ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

RVL: ΜΙΑ ΓΛΩΣΣΑ ΟΡΙΣΜΟΥ ΟΨΕΩΝ ΓΙΑ ΤΟ ΠΡΟΤΥΠΟ RDF/S

Αιμιλία Μ. Μαγκαναράκη

Μεταπτυχιακή Εργασία

Ηράκλειο, Νοέμβριος 2002

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

RVL: ΜΙΑ ΓΛΩΣΣΑ ΟΡΙΣΜΟΥ ΟΨΕΩΝ ΓΙΑ ΤΟ ΠΡΟΤΥΠΟ RDF/S

Εργασία που υποβλήθηκε από την

Αιμιλία Μ. Μαγκαναράκη

ως μερική εκπλήρωση των απαιτήσεων για την απόκτηση

ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΔΙΠΛΩΜΑΤΟΣ ΕΙΔΙΚΕΥΣΗΣ

Συγγραφέας:

_____

Μαγκαναράκη Αιμιλία

Τμήμα Επιστήμης Υπολογιστών

Εισηγητική Επιτροπή:

_____

Δημήτρης Πλεξουσάκης, Αναπληρωτής Καθηγητής, Επόπτης

_____

Πάνος Κωνσταντόπουλος, Καθηγητής, Μέλος

_____

Βασίλης Χριστοφίδης, Επίκουρος Καθηγητής, Μέλος, Επιβλέπων

Δεκτή:

_____

Πάνος Κωνσταντόπουλος, Καθηγητής,
Πρόεδρος Επιτροπής Μεταπτυχιακών Σπουδών

Ηράκλειο, Νοέμβριος 2002

RVL: ΜΙΑ ΓΛΩΣΣΑ ΟΡΙΣΜΟΥ ΟΨΕΩΝ ΓΙΑ ΤΟ ΠΡΟΤΥΠΟ RDF/S

Αιμιλία Μ. Μαγκαναράκη

Μεταπτυχιακή Εργασία

Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης

Περίληψη

Η ικανότητα μίας γλώσσας επερώτησης να υποστηρίζει μηχανισμούς για τον ορισμό όψεων πάνω σε σύνολα δεδομένων έχει αναγνωριστεί ως μία επιπρόσθετης αξίας λειτουργικότητα στην εκφραστική δύναμη μίας γλώσσας επερώτησης σε παραδοσιακά συστήματα βάσεων δεδομένων, όπως σχεσιακά και οντοκεντρικά, αλλά και σε σύγχρονες εφαρμογές βάσεων δεδομένων, που στηρίζονται στο ημιδομημένων δεδομένων και XML μοντέλο δεδομένων. Ο μηχανισμός ορισμού όψεων επιτρέπει την οργάνωση (μέρους) μίας βάσης δεδομένων σε λογικές μονάδες, δομώντας με αυτό τον τρόπο (τμήματα) της βάσης δεδομένων με τρόπο κατάλληλο για τις απαιτήσεις συγκεκριμένων εφαρμογών. Στα πλαίσια του Σημασιολογικού Ιστού, οι καταναλωτές του ιδίου συνόλου πληροφορίας θέτουν διαφορετικές απαιτήσεις στον τρόπο με τον οποίο αντιλαμβάνονται την πληροφορία και για το λόγο αυτό, απαιτούν την ύπαρξη μηχανισμών, όπως οι όψεις, για εξατομικευμένη πρόσβαση στις πληροφοριακές πηγές. Η ανάγκη αυτή γίνεται πιο επιτακτική, αν λάβουμε υπόψη μας την ποικιλομορφία των αναπαραστάσεων που χρησιμοποιούν οι καταναλωτές και οι προμηθευτές πληροφορίας για το φορμαλισμό, ακόμα και του ίδιου, πεδίου εφαρμογής.

Αναγνωρίζοντας την ανάγκη υποστήριξης ενός μηχανισμού ορισμού όψεων για το Σημασιολογικό Ιστό, η παρούσα εργασία προδιαγράφει μία γλώσσα ορισμού όψεων για την υποστήριξη της απαιτούμενης

λειτουργικότητας από εφαρμογές του Σημασιολογικού Ιστού. Βασιζόμενη στο μοντέλο δεδομένων του Πλαισίου Περιγραφής Πόρων (RDF/S) και εκμεταλλευόμενη την εκφραστική δύναμη της γλώσσας επερώτησης RDF/S γράφων RQL, η προτεινόμενη γλώσσα ορισμού όψεων, την οποία αποκαλούμε RVL, ενσωματώνει με ένα ομοιόμορφο τρόπο την απαιτούμενη λειτουργικότητα και τις ιδιαιτερότητες του υποκείμενου μοντέλου δεδομένων. Όντας η πρώτη ολοκληρωμένη προσπάθεια προδιαγραφής μίας γλώσσας ορισμού όψεων για το RDF/S, η RVL εκμεταλλεύεται το σύστημα τύπων της RQL και τα επίπεδα αφαίρεσης σε ένα RDF/S γράφο για να προδιαγράψει δύο τελεστές, που είναι σε θέση να υποστηρίξουν όλη την απαιτούμενη λειτουργικότητα. Η απλότητα αυτή αποτελεί και το σημαντικότερο πλεονέκτημα της RVL.

Επόπτης: Δημήτρης Πλεξουσάκης,

Επίκουρος Καθηγητής

RVL: A VIEW DEFINITION LANGUAGE FOR RDF/S

Aimilia M. Magkanaraki

Master's Thesis

Computer Science Department, University of Crete

Abstract

The ability of a query language to support mechanisms for defining views over a data set has been recognized as an added-value functionality to the expressive power of a query language in traditional database systems, such as relational and object-oriented systems, but also in modern database applications based on the semistructured and XML data models. The view definition mechanism enables the organization of (part of) a database into logic units, thus structuring (portions of) the database in a way suitable to the needs of specific applications. In the context of the Semantic Web, information consumers of the same information set, pose different requirements to the way they view information, hence necessitating the existence of mechanisms, such as views, to provide tailored access to the information sources. This need becomes more imperative, if we take into account the diversity of the representations used by information consumers and providers for formalizing, even the same, domain of discourse.

Recognizing the need of a view definition mechanism for the Semantic Web, the present thesis specifies a view definition language for supporting the functionality needed by semantic web applications. Based on the data model of the Resource Description Framework (RDF/S) and by taking advantage of the expressiveness of the RQL query language for RDF/S

graphs, the proposed view definition language, called RVL, incorporates in a uniform way the functionality needed and the peculiarities of the underlying data model. Being the first integrated effort for a view definition language specification, RVL exploits the RQL type system and the abstraction levels of an RDF/S graph to specify two operators, which are able to support all the necessary functionality. This minimality constitutes the most important advantage of RVL.

Supervisor: Dimitris Plexousakis,

Assistant Professor

Αφιερωμένο σε εκείνους που πίστεψαν σε εμένα

και πρωτίστως στους γονείς μου και τα αδέλφια μου,

Ελένη, Μαρία και Γιώργο.

Ευχαριστίες

Στο σημείο αυτό θα ήθελα να ευχαριστήσω όλους όσους, εμμέσως ή άμεσα, συνετέλεσαν στην ολοκλήρωση αυτής της εργασίας. Ιδιαίτερα θερμές ευχαριστίες αξίζουν αφενός στον επόπτη καθηγητή μου, κ. *Δημήτρη Πλεξουσάκη* και αφετέρου στον επιβλέποντα καθηγητή της εργασίας μου κ. *Βασίλη Χριστοφίδη*. Ο πρώτος ήταν πάντα πρόθυμος να διαβάσει ότι είχα να προτείνω (ακόμα και στο αεροπλάνο) προκειμένου να μου δώσει πολύτιμες συμβουλές. Ο δεύτερος ήταν πάντα εκεί για να μου δώσει μία ιδέα, μία διέξοδο (ή ένα ακόμα paper). Χωρίς την πολύτιμη βοήθεια, καθοδήγηση, συμπαράσταση και συνεργασία τους, η ολοκλήρωση αυτής της εργασίας θα ήταν αδύνατη.

Επίσης, θα ήθελα να ευχαριστήσω τα μέλη της ομάδας Πληροφοριακών Συστημάτων και Τεχνολογίας Λογισμικού του Ινστιτούτου Πληροφορικής, και ιδιαίτερα τον επικεφαλή της ομάδας, καθηγητή κ. *Πάνο Κωνσταντόπουλο*, που δέχτηκε τόσο πρόθυμα να είναι μέλος της εισηγητικής επιτροπής, αλλά και για την καθοδήγησή του.

Ένα πολύ μεγάλο ευχαριστώ ανήκει στους φίλους μου για την αμέριστη συμπαράσταση, την κατανόηση και για ότι ήταν πάντα εκεί, όταν τους χρειαζόμουν. Από φόβο μήπως ξεχάσω κάποιον, δεν επιχειρώ να τους αναφέρω ονομαστικά. Αυτό όμως δεν μειώνει τη βοήθεια και την ευγνωμοσύνη που τους χρωστώ.

Τέλος, αλλά πρωτίστως, ευχαριστώ τους γονείς μου *Μιχάλη* και *Νίκη* και τα αδέλφια μου *Ελένη*, *Μαρία* και *Γιώργο*, που με ανέχονται και με υποστηρίζουν αγόγγυστα τόσα χρόνια και για την αγάπη που τόσο απλόχερα μου προσφέρουν.

Περιεχόμενα

ΚΕΦΑΛΑΙΟ 1.....*1*

ΕΙΣΑΓΩΓΗ.....*1*

1.1 Ο Σημασιολογικός Ιστός 1

1.2 Ο μηχανισμός ορισμού όψεων 2

1.3 Αντικείμενο της εργασίας..... 3

1.4 Οργάνωση της εργασίας 4

ΚΕΦΑΛΑΙΟ 2.....*5*

ΒΑΣΙΚΗ ΤΕΧΝΟΛΟΓΙΑ ΣΗΜΑΣΙΟΛΟΓΙΚΟΥ ΙΣΤΟΥ.....*5*

2.1 Το πρότυπο RDF/S 6

2.1.1 Το μοντέλο δεδομένων του RDF/S..... 7

2.1.2 Οι τυπικές RDF/S κλάσεις και ιδιότητες..... 10

2.1.3 Οι εννοιολογικοί μηχανισμοί του RDF/S 13

2.2 Περιοχές εφαρμογής του RDF/S 15

2.3 Γλώσσες επερώτησης για RDF/S μεταδεδομένα..... 20

RQL.....26

RDQL.....26

SquishQL.....26

RDFQL.....26

RDFPath.....26

VERSA.....26

TRIPLE.....26

2.3.1 Σύγκριση των γλωσσών επερώτησης 26

ΚΕΦΑΛΑΙΟ 3.....*33*

Ο ΜΗΧΑΝΙΣΜΟΣ ΟΨΕΩΝ ΣΕ ΣΥΣΤΗΜΑΤΑ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ.....*33*

3.1 Η τεχνολογία των όψεων..... 34

3.1.1 Βασικές έννοιες και χρησιμότητα των όψεων..... 34

3.1.1.1 Σχεσιακό Μοντέλο Δεδομένων 36
3.1.1.2 Οντοκεντρικό Μοντέλο Δεδομένων 42
3.1.1.3 Μοντέλο Ημιδομημένων Δεδομένων και XML..... 55
3.1.2 Διαχείριση όψεων..... 63
3.2 Όψεις για το Σημασιολογικό Ιστό 66

ΚΕΦΑΛΑΙΟ 4.....71
Ο ΜΗΧΑΝΙΣΜΌΣ ΟΡΙΣΜΟΎ RDF/S ΌΨΕΩΝ.....71
4.1 Το τυπικό μοντέλο δεδομένων της RQL 72

4.1.1 Βασικές έννοιες 72
4.1.2 Το τυπικό μοντέλο δεδομένων της RQL..... 75
4.1.3 Το σύστημα τύπων του RQL μοντέλου δεδομένων..... 82
4.1.4 Διαφορές του RQL μοντέλου δεδομένων με τις RDF/S προδιαγραφές 87
4.1.5 Τυπική ερμηνεία των βασικών RQL επερωτήσεων..... 90
4.2 Το τυπικό μοντέλο των όψεων..... 93

4.2.1 Ορισμός RDF/S όψεων 93
4.2.2 Ιδεατά RDF σχήματα περιγραφών και βάσεις περιγραφών . 96
4.2.3 Επιλογές μοντελοποίησης του μηχανισμού ορισμού όψεων 100
4.2.3.1 Όψεις που διατηρούν ή/και δημιουργούν αντικείμενα..... 101
4.2.3.2 Σύνθεση όψεων 102
4.3 Η γλώσσα ορισμού RDF/S όψεων 103

4.3.1 Κατασκευή ενός ιδεατού σχήματος 104
4.3.2 Τελεστές της γλώσσας ορισμού όψεων 106
4.3.2.1 Τελεστής υποστασιοποίησης..... 108
4.3.2.1.1 Ιδεατές (Μετα)Κλάσεις 110
4.3.2.1.2 Ιδεατές Ιδιότητες 116
4.3.2.2 Τελεστής υπαλληλίας..... 120
4.3.3 Ιδιότητες της γλώσσας ορισμού όψεων RVL 127
4.3.3.1 Απλότητα 127
4.3.3.2 Γενικότητα 128
4.3.3.3 Λογική Ανεξαρτησία..... 128

4.3.3.4 Κλειστότητα..... 129

ΚΕΦΑΛΑΙΟ 5.....*131*

ΕΠΙΛΟΓΟΣ.....*131*

5.1 Μελλοντική εργασία 133

ΒΙΒΛΙΟΓΡΑΦΙΑ *135*

Κατάλογος Πινάκων

Πίνακας 2.1: Πεδία ορισμού και τιμών των βασικών RDF/S ιδιοτήτων..... 14

Πίνακας 2.2: Ενδεικτική επερώτηση RDF/S γλωσσών επερώτησης..... 26

Πίνακας 2.3: Βασικά χαρακτηριστικά RDF/S γλωσσών επερώτησης..... 27

Πίνακας 2.4: Εκφραστική δύναμη των RDF/S γλωσσών επερώτησης..... 29

Πίνακας 4.1: Τυπική ερμηνεία των βασικών RQL ερωτήσεων 90

Πίνακας 4.2: Τυπική ερμηνεία των βασικών εκφράσεων μονοπατιών της RQL . 91

Κατάλογος Εικόνων

Εικόνα 2.1: RDF/S γράφος 8

Εικόνα 2.2: RDF/S τυπικές κλάσεις και ιδιότητες 10

Εικόνα 2.3: Κατηγοριοποίηση 28 RDF/S σχημάτων ανάλογα με περιοχή εφαρμογής και σημασιολογικό βάθος 18

Εικόνα 3.1. Αρχιτεκτονική βάσεων δεδομένων ANSI/SPARC 35

Εικόνα 3.2. Δημιουργία ιδεατού σχήματος 50

Εικόνα 3.3. Περιγραφή ημιδομημένων βιβλιογραφικών δεδομένων 57

Εικόνα 3.4. Περιγραφή βιβλιογραφικών δεδομένων στην XML 58

Εικόνα 3.5: Λειτουργικότητα προς υποστήριξη 67

Εικόνα 4.1: Παράδειγμα πολιτισμικής εφαρμογής στο Σημασιολογικό Ιστό 74

Εικόνα 4.2: Υπογράφος ενός έγκυρου RDF/S γράφου σχήματος..... 80

Εικόνα 4.3: Το τυπικό RQL μοντέλο δεδομένων..... 86

Εικόνα 4.4: Παραγωγή ενός ιδεατού σχήματος περιγραφών βάσει πολλαπλών αρχικών σχημάτων περιγραφών 94

Εικόνα 4.5: Σύνθεση όψεων..... 102

Εικόνα 4.6: RVL επίπεδα αφαίρεσης..... 109

Εικόνα 4.7: Παράδειγμα χρήσης του τελεστή υπαλληλίας..... 123

Εικόνα 4.8: Παράδειγμα χρήσης του τελεστή υπαλληλίας..... 123

Εικόνα 4.9: Αναδρομική λειτουργία του τελεστή υπαλληλίας 126

ΚΕΦΑΛΑΙΟ 1

Εισαγωγή

1.1 Ο Σημασιολογικός Ιστός

Ο όρος *Σημασιολογικός Ιστός* (Semantic Web) χρησιμοποιήθηκε πρώτη φορά το 1998 από το δημιουργό του πρώτου φυλλομετρητή ιστοσελίδων και εξυπηρετητή διαδικτύου Tim Berners-Lee [BHL01]. Έκτοτε, καταβάλλεται σημαντική προσπάθεια από την επιστημονική κοινότητα για την υλοποίησή του πάνω από τον Παγκόσμιο Ιστό. Στο βασικότερο επίπεδό του, ο Σημασιολογικός Ιστός αποτελεί μία συλλογή από πληροφορία για τη διακινούμενη πληροφορία, επονομαζόμενη *μεταδεδομένα*, η οποία δεν είναι ορατή στον τελικό χρήστη. Τα μεταδεδομένα αυτά χρησιμοποιούνται για να περιγράψουν υπάρχοντα έγγραφα, ιστοσελίδες, βάσεις δεδομένων, προγράμματα κτλ που βρίσκονται στο Διαδίκτυο, έτσι ώστε οι εφαρμογές λογισμικού που κάνουν χρήση αυτών, να αποκτούν κατανόηση της σημασιολογίας του περιεχομένου τους. Απώτερος στόχος της όλης προσπάθειας είναι η ικανοποίηση των απαιτήσεων των συμμετεχόντων στην Κοινωνία της Πληροφορίας για ποιότητα των υπηρεσιών, η οποία συνίσταται κυρίως στην βελτιωμένη αναζήτηση, την εκτέλεση σύνθετων διεργασιών μέσω του Διαδικτύου και στην εξατομίκευση της πληροφορίες σύμφωνα με τις ανάγκες του εκάστοτε χρήστη.

Όντας το επόμενο βήμα εξέλιξης του Παγκοσμίου Ιστού, ο Σημασιολογικός Ιστός αναμένεται να διαθέσει προς επεξεργασία στους χρήστες τεράστιες ποσότητες πληροφοριακών πόρων (π.χ δεδομένα, έγγραφα, εικόνες, προγράμματα) σε συνδυασμό με διάφορα είδη περιγραφικής πληροφορίας (μεταδεδομένα). Η καλύτερη γνώση σχετικά με τη σημασία, τη χρήση, την προσπελασιμότητα, την εγκυρότητα ή την ποιότητα των διαδικτυακών πόρων, θα διευκολύνει σε μεγάλο βαθμό την αυτοματοποιημένη επεξεργασία του διαθέσιμου περιεχομένου και υπηρεσιών στο διαδίκτυο. Στοχεύοντας προς αυτή την κατεύθυνση, το *Πλαίσιο Περιγραφής Πόρων* (Resource Description Framework) [LS99] επιτρέπει τη δημιουργία και ανταλλαγή των μεταδεδομένων πληροφοριακών πόρων ως κοινά δεδομένα διαδικτύου. Για να ερμηνεύσει αυτά τα μεταδεδομένα, εντός αλλά και

μεταξύ διαφορετικών κοινοτήτων χρηστών, το RDF προσφέρει δυνατότητες ορισμού κατάλληλων σχημάτων (λεξιλογίων) μέσω της αντίστοιχης γλώσσας ορισμού RDF σχημάτων (RDF Schema Language) [BG00]. Με τον τρόπο αυτό, διαφορετικές κοινότητες χρηστών μπορούν είτε ανεξάρτητα να αναπτύξουν το δικό τους σχήμα είτε να επαναχρησιμοποιήσουν ήδη υπάρχοντα σχήματα άλλων ομάδων (πιθανώς με διαφορετικά ενδιαφέροντα) ως έχει ή επεκτείνοντάς τα, έτσι ώστε να ικανοποιήσουν τις συγκεκριμένες ανάγκες περιγραφής των πληροφοριακών πόρων τους.

Εκτός από την υποστήριξη της ικανότητας συνδυασμού πληροφορίας από πολλαπλές πληροφοριακές πηγές, οι καταναλωτές του ιδίου συνόλου πληροφορίας στο Σημασιολογικό Ιστό θέτουν διαφορετικές απαιτήσεις στον τρόπο με τον οποίο αντιλαμβάνονται την πληροφορία. Για το λόγο αυτό, απαιτούν την ύπαρξη μηχανισμών, όπως οι *όψεις*, για εξατομικευμένη πρόσβαση στις διεσπαρμένες πληροφοριακές πηγές. Η ανάγκη αυτή γίνεται πιο επιτακτική, αν λάβουμε υπόψη μας την ποικιλομορφία των αναπαραστάσεων που χρησιμοποιούν οι καταναλωτές και οι προμηθευτές πληροφορίας για το φορμαλισμό -ακόμα και του ίδιου- πεδίου εφαρμογής, την ύπαρξη μεγάλου αριθμού διαφορετικών κοινοτήτων χρηστών και την ποικιλία των ενδιαφερόντων και των απαιτήσεών τους.

1.2 Ο μηχανισμός ορισμού όψεων

Η έννοια των όψεων απαντάται για πρώτη φορά το 1975 σε σχεσιακά συστήματα βάσεων δεδομένων και έκτοτε αποτελεί ένα σημαντικό χαρακτηριστικό των συστημάτων διαχείρισης βάσεων δεδομένων τόσο σε σχεσιακά όσο και σε οντοκεντρικά, ημιδομημένα και XML περιβάλλοντα βάσεων δεδομένων. Ο όρος *όψη* σχετίζεται άμεσα με την αρχιτεκτονική βάσεων δεδομένων τριών επιπέδων *ANSI/SPARC* [A75], στην προδιαγραφή της οποίας χρησιμοποιήθηκε για πρώτη φορά. Το μοντέλο αυτό αρχιτεκτονικής βάσεων δεδομένων διακρίνει τρία επίπεδα, που εξασφαλίζουν την ανεξαρτησία των δεδομένων σε όλα τα επίπεδα: το *εσωτερικό επίπεδο* αναπαριστά τον τρόπο αποθήκευσης των δεδομένων στο φυσικό μέσο αποθήκευσης, το *εννοιολογικό επίπεδο* αποτελεί μία λογική αναπαράσταση του πραγματικού κόσμου και αντιστοιχεί στην έννοια του *εννοιολογικού σχήματος* και το *εξωτερικό επίπεδο* αναλογεί στην έννοια των *όψεων* ή του *εξωτερικού σχήματος*.

Ο μηχανισμός ορισμού όψεων αποτελεί ουσιαστικά μία τεχνική μοντελοποίησης των δεδομένων με διαφορετικό τρόπο για ποικίλες εφαρμογές, που

δεν επηρεάζει τον τύπο και τη δομή των αποθηκευμένων δεδομένων, επιτρέποντας με αυτό τον τρόπο σε εφαρμογές να προσαρμόζουν στις απαιτήσεις τους κατανεμημένα δεδομένα, χωρίς να επηρεάζουν τη λειτουργία άλλων εφαρμογών που χρησιμοποιούν το ίδιο σύνολο πληροφορίας. Η χρησιμότητά του και το εύρος των εφαρμογών του ποικίλει από την εξατομίκευση των δεδομένων στις ανάγκες των εκάστοτε χρηστών (τελικός χρήστης, προγραμματιστής εφαρμογών, ενδιάμεσα προγράμματα εφαρμογών κτλ) μέχρι την επιβεβαίωση περιορισμών ακεραιότητας. Ενδεικτικές εφαρμογές του μηχανισμού ορισμού όψεων αποτελούν η λογική ανεξαρτησία των δεδομένων, η υλοποίηση μηχανισμών προστασίας των δεδομένων, η ολοκλήρωση ετερογενών βάσεων δεδομένων, η μεσιτεία πληροφορίας από ετερογενείς πληροφοριακές πηγές και η εξέλιξη σχήματος σε οντοκεντρικά περιβάλλοντα.

1.3 Αντικείμενο της εργασίας

Αναγνωρίζοντας τη χρησιμότητα και την ανάγκη υποστήριξης ενός μηχανισμού ορισμού όψεων για το Σημασιολογικό Ιστό, το αντικείμενο της παρούσας εργασίας είναι η προδιαγραφή μίας γλώσσας ορισμού όψεων για την υποστήριξη της απαιτούμενης λειτουργικότητας από εφαρμογές του Σημασιολογικού Ιστού. Η επιστημονική συνεισφορά της συγκεκριμένης εργασίας εντοπίζεται στα εξής κυριότερα σημεία:

- Η παρουσίαση ενός τυπικού μοντέλου όψεων, που προσφέρει μία τυπική, εννοιολογική βάση στην προδιαγραφόμενη γλώσσα ορισμού όψεων
- Η προδιαγραφή μίας εκφραστικής γλώσσας ορισμού όψεων, που αποκαλούμε *RVL* (RDF View Language). Βασιζόμενη στο μοντέλο δεδομένων του Πλαισίου Περιγραφής Πόρων (RDF/S) και εκμεταλλευόμενη την εκφραστική δύναμη της γλώσσας επερώτησης RDF/S γράφων RQL, η προτεινόμενη γλώσσα ορισμού όψεων ενσωματώνει με ένα ομοιόμορφο τρόπο την απαιτούμενη λειτουργικότητα και τις ιδιαιτερότητες του υποκείμενου μοντέλου δεδομένων. Όντας η πρώτη ολοκληρωμένη προσπάθεια προδιαγραφής μίας γλώσσας ορισμού όψεων για το RDF/S, η RVL εκμεταλλεύεται το σύστημα τύπων της RQL και τα επίπεδα αφαίρεσης σε ένα RDF/S γράφο για να προδιαγράψει δύο τελεστές, που είναι σε θέση να υποστηρίξουν όλη την απαιτούμενη λειτουργικότητα. Η απλότητα αυτή αποτελεί και το σημαντικότερο πλεονέκτημα της RVL.

1.4 Οργάνωση της εργασίας

Η οργάνωση της συγκεκριμένης μεταπτυχιακής εργασίας έχει ως εξής:

- Στο **Κεφάλαιο 2** περιγράφονται τεχνολογίες σχετιζόμενες με το πρότυπο RDF/S. Ειδικότερα, παρουσιάζονται εν συντομία τα πρότυπα RDF και RDFS, αναφέρονται περιοχές εφαρμογής τους και καταγράφονται οι απαιτήσεις που θέτουν. Επιπλέον, παρουσιάζονται γλώσσες επερώτησης RDF/S δεδομένων και μέσω σύγκρισης της εκφραστικής τους δύναμης, αιτιολογείται η απόφαση να στηριχτούμε στη γλώσσα επερώτησης RQL.
- Στο **Κεφάλαιο 3** παρουσιάζεται η θεωρητική βάση του μηχανισμού ορισμού όψεων και ο τρόπος με τον οποίο μπορεί να χρησιμοποιηθεί στο περιβάλλον του Σημασιολογικού Ιστού. Επιχειρώντας αρχικά μία ανασκόπηση στη θεωρία των όψεων, παρουσιάζονται οι βασικές έννοιες του μηχανισμού ορισμού όψεων σε σχεσιακά, οντοκεντρικά, ημιδομημένα και XML περιβάλλοντα βάσεων δεδομένων. Εν συνεχεία, σκιαγραφείται η λειτουργικότητα που πρέπει να υποστηρίζει ένας μηχανισμός ορισμού όψεων στα πλαίσια του Σημασιολογικού Ιστού.
- Στο **Κεφάλαιο 4** γίνεται η τυπική παρουσίαση της προτεινόμενης γλώσσας ορισμού όψεων RVL. Για την πληρότητα της παρουσίασης, περιγράφεται αρχικά το τυπικό μοντέλο δεδομένων της RQL και η ερμηνεία των βασικότερων RQL επερωτήσεων. Κατόπιν, παρουσιάζουμε το τυπικό μοντέλο των RDF/S όψεων και την προδιαγραφόμενη γλώσσα ορισμού όψεων RVL.
- Το **Κεφάλαιο 5** καταλήγει τη συγκεκριμένη μεταπτυχιακή εργασία, ανακεφαλαιώνοντας την προσφορά της και αναφέροντας κατευθύνσεις για μελλοντική έρευνα.

ΚΕΦΑΛΑΙΟ 2

Βασική Τεχνολογία Σημασιολογικού Ιστού

Η χρησιμοποίηση του Παγκόσμιου Ιστού (World Wide Web) ως τεχνολογικό μέσο για την ανταλλαγή *μηχανικά αναγνώσιμης* πληροφορίας με τη μορφή εγγράφων, δεν ικανοποιεί πλέον τις διαφορετικές πληροφοριακές ανάγκες ενός σταθερά αυξανόμενου αριθμού χρηστών. Η νέα πρόκληση για το Διαδίκτυο είναι η μετατροπή του σε μία πλατφόρμα για την ανταλλαγή και επαναχρησιμοποίηση *μηχανικά κατανοήσιμης* γνώσης, δηλαδή δεδομένων και πληροφορίας των οποίων η σημασιολογία μπορεί να γίνει αντικείμενο αυτόματης επεξεργασίας από τους υπολογιστές. Ο **Σημασιολογικός Ιστός** (Semantic Web) [BHL01], ως επέκταση του τωρινού Διαδικτύου, στοχεύει στην αντιμετώπιση αυτής της πρόκλησης με την παροχή των αναγκαίων γλωσσών για την απόδοση σαφώς ορισμένης «ερμηνείας» στη διακινούμενη πληροφορία, διευκολύνοντας με αυτόν τον τρόπο την επικοινωνία τόσο μεταξύ υπολογιστών όσο και μεταξύ ανθρώπων.

Προκειμένου να τεθεί σε λειτουργία ο Σημασιολογικός Ιστός, απαιτείται η πρόσβαση σε δομημένες συλλογές πληροφορίας και σε σύνολα κανόνων εξαγωγής συμπερασμάτων που μπορούν να χρησιμοποιηθούν για να διεξάγουν αυτόματους συλλογισμούς1. Τα πρώτα βήματα για την ενσωμάτωση του Σημασιολογικού Ιστού στη δομή του υπάρχοντος Διαδικτύου βρίσκονται ήδη σε εξέλιξη και οι δύο πιο σημαντικές τεχνολογίες προς αυτή την κατεύθυνση είναι η EXtensible Markup Language (XML) [BPS+00] και το Πλαίσιο Περιγραφής Πόρων/Γλώσσα Σχήματος (Resource Description Framework/ Schema Language) ([LS99], [BG00]). Η χρήση της XML σηματοδότησε τη διάκριση της αναπαράστασης γνώσης από το περιεχόμενο ενός εγγράφου, επιτρέποντας στους χρήστες να προσθέσουν

---

[1] Αξίζει να σημειώσουμε ότι το εύρος των συλλογισμών αυτών βρίσκεται υπό αναζήτηση. Στην πλειοψηφία των σημερινών εφαρμογών του Σημασιολογικού Διαδικτύου, δηλωτικές γλώσσες επερώτησης στα δεδομένα και τη σημασιολογία τους ικανοποιούν σε σημαντικό βαθμό τις ανάγκες των εφαρμογών για αυτόματους συλλογισμούς.

(δενδρική) δομή στα έγγραφά τους με τη χρήση ετικετών που σχολιάζουν ιστοσελίδες ή τμήματα μίας ιστοσελίδας. Παρά το γεγονός ότι αρχικά απευθυνόταν στους περιορισμούς της HTML αναφορικά με την περιγραφή της δομής ενός εγγράφου, η XML έχει πλέον καταστεί αποδεκτή ως μία, ανεξάρτητη από οποιαδήποτε πλατφόρμα, σύνταξη για την μεταφορά δεδομένων μεταξύ προγραμμάτων και συστημάτων. Το RDF/S, από την άλλη πλευρά, παρέχει μία γενική «αρχιτεκτονική» για μεταδεδομένα (δεδομένα για δεδομένα), ορίζοντας ένα σύνολο από συμβάσεις που πρέπει να ακολουθούν εφαρμογές που ανταλλάσσουν μεταδεδομένα.

Σκοπός της ενότητας αυτής είναι να περιγράψει τεχνολογίες συσχετιζόμενες με το πρότυπο RDF. Ειδικότερα, θα παρουσιαστούν εν συντομία τα πρότυπα RDF και RDFS, που στο σύνολό τους θα αποκαλούμε με το ακρώνυμο RDF/S. Στη συνέχεια, θα αναφέρουμε περιοχές εφαρμογής του RDF/S και θα καταγράψουμε τις απαιτήσεις που θέτουν. Επίσης, θα παρουσιάσουμε γλώσσες επερώτησης RDF/S δεδομένων, που επιχειρούν να ικανοποιήσουν ανάγκες πραγματικών εφαρμογών και μέσω σύγκρισης της εκφραστικής τους δύναμης θα αιτιολογήσουμε την απόφασή μας να βασιστούμε στη γλώσσα επερώτησης RQL, προσφέροντας ένα μηχανισμό ορισμού όψεων προσαρμοσμένο στις ιδιαιτερότητες του Σημασιολογικού Ιστού.

2.1 Το πρότυπο RDF/S

Το *Πλαίσιο Περιγραφής Πόρων* (Resource Description Framework) [LS99] αποτελεί μέρος της δραστηριότητας για τη διαχείριση μεταδεδομένων στον Παγκόσμιο Ιστό, την οποία συντονίζει το World Wide Web Consortium (W3C) και απαρτίζει τη βάση για την κωδικοποίηση, ανταλλαγή, επεξεργασία και επαναχρησιμοποίηση μεταδεδομένων. Η βασική επιδίωξη του RDF είναι να επιτρέψει τον ορισμό της σημασιολογίας που εγκλείεται σε πληροφοριακούς πόρους με τυπικό, διαλειτουργικό και ανθρωπίνως αναγνώσιμο τρόπο μέσω ενός μηχανισμού για την περιγραφή πληροφοριακών πόρων, ο οποίος δεν κάνει καμία υπόθεση για τη φύση του συγκεκριμένου πεδίου εφαρμογής ή τη δομή του εγγράφου που περιέχει την πληροφορία. Εκτός από τη διαλειτουργικότητα μεταξύ συστημάτων, το RDF στοχεύει στην επαναχρησιμοποίηση, διαμοιρασμό και επεκτασιμότητα των ορισμών μεταδεδομένων και κατά συνέπεια, στην αυτόματη επεξεργασία πόρων που ανταλλάσσονται μέσω του Διαδικτύου.

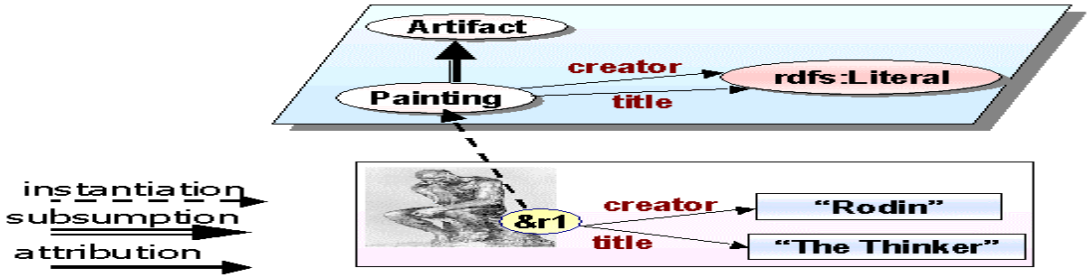
Τα δομικά στοιχεία του μοντέλου δεδομένων του Πλαισίου Περιγραφής Πόρων (RDF) [LS99] και της αντίστοιχης Γλώσσας Σχήματος (RDFS) [BG00] επιτρέπουν σε κοινότητες χρηστών που μοιράζονται πληροφορίες ενός τομέα γνώσης και ένα σύνολο από πληροφοριακούς πόρους, να ορίζουν ανεξάρτητα και χωρίς κεντρικό έλεγχο τη δική τους σημασιολογία μέσω μίας σαφώς ορισμένης, εννοιολογικής υποδομής. Αν και η σημασιολογία που αποδίδεται στους ίδιους πληροφοριακούς πόρους μπορεί να διαφέρει ακόμα και εντός της ίδιας κοινότητας χρηστών, η κοινή αυτή εννοιολογική υποδομή επιτρέπει τη συνεπή κωδικοποίηση, ανταλλαγή, και επεξεργασία των μεταδεδομένων (τροποποίηση, επερώτηση, αποθήκευση, κτλ). Η σημασιολογία κωδικοποιείται σε ένα σύνολο από τριάδες, κάθε τριάδα όντας το ανάλογο του υποκειμένου, ρήματος και αντικειμένου μίας στοιχειώδους πρότασης, η οποία μπορεί να γραφτεί με τη χρήση XML ετικετών. Η δομή αυτή αποδεικνύεται ένας φυσικός τρόπος για την περιγραφή της συντριπτικής πλειοψηφίας των πληροφοριακών πόρων που μπορούν να βρεθούν σε επιχειρησιακά περιβάλλοντα ή το Διαδίκτυο.

2.1.1 Το μοντέλο δεδομένων του RDF/S

Το μοντέλο δεδομένων του RDF βασίζεται στην έννοια του «*πόρου*» (resource). Οτιδήποτε, έννοια ή αντικείμενο, διαθέσιμο ή όχι στο Διαδίκτυο, μπορεί να μοντελοποιηθεί ως ένας πόρος αναγνωρίσιμος μοναδικά από ένα Universal Resource Identifier (URI) [BFM98]. Με τα δομικά στοιχεία του RDF μοντέλου δεδομένων μπορούμε είτε να περιγράψουμε ιδιότητες ενός πόρου (γνωρίσματα) είτε συσχετίσεις μεταξύ πόρων (σχέσεις) χρησιμοποιώντας «επώνυμες» ιδιότητες και τιμές και σχηματίζοντας τριάδες της μορφής <*υποκείμενο, κατηγόρημα, αντικείμενο*>, τις οποίες αποκαλούμε *δηλώσεις*. Τα μέρη της δήλωσης αντιστοιχούν στον περιγραφόμενο πόρο, την ιδιότητα που χρησιμοποιείται για την περιγραφή του και την τιμή της αντίστοιχα. Το σύνολο των δηλώσεων που αναφέρονται στο ίδιο υποκείμενο αποτελούν την *περιγραφή* ενός πόρου και σχηματίζουν δίκτυα πληροφορίας συσχετιζόμενων αντικειμένων. Επομένως, ένα RDF έγγραφο κάνει δηλώσεις ότι ορισμένα αντικείμενα (άτομα, ιστοσελίδες ή οτιδήποτε) έχουν ιδιότητες (π.χ όνομα) με ορισμένες τιμές (π.χ Αιμιλία). Το υποκείμενο και αντικείμενο μίας δήλωσης αναγνωρίζονται μοναδικά από ένα URI, όπως και τα κατηγορήματα, επιτρέποντας με αυτό τον τρόπο σε οποιονδήποτε χρήστη να ορίσει ένα νέο πόρο ή ένα νέο κατηγόρημα, απλώς δηλώνοντας ένα URI για αυτό κάπου στο Διαδίκτυο.

Ο ορισμός των γνωρισμάτων/σχέσεων που χρησιμοποιούνται για την περιγραφή των πόρων και της σημασιολογίας που φέρουν, επιτυγχάνεται μέσω του RDFS. Ένα RDF σχήμα (schema) εκφράζεται από το ίδιο μοντέλο δεδομένων [BG00] και αποτελείται από τις δηλώσεις κλάσεων, γνωρισμάτων και σχέσεων μεταξύ των κλάσεων, όπου μία κλάση χρησιμοποιείται για να ομαδοποιήσει λογικά ομοειδείς πόρους. Με άλλα λόγια, ο μηχανισμός των RDF σχημάτων προσφέρει ένα *σύστημα τύπων* για τα RDF μοντέλα, ένα *λεξιλόγιο* για τους έγκυρους όρους που μπορούν να χρησιμοποιηθούν για την περιγραφή των πληροφοριακών πόρων. Τα λεξιλόγια αυτά δημιουργούνται ανεξάρτητα, ενώ κοινότητες χρηστών μπορούν να επαναχρησιμοποιήσουν λεξιλόγια άλλων χρηστών είτε αυτούσια είτε εκλεπτύνοντάς τα σύμφωνα με τις ανάγκες της εκάστοτε κοινότητας.

Το μοντέλο δεδομένων του RDF/S μας προσφέρει ένα εννοιολογικό πλαίσιο για τον ορισμό και την χρήση μεταδεδομένων. Για τη δημιουργία όμως και ανταλλαγή αυτών των δεδομένων απαιτείται μία συγκεκριμένη σύνταξη. Η πιο διαδεδομένη μορφή σύνταξης για τη δημιουργία RDF/S εγγράφων βασίζεται στη χρήση της XML [BPS+00] και περιγράφεται στο [LS99]. Συνάμα, για την αναπαράσταση και οπτική απεικόνιση των RDF/S περιγραφών και σχημάτων χρησιμοποιείται, εκτός από το μοντέλο των δηλώσεων (τριάδων), ένα μοντέλο *κατευθυνόμενων γράφων με ετικέτες τόσο στις ακμές όσο και στους κόμβους* που μπορεί εύκολα να συνδυάσει πολλά διαφορετικά λεξιλόγια και να επεκταθεί προσθέτοντας απλώς περισσότερες ακμές. Το μοντέλο αυτό βασίζεται στα σημασιολογικά δίκτυα (semantic networks), ένα απλό σχήμα αναπαράστασης γνώσης που χρησιμοποιεί γράφους με ετικέτες στους κόμβους και κατευθυνόμενες ακμές με ετικέτες για να κωδικοποιήσει τη γνώση.



Εικόνα 2.1: RDF/S γράφος

Στο μοντέλο των κατευθυνόμενων γράφων που θα χρησιμοποιήσουμε για την αναπαράσταση RDF περιγραφών και σχημάτων, οι κόμβοι αναπαριστούν αντικείμενα (πόρους ή κλάσεις) και οι ακμές συμβολίζουν σχέσεις μεταξύ των κόμβων (ιδιότητες). Ως κόμβους αναπαριστούμε επίσης και τύπους Literal, δηλαδή

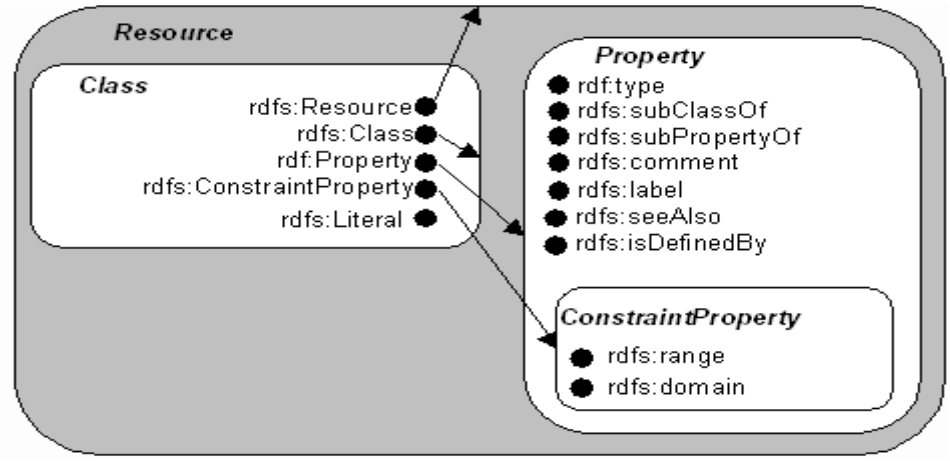
αλφαριθμητικά και άλλους βασικούς τύπους δεδομένων, όπως αυτοί ορίζονται στο XML σχήμα [XML99]. Οπτικά συμβολίζουμε τους κόμβους ως ελλείψεις και τα Literal ως παραλληλόγραμμα. Όσον αφορά τις ακμές, διακρίνουμε τρία είδη ακμών: *απόδοσης γνωρισμάτων* (μονό βέλος), *δημιουργίας στιγμιότυπων* (διακεκομμένο βέλος) και *υπαλληλίας* (διπλό βέλος). Με τη χρήση των ακμών απόδοσης γνωρισμάτων αναπαριστούμε γνωρίσματα κόμβων και σχέσεις μεταξύ τους, ενώ οι ακμές υπαλληλίας χρησιμοποιούνται για να δηλώσουν σε επίπεδο σχήματος ότι ένας κόμβος (κλάση) ή ιδιότητα είναι υποκατηγορία ενός πιο σημασιολογικά ευρύ κόμβου ή ιδιότητας αντίστοιχα. Σε αντίθεση με το τυπικό οντοκεντρικό μοντέλο, ο μηχανισμός των ακμών υπαλληλίας επιτρέπει το σχηματισμό ιεραρχιών κόμβων αλλά και ακμών. Οι ακμές δημιουργίας στιγμιότυπων παρέχουν το μηχανισμό που απαιτείται για τη χρήση όρων ενός ή πολλών σχημάτων στην περιγραφή πόρων και αποτελούν τη σύνδεση ανάμεσα στα πρότυπα RDF και RDFS. Ουσιαστικά επιτρέπουν τη δημιουργία στιγμιότυπων μίας κλάσης και την απόδοση τύπων σε πληροφοριακούς πόρους που περιγράφονται.

Για παράδειγμα, στον RDF/S γράφο της **Εικόνας 2.1**, διακρίνουμε σε επίπεδο σχήματος τις κλάσεις Artifact, Painting και rdfs:Literal (που αναπαριστά το σύνολο των Literal τύπων) και τις ιδιότητες creator και title, που αποδίδονται στην κλάση Painting. Η κλάση Painting ορίζεται ως υποκλάση της κλάσης Artifact χρησιμοποιώντας μία ακμή υπαλληλίας. Κάθε ένας από τους παραπάνω όρους αναγνωρίζεται μοναδικά από ένα URI, που για λόγους αναγνωσιμότητας δεν απεικονίζονται, αλλά παρουσιάζεται μία πιο αναγνώσιμη ετικέτα. Χρησιμοποιώντας τις κλάσεις και ιδιότητες που ορίζονται σε επίπεδο σχήματος, περιγράφουμε τον πόρο με αναγνωριστικό &r1, που αναπαριστά το gif αρχείο της εικόνας ενός γλυπτού με τίτλο (title) «The Thinker» και δημιουργό(creator) τον «Rodin». Η ακμή δημιουργίας στιγμιότυπων, που συνδέει τον κόμβο με ετικέτα &r1 με την κλάση Painting, δηλώνει το γεγονός ότι ο πόρος αυτός είναι τύπου Painting.

Ένας πολύ σημαντικός μηχανισμός για την ανάπτυξη RDF σχημάτων και τη δημιουργία RDF περιγραφών είναι ο *μηχανισμός ονοματοδοσίας XML* (XML namespace) [BHL99], ο οποίος επιτρέπει την επαναχρησιμοποίηση όρων από διαφορετικά σχήματα. Ένας χώρος ονοματοδοσίας είναι ένα σύνολο από κλάσεις και/ή ιδιότητες στο οποίον αποδίδεται ένα μοναδικό URI. Έτσι, ένα RDF σχήμα αποτελεί ένα χώρο ονοματοδοσίας που προσδιορίζεται μοναδικά από ένα URI. Με τη χρήση ενός XML χώρου ονοματοδοσίας, περιγραφικοί όροι (δηλαδή ονόματα κλάσεων ή/και ιδιοτήτων) αναγνωρίζονται μοναδικά από το URI του σχήματος, στο

οποίο ορίζονται (το οποίο παίζει το ρόλο ενός προθέματος) ως κανονικοί πόροι στο Διαδίκτυο.

Για να επιτύχει την μοντελοποίηση και περιγραφή των πληροφοριακών πόρων ενός πεδίου εφαρμογής, το RDF/S προσφέρει ένα σύνολο από δομικά στοιχεία και μηχανισμούς, τα οποία παρουσιάζονται εν συντομία στις επόμενες παραγράφους.



Εικόνα 2.2: RDF/S τυπικές κλάσεις και ιδιότητες

2.1.2 Οι τυπικές RDF/S κλάσεις και ιδιότητες

Η έννοια της κλάσης στο πρότυπο RDF/S [LS99, BG00] χρησιμοποιείται για να αναπαραστήσει ένα σύνολο από πληροφοριακούς πόρους, οι οποίοι ομοιάζουν βάσει ενός αριθμού κοινών χαρακτηριστικών. Το RDF/S προσφέρει ένα σύνολο από βασικές κλάσεις, οι οποίες μπορούν να χρησιμοποιηθούν για να μοντελοποιήσουν τη γνώση που εγκλείεται σε ένα τομέα γνώσης. Συγκεκριμένα, διακρίνουμε τις εξής κλάσεις[2] (**Εικόνα 2.2**):

- **rdfs:Resource**: οποιοσδήποτε πληροφοριακός πόρος που μπορεί να περιγραφεί με το RDF αποτελεί μέλος της κλάσης rdfs:Resource. Επομένως, μπορούμε να περιγράψουμε web σελίδες, π.χ "http://www.csd.uoc.gr", τμήματα αυτών ή συλλογές από σελίδες. Εκτός από πληροφοριακούς πόρους άμεσα συνδεδεμένους με το Διαδίκτυο, μπορούμε να περιγράψουμε αντικείμενα του

[2] Το πρόθεμα **rdfs** αντιστοιχεί στο URI "http://www.w3.org/2000/01/rdf-schema#" και το πρόθεμα **rdf** στο URI "http://www.w3.org/1999/02/22-rdf-syntax-ns#", τα οποία αντιστοιχούν στους χώρους ονοματοδοσίας των προτύπων RDFS και RDF αντίστοιχα.

φυσικού κόσμου, όπως πρόσωπα, εκδόσεις κτλ. Όπως έχουμε ήδη αναφέρει, κάθε πληροφοριακός πόρος διακρίνεται από ένα URI [BFM98] συν ένα προαιρετικό id.

- **rdfs:Class**: Μία κλάση κατά το πρότυπο RDF/S αντιστοιχεί στην βασική έννοια του «*τύπου*» ή «*κατηγορίας*» ενός πόρου και η σχέση μέλους ενός πόρου σε μία κλάση δηλώνει τον πόρο αυτό ως στιγμιότυπο της κατηγορίας αυτής. Ουσιαστικά, ορίζει εάν μία ιδιότητα μπορεί να αποδοθεί σε ένα αντικείμενο αυτής της κλάσης. Εντούτοις, μία *rdfs:Class* δεν επιβάλλει ένα συγκεκριμένο τύπο για τα μέλη της και πόροι που ανήκουν στην ίδια κλάση μπορεί να περιγράφονται με διαφορετικές ιδιότητες.
- **rdf:Property**: η κλάση *rdf:Property* χρησιμοποιείται για να αναπαραστήσει όλες τις ιδιότητες (γνωρίσματα ή σχέσεις) που μπορούν να αποδοθούν σε πόρους. Η ιδιαιτερότητα του RDF/S μοντέλου, σε σύγκριση με το οντοκεντρικό μοντέλο δεδομένων, έγκειται στο γεγονός ότι οι ιδιότητες στο RDF θεωρούνται «*πολίτες πρώτης κατηγορίας*» (first-class citizens), δεδομένου ότι μπορούν να οριστούν και να χρησιμοποιηθούν ανεξάρτητα από τις κλάσεις στις οποίες προσδίδονται. Στο RDF οι ιδιότητες είναι *μη-διατεταγμένες*, *προαιρετικές* και *πλειότιμες*, που σημαίνει ότι η χρήση τους είναι προαιρετική για την περιγραφή πόρων που ανήκουν στην κλάση στην οποία προσδίδονται ή μπορούν να χρησιμοποιηθούν περισσότερες από μία φορές.
- **rdfs:Literal**: η κλάση *rdfs:Literal* αντιπροσωπεύει το σύνολο των τιμών βασικών τύπων, για παράδειγμα συμβολοσειρές χαρακτήρων (string) και αριθμητικούς τύπους (πραγματικοί και ακέραιοι αριθμοί). Σύμφωνα με το μοντέλο γράφων που ακολουθεί το RDF/S, η κλάση rdfs:Literal αναπαριστά τους «αυτο-περιγραφόμενους» (self-denoting) κόμβους, δηλαδή κόμβους στους οποίους δεν μπορούμε να αποδώσουμε κάποια ιδιότητα. Με αυτό τον τρόπο, αν και μπορούμε να αποδώσουμε ιδιότητες στην ίδια την κλάση rdfs:Literal, οι ιδιότητες αυτές δεν μπορούν να αποδοθούν στα μέλη της κλάσης αυτής. Επομένως, οι τιμές literal μπορούν να χρησιμοποιηθούν στο πεδίο τιμών μίας ιδιότητας και όχι στο πεδίο ορισμού της.
- **rdf:Statement**: αντιπροσωπεύει την κλάση των δηλώσεων RDF, δηλαδή τριάδων της μορφής <*υποκείμενο, κατηγόρημα, αντικείμενο*>[3]. Η κλάση αυτή χρησιμοποιείται από τον μηχανισμό υποστασιοποίησης (reification), με τον όποιο

[3] Αν και η μορφή μίας τριάδας μπορεί να διαφέρει ανάλογα με την θέση των τμημάτων από τα οποία αποτελείται, το γεγονός το οποίο απεικονίζει είναι το ίδιο.

μπορούμε να εκφράσουμε περιγραφές για περιγραφές ή, με άλλα λόγια, δηλώσεις για κάποιες άλλες δηλώσεις.

- **rdfs:Container**: η κλάση *rdfs:Container* αναπαριστά το σύνολο των *συλλογών* πόρων που μπορούν να περιγραφούν από το RDF. Το RDF/S παρέχει 3 κλάσεις συλλογών (υποκλάσεις της κλάσης rdfs:Container), τις *rdf:Bag*, *rdf:Seq* και *rdf:Alt* για την αναπαράσταση μη-διατεταγμένων, διατεταγμένων και συλλογών εναλλακτικών τιμών αντίστοιχα. Οι κλάσεις αυτές αποτελούν μία φτωχή γλώσσα για την κατασκευή συλλογών, σε αντίθεση με γνωστές γλώσσες προγραμματισμού, και χρησιμοποιούνται κυρίως για την περιγραφή συλλογών.
- **rdfs:ContainerMembershipProperty**: η κλάση αυτή έχει ως μέλη τις ιδιότητες rdf:_1, rdf:_2 κτλ, οι οποίες χρησιμοποιούνται για να συσχετίσουν μία συγκεκριμένη «θέση» σε μία συλλογή με μία οντότητα.

Η βαρύτητα των ιδιοτήτων στο πρότυπο RDF/S φαίνεται από το γεγονός ότι θεωρούνται πολίτες πρώτης κατηγορίας. Αποτελούν τη βάση για την περιγραφή των πόρων και χρησιμοποιούνται για την αναπαράσταση υπό μορφή γράφων των RDF/S βάσεων περιγραφών. Όντας οι ίδιες πόροι, μπορούν να περιγραφούν χρησιμοποιώντας το πρότυπο RDF. Διακρίνουμε τις εξής βασικές ιδιότητες:

- **rdf:type**: η ιδιότητα rdf:type χρησιμοποιείται για να δηλώσει τη σχέση μέλους ενός πόρου σε μία κλάση και αναπαρίσταται γραφικά από τις ακμές δημιουργίας στιγμιότυπων. Η πολλαπλή χρήση της ιδιότητας αυτής επιτρέπει τη δήλωση ενός πόρου ως μέλος μίας ή περισσότερων κλάσεων.
- **rdfs:subClassOf**, **rdfs:subPropertyOf**: η χρήση των ιδιοτήτων *rdfs:subClassOf* και *rdfs:subPropertyOf* σχηματίζει ιεραρχίες κλάσεων και ιδιοτήτων αντίστοιχα. Γραφικά συμβολίζονται με ακμές υπαλληλίας (isA links). Η πολλαπλή χρήση των ιδιοτήτων αυτών δηλώνει μία κλάση/ιδιότητα ως υποκλάση/υποϊδιότητα πολλών διαφορετικών κλάσεων/ιδιοτήτων.
- **rdfs:domain, rdfs:range**: οι ιδιότητες αυτές χρησιμοποιούνται για να περιορίσουν το πεδίο ορισμού και τιμών μίας ιδιότητας. Με τον τρόπο αυτό, δηλώνουμε τον τύπο των πόρων στους οποίους μπορεί να εφαρμοστεί μία ιδιότητα και τον τύπο των πόρων από τους οποίους μπορεί να πάρει τιμές. Το μοντέλο RDF/S επιτρέπει την πολλαπλή χρήση των ιδιοτήτων αυτών, επομένως μία ιδιότητα μπορεί να αποδοθεί σε πολλές διαφορετικές κλάσεις και να πάρει τιμές από ένα σύνολο κλάσεων.
- **rdf:subject, rdf:predicate, rdf:object**: οι ιδιότητες αυτές, που αποδίδονται σε πόρους τύπου rdf:Statement, αντιπροσωπεύουν αντίστοιχα το υποκείμενο (subject), κατηγόρημα (subject) και αντικείμενο (subject) μίας δήλωσης (statement).

Ειδικότερα, το κατηγόρημα αναπαριστά μία ιδιότητα ή σε ένα γράφο την ακμή που συνδέει δύο κόμβους.

- **rdf:value**: η ιδιότητα αυτή χρησιμοποιείται για να δηλώσει την πρωτεύουσα τιμή (συνήθως μία συμβολοσειρά χαρακτήρων) μίας ιδιότητας όταν η τιμή της ιδιότητας είναι ένας δομημένος πόρος. Με άλλα λόγια, χρησιμοποιείται για την αναπαράσταση σχέσεων με πληθικότητα μεγαλύτερη του δύο π.χ τριαδικές σχέσεις.
- **rdfs:isDefinedBy**: η βοηθητική αυτή ιδιότητα χρησιμοποιείται για να δηλώσει το χώρο ονοματοδοσίας (namespace) ενός πόρου και συνήθως χρησιμοποιείται για την τεκμηρίωση της ανάπτυξης ενός σχήματος.
- **rdfs:comment, rdfs:label, rdfs:seeAlso**: όπως και η προηγούμενη ιδιότητα, χρησιμοποιούνται για να τεκμηριώσουν την ανάπτυξη ενός σχήματος. Συγκεκριμένα, η ιδιότητα *rdfs:comment* χρησιμοποιείται για να περιγράψει μία κλάση ή ιδιότητα, η *rdfs:label* για να αποδώσει ένα πιο αναγνώσιμο και κατανοητό όνομα για ένα πόρο, ενώ η *rdfs:also* δηλώνει ένα πόρο που παρέχει πληροφορία για τον πόρο που περιγράφεται.

Το σύνολο των κλάσεων και ιδιοτήτων που παρουσιάστηκαν παραπάνω αποτελούν τη βάση για την ανάπτυξη RDF σχημάτων και την περιγραφή πληροφοριακών πόρων. Στη συνέχεια θα αναφέρουμε τους εννοιολογικούς μηχανισμούς που βρίσκονται πίσω από τη χρήση των βασικών κλάσεων και ιδιοτήτων του RDF/S και οι οποίοι χαρακτηρίζουν το RDF/S ως μία απλή αλλά εκφραστική γλώσσα κατασκευής σχημάτων, τα οποία μπορούν να ανταλλαγούν και να επαναχρησιμοποιηθούν από χρήστες του Διαδικτύου.

2.1.3 Οι εννοιολογικοί μηχανισμοί του RDF/S

Οι εννοιολογικοί μηχανισμοί που προσδιορίζουν την εκφραστικότητα του RDF/S διακρίνονται σε *μηχανισμούς αφαίρεσης* και *μηχανισμούς έκφρασης περιορισμών*. Ειδικότερα, το πρότυπο RDF/S υποστηρίζει τους ακόλουθους μηχανισμούς αφαίρεσης:

- **(πολλαπλή) κληρονομικότητα** στην ιεραρχία των κλάσεων και ιδιοτήτων: ο μηχανισμός αυτός μας επιτρέπει να ορίσουμε μία κλάση (ιδιότητα) ως υποκλάση (υποϊδιότητα) πολλών κλάσεων (ιδιοτήτων) με την (πολλαπλή) χρήση της βασικής ιδιότητας *rdfs:subClassOf* (*rdfs:subPropertyOf*). Με τον τρόπο αυτό μία κλάση κληρονομεί τις ιδιότητες όλων των υπερκλάσεών της˙ ένας μηχανισμός που εξυπηρετεί σε μεγάλο βαθμό την ανάπτυξη σχημάτων.

- **(πολλαπλή) ταξινόμηση των πόρων**: η πολλαπλή ταξινόμηση πόρων χρησιμοποιείται για να δηλώσει ένα πόρο ως στιγμιότυπο κάποιων κλάσεων μέσω της βασικής ιδιότητας *rdf:type*. Λόγω του μηχανισμού αυτού, σε ένα πόρο μπορούν να αποδοθούν πολλές ιδιότητες από διαφορετικές κλάσεις, όχι απαραίτητα συναφούς σημασιολογίας. Έτσι, ένας πόρος μπορεί να περιγραφεί από

Ιδιότητα	Πεδίο ορισμού	Πεδίο Τιμών
rdfs:isDefinedBy	Rdfs:Resource	rdfs:Resource
rdf:subject	Rdf:Statement	rdfs:Resource
rdf:predicate	Rdf:Statement	rdf:Property
rdf:object	Rdf:Statement	Not specified
rdf:type	Rdfs:Resource	rdfs:Class
rdfs:subClassOf	rdfs:Class	rdfs:Class
rdfs:subPropertyOf	Rdf:Property	rdf:Property
rdfs:comment	Rdfs:Resource	rdfs:Literal
rdfs:label	Rdfs:Resource	rdfs:Literal
rdfs:domain	Rdf:Property	rdfs:Class
rdfs:range	Rdf:Property	rdfs:Class
rdfs:seeAlso	Rdfs:Resource	rdfs:Literal

Πίνακας 2.1: Πεδία ορισμού και τιμών των βασικών RDF/S ιδιοτήτων

πολλές «οπτικές γωνίες», για παράδειγμα ως αντικείμενο τέχνης (τύπου Artifact) ή ως αντικείμενο αγοραπωλησίας (τύπου Barter).

- **υποστασιοποίηση (reification)**: αν και ο μηχανισμός αυτός δεν εκφράζεται σε επίπεδο σχήματος, το RDF/S παρέχει μηχανισμό για την αναπαράσταση δηλώσεων. Ο μηχανισμός αυτός, τυπικά γνωστός ως υποστασιοποίηση από την περιοχή Αναπαράστασης Γνώσης, είναι εφαρμόσιμος σε επίπεδο δεδομένων και τα δομικά στοιχεία που χρησιμοποιούνται για το σκοπό αυτό είναι τα *statement*, *subject*, *predicate* και *type*. Η υποστασιοποίηση επιτρέπει το σχηματισμό δηλώσεων με αντικείμενο άλλες δηλώσεις.

Αν και το RDF/S δεν υποστηρίζει μηχανισμούς για να δηλώσει πολύπλοκους περιορισμούς πάνω στις ιδιότητες, όπως στην περίπτωση της DAML+OIL [CHH+01] ή των γλωσσών βασισμένων στο μοντέλο των frames, μπορούμε να δηλώσουμε απλούς περιορισμούς με τη χρήση των βασικών ιδιοτήτων *rdfs:domain* και *rdfs:range*. Ο **Πίνακας 2.1** παρουσιάζει τις βασικές ιδιότητες του RDF/S και πως αυτές περιορίζονται με τη χρήση των ιδιοτήτων περιορισμού του πεδίου ορισμού και τιμών τους.

Το σύνολο των εννοιολογικών μηχανισμών που υποστηρίζει καθιστά το RDF/S μία απλή γλώσσα κατασκευής εννοιολογικών σχημάτων (οντολογιών). Αν και δεν μπορούμε να εκφράσουμε κανόνες ή πολύπλοκους λογικούς περιορισμούς

με τα δομικά στοιχεία και τους εννοιολογικούς μηχανισμούς του RDF/S, διαπιστώνεται στο [MAC+02] ότι το πρότυπο αυτό απολαμβάνει ολοένα και μεγαλύτερης αποδοχής από την κοινότητα του Σημασιολογικού Ιστού για την κατασκευή απλών σχημάτων περιγραφής πληροφοριακών πόρων. Τα πρότυπα DAML+OIL [CHH+01] και Web Ontology Language (OWL) [DCH02] αποτελούν γλώσσες που, βασιζόμενες στο RDF/S, παράσχουν πιο πλούσιες σημασιολογικά δυνατότητες για την κατασκευή **οντολογιών**[4]. Πιο συγκεκριμένα, η DAML+OIL παρέχει βασικές έννοιες μοντελοποίησης που απαντώνται σε γλώσσες βασισμένες στην έννοια των πλαισίων (frame-based). Αποτελείται από ένα σύνολο από αξιώματα (axioms), π.χ για δήλωση γεγονότων σχετικά με κλάσεις, ιδιότητες και πόρους ή για την υπαλληλία/ισοδυναμία κλάσεων, επιτρέπει τον ορισμό περιορισμών σε ιδιότητες και παρέχει τη δυνατότητα ορισμού εκφράσεων κλάσεων με τη χρήση διαφόρων τελεστών. Την ίδια προσέγγιση ακολουθεί και η OWL που βασίζεται τόσο στο RDF/S όσο και στην DAML+OIL, της οποίας είναι επέκταση. Η επόμενη παράγραφος φιλοξενεί ενδεικτικές περιοχές εφαρμογής του RDF/S, παρουσιάζοντας επιγραμματικά εφαρμογές που κάνουν χρήση RDF/S τεχνολογίας.

2.2 Περιοχές εφαρμογής του RDF/S

Το σύνολο των δομικών στοιχείων που προσφέρει το RDF/S συνιστά ένα απλό μηχανισμό αναπαράστασης γνώσης, ανεξάρτητο από οποιοδήποτε πεδίο εφαρμογής. Η ουδετερότητα του ως προς τις ιδιαιτερότητες των πεδίων εφαρμογών καθιστά την τεχνολογία του εφαρμόσιμη σε ένα ευρύ πεδίο πρακτικών εφαρμογών και RDF/S σχήματα για την περιγραφή πληροφοριακών πόρων κοινοτήτων χρηστών έχουν αναπτυχθεί για ποικίλους τομείς εφαρμογών. Ενδεικτικά αναφέρουμε [MAC+02]:

- **Πολιτιστική κληρονομιά/Αρχειοθέτηση/Βιβλιοθήκες (Cultural heritage /Archives/Libraries)**: τα σχήματα αυτού του τομέα εξυπηρετούν δύο διαφορετικές λειτουργικότητες: είτε παρέχουν τυπικούς ορισμούς όρων και διαδικασιών που αναφέρονται σε βιβλιοθήκες, μουσεία και γενικότερα εφαρμογές πολιτιστικής

[4] Αν και ο όρος *Οντολογία* χρησιμοποιείται ευρέως στα πλαίσια του Σημασιολογικού Ιστού, δεν υπάρχει ένας κοινά αποδεκτός ορισμός του όρου αυτού [π.χ CJB99, G98, M99, U95]. Γενικά, μπορούμε να χαρακτηρίσουμε μία οντολογία ως μία ρητή και σαφή περιγραφή ενός πεδίου γνώσης. Οι οντολογίες παρέχουν ένα λεξιλόγιο για την αναπαράσταση, ανταλλαγή και επαναχρησιμοποίηση της γνώσης ενός τομέα εφαρμογής και ένα σύνολο σχέσεων που ισχύουν ανάμεσα στους όρους του εν λόγω λεξιλογίου.

κληρονομιάς είτε προσφέρουν κατευθυντήριες οδηγίες για την κωδικοποίηση, τη δομή και την ανταλλαγή πληροφορίας.

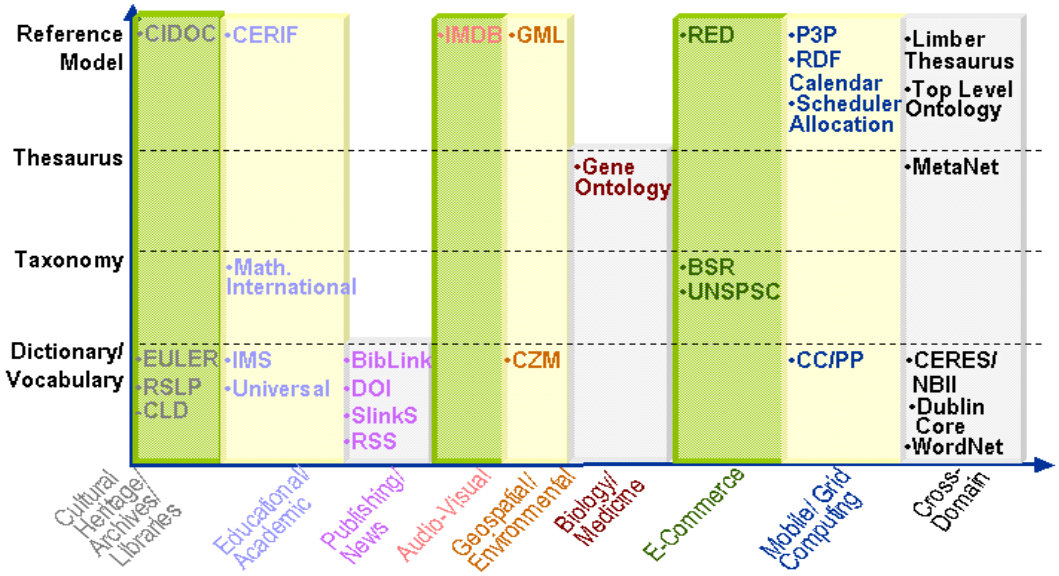
- **Εκπαίδευση/Ακαδημαϊκός χώρος (Education/Academic)**: τα σχήματα αυτού του είδους εξυπηρετούν είτε ως γλωσσάρια για τη διευκόλυνση της ανταλλαγής εκπαιδευτικών πόρων σε παγκόσμιο επίπεδο, όπως ασκήσεις, διαγράμματα, διαφάνειες και τηλεοπτικό υλικό είτε παρέχουν μέσα για τη περιγραφή και τυποποίηση ερευνητικών διαδικασιών και επιστημονικών εκδόσεων.
- **Εκδόσεις/Ηλεκτρονική Πληροφόρηση (Publishing/News)**: τα σχήματα αυτού του τομέα προσφέρουν λεξιλόγια για την κωδικοποίηση και ανταλλαγή πληροφόρησης ανάμεσα σε ιδιώτες και οργανισμούς μέσων μαζικής ενημέρωσης (πρακτορεία νέων, εφημερίδες κτλ). Αναφέρονται σε κάθε είδους έκδοση και ηλεκτρονικό υλικό, σε οποιαδήποτε μορφή (CDROM, DVD, διαφάνειες κτλ), τυπωμένο ή σε άμεση σύνδεση (on-line) και τις ιδιότητες τους, για παράδειγμα τον δημιουργό τους, τη περίοδο εγκυρότητας ή την έκδοσή τους.
- **Οπτικοακουστικός Τομέας (Audio-visual)**: τα σχήματα που εμπίπτουν σε αυτή την κατηγορία είναι ουσιαστικά σχήματα των βασικών όρων που χρησιμοποιούνται για να αναπαραστήσουν πληροφορία για άτομα της κινηματογραφικής βιομηχανίας, και της παραγωγής και διακίνησης πολυμέσων. Μπορεί επίσης να περιγράφουν πληροφορία σχετικά με διαδικασίες και γεγονότα σχετιζόμενα με κάθε πλευρά της παραγωγής και πώλησης κινηματογραφικών ταινιών και πολυμέσων, όπως τη διαφήμιση, τη διανομή ρόλων, την ηθοποιία κτλ.
- **Γεωγραφία/Περιβάλλον (Geospatial/Environmental)**: τα σχήματα αυτής της κατηγορίας περιγράφουν μία ορολογία εννοιών και κατευθυντήριων οδηγιών για την αναπαράσταση και διαμοιρασμό γεωγραφικής και περιβαλλοντικής πληροφορίας. Ο τομέας αυτός είναι προς το παρόν περισσότερο ανεξερεύνητος από τους υπόλοιπους από πλευράς αναπαράστασης γνώσης, αλλά περισσότερες προσπάθειες βρίσκονται ήδη σε εξέλιξη.
- **Βιολογία/Ιατρική (Biology/Medicine)**: τα σχήματα που έχουν κατηγοριοποιηθεί σε αυτό τον τομέα είναι κυρίως θησαυροί όρων και ελεγχόμενα λεξιλόγια που παρέχουν ορισμούς και σημασιολογικές σχέσεις ανάμεσα στους όρους. Η κυριότερη λειτουργικότητα τέτοιων σχημάτων είναι να διευκολύνουν τη διαλειτουργικότητα μεταξύ συστημάτων που αποθηκεύουν, επεξεργάζονται και επερωτούν βιολογικά και ιατρικά δεδομένα και να θέτουν τη βάση για την επικοινωνία μεταξύ ατόμων προσφέροντας επίσημα καταχωρημένους ορισμούς όρων.

- **Ηλεκτρονικό Εμπόριο (E-Commerce)**: σχήματα αυτής της κατηγορίας είναι κυρίως λεξιλόγια και ιεραρχίες που διευκρινίζουν τους όρους που χρησιμοποιούνται σε εφαρμογές ηλεκτρονικού εμπορίου, όπως διαχείριση κτηματομεσιτικών επενδύσεων, διαφήμιση ή πωλήσεις μέσω του διαδικτύου. Προσφέρουν μία κεντρική αναφορά σε επίσημα καταχωρημένους όρους σχετικούς με λογαριασμούς, πράκτορες, υπηρεσίες, οικονομικές συνδιαλλαγές κτλ, που χρησιμοποιούνται για να διευκολύνουν την επικοινωνία μεταξύ πελατών, προμηθευτών, επιχειρήσεων, πωλητών ή κάθε άλλης οντότητας που συμμετέχει σε οικονομικές συνδιαλλαγές.
- **Κινητή/Δικτυοκεντρική επεξεργασία πληροφοριών (Mobile/Grid Computing)**: περισσότερες πρωτοβουλίες σε αυτό τον τομέα έχουν ήδη δρομολογηθεί και αφορούν κυρίως λεξιλόγια όρων που επιτρέπουν την ανταλλαγή δεδομένων ανάμεσα σε συσκευές (π.χ τεχνικά χαρακτηριστικά ενός πελάτη ή ενός δικτύου), καθώς και δεδομένα για την κατανομή πόρων από ένα δρομολογητή του δικτύου.
- **Γενικές εφαρμογές (Cross –domain)**: τα σχήματα αυτά είναι συνήθως λεξιλόγια απαρτιζόμενα από γενικού σκοπού, περιγραφικούς όρους από ένα εκτεταμένο πεδίο εφαρμογών, που μπορούν να χρησιμοποιηθούν σε μία ποικιλία εφαρμογών. Για το λόγο αυτό, μερικά από τα σχήματα αυτά μπορούν να λειτουργήσουν ως μέσα ανταλλαγής ή ως θησαυροί γενικών όρων με σκοπό να διευκολύνουν περισσότερο την επικοινωνία και διαλειτουργικότητα.

Οι αναφερθείσες περιοχές εφαρμογών είναι απλά ενδεικτικές της ευρύτητας των πεδίων γνώσης στα οποία είναι εφαρμόσιμο το RDF/S. Στο [FBD+02] αναφέρονται επίσης ένα σύνολο από περιοχές εφαρμογών και πρακτικές υλοποιήσεις, στις οποίες εφαρμόζεται η τεχνολογία Σημασιολογικού Ιστού. Η περιοχή εφαρμογής στην οποία αναφέρεται ένα σχήμα είναι μία από τις διαστάσεις, τις οποίες μπορούμε να χρησιμοποιήσουμε για να κατηγοριοποιήσουμε RDF/S σχήματα. Η **Εικόνα 2.3** παρουσιάζει διαγραμματικά την ταξινόμηση 28 σχημάτων από τις 9 περιοχές εφαρμογών που αναφέρθησαν. Το *σημασιολογικό βάθος*, ο βαθμός δηλαδή στον οποίο χρησιμοποιούνται στην κατασκευή σχημάτων οι διάφορες σχέσεις που μπορούν να εκφραστούν με το RDF/S, (π.χ σχέσεις υπαλληλίας ιδιοτήτων και κλάσεων και ορισμένες από το χρήστη ιδιότητες) είναι η δεύτερη διάσταση που χρησιμοποιείται για να κρίνει, κατά κάποιο τρόπο, την πολυπλοκότητα κατασκευής και τη σημασιολογική επάρκεια ενός σχήματος. Με την ευρύτερη έννοια του όρου, μπορούμε να χαρακτηρίσουμε κάθε σχήμα ως **οντολογία**, εφόσον μπορεί να αποτελέσει ένα λεξιλόγιο όρων αποδεκτό από ένα σύνολο ατόμων και οργανισμών.

Γενικά, μπορούμε να διακρίνουμε τα εξής επίπεδα σημασιολογικού βάθους, τα οποία απεικονίζονται στον κάθετο άξονα της Εικόνας 2.3:

- **Λεξικά και Γλωσσάρια (Dictionaries, Vocabularies)**: τα σχήματα που αναπτύσσονται σε αυτό το επίπεδο είναι απλοί κατάλογοι από έννοιες και τους ορισμούς τους. Τις περισσότερες φορές αποτελούνται από δηλώσεις κλάσεων και η δομή τους είναι σχεδόν επίπεδη (δεν ορίζουν ιεραρχίες κλάσεων/ιδιοτήτων).
- **Ταξινομίες (Taxonomies)**: το χαρακτηριστικό των ταξινομιών είναι ότι η κυριότερη σχέση που ορίζουν μεταξύ των όρων είναι η σχέση υπαλληλίας. Το ιεραρχικό βάθος μίας ταξινομίας εξαρτάται από τη λεπτομέρεια στην οποία επιθυμεί να αναλύσει τους όρους ενός πεδίου ο κατασκευαστής του σχήματος.
- **Θησαυροί (Thesauri)**: εκτός του ορισμού σχέσεων ευρύτερου/στενότερου όρου με το σχηματισμό ιεραρχιών, ένας θησαυρός ορίζει επίσης και σχέσεις ισοδυναμίας, συσχέτισης και συνωνυμίας. Η φύση αυτών των σημασιολογικών σχέσεων είναι αυτό που διακρίνει τους θησαυρούς από τις ταξινομίες.
- **Μοντέλα Αναφοράς (Reference Models)**: ένα μοντέλο αναφοράς συνδυάζει όλες τις προαναφερθείσες σχέσεις για να συλλάβει τη σημασιολογία του πεδίου εφαρμογής. Αποτελείται από ένα λεξιλόγιο αναπαράστασης των όρων του πεδίου εφαρμογής και από τις λογικές δηλώσεις που περιγράφουν τη φύση των όρων, τις σχέσεις μεταξύ τους και τον τρόπο που μπορούν ή όχι να συσχετιστούν όροι μεταξύ τους.



Εικόνα 2.3: Κατηγοριοποίηση 28 RDF/S σχημάτων ανάλογα με περιοχή εφαρμογής και σημασιολογικό βάθος

Το πλήθος των διαφορετικών εφαρμογών στις οποίες είναι εφαρμόσιμη η τεχνολογία RDF/S, καθώς και η ποικιλομορφία στην ακρίβεια ανάπτυξης RDF/S σχημάτων

καθιστούν αναγκαία την ύπαρξη εργαλείων με δυνατότητα κλιμάκωσης για τη διαχείριση κάθε πτυχής της κατασκευής και χρήσης γνώσης κωδικοποιημένης στο RDF/S. Ειδικότερα κρίνονται αναγκαία:

- *Εργαλεία ανάπτυξης RDF/S εφαρμογών*: στην ομάδα αυτή συγκαταλέγονται εργαλεία και περιβάλλοντα κατασκευής και επαναχρησιμοποίησης RDF/S σχημάτων, περιγραφής πληροφοριακών πόρων, τεκμηρίωσης, αξιολόγησης, πλοήγησης σε RDF/S γράφους και οπτικοποίησης κτλ.
- *Εργαλεία ολοκλήρωσης σχημάτων*: τα εργαλεία της κατηγορίας αυτής στοχεύουν στην επίλυση του προβλήματος της συγχώνευσης ή ολοκλήρωσης σχημάτων από το ίδιο πεδίο εφαρμογών.
- *Εργαλεία αποθήκευσης και* επερώτησης *RDF/S δεδομένων*: τα εργαλεία αυτά παίζουν σημαντικό ρόλο, δεδομένου ότι επιτρέπουν τη διαχείριση γνώσης. Σε αυτά συγκαταλέγονται συστήματα βάσεων δεδομένων προσαρμοσμένων στο πρότυπο RDF/S, μηχανές σημασιολογικής αναζήτησης, μηχανές εξαγωγής γνώσης βάσει συμπερασματικών κανόνων (inference engines) κτλ.

Εκτός όμως από την ύπαρξη εργαλείων για τη διαχείριση γνώσης στο Σημασιολογικό Ιστό, οι εφαρμογές που αναπτύσσονται σε αυτό το πλαίσιο θέτουν απαιτήσεις, όπως η διαλειτουργικότητα των συστημάτων και η ολοκλήρωση των εργαλείων σε περιβάλλοντα εργασίας, απαιτήσεις που αφορούν την απόδοση και τη δυνατότητα κλιμάκωσης σε συνθήκες αυξημένου όγκου δεδομένων. Μία από τις πτυχές της δυνατότητας κλιμάκωσης είναι η **συναρμολογησιμότητα** (modularity) των (RDF/S) οντολογιών. Μία ογκώδης οντολογία δημιουργεί προβλήματα στην ομαλή και αποτελεσματική διαχείρισή της λόγω του μεγέθους της. Σε αυτό το πρόβλημα, ένας μηχανισμός όψεων μας δίνει τη δυνατότητα να διαχειριστούμε τμήματα αυτής ξεχωριστά, ορίζοντας αυτά τα τμήματα ως όψεις επί του συνολικού σχήματος. Οι χρήστες με αυτό τον τρόπο αποκτούν μία επιλεκτική εικόνα επί του συνόλου της πληροφορίας που περιέχεται στην οντολογία. Επίσης η συναρμολογησιμότητα μίας οντολογίας συντελεί στην εύκολη επέκτασή της, καθώς, λόγω του ότι οι χρήστες βλέπουν μόνο τμήμα αυτής της οντολογίας, μπορούν να το επεκτείνουν χωρίς προβλήματα. Για παράδειγμα, η UNSPSC[5] οντολογία είναι μία ταξινομία 16.000 όρων, την οποία ανέπτυξε και διαχειρίζεται η ένωση Electronic Commerce Code Management Association. Αποτελεί το πρώτο σύστημα ταξινόμησης προϊόντων και υπηρεσιών και παρέχει μία κοινή ορολογία για χρήση σε μία αγορά. Το μέγεθός της είναι τέτοιο που δυσχεραίνει τη χρήση της από κοινότητες εμπορίου που

[5] http://eccma.org/unspsc/

διαπραγματεύονται ένα μικρό τμήμα των προϊόντων που περιγράφει. Επομένως, ένας μηχανισμός ορισμού όψεων επιτρέπει σε μικρές ομάδες ατόμων να εργαστούν σε εύχρηστες οντολογίες, οι οποίες έχουν οριστεί ως τμήματα της UNSPSC οντολογίας, και οι οποίες ταιριάζουν στις δικές τους απαιτήσεις και ενδιαφέροντα.

Μία ακόμα απαίτηση που τίθεται από τις εφαρμογές πραγματικής κλίμακας στο Σημασιολογικό Ιστό είναι η ευκολία στην επαναχρησιμοποίηση γνώσης από διεσπαρμένες πηγές πληροφορίας. Ένας από τους τρόπους για την επαναχρησιμοποίηση γνώσης είναι η ύπαρξη βιβλιοθηκών αποθήκευσης σχημάτων προσβάσιμων από το Διαδίκτυο (repositories). Προς αυτή την κατεύθυνση έχουν ήδη γίνει προσπάθειες, οι κυριότερες των οποίων είναι οι SCHEMAS Project[6], DESIRE Registry[7], SWAG Dictionary[8] και Xmlns.com[9]. Ένας δεύτερος τρόπος αφορά την ύπαρξη μηχανισμών που μπορούν να προσφέρουν στο χρήστη γνώση από πολλά διαφορετικά σχήματα και βάσεις περιγραφών, σύμφωνα με τις εκάστοτε πληροφοριακές ανάγκες, όπως είναι ο μηχανισμός των όψεων. Η επόμενη παράγραφος εστιάζεται σε εργαλεία για την ανάκτηση γνώσης από διεσπαρμένες πηγές με τη χρήση RDF/S γλωσσών επερώτησης.

2.3 Γλώσσες επερώτησης για RDF/S μεταδεδομένα

Το RDF/S μας παρέχει τα κυριότερα δομικά στοιχεία για την αναπαράσταση γνώσης και πληροφορίας. Εντούτοις, η κατασκευή σχημάτων και η περιγραφή πόρων δεν είναι οι μοναδικές διαδικασίες που απαιτούνται για τη διαχείριση γνώσης στο Διαδίκτυο. Οι καταναλωτές πληροφορίας (άνθρωποι και μηχανικοί πράκτορες) πρέπει να μπορούν να χρησιμοποιούν και να επερωτούν RDF σχήματα και βάσεις περιγραφών, αποκτώντας με αυτό τον τρόπο γνώση από διεσπαρμένες πηγές πληροφορίας. Απόρροια αυτής της ανάγκης είναι η ύπαρξη ενός συνόλου γλωσσών επερώτησης και αντίστοιχων συστημάτων αποθήκευσης και επερώτησης RDF/S δεδομένων. Σκοπός της ενότητας είναι να παρουσιάσει εν συντομία τις κυριότερες γλώσσες επερώτησης RDF/S και να συγκρίνει τα βασικότερα χαρακτηριστικά τους από άποψη λειτουργικότητας. Η σύγκριση αυτή θα αιτιολογήσει την απόφασή μας

[6] http://www.schemas-forum.org/
[7] http://desire.ukoln.ac.uk/registry/
[8] http://webns.net/
[9] http://www.xmlns.com

να στηριχτούμε στη γλώσσα επερώτησης RQL [KAC+02] και να προσφέρουμε ένα μηχανισμό ορισμού όψεων βασισμένο σε αυτή.

Πιο συγκεκριμένα, η λειτουργικότητα που πρέπει να υποστηρίζει μία γλώσσα επερώτησης πρέπει πάντα να λαμβάνει υπόψη τη δομή και τις ιδιαιτερότητες του υποκείμενου μοντέλου δεδομένων. Ειδικότερα, οι γλώσσες επερώτησης για το RDF/S πρέπει να σέβονται τα μοντέλα δηλώσεων-τριάδων και κατευθυνόμενων γράφων με ετικέτες στους κόμβους και τις ακμές που χρησιμοποιούνται εναλλακτικά για να αναπαραστήσουν RDF/S σχήματα και περιγραφές. Οι κυριότερες γλώσσες που έχουν αναπτυχθεί με αυτό τον προσανατολισμό παρουσιάζονται περιληπτικά παρακάτω. Ο **Πίνακας 2.2** παρουσιάζει την ενδεικτική επερώτηση «*Αναζήτηση όλων των μηνυμάτων e-mail με συγκεκριμένο id μηνύματος*» εκφρασμένη στις γλώσσες επερώτησης που θα εξεταστούν. Ο ενδιαφερόμενος αναγνώστης μπορεί να αναζητήσει περισσότερες πληροφορίες και συγκριτική παρουσίαση των κυριότερων συστημάτων αποθήκευσης και επερώτησης RDF σχημάτων και περιγραφών στα [CPM+02] και [MKC+02].

- **ICS-FORTH RQL**: η RQL, που αναπτύχθηκε στα πλαίσια των ευρωπαϊκών προγραμμάτων C-Web (IST-1999-13479) και MesMuses (IST-2001- 26074), είναι μία δηλωτική γλώσσα επερώτησης βασιζόμενη σε ένα σύστημα τύπων η οποία ακολουθεί μία συναρτησιακή προσέγγιση τύπου OQL [KAC+02, K00]. Ορίζεται από ένα σύνολο βασικών επερωτήσεων και τελεστών σάρωσης (iterators) που μπορούν να χρησιμοποιηθούν συναρτησιακά για να δημιουργήσουν νέες επερωτήσεις. Εκτός του ότι προσαρμόζει τη λειτουργικότητα ημι-δομημένων ή XML γλωσσών επερώτησης στις ιδιαιτερότητες του RDF/S, την επεκτείνει επερωτώντας ομοιόμορφα τόσο περιγραφές πόρων όσο και (μετα)σχήματα.

Πιο συγκεκριμένα, η καινοτομία της RQL έγκειται στην ικανότητά της να συνδυάσει αρμονικά επερώτηση σε επίπεδο σχήματος και δεδομένων εξερευνώντας ταυτόχρονα –με διαφανή προς το χρήστη τρόπο- τις ιεραρχίες ετικετών και την πολλαπλή ταξινόμηση των πόρων. Συνεπώς, οι χρήστες μπορούν να επερωτήσουν πόρους σύμφωνα με το σχήμα που επιθυμούν, ανακαλύπτοντας συγχρόνως στη συνέχεια πώς οι ίδιοι πόροι περιγράφονται χρησιμοποιώντας κάποιο άλλο σχήμα. Η RQL έχει τη δυνατότητα να συνδυάσει μονοπάτια σχήματος για να εκτελέσει πιο πολύπλοκη πλοήγηση σχήματος, ένα είδος επερώτησης που δεν μπορεί να εκφραστεί από τις υπάρχουσες γλώσσες με δυνατότητες επερώτησης σχήματος. Επίσης, ένα από τα πλεονεκτήματα της γλώσσας είναι η υποστήριξη γενικευμένων εκφράσεων μονοπατιών (generalized path expressions) με μεταβλητές σε ετικέτες κόμβων (κλάσεων) και ακμών (ιδιοτήτων). Τέλος, υποστηρίζει επερωτήσεις με

συνολοθεωρητικούς τελεστές, XML Schema τύπους δεδομένων [XML99], συναρτήσεις συνάθροισης (aggregate) και ομαδοποίησης και αριθμητικές πράξεις επί τιμών δεδομένων. Συστήματα αποθήκευσης και επερώτησης που υποστηρίζουν τη γλώσσα RQL είναι τα **ICS-FORTH RDFSuite**[10] και **Sesame**[11], εκ των οποίων το δεύτερο υποστηρίζει υποσύνολο της γλώσσας.

- **ILRT SquishQL**: η SquishQL αποτελεί την πρόταση του ILRT[12] Semantic Web Research Group στην ανάγκη για μία γλώσσα επερώτησης για RDF βάσεις περιγραφών [MSR02]. Όντας μία απλή γλώσσα επερώτησης RDF γράφων, βασίζεται σε ένα μηχανισμό ταιριάσματος υπογράφων. Εκτός του ότι υποστηρίζει ένα μοντέλο επερώτησης βασισμένο σε μία αναπαράσταση γράφων σε τριάδες, εισάγει συναρτήσεις φιλτραρίσματος με τη μορφή Boolean εκφράσεων. Στην SquishQL υπάρχουν δύο κατηγορίες περιορισμών: *πρότυπα* (patterns) και *εκφράσεις φιλτραρίσματος*. Η γλώσσα προτύπων σχηματίζεται από τριάδες <subject,predicate,object> που περιγράφουν ακμές του γράφου και ένα τελεστή σύνδεσης. Για κάθε συστατικό της τριάδας, δηλαδή τα subject, predicate και object, επιτρέπει είτε μία μεταβλητή είτε μία τιμή, ενώ οι εκφράσεις φιλτραρίσματος περιορίζουν τις τιμές που μπορούν να πάρουν αυτές οι μεταβλητές. Συγκεκριμένα, το WHERE τμήμα μίας SquishQL επερώτησης προσδιορίζει το τμήμα του γράφου υπό επερώτηση ως μία λίστα από τριάδες και το AND τμήμα ορίζει την Boolean έκφραση που θα χρησιμοποιηθεί ως κριτήριο για να περιορίσει το πλήθος των αντικειμένων που επιστρέφονται ως απάντηση στην επερώτηση. Εντούτοις, η SquishQL δεν υποστηρίζει το μηχανισμό της υποστασιοποίησης και δεν μπορεί να εκφράσει μεταβατικότητα ή άλλες μορφές μονοπατιών αγνώστου μήκους.

Η SquishQL έχει αποτελέσει τη βάση ενός αριθμού RDF γλωσσών επερώτησης διαφορετικής πολυπλοκότητας. Μία γλώσσα παράγωγη της SquishQL είναι η RDQL, που αναπτύσσεται από την ομάδα Hewlett Packard Semantic Web Group [RDQL]. Η RDQL είναι μία σύνταξη και ένα API για επερώτηση της οποίας σκοπός είναι να λειτουργήσει ως μηχανισμός πρόσβασης σε επίπεδο υψηλότερο από ένα RDF API. Εξάγει πληροφορία από ένα RDF μοντέλο θεωρώντας το RDF ως δεδομένα και προσφέροντας επερωτήσεις με τριάδες και περιορισμούς πάνω σε ένα RDF μοντέλο, όπου ως μοντέλο θεωρείται ένα σύνολο από περιγραφές. Συστήματα διαχείρισης RDF/S δεδομένων που υποστηρίζουν την SquishQL είναι τα

---

[10] http://139.91.183.30:9090/RDF/index.html
[11] http://sesame.aidministrator.nl/
[12] Institute for Learning and Research Technology, University of Bristol (http://ilrt.org/)

Inkling[13], **RDFDB**[14] και **RDFStore**[15], ενώ το σύστημα **Jena**[16] υλοποιεί τη γλώσσα RDQL.

- **Intellidimension RDFQL**: Αναπτυσσόμενη από την εταιρία Intellidimension, η RDFQL είναι μία τύπου SQL, βασισμένη στο πρότυπο των δηλώσεων-τριάδων, γλώσσα επερώτησης [RDFQL]. Εκτός από την υποστήριξη λειτουργιών επερώτησης δομημένων RDF δεδομένων, ένα σημαντικό χαρακτηριστικό της RDFQL είναι η ικανότητά της να εξάγει συμπερασματικά νέες δηλώσεις από ήδη υπάρχουσες χρησιμοποιώντας συμπερασματικούς κανόνες ορισμένους από το χρήστη. Ένας RDFQL κανόνας αποτελείται από μία κεφαλίδα (head) και ένα σώμα (body). Η κεφαλίδα ορίζει τις τριάδες που θα δημιουργηθούν, εάν το σώμα του κανόνα αποτιμηθεί ως αληθές. Υπάρχουν δύο βασικοί τύποι κανόνων στην RDFQL: *κανόνες δηλώσεων* (statement rule) και *συναρτησιακοί κανόνες* (function rule). Ένας κανόνας δήλωσης, όταν εκτελείται, παράγει μία μοναδική δήλωση. Ως αποτέλεσμα, είναι δυνατόν να παραχθούν νέες ιδιότητες σε αντικείμενα βασιζόμενες στις ιδιότητες ενός ή περισσότερων άλλων αντικειμένων. Ένας συναρτησιακός κανόνας εκτελείται όταν καλείται από μία επερώτηση ή άλλο κανόνα. Οι συναρτησιακοί κανόνες είναι κατάλληλοι για την υλοποίηση αναδρομικών αλγορίθμων σε μία επερώτηση, όπως για παράδειγμα η επανάληψη πάνω σε μία δεντρική δομή.

Ως μία γλώσσα τύπου SQL, χρησιμοποιεί εντολές όπως INSERT, DELETE και SELECT για να διεξάγει λειτουργίες επερώτησης και εξαγωγής νέων δηλώσεων πάνω σε RDF τριάδες καθώς και εντολές ορισμού δεδομένων όπως CREATE TABLE ή CREATE DATABASE. Παραδείγματος χάριν, η εντολή SELECT είναι αντίστοιχη εκείνης της γλώσσας SQL με τη διαφορά ότι η RDFQL χρησιμοποιεί τον όρο USING αντί του FROM για να καθορίσει τις πηγές δεδομένων που θα χρησιμοποιηθούν στην επερώτηση, ενώ παράλληλα μπορούν να χρησιμοποιηθούν μεταβλητές στη λίστα των όρων που επιλέγονται, όπως αυτή ορίζεται στον SELECT όρο. Η RDFQL περιλαμβάνει τις περισσότερες SQL συναρτήσεις και τελεστές, που ορίζονται με τον όρο WHERE και δίνει τη δυνατότητα μέσω του όρου ORDER BY για διάταξη των αποτελεσμάτων ως προς μία κατεύθυνση διάταξης (αυξητική ή μειωτική). Επίσης, η RDFQL υποστηρίζει συναρτήσεις συνάθροισης, τελεστές σύγκρισης και τη χρήση ψευδωνύμων για κάθε χώρο ονοματοδοσίας

[13] http://swordfish.rdfweb.org/rdfquery/
[14] http://guha.com/rdfdb/
[15] http://rdfstore.sourceforge.net/
[16] http://www.hpl.hp.com/semweb/jena-top.html

χρησιμοποιώντας την εντολή NAMESPACE. Το σύστημα **RDF Gateway**[17] υποστηρίζει όλη τη λειτουργικότητα που προσφέρει η γλώσσα RDFQL.

- **RDFPath**: αν και βρίσκεται ακόμα υπό ανάπτυξη από την ομάδα RDFPath του Πανεπιστημίου Osnabrück και άλλους ενδιαφερόμενους φορείς, η RDFPath αποτελεί μία γλώσσα –παρόμοια με την XPath [CD99] για την XML- για τον εντοπισμό πληροφορίας κωδικοποιημένης σε RDF γράφους [Ko01]. Σκοπός της είναι να παρέχει γενικές τεχνικές για τον καθορισμό μονοπατιών ανάμεσα σε δύο τυχαίους κόμβους ενός RDF γράφου. Γενικά, μία RDFPath έκφραση είναι η σύνθεση μίας *πρωτεύουσας επιλογής* και αρκετών *βημάτων θέσης* και *φίλτρα*. Ειδικότερα, μία πρωτεύουσα επιλογή (primary selection) επιλέγει ένα αρχικό σύνολο κόμβων ενός δοθέντος RDF γράφου, π.χ οι εκφράσεις *resource()* και *literal()* επιλέγουν όλους τους πόρους και τα literals ενός RDF γράφου αντίστοιχα. Ένα βήμα θέσης (location step) καθορίζει ένα σύνολο κόμβων τους οποίους μπορούμε να προσπελάσουμε με ένα «βήμα» από ένα συγκεκριμένο κόμβο, π.χ η έκφραση *child()* επιλέγει τους άμεσους απογόνους ενός κόμβου, ενώ ένα φίλτρο (filter) επιλέγει υποσύνολα από ένα σύνολο δοθέντων αντικειμένων. Επομένως, κάθε πρωτεύουσα επιλογή ή βήμα θέσης επιλέγει ένα σύνολο αντικειμένων σχετικών με ένα συγκεκριμένο κόμβο και ο ρόλος των RDFPath φίλτρων είναι να εκλεπτύνει το σύνολο αυτό. Η RDFPath στοχεύει κυρίως σε RDF μεταδεδομένα, αλλά στις προθέσεις της ομάδας ανάπτυξης της γλώσσας είναι να υποστηρίξει RDF σχήματα και τις σχέσεις υπαλληλίας κλάσεων/ ιδιοτήτων του RDFS.

- **Versa RDF Query Language**: η Versa αποτελεί μία συνεχώς αναπτυσσόμενη, βασισμένη στο πρότυπο των γράφων γλώσσα επερώτησης RDF μοντέλων, που αρχικά αναπτύχθηκε από τους κατασκευαστές του συστήματος 4RDF[18][Og02]. Τα κυριότερα χαρακτηριστικά της συμπεριλαμβάνουν διάσχιση ακμών και επεξεργασία των περιεχομένων των κόμβων. Οι εκφράσεις (μεταβατικής) διάσχισης είναι ένα βασικό στοιχείο της Versa, εφόσον επιτρέπουν να ταιριάξουμε πρότυπα σε ένα μοντέλο σεβόμενοι τη δομή του ως κατευθυνόμενο γράφο, ενώ ταυτόχρονα επιτρέπουν διάσχιση και προς τις δύο κατευθύνσεις (εμπρός και πίσω). Σε μία διάσχιση προς τα εμπρός, μετακινούμαστε κατά μήκος των συγκεκριμένων ακμών από τα υποκείμενα στα αντικείμενα και κατόπιν γίνεται το φιλτράρισμα του αποτελέσματος, ενώ σε μία προς τα πίσω διάσχιση, ακολουθούμε την αντίθετη κατεύθυνση μετακινούμενοι από τα αντικείμενα προς τα υποκείμενα. Σε κάθε

[17] http://www.intellidimension.com/
[18] http://www.xml.com/pub/a/2000/10/11/rdf/index.html

περίπτωση, η φύση των αποτελεσμάτων της επερώτησης εξαρτάται από την υποκείμενη RDF υλοποίηση, η οποία καθορίζει τι θεωρείται πόρος για τους σκοπούς της Versa. Για παράδειγμα, παρουσία ενός RDF σχήματος, το αντικείμενο μίας δήλωσης μπορεί να θεωρηθεί ως πόρος στην περίπτωση που το κατηγόρημα της δήλωσης ορίζεται με πεδίο ορισμού. Γενικότερα, για να υποστηρίξει τη λειτουργικότητα που απαιτείται, η Versa παρέχει ενσωματωμένες συναρτήσεις, όπως η *all()* που ανακτά όλους τους πόρους του μοντέλου. Επίσης διαθέτει υπηρεσίες όπως λειτουργίες boolean λογικής και συνόλων, συνάθροισης, ταιριάσματος αλφαριθμητικών και άλλες βασικές λειτουργίες διαχείρισης τύπων δεδομένων. Επιπλέον, ο ερωτών μπορεί να σχηματίσει μία επερώτηση με πιο σύνθετα κριτήρια, π.χ να χρησιμοποιήσει την εντολή *all()-rdfs:label->contains("ab")*, αν επιθυμεί να ανακτήσει όλους τους πόρους των οποίων η ετικέτα περιέχει τους χαρακτήρες «ab», όπως επίσης και να παρουσιάσει τα αποτελέσματα μίας επερώτησης ταξινομημένα.

- **TRIPLE**: η γλώσσα TRIPLE είναι μία γλώσσα επερώτησης, εξαγωγής συμπερασμάτων και μετασχηματισμού RDF δεδομένων, που αναπτύχθηκε ως αποτέλεσμα της συνεργασίας ερευνητών από το Stanford University Database Group και το DFKI GmbH Kaiserslautern, Knowledge Management Department [SD01]. Βασιζόμενη στη λογική Horn και την F-Logic, επιχειρεί να υποστηρίξει με δηλωτικό τρόπο εφαρμογές που απαιτούν RDF συλλογιστική (reasoning) και μετασχηματισμό. Εντούτοις, σε αντίθεση με πολλές άλλες γλώσσες επερώτησης για το RDF, η TRIPLE επιτρέπει στη σημασιολογία γλωσσών που στηρίζονται στο RDF, όπως το RDFS και η Topic Maps[19], να οριστεί με κανόνες αντί να υποστηρίξει την ίδια λειτουργικότητα με ενσωματωμένη στη γλώσσα επερώτηση σημασιολογία. Στις περιπτώσεις εκείνες που ο ορισμός της σημασιολογίας της γλώσσας δεν είναι εύκολος με κανόνες, όπως για παράδειγμα η DAML+OIL [CHH+01], η TRIPLE παρέχει πρόσβαση σε εξωτερικά προγράμματα, όπως description logics ταξινομητές (classifiers). Επομένως, στην TRIPLE υπάρχουν δύο διαφορετικά είδη επιπέδων: συντακτικές επεκτάσεις της Horn λογικής για την υποστήριξη των βασικών RDF δομών, όπως πόροι και δηλώσεις, και τμήματα για τις σημασιολογικές επεκτάσεις του RDF, όπως το RDFS και η DAML+OIL, που υλοποιούνται είτε απευθείας στην TRIPLE είτε μέσω επικοινωνίας με εξωτερικά συστήματα συλλογιστικής, όπως ένας DL ταξινομητής. Συγκεκριμένα, η TRIPLE παρέχει υποστήριξη για πόρους και χώρους ονοματοδοσίας, συντομεύσεις (π.χ isa:=rdfs:subClassOf), μοντέλα (σύνολα RDF δηλώσεων), υποστασιοποίηση και κανόνες εκφράσιμους σε σύνταξη λογικής

[19] http://www.y12.doe.gov/sgml/sc34/document/0129.pdf

πρώτης τάξης. Η TRIPLE επιτρέπει επίσης Skolem συναρτήσεις, οι οποίες όταν χρησιμοποιηθούν σε κανόνες, μπορούν να μετατρέψουν ένα ή περισσότερα μοντέλα σε νέα. Συν τοις άλλοις, αντί για ορισμούς υποκειμένων, αντικειμένων και κατηγορημάτων, η TRIPLE επιτρέπει τη χρήση εκφράσεων μονοπατιών. Για παράδειγμα, μπορούμε να ορίσουμε (horn) κανόνες που αναζητούν έγγραφα με ένα καθορισμένο υποκείμενο. Τέλος, η TRIPLE διαθέτει μία ανθρωπίνως αναγνώσιμη ASCII σύνταξη, καθώς και μία σύνταξη βασισμένη στο RDF για την ανταλλαγή επερωτήσεων και κανόνων, πιθανώς μεταξύ επικοινωνούντων πρακτόρων. Το ομώνυμο σύστημα **TRIPLE**[20] υποστηρίζει όλη τη λειτουργικότητα της γλώσσας επερώτησης.

RQL	SELECT x FROM {x} email:message_id {id} WHERE id = "aimilia@examplenet.com" USING NAMESPACE email = http://www.hpl.hp.com/email/email#
RDQL	SELECT ?x WHERE (?x, <email:message_id>, 'aimilia@examplenet.com') USING email FOR <http://www.hpl.hp.com/email/email#>
SquishQL	SELECT ?x FROM http://test1/test WHERE (email::message_id ?x aimilia@examplenet.com) USING email FOR http://www.hpl.hp.com/email/email#
RDFQL	session.namespaces["pop3"] = "http://www.hpl.hp.com/email/email#"; session.namespaces["rdf"] = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"; session.namespaces["rdfs"] = "http://www.w3.org/2000/01/rdf-schema#"; var ds = new DataSource("pop3?server=mail.somewhere.net&user=someuser &pass=xxxx&action=list"); select ?msg ?from using #ds where {[rdf:type] ?msg [pop3:Message]} and {[pop3:property] ?msg ?prop} and {[rdfs:label] ?prop 'From'} and {[rdf:value] ?prop ?from} and regexp(?from, 'aimilia@examplenet.com')<>''
RDFPath	resource()[child(email:message_id)="aimilia@examplenet.com"]
VERSA	all() |- email:message_id -> eq('aimilia@examplenet.com')
TRIPLE	email:="http://www.hpl.hp.com/email/email#" FORALL C<- C[email:message_id -> aimilia@examplenet.com]@testbase

Πίνακας 2.2: Ενδεικτική επερώτηση RDF/S γλωσσών επερώτησης

2.3.1 Σύγκριση των γλωσσών επερώτησης

Οι αρχές στις οποίες βασίζεται το RDF/S είναι σημαντικά διαφορετικές από εκείνες των παραδοσιακών μοντέλων βάσεων δεδομένων όπως το οντοκεντρικό, σχεσιακό και ημι-δομημένων δεδομένων, γεγονός που δικαιολογεί την ύπαρξη γλωσσών επερώτησης προσαρμοσμένων στις ιδιαιτερότητές του. Έχοντας ήδη περιγράψει συνοπτικά τις κυριότερες γλώσσες επερώτησης για το RDF/S, επιχειρούμε να

[20] http://triple.semanticweb.org/

συγκρίνουμε την εκφραστική τους δύναμη βάσει ενός συνόλου κριτηρίων. Ο **Πίνακας 2.3** φιλοξενεί μία σύγκριση των βασικών χαρακτηριστικών των γλωσσών, ενώ ο **Πίνακας 2.4** παρουσιάζει μία πιο σύνθετη σύγκριση επί συνόλου κριτηρίων [CPM+02,MKC+02].

Ειδικότερα, στον πρώτο πίνακα αναφέρεται το *μοντέλο δεδομένων* που χρησιμοποιείται για την αναπαράσταση των RDF/S μεταδεδομένων, η *γλώσσα προέλευσης* πάνω στην οποία στηρίζεται η γλώσσα επερώτησης, η ικανότητά της να υποστηρίζει συναρτησιακή σύνθεση επερωτήσεων (*κλειστότητα*, τα αποτελέσματα μίας επερώτησης μπορούν να χρησιμοποιηθούν ως είσοδος σε μία άλλη επερώτηση), αν η γλώσσα επιτρέπει κάθε είδους δεδομένα ως είσοδο και έξοδο των επερωτήσεων (*ορθογωνικότητα*) και η *γενικότητά* της (η δυνατότητα να εκμεταλλευτεί όλα τα στοιχεία του RDF/S). Όπως παρατηρούμε, οι περισσότερες γλώσσες επερώτησης στηρίζονται σε ένα μοντέλο δεδομένων βασισμένο σε τριάδες, δηλαδή χρησιμοποιούν συλλογές δηλώσεων της μορφής <υποκείμενο, κατηγόρημα, αντικείμενο> για να κωδικοποιήσουν RDF/S βάσεις περιγραφών και σχήματα. Η RQL και η Versa ακολουθούν το μοντέλο των γράφων, το οποίο τους επιτρέπει να εκτελούν πιο πολύπλοκες επερωτήσεις πάνω σε γράφους περιγραφής πόρων. Η RDFPath, από την άλλη πλευρά, υιοθετεί ένα μοντέλο δέντρων και οι δομές της ορίζουν μονοπάτια ανάμεσα σε 2 τυχαίους κόμβους ενός RDF γράφου.

ΚΡΙΤΗΡΙΑ / ΓΛΩΣΣΑ.	Μοντέλο δεδομένων	Γλώσσα προέλευσης	Κλειστότητα	Ορθογωνικότητα δεδομένων εισόδου/εξόδου	Γενικότητα
RQL	Γράφοι	OQL τύπου	ΝΑΙ	ΝΑΙ	ΟΧΙ
SquishQL/RDQL	Τριάδες	SQL τύπου	ΟΧΙ	ΟΧΙ	ΟΧΙ
RDFQL	Τριάδες	SQL τύπου	ΟΧΙ	ΟΧΙ	ΟΧΙ
RDFPath	Δέντρα	Xpath τύπου	ΟΧΙ	ΟΧΙ	ΟΧΙ
VERSA	Γράφοι	LISP τύπου	ΝΑΙ	ΟΧΙ	ΟΧΙ
TRIPLE	Τριάδες	F-logic τύπου	ΟΧΙ	ΟΧΙ	ΟΧΙ

Πίνακας 2.3: Βασικά χαρακτηριστικά RDF/S γλωσσών επερώτησης

Τα μοντέλα δεδομένων που στηρίζονται σε τριάδες και δέντρα απαιτούν ακριβή γνώση της δομής ενός RDF γράφου στην επερώτηση, γνώση που δεν είναι απαραίτητη στα περισσότερα μοντέλα δεδομένων βασισμένα στους γράφους. Επιπλέον, η TRIPLE βασίζεται σε ένα μοντέλο στηριγμένο στη λογική, το οποίο της δίνει τη δυνατότητα ορισμού συμπερασματικών κανόνων. Μερική υποστήριξη συμπερασματικών κανόνων ορισμένων από το χρήστη προσφέρει και η RDFQL, π.χ

για να δηλώσει ιδιότητες των οποίων η διάσχιση μπορεί να γίνει μεταβατικά. Οι υπόλοιπες γλώσσες ακολουθούν μία προσέγγιση τύπου SQL, κυρίως με σκοπό να προσφέρουν δομές και λειτουργικότητα γνωστές σε σχεδιαστές βάσεων δεδομένων.

Όσο αφορά τις άλλες ιδιότητες των γλωσσών επερώτησης, η RQL είναι η μόνη γλώσσα που πληροί τις ιδιότητες της κλειστότητας και ορθογωνικότητας, ενώ η Versa πληροί μόνο την κλειστότητα. Όλες οι γλώσσες, με εξαίρεση την RQL, εκχωρούν προνόμια σε ορισμένα είδη εισόδου ή εξόδου. Για παράδειγμα, οι γλώσσες που ακολουθούν το μοντέλο των τριάδων, π.χ η SquishQL/RDQL και η RDFQL, λαμβάνουν ένα σύνολο από τριάδες ως είσοδο και επιστρέφουν ένα σύνολο/λίστα από πόρους ή literal τιμές. Σε αντίθεση, δεν υπάρχουν προνομιακά είδη δεδομένων εισόδου και εξόδου για την RQL, εφόσον είσοδος μπορεί να είναι μία κλάση, μία ιδιότητα, ένας πόρος ή μία τιμή, κτλ. Η RQL όμως, όπως και όλες οι γλώσσες) στερείται της ιδιότητας *γενικότητα*, καθότι δεν υποστηρίζει το σύνολο των χαρακτηριστικών του RDF/S (δεν υποστηρίζει υποστασιοποίηση). Το τελευταίο αυτό κριτήριο αποτελεί τη βάση της σύγκρισης που γίνεται στον Πίνακα 2.4, η οποία είναι πιο λεπτομερής και ουσιαστικά συγκρίνει την εκφραστική δύναμη των γλωσσών.

Η σύγκριση της εκφραστικής δύναμης των γλωσσών που παρουσιάστηκαν στην προηγούμενη παράγραφο γίνεται βάσει κριτηρίων που διακρίνονται σε 5 κύριες κατηγορίες: **Μηχανισμοί Μοντελοποίησης**, **Επερώτηση Σχήματος**, **Επερώτηση Δεδομένων**, **Επερώτηση Σχήματος/Δεδομένων** και **Επιπλέον Χαρακτηριστικά**. Τα κριτήρια της πρώτης κατηγορίας αναφέρονται στην ικανότητα της γλώσσας να υποστηρίξει στοιχεία του RDF/S, ειδικότερα αν δύναται να επερωτήσει κλάσεις, ιδιότητες και πόρους από διαφορετικούς *χώρους ονοματοδοσίας/πολλαπλά σχήματα*, αν υποστηρίζει *τύπους δεδομένων*, *συλλογές τιμών*, *υποστασιοποίηση*, και αν εκμεταλλεύεται τους μηχανισμούς της *πολλαπλής κληρονομικότητας* και *ταξινόμησης*. Σύμφωνα με αυτά τα κριτήρια, παρατηρούμε από τον Πίνακα 2.4 ότι όλες οι γλώσσες υποστηρίζουν επερώτηση σε χώρους ονοματοδοσίας και πολλαπλά σχήματα, καθώς και πολλαπλή ταξινόμηση και εξειδίκευση. Εντούτοις, δεν υποστηρίζουν όλες οι γλώσσες υποστασιοποίηση και συλλογές τιμών (container values). Στην πραγματικότητα, η υποστασιοποίηση υποστηρίζεται μόνο από την TRIPLE. Επιπλέον, οι περισσότερες γλώσσες επερώτησης υποστηρίζουν τους βασικούς τύπους δεδομένων, όπως αλφαριθμητικά και ακεραίους, με εξαίρεση τις RQL, VERSA και RDFQL που υποστηρίζουν πιο περίπλοκους τύπους δεδομένων, όπως πραγματικούς αριθμούς, ημερομηνίες, URI ή απαριθμητούς τύπους.

	ΓΛΩΣΣΕΣ / ΚΡΙΤΗΡΙΑ	RQL	SquishQL	RDFQL	RDFPath	VERSA	TRIPLE
Μηχανισμοί Μοντελοποίησης	Χώροι ονοματοδοσίας/ πολλαπλά σχήματα	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ
	Τύποι δεδομένων	αλφαριθμητικά, URI, ακέραιοι, πραγματικοί, ημερομηνίες, απαριθμητοί τύποι, τύποι θησαυρών	αλφαριθμητικά και ακέραιοι	Αλφαριθμητικά, ακέραιοι, πραγματικοί, ημερομηνίες και τύποι URI	Αλφαριθμητικά	αλφαριθμητικά, URI, αριθμοί, λίστες, σύνολα, booleans	αλφαριθμητικά, ακέραιοι
	Πολλαπλή κληρονομικότητα/ Πολλαπλή ταξινόμηση	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ
	Συλλογές τιμών	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΟΧΙ(?)	ΟΧΙ
	Υποστασιοποίηση	ΟΧΙ	ΟΧΙ	ΟΧΙ	ΟΧΙ	ΟΧΙ	ΝΑΙ
Επερώτηση Σχήματος	Διάσχιση προγόνων/απογόνων σε ιεραρχίες κλάσεων/ιδιοτήτων	ΝΑΙ	ΟΧΙ (μόνο άμεσα)	ΟΧΙ (μόνο άμεσα)	ΟΧΙ (μόνο άμεσα)	ΟΧΙ (μόνο άμεσα)	ΝΑΙ
	Συνθήκες φιλτραρίσματος σε ιεραρχίες κλάσεων/ιδιοτήτων	(αν)ισότητα, like, έλεγχος υπαλληλίας, επερώτηση χώρων ονοματοδοσίας	like (~), ισότητα	Λεξικογραφική διάταξη στα ονόματα κλάσεων και ιδιοτήτων, ισότητα	Ισότητα	(αν)ισότητα, έλεγχος εγκλεισμού ενός αλφαριθμητικού σε άλλο	(αν)ισότητα, έλεγχος υπαλληλίας
Επερώτηση Δεδομένων	Επερωτήσεις εύρεσης έκτασης κλάσεων/ιδιοτήτων	ΝΑΙ	ΝΑΙ (μόνο άμεσα)	ΝΑΙ (μόνο άμεσα)	ΝΑΙ (μόνο άμεσα)	ΝΑΙ	ΝΑΙ
	Πλήρη Boolean φίλτρα (άρνηση, διάζευξη, σύζευξη)	ΝΑΙ	ΟΧΙ (μόνο σύζευξη)	ΝΑΙ	ΟΧΙ (μόνο σύζευξη)	ΝΑΙ	ΝΑΙ
	Λειτουργίες με τελεστές συνόλων	ΝΑΙ	ΟΧΙ	ΟΧΙ	ΟΧΙ	ΝΑΙ	ΝΑΙ
	Αριθμητικές λειτουργίες	ΝΑΙ	ΟΧΙ	ΟΧΙ	ΟΧΙ	ΟΧΙ	ΟΧΙ
	Δομές για κατασκευή συλλογών τιμών	ΝΑΙ	ΟΧΙ	ΟΧΙ	ΟΧΙ	ΝΑΙ	ΟΧΙ
Επερώτηση Σχήματος/ Δεδομένων	Εκφράσεις γενικευμένων μονοπατιών	ΝΑΙ	ΟΧΙ	ΟΧΙ	ΟΧΙ	ΟΧΙ	ΟΧΙ
	Υπαρξιακοί/ Καθολικοί ποσοδείκτες	ΝΑΙ	ΟΧΙ	ΟΧΙ	ΟΧΙ	ΟΧΙ	ΝΑΙ
	Φωλιασμένες επερωτήσεις	ΝΑΙ	ΟΧΙ	ΟΧΙ	ΟΧΙ	ΝΑΙ	ΟΧΙ
Επιπλέον Χαρακτηριστικά	Συναρτήσεις συνάθροισης	ΝΑΙ	ΟΧΙ	ΝΑΙ (μόνο count)	ΟΧΙ	ΝΑΙ	ΟΧΙ
	Συναρτήσεις ομαδοποίησης	ΟΧΙ	ΟΧΙ	ΟΧΙ	ΟΧΙ	ΟΧΙ	ΟΧΙ
	Συναρτήσεις ταξινόμησης	ΟΧΙ	ΟΧΙ	ΝΑΙ	ΟΧΙ	ΝΑΙ	ΟΧΙ
	Ενσωματωμένες συναρτήσεις δεδομένων	ΝΑΙ (όροι θησαυρών)	ΟΧΙ	ΝΑΙ (μαθηματικές/αλφαριθμητικές/ημερομηνίας)	ΟΧΙ	ΝΑΙ (συναρτήσεις μετατροπής)	ΟΧΙ
	Υποστήριξη αυθαίρετων συναρτήσεων	ΟΧΙ	ΟΧΙ	ΝΑΙ	ΟΧΙ	ΟΧΙ	ΟΧΙ
	Ορισμένοι από το χρήστη συμπερασματικοί κανόνες	ΟΧΙ	ΟΧΙ	ΝΑΙ	ΟΧΙ	ΟΧΙ	ΝΑΙ
	Μηχανισμός ορισμού όψεων	ΟΧΙ	ΟΧΙ	ΟΧΙ	ΟΧΙ	ΟΧΙ	ΟΧΙ

Πίνακας 2.4: Εκφραστική δύναμη των RDF/S γλωσσών επερώτησης

Ο δεύτερος άξονας σύγκρισης αναφέρεται σε κριτήρια που δηλώνουν αν η γλώσσα επερώτησης μπορεί να εκμεταλλευτεί την ύπαρξη σχήματος. Ειδικότερα, αν δύναται να εκτελέσει *διάσχιση προγόνων/απογόνων σε ιεραρχίες κλάσεων/ιδιοτήτων*[21] και ποιες *συνθήκες φιλτραρίσματος* μπορεί να επιβάλλει *στις ιεραρχίες κλάσεων/ ιδιοτήτων*. Η ικανότητα διάσχισης των ιεραρχιών κλάσεων και ιδιοτήτων απαιτεί γνώση του σχήματος που υιοθετείται και μόνο η RQL και η TRIPLE μπορούν να την εκτελέσουν μεταβατικά. Επιπλέον, μόνο η RQL μπορεί να θέσει εκλεπτυσμένες συνθήκες φιλτραρίσματος στις ιεραρχίες των κλάσεων και ιδιοτήτων. Εκτός από συνθήκες (αν)ισότητας και ομοιότητας (like) πάνω στα ονόματα των κλάσεων και ιδιοτήτων, η RQL μπορεί επίσης να θέσει φίλτρα επιλογής βασισμένα σε ελέγχους υπαλληλίας, δηλαδή ελέγχους πάνω στις σχέσεις υπαλληλίας ανάμεσα στις κλάσεις/ιδιότητες, όπως επίσης και φίλτρα επί των χώρων ονοματοδοσίας στους οποίους ορίζονται οι όροι του σχήματος.

Το βασικό κριτήριο της κατηγορίας *Επερώτηση Δεδομένων* είναι η δυνατότητα της γλώσσας να υπολογίσει την *έκταση των κλάσεων/ιδιοτήτων*, με άλλα λόγια να βρει όλους τους πόρους που ορίζονται ως στιγμιότυπα μίας συγκεκριμένης κλάσης/ιδιότητας. Στην ίδια κατηγορία εμπίπτουν κριτήρια, όπως η υποστήριξη *Boolean φίλτρων*, *λειτουργιών με τελεστές συνόλων*, *αριθμητικών λειτουργιών* και *δομών για την κατασκευή συλλογών τιμών* (ακολουθίες (sequence) και μη διατεταγμένες συλλογές (bag)). Όπως παρατηρούμε από τον πίνακα, όλες οι γλώσσες είναι σε θέση να υπολογίσουν την έκταση μίας κλάσης/ιδιότητας είτε έμμεσα είτε άμεσα. Αυτό που δεν υποστηρίζουν οι περισσότερες γλώσσες είναι λειτουργίες με τελεστές συνόλων (ένωση, τομή, διαφορά) καθώς και αριθμητικές λειτουργίες πάνω σε τιμές δεδομένων. Το πρώτο γεγονός αποδίδεται κυρίως στο ότι δεν υποστηρίζουν δομές για την κατασκευή συλλογών τιμών (μόνο η RQL και η Versa διαθέτουν αυτό το χαρακτηριστικό). Επιπλέον, πλήρη Boolean φίλτρα υποστηρίζονται από τις RQL, RDFQL, VERSA και TRIPLE, εφόσον οι υπόλοιπες γλώσσες υποστηρίζουν μόνο σύζευξη.

Η κατηγορία *Επερώτηση Σχήματος/Δεδομένων* περιέχει κριτήρια αναφερόμενα στην ικανότητα της γλώσσας να συνδυάσει επερώτηση σχήματος και δεδομένων σε μία επερώτηση. Ο βασικός τρόπος για να εκτελέσουμε ένα τέτοιο

---

[21] Η διάσχιση προγόνων/απογόνων μίας ιεραρχίας κλάσεων/ιδιοτήτων μπορεί να θεωρηθεί ως μία μορφή *δομικής αναδρομής* (structural recursion), δηλαδή μία συγκεκριμένη μορφή αναδρομής που είναι πιο απλή να υλοποιηθεί από οποιεσδήποτε άλλες αυθαίρετες αναδρομικές συναρτήσεις. Επίσης είναι μία από τις διαδικασίες που συνιστούν τη λειτουργία της εξαγωγής συμπερασμάτων (infernence) στα πλαίσια του RDF/S.

σύνθετο ερώτημα είναι να χρησιμοποιήσουμε *γενικευμένες εκφράσεις μονοπατιών*. Μία γενικευμένη έκφραση μονοπατιού επερωτά δεδομένα και σχήμα ταυτόχρονα, επιτρέποντας με αυτόν τον τρόπο το σχηματισμό επερωτήσεων χωρίς να απαιτείται ακριβής γνώση των σχημάτων που χρησιμοποιούνται. Οι γενικευμένες εκφράσεις μονοπατιών σε συνδυασμό με την υποστήριξη *υπαρξιακών και καθολικών τελεστών* και *φωλιασμένων επερωτήσεων* είναι ενδεικτικά κριτήρια της εκφραστικής δύναμης της γλώσσας επερώτησης. Κρίνοντας βάσει αυτών των κριτηρίων, η RQL είναι η μόνη γλώσσα που είναι σε θέση να ενσωματώσει γνώση από σχήματα στην επερώτηση δεδομένων. Μάλιστα, υποστηρίζει γενικευμένες εκφράσεις μονοπατιών με μεταβλητές στις ετικέτες κόμβων και ακμών. Η TRIPLE και η RDFPath υποστηρίζουν μόνο μονοπάτια δεδομένων, ενώ η Versa μπορεί να διασχίσει μεταβατικά τις ιδιότητες. Συν τοις άλλοις, η RQL και η TRIPLE είναι οι μόνες γλώσσες επερώτησης που υποστηρίζουν καθολικούς και υπαρξιακούς τελεστές, ενώ μόνο η RQL και η Versa μπορούν να εκτελέσουν φωλιασμένες επερωτήσεις.

Η τελευταία κατηγορία φιλοξενεί ένα αριθμό από κριτήρια που μπορούν να χαρακτηρίσουν την αποτελεσματικότητα της γλώσσας επερώτησης και την προστιθέμενη αξία της χρήσης της από πραγματικές, ευρείας κλίμακας εφαρμογές. Η υποστήριξη συναρτήσεων *συνάθροισης* (min, max, average, sum, count), *ομαδοποίησης* (όρων αντίστοιχων με τον SQL όρο group_by), *ταξινόμησης* (για τη διάταξη των αποτελεσμάτων της επερώτησης) και *ενσωματωμένων συναρτήσεων δεδομένων* (π.χ μαθηματικές, αλφαριθμητικές συναρτήσεις μετατροπής), σε συνδυασμό με *ορισμένους από το χρήστη συμπερασματικούς κανόνες* είναι χαρακτηριστικά που ενδυναμώνουν την εκφραστικότητα μίας γλώσσας. Στην τελευταία αυτή κατηγορία μπορούμε να προσθέσουμε την ύπαρξη *μηχανισμού για του ορισμό όψεων* πάνω σε βάσεις περιγραφών. Από τα δεδομένα του Πίνακα 2.4, η RDFQL υποστηρίζει μόνο τη συνάρτηση count, η Versa τις συναρτήσεις min και max, ενώ η RQL προσφέρει ένα πιο ευρύ σύνολο συναρτήσεων υποστηρίζοντας τις συναρτήσεις min, max, count, average και sum, όπως αυτές είναι γνωστές από τις σχεσιακές βάσεις δεδομένων. Οι συναρτήσεις ταξινόμησης υποστηρίζονται μόνο από τις RDFQL και Versa, οι οποίες επίσης υποστηρίζουν ενσωματωμένες συναρτήσεις δεδομένων, π.χ για την μετατροπή ημερομηνιών και αλφαριθμητικών τιμών ή τον υπολογισμό αριθμητικών τιμών. Επιπλέον, η RDFQL, εκτός του ότι παρέχει δυνατότητες ορισμού δεδομένων (π.χ δημιουργία ενός πίνακα), επιτρέπει τον ορισμό από το χρήστη συμπερασματικών κανόνων και αυθαίρετων συναρτήσεων. Ορισμένοι από το χρήστη συμπερασματικοί κανόνες υποστηρίζονται επίσης από την TRIPLE και μπορούν να

χρησιμοποιηθούν, για παράδειγμα, για να διασχίσουν αντίστροφα μονοπάτια ιδιοτήτων. Μία λειτουργικότητα που δεν υποστηρίζεται από καμία γλώσσα επερώτησης είναι η ομαδοποίηση των αποτελεσμάτων μίας επερώτησης.

Όπως παρατηρείται στην τελευταία σειρά του Πίνακα 2.4, ο μηχανισμός ορισμού όψεων δεν υποστηρίζεται από καμία γλώσσα επερώτησης. Αν και η χρησιμότητα ενός μηχανισμού ορισμού όψεων έχει διαπιστωθεί στις σχεσιακές και οντοκεντρικές βάσεις δεδομένων, στον κόσμο του RDF/S πρέπει να εξεταστεί η χρησιμότητα και η λειτουργικότητα ενός τέτοιου μηχανισμού σε εφαρμογές Διαδικτύου, όπου το μοντέλο δεδομένων που υιοθετείται είναι διαφορετικό. Αντικείμενο του επόμενου κεφαλαίου είναι να παρουσιάσει, σε αντιπαράθεση με το σχεσιακό και οντοκεντρικό μοντέλο δεδομένων, τη χρησιμότητα και τη λειτουργικότητα που πρέπει να υποστηρίζει ένας μηχανισμός ορισμού όψεων στα πλαίσια του Σημασιολογικού Ιστού. Η σύγκριση των γλωσσών επερώτησης που έγινε στην ενότητα αυτή συνηγορεί στην απόφασή μας να στηριχτούμε στη γλώσσα RQL, λόγω της εκφραστικής της δύναμης που υπερτερεί των άλλων γλωσσών επερώτησης και της ικανότητά της για επερωτήσεις σε επίπεδο σχήματος. Ο μηχανισμός όψεων που προτείνουμε θα παρουσιαστεί σε επόμενο κεφάλαιο, όπου θα τεκμηριωθεί και η ευελιξία του κυρίως λόγω της RQL, που αποδεικνύεται ένα ισχυρό εργαλείο για την πλοήγηση και αναζήτηση δεδομένων σε σύνθετους γράφους.

ΚΕΦΑΛΑΙΟ 3

Ο Μηχανισμός Όψεων σε Συστήματα Βάσεων Δεδομένων

Είναι σύνηθες φαινόμενο της καθημερινής πρακτικής, ξεχωριστές εφαρμογές που έχουν μοντελοποιηθεί με διαφορετικό τρόπο, να χρειάζονται τα ίδια δεδομένα για να λειτουργήσουν. Η απλοϊκή λύση της αντιγραφής των δεδομένων και της μοντελοποίησης που ακολουθήθηκε είναι δαπανηρή σε όρους αποθήκευσης, συντήρησης των δεδομένων και επεκτασιμότητας. Ο **μηχανισμός των όψεων** είναι μία τεχνική μοντελοποίησης των δεδομένων με διαφορετικό τρόπο για ποικίλες εφαρμογές, που δεν επηρεάζει τον τύπο και τη δομή των αποθηκευμένων δεδομένων. Η τεχνική αυτή επιτρέπει σε εφαρμογές να προσαρμόζουν στις απαιτήσεις τους καταμερισμένα δεδομένα, χωρίς να επηρεάζουν τη λειτουργία άλλων εφαρμογών που κάνουν χρήση του ίδιου συνόλου πληροφορίας. Δεδομένων των αυξανόμενων αναγκών διαχείρισης δεδομένων που τίθονται από τις επιχειρήσεις και που δεν μπορούν να ικανοποιηθούν από τα υπάρχοντα συστήματα διαχείρισης δεδομένων, ο μηχανισμός των όψεων καθίσταται σημαντικός ως ένας τρόπος μοντελοποίησης και χρήσης δεδομένων σε νέες εφαρμογές [GM99].

Έχοντας ήδη παρουσιάσει συνοπτικά στο προηγούμενο κεφάλαιο το RDF/S και τις γλώσσες που έχουν αναπτυχθεί για την επερώτηση RDF/S γράφων, στην ενότητα αυτή παρουσιάζουμε τη θεωρητική βάση του μηχανισμού των όψεων και τον τρόπο με τον οποίο μπορεί να χρησιμοποιηθεί στο περιβάλλον του Σημασιολογικού Ιστού. Ειδικότερα, επιχειρώντας μία ανασκόπηση στη θεωρία των όψεων, θα παρουσιαστούν οι βασικές έννοιες που διέπουν το μηχανισμό ορισμού όψεων, τόσο κατά το σχεσιακό όσο και το οντοκεντρικό μοντέλο δεδομένων, που ομοιάζει σε πολλά σημεία με το μοντέλο δεδομένων του RDF/S, αλλά και σύμφωνα με το μοντέλο των ημιδομημένων δεδομένων και XML που υιοθετείται σε εφαρμογές διαδικτύου. Στη συνέχεια, θα σκιαγραφηθεί η λειτουργικότητα που πρέπει να υποστηρίζει ένας μηχανισμός ορισμού όψεων στα πλαίσια του Σημασιολογικού Ιστού.

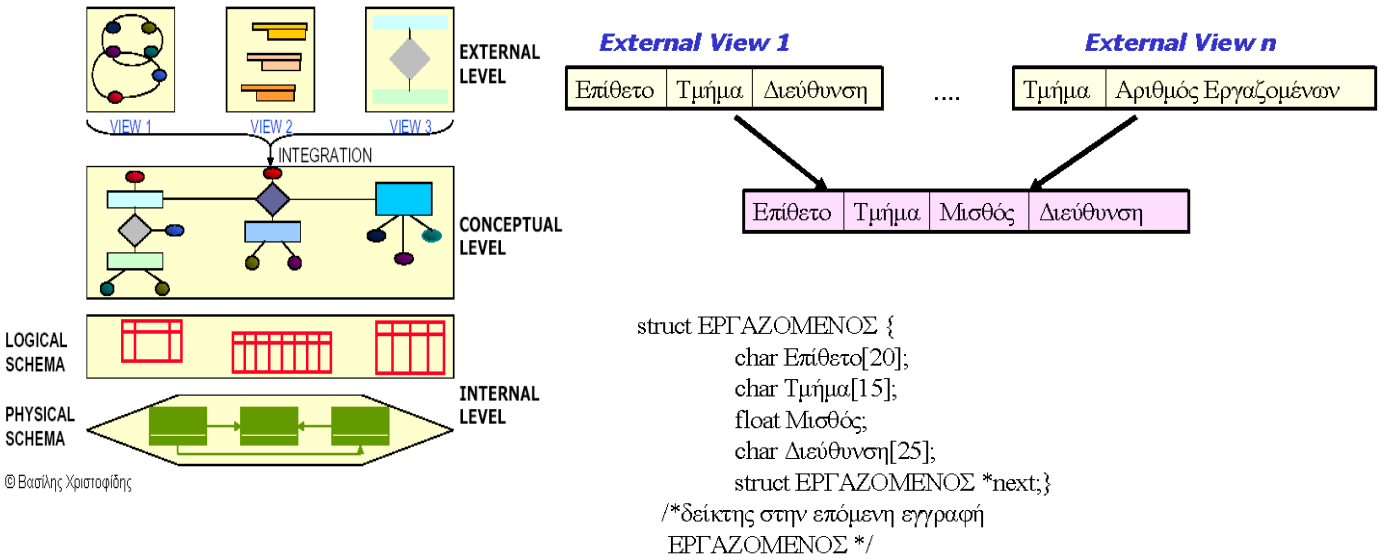
3.1 Η τεχνολογία των όψεων

Η υποστήριξη μηχανισμών ορισμού όψεων πάνω σε πληροφοριακές πηγές έχει αναγνωριστεί ως μία επιπρόσθετης αξίας λειτουργικότητα της εκφραστικής δύναμης μίας γλώσσας επερώτησης. Ο μηχανισμός ορισμού όψεων επιτρέπει την οργάνωση (μέρους) μίας βάσης δεδομένων σε λογικές μονάδες, δομώντας τη βάση δεδομένων, ή τμήματα αυτής, με τρόπο κατάλληλο προς τις απαιτήσεις μίας συγκεκριμένης εφαρμογής. Στα πλαίσια του Σημασιολογικού Ιστού, που χαρακτηρίζεται από τη διασπορά και ποικιλομορφία των πληροφοριακών πηγών, οι χρήστες του ίδιου υποσυνόλου πληροφορίας θέτουν απαιτήσεις εξατομικευμένης πρόσβασης στις πληροφοριακές πηγές και συνεπώς απαιτούν την ύπαρξη μηχανισμών λογικής ομαδοποίησης των δεδομένων, όπως είναι οι όψεις. Η ανάγκη ύπαρξης ενός τέτοιου μηχανισμού γίνεται πιο επιτακτική εάν λάβουμε υπόψη μας την ποικιλομορφία των αναπαραστάσεων που χρησιμοποιούν οι χρήστες και οι προμηθευτές πληροφορίας για τη μοντελοποίηση, ακόμα και του ίδιου, πεδίου εφαρμογής.

Δεδομένης της χρησιμότητας ενός μηχανισμού ορισμού όψεων, στην ενότητα αυτή παρουσιάζουμε το θεωρητικό υπόβαθρο της τεχνικής των όψεων κατά το σχεσιακό, οντοκεντρικό, ημιδομημένων δεδομένων και XML μοντέλο, αναφέροντας συνοπτικά τα βασικά στοιχεία του υποκείμενου μοντέλου δεδομένων (του τυπικού συστήματος που έχει χαρακτηριστικά τα οποία αφορούν τη δομή, την ακεραιότητα και το χειρισμό των δεδομένων). Εκτός από τα διαφορετικά μοντέλα δεδομένων και τις γλώσσες επερώτησης που χρησιμοποιούνται, οι γλώσσες ορισμού όψεων διαφέρουν και ως προς το σύνολο των λειτουργιών που στοχεύουν να υποστηρίξουν. Σκοπός της ενότητας είναι να παράσχει μία εποπτική εικόνα της τεχνικής των όψεων και της χρησιμότητας τους σε εφαρμογές βάσεων δεδομένων και διαδικτύου. Για να ολοκληρωθεί η εποπτική παρουσίαση του μηχανισμού όψεων, θα αναφερθούν συνοπτικά θέματα που άπτονται της διαχείρισης των όψεων και που χρήζουν αντιμετώπισης ανεξάρτητα από το υποκείμενο μοντέλο δεδομένων.

3.1.1 Βασικές έννοιες και χρησιμότητα των όψεων

Ο μηχανισμός ορισμού όψεων σχετίζεται με την *αρχιτεκτονική τριών επιπέδων ANSI/SPARC* [A75]. Πρόκειται για ένα μοντέλο αρχιτεκτονικής βάσεων δεδομένων,



Εικόνα 3.1. Αρχιτεκτονική βάσεων δεδομένων ANSI/SPARC

το οποίο απαρτίζεται από τα εξής τρία επίπεδα: *εξωτερικό επίπεδο* (external level) ή *επίπεδο όψεων* (view level), *εννοιολογικό επίπεδο* (conceptual level) και *εσωτερικό επίπεδο* (internal level). Η **Εικόνα 3.1** παρουσιάζει εποπτικά τη διάκριση των επιπέδων (αριστερό μέρος) και δίνει παράδειγμα αυτών, όπου γίνεται εμφανής η διαφορά ανάμεσα στα τρία επίπεδα (δεξιό μέρος).

Ειδικότερα, το εσωτερικό επίπεδο αντιστοιχεί στη φυσική αναπαράσταση της βάσης δεδομένων στο υπολογιστικό σύστημα και περιγράφει τον τρόπο με τον οποίο τα δεδομένα αποθηκεύονται φυσικά. Συνίσταται από το *λογικό σχήμα*, που αποτελεί τη λογική οργάνωση της πληροφορίας χρησιμοποιώντας το μοντέλο δεδομένων του συστήματος διαχείρισης βάσεων δεδομένων (π.χ πίνακες) και το *φυσικό σχήμα*, που αποτελεί τη φυσική οργάνωση των δεδομένων στη δευτερεύουσα μνήμη (π.χ αρχεία, εγγραφές). Το εννοιολογικό επίπεδο αναφέρεται στην όψη της *κοινότητας χρηστών* που χρησιμοποιεί τη βάση δεδομένων. Ουσιαστικά πρόκειται για τη μοντελοποίηση των δεδομένων του πραγματικού κόσμου που -όντας ανεξάρτητη από το υποκείμενο σύστημα διαχείρισης βάσεων δεδομένων (DBMS)- περιγράφει τα αποθηκευμένα δεδομένα στη βάση δεδομένων και τις σχέσεις μεταξύ αυτών. Ένα σύστημα διαχείρισης βάσεων δεδομένων παρέχει μία *γλώσσα ορισμού δεδομένων* (data definition language, DDL), που ορίζει το εννοιολογικό σχήμα σε όρους του υποκείμενου μοντέλου δεδομένων και μία *γλώσσα χειρισμού δεδομένων* (view manipulation language, DML) ή *γλώσσα επερώτησης*, με την οποία χειριζόμαστε τα δεδομένα.

Το εξωτερικό επίπεδο ή επίπεδο όψεων αποτελείται από τις όψεις των *επιμέρους χρηστών* πάνω τη βάση δεδομένων και περιγράφει το τμήμα της βάσης δεδομένων που είναι σχετικό με ένα συγκεκριμένο χρήστη, επιτρέποντας με αυτό τον τρόπο το σχηματισμό όψεων πάνω στα αποθηκευμένα δεδομένα σύμφωνα με τις πληροφοριακές απαιτήσεις του εκάστοτε χρήστη. Τα περισσότερα συστήματα διαχείρισης βάσεων δεδομένων παρέχουν μία γλώσσα για τον ορισμό όψεων, ονομαζόμενη *γλώσσα ορισμού δεδομένων όψεων* (view data definition language, VDDL) και δυνατότητες για την έκφραση επερωτήσεων και λειτουργιών πάνω στις όψεις, τις οποίες αποκαλούμε συλλογικά *γλώσσα χειρισμού δεδομένων όψεων* (view data manipulation language, VDML). Σε κάθε περίπτωση, αυτό που διαφέρει είναι το μοντέλο δεδομένων που χρησιμοποιείται από το σύστημα διαχείρισης βάσεων δεδομένων και όχι η υποκείμενη ιδέα.

Η αρχιτεκτονική ANSI/SPARC επιτυγχάνει με τη διάκριση των επιπέδων την **ανεξαρτησία δεδομένων**, δηλαδή την ικανότητα τροποποίησης του σχήματος σε ένα επίπεδο χωρίς να επηρεάζεται ο ορισμός σχήματος του υψηλότερου επιπέδου. Συνεπεία αυτής της αρχιτεκτονικής, όλοι οι χρήστες μπορούν να προσπελάσουν τα ίδια δεδομένα, ενώ δεν χρειάζεται να γνωρίζουν λεπτομέρειες της φυσικής αποθήκευσης της βάσης δεδομένων. Επίσης, η όψη ενός χρήστη (πρέπει να) είναι προστατευμένη από τις αλλαγές που γίνονται σε άλλες όψεις, οι δομές αποθήκευσης της βάσης δεδομένων μπορούν να αλλάξουν χωρίς να επηρεαστούν οι όψεις των χρηστών, η λογική δομή της βάσης δεδομένων είναι ανεπηρέαστη από αλλαγές σε ζητήματα φυσικής αποθήκευσης και η εννοιολογική δομή μίας βάσης δεδομένων δύναται να τροποποιείται χωρίς να επηρεάζονται όλοι οι χρήστες

3.1.1.1 Σχεσιακό Μοντέλο Δεδομένων

Η τεχνολογία όψεων είναι μία παλιά ιδέα που βρίσκει εφαρμογές σε νέα συμφραζόμενα. Η ιδέα της χρήσης όψεων για εξατομίκευση των δεδομένων χρονολογείται στις απαρχές του σχεσιακού μοντέλου δεδομένων. Τυπικά, το σχεσιακό μοντέλο δεδομένων αποτελείται από ένα πεπερασμένο σύνολο ονομάτων $A=\{A_1,A_2,\ldots,A_n\}$ που αποκαλούνται *γνωρίσματα*. Κάθε γνώρισμα A_i συσχετίζεται με ένα σύνολο τιμών V_i, το οποίο συμβολίζουμε με $dom(A_i)$ και αποκαλούμε *πεδίο*. Μία απεικόνιση που αναθέτει σε κάθε γνώρισμα A_i μία τιμή στο $dom(A_i)$ ονομάζεται *πλειάδα* (tuple) πάνω στο A και ένα σύνολο πλειάδων αποκαλείται *σχέση* (relation) πάνω στο A.

Η θεμελιώδης μαθηματική έννοια του σχεσιακού μοντέλου δεδομένων είναι η *συνολοθεωρητική σχέση*, η οποία ορίζεται ως ένα υποσύνολο του Καρτεσιανού γινομένου μίας λίστας από πεδία. Ως *σχήμα σχέσης* (relation schema) αναφέρουμε το συμβολισμό $R(A_1{:}dom(A_1),A_2{:}dom(A_2),\ldots,A_n{:}dom(A_n))$, που χρησιμοποιείται για να αναπαραστήσει μία μεταβλητή R, της οποίας οι τιμές είναι σχέσεις πάνω στο A. Η μεταβλητή R αποκαλείται *μεταβλητή σχέσης* και δηλώνει το όνομα της σχέσης. Μπορούμε να αναφέρουμε τους παρακάτω τυπικούς ορισμούς:

Ορισμός 1: Έστω ένα σχήμα σχέσης $R(A_1{:}dom(A_1),A_2{:}dom(A_2),\ldots,A_n{:}dom(A_n))$. Ένα **στιγμιότυπο σχέσης** ή απλούστερα μία **σχέση** r(R) είναι μία μαθηματική σχέση βαθμού *n* πάνω στα πεδία $dom(A_1)$, $dom(A_2)$, ..., $dom(A_n)$, η οποία είναι ένα υποσύνολο του Καρτεσιανού γινομένου των πεδίων που ορίζουν το R:

$$r(R) \subseteq (dom(A_1)\times dom(A_2)\times\ldots\times dom(A_n))$$

Ο συμβολισμός r(R) δηλώνει ότι το r είναι στιγμιότυπο του σχήματος σχέσης R.

Επομένως, ένα *στιγμιότυπο σχέσης* ή μία *σχέση* r(R) (απλούστερα r) $r=\{t_1,t_2,\ldots,t_n\}$ είναι ένα σύνολο από n-πλειάδες, όπου κάθε n-πλειάδα $t_i=(v_1,v_2,\ldots,v_n)$ είναι μία διατεταγμένη λίστα από n τιμές και κάθε v_i, $1\leq i\leq n$, είναι ένα στοιχείο του $dom(A_i)$ ή μία ειδική NULL τιμή.

Ορισμός 2: Ένα πεπερασμένο σύνολο σχημάτων σχέσεων αποτελεί το **σχήμα S μίας βάσης δεδομένων** (database schema), δηλαδή $S=\{R_1,R_2, \ldots,R_m\}$. Δοθέντος ενός σχήματος βάσης δεδομένων S, ένα πεπερασμένο σύνολο στιγμιότυπων σχέσεων αποτελεί ένα **στιγμιότυπο της βάσης δεδομένων** (database instance), δηλαδή $inst(S)=\{r_i(R_i) \mid 1\leq i\leq m$, r_i συμβολίζει μία σχέση πάνω στο $R_i\}$.

Χρησιμοποιώντας τον **Ορισμό 2**, ορίζουμε μία *σχεσιακή βάση δεδομένων* ως ένα πεπερασμένο σύνολο σχημάτων σχέσεων (σχήμα της βάσης δεδομένων) και ένα αντίστοιχο σύνολο από στιγμιότυπα σχέσεων (στιγμιότυπο της βάσης δεδομένων). Επίσης, κάθε ανάθεση τιμών στις μεταβλητές σχέσεων αποκαλείται *κατάσταση της βάσης δεδομένων* και το σύνολο όλων των καταστάσεων ονομάζεται *χώρος καταστάσεων της βάσης δεδομένων*. Με άλλα λόγια, η κατάσταση της βάσης δεδομένων καθορίζεται από τα δεδομένα της βάσης σε μία συγκεκριμένη χρονική στιγμή. Συνήθως, περιορίζουμε τις πιθανές καταστάσεις επιβάλλοντας περιορισμούς, π.χ λειτουργικές εξαρτήσεις, που πρέπει να ικανοποιούν οι καταστάσεις.

Για να ορίσουμε και να διαχειριστούμε το σχήμα μίας βάσης δεδομένων, απαιτείται μία γλώσσα επερώτησης, εκφρασμένης σε λογική ή άλγεβρα· με άλλα λόγια, μία γλώσσα για την περιγραφή επερωτήσεων σε μία σχεσιακή βάση δεδομένων. Η *Σχεσιακή Άλγεβρα* και η *SQL* (Structured Query Language) είναι γενικά αποδεκτές γλώσσες επερώτησης για το σχεσιακό μοντέλο δεδομένων. Η SQL, βασιζόμενη σε μία δηλωτική προσέγγιση, είναι η επικρατούσα γλώσσα επερώτησης σε επίπεδο εφαρμογών, ενώ η Σχεσιακή Άλγεβρα είναι μία *διαδικαστική* γλώσσα επερώτησης. Εφαρμόζεται σε ένα σύνολο σχέσεων χρησιμοποιώντας βασικούς και παράγωγους τελεστές, όπως επιλογή, προβολή, ένωση και διαίρεση, σύνδεση αντίστοιχα, και περιγράφει την ακολουθία με την οποία πρέπει να εφαρμοστούν οι τελεστές για να καθορίσουν το αποτέλεσμα μίας σχεσιακής έκφρασης, δηλαδή το αποτέλεσμα μίας επερώτησης.

Μία ονοματισμένη επερώτηση στη βάση δεδομένων χρησιμοποιώντας μία γλώσσα επερώτησης ονομάζεται **όψη** (view). Δοθέντος του σχήματος S μίας βάσης δεδομένων, μία όψη πάνω στο S ορίζεται δίδοντας το όνομα της σχέσης που αντιστοιχεί στην όψη και μία σχεσιακή έκφραση, που ορίζει την όψη και ονομάζεται *έκφραση ορισμού της όψης*, δηλαδή *<όνομα όψης>=<έκφραση ορισμού της όψης>*. Ειδικότερα, βασιζόμενοι στον ορισμό μίας όψης, μπορούμε να ορίσουμε μία σχέση όψης, της οποίας το όνομα είναι το όνομα της όψης και της οποίας το σχήμα απαρτίζεται από τα γνωρίσματα της έκφρασης που την ορίζει[22]. Τα ονόματα που χρησιμοποιούνται για τα γνωρίσματα, τις σχέσεις και τις όψεις προέρχονται από τον ίδιο χώρο ονομάτων. Ακολουθώντας ένα πιο τυπικό φορμαλισμό, μία όψη ορίζεται σύμφωνα με τον παρακάτω ορισμό.

Ορισμός 3: Έστω S το σχήμα μίας βάσης δεδομένων. Μία **όψη** (view) V πάνω στο S είναι ένα σχήμα σχέσης, το οποίο θα αποκαλούμε **σχήμα όψης**. Ο ορισμός μίας όψης είναι μία συνάρτηση *f* από τα στιγμιότυπα της βάσης δεδομένων στα στιγμιότυπα της όψης, η οποία υπολογίζει μία σχέση πάνω στα γνωρίσματα της V:

$$V : inst(S) \xrightarrow{f} r(V)$$

όπου $r(V) \subseteq (dom(A_1)\times\ldots\times dom(A_m))$ και A_i, $1\leq i\leq m$, είναι γνωρίσματα που ορίζουν τη V.

[22] Στην SQL για να ορίσουμε μία όψη χρησιμοποιούμε την εντολή ***CREATE VIEW V(A₁,…,Aₖ) AS Q***, όπου *V* είναι το όνομα της όψης, $A_1,\ldots, A_k$ είναι τα γνωρίσματά της και *Q* είναι η επερώτηση που ορίζει την όψη. Η όψη *V* δεν υπάρχει, αλλά όταν γίνει αντικείμενο επερώτησης, η *V* ή το σχετικό τμήμα της κατασκευάζεται. Για να κατασκευάσουμε την *V*, αποτιμούμε την επερώτηση *Q* και το σύνολο των πλειάδων που παράγονται από το *Q* αποτελούν τις πλειάδες στο *V*. Για να καταστρέψουμε μία όψη, χρησιμοποιούμε την εντολή ***DROP VIEW V***.

Μπορούμε να θεωρήσουμε ότι μία όψη είναι μία *νέα βάση δεδομένων* που προέρχεται από την αρχική και αποτελείται από ένα σύνολο πεδίων τιμών, γνωρισμάτων, σχέσεων και μία απεικόνιση που υπολογίζει την κατάσταση της όψης από την τρέχουσα κατάσταση της βάσης και η οποία αποκαλείται *απεικόνιση ορισμού της όψης* (view definition mapping). Άρα, αν *SS* είναι ο χώρος καταστάσεων της αρχικής βάσης δεδομένων και *f* είναι η απεικόνιση ορισμού της όψης, τότε *f(SS)* είναι ο χώρος καταστάσεων της όψης. Οι σχέσεις όψεων ολοκληρώνουν την πληροφορία (ή τμήματα αυτής) που περιέχεται σε σχέσεις της βάσης και για το λόγο αυτό, το περιεχόμενό τους υπολογίζεται σε αντίθεση με τα πρωτογενή δεδομένα που περιέχονται στις σχέσεις της βάσης δεδομένων. Κατά μια έννοια, η κατασκευή όψεων είναι η αντίστροφη διαδικασία της ολοκλήρωσης βάσεων δεδομένων, εφόσον για κάθε συλλογή δεδομένων που συνεισφέρει στη βάση, μπορούμε να κατασκευάσουμε όψεις που περιέχουν μόνο αυτά τα δεδομένα [U88].

Η χρησιμότητα των όψεων αποδεικνύεται από το πλήθος των εφαρμογών στις οποίες χρησιμοποιούνται και τη λειτουργικότητα που είναι σε θέση να υποστηρίξουν. Ειδικότερα, οι όψεις προωθούν τη **λογική ανεξαρτησία δεδομένων**. Στηριζόμενοι στην αρχιτεκτονική βάσεων δεδομένων τριών επιπέδων ANSI/SPARC που αναφέραμε προηγουμένως[23], η σχέση ανάμεσα στις όψεις (εξωτερικό επίπεδο) και το εννοιολογικό επίπεδο της βάσης δεδομένων στηρίζεται σε ένα είδος ανεξαρτησίας δεδομένων που ονομάζεται *λογική ανεξαρτησία δεδομένων* και αναφέρεται στο γεγονός ότι οι εφαρμογές που χρησιμοποιούν μία όψη δεν χρειάζεται να τροποποιηθούν, όταν το υποκείμενο σχήμα της βάσης δεδομένων αλλάξει. Πιο συγκεκριμένα, κατά τη διάρκεια του κύκλου ζωής μιας βάσης δεδομένων, πιθανότατα να καταστεί απαραίτητη η τροποποίηση του εννοιολογικού της σχήματος, για παράδειγμα, με την προσθήκη επιπλέον πληροφορίας για υπάρχουσες σχέσεις. Πολλές τροποποιήσεις στο εννοιολογικό σχήμα μπορούν να γίνουν χωρίς να επηρεαστούν οι υπάρχουσες όψεις, ενώ άλλες τροποποιήσεις μπορούν να γίνουν εάν επαναπροσδιοριστεί η συσχέτιση ανάμεσα

[23] Για λόγους ακρίβειας, αναφέρουμε ότι ο όρος *όψη* στα σχεσιακά περιβάλλοντα έχει σημασία, η οποία δεν είναι ακριβώς ταυτόσημη με τη σημασία που του αποδίδεται στην αρχιτεκτονική ANSI/SPARC. Στο εξωτερικό επίπεδο αυτής της αρχιτεκτονικής, η βάση δεδομένων γίνεται αντιληπτή ως μία *εξωτερική όψη*, που ορίζεται με ένα εξωτερικό σχήμα και οι διάφοροι χρήστες μπορεί να έχουν διαφορετικές εξωτερικές όψεις. Στα σχεσιακά συστήματα αντίθετα, μία άποψη είναι ένας *επώνυμος*, *παράγωγος* (*εικονικός*) πίνακας. Έτσι το σχεσιακό ανάλογο μίας εξωτερικής όψης ANSI/SPARC είναι (συνήθως) μία συλλογή πολλών σχέσεων, κάθε μια εκ των οποίων είναι μία όψη κατά τη σχεσιακή έννοια. Το *εξωτερικό σχήμα* αποτελείται από τους ορισμούς αυτών των όψεων [D95a].

στην όψη και το εννοιολογικό σχήμα. Και στις δύο περιπτώσεις όμως, δεν απαιτούνται αλλαγές στα προγράμματα εφαρμογών. Η μόνη τροποποίηση στο εννοιολογικό σχήμα που δεν θα μπορούσε να απεικονιστεί σε ένα επαναπροσδιορισμό μίας όψης σε όρους του τροποποιημένου εννοιολογικού σχήματος είναι η διαγραφή πληροφορίας που αντιστοιχεί σε πληροφορία παρούσα στην όψη. Τέτοιες αλλαγές θα απαιτούσαν λογικά επανεγγραφή μερικών προγραμμάτων εφαρμογών.

Από την οπτική γωνία του συστήματος διαχείρισης βάσεων δεδομένων, οι όψεις υποστηρίζουν τη λογική ανεξαρτησία των δεδομένων. Από την πλευρά των χρηστών της βάσης δεδομένων, η λογική ανεξαρτησία των δεδομένων είναι ισοδύναμη της **εξατομίκευσης της πληροφορίας**. Η τεχνική των όψεων παρέχει στους χρήστες τη δυνατότητα να ορίσουν ένα σύνολο από σχέσεις όψεων, δημιουργώντας με αυτόν τον τρόπο μία προσαρμοσμένη στις απαιτήσεις τους βάση δεδομένων. Για παράδειγμα, έστω η σχεσιακή βάση δεδομένων μίας εταιρίας, η οποία φυλάσσει πληροφορία σχετικά με πτυχές της επιχειρησιακής της δραστηριότητας, όπως δεδομένα για τους εργαζόμενους, τα τμήματα στα οποία εργάζονται, τις δραστηριότητες τις οποίες αναπτύσσουν και τους φορείς με τους οποίους συνεργάζεται. Αν και το σύνολο της προσφερόμενης πληροφορίας είναι απαραίτητο για την ομαλή λειτουργία της επιχείρησης, δεν ενδιαφέρει τους μεμονωμένους χρήστες της βάσης δεδομένων: ο διευθυντής προσωπικού της επιχείρησης ενδιαφέρεται μόνο για τμήματα της πληροφορίας που αφορούν τους εργαζόμενους και τα αντίστοιχα τμήματα, ενώ ο διευθυντής του διαφημιστικού τμήματος για τα προϊόντα, τις πωλήσεις τους και τους πελάτες. Συνεπώς, η «προσωπική» βάση δεδομένων του εκάστοτε χρήστη πληροφορίας αποτελείται από ένα σύνολο όψεων επί του συνόλου της διαχειριζόμενης πληροφορίας.

Εκτός του ότι διευκολύνουν τη γραφή προγραμμάτων εφαρμογών και προσαρμόζουν τα δεδομένα στις ανάγκες του εκάστοτε χρήστη, οι όψεις λειτουργούν και ως **μηχανισμός προστασίας των δεδομένων**, παρέχοντας ένα τρόπο πρόσβασης σε σύνολα δεδομένων μόνο από εξουσιοδοτημένους χρήστες. Ως ένα παράδειγμα της χρησιμότητας των όψεων στην ασφάλεια των δεδομένων, υποθέτουμε την ύπαρξη της σχέσης ΕΡΓΑΖΟΜΕΝΟΣ(Επίθετο, Τμήμα, Μισθός, Διεύθυνση) και επιθυμούμε οι χρήστες να έχουν πρόσβαση σε όλα τα γνωρίσματα της σχέσης πλην του γνωρίσματος «Μισθός». Για το λόγο αυτό, ορίζουμε την όψη ΣΤΟΙΧΕΙΑ_ΕΡΓΑΖΟΜΕΝΟΥ(Επίθετο, Τμήμα, Διεύθυνση). Αν και η όψη δεν υπάρχει φυσικά σαν αρχείο, μπορούμε να σχηματίσουμε SQL επερωτήσεις του τύπου:

SELECT Τμήμα

FROM ΣΤΟΙΧΕΙΑ_ΕΡΓΑΖΟΜΕΝΟΥ WHERE Επίθετο="Παπαδόπουλος"

για να ανακτήσουμε, για παράδειγμα, τα τμήματα της επιχείρησης στα οποία εργάζονται άτομα με επίθετο «Παπαδόπουλος». Με αυτό τον τρόπο, χρήστες που δεν επιτρέπεται να προσπελάσουν στοιχεία σχετικά με το μισθό των εργαζομένων, έχουν πρόσβαση μόνο στην όψη ΣΤΟΙΧΕΙΑ_ΕΡΓΑΖΟΜΕΝΟΥ, ενώ εξουσιοδοτημένοι χρήστες με πρόσβαση σε όλα τα στοιχεία έχουν πρόσβαση στη σχέση Εργαζόμενος.

Μπορούμε να διακρίνουμε δύο διαφορετικά είδη όψεων λαμβάνοντας υπόψη τη χρήση τους ως μηχανισμό προστασίας των δεδομένων [U88]. Το πρώτο είδος δεν επιτρέπει τροποποιήσεις στις όψεις και όψεις με αυτή τη λειτουργικότητα αποκαλούνται *όψεις μόνο για ανάγνωση* (read-only). Βρίσκουν εφαρμογή σε περιπτώσεις όπου ο ιδιοκτήτης μίας βάσης δεδομένων επιθυμεί να παραχωρήσει στους χρήστες δικαίωμα πρόσβασης στα δεδομένα του, αλλά ταυτόχρονα επιθυμεί να διατηρήσει το δικαίωμα αλλαγής της βάσης. Ο άλλος τύπος όψεων, οι *όψεις ανάγνωσης/εγγραφής* (read/write), επιτρέπουν την ανάγνωση και την εγγραφή σε αντικείμενα της όψης και μετατροπές στην όψη απεικονίζονται στο εννοιολογικό σχήμα. Ένα σοβαρό πρόβλημα με τις όψεις ανάγνωσης/εγγραφής είναι ότι ενημερώσεις πάνω σε αυτές συχνά έχουν παρενέργειες σε τμήματα της βάσης που δεν απεικονίζονται στις εν λόγω όψεις. Επίσης, στα σχεσιακά συστήματα είναι ασαφής η σημασία της διαγραφής κάποιων συστατικών μίας πλειάδας, εάν υπάρχουν γνωρίσματα που βρίσκονται εκτός της όψης και για αυτό το λόγο δεν πρέπει να διαγραφούν από το χρήστη που βλέπει μόνο την όψη. Για αυτούς τους λόγους, όλα τα συστήματα διαχείρισης βάσεων δεδομένων περιορίζουν τη δυνατότητα των χρηστών να ενημερώνουν όψεις σε λίγες, σαφείς περιπτώσεις. Για παράδειγμα, η SQL επιτρέπει τροποποίηση σε όψεις μόνο όταν η όψη παράγεται από μία σχέση με επιλογή και προβολή και δεν περιέχει συναθροίσεις [RG00, D95b].

Ως τελευταίο παράδειγμα της χρησιμότητας του μηχανισμού ορισμού όψεων στο σχεσιακό περιβάλλον, αναφέρουμε την **επαλήθευση περιορισμών ακεραιότητας** [S01]. Έστω για παράδειγμα, δύο σχέσεις μίας βάσης δεδομένων Α(Εργαζόμενος, Τμήμα) και Β(Τμήμα, Διευθυντής) και ο περιορισμός ότι οι τιμές του πεδίου «Τμήμα» στη σχέση Α πρέπει επίσης να εμφανίζονται στη σχέση Β. Μπορούμε να υλοποιήσουμε την επαλήθευση του περιορισμού ορίζοντας μία όψη ΑΒ, που αποτελείται από τις πλειάδες εκείνες που ανήκουν στη σχέση Α, αλλά όχι

στη Β, και αναφέρονται στο γνώρισμα «Τμήμα». Ένας πιθανός ορισμός της όψης ΑΒ μπορεί να γίνει με την SQL εντολή

CREATE VIEW AB AS (Select Τμήμα From A) minus (Select Τμήμα From B)

Ο περιορισμός ακεραιότητας ικανοποιείται όταν και μόνο όταν η αποτίμηση της παραπάνω SQL εντολής επιστρέφει μία κενή από περιεχόμενα όψη. Διαφορετικά, ο περιορισμός ακεραιότητας παραβιάζεται.

3.1.1.2 Οντοκεντρικό Μοντέλο Δεδομένων

Στη δεκαετία του 1980, έκαναν την εμφάνιση τους νέες εφαρμογές με απαιτήσεις στη διαχείριση μεγάλου όγκου (δομημένων) δεδομένων και συναλλαγών μεγάλης διάρκειας (σχεδιαστικά και βιομηχανικά συστήματα, επιστημονικές βάσεις δεδομένων, γεωγραφικά συστήματα πληροφοριών και βάσεις πολυμέσων), για τις οποίες τα παραδοσιακά συστήματα βάσεων δεδομένων, βασιζόμενα στο σχεσιακό μοντέλο δεδομένων, ήταν ανεπαρκή. Τα *ουτοκεντρικά* ή *αντικειμενοστραφή συστήματα διαχείρισης βάσεων δεδομένων* (OODBMS) αναπτύχθηκαν με σκοπό να ικανοποιήσουν τις απαιτήσεις αυτών των εφαρμογών και συνιστούν αποτέλεσμα της ολοκλήρωσης της τεχνολογίας των βάσεων δεδομένων με το οντοκεντρικό μοντέλο δεδομένων, που αναπτύχθηκε στις περιοχές των γλωσσών προγραμματισμού και μηχανικής λογισμικού. Ένα από τα σημαντικότερα χαρακτηριστικά των OODBMS είναι η ικανότητά τους να ορίζουν τόσο τη δομή των (σύνθετων) αντικειμένων που απαρτίζουν μία εφαρμογή όσο και τις λειτουργίες για τη διαχείριση αυτών. Τα *αντικειμενοσχεσιακά συστήματα διαχείρισης βάσεων δεδομένων* (Object-relational, ORDBMS) αποτελούν μία εναλλακτική, εξελικτική προσέγγιση ενσωματώνοντας αντικείμενα στο σχεσιακό μοντέλο δεδομένων και επεκτείνοντας υπάρχοντα σχεσιακά συστήματα διαχείρισης βάσεων δεδομένων με χαρακτηριστικά τυπικά του οντοκεντρικού μοντέλου δεδομένων, όπως ενθυλάκωση, πολυμορφισμός[24] και κληρονομικότητα. Η διττή φύση των αντικειμενοσχεσιακών συστημάτων αποδεικνύεται από την υποστήριξη της SQL ως γλώσσας επερώτησης και τη δυνατότητα διαχείρισης σύνθετων δεδομένων, συνδυασμός που αποφέρει καλές επιδόσεις, υποστήριξη ταυτόχρονης προσπέλασης σε δεδομένα και αξιοπιστία [CZ01].

---

[24] Ο πολυμορφισμός επιτρέπει στον ίδιο τελεστή ή σύμβολο να έχει διαφορετικές υλοποιήσεις, ανάλογα με τον τύπο των αντικειμένων στα οποία εφαρμόζεται.

Η οντοκεντρική προσέγγιση παρέχει την απαιτούμενη ευελιξία για την υλοποίηση εφαρμογών υψηλών απαιτήσεων, καθώς -στηριζόμενη στην έννοια των αντικειμένων ως αυθύπαρκτες οντότητες- δεν περιορίζεται στους τύπους δεδομένων και τις γλώσσες επερώτησης που διατίθενται από τα παραδοσιακά συστήματα βάσεων δεδομένων. Τις βάσεις του οντοκεντρικού μοντέλου δεδομένων αποτελούν οι έννοιες: **αντικείμενο**, **κλάση/τύπος** και **κληρονομικότητα**. Ειδικότερα, μία αντικειμενοστραφής βάση δεδομένων είναι μία συλλογή από αντικείμενα, κάθε ένα από τα οποία αναπαριστά μία οντότητα του πραγματικού κόσμου. Κάθε αντικείμενο αναγνωρίζεται μοναδικά από ένα *αναγνωριστικό* (OID)[25], έχει *κατάσταση* (συνίσταται από τις τιμές των ιδιοτήτων του) και *συμπεριφορά*, η οποία καθορίζεται από τις *λειτουργίες* που μπορούν να εκτελεστούν από και πάνω στο αντικείμενο. Οι ιδιότητες μπορεί να είναι είτε *γνωρίσματα* του αντικειμένου είτε *σχέσεις* ανάμεσα στο αντικείμενο και ένα ή περισσότερα άλλα αντικείμενα. Οι τιμές των γνωρισμάτων μπορεί να είναι σύνθετες, δηλαδή τιμές κατασκευασμένες από άλλα αντικείμενα και τιμές, χρησιμοποιώντας κάποιους κατασκευαστές τύπων.

Στο οντοκεντρικό μοντέλο δεδομένων, τα δεδομένα και οι λειτουργίες που τα διαχειρίζονται[26] ενθυλακώνονται σε μία μοναδική δομή: το *αντικείμενο*. Ένα αντικείμενο αποτελείται από μία *διεπαφή* (interface) και μία *υλοποίηση* (implementation). Η διεπαφή είναι η προδιαγραφή των λειτουργιών που μπορούν να εκτελεστούν πάνω στο αντικείμενο και είναι το μόνο τμήμα του αντικειμένου που είναι φανερό στους χρήστες. Αντιθέτως, η υλοποίηση περιέχει δεδομένα (την κατάσταση του αντικειμένου) και υλοποιήσεις λειτουργιών. Η **ενθυλάκωση** παρέχει ένα τύπο «λογικής ανεξαρτησίας δεδομένων», που επιτρέπει στα δεδομένα να μεταβάλλονται χωρίς να χρειάζεται να μεταβληθούν οι εφαρμογές που τα χρησιμοποιούν. Επίσης, παρέχει λογική επεκτασιμότητα των δεδομένων και ένας χρήστης μπορεί να επεκτείνει το σύνολο των τύπων χωρίς να υπάρχει διαφορά ανάμεσα στους παρεχόμενους από το σύστημα τύπους και εκείνους που ορίζονται από τον χρήστη.

---

[25] Η έννοια της ταυτότητας ενός αντικειμένου είναι αρκετά διαφορετική από την έννοια του κλειδιού που χρησιμοποιείται στο σχεσιακό μοντέλο δεδομένων για να αναγνωρίσει μοναδικά τις πλειάδες μίας σχέσης. Ένα κλειδί ορίζεται ως η τιμή ενός ή περισσότερων γνωρισμάτων, που μπορεί τροποποιηθεί, ενώ ένα OID είναι ανεξάρτητο από την κατάσταση του αντικειμένου, δηλαδή τις τιμές των γνωρισμάτων του.

[26] Για να διαχειριστούμε τα αντικείμενα μίας οντοκεντρικής βάσης δεδομένων χρησιμοποιούμε *μεθόδους*. Ο ορισμός μίας μεθόδου συνήθως αποτελείται από δύο συστατικά: μία *δήλωση* (signature) και μία *υλοποίηση* (implementation). Η δήλωση ορίζει το όνομα της μεθόδου, τα ονόματα και τους τύπους των ορισμάτων της μεθόδου και τον τύπο του αποτελέσματος για τις μεθόδους που επιστρέφουν μία τιμή αποτελέσματος, ενώ η υλοποίηση αποτελείται από το σύνολο των εντολών σε μία γλώσσα προγραμματισμού.

Δεν αναπαρίστανται όμως όλες οι οντότητες του πραγματικού κόσμου ως αντικείμενα. Το οντοκεντρικό μοντέλο δεδομένων κάνει διάκριση ανάμεσα σε αντικείμενα και *λεκτικές σταθερές* ή *τιμές* (literals). Η κυριότερη διαφορά ανάμεσα σε ένα αντικείμενο και μία λεκτική σταθερά είναι ότι ένα αντικείμενο έχει μία ταυτότητα (OID), που είναι ανεξάρτητη από την κατάστασή του, ενώ μία λεκτική σταθερά αναγνωρίζεται μόνο από την τιμή της. Οι λεκτικές σταθερές είναι παγκοσμίως γνωστές αφαιρέσεις και έχουν την ίδια σημασία για κάθε χρήστη, σε αντίθεση με τα αντικείμενα που αντιστοιχούν σε αφαιρέσεις των οποίων η σημασία καθορίζεται από την εκάστοτε εφαρμογή. Για να διευκολυνθεί η διαχείριση και η μοντελοποίηση μίας εφαρμογής, τόσο τα αντικείμενα όσο και οι τιμές κατηγοριοποιούνται σύμφωνα με τον τύπο τους. Με αυτόν τον τρόπο, όλα τα αντικείμενα ενός συγκεκριμένου τύπου έχουν το ίδιο σύνολο ιδιοτήτων και το ίδιο σύνολο λειτουργιών.

Ένας τύπος αντικειμένου ορίζεται μέσω μίας *κλάσης*. Η έννοια της κλάσης αποτελεί τη βάση της υποστασιοποίησης και ορίζει τη δομή και τη συμπεριφορά των αντικειμένων που την απαρτίζουν. Μία κλάση έχει την ικανότητα να δημιουργεί στιγμιότυπα (αντικείμενα) και το σύνολο όλων των στιγμιότυπων που ανήκουν σε μία κλάση ονομάζεται *έκταση* (extent). Η αυτόματη σύνδεση μίας έκτασης με την αντίστοιχη κλάση έχει το πλεονέκτημα της απλοποίησης της διαχείρισης των κλάσεων και των στιγμιότυπών τους. Σε αντίθεση, συστήματα στα οποία οι εκτάσεις γίνονται αντικείμενο διαχείρισης από το χρήστη παρέχουν μεγαλύτερη ευελιξία, αλλά σε βάρος της πολυπλοκότητας διαχείρισης των εκτάσεων των κλάσεων. Τη διαχείριση των κλάσεων διευκολύνει και ο μηχανισμός της *κληρονομικότητας*, που επιτρέπει σε μία κλάση, αποκαλούμενη *υποκλάση*, να οριστεί βάσει του ορισμού μίας άλλης κλάσης, που ονομάζεται *υπερκλάση*. Λόγω της κληρονομικότητας, η υποκλάση κληρονομεί τα γνωρίσματα και τις λειτουργίες της υπερκλάσης της, ενώ ταυτόχρονα μπορεί να διαθέτει τις δικές τις ιδιότητες και λειτουργίες. Πολλά συστήματα επιτρέπουν τον ορισμό περισσότερων της μιας άμεσης υπερκλάσης για μια κλάση (πολλαπλή κληρονομικότητα), ενώ άλλα συστήματα επιβάλλουν τον περιορισμό της μονής κληρονομικότητας, απαιτώντας μία κλάση να έχει το πολύ μία υπερκλάση. Και στις δύο περιπτώσεις, η εφαρμογή της κληρονομικότητας σε κλάσεις δημιουργεί ιεραρχίες, βάση των οποίων οργανώνονται εννοιολογικά οι κλάσεις. Ένα πρόβλημα που εμφανίζεται με την παρουσία πολλαπλής κληρονομικότητας είναι η *σχιζοφρένεια*, δηλαδή η επίλυση συγκρούσεων μεθόδων και παρουσιάζεται όταν μια μέθοδος κληρονομείται σε μια υποκλάση από διαφορετικές υπερκλάσεις και δεν επαναπροσδιορίζεται στην

υποκλάση. Το πρόβλημα αυτό, που δεν εμφανίζεται στο RDF/S λόγω της μοναδικότητας των ονομάτων ιδιοτήτων, οφείλεται στη δήλωση γνωρισμάτων και μεθόδων με το ίδιο όνομα και στην τοπικότητα των ονομάτων.

Συνοπτικά, παρουσιαζόμενο με τυπικό τρόπο, το οντοκεντρικό μοντέλο βάσεων δεδομένων αποτελείται από τα εξής βασικά δομικά στοιχεία και ορισμούς [C01]:

- *Σύνολο D ατομικών τιμών*, τέτοιο ώστε $D=\cup D_i$, όπου D_i είναι ατομικά πεδία (αλφαριθμητικά, ακέραιοι, πραγματικοί κτλ), *σύνολο ατομικών τύπων* (π.χ STRING, INT, REAL), *σύνολο A ονομάτων γνωρισμάτων*, *σύνολο O αναγνωριστικών αντικειμένων* (oid), *σύνολο C ονομάτων κλάσεων*.
- Ένα ***αντικείμενο*** είναι ένα ζεύγος (o,v), όπου $o \in O$ και v είναι μία τιμή, η οποία ορίζεται ως εξής: κάθε στοιχείο $v \in D$ είναι μία τιμή, κάθε στοιχείο $o \in O$ είναι μία τιμή, εάν $v_1, v_2, \ldots, v_n$ είναι τιμές και $a_1, a_2, \ldots, a_n$ είναι ονόματα γνωρισμάτων, τότε $[a_1:v_1, a_2:v_2, \ldots, a_n:v_n]$ είναι μία *τιμή πλειάδας*, $\{v_1, v_2, \ldots, v_n\}$ είναι μία *τιμή συνόλου* και $<v_1, v_2, \ldots, v_n>$ είναι μία *τιμή λίστας*.
- Ο ***τύπος μίας τιμής*** ορίζεται ως εξής: κάθε ατομικός τύπος είναι ένας τύπος, κάθε στοιχείο του C είναι ένας τύπος, εάν $t, t_1, t_2, \ldots, t_n$ είναι τύποι και $a_1, a_2, \ldots, a_n$ είναι ονόματα γνωρισμάτων, τότε $[a_1:t_1, a_2:t_2, \ldots, a_n:t_n]$ είναι ένας *τύπος πλειάδας*, $\{t\}$ είναι ένας *τύπος συνόλου* και $<t>$ είναι ένας *τύπος λίστας*. Το σύνολο των τύπων δοθέντος ενός υποσυνόλου ονομάτων κλάσεων C συμβολίζεται με *types(C)*.
- Μία ***ιεραρχία κλάσεων*** είναι μία τριάδα $(C, \sigma, \prec)$, όπου C είναι ένα υποσύνολο ονομάτων κλάσεων, σ είναι μία συνάρτηση που επιστρέφει για κάθε κλάση $c \in C$ τον τύπο της στο types(C) και $\prec$ είναι μία μερική διάταξη (ανακλαστική, αντισυμμετρική και μεταβατική) στο C.
- Δοθείσας μίας ιεραρχίας κλάσεων $(C, \sigma, \prec)$, η ***σχέση υποτύπων*** $\leq$ ορισμένη στο types(C) είναι η μικρότερη μερική διάταξη οριζόμενη ως:
  - εάν $c, c' \in C$: $c \prec c'$, τότε $c \leq c'$
  - εάν $t \leq t'$, τότε $\{t\} \leq \{t'\}$ και $<t> \leq <t'>$
  - για κάθε $j \in [1 \ldots m]$ υπάρχει $i \in [1..n]$: $b_j = a_i$ και $t_i \leq t'_j$ τότε $[a_1:t_1,\ldots,a_n:t_n] \leq [b_1:t'_1,\ldots,b_m:t'_m]$

  Μία ιεραρχία κλάσεων $(C, \sigma, \prec)$ είναι *καλώς ορισμένη* εάν για κάθε $c, c' \in C$: $c \prec c'$, τότε $\sigma(c) \leq \sigma(c')$.

- Έστω μία καλώς ορισμένη ιεραρχία κλάσεων (C, σ, $\prec$), όπου κάθε κλάση $c \in C$ έχει ένα σύνολο από συσχετιζόμενες μεθόδους. Η ***υπογραφή μίας μεθόδου*** είναι μία έκφραση της μορφής $m{:}c \times t_1 \times t_2 \times .. \times t_n \rightarrow t$, όπου m είναι το όνομα της μεθόδου, c είναι η κλάση στην οποία ορίζεται η μέθοδος, $t_1,...,t_n$ είναι οι τύποι των ορισμάτων της και t είναι ο τύπος του αποτελέσματος στο types(C). Το σύνολο όλων των υπογραφών μεθόδων που σχετίζονται με μία κλάση μίας ιεραρχίας συμβολίζεται με M. Οι υπογραφές είναι *καλώς ορισμένες* εάν για κάθε $m \in M$, όπου m ορίζεται σε δύο κλάσεις $c,c' \in C$, $c \prec c'$, $m{:}c \times t_1 \times t_2 \times .. \times t_n \rightarrow t$, $m{:}c' \times t'_1 \times t'_2 \times .. \times t'_n \rightarrow t'$, τότε ισχύει για κάθε $i \in [1..n]$ ότι $t_i \leq t_j'$ και $t \leq t'$.
- Δοθείσας μίας καλώς ορισμένης ιεραρχίας κλάσεων (C, σ, $\prec$), μία ***συνάρτηση απόδοσης πληθυσμού*** είναι μία συνάρτηση $\pi_d{:}C \rightarrow 2^O$ που συσχετίζει με κάθε όνομα κλάσης $c \in C$ ένα πεπερασμένο σύνολο $o \in O$, τέτοια ώστε, εάν c, $c' \in C$ και $c \neq c'$, τότε $\pi_d(c) \cap \pi_d(c') = \{\ \}$. Η ***ανάθεση αναγνωριστικών αντικείμενων*** είναι μία συνάρτηση π τέτοια ώστε για κάθε κλάση c: $\pi(c) = \cup\{\ \pi(c') \mid c' \prec c\}$. Ουσιαστικά, η συνάρτηση π υπολογίζει την έκταση μίας κλάσης $c \in C$.
- Δοθείσας μία συνάρτησης ανάθεσης αναγνωριστικών αντικειμένων π, η ***ερμηνεία ενός τύπου*** $t \in$ types(C), που συμβολίζεται dom(t), ορίζεται ως:
  - για κάθε ατομικό τύπο b: dom(b)=b
  - για κάθε κλάση $c \in C$, dom(c)= π(c)
  - $dom(\{t\}) = \{\{v_1,...,v_j\} \mid j \geq 0$ και κάθε $i \in [1..j]$, $v_i \in dom(t)\}$
  - $dom(<t>) = \{<v_1,...,v_j> \mid j \geq 0$ και κάθε $i \in [1..j]$, $v_i \in dom(t)\}$
  - $dom([a_1{:}t_1,\ a_2{:}t_2,\ ...,\ a_k{:}t_k]) = \{[a_1{:}v_1,\ a_2{:}v_2,\ ...,\ a_k{:}v_k,\ ...,\ a_{k+1}{:}v_{k+1}] \mid i \geq 0$ και κάθε $i \in [1..k]$, $v_i \in dom(t_i)\}$

Έχοντας ορίσει τα βασικά δομικά στοιχεία του οντοκεντρικού μοντέλου δεδομένων, μπορούμε να αναφέρουμε τους παρακάτω τυπικούς ορισμούς.

Ορισμός 4: Ένα **σχήμα S** μίας αντικειμενοστραφούς βάσης δεδομένων είναι μία πεντάδα (C,σ,$\prec$,M,G), όπου (C, σ, $\prec$) είναι μία καλώς ορισμένη ιεραρχία κλάσεων, M είναι ένα σύνολο καλώς ορισμένων υπογραφών μεθόδων και G είναι ένα σύνολο ονομάτων (π.χ σχέσεων) με ένα συσχετιζόμενο τύπο $\sigma(g) \in$ types(C) για κάθε $g \in G$.

Ορισμός 5: Έστω το σχήμα S μίας αντικειμενοστραφούς βάσης δεδομένων. Ένα **στιγμιότυπο** I του σχήματος S, αποκαλούμενο **αντικειμενοστραφής βάση δεδομένων** και συμβολιζόμενο inst(S), είναι μία τετράδα (π, υ, μ, γ), όπου π είναι η συνάρτηση απόδοσης πληθυσμού για κάθε κλάση: $O=\cup\{\pi(c) \mid c\in C\}$, υ είναι η συνάρτηση ανάθεσης τιμών για κάθε αντικείμενο o μίας κλάσης c: εάν $o\in\pi(c)$, τότε $\upsilon(o)\in dom(\sigma(c))$, μ είναι η συνάρτηση ανάθεσης τιμών για κάθε μέθοδο $m:c\times w\rightarrow t$ μίας κλάσης c: εάν $m\in M$, τότε $\mu(m:c\times w\rightarrow t)\in(dom(c)\times dom(w))^{dom(t)}$, και γ είναι η συνάρτηση ανάθεσης τιμών για κάθε όνομα $g\in G$: εάν $g\in G$, τότε $\gamma(g)\in dom(\sigma(g))$.

Όπως και στο σχεσιακό μοντέλο δεδομένων, για την προσπέλαση της αποθηκευμένης πληροφορίας (σχήμα και δεδομένα) χρησιμοποιούνται γλώσσες επερώτησης. Στο οντοκεντρικό μοντέλο όμως υπάρχουν δύο διαφορετικοί τρόποι πρόσβασης στα δεδομένα. Ο πρώτος ονομάζεται *πλοηγητικός* (navigational) και βασίζεται στα αναγνωριστικά των αντικειμένων και στις ιεραρχίες τις οποίες σχηματίζουν. Δοθέντος ενός αναγνωριστικού, είναι δυνατή η άμεση προσπέλαση του αντικειμένου που ταυτοποιείται από το αναγνωριστικό καθώς και η πλοήγηση διαμέσου των αντικειμένων που σχετίζονται με το συγκεκριμένο αντικείμενο. Ο δεύτερος τρόπος προσπέλασης ονομάζεται *συνδετικός (*associative) και βασίζεται σε τύπου SQL γλώσσες επερώτησης. Οι δύο τρόποι πρόσβασης στα δεδομένα μπορούν να χρησιμοποιηθούν συμπληρωματικά: μία επερώτηση αποτιμάται για να επιλέξει ένα σύνολο από αντικείμενα, που κατόπιν προσπελάζονται από τις εφαρμογές μέσω του πλοηγητικού μηχανισμού.

Υιοθετώντας ελαφρά διαφορετικό μοντέλο δεδομένων και υποστηρίζοντας διαφορετικά χαρακτηριστικά, έχουν αναπτυχθεί γλώσσες επερώτησης προσαρμοσμένες στις ιδιαιτερότητες του οντοκεντρικού μοντέλου δεδομένων. Το ODMG πρότυπο [CBC+00], που ορίζει τις αρχές του αντικειμενοστραφούς μοντέλου δεδομένων προδιαγράφει μία δηλωτική γλώσσα επερώτησης, την *OQL* [C98]. Αν και η OQL μπορεί να θεωρηθεί ως μία συμβατική επέκταση της SQL, στην πραγματικότητα διατηρεί μόνο κάποια συντακτικά πρότυπα, καθώς σημασιολογικά ακολουθεί το οντοκεντρικό μοντέλο, που είναι πολύ πιο εκφραστικό από το σχεσιακό μοντέλο δεδομένων. Μία άλλη γλώσσα επερώτησης, εφαρμοζόμενη σε ORDBMS, είναι το νέο SQL πρότυπο που αναπτύχθηκε από τους φορείς ANSI και ISO, η *SQL3*. Σε αντίθεση με τις προηγηθείσες εκδόσεις, η SQL3 υποτίθεται ότι αποτελεί μία γλώσσα με πλήρη υπολογιστική δύναμη. Αν και η κύρια δομή δεδομένων εξακολουθεί να είναι ο πίνακας (σχέση), η SQL3 προσπαθεί να προσθέσει την έννοια των αντικειμένων στην SQL, υποστηρίζοντας

πλήθος άλλων λειτουργιών, όπως αφηρημένους τύπους δεδομένων από το χρήστη, μεθόδους, αναγνωριστικά αντικειμένων, υποτύπους, κληρονομικότητα και πολυμορφισμό. Η διαφορά των δύο γλωσσών έγκειται στην κεντρική έννοια που χρησιμοποιούν: πίνακας ή αντικείμενα.

Εντούτοις, σε αντίθεση με το σχεσιακό μοντέλο δεδομένων, το οντοκεντρικό μοντέλο δεδομένων, όπως ορίζεται στο πρότυπο ODMG [CBC+00], δεν παρέχει υποστήριξη για όψεις. Για το λόγο αυτό, αναγνωρίζοντας τη σημασία και την αναγκαιότητα ύπαρξης ενός μηχανισμού ορισμού όψεων, έχει προταθεί ένα σύνολο άλλων προσεγγίσεων για την υποστήριξη όψεων σε αντικειμενοστραφείς βάσεις δεδομένων [π.χ JR99, AB91, R92, SAD94, LDB97a, LDB97b, LDB98, GBC+97]. Ενδεικτικά, συστήματα που έχουν αναπτυχθεί για την υποστήριξη αντικειμενοστραφών όψεων δεδομένων αποτελούν τα O_2 Views [AB91, YBS97, S95], AMOSII [JR99] και MultiView [R92]. Γενικά, εκτός του ότι βασίζονται σε διαφορετικά μοντέλα δεδομένων και εκμεταλλεύονται διαφορετικές γλώσσες επερώτησης για να ορίσουν τον πληθυσμό των κλάσεων, οι προτάσεις αυτές διαφέρουν ως προς τις λειτουργικότητες που στοχεύει να υποστηρίξει ο μηχανισμός όψεων, την προσέγγιση που ακολουθείται για την τοποθέτηση των *ιδεατών* κλάσεων (κλάσεων που ορίζονται σε μία όψη) στο σχήμα και τις πράξεις ενημέρωσης που επιτρέπονται πάνω στις όψεις. Για παράδειγμα, ένας μηχανισμός όψεων που προορίζεται για την υποστήριξη της εξέλιξης σχήματος μπορεί να είναι διαφορετικός από ένα μηχανισμό για τη χρήση όψεων στην επερώτηση. Επίσης, ένα άλλο σημείο στο οποίο διαφέρουν οι διάφορες προσεγγίσεις είναι το *επίπεδο λεπτομέρειας στον ορισμό των όψεων*, το οποίο ουσιαστικά καθορίζει τί θεωρείται όψη σε κάθε μία από τις προσεγγίσεις.

Ειδικότερα, ανάλογα με το επίπεδο λεπτομέρειας στον ορισμό των αντικειμενοστραφών όψεων (granularity), μπορούμε να διακρίνουμε *όψεις* και *σχήματα όψεων* (και στις δύο περιπτώσεις όμως χρησιμοποιούμε συνολικά τον όρο αντικειμενοστραφής όψη). Οι όψεις είναι ιδεατές κλάσεις και μπορούν να χρησιμοποιηθούν ακριβώς όπως οι κλάσεις (με τη διαφορά ότι η ενημέρωση τους δεν είναι εφικτή σε όλες τις περιπτώσεις) [SLT91]. Ένα σχήμα όψεων είναι ένα *ιδεατό σχήμα*, δηλαδή ένα σχήμα που αποτελείται από όψεις παρά από κλάσεις [GBC+97] και ορίζεται από εκφράσεις του τύπου:

DEFINE VIEW <*όνομα*>

{*ορισμοί κλάσεων και μεθόδων*}

όπου *όνομα* είναι το όνομα του σχήματος όψης και οι ιδεατές κλάσεις ορίζονται στο τμήμα που περικλείεται από τα άγκιστρα.

Σύμφωνα με το [AB91], μία όψη είναι ένα σύνολο από δηλώσεις δημιουργίας μίας κλάσης, των υποκλάσεων και γνωρισμάτων αυτής. Ακολουθώντας αυτή τη λογική, οι περισσότερες προσεγγίσεις υποστηρίζουν βασικές λειτουργίες για τον ορισμό μίας ιδεατής κλάσης, όπως απόκρυψη ή προβολή επιλεγμένων γνωρισμάτων και μεθόδων, εκλέπτυνση του ορισμού μίας κλάσης με κάποια γνωρίσματα ή μεθόδους και πράξεις συνόλων (ένωση, διαφορά, τομή) [YBS97, LDB97a, LBD97b, LDB98, SLT91]. Στο [GBC+97], μία όψη είναι ένα σύνολο από δηλώσεις, ένα σχήμα. Το ίδιο επίπεδο λεπτομέρειας στον ορισμό όψεων υιοθετείται στα [R92, SAD94]. Γενικότερα, τα σχήματα όψεων υποστηρίζουν την ικανότητα *αναδόμησης* ενός σχήματος, με τρόπο που να ικανοποιεί τις ανάγκες συγκεκριμένων εφαρμογών. Επιπλέον, είναι σε θέση να αναπαραστήσουν το εξωτερικό επίπεδο μίας βάσης δεδομένων σύμφωνα με το πνεύμα της αρχιτεκτονικής ANSI-SPARC, παρέχοντας στο χρήστη τη δυνατότητα να ορίζει ιδεατά σχήματα που αναπαριστούν το δικό του τρόπο θεώρησης των δεδομένων (και όχι μόνο μεμονωμένες κλάσεις).

Υιοθετώντας τη συλλογιστική των σχημάτων όψεων, δίνουμε τον ακόλουθο τυπικό ορισμό μίας αντικειμενοστραφούς όψης:

Ορισμός 6: Έστω S το σχήμα μίας αντικειμενοστραφούς βάσης δεδομένων. Μία **όψη** (view) V πάνω στο S είναι ένα νέο σχήμα, το οποίο θα αποκαλούμε **σχήμα όψης**. Ο ορισμός μίας όψης είναι μία συνάρτηση[27] f από τα στιγμιότυπα της αντικειμενοστραφούς βάσης δεδομένων στα στιγμιότυπα της όψης:

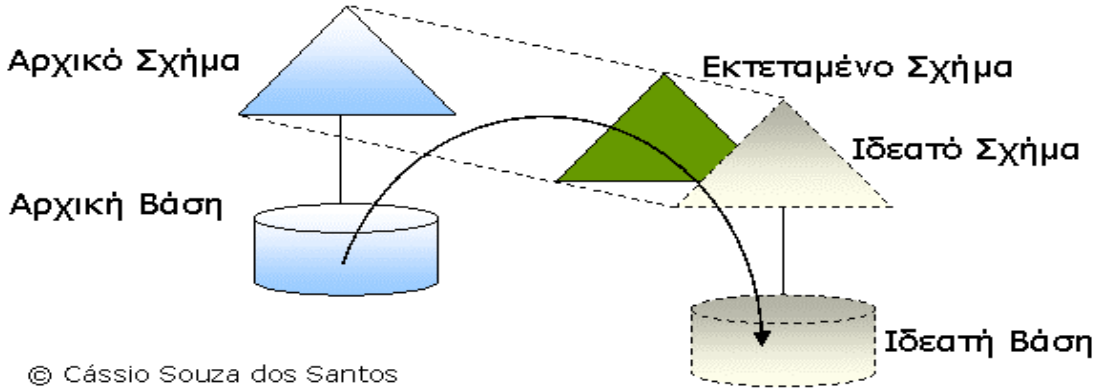
$$V : inst(S) \xrightarrow{f} inst(V)$$

όπου inst(V) ορίζεται όπως το inst(S).

Μπορούμε να θεωρήσουμε ότι ο ορισμός όψεων-κλάσεων είναι μία εκφυλισμένη μορφή του ορισμού σχημάτων όψεων. Στην περίπτωση αυτή, μία όψη ορίζεται ως μία επερώτηση σε μία ή περισσότερες κλάσεις (βασικές ή ιδεατές) της βάσης δεδομένων και της οποίας το αποτέλεσμα είναι μία νέα κλάση.

---

[27] Στην περίπτωση που η χρησιμοποιούμενη γλώσσα επερώτησης από το υποκείμενο μοντέλο δεδομένων υποστηρίζει τη δυνατότητα δημιουργίας νέων αντικειμένων, μία επερώτηση (που χρησιμοποιείται για να καθορίσει τον πληθυσμό μίας ιδεατής κλάσης) δεν είναι πλέον μία *απεικόνιση* ανάμεσα σε στιγμιότυπα, αλλά μία *σχέση* που ικανοποιεί ένα κριτήριο λειτουργικότητας, το οποίο διασφαλίζει ότι το αποτέλεσμα δεν εξαρτάται από τα επιλεγόμενα νέα αναγνωριστικά αντικειμένων [W98]. Συνεπώς, ο ορισμός μίας όψης σε αυτή την περίπτωση είναι μία σχέση παρά μία συνάρτηση.



Εικόνα 3.2. Δημιουργία ιδεατού σχήματος

Απομονώνοντας τις λεπτομέρειες της χρησιμοποιούμενης γλώσσας επερώτησης, οι όψεις ορίζονται εν γένει από μία δήλωση της μορφής:

DEFINE VIEW <*όνομα*> AS <*επερώτηση* >

Κατά την επεξεργασία αυτής της δήλωσης, ο όρος <όνομα> θα αποτελέσει το όνομα της ιδεατής κλάσης, ενώ μπορεί να υπάρχουν όροι που ορίζουν ρητώς τη λίστα γνωρισμάτων και μεθόδων της ιδεατής κλάσης. Όπως και στο σχεσιακό μοντέλο δεδομένων, τα ονόματα των όψεων και των βασικών κλάσεων προέρχονται από τον ίδιο χώρο ονοματοδοσίας. Η μόνη διαφορά είναι ότι η έκταση της όψης ορίζεται από την έκφραση <επερώτηση> και βασίζεται στις εκτάσεις άλλων (ιδεατών ή βασικών) κλάσεων. Με άλλα λόγια, ο ορισμός μίας όψης είναι παρόμοιος με τον ορισμό μίας κλάσης (όνομα, λίστα γνωρισμάτων, λίστα μεθόδων, λίστα περιορισμών) με τη διαφορά ότι διαθέτει μία συνάρτηση, που υπολογίζει τα μέλη της ιδεατής κλάσης βασιζόμενη στην κατάσταση της βάσης δεδομένων. Με αυτό τον τρόπο, τα περιεχόμενα μίας ιδεατής κλάσης γενικά δεν αποθηκεύονται άμεσα, αλλά υπολογίζονται κατά απαίτηση[28].

Η δημιουργία μίας όψης είναι ουσιαστικά το αποτέλεσμα δύο διαδοχικών βημάτων: ενός βήματος *επέκτασης* (extension), το οποίο ακολουθείται από ένα βήμα *προβολής* (projection) [S95]. Η **Εικόνα 3.2** παρουσιάζει εποπτικά τη διαδικασία δημιουργίας μίας όψης. Δοθέντος ενός σχήματος βάσης S, η επέκταση ορίζει ένα σχήμα ES[29], επεκτείνοντας το αρχικό σχήμα S με νέες κλάσεις

[28] Θα επικεντρωθούμε στην περίπτωση των όψεων-κλάσεων, που απαντώνται πιο συχνά στη βιβλιογραφία.

[29] Ο τρόπος δημιουργίας του επεκταμένου σχήματος και οι ιδιότητές του διαφέρουν από προσέγγιση σε προσέγγιση. Παραδείγματος χάριν, στην προσέγγιση MultiView [R92], το επεκταμένο σχήμα ονομάζεται *καθολικό σχήμα* και αποτελείται από βασικές και ιδεατές κλάσεις. Στο σύστημα O_2Views[S95], ένα επεκταμένο σχήμα αποτελείται από ιδεατές και

παραγόμενες από βασικές κλάσεις, γνωρίσματα και/ή μεθόδους. Το σχήμα ES ονομάζεται *επεκταμένο σχήμα* και αποτελεί την είσοδο της διαδικασίας προβολής, η οποία συνίσταται στην απόκρυψη πληροφορίας (κλάσεις, μέθοδοι και/ή γνωρίσματα) από το επεκταμένο σχήμα για την απόκτηση ενός σχήματος S΄, που αποτελεί το σχήμα όψης (ιδεατό σχήμα). Η όλη αυτή διαδικασία, την οποία μπορούμε να αποκαλέσουμε *μετασχηματισμό-όψης* (view-transformation), μας δίνει τη δυνατότητα να συνθέσουμε όψεις, αλλά πρέπει να πληροί κάποιους περιορισμούς σχετικά με την παραγωγή των κλάσεων, π.χ οι ιδεατές κλάσεις στο S΄ παράγονται ρητώς από τις κλάσεις στο S. Σε πολλές προσεγγίσεις, π.χ [R92, LDB97a, LDB97b, LDB98, SLT91], ορίζονται τελεστές που εκφράζουν μετασχηματισμούς όψεων και ουσιαστικά προδιαγράφουν τις διαδικασίες επέκτασης και προβολής.

Ένας μηχανισμός όψεων στα πλαίσια του οντοκεντρικού μοντέλου δεδομένων θα πρέπει να επιτρέπει στο χρήστη να αναδομεί τη βάση δεδομένων και να αλλάζει τη συμπεριφορά και τη δομή των αντικειμένων. Θα πρέπει επίσης να του επιτρέπει να ορίζει τιμές γνωρισμάτων, όπως επίσης να εισάγει νέες κλάσεις στην ιεραρχία των κλάσεων. Οι ιδεατές αυτές κλάσεις ουσιαστικά ορίζονται με τον καθορισμό του πληθυσμού τους και αποκτούν πληθυσμό είτε με την επιλογή υπαρχόντων αντικειμένων από άλλες κλάσεις είτε με τη δημιουργία νέων αντικειμένων. Μόλις δημιουργείται μία ιδεατή κλάση, η δομή και η συμπεριφορά της παράγονται από το σύστημα που υλοποιεί τον μηχανισμό όψεων ή μπορούν να οριστούν ρητώς από το χρήστη. Για παράδειγμα, ο χρήστης μπορεί να αποδώσει νέες μεθόδους σε μία ιδεατή κλάση ή να τροποποιήσει υπάρχουσες. Γενικά, μία όψη μπορεί να χρησιμοποιηθεί όπως μία κλάση και ένας μηχανισμός ορισμού αντικειμενοστραφών όψεων πρέπει να υποστηρίζει τα βασικά χαρακτηριστικά του υποκείμενου μοντέλου δεδομένων για να διευκολύνει τη διαφάνεια στη χρήση όψεων και βασικών κλάσεων. Με το σκεπτικό αυτό, πρέπει να υποστηρίζονται επίσης οι σχέσεις κληρονομικότητας (υπαλληλίας) ανάμεσα σε όψεις και η οργάνωση των όψεων σε ιεραρχίες όψεων, ακριβώς όπως οι κλάσεις οργανώνονται σε μία ιεραρχία κλάσεων.

Ως παράδειγμα ορισμού όψεων σε βάσεις δεδομένων που στηρίζονται στο οντοκεντρικό μοντέλο δεδομένων, αναφέρουμε το μοντέλο δεδομένων O_2 με την

εισαγόμενες κλάσεις. Ανεξάρτητα όμως από την υλοποίηση του επεκταμένου σχήματος, η συλλογιστική παραμένει η ίδια.

ομώνυμη γλώσσα επερώτησης [AB91]. Το παρακάτω σύμπλεγμα εκφράσεων ορίζει μία αντικειμενοστραφή όψη με όνομα My_View:

```
view My_View;
import class PERSON from database EMPLOYEE;
class ADULT includes (select P from PERSON where P.Age≥21);
    attribute ID of type string;
```

Αρχικά, στην όψη My_View γίνεται ορατή πληροφορία από τη βάση δεδομένων EMPLOYEE -συγκεκριμένα η κλάση PERSON- χρησιμοποιώντας τον όρο *import*. Στο συγκεκριμένο μοντέλο δεδομένων μία δήλωση τύπου import εισάγει μία κλάση (ή περισσότερες) μαζί με τις υποκλάσεις, τα αντικείμενα, τις τιμές και τη συμπεριφορά της εν λόγω κλάσης (ή κλάσεων). Επομένως, η κλάση PERSON γίνεται ορατή στην όψη My_View, μαζί με τις υποκλάσεις της, τα αντικείμενα, τις τιμές και τη συμπεριφορά της. Κατόπιν, ορίζεται η ιδεατή κλάση ADULT, που περιλαμβάνει όλα τα αντικείμενα της κλάσης PERSON, των οποίων η τιμή του γνωρίσματος Age είναι μεγαλύτερη ή ίση της τιμής 21. Σημασιολογικά, η κλάση ADULT ορίζεται ως υποκλάση της PERSON. Τέλος, αποδίδεται στην κλάση ADULT το γνώρισμα ID, το οποίο είναι τύπου αλφαριθμητικό. Το συγκεκριμένο μοντέλο παρέχει επίσης δυνατότητες ορισμού μεθόδων.

Η εκφραστικότητα του μηχανισμού ορισμού όψεων στο οντοκεντρικό μοντέλο δεδομένων συνοδεύεται όμως και από ένα πλήθος προβλημάτων, τα κυριότερα εκ των οποίων είναι [GBC+97]:

1. **Το οντοκεντρικό μοντέλο δεδομένων είναι πιο εκφραστικό από το σχεσιακό**. Ενώ ένα σχεσιακό σχήμα αποτελείται από ένα σύνολο ανεξάρτητων σχέσεων, ένα οντοκεντρικό σχήμα είναι μία ιεραρχία κλάσεων, όπου οι κλάσεις συνδέονται με σχέσεις κληρονομικότητας. Ένα μοντέλο όψεων πρέπει να ορίζει που θα τοποθετούνται οι όψεις στην (υπάρχουσα) ιεραρχία κλάσεων. Το πρόβλημα αυτό είναι γνωστό ως *πρόβλημα τοποθέτησης των όψεων* (view placement problem). Για την αντιμετώπιση του προβλήματος αυτού έχουν προταθεί τρεις λύσεις. Η πρώτη λύση προτείνει την αυτόματη τοποθέτηση των όψεων στην ιεραρχία των κλάσεων [AB91, R92, LDB97a, LDB97b, LDB98, SLT91). Στη δεύτερη πρόταση, ο χρήστης ορίζει ρητώς τη θέση των όψεων στην ιεραρχία κλάσεων, ενώ στην τρίτη περίπτωση οι ιεραρχίες κλάσεων και όψεων διατηρούνται σε χωριστές ιεραρχίες και μία σχέση συσχετίζει μία βασική κλάση και μία όψη [SAD94, GBC+97]. Στην πρώτη προσέγγιση, δημιουργούνται δύο προβλήματα: η τοποθέτηση των όψεων στο υπάρχον σχήμα της βάσης δεδομένων

είναι γενικά πρόβλημα μη-αποφασίσιμο χρησιμοποιώντας γενικές OQL ερωτήσεις και η ιεραρχία μπορεί να γίνει πολύ μεγάλη και δύσχρηστη. Στη δεύτερη λύση, η ανάμειξη κλάσεων και όψεων στην ίδια ιεραρχία μπορεί να προκαλέσει σύγχυση στους χρήστες, ενώ απαιτεί ταυτόχρονα ελέγχους από το σύστημα για να διασφαλιστεί ότι η καθορισμένη θέση της όψης είναι συνεπής με τον ορισμό της όψης. Η τρίτη περίπτωση δημιουργεί τα λιγότερα προβλήματα και συντελεί στη δημιουργία σχημάτων που είναι πιο κατανοητά από το χρήστη.

2. **Κλειστότητα των σχημάτων όψεων**. Στην περίπτωση των σχημάτων όψεων, χρειάζεται να διασφαλιστούν κάποιοι περιορισμοί συνέπειας. Ένας από τους περιορισμούς είναι η *κλειστότητα*, η οποία όταν ικανοποιείται, εξασφαλίζει ότι όλες οι κλάσεις (βασικές και ιδεατές) που χρησιμοποιούνται έμμεσα ή άμεσα, από την όψη ορίζονται επίσης εντός της όψης (είναι δηλαδή και οι ίδιες ιδεατές κλάσεις). Η κλειστότητα είναι μία ιδιότητα των όψεων, που βρίσκει εφαρμογή κυρίως στην περίπτωση των γνωρισμάτων, όπου υπάρχει η περίπτωση το πεδίο τιμών μίας ιδιότητας να μην είναι ορατό στην όψη. Στην περίπτωση αυτή, που η όψη είναι μη-κλειστή, η κλειστότητα επιτυγχάνεται με την εφαρμογή ενός αλγορίθμου που προσθέτει το μικρότερο σύνολο που χρειάζεται για να μετατραπεί η μη-κλειστή σε κλειστή όψη. Η κλειστότητα δεν αντιμετωπίζεται από πολλά μοντέλα όψεων. Το [R92] προτείνει ένα αλγόριθμο που υπολογίζει, δοθείσας μίας μη-κλειστής όψης, την αντίστοιχη κλειστή όψη, ενώ το [GBC+97] ορίζει τυπικά την ιδιότητα της κλειστότητας.

3. **Τα αντικείμενα έχουν μία ταυτότητα**. Μία όψη στο επίπεδο των δεδομένων θα έπρεπε να είναι μία κλάση. Το ζήτημα είναι τι θεωρούμε στιγμιότυπα της κλάσης: τιμές, υπάρχοντα ή νέα αντικείμενα. Το πρόβλημα αυτό δεν εμφανίζεται στο σχεσιακό μοντέλο δεδομένων, όπου δεν υπάρχει η έννοια της ταυτότητας. Στο οντοκεντρικό μοντέλο δεδομένων, μπορούμε να διακρίνουμε δύο είδη όψεων: *όψεις που διατηρούν αντικείμενα* (object-preserving views) και *όψεις που δημιουργούν αντικείμενα* (object-generating views). Το πρώτο είδος όψεων είναι όψεις που απλώς εξάγουν αντικείμενα από υπάρχουσες κλάσεις. Τα στιγμιότυπα αυτών των όψεων μπορούν να αναγνωριστούν από τα αναγνωριστικά των εξαγόμενων από τη βάση αντικειμένων. Το δεύτερο είδος όψεων είναι όψεις που δημιουργούν νέα αντικείμενα και τα στιγμιότυπα αυτών των όψεων πρέπει να αναγνωρίζονται από εκ νέου δημιουργηθέντα αναγνωριστικά αντικειμένων. Οι περισσότερες προσεγγίσεις θεωρούν μόνο όψεις που διατηρούν αντικείμενα [R92] και με αυτόν τον τρόπο παρέχουν ουσιαστικά διαφορετικές απόψεις των αποθηκευμένων αντικειμένων. Εντούτοις, οι όψεις αυτές δεν είναι αρκετά

ισχυρές για την υποστήριξη όλων των τύπων αναδιοργάνωσης μίας βάσης δεδομένων. Αντιθέτως, όψεις που έχουν τη δυνατότητα δημιουργίας νέων αντικειμένων μπορούν να χρησιμοποιηθούν σε περισσότερες περιπτώσεις [AHV95]. Το μοντέλο δεδομένων που προτείνεται από το [AB91] υποστηρίζει και τα δύο είδη όψεων. Ειδικότερα, διακρίνει 2 είδη όψεων: *ιδεατές κλάσεις* (virtual classes)[30], οι οποίες έχουν μόνο επιλεγμένα από ήδη υπάρχουσες κλάσεις αντικείμενα και *φανταστικές κλάσεις* (imaginary classes), οι οποίες αποκτούν πληθυσμό με αντικείμενα των οποίων τα αναγνωριστικά δημιουργούνται εκ νέου με κάθε αποτίμηση της όψης. Για το λόγο αυτό, ανακύπτει το πρόβλημα της ανάθεσης του ίδιου αναγνωριστικού στο ίδιο αντικείμενο σε διαφορετικές αποτιμήσεις της όψης. Οι ιδεατές κλάσεις ορίζονται με εξειδίκευση ή γενίκευση βασικών κλάσεων, ενώ οι φανταστικές κλάσεις ορίζονται από επερωτήσεις που επιστρέφουν ένα σύνολο από τιμές. Όψεις που δημιουργούν και διατηρούν αντικείμενα υποστηρίζουν επίσης τα [SAD94, LDB97a, LDB97b, LDB98, GBC+97].

Παρόλη τη δυσκολία στον ορισμό των αντικειμενοστραφών όψεων λόγω της πολυπλοκότητας του υποκείμενου μοντέλου, η ευελιξία ενός μηχανισμού όψεων σε αυτά τα συμφραζόμενα, και κατά συνέπεια ο βαθμός χρησιμότητάς του, είναι μεγάλη. Εκτός των τυπικών λειτουργικοτήτων που προσφέρουν οι σχεσιακές βάσεις δεδομένων, οι αντικειμενοστραφείς όψεις μπορούν να γίνουν αντικείμενο εκμετάλλευσης από πλήθος άλλων περιπτώσεων, όπως είναι η υποστήριξη της **εξέλιξης σχήματος** και η **ολοκλήρωση ετερογενών βάσεων δεδομένων**. Η χρήση των όψεων για την ολοκλήρωση ετερογενών σχημάτων βάσεων δεδομένων επιτελείται με τον ορισμό όψεων που ολοκληρώνουν σημασιολογικά ισοδύναμες κλάσεις που ανήκουν σε διαφορετικά σχήματα. Η χρήση των όψεων στην εξέλιξη σχήματος επιτρέπει στους χρήστες να πειραματιστούν με αλλαγές πάνω στο σχήμα, χωρίς να επηρεάζονται άλλοι χρήστες. Οι όψεις μπορούν να υποστηρίξουν επίσης το **μηχανισμό εκδόσεων** (versioning mechanism), έτσι ώστε οποιοδήποτε αντικείμενο που έχει αποθηκευτεί στη βάση δεδομένων να μπορεί να προσπελαστεί και να τροποποιηθεί από κάθε έκδοση σχήματος. Οι ιδιότητες του μηχανισμού όψεων καθορίζουν το είδος των αλλαγών σχήματος που μπορούν να προσομοιωθούν. Οι όψεις επιτρέπουν επίσης, όπως και στο σχεσιακό μοντέλο δεδομένων, τον **ορισμό**

---

[30] Η ορολογία που ακολουθούμε δεν κάνει διάκριση σε ιδεατές και φανταστικές κλάσεις. Οι κλάσεις που ορίζονται σε επίπεδο όψης ονομάζονται *ιδεατές*, ανεξάρτητα από το περιεχόμενό τους.

εξωτερικών σχημάτων (επίπεδο όψεων), όπου κάθε εξωτερικό σχήμα αποτελείται από ένα σύνολο όψεων και βασικών κλάσεων που καθορίζουν πως αντιλαμβάνεται ο χρήστης τη βάση δεδομένων.

3.1.1.3 Μοντέλο Ημιδομημένων Δεδομένων και XML

Η σύγκλιση ιδεών από τις κοινότητες βάσεων δεδομένων και διαχείρισης εγγράφων συντέλεσε στην επέκταση των τεχνικών διαχείρισης βάσεων δεδομένων, με τρόπο που να καθίσταται εφικτή η επεξεργασία μεγάλου όγκου πληροφοριών που βρίσκονται εκτός συστημάτων βάσεων δεδομένων, όπως δομημένα έγγραφα (π.χ SGML), σελίδες διαδικτυακών κόμβων (HTML), μη-αυστηρά δομημένα επιστημονικά ή εμπορικά δεδομένα (π.χ EDI) κτλ. Η δομή αυτών των (μη-σχεσιακών) δεδομένων είναι πολλές φορές ακανόνιστη, συχνά άγνωστη εκ των προτέρων, ενώ ακόμα και όταν είναι γνωστή, συμβαίνει να μεταβάλλεται χωρίς ειδοποίηση. Για την αναπαράσταση δεδομένων με ασαφώς ορισμένη ή ακανόνιστη δομή, αναπτύχθηκε το μοντέλο των ημιδομημένων δεδομένων ως ένα μοντέλο με ένα δυναμικό σύστημα τύπων, που παρέχει ένα μορφότυπο περιγραφής, στον οποίο τα δεδομένα είναι λιγότερο περιορισμένα από ότι είναι τυπικά στις βάσεις δεδομένων. Την ίδια στιγμή, η κοινότητα διαχείρισης εγγράφων ανέπτυξε τη γλώσσα XML [BPS+00] ως ένα μορφότυπο που προσθέτει περισσότερη δομή στα έγγραφα, προκειμένου να απλοποιήσει και να τυποποιήσει τη μετάδοση δεδομένων μέσω εγγράφων. Οι δύο αναπαραστάσεις, το μοντέλο ημιδομημένων δεδομένων και η XML, είναι πρακτικά παρόμοιες (με εξαίρεση κάποια σημεία, όπως η διάταξη των στοιχείων) και θα αναφέρουμε περιληπτικά τις βασικές έννοιες τους, στοχεύοντας στην παρουσίαση της χρησιμότητας του μηχανισμού όψεων στο περιβάλλον αυτό.

Αν και η δομή τους δεν είναι (τυπικά) ορισμένη, τα ημιδομημένα δεδομένα οργανώνονται σε σημασιολογικές οντότητες, οι οποίες με τη σειρά τους σχηματίζουν ομάδες παρόμοιας σημασίας. Εντούτοις, οι οντότητες της ίδιας ομάδας μπορεί να μην έχουν τα ίδια γνωρίσματα, η παρουσία μερικών γνωρισμάτων δεν επιβάλλεται, ενώ η διάταξη των γνωρισμάτων δεν είναι σημαντική. Το μοντέλο **OEM** (Object Exchange Model), που αναπτύχθηκε στα πλαίσια του συστήματος Tsimmis, αποτελεί ένα από τα μοντέλα δεδομένων που βρίσκει εφαρμογή σε ημιδομημένα δεδομένα [ASB99]. Βασιζόμενο στην έννοια των γράφων, αναπαριστά τα δεδομένα ως μία συλλογή από *αντικείμενα* (αναγνωριζόμενα μοναδικά από ένα αναγνωριστικό oid), κάθε ένα εκ των οποίων

μπορεί να είναι *ατομικό* ή *σύνθετο*. Η τιμή ενός ατομικού αντικειμένου είναι κάποιου βασικού τύπου (ακέραιος, αλφαριθμητικό, εικόνα, ήχος κτλ), ενώ η τιμή ενός σύνθετου αντικειμένου είναι ένα σύνολο από ζεύγη της μορφής *(γνώρισμα, αντικείμενο)*. Παρουσιαζόμενο με πιο τυπικό τρόπο και βασιζόμενοι σε μία παραλλαγή του βασικού OEM μοντέλου που αναπαριστά καλύτερα τη φύση των ημιδομημένων δεδομένων[31], ένα *OEM αντικείμενο* ορίζεται ως μία τετράδα της μορφής *(ετικέτα, oid, τύπος, τιμή)*, όπου «ετικέτα» είναι ένα αλφαριθμητικό χαρακτήρων που δηλώνει τί αναπαριστά το αντικείμενο, «oid» είναι το αναγνωριστικό του αντικειμένου, ενώ ο όρος «τύπος» μπορεί να είναι *ατομικός* (π.χ ακέραιος, αλφαριθμητικό) ή *σύνθετος* (σύνολο ή λίστα). Η «τιμή» μπορεί να είναι ατομική (τιμή ατομικού τύπου) ή ένα σύνολο/λίστα από αναγνωριστικά αντικειμένων, αν το αντικείμενο είναι ατομικό ή σύνθετο αντίστοιχα.

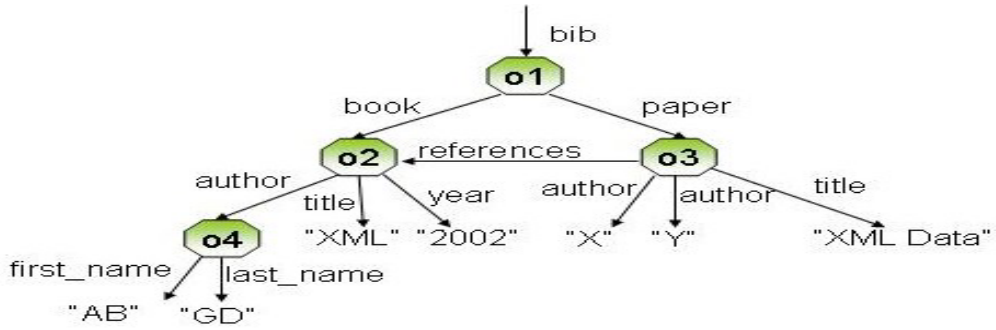
Γραφικά αναπαριστώμενα, τα ημιδομημένα δεδομένα συνιστούν ένα κατευθυνόμενο γράφο, του οποίου οι κατευθυνόμενες ακμές περιγράφονται με ετικέτες τις ετικέτες των αντικειμένων και οι οποίες αποτελούν τμήμα των δεδομένων. Οι κόμβοι του γράφου αναπαριστούν αντικείμενα: οι εσωτερικοί κόμβοι αναπαριστούν σύνθετες τιμές και οι κόμβοι-φύλλα (κόμβοι χωρίς εξερχόμενες ακμές) μία συσχετιζόμενη ατομική τιμή. Ένας OEM γράφος έχει μία ρίζα, δηλαδή ένα διακεκριμένο αντικείμενο, από το οποίο είναι προσπελάσιμα όλα τα άλλα αντικείμενα και για το λόγο αυτό, θα αποκαλείται *ριζωμένος κατευθυνόμενος γράφος*. Η **Εικόνα 3.3** παρουσιάζει ένα παράδειγμα αναπαράστασης βιβλιογραφικών δεδομένων στο μοντέλο ημιδομημένων δεδομένων. Ο κόμβος o1 αποτελεί τη ρίζα του γράφου, τα αντικείμενα o2, o3 και o4 είναι σύνθετα, ενώ το αντικείμενο «XML» είναι ατομικό.

Ορισμός 7: Μία **βάση (OEM) ημιδομημένων δεδομένων** είναι ένας ριζωμένος κατευθυνόμενος γράφος με ετικέτες στις ακμές.

Τα ημιδομημένα δεδομένα είναι *αυτοπεριγραφόμενα*, που σημαίνει ότι το σχήμα τους ενσωματώνεται στα δεδομένα και δεν υιοθετείται δομή εκ των προτέρων. Η σημασιολογική πληροφορία περικλείεται στις ετικέτες, οι οποίες είναι τμήμα των δεδομένων και μπορούν να αλλάξουν δυναμικά. Το γεγονός αυτό μας

[31] Τα OEM δεδομένα, όπως και τα ημιδομημένα δεδομένα, απαρτίζουν ένα γράφο. Η διαφορά έγκειται στο γεγονός ότι στο βασικό OEM μοντέλο οι ετικέτες προσαρτώνται στους κόμβους και όχι στις ακμές, όπως συμβαίνει στα ημιδομημένα δεδομένα [ASB99]. Οι περισσότερες όμως παραλλαγές του OEM μοντέλου δεδομένων, που χρησιμοποιούνται σε πλήθος εφαρμογών, κάνουν χρήση γράφων με ετικέτες στις ακμές, σύμβαση που θα ακολουθήσουμε και εμείς.

δίνει την ευελιξία να επεξεργαστούμε (αποθήκευση, επερώτηση κτλ) οποιαδήποτε δεδομένα και να διαχειριστούμε αλλαγές στη δομή των δεδομένων. Δημιουργεί όμως και ένα σύνολο προβλημάτων, όπως μη αποτελεσματική αποτίμηση και δυσκολία στο σχηματισμό επερωτήσεων. Για την επερώτηση ημιδομημένων δεδομένων έχουν αναπτυχθεί γλώσσες επερώτησης, όπως οι Lorel [AQM+97], UnQL [FFL+97] και YATL [CDS+88].



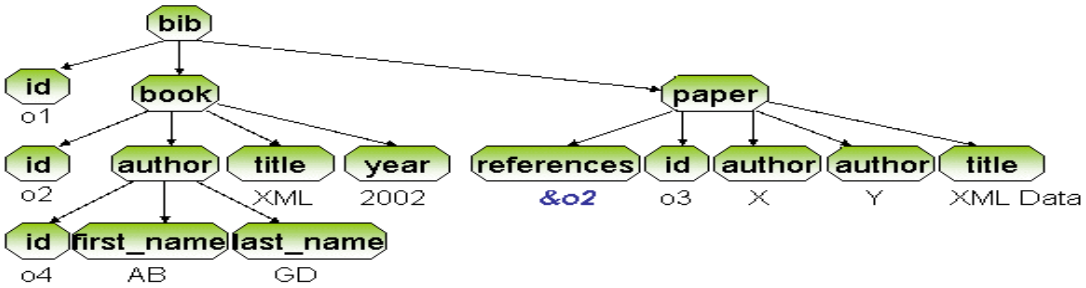
Εικόνα 3.3. Περιγραφή ημιδομημένων βιβλιογραφικών δεδομένων

Η XML [BPS+00]-προερχόμενη από την SGML- είναι μία γλώσσα (εκφραστικότερη της HTML) που επιτρέπει στους χρήστες να ορίζουν νέα ονόματα ετικετών. Τα δεδομένα στην XML ομαδοποιούνται σε *στοιχεία* (elements) που οριοθετούνται από ετικέτες και τα οποία στο σύνολό τους απαρτίζουν ένα *έγγραφο* (ένα στοιχείο είναι το αντίστοιχο του αντικειμένου στα ημιδομημένα δεδομένα). Τα στοιχεία μπορεί να είναι φωλιασμένα σε οποιοδήποτε βάθος και να περιέχουν κείμενο, άλλα στοιχεία (υποστοιχεία) ή και τα δύο. Η XML επιτρέπει τη συσχέτιση *γνωρισμάτων* με στοιχεία, τα οποία ορίζονται ως ζεύγη της μορφής (*όνομα, τιμή*), όπου η τιμή ενός γνωρίσματος είναι πάντα ένα αλφαριθμητικό. Ένα XML έγγραφο είναι *καλώς-ορισμένο* όταν το φώλιασμα των ετικετών γίνεται με συντακτικά σωστό τρόπο[32] και τα γνωρίσματα είναι μοναδικά. Επίσης, η XML επιτρέπει την απόδοση μοναδικού αναγνωριστικού σε κάθε στοιχείο, τα οποίο ορίζεται ως η τιμή ενός συγκεκριμένου XML γνωρίσματος, του *id*. Το αναγνωριστικό αυτό μπορεί να χρησιμοποιηθεί στην περιγραφή ενός άλλου στοιχείου μέσω μηχανισμών αναφοράς.

Τα XML δεδομένα μοιράζονται πολλά χαρακτηριστικά με τα ημιδομημένα δεδομένα: και τα δύο αναπαρίστανται καλύτερα με ένα γράφο, η δομή τους μπορεί να είναι ακαθόριστη, δεν είναι πάντα γνωστή εκ των προτέρων και μπορεί να

[32] Αυτό σημαίνει ότι ακολουθεί τη τυπική σύνταξη της XML: οι ετικέτες είναι συντακτικά σωστές και σωστά φωλιασμένες, κάθε ετικέτα συνοδεύεται με μία ετικέτα τέλους και υπάρχει πάντα ένα στοιχείο-ρίζα.

αλλάζει συχνά και χωρίς ειδοποίηση. Με άλλα λόγια, και τα δύο δεν απαιτούν την ύπαρξη σχήματος και είναι αυτοπεριγραφόμενα. Εντούτοις, διαφέρουν σε ένα αριθμό σημείων [S98]. Για παράδειγμα, η XML επιτρέπει τόσο σε κείμενο όσο και σε υποστοιχεία να παρίστανται σε ένα στοιχείο. Τα XML έγγραφα είναι εξ ορισμού *διατεταγμένα*, ενώ τα ημιδομημένα δεδομένα είναι μη διατεταγμένα. Διαφέρουν επίσης και στη χρησιμοποιούμενη γραφική αναπαράσταση (παρόμοια ως δέντρα, διαφορετικά ως γράφοι): τα ημιδομημένα δεδομένα αναπαρίστανται με ριζωμένους κατευθυνόμενους γράφους με ετικέτες στις ακμές, σε αντίθεση με τα XML δεδομένα, που αναπαρίστανται ως ριζωμένοι κατευθυνόμενοι γράφοι με ετικέτες στους κόμβους. Η **Εικόνα 3.4** παρουσιάζει το γράφο της Εικόνας 3.3 εκφρασμένο σύμφωνα με το μοντέλο των XML δεδομένων. Συν τοις άλλοις, η XML υποστηρίζει περισσότερα επιπλέον στοιχεία, όπως οντότητες[33] (entities), οδηγίες επεξεργασίας (processing instructions) και εισαγωγή σχολίων (comments). Για το λόγο αυτό, θα επικεντρωθούμε στο μοντέλο δεδομένων της XML, που αποτελεί το W3C πρότυπο για τη διακίνηση πληροφορίας στο διαδίκτυο.



Εικόνα 3.4. Περιγραφή βιβλιογραφικών δεδομένων στην XML

Το μοντέλο δεδομένων της XML στηρίζεται στην έννοια του δάσους *διατεταγμένων* δέντρων με ετικέτες στους κόμβους [ASB99]. Πιο συγκεκριμένα, ένα XML έγγραφο αναπαρίσταται ως ένας ριζωμένος κατευθυνόμενος γράφος[34], που περιέχει δύο ειδών κόμβους: κόμβους που αναπαριστούν *στοιχεία* ενός εγγράφου και κόμβους που αναπαριστούν *τιμές δεδομένων*. Κάθε κόμβος στοιχείου αναγνωρίζεται από ένα μοναδικό, αμετάβλητο αναγνωριστικό[35]. Σε κάθε δέντρο οι απόγονοι εσωτερικών κόμβων είναι διατεταγμένοι, φέρουν (αλφαριθμητικές) ετικέτες που δηλώνουν τί

[33] Οι οντότητες είναι επώνυμα τμήματα πληροφορίας, που μπορούν να αποθηκευτούν χωριστά από ένα έγγραφο, αλλά μπορούν να συμπεριληφθούν σε ένα έγγραφο μία ή περισσότερες φορές μέσω αναφοράς του ονόματός τους. Ουσιαστικά, επιτρέπουν τον ορισμό σύντομων αλφαριθμητικών που αναπαριστούν πιο περίπλοκη πληροφορία.
[34] Εξαιτίας της χρήσης μηχανισμών αναφοράς, ένα XML δέντρο μετατρέπεται σε ένα XML ριζωμένο κατευθυνόμενο γράφο.
[35] Το μοντέλο δεν θέτει κάποιο περιορισμό στη μορφή του αναγνωριστικού.

είναι το στοιχείο, ενώ τα φύλλα μπορεί να περιέχουν αναφορές σε κόμβους του ίδιου ή διαφορετικού δέντρου. Επομένως, τα XML έγγραφα έχουν την δυνατότητα να αναφέρονται σε στοιχεία μέσα σε άλλα XML έγγραφα. Τα ονόματα των ετικετών, σε αντίθεση με το σχεσιακό και οντοκεντρικό μοντέλο δεδομένων, ενδέχεται να προέρχονται από διαφορετικούς χώρους ονοματοδοσίας, χρησιμοποιώντας το μηχανισμό των χώρων ονοματοδοσίας της XML [BHL99]. Συν τοις άλλοις, ένας XML γράφος περιέχει κατευθυνόμενες ακμές τριών τύπων: *ακμές φωλιάσματος στοιχείων* (κάθε ακμή συνδέει ένα κόμβο πρόγονο με ένα κόμβο απόγονο, που μπορεί να είναι στοιχείο ή τιμή δεδομένων), *ακμές γνωρισμάτων* (συνδέουν στοιχεία με τιμές δεδομένων) και *ακμές αναφοράς* (συνδέουν στοιχεία με άλλα στοιχεία μέσω μηχανισμών αναφοράς, όπως IDREF, IDREFS, Xlinks, URI κτλ). Η διάταξη των απογόνων που συνδέονται μέσω ακμών φωλιάσματος με ένα κόμβο πρόγονο είναι η διάταξη με την οποία οι απόγονοι εμφανίζονται στο στοιχείο πρόγονο. Από τα παραπάνω, προκύπτει ο επόμενος ορισμός.

Ορισμός 8: Μία **βάση XML δεδομένων G** είναι ένας ριζωμένος κατευθυνόμενος γράφος με ετικέτες στους κόμβους, δηλαδή μία εξάδα της μορφής *(N, E, L, λ, r, ο)*, όπου N είναι το σύνολο των κόμβων που απαρτίζουν τον XML γράφο, E είναι μία «μία-προς-πολλά» σχέση ακμών από τους πατρικούς κόμβους στους κόμβους απογόνους, L είναι το σύνολο των χρησιμοποιούμενων ετικετών κόμβων και r είναι η διακεκριμένη ρίζα του γράφου. Το σύμβολο «λ» χρησιμοποιείται για να δηλώσει μία συνάρτηση απόδοσης ετικετών σε κάθε κόμβο του γράφου, δηλαδή λ:N→L, ενώ το σύμβολο «ο» δηλώνει μία συνάρτηση απόδοσης αναγνωριστικών στους κόμβους, τέτοια ώστε ο:N→O, όπου O είναι το σύνολο των αναγνωριστικών κόμβων.

Με άλλα λόγια, μία βάση XML δεδομένων είναι ένα σύνολο από XML έγγραφα. Όπως έχουμε ήδη αναφέρει, τα XML δεδομένα είναι αυτοπεριγραφόμενα. Εντούτοις, κάθε XML έγγραφο μπορεί να περιέχει μία προαιρετική περιγραφή της γραμματικής του, έτσι ώστε να μπορεί να χρησιμοποιηθεί από εφαρμογές που χρειάζεται να εκτελέσουν δομική επικύρωση των δεδομένων (structural validation). Το τμήμα αυτό της γραμματικής ονομάζεται *DTD* (Document Type Descriptor) ή *Ορισμός Τύπου Εγγράφου*. Ένα DTD καθορίζει μέσω γραμματικών πώς μπορούν να φωλιαστούν τα στοιχεία ενός εγγράφου: μπορούμε να ορίσουμε τη διάταξη των στοιχείων, ότι κάποια στοιχεία είναι προαιρετικά ή εμφανίζονται περισσότερες της μίας φοράς ή να αφήσουμε ακαθόριστη τη διάταξη των στοιχείων. Με άλλα λόγια, ένα DTD καθορίζει τα ονόματα όλων των ετικετών και οντοτήτων, τα ονόματα και τις προκαθορισμένες τιμές των γνωρισμάτων και κανόνες για τον τρόπο φωλιάσματος των ετικετών. Ένα

καλώς-ορισμένο έγγραφο είναι *έγκυρο* αν περιέχει ένα κατάλληλο DTD και ακολουθεί όλους τους περιορισμούς που τίθονται από αυτό.

Ως κάποιο σημείο, ένα DTD μπορεί να χρησιμοποιηθεί ως σχήμα για τα δεδομένα που αναπαρίστανται από το XML έγγραφο. Υπάρχουν όμως περιορισμοί που δεν του επιτρέπουν να καταστεί μία εκφραστική γλώσσα ορισμού XML σχημάτων: ένα DTD επιβάλλει μία διάταξη, δεν υπάρχει η έννοια του ατομικού τύπου (όλα αναπαρίστανται ως αλφαριθμητικά) και ένα DTD δεν περιορίζει τον τύπο των IDREF γνωρισμάτων (με αποτέλεσμα να μπορούμε να «δείξουμε» σε κάτι, χωρίς να ξέρουμε τι είναι). Για τους λόγους αυτούς, έχουν γίνει ένα σύνολο προτάσεων για την υιοθέτηση αντίστοιχων γλωσσών, η κυριότερη εκ των οποίων είναι το XML Schema [BFR+01]. Τα XML σχήματα ορίζουν τη δομή των XML εγγράφων και παρέχουν μηχανισμούς για τη συσχέτιση σημασιολογίας σε αυτά. Παρέχοντας τύπους δεδομένων στην XML, τα XML σχήματα αυξάνουν την εκφραστικότητα και τη χρησιμότητα της XML σε πλήθος εφαρμογών. Μολαταύτα, θα περιοριστούμε στη χρήση των DTD ως γλώσσα ορισμού XML σχημάτων, λόγω της μεγαλύτερης χρήσης τους σε σχέση με το XML Schema.

Υποθέτοντας ότι τα XML δεδομένα είναι εκ φύσεως αντικειμενοστραφή (τα στοιχεία ενός DTD μπορούν να θεωρηθούν ως κλάσεις οργανωμένες σε ιεραρχίες, όπου τα ονόματα κλάσεων αντιστοιχούν σε ονόματα στοιχείων, οι ιδιότητες αντιστοιχούν σε γνωρίσματα και η σχέση υπαλληλίας αντιστοιχεί σε σχέσεις φωλιάσματος), ένας μηχανισμός ορισμού όψεων θα πρέπει να επιτρέπει τον ορισμό ιδεατών XML δεδομένων [S98]. Με άλλα λόγια, ο κυριότερος στόχος μίας γλώσσας ορισμού XML όψεων θα πρέπει να είναι η υποστήριξη μηχανισμών για την εισαγωγή στην όψη αντικειμένων (στοιχείων) και ακμών ανάμεσα στα αντικείμενα από μία ή περισσότερες βάσεις δεδομένων, ενώ ταυτόχρονα θα πρέπει να επιτρέπεται η δημιουργία νέων αντικειμένων και ακμών στην όψη. Επομένως, μπορούμε να χρησιμοποιήσουμε τον ακόλουθο ορισμό:

Ορισμός 9: Έστω $G= G_1,..., G_n$ ένα σύνολο από βάσεις XML δεδομένων. Μία **όψη** (view) V πάνω στο G είναι μία νέα βάση XML δεδομένων, δηλαδή ένας ριζωμένος κατευθυνόμενος γράφος με ετικέτες στους κόμβους. Ο ορισμός μίας XML όψης είναι μία συνάρτηση f από το σύνολο των βάσεων δεδομένων στην όψη, η οποία υπολογίζει ουσιαστικά τα XML έγγραφα που απαρτίζουν την όψη:

$$V : G \xrightarrow{f} V$$

Συνεπώς, μία όψη κατά το μοντέλο δεδομένων της XML είναι ένας *υπογράφος* της αρχικής βάσης (ή των αρχικών βάσεων) -πιθανώς με νέα στοιχεία και ακμές που έχουν προστεθεί, του οποίου τα στοιχεία μπορεί να έχουν αντιγραφεί (μερικώς ή πλήρως) ή να είναι αναγνωριστικά αντικειμένων της πηγής[36]. Στην περίπτωση που μία βάση XML δεδομένων συνοδεύεται από ένα σύνολο DTD, τότε ο **Ορισμός 9** μπορεί να τροποποιηθεί, ώστε να θεωρήσουμε μία όψη ως ένα DTD, το οποίο περιγράφει ένα σύνολο XML εγγράφων. Στην περίπτωση αυτή, ο ορισμός μίας όψης είναι μία συνάρτηση f από τα XML έγγραφα της βάσης δεδομένων στα XML έγγραφα της όψης. Και στις δύο περιπτώσεις όμως, μία όψη είναι ένας χώρος εργασίας, όπως οποιαδήποτε άλλη βάση δεδομένων [AGM+97]. Ένας μηχανισμός ορισμού όψεων για XML δεδομένα πρέπει να στηρίζεται, όπως και στο σχεσιακό ή οντοκεντρικό μοντέλο δεδομένων, σε μία γλώσσα επερώτησης. Για τη διαχείριση XML δεδομένων έχουν αναπτυχθεί πολλές γλώσσες επερώτησης [FSW99], όπως τα δύο πρόσφατα W3C πρότυπα XPath [CD99] και XQuery [BCF+02].

Η διαφορά των XML δεδομένων στο διαδίκτυο με το σχεσιακό μοντέλο είναι ότι τα δεδομένα δεν αποθηκεύονται σε σχέσεις, αλλά συλλέγονται με ειδικά προγράμματα (crawler) από το διαδίκτυο. Για παράδειγμα, το σύστημα Xyleme [ACM+02] βασίζεται σε ένα εργαλείο αυτόματης ταξινόμησης που διαμερίζει τα έγγραφα σε θεματικές ομάδες (π.χ η ομάδα *τέχνη* περιέχει όλα τα έγγραφα που σχετίζονται με την τέχνη). Μία όψη στα πλαίσια του συγκεκριμένου συστήματος συνδυάζει αρκετές θεματικές ομάδες σε μία αφηρημένη θεματική ομάδα. Όπως στο σχεσιακό μοντέλο, μία όψη έχει ένα *πεδίο ορισμού*, ένα *σχήμα* και ένα *ορισμό*. Το πεδίο ορισμού μίας όψης είναι ένα σύνολο από θεματικές ενότητες. Το σχήμα μίας όψης είναι ένα αφηρημένο DTD που περιγράφει αφηρημένα έγγραφα, δηλαδή αυτά που περιλαμβάνονται στην όψη. Ο ορισμός μίας όψης είναι παραδοσιακά μία επερώτηση. Για τη μετάφραση μίας επερώτησης πάνω στην όψη, οι αφηρημένες έννοιες αντικαθίστανται από τους αντίστοιχους όρους της βάσης δεδομένων. Εκτός από το Xyleme [ACM+02], υπάρχοντα συστήματα που υποστηρίζουν XML όψεις είναι παραδείγματος χάρη, τα MIX [B99] και ActiveView [AAC+99].

Εν γένει, τα χαρακτηριστικά που (πρέπει να) φέρει μία XML όψη εξαρτώνται από το περιβάλλον στο οποίο εφαρμόζεται και τη λειτουργικότητα που επιθυμεί να παράσχει. Συνεπώς, για το σχεδιασμό ενός μηχανισμού ορισμού όψεων για το

[36] Ο ορισμός αυτός ισχύει και στην περίπτωση των ημιδομημένων δεδομένων.

μοντέλο δεδομένων της XML (και κατ' επέκταση του μοντέλου ημιδομημένων δεδομένων), λαμβάνονται υπόψη ένα σύνολο από παράγοντες. Κατά πρώτο, η XML εφαρμόζεται στο διαδίκτυο. Εξαιτίας της κατανεμημένης φύσης του διαδικτύου, τα δεδομένα προσπελάζονται και ενημερώνονται από πολλούς χρήστες. Ως αποτέλεσμα αυτών, οι XML όψεις πρέπει να λαμβάνουν υπόψη τους θέματα, όπως αντιγραφή δεδομένων και δυναμικά στοιχεία, όπως η *κοινοποίηση αλλαγών*· προβλήματα που αντιμετωπίζονται ήδη στις ομόσπονδες βάσεις δεδομένων. Επομένως, αυτό που απαιτείται είναι ένας δηλωτικός ορισμός όψεων στο διαδίκτυο με δυναμικά χαρακτηριστικά και έλεγχο αλλαγών, όπως επιχειρείται για παράδειγμα στο σύστημα Active View [AAC+99].

Επιπλέον, λόγω των κατανεμημένων δεδομένων και του κατανεμημένου χώρου εργασίας, μία όψη μπορεί να θεωρηθεί ως ένας χώρος εργασίας που μπορεί να περιέχει προηγούμενες επερωτήσεις και τα αποτελέσματα αυτών. Ταυτόχρονα, η διασπορά των πληροφοριακών πηγών και η δυναμική τους εγείρουν θέματα που σχετίζονται με την απάντηση επερωτήσεων χρησιμοποιώντας όψεις και τη διαχείριση αντιγράφων πληροφορίας. Σε αυτό το πλαίσιο πρέπει επίσης να καθοριστεί η σημασιολογία των επερωτήσεων, δηλαδή τα δεδομένα και τα αντικείμενα που πρέπει να μεταφερθούν από τις διασπαρμένες πηγές, ενώ οι όψεις στο διαδίκτυο πρέπει να προσεγγίζονται με ένα μοντέλο ατελούς πληροφορίας. Αυτό οφείλεται στο γεγονός ότι το διαδίκτυο είναι δυναμικό και οι πηγές δεδομένων δεν βρίσκονται πάντα σε σύνδεση με το διαδίκτυο. Ένα ζήτημα που ανακύπτει σε αυτή την περίπτωση είναι πώς να θέσουμε επερωτήσεις για να συμπληρώσουμε ελλιπείς απαντήσεις [A99]. Συν τοις άλλοις, εκτός από το να λαμβάνει υπόψη της παράγοντες του περιβάλλοντος, μία γλώσσα επερώτησης που επιθυμεί να υποστηρίζει XML όψεις πρέπει να πληροί ένα σύνολο από προϋποθέσεις. Μία συστηματική παρουσίαση αυτών πραγματοποιείται στο [BLP+98]. Παραδείγματος χάριν, μία γλώσσα ορισμού XML όψεων θα πρέπει πάντα να παράγει XML έγγραφα, τα οποία θα πρέπει να αντιμετωπίζονται όπως τα βασικά XML έγγραφα.

Παρά τις δυσκολίες και τη περιπλοκότητα στον ορισμό ενός μηχανισμού ορισμού όψεων για XML δεδομένα, η χρησιμότητα των όψεων, όπως και στα προηγούμενα μοντέλα δεδομένων είναι αδιαμφισβήτητη. Μία από τις πλευρές της χρησιμότητας των όψεων όταν αναφερόμαστε σε ημιδομημένα και XML δεδομένα είναι η χρήση τους για την **εισαγωγή κάποιας δομής**. Μία όψη μπορεί να ομαδοποιήσει τμήματα μίας βάσης δεδομένων σε λογικές μονάδες. Ο ορισμός και η επεξεργασία μίας επερώτησης πάνω σε μία όψη μπορεί να είναι απλούστερη και

αποτελεσματικότερη από την εφαρμογή της επερώτησης σε ολόκληρη τη βάση δεδομένων. Για παράδειγμα, ας υποθέσουμε μία ογκώδη αποθήκη δεδομένων που ολοκληρώνει πληροφορία σχετικά με εκατομμύρια άτομα από πολλές ετερογενείς πηγές. Στην αποθήκη δεδομένων, ένα άτομο μπορεί να αναπαρίσταται από αντικείμενα με πολλές διαφορετικές δομές, αλλά μία όψη μπορεί να αναπαραστήσει τα αντικείμενα με πιο δομημένο και κανονικό τρόπο. Με αυτό τον τρόπο, δημιουργούνται αυτοδύναμες βάσεις δεδομένων από την αρχική βάση δεδομένων.

Μία άλλη χρησιμότητα των XML όψεων είναι σε συστήματα wrapper-mediator, που χρησιμοποιούν την XML ως μέσο για την μοντελοποίηση και την ανταλλαγή πληροφορίας [CCS00]. Ο wrapper, που συσχετίζεται με κάθε πηγή, εξάγει μία XML όψη της πληροφορίας που αποθηκεύεται στην πηγή. Ο mediator (διαμεσολαβητής) είναι υπεύθυνος για την επιλογή, την αναδόμηση και τη συγχώνευση της πληροφορίας από αυτόνομες πηγές και για την παροχή μίας ολοκληρωμένης XML όψης της πληροφορίας. Σε αυτό το πλαίσιο, εκτός από την **ολοκλήρωση της πληροφορίας από ετερογενείς βάσεις δεδομένων**, μία χρησιμότητα των XML όψεων είναι η **έκδοση σχεσιακών βάσεων δεδομένων στο Διαδίκτυο**, χρησιμοποιώντας XML όψεις [B99]. Με αυτό τον τρόπο, δεδομένα που πριν αποθηκεύονταν σε παραδοσιακές σχεσιακές βάσεις αποκτούν ένα τρόπο μετάδοσης μέσω του διαδικτύου και επεξεργασίας από άλλες εφαρμογές, που σχεδόν σίγουρα θα στηρίζονται σε διαφορετικό (σχεσιακό) σχήμα. Είναι γενικά απλό να μετατρέψουμε δεδομένα από οποιαδήποτε πηγή σε XML, γεγονός που είναι επιθυμητό σε επιχειρήσεις που θέλουν να εξάγουν τις πληροφοριακές τους πηγές σε XML και με αυτόν τον τρόπο να τις καταστήσουν διαθέσιμες σε άλλες XML εφαρμογές στο Διαδίκτυο.

3.1.2 Διαχείριση όψεων

Ένα ζήτημα ορθογώνιο στη χρησιμότητα και το υποκείμενο μοντέλο δεδομένων είναι η διαχείριση των όψεων. Θέματα, όπως ο ορισμός, η επερώτηση, η συντήρηση και η ενημέρωση μίας όψης απαιτούν την προσοχή του διαχειριστή της βάσης δεδομένων και εμφανίζονται ανεξαρτήτως του μοντέλου δεδομένων που ακολουθείται (σχεσιακό, οντοκεντρικό ή οποιοδήποτε άλλο) και της λειτουργικότητας που στοχεύει να παράσχει ο μηχανισμός όψεων [S01]. Ο **ορισμός όψεων** αναφέρεται στη δημιουργία όψεων και είναι ένα θέμα που ανακύπτει σε όλες τις περιπτώσεις. Αυτό

που διαφέρει είναι η γλώσσα ορισμού δεδομένων όψεων και η αντίστοιχη σύνταξη σε επίπεδο εντολών που χρησιμοποιούνται. Για παράδειγμα, τα βασικά δομικά στοιχεία της γλώσσας SQL, τα οποία χρησιμοποιούνται για τον ορισμό και τη διαγραφή μίας όψης είναι τα ακόλουθα:

CREATE VIEW <*όνομα όψης*> AS <select-from-where *επερώτηση*>

DROP VIEW <*όνομα όψης*>

Ο πρώτος ορισμός δημιουργεί μία *ιδεατή* όψη με όνομα «όνομα όψης», ενώ ο δεύτερος καταστρέφει την όψη, με την προϋπόθεση ότι δεν χρησιμοποιείται από κάποια άλλη όψη. Οι ορισμοί αυτοί υποστηρίζουν μόνο *ιδεατές* και όχι *υλοποιημένες όψεις*[37]. Μία ιδεατή όψη διαφέρει από μία υλοποιημένη στο γεγονός ότι μόνο ο ορισμός μίας ιδεατής όψης αποθηκεύεται στη βάση δεδομένων, ενώ μία υλοποιημένη όψη αποθηκεύεται εξ ολοκλήρου (ορισμός και δεδομένα). Μία υλοποιημένη όψη λειτουργεί ως αντίγραφο δεδομένων για άμεση προσπέλαση, καθώς εξαλείφει την ανάγκη επαναϋπολογισμού του ορισμού της όψης κάθε φορά που αυτή χρησιμοποιείται. Για το λόγο αυτό, είναι ιδιαίτερα χρήσιμες σε εφαρμογές με υψηλό ρυθμό επερωτήσεων και πολυπλοκότητα δεδομένων.

Το είδος της υλοποίησης που επιλέγεται για την κατασκευή όψεων (ιδεατές ή υλοποιημένες όψεις) έχει επιπτώσεις σε όλα τα θέματα που αφορούν τη διαχείρισή τους, όπως η ***επερώτηση όψεων***. Όταν χρησιμοποιούνται ιδεατές όψεις, για την απάντηση μίας επερώτησης *Q* πάνω σε ένα σύνολο ιδεατών όψεων *V*, κάθε όνομα όψης που εμφανίζεται στην επερώτηση αντικαθίσταται από την έκφραση ορισμού της όψης δημιουργώντας μία νέα επερώτηση Q΄, η οποία τελικά αποτιμάται πάνω στη βάση δεδομένων για να παράγει την απάντηση στην επερώτηση Q. Η αντικατάσταση αυτή καθίσταται εφικτή λόγω της ιδιότητας της κλειστότητας που διαθέτει η SQL (οι τελεστές μίας σχεσιακής έκφρασης είναι σχέσεις και το αποτέλεσμα της αποτίμησης μίας σχεσιακής έκφρασης είναι πάντα μία σχέση). Γενικότερα όμως, υπάρχουν τουλάχιστον τρεις τρόποι για την επεξεργασία επερωτήσεων πάνω σε μία ιδεατή όψη: (α) η όψη υλοποιείται κατά απαίτηση όταν επερωτάται, (β) η επερώτηση πάνω στην όψη επαναδιατυπώνεται ως μία επερώτηση στη βάση δεδομένων, περιορίζοντας την επεξεργασία της επερώτησης σε

[37] Το ANSI SQL Standard (ANSI X3.135-1992, “Database Language SQL”) δεν υποστηρίζει υλοποιημένες όψεις. Προσπάθειες για την υποστήριξη υλοποιημένων όψεων γίνονται με αφορμή τη νέα έκδοση του προτύπου SQL (η τελευταία έκδοση του SQL προτύπου υιοθετήθηκε το 1999 και αποκαλείται συχνά SQL3 99. Τα τυπικά ονόματα του προτύπου αυτού είναι τα ANSI X3.135-1999 ή ISO/IEC 9075:1999, "Database Language SQL", Parts 1 ("Framework"), 2 ("Foundation"), and 5 ("Bindings")).

αντικείμενα που καθορίζονται από τον ορισμό της όψης και (γ) δοθείσας της επερώτησης, επανακαθορίζεται η όψη έτσι ώστε, όταν υλοποιηθεί, μόνο το τμήμα της όψης που είναι συναφές με την επερώτηση να κατασκευαστεί [AGM+97]. Στην περίπτωση που χρησιμοποιούνται υλοποιημένες όψεις, οι επερωτήσεις πάνω σε αυτές αποτιμούνται απευθείας πάνω στις όψεις και δεν χρειάζεται πρόσβαση στο αρχικό σύνολο δεδομένων της βάσης δεδομένων.

Μία εξίσου σημαντική δραστηριότητα, που αποτελεί μέρος της διαχείρισης όψεων, είναι η ***συντήρηση όψεων***. Η συντήρηση όψεων έχει να κάνει με το γεγονός ότι όταν αλλάζει μία βάση δεδομένων, οι υλοποιημένες όψεις πρέπει να τροποποιούνται αναλόγως για να είναι συνεπείς προς τη βάση δεδομένων. Οι αλλαγές μπορεί να προέρχονται από χρήστες που είτε μεταβάλλουν σχέσεις της βάσης δεδομένων είτε τροποποιούν στοιχεία των υλοποιημένων όψεων. Ανεξάρτητα από την πηγή των αλλαγών, η συνέπεια μίας όψης προς τη βάση δεδομένων επί της οποίας ορίζεται εξασφαλίζεται όταν, εφαρμόζοντας την έκφραση ορισμού της όψης πάνω στην τρέχουσα κατάσταση της βάσης, προκύπτει η τρέχουσα κατάσταση της όψης. Από τα παραπάνω, είναι προφανές ότι η διαχείριση όψεων αναφέρεται στις υλοποιημένες όψεις και σε περιπτώσεις αλλαγών διεκπεραιώνεται είτε με *επαναϋπολογισμό* της όψης είτε με *αυξητικό τρόπο*. Ο δεύτερος τρόπος είναι λιγότερο δαπανηρός από τον πρώτο και απαιτεί την ύπαρξη αλγορίθμων για τον υπολογισμό του τμήματος της πληροφορίας (και όχι του συνόλου της πληροφορίας, όπως στην πρώτη περίπτωση) που πρέπει τροποποιηθεί για να διατηρηθεί η συνέπεια της όψης (π.χ [AMR+98, KR02]). Η συντήρηση μίας όψης μπορεί να γίνει είτε εντός της συναλλαγής που ενημερώνει τη βάση δεδομένων (άμεση συντήρηση) είτε σε μία ξεχωριστή, επόμενη συναλλαγή, διαφορετική από εκείνη που ενημερώνει τη βάση δεδομένων (ετεροχρονισμένη συντήρηση).

Η συντήρηση όψεων σε ένα μοντέλο δεδομένων βασισμένο σε γράφους, όπως είναι το μοντέλο ημιδομημένων δεδομένων και της XML, είναι βασικά δύσκολη και σύνθετη διαδικασία. Αυτό οφείλεται στο γεγονός ότι, εξαιτίας του διαμοιρασμού δεδομένων, μία ενημέρωση σε ένα αντικείμενο της βάσης δεδομένων που προσπελάζεται μέσω ενός μονοπατιού μπορεί να επηρεάσει μία όψη που συμπεριλαμβάνει το ίδιο αντικείμενο από διαφορετικό μονοπάτι. Στην πραγματικότητα, το πρόβλημα της εξακρίβωσης του πότε μία όψη επηρεάζεται από μία συγκεκριμένη ενημέρωση είναι σημαντικό και η ανάλυση της επίδρασης μίας ενημέρωσης λαμβάνοντας υπόψη μας μόνο τη δήλωση ενημέρωσης και τον ορισμό της όψης είναι γενικά NP-hard πρόβλημα.

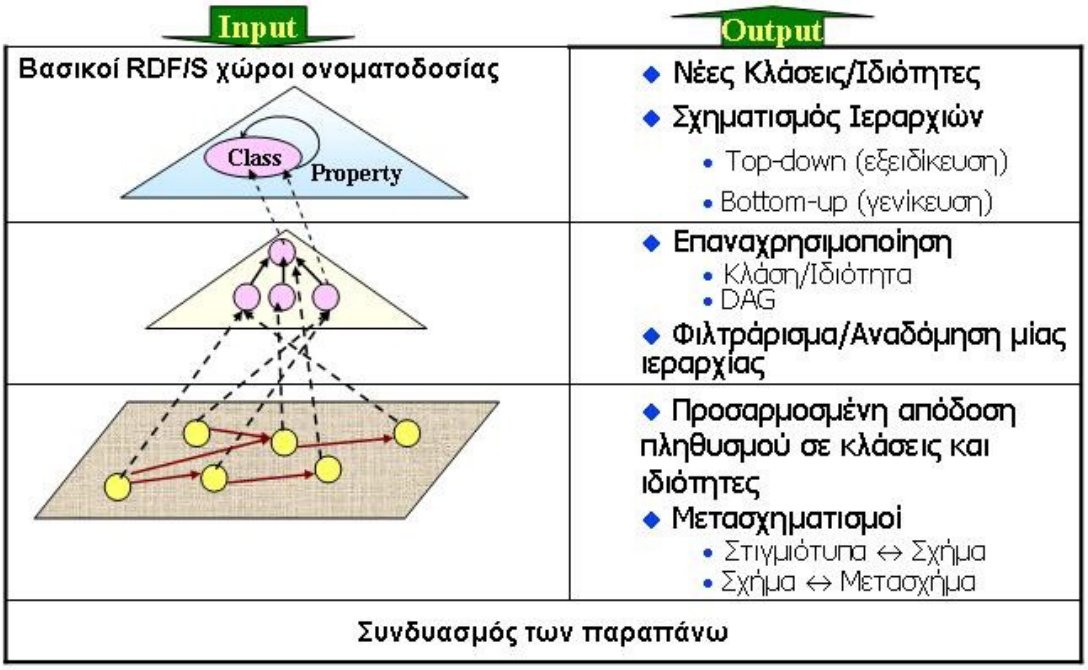
Η ***ενημέρωση των όψεων*** είναι μία συναφής προς τη συντήρηση όψεων διαδικασία και ασχολείται με την μετάφραση ενημερώσεων πάνω στην όψη σε ενημερώσεις στη βάση δεδομένων. Όπως έχουμε ήδη αναφέρει, μπορούμε να ενημερώσουμε μόνο όψεις ανάγνωσης/εγγραφής και η δυσκολία έγκειται αφενός στην εύρεση της κατάλληλης μετάφρασης και αφετέρου στο ότι ενημερώσεις όψεων που στηρίζονται σε άλλες όψεις μπορεί να προκαλέσουν αλυσιδωτές ενημερώσεις. Για κάθε μοντέλο δεδομένων έχουν αναπτυχθεί αλγόριθμοι που υπολογίζουν την κατάλληλη μετάφραση ενημερώσεων όψεων σε ενημερώσεις της βάσης δεδομένων, ανάλογα με το είδος των σχεσιακών (ή άλλων τελεστών) πάνω στους οποίους στηρίζεται ο ορισμός των όψεων. Κάθε μετάφραση είναι ασφαλής όταν εξασφαλίζει τη συνέπεια της όψης και στην περίπτωση που αναφερόμαστε σε υλοποιημένες όψεις, μία ασφαλής μετάφραση συνεπάγεται τη συντήρηση της όψης. Στο οντοκεντρικό μοντέλο δεδομένων, η δυνατότητα ενημέρωσης μίας όψης εξαρτάται από το είδος της όψης. Ειδικότερα, εάν μία όψη παράγεται από μία επερώτηση που διατηρεί τα δεδομένα της βάσης και δεν παράγει νέα αντικείμενα, η όψη μπορεί να ενημερωθεί και οι αλλαγές διαδίδονται στα αντικείμενα της βάσης με μοναδικό τρόπο [SLT91]. Η ενημέρωση όψεων είναι ένα θέμα που έχει απασχολήσει και εξακολουθεί να απασχολεί ερευνητές που ασχολούνται με θέματα βελτιστοποίησης επερωτήσεων (και όχι μόνο) [GM99, AMR+98, S80, BS81b, BS81a, YBS97, HKR02, SLT91, D95b...].

3.2 Όψεις για το Σημασιολογικό Ιστό

Στις προηγούμενες παραγράφους, παρουσιάσαμε εν είδει περίληψης τη θεωρητική υποδομή του μηχανισμού ορισμού όψεων σε σχεσιακά, οντοκεντρικά και XML μοντέλα δεδομένων, καθώς και ζητήματα που άπτονται της ύπαρξης όψεων. Σκοπός αυτής της παραγράφου είναι να καταγράψει τη λειτουργικότητα που απαιτείται για την υποστήριξη εφαρμογών που χρησιμοποιούν τεχνολογία βασισμένη στο RDF/S.

Ειδικότερα, η λειτουργικότητα που πρέπει να υποστηρίζει ένας μηχανισμός ορισμού όψεων εξαρτάται αφενός από τις απαιτήσεις του περιβάλλοντος στο οποίο χρησιμοποιείται και αφετέρου από τις ιδιαιτερότητες του υποκείμενου μοντέλου δεδομένων. Στην περίπτωση του Σημασιολογικού Ιστού, θεωρούμε ως υποκείμενο μοντέλο δεδομένων το μοντέλο των RDF/S γράφων, δηλαδή άκυκλων (σε επίπεδο σχήματος), κατευθυνόμενων γράφων με ετικέτες στις ακμές και στους κόμβους. Από την μελέτη των διαφόρων τύπων ανομοιογενειών που παρουσιάζονται στο

δυναμικό και ετερογενές περιβάλλον του Σημασιολογικού Ιστού [π.χ DS02], μπορούμε να καταλήξουμε σε ένα σύνολο από βασικές λειτουργίες, οι οποίες προσφέρουν την απαιτούμενη ευελιξία για την εφαρμογή του μηχανισμού όψεων σε ποικίλες περιοχές εφαρμογών του Σημασιολογικού Ιστού. Η **Εικόνα 3.5** παρουσιάζει εποπτικά τη ζητούμενη λειτουργικότητα.



Εικόνα 3.5: Λειτουργικότητα προς υποστήριξη

Η στήλη *Είσοδος* στην Εικόνα 3.5 αναφέρεται στα δεδομένα εισόδου της διαδικασίας ορισμού μίας όψης και η στήλη *Έξοδος* στο αποτέλεσμα της χρήσης των κατασκευών ορισμού μίας όψης. Η πρώτη περίπτωση αναφέρεται στη δημιουργία μίας όψης χωρίς τη χρήση προηγούμενης γνώσης. Η μόνη γνώση που χρησιμοποιείται είναι οι βασικοί RDF/S χώροι ονοματοδοσίας και το βασικό μετασχήμα του RDF/S (στοιχεία του RDF/S), βάσει του οποίου κατασκευάζονται οι RDF/S γράφοι. Αποτέλεσμα της διαδικασίας ορισμού όψεων σε αυτή την περίπτωση είναι *ο ορισμός ιδεατών κλάσεων, ιδιοτήτων* και *ιεραρχιών υπαλληλίας κλάσεων/ιδιοτήτων*. Οι ιεραρχίες υπαλληλίας μπορούν να σχηματιστούν μέσω *εξειδίκευσης* (top-down κατασκευή, δημιουργία υποκλάσεων σε μία κλάση) ή/και *γενίκευσης* (bottom-up κατασκευή, δημιουργία υπερκλάσεων μίας κλάσης).

Στη δεύτερη περίπτωση, λαμβάνουμε ως είσοδο μία ιεραρχία κλάσεων ή ιδιοτήτων. Η λειτουργικότητα που θέλουμε να υποστηρίξουμε είναι *η επαναχρησιμοποίηση των ορισμών κλάσεων και ιδιοτήτων*, αλλά και των ίδιων των ιεραρχιών, πιθανώς με αναδιοργανωμένη/φιλτραρισμένη μορφή. Το *φιλτράρισμα*

και *η αναδιοργάνωση των ιεραρχιών* είναι από τις βασικές λειτουργικότητες, δεδομένου ότι επιτρέπουν αντίστοιχα την επιλεκτική επαναχρησιμοποίηση ονομάτων κλάσεων/ιδιοτήτων και των μεταξύ τους σχέσεων υπαλληλίας και την αναδιάταξη των κλάσεων/ιδιοτήτων μίας ιεραρχίας, με τρόπο που να αντανακλά καλύτερα την αντίληψη του χρήστη για το συγκεκριμένο πεδίο εφαρμογής.

Εάν ως είσοδο έχουμε και το αντίστοιχο σύνολο δεδομένων της ιεραρχίας, όπως γίνεται στην τρίτη περίπτωση, *η προσαρμοσμένη απόδοση πληθυσμού* (customized population) σε κλάσεις και ιδιότητες και η *αλλαγή της εννοιολογικής κατάστασης* ενός στοιχείου είναι ανάμεσα στις βασικές λειτουργικότητες που απαιτούνται από εφαρμογές του πραγματικού κόσμου. Η πρώτη λειτουργικότητα ορίζει, κυρίως μέσω μίας επερώτησης πάνω σε έναν ή περισσότερους RDF/S γράφους, τον πληθυσμό (άμεσα στιγμιότυπα) μίας ιδεατής κλάσης/ιδιότητας. Η δεύτερη λειτουργικότητα αναφέρεται στην αλλαγή του επιπέδου αφαίρεσης ενός στοιχείου. Συγκεκριμένα, αναφέρεται στον ορισμό σε μία όψη ενός στιγμιότυπου ως στοιχείο σχήματος ή ενός στοιχείου σχήματος σε στοιχείο επιπέδου μετασχήματος και αντίστροφα. Με τον τρόπο αυτό, για παράδειγμα, ένα στιγμιότυπο ορίζεται ως μία κλάση στην όψη, ενώ μία μετακλάση «υποβιβάζεται» σε κλάση στο επίπεδο σχήματος. Πιο περίπλοκες όψεις μπορούν να οριστούν με το συνδυασμό όλων των παραπάνω περιπτώσεων.

Εν τέλει, η λειτουργικότητα που αναφέραμε προηγουμένως μπορεί να αναχθεί σε ένα σύνολο βασικών κατασκευών για τον ορισμό μίας όψης και συγκεκριμένα στη δημιουργία μίας νέας κλάσης/ιδιότητας, το φιλτράρισμα ενός συνόλου κλάσεων/ιδιοτήτων, τη δημιουργία σχέσεων υπαλληλίας, την απόδοση πληθυσμού σε μία κλάση/ιδιότητα και τη χρήση τελεστών αναδόμησης για το φιλτράρισμα/αναδιοργάνωση ιεραρχιών υπαλληλίας. Η πρόκληση για την υποστήριξη όψεων στο περιβάλλον του Σημασιολογικού Ιστού είναι η υποστήριξη όλων των παραπάνω λειτουργιών μέσω μίας γλώσσας, η οποία συνδυάζει την εκφραστικότητα και την ευελιξία με την απλότητα στον ορισμό των όψεων. Πιστεύουμε, ότι η γλώσσα ορισμού που αποτελεί το αντικείμενο αυτής της εργασίας είναι σε θέση να συνδυάσει την απλότητα με την εκφραστικότητα και την ευελιξία που χρειάζεται για την υποστήριξη εφαρμογών πραγματικής κλίμακας στο περιβάλλον του Σημασιολογικού Ιστού, εκμεταλλευόμενη τη δύναμη της υποκείμενης γλώσσας επερώτησης, της RQL.

Συν τοις άλλοις, το σύνολο των προαναφερθέντων λειτουργικοτήτων που – όπως θα διαπιστώσουμε στο επόμενο κεφάλαιο- υποστηρίζει ο μηχανισμός

ορισμού όψεων που προτείνεται, ενσωματώνει λειτουργίες που υποστηρίζονται σε άλλα περιβάλλοντα βάσεων δεδομένων και συγκεκριμένα στο σχεσιακό, οντοκεντρικό, ημιδομημένων δεδομένων και XML, λαμβάνοντας ταυτόχρονα υπόψη τις ιδιαιτερότητες του καινούργιου μοντέλου δεδομένων. Επομένως, το μεγαλύτερο πλεονέκτημα του μοντέλου όψεων που προτείνουμε είναι ότι το νέο μοντέλο όψεων βοηθάει στην *ενοποίηση με ομοιόμορφο τρόπο* όλων των λειτουργιών των προηγούμενων προσεγγίσεων, υποστηρίζοντας ταυτόχρονα λειτουργίες που σχετίζονται με τη φύση του νέου μοντέλου δεδομένων.

Για παράδειγμα, στο μοντέλο δεδομένων του RDF/S, οι ιδιότητες θεωρούνται «πολίτες πρώτης κατηγορίας», αποδεσμεύοντας τον ορισμό τους από τον ορισμό των κλάσεων, γεγονός που δεν συμβαίνει στο οντοκεντρικό μοντέλο δεδομένων. Η αποδέσμευση αυτή παρέχει μεγαλύτερη ευελιξία και καθιστά περιττούς τους τελεστές απόκρυψης ιδιοτήτων και μεθόδων (π.χ hide) που χρησιμοποιούνται στο οντοκεντρικό μοντέλο. Ταυτόχρονα, η λειτουργικότητα της πρώτης περίπτωσης της Εικόνας 3.5 (δηλαδή η δημιουργία νέων κλάσεων, ιδιοτήτων και ιεραρχιών κλάσεων) σε συνδυασμό με τη δυνατότητα προσαρμοσμένης απόδοσης πληθυσμού σε κλάσεις είναι λειτουργικότητα που απαντάται σε αντικειμενοστραφείς όψεις. Σε αυτό το μοντέλο δεδομένων, η γλώσσα επερώτησης χρησιμοποιείται μόνο ως εργαλείο για την απόδοση πληθυσμού σε μία κλάση και δεν κατέχει τη δυνατότητα επερώτησης του σχήματος μίας οντοκεντρικής βάσης δεδομένων (η έκφραση επερωτήσεων οι οποίες, π.χ ζητούν το σύνολο των υποκλάσεων μίας κλάσης, δεν είναι δυνατή). Αντιθέτως, η RQL δύναται όχι μόνο να εκμεταλλευτεί την ύπαρξη σχήματος, αλλά και να εκφράσει επερωτήσεις πάνω σε αυτό. Υποστηρίζουμε τη λειτουργικότητα αυτή, εκμεταλλευόμενοι ταυτόχρονα την ικανότητα της RQL για επερώτηση σχήματος προκειμένου να ενισχύσουμε την εκφραστική δύναμη της γλώσσας ορισμού όψεων.

Επίσης, η λειτουργικότητα της δεύτερης περίπτωσης που περιγράφεται στην Εικόνα 3.5 και οι μετασχηματισμοί της τρίτης περίπτωσης είναι λειτουργικότητα των ημιδομημένων δεδομένων, με τη διαφορά ότι η έννοια της ιεραρχίας είναι διαφορετική: οι ετικέτες κλάσεων αντιστοιχούν σε ονόματα στοιχείων, οι ιδιότητες αντιστοιχούν σε γνωρίσματα και η σχέση υπαλληλίας αντιστοιχεί σε σχέσεις φωλιάσματος. Παραδείγματος χάριν, όσο αφορά τους μετασχηματισμούς, η γλώσσα YATL [CDS+88], που αποτελεί ένα σύστημα ολοκλήρωσης/μετατροπής ημιδομημένων/XML δεδομένων από ετερογενείς πηγές δεδομένων, έχει τη δυνατότητα να μετασχηματίσει μία τιμή σε ένα όνομα στοιχείου και χρησιμοποιώντας RDF/S ορολογία, ένα στιγμιότυπο σε όνομα κλάσης. Επιπλέον,

ο μηχανισμός των χώρων ονοματοδοσίας της XML είναι λειτουργικότητα της XML, που χρησιμοποιεί το RDF/S για τη χρήση ονομάτων κλάσεων και ιδιοτήτων που ορίζονται σε πολλούς διαφορετικούς RDF/S γράφους. Εκμεταλλευόμαστε το μηχανισμό αυτό για να δημιουργήσουμε όψεις από διαφορετικές πληροφοριακές πηγές και για να αποδώσουμε υπόσταση (μοναδικό αναγνωριστικό) σε μία όψη, της οποίας πλέον τα βασικά δομικά στοιχεία (κλάσεις και ιδιότητες) μπορούν να χρησιμοποιηθούν για τον ορισμό άλλων όψεων.

Τέλος, ο συνδυασμός όλων των περιπτώσεων της Εικόνας 3.5, επιβάλλει την υποστήριξη της δημιουργίας ιδεατών RDF/S γράφων κατά αναλογία με τα ιδεατά σχήματα (view ή virtual schema) των αντικειμενοστραφών βάσεων δεδομένων. Ένα ιδεατό σχήμα ή σχήμα όψης είναι αυτόνομο από το αρχικό σχήμα της βάσης δεδομένων [W99, W98]. Η διάκριση αυτή οδηγεί στην επίτευξη τριών στόχων: τη *λογική ανεξαρτησία* που προσφέρεται από ένα σχήμα όψης, τον ισχυρό *μηχανισμό απόκρυψης και ασφάλειας των δεδομένων* και την *ικανότητα αναδόμησης* ενός σχήματος, με τρόπο που να ικανοποιεί τις ανάγκες των εκάστοτε χρηστών. Ένα σχήμα που αποτελείται από βασικές και ιδεατές κλάσεις, όπως συμβαίνει στην αντίθετη περίπτωση, έχει δύο κύρια μειονεκτήματα. Πρώτον, δεν παρέχει ισχυρό μηχανισμό απόκρυψης των δεδομένων, καθώς ένας χρήστης μπορεί να πλοηγείται μέσω των ιδιοτήτων και σχέσεων υπαλληλίας στο βασικό σχήμα και να έχει πρόσβαση σε δεδομένα που δεν ορίζονται στην όψη. Δεύτερον, δεν παρέχει πλήρη λογική ανεξαρτησία δεδομένων, γιατί δεν υπάρχει διάκριση ανάμεσα στα εξωτερικά σχήματα (όψεις) των διαφόρων χρηστών και το λογικό σχήμα, το οποίο αναπτύσσεται με την ενσωμάτωση ιδεατών κλάσεων. Η δημιουργία μίας ιδεατής κλάσης και η ενσωμάτωση της στο σχήμα της βάσης δεδομένων (λογικό σχήμα), εκτός του ότι αυξάνει το μέγεθος και την πολυπλοκότητα του σχήματος οδηγώντας σε δυσκολίες στη διαχείριση, δημιουργεί σύγχυση στους χρήστες, εφόσον η δημιουργία μίας όψης από ένα χρήστη έχει επίδραση στο σχήμα, επί του οποίου κάποιος άλλος χρήστης ορίζει τις δικές του όψεις. Η δημιουργία ιδεατών σχημάτων, και στα πλαίσια του RDF/S, ιδεατών γράφων μας δίνει όλη την απαιτούμενη ευελιξία για την αντιμετώπιση αυτών των προβλημάτων.

Σκοπός της επόμενης ενότητας είναι, εκτός από το να παρουσιάσει με τυπικό τρόπο το μοντέλο δεδομένων της γλώσσας επερώτησης RQL, να παρουσιάσει το τυπικό μοντέλο των όψεων και τη λειτουργικότητα που υποστηρίζει ο μηχανισμός όψεων, σύμφωνα με τη δηλωτική γλώσσα ορισμού όψεων που προτείνουμε.

ΚΕΦΑΛΑΙΟ 4

Ο Μηχανισμός Ορισμού RDF/S Όψεων

Όπως εξετάσαμε στο προηγούμενο κεφάλαιο, οι όψεις αποτελούν μία τεχνολογία με σημαντική προσφορά υπηρεσιών σε σχεσιακά, οντοκεντρικά, ημιδομημένα και XML περιβάλλοντα βάσεων δεδομένων. Η κυριότερη χρήση τους συνίσταται στην προσαρμογή των αποθηκευμένων δεδομένων στις ανάγκες των επιμέρους εφαρμογών και χρηστών ανεξάρτητα από την ιδιότητά τους (τελικοί χρήστες της πληροφορίας, διαχειριστές του συστήματος βάσεων δεδομένων, ενδιάμεσα προγράμματα εφαρμογών κτλ.). Αναγνωρίζοντας τη χρησιμότητά τους σε παραδοσιακά περιβάλλοντα βάσεων δεδομένων, επιχειρούμε να παράσχουμε μέσω ενός μηχανισμού ορισμού όψεων για το RDF/S, λειτουργικότητα προσαρμοσμένη αφενός στις απαιτήσεις του νέου περιβάλλοντος εργασίας και αφετέρου στις ιδιαιτερότητες του υποκείμενου μοντέλου δεδομένων. Στηριζόμενοι στην RQL, μία εκφραστική, δηλωτική γλώσσα επερώτησης RDF/S γράφων, προδιαγράφουμε μία γλώσσα ορισμού όψεων που εκμεταλλεύεται την εκφραστικότητα αυτή για να σχηματίσει σύνθετες όψεις, οι οποίες μοντελοποιούν με διαφορετικό τρόπο τους περιγραφόμενους πληροφοριακούς πόρους.

Σκοπός της συγκεκριμένης ενότητας είναι η τυπική παρουσίαση της προτεινόμενης γλώσσας ορισμού όψεων και του μοντέλου δεδομένων στο οποίο στηρίζεται. Προκειμένου να εξασφαλιστεί η πληρότητα της παρουσίασης, περιγράφουμε αρχικά το τυπικό μοντέλο δεδομένων στο οποίο στηρίζεται η RQL, αναφέροντας ταυτόχρονα την τυπική ερμηνεία των βασικών RQL επερωτήσεων. Κατόπιν, παρουσιάζουμε το μοντέλο δεδομένων των RDF/S όψεων και ζητήματα που άπτονται του τυπικού ορισμού τους. Το κεφάλαιο ολοκληρώνεται με την παρουσίαση της προτεινόμενης γλώσσας, που χρησιμοποιείται για τον ορισμό των RDF/S όψεων.

4.1 Το τυπικό μοντέλο δεδομένων της RQL

Η ενότητα αυτή παρουσιάζει ένα τυπικό μοντέλο δεδομένων για το RDF/S, πάνω στο οποίο στηρίζεται η γλώσσα επερώτησης *RQL* [KAC+02]. Το τυπικό αυτό μοντέλο γεφυρώνει το πρότυπο RDF [LS99] με τη γλώσσα κατασκευής RDF σχημάτων [BG00]. Συγκρινόμενο με τις προδιαγραφές του RDF/S, η κυριότερη συνεισφορά του RQL μοντέλου δεδομένων έγκειται στην εισαγωγή ενός ημιδομημένου συστήματος τύπων για τα RDF σχήματα και στην αναπαράσταση των RDF περιγραφών ως ατομικές ή σύνθετες τιμές δεδομένων. Η σύνδεση ανάμεσα στους δύο κόσμους επιτυγχάνεται με μία συνάρτηση ερμηνείας τύπων, η οποία (α) δεν επιβάλλει αυστηρούς τύπους στις περιγραφές (π.χ, ένας πόρος μπορεί να περιγραφεί ελεύθερα χρησιμοποιώντας προαιρετικές και επαναλαμβανόμενες ιδιότητες που δεν είναι στενά συνδεδεμένες με κλάσεις), (β) επιτρέπει διαφορετικές περιγραφές των ίδιων πόρων (π.χ, ταξινομώντας τους πόρους κάτω από πολλαπλές κλάσεις, όχι απαραίτητα συνδεδεμένες με σχέσεις υπαλληλίας) και (γ) διευκολύνει την εύκολη επέκταση σχημάτων (π.χ, μέσω εξειδίκευσης κλάσεων και ιδιοτήτων).

Στις επόμενες παραγράφους παρουσιάζονται οι βασικές έννοιες του μοντέλου δεδομένων που υιοθετείται και περιγράφεται το σύστημα τύπων που χρησιμοποιείται από την RQL. Επίσης, αναφέρονται συνοπτικά οι περιορισμοί που επιβάλλονται για τη συνέπεια του τυπικού μοντέλου σε σχέση με τις RDF/S προδιαγραφές και η τυπική ερμηνεία των βασικών RQL επερωτήσεων, το σύνολο των οποίων χρησιμοποιεί η γλώσσα ορισμού όψεων.

4.1.1 Βασικές έννοιες

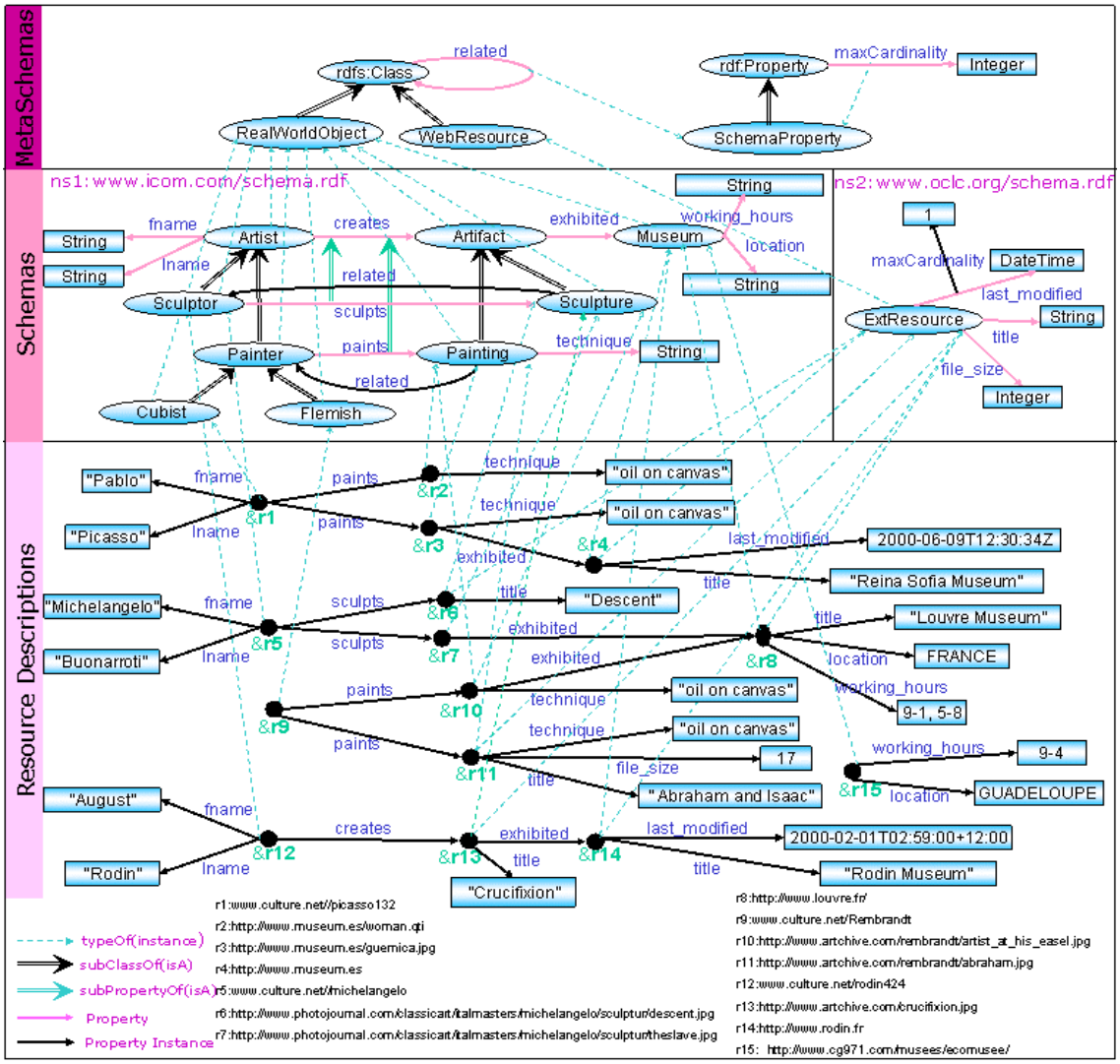
Βασιζόμενο σε ένα μοντέλο κατευθυνόμενων γράφων με ετικέτες σε κόμβους και ακμές, διαφορετικό από το τυπικό οντοκεντρικό μοντέλο, το μοντέλο που παρουσιάζεται διαχωρίζει τους τύπους κλάσεων και ιδιοτήτων και βασίζεται σε ένα μηχανισμό δημιουργίας στιγμιότυπων γράφων, που επιτρέπει την ερμηνεία των στιγμιότυπων από ένα ή περισσότερα RDF σχήματα [ACK+01, KC02]. Όπως έχουμε ήδη αναφέρει στο **Κεφάλαιο 2**, οι RDF περιγραφές πόρων αναπαρίστανται ως *κατευθυνόμενοι γράφοι με ετικέτες*, των οποίων οι κόμβοι καλούνται *πόροι* (ή *βασικές τιμές* και *τιμές συλλογών*) και οι ακμές *ιδιότητες*. Τα RDF σχήματα αναπαρίστανται επίσης ως *κατευθυνόμενοι άκυκλοι γράφοι με ετικέτες* και ουσιαστικά ορίζουν λεξιλόγια ετικετών για κόμβους γράφων, επονομαζόμενους *κλάσεις* (ή *βασικοί τύποι*

και *τύποι συλλογών*) και για ακμές, αποκαλούμενες *τύποι ιδιοτήτων*. Σε αντίθεση με το οντοκεντρικό μοντέλο δεδομένων, τόσο οι κλάσεις όσο και οι τύποι ιδιοτήτων μπορούν να οργανωθούν σε ιεραρχίες.

Η βασικότερη έννοια του RDF/S μοντέλου δεδομένων είναι η έννοια του *πόρου* (resource). Λαμβάνοντας ως κριτήριο το επίπεδο αφαίρεσης στο οποίο ανήκουν και το οποίο χαρακτηρίζει την ικανότητά τους να περιέχουν μέλη, μπορούμε να διακρίνουμε τους RDF πόρους σε *αντικείμενα* (token), *κλάσεις* και *μετακλάσεις*. Ειδικότερα, στο RDF/S μοντέλο δεδομένων που ακολουθούμε μπορούμε να διακρίνουμε τρία επίπεδα αφαίρεσης, που βοηθούν στην κατασκευή σημασιολογικά πλούσιων σχημάτων (Ενότητα 4.1.4). Το κατώτατο επίπεδο αφαίρεσης είναι το *επίπεδο δεδομένων*, όπου γίνεται η περιγραφή των πληροφοριακών πόρων με τη χρήση λεξιλογίων, δηλαδή σύνολα ετικετών κλάσεων και ιδιοτήτων από το αμέσως ανώτερο επίπεδο αφαίρεσης, το *επίπεδο σχήματος*. Ως *αντικείμενα* επομένως χαρακτηρίζουμε σύνολα πραγματικών πόρων, είτε αντικείμενα είτε βασικές τιμές (literals) που ορίζονται στο επίπεδο δεδομένων. Εφεξής, χρησιμοποιούμε τον όρο «πόρος» για να αναφερθούμε σε αντικείμενα που ορίζονται σε επίπεδο δεδομένων, δηλαδή πόρους δεδομένων, οι οποίοι στο σύνολό τους αποτελούν μέλη της βασικής RDF/S κλάσης **rdfs:Resource**.

Το επίπεδο σχήματος είναι το επίπεδο αφαίρεσης όπου αναπτύσσονται RDF σχήματα για να διευκολύνουν τη σημασιολογική περιγραφή των πόρων. Σε αυτό το επίπεδο, οι *κλάσεις* αναπαριστούν αφηρημένες οντότητες και αναφέρονται συλλογικά σε σύνολα παρομοίων αντικειμένων. Το ανώτατο σημασιολογικά επίπεδο, το *RDF/S επίπεδο* ή *επίπεδο μετασχήματος*, χρησιμοποιείται για την ομαδοποίηση κατασκευών σχήματος σε σημασιολογικές μονάδες που αποκαλούμε *μετακλάσεις*. Διακρίνουμε δύο είδη μετακλάσεων: τις *βασικές* RDF/S μετακλάσεις και τις *ορισμένες από το χρήστη* μετακλάσεις, οι οποίες με τη σειρά τους διακρίνονται σε *μετακλάσεις κλάσεων* και *μετακλάσεις ιδιοτήτων*. Οι βασικές RDF/S μετακλάσεις είναι οι **rdfs:Class** και **rdf:Property**, ενώ οι άμεσες υποκλάσεις αυτών, οι οποίες δημιουργούνται με τη χρήση της βασικής ιδιότητας *rdfs:subClassOf*, αποτελούν τις μετακλάσεις κλάσεων και ιδιοτήτων αντίστοιχα. Μία μετακλάση κλάσεων έχει ως μέλη της κλάσεις, ενώ μια μετακλάση ιδιοτήτων έχει ως μέλη της ιδιότητες.

Η **Εικόνα 4.1** παρουσιάζει εποπτικά τη διάκριση των επιπέδων αφαίρεσης σε μία πολιτιστική εφαρμογή για την περιγραφή πληροφοριακών πόρων, όπως έργα τέχνης, δημιουργούς και μουσεία. Τα σχήματα που απεικονίζονται περιγράφουν



Εικόνα 4.1: Παράδειγμα πολιτισμικής εφαρμογής στο Σημασιολογικό Ιστό

τους καλλιτεχνικούς πόρους τόσο σύμφωνα με την καλλιτεχνική τους υπόσταση όσο και από τη διοικητική τους πλευρά, δηλαδή ως διαδικτυακούς πόρους τους οποίους διαχειρίζεται η εφαρμογή.

Το επάνω τμήμα της Εικόνας 4.1 αναφέρεται στο μετασχήμα της εφαρμογής και αποτελείται από τις μετακλάσεις κλάσεων *RealWorldObject* και *WebResource* και τη μετακλάση ιδιοτήτων *SchemaProperty*, της οποίας μέλη είναι οι ιδιότητες *related* και *maxCardinality*. Το μεσαίο τμήμα της εικόνας αναφέρεται στο σχήμα της εφαρμογής. Στην πραγματικότητα πρόκειται για δύο σχήματα[38], εκ των οποίων

[38] Ο ενδιαφερόμενος αναγνώστης μπορεί να πειραματιστεί με τα σχήματα αυτά χρησιμοποιώντας το on-line πρόγραμμα επίδειξης της γλώσσας επερώτησης RQL [RQL]

το ένα (αριστερά) μοντελοποιεί την καλλιτεχνική περιγραφή των διαχειριζόμενων πόρων και το άλλο (δεξιά) παρέχει όλες τις έννοιες που χρειαζόμαστε για να περιγράψουμε τους πόρους βάσει της διοικητικής πληροφορίας που περιέχουν, π.χ τον τίτλο (*title*), την ημερομηνία που τροποποιήθηκε τελευταία φορά (*last_modified*) και το μέγεθος του εκάστοτε πόρου (*file_size*). Εν συντομία, θα χρησιμοποιούμε τα προθέματα *ns1* και *ns2* για να αναφερθούμε αντίστοιχα στα δύο σχήματα. Το σχήμα ns1 διακρίνει βασικές οντότητες, όπως *Artist*, *Artifact* και *Museum*. Μέσω της συμμετοχής τους σε ιεραρχίες, οι κλάσεις *Artist* και *Artifact* εξειδικεύονται περαιτέρω σε *Painter* και *Sculptor* και *Painting* και *Sculpture* αντίστοιχα. Μεταξύ των κλάσεων αναπτύσσεται ένα σύνολο σχέσεων, όπως *creates* και *exhibited*, ενώ ορισμένες κλάσεις φέρουν γνωρίσματα, όπως *working_hours* και *technique*. Σχηματίζοντας μία ιεραρχία ιδιοτήτων, η ιδιότητα *creates* διακρίνεται περαιτέρω στις ιδιότητες *sculpts* και *paints*. Το κάτω τμήμα της εικόνας αναφέρεται στη βάση περιγραφών που έχει σχηματιστεί (επίπεδο δεδομένων) και απεικονίζει ένα τμήμα των περιγραφών που διαθέτει. Για την περιγραφή των πόρων, ισχύουν οι μηχανισμοί του RDF που έχουμε περιγράψει στο **Κεφάλαιο 2**: μία ιδιότητα μπορεί να αποδοθεί περισσότερες από μία φορά σε ένα πόρο (π.χ η ιδιότητα *paints* στον πόρο &r1), η χρήση της μπορεί να είναι προαιρετική (π.χ. η ιδιότητα *lname* δεν αποδίδεται στον πόρο &r9) και ένας πόρος μπορεί να ταξινομηθεί σε πολλές, όχι απαραίτητα συνδεδεμένες με σχέση υπαλληλίας μεταξύ τους κλάσεις (π.χ ο πόρος &r11).

4.1.2 Το τυπικό μοντέλο δεδομένων της RQL

Ακολουθώντας μία πιο τυπική περιγραφή του μοντέλου δεδομένων του RDF/S πάνω στο οποίο στηρίζεται η RQL, υποθέτουμε αρχικά την ύπαρξη των παρακάτω αριθμήσιμα άπειρων και ανά δύο ξένων μεταξύ τους συνόλων συμβόλων:

- $\mathcal{M}$= {m_1, m_2, …}: ονόματα μετακλάσεων
- $\mathcal{C}$ = {c_1, c_2, …}: ονόματα κλάσεων
- $\mathcal{P}$= {p_1, p_2, …}: ονόματα ιδιοτήτων
- $\mathcal{U}$= {u_1, u_2, …}: URI πόρων
- $\mathcal{O}$= {o_1, o_2, …}: τιμές συλλογών
- $\mathcal{L}$= βασικοί τύποι, αλφαριθμητικά, ακέραιοι, ημερομηνίες κτλ.

Τα σύνολα αυτά χρησιμοποιούνται για να ονοματίσουμε τους κόμβους και τις ακμές που συνθέτουν ένα RDF/S γράφο (σε επίπεδο δεδομένων, σχήματος και μετασχήματος). Ειδικότερα, το σύνολο $\mathcal{M}$ αναπαριστά το σύνολο των ονομάτων μετακλάσεων. Δεδομένου ότι διακρίνουμε τις μετακλάσεις σε μετακλάσεις κλάσεων και ιδιοτήτων, συμβολίζουμε με $\mathcal{M}_C$ και $\mathcal{M}_P$ το σύνολο των μετακλάσεων κλάσεων και ιδιοτήτων αντίστοιχα ($\mathcal{M}= \mathcal{M}_C \cup \mathcal{M}_P$). Το σύνολο $\mathcal{M}_C$ συμπεριλαμβάνει επίσης το βασικό RDF/S όνομα *rdfs:Class* και το σύνολο $\mathcal{M}_P$ το βασικό RDF/S όνομα *rdf:Property*, τα οποία αναπαριστούν αντίστοιχα τη ρίζα της ιεραρχίας μετακλάσεων κλάσεων και τη ρίζα της ιεραρχίας μετακλάσεων ιδιοτήτων. Το σύνολο $\mathcal{C}$, εκτός από τα ονόματα των ορισμένων από τους χρήστες κλάσεων, περιέχει και το βασικό όνομα *rdfs:Resource*, το οποίο αποτελεί τη ρίζα της ιεραρχίας των κλάσεων (επίπεδο σχήματος).

Αν και δεν αναπαρίστανται στην Εικόνα 4.1, το RDF/S υποστηρίζει επίσης και δομημένες τιμές, που αποκαλούμε *συλλογές* (containers) (επίπεδο δεδομένων). Το σύνολο $\mathcal{P}$, εκτός από τα ονόματα ιδιοτήτων, περιλαμβάνει επίσης τις αριθμητικές ετικέτες ({1,2,3...}), οι οποίες χρησιμοποιούνται ως ονόματα ιδιοτήτων για τα μέλη των τιμών συλλογών. Το σύνολο όλων των τιμών συλλογών συμβολίζεται με $\mathcal{O}$. Κάθε τιμή RDF/S συλλογών μπορεί να αναγνωριστεί μοναδικά από ένα URI και μπορεί να είναι στιγμιότυπο ενός και μόνο ενός τύπου συλλογών στο *Bt*, δηλαδή *rdf:Bag*, *rdf:Seq* και *rdf:Alt*. Επιπλέον, το πεδίο κάθε βασικού τύπου *t* στο *Lt* (π.χ, αλφαριθμητικό, ακέραιος, ημερομηνία κτλ) συμβολίζεται με *dom(t)*, ενώ το $\mathcal{L}$ αναπαριστά το σύνολο $\cup_{t \in Lt}$dom(t), δηλαδή τον ορισμό του βασικού RDF/S ονόματος *rdfs:Literal*. Στην πραγματικότητα, το *Lt* αναπαριστά το σύνολο όλων των βασικών XML τύπων δεδομένων που μπορούν να χρησιμοποιηθούν σε ένα RDF σχήμα [XML99, MM00].

Κάθε RDF σχήμα χρησιμοποιεί ένα πεπερασμένο αριθμό από ονόματα μετακλάσεων $M \subseteq \mathcal{M}$, κλάσεων $C \subseteq \mathcal{C}$ και ιδιοτήτων $P \subseteq \mathcal{P}$, καθώς και τα σύνολα των ονομάτων τύπων *Lt* και *Bt*. Τα ονόματα ιδιοτήτων ορίζονται χρησιμοποιώντας ονόματα μετακλάσεων, κλάσεων ή βασικών τύπων και τύπων συλλογών, έτσι ώστε για κάθε $p \in P$, $domain(p) \in M \cup C$ και $range(p) \in M \cup C \cup Bt \cup Lt$ (10.1).

Ορισμός 10: Ένας **RDF/S γράφος σχήματος RS** είναι μία εξάδα $RS=(V_S, E_S, \psi, \lambda, \prec, N)$, όπου $N=M \cup C \cup P \cup Bt \cup Lt$ και :

- V_S είναι ένα σύνολο από κόμβους και E_S είναι ένα σύνολο από ακμές, όπου $V_S=M \cup C \cup Bt \cup Lt$ και $E_S=P$ (*10.1*)
- ψ είναι μία συνάρτηση πρόσπτωσης ψ: $E_S \rightarrow V_S \times V_S$ (*10.2*)

- λ είναι μία συνάρτηση απόδοσης ετικετών λ: $V_S \cup E_S \rightarrow 2^M$ (*10.3*)

- $\prec$ είναι μία σχέση υπαλληλίας, τέτοια ώστε:

  - *rdfs:Class* είναι η ρίζα της ιεραρχίας μετακλάσεων κλάσεων

  - *rdf:Property* είναι η ρίζα της ιεραρχίας μετακλάσεων ιδιοτήτων

  - *rdfs:Resource* είναι η ρίζα της ιεραρχίας κλάσεων (*10.4*) □

Η συνάρτηση πρόσπτωσης *ψ* αναπαριστά τα πεδία ορισμού (*rdfs:domain*) και τιμών (*rdfs:range*) των ιδιοτήτων και ενσωματώνει τον περιορισμό ότι το πεδίο ορισμού και τιμών μίας ιδιότητας πρέπει να είναι μοναδικά (10.2). Χρησιμοποιώντας το παράδειγμα της Εικόνας 4.1, η ακμή (ιδιότητα) *creates* συνδέει τους κόμβους (κλάσεις) *Artist* και *Artifact*. Η συνάρτηση απόδοσης ετικετών *λ* αναπαριστά ουσιαστικά την ιδιότητα *rdf:type*, η οποία συνδέει τα ονόματα στο επίπεδο σχήματος με εκείνα στο επίπεδο μετασχήματος (10.3). Συγκεκριμένα, εφαρμοζόμενη στους κόμβους ενός RDF/S γράφου σχήματος, η *λ* επιστρέφει το όνομα μίας ή περισσότερων μετακλάσεων κλάσεων (που μπορεί να ορίζονται σε πολλές διαφορετικές καλά δομημένες ιεραρχίες), ενώ εφαρμοζόμενη στις ακμές επιστρέφει ένα σύνολο ονομάτων μετακλάσεων ιδιοτήτων. Για παράδειγμα, η κλάση σχήματος *Artist* συνδέεται με τη μετακλάση κλάσεων *RealWorldObject* μέσω μίας ακμής *rdf:type*. Οι συναρτήσεις πρόσπτωσης *ψ* και απόδοσης ετικετών *λ* είναι *καθολικές* στα σύνολα Es και Vs∪Es αντίστοιχα, δηλαδή ορίζονται για όλες τις πιθανές τιμές των συνόλων αυτών. Αυτό δεν αποκλείει την περίπτωση κόμβων που δεν συνδέονται μέσω κάποιας ακμής σχήματος. Επίσης, υποθέτουμε ότι οι ετικέτες των κόμβων και ακμών ενός γράφου σχήματος είναι μοναδικές, δηλαδή ισχύει η υπόθεση της *ύπαρξης μοναδικών ονομάτων* (unique name assumption) στο *N* (πιθανώς χρησιμοποιώντας URI χώρων ονοματοδοσίας για αποσαφήνιση). Τα ονόματα (μετα)σχήματος οργανώνονται επίσης σε ιεραρχίες εξειδίκευσης μέσω της σχέσης υπαλληλίας «$\prec$», που αναπαριστά τις βασικές ιδιότητες (ακμές) *rdfs:subClassOf* και *rdfs:subPropertyOf* (10.4).

Προκειμένου να παράσχουμε ένα σαφή ορισμό της σημασιολογίας του RDF/S μοντέλου δεδομένων, εισάγουμε την έννοια ενός *έγκυρου RDF/S γράφου σχήματος*, που επιβάλλει κατάλληλους περιορισμούς στους σχηματιζόμενους RDF/S γράφους σχήματος.

Ορισμός 11: Ένας RDF/S γράφος σχήματος $RS=(V_S, E_S, \psi, \lambda, \prec, N)$ είναι **έγκυρος** αν και μόνο αν:

- Για τη συνάρτηση απόδοσης ετικετών λ, ισχύει ότι:

  - αν $c \in C$ και $\lambda(c)=\{m_1,\ldots, m_n\}$ για κάθε $i \in [1\ldots n]$, $m_i \in M_C$ *(11.1)*

  - αν $p \in P$ και $\lambda(p)=\{m_1,\ldots, m_n\}$ για κάθε $i \in [1\ldots n]$, $m_i \in M_P$ *(11.2)*

- Για τη σχέση υπαλληλίας $\prec$, ισχύει ότι:

  - η σχέση $\prec$ είναι μία αυστηρή μερική διάταξη (μη ανακλαστική, αντισυμμετρική και μεταβατική σχέση[39]) *(11.3)*

  - αν $p_1, p_2 \in P$ και $p_1 \prec p_2$, τότε $domain(p_1) \preccurlyeq domain(p_2)$ και $range(p_1) \preccurlyeq range(p_2)$ *(11.4)*

  - για κάθε $c \prec c'$, $c, c' \in C$ *(11.5)*

  - για κάθε $p \prec p'$, $p, p' \in P$ *(11.6)*

  - για κάθε $m \prec m'$, είτε $m, m' \in M_C$ είτε $m, m' \in M_P$ *(11.7)* □

Ο περιορισμός 11.1 δηλώνει ότι κάθε κλάση πρέπει να είναι στιγμιότυπο μόνο μετακλάσεων κλάσεων. Αντίστοιχα, ο περιορισμός 11.2 δηλώνει ότι μία ιδιότητα πρέπει να είναι στιγμιότυπο μόνο μετακλάσεων ιδιοτήτων. Ο περιορισμός 11.3 επιβάλλει ότι η σχέση υπαλληλίας είναι ουσιαστικά μία *άκυκλη, δυαδική σχέση διάταξης*, έτσι ώστε οι RDF/S ιεραρχίες σε επίπεδο σχήματος και μετασχήματος να σχηματίζουν ένα *κατευθυνόμενο άκυκλο γράφο* (Directed Acyclic Graph, DAG). Με άλλα λόγια, μία RDF/S ιεραρχία υπαλληλίας είναι ουσιαστικά ένα *ημίπλεγμα* (semi-lattice) με ελάχιστο ανώτατο όριο όλων των κόμβων τη ρίζα της ιεραρχίας[40]. Επίσης, συμβολίζουμε με «$\prec^*$» μία ιεραρχία ανεπτυγμένη στην πλήρους μεταβατικής κλειστότητας μορφή της (δηλαδή υπάρχουν όλα τα μονοπάτια από ένα κόμβο στους προγόνους του μέχρι τη ρίζα της ιεραρχίας). Ο

---

[39] Μία σχέση R είναι *μη ανακλαστική* (irreflexive), όταν δεν ισχύει iRi. Στην περίπτωση των ιεραρχιών κλάσεων/μετακλάσεων και ιδιοτήτων, σημαίνει $i \not\prec i$. Μία σχέση R είναι αντισυμμετρική, όταν: αν iRj, τότε δεν ισχύει jRi. Μία σχέση R είναι μεταβατική, όταν: αν iRj και jRk, τότε iRk. Το σύμβολο $\preccurlyeq$ επεκτείνει το σύμβολο $\prec$ με ισότητα (ισχύει η ανακλαστική ιδιότητα).

[40] Ένα *ημίπλεγμα* είναι μία μερική διάταξη $(N, \leq)$, όπου κάθε πεπερασμένο υποσύνολο $A \subseteq N$ έχει ένα ελάχιστο ανώτατο όριο. Ένα *ανώτατο όριο* b ενός υποσυνόλου A ενός συνόλου N με μερική διάταξη $\leq$ είναι μία τιμή που είναι μεγαλύτερη από όλα τα στοιχεία του συνόλου, δηλαδή: $\forall_{a \in A}\ a \leq b$. Ένα *ελάχιστο ανώτατο όριο* έχει την επιπρόσθετη ιδιότητα ότι είναι μικρότερο από όλα τα άλλα όρια c, δηλαδή: $\forall c \in N(\forall_{a \in A}\ a \leq c) \Leftrightarrow b \leq c$. Στην περίπτωση των RDF/S ιεραρχιών, η σχέση διάταξης «$\leq$» ισοδυναμεί με τη σχέση υπαλληλίας «$\preccurlyeq$».

περιορισμός 11.4 επιβάλλει ότι τα πεδία ορισμού και τιμών μιας υπο-ιδιότητας πρέπει να είναι υπάλληλα ονόματα των αντίστοιχων πεδίων της υπερ-ιδιότητας. Οι περιορισμοί 11.5-11.7 επιβάλλουν την εφαρμογή της σχέσης υπαλληλίας ανάμεσα σε ονόματα (μετα)σχήματος του ιδίου τύπου (για παράδειγμα, ιεραρχίες μεταξύ μετακλάσεων και κλάσεων δεν επιτρέπονται). Στην περίπτωση που ισχύουν οι περιορισμοί 11.3-11.7, λέμε ότι η ιεραρχία ονομάτων (μετα)κλάσεων και ιδιοτήτων είναι *καλώς δομημένη*. Εφεξής, χρησιμοποιώντας τον όρο RDF/S γράφος σχήματος, θα εννοούμε ένα έγκυρο RDF/S γράφο σχήματος.

Οι παραπάνω περιορισμοί διασφαλίζουν ότι η *ένωση δύο έγκυρων RDF/S γράφων σχήματος είναι πάντα έγκυρη* σε σχέση με τη σημασιολογία εγκλεισμού (inclusion semantics) της υπαλληλίας (μετα)κλάσεων και ιδιοτήτων. Στο σημείο αυτό μάλιστα μπορούμε να αναφέρουμε τους τυπικούς ορισμούς της *ένωσης δύο έγκυρων RDF/S γράφων σχήματος* και του *υπογράφου ενός έγκυρου RDF/S γράφου σχήματος*. Για τον τυπικό ορισμό του υπογράφου, χρησιμοποιούμε τις συναρτήσεις *domain** και *range**, οι οποίες επιστρέφουν αντιστοίχως το σύνολο των (μετα)κλάσεων στις οποίες μπορεί να εφαρμοστεί ή να καταλήξει μία ιδιότητα. Πιο συγκεκριμένα, ορίζουμε τη συνάρτηση $domain^*: P \rightarrow (M \cup C)$, τέτοια ώστε για κάθε $p \in P$, $domain^*(p)=\{\{c_i\} \mid c_i \preccurlyeq domain(p)\}$, όπου $domain(p)$ είναι συνάρτηση που επιστρέφει το πεδίο ορισμού μίας ιδιότητας. Αντίστοιχα, ορίζουμε τη συνάρτηση $range^*: P \rightarrow (M \cup C \cup Bt \cup Lt)$, τέτοια ώστε για κάθε $p \in P$, $range^*(p)=\{\{c_i\} \mid c_i \preccurlyeq range(p)\}$, όπου $range(p)$ είναι συνάρτηση που επιστρέφει το πεδίο τιμών μίας ιδιότητας.

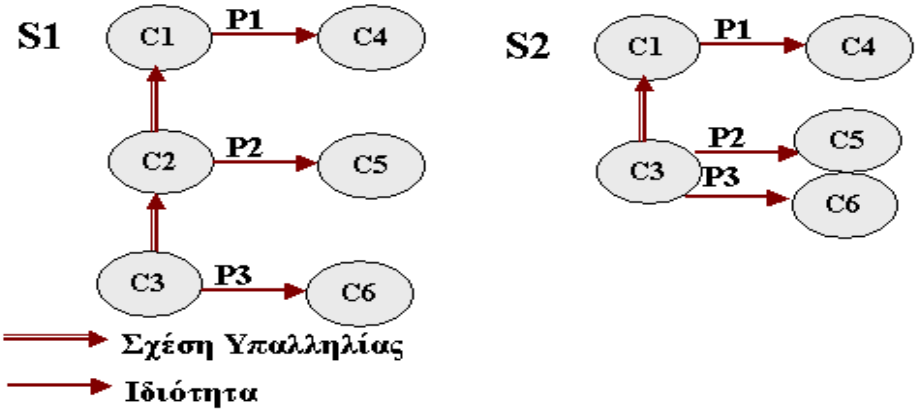
Ορισμός 12: Η **ένωση δύο έγκυρων RDF/S γράφων σχήματος** $RS=(V_S, E_S, \psi, \lambda, \prec, N)$ και $RS'=(V_S', E_S', \psi', \lambda', \prec', N')$ είναι ένας έγκυρος RDF/S γράφος σχήματος $RS''=(V_S'', E_S'', \psi'', \lambda'', \prec'', N'')$, όπου $N''=N \cup N'$ και $\prec''=\prec^* \cup \prec^{*\prime}$. Συμβολίζουμε το γεγονός αυτό με $RS''=RS \sqcup RS'$. □

Ορισμός 13: Ένας έγκυρος RDF/S γράφος σχήματος $RS'=(V_S', E_S', \psi', \lambda', \prec', N')$ είναι **υπογράφος** ενός έγκυρου RDF/S γράφου σχήματος $RS=(V_S, E_S, \psi, \lambda, \prec, N)$, συμβολιζόμενο ως $RS' \Subset RS$, αν και μόνο αν $N' \subseteq N$, $\prec' \subseteq \prec^*$ και ισχύει ότι:

- για κάθε $c, c', c'' \in (M' \cup C' \cup P')$ και $c \preccurlyeq' c'$, $\nexists c''$ τέτοιο ώστε $c \preccurlyeq' c''$ και $c'' \preccurlyeq' c'$ *(13.1)*

- για κάθε $p \in P'$ και $c, c' \in (M' \cup C')$, $domain(p) \in domain^*(p)$, $range(p) \in range^*(p)$ και, εάν $domain(p)=c$, $\nexists c' \in domain^*(p)$, τέτοιο ώστε $c \preccurlyeq' c'$. Αντίστοιχα για το πεδίο τιμών *(13.2)*. □

Όσον αφορά τον **Ορισμό 12** και την έννοια της ένωσης δύο έγκυρων RDF/S γράφων σχήματος, από τον ορισμό της σχέσης υπαλληλίας «$\prec$» μπορούμε να καταλάβουμε ότι οι ιεραρχίες που σχηματίζονται από την ένωση δύο RDF/S γράφων σχήματος είναι άκυκλες. Σύμφωνα με τον **Ορισμό 13**, ένας υπογράφος ενός έγκυρου RDF/S γράφου σχήματος αποτελείται από ένα υποσύνολο των ονομάτων του υπεργράφου του, τις ιεραρχίες και το δίκτυο των ιδιοτήτων που αναπτύσσονται ανάμεσά τους. Ειδικότερα, η συνθήκη 13.1 εξασφαλίζει ότι οι ιεραρχίες που σχηματίζονται σε ένα υπογράφο είναι *ελάχιστες*, που σημαίνει ότι δεν υπάρχουν πλεονασματικές δηλώσεις υπαλληλίας μεταξύ μίας (μετα)κλάσης (ιδιότητας) και των υπερκλάσεών (υπερ-ιδιοτήτών) της. Η συνθήκη 13.2 εξασφαλίζει ότι στην περίπτωση που το πεδίο ορισμού (τιμών) μίας ιδιότητας δεν ορίζεται ως κόμβος του υπογράφου, αλλά η ιδιότητα που εφαρμόζεται (καταλήγει) σε αυτόν θεωρείται τμήμα του υπογράφου, η ιδιότητα αυτή ορίζεται με μοναδικό πεδίο ορισμού (τιμών) την «άμεση» υποκλάση του κόμβου αυτού, η οποία ανήκει στον υπογράφο. Η **Εικόνα 4.2** παρουσιάζει γραφικά την έννοια του υπογράφου ενός RDF/S γράφου σχήματος. Στην εικόνα αυτή, ο RDF/S γράφος σχήματος *S2* αποτελεί υπογράφο του *S1*. Εναλλακτικά, δεδομένου ότι μπορούμε να αναπαραστήσουμε ένα RDF/S γράφο ως ένα σύνολο από τριάδες, μπορούμε να θεωρήσουμε ότι ένας υπογράφος ενός RDF/S σχήματος απαρτίζεται από ένα υποσύνολο του συνόλου των τριάδων που συνιστούν τον υπεργράφο.



Εικόνα 4.2: Υπογράφος ενός έγκυρου RDF/S γράφου σχήματος

Οι περιγραφές πόρων ορίζονται χρησιμοποιώντας ένα πεπερασμένο σύνολο αναγνωριστικών πόρων (URI) $U \subseteq \mathcal{U}$, βασικών τιμών $L \subseteq \mathcal{L}$ ή τιμών συλλογών $O \subseteq \mathcal{O}$ και ονόματα ιδιοτήτων P, έτσι ώστε κάθε $p \in P$ να ξεκινάει από ένα κόμβο στο U και να καταλήγει σε ένα κόμβο στο $U \cup L \cup O$ (14.1).

Ορισμός 14: Μία **RDF περιγραφή πόρων RD**, στιγμιότυπο ενός RDF/S γράφου σχήματος $RS=(V_S, E_S, \psi, \lambda, \prec, N)$, είναι μία πεντάδα $RD=(RS, V_D, E_D, \psi, \lambda)$, τέτοια ώστε:

- V_D είναι ένα σύνολο από κόμβους και E_D είναι ένα σύνολο από ακμές, όπου $V_D=U\cup L\cup O$ και $E_D=P$ *(14.1)*

- ψ είναι μία συνάρτηση πρόσπτωσης ψ: $E_D \rightarrow V_D \times V_D$ *(14.2)*

- λ είναι μία συνάρτηση απόδοσης ετικετών λ: $V_D \rightarrow 2^C \cup Lt \cup Bt$ *(14.3)* □

Η συνάρτηση πρόσπτωσης *ψ* αναπαριστά το σύνολο των σχέσεων και γνωρισμάτων που αποδίδονται στους πόρους (14.2). Η συνάρτηση απόδοσης ετικετών *λ* αναπαριστά ουσιαστικά την ιδιότητα *rdf:type*, η οποία συνδέει τον RDF γράφο δεδομένων με ένα RDF/S γράφο σχήματος (14.3). Συγκεκριμένα, εφαρμοζόμενη σε κόμβους πόρων, η *λ* επιστρέφει το όνομα μίας ή περισσότερων κλάσεων (που μπορεί να ορίζονται σε πολλές διαφορετικές καλά δομημένες ιεραρχίες), ενώ εφαρμοζόμενη σε κόμβους τιμών επιστρέφει το όνομα ενός βασικού τύπου ή ενός τύπου συλλογών. Οι συναρτήσεις πρόσπτωσης *ψ* και απόδοσης ετικετών *λ* είναι *καθολικές* στα σύνολα E_D και V_D αντίστοιχα, δηλαδή ορίζονται για όλες τις πιθανές τιμές των συνόλων αυτών.

Όπως και στην περίπτωση των RDF/S σχημάτων, εισάγουμε στη συνέχεια την έννοια μίας *έγκυρης RDF περιγραφής πόρων*.

Ορισμός 15: Μία RDF περιγραφή πόρων $RD=(RS, V_D, E_D, \psi, \lambda)$ είναι **έγκυρη**, αν και μόνο αν RS είναι ένας έγκυρος RDF/S γράφος σχήματος και:

- για κάθε κόμβο $n \in V_D$
  - αν $n \in U \Rightarrow \lambda(n) \subseteq C$ *(15.1)*
  - αν $n \in L \Rightarrow \lambda(n) \in Lt$ *(15.2)*
  - αν $n \in O \Rightarrow \lambda(n) \in Bt$ *(15.3)*
- για κάθε $p \in E_D$ από ένα κόμβο n σε ένα κόμβο n΄:
  - $\exists c \in \lambda(n), c \preccurlyeq domain(p) \wedge \exists c' \in \lambda(n'), c' \preccurlyeq range(p)$ *(15.4)* □

Ο περιορισμός 15.1 δηλώνει ότι ένας πόρος δεδομένων μπορεί να είναι στιγμιότυπο μόνο κλάσεων. Ο περιορισμός 15.2 (15.3) δηλώνει ότι μία βασική τιμή (τιμή συλλογής) είναι στιγμιότυπο ενός και μόνο ενός βασικού τύπου (τύπου συλλογών). Ο περιορισμός 15.4 επιβάλλει ότι ένας πόρος δεδομένων (ή βασική τιμή ή τιμή συλλογών) στον οποίο εφαρμόζεται (καταλήγει) μία ιδιότητα, πρέπει να

είναι στιγμιότυπο της κλάσης (ή του τύπου) που αποτελεί το πεδίο ορισμού (τιμών) της ιδιότητας. Οι παραπάνω περιορισμοί εξασφαλίζουν ότι η *ένωση δύο έγκυρων RDF περιγραφών πόρων είναι πάντα έγκυρη* σε σχέση με τη σημασιολογία εγκλεισμού της υπαλληλίας κλάσεων και ιδιοτήτων. Εφεξής, χρησιμοποιώντας τον όρο «RDF περιγραφή πόρων», θα εννοούμε μία έγκυρη RDF περιγραφή πόρων.

4.1.3 Το σύστημα τύπων του RQL μοντέλου δεδομένων

Σχήματα ή τύποι έχουν παραδοσιακά χρησιμοποιηθεί στον κόσμο των βάσεων δεδομένων από γλώσσες επερώτησης, όπως η OQL [CBC+00], για ποικίλους λόγους:

- **Σαφής ερμηνεία των δεδομένων**: ένα σύστημα τύπων παρέχει μία σαφή *κατανόηση της φύσης των δεδομένων* που επιστρέφονται από μία επερώτηση. Για παράδειγμα, στα πλαίσια του RDF/S μοντέλου δεδομένων, μπορούμε να καταλάβουμε ότι ένα URI αναγνωρίζει μοναδικά ένα πόρο δεδομένων και όχι μία κλάση ή μία ιδιότητα.
- **Ανίχνευση λαθών και ασφάλεια**: η ύπαρξη κανόνων τύπων μας επιτρέπει αφενός να *εξασφαλίσουμε την ασφάλεια των πράξεων* που τελούνται και αφετέρου να *ελέγξουμε την εγκυρότητα της σύνθεσής τους*. Για παράδειγμα, αριθμητικές πράξεις πάνω σε ονόματα κλάσεων δεν έχουν έννοια.
- **Καλύτερη απόδοση**: ένα σύστημα τύπων μπορεί να προσφέρει πολύτιμες πληροφορίες για το σχεδιασμό αποτελεσματικών μέσων αποθήκευσης. Στα συμφραζόμενα του RDF/S, εκτός από τη σχεδίαση αποδοτικότερης αποθήκευσης για τους RDF/S γράφους, ένα σύστημα τύπων μπορεί να διευκολύνει την αποτελεσματική επεξεργασία των επερωτήσεων (π.χ επαναδιατύπωση εκφράσεων μονοπατιών).

Για την υιοθέτηση ενός συστήματος τύπων για το RDF/S μοντέλο δεδομένων πρέπει να ληφθούν υπόψη ένα σύνολο παραγόντων. Αρχικά, σε αντίθεση με τα σχήματα του οντοκεντρικού μοντέλου δεδομένων, οι RDF/S ιδιότητες (γνωρίσματα και σχέσεις) αποτελούν αυθύπαρκτες οντότητες. Παραδείγματος χάριν, η ιδιότητα *creates* μπορεί να επερωτηθεί, ανεξάρτητα αν εφαρμόζεται σε πόρους που είναι στιγμιότυπα της κλάσης *Artist* ή της υποκλάσης της *Painter*. Επίσης, εξαιτίας της ύπαρξης επιπέδων αφαίρεσης (δεδομένα-σχήμα-μετασχήμα), τα στιγμιότυπα ενός τύπου μπορεί να είναι δεδομένα για κάποιον άλλο τύπο. Για παράδειγμα, τα στιγμιότυπα της μετακλάσης *rdf:Property* είναι ιδιότητες σχήματος, όπως η ιδιότητα *creates*, ενώ τα στιγμιότυπα αυτής της ιδιότητας είναι (δυαδικές) ακολουθίες. Τέλος, οι τιμές

συλλογών μπορεί να έχουν ετερογενή περιεχόμενα, όπως βασικές τιμές ή τιμές άλλων συλλογών, αναγνωριστικά πόρων ή ονόματα (μετα)κλάσεων και ιδιοτήτων.

Λαμβάνοντας υπόψη τις παραπάνω διαπιστώσεις, το σύστημα τύπων του μοντέλου δεδομένων της RQL, το οποίο υιοθετούμε ορίζεται ως:

$$\tau = \tau_{Mc}|\tau_{Mp}|\tau_C|\tau_P[\tau,\tau]|\tau_U|\tau_L|\{\tau\}|[1{:}\tau_1, 2{:}\tau_2, \ldots, n{:}\tau_n]|(1{:}\tau_1 + 2{:}\tau_2 + \ldots + n{:}\tau_n)$$

όπου τ_{Mc} είναι ο τύπος μίας μετακλάσης κλάσεων, τ_{Mp} μίας μετακλάσης ιδιοτήτων, τ_C είναι ο τύπος μίας κλάσης, $\tau_P[\tau,\tau]$ ο τύπος μίας ιδιότητας, τ_U ο τύπος για τα URI αναγνωριστικά πόρων (συμπεριλαμβανόμενων και των URI των χώρων ονοματοδοσίας), τ_L είναι ένας βασικός τύπος στο *Lt*, {.} είναι ο τύπος πολυσυνόλου (Bag), [.] είναι ο τύπος μίας ακολουθίας (Sequence) και (.) είναι ο τύπος Alternative. Ο τελευταίος τύπος χρησιμοποιείται στο μοντέλο της RQL για να αναπαραστήσει τη σημασιολογία της ένωσης ή μεταβλητών τύπων (variant) [C88] και είναι επίσης διατεταγμένος. Δεδομένης της προκαθορισμένης διάταξης των ετικετών για τις ακολουθίες και τους τύπους alternative, οι ετικέτες μπορούν να παραληφθούν (για τα πολυσύνολα, οι ετικέτες δεν έχουν σημασία). Επίσης, δεν ορίζεται σχέση υποτύπων (subtyping) στο RDF/S. Οι τύποι που απαρτίζουν το σύστημα τύπων είναι αμοιβαία αποκλειόμενοι και το σύνολο όλων των τύπων που μπορούν να κατασκευαστούν από τα ονόματα (μετα)κλάσεων και ιδιοτήτων, τα αναγνωριστικά πόρων και τις βασικές τιμές ή τις τιμές συλλογών συμβολίζεται με **T**.

Η RQL εκμεταλλεύεται την ύπαρξη του συστήματος τύπων και με τον τρόπο αυτό είναι σε θέση να υποστηρίξει σύνθετες επερωτήσεις, που δημιουργούνται με συναρτησιακό τρόπο. Για τη γλώσσα ορισμού όψεων που θα προδιαγράψουμε εν συνεχεία, το σύστημα τύπων αποτελεί σημαντικό εργαλείο, καθώς επιτρέπει αφενός τον έλεγχο της εγκυρότητας των ορισμών όψεων και αφετέρου την εξαγωγή συμπερασμάτων σχετικά με τον τύπο των ιδεατών κατασκευών. Επίσης, το σύστημα τύπων της RQL μας παρέχει όλα τα απαραίτητα στοιχεία για την αναπαράσταση συλλογών με ομοιογενή αλλά και ετερογενή περιεχόμενα, καθώς επίσης και για την ομοιόμορφη ερμηνεία των μετακλάσεων, κλάσεων και ιδιοτήτων που ορίζονται σε RDF σχήματα. Οι RDF/S κλάσεις και μετακλάσεις μπορούν να ερμηνευτούν ως μοναδιαίες σχέσεις (δηλαδή πολυσύνολα) τύπου $\{\tau_U\}$ (επίπεδο δεδομένων) και $\{(\tau_C+\tau_P)\}$ (επίπεδο σχήματος) αντίστοιχα, ενώ οι ιδιότητες μπορούν να ερμηνευτούν στο επίπεδο δεδομένων ως δυαδικές σχέσεις (δηλαδή πολυσύνολα από ακολουθίες) τύπου $\{[\tau_U,\tau_U]\}$ για σχέσεις και $\{[\tau_U, \tau_L]\}$ για ατομικά γνωρίσματα. Όταν το πεδίο τιμών μίας ιδιότητας είναι τύπου συλλογής, τότε η αντίστοιχη ερμηνεία περιλαμβάνει τιμές συλλογών ενός κατάλληλου τύπου (bag, sequence ή

alternative). Ο συμβολισμός $\tau_P[\tau,\tau]$ για τους τύπους ιδιοτήτων δηλώνει τον ακριβή τύπο του πεδίου ορισμού και τιμών της ιδιότητας (πρώτη και δεύτερη θέση στην ακολουθία αντίστοιχα). Ιδιότητες των οποίων το πεδίο ορισμού και τιμών είναι μετακλάσεις ερμηνεύονται (σε επίπεδο σχήματος) ως $\{[(\tau_C+\tau_P), (\tau_C+\tau_P)]\}$, ανάλογα αν το πεδίο ορισμού ή τιμών της ιδιότητας είναι μετακλάση κλάσεων ή μετακλάση ιδιοτήτων.

Γενικά, τα ονόματα (μετα)σχήματος στο RDF/S έχουν διττή φύση, τόσο ως ετικέτες όσο και ως συλλογές (η αποσαφήνιση εξαρτάται από τα συμφραζόμενα της λειτουργίας που επιτελείται). Παραδείγματος χάριν, το όνομα *Artist* αναφέρεται στο όνομα του κόμβου σχήματος με την ίδια ετικέτα (ετικέτα κλάσης), αλλά και στα αναγνωριστικά πόρων που ορίζονται ως άμεσα και έμμεσα στιγμιότυπα της κλάσης αυτής. Η υποστασιοποίηση (instantiation) των ονομάτων (μετα)σχήματος με ένα πεπερασμένο σύνολο αναγνωριστικών πόρων URIs (ή ονομάτων σχήματος) αναπαρίσταται με κατάλληλες *συναρτήσεις απόδοσης πληθυσμού* (population function).

Ορισμός 16: Μία **συνάρτηση απόδοσης πληθυσμού στις κλάσεις**, $\pi_C:C\rightarrow 2^U$, αναθέτει ένα πεπερασμένο σύνολο αναγνωριστικών πόρων σε μία κλάση σχήματος, έτσι ώστε:

- για κάθε $c, c' \in C$, $c \preccurlyeq c'$, $v \in \pi_C \Rightarrow v \notin \pi_{C'}$ *(16.1)* □

Προκειμένου να αναπαραστήσουμε την απόδοση μελών σε μετακλάσεις κλάσεων και ιδιοτήτων, ορίζουμε επίσης τις συναρτήσεις απόδοσης πληθυσμού $\pi_{M_C}:M_C\rightarrow 2^C$ και $\pi_{M_P}: M_P\rightarrow 2^P$. Αξίζει να σημειώσουμε ότι οι συναρτήσεις απόδοσης πληθυσμού π είναι ουσιαστικά οι αντίστροφες συναρτήσεις των συναρτήσεων απόδοσης ετικετών λ, που χρησιμοποιούνται σε κάθε επίπεδο αφαίρεσης (Ορισμοί 10 και 14). Οι συναρτήσεις απόδοσης πληθυσμού είναι *μερικές*, εξαιτίας του γεγονότος ότι ενδέχεται να υπάρχουν (μετα)κλάσεις χωρίς πληθυσμό. Επιπλέον, λόγω της πολλαπλής ταξινόμησης πόρων (ή ονομάτων κλάσεων και ιδιοτήτων) κάτω από διαφορετικές (μετα)κλάσεις, οι αναθέσεις πληθυσμών π_C σε κλάσεις (ή π_{M_C} και π_{M_P}) είναι *μη ξένες* μεταξύ τους (non-disjoint). Εντούτοις, η συνθήκη 16.1 επιβάλλει ότι ένα αναγνωριστικό πόρου (αντίστοιχα ένα όνομα κλάσης ή ιδιότητας) εμφανίζεται μόνο μία φορά στην έκταση μίας (μετα)κλάσης, έστω και αν μπορεί να ταξινομηθεί περισσότερες από μία φορές στις υποκλάσεις της (με άλλα λόγια, ανήκει στην «πλησιέστερη» κλάση σε σχέση με την οριζόμενη ιεραρχία υπαλληλίας). Πρέπει να τονιστεί ότι εξ ορισμού χρησιμοποιούμε μία *επεκταμένη*

ερμηνεία (μετα)κλάσεων, συμβολιζόμενη ως π^*, που συμπεριλαμβάνει στα άμεσα στιγμιότυπα μίας (μετα)κλάσης και τα στιγμιότυπα των υποκλάσεων της.

Το σύνολο όλων των τιμών που μπορεί κάποιος να κατασκευάσει από τα ονόματα κλάσεων και ιδιοτήτων, τα URIs πόρων και τις βασικές τιμές ή τιμές συλλογών χρησιμοποιώντας το RQL σύστημα τύπων συμβολίζεται ως ***V*** και η *συνάρτηση ερμηνείας* (interpretation function) [[.]] των RQL τύπων ορίζεται ακολούθως ως:

Ορισμός 17: Δοθείσας μίας συνάρτησης απόδοσης πληθυσμού π στα ονόματα (μετα)σχήματος, η **συνάρτηση ερμηνείας [[.]]** ορίζεται ως εξής:

- Για τους ατομικούς τύπους:: $[[\ \tau_L\ ]] = dom(\tau_L)$

- Για τους τύπους πόρων: $[[\ \tau_U\ ]] = u \in U$

- Για τύπους μετακλάσεων: $[[\ \tau_m\ ]] = \{\nu \mid \nu \in \pi_M^*(m)\}$

- Για τύπους κλάσεων: $[[\ \tau_C\ ]] = \{\nu \mid \nu \in \pi_C^*(c)\}$

- Για τύπους ιδιοτήτων: $[[\ \tau_P[\tau_1,\tau_2]\ ]] = \{[v_1, v_2] \mid v_1 \in [[\tau_1]], v_2 \in [[\tau_2]]\} \cup \{[[\tau_{P'}]] \mid p' \prec p\}$

- Για τον τύπο πολυσυνόλου (Bag): $[[\ \{\tau\}\ ]] = \{\{v_1, v_2, \ldots, v_j\} \mid j>0$, για κάθε $i \in [1 \ldots j], v_i \in [[\tau]]\}$

- Για τον τύπο ακολουθίας (Seq): $[[\ [\tau]\ ]] = \{[1{:}v_1, 2{:}v_2, \ldots, n{:}v_n] \mid n>0$, για κάθε $i \in [1 \ldots n], v_i \in [[\tau_i]]\}$

- Για τον τύπο ένωσης (Alt): $[[\ (1{:}\tau_1 + 2{:}\tau_2 + \ldots + n{:}\tau_n)\ ]] = \{i{:}v_i \mid$ για κάθε $i \in [1 \ldots n], v_i \in [[\tau_i]]\}$ □

Προκειμένου να εξασφαλίσουμε μία *συνολοθεωρητική* ερμηνεία των RQL τύπων, ο περιορισμός 16.1 για την απόδοση πληθυσμού σε (μετα)κλάσεις εφαρμόζεται επίσης στην ερμηνεία των υπο-ιδιοτήτων. Στο εξής, χρησιμοποιούμε τους όρους *έκταση κλάσης* και *ιδιότητας* για να συμβολίσουμε τις αντίστοιχες ερμηνείες τους, ενώ θα χρησιμοποιούμε το συμβολισμό **^[[.]]** για να αναφερθούμε μόνο τα άμεσα στιγμιότυπα μίας (μετα)κλάσης (ή ιδιότητας).

Συνολικά, εκμεταλλευόμενοι τους RQL τύπους *T* και τιμές *V* μπορούμε να παρουσιάσουμε με τυπικό τρόπο τις έννοιες του *σχήματος περιγραφών* και της *βάσης περιγραφών*.

Ορισμός 18 Ένα **σχήμα περιγραφών S** είναι μία δυάδα $S=(RS, \sigma)$, όπου $RS=(V_S, E_S, \psi, \lambda, \prec, N)$ είναι ένας έγκυρος RDF/S γράφος σχήματος και σ είναι μία συνάρτηση απόδοσης τύπων $\sigma{:}N \rightarrow T$ □

Η συνάρτηση απόδοσης τύπων *σ*, η οποία είναι *καθολική* στο σύνολο N, συσχετίζει ονόματα (μετα)κλάσεων με τύπους (μετα)κλάσεων και ονόματα ιδιοτήτων

με τύπους ιδιοτήτων. Επιπλέον, συσχετίζει τους βασικούς XML Schema τύπους ή τους τύπους RDF/S συλλογών με τους RQL βασικούς τύπους ή τύπους συλλογών.

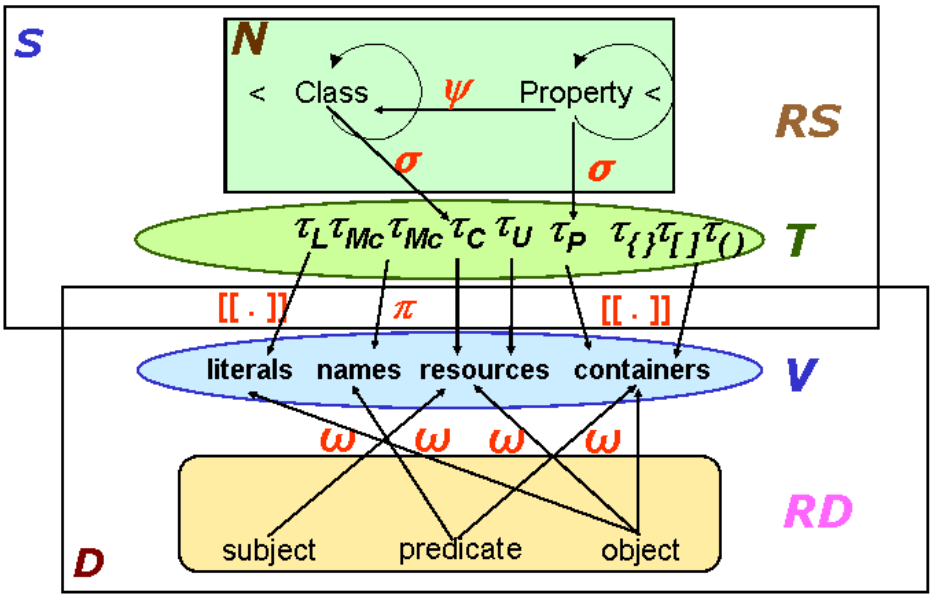
Ορισμός 19: Μία **βάση περιγραφών D**, στιγμιότυπο ενός σχήματος περιγραφών S=(RS, σ), είναι μία δυάδα D=(RD, ω), όπου RD=(RS, V_D, E_D, ψ, λ) είναι ένα σύνολο έγκυρων RDF περιγραφών πόρων και ω είναι μία συνάρτηση απόδοσης τιμών $\omega: V_D \cup E_D \rightarrow V$, τέτοια ώστε

- για κάθε $n \in V_D$, $\omega(n) \in [[\sigma(\lambda(n))]]$ *(19.1)*

- για κάθε $p \in E_D$ από τον κόμβο n στον κόμβο n΄, $[\omega(n), \omega(n')] \in [[\sigma(p)]]$ *(19.2)* □

Η συνάρτηση απόδοσης τιμών *ω* συσχετίζει τους κόμβους και τις ακμές RDF περιγραφών πόρων με μία από τις τιμές στο *V*. Ο περιορισμός 19.1 (19.2) επιβάλλει ότι η τιμή ενός κόμβου (ιδιότητας) ανήκει στην ερμηνεία του τύπου που προσαρτάται στην ετικέτα αυτού του κόμβου (ακμής). Επίσης, οι ατομικοί κόμβοι που αποτιμώνται σε βασικές τιμές ανήκουν στην ερμηνεία βασικών τύπων όπως αλφαριθμητικά, ακέραιοι, ημερομηνίες κτλ.

Στην **Εικόνα 4.3** συνοψίζουμε με γραφικό τρόπο τους τυπικούς ορισμούς που εισάγαμε σε αυτή την ενότητα. Ένας *RDF/S γράφος σχήματος RS* αποτελείται



Εικόνα 4.3: Το τυπικό RQL μοντέλο δεδομένων

από ένα σύνολο ονομάτων *N*, που συνδέονται με σχέσεις υπαλληλίας (≺) και ακμές ιδιοτήτων (*ψ*). Μία *RDF περιγραφή πόρων RD* είναι επίσης ένας γράφος, που απαρτίζεται από ένα σύνολο αναγνωριστικών πόρων URI και ατομικές τιμές (ή τιμές

συλλογών), συνδεδεμένων με ακμές ιδιοτήτων. Και οι δύο γράφοι μπορούν να αναπαρασταθούν χρησιμοποιώντας τριάδες της μορφής <*υποκείμενο, κατηγόρημα, αντικείμενο*>. Κατόπιν, όταν τα *RS* και *RD* ικανοποιούν κατάλληλους περιορισμούς εγκυρότητας, είμαστε σε θέση να απεικονίσουμε τα ονόματα σχήματος *N* σε ένα πεπερασμένο σύνολο RQL τύπων *T* (συνάρτηση *σ*) και τις τριάδες περιγραφών σε ένα πεπερασμένο σύνολο RQL τιμών *V* (συνάρτηση *ω*). Τα *RS* και *T* απαρτίζουν ένα *σχήμα περιγραφών S* και τα *RD* και *V* αποτελούν μία *βάση περιγραφών D*, τα οποία συνδέονται μέσω μίας συνάρτησης ερμηνείας τύπων [[.]], που συσχετίζει RQL τύπους με τιμές.

4.1.4 Διαφορές του RQL μοντέλου δεδομένων με τις RDF/S προδιαγραφές

Σκοπός της παραγράφου αυτής είναι να παρουσιάσει εν είδει σύνοψης όλους τους περιορισμούς που θέτει για λόγους εγκυρότητας το τυπικό RQL μοντέλο δεδομένων σε αντιπαράθεση με τις RDF/S προδιαγραφές. Πιο συγκεκριμένα, στα RDF/S [LS99, BG00] και RDF Semantics [H02], η διάκριση των επιπέδων αφαίρεσης (δεδομένα, σχήμα, μετασχήμα) δεν δηλώνεται σαφώς, σε αντίθεση με το RQL μοντέλο δεδομένων, που βασίζεται στην ύπαρξή τους. Αν και αυτές οι προδιαγραφές δεν υποθέτουν την ύπαρξη επιπέδων αφαίρεσης, δεν την απαγορεύουν. Έτσι, μία κλάση μπορεί να έχει στιγμιότυπα από διαφορετικά σημασιολογικά επίπεδα, όπως άλλες κλάσεις, ιδιότητες ή πόρους δεδομένων. Σε αντίθεση με γνωστές γλώσσες αναπαράστασης γνώσης, όπως η UML [UML] ή η Telos [MBJ+90], όλη η RDF/S πληροφορία (αντικείμενα, κλάσεις και μετακλάσεις) αναπαρίσταται ενιαία με τη μορφή ενός γράφου. Από την άλλη πλευρά, στις γλώσσες DAML+OIL [CHH+01] και OWL [DCH02] το μετασχήμα είναι σταθερό και δεν επιτρέπεται η επέκτασή του με ορισμένες από το χρήστη μετακλάσεις (παρόλο που οι βασικές RDF/S μετακλάσεις *rdfs:Class* και *rdf:Property* εξειδικεύονται για να ορίσουν το DAML+OIL μεταμοντέλο). Όπως τεκμηριώνεται στο [MAC+02], η ευελιξία αυτή ενδέχεται να προκαλέσει προβλήματα σημασιολογικής συνέπειας στην ανάπτυξη εφαρμογών. Το σύστημα τύπων της RQL διακρίνει με σαφήνεια τα διαφορετικά RDF/S επίπεδα αφαίρεσης και προσφέρει κατάλληλες συναρτήσεις ερμηνείας για τη μετάβαση από το ένα επίπεδο στο άλλο. Αποτέλεσμα της θεώρησης αυτής είναι οι ακόλουθοι περιορισμοί, οι οποίοι δεν ορίζονται στα [LS99, BG00, H02]: (α) μία κλάση πρέπει να είναι στιγμιότυπο μίας μετακλάσης κλάσεων και όχι μίας άλλης κλάσης σε

επίπεδο σχήματος, (β) δεν επιτρέπονται σχέσεις υπαλληλίας μεταξύ μετακλάσεων και κλάσεων και (γ) μία μετακλάση δεν μπορεί να είναι στιγμιότυπο κάποιου άλλου κόμβου, εφόσον δεν ορίζονται επίπεδα αφαίρεσης υψηλότερα από το μετασχήμα.

Μία άλλη βασική έννοια του RDF/S, η οποία συνιστά μία σημαντική διαφορά του με το οντοκεντρικό μοντέλο δεδομένων, είναι η θεώρηση των ιδιοτήτων ως «πολίτες πρώτης κατηγορίας», που σημαίνει ότι η διαχείρισή τους γίνεται ανεξάρτητα από τις κλάσεις ή μετακλάσεις στις οποίες αποδίδονται. Εντούτοις, πρακτικά οι ιδιότητες δεν χρησιμοποιούνται ως πολίτες πρώτης κατηγορίας. Στο RDF/S, τόσο μία μετακλάση κλάσεων όσο και μία μετακλάση ιδιοτήτων είναι μία κλάση, σε αντίθεση με την Telos [MBJ+90], όπου μία μετακλάση κλάσεων είναι μία κλάση, αλλά μία μετακλάση ιδιοτήτων (μετα-ιδιότητα) είναι μία ιδιότητα. Για αυτό το λόγο, ενώ στην Telos μια μετα-ιδιότητα μπορεί να έχει πεδίο ορισμού και τιμών, στο RDF/S μοντέλο δεδομένων δεν μπορεί. Επιπλέον, στο επίπεδο δεδομένων μία ιδιότητα δεν μπορεί να είναι *τύπου ιδιότητας*, όπως στην περίπτωση των πόρων και των κλάσεων. Ο περιορισμός αυτός αποδίδεται στο ότι η ιδιότητα *rdf:type*, η οποία αποτελεί τη βάση της υποστασιοποίησης, είναι εφαρμόσιμη μόνο σε κλάσεις και το RDF/S δεν υποστηρίζει ένα μηχανισμό στιγμιότυπων για τις ιδιότητες σε επίπεδο δεδομένων. Στο υιοθετούμενο μοντέλο δεδομένων, χειριζόμαστε τις (μετα)ιδιότητες σύμφωνα με τον τρόπο που ορίζουν οι προδιαγραφές του RDF/S.

Όσον αφορά τον ορισμό και τις ιεραρχίες ιδιοτήτων, ένας πρώτος περιορισμός που τίθεται δηλώνει ότι *τα πεδία ορισμού και τιμών μίας ιδιότητας πρέπει να ορίζονται πάντα και να είναι μοναδικά* [ACK+01]. Ο περιορισμός αυτός έρχεται σε αντίθεση με το RDF Semantics [H02], που επιτρέπει την (προαιρετική) δήλωση πολλαπλών πεδίων ορισμού και τιμών για μία ιδιότητα, στα οποία αποδίδει συζευκτική σημασιολογία. Στην περίπτωση αυτή, μία ιδιότητα με μη καθορισμένο πεδίο ορισμού/τιμών μπορεί να λάβει ως τιμές τόσο πόρους όσο και βασικές τιμές. Παραδείγματος χάριν, στο παράδειγμα της Εικόνας 4.1, εάν η ιδιότητα *technique* έχει ακαθόριστο πεδίο τιμών, το αντικείμενο *oil-on-canvas* μπορεί να ερμηνευτεί τόσο ως ένα αλφαριθμητικό όσο και ως ένα αναγνωριστικό πόρου. Εφόσον η τομή των *rdfs:Class* και *rdfs:Literal* είναι κενή (δηλαδή τα σύνολα αναγνωριστικών πόρων $\mathcal{U}$ και βασικών τιμών $\mathcal{L}$), η ελευθερία αυτή ενδέχεται να οδηγήσει σε σημασιολογικές ασυνέπειες: οι πόροι πρέπει να αναγνωρίζονται μοναδικά από το URI τους, ενώ οι βασικές τιμές από την τιμή τους. Διαφορετικά, αλλάζοντας την βασική τιμή *oil-on-canvas* σε *aquarelle*, θα σήμαινε ότι όλες οι ιδιότητες με τιμή *oil-on-canvas* θα έπρεπε να ενημερωθούν με την τιμή *aquarelle* ως νέα τιμή.

Επιπλέον, αντίθετα με το RQL μοντέλο δεδομένων, το RDF Semantics [H02] εισάγει κανόνες για να συνάγει ότι ο κόμβος αφετηρία μίας ιδιότητας είναι τύπου *πεδίου ορισμού* και ότι ο κόμβος στόχος είναι τύπου *πεδίου τιμών*. Για παράδειγμα, εάν ένας τίτλος (*title*) αποδοθεί στον πόρο *&r2*, τότε ο πόρος αυτός θα ταξινομηθεί αυτόματα κάτω από όλες τις κλάσεις που δηλώνονται ως πεδίο ορισμού της ιδιότητας *title* (δηλαδή αποδίδεται μία σημασιολογία τομής). Εντούτοις, η ταξινόμηση του πόρου *&r2* ως *Painting* και/ή *ExtResource* θα πρέπει να είναι στην πλήρη δικαιοδοσία του μηχανικού γνώσης. Επιπρόσθετα, στις RDF/S προδιαγραφές οι εξειδικευμένες ιδιότητες δεν διατηρούν τη σημασιολογία εγκλεισμού συνόλων του πεδίου ορισμού και τιμών τους. Για παράδειγμα, η υπο-ιδιότητα *sculpts* της ιδιότητας *creates* μπορεί να έχει ως πεδίο ορισμού (τιμών) μία υπερκλάση της κλάσης *Artist* (*Artifact*). Αντιθέτως, στο RQL μοντέλο δεδομένων τα πεδία ορισμού και τιμών των υπο-ιδιοτήτων είναι υπάλληλοι όροι των αντίστοιχων πεδίων των υπερ-ιδιοτήτων τους. Η σημασιολογία των ιδιοτήτων καθίσταται τελείως ασαφής, όταν οι υπο-ιδιότητες ορίζονται με πολλαπλά πεδία ορισμού και τιμών. Τέτοια ευελιξία αναπαράστασης, συνδυαζόμενη με τους προηγούμενους συμπερασματικούς κανόνες (αποκαλούμενους *παραγωγική ερμηνεία* (generative interpretation) στο [H02]), επιβάλλει σοβαρούς περιορισμούς μοντελοποίησης και επεκτασιμότητας για εφαρμογές Σημασιολογικού Ιστού, ειδικότερα όταν χρησιμοποιούνται πολλαπλοί χώροι ονοματοδοσίας αλληλοσυνδεόμενων σχημάτων.

Τέλος, σε αντίθεση με τη νέα έκδοση της RDFS προδιαγραφής [BG02] και του RDF Semantics [H02], το τυπικό RQL μοντέλο δεδομένων απαγορεύει την ύπαρξη κύκλων στις ιεραρχίες υπαλληλίας. Η ύπαρξη κύκλων υποστηρίζεται επίσης από την DAML+OIL [CHH+01] και την OWL [DCH02]. Σύμφωνα με αυτές τις προδιαγραφές, εάν ένας πόρος δηλώνεται ως στιγμιότυπο μίας κλάσης σε ένα κύκλο υπαλληλίας, τότε θα είναι επίσης στιγμιότυπο όλων των άλλων κλάσεων του κύκλου. Με άλλα λόγια, όλες οι κλάσεις που συμμετέχουν στον κύκλο θα έχουν την ίδια έκταση. Με αυτόν τον τρόπο, οι κύκλοι σε μία ιεραρχία υπαλληλίας χρησιμοποιούνται κυρίως για να προσφέρουν διαφορετικά ονόματα για την ίδια (μετα)κλάση ή ιδιότητα. Το σκεπτικό πίσω από αυτή την επιλογή μοντελοποίησης είναι η λογική εξαγωγή της ισοδυναμίας κλάσεων. Εντούτοις, η εισαγωγή κύκλων μπορεί να επηρεάσει σημαντικά τη σημασιολογία ήδη δημιουργημένων RDF/S σχημάτων και περιγραφών, ιδιαίτερα όταν οι δηλώσεις υπαλληλίας διασπείρονται σε πολλούς, διαφορετικούς χώρους ονοματοδοσίας. Για το λόγο αυτό, είναι προτιμότερο οι κύκλοι να χρησιμοποιούνται ως περιορισμός της σημασιολογικής

εγκυρότητας της ένωσης σχημάτων, παρά ως κανόνας εξαγωγής συμπερασμάτων σχετικά με την ισοδυναμία κλάσεων/ιδιοτήτων. Τη λογική αυτή ακουλουθεί το τυπικό RQL μοντέλο δεδομένων. Άλλωστε, οι γλώσσες DAML+OIL και OWL υποστηρίζουν άμεσους μηχανισμούς για τη δήλωση ισοδύναμων κλάσεων/ιδιοτήτων χωρίς να απαιτείται η δημιουργία κύκλων στην ιεραρχία, π.χ με τη χρήση των βασικών DAML+OIL ιδιοτήτων *equivalentTo, samePropertyAs* και *sameClassAs*. Το RQL σύστημα τύπων μπορεί εύκολα να αναπαραστήσει τέτοιους τύπους ιδιοτήτων που έχουν ως πεδίο ορισμού και τιμών μετακλάσεις.

4.1.5 Τυπική ερμηνεία των βασικών RQL επερωτήσεων

Το μοντέλο δεδομένων και το σύστημα τύπων που παρουσιάστηκαν σε προηγούμενες παραγράφους, επιτρέπουν τον ορισμό μίας καλώς ορισμένης σημασιολογίας μίας δηλωτικής γλώσσας επερώτησης που περιλαμβάνει αναδρομή και συναρτησιακή σύνθεση. Όπως εξετάσαμε στο **Κεφάλαιο 2**, η RQL [KAC+02, KC02] αποτελεί μία δηλωτική γλώσσα επερώτησης RDF/S γράφων, που έχει την ικανότητα να συνδυάζει αρμονικά επερώτηση σε επίπεδο (μετα)σχήματος και δεδομένων. Η RQL ορίζει ένα σύνολο βασικών επερωτήσεων και τελεστών σάρωσης που μπορούν να συνδυαστούν με συναρτησιακό τρόπο για να δημιουργήσουν νέες επερωτήσεις.

Έκφραση	Ερμηνεία
subClassOf(c)	$\{c' \in C \mid c' \prec c\}$
subClassOf^(c)	$\{c' \in C \mid c' \prec c, \nexists c'' \in C, c'' \prec c$ και $c' \prec c''\}$
superClassOf(c)	$\{c' \in C \mid c \prec c'\}$
superClassOf^(c)	$\{c' \in C \mid c \prec c', \nexists c'' \in C, c \prec c''$ και $c'' \prec c'\}$
SubPropertyOf(p)	$\{p' \in P \mid p' \prec p\}$
SubPropertyOf^(p)	$\{p' \in P \mid p' \prec p, \nexists p'' \in P, p'' \prec p$ και $p' \prec p''\}$
superPropertyOf(p)	$\{p' \in P \mid p \prec p'\}$
superPropertyOf^(p)	$\{p' \in P \mid p \prec p', \nexists p'' \in P, p \prec p''$ και $p'' \prec p'\}$
P	$[[\ \tau_P\ ]]$
^p	$[[\ \tau_p\ ]] - \cup_{p' \prec p} ([[\ \tau_{p'}\ ]])$
col_1 intersect col_2	$[[\ \tau_{col1}\ ]] \cap [[\ \tau_{col2}\ ]]$
col_1 union col_2	$[[\ \tau_{col1}\ ]] \cup [[\ \tau_{col2}\ ]]$
col_1 minus col_2	$[[\ \tau_{col1}\ ]] \setminus [[\ \tau_{col2}\ ]]$
count(col)	n, if $\exists i_1, \ldots, i_n, \exists v_{i1}, \ldots, v_{in}, \forall j, i_1 \leq j \leq i_n, (j, v_j) \in [[\ \tau_{col}\ ]]$
seq [n]	v_n, if $1 \leq n \leq l, (n, v_n) \in [[\ \tau_{seq}\ ]]$

Πίνακας 4.1: Τυπική ερμηνεία των βασικών RQL ερωτήσεων

Ο **Πίνακας 4.1** παρουσιάζει την τυπική ερμηνεία των βασικότερων RQL επερωτήσεων. Η δυνατότητα σύνθεσης βασικών εκφράσεων επερώτησης σε πιο πολύπλοκες επερωτήσεις εξασφαλίζεται με τη χρήση του συστήματος τύπων. Ειδικότερα, για να είναι έγκυρη μία έκφραση επερώτησης, οι τελεστέοι της έκφρασης πρέπει να είναι ενός συγκεκριμένου τύπου, ο οποίος ελέγχεται κατά τη διάρκεια της μεταγλώττισης. Για παράδειγμα, η έκφραση «*domain(X)*», που επιστρέφει το πεδίο ορισμού της ιδιότητας X, μπορεί να εφαρμοστεί σε τελεστέους τύπου ιδιότητας (τ_P), διαφορετικά επιστρέφεται ένα σφάλμα τύπου. Συνολικά, κάθε χρησιμοποιούμενη ετικέτα συνοδεύεται από τον τύπο της, ο οποίος εξασφαλίζει τη συναρτησιακή σύνθεση των εκφράσεων επερώτησης.

	Έκφραση Μονοπατιού	Ερμηνεία
Μονοπάτι Δεδομένων	c{X}	$\{ v \mid v \in [[\tau_C]] \}$
	{X}p{Y}	$\{ [v_1, v_2] \mid [v_1, v_2] \in [[\tau_P[\tau_1,\tau_2]]] \}$
	{X}@P{Y}	$\{ [v_1, p, v_2] \mid p \in [[\tau_{Mp}]], [v_1, v_2] \in ^{\wedge}[[\tau_P[\tau_1,\tau_2]]] \}$
	$X{Y}	$\{ [c, v] \mid c \in [[\tau_{Mc}]], v \in ^{\wedge}[[\tau_C]] \}$
Μονοπάτι Σχήματος	Class{X}	$\{ c \mid c \in [[\tau_{Mc}]] \}$
	$X	$\{ c \mid c \in [[\tau_{Mc}]] \}$
	Property{P}	$\{ p \mid p \in [[\tau_{Mp}]] \}$
	@P	$\{ p \mid p \in [[\tau_{Mp}]] \}$
	c{$C}	$\{ c' \mid c' \in [[\tau_{Mc}]], c' \preccurlyeq c \}$
	{$X}p{$Y}	$\{ [c_1, c_2] \mid c_1,c_2 \in [[\tau_{Mc}]], c_1 \preccurlyeq domain(p), c_2 \preccurlyeq range(p) \}$
	$X{$Y}	$\{ [c_1, c_2] \mid c_1,c_2 \in [[\tau_{Mc}]], c_2 \preccurlyeq c_1 \}$
	{$X}@P{$Y}	$\{ [c_1, p, c_2] \mid p \in [[\tau_{Mp}]], c_1,c_2 \in [[\tau_{Mc}]], c_1 \preccurlyeq domain(p), c_2 \preccurlyeq range(p) \}$
Μικτό Μονοπάτι	C{X;$C}	$\{ [v, c'] \mid c' \in [[\tau_{Mc}]], c' \preccurlyeq c, v \in ^{\wedge}[[\tau_{C'}]] \}$
	{X;$Z}p{Y;$W}	$\{ [v_1, c_1, v_2, c_2] \mid c_1,c_2 \in [[\tau_{Mc}]], c_1 \preccurlyeq domain(p), v_1 \in ^{\wedge}[[\tau_{C_1}]], c_2 \preccurlyeq range(p), v_2 \in ^{\wedge}[[\tau_{C_2}]], [v_1, v_2] \in [[\tau_P[\tau_1,\tau_2]]] \}$
	P{Y;$W}	$\{ [v_2, c_2] \mid c_2 \in [[\tau_{Mc}]], c_2 \preccurlyeq range(p), v_2 \in ^{\wedge}[[\tau_{C_2}]], [v_1,v_2] \in [[\tau_P[\tau_1,\tau_2]]] \}$
	{X}p{Y;$W}	$\{ [v_1,v_2,c_2] \mid c_2 \in [[\tau_{Mc}]], c_2 \preccurlyeq range(p), v_2 \in ^{\wedge}[[\tau_{C_2}]], [v_1,v_2] \in [[\tau_P[\tau_1,\tau_2]]] \}$
	{X}p{$W}	$\{ [v_1, c_2] \mid c_2 \in [[\tau_{Mc}]], c_2 \preccurlyeq range(p), v_2 \in ^{\wedge}[[\tau_{C_2}]], [v_1, v_2] \in [[\tau_P[\tau_1,\tau_2]]] \}$
	{$Z}p{Y;$W}	$\{ [c_1, v_2, c_2] \mid c_1,c_2 \in [[\tau_{Mc}]], c_1 \preccurlyeq domain(p), v_1 \in ^{\wedge}[[\tau_{C_1}]], c_2 \preccurlyeq range(p), v_2 \in ^{\wedge}[[\tau_{C_2}]], [v_1, v_2] \in [[\tau_P[\tau_1,\tau_2]]] \}$
	{X;$Z}@P{Y;$W}	$\{ [v_1, c_1, p, v_2, c_2] \mid p \in [[\tau_{Mp}]], c_1,c_2 \in [[\tau_{Mc}]], c_1 \preccurlyeq domain(p), v_1 \in ^{\wedge}[[\tau_{C_1}]], c_2 \preccurlyeq range(p), v_2 \in ^{\wedge}[[\tau_{C_2}]], [v_1,v_2] \in ^{\wedge}[[\tau_P[\tau_1,\tau_2]]] \}$

Πίνακας 4.2: Τυπική ερμηνεία των βασικών εκφράσεων μονοπατιών της RQL

Ένα από τα πλεονεκτήματα της RQL γλώσσας είναι η υποστήριξη γενικευμένων εκφράσεων μονοπατιών με μεταβλητές σε ετικέτες κόμβων (κλάσεων) και ακμών

(ιδιοτήτων). Για τον σχηματισμό RQL εκφράσεων μονοπατιών χρησιμοποιούνται δομές της μορφής SELECT-FROM-WHERE. Τα φίλτρα αυτά προσφέρουν ένα αποτελεσματικό εργαλείο για την προσπέλαση συλλογών με RDF δεδομένα ή πληροφορία σχήματος οποιουδήποτε τύπου. Το αποτέλεσμα ενός RQLφίλτρου αναπαρίσταται με μια τιμή RDF πολυσυνόλου (bag), ενώ οι διατεταγμένες πλειάδες μπορούν να αναπαρασταθούν ως RDF ακολουθίες, οι οποίες προσπελάζονται μέσω δεικτών θέσης (βασική RQL επερώτηση «seq[n]», Πίνακας 4.1).

Πιο συγκεκριμένα, όπως και στην SQL, ο όρος SELECT ορίζει μία προβολή πάνω στις μεταβλητές των οποίων οι τιμές θα συμμετάσχουν στο αποτέλεσμα. Επιπλέον, μπορούμε να χρησιμοποιήσουμε τον όρο “SELECT *” για να συμπεριλάβουμε στο αποτέλεσμα τις τιμές όλων των μεταβλητών. Ο όρος SELECT κατασκευάζει μία διατεταγμένη πλειάδα, της οποίας η πληθικότητα εξαρτάται από τον αριθμό των μεταβλητών που προβάλλονται. Ο όρος FROM φιλοξενεί τις οριζόμενες εκφράσεις μονοπατιών, οι οποίες πρακτικά καθορίζουν τα τμήματα του RDF/S γράφου που θα λάβουν μέρος στην αποτίμηση της επερώτησης. Ουσιαστικά, μία έκφραση μονοπατιού αποτελείται από μία σειρά βημάτων. Κάθε βήμα αναπαριστά κίνηση προς μία συγκεκριμένη κατεύθυνση προσδιορίζοντας ετικέτες κόμβων, ενώ σε κάθε βήμα μπορούν να εφαρμοστούν ένα ή περισσότερα κατηγορήματα για τον αποκλεισμό των κόμβων που δεν ικανοποιούν μία συγκεκριμένη συνθήκη. Αυτές οι συνθήκες φιλτραρίσματος ορίζονται στον (προαιρετικό) όρο WHERE. Το αποτέλεσμα κάθε βήματος είναι μία λίστα κόμβων που εξυπηρετούν ως σημείο εκκίνησης για το επόμενο βήμα. Ο **Πίνακας 4.2** παρουσιάζει την τυπική ερμηνεία των βασικότερων εκφράσεων μονοπατιών της RQL, όπου γίνεται διάκριση σε μονοπάτια δεδομένων, σχήματος και μικτά μονοπάτια. Επίσης, μπορεί να χρησιμοποιηθεί ο (προαιρετικός) όρος USING NAMESPACE για τη δήλωση μεταβλητών που αντιστοιχούν σε προθέματα χώρων ονοματοδοσίας. Τα προθέματα αυτά μπορούν στη συνέχεια να χρησιμοποιηθούν σε μία επερώτηση για να δηλωθεί με ακρίβεια η προέλευση ενός ή περισσότερων ονομάτων και να αποσαφηνιστούν περιπτώσεις, όπου το ίδιο όνομα ορίζεται σε πολλούς διαφορετικούς χώρους ονοματοδοσίας.

Για παράδειγμα, η RQL επερώτηση:

```
SELECT X
FROM subClassOf(ns1:Artist){X}
WHERE X like "*P*"
USING NAMESPACE ns1=&http://www.art.com/culture.rdf#
```

αναζητά όλες τις υποκλάσεις της κλάσης Artist, όπως αυτή ορίζεται στο χώρο ονοματοδοσίας με πρόθεμα ns1 και URI http://www.art.com/culture.rdf#. Ο περιορισμός που τίθεται στον όρο WHERE δηλώνει ότι το όνομα των υποκλάσεων πρέπει να περιέχει το γράμμα P (π.χ Painter). Περισσότερα χαρακτηριστικά παραδείγματα της εκφραστικής δύναμης της RQL μπορεί να αντλήσει ο ενδιαφερόμενος αναγνώστης από το [KC02]. Την εκφραστικότητα αυτής της γλώσσας χρησιμοποιούμε για την κατασκευή RDF/S όψεων, όπως αυτές ορίζονται με τυπικό τρόπο στην επόμενη ενότητα.

4.2 Το τυπικό μοντέλο των όψεων

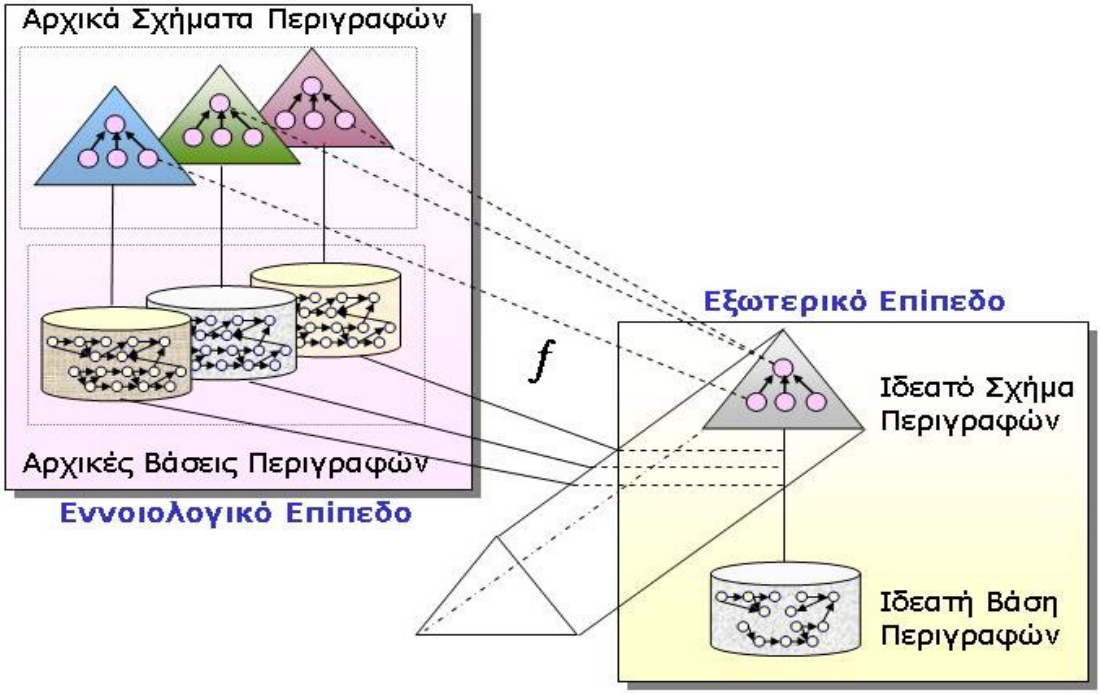
Σκοπός της συγκεκριμένης ενότητας είναι να επεκτείνει το τυπικό RQL μοντέλο δεδομένων, παρουσιάζοντας το μοντέλο δεδομένων των όψεων που υποστηρίζει ο προτεινόμενος μηχανισμός ορισμού RDF/S όψεων. Αν και σε ένα σχεσιακό περιβάλλον μία όψη μπορεί να θεωρηθεί ως μία επερώτηση στη βάση δεδομένων, στα πλαίσια ενός σημασιολογικού δικτύου, όπως είναι οι RDF/S γράφοι, μία όψη δεν περιορίζεται σε μία επερώτηση που επιστρέφει μέρος του σημασιολογικού δικτύου. Ένας από τους βασικούς στόχους του μηχανισμού όψεων στα πλαίσια του RDF/S είναι, εκτός από την ανάκτηση ενός (τμήματος) γράφου, η αναδιοργάνωση των ιεραρχιών κλάσεων και ιδιοτήτων και η θέαση των ίδιων δεδομένων από πολλές διαφορετικές «οπτικές γωνίες» με τη δημιουργία ιδεατών σχημάτων.

4.2.1 Ορισμός RDF/S όψεων

Μία από τις πρώτες σχεδιαστικές αποφάσεις που πρέπει να ληφθεί για την προδιαγραφή ενός μηχανισμού ορισμού όψεων είναι το *επίπεδο λεπτομέρειας* στο οποίο θα γίνονται οι ορισμοί των όψεων και κατά συνέπεια ο ορισμός του τι σημαίνει όψη στο νέο μοντέλο δεδομένων. Όπως έχουμε ήδη αναφέρει στο οντοκεντρικό μοντέλο δεδομένων, που ομοιάζει σε πολλά σημεία με το RDF/S, μία αντικειμενοστραφής όψη μπορεί να οριστεί σε *επίπεδο όψης* ή *σχήματος όψης*. Ο μηχανισμός ορισμού όψεων που προτείνουμε για το πρότυπο RDF/S υποστηρίζει όψεις σύμφωνα με το πνεύμα της αρχιτεκτονικής ANSI/SPARC [A75]. Στο εξωτερικό επίπεδο αυτής της αρχιτεκτονικής, η βάση δεδομένων γίνεται αντιληπτή ως μία

εξωτερική όψη, που ορίζεται με ένα εξωτερικό σχήμα. Συνεπώς, μία **RDF/S όψη** - κατά τη δική μας προσέγγιση- είναι ένα **ιδεατό σχήμα περιγραφών**[41]. Η έκταση της RDF/S όψης αντιστοιχεί σε μία **ιδεατή** ή **παραγόμενη βάση περιγραφών**, που ορίζεται ως ένα στιγμιότυπο του ιδεατού σχήματος περιγραφών και παράγεται βάσει μίας ή περισσότερων αρχικών βάσεων περιγραφών.

Η **Εικόνα 4.4** παρουσιάζει εποπτικά τη δημιουργία ενός ιδεατού σχήματος περιγραφών. Ένα ιδεατό σχήμα ορίζεται πάνω σε ένα ή περισσότερα σχήματα (βασικά ή ιδεατά), τα οποία ονομάζονται **αρχικά** σχήματα. Προκειμένου να δημιουργήσουμε ένα ομοιογενές μοντέλο, διατηρώντας ταυτόχρονα τα χαρακτηριστικά του υποκείμενου μοντέλου δεδομένων, ελαχιστοποιούμε τις διαφορές ανάμεσα σε ένα ιδεατό και ένα **πραγματικό** ή **βασικό** σχήμα. Με τον τρόπο αυτό, η διάκριση όψεων και βασικών σχημάτων δεν γίνεται ορατή στο χρήστη και οι εφαρμογές που χρησιμοποιούν μία όψη ορίζονται με τον ίδιο τρόπο, όπως πάνω σε ένα πραγματικό σχήμα. Επιπλέον, η *σύνθεση όψεων* γίνεται με φυσικό τρόπο, επιτρέποντας τη δημιουργία όψεων πάνω σε όψεις. Επομένως, ένα



Εικόνα 4.4: Παραγωγή ενός ιδεατού σχήματος περιγραφών βάσει πολλαπλών αρχικών σχημάτων περιγραφών

[41] Για λόγους συντομίας, θα χρησιμοποιούμε εναλλακτικά τον όρο «ιδεατό σχήμα».

ιδεατό σχήμα περιγραφών είναι ένα νέο σχήμα του ίδιου τύπου όπως εκείνα από τα οποία προέρχεται, συν μία αντιστοίχηση που υπολογίζει την κατάσταση/έκταση της όψης από την τρέχουσα ή τις τρέχουσες βάσεις περιγραφών. Αυτή η αντιστοίχηση αποκαλείται *απεικόνιση ορισμού της όψης* (view definition mapping) και αναπαρίσταται στην Εικόνα 4.4 με τη συνάρτηση «f». Ουσιαστικά, η απεικόνιση ορισμού μίας όψης είναι ένα σύνολο RQL εκφράσεων επερώτησης στο αρχικό σύνολο πληροφορίας ((μετα)σχήμα και δεδομένα).

Ο ορισμός μίας όψης καθορίζει ένα σύνολο από **ιδεατά ονόματα** (μετα)κλάσεων και ιδιοτήτων και τις αντίστοιχες ιεραρχίες, σε αντίθεση με τα ονόματα (μετα)κλάσεων και ιδιοτήτων της βάσης περιγραφών, που αποκαλούνται **βασικά**. Τα ονόματα κλάσεων και ιδιοτήτων (βασικά ή ιδεατά) προέρχονται από διαφορετικούς χώρους ονοματοδοσίας. Χρησιμοποιώντας τον μηχανισμό ονοματοδοσίας της XML [BHL99], μία όψη ορίζει ένα νέο χώρο ονοματοδοσίας. Για να παρέχουμε όλη τη δυνατή ευελιξία και διαφάνεια στη χρήση ιδεατών και βασικών (μετα)κλάσεων από την πλευρά του χρήστη, χειριζόμαστε τις ιδεατές κλάσεις και ιδιότητες ως οποιαδήποτε άλλη, βασική κλάση ή ιδιότητα, στις οποίες μάλιστα αποδίδουμε ένα *καθολικό* αναγνωριστικό. Το αναγνωριστικό αυτό συντίθεται από το χώρο ονοματοδοσίας της όψης και το όνομα της κλάσης/ιδιότητας, κατά αναλογία με τις βασικές κλάσεις/ιδιότητες και παίζει το ρόλο ενός URI, το οποίο μπορεί να χρησιμοποιηθεί στον ορισμό άλλων ιδεατών κλάσεων/ιδιοτήτων. Εκτός όμως από τις νέες ιδεατές (μετα)κλάσεις και ιδιότητες που ορίζουμε σε μία όψη, μπορούμε να *επαναχρησιμοποιήσουμε* στον ορισμό μίας όψης ένα σύνολο αρχικών ονομάτων (μετα)κλάσεων και ιδιοτήτων. Στην περίπτωση αυτή, μία επαναχρησιμοποιούμενη κατασκευή προσδιορίζεται πλέον από το χώρο ονοματοδοσίας του ιδεατού σχήματος, αποκτώντας ιδεατή υπόσταση. Αξίζει να σημειώσουμε, ότι η επαναχρησιμοποίηση ενός ονόματος αρχικής κατασκευής δεν αναφέρεται στο σύνολο της συσχετιζόμενης με αυτή πληροφορίας, π.χ υποκλάσεις, υπερκλάσεις, στιγμιότυπα κτλ.

Έχοντας ορίσει ένα σύνολο ιδεατών (μετα)κλάσεων και ιδιοτήτων, μπορούμε να τις διατάξουμε σε *ιεραρχίες (μετα)κλάσεων και ιδιοτήτων* αντίστοιχα. Σύμφωνα με την προσέγγισή μας, σε μία όψη η ιεραρχία (μετα)κλάσεων και ιδιοτήτων ($\prec$) αναφέρεται στις σχέσεις υπαλληλίας ανάμεσα στις ιδεατές (μετα)κλάσεις/ιδιότητες και τις αρχικές (μετα)κλάσεις/ιδιότητες που *επαναχρησιμοποιούνται* στον ορισμό της όψης (που ορίζονται και αυτές ως ιδεατές στην όψη) και όχι στο σύνολο των βασικών κλάσεων και ιδιοτήτων. Σε αντίθεση με τη δημιουργία καθολικών

ιεραρχιών ανάμεσα σε ιδεατές και βασικές κλάσεις, ο περιορισμός αυτός δημιουργεί μία διάκριση ανάμεσα στις ιεραρχίες του αρχικού σχήματος και του ιδεατού σχήματος, προσφέροντας μεγαλύτερη ευελιξία στη δημιουργία ιδεατών σχημάτων (που δεν περιορίζονται πλέον από τα αρχικά σχήματα). Με τον τρόπο αυτό, μπορούμε να αναδομήσουμε τις αρχικές ιεραρχίες στο ιδεατό σχήμα με όποιο τρόπο επιθυμούμε και να επιτρέψουμε την πρόσβαση στα αποθηκευμένα δεδομένα με τρόπο που να εξυπηρετεί τις ανάγκες και προτιμήσεις θέασης των δεδομένων του εκάστοτε χρήστη.

Επιπλέον, η δημιουργία μίας ιδεατής κλάσης σε μία RDF/S όψη δεν εξαρτάται από την έκταση μίας ή περισσότερων αρχικών κλάσεων (σε αντίθεση με το οντοκεντρικό μοντέλο δεδομένων)[42]. Συν τοις άλλοις, μπορούμε να δημιουργήσουμε μία ιδεατή κλάση από ατομικές τιμές ή πόρους στο επίπεδο δεδομένων. Έχοντας ορίσει μία ιδεατή (μετα)κλάση (αντίστοιχα μία ιδεατή ιδιότητα), υπάρχει η δυνατότητα προσαρμοσμένης απόδοσης πληθυσμού από μία ή περισσότερες αρχικές (μετα)κλάσεις, που όχι μόνο δεν ανήκουν στο ίδιο τμήμα της ιεραρχίας, αλλά ορίζονται και σε διαφορετικά σχήματα. Η αποδέσμευση του ορισμού μίας ιδεατής κλάσης/ιδιότητας από την απόδοση πληθυσμού σε αυτή, μας δίνει τη δυνατότητα να ορίζουμε ιδεατά σχήματα περιγραφών, τα οποία από μόνα τους αποτελούν ένα σύνολο γνώσης που μπορεί να επερωτηθεί και να συνδυάζουμε γνώση -όπου απαιτείται από τις εκάστοτε εφαρμογές- από πολλαπλές βάσεις περιγραφών. Παράλληλα, εκτός από αντικείμενα που υπάρχουν αποθηκευμένα στις αρχικές βάσεις, σε μία ιδεατή κλάση/ιδιότητα μπορεί να αποδοθούν *νέα* αντικείμενα, τα οποία αποκαλούμε *ιδεατά στιγμιότυπα* (object-generating views). Η δυνατότητα δημιουργίας ιδεατών στοιχείων μίας όψης συνεπάγεται την ικανότητα δημιουργίας νέων αναγνωριστικών αντικειμένων (URI), τα οποία θα ταυτοποιούν τα ιδεατά στοιχεία.

4.2.2 Ιδεατά RDF σχήματα περιγραφών και βάσεις περιγραφών

Κατά αναλογία με το οντοκεντρικό μοντέλο δεδομένων, ένα ιδεατό σχήμα περιγραφών αποτελεί το προϊόν μίας διαδικασίας δύο βημάτων [S95]. Στο πρώτο

[42] Αυτό αποδίδεται κυρίως σε δύο λόγους: α) τη διάκριση του επιπέδου σχήματος από το επίπεδο δεδομένων και β) το γεγονός ότι οι κλάσεις στο RDF/S περιέχουν αντικείμενα που περιγράφονται με διαφορετικό τρόπο από (πολλαπλά) σχήματα και δεν επιβάλλουν κάποιο τύπο στα μέλη τους.

βήμα, τα αρχικά σχήματα περιγραφών επεκτείνονται με τον ορισμό νέων κλάσεων, ιδιοτήτων ή ολόκληρων ιεραρχιών κλάσεων και ιδιοτήτων, δημιουργώντας ένα **επεκταμένο** σχήμα. Ένα επεκταμένο σχήμα αποτελεί ενδιάμεσο προϊόν της διαδικασίας δημιουργίας ιδεατών σχημάτων και υφίσταται μόνο κατά τη διάρκεια αυτής. Για λόγους απλότητας στους ορισμούς, θεωρούμε ότι τα αρχικά σχήματα στο σύνολο τους συνθέτουν ένα ευρύτερο αρχικό σχήμα, το οποίο απαρτίζεται από το σύνολο των ονομάτων κλάσεων και ιδιοτήτων. Εξαιτίας της ύπαρξης καθολικών αναγνωριστικών και των περιορισμών εγκυρότητας του τυπικού RQL μοντέλου δεδομένων, το ευρύτερο αυτό σχήμα περιγραφών είναι έγκυρο και δεν εμφανίζει ασυνέπειες, όπως απόδοση ίδιου ονόματος σε διαφορετικές οντότητες, εισαγωγή νέου πεδίου ορισμού/τιμών σε ιδιότητες και κύκλους στις ιεραρχίες κλάσεων και ιδιοτήτων. Άλλωστε, σύμφωνα με τον τυπικό **Ορισμό 12**, η ένωση έγκυρων RDF/S γράφων σχήματος (και κατ' επέκταση σχημάτων περιγραφών) είναι ένας έγκυρος γράφος σχήματος (σχήμα περιγραφών)[43]. Στο δεύτερο βήμα, αποκρύπτεται πληροφορία από το επεκταμένο σχήμα για την απόκτηση ενός νέου σχήματος, που αποτελεί το **ιδεατό** σχήμα. Η απόκρυψη πληροφορίας ουσιαστικά συνίσταται στην επιλογή των (μετα)κλάσεων/ιδιοτήτων και των αντίστοιχων ιεραρχιών που θα αποτελέσουν τα δομικά στοιχεία της όψης.

Επομένως, ένα ιδεατό σχήμα περιγραφών ορίζεται ως *η προβολή* μίας *επέκτασης* του αρχικού του σχήματος. Πριν αναφέρουμε τους τυπικούς ορισμούς του ιδεατού σχήματος, παρουσιάζουμε τους τυπικούς ορισμούς της *επέκτασης* και *προβολής* ενός σχήματος περιγραφών.

Ορισμός 20: Ένα σχήμα περιγραφών $S'=(RS', \sigma')$ είναι μία **επέκταση** ενός σχήματος περιγραφών $S=(RS, \sigma)$, εάν και μόνο αν $RS \in RS'$ και $\sigma'_{|S}=\sigma$. Συμβολίζουμε την περίπτωση αυτή ως $S \sqsubseteq S'$ □

Ορισμός 21: Η **προβολή** ενός σχήματος περιγραφών $S'=(RS', \sigma')$ πάνω στο RS είναι ένα σχήμα περιγραφών $S=(RS, \sigma'_{|RS})$, όπου $RS \in RS'$ □

Από τους παραπάνω ορισμούς, καθίσταται φανερό ότι το S΄ είναι μία επέκταση της προβολής του (σχήμα S), που σημαίνει ότι οι δύο πράξεις είναι

[43] Ο Ορισμός 12 σχετικά με την ένωση έγκυρων RDF/S γράφων σχήματος είναι εφαρμόσιμος στην περίπτωση σχημάτων περιγραφών, δεδομένου ότι οι ετικέτες διατηρούν (και συνοδεύονται) πάντα από τον τύπο τους.

αντίστροφη η μία της άλλης. Η ιδιότητα αυτή της προβολής και επέκτασης βρίσκει εφαρμογή στη σύνθεση όψεων, όπου εναλλάσσοντας τις δύο αυτές πράξεις, μπορούμε να δημιουργήσουμε ακολουθίες ορισμών όψεων πάνω σε αρχικά και ιδεατά σχήματα περιγραφών. Ο **Ορισμός 20** δηλώνει ότι στο σχήμα S΄ ορίζονται επιπλέον των αρχικών (μετα)κλάσεις και ιδιότητες, καθώς και οι μεταξύ τους σχέσεις υπαλληλίας. Οι επιπλέον αυτές (μετα)κλάσεις και ιδιότητες είναι ιδεατές και διακρίνονται από ένα μοναδικό URI. Από τον ορισμό των υπογράφων, είναι κατανοητό ότι οι ιεραρχίες (μετα)κλάσεων και ιδιοτήτων και τα πεδία ορισμού/τιμών των ιδιοτήτων του αρχικού σχήματος περιγραφών S, όπως «φαίνονται» μέσα από την επέκταση του, δεν μεταβάλλονται. Αμετάβλητες παραμένουν και οι ετικέτες που αποδίδονται σε (μετα)κλάσεις και ιδιότητες του αρχικού σχήματος, καθώς και οι τύποι τους ($\sigma'_{|S}=\sigma$). Το γεγονός αυτό μας εξασφαλίζει την ανεξαρτησία των όψεων από τα αρχικά σχήματα και βάσεις περιγραφών.

Έχοντας πλέον ορίσει τυπικά τις έννοιες της επέκτασης και προβολής ενός σχήματος περιγραφών, παρουσιάζουμε τους τυπικούς ορισμούς του *επεκταμένου* και του *ιδεατού σχήματος περιγραφών*.

Ορισμός 22: Μία επέκταση S΄ του S είναι ένα **επεκταμένο σχήμα** του S, αν και μόνο αν υπάρχει μία μερική συνάρτηση $\boldsymbol{f}$: $S' \rightarrow 2^S$, η οποία αντιστοιχεί κάποια ιδεατή (μετα)κλάση ή ιδιότητα $n \in N'$ σε ένα σύνολο από αρχικές (μετα)κλάσεις ή ιδιότητες στο N και η οποία έχει την ιδιότητα ότι $\boldsymbol{f}_{|S}$=id (id είναι η συνάρτηση ταυτότητας) □

Ορισμός 23: Η **προβολή** S΄ ενός επεκταμένου σχήματος S΄΄ του S είναι ένα **ιδεατό σχήμα περιγραφών**. Το S είναι το αρχικό σχήμα περιγραφών του S΄. Η τριάδα δ=(S, $\boldsymbol{f}$, S΄) δηλώνει την παραγωγή του S΄ από το S με τη χρήση της συνάρτησης αντιστοιχίας $\boldsymbol{f}$ □

Ο **Ορισμός 22** και ειδικότερα η συνάρτηση αντιστοιχίας *f* ορίζει τον τρόπο με τον οποίο μία όψη παράγεται πάνω σε ένα αρχικό σχήμα. Ουσιαστικά, η συνάρτηση *f* αντιστοιχεί σε μία RQL επερώτηση πάνω στο αρχικό σχήμα περιγραφών. Η συνάρτηση αυτή είναι *μερική* εξαιτίας της δυνατότητας να παράγουμε ιδεατές (μετα)κλάσεις ή ιδιότητες χωρίς τη χρήση γνώσης από το αρχικό σχήμα και την αρχική βάση περιγραφών. Η τριάδα δ στον **Ορισμό 23** μπορεί να γραφτεί ως δ=({S_1,S_2, …, S_n}, *f*, S΄) για να δηλώσει ρητά την παραγωγή του ιδεατού σχήματος S΄ από ένα σύνολο αρχικών σχημάτων περιγραφών. Στην

περίπτωση αυτή, η συνάρτηση αντιστοιχίας f δηλώνεται επίσης ως f:S΄→$2^{S_1 \cup S_2 \cup .. \cup S_n}$, όπου S_i, $1 \leq i \leq n$, το σύνολο των αρχικών σχημάτων. Το παραγόμενο ιδεατό σχήμα, το οποίο αποτελεί την προβολή ενός επεκταμένου σχήματος, είναι ένα *ιδεατό σχήμα περιγραφών* που πληροί όλους τους περιορισμούς του υποκείμενου μοντέλου δεδομένων και δεν περιέχει πλεονασματικές δηλώσεις.

Καταληκτικά, από το σύνολο των παραπάνω ορισμών συμπεραίνεται ότι το σύνολο όλης της πληροφορίας που δημιουργείται μέσω των δηλώσεων ορισμού της όψης σχηματίζει ένα επεκταμένο σχήμα, το οποίο αποτελεί επέκταση των αρχικών σχημάτων με ιδεατές κατασκευές. Η προβολή του επεκταμένου σχήματος, η οποία ουσιαστικά συνιστά το ιδεατό σχήμα, δημιουργείται με την απόκρυψη όλης της πληροφορίας που δεν αναφέρεται στις ιδεατές (μετα)κλάσεις και ιδιότητες ή αποτελεί περιττή πληροφορία στη όψη, εφόσον η ερμηνεία που θεωρούμε είναι συνολοθεωρητική (πολλαπλές ακμές *rdf:type* και σχέσεις υπαλληλίας). Ουσιαστικά, κατά τη προβολή του επεκταμένου σχήματος εξασφαλίζεται ότι η πληροφορία που υπάρχει στην όψη είναι η ελάχιστη δυνατή, που διατηρεί την όψη και την ιδεατή βάση περιγραφών συνεπείς[44]. Με διαδικασία αντίστοιχη με εκείνη των ιδεατών σχημάτων περιγραφών, ορίζονται και οι ιδεατές βάσεις περιγραφών. Αρχικά δίνουμε τους ορισμούς της *επέκτασης* και της *προβολής* μίας βάσης περιγραφών.

Ορισμός 24: Μία βάση περιγραφών D΄=(RD΄, ω΄), στιγμιότυπο ενός σχήματος περιγραφών S΄=(RS΄, σ΄), είναι μία **επέκταση μίας βάσης περιγραφών** D=(RD,ω), στιγμιότυπο ενός RDF

[44] Οι ιεραρχίες που αναπτύσσονται σε ένα σχήμα περιγραφών μπορεί να είναι ανεπτυγμένες στην πλήρους μεταβατικής κλειστότητας μορφή τους, γεγονός που εισάγει ένα σύνολο από πλεονασματικές σχέσεις υπαλληλίας στο επεκταμένο σχήμα και οι οποίες πρέπει να εξαλειφθούν στο ιδεατό σχήμα (όψη). Στα πλαίσια μίας αποτελεσματικής υλοποίησης, αντί για εσωτερικά, τεχνητά αναγνωριστικά (id) που αποδίδονται στις ιδεατές κατασκευές από τα ονόματά τους, μπορούμε να εκμεταλλευτούμε σχήματα κωδικοποίησης ετικετών, τα οποία αποδίδουν εσωτερικές ετικέτες που κωδικοποιούν την ιεραρχία. Μία από τις ιδιότητες ενός σχήματος κωδικοποίησης είναι ότι δεν απαιτείται η αφαίρεση όλων των πλεονασματικών σχέσεων υπαλληλίας για την εφαρμογή της κωδικοποίησης, δεδομένου ότι αυτή αφαιρεί τους πλεονασμούς μέσω των ετικετών που αποδίδει. Με τον τρόπο αυτό, η επερώτηση στις κωδικοποιημένες ετικέτες δεν επιστρέφει τους πλεονασματικούς συνδέσμους υπαλληλίας. Σημαντική εργασία πάνω σε σχήματα κωδικοποίησης που μπορούν να εφαρμοστούν σε RDF σχήματα, γίνεται από το συνάδελφο Σωτήρη Τουρτούνη, στα πλαίσια της μεταπτυχιακής εργασίας του με επόπτη τον καθηγητή κ. Δημήτρη Πλεξουσάκη και επιβλέποντα καθηγητή τον κ. Βασίλη Χριστοφίδη και η οποία πρόκειται να παρουσιαστεί σύντομα.

σχήματος S=(RS, σ) (αντίστροφα, η D είναι μία προβολή του D΄), αν και μόνο αν S⊑S΄, και $\omega'|_D=\omega$ □

Από τον παραπάνω ορισμό, συμπεραίνεται ότι η απόδοση τιμών στους κόμβους και τις ακμές μίας αρχικής βάσης περιγραφών D ενός σχήματος περιγραφών S, δεν μεταβάλλεται στην επέκταση D΄ αυτής της βάσης. Το αντίστοιχο συμβαίνει και στην περίπτωση της προβολής. Έχοντας ορίσει την έννοια της προβολής και της επέκτασης, ορίζουμε μία ιδεατή βάση περιγραφών D΄ ως ένα στιγμιότυπο ενός ιδεατού σχήματος S΄, το οποίο ορίζεται με επέκταση και προβολή.

Ορισμός 25: Η **έκταση μίας ιδεατής βάσης περιγραφών** D΄, παραγόμενης από μία αρχική βάση περιγραφών D και στιγμιότυπο ενός ιδεατού σχήματος περιγραφών S΄=(RS΄, σ΄), ορίζεται ως μία δυάδα D΄=(RD΄, ω΄), όπου RD΄=(RS΄, V_D΄, E_D΄, ψ΄, λ΄) είναι ένα σύνολο έγκυρων RDF περιγραφές πόρων και ω΄ είναι μία συνάρτηση απόδοσης τιμών ω΄:V_D΄∪E_D΄ → V, τέτοια ώστε

- για κάθε n∈V_D΄, ω΄(n)∈[[σ΄(λ΄(n))]]

- για κάθε p∈E_D΄ από τον κόμβο n στον κόμβο n΄, [ω΄(n), ω΄(n΄)]∈[[σ΄(p)]] □

Σε αντίθεση με μία αρχική βάση περιγραφών, τα περιεχόμενα μίας ιδεατής βάσης περιγραφών D΄ υπολογίζονται κατά απαίτηση όταν ενεργοποιείται η όψη, βάσει μίας ή περισσότερων αρχικών βάσεων περιγραφών, οι οποίες αποτελούν στιγμιότυπα των αρχικών σχημάτων περιγραφών της όψης.

4.2.3 Επιλογές μοντελοποίησης του μηχανισμού ορισμού όψεων

Σύμφωνα με τους τυπικούς ορισμούς που παρουσιάστηκαν στην παραπάνω παράγραφο, μία RDF/S όψη είναι ένα ιδεατό σχήμα περιγραφών, που περιέχει ιδεατές (μετα)κλάσεις, ιδιότητες και ιεραρχίες ανάμεσα σε αυτές. Η έκταση της όψης ισοδυναμεί με μία ιδεατή βάση περιγραφών, η οποία παράγεται από μία ή περισσότερες αρχικές βάσεις περιγραφών. Εκτός από τους τυπικούς ορισμούς, θέματα όπως η δημιουργία όψεων που διατηρούν ή δημιουργούν αντικείμενα και η σύνθεση όψεων, χρήζουν αναφοράς για την πληρότητα της παρουσίασης του τυπικού μοντέλου όψεων. Το στόχο αυτό επιτυγχάνει η συγκεκριμένη παράγραφος.

4.2.3.1 Όψεις που διατηρούν ή/και δημιουργούν αντικείμενα

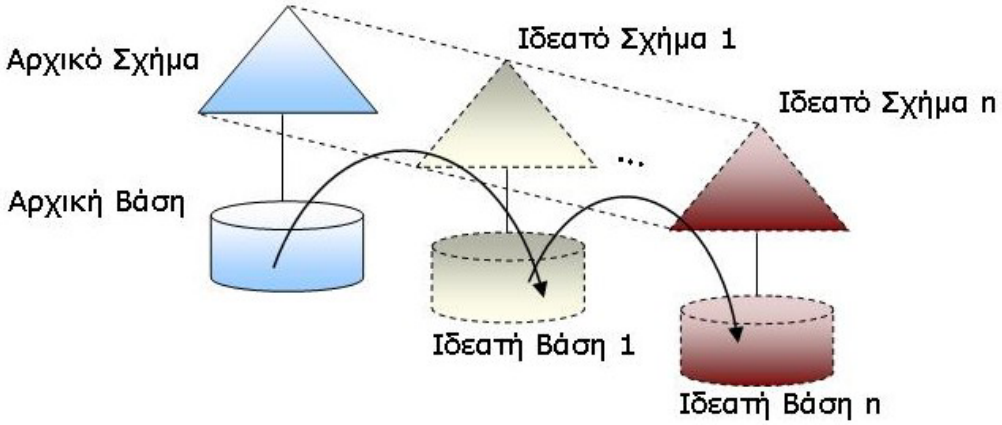
Το μοντέλο δεδομένων στο οποίο βασιζόμαστε ακολουθεί μία οντοκεντρική προσέγγιση προσαρμοσμένη σε ημιδομημένα δεδομένα. Ουσιαστικά, αποτελείται από μοναδικά ονόματα (μετα)κλάσεων και ιδιοτήτων οργανωμένων σε ιεραρχίες εξειδίκευσης. Αποδίδοντας στις ιδεατές κατασκευές αναγνωριστικό URI, επιτρέποντας τη χρησιμοποίησή τους στον ορισμό άλλων όψεων και υποστηρίζοντας τη δημιουργία ιεραρχιών είναι κάποιοι από τους τρόπους με τους οποίους αποδεικνύεται ότι χειριζόμαστε τις ιδεατές κατασκευές όπως τις πραγματικές. Ένας άλλος τρόπος είναι η *απόδοση πληθυσμού* σε αυτές. Η απόδοση πληθυσμού σε κατασκευές ιδεατού σχήματος μπορεί να γίνει με δύο τρόπους: είτε με άμεση ανάθεση ενός στιγμιότυπου στον πληθυσμό της κατασκευής είτε μέσω μίας επερώτησης στο αρχικό σύνολο πληροφορίας (δεδομένα και σχήμα). Η επερώτηση αυτή φιλτράρει τα αρχικά δεδομένα και επιστρέφει το σύνολο των δεδομένων που θα αποτελέσουν τον πληθυσμό της ιδεατής (μετα)κλάσης ή ιδιότητας. Τα δεδομένα που επιστρέφονται και αφορούν το επίπεδο σχήματος (είναι δηλαδή στιγμιότυπα μετακλάσεων), διακρίνονται πλέον από το χώρο ονοματοδοσίας της όψης. Ένας περιορισμός που τίθεται είναι ότι η επερώτηση αυτή δεν μπορεί να αναφέρεται σε δεδομένα της όψης στην οποία χρησιμοποιείται.

Εκτός όμως από τη διαδικασία απόδοσης μελών στις ιδεατές κλάσεις με στιγμιότυπα κλάσεων που έχουν ήδη αποθηκευτεί (**object-preserving views**), μία ιδεατή κλάση μπορεί να αποκτήσει και *ιδεατά στιγμιότυπα* (virtual instances), τα οποία ορίζονται μόνο μέσα στην όψη (**object-generating views**) [AB91, SAD94, YBS97, LDB97a, LDB97b, GBC+97]. Τα ιδεατά στιγμιότυπα υπάρχουν μόνο κατά τη διάρκεια ενεργοποίησης της όψης και δεν αποθηκεύονται, γεγονός που εξασφαλίζει την ανεξαρτησία των αποθηκευμένων δεδομένων και των όψεων (για την ταυτοποίησή τους χρησιμοποιείται συνάρτηση δημιουργίας αναγνωριστικών). Οι όψεις που ορίζονται με αυτό τον τρόπο αυξάνουν την ικανότητα αποθήκευσης δεδομένων στις όψεις. Και στις δύο περιπτώσεις όμως, πριν κατασκευαστεί η ιδεατή βάση, θα πρέπει να ελεγχθεί ότι ένα στιγμιότυπο ανήκει στην πιο ειδική του κλάση και δεν υπάρχουν πλεονασματικές δηλώσεις απόδοσης στιγμιότυπων σε ιδεατές (μετα)κλάσεις και ιδιότητες.

4.2.3.2 Σύνθεση όψεων

Ένα από τα ζητήματα που χρήζουν αναφοράς στα πλαίσια ενός τυπικού μοντέλου όψεων είναι η *σύνθεση όψεων*. Η δυνατότητα σύνθεσης όψεων μας επιτρέπει να δημιουργούμε όψεις πάνω σε βασικά και ιδεατά σχήματα περιγραφών. Μπορούμε να θεωρήσουμε δύο επίπεδα σύνθεσης όψεων: στο *εσωτερικό* ενός ιδεατού σχήματος και *ανάμεσα* σε δύο ή περισσότερα διαφορετικά, ιδεατά σχήματα [S95]. Η σύνθεση στο εσωτερικό ενός ιδεατού σχήματος αντιστοιχεί στον ορισμό της έκτασης μίας ιδεατής κλάσης βάσει της έκτασης μίας αρχικής κλάσης, η οποία ορίζεται ως ιδεατή κλάση στην ίδια όψη. Για λόγους ανεξαρτησίας των δεδομένων και απαγόρευσης κυκλικών δηλώσεων (μία κλάση χρησιμοποιεί μία άλλη κλάση στον ορισμό της και το αντίστροφο), απαγορεύουμε αυτό τον τρόπο σύνθεσης ιδεατών κλάσεων και δεν επιτρέπουμε μία επερώτηση να αναφέρεται στις ιδεατές κατασκευές της οριζόμενης όψης.

Η σύνθεση όψεων σε επίπεδο διαφορετικών, ιδεατών σχημάτων, μας επιτρέπει να δημιουργήσουμε ιδεατές (μετα)κλάσεις ή ιδιότητες, οι οποίες ορίζονται βάσει ιδεατών (μετα)κλάσεων ή ιδιοτήτων άλλων ιδεατών σχημάτων. Οι ιδεατές (μετα)κλάσεις/ιδιότητες που ορίζονται βάσει ιδεατών (μετα)κλάσεων/ιδιοτήτων μπορούν να οριστούν αυξητικά σε διαφορετικά ιδεατά σχήματα, τα οποία παράγονται το ένα από το άλλο. Η διαδικασία αυτή δημιουργεί μία ακολουθία δημιουργίας ιδεατών σχημάτων και κάθε ενδιάμεσο ιδεατό σχήμα μπορεί να θεωρηθεί ως ένα επίπεδο στη διαδικασία παραγωγής του τελικού ιδεατού σχήματος. Η **Εικόνα 4.5** απεικονίζει γραφικά τη διαδικασία σύνθεσης όψεων, η οποία απαρτίζεται από μία ακολουθία ορισμών ιδεατών σχημάτων.



Εικόνα 4.5: Σύνθεση όψεων

Μπορούμε να επιβάλλουμε μία διάταξη πάνω στα παραγόμενα ιδεατά σχήματα μίας ακολουθίας ορισμών ιδεατών σχημάτων. Σε αυτή την περίπτωση, βρίσκει εφαρμογή ο παρακάτω τυπικός ορισμός [S95].

Ορισμός 26: Η **παραγωγή ενός ιδεατού σχήματος** $\delta=(\{S_1, S_2, \ldots, S_n\}, f, S')$ συνεπάγεται μία σχέση διάταξης $\lhd$ ανάμεσα στα αρχικά σχήματα $S_1, S_2, \ldots, S_n$ και στο ιδεατό σχήμα S', τέτοια ώστε για όλα τα αρχικά S_i, $S' \lhd S_i$. Ένα σύνολο παραγωγών ιδεατών σχημάτων $\Delta=\{\delta_1, \delta_2, \ldots \delta_n\}$, είναι καλώς ορισμένο, αν η σχέση διάταξης $\lhd$ διατηρείται στο σύνολο όλων των σχημάτων που συμμετέχουν στις παραγωγές □

Η σχέση διάταξης εξασφαλίζει ότι κατασκευές ενός ιδεατού σχήματος που χρησιμοποιούνται στον ορισμό ενός άλλου ιδεατού σχήματος έχουν ήδη οριστεί. Το ίδιο συμβαίνει και με τις εκτάσεις των ιδεατών βάσεων περιγραφών. Για λόγους συνέπειας, απαγορεύουμε τους κύκλους σε επίπεδο παραγωγής σχημάτων, καθώς αυτό θα σήμαινε ότι πιθανότατα ιδεατές κλάσεις χρησιμοποιούνται στο ορισμό των βασικών σχημάτων, γεγονός που απαγορεύεται για λόγους ανεξαρτησίας των αποθηκευμένων δεδομένων και των ιδεατών δεδομένων.

4.3 Η γλώσσα ορισμού RDF/S όψεων

Έχοντας ορίσει το τυπικό μοντέλο των όψεων, παρουσιάζουμε στο σημείο αυτό την προτεινόμενη *γλώσσα ορισμού δεδομένων όψεων*, η οποία προδιαγράφει το σύνολο των βασικών κατασκευών που απαιτούνται για την υποστήριξη της ζητούμενης από τις εφαρμογές λειτουργικότητας. Η γλώσσα αυτή αποτελεί μία από πρώτες (και ελάχιστες) προσπάθειες προδιαγραφής μίας γλώσσας ορισμού όψεων για το πρότυπο RDF/S. Ενταγμένη σε ένα γενικότερο, ομοιόμορφο τυπικό μοντέλο που διατηρεί τα στοιχεία του υποκείμενου μοντέλου δεδομένων, αποτελεί μία δηλωτική γλώσσα που χαρακτηρίζεται από την απλότητα στον ορισμό των όψεων. Η απλότητα αυτή δεν αποτελεί τροχοπέδη στην ευελιξία της γλώσσας ούτε υποβιβάζει την εκφραστική της δύναμη. Η εκφραστικότητά της διαπιστώνεται τόσο από την εκφραστική δύναμη της υποκείμενης γλώσσας πλοήγησης RDF/S γράφων όσο και από τις παρεχόμενες

λειτουργίες, που στο σύνολό τους αποτελούν ένα χρήσιμο προγραμματιστικό εργαλείο για την μοντελοποίηση του ίδιου συνόλου πληροφορίας σύμφωνα με τις πληροφοριακές ανάγκες διαφορετικών εφαρμογών.

Η λειτουργικότητα που υποστηρίζει η γλώσσα ορισμού όψεων προδιαγράφηκε στο προηγούμενο κεφάλαιο (**Ενότητα 3.2**) και μπορεί να αναχθεί σε ένα σύνολο βασικών κατασκευών για τον ορισμό μίας RDF/S όψης. Συγκεκριμένα, ανάγεται στη δημιουργία μίας νέας (μετα)κλάσης/ιδιότητας, το φιλτράρισμα ενός συνόλου (μετα)κλάσεων/ιδιοτήτων, τη δημιουργία σχέσεων υπαλληλίας, την απόδοση πληθυσμού σε μία (μετα)κλάση/ιδιότητα και το φιλτράρισμα/αναδιοργάνωση αρχικών ιεραρχιών υπαλληλίας. Για να είναι σε θέση η RQL να υποστηρίξει το σχηματισμό εκφράσεων που ορίζουν όψεις, πρέπει να προστεθούν δομές για τον ορισμό των βασικών στοιχείων που απαρτίζουν μία όψη. Θα ονομάζουμε εφεξής για λόγους συντομίας **RVL** (RQL-View Language) τη βασική γλώσσα επερώτησης/όψεων που προσφέρει κατάλληλες συναρτήσεις για την κατασκευή (μετα)κλάσεων, ιδιοτήτων και ιεραρχιών υπαλληλίας στο νέο σχήμα (όψη). Το πλεονέκτημα της RVL είναι η απλότητά της, εφόσον επιτελεί τις παραπάνω λειτουργίες με δύο βασικούς τελεστές (τελεστής υποστασιοποίησης και τελεστής υπαλληλίας), εκμεταλλευόμενη αφενός το υποκείμενο μοντέλο δεδομένων (σύστημα τύπων) και την ύπαρξη επιπέδων αφαίρεσης και αφετέρου την εκφραστική δύναμη της βασικής γλώσσας επερώτησης.

4.3.1 Κατασκευή ενός ιδεατού σχήματος

Τυπικά, μία RDF/S όψη ορίζεται ως ένα ιδεατό RDF σχήμα, αποτελούμενο από ένα σύνολο από ορισμούς κλάσεων και ιδιοτήτων και τις ιεραρχίες που ορίζονται μεταξύ τους. Πρακτικά, μία RDF/S όψη αποτελείται από ένα σύνολο από δηλώσεις ορισμού τμημάτων της όψης. Μία δήλωση ορισμού αναφέρεται στη δημιουργία νέων ιδεατών (μετα)κλάσεων/ιδιοτήτων, στην επαναχρησιμοποίηση ενός συνόλου (μετα)κλάσεων/ιδιοτήτων, στην απόδοση μελών στις ιδεατές (μετα)κλάσεις/ιδιότητες και στη δημιουργία ή επαναχρησιμοποίηση σχέσεων υπαλληλίας ανάμεσα στις κλάσεις (ιδεατές και βασικές), χρησιμοποιώντας εκφράσεις της γλώσσας ορισμού όψεων. Όντας ένας ιδεατός χώρος ονοματοδοσίας, μία RDF/S όψη διακρίνεται από ένα μοναδικό URI, το οποίο αποδίδεται από τον κατασκευαστή της όψης. Το URI αυτό αποτελεί το πρώτο συνθετικό των μοναδικών αναγνωριστικών που

προσδιορίζουν τις ιδεατές κατασκευές. Γενικά, ο ορισμός μίας RDF/S όψης έχει τη μορφή:

[VIEW *operator*
FROM *RQL_path_expression*
WHERE *filtering_conditions*
USING NAMESPACE *root_schema_namespace*]
[……]
USING NAMESPACE *root_schema_namespace*
CREATE NAMESPACE *RVL_view_namespace*

Μία όψη αποτελείται από ένα σύνολο δηλώσεων ορισμού όψης, η μορφή των οποίων αναπαρίσταται μέσα στις αγκύλες. Ο όρος *VIEW*, όταν χρησιμοποιείται με έναν από τους τελεστές που προδιαγράφει η γλώσσα RVL, δημιουργεί στο ιδεατό σχήμα κατασκευές του τύπου που ορίζει ο χρησιμοποιούμενος τελεστής. Ο όρος *FROM* υποδεικνύει την έκφραση μονοπατιού που χρησιμοποιείται για την αποτίμηση της μεταβλητής ή των μεταβλητών που ορίζονται στον όρο *VIEW*, σε μία προσπάθεια να σχηματίσουμε, με μία δήλωση ορισμού όψης, ένα σύνολο από εκφράσεις ορισμού όψης. Ο όρος αυτός δεν μπορεί να αναφέρεται σε τμήματα του χώρου ονοματοδοσίας της οριζόμενης όψης. Ο όρος *WHERE* φιλοξενεί το σύνολο των κριτηρίων φιλτραρίσματος της ανακτημένης πληροφορίας από το αρχικό σχήμα και την αρχική βάση περιγραφών, ενώ ο όρος *USING NAMESPACE* χρησιμοποιείται για να δηλώσει το πρόθεμα ενός ή περισσότερων χώρων ονοματοδοσίας που χρησιμοποιούνται στη συγκεκριμένη RQL επερώτηση. Ακολουθεί η προαιρετική χρήση της πρότασης *USING NAMESPACE*[45], με την οποία δηλώνουμε -για λόγους ευχρηστίας- καθολικά προθέματα που αντιστοιχούν σε χώρους ονοματοδοσίας, από τους οποίους επαναχρησιμοποιούμε συνεχώς κλάσεις και ιδιότητες, συμπεριλαμβανομένων και χώρων ονοματοδοσίας όψεων (δηλαδή τα αρχικά σχήματα). Και στις δύο περιπτώσεις (τοπική και καθολική δήλωση χώρων ονοματοδοσίας), η δήλωση των προκαθορισμένων *rdf* και *rdfs* χώρων ονοματοδοσίας είναι προαιρετική, δεδομένου ότι είναι ενσωματωμένοι στο χώρο ονοματοδοσίας κάθε όψης (όπως γίνεται και στα βασικά RDF σχήματα).

Τέλος, έπεται η δήλωση *CREATE NAMESPACE*, όπου δηλώνουμε το URI του χώρου ονοματοδοσίας που θα δημιουργήσουμε για την όψη -κατά αντιστοιχία με

[45] Όταν δεν χρησιμοποιείται κάποιος χώρος ονοματοδοσίας, τότε εννοείται αυτός της όψης.

το χώρο ονοματοδοσίας ενός σχήματος- και ένα πρόθεμα (*nsname*) του χώρου αυτού, που διευκολύνει την εύκολη αναφορά του. Για παράδειγμα, η δήλωση

CREATE NAMESPACE myview=&http://www.ics.forth.gr/view.rdf

δημιουργεί ένα ιδεατό σχήμα με URI *http://www.ics.forth.gr/view.rdf*, ο οποίος αναγνωρίζεται συνοπτικά με το πρόθεμα *myview* (το πρόθεμα δεν είναι υποχρεωτικό). Το URI αυτό χαρακτηρίζει με μοναδικό τρόπο τα στοιχεία της αντίστοιχης όψης. Για παράδειγμα, μία ιδεατή κλάση «Culture_products», η οποία ορίζεται στην όψη του παραπάνω παραδείγματος, θα αναγνωρίζεται μοναδικά από το URI http://www.ics.forth.gr/view.rdf#Culture_products. Η δυνατότητα σύνθεσης όψεων που υποστηρίζει ο συγκεκριμένος μηχανισμός ορισμού όψεων, μας επιτρέπει να χρησιμοποιήσουμε το αποδιδόμενο στην όψη αναγνωριστικό στον ορισμό μίας άλλης όψης.

4.3.2 Τελεστές της γλώσσας ορισμού όψεων

Όπως έχουμε ήδη αναφέρει, το πλεονέκτημα της προτεινόμενης γλώσσας ορισμού όψεων είναι ότι επιτελεί τις απαιτούμενες λειτουργίες για τον ορισμό RDF/S όψεων με τη χρήση δύο τελεστών: του **τελεστή υποστασιοποίησης** (**()**) και του **τελεστή υπαλληλίας** (**< >**). Κατά αναλογία με τη χρησιμοποιούμενη γλώσσα επερώτησης RQL, για να εξασφαλιστεί η εγκυρότητα μίας δήλωσης ορισμού όψης αλλά και για να εξαχθεί ο τύπος μίας ιδεατής κατασκευής, οι τελεστέοι μίας έκφρασης πρέπει να είναι ενός συγκεκριμένου τύπου, ο οποίος ελέγχεται κατά τη διάρκεια της μεταγλώττισης. Η RVL χρησιμοποιεί το σύστημα τύπων της RQL, το οποίο της εξασφαλίζει τη δυνατότητα σύνθεσης βασικών εκφράσεων ορισμού όψεων σε πιο πολύπλοκες εκφράσεις.

Επίσης, η ύπαρξη του συστήματος τύπων μας επιτρέπει να αποφύγουμε τη ρητή δήλωση κάποιων ορισμών όψεων. Αυτό σημαίνει ότι, δεδομένης της γνώσης του τύπου ενός στοιχείου από ένα αρχικό σύνολο πληροφορίας, μπορούμε να εισάγουμε τη σχετική πληροφορία στην όψη, χωρίς να απαιτείται η άμεση δήλωσή της μέσω της κατάλληλης έκφρασης ορισμού όψης. Στην περίπτωση αυτή, το στοιχείο (μετα/κλάση, ιδιότητα, πόρος) εισάγεται στην όψη με τον τύπο που διατηρεί στο αρχικό σχήμα. Ενδεικτικές περιπτώσεις που ενδέχεται να αξιοποιήσουμε τη λειτουργικότητα αυτή είναι η δήλωση του πεδίου ορισμού και τιμών μίας ιδιότητας ή η επαναχρησιμοποίηση ιεραρχιών υπαλληλίας. Το γεγονός αυτό βέβαια δεν μας εμποδίζει να εισάγουμε ένα αρχικό στοιχείο μέσα στην όψη

με τον τύπο που επιθυμούμε. Στην περίπτωση αυτή, χρησιμοποιούμε τον κατάλληλο τελεστή απόδοσης τύπου (τελεστής υποστασιοποίησης).

Η εφαρμογή ενός τελεστή στους τελεστέους μίας έκφρασης ορισμού όψης παράγει ένα αποτέλεσμα ενός συγκεκριμένου τύπου και οι συνθήκες εγκυρότητας που πρέπει να ισχύουν πάνω στους τελεστέους μίας RVL έκφρασης χρησιμοποιούνται για τον έλεγχο τύπων. Αναπαριστούμε τη διαδικασία αυτή με κανόνες τύπων (typing rules) της μορφής: $\tau_\alpha \times \tau_\beta \rightarrow \tau_\gamma$. Οι κανόνες τύπων ερμηνεύονται ως εξής*: αν οι δύο τελεστέοι της έκφρασης είναι αντίστοιχα του τύπου* τ_α *και* τ_β*, τότε η εφαρμογή του τελεστή σε αυτούς παράγει ένα αποτέλεσμα του τύπου* τ_γ. Κάθε λειτουργία που επιτελεί ένας συγκεκριμένος τελεστής συνοδεύεται από ένα σύνολο κανόνων τύπων και κατά περίπτωση από κάποιους περιορισμούς (π.χ μοναδικότητα ονομάτων). Οι κανόνες τύπων και οι περιορισμοί που αντιστοιχούν στους τελεστές παρουσιάζονται στην επόμενη ενότητα, όπου περιγράφεται ανά λειτουργικότητα η χρήση των τελεστών που προδιαγράφει η RVL.

Όπως είδαμε στην **Ενότητα 2.1.1**, μία RDF βάση μπορεί να αναπαρασταθεί με ένα επίπεδο τρόπο ως ένα σύνολο από τριάδες της μορφής <υποκείμενο, κατηγόρημα, αντικείμενο>. Επιχειρώντας μία (λογική) ερμηνεία των τελεστών που προδιαγράφει η γλώσσα ορισμού όψεων RVL, χρησιμοποιούμε τη δομή των τριάδων, υποθέτοντας ότι κάθε κατασκευαστής (τελεστής) συνεισφέρει ένα σύνολο από τριάδες στο παραγόμενο ιδεατό σχήμα, δηλαδή την όψη. Με άλλα λόγια, το αποτέλεσμα της εφαρμογής ενός τελεστή πάνω σε ένα σύνολο από τελεστέους αναπαρίσταται με τη μορφή (ενός συνόλου) τριάδων. Το σύνολο όλων των τριάδων που αντιστοιχούν στις εφαρμογές των τελεστών της RVL στην πληροφορία των αρχικών σχημάτων και βάσεων περιγραφών, όπως αυτές καθορίζονται στις δηλώσεις ορισμού μίας όψης, αναπαριστούν ουσιαστικά το παραγόμενο σύνολο πληροφορίας (ιδεατό σχήμα και βάση περιγραφών). Πιο συγκεκριμένα και σύμφωνα με το τυπικό μοντέλο που παρουσιάσαμε, η εφαρμογή των τελεστών στους τελεστέους, όπως ορίζουν οι εκφράσεις ορισμού της όψης, αντιστοιχούν στη δημιουργία ενός επεκταμένου σχήματος. Η προβολή αυτού του σχήματος αποτελεί το ιδεατό σχήμα και συνίσταται στην αφαίρεση πλεονασματικών ακμών υποστασιοποίησης και σχέσεων υπαλληλίας. Η αφαίρεση αυτή γίνεται π.χ βάσει των κανόνων ότι ένα στιγμιότυπο πρέπει να ανήκει στην πιο ειδική του κλάση και ότι οι ιεραρχίες που σχηματίζονται σε ένα ιδεατό σχήμα πρέπει να είναι ελάχιστες (δηλαδή δεν περιέχουν πλεονάζουσες ακμές υπαλληλίας).

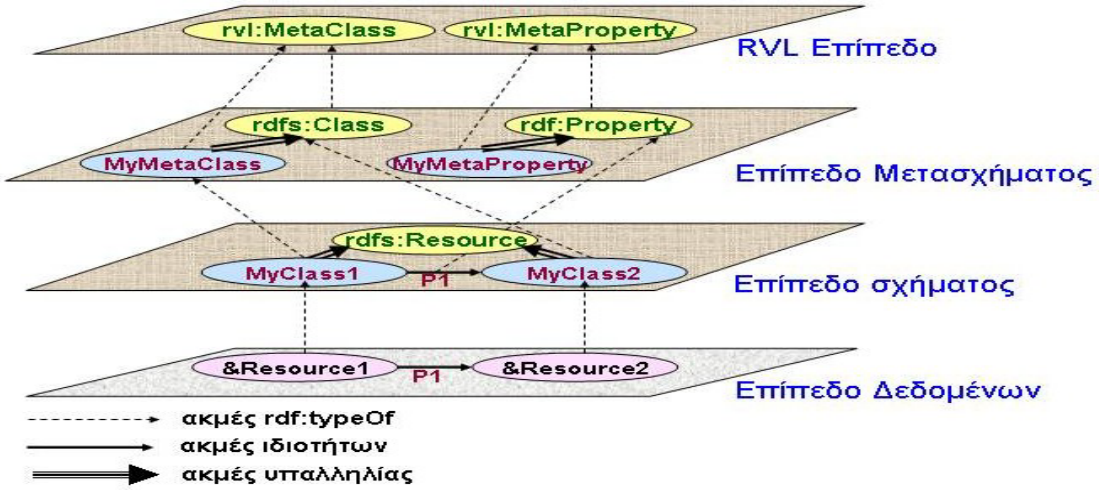
Στη συνέχεια, θα παρουσιάσουμε τους τελεστές που υποστηρίζει η γλώσσα RVL και θα δείξουμε τη λειτουργικότητά τους χρησιμοποιώντας το παράδειγμα της πολιτισμικής εφαρμογής, η γραφική απεικόνιση της οποίας φιλοξενείται στην **Εικόνα 4.1**. Εφόσον η RVL μας επιτρέπει να ορίζουμε ιδεατά σχήματα, μπορούμε να ορίσουμε ακόμα και το σχήμα του παραδείγματος αυτού ως ένα ιδεατό σχήμα, χρησιμοποιώντας RVL εκφράσεις ορισμού όψεων.

4.3.2.1 Τελεστής υποστασιοποίησης

Ο τελεστής υποστασιοποίησης εκμεταλλεύεται την ύπαρξη των επιπέδων αφαίρεσης που διακρίνει το RDF/S μοντέλο δεδομένων για να προσφέρει ένα σύνολο από υπηρεσίες. Ειδικότερα, μέσω του τελεστή υποστασιοποίησης, που συμβολίζεται με **()**, επιτελούμε τις λειτουργίες:

- Δημιουργία ιδεατών (μετα)κλάσεων και ιδιοτήτων
- Απόδοση πληθυσμού σε ιδεατές μετακλάσεις, κλάσεις και ιδιότητες.

Η δύναμη του τελεστή υποστασιοποίησης αποδίδεται αφενός στο σύστημα τύπων που χρησιμοποιείται και αφετέρου στη διάκριση των επιπέδων αφαίρεσης. Για να είναι σε θέση να υποστηρίξει τη δημιουργία ιδεατών κατασκευών σε επίπεδο μετασχήματος (δηλαδή ιδεατές μετακλάσεις κλάσεων και ιδιοτήτων), η RVL διακρίνει ένα ακόμα επίπεδο μετασχήματος, το *RVL επίπεδο*. Η **Εικόνα 4.6** αναπαριστά γραφικά τα επίπεδα αφαίρεσης που διακρίνει η RVL, προκειμένου να είναι σε θέση να δημιουργήσει ιδεατές κατασκευές σε όλα τα εννοιολογικά επίπεδα: δεδομένων, σχήμα και μετασχήμα.



Εικόνα 4.6: RVL επίπεδα αφαίρεσης

Στο RVL επίπεδο ορίζονται οι RVL μετακλάσεις *rvl:MetaClass και rvl:MetaProperty*, των οποίων στιγμιότυπα αποτελούν αντίστοιχα μετακλάσεις κλάσεων και μετακλάσεις ιδιοτήτων (οι ειδικά δεσμευμένες αυτές μετακλάσεις έχουν ειδικό, ενσωματωμένο στη γλώσσα RVL τύπο τ_{Wc} και τ_{Wp} αντίστοιχα). Δεδομένης της διάκρισης των επιπέδων αφαίρεσης, η εφαρμογή του τελεστή υποστασιοποίησης στο *RVL επίπεδο* δημιουργεί μετακλάσεις κλάσεων και ιδιοτήτων, ενώ η εφαρμογή του σε ιδεατές κατασκευές στο *επίπεδο μετασχήματος* δημιουργεί κλάσεις/ιδιότητες και αποδίδει πληθυσμό στις ιδεατές μετακλάσεις. Ειδικότερα, η απόδοση πληθυσμού στις βασικές μετακλάσεις *rdfs:Class* και *rdf:Property* ισοδυναμεί με τη δημιουργία κλάσεων και ιδιοτήτων αντίστοιχα, ενώ η εφαρμογή του σε ορισμένες από το χρήστη μετακλάσεις δημιουργεί στιγμιότυπα αυτών. Εφαρμόζοντας τον τελεστή υποστασιοποίησης σε ιδεατές κατασκευές στο *επίπεδο σχήματος* (ορισμένες από το χρήστης ιδεατές κλάσεις και ιδιότητες), αποδίδουμε πληθυσμό σε αυτές. Ο βασικός περιορισμός που πρέπει να ισχύει σε όλες τις περιπτώσεις είναι *η μοναδικότητα των ονομάτων*, δηλαδή τα ονόματα ιδεατών (μετα)κλάσεων και ιδιοτήτων είναι μοναδικά. Επιπλέον, εξαιτίας της δυνατότητάς του να διακρίνει τα επίπεδα αφαίρεσης ενός RDF/S γράφου, ο τελεστής υποστασιοποίησης μπορεί να χρησιμοποιηθεί για να μεταβάλλει την εννοιολογική στάθμη των στοιχείων ενός αρχικού συνόλου πληροφορίας μέσα στην όψη. Η γενική σύνταξη[46] του τελεστή είναι:

VIEW Symbol**(**Expression**)**

{FROM RQL_path_expression}

{WHERE <filter>}

{USING NAMESPACE ns1=&URI,…}

όπου Symbol=*Variable*|*String*|*Qnames* (δηλαδή URI της μορφής namespace:name) και Expression=*Symbol*|*Function*, όπου ο συμβολισμός *Function* αντιπροσωπεύει συναρτήσεις που επιστρέφουν αντικείμενα τύπου τ_{string}.

---

[46] Στην περίπτωση της δημιουργίας ιδιοτήτων, ο τελεστής υποστασιοποίησης έχει διαφορετική σύνταξη, δεδομένου ότι δεν αρκεί η δήλωση μίας ιδιότητας ως στιγμιότυπο της βασικής μετακλάσης rdf:Property, αλλά απαιτείται και η δήλωση του πεδίου ορισμού και τιμών της. Σύμφωνα με τους περιορισμούς του RDF/S μοντέλου δεδομένων που υιοθετήσαμε, τα πεδία αυτά όχι μόνο πρέπει να συνοδεύουν τον ορισμό μίας ιδιότητας, αλλά πρέπει να είναι και μοναδικά. Την δομή του τελεστή υποστασιοποίησης σε αυτή την περίπτωση θα εξετάσουμε αναλυτικά παρακάτω. Μολαταύτα, η λειτουργικότητα που προσφέρει ο τελεστής είναι κοινή και στις δύο περιπτώσεις.

Οι τριάδες που αντιστοιχούν στη λογική ερμηνεία του τελεστή υποστασιοποίησης έχουν ως κατηγόρημα τη βασική ιδιότητα *rdf:type*, ως αντικείμενο το όνομα ή τα ονόματα που προκύπτουν από την αξιολόγηση της μεταβλητής στην αριστερή μεριά του κατασκευαστή και ως υποκείμενο ένα από τα στοιχεία μέσα στις παρενθέσεις. Για παράδειγμα, η RVL έκφραση

VIEW rdfs:Class("Artist")

USING NAMESPACE rdfs=&http://www.w3.org/2000/01/rdf-schema#

που ορίζει μία νέα ιδεατή κλάση με το όνομα «Artist», συνεισφέρει στην όψη την τριάδα *<nsview:Artist rdf:type rdfs:Class>*, όπου με nsview συμβολίζουμε στο εξής τη συντόμευση του URI του χώρου ονοματοδοσίας που ανατίθεται στην όψη με τη χρήση του όρου CREATE NAMESPACE.

Οι εκφράσεις

VIEW rvl:MetaClass("WebResource")

και

VIEW rvl:MetaProperty("SchemaProperty")

ορίζουν αντίστοιχα μία ιδεατή μετακλάση κλάσεων (*nsview:WebResource*) και μία ιδεατή μετακλάση ιδιοτήτων (*nsview:SchemaProperty*). Η συνεισφορά των εκφράσεων αυτών στο ιδεατό σχήμα είναι αντίστοιχα οι τριάδες

<nsview:WebResource rdfs:subClassof rdfs:Class>

και

<nsview:SchemaProperty rdfs:subClassOf rdfs:Class>

δεδομένου ότι με αυτό τον τρόπο ορίζουμε μετακλάσεις κλάσεων και ιδιοτήτων σε ένα RDF σχήμα (**Ενότητα 4.1.1**). Σε όλες τις περιπτώσεις, αυτό που παράγει ο τελεστής υποστασιοποίησης είναι το ελάχιστο σύνολο δηλώσεων υποστασιοποίησης (rdf:type δηλώσεις) που κρατούν την όψη συνεπή (αφαιρούνται όλες οι πλεονασματικές δηλώσεις υποστασιοποίησης). Στις επόμενες παραγράφους εξετάζουμε τη λειτουργία του τελεστή υποστασιοποίησης για ιδεατές (μετα)κλάσεις και ιδιότητες. Ανά περίπτωση, παρουσιάζουμε και τους κανόνες τύπων που χρησιμοποιούνται.

4.3.2.1.1 Ιδεατές (Μετα)Κλάσεις

ΔΗΜΙΟΥΡΓΙΑ ΙΔΕΑΤΩΝ (ΜΕΤΑ)ΚΛΑΣΕΩΝ: Οι ιδεατές (μετα)κλάσεις μας προσφέρουν έναν τρόπο να αποδώσουμε μία διαφορετική εννοιολογική οργάνωση σε

ένα σύνολο από πληροφοριακούς πόρους. Για τη δημιουργία μίας ή περισσότερων ιδεατών (μετα)κλάσεων χρησιμοποιούμε τον τελεστή της υποστασιοποίησης. Η σύνταξη του τελεστή σε αυτές τις περιπτώσεις είναι:

VIEW Symbol(Expression)

{FROM <path_expression>}

{WHERE <filter>}

{USING NAMESPACE nsname1=&URI,…}

όπου Symbol=*Class*|*rvl:MetaClass*|*rvl:MetaProperty*, Expression= *Variable*|*String*|*Qnames*|*Function*, με το συμβολισμό *Function* να αντιπροσωπεύει συναρτήσεις που επιστρέφουν αντικείμενα τύπου τ_{string}. Ο όρος FROM χρησιμοποιείται στην περίπτωση που χρησιμοποιούμε μεταβλητή εντός των παρενθέσεων και ο όρος WHERE δηλώνει τα κριτήρια φιλτραρίσματος.

Για την ορθότητα της χρήσης του τελεστή, επιβάλλουμε την ύπαρξη του ακόλουθου κανόνα τύπου:

$$\tau_{Mc} \times (\text{string} \mid \tau_C \mid \tau_{Mc} \mid \tau_{Mp} \mid \tau_p \mid \tau_U) \rightarrow \tau_C \qquad (1)$$

Ο πρώτος τελεστέος αντιπροσωπεύει τις βασικές μετακλάσεις στις οποίες αποδίδουμε πληθυσμό και τα στοιχεία εντός των παρενθέσεων δηλώνουν τους εναλλακτικούς τύπους που μπορεί να έχει ο δεύτερος τελεστέος του τελεστή υποστασιοποίησης. Ο **Κανόνας 1** προδιαγράφει τον τρόπο δημιουργίας κλάσεων[47]. Σύμφωνα με τον κανόνα αυτό, μπορούμε να δημιουργήσουμε (μετα)κλάσεις βάσει απλών αλφαριθμητικών και αρχικών κλάσεων, ιδιοτήτων, μετακλάσεων ή ακόμα και πόρων. Με αυτό τον τρόπο, με τη χρήση του τελεστή υποστασιοποίησης και χρησιμοποιώντας το σύστημα τύπων του RDF/S μοντέλου δεδομένων, μπορούμε να μεταβάλλουμε μέσα στην όψη την εννοιολογική στάθμη ενός αρχικού στοιχείου. Για παράδειγμα, μία ιδεατή μετακλάση ενός αρχικού σχήματος μπορεί να οριστεί ως μία ιδεατή κλάση μέσα στην οριζόμενη όψη. Η δυνατότητα αυτή μας προσφέρει μεγάλη ευελιξία, ειδικότερα σε περιβάλλοντα όπου χρειάζεται να αντιμετωπίσουμε σημαντικές ανομοιογένειες στη μοντελοποίηση των πληροφοριακών πόρων.

Η πιο απλή περίπτωση δημιουργίας μίας ιδεατής (μετα)κλάσης πραγματοποιείται χρησιμοποιώντας ένα αλφαριθμητικό. Αν αντί για ένα

[47] Δεν μπορούμε να δημιουργήσουμε ιδεατές (μετα)κλάσεις από ακέραιους, πραγματικούς, τύπους Boolean ή ημερομηνίες.

αλφαριθμητικό, χρησιμοποιήσουμε μια μεταβλητή, τότε μπορούμε να ορίσουμε ένα *σύνολο* από ιδεατές (μετα)κλάσεις. Για παράδειγμα, η έκφραση ορισμού όψης

VIEW Class(X)

FROM subclassof(ns1:Artist,1){X}

USING NAMESPACE ns1=&http://www.icom.com/schema.rdf#

ορίζει ένα σύνολο από ιδεατές κλάσεις. Ειδικότερα, η αποτίμηση της επερώτησης που καθορίζεται στον όρο FROM, αποτελεί το σύνολο των στοιχείων που θα οριστούν ως ιδεατές κλάσεις στην όψη. Στη συγκεκριμένη περίπτωση, ο τύπος των στοιχείων αυτών είναι τ_C και εφαρμόζεται ο Κανόνας 1. Επομένως, οι κλάσεις *ns1:Sculptor* και *ns1:Painter*, που αποτελούν τις υποκλάσεις της κλάσης *ns1:Artist* σε βάθος 1, ορίζονται ως κλάσεις του ιδεατού σχήματος και η λογική ερμηνεία της έκφρασης είναι το σύνολο των τριάδων

<nsview:Sculptor rdf:type rdfs:Class>
<nsview:Painter rdf:type rdfs:Class>.

Αν αντί για τον τελεστέο *Class* στην παραπάνω έκφραση, χρησιμοποιήσουμε τον τελεστέο rvl:MetaClass (rvl:MetaProperty), τότε δημιουργούμε τις αντίστοιχες ιδεατές μετακλάσεις κλάσεων (ιδιοτήτων). Μία παραλλαγή της έκφρασης αυτής και ειδικότερα η

VIEW Class("Greek_"+X)

FROM subclassof(ns1:Artist,1){X}

USING NAMESPACE ns1=&http://www.icom.com/schema.rdf#

μας δίνει τη δυνατότητα να κάνουμε κάποιου είδους μετονομασία στις επαναχρησιμοποιούμενες κατασκευές. Αυτό καθίσταται δυνατό λόγω του ότι η RQL υποστηρίζει τη συναρτησιακή σύνθεση των τελεστών και μετατροπές τύπων (type coercions). Στη συγκεκριμένη περίπτωση, ο τύπος τ_C της μεταβλητής X μετατρέπεται σε τύπου αλφαριθμητικό εξαιτίας του «+» (που μετατρέπει το τ_C.σε αλφαριθμητικό). Επομένως, με την παραπάνω έκφραση σχηματίζονται οι ιδεατές κλάσεις *nsview:Greek_Sculptor* και *nsview:Greek_Painter* και η λογική ερμηνεία της έκφρασης είναι το σύνολο των πλειάδων

<nsview:Greek_Sculptor rdf:type rdfs:Class>
<nsview:Greek_Painter rdf:type rdfs:Class>.

Επίσης, η έκφραση

VIEW Class(X)

FROM ns1:technique{X}

USING NAMESPACE ns1=&http://www.icom.com/schema.rdf#

δημιουργεί ένα σύνολο από ιδεατές κλάσεις χρησιμοποιώντας ένα σύνολο από αρχικές τιμές δεδομένων, δεδομένου ότι η RQL επερώτηση στον όρο FROM επιστρέφει το σύνολο των τιμών που έχουν αποδοθεί στο γνώρισμα *technique* της κλάσης *Painting* (τύπου τ_U). Ο κανόνας που χρησιμοποιείται σε αυτή την περίπτωση είναι ο Κανόνας 1, με δεύτερο τελεστέο τον τύπο τ_U. Με τον τρόπο αυτό, ουσιαστικά προάγουμε μία τιμή από το επίπεδο δεδομένων στο επίπεδο κλάσεων. Έτσι, στο συγκεκριμένο παράδειγμα, δημιουργείται η κλάση *nsview:oil_on_canvas*, την οποία στη συνέχεια μπορούμε να χρησιμοποιήσουμε π.χ για να οργανώσουμε τα έργα ζωγραφικής που χρησιμοποιούν αυτή την τεχνική. Η λογική ερμηνεία της έκφρασης είναι η τριάδα *<nsview:oil_on_canvas rdf:type rdfs:Class>*.

Στην περίπτωση που δεν θέλουμε να μεταβάλλουμε τον τύπο των αρχικών στοιχείων, η RVL υποστηρίζει συντομογραφίες του τύπου

VIEW X

FROM subClassOf(ns1:Artist){X}

USING NAMESPACE ns1=&http://www.icom.com/schema.rdf#

που εισάγει στην όψη τις υποκλάσεις της κλάσης *ns1:Artist* μαζί με τον τύπο τους. Στη συγκεκριμένη περίπτωση, η ερμηνεία της παραπάνω έκφρασης είναι το σύνολο των τριάδων

<nsview:Painter rdf:type rdfs:Class>
<nsview:Scupltor rdf:type rdfs:Class>
<nsview:Cubist rdf:type rdfs:Class>
<nsview:Flemish rdf:type rdfs:Class>

Ανάλογες κατασκευές και πειραματισμούς μπορούμε να κάνουμε μεταβάλλοντας τον τύπο των αρχικών στοιχείων που χρησιμοποιούνται στη δημιουργία ιδεατών (μετα)κλάσεων.

ΑΠΟΔΟΣΗ ΠΛΗΘΥΣΜΟΥ ΣΕ ΙΔΕΑΤΕΣ (ΜΕΤΑ)ΚΛΑΣΕΙΣ: Η δημιουργία ιδεατών (μετα)κλάσεων αποτελεί έναν τρόπο διαφορετικής μοντελοποίησης σε επίπεδο (μετα)σχήματος. Σε επίπεδο δεδομένων, απαιτείται η ταξινόμηση των πληροφοριακών πόρων στο ιδεατό σχήμα, δηλαδή η απόδοση πληθυσμού σε

ιδεατές (μετα)κλάσεις[48]. Αυτό επιτυγχάνεται από τον τελεστή υποστασιοποίησης, ο οποίος στην περίπτωση αυτή συντάσσεται ως εξής:

VIEW Symbol(Expression)

{FROM <path_expression>}

{WHERE <filter>}

{USING NAMESPACE nsname1=&URI,…}

όπου Symbol=*Variable*|*Qnames*, Expression= *Variable*|*Qnames*|*URI*|*Function* και *Function* αντιπροσωπεύει συναρτήσεις που επιστρέφουν αντικείμενα τύπου τ_U ή τ_C ή τ_P. Ο περιορισμός που τίθεται σε αυτή την περίπτωση είναι το Symbol να αναφέρεται σε ονόματα κλάσεων και ιδιοτήτων που έχουν οριστεί μέσα στην όψη.

Εξαιτίας της διάκρισης επιπέδων αφαίρεσης και του συστήματος τύπων που χρησιμοποιείται, μία ιδεατή μετακλάση κλάσεων πρέπει να έχει ως στιγμιότυπα κλάσεις, μία μετακλάση ιδιοτήτων πρέπει να έχει στιγμιότυπα ιδιότητες, ενώ μία κλάση πόρους. Εξασφαλίζουμε τα παραπάνω με τη χρήση των ακόλουθων κανόνων τύπων:

$\tau_C \times \tau_U \rightarrow \tau_U$ (2)

$\tau_{Mc} \times \tau_C \rightarrow \tau_C$ (3)

$\tau_{Mp} \times \tau_P \rightarrow \tau_P$ (4)

Ο **Κανόνας 2** αναφέρεται στην απόδοση πληθυσμού στις κλάσεις, όπου ουσιαστικά δημιουργούμε στιγμιότυπα αυτών. Ο **Κανόνας 3** αποτελεί υποπερίπτωση του **Κανόνα 1** και εφαρμόζεται στην περίπτωση των ορισμένων από το χρήστη μετακλάσεων κλάσεων. Ο **Κανόνας 4** αναφέρεται στην απόδοση πληθυσμού σε ορισμένες από το χρήστη μετακλάσεις ιδιοτήτων.

Παραδείγματος χάριν, η έκφραση

VIEW Painter(&www.culture.net/dali)

αποδίδει τον τύπο *Painter* στο URI www.culture.net/dali και η λογική ερμηνεία της έκφρασης είναι *<www.culture.net/dali rdf:type nsview:Painter>*. Η έκφραση

VIEW WebResource(ExtResource)

έχει λογική ερμηνεία *<nsview:ExtResource rdf:type nsview:WebResource>* και για να είναι ορθή πρέπει ο τύπος του ονόματος *nsview:ExtResource* να είναι κλάση, αν

[48] Το επίπεδο σχήματος αποτελεί το επίπεδο δεδομένων για το επίπεδο μετασχήματος. Για το λόγο αυτό, χρησιμοποιούμε τον όρο *απόδοση πληθυσμού σε μετακλάσεις*.

ο τύπος του *nsview:WebResource* είναι μετακλάση κλάσεων (Κανόνας 3). Αν επίσης έχουμε την έκφραση

VIEW PicassoWorksInOil(Y)

FROM ns1:Artist{X}.paints{Y}.technique{Z}, {X}lname{W}

WHERE W like "*Picasso*" and Z like "oil*"

USING NAMESPACE ns1=&http://www.icom.com/schema.rdf#

όπου το όνομα *PicassoWorksInOil* έχει ήδη οριστεί ως κλάση, τότε αποδίδουμε στην κλάση αυτή το σύνολο των πόρων που αποδίδονται στον καλλιτέχνη Picasso και έχουν ζωγραφιστεί με την τεχνική oil on canvas, δηλαδή τους πόρους &r2 και &r3 (Κανόνας 2). Αν χρησιμοποιήσουμε μεταβλητή στα αριστερά του τελεστή, δηλαδή σχηματίζουμε εκφράσεις της μορφής

VIEW X(&http://www.louvre.fr/)

FROM bag(ns2:ExtResource, ns1:Museum){X}

USING NAMESPACE ns1=&http://www.icom.com/schema.rdf#,

ns2=&www.oclc.org/schema.rdf

τότε αποδίδουμε ένα σύνολο τύπων σε ένα πόρο, άρα έχουμε πολλαπλή ταξινόμηση του πόρου. Στη συγκεκριμένη περίπτωση, ο πόρος *www.louvre.fr* αποδίδεται στις ιδεατές κλάσεις *nsview:Museum* και *nsview:ExtResource*, οι οποίες ορίζονται μέσα στην όψη (Κανόνας 2). Η λογική ερμηνεία αυτής της έκφρασης είναι αυτονόητη. Στην περίπτωση που χρησιμοποιούνται μεταβλητές και στις δύο θέσεις του τελεστή, σχηματίζονται εκφράσεις, των οποίων η σημασιολογία ισοδυναμεί με την απόδοση ενός συνόλου τύπων σε ένα σύνολο πόρων (σαν καρτεσιανό γινόμενο των τιμών X και Y). Τέλος, στην περίπτωση που έχουμε εκφράσεις της μορφής

VIEW ExtResource(&X)

FROM subClassOf(ns1:Artist, 1){X}

USING NAMESPACE ns1=&http://www.icom.com/schema.rdf#

και η ExtResource είναι μία ιδεατή κλάση, παρατηρούμε άλλη μία λειτουργικότητα του τελεστή υποστασιοποίησης, που μετατρέπει ένα σύνολο αρχικών κλάσεων σε πόρους στο επίπεδο δεδομένων. Στη συγκεκριμένη περίπτωση, και εφαρμόζοντας τον Κανόνα 2, ο τύπος κλάσης της μεταβλητής X μετατρέπεται στον τύπο τ_U λόγω του τελεστή &, που μετατρέπει τον τύπο του X σε τύπο URI. Η λογική ερμηνεία της παραπάνω έκφρασης είναι το σύνολο των τριάδων

<http://www.icom.com/schema.rdf#Painter rdf:type nsview:ExtResource>

<http://www.icom.com/schema.rdf#Sculptor rdf:type nsview:ExtResource>

Ανάλογες εκφράσεις και πειραματισμοί μπορούν να γίνουν μεταβάλλοντας τους τελεστέους του τελεστή υποστασιοποίησης. Με τον τρόπο αυτό, αναδεικνύεται η εκφραστικότητα της γλώσσας RVL.

4.3.2.1.2 Ιδεατές Ιδιότητες

Αν οι ιδεατές (μετα)κλάσεις μας επιτρέπουν να αποδώσουμε μία διαφορετική εννοιολογική οργάνωση σε ένα σύνολο από πληροφοριακούς πόρους, οι ιδεατές ιδιότητες μας επιτρέπουν να ορίσουμε σημασιολογικές συσχετίσεις ανάμεσα σε ιδεατές (μετα)κλάσεις και να δηλώσουμε γνωρίσματα που περιγράφουν καλύτερα, σύμφωνα με τα δικά μας κριτήρια, τη φύση μίας κατηγορίας αντικειμένων. Σύμφωνα με το RDF/S, οι ιδιότητες αντιμετωπίζονται (σχεδόν) ως αυθύπαρκτες οντότητες, που σημαίνει ότι ορίζονται ανεξάρτητα από την κλάση στην οποία αποδίδονται, ενώ μπορούν να εξειδικευτούν σχηματίζοντας ιεραρχίες υπαλληλίας με τη χρήση της βασικής ιδιότητας *rdfs:subPropertyOf*. Την ίδια αντιμετώπιση ακολουθούμε και σε μία RDF/S όψη, γεγονός που σημαίνει ότι οι περιορισμοί που θέτει το RDF/S μοντέλο στις ιδιότητες εξακολουθούν να ισχύουν στην περίπτωση της δημιουργίας όψεων.

ΔΗΜΙΟΥΡΓΊΑ ΙΔΕΑΤΗΣ ΙΔΙΟΤΗΤΑΣ: Η δημιουργία μίας ιδεατής ιδιότητας συνοδεύεται από τη δήλωση του πεδίου ορισμού και τιμών της. Για να λάβει υπόψη του αυτή την ιδιαιτερότητα, ο τελεστής υποστασιοποίησης έχει μία διαφορετική σύνταξη σε σχέση με τη σύνταξη του στην περίπτωση των ιδεατών (μετα)κλάσεων. Επίσης, η λογική ερμηνεία του είναι πιο σύνθετη από εκείνη στην περίπτωση των ιδεατών κλάσεων, δεδομένου ότι πρέπει να προστεθούν και δηλώσεις που αφορούν το πεδίο ορισμού και τιμών της ιδεατής ιδιότητας. Συγκεκριμένα, η γενική του σύνταξη είναι:

VIEW Property(Expression1, Expression2, Expression3)

{FROM <path_expression>}

{WHERE <filter>}

{USING NAMESPACE nsname1=&URI,…}

όπου Expression1=*String*|*Variable*|*Function1* και *Function1* αντιπροσωπεύει συναρτήσεις που επιστρέφουν αντικείμενα τύπου τ_{string}. Επίσης, Expression2= *QNames*|*Function2* και *Function2* αντιπροσωπεύει συναρτήσεις που επιστρέφουν

αντικείμενα τύπου τ_C, τ_{M_C}.*και* τ_{M_P}, ενώ Expression3=*Qnames*|*String*|*Function3* και *Function3* αντιπροσωπεύει συναρτήσεις που επιστρέφουν αντικείμενα τύπου τ_C, τ_{M_C}, τ_{M_P} και τ_L. Ο πρώτος τελεστέος εντός της παρενθέσεως, αντιστοιχεί στο όνομα της ιδεατής ιδιότητας, ο δεύτερος στο πεδίο ορισμού και ο τρίτος στο πεδίο τιμών της. Επομένως, οι τελεστέοι στη δεύτερη και τρίτη θέση εντός του τελεστή, εκτός από ιδεατά ονόματα (μετα)κλάσεων και ιδιοτήτων (συμπεριλαμβανομένων και των βασικών (μετα)κλάσεων *rdfs:*Class, *rdfs:Resource, rdfs:Literal* κτλ) μπορεί να είναι σύνθετες εκφράσεις, των οποίων όμως ο τύπος ανήκει στο σύνολο $\{\tau_C, \tau_{Mc}, \tau_{Mp}\}$ και οι οποίες αναφέρονται σε μία μόνο ιδεατή κατασκευή αυτού του τύπου. Για παράδειγμα, μπορεί να είναι εκφράσεις του τύπου Class("stringname"), domain/ range(property1) ή domain/range(Variable). Μεταβλητή μπορεί να χρησιμοποιηθεί και στην πρώτη θέση του τελεστή για τη δημιουργία ενός συνόλου ιδιοτήτων. Γενικά, ο κανόνας τύπου που ισχύει στην περίπτωση δημιουργίας ιδεατών ιδιοτήτων είναι:

$$\tau_{Mp} \times (\text{string} | \tau_p, \tau_C | \tau_{Mc} | \tau_{Mp}, \tau_C | \tau_{Mc} | \tau_{Mp} | \tau_L) \rightarrow \tau_p\ [\tau_C | \tau_{Mc} | \tau_{Mp}, \tau_C | \tau_{Mc} | \tau_{Mp} | \tau_L] \quad (5)$$

Ο κανόνας αυτός εξασφαλίζει ότι μία ιδεατή ιδιότητα αποδίδεται σε ιδεατές κατασκευές τύπου (μετα)κλάσης και παίρνει τιμές από κατασκευές (μετα)κλάσεων και ατομικές τιμές δεδομένων.

Η πιο απλή περίπτωση δημιουργίας μίας ιδεατής ιδιότητας συνίσταται σε RVL εκφράσεις ορισμού όψης της μορφής

VIEW rdf:Property("creates", Artist, Artifact)

USING NAMESPACE rdf=& http://www.w3.org/1999/02/22-rdf-syntax-ns#

ορίζει την ιδεατή ιδιότητα *nsview:creates* με πεδίο ορισμού την ιδεατή κλάση *nsview:Artist* και πεδίο τιμών την ιδεατή κλάση *nsview:Artifact*, οι οποίες ορίζονται μέσα στην όψη. Η λογική ερμηνεία της έκφρασης είναι το σύνολο των τριάδων

<nsview:creates rdf:type rdf:Property>
<nview:creates rdfs:domain nsview:Artist>
<nsview:creates rdfs:range nsview:Artifact>

Επίσης, μπορούμε επίσης να ορίσουμε αντίστροφες ιδιότητες με εκφράσεις του τύπου

VIEW Property("created_by", range(ns1:creates), domain(ns:1creates))

USING NAMESPACE ns1=&http://www.icom.com/schema.rdf#

η οποία δημιουργεί την ιδεατή ιδιότητα *nsview:created_by* με πεδίο τιμών την ιδεατή κλάση *nsview:Artifact* και πεδίο τιμών την ιδεατή κλάση *nsview:Artist* (ο

τύπος τους εισάγεται επίσης στην όψη). Σε αυτό το παράδειγμα, μπορούμε να παρατηρήσουμε τη δυνατότητα συναρτησιακή σύνθεσης της RVL. Ο τύπος των (μοναδικών) στοιχείων που επιστρέφουν οι εκφράσεις range(ns1:creates) και domain(ns:1creates) είναι κλάση και επιτρέπεται η χρήση τους στην παραπάνω έκφραση ορισμού όψης σύμφωνα με τον Κανόνα 5. Η λογική ερμηνεία της έκφρασης ορισμού όψης είναι προφανής.

Χρησιμοποιώντας μεταβλητή στην πρώτη θέση του τελεστή μπορούμε να δημιουργήσουμε ένα σύνολο από ιδιότητες. Για παράδειγμα, η έκφραση

VIEW Property(@P, Class("Creator"), range(@P))

FROM {;Artist}@P

WHERE namespace(Artist)=&http://www.icom.com/schema.rdf#

USING NAMESPACE ns1=&http://www.icom.com/schema.rdf#

δημιουργεί το σύνολο των ιδεατών ιδιοτήτων {*nsview:last_name, nsview:first_name, nsview:creates*}, καθώς η επερώτηση που χρησιμοποιείται στον όρο FROM προσδιορίζει το σύνολο των ιδιοτήτων που μπορούν να αποδοθούν στην κλάση *ns1:Artist*. Οι τύποι των εκφράσεων @P, Class("Creator") και range(@P) είναι αντίστοιχα τ_p, τ_C και τ_C. Ως πεδίο ορισμού των ιδιοτήτων αυτών δηλώνεται η ιδεατή κλάση *nsview:Creator*, η οποία μάλιστα ορίζεται στην παραπάνω έκφραση λόγω της δυνατότητας σύνθεσης εκφράσεων ορισμού όψεων. Ως πεδίο τιμών των ιδιοτήτων ορίζονται τα αντίστοιχα πεδία τιμών των αρχικών ιδιοτήτων. Η λογική ερμηνεία της έκφρασης αυτής είναι το σύνολο των τριάδων

<nsview:Creator rdf:type rdfs:Class>
<nsview:last_name rdf:type rdf:Property>
<nsview:first_name rdf:type rdf:Property>
<nsview:creates rdf:type rdf:Property>
<nsview:last_name rdfs:domain nsview:Creator>
<nsview:first_name rdfs:domain nsview:Creator>
<nsview:creates rdfs:domain nsview:Creator>
<nsview:last_name rdfs:range xml:string>
<nsview:first_name rdfs:range xml:string>
<nsview:creates rdfs:range nsview:Artifact>
<nsview:Artifact rdf:type rdfs:Class>

όπου το πρόθεμα *xml* αντιστοιχεί στο χώρο ονοματοδοσίας των βασικών XML datatypes[49]. Στην περίπτωση που δεν θέλουμε να μεταβάλουμε τον τύπο των αρχικών στοιχείων από ένα αρχικό σχήμα και επιθυμούμε την εισαγωγή ενός συνόλου στοιχείων στην όψη, η RVL υποστηρίζει επίσης συντομογραφίες του τύπου

VIEW @P

FROM {;ns1:Artist}@P

USING NAMESPACE ns1=&http://www.icom.com/schema.rdf#

οι οποίες εισάγουν στην όψη ένα σύνολο από ιδιότητες, μαζί με τα πεδία ορισμού και τιμών τους (και τους τύπους τους). Στη συγκεκριμένη έκφραση, επαναχρησιμοποιούνται στην όψη οι αρχικές ιδιότητες *ns1:creates*, *ns1:last_name*, *ns1:first_name* με τα αντίστοιχα πεδία ορισμού και τιμών τους.

Οι παραπάνω εκφράσεις αποτελούν απλώς δείγμα των πιθανών εκφράσεων που μπορούν να οριστούν με τη χρήσης της γλώσσας RVL.

ΑΠΟΔΟΣΗ ΠΛΗΘΥΣΜΟΥ ΣΕ ΙΔΕΑΤΕΣ ΙΔΙΟΤΗΤΕΣ: Εκτός όμως από τη δημιουργία ιδεατών ιδιοτήτων, ο τελεστής υποστασιοποίησης χρησιμοποιείται και για την απόδοση πληθυσμού στις ιδεατές ιδιότητες. Η γενική σύνταξη του τελεστή σε αυτή την περίπτωση είναι

VIEW PropertyName(Constant|Variable1, Constant|Value|Variable2)

{FROM <path_expression>}

{WHERE <filter>}

{USING NAMESPACE nsname1=&URI,…}

όπου Constant=*Qname*|*URI* και Value είναι μία τιμή ατομικού τύπου. Ο περιορισμός που τίθεται είναι ότι Constant|eval(Variable1)∈ [[domain(PropertyName)]] και Value|eval(Variable2)∈[[range(PropertyName)]], όπου με [[]] συμβολίζουμε την έκταση της αντίστοιχης κατασκευής (**Ορισμός 18**) και με eval(X) συμβολίζουμε την αποτίμηση της μεταβλητής X. Τα PropertyName καιConstant (αν είναι (μετα)κλάση) πρέπει να ορίζονται στην όψη. Ο κανόνας τύπου που εφαρμόζεται είναι:

[49] Υπενθυμίζουμε ότι οι αρχικές κατασκευές που επαναχρησιμοποιούνται στην κλάση, αν δεν ορίζεται διαφορετικά με τη χρήση του τελεστή υποστασιοποίησης, διατηρούν τον τύπο που έχουν στο αρχικό σύνολο πληροφορίας. Στη συγκεκριμένη περίπτωση, το *nsview:Artifact,* αν δεν ορίζεται διαφορετικά, έχει τύπο κλάσης. Το ίδιο ισχύει και στην περίπτωση της δημιουργίας ιεραρχιών.

$$\tau_P \times (\tau_U \mid \tau_C \mid \tau_P, \tau_U \mid \tau_C \mid \tau_P \mid \text{string} \mid \text{integer} \mid \text{real} \mid \text{date} \mid \text{boolean}) \rightarrow$$
$$[\tau_U \mid \tau_C \mid \tau_P, \tau_P, \tau_U \mid \tau_C \mid \tau_P \mid \text{string} \mid \text{integer} \mid \text{real} \mid \text{date} \mid \text{boolean}] \quad (6)$$

Για παράδειγμα, μία έκφραση απόδοσης πληθυσμού σε μία ιδεατή ιδιότητα *nsview:creates* είναι η εξής:

VIEW creates(&www.culture.net/picasso32, X)

FROM {Y}ns1:creates{X}, {Y}last_name{W}

WHERE W like "*Picasso*"

USING NAMESPACE ns1=&…

η οποία αποδίδει πολλαπλές φορές την ιδεατή ιδιότητα στον πόρο με αναγνωριστικό *www.culture.net/picasso32*. Η λογική ερμηνεία της έκφρασης είναι το σύνολο των τριάδων

<www.culture.net/picasso32 nsview:creates http://www.museum.es/woman.qti>
<www.culture.net/picasso32 nsview:creates http://www.museum.es/guernica.jpg>

Μία άλλη έκφραση ορισμού όψης μπορεί να χρησιμοποιηθεί για να αποδώσει την ίδια τιμή/πόρο σε μία ιδιότητα ενός συνόλου πόρων. Για παράδειγμα, η έκφραση

VIEW last_modified(X, 2002-06-09)

FROM ns1:Museum{X}.ns2:last_modified{Y}

WHERE Y>2000-01-01

USING NAMESPACE ns1=&www.icom.com/schema.rdf#,

ns2=&www.ocic.org/schema.rdf#

αποδίδει στο σύνολο των πόρων που αποτελούν την αποτίμηση της μεταβλητής X την τιμή 2002-06-09. Για να είναι ορθή η πράξη αυτή, θα πρέπει το πεδίο τιμών της ιδεατής ιδιότητας *nsview:last_modified* να είναι τύπου ημερομηνίας και το eval(X) θα πρέπει να ανήκει στην ερμηνεία του πεδίου ορισμού της ιδεατής ιδιότητας. Η λογική ερμηνεία της έκφρασης είναι το σύνολο των τριάδων

<http://www.rodin.fr nsview:last_modified *2002-06-09>*
<http://www.museum.es nsview:last_modified *2002-06-09>.*

4.3.2.2 Τελεστής υπαλληλίας

Ο τελεστής υπαλληλίας, που συμβολίζεται με **< >**, χρησιμοποιείται βασικά για το σχηματισμό ιεραρχιών μετακλάσεων, κλάσεων και ιδιοτήτων. Ο τελεστής αυτός

συνοδεύεται από ένα σύνολο περιορισμών, οι οποίοι επιβάλλονται από το υποκείμενο μοντέλο δεδομένων. Ειδικότερα, καθώς απαγορεύονται οι κύκλοι σε μία ιεραρχία ενός βασικού σχήματος, μία από τις προϋποθέσεις για την εφαρμογή του τελεστή υπαλληλίας είναι η διασφάλιση της μη δημιουργίας κύκλων στις ιεραρχίες που σχηματίζονται στο ιδεατό σχήμα. Επίσης, ο τελεστής υπαλληλίας πρέπει να εφαρμόζεται σε τελεστέους του ίδιου τύπου (τύπου κλάσης, μετακλάσης και ιδιότητας), καθώς δεν επιτρέπεται ο σχηματισμός ιεραρχιών μεταξύ αντικειμένων διαφορετικού τύπου. Στην περίπτωση των ιδιοτήτων, επιβάλλεται ο περιορισμός της συμβατότητας των πεδίων ορισμού και τιμών μίας ιδιότητας και των υπερ-ιδιοτήτων της[50].

Για να μπορέσουμε να χρησιμοποιήσουμε με επιτυχία τον τελεστή υπαλληλίας, θα πρέπει οι τύποι των τελεστέων να συμμορφώνονται με τους κανόνες τύπων:

$\tau_C \times \tau_C \rightarrow [\tau_C, \tau_C]$ (7)

$\tau_P \times \tau_P \rightarrow [\tau_P, \tau_P]$ (8)

$\tau_{Mc} \times \tau_{Mc} \rightarrow [\tau_{Mc}, \tau_{Mc}]$ (9)

$\tau_{Mp} \times \tau_{Mp} \rightarrow [\tau_{Mp}, \tau_{Mp}]$ (10)

Οι κανόνες αυτοί εξασφαλίζουν ότι οι ιεραρχίες σχηματίζονται ανάμεσα σε αντικείμενα του ιδίου τύπου.

Η γενική σύνταξη του τελεστή υπαλληλίας ομοιάζει με εκείνη του τελεστή υποστασιοποίησης και συγκεκριμένα είναι:

VIEW Symbol1< Symbol2 >

{FROM RQL_path_expression}

{WHERE <filter>}

{USING NAMESPACE ns1=&URI,…}

όπου Symbol1=*Qnames*|*Variable* και Symbol2=*Qnames*|*Variable*|*Function*, όπου *Function* αντιπροσωπεύει συναρτήσεις που επιστρέφουν αντικείμενα του τύπου κλάσης, μετακλάσης και ιδιοτήτων. Για την ορθότητα της χρήσης του τελεστή και για

[50] Στην περίπτωση που το πεδίο τιμών μίας υπερ-ιδιότητας είναι τύπου τ_L (αλφαριθμητικό, ακέραιος, πραγματικός, ημερομηνία ή Boolean), η συμβατότητα των πεδίων ορισμού και τιμών κατά το σχηματισμό ιεραρχιών ιδιοτήτων ισοδυναμεί με την ταυτότητα των τύπων (δηλαδή πρέπει να είναι του ίδιου τύπου).

να μην δημιουργούνται κύκλοι στις ιεραρχίες, πρέπει να ισχύει ότι Symbol1≠Symbol2. Αυτό σημαίνει ότι στην περίπτωση που χρησιμοποιούμε ονόματα (μετα)κλάσεων και ιδιοτήτων, αυτά πρέπει να είναι διαφορετικά. Επίσης, σε όλες τις περιπτώσεις χρησιμοποίησης μεταβλητών, τα σύνολα που επιστρέφονται από την αποτίμηση των μεταβλητών πρέπει να είναι αμοιβαίως αποκλειόμενα. Η χρήση μεταβλητής στα αριστερά του τελεστή, μας επιτρέπει να δηλώσουμε ένα σύνολο από υπερκλάσεις/υπερ-ιδιότητες σε μία κλάση/ιδιότητα (πολλαπλή κληρονομικότητα), ενώ η χρήση μεταβλητής εντός του τελεστή δημιουργεί ένα σύνολο υποκλάσεων/υπο-ιδιοτήτων για την κλάση/ιδιότητα που ορίζεται στα αριστερά του τελεστή. Η χρήση μεταβλητών και στις δύο θέσεις δημιουργεί ένα δέντρο βάθους 1, το οποίο ουσιαστικά αποτελεί το καρτεσιανό γινόμενο των τιμών που αντιστοιχούν στις αποτιμήσεις των μεταβλητών.

Η λογική ερμηνεία του τελεστή υπαλληλίας είναι η προσθήκη στο παραγόμενο ιδεατό σχήμα ενός συνόλου τριάδων με κατηγόρημα τις βασικές RDF/S ιδιότητες *rdfs:subClassOf* ή *rdfs:subPropertyOf*, ανάλογα αν σχηματίζουμε ιεραρχίες (μετα)κλάσεων ή ιδιοτήτων αντίστοιχα. Κάθε αρχικό αντικείμενο (μετακλάση, κλάση, ιδιότητα) αναγνωρίζεται πλέον μέσα στην όψη από το URI του χώρου ονοματοδοσίας της όψης και το όνομα που φέρει.

Η πιο απλή περίπτωση δημιουργίας σχέσης υπαλληλίας ανάμεσα σε ιδεατές κλάσεις ορίζεται με εκφράσεις της μορφής

VIEW Person<Artist> ή VIEW Person<Class("Artist")>

όπου η δεύτερη περίπτωση αποτελεί ένα τρόπο σύνθεσης βασικών εκφράσεων όψεων (Κανόνας 7). Στην πρώτη περίπτωση, η ερμηνεία της έκφρασης είναι η τριάδα *<nsview:Artist rdfs:subClassOf nsview:Person>*, ενώ στη δεύτερη περίπτωση είναι το σύνολο των τριάδων

<nsview:Artist rdf:type rdfs:Class>
<nsview:Artist rdfs:subClassOf nsview:Person>

Στην περίπτωση που επιθυμούμε να σχηματίσουμε ιεραρχίες ιδιοτήτων, σχηματίζουμε εκφράσεις της μορφής

VIEW creates<composes>

όπου η ιδεατή ιδιότητα *nsview:composes* δηλώνεται ως υπο-ιδιότητα της ιδεατής ιδιότητας *nsview:creates* (Κανόνας 8). Η λογική ερμηνεία της έκφρασης είναι *<nsview:composes rdfs:subPropertyOf nsview:creates>* και για να εφαρμοστεί ο τελεστής θα πρέπει το *domain(composes)*$\preccurlyeq$*domain(creates)* και range*(composes)*$\preccurlyeq$ *range(creates).*

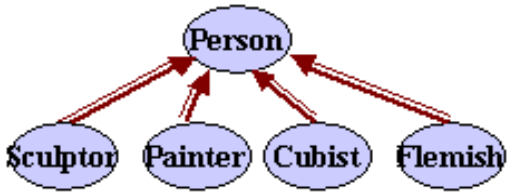
Χρησιμοποιώντας μεταβλητές, μπορούμε να σχηματίσουμε ιεραρχίες βάθους 1, είτε ορίζοντας τις υπερκλάσεις μίας ιδεατής κλάσης είτε ορίζοντας τις υποκλάσεις της. Για παράδειγμα, η έκφραση

VIEW Person<X>

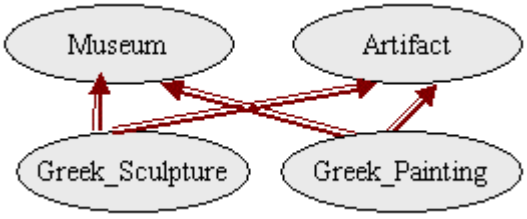
FROM subclassof(ns1:Artist){X}

USING NAMESPACE ns1=&http://www.icom.com/schema.rdf#

δημιουργεί την ιεραρχία ιδεατών κλάσεων που απεικονίζεται στην **Εικόνα 4.7**. Ο τύπος της μεταβλητής X είναι κλάση και χρησιμοποιείται ο Κανόνας 7. Επομένως, το σύνολο των κλάσεων που ανήκουν στην αποτίμηση της μεταβλητής X, δηλαδή οι κλάσεις *nsview:Sculptor, nsview:Painter, nsview:Cubist* και*nsview:Flemish* ορίζονται υποκλάσεις της ιδεατής κλάσης *nsview:Person*.



Εικόνα 4.7: Παράδειγμα χρήσης του τελεστή υπαλληλίας



Εικόνα 4.8: Παράδειγμα χρήσης του τελεστή υπαλληλίας

Στην περίπτωση που χρησιμοποιούνται μεταβλητές και στις δύο θέσεις του τελεστή, μπορούμε να σχηματίσουμε εκφράσεις της μορφής

VIEW X<Class($Y+"Greek_")>

FROM subClassOf(ns1:Artifact){Y}, bag(ns1:Museum, ns1:Artifact){X}

USING NAMESPACE ns1=&http://www.icom.com/schema.rdf#

Η παραπάνω έκφραση δημιουργεί το γράφο της **Εικόνας 4.8** (Κανόνας 7) και η λογική ερμηνεία της είναι το σύνολο των τριάδων

<nsview:Greek_Sculpture rdf:type rdfs:Class>
<nsview:Greek_Painting rdf:type rdfs:Class>

<nsview:Greek_Sculpture rdfs:subClassOf nsview:Museum>
<nsview:Greek_Painting rdfs:subClassOf nsview:Museum>
<nsview:Greek_Sculpture rdfs:subClassOf nsview:Artifact>
<nsview:Greek_Painting rdfs:subClassOf nsview:Artifact>

Στην περίπτωση που δεν ορίζεται διαφορετικά, εισάγεται και ο τύπος των αρχικών κλάσεων, που ανήκουν στην αποτίμηση της μεταβλητής X.

Μια ιδιαίτερα χρήσιμη υπολειτουργία του τελεστή υπαλληλίας τίθεται σε εφαρμογή όταν οι μεταβλητές που χρησιμοποιούνται στις δύο θέσεις του τελεστή υπαλληλίας συνδέονται με σχέσεις υπαλληλίας στο αρχικό σχήμα. Στην περίπτωση αυτή, δημιουργούνται ιδεατές ιεραρχίες μέσω της αναδόμησης αρχικών ιεραρχιών. Ο τελεστής υπαλληλίας λαμβάνει ως είσοδο μία ιεραρχία, πιθανώς επεκταμένη στην πλήρους μεταβατικής κλειστότητας (complete transitive closure) μορφή της (δηλαδή υπάρχουν όλα τα μονοπάτια από ένα κόμβο στους προγόνους του μέχρι την ρίζα της ιεραρχίας), για να παράγει μία αναδομημένη, φιλτραρισμένη, ελάχιστη ιεραρχία. Ως *ελάχιστη ιεραρχία* χαρακτηρίζουμε την ιεραρχία εκείνη που φέρει τον ελάχιστο δυνατό αριθμό μονοπατιών από κάθε κόμβο στους προγόνους του, διατηρώντας τις περιπτώσεις πολλαπλής κληρονομικότητας, όπου εμφανίζονται. Στην περίπτωση αυτή, ο όρος FROM παρέχει την είσοδο που χρειάζεται ο τελεστής, δηλαδή μία ιεραρχία υπαλληλίας (μετα)κλάσεων ή ιδιοτήτων (πιθανώς σε πλήρη μεταβατική κλειστότητα), ενώ ο όρος WHERE καθορίζει τις συνθήκες φιλτραρίσματος. Η μη χρησιμοποίηση του όρου WHERE ισοδυναμεί με την εισαγωγή της ιεραρχίας στην όψη με την μορφή που έχει στο αρχικό σχήμα (ή αν είναι επεκταμένη στην πλήρη μεταβατική κλειστότητα εισάγεται ως ελάχιστη ιεραρχία). Η διαφορά έγκειται στο ότι οι κόμβοι της είναι πλέον ιδεατές (μετα)κλάσεις ή ιδιότητες. Στην αντίθετη περίπτωση, όπου χρησιμοποιούμε τον όρο WHERE, αυτό που εισάγεται είναι μία ελάχιστη ιεραρχία που περιλαμβάνει μόνο τους κόμβους που επιτρέπονται από τα κριτήρια φιλτραρίσματος.

Δοθείσας μίας αρχικής ιεραρχίας, η λογική ερμηνεία του τελεστή υπαλληλίας στην περίπτωση αυτή ανάγεται σε ένα σύνολο τριάδων που πρέπει να αφαιρεθούν ή να προστεθούν στο σύνολο των τριάδων που αναπαριστούν την ιεραρχία. Αν με D συμβολίσουμε το σύνολο των κόμβων που αποκόπτονται συνεπεία της επιβολής των κριτηρίων φιλτραρίσματος, δηλαδή D={σύνολο αποκοπτόμενων κόμβων}, οι τριάδες που πρέπει να *αφαιρεθούν* ακολουθούν το πρότυπο

<X rdfs:subClassOf[51] superclassof^(X)>

<subClassof^(X) rdfs:subClassOf X>

ενώ το σύνολο των τριάδων που *προστίθενται* έχουν τη μορφή

<subClassof^(X) rdfs:subClassOf closest_undeleted_ancestor(X)>

όπου X∈D, superclassof^(X) συμβολίζει τις άμεσες υπερκλάσεις του X, subClassof^(X) συμβολίζει τις άμεσες υποκλάσεις του X και closest_undeleted_ancestor(X) αναπαριστά τους πιο κοντινούς ιεραρχικά προγόνους του X που δεν ανήκουν στο D. Αντίστοιχη διαδικασία γίνεται στην περίπτωση των ιδιοτήτων, όπου η ιδιότητα rdfs:subClassOf αντικαθίσταται από την ιδιότητα rdfs:subPropertyOf. Ένας περιορισμός που τίθεται για την ορθότητα του αποτελέσματος είναι ότι κατά την πρόσθεση των τριάδων πρέπει να ισχύει ότι subClassOf^(X)∉D, δηλαδή οι άμεσες υποκλάσεις του αποκοπτόμενου κόμβου δεν πρέπει να ανήκουν στο σύνολο D των αποκοπτόμενων κόμβων. Προκειμένου να προκύψει μία ελάχιστη ιεραρχία, επιβάλλεται επίσης ο περιορισμός ότι αν ισχύει «x isA y isA z», τότε δεν θα πρέπει να υπάρχει το μονοπάτι «x isA z», όπου isA αναπαριστά τη σχέση υπαλληλίας (*περιορισμός δημιουργίας ελάχιστων ιεραρχιών*).

Στην περίπτωση που έχουμε μία ιεραρχία σε πλήρη μεταβατική κλειστότητα, αρχικά αφαιρούμε μόνο το σύνολο των τριάδων που έχουν τη μορφή

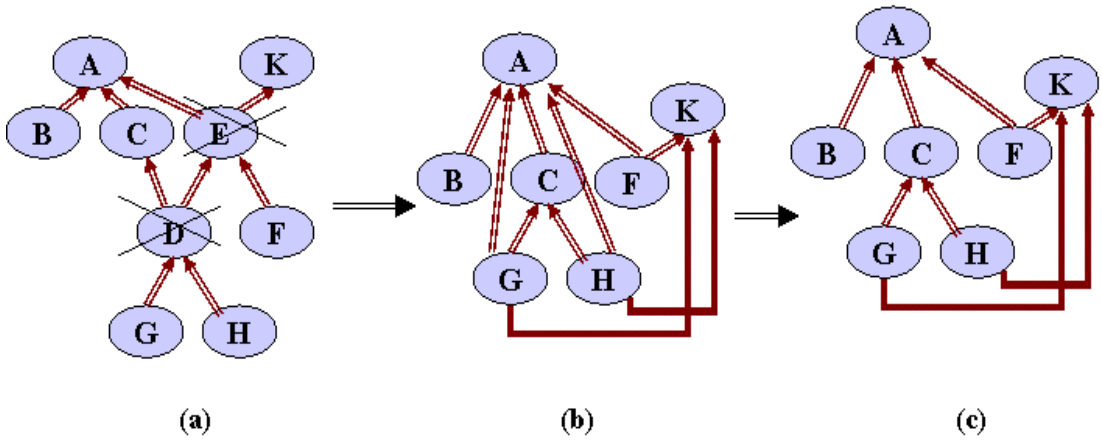
<X rdfs:subClassOf superclassof(X)>

<subClassof(X) rdfs:subClassOf X>

δηλαδή αφαιρούμε τις εισερχόμενες και εξερχόμενες rdfs:subClassOf ακμές από το σύνολο των κόμβων που αποκόπτουμε. Στο σύνολο των τριάδων που προκύπτουν εφαρμόζουμε τον περιορισμό της δημιουργίας ελάχιστων ιεραρχιών, όπως και στην περίπτωση των απλών ιεραρχιών.

Εξαιτίας της χρησιμότητας που προσφέρει ο τελεστής υπαλληλίας στην περίπτωση αρχικών ιεραρχιών, θα εξετάσουμε τη λειτουργία του χρησιμοποιώντας ένα αφηρημένο παράδειγμα. Έστω λοιπόν η ιεραρχία που απεικονίζεται στο τμήμα (a) της **Εικόνας 4.9** και ότι οι κόμβοι που ορίζονται από τον όρο WHERE προς αποκοπή είναι οι *E* και *D*.

[51] Αντίστοιχα πράττουμε και στην περίπτωση των ιεραρχιών ιδιοτήτων.



Εικόνα 4.9: Αναδρομική λειτουργία του τελεστή υπαλληλίας

Δοθέντος του συνόλου των τριάδων που αναπαριστούν την ιεραρχία, το σύνολο των isA συνδέσμων που πρέπει να αποκοπεί βάσει των κανόνων που ισχύουν για τον τελεστή αναδρομικότητας είναι το {<E isa A>, <E isa K>, <D isa G>, <D isa E>, <F isa E>, <G isa D>, <H isa D>}, ενώ το σύνολο των isA συνδέσμων που πρέπει να προστεθεί είναι το {<F isa K>, <F isa A>, <G isa C>, <H isa C>, <G isa A>, <H isa A>, <G isa K>, <H isa K>}. Ο γράφος που προκύπτει απεικονίζεται στο τμήμα (b) της Εικόνας 4.9. Η ιεραρχία αυτή δεν είναι ελάχιστη, δεδομένου ότι υπάρχουν επιπλέον μονοπάτια από έναν κόμβο στους προγόνους του, π.χ από τον κόμβο *G* στον κόμβο *A*. Εντοπίζοντας και αφαιρώντας όλες τις περιπτώσεις αυτές, η ιεραρχία που προκύπτει, και απεικονίζεται στο τμήμα (c) της εικόνας αποτελεί την έξοδο του τελεστή αναδρομικότητας, δηλαδή την ελάχιστη ιεραρχία. Η ελάχιστη αυτή ιεραρχία αποτελείται από το σύνολο των τριάδων {<F isa K>, <F isa A>, <G isa C>, <H isa C>, <G isa K>, <H isa K>} και ουσιαστικά διαμορφώνεται κατά το στάδιο της προβολής του επεκταμένου σχήματος που σχηματίζεται με τη χρήση των τελεστών υποστασιοποίησης και υπαλληλίας. Η ίδια διαδικασία ακολουθείται στην περίπτωση που έχουμε ιεραρχία σε πλήρη μεταβατική κλειστότητα. Και στις δύο περιπτώσεις (ιεραρχία και πλήρη μεταβατική κλειστότητα), ο τελεστής αναδρομικότητας μας επιτρέπει να εισάγουμε μέσα στην όψη τμήματα πληροφορίας, προερχόμενα από έναν ή περισσότερους χώρους ονοματοδοσίας, που αφορούν ιεραρχίες σε επίπεδο σχήματος και μετασχήματος. Για παράδειγμα, η έκφραση

```
VIEW $X<$Y>
FROM $X($Y}
USING NAMESPACE ns1=&http://www.icom.com/schema.rdf#
```

εισάγει στην όψη όλες τις ιεραρχίες των κλάσεων που έχουν οριστεί στο χώρο ονοματοδοσίας ns1. Οι κόμβοι της ιδεατής ιεραρχίας είναι ιδεατές κλάσεις.

4.3.3 Ιδιότητες της γλώσσας ορισμού όψεων RVL

Η προδιαγραφή μίας γλώσσας ορισμού όψεων πρέπει να βασίζεται σε ένα σύνολο κριτηρίων, τα οποία αποτελούν γνώμονα ανάπτυξης. Μπορούμε να διακρίνουμε τα εξής βασικά κριτήρια προδιαγραφής μίας γλώσσας ορισμού όψεων, που βρίσκουν εφαρμογή στην προδιαγραφή της γλώσσας ορισμού RDF/S όψεων που προτείνεται στα πλαίσια αυτής της εργασίας: *απλότητα*, *γενικότητα*, *λογική ανεξαρτησία*, *κλειστότητα* και *συνέπεια*. Οι επόμενες παράγραφοι αναφέρονται συνοπτικά στα κριτήρια αυτά.

4.3.3.1 Απλότητα

Μία γλώσσα ορισμού όψεων πρέπει να προσφέρει το ελάχιστο σύνολο βασικών κατασκευών που απαιτούνται για τη δημιουργία μίας όψης, χωρίς αυτό να συνεπάγεται μείωση της εκφραστικής της δύναμης και των υποστηριζόμενων λειτουργιών. Επιπλέον, οι ορισμοί των ιδεατών κατασκευών πρέπει να γίνονται με σαφή τρόπο και σαφή πρέπει να είναι επίσης τα αποτελέσματα μίας τέτοιας δήλωσης. Η *RVL* θεωρούμε ότι πληροί και τα δύο κριτήρια. Από τη μία πλευρά, προδιαγράφει μόνο δύο τελεστές (υποστασιοποίησης και υπαλληλίας), που της παρέχουν όμως εκφραστική δύναμη με την οποία δύναται να ανταποκριθεί στις απαιτήσεις της πλειονότητας των εφαρμογών Σημασιολογικού Ιστού. Από την άλλη πλευρά, η λογική ερμηνεία κάθε τελεστή και το σύστημα τύπων που χρησιμοποιείται καθιστά σαφή τα αποτελέσματα της εφαρμογής ενός RVL τελεστή στους απαιτούμενους τελεστέους. Όλα αυτά επιτυγχάνονται με SQL-τύπου εκφράσεις ορισμού όψεων, των οποίων η σύνταξη είναι γνώριμη από εφαρμογές παραδοσιακών βάσεων δεδομένων και η σημασιολογία είναι διαισθητική. Στα πλαίσια της απλότητας μπορούμε επίσης να κατατάξουμε το *δηλωτικό ορισμό όψεων*. Αυτό επιβεβαιώνεται από τη χρήση της γλώσσας επερώτησης RQL στον ορισμό των όψεων. Η απόδοση πληθυσμού σε (μετα)κλάσεις/ιδιότητες, ακόμα και η δημιουργία στοιχείων επιπέδου (μετα)σχήματος στηρίζεται στις επερωτήσεις που ενθυλακώνονται στις δηλώσεις ορισμού όψης.

4.3.3.2 Γενικότητα

Μία άλλη ιδιότητα που πρέπει να διαθέτει μία γλώσσα ορισμού όψεων είναι η *διατήρηση του υποκείμενου μοντέλου δεδομένων*. Η ιδιότητα αυτή σχετίζεται με την ιδιότητα της *γενικότητας*, που ορίζει ότι η γλώσσα ορισμού όψεων πρέπει να παρέχει υποστήριξη σε όλα τα στοιχεία του υποκείμενου μοντέλου. Η γλώσσα RVL δεν μεταβάλλει το υποκείμενο μοντέλο δεδομένων και το σύστημα τύπων που ακολουθείται, πληρώντας με τον τρόπο αυτό την ιδιότητα της διατήρησης του υποκείμενου μοντέλου δεδομένων. Από την άλλη πλευρά καλύπτει όλα τα στοιχεία του υποκείμενου μοντέλου, εκτός από εκείνο της υποστασιοποίησης που δεν υποστηρίζεται ούτε από το μοντέλο της χρησιμοποιούμενης γλώσσας επερώτησης. Επομένως, η RVL προδιαγράφει μόνο ένα (μικρό) σύνολο απαραίτητων τελεστών για την προσθήκη αντικειμένων στις όψεις και δεν επιφέρει αλλαγές στην εννοιολογική δομή που χρησιμοποιείται από τα αποθηκευμένα δεδομένα.

4.3.3.3 Λογική Ανεξαρτησία

Η *λογική ανεξαρτησία* είναι μία από τις σημαντικότερες ιδιότητες που οφείλει να εξασφαλίζει μία γλώσσα ορισμού όψεων, καθώς αποτελεί τον πυρήνα της φιλοσοφίας του μηχανισμού ορισμού όψεων. Αυτό που επιβάλλει η λογική ανεξαρτησία είναι ότι οι ορισμοί των όψεων απομονώνονται από εκείνους των αποθηκευμένων δεδομένων. Οι συγγραφείς του [SAD94] αναφέρονται σε ένα σύνολο περιορισμών που πρέπει να τηρούνται για την επίτευξη της ανεξαρτησίας δεδομένων και όψεων. Οι περιορισμοί αυτοί, όπως προσαρμόζονται στο RDF/S μοντέλο δεδομένων, είναι:

1. Τα πραγματικά σχήματα δεν μπορούν να χρησιμοποιούν ορισμούς από ιδεατά σχήματα. Τα ιδεατά σχήματα όμως μπορούν να στηρίζονται σε ορισμούς από κάθε πραγματικό ή ιδεατό σχήμα.
2. Οι ιδεατές κατασκευές που ορίζονται σε μία όψη υπάρχουν μόνο κατά τη διάρκεια ενεργοποίησης μίας όψης και δεν αποθηκεύονται. Αυτό ισοδυναμεί με την μη υποστήριξη υλοποιημένων όψεων (των οποίων τα περιεχόμενα αποκτούν φυσική υπόσταση).
3. Τα βασικά στιγμιότυπα μπορούν να γίνουν ιδεατά ως αποτέλεσμα της ενεργοποίησης μίας όψης. Επίσης, οι επαναχρησιμοποιούμενες κλάσεις μπορούν να επαναπροσδιοριστούν στα ιδεατά σχήματα, αλλά δεν μπορούν σε βασικά.

Επιπλέον, ένα σύστημα διαχείρισης βάσεων δεδομένων παρέχει ανεξαρτησία όψεων όταν η δήλωση και η σημασιολογία των υπαρχόντων σχημάτων όψεων δεν επηρεάζονται από τον ορισμό νέων [R92].

Οι παραπάνω ορισμοί πληρούνται από την RVL. Σύμφωνα με το τυπικό μοντέλο όψεων που χρησιμοποιούμε, ένα βασικό σχήμα σχετίζεται με μία αποθηκευμένη βάση περιγραφών. Ένα ιδεατό σχήμα, το οποίο διακρίνεται από τα αρχικά σχήματα, σχετίζεται με μία ιδεατή βάση περιγραφών. Αυτό συνεπάγεται ότι οι όψεις που δημιουργούνται δεν επηρεάζουν τα υπάρχοντα σχήματα (ιδεατά και βασικά). Επιπλέον, το γεγονός ότι οι ιδεατές κατασκευές διακρίνονται από τις αρχικές και οι ιεραρχίες μέσα σε μία όψη αναφέρονται σε ιδεατές κατασκευές είναι ένας ακόμα λόγος που πιστοποιεί την ανεξαρτησία όψεων που επιτυγχάνει η RVL.

4.3.3.4 Κλειστότητα

Η κλειστότητα αποτελεί μία έννοια την οποία αναφέραμε σε προηγούμενο κεφάλαιο, αναφερόμενοι στο οντοκεντρικό μοντέλο δεδομένων. Ως *κλειστότητα* (closure) ενός ιδεατού σχήματος αναφορικά με το δίκτυο των ιδιοτήτων και τις ιεραρχίες υπαλληλίας χαρακτηρίζουμε την ιδιότητα που ορίζει ότι κάθε ιδιότητα ή σχέση υπαλληλίας που χρησιμοποιείται από μία ιδεατή (μετα)κλάση ή ιδιότητα πρέπει να ορίζεται στο ιδεατό σχήμα. Στην περίπτωση που ένα σχήμα δεν είναι κλειστό, εισάγεται στην όψη το ελάχιστο σύνολο πληροφορίας που είναι απαραίτητο για την ικανοποίηση αυτής της ιδιότητας. Η ιδιότητα της κλειστότητας μας εξασφαλίζει ότι οτιδήποτε χρειάζεται μία όψη για να είναι συνεπής, περιέχεται μέσα στην όψη.

Στο μοντέλο δεδομένων της RVL, ο τρόπος δήλωσης και κατασκευής ιδεατών σχημάτων μας εξασφαλίζει ότι οι όψεις που σχηματίζονται είναι κλειστές. Το γεγονός αυτό αποδίδεται σε δύο λόγους. Κατά πρώτον, ο περιορισμός της δήλωσης του πεδίου ορισμών και τιμών μίας ιδιότητας κατά τον ορισμό της, μας εξασφαλίζει ότι το δίκτυο των ιδιοτήτων που σχηματίζεται σε ένα ιδεατό σχήμα είναι πλήρες. Αυτό σημαίνει ότι κάθε ιδιότητα συνοδεύεται από το πεδίο ορισμού και τιμών της, που είτε δηλώνονται ρητά στην όψη είτε επαναχρησιμοποιούνται από ένα αρχικό σχήμα. Κατά δεύτερον, οι ιεραρχίες που σχηματίζονται σε μία όψη σχηματίζονται ανάμεσα σε ιδεατές (μετα)κλάσεις ή ιδιότητες που ορίζονται στην όψη (άρα οτιδήποτε χρειάζεται για το σχηματισμό τους ορίζεται μέσα στην όψη). Συνεπώς, η RVL γλώσσα

είναι μία γλώσσα ορισμού ιδεατών σχημάτων, που πληροί την ιδιότητα της κλειστότητας.

ΚΕΦΑΛΑΙΟ 5

Επίλογος

Η πρόκληση για την παροχή ενός μηχανισμού ορισμού όψεων στο περιβάλλον του Σημασιολογικού Ιστού είναι η υποστήριξη των απαιτούμενων από τις εφαρμογές λειτουργικοτήτων μέσω μίας γλώσσας ορισμού όψεων, η οποία συνδυάζει την εκφραστικότητα και την ευελιξία με την απλότητα στον ορισμό όψεων. Το σύνολο των λειτουργικοτήτων αυτών, όπως εκφράζεται στο μοντέλο δεδομένων του RDF/S, ανάγεται στη δημιουργία και απόδοση πληθυσμού σε ιδεατές κατασκευές (μετακλάσεις κλάσεων και ιδιοτήτων, κλάσεις και ιδιότητες), καθώς και στη δημιουργία ιδεατών ιεραρχιών. Η γλώσσα ορισμού RDF/S όψεων που προδιαγράψαμε (RVL) και η οποία αποτελεί το αντικείμενο της συγκεκριμένης μεταπτυχιακής εργασίας, είναι σε θέση να αντιμετωπίσει αυτή την πρόκληση και να υποστηρίξει εφαρμογές πραγματικής κλίμακας στο Σημασιολογικό Ιστό. Ενδεικτικά αναφέρουμε περιοχές εφαρμογών, όπως η *εξατομίκευση πληροφορίας* [IBM00, STC02], η *ολοκλήρωση πληροφορίας*[52] και η *διαχείριση οντολογιών* [KF01]. Οι περιοχές αυτές είναι ενδεικτικές και δεν περιορίζουν σε κανένα βαθμό το εύρος των εφαρμογών των όψεων, καθώς περιοχές εφαρμογών (π.χ προστασία δεδομένων) που ισχύουν σε σχεσιακά, οντοκεντρικά, ημιδομημένα και XML περιβάλλοντα βάσεων δεδομένων εξακολουθούν να βρίσκουν εφαρμογή στο Σημασιολογικό Ιστό.

Λαμβάνοντας υπόψη τις ιδιαιτερότητες του RDF/S μοντέλου δεδομένων, ο προτεινόμενος μηχανισμός ορισμού όψεων ενσωματώνει λειτουργικότητα που υποστηρίζεται σε παραδοσιακά συστήματα βάσεων δεδομένων, αλλά και σε σύγχρονες εφαρμογές, για να παράσχει περιεχόμενο προσαρμοσμένο στις απαιτήσεις των επιμέρους χρηστών. Η προδιαγραφή της γλώσσας ορισμού όψεων RVL βασίστηκε σε ένα τυπικό μοντέλο όψεων, το οποίο αντιμετωπίζει όλα τα θέματα που άπτονται του ορισμού τους. Βασισμένη στο σύστημα τύπων της RQL

---

[52] Η ολοκλήρωση είναι μία έννοια που εμφανίζεται σε πολλά συμφραζόμενα: *ολοκλήρωση συστημάτων* [JR99], *οντολογιών* [CGL01, S00a, K01, SM01a, SM01b, GPS98, RB01, AF99, WVV+01], *καταλόγων προϊόντων* [OF01a, OF01b, OF01c, SHF+00], *δεδομένων προϊόντων* [FDS+01, ABM01], *γλωσσών οντολογιών* [SE01] κτλ...

γλώσσας επερώτησης RDF/S γράφων, η RVL εκμεταλλεύεται όλη την εκφραστική δύναμη της RQL για να δημιουργήσει κατασκευές σε όλα τα επίπεδα αφαίρεσης μίας RDF/S εφαρμογής (επίπεδο δεδομένων, σχήματος και μετασχήματος). Η ικανότητά της αυτή να διακρίνει τα επίπεδα αφαίρεσης μίας RDF/S εφαρμογής, σε συνδυασμό με το υιοθετούμενο σύστημα τύπων, της δίνει τη δυνατότητα να επιτελεί το σύνολο των ζητουμένων λειτουργικοτήτων με τη χρήση μόνο δύο τελεστών: του *τελεστή υποστασιοποίησης* και του *τελεστή υπαλληλίας*. Το γεγονός αυτό δεν μειώνει την εκφραστική δύναμη της RVL. Αντιθέτως, η απλότητα στον ορισμό των απαιτούμενων ιδεατών κατασκευών αποτελεί ένα από τα μεγαλύτερα πλεονεκτήματά της.

Στο σημείο αυτό αξίζει να τονίσουμε ότι σε αντίθεση με το σχεσιακό μοντέλο δεδομένων, όπου όλη η πληροφορία αναφέρεται στον ίδιο χώρο ονοματοδοσίας, στο RDF/S μοντέλο δεδομένων οι χώροι ονοματοδοσίας είναι πολλοί και πιθανότατα εκφράζουν διαφορετικές θεωρήσεις για το ίδιο σύνολο πληροφορίας. Σε ένα περιβάλλον διαχείρισης γνώσης (knowledge management), το γεγονός αυτό ισοδυναμεί με την ύπαρξη ορθόγωνων απόψεων επί του ίδιου συνόλου δεδομένων. Σύμφωνα με τη λογική αυτή, μία όψη δεν θεωρείται απλώς ένα κομμάτι της αποθηκευμένης πληροφορίας, αλλά μια (νέα) θεώρηση πάνω σε αυτή, που πιθανώς να διαφέρει εξ ολοκλήρου από την θεώρηση που εκφράζει το σχήμα που χρησιμοποιείται για την αποθήκευση της πληροφορίας. Με άλλα λόγια, μία όψη στα συμφραζόμενα του RDF/S μπορεί να χρησιμοποιηθεί για την μοντελοποίηση της αποθηκευμένης πληροφορίας με τρόπο εντελώς διαφορετικό από τη μοντελοποίηση που χρησιμοποιήθηκε για την αποθήκευσή της, ισοδυναμώντας με τη δυνατότητα *ισχυρής αναδόμησης* (heavy-restructuring) των δεδομένων.

Η ικανότητα ισχυρής αναδόμησης εξασφαλίζεται από την εκφραστικότητα της γλώσσας επερώτησης RQL (που δύναται να επερωτήσει με ευελιξία δεδομένα σε όλα τα σημασιολογικά επίπεδα) και δικαιολογείται από τις ανάγκες των εφαρμογών στο Σημασιολογικό Ιστό, όπου το εύρος των πραγμάτων που χρειάζεται να ολοκληρωθεί με εργαλείο το μηχανισμό των όψεων είναι πολύ μεγάλο και οι ετερογένειες που υπάρχουν εντοπίζονται τόσο σε *συντακτικό* επίπεδο όσο και σε *δομικό* και *σημασιολογικό* [SHF+00]. Ενδεικτικά, σε *επίπεδο γλώσσας ή μεταμοντέλου* ενδέχεται να χρησιμοποιείται διαφορετική *σύνταξη* ή *λογική αναπαράσταση* και η *σημασιολογία* των δομών της γλώσσας αναπαράστασης γνώσης να είναι διαφορετική. Επίσης, η *εκφραστική δύναμη* των γλωσσών μπορεί να διαφέρει, καθώς ορισμένες γλώσσες ορισμού οντολογιών (σχημάτων) μπορούν να εκφράσουν σημασιολογικά πλουσιότερες έννοιες από άλλες. Ταυτόχρονα *σε*

επίπεδο οντολογίας ή μοντέλου παρουσιάζονται διαφορές στον τρόπο που μοντελοποιείται το πεδίο εφαρμογής (εννοιολογικές, επίπεδο λεπτομέρειας στην ανάπτυξη, διαφορές επεξήγησης ή ορολογικές διαφοροποιήσεις κτλ.) [K01, DS02]. Επομένως, για να εξισορροπηθεί η αυτονομία των κοινοτήτων χρηστών ως προς την ανάπτυξη οντολογιών και εννοιολογικών σχημάτων με την ανάγκη για διαλειτουργικότητα και αποτελεσματικό διαμοιρασμό πληροφορίας, η γλώσσα ορισμού όψεων πρέπει να είναι σε θέση να υποστηρίζει το χαρακτηρισμό των σχέσεων μεταξύ διαφορετικών πληροφοριακών πηγών. Τη δυνατότητα αυτή για ισχυρή αναδόμηση θεωρούμε ότι διαθέτει η προτεινόμενη γλώσσα ορισμού όψεων RVL.

5.1 Μελλοντική εργασία

Μία από τις άμεσες προτεραιότητές μας αποτελεί η *υλοποίηση* του προτεινόμενου μηχανισμού ορισμού όψεων. Εκτός από την τεχνική πλατφόρμα που απαιτείται για το σκοπό αυτό, απαραίτητη προϋπόθεση είναι η προδιαγραφή της αρχιτεκτονικής του συστήματος, όπου πρέπει να αντιμετωπιστούν ζητήματα, όπως ο καθορισμός δικαιωμάτων ορισμού όψεων, ο χειρισμός/αποθήκευση των δηλώσεων όψεων και η αποτελεσματική υλοποίηση των μηχανισμών εξασφάλισης των περιορισμών του υποκείμενου μοντέλου. Στα πλαίσια αυτά, ενδιαφέρον είναι το πρόβλημα της υποστήριξης *γραφικών διεπαφών χρήσης για του ορισμό όψεων*. Με τον τρόπο αυτό, ο μηχανισμός ορισμού όψεων από προγραμματιστικό εργαλείο μετατρέπεται σε εφαρμογή τελικού χρήστη και οι ιδιαιτερότητες της γλώσσας ορισμού όψεων γίνονται διαφανείς στον τελικό αποδέκτη της λειτουργικότητάς της.

Οι όψεις που υποστηρίζει η γλώσσα RVL ισοδυναμούν με *ιδεατές όψεις*, καθώς η ύπαρξή τους εξαρτάται από την ενεργοποίηση του ορισμού τους. Μία ενδιαφέρουσα πτυχή του μηχανισμού ορισμού όψεων, που αποτελεί ταυτόχρονα έναυσμα για μελλοντική εργασία, είναι η μελέτη της υποστήριξης *υλοποιημένων όψεων* από την RVL. Αν και ο τρόπος με τον οποίο δημιουργούνται οι RVL όψεις εξασφαλίζει τη λογική ερμηνεία των δεδομένων, υπάρχουν ζητήματα που άπτονται της υλοποίησης και της αποθήκευσης των ιδεατών όψεων στο φυσικό μέσο αποθήκευσης, τα οποία χρήζουν αντιμετώπισης. Επίσης, άμεσα συσχετιζόμενο θέμα έρευνας είναι η *ενημέρωση* και *συντήρηση* των όψεων. Το πρόβλημα των ενημερώσεων όψεων και των ιδεατών αντικειμένων ενσωματώνεται στην προβληματική της ενημέρωσης δεδομένων πάνω στα οποία επιβάλλουμε ένα

σύνολο από περιορισμούς ακεραιότητας. Η μετάφραση των ενημερώσεων σε μία όψη (ή βάση δεδομένων) σε ενημερώσεις στη βάση δεδομένων (ή όψη) έχει απασχολήσει την επιστημονική κοινότητα των βάσεων δεδομένων δεκαετίες, καθώς οι παράμετροι ενός τέτοιου προβλήματος είναι πολύπλοκες. Στα πλαίσια του RDF/S, όπου το μοντέλο δεδομένων παρουσιάζει ιδιαιτερότητες, το πρόβλημα αυτό καθίσταται ιδιαίτερα ενδιαφέρον και ταυτόχρονα περίπλοκο, γεγονός που δικαιολογεί την περαιτέρω μελέτη.

Συν τοις άλλοις, σε παραδοσιακά συστήματα βάσεων δεδομένων, ένα ζήτημα που έχει απασχολήσει κατά κόρον τους επιστήμονες που ασχολούνται με όψεις είναι η *βελτιστοποίηση επερωτήσεων με τη χρήση όψεων*. Ιδιαίτερα σε ένα περιβάλλον, όπως ο Σημασιολογικός Ιστός, όπου η μεσιτεία πληροφορίας είναι μείζονος σημασίας, η χρήση των όψεων στην απάντηση και βελτιστοποίηση επερωτήσεων λαμβάνει ιδιαίτερη σημασία. Τέλος, σημαντικό θέμα προς μελλοντική διερεύνηση αποτελεί ο *ορισμός δικαιωμάτων πρόσβασης πάνω σε όψεις*, ο οποίος ισοδυναμεί με τον καθορισμό των χρηστών που έχουν δικαίωμα ορισμού και χρήσης όψεων.

Τα παραπάνω ζητήματα μελλοντικής έρευνας είναι μόνο ενδεικτικά του πλούσιου πεδίου επιστημονικής ενασχόλησης που προσφέρουν οι όψεις. Εν μέρει μάλιστα δικαιολογούν και την επιθυμία μας να ασχοληθούμε με ένα μηχανισμό ορισμού όψεων σε ένα νέο περιβάλλον διακίνησης και αποθήκευσης πληροφορίας, που θέτει νέες απαιτήσεις και προκλήσεις.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[A99] Serge Abiteboul. "On Views and XML". In Proceedings of the ACM Symposium on Principles of Database Systems, Philadelphia, Pennsylvania. (pp. 1-9). May 31 - June 2, 1999.

[AAC+99] Serge Abiteboul, Bernd Amann, Sophie Cluet, Anat Eyal, Laurent Mignet, Tova Milo. "Active Views for Electronic Commerce". In Proceedings of the 25th International Conference on Very Large Data Bases, Edinburgh, Scotland, United Kingdom. September 7-10, 1999.

[ABM01] Serge Abiteboul, Omar Benjelloun, Tova Milo. "*Towards a Flexible Model for Data and Web Services* Integration". In Proceedings of the International Workshop on Foundations of Models and Languages for Data and Objects, Italy. September 16-18, 2001.

[AB91] Serge Abiteboul, Anthony Bonner. "*Object and Views*". In Proceedings of the ACM SIGMOD International Conference on Management of Data, Denver, Colorado (pp. 238-247). May, 1991.

[AGM+97] Serge Abiteboul, Roy Goldman, Jason McHugh, Vassilis Vassalos, Yue Zhuge. "*Views for Semistructured Data*". In Proceedings of the Workshop on Management of Semistructured Data, Tucson, Arizona. May, 1997.

[AHV95] S. Abiteboul, R. Hull, V. Vianu. "*Foundations of Databases*". Addison Wesley. 1995.

[AMR+98] Serge Abiteboul, Jason McHugh, Michael Rys, Vassilis Vassalos, Janet L.Wiener. "*Incremental Maintenance of Materialized Views Over Semistructured Data*". In Proceedings of the 24th International Conference on Very Large Databases, New York. August, 1998.

[AQM+97] S. Abiteboul, D. Quass, J. McHugh, J. Widom, J. Wiener. "*The Lorel query language for semistructured data*". International Journal on Digital Libraries, 1(1):68-88. April, 1997.

[ASB99] Serge Abiteboul, Dan Suciu, Peter Buneman. "*Data on the Web: From Relations to Semistructured Data and Xml*". Morgan Kaufmann Series in Data Management Systems. 1999.

[ACM+02] Vincent Aguilera, Sophie Cluet, T. Milo, Pierangelo Veltri, Dan Vodislav. "*Views in a Large Scale XML Repository*". In Proceedings of the 28th VLDB Conference, Hong Kong, China. August 20-23, 2002.

[ACK+01] S. Alexaki, V. Christophides, G. Karvounarakis, D. Plexousakis, K. Tolle. "*The ICS-FORTH RDFSuite: Managing Voluminous RDF Description Bases*". In Proceedings of the 2nd International Workshop on the Semantic Web (SemWeb'01), in conjunction with Tenth International World Wide Web Conference (WWW10), Hongkong (pp. 1-13). May 1, 2001.

[AF99] Bernd Amann, Irini Foundulaki. "*Integrating Ontologies and Thesauri to Build RDF Schemas*". In Proceedings of the Third European Conference for Digital Libraries ECDL'99, Bibliotheque Nationale de France, Paris, France. September 22-24, 1999.

[A75] ANSI/X3/SPARC Study Group on Database Management Systems. Interim Report. ACM SIGMOD Bulletin 7, N2. 1975.

[BS81a] F. Bancilhon, N. Spyratos. "*Independent Components of Databases*". In Proceedings of the 7th International Conference on Very Large Data Bases, Cannes, France (pp. 398-408). September 9-11, 1981.

[BS81b] F. Bancilhon, N. Spyratos. "*Update Semantics of Relational Views*". In Proceedings of the ACM Transactions on Database Systems (TODS), V.6 N.4, (pp. 557-575). December, 1981.

[B99] Chaitanya Baru. "*XViews: XML Views of Relational Schemas*". In Proceedings of the 10th International Workshop on Database & Expert Systems Applications, Florence, Italy (pp. 700-705). September 1-3, 1999.

[BLP+98] C. Baru, Bertram Ludäscher, Yannis Papakonstantinou, Pavel Velikhov, Victor Vianu. "*Features and Requirements for an XML View Definition Language: Lessons from XML Information Mediation*". In Proceedings of the Query Languages Workshop (QL '98), Boston, Massachussets, USA. December 3-4, 1998.

[BFM98] T. Berners-Lee, R. Fielding, L. Masinter. "*Uniform Resource Identifiers (URI): Generic Syntax*". RFC 2396. August 1998.
Available at <http://www.ietf.org/rfc/rfc2396.txt>

[BHL01] Tim Berners-Lee, James Hendler, Ora Lassila. "*The Semantic Web*". Scientific American. May 2001.
Available at <http://www.sciam.com/2001/0501issue/0501berners-lee.html>

[BCF+02] Scott Boag, Don Chamberlin, Mary F. Fernandez, Daniela Florescu, Jonathan Robie, Jérôme Siméon. "*XQuery 1.0: An XML Query Language*". W3C Working Draft. August 16, 2002.
Available at <http://www.w3.org/TR/xquery/>

[BHL99] Tim Bray, Dave Hollander, Andrew Layman. "*Namespaces in XML*". W3C Recommendation. January 14, 1999.

[BPS+00] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler. "*Extensible Markup Language (XML) 1.0 (Second Edition)*". W3C Recommendation. October 6, 2000.

[BG00] D. Brickley, R.V. Guha. "*Resource Description Framework Schema (RDF/S) Specification 1.0*". W3C Candidate Recommendation. March 27, 2000.
Available at <http://www.w3.org/TR/2000/CR-rdf-schema-20000327/>

[BG02] Dan Brickley, R.V. Guha. "*RDF Vocabulary Description Language 1.0: RDF Schema*". W3C Working Draft. April 30, 2002.
Available at <http://www.w3.org/TR/rdf-schema/>

[BFR+01] Allen Brown, Matthew Fuchs, Jonathan Robie, Philip Wadler. "*XML Schema: Formal Description*". W3C Working Draft. September 25, 2001.
Available at <http://www.w3.org/TR/xmlschema-formal/>

[CGL01] Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini. "*Ontology of Integration and Integration of Ontologies*". In Proceedings of the 2001 Description Logic Workshop (DL'01), Stanford University, California, USA (pp. 10-19). August 1-3, 2001.

[C88] Cardelli, L. "*A Semantics of Multiple Inheritance*". Information and Computation, 76(2/3):138-164. 1988.

[CBC+00] R. G. G. Cattell (Editor), Douglas K. Barry (Editor), Rick Catell (Editor), Mark Berler (Editor), Jeff Eastman (Editor), David Jordan (Editor), Craig Russell, Olaf Schadow, Torsten Stanienda, Fernando Velez. "*The Object Data Standard: ODMG 3.0*". Morgan Kaufmann Publishers. January, 2000.

[CJB99] B. Chandrasekaran, J. R. Josephson and V. R. Benjamins. "*What Are Ontologies, and Why Do We Need Them*?". IEEE Intelligent Systems, Vol. 14, No. 1. January/February, 1999.

[CZ01] Akmal B. Chaudhri, Roberto Zicari. "Succeeding with Object Databases: A Practical Look at Today's Implementations with Java and XML". Wiley Computer Publishing. 2001.

[C01] Vassilis Christophides. "*From Relational to Object Database Management Systems*".Course Notes, Advanced Issues of Databases, University of Crete, Greece. December 2001.
Available at <http://www.csd.uch.gr/~hy562/Handouts.htm>

[CCS00] Vassilis Christophides, Sophie Cluet, Jérôme Siméon. "*On Wrapping Query Languages and Efficient XML Integration*". In Proceedings of the 2000 ACM SIGMOD Conference on Management of Data, Dallas, Texas (pp. 141-152). ACM Press. May, 2000.

[CPM+02] Ontoweb: Ontology-based Information Integration Exchange for Knowledge Management and Electronic Commerce (IST-2000-29243). "*A Survey on Ontology Tools*". *Chapter 6: "Ontology Storage and Querying Tools*", Editor: Vassilis Christophides. Contributors: Dimitris Plexousakis, Aimilia Magkanaraki, Ta Tuan Anh, Grigoris Karvounarakis. 2002.

[CD99] James Clark, Steve DeRose. "*XML Path Language (XPath)*". W3C Recommendation, Version 1.0. November 16, 1999.
Available at <http://www.w3.org/TR/xpath>

[C98] Sophie Cluet. "*Designing OQL: Allowing Objects to be Queried*". Information Systems 23(5): 279-305. 1998.

[CDS+88] Sophie Cluet, Claude Delobel, Jerome Simeon, Katarzyna Smaga. "*Your Mediators Need Data Conversion!*". In Proceedings of ACM-SIGMOD International Conference on Management of Data, Seattle (pp. 177-188). 1988.

[CHH+01] Dan Connolly, Frank van Harmelen, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, Lynn Andrea Stein. "*DAML+OIL (March 2001) Reference Description*". W3C Note. December 18, 2001.
Available at <http://www.w3.org/TR/daml+oil-reference>

[D95a] C. J. Date. "*Εισαγωγή στα Συστήματα Βάσεων Δεδομένων*" (Ελληνική Έκδοση). Τόμος Α΄. Έκτη Αμερικανική Έκδοση. Εκδόσεις Κλειδάριθμος. 1995.

[D95b] C. J. Date. "*Εισαγωγή στα Συστήματα Βάσεων Δεδομένων*" (Ελληνική Έκδοση). Τόμος Β΄. Έκτη Αμερικανική Έκδοση. Εκδόσεις Κλειδάριθμος. 1995.

[DCH02] Mike Dean, Dan Connolly, Frank van Harmelen, James Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, Lynn Andrea

Stein. "*OWL Web Ontology Language 1.0 Reference*". W3C Working Draft. July 29, 2002.
Available at <http://www.w3.org/TR/2002/WD-owl-ref-20020729/>

[DS02] David Durand, Paul Saton. "*Semantic Heterogeneity Among Document Encoding Schemes*". Final Report for NIST Federal Assistance Contract 60NANB0D0115. January 15, 2002.
Available at <http://www.stg.brown.edu/projects/semantic/semantic_stg.pdf>

[FBD+02] D. Fensel, C. Bussler, Y. Ding, V. Kartseva, M. Klein, M. Korotkiy, B. Omelayenko, R. Siebes. "*Semantic Web Application Areas*". In Proceedings of the 7th International Workshop on Applications of Natural Language to Information Systems, Stockholm, Sweden. June 27-28, 2002.

[FDS+01] Dieter Fensel, Y. Ding, E. Schulten, B. Omelayenko, G. Botquin, M. Brown, A. Flett. "*Product Data Integration in B2B E-Commerce*". IEEE Intelligent System, 16(4). 2001.

[FFL+97] M. Fernandez, D. Florescu, A. Levy, D. Suciu. "*A query language and processor for a web site management system*". In Proceedings of the Workshop on Management of Semistructured Data, Tucson, Arizona. May, 1997.

[FSW99] Mary Fernandez, Jérôme Siméon, Philip Wadler (Editors). "*XML Query Languages: Experiences and Exemplars*". 1999.
Available at <http://www.w3.org/1999/09/ql/docs/xquery.html>

[GPS98] A. Gangemi, D. Pisanelli, G. Steve. "*Ontology Integration: Experiences with Medical Terminologies*". In: N. Guarino (ed.): Formal Ontology in Information Systems (pp. 163-178). IOS Press. 1998.

[G98] Nicola Guarino. "*Formal Ontology in Information Systems*". In Proceedings of FOIS'98, Trento, Italy. Amsterdam, IOS Press (pp. 3-15). June 6-8, 1998.

[GBC+97] Giovanna Guerrini, Elisa Bertino, Barbara Catania, Jesus Garcia-Molina. "*A formal model of views for object-oriented database systems*". Theory and Practice of Object Systems, 3(3): 157-183. 1997.

[GM99] Ashish Gupta, Inderpal Singh Mumick (Editors). "*Materialized Views: Techniques, Implementations and Applications*". Cambridge, MA: The MIT Press. 1999.

[H02] Patrick Hayes. "*RDF Semantics*". W3C Working Draft. November 12, 2002.
Available at: <http://www.w3.org/TR/rdf-mt/>

[HKR02] Hong Hu, Diane K. Kramer, Elke Rundensteiner. "*XEM: XML Evolution Management*". Technical Report WPI-CS-TR-02-09, Worcester Polytechnic Institute, Dept. of Computer Science. 2002.
Available at <http://davis.wpi.edu/dsrg/WEB_DB/PAPERS/XEM-Journal-02/xem-tr.pdf>

[IBM00] IBM High-Volume Web site team. "*Web site personalization*". January 2000. Available at
<http://www7b.boulder.ibm.com/wsdd/library/techarticles/hvws/personalize.html>

[JR99] Vanja Josifovski, Tore Risch. "*Integrating Heterogeneous Overlapping Databases Through Object-Oriented Transformations*". In Proceedings of the 25th International Conference on Very Large Data Bases, Edinburgh, Scotland, United Kingdom. September 7-10, 1999.

[K00] Grigoris Karvounarakis. "*Querying RDF metadata for Community Web Portals*". Master's Thesis, University of Crete. 2000.

[KAC+02] G. Karvounarakis, S. Alexaki, V. Christophides, D. Plexousakis, Michel Scholl. "*RQL: A Declarative Query Language for RDF*". In Proceedings of the Eleventh International World Wide Web Conference (WWW'02), Honolulu, Hawaii, USA. May 7-11, 2002.

[KC02] Grigoris Karvounarakis, Vassilis Christophides. "*The RQL v1.5 User Manual*". 2002.
Available at <http://139.91.183.30:9090/RDF/RQL/Manual.html>

[K01] Michel Klein. "*Combining and Relating Ontologies: An Analysis of Problems and Solutions*". In Proceedings of the IJCAI'01 Workshop on Ontologies and Information Sharing, Seattle, USA. August 4-5, 2001.

[KF01] Michel Klein, Dieter Fensel. "*Ontology Versioning for the Semantic Web*". In Proceedings of the International Semantic Web Working Symposium (SWWS), Stanford University, California, USA (pp. 75-91). July 30 – August 1, 2001.

[KR02] Andreas Koeller, Elke A. Rundensteiner. "*Incremental Maintenance of Schema-Restructuring Views*". C.S. Jensen et al. (Eds): EDBT 2002, LNCS 2287, pp. 354-367, 2002. Springer-Verlag Berlin Heidelberg 2002

[Ko01] Stefan Kokkelink. "*Transforming RDF with RDFPath*". Working Draft. March 2001. Available at
<http://zoe.mathematik.uni-osnabrueck.de/QAT/Transform/RDFTransform.pdf>

[LDB97a] Zoe Lacroix, Claude Delobel, Philip Breche. "*Object Views and Database Restructuring*". In Proceedings of 6th International Workshop on Database Programming Languages (DBPL97), Estes Park, Colorado, USA. LNCS 1369 (pp 180-201). August, 1997.

[LDB97b] Zoe Lacroix, Claude Delobel, Philippe Breche. "*Object Views Constructed with an Object Algebra*". Actes des 13 iemes Journées Bases de Données Avancées (BDA' 97) Conference, France. September, 1997.

[LDB98] Zoe Lacroix, Claude Delobel, Philippe Breche. "*Object Views*". Networking and Information Systems Journal, Volume 1 (2-3), Databases and Information Systems, Hermès, Paris. 1998

[LS99] O. Lassila, R. Swick. "*Resource Description Framework (RDF) Model and Syntax Specification*". W3C Recommendation. February 1999.
Available at <http://www.w3.org/TR/REC-rdf-syntax/>

[MAC+02] A. Magkanaraki, S. Alexaki, V. Christophides, and Dimitris Plexousakis. "*Benchmarking RDF Schemas for the Semantic Web*". In Proceedings of the First International Semantic Web Conference (ISWC'02), Sardinia, Italy. June 9-12, 2002.

[MKC+02] A. Magkanaraki, G. Karvounarakis, V. Christophides, D. Plexousakis, Ta Tuan Ahn. "*Ontology Storage and Querying*". ICS-FORTH Technical Report No 308. April 2002.
Available at <http://139.91.183.30:9090/RDF/publications/tr308.pdf>

[MM00] Maloney, M., Malhotra, A. "*XML Schema part 2: Datatypes*". W3C Candidate Recommendation, 2000.

[MSR02] Libby Miller, Andy Seaborne, Alberto Reggiori. "*Three Implementations of SquishQL, a Simple RDF Query Language*". In Proceedings of the 1st International Semantic Web Conference (ISWC2002), Sardinia, Italy (pp. 423-435). June 9-12, 2002.

[M99] Paul Mulholland. "*Introduction to Ontologies*". Internal Report, RichODL-OU-3/1999. September 17, 1999.

[MBJ+90] J. Mylopoulos, A. Borgida, M. Jarke, M. Koubarakis. "*Telos - a language for representing knowledge about information systems*". ACM Transactions on Information Systems, 8(4):325-362. 1990.

[Og02] Uche Ogbuji. "*Versa by example*". 2002.
Available at <http://www.xmlhack.com/read.php?item=1560>

[OF01a] Borys Omelayenko, Dieter Fensel. "*Scalable Document Integration for B2B Electronic Commerce*". Special Issue of Electronic Commerce Research Journal on B2B Research. 2001.

[OF01b] Borys Omelayenko, Dieter Fensel. "*An Analysis of Integration Problems of XML-based Catalogs for B2B E-Commerce*". In Proceedings of the 9th IFIP 2.6 Working Conference on Database Semantics (DS-9), Hong-Kong (pp. 232-246). April 25-28, 2001.

[OF01c] Borys Omelayenko, Dieter Fensel. "*A two-layered Integration Approach for Product Information in B2B E-Commerce*". In: K. Bauknecht, S.-K. Madria, G. Pernul (eds.), Electronic Commerce and Web Technologies, Proceedings of the Second International Conference on Electronic Commerce and Web Technologies (EC WEB-2001), LNCS 2115, Munich, Germany (pp. 226-239). September 4-6, 2001.

[RB01] Erhard Rahm, Philip A. Bernstein. "*On Matching Schemas Automatically*". Technical Report, MSR-TR-2001-17. February, 2001.
Available at <http://dol.uni-leipzig.de/pub/2001-5>

[RG00] Raghu Ramakrishnan, Johannes Gehrke. "*Database Management Systems".* McGraw-Hill Science/Engineering/Math. 2nd Edition. June 2000.

[RDFQL] RDFQL Reference Manual.
Available <http://www.intellidimension.com/RDFGateway/Docs/querying.asp>

[RDQL] RDF Data Query Language.
Available at <http://www.hpl.hp.com/semweb/rdql.html/>

[RQL] RQL On-Line Demo.
Available at <http://139.91.184.28:8080/RQLdemo/>

[R92] Elke A. Rundensteiner. "*MultiView: A Methodology for Supporting Multiple View Schemata in Object-Oriented Databases*". In Proceedings of the 18th International Conference on Very Large Data Bases (VLDB'92), Vancouver, Canada (pp. 187 – 198). August, 1992.

[S95] Casio Souza dos Santos. "*Un Mécanisme de Vues pour les syst*èmes *de Gestion de Bases de Donneés Objet*". PhD thesis, Université de Paris Sud-Centre d' Orsay, Paris, France. November, 1995.

[SAD94] Cassio Souza dos Santos, Serge Abiteboul, Claude Delobel. "*Virtual Schemas and Bases*". In: M.Jarke, J. Bubenko and K. Jeffery (editors): Proceedings of the Fourth International Conference on Extending Database

Technology, St John's College, Cambridge, UK. Number 779 in Lecture Notes in Computer Science (pp. 81-94). 1994.

[SLT91] Marc H. Scholl, Christian Laasch, Markus Tresch. "*Updatable Views in Object-Oriented Databases*". In Proceedings of the 2nd Deductive and Object-Oriented Databases Conference (DOOD), Munich (pp 189-207). December 16-18, 1991.

[SD01] Michael Sintek, Stefan Decker. "*TRIPLE-An RDF Query, Inference, and Transformation Language*". In Proceedings of the Deductive Databases and Knowledge Management Workshop (DDLP' 2001), Japan. October, 2001

[S80] Nicolas Spyratos. "*Translation Structures of Relational Views*". In Proceedings of the 6th VLDB Conference (Montreal, Canada, Oct. 1-3). ACM, New York, 1980, (pp. 411—416).

[S01] Nicolas Spyratos. "*Conceptual Aspects of Information Integration*". Course Notes: Advanced Topics of Databases, University of Crete, Greece. December 2001.
Available at <http://www.csd.uch.gr/~hy562/Handouts.htm>

[STC02] Nicolas Spyratos, Yannis Tzitzikas, Vassilis Christophides. "*On Personalizing the Catalogs of Web Portals*". In Proceedings of the 15th International FLAIRS Conference (FLAIRS'02), Pensacola, Florida. May, 2002.

[SHF+00] Heiner Stuckenschmidt, Frank van Harmelen, Dieter Fensel, Michel Klein, Ian Horrocks. "*Catalogue Integration: A Case Study in Ontology Based Semantic Translation*". Technical Report, AI Department, Vreije Univsersiteit Amsterdam. March 9, 2000.
Available at < http://www.tzi.de/~heiner/public/CatalogueIntegration.pdf>

[SE01] Heiner Stuckenschmidt, Jerome Euzenat. "*Ontology Language Integration: A Constructive Approach*". In Proceedings of the Workshop on Application of Description Logics at the Joint German and Austrian Conference on AI, Vienna. September 18, 2001. CEUR- Workshop Proceedings No. 44.

[SM01a] Gerd Stumme, Alexander Maedche. "*Ontology Merging for Federated Ontologies on the Semantic Web using FCA-Merge*". In Proceedings of the IJCAI'01 Workshop on Ontologies and Information Sharing, Seattle, USA. August 4-5, 2001.

[SM01b] Gerd Stumme, Alexander Maedche. "*FCA-Merge: Bottomup Merging of Ontologies*". In Proceedings of the 17th International Joint Conference on

Artificial Intelligence (IJCAI'01), Seattle, USA. August 1-6, 2001. San Francisco/CA: Morgen Kaufmann.

[S00a] Kazimierz Subieta. "*Mapping Heterogenous Ontologies Through Object Views*". In: M. Roantree, Hasselbring, S. Conrad (eds): Engineering Federated Information Systems. Proc. of the 3rd Workshop EFIS 2000, (pp. 1-10), Dublin. June 9-20, 2000. IOS Press, 2000.

[S98] Dan Suciu. "*Semistructured Data and XML*". In Proceedings of International Conference on Foundations of Data Organization, Japan. Nov. 12-13, 1998.

[U88] Jeffrey D. Ullman. "*Principles of Database and Knowledge-Base Systems*". Volume I. Computer Science Press. 1988.

[UML] OMG Unified Modeling Language Specification. Version 1.4. September, 2001.
Available at <http://cgi.omg.org/docs/formal/01-09-67.pdf>

[U95] Mike Uschold. "*Building Ontologies: Towards a Unified Methodology*". AIAI-TR-197. September 1995. Presented at Expert Systems '96 Conference in Cambridge, UK.

[WVV+01] H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, S. Hübner. "*Ontology-based Integration of Information: A Survey of Existing Approaches*". In Proceedings of IJCAI-01 Workshop: Ontologies and Information Sharing, Seattle, WA (vol. pp.108-117). 2001.

[W98] Robert Wrembel. "*Object-Oriented Views: Virtues and Limitations*". In Proceedings of the 13th International Symposium on Computer and Information Sciences (ISCIS'98), Antalya, Turkey. October, 1998.

[W99] Robert Wrembel. "*On a Formal Model of an Object-Oriented Database with Views Supporting Data Materialization*". In Proceedings of the 3rd East-European Conference on Advances in Databases and Information Systems (ADBIS'99), Maribor, Slovenia. September, 1999.

[XML99] The XML Schema Data types.
Available at <http://www.w3.org/1999/XMLSchema-datatypes>

[YBS97] Sihem Amer-Yahia, Philippe Breche, Cassio Souza Dos Santos "*Object Views and Updates*". Engineering of Information Systems Journal 5(1). 1997.