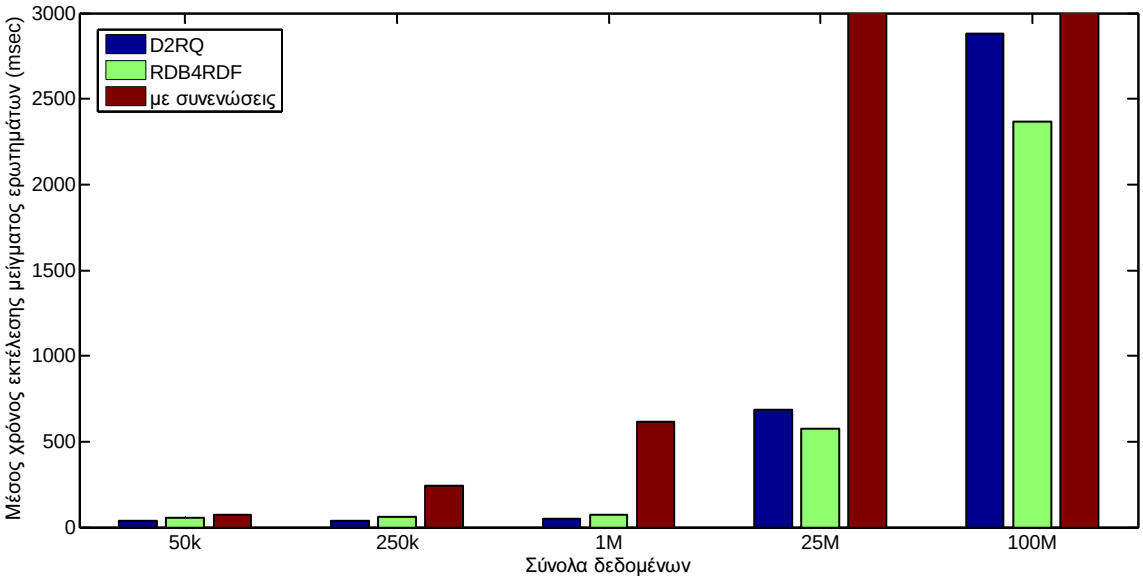
Η υποστήριξη της R2RML από τη μηχανή επανεγγραφής του D2RQ, βρίσκεται ακόμα σε πρώιμο στάδιο και χαρακτηριστικό είναι η τελευταία σχετική προέκδοση του λογισμικού[18], η οποία χρησιμοποιήθηκε στα εν λόγω πειράματα,

[17] Η δυνατότητα RDF Views του Virtuoso Universal Server προς το παρόν δεν υποστηρίζει το χαρακτηριστικό `rr:sqlQuery` της R2RML για τη χρήση όψεων στην αντιστοιχία, ένα χαρακτηριστικό απαραίτητο για την εφαρμογή της BSBM μεθοδολογίας. Επίσης, το εν λόγω σύστημα δε συνδέεται με εξωτερική σχεσιακή ΒΔ, αλλά χρησιμοποιεί δικό του σχεσιακό σύστημα αποθήκευσης.

[18] http://download.d2rq.org/d2rq-r2rml-preview-v4.tar.gz

είναι σε θέση να εκτελέσει μονάχα τα BSBM ερωτήματα Q_8, Q_{10} και Q_{11}. Ως εκ τούτου, εκτελούμε ένα μειωμένο μείγμα ερωτημάτων στο οποίο περιλαμβάνονται μόνο τα προηγούμενα ερωτήματα, ακολουθώντας την ίδια τακτική με προηγουμένως, δηλαδή την αρχική εκτέλεση 50 μειγμάτων για την ενεργοποίηση των μηχανών και 500 μειγμάτων για τον υπολογισμό του μέσου χρόνου εκτέλεσης, για καθένα από τα 50k, 250k, 1M, 25M και 100M σύνολα δεδομένων.

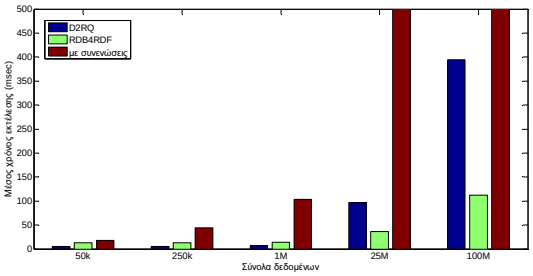
Το σχήμα 5.6 απεικονίζει το μέσο συνολικό χρόνο εκτέλεσης ενός μειωμένου μείγματος για τα D2RQ, RDB4RDF και την προσέγγιση χωρίς τη βελτιστοποίηση αποφυγής συνενώσεων. Η τελευταία, όπως είδαμε και προηγουμένως, είναι πρακτικά ανεφάρμοστη με μέσους χρόνους 70.6 και 109.7 sec για τα 25M και 100M σύνολα αντίστοιχα. Το σχήμα 5.6 αναδεικνύει την υπεροχή του D2RQ για τα μικρότερα μεγέθη ΒΔ, ενώ στα σύνολα 25M και 100M φαίνεται να υπερέχει το RDB4RDF. Το αποτέλεσμα είναι σε ένα βαθμό αναμενόμενο, καθώς ούτε το D2RQ λαμβάνει υπόψη του τις $s \bowtie s$ και $s \bowtie o$ συνενώσεις μεταξύ των προτύπων τριάδας του SPARQL ερωτήματος, αλλά πραγματοποιεί έμμεσα συνενώσεις πινάκων εκτελώντας SQL ερωτήματα με συνθήκες επιλογής (στοιχείο WHERE) που καθορίζονται από τα αποτελέσματα της εκτέλεσης προηγούμενων SQL ερωτημάτων.



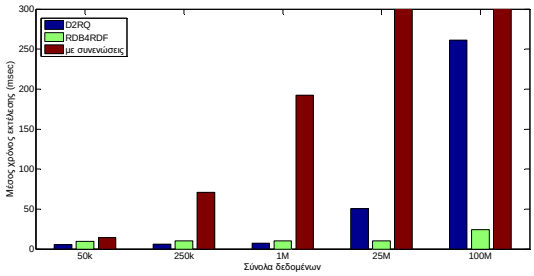
Σχήμα 5.6: *Μέσοι χρόνοι εκτέλεσης μειωμένου μείγματος Q_8, Q_{10}, Q_{11} για D2RQ και RDB4RDF*

Αναλυτικά, η σύγκριση των επιδόσεων των τριών αλγορίθμων για καθένα από τα ερωτήματα του μειωμένου μείγματος πραγματοποιείται στο σχήμα 5.7.

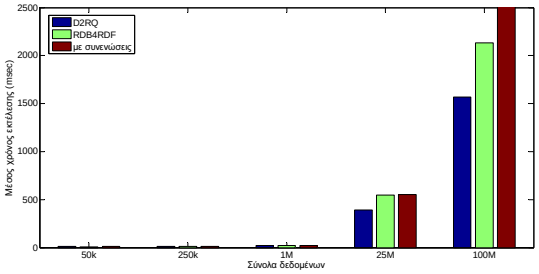
Γενικά, παρατηρείται μια υπεροχή του D2RQ σε σύγκριση με το RDB4RDF για τα ερωτήματα Q_8 και Q_{10} για σύνολα δεδομένων μέχρι 1M, ενώ οι χρόνοι εκτέλεσης του Q_{11} για τα ίδια σύνολα είναι παραπλήσιοι για τα δύο εργαλεία. Αντίθετα, στα μεγάλα σύνολα δεδομένων 25M και 100M η κατάσταση αντιστρέφεται, τα οφέλη από την αποφυγή συνενώσεων γίνονται εμφανή και το RDB4RDF εμφανίζει σαφώς χαμηλότερο χρόνο εκτέλεσης για τα Q_8, Q_{10}. Το Q_8 αποτελείται από 6 πρότυπα τριάδας και 4 διαδοχικά προαιρετικά πρότυπα



(α) *Μέσος χρόνος εκτέλεσης* Q_8



(β) *Μέσος χρόνος εκτέλεσης* Q_{10}



(γ) *Μέσος χρόνος εκτέλεσης* Q_{11}

Σχήμα 5.7: *Μέσοι χρονοι εκτέλεσης* Q_8, Q_{10}, Q_{11} *για D2RQ και RDB4RDF*

τριάδας αποτελώντας έναν BDPLeftJoin τελεστή, ο οποίος αντιμετωπίζεται αποδοτικά από τον αλγόριθμο του RDB4RDF και οδηγεί σε μόλις 2 συνενώσεις πινάκων, ενώ το Q_{10}, αν και αποτελείται από 7 πρότυπα τριάδας, οδηγεί σε ένα επίπεδο SQL ερώτημα με επίσης 2 συνενώσεις πινάκων. Ανάλογα οφέλη δεν μπορούν να υπάρξουν στην περίπτωση του Q_{11}, το οποίο όπως εξηγήσαμε προηγουμένως οδηγεί σε ένωση 11 SQL ερωτημάτων. Για το δεδομένο ερώτημα, αυτό έχει ως αποτέλεσμα όχι μόνο να υπερτερεί σημαντικά το D2RQ για τα 25M και 100M σύνολα έναντι του RDB4RDF, αλλά μάλιστα να εμφανίζει το τελευταίο παραπλήσιο χρόνο για το σύνολο 25M ακόμα και όταν αφαιρείται το κομμάτι του αλγορίθμου που αποφεύγει τις συνενώσεις πινάκων.

Οι συγκεκριμένες μετρήσεις, αν και περιορισμένης έκτασης, αποτελούν μια πρώτη ένδειξη της αποδοτικότητας του RDB4RDF ιδιαίτερα για ογκώδεις σχεσιακές ΒΔ. Η βελτίωση στο χρόνο εκτέλεσης εμφανίζεται κυρίως σε περιπτώσεις SPARQL ερωτημάτων με πρότυπα γράφου που εμφανίζουν μεγάλο αριθμό $s \bowtie s$ ή και $s \bowtie o$ συνενώσεων ή διαδοχικά προαιρετικά πρότυπα τριάδας με το ίδιο υποκείμενο. Τέτοια SPARQL ερωτήματα και, ιδιαίτερα, ερωτήματα που έχουν τη μορφή αστέρα είναι πολύ συνηθισμένα στην πράξη καθώς επιτρέπουν την ανάκτηση βασικής πληροφορίας για μια οντότητα. Υπενθυμίζεται ότι προκειμένου να είναι εφαρμόσιμη η προτεινόμενη βελτιστοποίηση, πρέπει οι κοινοί όροι των προτύπων τριάδων να προέρχονται από τον ίδιο λογικό πίνακα και μάλιστα, να παράγονται από ένα σύνολο στηλών που αποτελεί κλειδί αυτού του πίνακα.

5.5 Συμπεράσματα και μελλοντική εργασία

Σε αυτό το κεφάλαιο, μελετήθηκε το πρόβλημα της δυναμικής πρόσβασης στα περιεχόμενα σχεσιακών ΒΔ μέσω SPARQL ερωτημάτων, υπό την παρου-

σία R2RML αντιστοιχιών. Οι μέθοδοι δυναμικής πρόσβασης υπερέχουν σαφώς έναντι στατικών μεθόδων, καθώς δε χρειάζονται κάποια μέθοδο συγχρονισμού μεταξύ των σχεσιακών δεδομένων και του RDF γράφου ούτε και επιπρόσθετο αποθηκευτικό χώρο για την αποθήκευση του τελευταίου. Προτάθηκε μια αρχιτεκτονική ενός ολοκληρωμένου συστήματος αντιστοιχίας σχεσιακής ΒΔ και RDF γράφων και οντολογιών, της οποίας ακρογωνιαίο κομμάτι αποτελεί η μηχανή επανεγγραφής SPARQL ερωτημάτων σε ισοδύναμα SQL. Ο αλγόριθμος που παρουσιάστηκε είναι, από όσο γνωρίζουμε, ο πρώτος πλήρης αλγόριθμος SPARQL-σε-SQL επανεγγραφής που υποθέτει την ύπαρξη R2RML αντιστοιχιών. Βασικός στόχος του εν λόγω αλγορίθμου είναι η παραγωγή ενός κατά το δυνατόν επίπεδου SQL ερωτήματος με τον ελάχιστο αριθμό συνενώσεων πινάκων, σε αντίθεση με τους μέχρι τώρα προταθέντες αλγορίθμους SPARQL-σε-SQL επανεγγραφής που παράγουν μη αποδοτικά ερωτήματα με υψηλό βαθμό εμφώλευσης και μεγάλο αριθμό συνενώσεων. Ο προτεινόμενος αλγόριθμος χρησιμοποιεί ένα SQL μοντέλο, το οποίο και εμπλουτίζεται κατά τη διαδικασία επανεγγραφής, αντί να εμφωλεύονται αυτούσια SQL ερωτήματα το ένα μέσα στο άλλο.

Οι μετρήσεις που πραγματοποιήθηκαν αποκάλυψαν την υπεροχή του προτεινόμενου αλγορίθμου έναντι παραδοσιακών αλγορίθμων επανεγγραφής, οι οποίοι δεν πραγματοποιούν βελτιστοποιήσεις για την αποφυγή συνενώσεων μεταξύ πινάκων. Η υπεροχή αυτή είναι εμφανής στην περίπτωση σχεσιακών ΒΔ μεγάλου όγκου για SPARQL ερωτήματα με πρότυπα γράφου που έχουν τη μορφή αστέρα ή τουλάχιστον μεγάλο αριθμό $s \bowtie s$ συνενώσεων. Επίσης, η απόδοση του προτεινόμενου αλγορίθμου φαίνεται να είναι συγκρίσιμη με αυτή του ώριμου εργαλείου D2RQ για μικρά σύνολα δεδομένων μέχρι και 1 εκατομμυρίου τριάδων, ενώ υπερέχει για μεγαλύτερα σύνολα δεδομένων της τάξης των 25 και 100 εκατομμυρίων τριάδων.

Μια αδυναμία του προτεινόμενου αλγορίθμου αφορά σε στοιχεία της R2RML (όπως π.χ. οι εκφράσεις προτύπων) και σε συναρτήσεις της SPARQL, που δεν έχουν αντίστοιχό τους στην SQL. Η αντιμετώπιση αυτών των χαρακτηριστικών πραγματοποιείται στο τελευταίο στάδιο του αλγορίθμου, κατά την επεξεργασία των αποτελεσμάτων της εκτέλεσης των SQL ερωτημάτων, γεγονός που επιδρά αρνητικά στην απόδοση του συστήματος ιδιαίτερα σε περιπτώσεις ΒΔ μεγάλου όγκου. Αν και ο τρέχων αλγόριθμος μπορεί εύκολα να προσαρμοστεί ώστε να θεωρεί ότι τα εκτός-SQL χαρακτηριστικά θα είναι υλοποιημένα στη ΒΔ με τη μορφή αποθηκευμένων διαδικασιών, θεωρούμε ότι μια τέτοια υπόθεση απέχει από το ιδανικό σενάριο λειτουργίας, στο οποίο η ΒΔ δεν πρέπει να αλλοιώνεται με την προσθήκη στοιχείων άσχετων με το πεδίο ενδιαφέροντος που αυτή μοντελοποιεί. Αντίθετα, ο αλγόριθμος θα μπορούσε να προσαρμοστεί με τέτοιον τρόπο ώστε, όπου χρειάζεται, να εκτελεί ενδιάμεσα SQL ερωτήματα, να δημιουργεί SPARQL αντιστοιχίες από τα αποτελέσματά τους και να τις τροφοδοτεί στα επόμενα στάδια επεξεργασίας του SPARQL δέντρου, όπως προτείνεται στο [94] και πραγματοποιείται στο D2RQ.

Επίσης, αφήνουμε ως μελλοντική εργασία μια πιο εξαντλητική διερεύνηση όλων των δυνατών SPARQL εκφράσεων που μπορούν να χρησιμοποιηθούν εντός ενός φίλτρου, όπως και των υπόλοιπων SPARQL τελεστών που δεν αναφέρθηκαν σε αυτό το κεφάλαιο (π.χ. BIND, VALUES). Παράλληλα, στοχεύουμε σε μια πιο εκτεταμένη αξιολόγηση του αλγορίθμου και τη σύγκρισή του με

περισσότερα εργαλεία δυναμικής πρόσβασης μέσω SPARQL. Σε αυτή την κατεύθυνση, πιθανώς θα παρουσίαζε ενδιαφέρον ο ορισμός μιας νέας πρότυπης διαδικασίας αξιολόγησης της επίδοσης μηχανών SPARQL-σε-SQL επανεγγραφής, περισσότερο εστιασμένη σε χαρακτηριστικά της R2RML από ό,τι η μεθοδολογία του Berlin SPARQL Benchmark. Η νέα αυτή μεθοδολογία θα πρέπει να παρέχει, πέρα από ένα μείγμα SPARQL ερωτημάτων, και ένα σύνολο αντιστοιχιών κλιμακούμενης πολυπλοκότητας για τη λεπτομερέστερη αξιολόγηση των επιδόσεων της μηχανής επανεγγραφής.

Αξίζει να σημειωθεί ότι οι προτεινόμενες βελτιστοποιήσεις δε λαμβάνουν υπόψη τους στατιστικά στοιχεία πινάκων της ΒΔ και συνεπώς, ακολουθείται μια ενιαία στρατηγική εκτέλεσης, ανεξαρτήτως της υποκείμενης ΒΔ. Μελλοντικά, στόχος είναι η διερεύνηση μεθόδων που θα προσαρμόζουν το πλάνο εκτέλεσης ενός δοθέντος SPARQL ερωτήματος ανάλογα με την εκτίμηση του μεγέθους κάθε πίνακα της ΒΔ, έτσι ώστε να λαμβάνεται δυναμικά μια απόφαση σχετικά με το αν υπάρχει όφελος από την εφαρμογή των μετασχηματισμών που προτάθηκαν στην παράγραφο 5.3.2.

Από εκεί και πέρα, τα επόμενα βήματα στο πλαίσιο της συγκεκριμένης ερευνητικής κατεύθυνσης υπαγορεύονται ουσιαστικά από την προτεινόμενη αρχιτεκτονική και τα υποσυστήματα που αυτή περιλαμβάνει. Ένα ανοικτό ερευνητικό θέμα είναι η *ενημέρωση εικονικών RDF γράφων* ή, με άλλα λόγια, η ενημέρωση σχεσιακών ΒΔ μέσω SPARQL Update αιτημάτων υπό την παρουσία R2RML αντιστοιχιών. Το πρόβλημα αυτό είναι παρόμοιο με το κλασικό πρόβλημα της ενημέρωσης όψεων σχεσιακών ΒΔ, στόχος του οποίου είναι η εύρεση της βέλτιστης – σύμφωνα με δεδομένα κριτήρια – πράξης ενημέρωσης στο σχήμα μιας ΒΔ, η οποία επιτυγχάνει το επιθυμητό αποτέλεσμα σε μια ορισμένη όψη του σχήματος. Πλήθος σχετικών εργασιών έχει ασχοληθεί με το συγκεκριμένο θέμα και ένας από τους μελλοντικούς στόχους μας είναι η εφαρμογή των σχετικών αποτελεσμάτων στην περίπτωση της R2RML και της SPARQL Update. Συνοπτικά, οραματιζόμαστε δύο τρόπους λειτουργίας του υποσυστήματος ενημέρωσης: ο πρώτος θα θεωρεί μόνο R2RML αντιστοιχίες μιας περιορισμένης μορφής οι οποίες θα εξασφαλίζουν ότι για κάθε πιθανό SPARQL Update αίτημα, υπάρχει μοναδική λογική ενημέρωση της σχεσιακής ΒΔ, ενώ ο δεύτερος θα χρησιμοποιεί μια κλασική R2RML αντιστοιχία, επιτρέποντας συγκεκριμένα SPARQL Update αιτήματα και απορρίπτοντας εκείνα τα αιτήματα που δε συνεπάγονται μοναδική λογική ενημέρωση της ΒΔ. Ο αλγόριθμος επανεγγραφής των SPARQL Update αιτημάτων σε ισοδύναμα SQL DML αιτήματα θα βασίζεται σε μεγάλο βαθμό στον αλγόριθμο που παρουσιάστηκε σε αυτό το κεφάλαιο, δεδομένου του ότι η SPARQL Update χρησιμοποιεί την ίδια φιλοσοφία ταιριάσματος προτύπων γράφου με τη SPARQL.

Ένα άλλο ενδιαφέρον ζήτημα αποτελεί η *πραγματοποίηση συλλογισμού σε εικονικούς RDF γράφους*. Η χρήση υπαρχόντων εργαλείων συλλογισμού για αυτό το σκοπό δεν είναι εφικτή, καθώς ο RDF γράφος δεν είναι διαθέσιμος σε φυσική μορφή. Η ενδεδειγμένη λύση σε αυτή την περίπτωση είναι η επανεγγραφή του αρχικού SPARQL ερωτήματος έτσι ώστε να ενσωματώνει τα οντολογικά αξιώματα τα οποία εμπλέκουν όρους που χρησιμοποιεί ο RDF γράφος. Η τακτική αυτή έχει αποδειχθεί ότι μπορεί να υποκαταστήσει το συλλογισμό με οντολογίες εκφραστικότητας μικρότερης ή ίσης με αυτήν της OWL QL, αλλά μέχρι στιγμής δεν έχει εφαρμοστεί σε συστήματα SPARQL-σε-SQL

επανεγγραφής ούτε έχει εξεταστεί η επίδραση των R2RML αντιστοιχιών σε μια τέτοια διαδικασία. Επιπλέον, ειδικά για την περίπτωση RDFS οντολογιών, όπου ο συνεπαγόμενος RDF γράφος μπορεί να υλοποιηθεί, στόχος μας είναι η εξέταση μιας εναλλακτικής μεθόδου συλλογισμού, στην οποία τα αξιώματα μιας RDFS οντολογίας θα λαμβάνονται υπόψη για την παραγωγή ενός «συνεπαγόμενου» R2RML γράφου αντιστοιχίας, ο οποίος θα μπορεί να χρησιμοποιηθεί από τον αλγόριθμο SPARQL-σε-SQL επανεγγραφής για την επερώτηση και συνεπαγόμενων RDF προτάσεων.

Η *ανακάλυψη αντιστοιχιών* του σχήματος αλλά και του περιεχομένου της ΒΔ με εξωτερικές οντολογίες και λεξιλόγια του Σύννεφου Συνδεδεμένων Δεδομένων αποτελεί ένα ακόμη ανοικτό θέμα με το οποίο σκοπεύουμε να ασχοληθούμε μελλοντικά, προκειμένου να διευκολύνουμε την παραγωγή πραγματικά Συνδεδεμένων Δεδομένων από το περιεχόμενο μιας σχεσιακής ΒΔ. Στο πλαίσιο αυτό, θα εξεταστεί η εφαρμογή γνωστών τεχνικών λεξικολογικής και δομικής ομοιότητας που έχουν προταθεί και εφαρμοστεί στα προβλήματα αντιστοιχίας σχημάτων (schema matching), ευθυγράμμισης οντολογιών (ontology alignment) και διασύνδεσης εγγραφών (record linkage).

Τέλος, μέρος των μελλοντικών στόχων αποτελεί η ενσωμάτωση της δυνατότητας ορισμού κανόνων για την αυτόματη παραγωγή μιας οντολογίας από το σχεσιακό σχήμα και, κατ' επέκταση, την αυτόματη παραγωγή μιας αντιστοιχίας που θα αναγνωρίζει πιο σύνθετες σχεσιακές δομές από ό,τι η Άμεση Αντιστοιχία. Επίσης, η δυνατότητα αυτόματης ενημέρωσης της αντιστοιχίας όταν μεταβάλλεται το σχεσιακό σχήμα αποτελεί ένα ακόμα χαρακτηριστικό που πρέπει να διαθέτει ένα πλήρες σύστημα αντιστοιχίας και για το σκοπό αυτό, σκοπεύουμε να προσαρμόσουμε κατάλληλα σχετικές προσεγγίσεις που έχουν προταθεί ως λύση στο πρόβλημα της προσαρμογής αντιστοιχιών μεταξύ σχεσιακών σχημάτων.

Κεφάλαιο 6

Διαχείριση και επεξεργασία αισθητήριων δεδομένων με χρήση τεχνολογιών Σημασιολογικού Ιστού

Περιεχόμενα

Στο παρόν κεφάλαιο, ορίζεται ένα γενικό πλαίσιο διαχείρισης δεδομένων δικτύων αισθητήρων με χρήση τεχνολογιών Σημασιολογικού Ιστού και περιγράφεται μια επέκταση σε ένα ήδη υλοποιημένο σύστημα διαχείρισης και σημασιολογικής επεξεργασίας αισθητήριων δεδομένων, η οποία το ενισχύει με την ικανότητα αποδοτικότερου χειρισμού ροών δεδομένων. Όπως αναφέρθηκε και στην ενότητα 2.3, τα δίκτυα αισθητήρων αποτελούν πλέον μέρος της καθημερινής ζωής, χρησιμοποιούμενα σε μεγάλο εύρος εφαρμογών, από εφαρμογές υγείας, παρακολούθησης χώρων και κυκλοφοριακού προγραμματισμού, μέχρι στρατιωτικές και οικονομικές εφαρμογές. Παγκοσμίως, ο αριθμός εγκατεστημένων δικτύων αισθητήρων αυξάνεται συνεχώς, προκαλώντας ταυτόχρονη έκρηξη στον όγκο των διαθέσιμων δεδομένων που παράγονται από αυτά. Την ίδια στιγμή, έχει γίνει πολύ εύκολη η εξαγωγή και ο διαμοιρασμός δεδομένων από δίκτυα αισθητήρων με τη βοήθεια εφαρμογών και υποδομών όπως το Cosm[1], το SensorMap[2] και το μεσισμικό GSN[3].

Εντούτοις, φαίνεται να βρισκόμαστε σε μια κατάσταση όπου «υπάρχουν πολλά δεδομένα αλλά λίγη γνώση» [176]. Η κατάσταση αυτή δημιουργεί την

[1] https://cosm.com/
[2] http://atom.research.microsoft.com/sensewebv3/sensormap/
[3] http://sourceforge.net/apps/trac/gsn/

ανάγκη για επινόηση μεθόδων και ανάπτυξη συστημάτων που μπορούν να επεξεργαστούν μεγάλο όγκο δεδομένων και να εξάγουν χρήσιμα γεγονότα υψηλού επιπέδου, τα οποία έχουν μεγαλύτερη σημασία για τους τελικούς χρήστες των παραπάνω εφαρμογών από ό,τι τα πρωτογενή ανεπεξέργαστα δεδομένα που παράγονται από τις αισθητήριες συσκευές ενός δικτύου. Παράλληλα, η έλλειψη κοινών προτύπων επικοινωνίας και αναπαράστασης μειώνει τα περιθώρια ολοκλήρωσης διαφορετικών δικτύων και δυσχεραίνει το συνδυασμό των δεδομένων που προέρχονται από αυτά, δυσκολεύοντας τη σύγκριση και τη συσχέτιση συμβάντων που εντοπίζονται από διαφορετικά δίκτυα. Στην ενότητα 6.2, προτείνεται ένα πλαίσιο διαχείρισης και επεξεργασίας δεδομένων δικτύων αισθητήρων που αξιοποιεί τεχνολογίες Σημασιολογικού Ιστού και την ευελιξία που αυτές προσφέρουν, ώστε να επιτευχθεί διαλειτουργικότητα ετερογενών δικτύων αισθητήρων και αναγνώριση συμβάντων υψηλού επιπέδου μέσω εφαρμογής διαδικασιών συλλογισμού.

Η προτεινόμενη αρχιτεκτονική υιοθετείται από το σύστημα σημασιολογικής επεξεργασίας πολυμεσικών αισθητήριων δεδομένων ΠΡΙΑΜΟΣ [113], το οποίο περιγράφεται συνοπτικά στην ενότητα 6.3 και το οποίο χρησιμοποιεί συντακτικούς και σημασιολογικούς κανόνες για τη μετατροπή πρωτογενών δεδομένων, μορφοποιημένων σε XML, σε RDF δεδομένα που χρησιμοποιούν όρους από μια κατάλληλη οντολογία εφαρμογής. Εντούτοις, ένα από τα βασικά μειονεκτήματα του ΠΡΙΑΜΟΣ είναι η μειωμένη απόδοσή του με την πάροδο του χρόνου λειτουργίας του, λόγω της συσσώρευσης μεγάλου όγκου αισθητήριων δεδομένων. Για την αντιμετώπιση αυτής της αδυναμίας, στην ενότητα 6.4, παρουσιάζεται μια επέκταση του ΠΡΙΑΜΟΣ, στην οποία παρέχεται η δυνατότητα ορισμού παραθύρων επί της εισερχόμενης ροής αισθητήριων παρατηρήσεων. Η συγκεκριμένη επέκταση εισάγει μάλιστα και μια νέα κατηγορία παραθύρου, η οποία προσαρμόζει τα κλασικά παράθυρα – γνωστά από το πρόβλημα της διαχείρισης ροών δεδομένων (παράγραφος 2.3.2) – στο πλαίσιο του Σημασιολογικού Ιστού.

6.1 Σχετικές εργασίες

Ένα ολοκληρωμένο και λειτουργικό σύστημα σημασιολογικής επεξεργασίας δικτύων αισθητήρων αποτελείται από πλήθος συστατικών και υποσυστημάτων που συνεργάζονται μεταξύ τους, ώστε να οργανώσουν με αποδοτικό τρόπο τον τεράστιο όγκο των παραγόμενων δεδομένων και να δημιουργήσουν από αυτά χρήσιμη για τον τελικό χρήστη πληροφορία και γνώση. Τέτοια συστατικά περιλαμβάνουν εννοιολογικά μοντέλα και οντολογίες που περιγράφουν με τυπικό τρόπο το γνωστικό πεδίο ενδιαφέροντος στο οποίο εντάσσεται η χρήση των δεδομένων του δικτύου αισθητήρων, μηχανές συλλογισμού για εξαγωγή πληροφορίας που υπονοείται από το συνδυασμό οντολογικών αξιωμάτων και παραγόμενων δεδομένων, μεθόδους και συστήματα σημασιολογικής επισημείωσης των πρωτογενών δεδομένων, αλλά και συστήματα διαχείρισης ροών RDF δεδομένων. Στην τρέχουσα ενότητα, παρουσιάζουμε τη σχετική βιβλιογραφία, η οποία περιλαμβάνει εργασίες που εστιάζουν αποκλειστικά σε κάποιο από τα παραπάνω θέματα, με τα οποία ασχολείται το παρόν κεφάλαιο, αλλά και εργασίες που προτείνουν ολοκληρωμένες αρχιτεκτονικές και συστήματα σημασιολογικής επεξεργασίας.

Γενικά πλαίσια και αρχιτεκτονικές σημασιολογικής επεξεργασίας αισθητήριων δεδομένων

Μέχρι σήμερα, έχει προταθεί στη βιβλιογραφία μεγάλος αριθμός γενικών αρχιτεκτονικών και πλαισίων που προτείνουν μια πολυεπίπεδη οργάνωση των λειτουργιών που απαιτούνται για την ανάκτηση, επεξεργασία και διαχείριση δεδομένων από ημιδομημένες πηγές δεδομένων, όπως είναι τα δίκτυα αισθητήρων. Κάποια από τα προταθέντα πλαίσια είναι αμιγώς θεωρητικά, ενώ άλλα συνοδεύονται και από αντίστοιχη υλοποίηση. Καθώς τα πρωτογενή αισθητήρια δεδομένα από μόνα τους δύσκολα επιδέχονται ερμηνείας, υπάρχει μια έντονη τάση προς τη χρήση τεχνολογιών Σημασιολογικού Ιστού για την απόδοση σημασίας σε αυτά και αρκετά από τα προταθέντα πλαίσια ενσωματώνουν σχετικές τεχνικές. Εκτός αυτού, η χρήση τεχνολογιών Σημασιολογικού Ιστού στη διαχείριση δικτύων αισθητήρων διευκολύνει την ανακάλυψη και πρόσβαση σε ετερογενείς αισθητήρες, καθώς και την αυτοματοποιημένη ανάπτυξη και επαναχρησιμοποίηση σύνθετων εφαρμογών χρήστη.

Δύο πρώιμες αρχιτεκτονικές [128, 130] προτείνουν το συνδυασμό βασικών υπηρεσιών με διαθέσιμη σημασιολογία για το δίκτυο αισθητήρων και την υπερκείμενη εφαρμογή. Οι βασικές αυτές υπηρεσίες (όπως π.χ. ένας εκτιμητής κίνησης) αναλαμβάνουν την επεξεργασία των πρωτογενών δεδομένων. Ο ορισμός των υπηρεσιών στο [130] έχει τη μορφή προτάσεων λογικής πρώτης τάξης που χρησιμοποιούν όρους από συγκεκριμένο λεξιλόγιο, γεγονός που επιτρέπει την κατάλληλη σύνθεση και προγραμματισμό αυτών των υπηρεσιών, προκειμένου να δοθούν απαντήσεις στα ερωτήματα των τελικών χρηστών. Παρόμοια είναι και η προσέγγιση του [128], όπου γνώση σχετική με το δίκτυο αισθητήρων κωδικοποιείται σε μια γλώσσα, βασισμένη σε λογική πρώτης τάξης και ένας αλγόριθμος επανεγγραφής ερωτημάτων χρησιμοποιείται για την απάντηση επερωτήσεων.

Κάποιες θεωρητικές, κατά κύριο λόγο, αρχιτεκτονικές που δίνουν ιδιαίτερο βάρος στη σημασιολογική επεξεργασία των αισθητήριων δεδομένων παρουσιάζονται στα [106, 127, 140]. Στο [106], προτείνεται μια διαστρωματωμένη αρχιτεκτονική με τέσσερα επίπεδα: α) το επίπεδο δεδομένων μέσω του οποίου παρέχεται πρόσβαση σε ανεπεξέργαστες αισθητήριες παρατηρήσεις, β) το οντολογικό επίπεδο, στο οποίο η σημασία των παρατηρήσεων ορίζεται μέσω σύνδεσης με κατάλληλες οντολογίες, γ) το επίπεδο επεξεργασίας, όπου εφαρμόζονται κανόνες και διαδικασίες συλλογισμού στα δεδομένα που προέρχονται από το προηγούμενο επίπεδο και δ) το επίπεδο εφαρμογής, το οποίο περιλαμβάνει πλήθος εφαρμογών πελάτη που χρησιμοποιούν τα δεδομένα του δικτύου αισθητήρων.

Το SWAP [140] είναι ένα πλαίσιο διαχείρισης ενός δικτύου αισθητήρων, που συνδυάζει τεχνολογία πρακτόρων λογισμικού και οντολογικές δομές για την ανακάλυψη, επαναχρησιμοποίηση και συνδυασμό αισθητήριων παρατηρήσεων. Η γενική αρχιτεκτονική που προτείνεται, αν και τριών επιπέδων, μοιάζει με αυτήν του [106] και εκτός από το χαμηλότερο επίπεδο πρόσβασης σε παρατηρήσεις αισθητήρων οι οποίες θεωρούνται ότι είναι εκφρασμένες σε όρους κατάλληλης οντολογίας, περιέχει επίσης ένα επίπεδο γνώσης στο οποίο δρουν πράκτορες λογισμικού που επεξεργάζονται τα αισθητήρια δεδομένα, εφαρμόζοντας επί αυτών απλούς ή και περισσότερο σύνθετους αλγορίθμους. Τέλος, το επίπεδο εφαρμογών περιέχει πράκτορες που παρέχουν διεπαφές προς τον

ανθρώπινο χρήστη, συνδυάζοντας εξόδους από πράκτορες του επιπέδου γνώσης.

Κινούμενο στην ίδια λογική, το ES3N [127] αποτελεί ένα υλοποιημένο σύστημα για τη σημασιολογική διαχείριση δεδομένων σε ένα δίκτυο ετερογενών αισθητήρων. Τα δεδομένα συλλέγονται και διατηρούνται σε ένα RDF σύστημα αποθήκευσης, στο οποίο μπορούν να υποβάλλονται SPARQL ερωτήματα. Παρομοίως, στο [157] περιγράφεται μια υλοποίηση ενός σημασιολογικού δικτύου αισθητήρων, όπου σε ένα triple store σύστημα συγκεντρώνονται σε RDF μορφή μετρήσεις, μεταδεδομένα και προβλέψεις σχετικές με την κατάσταση αισθητήρων. Τα δεδομένα αυτά ανακτώνται από το δίκτυο αισθητήρων μέσω διεπαφών Ιστού, ενώ η αναζήτηση σημασιολογικών οντοτήτων πραγματοποιείται μέσω SPARQL ερωτημάτων που τίθενται στο triple store της συγκεκριμένης υλοποίησης. Διαφορετικός είναι ο στόχος του [146], όπου προτείνεται ένα γενικό πλαίσιο για την εκμετάλλευση της σημασίας αισθητήριων δεδομένων σε λειτουργίες ενός δικτύου αισθητήρων, όπως η δρομολόγηση και η επεξεργασία ερωτημάτων.

Τέλος, στο [107] παρουσιάζεται μια αρχιτεκτονική που προτείνει την εισαγωγή ενός υπερκείμενου στρώματος υπηρεσιών που θα προσδώσει αντίληψη περιβάλλοντος στις υπηρεσίες-πρότυπα του SWE. Οι υπηρεσίες αυτές είναι υπεύθυνες: α) για τη σημασιολογική επισημείωση – δηλαδή τη σύνδεση με έννοιες μιας οντολογίας – των SWE υπηρεσιών αλλά και των αισθητήριων παρατηρήσεων, β) την αποθήκευση και επερώτηση των χρησιμοποιούμενων οντολογιών και γ) την πραγματοποίηση συλλογισμού στις επισημειωμένες αισθητήριες μετρήσεις.

Στην ενότητα 6.2, παρουσιάζεται μια αρχιτεκτονική τριών επιπέδων βασισμένη σε πρότυπες τεχνολογίες Σημασιολογικού Ιστού για την αποτελεσματική διαχείριση και επεξεργασία αισθητήριων παρατηρήσεων, που μοιράζεται αρκετά κοινά στοιχεία με κάποια από τα παραπάνω πλαίσια.

Μοντέλα και οντολογίες αναπαράστασης δικτύων αισθητήρων

Οι οντολογίες, ως εννοιολογικά μοντέλα που περιγράφουν ένα τομέα ενδιαφέροντος, αποτελούν πρωτεύον συστατικό σε αρχιτεκτονικές και συστήματα σημασιολογικής επεξεργασίας δεδομένων από δίκτυα αισθητήρων. Οι οντολογίες κωδικοποιούν τη γνώση που χρειάζονται τα συστήματα αυτά για να ερμηνεύσουν τις αισθητήριες παρατηρήσεις, να βελτιώσουν τη σημασιολογική διαλειτουργικότητα αυτών και να διευκολύνουν το συνδυασμό μετρήσεων από ετερογενείς αισθητήρες που χρησιμοποιούν διαφορετική χρονική και χωρική ανάλυση, διαφορετικές μονάδες μέτρησης και μορφότυπα.

Μια κατηγοριοποίηση οντολογιών που μπορούν να χρησιμοποιηθούν για την περιγραφή και τον διαμοιρασμό αισθητήριων παρατηρήσεων παρουσιάζεται στο [32]. Συγκεκριμένα, διακρίνονται οι εξής – επικαλυπτόμενες – κατηγορίες: α) οντολογίες συσκευών, οι οποίες παρέχουν μια ταξινομία των συσκευών που συναντώνται σε εφαρμογές αίσθησης περιβάλλοντος, β) οντολογίες περιβάλλοντος, που εστιάζουν σε περιβαλλοντική πληροφορία (π.χ. θερμοκρασία, τοποθεσία) η οποία ανακτάται συνήθως από αισθητήρες, γ) οντολογίες δεδομένων, οι οποίες ασχολούνται με τα δεδομένα που συλλέγονται από αισθητήριες συσκευές και δ) οντολογίες πεδίου που περιγράφουν ένα συγκεκριμένο

γνωστικό πεδίο, που συνήθως αποτελεί και το πεδίο εφαρμογής ενός δικτύου αισθητήρων.

Ανασκοπήσεις των σημαντικότερων οντολογιών για δίκτυα αισθητήρων πραγματοποιούνται στην τελική αναφορά της ομάδας εργασίας Semantic Sensor Network Incubator Group του W3C [26], όπως επίσης και στο [65]. Αποτέλεσμα των εργασιών αυτής της ομάδας ήταν ο ορισμός της οντολογίας Σημασιολογικού Δικτύου Αισθητήρων [64], μιας OWL οντολογίας για τη μοντελοποίηση των δυνατοτήτων αισθητήριων συσκευών, συστημάτων και διαδικασιών.

Σημασιολογική επισημείωση

Η διαδικασία συσχέτισης δεδομένων με όρους εννοιολογικών μοντέλων που ορίζουν με τρόπο τυπικό τη σημασία των πρώτων ονομάζεται σημασιολογική επισημείωση. Η σημασιολογική επισημείωση αποτελεί το απαραίτητο πρώτο βήμα σε ένα σύστημα σημασιολογικής επεξεργασίας ημιδομημένων δεδομένων και σχετίζεται με πληθώρα ωφελειών, πολλές από τις οποίες έχουν ήδη αναφερθεί, όπως η σημασιολογική διαλειτουργικότητα των δεδομένων και η δυνατότητα ολοκλήρωσής τους ακόμα και αν προέρχονται από ετερογενείς πηγές. Η σύνδεση αισθητήριων παρατηρήσεων με γνωστά μοντέλα και οντολογίες βοηθά στην αποφυγή του εγκλεισμού τους στο σύστημα από το οποίο έχουν προέλθει και επιτρέπει την επαναχρησιμοποίησή τους από εξωτερικά συστήματα και εφαρμογές. Πέρα από την επισημείωση παρατηρήσεων όμως, ιδιαίτερη αξία έχει και η έκφραση των χαρακτηριστικών ενός δικτύου αισθητήρων σε όρους γνωστών λεξιλογίων, η οποία μεταξύ άλλων διευκολύνει και την αυτοματοποίηση εσωτερικών λειτουργιών του δικτύου αισθητήρων, όπως η δρομολόγηση ή η συνάθροιση δεδομένων [157].

Η έκφραση δεδομένων από δίκτυα αισθητήρων ως Συνδεδεμένα Δεδομένα (Linked Data) επιτρέπει στους διαχειριστές αυτών των δικτύων αλλά και σε χρήστες που καταναλώνουν τα δεδομένα τους να συνδέσουν τις περιγραφές αισθητήρων με έναν απεριόριστο όγκο δεδομένων που υπάρχουν στον Ιστό. Συνεπώς, οι ενδιαφερόμενοι χρήστες είναι σε θέση να συνδυάσουν δεδομένα από το φυσικό κόσμο με θεωρητική γνώση από τον Ιστό προκειμένου να εξάγουν νέα συμπεράσματα, να συνεισφέρουν στην επιχειρηματική ευφυΐα, να δημιουργήσουν πρόσφορο έδαφος για έξυπνα περιβάλλοντα, καθώς και να υποστηρίξουν αυτοματοποιημένα συστήματα και διαδικασίες λήψης αποφάσεων [27].

Ένα σύστημα σημασιολογικής επισημείωσης δεδομένων από δίκτυα αισθητήρων, το οποίο είναι ενδεικτικό της φιλοσοφίας που ακολουθείται από την πλειοψηφία των συστημάτων της βιβλιογραφίας είναι αυτό που παρουσιάζεται στο [150]. Το συγκεκριμένο σύστημα εξάγει αισθητήριες παρατηρήσεις αλλά και περιγραφές αισθητήρων σε μορφή RDF. Τα ανεπεξέργαστα δεδομένα αντλούνται αρχικά μέσω κατάλληλων υπηρεσιών Ιστού σε μορφή CSV, μετατρέπονται σε O&M και τέλος σε RDF γράφο που χρησιμοποιεί την οντολογία GeoNames[4], προκειμένου να επιτρέψει τη σύνδεση των παραγόμενων RDF δεδομένων με άλλα σύνολα δεδομένων στο Σύννεφο Συνδεδεμένων Δεδομένων. Τα RDF δεδομένα αποθηκεύονται σε triple store σύστημα, το οποίο παρέχει ένα τελικό σημείο SPARQL για την πρόσβαση σε αυτό, ενώ έχει αναπτυχθεί

[4] http://www.geonames.org/ontology/documentation.html

και γραφική διεπαφή για τελικούς χρήστες μη εξοικειωμένους με τη SPARQL. Συστήματα που ακολουθούν παρόμοια τακτική για την επισημείωση μονάχα αισθητήριων παρατηρήσεων είναι τα Sense2Web [27] και τα συστήματα που προτείνονται στα [124, 148].

Άξια αναφοράς είναι και η προσέγγιση της SemSOS υπηρεσίας [99], η οποία αποτελεί μια σημασιολογική εκδοχή του SOS προτύπου του SWE. Αυτή βασίζεται σε μια OWL βάση γνώσης, η οποία περιέχει τις παρατηρήσεις ενός δικτύου αισθητήρων, επισημειωμένες με όρους από την O&M-OWL οντολογία καθώς και από άλλες γεωχωρικές, θεματικές και χρονικές οντολογίες. Η συγκεκριμένη βάση γνώσης ενσωματώνει και δυνατότητες συλλογισμού μέσω κανόνων, οι οποίοι παράγουν νέα γνώση από την ήδη υπάρχουσα πληροφορία. Η SemSOS υπηρεσία χρησιμοποιεί μηχανισμούς που μετατρέπουν τα SOS αιτήματα σε SPARQL ερωτήματα που τίθενται στη βάση γνώσης και στην αντίθετη κατεύθυνση, μηχανισμούς που μετατρέπουν τα SPARQL αποτελέσματα σε επισημειωμένες SensorML ή O&M περιγραφές, ως απόκριση στο αρχικό SOS αίτημα. Η υπηρεσία αυτή μπορεί να χρησιμοποιηθεί για τη σημασιολογική ανακάλυψη αισθητήρων με βάση τα χαρακτηριστικά τους ή/και την τοποθεσία τους.

Τέλος, στο [25], προτείνεται η δημοσίευση RDF ροών δεδομένων ως Συνδεδεμένα Δεδομένα και η REST[5] πρόσβαση σε αυτές μέσω HTTP. Οι ροές RDF προτάσεων δημιουργούνται ως απόκριση σε ένα συνεχές SPARQL ερώτημα, εκφρασμένο στη C-SPARQL γλώσσα και περιγράφονται με ειδικό λεξιλόγιο που προσδιορίζει π.χ. τη χρονική στιγμή της τελευταίας ενημέρωσης της ροής ή τον τύπο του χρησιμοποιούμενου παραθύρου.

Συστήματα διαχείρισης ροών

Η διαχείριση ενός μεγάλου αριθμού αισθητήριων παρατηρήσεων, οι οποίες καθίστανται διαθέσιμες σε ένα σύστημα σημασιολογικής επεξεργασίας με ραγδαίο συνήθως ρυθμό, αποτελεί ένα ακόμα πρόβλημα άξιο αναφοράς στο πλαίσιο του σημασιολογικού εμπλουτισμού ημιδομημένης πληροφορίας και ένας διόλου ευκαταφρόνητος αριθμός προσεγγίσεων έχει προταθεί μέχρι σήμερα στη σχετική βιβλιογραφία.

Το Semantic Streams [201] είναι ένα πλαίσιο σημασιολογικής επεξεργασίας που δίνει τη δυνατότητα σε χρήστες να θέτουν δηλωτικά ερωτήματα υψηλού επιπέδου σε σημασιολογικές ερμηνείες αισθητήριων δεδομένων. Η αναπαράσταση των διαθέσιμων φυσικών αισθητήρων γίνεται μέσω Prolog κανόνων που χρησιμοποιούν έναν αριθμό προκαθορισμένων κατηγορημάτων ειδικής σημασίας. Τα ερωτήματα που τίθενται από το χρήστη στο σύστημα είναι προτάσεις λογικής πρώτης τάξης που χρησιμοποιούν αυτά τα προκαθορισμένα κατηγορήματα. Για την απόδειξη αυτών των προτάσεων, εφαρμόζεται η ανάστροφη διαδικασία εκτέλεσης (backward chaining) των διαθέσιμων κανόνων. Εντούτοις, το συγκεκριμένο πλαίσιο δε φαίνεται να λαμβάνει υπόψη του τα προβλήματα και τις ιδιαιτερότητες που θέτει η άπειρη φύση μιας ροής αισθητήριων παρατηρήσεων.

[5]Το ακρωνύμιο REST (Representational State Transfer) αναφέρεται σε μια φιλοσοφία σχεδιασμού υπηρεσιών που βασίζεται στη μεταφορά περιγραφών για την κατάσταση ενός πόρου. Συνήθως, για τη μεταφορά αυτή, χρησιμοποιείται το HTTP πρωτόκολλο και οι γνωστές μέθοδοι GET, PUT, POST, DELETE.

Η αρχιτεκτονική που παρουσιάζεται στο [45] ακολουθεί παρόμοια λογική με αυτή του Semantic Streams. Βασικά συστατικά αποτελούν οι φυσικοί αισθητήρες ενός ή περισσότερων δικτύων καθώς και επεξεργαστικά στοιχεία λογισμικού που εφαρμόζουν απλούς ή σύνθετους αλγορίθμους σε δεδομένα καθορισμένου τύπου. Τα χαρακτηριστικά των αισθητήρων αλλά και των στοιχείων λογισμικού που επεξεργάζονται τις ενδείξεις των τελευταίων αναπαρίστανται με σημασιολογικές περιγραφές που χρησιμοποιούν κατάλληλες OWL οντολογίες πεδίου και οι οποίες επιτρέπουν την αυτόματη σύνθεση αισθητήρων και διαθέσιμων επεξεργαστικών στοιχείων λογισμικού ώστε να ικανοποιηθεί κάποιος στόχος υψηλού επιπέδου.

Ένας σημαντικός αριθμός προσεγγίσεων προτείνει τον ορισμό επεκτάσεων της SPARQL κατάλληλων για τον ορισμό παραθύρων και συνεχών ερωτημάτων σε ροές RDF δεδομένων. Άξιες αναφοράς είναι η Streaming SPARQL [43], η C-SPARQL [22], η CQELS [122] και η SPARQL$_{Stream}$ [50], κάθε μία εκ των οποίων υποστηρίζεται και από αντίστοιχο μοντέλο επεξεργασίας.

Σε γενικές γραμμές, θα μπορούσαμε να πούμε ότι υπάρχουν δυο ειδών στρατηγικές για τη διαχείριση άπειρων ροών RDF δεδομένων. Η πρώτη θεωρεί πεπερασμένα υποσύνολα της RDF ροής – τα γνωστά παράθυρα από το πρόβλημα της διαχείρισης ροών σχεσιακών δεδομένων – και τα αντιμετωπίζει ως στατική αμετάβλητη πληροφορία την οποία επεξεργάζεται σύμφωνα με κλασικές μεθοδολογίες αναπαράστασης γνώσης και σχετικά εργαλεία Σημασιολογικού Ιστού. Χαρακτηριστικό παράδειγμα αποτελεί το σύστημα που προτείνεται στο [22] για την αποτίμηση C-SPARQL ερωτημάτων. Στο σύστημα αυτό, χρησιμοποιείται ένα τυπικό σύστημα διαχείρισης ροών για την εφαρμογή παραθύρου επί της εισερχόμενης ροής, όπως αυτό ορίζεται στο εκτελούμενο C-SPARQL ερώτημα και στη συνέχεια, η αποτίμηση του C-SPARQL ερωτήματος γίνεται με τη χρήση μιας τυπικής SPARQL μηχανής εκτέλεσης η οποία λαμβάνει υπόψη της μονάχα τον RDF γράφο που έχει σχηματιστεί από την εφαρμογή του παραθύρου.

Η δεύτερη κατηγορία συστημάτων συνδυάζει στατική και μεταβαλλόμενη RDF πληροφορία και προσπαθεί να ανακαλύψει αποδοτικότερους μηχανισμούς εκτέλεσης συνεχών ερωτημάτων. Ένα τέτοιο σύστημα είναι το [24], το οποίο αναλαμβάνει την εκτέλεση C-SPARQL ερωτημάτων που παράλληλα αναφέρονται και σε στατικές πηγές RDF δεδομένων. Το συγκεκριμένο σύστημα διακρίνει μεταξύ στατικών και ταχέως μεταβαλλόμενων προτύπων γράφου αναθέτοντας την αποτίμηση των τελευταίων στο σύστημα διαχείρισης ροών δεδομένων STREAM [13], αφού έχει προηγηθεί επανεγγραφή του αντίστοιχου προτύπου γράφου σε ένα ισοδύναμο CQL ερώτημα[6]. Με αυτόν τον τρόπο, η συγκεκριμένη υλοποίηση συνδυάζει μια στατική RDF βάση γνώσης με ροές εικονικών RDF τριάδων, δεδομένου ότι οι εισερχόμενες ροές δεδομένων απλώς *ερμηνεύονται* ως RDF ροές, χωρίς να *υλοποιούνται* σε RDF μορφή.

Η ίδια φιλοσοφία εκτέλεσης παρουσιάζεται και στην αρχιτεκτονική του [123], με τη διαφορά ότι για την αποτίμηση του «μεταβαλλόμενου» τμήματος ενός CQELS ερωτήματος, χρησιμοποιείται μια φυσική μηχανή αποτίμησης συνεχών ερωτημάτων για RDF ροές δεδομένων. Η συγκεκριμένη μηχανή [122] χρησιμοποιεί μηχανισμούς κρυφής μνήμης (caching) και ευρετικές μεθόδους

[6] Η CQL (Continuous Query Language) είναι μια δηλωτική, βασισμένη σε SQL, γλώσσα για την υποβολή συνεχών ερωτημάτων σε ροές σχεσιακών δεδομένων.

για το μετασχηματισμό του CQELS δέντρου τελεστών σε ένα βέλτιστο πλάνο εκτέλεσης.

Ένα ακόμα σύστημα που επιτρέπει την επερώτηση ροών *εικονικών* RDF δεδομένων, αυτή τη φορά μέσω της γλώσσας SPARQL*Stream*, προτείνεται στο [50]. Η συγκεκριμένη προσέγγιση βασίζεται στην υπόθεση ότι κάθε ροή δεδομένων που προέρχεται από μια δεδομένη πηγή (π.χ. έναν αισθητήρα) αποθηκεύεται σε ένα σχεσιακό σχήμα. Μια τέτοια πρωταρχική ροή αντιστοιχείται σε μια παραγόμενη RDF ροή μέσω της γλώσσας αντιστοιχιών S_2O, η οποία αποτελεί επέκταση της γλώσσας αντιστοιχίας σχεσιακών ΒΔ και οντολογιών R_2O [29]. Η προτεινόμενη μηχανή εκτέλεσης συνεχών ερωτημάτων αναλαμβάνει την επανεγγραφή SPARQL*Stream* σε SNEEql, μια επέκταση της SQL για την επερώτηση ροών σχεσιακών δεδομένων.

Τέλος, αξίζει να αναφερθεί η προσέγγιση του [166], όπου προτείνεται μια επέκταση του RDF μοντέλου για την αναπαράσταση χρονικά μεταβαλλόμενης RDF πληροφορίας. Σύμφωνα με το χρονικά επισημειωμένο RDF μοντέλο που προτείνεται (TA-RDF), κάθε μεμονωμένος RDF πόρος μπορεί να είναι επισημειωμένος με κάποιο χρονικό ορόσημο. Εισάγεται ειδικό λεξιλόγιο για τον ορισμό TA-RDF γράφων, ορίζεται η σημασιολογία τους καθώς και μια παραλλαγή της SPARQL για την υποβολή ερωτημάτων στιγμιοτύπου σε αυτούς. Στην υλοποίηση που παρουσιάζεται στο [166], η χρονική επισημείωση ενός RDF μοντέλου δεν προσθέτει επιπλέον τριάδες, αλλά θέτει τα κατάλληλα ευρετήρια στη μηχανή αποθήκευσης RDF, ευρετηριάζοντας τους RDF πόρους σύμφωνα με το χρονικό ορόσημό τους αλλά και τη ροή στην οποία ανήκουν.

Πραγματοποίηση συλλογισμού σε ροή

Μεγαλύτερη αξία από την απλή επερώτηση ροών RDF δεδομένων έχει η πραγματοποίηση συλλογισμού σε αυτές, υπό την παρουσία χρονικά αμετάβλητης πληροφορίας, προκειμένου να αναγνωριστούν σύνθετες καταστάσεις και γεγονότα ενδιαφέροντος. Το συγκεκριμένο πρόβλημα είναι σχετικά πρόσφατο και οι προσεγγίσεις που πραγματικά αντιμετωπίζουν τις προκλήσεις του συλλογισμού σε μια άπειρη ροή είναι ολιγάριθμες.

Στο [199], παρουσιάζεται ένα σύστημα για την πραγματοποίηση συλλογισμού σε ροές δεδομένων. Η κύρια ιδέα του συστήματος βασίζεται στον προϋπολογισμό της ιεραρχίας κλάσεων της RDFS ή OWL οντολογίας της εφαρμογής, η οποία αποτελεί και το στατικό τμήμα της συνολικής βάσης γνώσης, και στην αποθήκευση αυτών σε ένα σύστημα διαχείρισης ροής, το οποίο αναλαμβάνει την εκτέλεση συνεχών ερωτημάτων συνδυάζοντας τις εισερχόμενες τριάδες με τις στατικές οντολογικές τριάδες. Επίσης, το σύστημα επαυξάνει την εισερχόμενη ροή RDF προτάσεων με νέες συνεπαγόμενες προτάσεις για συγκεκριμένα RDFS/OWL αξιώματα. Τα υποστηριζόμενα από το σύστημα συνεχή ερωτήματα είναι αρκετά περιορισμένα, καθώς είναι σε θέση να αναγνωρίσουν μόνο οντολογικά άτομα μιας δεδομένης κλάσης ενδιαφέροντος.

Μια διαφορετική πρόταση για την πραγματοποίηση συλλογισμού σε ροή RDF δεδομένων βασίζεται στη σταδιακή ενημέρωση της βάσης γνώσης καθώς RDF προτάσεις προστίθενται ή αφαιρούνται από αυτή [23]. Η κεντρική ιδέα της προσέγγισης αυτής περιλαμβάνει τον υπολογισμό των συνεπαγόμενων προτάσεων για κάθε νέα διαθέσιμη RDF πρόταση, την εύρεση των προτάσεων

που έχουν προκύψει από RDF τριάδες οι οποίες δεν είναι πλέον έγκυρες (δηλαδή, τριάδες που δεν περιλαμβάνονται στο τρέχον παράθυρο αποτίμησης του ερωτήματος) και την κατάλληλη ενημέρωση της βάσης γνώσης του συστήματος.

Στο [12] προτείνεται η EP-SPARQL, μια επέκταση της SPARQL η οποία ενσωματώνει χρονικούς τελεστές και συναρτήσεις και αντίστοιχη μηχανή εκτέλεσης, η οποία ακολουθεί μια προσέγγιση βασισμένη σε λογικούς κανόνες. Ένα EP-SPARQL ερώτημα μεταφράζεται σε ένα σύνολο Prolog κανόνων, οπότε η διαδικασία συλλογισμού πραγματοποιείται μέσω κατάλληλης μηχανής κανόνων, η οποία λαμβάνει υπόψη της και στατική γνώση, επίσης εκφρασμένη με τη μορφή Prolog κανόνων.

Τέλος, το DyKnow [96] είναι ένα μεσισμικό για το συλλογισμός σε ροές δεδομένων. Βασικά συστατικά του συστήματος είναι οι ροές και οι διαδικασίες γνώσης, με τις τελευταίες να δέχονται κατά κανόνα ως είσοδο μία ή περισσότερες ροές και να τις επεξεργάζονται δημιουργώντας μία ή περισσότερες ροές εξόδου. Το DyKnow δε χρησιμοποιεί τεχνολογίες Σημασιολογικού Ιστού για την πραγματοποίηση συλλογισμού και την εξαγωγή συμβάντων υψηλού επιπέδου, αλλά αντίθετα κάνει χρήση ενός αλγορίθμου αναγνώρισης αλληλουχιών γεγονότων, τα οποία είναι γνωστά και ως χρονικά (chronicles).

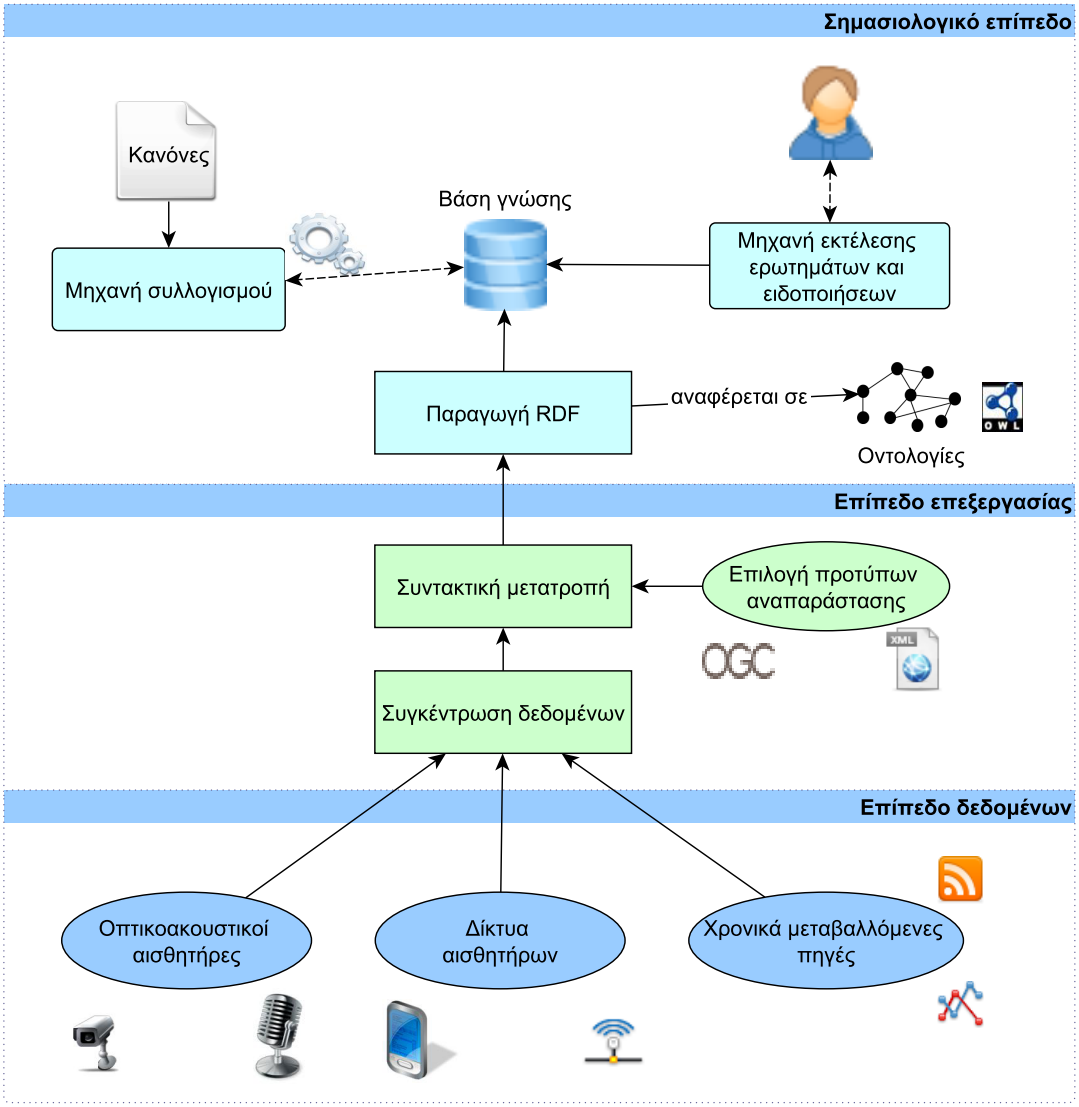
6.2 Αρχιτεκτονική σημασιολογικής επεξεργασίας δικτύων αισθητήρων

Σε αυτήν την ενότητα, παρουσιάζεται εν συντομία μια γενική αρχιτεκτονική ενός συστήματος για τη διαχείριση και σημασιολογική επεξεργασία των δεδομένων ενός δικτύου αισθητήρων. Η συγκεκριμένη αρχιτεκτονική μοιράζεται αρκετά κοινά στοιχεία με κάποιες από τις αρχιτεκτονικές και τα συστήματα που παρουσιάστηκαν στην αντίστοιχη παράγραφο της ενότητας 6.1 και τα συνδυάζει σε ένα ενοποιημένο πλαίσιο. Στόχος είναι η πρόταση μιας αρθρωτής (modular) αρχιτεκτονικής που θα λειτουργήσει ως βάση για την ανάπτυξη ευέλικτων συστημάτων επίγνωσης περιβάλλοντος (context-aware systems), τα οποία μεταξύ άλλων:

α) θα καλύπτουν την ετερογένεια των υποκείμενων δικτύων αισθητήρων,
β) θα αξιοποιούν τη σημασία του περιβάλλοντός τους και θα τη συνδυάζουν με εξωτερική γνώση προς όφελος των τελικών χρηστών,
γ) θα είναι εύκολα προσαρμόσιμα σε διαφορετικές εφαρμογές και σενάρια χρήσης και
δ) θα επιτρέπουν την επαναχρησιμοποίηση και αξιοποίηση των παραγόμενων δεδομένων από τρίτες εφαρμογές

Το συγκεκριμένο πλαίσιο έχει προταθεί στις δημοσιεύσεις [205, 206], είναι εφαρμόσιμο σε ένα ευρύ φάσμα κατηγοριών δικτύων αισθητήρων και εφαρμογών και αποτελείται από ένα ***επίπεδο δεδομένων***, όπου πραγματοποιείται η ανακάλυψη, δρομολόγηση και συλλογή των πρωτογενών δεδομένων από το υποκείμενο δίκτυο αισθητήρων, ένα ***επίπεδο επεξεργασίας***, που είναι υπεύθυνο για τη συντακτική μορφοποίηση και επεξεργασία των δεδομένων και ένα ***σημασιολογικό επίπεδο***, όπου τα δεδομένα επισημειώνονται με πληροφορία σχετική με το περιβάλλον τους, εκφρασμένη με τη μορφή σχετικών οντολογιών

και εφαρμόζονται αλγόριθμοι συλλογισμού σε αυτά για την εξαγωγή γεγονότων υψηλού επιπέδου. Η προτεινόμενη αρχιτεκτονική απεικονίζεται στο σχήμα 6.1 και τα επιμέρους επίπεδα αυτής αναλύονται στις επόμενες παραγράφους.



Σχήμα 6.1: *Αρχιτεκτονική σημασιολογικής επεξεργασίας αισθητήριων δεδομένων*

Αξίζει να σημειωθεί ότι ο διαχωρισμός που προτείνεται μέσω αυτού του γενικού πλαισίου συνεπάγεται όλα τα πλεονεκτήματα ενός αρθρωτού συστήματος, με σημαντικότερο όλων το γεγονός ότι κάθε επίπεδο μπορεί να θεωρηθεί ανεξάρτητο των υπολοίπων, έτσι ώστε αποφάσεις που αφορούν σε στρατηγικές και τρόπους υλοποίησης σε ένα επίπεδο να μην επηρεάζουν τα υπόλοιπα. Αυτό επιτρέπει την επαναχρησιμοποίηση και εφαρμογή κλασικών μεθόδων διαχείρισης δεδομένων σε δίκτυα αισθητήρων, όπου πρωταρχικός στόχος είναι η ελαχιστοποίηση της ενεργειακής κατανάλωσης λόγω των περιορισμένων πόρων του δικτύου.

Συνοπτικά, σύμφωνα με το προτεινόμενο πλαίσιο, ο διαχειριστής του συστήματος είναι σε θέση, στο *επίπεδο δεδομένων* και ανάλογα με την τοπολογία του υποκείμενου δικτύου, να επιλέξει τον καταλληλότερο αλγόριθμο δρομο-

λόγησης και συνάθροισης δεδομένων, ο οποίος θα ελαχιστοποιήσει την ενεργειακή κατανάλωση και θα κάνει βέλτιστη χρήση του διαθέσιμου εύρους ζώνης. Το *επίπεδο επεξεργασίας* επιτρέπει την επιλογή του κατάλληλου μορφοτύπου στο οποίο αναπαριστώνται οι πρωτογενείς αισθητήριες παρατηρήσεις, ενώ υποστηρίζει συναρτήσεις μετασχηματισμού μεταξύ διαφορετικών μορφοτύπων και προτύπων αναπαράστασης. Το *σημασιολογικό επίπεδο* της αρχιτεκτονικής συνδέει τα δεδομένα του επιπέδου επεξεργασίας με εννοιολογικά μοντέλα και οντολογίες, προσδίδοντας στο σύστημα επίγνωση περιβάλλοντος.

Παραλείποντας το σημασιολογικό επίπεδο, μπορούμε να θεωρήσουμε ότι το προτεινόμενο πλαίσιο περιγράφει ακόμα και συστήματα που δε χρησιμοποιούν μια τυπικά ορισμένη σημασία του περιβάλλοντός τους. Η παρουσία λοιπόν του ανώτερου επιπέδου της αρχιτεκτονικής είναι προαιρετική και εξαρτάται από το αν η προσθήκη σημασιολογίας βελτιώνει σημαντικά την αποτελεσματικότητα μιας εφαρμογής ή όχι. Εντούτοις, η παρουσία του σημασιολογικού επιπέδου συνδέεται κατά κανόνα με αρκετά από τα πλεονεκτήματα που αναφέρθηκαν στην αρχή αυτής της ενότητας, μειώνει το χρόνο ανάπτυξης εφαρμογών που αξιοποιούν δεδομένα από ένα δίκτυο αισθητήρων και επιτρέπει τη σημασιολογική επερώτηση αυτών των δεδομένων από τον τελικό χρήστη. Στις επόμενες παραγράφους, τα τρία επίπεδα της προτεινόμενης αρχιτεκτονικής περιγράφονται αναλυτικά.

Επίπεδο δεδομένων

Το επίπεδο δεδομένων αναλαμβάνει την ανακάλυψη, συλλογή και συνάθροιση πρωτογενών δεδομένων σε έναν κεντρικό κόμβο. Η αποδοτική συνάθροιση δεδομένων είναι κρίσιμος παράγοντας για την μείωση της συνολικής ενέργειας μετάδοσης και, κατ' επέκταση, τη μεγιστοποίηση του χρόνου ζωής του δικτύου. Ανάλογα με την τοπολογία του δικτύου αισθητήρων και το επιλεγμένο κριτήριο βελτιστοποίησης (π.χ. συνολική κατανάλωση ενέργειας, χρησιμοποιούμενο εύρος ζώνης, συνολική καθυστέρηση μετάδοσης δεδομένων), υπολογίζεται και εφαρμόζεται η αντίστοιχη στρατηγική για τη συνάθροιση των δεδομένων.

Η *συνάθροιση των δεδομένων* σε έναν κεντρικό κόμβο μπορεί να πραγματοποιηθεί μέσω δομημένων (structured) και αδόμητων (structure-free) τεχνικών [82]. Στις μεν πρώτες, ακολουθείται μια σταθερή διαδρομή για τη συγκέντρωση των δεδομένων, γεγονός που συνεπάγεται χαμηλό κόστος για τη συντήρηση της διαδρομής και καθιστά τις τεχνικές αυτές κατάλληλες για εφαρμογή σε στατικά περιβάλλοντα, όπου οι μεταβολές στο δίκτυο είναι σπάνιες. Αντίθετα, στις περιπτώσεις δυναμικού περιβάλλοντος, το κόστος για την εύρεση και συντήρηση της βέλτιστης διαδρομής μπορεί να είναι τόσο μεγάλο ώστε να υπερβαίνει τα οφέλη που προκύπτουν από τη συνάθροιση των δεδομένων. Επιπλέον, οι δομημένες προσεγγίσεις είναι ευαίσθητες στην καθυστέρηση που εισάγεται από τους ενδιάμεσους κόμβους,στη συχνότητα μετάδοσης δεδομένων καθώς και στο μέγεθος του δικτύου. Αδόμητες προσεγγίσεις, στις οποίες οι αποφάσεις για τη δρομολόγηση των δεδομένων λαμβάνονται δυναμικά, ταιριάζουν περισσότερο σε δίκτυα αισθητήρων όπου κόμβοι συνεχώς εισέρχονται και εξέρχονται του δικτύου, με αποτέλεσμα να μην ενδείκνυται μια σταθερή διαδρομή για τη συγκέντρωση των δεδομένων. Ένας κεντρικός

κόμβος είναι υπεύθυνος για την ανακάλυψη νέων κόμβων, ενώ επίσης καθορίζει και τη στρατηγική της ανάκτησης των δεδομένων, η οποία μπορεί να είναι push-based, οπότε οι αισθητήριες παρατηρήσεις αποστέλλονται από τον αντίστοιχο αισθητήρα στον κεντρικό κόμβο, ή pull-based, όταν ο κεντρικός κόμβος ανακτά περιοδικά μετρήσεις από τους αισθητήρες του δικτύου.

Ένα άλλο ζήτημα που πρέπει να ληφθεί υπόψη σε αυτό το χαμηλότερο επίπεδο της αρχιτεκτονικής είναι αυτό της *ασφάλειας των δεδομένων*, ιδιαίτερα σε περιπτώσεις όπου ένα δίκτυο παράγει, αποθηκεύει και διαχειρίζεται ευαίσθητα δεδομένα σε εχθρικά περιβάλλοντα. Οι σημαντικότεροι μέθοδοι πρόληψης ενάντια σε επιθέσεις από κακόβουλους χρήστες αναφέρονται στο [200]. Επίσης, όταν χρειάζεται να προστατευθεί πληροφορία σχετική με την τοποθεσία ενός παρατηρούμενου μεγέθους, μηχανισμοί ανωνυμίας πρέπει να υλοποιηθούν προκειμένου να διασφαλιστεί η εμπιστευτικότητα αυτής της πληροφορίας.

Ανάλογα με την περίπτωση του δικτύου, το είδος των αισθητήρων από το οποίο αυτό αποτελείται, αλλά και την εφαρμογή για την οποία χρησιμοποιείται, χρειάζεται να ληφθεί υπόψη η σχέση ανταλλαγής μεταξύ του επιθυμητού επιπέδου ασφαλείας και της αποδοτικότητας του συστήματος, προκειμένου να επιλεγεί η κατάλληλη στρατηγική και λύση για τα προαναφερθέντα ζητήματα.

Επίπεδο επεξεργασίας

Το επίπεδο επεξεργασίας αντιμετωπίζει την ετερογένεια των αισθητήριων συσκευών του δικτύου και των αναπαραστάσεων των παρατηρήσεων που προέρχονται από αυτές, υιοθετώντας ένα κοινό πρότυπο αναπαράστασης. Αυτό το πρότυπο αναπαράστασης πρέπει να είναι κατάλληλο για επεξεργασία από υπολογιστή, καθώς οι επεξεργασμένες παρατηρήσεις προωθούνται στο σημασιολογικό επίπεδο για περαιτέρω ανάλυση. Προτείνεται η χρήση της XML ως γλώσσας αναπαράστασης, καθώς αντιπροσωπεύει την ευκολότερη λύση για την έκφραση πληροφορίας με μια απλή δομή, ενώ η δημιουργία και επεξεργασία XML εγγράφων απλοποιείται σημαντικά με τη χρήση πλήθους διαθέσιμων σχετικών εργαλείων και προγραμματιστικών βιβλιοθηκών. Επιπλέον, υπάρχει ένας σημαντικός αριθμός προτύπων για την περιγραφή δικτύων αισθητήρων και αισθητήριων παρατηρήσεων, τα οποία βασίζονται σε XML, όπως για παράδειγμα τα SWE πρότυπα SensorML και O&M. Η υιοθέτηση των συγκεκριμένων προτύπων για την έκφραση των αισθητήριων παρατηρήσεων εξασφαλίζει τη συντακτική διαλειτουργικότητα αυτών και καθιστά δυνατή την επεξεργασία και το συνδυασμό τους.

Επίσης, το επίπεδο επεξεργασίας αναλαμβάνει τη συγκέντρωση και την ενοποίηση ολόκληρου του όγκου των παραγόμενων δεδομένων. Στις περισσότερες περιπτώσεις μάλιστα, είναι απαραίτητος ο υπολογισμός συνόψεων των πρωτογενών δεδομένων, καθώς η διατήρηση του συνόλου των παρατηρήσεων είναι συχνά άσκοπη και ανέφικτη. Ένα παράδειγμα μπορεί να αποτελεί η μέτρηση της θερμοκρασίας σε μια περιοχή με τη βοήθεια ενός δικτύου αισθητήρων που μπορεί να περιλαμβάνει δεκάδες ή εκατοντάδες αισθητήρες θερμοκρασίας. Η επεξεργασία και διατήρηση των μεμονωμένων μετρήσεων από κάθε αισθητήρα είναι ασύμφορη, καθώς όχι μόνο υπερφορτώνει το δίκτυο, αλλά απαιτεί και τεράστιο αποθηκευτικό χώρο. Μια σύνοψη των παρατηρήσεων

μέσω κάποιας συναθροιστικής συνάρτησης (όπως η μέση τιμή) είναι αρκετή για την περιγραφή των συνθηκών που επικρατούν στην περιοχή ενδιαφέροντος.

Οι διαδικασίες συντακτικής μετατροπής και επεξεργασίας των πρωτογενών παρατηρήσεων χρειάζονται αρκετή υπολογιστική ισχύ και, ως εκ τούτου, πραγματοποιούνται σε μια κεντρική υποδομή. Το επίπεδο επεξεργασίας ενός συστήματος πρέπει να υποστηρίζει περισσότερα του ενός πρότυπα αναπαράστασης και η επιλογή αυτών που θα χρησιμοποιηθούν γίνεται από το διαχειριστή του συστήματος, ανάλογα με το είδος των εγκατεστημένων αισθητήρων και τη φύση των λαμβανόμενων παρατηρήσεων.

Σημασιολογικό επίπεδο

Το σημασιολογικό επίπεδο συνδυάζει τις επεξεργασμένες αισθητήριες παρατηρήσεις με μεταδεδομένα που περιγράφουν το περιβάλλον της εφαρμογής, προσφέροντας στον τελικό χρήστη μια άποψη υψηλού επιπέδου για το δίκτυο αισθητήρων και τα φαινόμενα που αυτό παρατηρεί. Τα μεταδεδομένα αυτά εκφράζονται σε όρους συγκεκριμένων οντολογιών, οι οποίες περιγράφουν τον τομέα ενδιαφέροντος που σχετίζεται με την εφαρμογή και οι οποίες επιλέγονται από το διαχειριστή του συστήματος. Η παρουσία του σημασιολογικού επιπέδου στο προτεινόμενο πλαίσιο είναι προαιρετική, προκειμένου να καλύψει και περιπτώσεις υπαρχόντων συστημάτων διαχείρισης που δε λαμβάνουν υπόψη τους τη σημασία του περιβάλλοντός τους. Η απόφαση για την παρουσία του σημασιολογικού επιπέδου σε ένα σύστημα είναι σύνθετη και πρέπει να λαμβάνει υπόψη της τη σχέση ανταλλαγής που υπάρχει μεταξύ του επιπλέον υπολογιστικού φόρτου και της ποιότητας της προσφερόμενης υπηρεσίας ή εφαρμογής.

Τα βασικά συστατικά που μπορούν να χρησιμοποιηθούν στο σημασιολογικό επίπεδο είναι τα ακόλουθα:

Κανόνες. Οι κανόνες του σημασιολογικού επιπέδου αποτελούν τον οδηγό για τη μετατροπή των XML εγγράφων του επιπέδου επεξεργασίας σε RDF γράφους, ενώ επίσης καθορίζουν τη λογική και ευφυΐα της εφαρμογής. Με αυτό το σκεπτικό, διαχωρίζουμε τους κανόνες ανάλογα με τη χρήση τους: σε αυτούς που χρησιμοποιούνται για την αντιστοιχία των αισθητήριων παρατηρήσεων σε RDF και σε αυτούς που παράγουν νέες υπονοούμενες RDF προτάσεις δημιουργώντας με αυτόν τον τρόπο νέα γνώση. Εκτός από την παραγωγή νέας γνώσης, η τελευταία κατηγορία κανόνων μπορεί να οδηγήσει και στην πραγματοποίηση δράσεων ή τη σήμανση συναγερμών, όταν ικανοποιούνται οι αντίστοιχες προϋποθέσεις ενός κανόνα. Για τις ανάγκες του σημασιολογικού επιπέδου και προκειμένου να υπάρχει όσο το δυνατόν μεγαλύτερη ευελιξία, οι κανόνες ακολουθούν το μοντέλο *γεγονός-συνθήκη-δράση* (event-condition-action ή ECA) από το χώρο των ενεργών (active) βάσεων δεδομένων.Σύμφωνα με αυτό, κάθε κανόνας αποτελείται από τρία μέρη:

α) το *γεγονός* που ενεργοποιεί τον κανόνα,
β) τη *συνθήκη*, η οποία προσδιορίζει αν πρέπει να εκτελεσθεί η δράση του κανόνα και

γ) τη *δράση* που θα εκτελεσθεί αν ο κανόνας είναι ενεργοποιημένος και ισχύει η συνθήκη.

Οι ECA κανόνες ταιριάζουν σε αρχιτεκτονικές και συστήματα που βασίζονται σε γεγονότα, όπως κατ' εξοχήν αποτελούν τα δίκτυα αισθητήρων, και κωδικοποιούν την αναμενόμενη συμπεριφορά και ευφυΐα της εφαρμογής. Ένα παράδειγμα τέτοιου κανόνα θα μπορούσε να αναφέρει «Αν η θερμοκρασία σε ένα χώρο είναι πάνω από 30^oC, άνοιξε το σύστημα κλιματισμού». ECA κανόνες με κατάλληλες συνθήκες και δράσεις μπορούν να χρησιμοποιηθούν και για τη δημιουργία RDF προτάσεων από αισθητήριες παρατηρήσεις. Σε αυτή την τελευταία περίπτωση, ένα παράδειγμα συνθήκης θα μπορούσε να είναι «Αν το XML έγγραφο έχει ένα στοιχείο `/measurement/temperature`», ενώ ένα παράδειγμα δράσης θα μπορούσε να είναι «πρόσθεσε στον RDF γράφο την πρόταση `<χώρος1> <έχειθερμοκρασία>` ''28''». Εναλλακτικά, για την πραγματοποίηση της αντιστοιχίας, θα μπορούσαν να χρησιμοποιηθούν XSLT μετασχηματισμοί[7] για τη μετατροπή από XML σε RDF/XML, όμως μια τέτοια λύση καθιστά την αντιστοιχία λιγότερο ευανάγνωστη και κατ' επέκταση, δυσκολότερα παραμετροποιήσιμη.

Οντολογίες. Οι οντολογίες αποτελούν τα εννοιολογικά μοντέλα που περιγράφουν με τυπικό τρόπο τη γνώση που χρειάζεται η εφαρμογή για τη λειτουργία της. Οι οντολογίες μπορεί να είναι *θεματικές*, *χωρικές* και *χρονικές*, περιγράφοντας το θεματικό τομέα ενδιαφέροντος, χωρικές και χρονικές έννοιες αντίστοιχα. Η επαναχρησιμοποίηση γνωστών οντολογιών του Σημασιολογικού Ιστού ενισχύει τη σημασιολογική διαλειτουργικότητα των παρατηρήσεων και διευκολύνει σημαντικά το συνδυασμό της παραγόμενης γνώσης με γνώση από άλλες πηγές και τη συσχέτιση με γεγονότα που ανιχνεύονται από άλλα δίκτυα αισθητήρων. Η εκφραστικότητα της γλώσσας στην οποία είναι εκφρασμένες οι χρησιμοποιούμενες οντολογίες, καθώς και χαρακτηριστικά τους όπως η παρουσία σύνθετων αξιωμάτων ή ο συνολικός αριθμός των ορολογικών αξιωμάτων της οντολογίας επηρεάζουν τη διαδικασία συλλογισμού και συνεπώς, και την απόδοση του συστήματος.

Σύστημα αποθήκευσης. Οι επιλεχθείσες οντολογίες εφαρμογής και τα επισημειωμένα δεδομένα του δικτύου αισθητήρων αποθηκεύονται σε ένα triple store σύστημα, το οποίο, όπως είδαμε και στο κεφάλαιο 3, είναι υπεύθυνο για την αποδοτική αποθήκευση, διαχείριση και επερώτηση ενός RDF γράφου. Η επιλογή ενός triple store συστήματος βασίζεται κατά κύριο λόγο στην απόδοσή του, καθώς και στις δυνατότητες και τη γλώσσα της προγραμματιστικής διεπαφής την οποία αυτό παρέχει. Η συντριπτική πλειοψηφία των διαθέσιμων triple store συστημάτων παρέχει τη δυνατότητα επερώτησης των περιεχομένων τους μέσω της γλώσσας ερωτημάτων SPARQL, η οποία είναι η πλέον διαδεδομένη γλώσσα υποβολής ερωτημάτων σε RDF γράφους.

Μηχανή συλλογισμού. Όπως αναφέρθηκε και στην παράγραφο 2.2.2, η διαδικασία του συλλογισμού οδηγεί σε νέα συμπεράσματα που προκύπτουν από κατηγορηματικά δηλωθείσες προτάσεις και αξιώματα και αποτελεί κλειδί για

[7] Η XSLT (Extensible Stylesheet Language Transformations) είναι μια γλώσσα, βασισμένη στην XML, για τον ορισμό μετασχηματισμών από ένα XML έγγραφο σε ένα άλλο.

την επιτυχία μιας εφαρμογής βασισμένης σε σημασιολογία. Ως εκ τούτου, η μηχανή συλλογισμού θεωρείται βασικό συστατικό του σημασιολογικού επιπέδου, καθώς εκτός από τη δημιουργία νέας γνώσης, παρέχει και μια σειρά υπηρεσιών, όπως π.χ. έλεγχος της συνέπειας μιας βάσης γνώσης. Κάθε μηχανή συλλογισμού έχει διαφορετικά χαρακτηριστικά, εφαρμόζεται σε βάσεις γνώσης συγκεκριμένης οντολογικής γλώσσας και η απόδοσή της μπορεί να έχει βελτιστοποιηθεί για συγκεκριμένα σενάρια χρήσης, π.χ. για συλλογισμό σε βάσεις γνώσης με μικρό σώμα ορολογίας (TBox) και ογκώδες σώμα ισχυρισμών (ABox). Αυτά τα χαρακτηριστικά πρέπει να ληφθούν υπόψη κατά την επιλογή της μηχανής συλλογισμού, η οποία επηρεάζει και την απόδοση του συνολικού συστήματος. Συνεπώς, θα πρέπει να επιλεγεί εκείνη η μηχανή συλλογισμού, η οποία ταιριάζει περισσότερο στις συνθήκες λειτουργίας της εφαρμογής.

Τεχνικές που εστιάζουν κατά κύριο λόγο στο επίπεδο δεδομένων περιγράφονται στη διδακτορική διατριβή [3], ενώ το σύστημα ΠΡΙΑΜΟΣ (ενότητα 6.3) και η επέκταση που προτείνεται σε αυτό το κεφάλαιο (ενότητα 6.4) καλύπτουν το επίπεδο επεξεργασίας και το σημασιολογικό επίπεδο.

6.3 Σύντομη παρουσίαση του συστήματος ΠΡΙΑΜΟΣ

Το σύστημα ΠΡΙΑΜΟΣ (ΠΡοσαρμοστικά συστήματα πραγματικού χρόνου για εξόρυξη σημασΙολογίας και ευφυείς διεπΑφές - επίδειξη σε εφαρΜΟγές ασφάλειαΣ και επικοινωνίας του Πολίτη) [113] αποτελεί ένα σύστημα με επίγνωση περιβάλλοντος για τη σημασιολογική επισημείωση αισθητήριων παρατηρήσεων και την πραγματοποίηση συλλογισμού σε αυτές σε πραγματικό χρόνο. Στην τρέχουσα ενότητα, θα γίνει μια συνοπτική περιγραφή του συστήματος ΠΡΙΑΜΟΣ, στο οποίο αναπτύχθηκε κατάλληλη επέκταση (ενότητα 6.4) για τη βελτίωση της απόδοσής του μετά από κάποιο χρονικό διάστημα λειτουργίας, οπότε και η συσσώρευση μεγάλου αριθμού αισθητήριων παρατηρήσεων αυξάνει την απόκριση του συστήματος σε μη αποδεκτά επίπεδα.

Το ΠΡΙΑΜΟΣ ακολουθεί σε γενικές γραμμές την αρχιτεκτονική τριών επιπέδων που παρουσιάστηκε στην ενότητα 6.2, εστιάζοντας κυρίως σε δίκτυα πολυμεσικών αισθητήρων (όπως π.χ. κάμερες και μικρόφωνα), τα οποία αποτελούν το επίπεδο δεδομένων. Ένα τμήμα του επιπέδου επεξεργασίας, το οποίο αποτελείται από αλγόριθμους ανάλυσης εικόνας και εξαγωγής χαρακτηριστικών, θεωρείται εξωτερικό του συστήματος ΠΡΙΑΜΟΣ, το οποίο δέχεται ως είσοδο τα εξαχθέντα χαρακτηριστικά κωδικοποιημένα σε XML μηνύματα. Η υιοθέτηση της XML ως γλώσσας κωδικοποίησης των εισερχόμενων μηνυμάτων αποτελεί μια γενική προδιαγραφή που επιτρέπει τη χρήση του ΠΡΙΑΜΟΣ και σε περιπτώσεις άλλων κατηγοριών αισθητήρων ή και δεδομένων διαφορετικών θεματικών τομέων, όπως για παράδειγμα οικονομικά δεδομένα, δεδομένα υγείας ή δεδομένα χρήσης κοινωνικών δικτύων. Η μορφή των εισερχόμενων XML μηνυμάτων είναι παραμετροποιήσιμη, όπως εξάλλου και τα περισσότερα βασικά συστατικά του επιπέδου επεξεργασίας και του σημασιολογικού επιπέδου: οι κανόνες αντιστοιχίας σε RDF, οι κανόνες εξαγωγής συμπερασμάτων και οι οντολογίες εφαρμογής. Η επικοινωνία του δικτύου αισθητήρων με

το ΠΡΙΑΜΟΣ πραγματοποιείται μέσω υπηρεσιών Ιστού και η μετάδοση των μηνυμάτων ακολουθεί το πρωτόκολλο SOAP[8].

Οι κανόνες που δέχεται και είναι σε θέση να επεξεργαστεί το ΠΡΙΑΜΟΣ ακολουθούν το γενικό μοντέλο ECA και διατυπώνονται σε μια ιδιότυπη συντακτική μορφή, η οποία παρουσιάζεται αναλυτικά στο [113]. Οι κανόνες αυτοί ορίζονται από το διαχειριστή του συστήματος και καθορίζουν τόσο τις λεπτομέρειες της μετατροπής των αισθητήριων παρατηρήσεων σε RDF όσο και τη συμπεριφορά του συστήματος ανάλογα με το περιεχόμενο της βάσης γνώσης του. Οι μεν πρώτοι κανόνες ονομάζονται *κανόνες αντιστοιχίας*, ενώ οι δεύτεροι *σημασιολογικοί κανόνες*. Τα δύο είδη κανόνων ακολουθούν την ίδια σύνταξη, αλλά μπορούν να περιέχουν διαφορετικές συνθήκες και δράσεις. Μια σύνοψη των συνθηκών και δράσεων που υποστηρίζει η γλώσσα κανόνων του ΠΡΙΑΜΟΣ φαίνεται στον πίνακα 6.1.

Πίνακας 6.1: *Χαρακτηριστικά γλώσσας κανόνων του ΠΡΙΑΜΟΣ*

Κανόνες αντιστοιχίας		**Σημασιολογικοί κανόνες**	
Συνθήκες	*Δράσεις*	*Συνθήκες*	*Δράσεις*
ύπαρξη XML στοιχείου	προσθήκη ατόμου στη βάση γνώσης	ύπαρξη ατόμων συγκεκριμένης κλάσης	προσθήκη ατόμου στη βάση γνώσης
σύγκριση τιμής XML στοιχείου με σταθερά	προσθήκη ζεύγους ιδιότητας-τιμής	ύπαρξη αποτελεσμάτων SPARQL ερωτήματος	εκτέλεση SPARQL/Update ερωτήματος
ύπαρξη αδερφών XML στοιχείων	εκτέλεση SPARQL/Update ερωτήματος	ύπαρξη τιμής για συγκεκριμένη ιδιότητα	κλήση υπηρεσίας ιστού
		ύπαρξη υποκλάσεων για συγκεκριμένη κλάση	δημιουργία υποκλάσης
			εκτέλεση εντολής συστήματος

Ενδεικτικά, ένας κανόνας αντιστοιχίας που εμπλουτίζει τη βάση γνώση του συστήματος με νέα οντολογικά άτομα θα μπορούσε να είναι και ο ακόλουθος:

```
if,xml element exists,/Event/Person/@id,then,
insert individual in class,foaf:Person,named after,/Event/Person/@id
```

Ο κανόνας αυτός ελέγχει την ύπαρξη ενός συγκεκριμένου μονοπατιού σε ένα XML έγγραφο και, αν υπάρχει, προσθέτει στη βάση γνώσης ένα νέο οντολογικό άτομο της κλάσης `foaf:Person`, το μοναδικό αναγνωριστικό (URI) του οποίου κατασκευάζεται σύμφωνα με την τιμή του XML γνωρίσματος `/Event/Person/@id`. Παρατηρώντας τη δεύτερη στήλη του πίνακα 6.1, γίνεται φανερό ότι κάθε δράση ενός κανόνα αντιστοιχίας οδηγεί στην προσθήκη ενός συνόλου RDF προτάσεων G, με εξαίρεση την περίπτωση της εκτέλεσης ενός SPARQL DELETE ερωτήματος που μπορεί να προκαλέσει τη διαγραφή μίας ή περισσότερων RDF προτάσεων.

[8]Το SOAP (Simple Object Access Protocol) είναι ένα πρωτόκολλο, βασισμένο στη γλώσσα XML, για την ανταλλαγή δομημένης πληροφορίας μέσω υπηρεσιών Ιστού. Παραδείγματα τέτοιας δομημένης πληροφορίας αποτελούν τα ορίσματα εισόδου με τα οποία καλείται μια υπηρεσία, αλλά και πιθανές οδηγίες για την επεξεργασία ενός SOAP μηνύματος.

Αντίστοιχα, ένα παράδειγμα σημασιολογικού κανόνα θα μπορούσε να ήταν το επόμενο:

```
if,class has individuals,DangerousEvent,then,setAlert
```

Ο κανόνας αυτός ελέγχει την ύπαρξη οντολογικών ατόμων που ανήκουν στην κλάση `DangerousEvent` και αν υπάρχουν, εκτελεί την εντολή συστήματος `setAlert`. Η δράση ενός σημασιολογικού κανόνα, λοιπόν, εκτός από την μεταβολή της βάσης γνώσης του συστήματος, μπορεί να περιλαμβάνει την εκτέλεση μιας εντολής συστήματος ή την κλήση μιας υπηρεσίας ιστού. Η συνθήκη ενός σημασιολογικού κανόνα αναφέρεται στη βάση γνώσης που έχει προκύψει μετά από την εφαρμογή των κανόνων αντιστοιχίας και την πραγματοποίηση συλλογισμού για τον υπολογισμό των υπονοούμενων αξιωμάτων και ισχυρισμών. Ο συλλογισμός σε αυτό το σημείο είναι απαραίτητος προκειμένου να εξασφαλιστεί η ενεργοποίηση όλων των δυνατών σημασιολογικών κανόνων, ακόμα και αυτών που ενεργοποιούνται από υπονοούμενα γεγονότα. Η γλώσσα κανόνων του ΠΡΙΑΜΟΣ είναι ευανάγνωστη, ενώ η συγγραφή των κανόνων της εφαρμογής υποβοηθείται από τη γραφική διεπαφή του ΠΡΙΑΜΟΣ.

Δεδομένης μιας υπάρχουσας οντολογίας εφαρμογής (η οποία αποτελεί την αρχική βάση γνώσης του συστήματος), ενός συνόλου κανόνων αντιστοιχίας και ενός συνόλου σημασιολογικών κανόνων, όπως αυτά έχουν επιλεγεί από το διαχειριστή της εφαρμογής, η διαδικασία που ακολουθεί το ΠΡΙΑΜΟΣ για κάθε εισερχόμενο XML μήνυμα που περιέχει κάποιο χαρακτηριστικό χαμηλού επιπέδου σκιαγραφείται στον αλγόριθμο 18.

Αλγόριθμος 18 *Επεξεργασία εισερχόμενου μηνύματος ΠΡΙΑΜΟΣ*

Είσοδος: XML μήνυμα *message*, βάση γνώσης συστήματος KB, σύνολο κανόνων αντιστοιχίας *mappingRules*, σύνολο σημασιολογικών κανόνων *semanticRules*

Έξοδος: ανανεωμένη βάση γνώσης KB' ή/και αποτελέσματα εκτέλεσης εξωτερικών διαδικασιών

```
1: function ΕΠΕΞΕΡΓΑΣΙΑ_ΜΗΝΥΜΑΤΟΣ(message, KB, mappingRules, semanticRules)
2:     for all rule στο mappingRules do
3:         if συνθήκη του rule αποτιμάται σε true then
4:                                          # ο έλεγχος της συνθήκης αναφέρεται στο message
5:             εκτέλεσε τη δράση του rule        # η δράση μεταβάλλει το KB, KB' ← KB ∪ G
6:         end if
7:     end for
8:     KB' ← Εφαρμογή συλλογισμού σε KB'
9:     for all rule στο semanticRules do
10:        if συνθήκη του rule αποτιμάται σε true then
11:                                          # ο έλεγχος της συνθήκης αναφέρεται στο KB'
12:            εκτέλεσε τη δράση του rule
13:        end if
14:    end for
15: end function
```

Είναι φανερό ότι ο αλγόριθμος 18 μπορεί να οδηγήσει σε ταχεία αύξηση του συνολικού μεγέθους της βάσης γνώσης, λόγω της συνεχούς άφιξης XML μηνυμάτων που εμπλουτίζουν την πληροφορία του συστήματος. Αυτή η αύξηση έχει αντίκτυπο στην απόδοση του συστήματος, καθώς, τόσο ο έλεγχος εφαρμογής και η εκτέλεση των κανόνων όσο και η πραγματοποίηση συλλογισμού

επηρεάζονται αρνητικά από το μέγεθος της βάσης γνώσης. Αυτό επιβεβαιώνεται από τις μετρήσεις απόδοσης που παρουσιάζονται στο [113], αλλά και από το σχήμα 6.4. Αναπόφευκτα λοιπόν, με την πάροδο του χρόνου, ο χρόνος επεξεργασίας κάθε μηνύματος θα ξεπεράσει ένα ανώτατο αποδεκτό όριο, πέραν του οποίου η απόκριση του συστήματος κρίνεται ως μη ικανοποιητική. Το όριο αυτό εξαρτάται από το είδος της εφαρμογής και τις απαιτήσεις του χρήστη και μπορεί να κυμαίνεται από μερικά msec μέχρι αρκετά δευτερόλεπτα. Για παράδειγμα, το ανώτατο επιτρεπτό όριο χρονικής απόκρισης σε μια εφαρμογή παρακολούθησης της υγείας ενός ασθενούς θα πρέπει να είναι σαφώς μικρότερο από το αντίστοιχο όριο σε μια εφαρμογή εύρεσης χώρου στάθμευσης ή υπολογισμού της βέλτιστης διαδρομής σε ένα αστικό περιβάλλον, όπου ο τελικός χρήστης συνήθως έχει τη δυνατότητα να θυσιάσει κάποια δευτερόλεπτα για μια καλύτερη υπόδειξη.

Στο [113], προκειμένου να αντιμετωπιστεί αυτή η εγγενής αδυναμία του ΠΡΙΑΜΟΣ και να διατηρείται η απόκριση του συστήματος εντός επιθυμητών ορίων, προτείνεται μια περιοδική διαδικασία επαναφοράς της βάσης γνώσης σε κάποια αρχική κατάσταση, μεταφέροντας την υπάρχουσα πληροφορία σε μια δευτερεύουσα μόνιμη βάση γνώσης, η οποία μπορεί να χρησιμοποιηθεί για ιστορικά ερωτήματα και δε συμμετέχει στην εξαγωγή συμπερασμάτων σε πραγματικό χρόνο[9]. Μειονέκτημα αυτής της προσέγγισης αποτελεί το γεγονός ότι η διαδικασία επαναφοράς της βάσης γνώσης μπορεί να πραγματοποιηθεί σε απρόβλεπτες χρονικές στιγμές, ακόμα και σε συγκυρίες όπου υπάρχει σημαντική ροή εισερχόμενων αισθητήριων παρατηρήσεων. Σε τέτοιες συγκυρίες, ουσιαστικά χάνεται όλη η «μνήμη» του συστήματος, γεγονός που έχει επιπτώσεις στην αποτελεσματικότητα του συστήματος. Η συγκεκριμένη αδυναμία μετριάζεται μέσω της χρήσης παραθύρων, όπως προτείνεται στην ενότητα 6.4.

Το σύστημα ΠΡΙΑΜΟΣ είναι υλοποιημένο σε Java, χρησιμοποιεί το προγραμματιστικό πλαίσιο Jena[10] για τη διαχείριση και επερώτηση της βάσης γνώσης του συστήματος, η οποία διατηρείται σε μια MySQL βάση δεδομένων, ενώ χρησιμοποιεί επίσης τη μηχανή συλλογισμού Pellet [179].

6.4 Μια επέκταση βασισμένη σε παράθυρα

Σε αυτή την ενότητα, παρουσιάζουμε μια επέκταση στο σύστημα ΠΡΙΑΜΟΣ, η οποία είναι το αντικείμενο της δημοσίευσης [181] και η οποία προσαρμόζει την έννοια του παραθύρου δεδομένων στο πρόβλημα της διαχείρισης ροών RDF δεδομένων και της πραγματοποίησης συλλογισμού σε αυτά. Η προσέγγιση αυτή εκμεταλλεύεται τα γνωστά πλεονεκτήματα των παραθύρων δεδομένων, όπως αυτά αναφέρθηκαν στην παράγραφο 2.3.2, για να αντιμετωπίσει τις αδυναμίες του ΠΡΙΑΜΟΣ όταν το μέγεθος της βάσης γνώσης αυξάνεται υπερβολικά. Η θεώρηση ενός παραθύρου ενδιαφέροντος επί των συνολικών διαθέσιμων RDF δεδομένων εξοικονομεί πόρους του συστήματος, όπως μνήμη και επεξεργαστική ισχύ, προκειμένου αυτό να είναι σε θέση να αναγνωρίζει σε πραγματικό χρόνο και να αντιδρά στην εμφάνιση κάποιου συμβάντος. Η συγ-

[9] Σύμφωνα με το [33], ένα σύστημα ή μια διαδικασία πραγματικού χρόνου εξασφαλίζει απόκριση εντός ενός δεδομένου χρονικού διαστήματος.

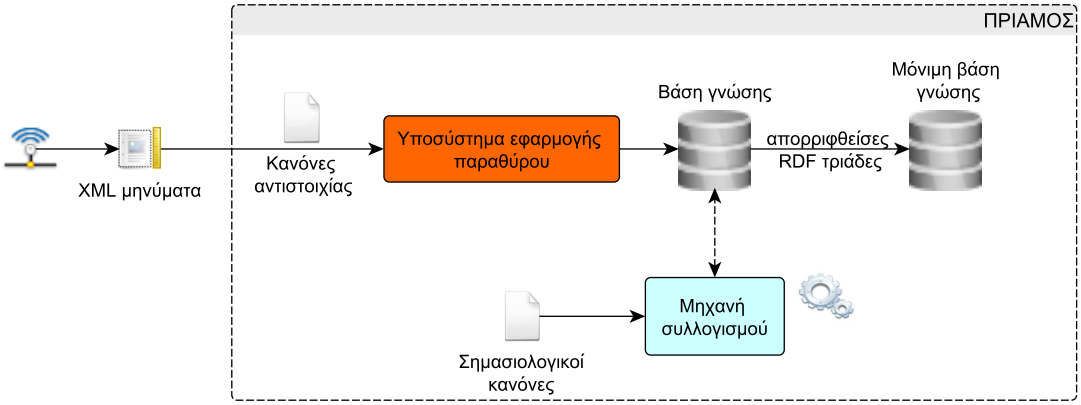
[10] http://jena.apache.org/

κεκριμένη λύση βασίζεται στο γεγονός ότι στην πλειοψηφία των εφαρμογών που διαχειρίζονται ροές δεδομένων, τα παλαιότερα δεδομένα έχουν μικρότερη αξία και σημασία από τα πιο πρόσφατα και κάποια στιγμή, καθίστανται άχρηστα. Επομένως, είναι πιθανό, υπό προϋποθέσεις, η θεώρηση της συνολικής βάσης γνώσης να μη συνεισφέρει ουσιαστικά χρήσιμα αποτελέσματα σε σχέση με τη θεώρηση ενός παραθύρου.

Η αρχιτεκτονική του συνολικού συστήματος απεικονίζεται στο σχήμα 6.2 και επεκτείνει το ΠΡΙΑΜΟΣ με ένα υποσύστημα εφαρμογής παραθύρου που χρησιμοποιεί SPARQL ερωτήματα για αυτό το σκοπό. Η προσθήκη του υποσυστήματος αυτού διαμορφώνει τον αλγόριθμο 18, προσθέτοντας μετά την εφαρμογή των σημασιολογικών κανόνων (γραμμές 9-14) τα στάδια:

1. του περιορισμού του μεγέθους της βάσης γνώσης, μέσω της θεώρησης ενός υποσυνόλου των συνολικών RDF προτάσεων, το οποίο υποδεικνύεται από μια συνάρτηση παραθύρου και

2. της μεταφοράς των απορριφθεισών RDF προτάσεων σε μια δευτερεύουσα μόνιμη βάση γνώσης, στην οποία μπορούν να τεθούν ιστορικά ερωτήματα.



Σχήμα 6.2: *Προσαρμοσμένη αρχιτεκτονική του ΠΡΙΑΜΟΣ και της προτεινόμενης επέκτασής του*

Με βάση τα παραπάνω ο χρόνος επεξεργασίας ενός XML μηνύματος από το σύστημα μπορεί να γραφτεί ως το άθροισμα των παρακάτω συνιστωσών:

$$t_{process} = t_{reason} + t_{map} + t_{semantic} + t_{win} + t_{move} \tag{6.1}$$

όπου t_{reason} ο χρόνος για την πραγματοποίηση συλλογισμού, t_{map} ο χρόνος εκτέλεσης των κανόνων αντιστοιχίας, $t_{semantic}$ ο χρόνος εκτέλεσης των σημασιολογικών κανόνων, t_{win} ο χρόνος υπολογισμού του παραθύρου εφόσον απαιτείται ανανέωση αυτού και t_{move} ο χρόνος μεταφοράς των απορριφθεισών τριάδων, εφόσον υπάρχουν τέτοιες, στη μόνιμη βάση γνώσης. Ο κύριος παράγοντας που επηρεάζει το σύνολο των επιμέρους αυτών συνιστωσών είναι το μέγεθος της βάσης γνώσης και κατά δεύτερο λόγο, το πλήθος των κανόνων αντιστοιχίας και των σημασιολογικών κανόνων, το οποίο επηρεάζει τα t_{map} και $t_{semantic}$ αντίστοιχα,

ενώ η πολυπλοκότητα των εισερχόμενων XML μηνυμάτων έχει αμελητέα επίδραση στο t_{map}.

Σε ό,τι αφορά στα είδη παραθύρων που εξετάζονται στο δεδομένο πλαίσιο, εκτός από το κλασικό ολισθαίνον παράθυρο, ορίζουμε και εφαρμόζουμε μια παραλλαγή του επάλληλου παραθύρου, ενώ εισάγουμε και μια νέα κατηγορία παραθύρων, τα παράθυρα *οντολογικών ατόμων*.

6.4.1 Παράθυρα οντολογικών ατόμων

Ο αρχικός σχεδιασμός του ΠΡΙΑΜΟΣ είναι τέτοιος ώστε να δέχεται ως είσοδο μια ροή XML μηνυμάτων, χωρίς να επιβάλλει κάποιο δομικό περιορισμό ή περιορισμό περιεχομένου σε αυτά. Αυτό σημαίνει ότι δεν είναι εξασφαλισμένη η παρουσία ενός κατηγορηματικού χρονικού οροσήμου (explicit timestamp), το οποίο αναφέρεται στη χρονική στιγμή κατά την οποία ισχύει μια παρατήρηση (βλέπε και παράγραφο 2.3.2). Ένα κατηγορηματικό χρονικό ορόσημο έχει μεγαλύτερη αξία για μια υπερκείμενη εφαρμογή σε σύγκριση με ένα υπονοούμενο χρονικό ορόσημο, το οποίο υποδηλώνει τη χρονική στιγμή άφιξης μιας παρατήρησης στο σύστημα και επηρεάζεται από πιθανές καθυστερήσεις του μέσου μετάδοσης, συνήθως ανεξάρτητες του παρατηρηθέντος φαινομένου. Για αυτό το λόγο, στο πλαίσιο που περιγράφεται στην τρέχουσα ενότητα, υποθέτουμε την παρουσία ενός χρονικού οροσήμου σε κάθε μεταδιδόμενο XML μήνυμα.

Όπως αναφέρθηκε και στην ενότητα 6.3, κάθε εισερχόμενο XML μήνυμα μπορεί να οδηγήσει στη μεταβολή της βάσης γνώσης του συστήματος, προσθέτοντας ή αφαιρώντας (στην περίπτωση εκτέλεσης ενός SPARQL/Update ερωτήματος) RDF προτάσεις. Καθώς ο χρόνος για την πραγματοποίηση συλλογισμού και για την απάντηση ερωτημάτων σε μια βάση γνώσης εξαρτάται θετικά από το μέγεθός της, ο στόχος της μείωσης του συνολικού χρόνου επεξεργασίας ενός XML μηνύματος μπορεί να επιτευχθεί με εφαρμογή μιας συνάρτησης παραθύρου στην *παραγόμενη* ροή RDF προτάσεων. Είναι προφανές ότι η εφαρμογή κάποιου παραθύρου απευθείας στην XML ροή δεν επιτρέπει τον ακριβή έλεγχο του μεγέθους του RDF παραθύρου, εφόσον κάθε XML μήνυμα συνεπάγεται διαφορετικό πλήθος προστιθέμενων ή αφαιρούμενων RDF προτάσεων.

Μια ροή RDF προτάσεων μπορεί να θεωρηθεί ειδική περίπτωση της γενικότερης έννοιας μιας ροής δεδομένων, όπως αυτή περιγράφηκε στην παράγραφο 2.3.2, δηλαδή ένα σύνολο ζευγών $(\langle s, p, o\rangle, \tau)$, όπου τ το χρονικό ορόσημο μιας RDF τριάδας. Η πλειοψηφία των προσεγγίσεων σημασιολογικής επεξεργασίας ροών δεδομένων που αναλύθηκαν στην ενότητα 6.1 θεωρεί ροές RDF προτάσεων και αντίστοιχοι τυπικοί ορισμοί υπάρχουν στα [22, 122].

Στην περίπτωση του ΠΡΙΑΜΟΣ, όπως ήδη αναφέρθηκε, ένα εισερχόμενο XML μήνυμα μπορεί να οδηγήσει στην παραγωγή μίας ή περισσότερων RDF τριάδων, οι οποίες θα μοιράζονται το ίδιο χρονικό ορόσημο, αυτό του XML μηνύματος. Σε αντιδιαστολή με τις μέχρι τώρα προταθείσες ροές RDF προτάσεων, εισάγουμε ένα νέο είδος ροής, το οποίο προκύπτει *ομαδοποιώντας τις RDF τριάδες που διαθέτουν ίδιο υποκείμενο και χρονικό ορόσημο*. Δεδομένου του γεγονότος ότι οι τριάδες που παράγονται από τους κανόνες αντιστοιχίας δεν περιέχουν κενούς κόμβους (blank nodes), αυτή η ομαδοποίηση αντιστοιχεί στη *συνοπτική περιορισμένη περιγραφή* (concise bounded description ή CBD)

του κοινού υποκειμένου αυτών των τριάδων. Το CBD [183] ενός RDF πόρου αποτελείται από τις RDF τριάδες που έχουν ως υποκείμενο το συγκεκριμένο πόρο καθώς και από εκείνες τις RDF τριάδες που έχουν ως υποκείμενο κάποιον κενό κόμβο που εμφανίζεται ως αντικείμενο σε κάποια τριάδα του CBD. Όπως φανερώνει και το όνομά του, το CBD είναι ένας γράφος που περιέχει προτάσεις που περιγράφουν ένα δεδομένο πόρο και χρησιμοποιείται συχνά σε εφαρμογές που ζητούν πληροφορία για αυτόν π.χ. μέσω ενός SPARQL DESCRIBE ερωτήματος.

Παράδειγμα 6.4.1. Έστω οι ακόλουθες RDF τριάδες που έχουν προστεθεί στη βάση γνώσης λόγω της ενεργοποίησης κατάλληλων κανόνων αντιστοιχίας σε δύο εισερχόμενα XML μηνύματα με χρονικά ορόσημα $\tau_1 = 4$ και $\tau_2 = 5$ αντίστοιχα. Οι δύο πρώτες αποτελούν το CBD του πόρου `http://example.org/person34`, ενώ οι δύο τελευταίες το CBD του πόρου `http://example.org/person35`.

```
@prefix : <http://example.org/>.
:person34 :hasXLocation ‘‘5’’.
:person34 :hasYLocation ‘‘-3’’.
:person35 :hasXLocation ‘‘12’’.
:person35 :hasYLocation ‘‘8’’.
```

Ορισμός 6.4.1. (Ροή οντολογικών ατόμων) Μια *ροή οντολογικών ατόμων* s_i ορίζεται ως ένα σύνολο ζευγών $(cbd(s_i), \tau_i)$, όπου $cbd(s_i)$ η συνοπτική περιορισμένη περιγραφή του s_i και τ_i το αντίστοιχο χρονικό ορόσημο αυτής.

Με άλλα λόγια, ο ορισμός 6.4.1 θεωρεί μια ροή από βασικούς υπογράφους που περιγράφουν μια οντότητα, προσέγγιση που μοιάζει αρκετά με τη ροή *RDF μορίων* (molecules) που παρουσιάζεται στο [72]. Η ελάχιστη μονάδα που θεωρείται στο [72] είναι το RDF μόριο, το οποίο αποτελεί την ελάχιστη μονάδα στην οποία μπορεί να διασπαστεί ένας RDF γράφος χωρίς απώλεια πληροφορίας [73] και το οποίο αποτελεί παρεμφερή έννοια με το CBD. Στο [72] αναφέρεται ότι συνήθως το χρονικό ορόσημο τ που ανατίθεται σε κάθε RDF μόριο είναι το μέγιστο μεταξύ των χρονικών οροσήμων των RDF τριάδων που το αποτελούν, θεώρηση που υιοθετείται και στο τρέχον πλαίσιο. Σύμφωνα με την υπόθεση αυτή, μια συνάρτηση παραθύρου θα δώσει προτεραιότητα στη διατήρηση ενός CBD που έχει ανανεωθεί πιο πρόσφατα σε σχέση με ένα άλλο, ακόμα και αν ο χρόνος αρχικής άφιξης του δεύτερου στο σύστημα είναι μεταγενέστερος από αυτόν του πρώτου. Η συμπεριφορά αυτή είναι επιθυμητή, εφόσον το πρώτο CBD θα περιέχει νεότερη πληροφορία σε σχέση με το δεύτερο.

Η αναπαράσταση της πληροφορίας για το χρονικό ορόσημο ενός CBD αποθηκεύεται στη βάση γνώσης με τη βοήθεια ενός ειδικού κατηγορήματος p_{time} και την αντίστοιχη τριάδα $\langle s_i, p_{time}, \tau_i \rangle$. Η τριάδα αυτή είναι αρκετή, καθώς το s_i προσδιορίζει μοναδικά το αντίστοιχο CBD του. Συνεπώς, για κάθε νέο οντολογικό άτομο που προστίθεται στη βάση γνώσης, προστίθεται και μια επιπλέον RDF πρόταση που δηλώνει το χρονικό ορόσημό του. Συνεχίζοντας το παράδειγμα 6.4.1, τα δύο εισερχόμενα XML μηνύματα θα προκαλέσουν την προσθήκη των προτάσεων $\langle$`:person34`$, p_{time}, \tau_1 \rangle$ και $\langle$`:person35`$, p_{time}, \tau_2 \rangle$ στη βάση γνώσης του συστήματος.

Η ανάθεση σε ένα CBD του χρονικού οροσήμου που αντιστοιχεί στην πιο πρόσφατη RDF πρόταση που ανήκει σε αυτό προϋποθέτει τη δυνατότητα ενημέρωσης της τιμής του χρονικού οροσήμου, όταν προστεθεί στο παράθυρο μια RDF πρόταση που βρίσκεται ήδη εντός παραθύρου. Για το σκοπό αυτό, δανειζόμαστε στοιχεία από το μοντέλο του *παραθύρου κατηγορήματος* (predicate window), το οποίο ακολουθεί την *προσέγγιση των αρνητικών πλειάδων* (negative tuple approach) [86]. Η συμμετοχή ενός στοιχείου ροής σε ένα παράθυρο κατηγορήματος δεν καθορίζεται από το χρονικό ορόσημο του στοιχείου, αλλά από μια συγκεκριμένη συνθήκη την οποία αυτό πρέπει να πληροί. Έτσι, μπορούν να οριστούν υποσύνολα μιας ροής που δε θα μπορούσαν να σχηματιστούν με ένα ολισθαίνον παράθυρο, όπως π.χ. οι ενδείξεις αισθητήρων με τιμή θερμοκρασίας μεγαλύτερη των 60^oC. Για κάθε παράθυρο κατηγορήματος, ορίζεται ένα *γνώρισμα συσχέτισης* (correlation attribute), το οποίο κατά κάποιον τρόπο αποτελεί το γνώρισμα-κλειδί για τη συσχέτιση δύο πλειάδων της ροής. Αν δύο πλειάδες με χρονικά ορόσημα τ_1 και τ_2 αντίστοιχα, $\tau_2 > \tau_1$, διαθέτουν την ίδια τιμή για το γνώρισμα συσχέτισης, τότε η δεύτερη ονομάζεται *πλειάδα ενημέρωσης* (update tuple) και αντικαθιστά την πρώτη. Επίσης, μια πλειάδα που προστίθεται στο παράθυρο ονομάζεται *θετική*, ενώ μια πλειάδα που απορρίπτεται από αυτό ονομάζεται *αρνητική*.

Παράδειγμα 6.4.2. Έστω ότι το σχήμα που ακολουθούν οι πλειάδες μιας ροής έχει τη μορφή ⟨`Αναγνωριστικό_Αισθητήρα, Θερμοκρασία, Χρονικό_Όρόσημο`⟩ και το γνώρισμα συσχέτισης είναι το `Αναγνωριστικό_Αισθητήρα`. Αυτό συνεπάγεται ότι η πλειάδα $\langle 3, 58, 14\rangle$ ενημερώνει και αντικαθιστά την πλειάδα $\langle 3, 62, 13\rangle$, εφόσον έχει την ίδια τιμή για το γνώρισμα συσχέτισης και μεταγενέστερο χρονικό ορόσημο.

Ορισμός 6.4.2. (Συνάρτηση συσχέτισης) Έστω $\mathbb{S}(t) = \{(\langle s_i, p_i, o_i\rangle, \tau_i), i = 1, 2, \dots\}$ μια ροή RDF προτάσεων και έστω η συνάρτηση $corr : (IB \times I \times IBL)^2 \rightarrow \{0, 1\}$, όπου I το σύνολο όλων των δυνατών IRIs, B το σύνολο όλων των κενών κόμβων RDF γράφων, L το σύνολο όλων των RDF λεκτικών, $IB = I \cup B$ και $IBL = I \cup B \cup L$. Η συνάρτηση $corr$ ονομάζεται *συνάρτηση συσχέτισης* και ισχύει ότι αν $corr(\langle s_1, p_1, o_1\rangle, \langle s_2, p_2, o_2\rangle) = 1$, τότε οι RDF προτάσεις $\langle s_1, p_1, o_1\rangle$ και $\langle s_2, p_2, o_2\rangle$ συσχετίζονται και αυτή με το μεταγενέστερο χρονικό ορόσημο ενημερώνει την άλλη.

Ο ορισμός 6.4.2 επεκτείνει την έννοια του γνωρίσματος συσχέτισης, το οποίο δεν προσφέρεται για την επιλεκτική ενημέρωση RDF προτάσεων με βάση το κατηγόρημά τους. Αυτό γίνεται φανερό αν παρομοιάσουμε μια RDF πρόταση με μια πλειάδα με τρία γνωρίσματα ⟨`Υποκείμενο, Κατηγόρημα, Αντικείμενο`⟩. Η επιλογή κανενός από τα παραπάνω γνωρίσματα ως γνώρισμα συσχέτισης δεν οδηγεί στο επιθυμητό αποτέλεσμα. Για παράδειγμα, η επιλογή του υποκειμένου ως γνώρισμα συσχέτισης δε θα επέτρεπε την παρουσία δύο RDF προτάσεων με κοινό υποκείμενο, αφού η μία θα ερμηνευόταν ως πρόταση ενημέρωσης (update statement) για την άλλη. Αντίθετα, η χρήση μιας συνάρτησης συσχέτισης λύνει αυτό το πρόβλημα, όπως φαίνεται και στο παράδειγμα 6.4.3.

Παράδειγμα 6.4.3. Η ενημέρωση του χρονικού οροσήμου ενός CBD επιτυγχάνεται με τη συνάρτηση συσχέτισης $corr(t_1, t_2) = t_1.s = t_2.s \wedge t_1.p = p_{time} \wedge t_2.p = p_{time}$,

όπου χρησιμοποιείται ο συμβολισμός $t_i.s$ και $t_i.p$ για τη δήλωση του υποκειμένου και του κατηγορήματος αντίστοιχα της τριάδας t_i. Αν λοιπόν, σύμφωνα με το παράδειγμα 6.4.1, το χρονικό ορόσημο του CBD του οντολογικού ατόμου :person34 είναι $\tau_1 = 4$ (ισχύει δηλαδή $\langle$:person34, p_{time},"4"$\rangle$) και ένα μεταγενέστερο XML μήνυμα προκαλέσει την προσθήκη της τριάδας $\langle$:person34, p_{time},"7"$\rangle$, τότε η τελευταία, σύμφωνα με την προηγούμενη συνάρτηση συσχέτισης, ενημερώνει και αντικαθιστά την πρώτη.

Η συνάρτηση συσχέτισης μπορεί να επεκταθεί προκειμένου να συμπεριλάβει και άλλες RDF ιδιότητες, ανάλογα με το αν η εκάστοτε εφαρμογή απαιτεί την ενημέρωση της τιμής τους ή τη διατήρηση της εξέλιξης όλων των τιμών. Για παράδειγμα, είναι πιθανό για μια εφαρμογή να έχει νόημα η διατήρηση όλων των τιμών που έχουν αναφερθεί για τις ιδιότητες :hasXLocation και :hasYLocation, αν χρειάζεται να εξαχθεί κάποιο συμπέρασμα σχετικά με την κατεύθυνση της κίνησης ή την ταχύτητα ενός αντικειμένου, ενώ για μια άλλη εφαρμογή μπορεί να έχει ενδιαφέρον μονάχα η τρέχουσα θέση ενός αντικειμένου. Στην τελευταία περίπτωση, χρειάζεται να ενημερώνεται διαρκώς η τιμή των δύο ιδιοτήτων με τις πλέον πρόσφατες τιμές, οπότε η συνάρτηση συσχέτισης θα έχει ως εξής: $corr(t_1,t_2) = t_1.s = t_2.s \wedge ((t_1.p = p_{time} \wedge t_2.p = p_{time}) \vee (t_1.p =$:hasXLocation $\wedge\, t_2.p =$:hasXLocation$) \vee (t_1.p =$:hasYLocation $\wedge\, t_2.p =$:hasYLocation$))$. Συνοψίζοντας λοιπόν, η συνάρτηση συσχέτισης είναι αυτή που καθορίζει τη συμπεριφορά του συστήματος όσον αφορά στη στρατηγική ενημέρωσης RDF τριάδων με συγκεκριμένα κατηγορήματα.

Η ελευθερία στην επιλογή των RDF ιδιοτήτων, των οποίων οι τιμές θα ενημερώνονται κατά την εξέλιξη της ροής δεδομένων συνεπάγεται και ένα από τα μειονεκτήματα του παραθύρου οντολογικών ατόμων, καθώς μπορεί να οδηγήσει στην παρουσία RDF τριάδων που μένουν διαρκώς εντός παραθύρου (long-living triples). Αυτό μπορεί να οδηγήσει σε αύξηση του συνολικού μήκους του παραθύρου και σε μειωμένη απόδοση του συστήματος, με αντάλλαγμα βέβαια τη διαθεσιμότητα περισσότερης γνώσης για ένα οντολογικό άτομο, η οποία συνήθως συνοδεύεται και από περισσότερα συμπεράσματα. Μια λύση στο πρόβλημα αυτό είναι ο ορισμός ενός μέγιστου χρονικού ορίου παραμονής μιας RDF τριάδας εντός παραθύρου, πέραν του οποίου αυτή απορρίπτεται.

Καθώς ο ορισμός 6.4.1 υπάγεται στο γενικότερο ορισμό μιας ροής δεδομένων, οι ορισμοί των διαφόρων ειδών παραθύρων που δόθηκαν στην παράγραφο 2.3.2 ισχύουν κανονικά και για μια ροή οντολογικών ατόμων. Το ίδιο ισχύει και για τους ορισμούς φυσικών παραθύρων, με τη μόνη διαφορά ότι πλέον οι παράμετροί τους, όπως το μήκος και το βήμα προόδου, εκφράζονται σε πλήθος οντολογικών ατόμων.

Αξίζει να τονιστεί ότι τα φυσικά παράθυρα οντολογικών ατόμων διαφέρουν σημαντικά από τα τυπικά φυσικά παράθυρα RDF προτάσεων που έχουν προταθεί μέχρι τώρα στη βιβλιογραφία. Εξ ορισμού, τα παράθυρα οντολογικών ατόμων εστιάζουν περισσότερο στην έννοια της οντότητας σε σύγκριση με τα φυσικά παράθυρα RDF προτάσεων, γεγονός που προσφέρει σημαντικά πλεονεκτήματα σε σχέση με τα τελευταία. Το μήκος ενός παραθύρου οντολογικών ατόμων μπορεί να μην έχει σταθερό αριθμό RDF προτάσεων, αλλά εξασφαλίζει ότι όλες οι προτάσεις που περιγράφουν ένα οντολογικό άτομο διατηρούνται εντός του παραθύρου, με την προϋπόθεση βέβαια ότι το συγκεκριμένο άτομο έχει διατηρηθεί εντός παραθύρου από τη στιγμή της πρώτης άφιξής του στο

σύστημα. Αντιθέτως, η εφαρμογή ενός φυσικού παραθύρου RDF προτάσεων παρουσιάζει το μειονέκτημα ότι λαμβάνει υπόψη της μονάχα το χρονικό ορόσημο μιας RDF πρότασης και όχι τη σχέση της με άλλες RDF προτάσεις, οδηγώντας συχνά σε καταστάσεις ελλιπούς γνώσης, οι οποίες έχουν αντίκτυπο και στην αποτελεσματικότητα της εφαρμογής. Το πρόβλημα αυτό αντιμετωπίζεται σε μεγάλο βαθμό από τα φυσικά παράθυρα οντολογικών ατόμων, υπό την προϋπόθεση ότι αυτά διαθέτουν το κατάλληλο μήκος για να καλύψουν τις ανάγκες της εφαρμογής σε ποσότητα γνώσης. Το επόμενο απλό παράδειγμα εξηγεί καλύτερα αυτή την υπεροχή των παραθύρων οντολογικών ατόμων.

Παράδειγμα 6.4.4. Έστω μια εφαρμογή επιτήρησης ενός εργαστηριακού χώρου η οποία καταγράφει την παρουσία ατόμων σε αυτόν. Τα άτομα αυτά μπορεί να είναι ακαδημαϊκό προσωπικό, φοιτητές ή και άτομα άλλης ιδιότητας. Μια ροή XML μηνυμάτων που περιέχουν παρατηρήσεις από κατάλληλους αισθητήρες (π.χ. κάμερες) προκαλεί την εισαγωγή μιας σειράς RDF προτάσεων στη βάση γνώσης του συστήματος. Ο επόμενος πίνακας συνοψίζει τον αριθμό των RDF τριάδων που έχουν ως υποκείμενο μια συγκεκριμένη οντότητα καθώς και το χρονικό ορόσημο αυτών.

Οντότητα	Πλήθος τριάδων	Χρονικό ορόσημο
:student3	4	1
:professor5	3	2
:student8	4	3
:student14	6	4
⋮	⋮	⋮
:student3	2	13
:student7	6	14
:student35	4	15

Η θεώρηση ενός παραθύρου RDF τριάδων για την τρέχουσα ροή καθιστά δύσκολη την ορθή επιλογή του μήκους του παραθύρου, εφόσον ο αριθμός των τριάδων που παράγονται για κάθε οντολογικό άτομο με την έλευση ενός XML μηνύματος είναι σε γενικές γραμμές απρόβλεπτος. Αν, για παράδειγμα, μεταξύ των χρονικών στιγμών $t_1 = 4$ και $t_2 = 13$ εντοπιστούν 20 διαφορετικοί φοιτητές, για καθέναν εκ των οποίων παραχθούν 4 τριάδες, γίνεται κατανοητό ότι ένα ολισθαίνον παράθυρο μήκους 100 τριάδων που εφαρμόζεται τη χρονική στιγμή $t_3 = 14$ θα περιέχει ελλιπή γνώση για την οντότητα :professor5 (συγκεκριμένα, θα περιέχει 2 από τις συνολικά 3 τριάδες). Επιπλέον, τη χρονική στιγμή $t_4 = 15$, όλη η γνώση για τον :professor5 θα βρεθεί εκτός παραθύρου. Αυτό θα προκαλέσει την ενεργοποίηση σημασιολογικών κανόνων που π.χ. ελέγχουν την ύπαρξη καθηγητή στο χώρο και σημαίνουν κάποιο συναγερμό στην περίπτωση απουσίας του. Αυτός ο λανθασμένος συναγερμός μπορεί να αποτραπεί με τη θεώρηση ενός παραθύρου οντολογικών ατόμων κατάλληλου μήκους, π.χ. έστω 50 αν είναι γνωστό ότι η χωρητικότητα του συγκεκριμένου εργαστηρίου κυμαίνεται σε αυτά τα επίπεδα. Το συγκεκριμένο παράθυρο θα διατηρήσει εντός της προσωρινής βάσης γνώσης όλες τις τριάδες που αναφέρονται στα 50 πιο πρόσφατα οντολογικά άτομα. Επιπλέον, αν υπάρχει κάποια εκτίμηση για τη διάρκεια των παρατηρούμενων φαινομένων, αυτή μπορεί να

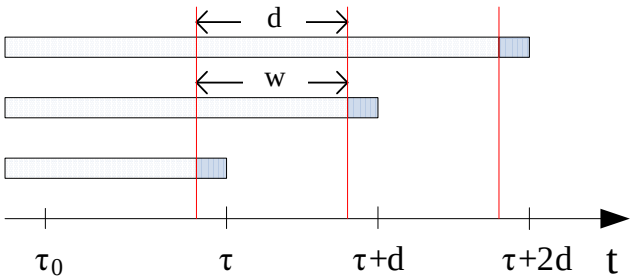
χρησιμεύσει ώστε να τεθεί ένα ρεαλιστικό κατώφλι μέγιστης παραμονής ενός οντολογικού ατόμου στο παράθυρο. Παραδείγματος χάριν, για την εκτέλεση μιας εργαστηριακής άσκησης 2 ωρών, ο χρόνος μέγιστης παραμονής ενός οντολογικού ατόμου στο παράθυρο μπορεί να τεθεί στις 2,5 ώρες.

Εκφράζοντας το μήκος του παραθύρου συναρτήσει εννοιών της εφαρμογής που είναι πιο κοντά στην αντίληψη του τελικού χρήστη, μπορεί να ερευνηθεί ευκολότερα η σχέση ανταλλαγής μεταξύ του χρόνου απόκρισης και της πληρότητας των συμπερασμάτων που εξάγονται από το σύστημα.

Το υλοποιημένο σύστημα επιτρέπει την εφαρμογή ολισθαίνοντων παραθύρων RDF προτάσεων και οντολογικών ατόμων, καθώς και μιας παραλλαγής του επάλληλου παραθύρου (βλέπε εξίσωση (2.8)), την οποία ονομάζουμε *σταδιακά συμπληρούμενο επάλληλο* παράθυρο. Η διαφορά του συγκεκριμένου παραθύρου με τον ορισμό του κλασικού επάλληλου παραθύρου έγκειται στη σταδιακή συμπλήρωση του πρώτου καθώς αφικνούνται νέα δεδομένα, σε αντίθεση με την ολική ανανέωση που ισχύει στο δεύτερο και η οποία λαμβάνει χώρα όταν υπάρχει διαθέσιμη ικανή ποσότητα δεδομένων για να συμπληρώσει ολόκληρο το παράθυρο. Ένα σταδιακά συμπληρούμενο επάλληλο παράθυρο δεν έχει σταθερό μήκος, ενώ η ανανέωσή του αρχίζει ανά τακτά χρονικά διαστήματα ίσα με το μήκος του παραθύρου, όπως ακριβώς και στην περίπτωση του κλασικού επάλληλου παραθύρου. Η φυσική εκδοχή του σταδιακά συμπληρούμενου επάλληλου παραθύρου – η οποία ισχύει ως έχει τόσο για ροές RDF προτάσεων όσο και για ροές οντολογικών ατόμων – δίνεται στην εξίσωση (6.2), ενώ αντίστοιχος είναι και ο ορισμός της χρονικής εκδοχής του.

$$w_{tum,gf,ph}(\mathbb{S}, W, \nu_0, \nu) = \begin{cases} w_{ph}(\mathbb{S}, \nu_0, \nu), & \nu_0 \leq \nu < \nu_0 + W - 1 \\ w_{ph}(\mathbb{S}, \nu - mod(\nu - \nu_0, W), \nu), & \nu_0 + W - 1 \leq \nu \wedge mod(\nu - \nu_0, W) \neq 0 \\ w_{ph}(\mathbb{S}, \nu - W + 1, \nu), & \nu_0 + W - 1 \leq \nu \wedge mod(\nu - \nu_0, W) = 0 \end{cases} \quad (6.2)$$

Διαγραμματικά, η εφαρμογή ενός χρονικού σταδιακά συμπληρούμενου επάλληλου παραθύρου στις χρονικές στιγμές τ, $\tau + d$ και $\tau + 2d$ απεικονίζεται στο σχήμα 6.3. Επιλέγουμε να απεικονίσουμε τη χρονική εκδοχή του παραθύρου για ευκολότερη αντιπαραβολή με το σχήμα 2.4 που απεικονίζει τις υπόλοιπες γνωστές κατηγορίες παραθύρων.



Σχήμα 6.3: *Σταδιακά συμπληρούμενο επάλληλο παράθυρο*

6.4.2 Αξιολόγηση επέκτασης

Σε αυτή την παράγραφο, μετράται η επίδοση του συστήματος για τα δύο είδη παραθύρων που αναφέρθηκαν στην παράγραφο 6.4.1 και επιβεβαιώνεται ότι η υλοποιημένη επέκταση επιτυγχάνει τη διατήρηση του χρόνου απόκρισης του συστήματος εντός συγκεκριμένων χρονικών ορίων, καθιστώντας το ένα σύστημα πραγματικού χρόνου. Επίσης, μετράται ο χρόνος της διαδικασίας εφαρμογής παραθύρου, ο οποίος σε κάθε περίπτωση αποτελεί μικρό ποσοστό του συνολικού χρόνου απόκρισης. Υπενθυμίζεται ότι αναλυτική αξιολόγηση της επίδοσης του συστήματος ΠΡΙΑΜΟΣ υπό διαφορετικές συνθήκες παρουσίας παρόχων δεδομένων και πλήθους κανόνων έχει πραγματοποιηθεί στη δημοσίευση [113] και συνεπώς, θεωρείται ότι ξεφεύγει των στόχων της τρέχουσας ενότητας.

Για τις ανάγκες των πειραματικών μετρήσεων που πραγματοποιήθηκαν, ακολουθήθηκε ένα σενάριο επιτήρησης ενός δωματίου με κάμερες, κατά το οποίο σημαίνουν κατάλληλοι συναγερμοί για διαφορετικά συμβάντα. Η OWL οντολογία εφαρμογής που χρησιμοποιήθηκε για το δεδομένο σενάριο είναι απλή, χωρίς σύνθετα OWL αξιώματα, περιέχει έννοιες όπως Δράστης, Συσκευή, Φοιτητής, Γεγονός, Τοποθεσία, ιδιότητες όπως έχειΤοποθεσία και τα συγκεντρωτικά στατιστικά στοιχεία της φαίνονται στον επόμενο πίνακα.

Αριθμός κλάσεων	31
Αριθμός ιδιοτήτων αντικειμένου	3
Αριθμός ιδιοτήτων τύπων δεδομένων	7
Αριθμός ατόμων	11
Αριθμός αξιωμάτων	62
Βάθος ιεραρχίας κλάσεων	4

Για τις ανάγκες της προσομοίωσης, θεωρούμε μια ροή 5000 XML μηνυμάτων που ακολουθούν το επόμενο πρότυπο με τυχαίες τιμές για τα γνωρίσματα που αναφέρονται σε αυτό.

```
<?xml version='1.0' encoding='UTF-8'?>
<Message>
    <Event id='{@eventId}'>
        <Tracker type='FaceTracker'>
            <TimeStamp value = '{@value}'/>
            <person id='{@personId}' certainty='{@certainty}' >
                <location2d x='{@x}' y='{@y}' />
            </person>
        </Tracker>
    </Event>
</Message>
```

Επίσης, χρησιμοποιούμε τον επόμενο κανόνα αντιστοιχίας, ο οποίος για καθένα από τα εισερχόμενα XML μηνύματα της παραπάνω μορφής δημιουργεί ένα νέο οντολογικό άτομο που ανήκει στην OWL κλάση Human και αναθέτει τιμές σε 4 OWL ιδιότητες τύπου δεδομένων, οδηγώντας συνολικά στην προσθήκη 5 RDF τριάδων στη βάση γνώσης του συστήματος:

```
if,xml element exists,/Message/Event/Tracker/person/@id,then,
insert individual in class,Human,named after,/Message/Event/Tracker/person/@id,
and set datatype property,hasXLocation,/Message/Event/Tracker/person/location2d/@x,
and set datatype property,hasYLocation,/Message/Event/Tracker/person/location2d/@y,
and set datatype property,hasCertainty,/Message/Event/Tracker/person/@certainty,
and set datatype property,hasTime,/Message/Event/Tracker/TimeStamp/@value
```

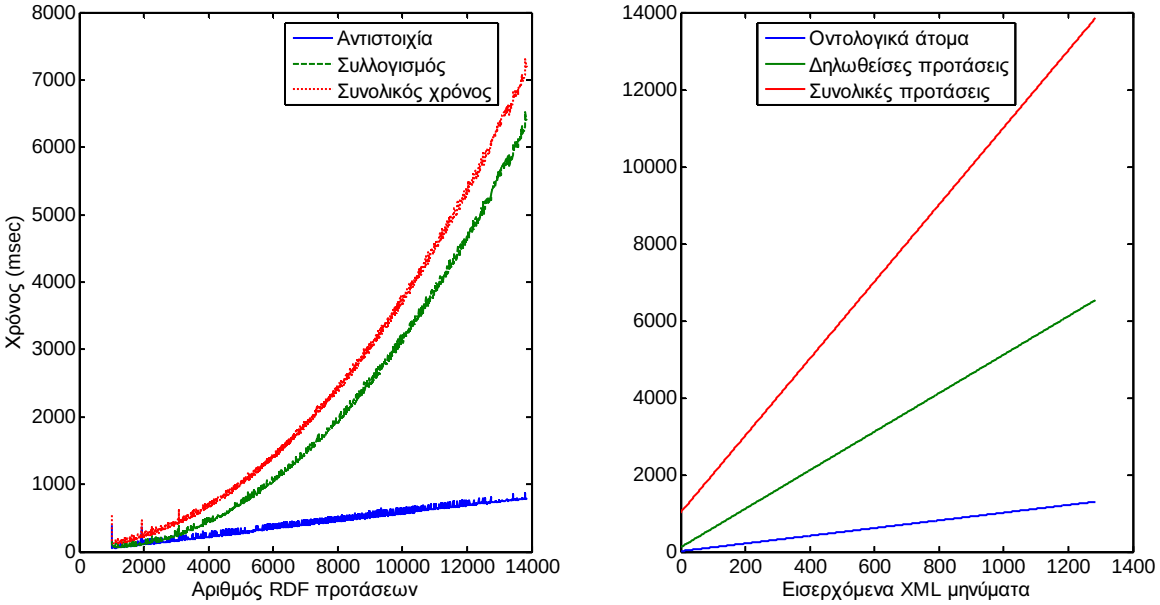
Κάθε πείραμα συνίσταται στην επεξεργασία από το σύστημα του συνόλου των XML μηνυμάτων και στη μέτρηση των επιμέρους συνιστωσών του χρόνου απόκρισης (εξίσωση (6.1)), για ολισθαίνοντα και σταδιακά συμπληρούμενα επάλληλα παράθυρα οντολογικών ατόμων με διαφορετικά μήκη. Στόχος είναι η πειραματική επιβεβαίωση της αναμενόμενης, με βάση τον ορισμό τους, συμπεριφοράς των παραπάνω παραθύρων καθώς και η αποδοτικότητα του υποσυστήματος εφαρμογής παραθύρου. Σημειώνουμε ότι για τις συγκεκριμένες πειραματικές εκτελέσεις, ισχύει $t_{semantic} = 0$ εφόσον δεν χρησιμοποιούμε κάποιο σημασιολογικό κανόνα, ενώ επίσης $t_{move} = 0$, καθώς διατηρούμε απενεργοποιημένη τη διαδικασία μεταφοράς των τριάδων που εξέρχονται από το παράθυρο στη δευτερεύουσα βάση γνώσης και οι οποίες πλέον διαγράφονται μόνιμα.

Οι μετρήσεις πραγματοποιήθηκαν σε σύστημα με επεξεργαστή Intel® Core™ 2 Duo @ 2.93GHz, με 2012 MB συνολική μνήμη σε περιβάλλον Windows XP.

Αρχικά, απενεργοποιούμε την εφαρμογή παραθύρου, προκειμένου να επαληθεύσουμε τη συμπεριφορά του ΠΡΙΑΜΟΣ και να παρατηρήσουμε ότι γρήγορα ο χρόνος απόκρισης αυξάνεται εκτός αποδεκτών ορίων. Στο αριστερό διάγραμμα του σχήματος 6.4, απεικονίζονται οι δύο κύριες συνιστώσες του χρόνου απόκρισης καθώς και ο συνολικός χρόνος συναρτήσει του συνολικού αριθμού – κατηγορηματικά δηλωθέντων και υπονοούμενων – προτάσεων της βάσης γνώσης. Στο δεξιό διάγραμμα του ίδιου σχήματος, απεικονίζεται το πλήθος των κατηγορηματικά δηλωθέντων και των συνολικών RDF προτάσεων, όπως και το πλήθος των οντολογικών ατόμων συναρτήσει του αριθμού των εισερχόμενων XML μηνυμάτων. Τα τρία αυτά μεγέθη είναι απόλυτα συσχετισμένα και η αύξησή τους είναι γραμμική, εφόσον για κάθε XML μήνυμα ενεργοποιείται ο μοναδικός κανόνας αντιστοιχίας, ο οποίος προσθέτει σταθερό αριθμό (5) προτάσεων στη βάση γνώσης. Επίσης, το σχήμα 6.4 αποκαλύπτει ότι η διαδικασία συλλογισμού είναι αυτή που κυριαρχεί και ευθύνεται για το μεγάλο χρόνο απόκρισης, ιδιαίτερα όσο αυξάνεται το μέγεθος της βάσης γνώσης.

Στη συνέχεια, εξετάζεται η εφαρμογή ενός σταδιακά συμπληρούμενου επάλληλου παραθύρου οντολογικών ατόμων και η συμπεριφορά του συστήματος για μήκη παραθύρου ίσα με 250, 500 και 1000 οντολογικά άτομα (ή ισοδύναμα, μήκη παραθύρου της τάξης των 1250, 2500 και 5000 προτάσεων[11]). Για κάθε πειραματική εκτέλεση, μετράμε τις επιμέρους συνιστώσες του συνολικού χρόνου επεξεργασίας κάθε XML μηνύματος, όπως και το πλήθος οντολογικών ατόμων και RDF προτάσεων που υπάρχουν ανά πάσα στιγμή στο παράθυρο. Τα δεξιά διαγράμματα του σχήματος 6.5 δείχνουν την εξέλιξη των οντολογικών ατόμων, η οποία υπαγορεύεται από την εξίσωση (6.2) και σύμφωνα με την οποία, ανά συγκεκριμένο αριθμό ατόμων ίσου με το μήκος του παραθύρου,

[11] Στην πραγματικότητα, τα παράθυρα θα περιέχουν έναν ελαφρώς μεγαλύτερο αριθμό RDF προτάσεων, συνυπολογίζοντας και τα αξιώματα της οντολογίας εφαρμογής.



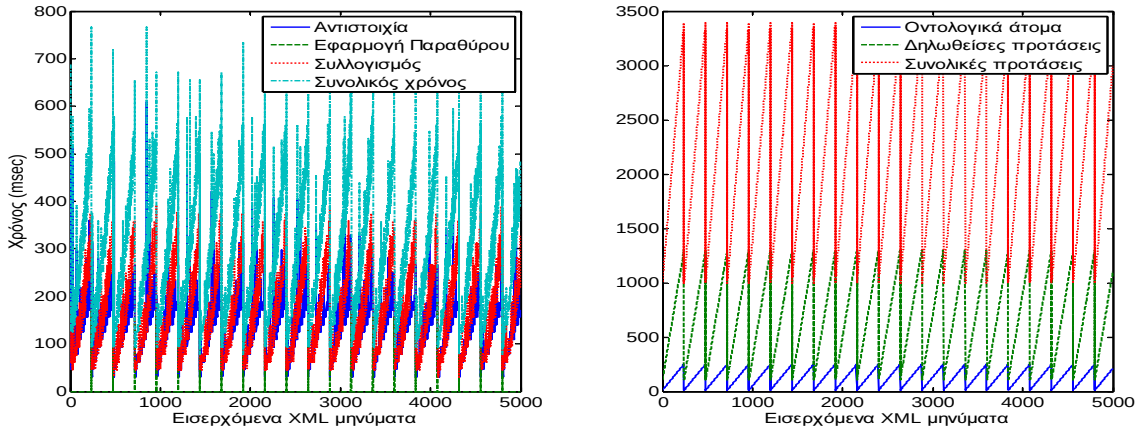
Σχήμα 6.4: *Χρόνος απόκρισης συστήματος χωρίς εφαρμογή παραθύρου, πλήθος οντολογικών ατόμων και RDF προτάσεων*

το παράθυρο ανανεώνεται πλήρως απορρίπτοντας όλο το περιεχόμενό του και αρχίζει να συμπληρώνεται σταδιακά με τις πιο πρόσφατες RDF προτάσεις, εξασφαλίζοντας την επικαιρότητα των δεδομένων του παραθύρου. Αντίθετα, σε ένα τυπικό επάλληλο παράθυρο, οι νέες RDF προτάσεις περιλαμβάνονται στο παράθυρο μόνο όταν συγκεντρωθούν τόσα οντολογικά άτομα όσα και το μήκος του παραθύρου.

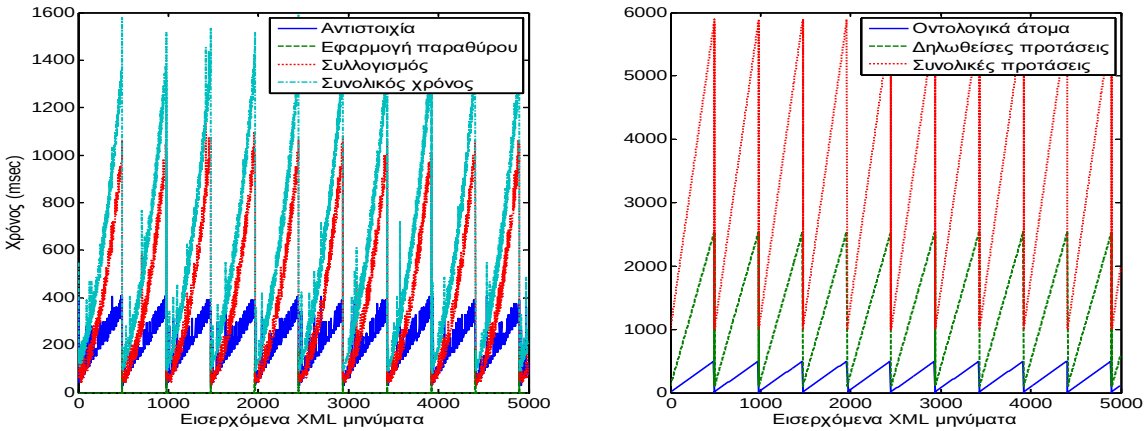
Ο αριθμός των RDF τριάδων που περιέχονται στο παράθυρο έχει, όπως αναμενόταν, άμεση επίδραση στο συνολικό χρόνο απόκρισης και αυτή η αντιστοιχία γίνεται φανερή από την αντιπαράθεση των διαγραμμάτων της αριστερής στήλης με αυτά της δεξιάς στο σχήμα 6.5. Το γεγονός ότι το άνω όριο του χρόνου απόκρισης που εμφανίζεται αμέσως πριν την ανανέωση κάθε στιγμιοτύπου παραθύρου παραμένει σταθερό για κάθε στιγμιότυπο παραθύρου οφείλεται στον αριθμό των RDF προτάσεων που περιέχει το παράθυρο και ο οποίος παραμένει σταθερός δεδομένων των υποθέσεων του τρέχοντος πειράματος. Όπως αναφέρθηκε και στην παράγραφο 6.4.1, τα παράθυρα οντολογικών ατόμων δεν έχουν σταθερό αριθμό RDF προτάσεων και κατ' επέκταση, δεν εξασφαλίζουν σταθερό άνω όριο χρόνου απόκρισης, σε αντίθεση με τα παράθυρα RDF προτάσεων, τα οποία διαθέτουν αυτό το χαρακτηριστικό.

Η αύξηση των προτάσεων της βάσης γνώσης επιδρά με διαφορετικό τρόπο στις διαδικασίες συλλογισμού και εκτέλεσης των κανόνων αντιστοιχίας, με το χρόνο συλλογισμού να παραμένει κυρίαρχος, καθορίζοντας σε μεγάλο ποσοστό το συνολικό χρόνο επεξεργασίας ενός μηνύματος από το ΠΡΙΑΜΟΣ. Αντίθετα, ο χρόνος εφαρμογής παραθύρου είναι αμελητέος και συνεισφέρει στο συνολικό χρόνο απόκρισης μόνο κατά την πλήρη ανανέωση του παραθύρου, οπότε και διαγράφονται όλα τα περιεχόμενά του.

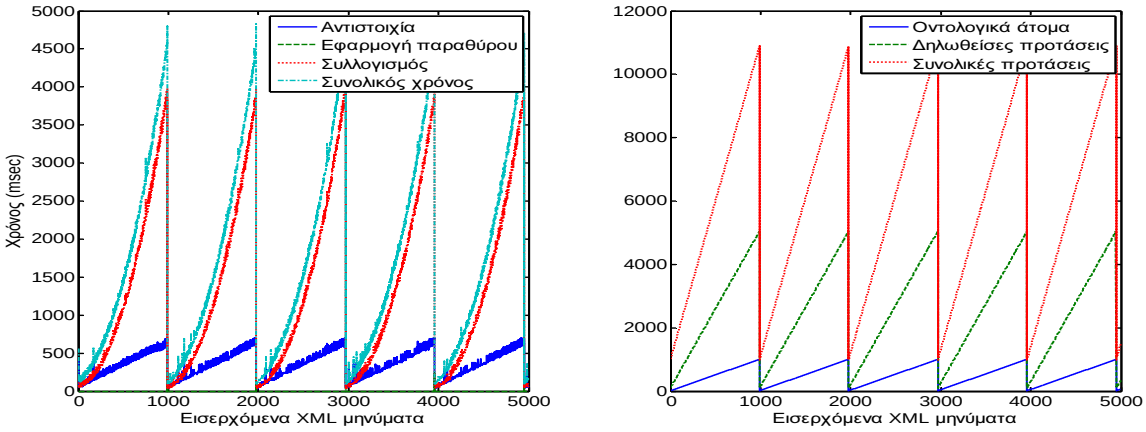
Τέλος, δοκιμάζουμε την εφαρμογή ενός ολισθαίνοντος παραθύρου οντολο-



(α) *Σταδιακά συμπληρούμενο επάλληλο παράθυρο μήκους 250 ατόμων*



(β) *Σταδιακά συμπληρούμενο επάλληλο παράθυρο μήκους 500 ατόμων*



(γ) *Σταδιακά συμπληρούμενο επάλληλο παράθυρο μήκους 1000 ατόμων*

Σχήμα 6.5: *Χρόνος απόκρισης συστήματος για σταδιακά συμπληρούμενο επάλληλο παράθυρο, πλήθος οντολογικών ατόμων και RDF προτάσεων*

γικών ατόμων μήκους 250, 500 και 1000 ατόμων και μοναδιαίο βήμα προόδου (θέτοντας δηλαδή $D = 1$ στην εξίσωση (2.7)). Τα μετρούμενα μεγέθη είναι ίδια με αυτά του σταδιακά συμπληρούμενου επάλληλου παραθύρου και παρουσιάζονται στο σχήμα 6.6. Σημειώνουμε ότι, για λόγους ευκρίνειας της απεικόνισης,

στα διαγράμματα της αριστερής στήλης του σχήματος 6.6 σχεδιάζεται ο κινητός μέσος όρος με ορίζοντα 100 στοιχείων για όλες τις συνιστώσες του χρόνου απόκρισης, οπότε και έχουν απαλειφθεί οι διακυμάνσεις που παρουσιάζονται και οι οποίες οφείλονται στο περιβάλλον δοκιμής στο οποίο πραγματοποιήθηκαν οι μετρήσεις. Ενδεικτικά, ο ακριβής χρόνος απόκρισης απεικονίζεται στο σχήμα 6.7 για το παράθυρο μήκους 250 ατόμων.

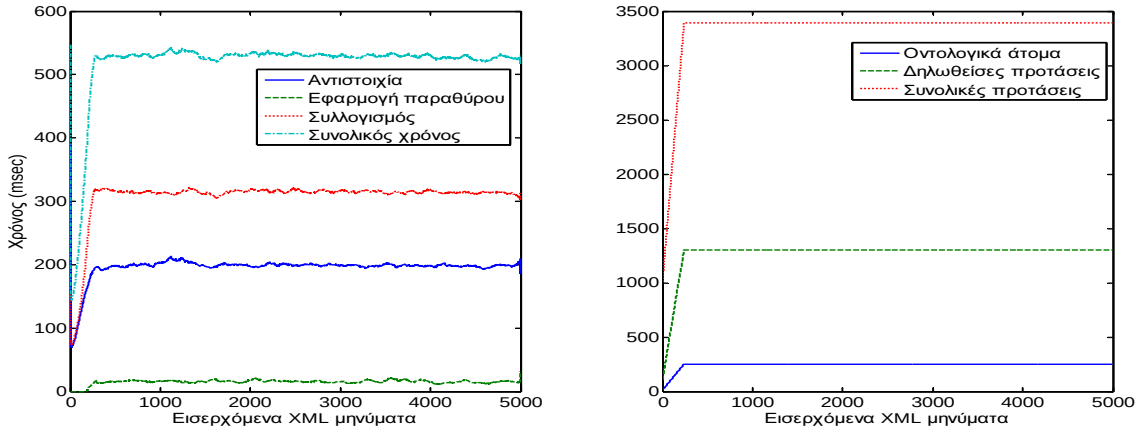
Ο υπολογισμός του νέου στιγμιοτύπου του παραθύρου πραγματοποιείται με το ακόλουθο SPARQL ερώτημα στη βάση γνώσης, προκειμένου να επιλεγούν τα D αρχαιότερα οντολογικά άτομα – όπου D το βήμα προόδου του παραθύρου – και να αφαιρεθούν από το παράθυρο τα CBD αυτών των ατόμων:

```
SELECT ?x
WHERE{?x :hasTime ?time.}
ORDER BY xsd:integer(?time) LIMIT {@D}
```

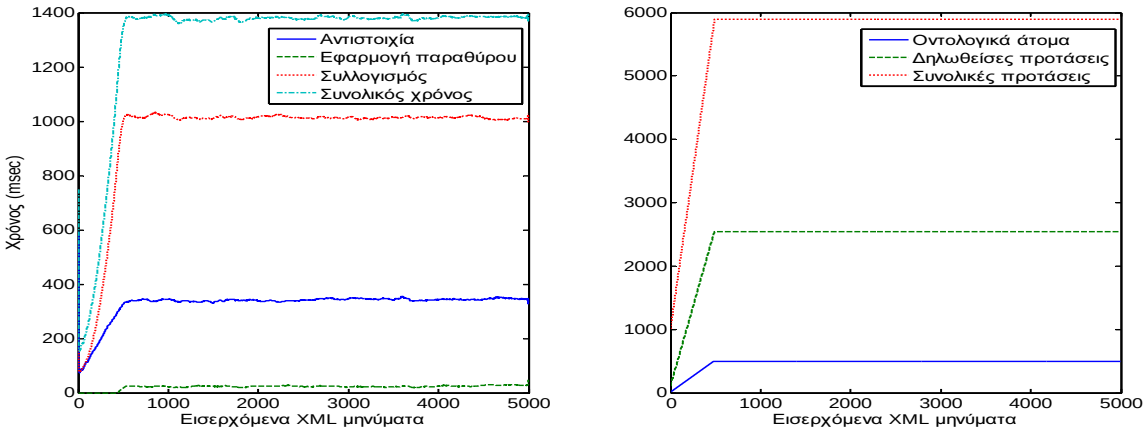
όπου hasTime η ιδιότητα που παίζει το ρόλο του κατηγορήματος p_{time} στην παράγραφο 6.4.1. Αξίζει να παρατηρήσουμε ότι ο χρόνος για τον υπολογισμό κάθε στιγμιοτύπου παραθύρου αντιπροσωπεύει ένα μικρό ποσοστό (κοντά στο 3-5%) του συνολικού χρόνου απόκρισης. Το σχήμα 6.6 δείχνει ότι η εφαρμογή ενός ολισθαίνοντος παραθύρου RDF προτάσεων συνεπάγεται μια σταθερή χρονική απόκριση, η οποία μπορεί μεν να είναι μεγαλύτερη από τη μέση χρονική απόκριση που επιτυγχάνεται με ένα σταδιακά συμπληρούμενο επάλληλο παράθυρο, αλλά αποφεύγει την αδυναμία του τελευταίου να αναγνωρίσει συμβάντα στα σημεία «επανεκκίνησης» κάθε στιγμιοτύπου παραθύρου. Αυτό γίνεται φανερό και από το σχήμα 6.8, όπου απεικονίζονται στο ίδιο διάγραμμα οι κινητοί μέσοι όροι των συνολικών χρόνων απόκρισης για τα δύο παράθυρα.

Η σχέση ανταλλαγής μεταξύ χρόνου απόκρισης και πληρότητας των εξαγόμενων συμπερασμάτων καθώς και οι απαιτήσεις και περιορισμοί της εκάστοτε εφαρμογής αποτελούν τους κυριότερους παράγοντες που πρέπει να λαμβάνονται υπόψη κατά την επιλογή του μήκους και του είδους του χρησιμοποιούμενου παραθύρου. Αν η επίτευξη αυστηρών χρονικών ορίων απόκρισης αποτελεί ύψιστη προτεραιότητα για την εφαρμογή, τότε προτείνεται η χρήση ενός ολισθαίνοντος παραθύρου RDF προτάσεων, λόγω της σταθερής χρονικής απόκρισης που αυτό επιτυγχάνει. Η επιλογή του μήκους μπορεί να πραγματοποιηθεί μετά από μια δοκιμαστική περίοδο λειτουργίας του συστήματος με πραγματικά ή ιστορικά δεδομένα χωρίς τη χρήση παραθύρου και την καταγραφή της επίδοσής του σε ένα διάγραμμα παρόμοιο με το αριστερό διάγραμμα του σχήματος 6.4. Το διάγραμμα αυτό μπορεί να λειτουργήσει ως εκτιμητής του μέγιστου πλήθους των RDF προτάσεων που αντιστοιχούν στο άνω όριο απόκρισης που έχει τεθεί ως απαίτηση από την εφαρμογή. Παραδείγματος χάριν, με βάση το σχήμα 6.4, μια εφαρμογή που απαιτεί απόκριση κάτω του 1 sec, μπορεί να επιλέξει ένα παράθυρο μήκους περίπου 5000 RDF προτάσεων.

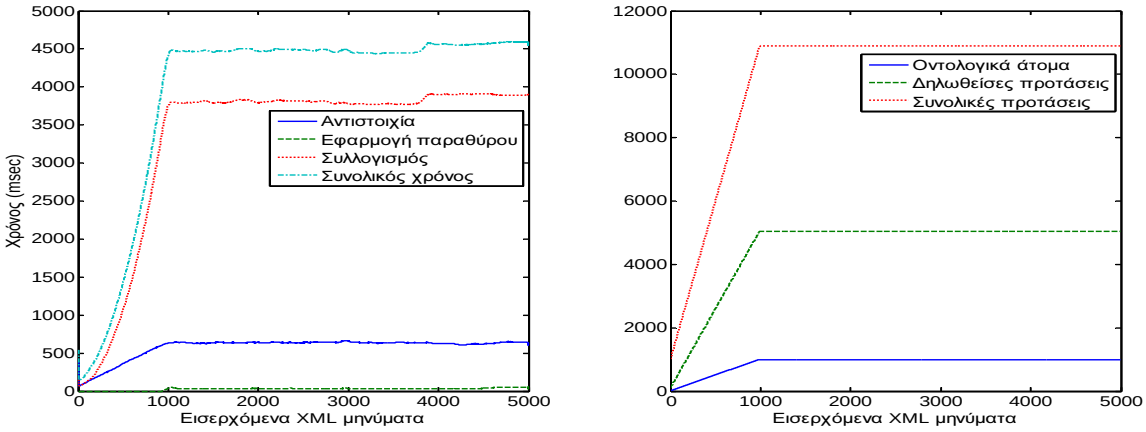
Στην περίπτωση που προτεραιότητα της εφαρμογής είναι η ποιότητα των εξαγόμενων συμπερασμάτων, τότε προτείνεται η χρήση ενός παραθύρου οντολογικών ατόμων. Το μέγεθος του παραθύρου μπορεί να προσδιοριστεί άμεσα αν υπάρχει κάποια εκτίμηση για το περιβάλλον λειτουργίας της εφαρμογής και το πλήθος των οντοτήτων που η τελευταία χρειάζεται να παρακολουθεί, ενώ αν δεν είναι δυνατόν να γίνει τέτοια εκτίμηση, προτείνεται η δοκιμαστική λειτουργία του συστήματος για διάφορα μήκη παραθύρων με την παρουσία



(α) *Ολισθαίνον παράθυρο μήκους 250 ατόμων*



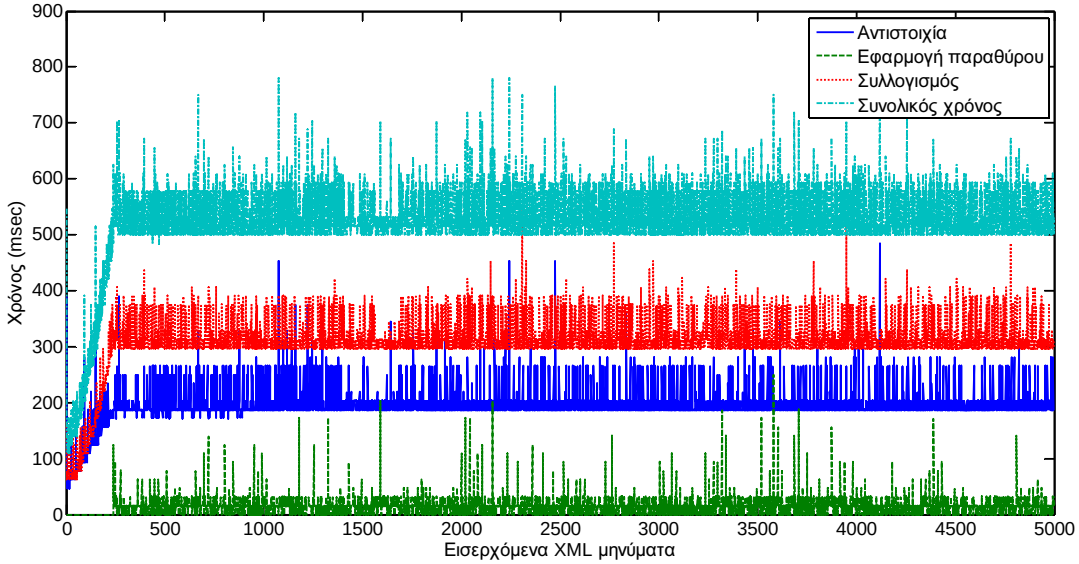
(β) *Ολισθαίνον παράθυρο μήκους 500 ατόμων*



(γ) *Ολισθαίνον παράθυρο μήκους 1000 ατόμων*

Σχήμα 6.6: *Κινητός μέσος όρος χρόνου απόκρισης συστήματος για ολισθαίνον παράθυρο με μοναδιαίο βήμα προόδου, πλήθος οντολογικών ατόμων και RDF προτάσεων*

των απαιτούμενων κανόνων αντιστοιχίας και σημασιολογικών κανόνων, προκειμένου να επιλεγεί το μήκος εκείνο που επιτυγχάνει την ορθή αναγνώριση των συμβάντων ενδιαφέροντος και ενεργοποιεί ορθούς συναγερμούς. Τέλος, αν

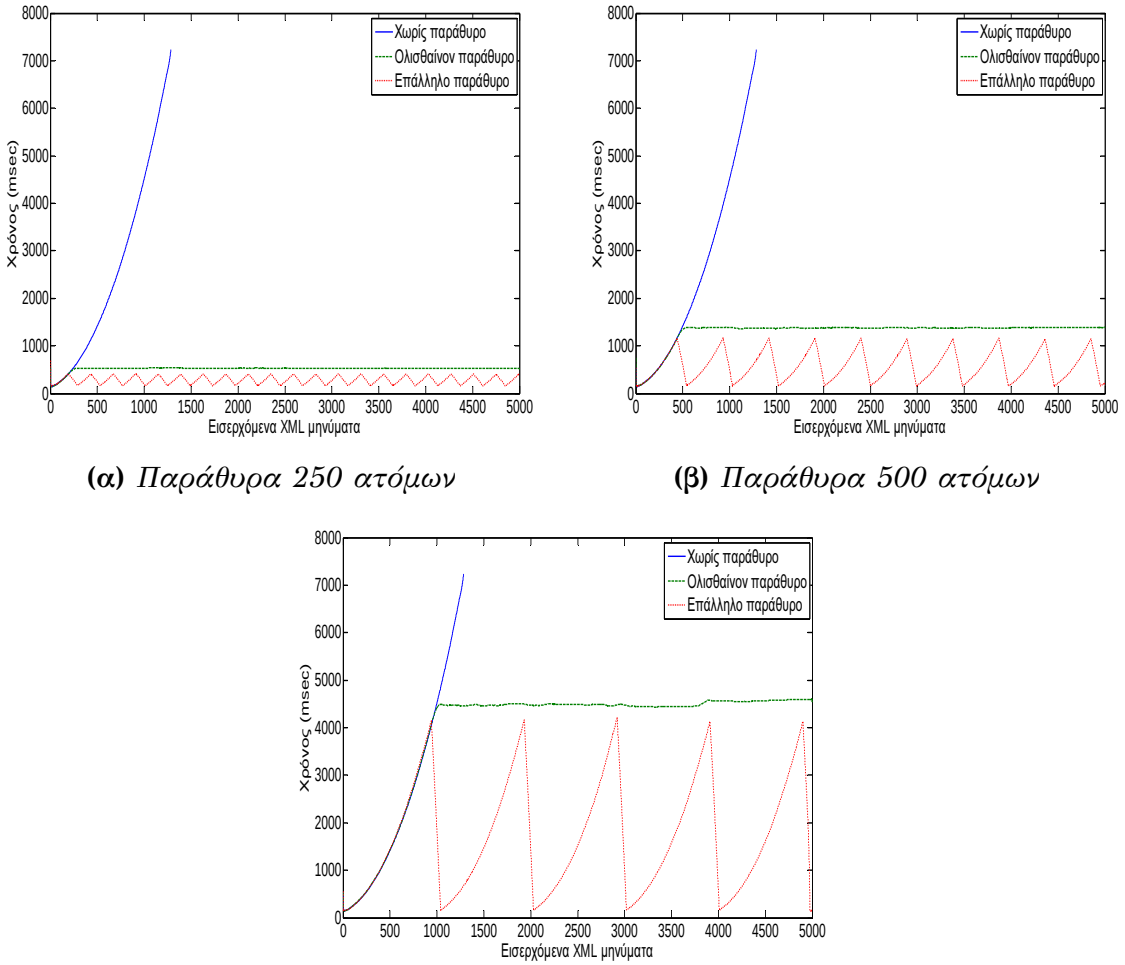


Σχήμα 6.7: *Χρόνος απόκρισης συστήματος για ολισθαίνον παράθυρο μήκους 250 ατόμων*

οι απαιτήσεις της εφαρμογής δεν κλίνουν αποκλειστικά προς μία κατεύθυνση (είτε προς την τήρηση αυστηρής χρονικής απόκρισης είτε προς την επίτευξη ποιοτικών αποτελεσμάτων), μπορούν να συνδυαστούν οι δύο παραπάνω τεχνικές για την επιλογή του βέλτιστου είδους και μήκους παραθύρου.

6.5 Συμπεράσματα και μελλοντική εργασία

Σε αυτό το κεφάλαιο, μελετήθηκε το πρόβλημα της σημασιολογικής επεξεργασίας ημιδομημένων δεδομένων, με έμφαση σε δεδομένα που προέρχονται από δίκτυα αισθητήρων. Παρουσιάστηκε μια αρθρωτή αρχιτεκτονική βασισμένη σε τεχνολογίες Σημασιολογικού Ιστού για τη συγκέντρωση, διαχείριση, σημασιολογικό εμπλουτισμό και εφαρμογή συλλογισμού σε μετρήσεις και αισθητήριες παρατηρήσεις. Ιδιαίτερο βάρος δόθηκε στη χρονική διάσταση των δεδομένων δικτύων αισθητήρων και στην αποτελεσματική διαχείριση των ροών που αυτά δημιουργούν. Συγκεκριμένα, η γνωστή από το χώρο των βάσεων δεδομένων τεχνική της εφαρμογής παραθύρων επί μιας ροής δεδομένων προσαρμόστηκε για τις ανάγκες RDF δεδομένων και οδήγησε στην επέκταση ενός ήδη υλοποιημένου συστήματος σημασιολογικής επισημείωσης και εφαρμογής συλλογισμού σε ημιδομημένα δεδομένα. Η επέκταση αυτή έδειξε ότι η εφαρμογή ενός κατάλληλου παραθύρου στα εισερχόμενα δεδομένα μπορεί να οδηγήσει σε σταθερό χρόνο απόκρισης του συστήματος, καθιστώντας το σύστημα πραγματικού χρόνου. Επίσης, προτάθηκε μια νέα κατηγορία παραθύρου, τα παράθυρα οντολογικών ατόμων, τα οποία μπορούν να ξεπεράσουν την αδυναμία της ελλιπούς γνώσης για μια συγκεκριμένη οντότητα, αδυναμία η οποία χαρακτηρίζει τα παράθυρα RDF τριάδων. Επιπλέον, τα παράθυρα οντολογικών ατόμων καθιστούν ευκολότερη τη διερεύνηση της σχέσης ανταλλαγής που ισχύει μεταξύ του χρόνου επεξεργασίας των δεδομένων και της



(α) *Παράθυρα 250 ατόμων* **(β)** *Παράθυρα 500 ατόμων*

(γ) *Παράθυρα 1000 ατόμων*

Σχήμα 6.8: *Σύγκριση κινητών μέσων όρων χρόνων απόκρισης συστήματος με και χωρίς εφαρμογή παραθύρου*

ποιότητας της εξαγόμενης γνώσης, καθώς οι παράμετροί τους εκφράζονται σε όρους οντοτήτων της εφαρμογής, το πλήθος των οποίων μπορεί να εκτιμηθεί πιο εύκολα σε σύγκριση με το απαιτούμενο πλήθος των RDF τριάδων.

Τα παράθυρα οντολογικών ατόμων είναι ίσως η πιο απλή μορφή μιας νέας κατηγορίας παραθύρων που οραματιζόμαστε και η οποία μπορεί να χαρακτηριστεί ως μια εξέλιξη των παραθύρων κατηγορήματος, στα οποία η συμμετοχή ενός στοιχείου στο στιγμιότυπο ενός παραθύρου εξαρτάται από την τιμή ενός γνωρίσματός του. Αντίστοιχα, ένα νέο είδος σημασιολογικού παραθύρου θα μπορούσε να χαρακτηρίζεται από μια περισσότερο σύνθετη συνθήκη, εκφρασμένη με τη μορφή ενός σημασιολογικού ερωτήματος. Η συνθήκη αυτή μπορεί να λαμβάνει υπόψη της όχι μόνο ιδιότητες του ίδιου του στοιχείου αλλά και τις σχέσεις του με άλλα στοιχεία, επιτρέποντας το σχηματισμό πιο επιλεκτικών παραθύρων, δυνατότητα που μοιάζει να έχει σημαντικές προοπτικές αξιοποίησης από εφαρμογές. Η συνεχής πρόοδος σε συστήματα διαχείρισης και επερώτησης RDF συνόλων δεδομένων εγγυάται ότι ο χρόνος σχηματισμού του παραθύρου θα επιβαρύνει ελάχιστα τη συνολική απόδοση του συστήματος και ένδειξη για αυτό μοιάζει να αποτελεί το χαμηλό ποσοστό του χρόνου σχηματι-

σμού του παραθύρου επί του συνολικού χρόνου απόκρισης που παρατηρήθηκε στις μετρήσεις του υπό εξέταση συστήματος.

Ασφαλώς, η λύση που παρουσιάστηκε στο παρόν κεφάλαιο απέχει αρκετά από το να χαρακτηριστεί βέλτιστη, δεδομένων των εγγενών αδυναμιών του συστήματος στο οποίο υλοποιήθηκε η παραθυρική αυτή επέκταση. Η βασικότερη εξ αυτών, η οποία έχει φανερό αντίκτυπο στη συνολική απόδοση του συστήματος είναι η αντιμετώπιση του συλλογισμού ως μιας αδιαφανούς διαδικασίας που επιτελείται από ένα εξωτερικό σύστημα συλλογισμού, το οποίο θεωρεί κάθε στιγμιότυπο παραθύρου ως μια στατική βάση γνώσης ανεξάρτητη από γειτονικά στιγμιότυπα. Για την επίτευξη καλύτερης απόδοσης, είναι δεδομένο ότι χρειάζεται να μελετηθούν και να εφαρμοστούν τεχνικές σταδιακού συλλογισμού, οι οποίες ενημερώνουν μέρος των συμπερασμάτων της βάσης γνώσης, καθώς μεταβάλλεται ένα μέρος αυτής, χωρίς να επανυπολογίζουν από την αρχή όλες τις υπονοούμενες προτάσεις. Επίσης, για την έκφραση των κανόνων του συστήματος, θα μπορούσαν να διερευνηθούν περισσότερο πρότυποι τρόποι έκφρασης – π.χ. XSLT για τους κανόνες αντιστοιχίας και SPARQL ερωτήματα για τους σημασιολογικούς κανόνες – οι οποίοι θα αξιοποιούν ώριμα σχετικά εργαλεία για την αποδοτικότερη διαχείριση του συνόλου κανόνων του συστήματος.

Ένα ακόμα ζήτημα που χρήζει διερεύνησης σχετίζεται με το κατά πόσο είναι ενδεδειγμένη λύση για το χειρισμό ροών δεδομένων η εκτέλεση ενός SPARQL ερωτήματος σε μια RDF βάση δεδομένων. Μια εναλλακτική προσέγγιση θα απαιτούσε τον επανασχεδιασμό του όλου συστήματος ώστε να εκμεταλλεύεται τις δυνατότητες διαχείρισης ροών κάποιου DSMS συστήματος, το οποίο και θα προσάρμοζε κατάλληλα ώστε να υποβάλλει συνεχή ερωτήματα σε ροές RDF δεδομένων, εφαρμόζοντας παράλληλα σε αυτά αλγορίθμους σταδιακού συλλογισμού.

Τέλος, η προσέγγιση που παρουσιάστηκε σε αυτό το κεφάλαιο δε λαμβάνει υπόψη της το ζήτημα της διαταραχής της διάταξης των στοιχείων της ροής που καταφθάνουν στο σύστημα. Το συγκεκριμένο πρόβλημα μπορεί να οδηγήσει σε παράβλεψη των στοιχείων μιας ροής που έρχονται καθυστερημένα και τα οποία θα έπρεπε υπό κανονικές συνθήκες να συμπεριληφθούν στο στιγμιότυπο ενός παραθύρου. Μια λύση στο πρόβλημα της διαταραχής είναι η υιοθέτηση ενός μέγιστου χρονικού ή φυσικού (εκφρασμένου σε αριθμό RDF τριάδων) ορίου, πέραν του οποίου μπορεί να υποτεθεί ότι δε θα καταφθάσει κάποια καθυστερημένη RDF τριάδα. Αυτή η τεχνική οδηγεί σε αντίστοιχη καθυστέρηση του υπολογισμού του παραθύρου, προκειμένου να εξασφαλιστεί ότι θα είναι διαθέσιμες όλες οι RDF τριάδες με τα απαιτούμενα χρονικά ορόσημα.

Κεφάλαιο 7

Συμπεράσματα και μελλοντική έρευνα

Στο κεφάλαιο αυτό, συμπυκνώνουμε τα κύρια συμπεράσματα της παρούσας διατριβής και συγκεντρώνουμε τη μελλοντική έρευνα για τους θεματικούς άξονες που εξετάστηκαν. Το κύριο θέμα που μας απασχόλησε ήταν η παραγωγή σημασιολογικού περιεχομένου από πηγές δεδομένων που δεν μπορούν να ενσωματωθούν ως έχουν στο Σημασιολογικό Ιστό και τα δεδομένα τους δεν μπορούν να αξιοποιηθούν στη μορφή που είναι από εφαρμογές που κάνουν χρήση σημασιολογικών τεχνολογιών. Η παραγωγή σημασιολογικού περιεχομένου μπορεί να ιδωθεί και ως μια πτυχή του ζητήματος της σημασιολογικής επισημείωσης δεδομένων, όπου τα δεδομένα επαυξάνονται με τη σημασία τους, ορισμένη σε τυπική μορφή, επεξεργάσιμη από μηχανή. Τα δύο είδη πηγών δεδομένων που εξετάστηκαν και τα οποία σηματοδοτούν τους δύο κύριους άξονες της διατριβής ήταν οι σχεσιακές βάσεις δεδομένων (κατά κύριο λόγο) και ημιδομημένες πηγές όπως δίκτυα αισθητήρων.

Στο κεφάλαιο 3 πραγματοποιήθηκε μια εκτεταμένη βιβλιογραφική επισκόπηση του θέματος της συμμετοχής, αξιοποίησης και ενσωμάτωσης σχεσιακών βάσεων δεδομένων στο Σημασιολογικό Ιστό. Κίνητρο για την πραγματοποίηση αυτής της επισκόπησης αποτέλεσε η πληθώρα σχετικών προσεγγίσεων στη βιβλιογραφία καθώς και η ανάγκη για μια συστηματική κατηγοριοποίησή τους, δεδομένης της συχνά παραπλανητικής χρήσης του γενικού όρου «αντιστοιχία σχεσιακής ΒΔ με οντολογία», ο οποίος χρησιμοποιείται για την αναφορά σε αυτές. Η συγκεκριμένη επισκόπηση έχει αρκετά καινοτομικά στοιχεία καθώς προτείνει ένα πλήρες σύστημα ταξινόμησης σε μη επικαλυπτόμενες κατηγορίες προσεγγίσεων. Αναλυτικά, αναγνωρίστηκαν τα προβλήματα της δημιουργίας μιας οντολογίας πεδίου ή μιας οντολογίας σχεσιακού σχήματος από μια σχεσιακή ΒΔ, του ορισμού ενός RDF γράφου από τα περιεχόμενα ενός στιγμιοτύπου σχεσιακής ΒΔ, με αναφορά σε περισσότερες από μία οντολογίες, καθώς και το πρόβλημα της ανακάλυψης αντιστοιχιών μεταξύ ενός σχεσιακού σχήματος και μιας δεδομένης οντολογίας. Για καθένα από αυτά τα προβλήματα, αναγνωρίστηκαν τα κίνητρα και οι ωφέλειες που προκύπτουν από τη λύση τους και ορίστηκαν ξεχωριστά πλαίσια για τη σύγκριση μεθόδων που καταπιάνονται με αυτά. Τέλος, παρουσιάστηκαν οι προοπτικές του συγκεκριμένου πεδίου και οι επικρατέστερες μελλοντικές ερευνητικές κατευθύνσεις. Ευελπιστούμε ότι η συγκεκριμένη επισκόπηση θα λειτουργήσει ως οδηγός και

θα συνεισφέρει στον καλύτερο συντονισμό μελλοντικών ερευνητικών προσπαθειών στο πεδίο αυτό.

Στο κεφάλαιο 4 παρουσιάστηκε ένα σύστημα ορισμού απλών αντιστοιχιών μεταξύ μιας σχεσιακής ΒΔ και μιας δεδομένης OWL οντολογίας. Το σύστημα VisAVis ήταν ένα από τα πρώτα συστήματα αντιστοιχίας στη σχετική βιβλιογραφία και κύριο προτέρημά του αποτελεί η χρήση της SQL για τον ορισμό της αντιστοιχίας, γεγονός που δεν επιβάλλει στο χρήστη την απαίτηση εκμάθησης μιας πολύπλοκης γλώσσας αντιστοιχίας. Το είδος της αντιστοιχίας που εισάγεται στο κεφάλαιο αυτό ορίζεται τυπικά, όπως και οι επιπτώσεις ενός συνόλου αντιστοιχιών σε μια βάση γνώσης, με την τελευταία να επεκτείνεται με ένα εικονικό σώμα ισχυρισμών. Παράλληλα, ορίζονται διαδικασίες συλλογισμού σε αυτό το εικονικό σώμα ισχυρισμών, εξετάζοντας δύο εναλλακτικές ερμηνείες για τα οντολογικά αξιώματα: μία που ακολουθεί την OWL σημασιολογία και μία που αντιμετωπίζει τα αξιώματα ως περιορισμούς ακεραιότητας. Οι προτεινόμενες διαδικασίες συλλογισμού συνίστανται στην εκτέλεση κατάλληλων SQL ερωτημάτων. Ο τυπικός ορισμός της προτεινόμενης αντιστοιχίας μας βοηθά να εντοπίσουμε ελλείψεις και να αναγνωρίσουμε τα ελάχιστα εκείνα στοιχεία που πρέπει να διαθέτει μια στοιχειώδης γλώσσα αντιστοιχίας, ήτοι ένα μηχανισμό παραγωγής IRI και τη δυνατότητα αντιστοιχίας ιδιοτήτων. Συνεπώς, το κόστος της απλότητας της προσέγγισής μας είναι η απουσία των συγκεκριμένων χαρακτηριστικών. Τέλος, προτείνονται τρόποι αξιοποίησης αντιστοιχιών αυτού του είδους σε σενάρια ολοκλήρωσης και ανταλλαγής δεδομένων.

Στο κεφάλαιο 5, παρουσιάστηκε ένας αλγόριθμος για την επανεγγραφή SPARQL ερωτημάτων σε ισοδύναμα SQL, λαμβάνοντας υπόψη R2RML αντιστοιχίες μεταξύ μιας σχεσιακής ΒΔ και ενός RDF γράφου. Ο αλγόριθμος αυτός αποτελεί τον πυρήνα ενός συστήματος που επιτρέπει την επερώτηση εικονικών RDF γράφων ή ισοδύναμα, τη δυναμική πρόσβαση στα περιεχόμενα μιας ΒΔ μέσω SPARQL. Εξ όσων γνωρίζουμε, ο προτεινόμενος αλγόριθμος αποτελεί τον πρώτο ή έναν εκ των πρώτων ολοκληρωμένων και τεκμηριωμένων αλγορίθμων SPARQL-σε-SQL επανεγγραφής, υπό την παρουσία R2RML αντιστοιχιών. Επίσης, ένα από τα χαρακτηριστικά του εν λόγω αλγορίθμου που τον διαφοροποιεί από τους υπόλοιπους είναι η δυνατότητα να παράγει όσο το δυνατόν πιο επίπεδα SQL ερωτήματα με το μικρότερο δυνατό αριθμό συνενώσεων, αξιοποιώντας κοινές μεταβλητές μεταξύ των προτύπων τριάδων του εισερχόμενου SPARQL ερωτήματος. Η χρήση ενός αλγορίθμου SPARQL-σε-SQL επανεγγραφής επιτρέπει σε ένα σύστημα αντιστοιχίας να λειτουργήσει δυναμικά, ανακτώντας πάντα επίκαιρα δεδομένα της ΒΔ και αποφεύγοντας την ανάγκη επιπρόσθετου αποθηκευτικού χώρου για τη διατήρηση του RDF γράφου σε φυσική μορφή. Τα αποτελέσματα της εφαρμογής του προτεινόμενου αλγορίθμου έδειξαν ότι αυτός υπερέχει σχετικών προσεγγίσεων σε περιπτώσεις σχεσιακών ΒΔ μεγάλου όγκου.

Τέλος, το κεφάλαιο 6 καταπιάνεται με το ζήτημα της παραγωγής σημασιολογικού περιεχομένου από ημιδομημένες πηγές όπως δίκτυα αισθητήρων. Παρουσιάζεται μια αρχιτεκτονική σημασιολογικής επεξεργασίας αισθητήριων παρατηρήσεων για την αναγνώριση γεγονότων υψηλού επιπέδου και προτείνεται μια επέκταση σε ένα σύστημα επίγνωσης περιβάλλοντος, βασισμένη σε παράθυρα, με στόχο τη βελτίωση της απόδοσής του υπό καθεστώς συνεχούς λειτουργίας. Η εφαρμογή ολισθαίνοντων και μιας παραλλαγής επάλληλων φυ-

σικών παραθύρων RDF προτάσεων έδειξε ότι ο συνολικός χρόνος απόκρισης παραμένει εντός συγκεκριμένων χρονικών ορίων τα οποία εξαρτώνται από το επιλεγόμενο μήκος του παραθύρου. Επίσης, ορίστηκε και δοκιμάστηκε η εφαρμογή ενός νέου είδους φυσικού παραθύρου, του παραθύρου οντολογικών ατόμων, το μήκος του οποίου ορίζεται με βάση το πλήθος οντοτήτων ενδιαφέροντος και το οποίο περιέχει εκείνες τις RDF προτάσεις που περιγράφουν τις οντότητες που βρίσκονται εντός του παραθύρου. Το παράθυρο οντολογικών ατόμων βοηθά να ξεπεραστεί το πρόβλημα της ελλιπούς γνώσης για μια οντότητα, το οποίο χαρακτηρίζει τα φυσικά παράθυρα RDF προτάσεων, ενώ καθιστά ευκολότερη και την κατανόηση της σχέσης ανταλλαγής μεταξύ του χρόνου απόκρισης του συστήματος και της ποιότητας και πληρότητας των εξαγόμενων συμπερασμάτων, καθώς οι παράμετροι των παραθύρων εκφράζονται σε όρους οντοτήτων του πεδίου της εφαρμογής. Εντούτοις, δεν μπορεί να εγγυηθεί ότι ο χρόνος απόκρισης θα βρίσκεται εντός συγκεκριμένων χρονικών ορίων, καθώς το συγκεκριμένο είδος παραθύρου δεν έχει σταθερό αριθμό RDF προτάσεων.

Σε ό,τι αφορά στα μελλοντικά ερευνητικά θέματα που σχετίζονται με τους θεματικούς άξονες της παρούσας διατριβής, αυτά αναφέρθηκαν σε μεγάλο βαθμό στις ενότητες 5.5 και 6.5 και συνοψίζονται στα επόμενα.

Αρχικά, σχετικά με το ζήτημα της διεπαφής σχεσιακών ΒΔ με το Σημασιολογικό Ιστό, εκτός από την περαιτέρω βελτιστοποίηση του αλγορίθμου που παρουσιάστηκε στο κεφάλαιο 5, στόχος είναι η ενασχόληση με όλες τις πτυχές ενός ολοκληρωμένου συστήματος αντιστοιχίας που καθιστά μια σχεσιακή ΒΔ μέρος του Ιστού Δεδομένων. Κατ' αρχήν, το ζήτημα της ενημέρωσης ενός εικονικού RDF γράφου που αντικατοπτρίζει τα περιεχόμενα μιας ΒΔ είναι ένα κρίσιμο πρόβλημα, παρόμοιο με το πρόβλημα της ενημέρωσης σχεσιακών όψεων, και η αντιμετώπισή του ουσιαστικά θα οδηγήσει στην προσέγγιση του οράματος ενός Ιστού Δεδομένων ανάγνωσης και εγγραφής. Εξίσου σημαντικό είναι και το ζήτημα του συλλογισμού σε εικονικούς RDF γράφους, καθώς δεν επαρκούν υπάρχοντα εργαλεία συλλογισμού τα οποία μπορούν να εφαρμοστούν μονάχα σε υλοποιημένους RDF γράφους. Προς αυτή την κατεύθυνση, θα χρειαστεί να προσαρμοστούν αλγόριθμοι επανεγγραφής, έτσι ώστε να ενσωματώνεται στο αρχικό SPARQL ερώτημα και η επίδραση οντολογικών αξιωμάτων. Ακόμα ένα ανοικτό ερευνητικό πρόβλημα είναι το ζήτημα της αυτόματης ανακάλυψης αντιστοιχιών μεταξύ των περιεχομένων μιας σχεσιακής ΒΔ και οντοτήτων του πραγματικού κόσμου, οι οποίες αναπαρίστανται ως όροι οντολογιών και λεξιλογίων του Σημασιολογικού Ιστού. Λύσεις στο εν λόγω πρόβλημα, εκτός της διευκόλυνσης του χρήστη που είναι υπεύθυνος για τον ορισμό μιας αντιστοιχίας, θα οδηγήσουν στην παραγωγή πραγματικά Συνδεδεμένων Δεδομένων, αυξάνοντας τη συνεκτικότητα του Σύννεφου Συνδεδεμένων Δεδομένων και προσεγγίζοντας ακόμα περισσότερο το όραμα του Ιστού Δεδομένων. Τέλος, ένα ακόμα θέμα άξιο διερεύνησης είναι η αυτόματη ενημέρωση μιας αντιστοιχίας σε περίπτωση μεταβολής του σχεσιακού σχήματος.

Τέλος, όσον αφορά στο πρόβλημα της αποτελεσματικής σημασιολογικής επεξεργασίας και συλλογισμού σε ροές RDF δεδομένων, παρουσιάζει ιδιαίτερο ενδιαφέρον η μελέτη παραθύρων που σχηματίζονται με βάση κάποια σύνθετη συνθήκη η οποία μπορεί να εκφραστεί ως ένα σημασιολογικό ερώτημα. Τέτοια παράθυρα αποτελούν επέκταση της έννοιας των παραθύρων κατηγορήματος και μπορεί να λαμβάνουν υπόψη τους ακόμα και τη σχέση ενός στοιχείου της

ροής με κάποιο άλλο, σε αντίθεση με τις μέχρι τώρα κατηγορίες παραθύρων που σχηματίζονται με βάση το χρονικό ορόσημο ή κάποιο άλλο γνώρισμα ενός στοιχείου. Επίσης, αξίζει να διερευνηθεί μια εναλλακτική αντιμετώπιση των προκλήσεων που θέτει ο συλλογισμός σε RDF ροές, με χρήση τεχνικών σταδιακού συλλογισμού, καθώς και αξιοποίηση υπαρχόντων συστημάτων διαχείρισης ροών για το σχηματισμό των παραθύρων.

■

Γλωσσάριο όρων

annotation : επισημείωση

benchmark : μεθοδολογία μέτρησης επιδόσεων

Closed World Assumption : Υπόθεση Κλειστού Κόσμου

conjunctive query : συζευκτικό ερώτημα

continuous query : συνεχές ερώτημα

Direct Mapping : Άμεση Αντιστοιχία

Extended Entity-Relationship Model : Εκτεταμένο Μοντέλο Οντοτήτων-Συσχετίσεων

Deep Web : Βαθύς Ιστός

integration : ολοκλήρωση

Linked Data : Συνδεδεμένα Δεδομένα

microformat : μικροπρότυπο

object-relational impedance mismatch : ασυμφωνία σχεσιακού και αντικειμενοστρεφούς μοντέλου

ontology-based data access : πρόσβαση σε δεδομένα με χρήση οντολογιών

ontology-based integration : ολοκλήρωση με χρήση οντολογιών

ontology learning : οντολογική μάθηση

Open World Assumption : Υπόθεση Ανοικτού Κόσμου

query rewriting : επανεγγραφή ερωτημάτων

reasoning : συλλογισμός

reverse engineering : αντίστροφη μηχανική

scalability : κλιμακωσιμότητα

Semantic Web : Σημασιολογικός Ιστός

Sensor Web : Ιστός Αισθητήρων

sliding window : ολισθαίνον παράθυρο

SPARQL endpoint : τελικό σημείο SPARQL

timestamp : χρονικό ορόσημο

tumbling window : επάλληλο παράθυρο

Unique Name Assumption : Υπόθεση Μοναδικών Ονομάτων

Uniform Resource Identifier : Ενιαίο Αναγνωριστικό Πόρου

Web of data : Ιστός δεδομένων

wrapper : σύστημα-κέλυφος

Ακρωνύμια

ΒΔ : Βάση Δεδομένων

ΣΔΒΔ : Σύστημα Διαχείρισης Βάσης Δεδομένων

BGP : Basic Graph Pattern

CSV : Comma-Separated Values

CWA : Closed World Assumption

DBMS : Database Management System

DSMS : Data Stream Management System

DDL : Data Definition Language

DML : Data Manipulation Language

EER : Extended Entity-Relationship Model

GAV : Global as View

HTML : HyperText Markup Language

IRI : Internationalized Resource Identifier

LAV : Local as View

OBDA : Ontology-Based Data Access

OWA : Open World Assumption

R2RML : RDB to RDF Mapping Language

RDF : Resource Description Framework

RDFa : Resource Description Framework-in-attributes

RDQL : RDF Data Query Language

REST : Representational State Transfer

SPARQL : SPARQL Protocol and RDF Query Language

SQL : Structured Query Language

SSN : Semantic Sensor Network

UML : Universal Modeling Language

UNA : Unique Name Assumption

URI : Uniform Resource Identifier

XHTML : Extensible HyperText Markup Language

XML : eXtensible Markup Language

Βιβλιογραφία

[1] Δ.-Ε. Σπανός και Μ. Χαλάς, *Απεικόνιση Οντολογιών σε Σχήματα Σχεσιακών Βάσεων Δεδομένων με σκοπό την Ανάκτηση Δεδομένων Σημασιολογικού Περιεχομένου*, Διπλωματική Εργασία, Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών, Εθνικό Μετσόβιο Πολυτεχνείο, Σεπτέμβριος 2005.

[2] ISO/IEC 9075-14:2008, SQL Part 14: XML-Related Specifications (SQL/XML), International Organization for Standardization, 27 January 2009.

[3] Α. Ζαφειρόπουλος, *Παροχή Προηγμένων Υπηρεσιών με Επίγνωση Περιβάλλοντος σε Δυναμικά Ετερογενή Δίκτυα*, Διδακτορική Διατριβή, Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών, Εθνικό Μετσόβιο Πολυτεχνείο, Φεβρουάριος 2011.

[4] D. Abadi, Y. Ahmad, M. Balazinska, U. Cetintemel, M. Cherniack, J. Hwang, W. Lindner, A. Maskey, A. Rasin, E. Ryvkina, N. Tatbul, Y. Xing and S. Zdonik, The Design of the Borealis Stream Processing Engine, in *Second Biennial Conference on Innovative Data Systems Research (CIDR 2005)*, pp. 277–289, 2005.

[5] S. Abiteboul, Querying Semi-Structured Data, in *Proceedings of the 6th International Conference on Database Theory (ICDT '97)*, pp. 1–18, 1997.

[6] N. Alalwan, H. Zedan and F. Siewe, Generating OWL Ontology for Database Integration, in P. Dini, J. Hendler, J. Noll, R. Witte, F. Zavoral, D. Roman, U. Straccia, M. Paolucci, P. Yeh, S. Athenikos, N. Dokoohaki, A. Kumar and M. Nagarajan, eds., *Proceedings of the Third International Conference on Advances in Semantic Processing (SEMAPRO 2009)*, pp. 22–31, IEEE, 2009.

[7] K. M. Albarrak and E. H. Sibley, Translating Relational & Object-Relational Database Models into OWL Models, in S. Rubin and S.-C. Chen, eds., *Proceedings of the 2009 IEEE International Conference on Information Reuse & Integration (IRI 2009)*, pp. 336–341, IEEE, 2009.

[8] R. Alhajj, Extracting the Extended Entity-Relationship Model from a Legacy Relational Database, *Information Systems*, **28**(6), pp. 597–618, 2003.

[9] Y. An, A. Borgida, R. J. Miller and J. Mylopoulos, A Semantic Approach to Discovering Schema Mapping Expressions, in R. Chirkova and V. Oria, eds., *Proceedings of 2007 IEEE 23rd International Conference on Data Engineering (ICDE 2007)*, pp. 206–215, IEEE, 2007.

[10] Y. An, A. Borgida and J. Mylopoulos, Discovering the Semantics of Relational Tables through Mappings, *Journal on Data Semantics*, **VII**, pp. 1–32, 2006.

[11] Y. An, X. Hu and I.-Y. Song, Round-Trip Engineering for Maintaining Conceptual-Relational Mappings, in Z. Bellahsène and M. Léonard, eds., *Advanced*

Information Systems Engineering: 20th International Conference (CAiSE 2008), *Lecture Notes on Computer Science*, vol. 5074, pp. 296–311, Springer, 2008.

[12] D. Anicic, P. Fodor, S. Rudolph and N. Stojanovic, EP-SPARQL : A Unified Language for Event Processing and Stream Reasoning, in *Proceedings of the 20th International Conference on World Wide Web (WWW '11)*, pp. 635–644, 2011.

[13] A. Arasu, B. Babcock, S. Babu, J. Cieslewicz, M. Datar, K. Ito, R. Motwani, U. Srivastava and J. Widom, STREAM: The Stanford Data Stream Management System, Tech. rep., Stanford InfoLab, 2004.

[14] M. Arenas, A. Bertails, E. Prud'hommeaux and J. Sequeda, A Direct Mapping of Relational Data to RDF , available at: http://www.w3.org/TR/rdb-direct-mapping/, August 2012.

[15] I. Astrova, Reverse Engineering of Relational Databases to Ontologies, in C. J. Bussler, J. Davies, D. Fensel and R. Studer, eds., *The Semantic Web: Research and Applications: First European Semantic Web Symposium (ESWS 2004)*, *Lecture Notes in Computer Science*, vol. 3053, pp. 327–341, Springer, 2004.

[16] I. Astrova, Rules for Mapping SQL Relational Databases to OWL Ontologies, in M.-A. Sicilia and M. D. Lytras, eds., *Metadata and Semantics*, pp. 415–424, Springer, 2009.

[17] P. Atzeni, S. Paolozzi and P. Del Nostro, Ontologies and Databases: Going Back and Forth, in *Proceedings of the 4th International VLDB Workshop on Ontology-based Techniques for Databases in Information Systems and Knowledge Systems (ODBIS 2008)*, pp. 9 – 16, 2008.

[18] S. Auer, S. Dietzold, J. Lehmann, S. Hellmann and D. Aumueller, Triplify: Light-Weight Linked Data Publication from Relational Databases, in *Proceedings of the 18th International Conference on World Wide Web*, pp. 621–630, ACM, 2009.

[19] F. Baader, D. L. McGuinness, P. F. Patel-Schneider and D. Nardi, *The Description Logic Handbook: Theory, Implementation, and Applications*, Cambridge University Press, 2nd ed., 2007.

[20] B. Babcock, S. Babu, M. Datar, R. Motwani and J. Widom, Models and Issues in Data Stream Systems, in *Proceedings of the Twenty-First ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS '02)*, pp. 1–16, 2002.

[21] M. Baglioni, M. V. Masserotti, C. Renso and L. Spinsanti, Building Geospatial Ontologies from Geographical Databases, in F. Fonseca, M. A. Rodríguez and S. Levashkin, eds., *GeoSpatial Semantics: Second International Conference (GeoS 2007)*, *Lecture Notes in Computer Science*, vol. 4853, pp. 195–209, Springer, 2007.

[22] D. F. Barbieri, D. Braga, S. Ceri, E. Della Valle and M. Grossniklaus, Continuous Queries and Real-Time Analysis of Social Semantic Data with C-SPARQL, in *Proceedings of the 2nd Workshop on Social Data on the Web (SDoW2009)*, *collocated with the 8th International Semantic Web Conference (ISWC 2009)*, 2009.

[23] D. F. Barbieri, D. Braga, S. Ceri, E. Della Valle and M. Grossniklaus, Incremental Reasoning on Streams and Rich Background Knowledge, in *The Semantic Web: Research and Applications*, *Proceedings of the 7th 7th Extended Semantic Web Conference (ESWC '10)*, pp. 1–15, 2010.

[24] D. F. Barbieri, D. Braga, S. Ceri and M. Grossniklaus, An Execution Environment for C-SPARQL Queries, in *Proceedings of the 13th International Conference on Extending Database Technology (EDBT '10)*, pp. 441–452, 2010.

[25] D. F. Barbieri and E. Della Valle, A Proposal for Publishing Data Streams as Linked Data, in *Proceedings of the WWW2010 Workshop on Linked Data on the Web (LDOW 2010)*, 2010.

[26] P. Barnaghi, M. Compton, O. Corcho, R. García Castro, J. Graybeal, A. Herzog, K. Janowicz, H. Neuhaus, A. Nikolov and K. Page, Semantic Sensor Network XG Final Report, available at: http://www.w3.org/2005/Incubator/ssn/XGR-ssn/, June 2011.

[27] P. Barnaghi and M. Presser, Publishing Linked Sensor Data, in *Proceedings of the 3rd International Workshop on Semantic Sensor Networks (SSN10)*, 2010.

[28] J. Barrasa - Rodriguez and A. Gómez-Pérez, Upgrading Relational Legacy Data to the Semantic Web, in *Proceedings of the 15th International Conference on World Wide Web (WWW'06)*, pp. 1069–1070, ACM, 2006.

[29] J. Barrasa, O. Corcho and A. Gómez-Pérez, R2O, an Extensible and Semantically Based Database-to-Ontology Mapping Language, in *Proceedings of the 2nd Workshop on Semantic Web and Databases (SWDB 2004)*, 2004.

[30] D. Beckett and J. Grant, SWAD-Europe Deliverable 10.2: Mapping Semantic Web Data with RDBMSes, Tech. rep., 2003, available at: http://www.w3.org/2001/sw/Europe/reports/scalable_rdbms_mapping_report/.

[31] A. Behm, A. Geppert and K. R. Dittrich, On The Migration of Relational Schemas and Data to Object-Oriented Database Systems, in *Proceeding of the 5th International Conference on Re-Technologies for Information Systems (ReTIS 1997)*, pp. 13–33, 1997.

[32] D. Bell, B. R. Heravi and M. Lycett, Sensory Semantic User Interfaces (SenSUI), in *Proceedings of the 2nd International Workshop on Semantic Sensor Networks (SSN09), collocated with the 8th International Semantic Web Conference (ISWC 2009)*, pp. 96–109, 2009.

[33] M. Ben-Ari, *Principles of Concurrent and Distributed Programming*, chap. 13, Addison-Wesley, second ed., 2006.

[34] C. Ben Necib and J.-C. Freytag, Semantic Query Transformation Using Ontologies, in B. C. Desai and G. Vossen, eds., *Proceedings of 9th International Database Engineering & Application Symposium (IDEAS 2005)*, pp. 187–199, IEEE, 2005.

[35] T. Berners-Lee, Relational Databases on the Semantic Web, available at: http://www.w3.org/DesignIssues/RDB-RDF.html, 1998.

[36] T. Berners-Lee, Semantic Web Road Map, available at: http://www.w3.org/DesignIssues/Semantic.html, September 1998.

[37] T. Berners-Lee, Linked Data - Design Issues, available at: http://www.w3.org/DesignIssues/LinkedData.html, July 2006.

[38] T. Berners-Lee, J. Hendler and O. Lassila, The Semantic Web, *Scientific American*, (May), 2001.

[39] C. Bizer and R. Cyganiak, D2R Server - Publishing Relational Databases on the Semantic Web, *poster in 5th International Semantic Web Conference (ISWC 2006)*, 2006.

[40] C. Bizer and A. Schultz, The Berlin SPARQL Benchmark, *International Journal On Semantic Web and Information Systems*, **5**(2), pp. 1–24, 2009.

[41] C. Bizer and A. Seaborne, D2RQ - Treating non-RDF Databases as Virtual RDF Graphs, *poster in 3rd International Semantic Web Conference (ISWC 2004)*, 2004.

[42] C. Blakeley, Virtuoso RDF Views – Getting Started Guide, available at: http://www.openlinksw.co.uk/virtuoso/Whitepapers/pdf/Virtuoso_SQL_to_RDF_Mapping.pdf, OpenLink Software, 2007.

[43] A. Bolles, M. Grawunder and J. Jacobi, Streaming SPARQL - Extending SPARQL to Process Data Streams, in *The Semantic Web: Research and Applications*, *Proceedings of the 5th European Semantic Web Conference (ESWC 2008)*, pp. 448–462, Springer, 2008.

[44] M. Botts and A. Robin, Bringing the Sensor Web together, *Geosciences*, **6**, pp. 46–53, 2007.

[45] E. Bouillet, M. Feblowitz, Z. Liu, A. Ranganathan, A. Riabov and F. Ye, A Semantics-based Middleware for Utilizing Heterogeneous Sensor Networks, in *Proceedings of the Third IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS 2007)*, pp. 174–188, 2007.

[46] D. Brickley and R. Guha, RDF Vocabulary Description Language 1.0: RDF Schema, available at: http://www.w3.org/TR/rdf-schema/, February 2004.

[47] A. Buccella, M. R. Penabad, F. R. Rodriguez, A. Farina and A. Cechich, From Relational Databases to OWL Ontologies, in *Proceedings of 6th Russian Conference on Digital Libraries (RCDL 2004)*, 2004.

[48] G. Būmans and K. Čerāns, RDB2OWL : a Practical Approach for Transforming RDB Data into RDF/OWL, in A. Paschke, N. Henze and T. Pellegrini, eds., *Proceedings of the 6th International Conference on Semantic Systems (I-SEMANTICS 2010)*, ACM, 2010.

[49] K. Byrne, Having Triplets - Holding Cultural Data as RDF, in M. Larson, K. Fernie, O. J. and J. Cigarran, eds., *Proceedings of the ECDL 2008 Workshop on Information Access to Cultural Heritage*, 2008.

[50] J.-P. Calbimonte, O. Corcho and A. J. G. Gray, Enabling Ontology-based Access to Streaming Data Sources, in *The Semantic Web*, *Proceedings of the 9th International Semantic Web Conference (ISWC '10)*, pp. 96–111, 2010.

[51] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, A. Poggi, M. Rodriguez-Muro, R. Rosati, M. Ruzzi and D. F. Savo, The MASTRO System for Ontology-based Data Access, *Semantic Web Journal*, **2**(1), pp. 43–53, 2011.

[52] J. J. Carroll, C. Bizer, P. Hayes and P. Stickler, Named Graphs, *Web Semantics: Science, Services and Agents on the World Wide Web*, **3**(4), pp. 247–267, 2005.

[53] J. J. Carroll and P. Stickler, TriX: RDF Triples in XML, Tech. Rep. HPL-2004-56, HP Laboratories, Bristol, 2004.

[54] F. Cerbah, Mining the Content of Relational Databases to Learn Ontologies with Deeper Taxonomies, in Y. Li, G. Pasi, C. Zhang, N. Cercone and L. Cao, eds., *Proceedings of 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology Workshops(WI-IAT 2008)*, pp. 553–557, IEEE, 2008.

[55] P.-A. Champin, G.-J. Houben and P. Thiran, CROSS: an OWL Wrapper for Reasoning on Relational Databases, in C. Parent, K.-D. Schewe, V. C. Storey and B. Thalheim, eds., *Conceptual Modeling - ER 2007: 26th International Conference on Conceptual Modeling*, *Lecture Notes in Computer Science*, vol. 4801, pp. 502–517, Springer, 2007.

[56] S. Chandrasekaran, O. Cooper, A. Deshpande, M. Franklin, J. Hellerstein, W. Hong, S. Krishnamurthy, S. Madden, V. Raman, F. Reiss and M. Shah, TelegraphCQ: Continuous Dataflow Processing for an Uncertain World, in *Proceedings of the First Biennial Conference on Innovative Data Systems Research (CIDR 2003)*, 2003.

[57] A. Chebotko, S. Lu and F. Fotouhi, Semantics Preserving SPARQL-to-SQL Translation, *Data & Knowledge Engineering*, **68**(10), pp. 973–1000, 2009.

[58] A. Chebotko, S. Lu, H. M. Jamil and F. Fotouhi, Semantics Preserving SPARQL-to-SQL Query Translation for Optional Graph Patterns, Tech. Rep. TR-DB-052006-CLJF, Wayne State University, 2006.

[59] H. Chen, Z. Wu, Y. Mao and G. Zheng, DartGrid: a Semantic Infrastructure for Building Database Grid Applications, *Concurrency and Computation: Practice and Experience*, **18**(14), pp. 1811–1828, 2006.

[60] J. Chen, D. DeWitt, F. Tian and Y. Wang, NiagaraCQ: A Scalable Continuous Query System for Internet Databases, in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data (SIGMOD '00)*, pp. 379–390, ACM, 2000.

[61] P. P.-S. Chen, The Entity-Relationship Model - Toward a Unified View of Data, *ACM Transactions on Database Systems (TODS)*, **1**(1), pp. 9–36, 1976.

[62] R. H. Chiang, T. M. Barron and V. C. Storey, Reverse Engineering of Relational Databases: Extraction of an EER Model from a Relational Database, *Data & Knowledge Engineering*, **12**(2), pp. 107–142, 1994.

[63] E. Codd, A Relational Model of Data for Large Shared Data Banks, *Communications of the ACM*, **13**(6), pp. 377–387, 1970.

[64] M. Compton, P. Barnaghi, L. Bermudez, R. García-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, V. Huang, K. Janowicz, W. D. Kelsey, D. Le Phuoc, L. Lefort, M. Leggieri, H. Neuhaus, A. Nikolov, K. Page, A. Passant, A. Sheth and K. Taylor, The SSN Ontology of the W3C Semantic Sensor Network Incubator Group, *Web Semantics: Science, Services and Agents on the World Wide Web*, **17**, pp. 25–32, 2012.

[65] M. Compton, C. Henson, L. Lefort, H. Neuhaus and A. Sheth, A Survey of the Semantic Specification of Sensors, in *Proceedings of the 2nd International Workshop on Semantic Sensor Networks (SSN09), collocated with the 8th International Semantic Web Conference (ISWC 2009)*, pp. 17–32, 2009.

[66] C. Curino, G. Orsi, E. Panigati and L. Tanca, Accessing and Documenting Relational Databases through OWL Ontologies, in T. Andreasen, R. R. Yager, H. Bulskov, H. Christiansen and H. Legind Larsen, eds., *Flexible Query Answering Systems: 8th International Conference (FQAS 2009)*, *Lecture Notes in Computer Science*, vol. 5822, pp. 431–442, Springer, 2009.

[67] R. Cyganiak, A Relational Algebra for SPARQL, Tech. Rep. HPL-2005-170, Hewlett-Packard, 2005.

[68] S. Das and J. Srinivasan, Database Technologies for RDF, in S. Tessaris, E. Franconi, T. Eiter, C. Gutierrez, S. Handschuh, M.-C. Rousset and R. A. Schmidt, eds., *Reasoning Web. Semantic Technologies for Information Systems: 5th International Summer School 2009*, *Lecture Notes in Computer Science*, vol. 5689, pp. 205–221, Springer, 2009.

[69] S. Das, S. Sundara and R. Cyganiak, R2RML: RDB to RDF Mapping Language, available at: http://www.w3.org/TR/r2rml/, September 2012.

[70] J. de Bruijn, F. Martin-Recuerda, D. Manov and M. Ehrig, State-of-the-art Survey on Ontology Merging and Aligning, SEKT Project, Deliverable D4.2.1, 2004.

[71] M. del Mar Roldan-Garcia and J. F. Aldana-Montes, A Survey on Disk Oriented Querying and Reasoning on the Semantic Web, in *Proceedings of 22nd International Conference on Data Engineering Workshops (ICDEW'06)*, IEEE, 2006.

[72] E. Della Valle, S. Ceri, D. F. Barbieri, D. Braga and A. Campi, A First Step Towards Stream Reasoning, in *Proceedings of the Future Internet Symposium 2008 (FIS 2008)*, pp. 72–81, Springer, 2008.

[73] L. Ding, T. Finin, A. Joshi, Y. Peng, P. Pinheiro da Silva and D. L. McGuinness, Tracking RDF Graph Provenance using RDF Molecules, in *Proceedings of the 4th International Semantic Web Conference (ISWC 2005)*, 2005.

[74] C. Dolbear and G. Hart, Ontological Bridge Building - Using Ontologies to Merge Spatial Datasets, in D. L. McGuinness, P. Fox and B. Brodaric, eds., *Semantic Scientific Knowledge Integration: AAAI Spring Symposium (AAAI/SSS Workshop)*, pp. 15–20, 2008.

[75] E. Dragut and R. Lawrence, Composing Mappings between Schemas using a Reference Ontology, in R. Meersman and Z. Tari, eds., *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE*, *Lecture Notes in Computer Science*, vol. 3290, pp. 783–800, Springer, 2004.

[76] V. Eisenberg and Y. Kanza, D2RQ/Update: Updating Relational Data via Virtual RDF, in *Proceedings of the 21st International conference companion on World Wide Web (WWW'12 Companion)*, pp. 497–498, 2012.

[77] B. Elliott, E. Cheng, C. Thomas-Ogbuji and Z. M. Ozsoyoglu, A Complete Translation from SPARQL into Efficient SQL, in B. C. Desai, ed., *Proceedings of the 2009 International Database Engineering & Applications Symposium (IDEAS '09)*, pp. 31–42, ACM, 2009.

[78] R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, The Benjamin/Cummings Publishing Company, Inc., San Francisco, CA, USA, 6th ed., 2010.

[79] R. Fagin, P. G. Kolaitis, R. J. Miller and L. Popa, Data Exchange: Semantics and Query Answering, *Theoretical Computer Science*, **336**(1), pp. 89–124, 2005.

[80] M. Fahad, ER2OWL: Generating OWL Ontology from ER Diagram, in Z. Shi, E. Mercier-Laurent and D. Leake, eds., *Intelligent Information Processing IV: 5th IFIP International Conference on Intelligent Information Processing*, pp. 28–37, Springer, 2008.

[81] C. Fahrner and G. Vossen, A Survey of Database Design Transformations based on the Entity-Relationship Model, *Data & Knowledge Engineering*, **15**(3), pp. 213–250, 1995.

[82] K.-W. Fan, S. Liu and P. Sinha, Structure-Free Data Aggregation in Sensor Networks, *IEEE Transactions on Mobile Computing*, **6**(8), pp. 929–942, 2007.

[83] M. Fisher, M. Dean and G. Joiner, Use of OWL and SWRL for Semantic Relational Database Translation, in K. Clark and P. F. Patel-Schneider, eds., *Proceedings of the Fourth OWLED Workshop on OWL: Experiences and Directions*, 2008.

[84] A. Garrote and M. N. Moreno García, RESTful writable APIs for the web of Linked Data using relational storage solutions, in *Proceedings of the 4th Linked Data on the Web Workshop (LDOW2011)*, 2011.

[85] J. Geller, S. A. Chun and Y. J. An, Toward the Semantic Deep Web, *Computer*, **41**(9), pp. 95–97, 2008.

[86] T. M. Ghanem, W. G. Aref and A. K. Elmagarmid, Exploiting Predicate-Window Semantics over Data Streams, *ACM SIGMOD Record*, **35**(1), pp. 3–8, 2006.

[87] R. Ghawi and N. Cullot, Database-to-Ontology Mapping Generation for Semantic Interoperability, in *3rd International Workshop on Database Interoperability (InterDB 2007), held in conjunction with VLDB 2007*, 2007.

[88] L. Golab and M. T. Özsu, Issues in Data Stream Management, *ACM SIGMOD Record*, **32**(2), pp. 5–14, 2003.

[89] A. Gomez-Perez, O. Corcho-Garcia and M. Fernandez-Lopez, *Ontological Engineering*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1st ed., 2003.

[90] A. J. G. Gray, N. Gray and I. Ounis, Can RDB2RDF Tools Feasibly Expose Large Science Archives for Data Integration?, in L. Aroyo, P. Traverso, F. Ciravegna, P. Cimiano, T. Heath, E. Hyvönen, R. Mizoguchi, E. Oren, M. Sabou and E. Simperl, eds., *The Semantic Web: Research and Applications: 6th European Semantic Web Conference (ESWC 2009)*, *Lecture Notes in Computer Science*, vol. 5554, pp. 491–505, Springer, 2009.

[91] T. Gruber, Toward Principles for the Design of Ontologies Used for Knowledge Sharing, *International Journal of Human-Computer Studies*, **43**(5-6), pp. 907–928, 1995.

[92] N. Guarino, Formal Ontology and Information Systems, in *Formal Ontologies in Information Systems: Proceedings of the 1st International Conference (FOIS'98)*, pp. 3–15, IOS Press, 1998.

[93] S. Harris and A. Seaborne, SPARQL 1.1 Query Language, available at: http://www.w3.org/TR/sparql11-query/, July 2012.

[94] S. Harris and N. Shadbolt, SPARQL Query Processing with Conventional Relational Database Systems, in *Web Information Systems Engineering, Proceedings of WISE 2005 International Workshops*, pp. 235–244, Springer, 2005.

[95] T. Heath and C. Bizer, *Linked Data: Evolving the Web into a Global Data Space*, Morgan & Claypool Publishers, San Rafael, 2011.

[96] F. Heintz, J. Kvarnstrom and P. Doherty, Stream Reasoning in DyKnow: A Knowledge Processing Middleware System, in *Proceedings of the 1st International Workshop on Stream Reasoning (SR 2009), collocated with the 6th European Semantic Web Conference (ESWC 2009)*, 2009.

[97] S. Hellmann, J. Unbehauen, A. Zaveri, J. Lehmann, S. Auer, S. Tramp, H. Williams, O. Erling, T. Thibodeau Jr, K. Idehen, A. Blumauer and H. Nagy, Report on Knowledge Extraction from Structured Sources, LOD2 Project, Deliverable 3.1.1, available at: http://static.lod2.eu/Deliverables/deliverable-3.1.1.pdf, 2011.

[98] J. Hendler, Web 3.0: Chicken Farms on the Semantic Web, *Computer*, **41**(1), pp. 106–108, 2008.

[99] C. Henson, J. Pschorr, A. Sheth and K. Thirunarayan, SemSOS: Semantic Sensor Observation Service, in *Proceedings of International Symposium on Collaborative Technologies and Systems (CTS 2009)*, pp. 44–53, IEEE, 2009.

[100] M. Hert, G. Reif and H. C. Gall, Updating Relational Data via SPARQL/Update, in F. Daniel, L. Delcambre, F. Fotouhi, I. Garrigòs, G. Guerrini, J.-N. Mazòn, M. Mesiti, S. Müller-Feuerstein, J. Trujillo, T. M. Truta, B. Volz, E. Waller, L. Xiong and E. Zimányi, eds., *Proceedings of the 2010 EDBT/ICDT Workshops*, ACM, 2010.

[101] M. Hert, G. Reif and H. C. Gall, A Comparison of RDB-to-RDF Mapping Languages, in C. Ghidini, A.-C. Ngonga Ngomo, S. Lindstaedt and T. Pellegrini, eds., *Proceedings of the 7th International Conference on Semantic Systems (I-SEMANTICS 2011)*, pp. 25–32, ACM, 2011.

[102] G. Hillairet, F. Bertrand and J. Y. Lafaye, MDE for Publishing Data on the Semantic Web, in F. Silva Parreiras, J. Z. Pan, U. Assmann and J. Henriksson, eds., *Transforming and Weaving Ontologies in Model Driven Engineering: Proceedings of the 1st International Workshop (TWOMDE 2008)*, pp. 32–46, 2008.

[103] P. Hitzler, M. Krötzsch and S. Rudolph, *Foundations of Semantic Web Technologies*, Textbooks in Computing, Chapman & Hall/CRC, Boca Raton, FL, USA, 2009.

[104] K. Hose and R. Schenkel, Towards Benefit-Based RDF Source Selection for SPARQL Queries, in *Proceedings of the 4th International Workshop on Semantic Web Information Management (SWIM '12)*, 2012.

[105] W. Hu and Y. Qu, Discovering Simple Mappings between Relational Database Schemas and Ontologies, in K. Aberer, K.-S. Choi, N. Noy, D. Allemang, K.-I. Lee, L. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, P. Cudré-Mauroux and , eds., *The Semantic Web: 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference (ISWCN 2007 + ASWC 2007)*, *Lecture Notes in Computer Science*, vol. 4825, pp. 225–238, Springer, 2007.

[106] V. Huang and M. K. Javed, Semantic Sensor Information Description and Processing, in *Proceedings of the Second International Conference on Sensor Technologies and Applications (SENSORCOMM '08)*, pp. 456–461, IEEE, 2008.

[107] K. Janowicz, S. Schade, A. Broring, C. Kessler, C. Stasch, P. Maue and T. Diekhof, A Transparent Semantic Enablement Layer for the Geospatial Web, in *Terra Cognita 2009 Workshop, in Conjunction with the 8th International Semantic Web Conference (ISWC 2009)*, 2009.

[108] P. Johannesson, A Method for Transforming Relational Schemas into Conceptual Schemas, in *Proceedings of the 10th International Conference on Data Engineering (ICDE 1994)*, pp. 190–201, IEEE, 1994.

[109] D. Jurić, M. Banek and Z. Skočir, Uncovering the Deep Web: Transferring Relational Database Content and Metadata to OWL Ontologies, in I. Lovrek, R. J. Howlett and L. C. Jain, eds., *Knowledge-Based Intelligent Information and Engineering Systems: 12th International Conference (KES 2008)*, *Lecture Notes in Computer Science*, vol. 5177, pp. 456–463, Springer, 2008.

[110] Y. Kalfoglou and M. Schorlemmer, Ontology Mapping: the State of the Art, *The Knowledge Engineering Review*, **18**(1), pp. 1–31, 2003.

[111] V. Kashyap, Design and Creation of Ontologies for Environmental Information Retrieval, in *First Agricultural Service Ontology (AOS) Workshop*, 2001.

[112] S. Kiminki, J. Knuuttila and V. Hirvisalo, SPARQL to SQL Translation Based on an Intermediate Query Language, in *Proceedings of the 6th International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS2010)*, pp. 32–47, 2010.

[113] N. Konstantinou, E. Solidakis, A. Zafeiropoulos, P. Stathopoulos and N. Mitrou, A Context-Aware Middleware for Real-Time Semantic Enrichment of Distributed Multimedia Metadata, *Multimedia Tools and Applications*, **46**(2-3), pp. 425–461, 2010.

[114] N. Konstantinou, D.-E. Spanos, M. Chalas, E. Solidakis and N. Mitrou, VisAVis: An Approach to an Intermediate Layer between Ontologies and Relational Database Contents, in F. Frasincar, G.-J. Houben and P. Thiran, eds., *Proceedings of the CAiSE'06 3rd International Workshop on Web Information Systems Modeling (WISM'06)*, pp. 1050–1061, 2006.

[115] N. Konstantinou, D.-E. Spanos and N. Mitrou, Ontology and Database Mapping: A Survey of Current Implementations and Future Directions, *Journal of Web Engineering*, **7**(1), pp. 1–24, 2008.

[116] N. Konstantinou, D.-E. Spanos, P. Stavrou and N. Mitrou, Technically Approaching the Semantic Web Bottleneck, *International Journal of Web Engineering and Technology*, **6**(1), pp. 83–111, 2010.

[117] M. Korotkiy and J. L. Top, From Relational Data to RDFS Models, in N. Koch, P. Fraternali and M. Wirsing, eds., *Web Engineering: 4th International Conference (ICWE 2004)*, *Lecture Notes in Computer Science*, vol. 3140, pp. 430–434, Springer, 2004.

[118] A. Kupfer, S. Eckstein, K. Neumann and B. Mathiak, Handling Changes of Database Schemas and Corresponding Ontologies, in J. F. Roddick, V. R. Benjamins, S. S.-S. Cherfi, R. Chiang, C. Claramunt, R. A. Elmasri, F. Grandi, H. Han, M. Hepp, M. D. Lytras, V. D. Misic, G. Poels, I.-Y. Song, J. Trujillo and C. Vangenot, eds., *Advances in Conceptual Modeling - Theory and Practice: ER 2006 Workshops*, *Lecture Notes in Computer Science*, vol. 4231, pp. 227–236, Springer, 2006.

[119] N. Lammari, I. Comyn-Wattiau and J. Akoka, Extracting Generalization Hierarchies from Relational Databases: A Reverse Engineering Approach, *Data & Knowledge Engineering*, **63**(2), pp. 568–589, 2007.

[120] G. Lausen, Relational Databases in RDF: Keys and Foreign Keys, in V. Christophides, M. Collard and C. Gutierrez, eds., *Semantic Web, Ontologies and Databases: VLDB Workshop (SWDB-ODBIS 2007)*, *Lecture Notes in Computer Science*, vol. 5005, pp. 43–56, Springer, 2007.

[121] G. Lausen, M. Meier and M. Schmidt, SPARQLing Constraints for RDF, in A. Kemper, P. Valduriez, N. Mouaddib, J. Teubner, M. Bouzeghoub, V. Markl, L. Amsaleg and I. Manolescu, eds., *Advances in Database Technology: Proceedings of the 11th International conference on Extending Database Technology (EDBT '08)*, pp. 499–509, ACM, 2008.

[122] D. Le-Phuoc, M. Dao-Tran, J. X. Parreira and M. Hauswirth, A Native and Adaptive Approach for Unified Processing of Linked Streams and Linked Data, in *Proceedings of the 10th International Semantic Web Conference (ISWC 2011)*, pp. 370–388, Springer, 2011.

[123] D. Le-Phuoc, H. Q. Nguyen-Mau, J. X. Parreira and M. Hauswirth, A Middleware Framework for Scalable Management of Linked Streams, *Web Semantics: Science, Services and Agents on the World Wide Web*, **16**, pp. 42–51, 2012.

[124] M. Leggieri, A. Passant and M. Hauswirth, inContext-Sensing: LOD Augmented Sensor Data, in *Proceedings of the 10th International Semantic Web Conference (ISWC 2011)*, 2011.

[125] M. Lenzerini, Data Integration: A Theoretical Perspective, in *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pp. 233–246, 2002.

[126] D. Levshin, Mapping Relational Databases to the Semantic Web with Original Meaning, in D. Karagiannis and Z. Jin, eds., *Knowledge Science, Engineering and Management: Third International Conference (KSEM 2009)*, *Lecture Notes in Computer Science*, vol. 5914, pp. 5–16, Springer, 2009.

[127] M. Lewis, D. Cameron, S. Xie and I. B. Arpinar, ES3N: A Semantic Approach to Data Management in Sensor Networks, in *Proceedings of the Semantic Sensor Networks Workshop (SSN06), co-located with the 5th International Semantic Web Conference (ISWC 2006)*, 2006.

[128] L. Li and K. Taylor, A Framework for Semantic Sensor Network Services, in *Proceedings of the 6th International Conference on Service-Oriented Computing (ICSOC 2008)*, pp. 347–361, Springer, 2008.

[129] M. Li, X. Du and S. Wang, A Semi-Automatic Ontology Acquisition Method for the Semantic Web, in W. Fan, Z. Wu and J. Yang, eds., *Advances in Web-Age Information Management: 6th International Conference (WAIM 2005)*, *Lecture Notes in Computer Science*, vol. 3739, pp. 209–220, Springer, 2005.

[130] J. Liu and F. Zhao, Towards Semantic Services for Sensor-Rich Information Systems, in *Proceedings of the 2nd International Conference on Broadband Networks, 2005 (BroadNets 2005)*, pp. 967–974, IEEE, 2005.

[131] L. Liu, C. Pu and W. Tang, Continual Queries for Internet Scale Event-Driven Information Delivery, *IEEE Transactions on Knowledge and Data Engineering*, **11**(4), pp. 610–628, 1999.

[132] J. Lu, F. Cao, L. Ma, Y. Yu and Y. Pan, An Effective SPARQL Support over Relational Databases, in V. Christophides, M. Collard and C. Gutierrez, eds., *Semantic Web, Ontologies and Databases: VLDB Workshop (SWDB-ODBIS 2007)*, *Lecture Notes in Computer Science*, vol. 5005, pp. 57–76, Springer, 2007.

[133] L. Lubyte and S. Tessaris, Automatic Extraction of Ontologies Wrapping Relational Data Sources, in S. S. Bhowmick, J. Küng and R. Wagner, eds., *Database and Expert Systems Applications: 20th International Conference (DEXA 2009)*, *Lecture Notes in Computer Science*, vol. 5690, pp. 128–142, Springer, 2009.

[134] L. Ma, C. Wang, J. Lu, F. Cao, Y. Pan and Y. Yu, Effective and Efficient Semantic Web Data Management over DB2, in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data (SIGMOD '08)*, pp. 1183–1194, ACM Press, 2008.

[135] F. Maali, R. Cyganiak and V. Peristeras, Re-using Cool URIs: Entity Reconciliation Against LOD Hubs, in *Proceedings of the 4th Linked Data on the Web Workshop (LDOW 2011)*, 2011.

[136] A. Maedche and S. Staab, Ontology Learning for the Semantic Web, *IEEE Intelligent Systems*, **16**(2), pp. 72–79, 2001.

[137] D. Maier, P. Tucker and M. Garofalakis, *Stream Data Management*, chap. Filtering, Punctuation, Windows and Synopses, pp. 35–58, Springer, 2005.

[138] P. Martin, J. R. Cordy and R. Abu-Hamdeh, Information Capacity Preserving Translations of Relational Schemas Using Structural Transformation, Tech. rep., Ontario, Canada, 1995, iSSN 0836-0227-95-392.

[139] A. Miller and D. McNeil, Revelytix RDB Mapping Language Specification, available at: http://www.knoodl.com/ui/groups/Mapping_Ontology_Community/wiki/User_Guide/media/RDB_Mapping_Specification_v0.2, Revelytix, 2010.

[140] D. Moodley and I. Simonis, A New Architecture for the Sensor Web: The SWAP Framework, in *Proceedings of the Semantic Sensor Networks Workshop (SSN06)*, *co-located with the 5th International Semantic Web Conference (ISWC 2006)*, 2006.

[141] B. Motik, On the Properties of Metamodeling in OWL, *Journal of Logic and Computation*, **17**(4), pp. 617–637, 2007.

[142] B. Motik, I. Horrocks and U. Sattler, Bridging the Gap between OWL and Relational Databases, in C. Williamson and M. E. Zurko, eds., *Proceedings of the 16th International Conference on World Wide Web (WWW 2007)*, pp. 807–816, 2007.

[143] B. Motik, P. F. Patel-Schneider and B. Parsia, OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax, available at: http://www.w3.org/TR/owl2-syntax/, October 2009.

[144] V. Mulwad, T. Finin, Z. Syed and A. Joshi, Using Linked Data to Interpret Tables, in O. Hartig, A. Harth and J. Sequeda, eds., *Proceedings of the First International Workshop on Consuming Linked Data (COLD 2010)*, 2010.

[145] I. Myroshnichenko and M. C. Murphy, Mapping ER Schemas to OWL Ontologies, in S.-C. Chen, R. Glasberg, J. Heflin, K. K. Schuler, L. Kof, I. Oliver, P. Pantel and E. Simperl, eds., *Proceedings of the 2009 IEEE International Conference on Semantic Computing (ICSC 2009)*, pp. 324–329, IEEE, 2009.

[146] L. M. Ni, Y. Zhu, J. Ma, Q. Luo, Y. Liu, S. Cheung, Q. Yang, M. Li and M.-y. Wu, Semantic Sensor Net: an Extensible Framework, *International Journal of Ad Hoc and Ubiquitous Computing*, **4**(3), pp. 157–167, 2009.

[147] C. Nyulas, M. O'Connor and S. Tu, DataMaster - a Plug-in for Importing Schemas and Data from Relational Databases into Protégé, in *10th International Protégé Conference*, 2007.

[148] K. R. Page, D. C. De Roure, K. Martinez, J. D. Sadler and O. Y. Kit, Linked Sensor Data: RESTfully Serving RDF and GML, in *Proceedings of the 2nd International Workshop on Semantic Sensor Networks (SSN 09)*, 2009.

[149] P. Papapanagiotou, P. Katsiouli, V. Tsetsos, C. Anagnostopoulos and S. Hadjiefthymiades, RONTO: Relational to Ontology Schema Matching, *AIS SIGSEMIS Bulletin*, **3**(3-4), pp. 32–36, 2006.

[150] H. Patni, C. Henson and A. Sheth, Linked Sensor Data, in *Proceedings of the 2010 International Symposium on Collaborative Technologies and Systems (CTS 2010)*, pp. 362–370, IEEE, 2010.

[151] K. Patroumpas and T. Sellis, Window Specification over Data Streams, in *Current Trends in Database Technology: Proceedings of International Conference on Semantics of a Networked World: Semantics of Sequence and Time Dependent Data (ICSNW '06)*, pp. 445–464, Springer, 2006.

[152] J. Pérez, M. Arenas and C. Gutierrez, Semantics and Complexity of SPARQL, *ACM Transactions on Database Systems*, **34**(3), 2009.

[153] C. Pérez De Laborda and S. Conrad, Relational.OWL - A Data and Schema Representation Format Based on OWL, in S. Hartmann and M. Stumptner, eds., *Proceedings of the Second Asia-Pacific Conference on Conceptual Modeling (APCCM2005)*, pp. 89–96, 2005.

[154] C. Pérez De Laborda and S. Conrad, Database to Semantic Web Mapping using RDF Query Languages, in D. W. Embley, A. Olivé and S. Ram, eds., *Conceptual Modeling - ER 2006: 25th International Conference on Conceptual Modeling*, *Lecture Notes in Computer Science*, vol. 4215, pp. 241–254, 2006.

[155] C. Pérez De Laborda, M. Zloch and S. Conrad, RDQuery - Querying Relational Databases on-the-fly with RDF-QL, *poster in the 15th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2006)*, 2006.

[156] J. Petrini and T. Risch, SWARD: Semantic Web Abridged Relational Databases, in *Proceedings of 18th International Conference on Database and Expert Systems Applications (DEXA 2007)*, pp. 455–459, IEEE, 2007.

[157] D. Pfisterer, K. Romer, D. Bimschas, O. Kleine, R. Mietz, C. Truong, H. Hasemann, A. Kroller, M. Pagel, M. Hauswirth, M. Karnstedt, M. Leggieri, A. Passant and R. Richardson, SPITFIRE: Toward a Semantic Web of Things, *IEEE Communications Magazine*, **49**(11), pp. 40–48, 2011.

[158] A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini and R. Rosati, Linking Data to Ontologies, *Journal on Data Semantics*, **10**, pp. 133–173, 2008.

[159] A. Poggi, M. Rodriguez-Muro and M. Ruzzi, Ontology-based Database Access with DIG-MASTRO and the OBDA Plugin for Protégé, in K. Clark and P. F. Patel-Schneider, eds., *Proceedings of the Fourth OWLED Workshop on OWL: Experiences and Directions*, 2008.

[160] S. Polfliet and R. Ichise, Automated Mapping Generation for Converting Databases into Linked Data, in A. Polleres and H. Chen, eds., *Proceedings of the ISWC 2010 Posters & Demonstrations Track: Collected Abstracts*, pp. 173–176, 2010.

[161] W. J. Premerlani and M. R. Blaha, An Approach for Reverse Engineering of Relational Databases, *Communications of the ACM*, **37**(5), pp. 42–49, 1994.

[162] E. Prud'hommeaux and A. Bertails, A Mapping of SPARQL Onto Conventional SQL, W3C paper, available at: http://www.w3.org/2008/07/MappingRules/StemMapping, 2008.

[163] R. Ramakrishnan and J. Gehrke, *Database Management Systems*, McGraw-Hill, New York City, NY, 3rd ed., 2002.

[164] S. Ramanathan and J. Hodges, Extraction of Object-Oriented Structures from Existing Relational Databases, *ACM SIGMOD Record*, **26**(1), pp. 59–64, 1997.

[165] S. Ramanujam, V. Khadilkar, L. Khan, M. Kantarcioglu, B. Thuraisingham and S. Seida, Update-Enabled Triplification of Relational Data into Virtual RDF Stores, *International Journal of Semantic Computing*, **4**(4), pp. 423–451, 2010.

[166] A. Rodriguez, R. McGrath, Y. Liu and J. Myers, Semantic Management of Streaming Data, in *Proceedings of the 2nd International Workshop on Semantic Sensor Networks (SSN09), collocated with the 8th International Semantic Web Conference (ISWC 2009)*, pp. 80–95, 2009.

[167] S. Sahoo, W. Halb, S. Hellmann, K. Idehen, T. Thibodeau, S. Auer, J. Sequeda and A. Ezzat, A Survey of Current Approaches for Mapping of Relational Databases to RDF, *W3C RDB2RDF Incubator Group Report*, available at: http://www.w3.org/2005/Incubator/rdb2rdf/RDB2RDF_SurveyReport.pdf, 2009.

[168] P. E. Salas, K. K. Breitman, J. F. Viterbo and M. A. Casanova, Interoperability by Design using the StdTrip Tool: an a priori Approach, in A. Paschke, N. Henze and T. Pellegrini, eds., *Proceedings of the 6th International Conference on Semantic Systems (I-SEMANTICS 2010)*, ACM, 2010.

[169] M. Schmidt, T. Hornung, G. Lausen and C. Pinkel, SP^{2}Bench: A SPARQL Performance Benchmark, in J. Li and P. S. Yu, eds., *Proceedings of the 25th International Conference on Data Engineering (ICDE 2009)*, pp. 222–233, IEEE, 2008.

[170] M. Schneider and G. Sutcliffe, Reasoning in the OWL 2 Full Ontology Language using First-Order Automated Theorem Proving, in N. Bjørner and V. Sofronie-Stokkermans, eds., *Automated Deduction - CADE-23: 23rd International Conference on Automated Deduction*, *Lecture Notes in Computer Science*, vol. 6803, pp. 461–475, Springer, 2011.

[171] A. Seaborne, D. Steer and S. Williams, SQL-RDF, in *W3C Workshop on RDF Access to Relational Databases*, 2007.

[172] J. Sequeda and D. Miranker, SPARQL Execution as Fast as SQL Execution on Relational Data, in *Proceedings of Posters and Demos at the 10th International Semantic Web Conference (ISWC 2011)*, 2011.

[173] J. F. Sequeda and D. P. Miranker, Ultrawrap: SPARQL Execution on Relational Data, Tech. Rep. TR-2078, University of Texas, Austin, Department of Computer Science, 2012.

[174] J. F. Sequeda, S. H. Tirmizi, O. Corcho and D. P. Miranker, Direct Mapping SQL Databases to the Semantic Web: A Survey, Tech. Rep. TR-09-04, University of Texas, Austin, Department of Computer Science, 2009.

[175] G. Shen, Z. Huang, X. Zhu and X. Zhao, Research on the Rules of Mapping from Relational Model to OWL, in B. Cuenca-Grau, P. Hitzler, C. Shankey and E. Wallace, eds., *Proceedings of the OWLED'06 Workshop on OWL: Experiences and Directions*, 2006.

[176] A. Sheth, C. Henson and S. S. Sahoo, Semantic Sensor Web, *IEEE Internet Computing*, **12**(4), pp. 78–83, Jul. 2008.

[177] A. Sheth and R. Meersman, Amicalola Report: Database and Information Systems Research Challenges and Opportunities in Semantic Web and Enterprises, *ACM SIGMOD Record*, **31**(4), pp. 98–106, 2002.

[178] A. P. Sheth, Changing Focus on Interoperability in Information Systems: from System, Syntax, Structure to Semantics, in M. Goodchild, M. Egenhofer, R. Fegeas and C. Kottman, eds., *Interoperating Geographic Information Systems*, Kluwer Academic Publishers, 1999.

[179] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur and Y. Katz, Pellet: A Practical OWL-DL Reasoner, *Web Semantics: Science, Services and Agents on the World Wide Web*, **5**(2), pp. 51–53, 2007.

[180] K. Sonia and S. Khan, R2O Transformation System: Relation to Ontology Transformation for Scalable Data Integration, in J. Bernardino and B. C. Desai, eds., *Proceedings of the 2008 International Database Engineering & Applications Symposium (IDEAS '08)*, pp. 291–295, ACM, 2008.

[181] D.-E. Spanos, P. Stavrou, N. Konstantinou and N. Mitrou, SensorStream: A Semantic Real-Time Stream Management System, *International Journal of Ad Hoc and Ubiquitous Computing*, **11**(2/3), pp. 178–193, 2012.

[182] D.-E. Spanos, P. Stavrou and N. Mitrou, Bringing Relational Databases into the Semantic Web: A Survey, *Semantic Web Journal*, **3**(2), pp. 169–209, 2012.

[183] P. Stickler, CBD - Concise Bounded Description, available at: http://www.w3.org/Submission/CBD/, June 2005, W3C Member Submission.

[184] L. Stojanovic, N. Stojanovic and R. Volz, Migrating Data-Intensive Web Sites into the Semantic Web, in G. B. Lamont, ed., *Proceedings of the 2002 ACM Symposium on Applied Computing (SAC'02)*, pp. 1100–1107, ACM, 2002.

[185] M. Svihla and I. Jelinek, Two Layer Mapping from Database to RDF, in *Proceedinfs of the Sixth International Scientific Conference Electronic Computers and Informatics (ECI 2004)*, pp. 270–275, 2004.

[186] M. Svihla and I. Jelinek, Benchmarking RDF Production Tools, in R. Wagner, N. Revell and G. Pernul, eds., *Database and Expert Systems Applications: 18th International Conference(DEXA 2007)*, *Lecture Notes in Computer Science*, vol. 4653, pp. 700–709, Springer, 2007.

[187] V. Tannen, V. Christophides, G. Karvounarakis, I. Koffina, G. Kokkinidis, A. Magkanaraki, D. Plexousakis and G. Serfiotis, The ICS-FORTH SWIM: A Powerful Semantic Web Integration Middleware, in *Proceedings of the First International Workshop on Semantic Web and Databases (SWDB 2003)*, pp. 381–393, 2003.

[188] S. H. Tirmizi, J. F. Sequeda and D. P. Miranker, Translating SQL Applications to the Semantic Web, in S. S. Bhowmick, J. Küng and R. Wagner, eds., *Database and Expert Systems Applications: 19th International Conference (DEXA 2008)*, *Lecture Notes in Computer Science*, vol. 5181, pp. 450–464, Springer, 2008.

[189] Q. Trinh, K. Barker and R. Alhajj, RDB2ONT: A Tool for Generating OWL Ontologies From Relational Database Systems, in T. Atmaca, P. Dini, P. Lorenz and J. Neuman de Sousa, eds., *Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services (AICT-ICIW'06)*, IEEE, 2006.

[190] P. Tsialiamanis, L. Sidirourgos, I. Fundulaki, V. Christophides and P. Boncz, Heuristics-based Query Optimisation for SPARQL, in *Proceedings of the 15th International Conference on Extending Database Technology (EDBT '12)*, pp. 324–335, 2012.

[191] S. R. Upadhyaya and P. S. Kumar, ERONTO: a Tool for Extracting Ontologies from Extended E/R Diagrams, in L. M. Liebrock, ed., *Proceedings of the 2005 ACM Symposium on Applied Computing (SAC'05)*, pp. 666 – 670, ACM, 2005.

[192] M. Y. Vardi, The Complexity of Relational Query Languages, in *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing (STOC'82)*, pp. 137–146, ACM, 1982.

[193] K. N. Vavliakis, T. K. Grollios and P. A. Mitkas, RDOTE - Transforming Relational Databases into Semantic Web Data, in A. Polleres and H. Chen, eds., *Proceedings of the ISWC 2010 Posters & Demonstrations Track: Collected Abstracts*, pp. 121–124, 2010.

[194] Y. Velegrakis, R. J. Miller and L. Popa, Preserving Mapping Consistency under Schema Changes, *The VLDB Journal*, **13**(3), pp. 274–293, 2004.

[195] M.-E. Vidal, E. Ruckhaus, T. Lampo, A. Martínez, J. Sierra and A. Polleres, Efficiently Joining Group Patterns in SPARQL Queries, in *The Semantic Web: Research and Applications*, *Proceedings of the 7th Extended Semantic Web Conference (ESWC 2010)*, pp. 228–242, 2010.

[196] J. Volz, C. Bizer, M. Gaedke and G. Kobilarov, Discovering and Maintaining Links on the Web of Data, in A. Bernstein, D. R. Karger, T. Heath, L. Feigenbaum, D. Maynard, E. Motta and K. Thirunarayan, eds., *The Semantic Web - ISWC 2009: 8th International Semantic Web Conference*, *Lecture Notes in Computer Science*, vol. 5823, pp. 650–665, Springer, 2009.

[197] R. Volz, S. Handschuh, S. Staab, L. Stojanovic and N. Stojanovic, Unveiling the Hidden Bride: Deep Annotation for Mapping and Migrating Legacy Data to the

Semantic Web, *Web Semantics: Science, Services and Agents on the World Wide Web*, **1**(2), pp. 187–206, 2004.

[198] H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann and S. Hübner, Ontology-Based Integration of Information - A Survey of Existing Approaches, in A. Gómez-Pérez, M. Gruninger, H. Stuckenschmidt and M. Uschold, eds., *Proceedings of the IJCAI-01 Workshop on Ontologies and Information Sharing*, pp. 108–117, 2001.

[199] O. Walavalkar, A. Joshi, T. Finin and Y. Yesha, Streaming Knowledge Bases, in *Proceedings of the Fourth International Workshop on Scalable Semantic Web Knowledge Base Systems*, 2008.

[200] J. P. Walters, Z. Liang, W. Shi and V. Chaudhary, Wireless Sensor Network Security: A Survey, in Y. Xiao, ed., *Security in Distirbuted, Grid, Mobile and Pervasive Computing*, chap. 17, CRC Press, 2007.

[201] K. Whitehouse, F. Zhao and J. Liu, Semantic Streams: A Framework for Composable Semantic Interpretation of Sensor Data, in *Proceedings of the Third European Conference on Wireless Sensor Networks (ESWN '06)*, pp. 5–20, 2006.

[202] Z. Xu, X. Cao, Y. Dong and W. Su, Formal Approach and Automated Tool for Translating ER Schemata into OWL Ontologies, in H. Dai, R. Srikant and C. Zhang, eds., *Advances in Knowledge Discovery and Data Mining: 8th Pacific-Asia Conference (PAKDD 2004)*, *Lecture Notes in Computer Science*, vol. 3056, pp. 464–475, Springer, 2004.

[203] Z. Xu, S. Zhang and Y. Dong, Mapping between Relational Database Schema and OWL Ontology for Deep Annotation, in T. Nishida, Z. Shi, U. Visser, X. Wu, J. Liu, B. Wah, W. Cheung and Y.-M. Cheung, eds., *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 548–552, IEEE, 2006.

[204] C. Yu and L. Popa, Semantic Adaptation of Schema Mappings when Schemas Evolve, in *Proceedings of the 31st International Conference on Very Large Data Bases*, VLDB '05, pp. 1006–1017, VLDB Endowment, 2005.

[205] A. Zafeiropoulos, N. Konstantinou, S. Arkoulis, D.-E. Spanos and N. Mitrou, A Semantic-based Architecture for Sensor Data Fusion, in *Proceedings of the Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM'08)*, pp. 116–121, 2008.

[206] A. Zafeiropoulos, D.-E. Spanos, S. Arkoulis, N. Konstantinou and N. Mitrou, Data Management in Sensor Networks using Semantic Web Technologies, in H. Jin, ed., *Data Management in the Semantic Web*, chap. 11, p. 436, Nova Publishers, 2011.

[207] S. Zhao and E. Chang, From Database to Semantic Web Ontology: An Overview, in R. Meersman, Z. Tari and P. Herrero, eds., *On the Move to Meaningful Internet Systems: OTM 2007 Workshops*, *Lecture Notes in Computer Science*, vol. 4806, pp. 1205–1214, Springer, 2007.

[208] C. Zhou, C. Xu, H. Chen and K. Idehen, Browser-based Semantic Mapping Tool for Linked Data in Semantic Web, in C. Bizer, T. Heath, K. Idehen and T. Berners-Lee, eds., *Proceedings of the WWW 2008 Workshop on Linked Data on the Web (LDOW 2008)*, 2008.

Κατάλογος δημοσιεύσεων

Περιοδικά

1. N. Konstantinou, D.-E. Spanos, N. Houssos, N. Mitrou, "Exposing Scholarly Information as Linked Open Data: RDFizing DSpace contents", *The Electronic Library*, **32**(6) 2013 (to appear).

2. D.-E. Spanos, P. Stavrou, N. Konstantinou, N. Mitrou, "SensorStream: A Semantic Real-Time Stream Management System", *International Journal of Ad Hoc and Ubiquitous Computing*, **11**(2/3), pp. 178-193, 2012.

3. D.-E. Spanos, P. Stavrou, N. Mitrou, "Bringing Relational Databases into the Semantic Web: A Survey", *Semantic Web*, **3**(2), pp. 169-209, 2012. *(7 ετεροαναφορές)*

4. N. Konstantinou, D.-E. Spanos, P. Stavrou and N. Mitrou, "Technically Approaching the Semantic Web Bottleneck", *International Journal of Web Engineering and Technology*, **6**(1), pp. 83-111, 2010. *(3 ετεροαναφορές)*

5. S. Arkoulis, D.-E. Spanos, S. Barbounakis, A. Zafeiropoulos and N. Mitrou, "Cognitive Radio-Aided Wireless Sensor Networks for Emergency Response", *Measurement Science and Technology*, **21**(12), 2010. *(2 ετεροαναφορές)*

6. N. Konstantinou, D.-E. Spanos, N. Mitrou, "Ontology and Database Mapping: A Survey of Current Implementations and Future Directions", *Journal of Web Engineering*, **7**(1), pp. 1-24, 2008. *(25 ετεροαναφορές)*

Κεφάλαια σε Βιβλία

1. A. Zafeiropoulos, D.-E. Spanos, S. Arkoulis, N. Konstantinou, N. Mitrou, "Data Management in Sensor Networks Using Semantic Web Technologies", in H. Jin, ed., *Data Management in the Semantic Web*, ch. 11, Nova Publishers, Hauppauge, New York, p. 436, 2011. *(5 ετεροαναφορές)*

Συνέδρια

1. I. Papaioannou, P. Stavrou, A. Zafeiropoulos, D.-E. Spanos, S. Arkoulis, N. Mitrou, "Mechanisms for Distributed Data Fusion and Reasoning in Wireless Sensor Networks", Proceedings of the 17th International Workshop on Energy-Aware Communications (EUNICE 2011), pp. 221-224, 2011.

2. A. Zafeiropoulos, N. Konstantinou, S. Arkoulis, D.-E. Spanos, N. Mitrou, "A Semantic-based Architecture for Sensor Data Fusion", Proceedings of the Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM '08), pp.116-121, 2008. *(5 ετεροαναφορές)*

3. N. Konstantinou, D.-E. Spanos, M. Chalas, E. Solidakis, N. Mitrou, "VisAVis: An Approach to an Intermediate Layer between Ontologies and Relational Database Contents", Proceedings of the CAiSE'06 Third International Workshop on Web Information Systems Modeling (WISM '06), 2006. *(34 ετεροαναφορές)*