

Πανεπιστήμιο Κρήτης
Σχολή Θετικών Επιστημών
Τμήμα Επιστήμης Υπολογιστών

Υποστήριξη της Υποστασιοποίησης στο μοντέλο του **RDF**

Τζορμπατζάκη Ι. Αγγελική

Μεταπτυχιακή Εργασία

Ηράκλειο, Φεβρουάριος 2003

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Μεταπτυχιακή εργασία που υποβλήθηκε από την
Αγγελική Ι. Τζορμπατζάκη
ως μερική εκπλήρωση των απαιτήσεων για την απόκτηση
ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΔΙΠΛΩΜΑΤΟΣ ΕΙΔΙΚΕΥΣΗΣ

ΕΙΣΗΓΗΤΙΚΗ ΕΠΙΤΡΟΠΗ:

ΔΗΜΗΤΡΗΣ ΠΛΕΞΟΥΣΑΚΗΣ, Επόπτης
Αναπληρωτής Καθηγητής

ΠΑΝΟΣ ΚΩΝΣΤΑΝΤΟΠΟΥΛΟΣ, Μέλος
Καθηγητής

ΒΑΣΙΛΗΣ ΧΡΙΣΤΟΦΙΔΗΣ, Μέλος,
Επίκουρος Καθηγητής

ΕΓΚΡΙΣΗ:

ΠΑΝΟΣ ΚΩΝΣΤΑΝΤΟΠΟΥΛΟΣ
Πρόεδρος Επιτροπής Μεταπτυχιακών Σπουδών

Ηράκλειο, Φεβρουάριος 2003

*Αφιερωμένο σε όλους όσους μου συμπαραστάθηκαν και πίστεψαν σε μένα και
πάνω απ' όλα στην οικογένεια μου, τους γονείς, τα αδέρφια και
τα ανίψια μου*

Υποστήριξη της υποστασιοποίησης στο μοντέλο του RDF

Τζορμπατζάκη Ι. Αγγελική

Μεταπτυχιακή Εργασία

Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

Περίληψη

Ο Παγκόσμιος Ιστός αρχικά δημιουργήθηκε προκειμένου οι πληροφορίες που βρίσκονται σε αυτόν να «καταναλώνονται» από ανθρώπους, και παρά το γεγονός ότι όλα όσα υπάρχουν σε αυτόν *μπορούν να γραφούν από μηχανές*, κανένα από αυτά τα δεδομένα δεν είναι *κατανοητό για τις μηχανές*. Είναι πολύ δύσκολο να αυτοματοποιηθούν τα πάντα στον Παγκόσμιο Ιστό, και εξαιτίας του όγκου των πληροφοριών που περιέχονται σε αυτόν, είναι αδύνατη η χειρονακτική διαχείριση τους. Μία λύση, ευρύτατα διαδεδομένη, είναι η χρήση μετα-δεδομένων για την περιγραφή των δεδομένων που περιέχονται στον Παγκόσμιο Ιστό. Τα μετα-δεδομένα είναι «δεδομένα σχετικά με δεδομένα». Συγκεκριμένα για την περίπτωση του Παγκόσμιου Ιστού τα μετα-δεδομένα είναι «δεδομένα που περιγράφουν πόρους του Παγκόσμιου Ιστού».

Το Πλαίσιο Περιγραφής Πόρων (Resource Description Framework (RDF)) είναι μια θεμελίωση για την επεξεργασία μετα-δεδομένων. Το RDF παρέχει διαλειτουργικότητα μεταξύ των εφαρμογών που ανταλλάζουν πληροφορία κατανοητή για τις μηχανές στον Παγκόσμιο Ιστό. Μία κατηγορία εφαρμογών στις οποίες έχουμε ανταλλαγή πληροφοριών στον Παγκόσμιο Ιστό είναι οι τα συστήματα Πολλαπλών-Πρακτόρων (Multi-Agents). Τα συστήματα αυτά έχουν την δυνατότητα να επικοινωνούν με άλλα απομακρυσμένα συστήματα και να εκτελούν λειτουργίες αυτόνομα χωρίς να είναι απαραίτητη η συνεχής επικοινωνία με τον χρήστη. Για παράδειγμα, τέτοια συστήματα θα μπορούσαν να διαχειριστούν τον τεράστιο όγκο των πληροφοριών του Παγκόσμιου Ιστού. Ωστόσο, απαραίτητη προϋπόθεση για να συμβούν όλα τα παραπάνω είναι οι πράκτορες να μπορούν να επικοινωνούν μεταξύ τους. Η απλή μεταφορά δυαδικών ψηφίων δεν είναι αρκετή για μια αποτελεσματική επικοινωνία μεταξύ των πρακτόρων. Οι πράκτορες χρειάζεται να επικοινωνούν σε υψηλότερο επίπεδο και να αναπαριστούν περίπλοκες έννοιες, και στο σημείο αυτό είναι χρήσιμη το RDF. Το RDF επιτρέπει την δημιουργία κλάσεων ιεραρχιών και μπορεί να χρησιμοποιηθεί για την αναπαράσταση οντολογιών, επομένως μπορεί να χρησιμοποιηθεί για την περιγραφή των πληροφοριών και τις διαπραγματεύσεις μεταξύ των πρακτόρων.

Η εργασία μας προτείνει ένα εμπλουτισμό του RDF/S ώστε να είναι δυνατή η χρήση της σε συστήματα πολλαπλών πρακτόρων, σε συστήματα ανίχνευσης αλλαγών αποθηκών πληροφορίας και γενικότερα να δίνει την δυνατότητα στο RDF να αποθηκεύει πληροφορίες που έχουν λεχθεί ανεξάρτητα από το αν είναι σίγουρη η αλήθεια τους ή όχι. Ο εμπλουτισμός του RDF/S έγκειται στην υποστήριξη του μηχανισμού της υποστασιοποίησης. Η υποστασιοποίηση (Reification) δίνει την δυνατότητα μετατροπής μίας πρότασης σε ένα αντικείμενο με μοναδικό αναγνωριστικό, έτσι ώστε να μπορούν να γίνουν ισχυρισμοί για την πρόταση αυτή. Για παράδειγμα, η πρόταση “Ο Picasso ζωγράφησε την Guernica” μπορεί να θεωρηθεί ως ένα νέο αντικείμενο και σε αυτή να αποδοθούν ιδιότητες, όπως ποιος την διατύπωσε, πότε την διατύπωσε και σε ποιο χώρο. Το παραπάνω παράδειγμα αποτελεί ένα δείγμα της υποστασιοποίησης σε επίπεδο δεδομένων. Ωστόσο, στην παρούσα εργασία μελετάται η χρήση της υποστασιοποίησης και σε επίπεδο σχήματος, προκειμένου να καταστεί δυνατή η αποθήκευση πληροφοριών για τις Εκδόσεις των RDF Σχημάτων.

Επίσης, στην εργασία αυτή παρουσιάζεται ένα μοντέλο υπεργράφων το οποίο αντικαθιστά τον απλό κατευθυνόμενο γράφο που χρησιμοποιείται για την αναπαράσταση του σχήματος και των μεταδεδομένων στο RDF/S. Ένα άλλο θέμα, το οποίο εξετάζεται είναι η αποθήκευση των υποστασιοποιημένων προτάσεων στις αποθήκες μεταδεδομένων. Τέλος, προτείνεται ο εμπλουτισμός της γλώσσας επερωτήσης μεταδεδομένων, RQL, με τα στοιχεία εκείνα που απαιτούνται προκειμένου να υποστηριχθούν επερωτήσεις υποστασιοποιημένων προτάσεων

Επόπτης: Πλεξουσάκης Δημήτριος
Καθηγητής Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

Reification Support in the model of RDF/S

Tzorbatzaki I. Aggeliki

Master's Thesis

Computer Science Department
University Of Crete

Abstract

The World Wide Web was initially created in order that the information that is found in it to be “consumed” by people, and despite the fact that all that exist in it can be written by machines, none of that data is machine-comprehensible. It is very difficult everything that exists in the World Wide Web to be automated, and because of the volume of information that is contained in it, it is impossible for them to be managed manually. A solution, very widely widespread, is the use of metadata for the description of data that is contained in the World Wide Web. Metadata is “data about data”. Specifically, for the case of World Wide Web metadata is “data describing web resources”.

The Resource Description Framework (RDF) is a foundation for processing metadata. RDF provides interoperability between the applications that exchange machine-understandable information on the Web. A category of applications, in which we have exchange of information in the World Wide Web, are the Multi Agent systems. Multi Agent systems have the ability to communicate with other remoted systems and to execute operations autonomously without being necessary the continuing communication with the user. For example, such systems could manage the enormous volume of information of World Wide Web. However, basic condition in order for all the above to happen is the agents to be able to communicate with each other. The simple transmission of binary digits is not enough for an effective communication between the agents. The agents need to communicate in a higher level and to represent complex concepts, at this point RDF is useful. RDF allows the creation of classes of hierarchies and can be used for the representation of ontologies, consequently it can be used for the description of information and the negotiations between the agents.

Our work proposes an enrichment of RDF/S in order to be used in Multi Agents systems, in changes' detection systems of repository information and more generally to give the ability to RDF to store information that has been said, independently from the fact that their truth is certain or not. RDF/S's enrichment lies in the support of the reification mechanism. Reification gives the ability of the transformation of a statement in an object with a unique identifier, so that assertions can be made for this statement. For example, the statement “Picasso painted Guernica” can be considered as a new object and in this statement new attributes can be ascribed, like who and

when formulated it. The example above constitutes a reification sample in the data level. However, in the present work is also being studied the use of reification in schema level, in order the storage of information on the RDF Schema Versioning to be possible.

Additionally, in this work a hypergraph model, which replaces the simple directed graph that is used for the representation of schema and metadata in the RDF/S, is presented. Another issue, which is examined is the storage of reified statements in metadata repositories. Finally, the enrichment of metadata query language, RQL, with the required elements so that queries of reified statements can be supported, is proposed.

Supervisor: Dimitris Plexousakis
Professor of Computer Science
University Of Crete

Ευχαριστίες

Στο σημείο αυτό θα ήθελα να ευχαριστήσω, όλους όσους με βοήθησαν να φέρω σε πέρας την παρούσα εργασία. Πρώτα από όλα θα ήθελα να ευχαριστήσω θερμά τον επόπτη καθηγητή μου κ. Δημήτρη Πλεξουσάκη για την πολύτιμη βοήθεια, καθοδήγηση και συμπαράσταση που μου πρόσφερε σε όλη τη διάρκεια της μεταπτυχιακής εργασίας. Νιώθω υποχρεωμένη να του εκφράσω την ευγνωμοσύνη μου όχι μόνο για την άψογη συνεργασία την οποία είχαμε αλλά και για το γεγονός ότι μου έδωσε την δυνατότητα να ολοκληρώσω την μεταπτυχιακή μου εργασία και επέδειξε υπομονή και κατανόηση.

Ιδιαίτερος θα ήθελα να ευχαριστήσω τον κ. Πάνο Κωνσταντόπουλο, μέλος της Εισηγητικής Επιτροπής της Μεταπτυχιακής μου Εργασίας τόσο γιατί με τίμησε με την παρουσία του ως μέλος αυτής όσο και για την ευκαιρία που μου έδωσε να εργαστώ ως μέλος της Ομάδας του και να συμμετάσχω στο Μεταπτυχιακό Πρόγραμμα σπουδών του Τμήματος Επιστήμης Υπολογιστών. Επίσης, θα ήθελα να ευχαριστήσω τον κ. Βασίλη Χριστοφίδη, μέλος της Εισηγητικής Επιτροπής, ο οποίος με καθοδήγησε με τις εποικοδομητικές παρατηρήσεις του.

Θα ήθελα να ευχαριστήσω όλα τα μέλη της ομάδας Πληροφοριακών συστημάτων και Τεχνολογίας και ιδιαίτερα την Σοφία Αλεξάκη, τον Γρηγόρη Καρβουναράκη και την Αιμιλία Μαγκαναράκη για τον χρόνο που μου προσέφεραν και για το ότι ήταν πρόθυμοι οποιαδήποτε στιγμή να μου προσφέρουν τις πολύτιμες γνώσεις τους.

Θα ήθελα ακόμα να ευχαριστήσω όλους τους φίλους μου και ιδιαίτερα, την Σταυρούλα και την Ηλέρια, για τις δύσκολες αλλά και τις ωραίες στιγμές που περάσαμε μαζί σε όλη τη διάρκεια των μεταπτυχιακών μας σπουδών. Όπως επίσης και τη Σοφία που άκουγε υπομονετικά τους προβληματισμούς μου. Ένα ιδιαίτερο ευχαριστώ οφείλω στο Βασίλη για την αμέριστη συμπαράσταση και κατανόηση που έδειξε σε μια περίοδο ιδιαίτερα «πιεστική» για μένα.

Θα ήθελα επίσης να ευχαριστήσω την αδερφή μου Ελένη που μου συμπαραστάθηκε ουσιαστικά σε δύσκολες στιγμές της «μεταπτυχιακής πορείας» μου, αλλά και τις αδερφές μου Μαριάνθη και Μαρίνα για την ηθική συμπαράσταση και για το γεγονός ότι πάντα τις ένιωθα στο πλευρό μου. Όσο για τον αδερφό μου Μανόλη, σε αυτόν οφείλω ένα μεγάλο ευχαριστώ εκτός των άλλων και για τις συζητήσεις που κάναμε κάθε φορά που ένιωθα ότι το είχα ανάγκη.

Ωστόσο, το μεγαλύτερο ευχαριστώ το χρωστώ στους γονείς μου, για την κάθε στιγμή που ήταν κοντά μου, για την αγάπη που μου πρόσφεραν απλόχερα και για όλα αυτά που έχουν στερηθεί για χάρη μου.

Τέλος, θα ήθελα να ευχαριστήσω τις γιαγιάδες μου Μαριάνθη και Μαρίνα, που αν και δεν είναι στη ζωή, νιώθω ότι σε αυτές χρωστάω πολλά πράγματα που έχω καταφέρει στη ζωή μου.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1..... 1

1.1 ΕΦΑΡΜΟΓΕΣ ΤΟΥ ΠΛΑΙΣΙΟΥ ΠΕΡΙΓΡΑΦΗΣ ΠΟΡΩΝ.....	1
1.2 ΑΝΤΙΚΕΙΜΕΝΟ ΤΗΣ ΕΡΓΑΣΙΑΣ	2
1.3 ΟΡΓΑΝΩΣΗ ΤΗΣ ΕΡΓΑΣΙΑΣ.....	3

ΚΕΦΑΛΑΙΟ 2..... 5

2.1 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ ΤΟΥ RDF.....	5
2.2 ΣΤΟΧΟΙ – ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ RDF	6
2.3 ΤΑ ΒΑΣΙΚΑ ΣΤΟΙΧΕΙΑ ΤΗΣ RDF	7
2.3.1 ΠΟΡΟΙ ΤΟΥ RDF	7
2.3.2 ΙΔΙΟΤΗΤΕΣ ΤΟΥ RDF	8
2.3.3 ΠΡΟΤΑΣΕΙΣ ΤΟΥ RDF	8
2.4 RDF ΓΡΑΦΟΙ.....	9
2.4.1 RDF ΣΥΛΛΟΓΕΣ	11
2.4.2 ΥΠΟΣΤΑΣΙΟΠΟΙΗΜΕΝΕΣ ΔΗΛΩΣΕΙΣ (REIFIED STATEMENTS).....	15
2.5 ΓΛΩΣΣΑ ΠΕΡΙΓΡΑΦΗΣ ΣΧΗΜΑΤΩΝ (RDF/S)	17
2.4.1 ΒΑΣΙΚΕΣ ΚΛΑΣΕΙΣ ΤΟΥ RDF/S.....	19
2.4.1 ΒΑΣΙΚΕΣ RDF/S ΙΔΙΟΤΗΤΕΣ.....	21
2.6 ΑΝΑΠΑΡΑΣΤΑΣΗ RDF ΜΕ ΤΡΙΑΔΕΣ.....	25
2.7 ΣΥΝΤΑΚΤΙΚΟ RDF/XML.....	26
2.7.1 ΠΕΡΙΓΡΑΦΕΣ (DESCRIPTIONS)	26
2.7.2 ΣΥΛΛΟΓΕΣ	29
2.7.3 ΥΠΟΣΤΑΣΙΟΠΟΙΗΣΗ.....	30
2.7.4 RDF vs. XML.....	31

ΚΕΦΑΛΑΙΟ 3..... 33

3.1 ΥΠΟΣΤΑΣΙΟΠΟΙΗΣΗ.....	33
3.1.1 ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΤΩΝ ΠΡΟΤΑΣΕΩΝ ΣΤΟ RDF.....	33
3.1.2 ΤΥΠΙΚΟ ΜΟΝΤΕΛΟ ΤΟΥ RDF ΚΑΙ ΥΠΟΣΤΑΣΙΟΠΟΙΗΣΗ	37
3.2.3 ΘΕΩΡΗΣΕΙΣ ΓΙΑ ΤΗΝ ΥΠΟΣΤΑΣΙΟΠΟΙΗΣΗ.....	39
3.3 ΓΡΑΦΟΙ ΚΑΙ ΥΠΕΡΓΡΑΦΟΙ ΣΤΟ RDF.....	41
3.3.1 ΜΟΝΤΕΛΟ ΥΠΕΡΓΡΑΦΩΝ.....	44

3.4 ΧΡΗΣΕΙΣ ΤΗΣ ΥΠΟΣΤΑΣΙΟΠΟΙΗΣΗΣ.....	46
3.5 ΥΠΑΡΧΟΥΣΕΣ ΥΛΟΠΟΙΗΣΕΙΣ	48
3.5.1 JENA.....	48
3.5.1.3 ΥΛΟΠΟΙΗΣΗ ΤΗΣ ΣΧΕΣΙΑΚΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ (RDB)	50
3.5.2 TRIPLE.....	52
3.6 ΣΥΜΠΕΡΑΣΜΑΤΑ	53

ΚΕΦΑΛΑΙΟ 4.....55

4.1 RDF/XML ΣΥΝΤΑΞΗ - ΠΑΡΑΔΕΙΓΜΑΤΑ	55
4.1.1 ΓΕΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ RDF/XML ΣΥΝΤΑΞΗΣ	55
4.1.2 RDF/XML ΑΝΑΠΑΡΑΣΤΑΣΗ ΣΧΗΜΑΤΩΝ	57
4.1.3 RDF/XML ΑΝΑΠΑΡΑΣΤΑΣΗ ΥΠΟΣΤΑΣΙΟΠΟΙΗΜΕΝΩΝ ΔΗΛΩΣΕΩΝ.....	59
4.2 ΣΥΝΤΑΚΤΙΚΟΣ ΣΗΜΑΣΙΟΛΟΓΙΚΟΣ ΑΝΑΛΥΤΗΣ RDF ΜΕΤΑΔΕΔΟΜΕΝΩΝ (VRP) 60	
4.2.1 ΜΟΝΤΕΛΟ ΑΝΤΙΚΕΙΜΕΝΩΝ ΤΟΥ VRP	60
4.2.3 ΣΗΜΑΣΙΟΛΟΓΙΚΟΣ ΈΛΕΓΧΟΣ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ.....	62
4.3 ΥΠΑΡΧΟΝ ΜΟΝΤΕΛΟ ΣΧΕΣΙΑΚΗΣ ΑΝΑΠΑΡΑΣΤΑΣΗΣ RDF ΜΕΤΑΔΕΔΟΜΕΝΩΝ 63	
4.3.1 ΠΕΡΙΓΡΑΦΗ ΠΙΝΑΚΩΝ	64
4.4.2 ΤΡΟΠΟΠΟΙΗΣΕΙΣ ΣΤΗΝ ΣΧΕΣΙΑΚΗ ΒΑΣΗ ΑΠΟΘΗΚΕΥΣΗΣ RDF ΜΕΤΑΔΕΔΟΜΕΝΩΝ	68

ΚΕΦΑΛΑΙΟ 5.....75

5.1 Το τυπικό μοντέλο για το RDF/S	75
5.1.1 Το σύστημα τύπων.....	77
5.2 Η ΓΛΩΣΣΑ ΕΠΕΡΩΤΗΣΕΩΝ ΜΕΤΑΔΕΔΟΜΕΝΩΝ RDF.....	79
5.3 ΥΠΟΣΤΗΡΙΞΗ ΕΠΕΡΩΤΗΣΕΩΝ ΥΠΟΣΤΑΣΙΟΠΟΙΗΣΗΣ ΣΤΗΝ RQL	82

ΚΕΦΑΛΑΙΟ 6.....91

6.1 ΑΝΑΠΑΡΑΣΤΑΣΗ ΜΗ-ΔΥΑΔΙΚΩΝ ΣΧΕΣΕΩΝ.....	91
6.2 ΕΠΙΚΟΙΝΩΝΙΑ ΜΕΤΑΞΥ AGENTS.....	94
6.2.1 ΕΙΣΑΓΩΓΗ.....	94
6.2.2 ΑΛΛΗΛΕΠΙΔΡΑΣΗ ΜΕΤΑΞΥ ΠΡΑΚΤΟΡΩΝ.....	95

ΚΕΦΑΛΑΙΟ 7.....109

ΒΙΒΛΙΟΓΡΑΦΙΑ.....111

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

2.1 ΠΑΡΑΔΕΙΓΜΑΤΑ RDF ΓΡΑΦΩΝ.....	10
2.2 ΓΡΑΦΟΣ RDF ΜΕ ΔΟΜΗΜΕΝΗ ΤΙΜΗ	11
2.3 ΠΑΡΑΔΕΙΓΜΑ RDF ΠΑΡΑΛΛΑΓΗΣ	13
2.4 RDF ΓΡΑΦΟΣ ΜΕ ΤΙΜΗ ΙΔΙΟΤΗΤΑΣ ΕΝΑ ΠΟΛΥΣΥΝΟΛΟ.....	13
2.5 ΓΡΑΦΟΣ ΜΕ ΤΙΜΗ ΙΔΙΟΤΗΤΑΣ ΜΙΑ ΠΑΡΑΛΛΑΓΗ.....	14
2.6 RDF ΓΡΑΦΟΣ ΜΕ ΠΛΕΙΟΤΙΜΕΣ ΙΔΙΟΤΗΤΕΣ.....	14
2.7 RDF ΓΡΑΦΟΣ ΜΕ ΤΙΜΗ ΙΔΙΟΤΗΤΑΣ ΕΝΑ ΠΟΛΥΣΥΝΟΛΟ.....	15
2.8 ΥΠΟΣΤΑΣΙΟΠΟΙΗΜΕΝΗ ΔΗΛΩΣΗ.....	16
2.9 ΥΠΟΣΤΑΣΙΟΠΟΙΗΜΕΝΗ ΔΗΛΩΣΗ ΣΤΗΝ ΟΠΟΙΑ ΕΧΕΙ ΑΠΟΔΟΘΕΙ Η ΙΔΙΟΤΗΤΑ NS1:SAID	17
2.10 ΠΑΡΑΔΕΙΓΜΑ RDFS ΣΧΗΜΑΤΟΣ.....	19
2.11 ΙΕΡΑΡΧΙΑ RDF/S ΚΛΑΣΕΩΝ	19
2.12 ΠΕΡΙΟΡΙΣΜΟΙ ΠΟΥ ΔΗΛΩΝΟΝΤΑΙ ΣΤΟ RDF ΣΧΗΜΑ	21
2.13 Η ΣΧΕΣΗ CREATES ΚΑΙ Η ΥΠΟΣΧΕΣΗ ΤΗΣ RAINTS.....	24
2.14 ΣΥΝΤΑΚΤΙΚΗ ΑΝΑΠΑΡΑΣΤΑΣΗ ΕΝΟΣ ΑΠΛΟΥ RDF ΓΡΑΦΟΥ.....	27
2.15 ΕΝΑ ΕΝΔΕΙΚΤΙΚΟ ΠΑΡΑΔΕΙΓΜΑ ΓΡΑΦΟΥ ΠΕΡΙΓΡΑΦΩΝ RDF.....	28
2.16 ΣΥΝΤΑΚΤΙΚΗ ΑΝΑΠΑΡΑΣΤΑΣΗ ΜΙΑΣ ΣΥΛΛΟΓΗΣ ΤΥΠΟΥ BAG.....	29
2.17 ΣΥΝΤΑΚΤΙΚΗ ΑΝΑΠΑΡΑΣΤΑΣΗ ΜΙΑΣ ΣΥΛΛΟΓΗΣ ΤΥΠΟΥ ALT.....	29
2.18 ΣΥΝΤΑΚΤΙΚΗ ΠΕΡΙΓΡΑΦΗ ΠΡΟΤΑΣΕΩΝ ΥΨΗΛΟΤΕΡΗΣ ΤΑΞΗΣ ΜΕ ΥΠΟΣΤΑΣΙΟΠΟΙΗΣΗ	29
2.19 ΥΠΟΣΤΑΣΙΟΠΟΙΗΣΗ ΤΩΝ ΠΡΟΤΑΣΕΩΝ ΠΟΥ ΠΕΡΙΕΧΟΝΤΑΙ ΣΕ ΕΝΑ ΣΤΟΙΧΕΙΟ ΠΕΡΙΓΡΑΦΗΣ.....	30
2.20 ΔΙΑΦΟΡΕΤΙΚΕΣ ΣΥΝΤΑΚΤΙΚΕΣ ΑΝΑΠΑΡΑΣΤΑΣΕΙΣ ΤΗΣ ΙΔΙΑΣ ΠΛΗΡΟΦΟΡΙΑΣ ΣΕ XML.....	32
3.1 ΣΥΝΤΑΚΤΙΚΗ ΑΝΑΠΑΡΑΣΤΑΣΗ ΥΠΟΣΤΑΣΙΟΠΟΙΗΜΕΝΗΣ ΠΡΟΤΑΣΗΣ	35
3.2 ΓΡΑΦΟΣ ΥΠΟΣΤΑΣΙΟΠΟΙΗΜΕΝΗΣ ΠΡΟΤΑΣΗΣ	35
3.3 ΧΡΗΣΗ ΤΟΥ BAG ΓΙΑ ΤΗΝ ΑΝΑΠΑΡΑΣΤΑΣΗ ΟΜΑΔΟΠΟΙΗΜΕΝΩΝ ΠΡΟΤΑΣΕΩΝ.....	36
3.4 ΓΡΑΦΟΣ ΜΙΑΣ ΑΠΛΗΣ ΠΡΟΤΑΣΗΣ (STATEMENT).....	37
3.5 ΓΡΑΦΙΚΗ ΑΝΑΠΑΡΑΣΤΑΣΗ ΑΠΛΗΣ ΠΡΟΤΑΣΗΣ.....	38
3.6 ΠΑΡΑΔΕΙΓΜΑΤΑ RDF ΓΡΑΦΩΝ.....	42
3.7 ΥΠΟΣΤΑΣΙΟΠΟΙΗΜΕΝΗ ΔΗΛΩΣΗ.....	43
3.8 ΠΑΡΑΔΕΙΓΜΑ ΥΠΕΡΓΡΑΦΟΥ.....	44
3.9 ΑΠΛΟ ΠΑΡΑΔΕΙΓΜΑ ΥΠΕΡΓΡΑΦΟΥ.....	45
3.10 ΠΑΡΑΔΕΙΓΜΑ ΕΜΦΩΛΕΥΜΕΝΟΥ ΥΠΕΡΓΡΑΦΟΥ	45
4.1 RDF/XML ΑΝΑΠΑΡΑΣΤΑΣΗ	56
4.2 ΣΥΝΤΟΜΕΥΜΕΝΗ RDF/XML ΑΝΑΠΑΡΑΣΤΑΣΗ	57
4.3 RDF ΣΧΗΜΑ – ΙΕΡΑΡΧΙΑ ΚΛΑΣΕΩΝ.....	58
4.4 RDF ΣΧΗΜΑ ΓΙΑ ΤΗΝ ΠΕΡΙΓΡΑΦΗ ΣΕΛΙΔΩΝ	58
4.5 RDF ΑΝΑΠΑΡΑΣΤΑΣΗ ΥΠΟΣΤΑΣΙΟΠΟΙΗΜΕΝΗΣ ΠΡΟΤΑΣΗΣ.....	59
4.5 ΙΕΡΑΡΧΙΑ ΚΛΑΣΕΩΝ ΤΟΥ ΜΟΝΤΕΛΟΥ VRP.....	60

4.6 Το Μοντέλο Αντικειμένων Του VRP	62
4.7 Σχεσιακή Αναπαράσταση RDF Σχημάτων.....	64
4.8 Πίνακας Δεδομένων.....	67
4.9 Σχεσιακή Αναπαράσταση Με Υποστازیοποίηση	70
4.10 Αναπαράσταση Υποστازیοποιημένων Δεδομένων	72
6.1 Τριγώνικη Σχέση	91
6.2.Αναπαράσταση Τριγώνικης Σχέσης με δύο δυαδικές.....	91
6.3 Αναπαράσταση Τριγώνικης Σχέσης με υποστازیοποίηση.....	92
6.4 Παραδειγμα Τετραδικής Σχέσης.....	93
6.5 Αναπαράσταση Τετραδικής Σχέσης με υποστازیοποίηση.....	94
6.6 Παραδειγμα ΚQML Μηνύματος	97
6.7 Επικοινωνία Μεταξύ Πρακτόρων.....	102
6.8 Παραδειγμα Πολλαπλών Πρακτόρων	104

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

5.1 ΤΥΠΙΚΗ ΕΡΜΗΝΕΙΑ ΤΩΝ ΒΑΣΙΚΩΝ RQL ΕΡΩΤΗΣΕΩΝ.....	80
5.2 ΤΥΠΙΚΗ ΕΡΜΗΝΕΙΑ ΤΩΝ ΒΑΣΙΚΩΝ ΕΚΦΡΑΣΕΩΝ ΜΟΝΟΠΑΤΙΩΝ ΤΗΣ RQL	81

ΚΕΦΑΛΑΙΟ 1

Εισαγωγή

1.1 Εφαρμογές του Πλαισίου Περιγραφής Πόρων

Ο Παγκόσμιος Ιστός αρχικά δημιουργήθηκε προκειμένου οι πληροφορίες που βρίσκονται σε αυτόν να «καταναλώνονται» από ανθρώπους, και παρά το γεγονός ότι όλα όσα υπάρχουν σε αυτόν *μπορούν να γραφούν από μηχανές*, κανένα από αυτά τα δεδομένα δεν είναι *κατανοητό για τις μηχανές*. Είναι πολύ δύσκολο να αυτοματοποιηθούν τα πάντα στον Παγκόσμιο Ιστό, και εξαιτίας του όγκου των πληροφοριών που περιέχονται σε αυτόν, είναι αδύνατη η χειρονακτική διαχείριση τους. Μία λύση, ευρύτατα διαδεδομένη, είναι η χρήση μετα-δεδομένων για την περιγραφή των δεδομένων που περιέχονται στον Παγκόσμιο Ιστό. Τα μετα-δεδομένα είναι «δεδομένα σχετικά με δεδομένα» (για παράδειγμα, ο κατάλογος μιας βιβλιοθήκης είναι μετα-δεδομένα, διότι περιγράφει τις εκδόσεις). Συγκεκριμένα για την περίπτωση του Παγκόσμιου Ιστού τα μετα-δεδομένα είναι «δεδομένα που περιγράφουν πόρους του Παγκόσμιου Ιστού». Η διάκριση μεταξύ «δεδομένων» και «μετα-δεδομένων» δεν είναι απόλυτη, είναι μια διάκριση που δημιουργήθηκε αρχικά από μια συγκεκριμένη εφαρμογή, και πολλές φορές ο ίδιος πόρος μπορεί να μεταφράζεται και με τους δύο όρους («δεδομένα» και «μετα-δεδομένα») ταυτόχρονα.

Το Πλαίσιο Περιγραφής Πόρων (Resource Description Framework (RDF)) είναι μια θεμελίωση για την επεξεργασία μετα-δεδομένων. Το RDF παρέχει διαλειτουργικότητα μεταξύ των εφαρμογών που ανταλλάζουν πληροφορία κατανοητή για τις μηχανές στον Παγκόσμιο Ιστό. Μία κατηγορία εφαρμογών στις οποίες έχουμε ανταλλαγή πληροφοριών στον Παγκόσμιο Ιστό είναι τα συστήματα Πολλαπλών-Πρακτόρων (Multi-Agents). Τα συστήματα αυτά έχουν την δυνατότητα να επικοινωνούν με άλλα απομακρυσμένα συστήματα και να εκτελούν λειτουργίες αυτόνομα χωρίς να είναι απαραίτητη η συνεχής επικοινωνία με τον χρήστη. Για παράδειγμα, τέτοια συστήματα θα μπορούσαν να διαχειριστούν τον τεράστιο όγκο των πληροφοριών του Παγκόσμιου Ιστού. Ωστόσο, απαραίτητη προϋπόθεση για να συμβούν όλα τα παραπάνω είναι οι πράκτορες να μπορούν να επικοινωνούν μεταξύ τους. Η απλή μεταφορά δυαδικών ψηφίων δεν είναι αρκετή για μια αποτελεσματική επικοινωνία μεταξύ των πρακτόρων. Οι πράκτορες χρειάζεται να επικοινωνούν σε υψηλότερο επίπεδο και να αναπαριστούν περίπλοκες έννοιες, και στο σημείο αυτό είναι χρήσιμο το RDF [O00]. Το RDF επιτρέπει την δημιουργία κλάσεων ιεραρχιών, αλλά επίσης μπορεί να αναπαραστήσει οντολογίες, επομένως μπορεί να χρησιμοποιηθεί τόσο για την περιγραφή των πληροφοριών όσο και για τις διαπραγματεύσεις μεταξύ των

πρακτόρων. Πέραν των παραπάνω όμως, στα συστήματα πρακτόρων, λόγω του ότι γίνεται πολύ συχνά ανταλλαγή πληροφοριών, θέτονται ζητήματα προέλευσης της πληροφορίας, δηλαδή είναι απαραίτητο να είναι γνωστή η πηγή της πληροφορίας, ώστε να υπάρχει η δυνατότητα ελέγχου της αλήθειας αυτής. Το θέμα της προέλευσης της πληροφορίας σχετίζεται άμεσα με την πιστότητα αυτής. Σε περίπτωση που η πληροφορία προέρχεται από μία αξιόπιστη πηγή συνεπάγεται άμεσα ότι η πληροφορία θεωρείται έγκυρη, σε αντίθεση με μία πληροφορία η οποία δεν προέρχεται από αξιόπιστη πηγή. Είναι εμφανές, λοιπόν, ότι η παραπάνω αξιολόγηση δεν θα ήταν εφικτή σε περίπτωση μη ύπαρξης της προέλευσης της πληροφορίας. Ωστόσο, αν και το RDF μπορεί να προσφέρει πολλά σε εφαρμογές πρακτόρων όπως ανταλλαγή και διαμοιρασμό των πληροφοριών, ακόμα δεν προσφέρει την δυνατότητα αυτή.

Ένα άλλο ζήτημα, που ανακύπτει πολλές φορές στις εφαρμογές πρακτόρων που διαχειρίζονται μεταδεδομένα είναι ότι έρχονται αντιμέτωποι με σχήματα μεταδεδομένων τα οποία τους είναι εντελώς άγνωστα και επομένως δεν μπορούν να τα επεξεργαστούν. Ωστόσο πολλά από αυτά τα σχήματα μεταδεδομένων μπορεί να προέρχονται από κάποια αρχικά σχήματα τα οποία είναι ήδη γνωστά στους πράκτορες και επομένως τα μεταδεδομένα που περιέχουν μπορούν να διαχειριστούν από αυτούς. Απαραίτητη προϋπόθεση για να συμβεί κάτι τέτοιο είναι να υπάρχει ένας μηχανισμός με τον οποίο θα είναι δυνατή η εύρεση του αρχικού σχήματος από το οποίο προέκυψε το υπάρχον. Αυτό συνεπάγεται ότι κάθε φορά που γίνεται μία προσθήκη στο αρχικό σχήμα μεταδεδομένων πρέπει να αποθηκεύονται πληροφορίες για την αλλαγή αυτή, όπως για παράδειγμα από ποιόν έγινε η αλλαγή και πότε.

Γενικότερα όμως, ο Παγκόσμιος Ιστός αποτελείται από πάρα πολλά τόσο δεδομένα όσο και μεταδεδομένα. Οποιοσδήποτε μπορεί να δημιουργήσει μία ιστοσελίδα στον Παγκόσμιο Ιστό και να προσθέσει μεταδεδομένα με την βοήθεια είτε της RDF είτε της XML. Παρόλα αυτά, η ποσότητα των πληροφοριών δεν συνεπάγεται άμεσα και ποιότητα. Ένα πρόβλημα που προκύπτει και ζητάει λύση όλο και πιο επιτακτικά είναι η αξιοπιστία των μεταδεδομένων. Αξιοπιστία των μεταδεδομένων σημαίνει ότι πρέπει να προσθέτονται πληροφορίες στα μεταδεδομένα σχετικά με το ποιος τα έχει διατυπώσει και την χρονική στιγμή που διατυπώθηκαν αυτά, καθώς μία πληροφορία που ισχύει την παρούσα χρονική στιγμή μπορεί να μην ισχύει την επόμενη, ειδικά στο Περιβάλλον του Παγκόσμιου Ιστού που χαρακτηρίζεται από μια συνεχής μεταβλητότητα. Έτσι για παράδειγμα, η πληροφορία που μας παρέχει η πρόταση «Ο τίτλος της σελίδας με URI <http://www.w3c.org> είναι *World Wide Web Consortium*» μπορεί να είναι αληθής μόνο για το χρονικό διάστημα από 1-12-1999 έως 31-8-2001, ενώ από 1-09-2001 μέχρι την παρούσα στιγμή μπορεί να έχει μεταβληθεί. Επίσης η πληροφορία ότι «Το RDF περιγράφει πληροφοριακούς πόρους» μπορεί να θεωρηθεί αξιόπιστη μόνο αν για παράδειγμα έχει διατυπωθεί από την Ora Lassila.

1.2 Αντικείμενο της εργασίας

Η εργασία μας προτείνει ένα εμπλουτισμό του RDF ώστε να είναι δυνατή η χρήση του σε συστήματα πολλαπλών πρακτόρων, σε συστήματα ανίχνευσης αλλαγών αποθηκών πληροφορίας και γενικότερα να δίνει την δυνατότητα στο RDF να αποθηκεύει πληροφορίες που έχουν λεχθεί ανεξάρτητα από το αν είναι σίγουρη η αλήθεια τους ή όχι.

Ο εμπλουτισμός του RDF έγκειται στην υποστήριξη του μηχανισμού της υποστασιοποίησης. Η υποστασιοποίηση (Reification) δίνει την δυνατότητα μετατροπής μίας πρότασης σε ένα αντικείμενο με μοναδικό αναγνωριστικό, έτσι ώστε να μπορούν να γίνουν ισχυρισμοί για την πρόταση αυτή. Για παράδειγμα, η πρόταση “Ο Picasso ζωγράφησε την Guernica” μπορεί να θεωρηθεί ως ένα νέο αντικείμενο και σε αυτή να αποδοθούν ιδιότητες, όπως ποιος την διατύπωσε, πότε την διατύπωσε και σε ποιο χώρο. Το παραπάνω παράδειγμα αποτελεί ένα δείγμα της υποστασιοποίησης σε επίπεδο δεδομένων.

Ωστόσο, πολλές φορές είναι απαραίτητο να υποστασιοποιήσουμε, δηλαδή να μετατρέψουμε σε αντικείμενα στα οποία στη συνέχεια θα μπορούμε να αναφερθούμε όχι μόνο προτάσεις δεδομένων αλλά και προτάσεις που περιγράφουν το σχήμα του RDF. Για παράδειγμα, ποιος από τους διαχειριστές του σχήματος περιγραφής του RDF, για ένα μουσείο προσέθεσε ότι “Η κλάση Painter είναι υποκλάση της κλάσης Artist”.

Η επιστημονική συνεισφορά αυτής της εργασίας έγκειται στα εξής:

- Παρουσιάζεται ένα μοντέλο υπεργράφων το οποίο δίνει την δυνατότητα υποστήριξης του μηχανισμού της υποστασιοποίησης.
- Ο μηχανισμός της υποστασιοποίησης υποστηρίζεται όχι μόνο σε επίπεδο δεδομένων, όπως συμβαίνει στις μέχρι τώρα υπάρχουσες υλοποιήσεις αλλά και σε επίπεδο σχήματος του RDF.
- Προτείνονται οι απαιτούμενες αλλαγές στην σχεσιακή βάση αναπαράστασης των RDF μεταδεδομένων ώστε να είναι δυνατή η υλοποίηση του μηχανισμού της υποστασιοποίησης.
- Προτείνεται ο εμπλουτισμός της δηλωτικής γλώσσας RQL, η οποία επιτρέπει την επερώτηση περιγραφών και σχημάτων RDF, με τα στοιχεία εκείνα τα οποία θα καταστήσουν δυνατή την επερώτηση των υποστασιοποιημένων δηλώσεων.

1.3 Οργάνωση της εργασίας

Η εργασία μας έχει οργανωθεί ως εξής:

- Στο κεφάλαιο 2, παρουσιάζονται οι βασικές έννοιες του μοντέλου δεδομένων του RDF και του RDFS.
- Στο κεφάλαιο 3, γίνεται μία πιο αναλυτική περιγραφή της έννοιας της υποστασιοποίησης, και παραθέτονται απόψεις και ορισμοί για το αμφιλεγόμενο ζήτημα της υποστασιοποίησης. Στη συνέχεια, παρουσιάζονται οι υπάρχουσες υλοποιήσεις συστημάτων RDF που υποστηρίζουν την υποστασιοποίηση. Επίσης, περιγράφεται το μοντέλο

υπερ-γράφων που σχεδιάσαμε για την αναπαράσταση του RDF και δίνονται οι ορισμοί του μοντέλου μας και οι ορισμοί της υποστασιοποίησης στο μοντέλο μας. Τέλος, παρουσιάζονται τα υπάρχοντα συστήματα στα οποία έχει υλοποιηθεί η υποστασιοποίηση.

- Στο κεφάλαιο 4, προτείνεται ένας τρόπος αναπαράστασης και αποθήκευσης των RDF μεταδεδομένων στο Σχεσιακό Σύστημα διαχείρισης Βάσεων Δεδομένων με υποστήριξη της Υποστασιοποίησης.
- Στο κεφάλαιο 5, παρουσιάζεται η υπάρχουσα γλώσσα επερώτησης RQL. Στην συνέχεια, παραθέτονται οι προτεινόμενες αλλαγές ώστε να είναι δυνατή η ανάκτηση πληροφοριών υποστασιοποιημένων δηλώσεων τόσο σε επίπεδο δεδομένων όσο και σε επίπεδο σχήματος.
- Τέλος, στο κεφάλαιο 6, γίνεται μια σύνοψη της επιστημονικής συνεισφοράς και παρουσιάζονται σενάρια χρήσης της υποστασιοποίησης.

ΚΕΦΑΛΑΙΟ 2

Το Πρότυπο RDF/S

Στο κεφάλαιο αυτό θα γίνει μια παρουσίαση του **Περιβάλλοντος Περιγραφής Πόρων** (Resource Description Framework - RDF), το οποίο αποτελεί πρότυπο του World Wide Web Consortium (W3C). Το RDF είναι μια θεμελίωση για την επεξεργασία μετα-δεδομένων και παρέχει διαλειτουργικότητα μεταξύ των εφαρμογών που ανταλλάζουν πληροφορία κατανοητή για τις μηχανές στον Παγκόσμιο Ιστό. Επίσης, δίνει έμφαση σε λειτουργίες που κάνουν δυνατή της αυτοματοποίηση της επεξεργασίας των πόρων του Παγκόσμιου Ιστού. Το RDF μπορεί να χρησιμοποιηθεί σε μία ποικιλία περιοχών εφαρμογής, για παράδειγμα: στην *ανακάλυψη πόρων* για την παροχή καλύτερων δυνατοτήτων στις μηχανές αναζήτησης, στην *δημιουργία καταλόγων* για την περιγραφή των περιεχομένων και σχέσεων μεταξύ των αυτών που είναι διαθέσιμα σε ένα Δικτυακό Τόπο, μία σελίδα ή μια ψηφιακή βιβλιοθήκη ή από συστήματα *ευφώνων πρακτόρων* για την διευκόλυνση του διαμοιρασμού και της ανταλλαγής πληροφοριών καθώς και σε πολλές άλλες εφαρμογές.

2.1 Ιστορική Αναδρομή του RDF

Το RDF προτάθηκε από το World-Wide-Web Consortium (W3C) ως επέκταση της Πλατφόρμας για επιλογή περιεχομένου στο Διαδίκτυο (PICS – Platform for Internet Content Selection), η οποία προτάθηκε από το W3C το 1995. Το PICS αποτελούσε ένα μηχανισμό για την περιγραφή των περιεχομένων ιστοσελίδων και την αξιολόγηση αυτών. Οι πληροφορίες αυτές ανταλλάσσονταν μεταξύ ενός Εξυπηρετητή Ιστού (Web Server) και των πελατών-φυλλομετρητών (Web Browsers). Τέτοιες πληροφορίες θα μπορούσαν να αφορούν για παράδειγμα στην καταλληλότητα των ιστοσελίδων για μικρά παιδιά ή στον συγγραφέα ενός άρθρου μιας ιστοσελίδας. Οι πληροφορίες αυτές αξιολογούνταν από τον εκάστοτε οργανισμό ή άτομο ανάλογα με τις τρέχουσες απαιτήσεις, καθώς σε κάθε περίπτωση τα κριτήρια για την αξιολόγηση μιας σελίδας ποικίλουν. Στην περίπτωση ενός σχολικού δικτύου, για παράδειγμα, θα ήταν απαραίτητη η αποτροπή της πρόσβασης σε δικτυακούς τόπους με βίαιο ή ακατάλληλο περιεχόμενο. Αυτή η δυνατότητα δίδονταν στο σύστημα με ρύθμιση των φυλλομετρητών ώστε να μην εμφανίζονται αυτές οι ιστοσελίδες. Η δημιουργία του μηχανισμού αυτού ανέκυψε από την ανάγκη για περιορισμό και αξιολόγηση της πληθώρας των περιεχομένων του Παγκόσμιου Ιστού.

Η αρχική ιδέα λοιπόν είχε γεννηθεί, και ήταν η ανταλλαγή πληροφοριών κατανοητών τόσο από τον άνθρωπο όσο και από τον υπολογιστή. Μέχρι τη στιγμή αυτή, οι πληροφορίες που ανταλλάσσονταν στον Παγκόσμιο Ιστό, ήταν να μεν κατανοητές από τον άνθρωπο, αλλά για τον υπολογιστή δεν είχαν κανένα νόημα, καθώς αυτός αποτελούσε απλά το μέσο για την μεταφορά και την παρουσίαση των πληροφοριών. Προέκυψε, λοιπόν, η ανάγκη περιγραφής των πόρων του Παγκόσμιου Ιστού, με μία νέα μορφή αναγνώσιμη από τον άνθρωπο και κατανοητή από τον υπολογιστή. Για το σκοπό αυτό δημιουργήθηκε μία νέα ομάδα εργασίας από το W3C, στην οποία συμμετείχαν μέλη από διάφορα ερευνητικά ιδρύματα και εταιρείες. Τα μέλη της νεοϊδρυθείσας αυτής ομάδας συνεργάστηκαν για την δημιουργία ενός κοινώς αποδεκτού προτύπου για την περιγραφή πόρων στον Παγκόσμιο Ιστό.

Το πρότυπο αυτό δεν έμεινε ανεπηρέαστο από άλλες περιοχές και κοινότητες. Οι βασικότερες επιρροές που δέχτηκε ήταν, φυσικά, από την Κοινότητα Προτυποποίησης του Παγκόσμιου Ιστού, του W3C. Επίσης άντλησε στοιχεία τόσο από την περιοχή των ψηφιακών βιβλιοθηκών όσο και από την περιοχή της αναπαράστασης γνώσεων. Δέχτηκε σημαντικές επιδράσεις από την περιοχή των δομημένων εγγράφων, από την SGML και ιδιαίτερα από την XML. Τέλος, καθώς η ομάδα περιλάμβανε μέλη από ομάδες άλλων προσπαθειών προτυποποίησης, όπως το Dublin Core [DC], το Warwick Framework [LLJ96] και την αρχιτεκτονική Metadata Content Framework [GB97], επηρεάστηκε σε σημαντικό βαθμό και από αυτές.

2.2 Στόχοι – Χαρακτηριστικά του RDF

Ο απώτερος στόχος του RDF είναι ο ορισμός ενός μηχανισμού περιγραφής πληροφοριακών πόρων, ο οποίος θα είναι κατάλληλος τόσο για την περιγραφή των πληροφοριακών πόρων ανεξάρτητα από το πεδίο εφαρμογής, όσο και για την αναπαράσταση της γνώσης σε ένα συγκεκριμένο πεδίο εφαρμογής, μέσω εννοιών και ιδιοτήτων.

Οι βασικότεροι στόχοι του RDF, πλην των παραπάνω, συνοψίζονται στους εξής:

- **Εύκολη Ανταλλαγή και Μεταφορά μεταδεδομένων:** Τα μεταδεδομένα του RDF μπορούν να μεταφερθούν εύκολα μεταξύ των υπολογιστών και να αποθηκευτούν είτε σε απλά συστήματα αρχείων είτε σε εξειδικευμένα σχεσιακά συστήματα διαχείρισης μεταδεδομένων.
- **Επεκτασιμότητα και Επαναχρησιμοποίηση των Σχημάτων:** Προκειμένου να διευκολυνθεί ο ορισμός των μεταδεδομένων, το RDF έχει ένα σύστημα κλάσεων που μοιάζει πολύ με τα οντοκεντρικά συστήματα προγραμματισμού και τα συστήματα μοντελοποίησης. Μία συλλογή από κλάσεις ονομάζεται σχήμα. Οι κλάσεις οργανώνονται σε μία ιεραρχία και προσφέρουν επεκτασιμότητα μέσω της δυνατότητας προσθήκης υποκλάσεων στο αρχικό σχήμα. Με αυτόν τον τρόπο, προκειμένου να δημιουργηθεί ένα σχήμα που διαφέρει λίγο σε σχέση με ένα υπάρχον, δεν είναι απαραίτητη η δημιουργία του νέου σχήματος από την αρχή. Αντίθετα, βασιζόμενοι στο υπάρχον σχήμα και με διαδοχικές τροποποιήσεις είναι δυνατόν να προκύψει το νέο σχήμα.

- **Δυνατότητα Περιγραφής του Σχήματος:** Τα σχήματα περιγραφής μεταδεδομένων αποτελούνται από κλάσεις και ιδιότητες και μπορούν να περιγραφούν τα ίδια με τη βοήθεια του RDF. Επομένως, είναι δυνατή η δημιουργία μεταδεδομένων για το ίδιο το σχήμα, γεγονός που καθιστά δυνατή τόσο την ανάκτηση πληροφοριών σχετικά με τις κλάσεις και τις ιδιότητες, όσο και την αποθήκευση πληροφοριών για το σχήμα. Έτσι, για παράδειγμα, μπορεί να ερωτηθεί αν υπάρχει κάποια ιδιότητα ορισμένη σε κάποιο σχήμα RDF που περιγράφει το δημιουργό μιας σελίδας του Παγκόσμιου Ιστού.
- **Δυνατότητα Περιγραφής των ίδιων των μεταδεδομένων:** Μία πολύ σημαντική ιδιότητα του RDF είναι η δυνατότητα που παρέχει για ύπαρξη μεταδεδομένων για τα μεταδεδομένα των πληροφοριακών πόρων. Στον Παγκόσμιο Ιστό ο καθένας μπορεί να δημιουργήσει μεταδεδομένα. Το θέμα που ανακύπτει σε αυτή την περίπτωση είναι πόσο αξιόπιστα είναι τα μεταδεδομένα αυτά, και βέβαια η αξιοπιστία των μεταδεδομένων έχει άμεση σχέση με την προέλευση αυτών. Αν για παράδειγμα, τα μεταδεδομένα προέρχονται από μία γνωστή επιστημονική κοινότητα, τότε μπορούν να χρησιμοποιηθούν με ασφάλεια. Η πληροφορία που μας παρέχει η πρόταση «Ο τίτλος της σελίδας με URI <http://www.w3c.org> είναι *World Wide Web Consortium*», είναι πολύ περισσότερο χρήσιμη αν ξέρουμε ποιος έχει διατυπώσει την πρόταση αυτή και σε ποια χρονική στιγμή.

2.3 Τα Βασικά Στοιχεία της RDF

Το βασικό μοντέλο δεδομένων του RDF αποτελείται από τρεις βασικούς τύπους αντικειμένων, *τους Πόρους (Resources)*, *τις Ιδιότητες (Properties)* και *τις Προτάσεις (Statements)*:

2.3.1 Πόροι του RDF

Οτιδήποτε μπορεί να περιγραφεί με RDF εκφράσεις ονομάζεται *πόρος*. Οι πόροι μπορεί να είναι:

- Δικτυακοί Τόποι, όπως για παράδειγμα το <http://ics.forth.gr>
- Συγκεκριμένα έγγραφα ή σελίδες (HTML, XML, κλπ.)
- Συγκεκριμένα τμήματα τέτοιων εγγράφων, καθώς αυτά μπορούν να περιγραφούν με ένα μοναδικό αναγνωριστικό πόρου (Uniform Resource Identifier - URI).
- Τέλος, αντικείμενα που δεν είναι προσπελάσιμα από τον παγκόσμιο Ιστό, όπως π.χ. βιβλία, πίνακες ζωγραφικής, πρόσωπα κτλ.

Σε κάθε πόρο αποδίδεται ένα μοναδικό αναγνωριστικό (Uniform Resource Identifier - URI), στο οποίο μπορεί να προστεθεί προαιρετικά μία άγκυρα (anchor ids). Η άγκυρα περιγράφεται με τον χαρακτήρα «#». Για παράδειγμα το URI

http://www.ics.forth.gr/proj/isst/RDF αποδίδεται στην σελίδα με την αντίστοιχη ηλεκτρονική διεύθυνση. Ενώ, το URI *http://www.ics.forth.gr/RDF/Dublin_Core.rdf#Title* χαρακτηρίζει την ιδιότητα *Title* που ορίζεται στο αρχείο *http://www.ics.forth.gr/RDF/Dublin_Core.rdf*.

Οτιδήποτε μπορεί να έχει ένα URI. Η επεκτασιμότητα των URI επιτρέπει την εισαγωγή αναγνωριστικών για κάθε δυνατή οντότητα.

2.3.2 Ιδιότητες του RDF

Η *ιδιότητα* είναι είτε ένα γνώρισμα, χαρακτηριστικό ή ιδιότητα που χρησιμοποιείται για να περιγράψει ένα πόρο είτε μία σχέση μεταξύ πόρων. Κάθε ιδιότητα έχει ορισμένη σημασιολογία, και πιθανόν κάποιο πεδίο ορισμού, πεδίο τιμών και συσχετίσεις με άλλες ιδιότητες.

Οι ιδιότητες αποτελούν μια ειδική κατηγορία πόρων. Όπως, οι πόροι έτσι και οι ιδιότητες χαρακτηρίζονται από μοναδικά αναγνωριστικά (URIs). Η ύπαρξη μοναδικών αναγνωριστικών δίνει την δυνατότητα περιγραφής των ίδιων των ιδιοτήτων από εκφράσεις RDF, γεγονός που συμβάλλει στην απλότητα του μοντέλου.

Ένα χαρακτηριστικό των ιδιοτήτων στο RDF είναι ότι αποτελούν πολίτες πρώτης κατηγορίας. Πράγματι, στο RDF η τριάδα, η οποία αποτελεί την στοιχειώδη οντότητα πληροφορίας, οφείλεται στον ορισμό της ιδιότητας, η οποία ουσιαστικά περιγράφει τη σημασιολογία της πρότασης. Τέλος, οι ιδιότητες μπορούν να οριστούν και να χρησιμοποιηθούν ανεξάρτητα από τις κλάσεις, αν και η έννοια των κλάσεων εξακολουθεί να υφίστανται.

Παρόμοια με άλλα μοντέλα ημιδομημένων δεδομένων, οι ιδιότητες στο RDF είναι προαιρετικές (optional) και πιθανόν πλειότιμες (multi-valued). Αυτό συνεπάγεται ότι για κάποιο πόρο που ανήκει στο πεδίο ορισμού μιας ιδιότητας, μπορεί είτε να μην οριστεί η ιδιότητα αυτή είτε να οριστεί παραπάνω από μία φορές. Το αποτέλεσμα της ιδιαιτερότητας αυτής του RDF είναι ότι σε αντίθεση με το σχεσιακό μοντέλο, στα στιγμιότυπα μεταδεδομένων δεν εμφανίζονται μηδενικές τιμές (null), γεγονός που παρέχει μεγαλύτερη ευελιξία τόσο στην επερώτηση όσο και στην αναπαράσταση των σχημάτων του RDF.

2.3.3 Προτάσεις του RDF

Μία πρόταση είναι ένας συγκεκριμένος πόρος, που υποστασιοποιεί (reify) μία τριάδα. Η τριάδα αυτή έχει την μορφή *πόρος-ιδιότητα-τιμή*, επομένως ο πόρος έχει τουλάχιστον τρεις ιδιότητες, που ορίζονται από το μοντέλο του RDF, τις : *rdf:subject*, *rdf:predicate* και *rdf:object* με τις αντίστοιχες τιμές από τα μέρη της τριάδας. Το αντικείμενο μίας πρότασης, *rdf:object* μπορεί να είναι είτε ένας πόρος είτε ένα Literal. Ο πόρος αυτός μπορεί να έχει με τη σειρά του άλλες ιδιότητες. Ο τύπος Literal συμπεριλαμβάνει τα αλφαριθμητικά και τους άλλους ατομικούς τύπους

δεδομένων που ορίζονται στο XML Schema. Τέλος, ένας τέτοιος πόρος έχει την ιδιότητα `rdf:type`, με τιμή `rdf:Statement`.

Η υποστασιοποίηση (reification) των τριάδων αποτελεί θεμελιώδη έννοια στο μοντέλο του RDF, και μπορεί να χρησιμοποιηθεί για τον ορισμό του ίδιου του RDF με περιγραφές RDF. Η υποστασιοποίηση των προτάσεων, δηλαδή η δυνατότητα θεώρησης των ίδιων των προτάσεων ως πόρων επιτρέπει να αναθέτονται ιδιότητες σε προτάσεις (η πρόταση αποτελεί το υποκείμενο της πρότασης) και να θέτονται προτάσεις ως τιμές ιδιοτήτων (η πρόταση αποτελεί το αντικείμενο της πρότασης).

Έτσι, για παράδειγμα, μπορούν να αναπαρασταθούν στο RDF προτάσεις της μορφής:

“Ο Picasso ζωγράφισε την Guernica” δηλώθηκε από τον Paul
“Ο Picasso ζωγράφισε την Guernica” δηλώθηκε στις 20/11/2000

Οπότε και έχουμε μία συλλογή προτάσεων που χαρακτηρίζουν την πρόταση αυτή.

Επίσης, μπορούμε να έχουμε εκφράσεις της μορφής:

Ο John δήλωσε ότι “Ο Rodin ζωγράφισε την Guernica”

Η δυνατότητα σχηματισμού προτάσεων που έχουν ως μέλη τους άλλες προτάσεις, ανάγει το μοντέλο του RDF από πρωτοβάθμιο (first-order), σε ανώτερης τάξης (higher order). Επίσης, καθώς οι υποστασιοποιημένες προτάσεις αποτελούν πόρους, μπορούν να αποτελέσουν το υποκείμενο ή το αντικείμενο μίας άλλης πρότασης. Με τον τρόπο αυτό δημιουργούνται φωλιασμένες εκφράσεις στο RDF. Για παράδειγμα, στο RDF μπορεί να αναπαρασταθεί η παρακάτω φράση:

Ο Paul είπε ότι “Ο John είπε ότι “Ο Rodin ζωγράφισε την Guernica” ”

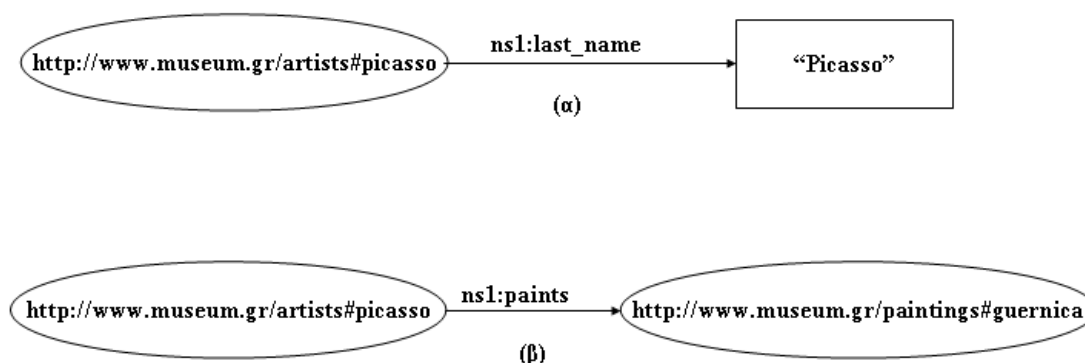
2.4 RDF Γράφοι

Ένας τρόπος αναπαράστασης των RDF περιγραφών είναι οι γράφοι. Συγκεκριμένα, οι γράφοι που χρησιμοποιεί το RDF είναι οι *κατευθυνόμενοι γράφοι με ετικέτες*. Αυτοί οι γράφοι ονομάζονται και διαγράμματα κόμβων και ακμών (nodes and arcs diagrams). Οι κόμβοι αντιστοιχούν στους πόρους ή σε Literal ενώ οι ακμές στις ιδιότητες. Οι κόμβοι που αντιστοιχούν σε πόρους απεικονίζονται με ελλειψοειδή σχήματα, ενώ αυτοί που αντιστοιχούν σε Literal απεικονίζονται με παραλληλόγραμμα σχήματα. Δεδομένου ότι το πεδίο ορισμού μίας ιδιότητας είναι πάντα ένας πόρος ενώ το σύνολο τιμών είναι είτε ένας πόρος είτε ένα Literal, συνεπάγεται ότι μία ακμή στο γράφο θα ξεκινάει πάντα από μία έλλειψη και θα καταλήγει είτε σε μία έλλειψη είτε σε ένα παραλληλόγραμμο.

Στο σχήμα 2.1 απεικονίζεται μία RDF περιγραφή με τη βοήθεια ενός κατευθυνόμενου γράφου με ετικέτες.

Στο παράδειγμα αυτό, ο γράφος 2.1(α) δηλώνει ότι ο κόμβος <http://www.museum.gr/artists#picasso> έχει μία ιδιότητα, την http://www.culture.gr/schema.rdf#last_name, με τιμή “Picasso”.

Ο γράφος 2.1(β) δηλώνει ότι ο κόμβος <http://www.museum.gr/artists#picasso> έχει μία ιδιότητα <http://www.culture.gr-/schema.rdf#creates> με τιμή <http://www.museum.gr/paintings#Guernica>.



Σχήμα 2.1 Παραδείγματα RDF γράφων

Σε φυσική γλώσσα ο γράφος του σχήματος 2.1(α) αναπαριστά τη φράση:

Το επώνυμο (ns1:last_name) του <http://www.museum.gr/artists#picasso> είναι “Picasso”.

και ο γράφος του σχήματος 2.1(β) αναπαριστά τη φράση:

Ο δημιουργός (ns1:creator) της Guernica, <http://www.museum.gr/paintings#Guernica> είναι ο <http://www.museum.gr/artists#picasso>.

Η τριάδα *πόρος-ιδιότητα πόρου-τιμή* ονομάζεται πρόταση στην ορολογία του RDF. Τα τρία αυτά μέρη της πρότασης, αποτελούν αντίστοιχα το υποκείμενο, το κατηγορημα και το αντικείμενο της πρότασης.

Στο παραπάνω παράδειγμα χρησιμοποιήθηκε το πρόθεμα ns1 για τις ιδιότητες last_name και creator, στη θέση του πλήρους URI http://www.culture.gr/schema.rdf#last_name και <http://www.culture.gr/schema.rdf#creator> αντίστοιχα. Η συντομογραφία αυτή είναι εφικτή λόγω της δυνατότητας που παρέχει η XML για ορισμό προθεμάτων τα οποία υποδεικνύουν το έγγραφο στο οποίο έχει οριστεί το αντίστοιχο στοιχείο. Η δυνατότητα αυτή είναι ιδιαίτερα χρήσιμη στο RDF για τον ορισμό του σχήματος – λεξιλογίου (RDF Schema) στο οποίο έχει οριστεί η αντίστοιχη ιδιότητα. Ουσιαστικά ένα σχήμα ορίζει τις λέξεις – και τη σημασία αυτών - που μπορούν να χρησιμοποιηθούν για το σχηματισμό προτάσεων. Επομένως, το σχήμα είναι το αντίστοιχο ενός λεξικού που περιέχει όλες τις λέξεις που μπορούμε να χρησιμοποιήσουμε καθώς και τη σημασία τους.

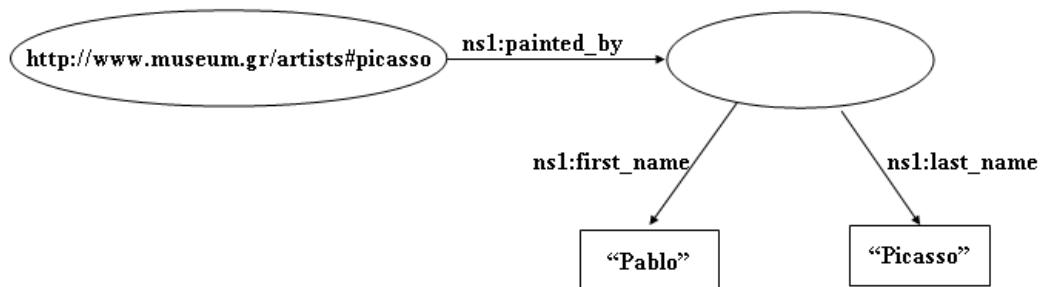
Σύμφωνα με τα παραπάνω γίνεται σαφές ότι, απαραίτητη προϋπόθεση για την χρήση του RDF, για την ανταλλαγή και τον διαμοιρασμό πληροφορίας με σημασιολογικά σαφή τρόπο, είναι η ύπαρξη κοινώς αποδεκτών και σαφώς ορισμένων σχημάτων. Παρακάτω θα γίνει αναλυτική αναφορά στα βασικά στοιχεία ορισμού τέτοιων σχημάτων, οι πλήρεις προδιαγραφές των οποίων περιγράφονται στο [BG00].

Εκτός όμως των παραπάνω απλών προτάσεων που αναφέρθηκαν πολλές φορές είναι απαραίτητη η αναπαράσταση πιο πολύπλοκων προτάσεων.

Για παράδειγμα, η πρόταση

“*Το <http://www.museum.gr/paintings#guernica> ζωγραφίστηκε (ns1:paints) από το άτομο, με όνομα Pablo(ns1:first_name) και επώνυμο(ns1:last_name) Picasso*”.

Η πρόταση αυτή απεικονίζεται στο σχήμα 2.2. Προκειμένου να περιγραφεί το άτομο με όνομα Pablo και επώνυμο Picasso είναι απαραίτητη η εισαγωγή ενός νέου πόρου – το literal δεν αρκεί – και η απόδοση ιδιοτήτων σε αυτόν. Ο νέος αυτός πόρος ονομάζεται *ανώνυμος πόρος*, και αναπαριστά την οντότητα του ατόμου που έχει ζωγραφίσει τον πίνακα. Το αναγνωριστικό του ανώνυμου πόρου είναι άγνωστο και για το λόγο αυτό χρησιμοποιείται μία κενή έλλειψη για την αναπαράστασή του. Σε επίπεδο εφαρμογής είναι απαραίτητη η απόδοση κάποιου μοναδικού αναγνωριστικού σε όλους τους ανώνυμους πόρους ώστε να είναι δυνατή η επεξεργασία και η διαφοροποίηση αυτών.



Σχήμα 2.2 Γράφος RDF με δομημένη τιμή

Παρακάτω θα αναφερθούμε σε ένα σύνολο πόρων και ιδιοτήτων που επιτρέπουν την αναπαράσταση πιο πολύπλοκων εκφράσεων.

2.4.1 RDF Συλλογές

Συχνά είναι απαραίτητη η αναφορά σε μία συλλογή από πόρους, για παράδειγμα όταν θέλουμε να πούμε ότι μια εργασία δημιουργήθηκε από περισσότερα από ένα άτομα ή να φτιάξουμε ένα κατάλογο με τους φοιτητές που παρακολουθούν ένα μάθημα. Για την διατήρηση λιστών που περιέχουν πόρους ή literals, χρησιμοποιούνται οι Συλλογές RDF (RDF Containers).

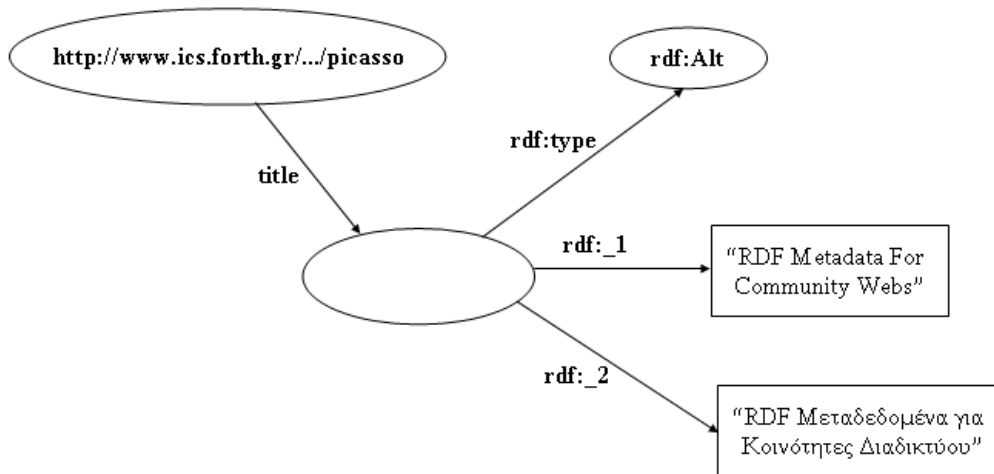
Στο RDF ορίζονται τρία είδη συλλογών τα: Bag, Sequence και Alternative:

Bag
(Πολυσύνολο) Καλείται μία μη ταξινομημένη λίστα από πόρους ή literals. Χρησιμοποιείται για να δηλώσει ότι μία ιδιότητα έχει πολλαπλές τιμές και δεν έχει σημασία η σειρά των τιμών αυτών. Ένα πολυσύνολο μπορεί να περιέχει τους μαθητές μίας τάξης αφού δεν μας ενδιαφέρει η σειρά τους. Τα πολυσύνολα μπορεί να περιέχουν πολλές φορές τα ίδια μέλη χωρίς να δημιουργείται πρόβλημα.

Sequence
(Ακολουθία) Καλείται μία ταξινομημένη λίστα από πόρους ή literals. Χρησιμοποιείται για να δηλώσει ότι μια ιδιότητα έχει πολλαπλές τιμές και η **σειρά** των τιμών αυτών έχει σημασία. Η *ακολουθία* μπορεί να χρησιμοποιηθεί, για παράδειγμα, για μία λίστα ονομάτων των εργαζομένων μιας εταιρείας με αλφαβητική σειρά. Η ακολουθία μπορεί να περιέχει επαναλαμβανόμενες τιμές.

Alternative
(Παραλλαγή) Καλείται μία λίστα από πόρους ή literals που αναπαριστούν εναλλακτικές τιμές για τιμή μιας ιδιότητας. Η παραλλαγή μπορεί να περιέχει για παράδειγμα, τον τίτλο ενός βιβλίου μεταφρασμένο σε διάφορες γλώσσες ή μία λίστα από εναλλακτικούς δικτυακούς τόπους από τους οποίους μπορεί να ανακτηθεί ένας πόρος (π.χ. πρόγραμμα). Οι εφαρμογές που διαχειρίζονται παραλλαγές «γνωρίζουν» ότι μπορούν να επιλέξουν οποιαδήποτε από τα μέλη της λίστας σαν τιμή της ιδιότητας. Το πρώτο μέλος μιας παραλλαγής αποτελεί την προκαθορισμένη τιμή, αλλά μπορεί να επιλεγθεί οποιοδήποτε άλλο μέλος της παραλλαγής ως τιμή της ιδιότητας.

Για την αναπαράσταση μιας συλλογής πόρων, το RDF χρησιμοποιεί ένα επιπλέον πόρο που προσδιορίζει το είδος της συγκεκριμένης συλλογής. Ο πόρος αυτός πρέπει να δηλώνεται ως στιγμιότυπο ενός από τους τύπους αντικειμένων συλλογών που ορίζονται παραπάνω (Bag, Sequence, Alt). Η ιδιότητα *type*, χρησιμοποιείται για να γίνει η παραπάνω δήλωση. Η σχέση μέλους μεταξύ του πόρου της συλλογής και των πόρων που αποτελούν τα μέλη της συλλογής αυτής ορίζεται από ένα σύνολο ιδιοτήτων που ορίζονται ειδικά για το σκοπό αυτό. Αυτές οι ιδιότητες μέλους ονομάζονται απλά "_1", "_2", "_3", κτλ. Οι πόροι συλλογών μπορεί να έχουν και επιπλέον ιδιότητες πλην αυτών των ιδιοτήτων μέλους και της ιδιότητας *type*. Οποιαδήποτε άλλη επιπλέον ιδιότητα περιγράφει τη συλλογή.



Σχήμα 2.3 Παράδειγμα RDF Παραλλαγής

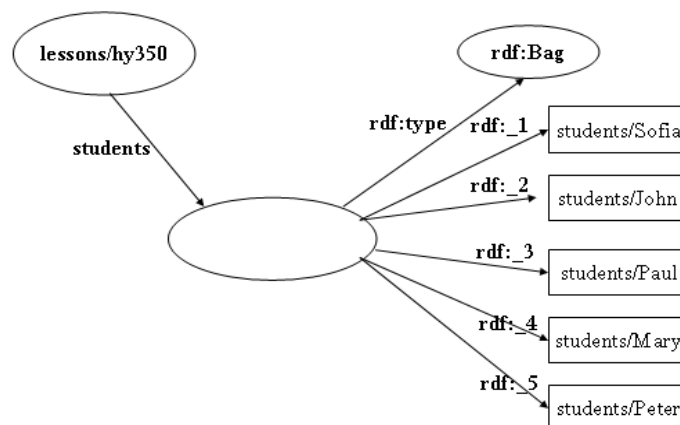
Στο σχήμα 2.3 απεικονίζεται μια παραλλαγή που έχει σαν μέλη της τον τίτλο μιας δημοσίευσης μεταφρασμένο σε δύο γλώσσες. Στην αναπαράσταση αυτή παρατηρούμε ότι η τιμή της ιδιότητας *rdf:type* είναι `rdf:Alt`, δηλαδή παραλλαγή (Alternative). Τα μέλη συλλογής αποδίδονται στον πόρο της συλλογής με τις ιδιότητες `_1`, `_2`, `_3` ... οι οποίες έχουν οριστεί για το σκοπό αυτόν.

Μία κοινή χρήση των πολυσυνόλων είναι ως η τιμή μιας ιδιότητας. Όταν χρησιμοποιείται με αυτόν τον τρόπο, η πρόταση έχει πάλι μόνο ένα αντικείμενο ανεξάρτητα από τον αριθμό των μελών του πολυσυνόλου. Ο πόρος που αναπαριστά το πολυσύνολο αποτελεί το αντικείμενο της πρότασης.

Για παράδειγμα, η πρόταση

Οι μαθητές του μαθήματος HY350 είναι οι Sofia, John, Paul, Mary, και Peter.

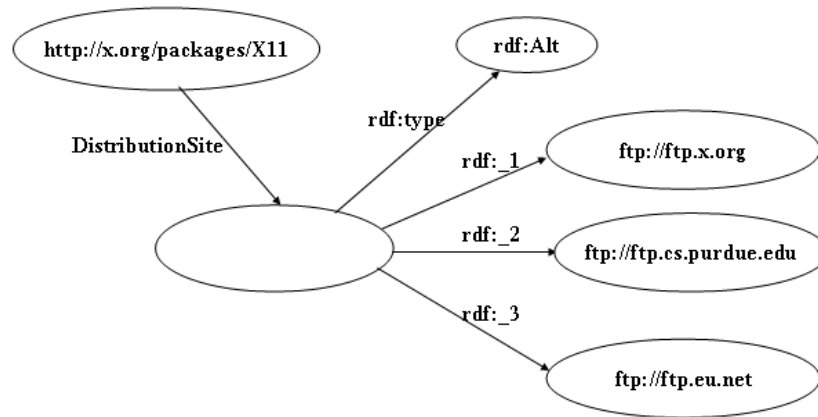
αναπαριστάνεται στο μοντέλο του RDF με το σχήμα 2.4:



Σχήμα 2.4 RDF γράφος με τιμή ιδιότητας ένα πολυσύνολο

Η πρόταση

Ο πηγαίος κώδικας για τα X11 μπορεί να βρεθεί στα *ftp.x.org*, *ftp.cs.purdue.edu*, ή *ftp.eu.net*.
μοντελοποιείται στην RDF με τη βοήθεια του σχήματος 2.5.



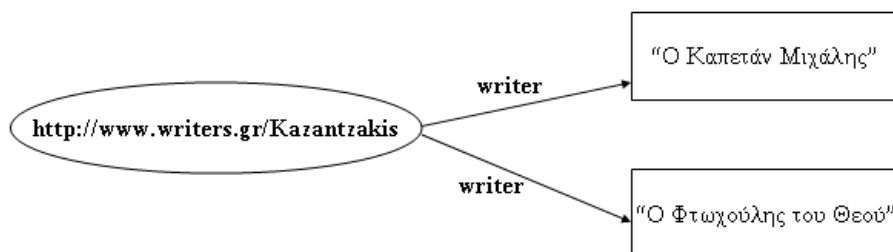
Σχήμα 2.5 Γράφος με τιμή ιδιότητας μία Παραλλαγή

Η λειτουργικότητα τόσο των ακολουθιών όσο και των συλλογών είναι εμφανής. Ωστόσο από τα παραπάνω ίσως δεν είναι φανερή η διαφορά μεταξύ πολυσυνόλων και επαναλαμβανόμενων ιδιοτήτων (χρήση της ίδιας ιδιότητας πολλαπλές φορές). Η διαφορά μεταξύ πολυσυνόλων και επαναλαμβανόμενων ιδιοτήτων είναι ότι στις επαναλαμβανόμενες ιδιότητες οι διαφορετικές τιμές της ιδιότητας δεν έχουν σχέση μεταξύ τους. Ας εξετάσουμε ένα παράδειγμα για γίνει περισσότερο κατανοητή η διάκριση αυτή.

Έστω η πρόταση:

Ο Καζαντζάκης έχει γράψει τα βιβλία Καπετάν Μιχάλης και Ο Φτωχούλης του Θεού.

Η μοντελοποίηση της παραπάνω πρότασης φαίνεται στο παρακάτω σχήμα (Σχήμα 2.6).

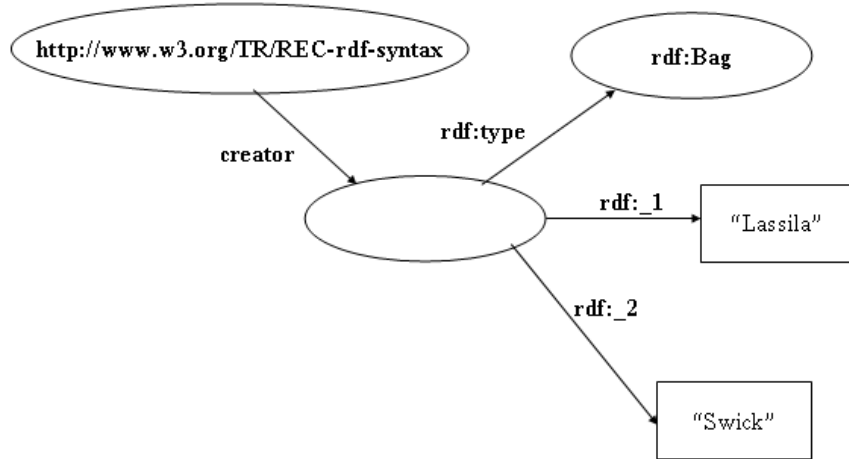


Σχήμα 2.6 RDF γράφος με πλειότιμες ιδιότητες

Τα παραπάνω βιβλία δεν έχουν κάποια ιδιαίτερη σχέση, πλην του ότι έχουν γραφεί από τον ίδιο συγγραφέα. Αντίθετα, τα πολυσύνολα χρησιμοποιούνται όταν υπάρχει ιδιαίτερη σημασιολογία μεταξύ των τιμών της ιδιότητας. Για παράδειγμα, η πρόταση:

Το RDF&MS έχει γραφεί από τους Lassila και Stick

μοντελοποιείται με τη βοήθεια του πολυσυνόλου ακριβώς επειδή πρέπει να εκφραστεί ότι το RDF&MS αποτελεί το αποτέλεσμα συλλογικής δουλειάς και όχι του καθένα ξεχωριστά. Ο γράφος με τον οποίο μοντελοποιείται η παραπάνω πρόταση φαίνεται στο σχήμα 2.7.



Σχήμα 2.7 RDF γράφος με τιμή ιδιότητας ένα πολυσύνολο

2.4.2 Υποστασιοποιημένες Δηλώσεις (Reified Statements)

Συχνά προκύπτει το πρόβλημα αν ένα πακέτο μεταδεδομένων είναι αξιόπιστο. Η RDF λύνει το πρόβλημα αυτό χωρίς να εισάγει κάποιο λεξιλόγιο, που θα έπρεπε κάποιος να το ξαναεπινοήσει για κάθε επιπλέον επίπεδο, αλλά με μία δομή η οποία δημιουργεί αυτόματα πόρους πρώτης κλάσης από δεύτερης κλάσης. Αυτή η διαδικασία του RDF ονομάζεται Υποστασιοποίηση (Reification). Με τη βοήθεια της υποστασιοποίησης είναι δυνατή η περιγραφή των ίδιων των μεταδεδομένων. Για παράδειγμα, μέσω της υποστασιοποίησης μπορεί να εκφραστεί η πηγή μίας δήλωσης ή η χρονική στιγμή διατύπωσης αυτής.

Για να περιγραφεί μία δήλωση πρέπει πρώτα να δημιουργηθεί η υποστασιοποιημένη δήλωση της. Μία υποστασιοποιημένη δήλωση είναι ένας νέος πόρος ο οποίος χαρακτηρίζεται από τέσσερις υποχρεωτικές ιδιότητες, τις: ***rdf:subject***, ***rdf:predicate***, ***rdf:object*** και ***rdf:type***.

Η τιμή της ιδιότητας ***rdf:subject*** είναι το αντίστοιχο υποκείμενο (subject) της πρότασης που υποστασιοποιείται.

Η τιμή της ιδιότητας ***rdf:predicate*** είναι το αντίστοιχο κατηγορούμενο (predicate) της πρότασης που υποστασιοποιείται.

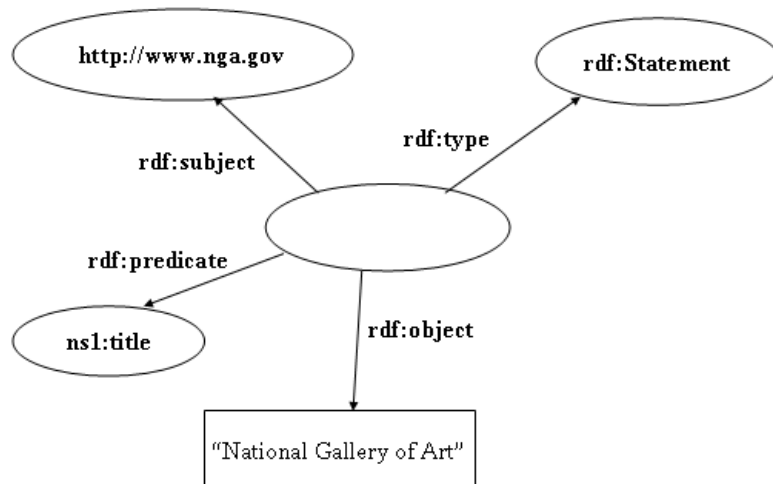
Η τιμή της ιδιότητας ***rdf:object*** είναι το αντίστοιχο αντικείμενο (object) της πρότασης που υποστασιοποιείται.

Η τιμή της ιδιότητας ***rdf:type***, για κάθε πόρο που αντιστοιχεί σε μία υποστασιοποιημένη δήλωση, είναι η κλάση Statement.

Ας εξετάσουμε τώρα το γράφο αναπαράστασης μίας υποστασιοποιημένης δήλωσης. Για παράδειγμα, η πρόταση:

“Ο τίτλος της σελίδας με URI <http://www.nga.org> είναι National Gallery Of Art”.

μοντελοποιείται ως εξής (Σχήμα 2.8):



Σχήμα 2.8 Υποστασιοποιημένη Δήλωση

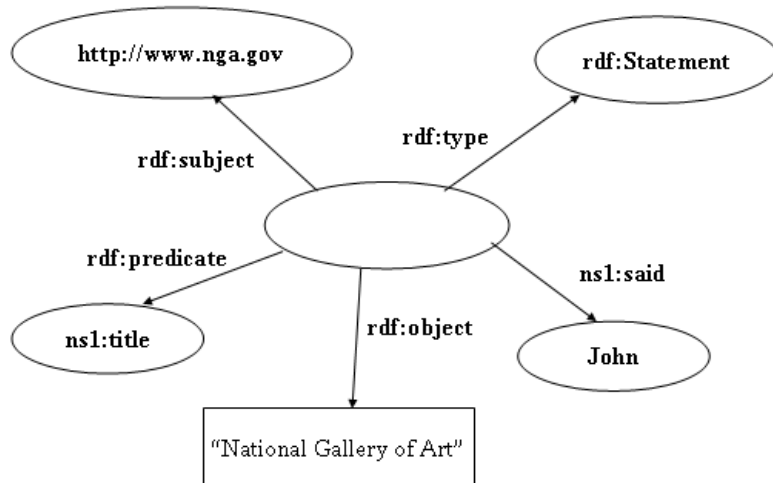
Μία υποστασιοποιημένη δήλωση μπορεί να χρησιμοποιηθεί τόσο σαν υποκείμενο όσο και σαν αντικείμενο μίας άλλης δήλωσης.

Η αναπαράσταση της πρότασης:

Ο John είπε ότι “Ο τίτλος της σελίδας με URI <http://www.nga.org> είναι National Gallery Of Art”.

μοντελοποιείται στο σχήμα 2.9

Πρέπει να τονιστεί ότι η υποστασιοποιημένη δήλωση δεν υποκαθιστά την δήλωση. Σε ένα γράφο μπορούν να υπάρχουν και οι δύο ή μόνο μία από αυτές. Ωστόσο ένας RDF γράφος εκφράζει ένα γεγονός μόνο όταν η δήλωση που το εκφράζει περιέχεται στο γράφο ανεξάρτητα από το αν υπάρχει η υποστασιοποιημένη δήλωση ή όχι.



Σχήμα 2.9 Υποστασιοποιημένη δήλωση στην οποία έχει αποδοθεί η ιδιότητα ns1:said

2.5 Γλώσσα Περιγραφής Σχημάτων (RDF/S)

Το Πλαίσιο Περιγραφής Πόρων (RDF) αποτελεί ένα απλό μοντέλο για την περιγραφή σχέσεων μεταξύ πόρων, με τη βοήθεια ιδιοτήτων και τιμών. Οι RDF ιδιότητες μπορούν να θεωρηθούν ως χαρακτηριστικά των πόρων και με αυτή την έννοια αντιστοιχούν στα «παραδοσιακά» ζεύγη χαρακτηριστικών-τιμών. Επίσης, οι ιδιότητες του RDF αναπαριστούν σχέσεις μεταξύ των πόρων. Αυτό συνεπάγεται, ότι το μοντέλο δεδομένων του RDF μπορεί να παρομοιαστεί με ένα διάγραμμα οντοτήτων – συσχετίσεων. Παρόλα αυτά, το μοντέλο δεδομένων του RDF από μόνο του, δεν παρέχει μηχανισμό ούτε για την περιγραφή των ιδιοτήτων, ούτε για την περιγραφή των σχέσεων μεταξύ των ιδιοτήτων και άλλων πόρων.

Η ανάγκη, λοιπόν, ενός μηχανισμού για την περιγραφή τόσο των ιδιοτήτων όσο και των σχέσεων μεταξύ των ιδιοτήτων οδήγησε στο να προταθεί μία γλώσσα ορισμού σχημάτων (συνόλων κλάσεων και ιδιοτήτων), η RDF Schema Specification Language (RDFS). Η RDFS παρέχει ένα σύνολο βασικών τύπων (κλάσεων και ιδιοτήτων) που χρησιμοποιούνται για τη δημιουργία των RDF σχημάτων. Για παράδειγμα, με τη βοήθεια του RDFS, είναι δυνατός ο προσδιορισμός μηχανισμών καθορισμού κλάσεων πόρων, ο περιορισμός των πιθανών συνδυασμών κλάσεων και σχέσεων και η ανίχνευση παραβίασης αυτών των περιορισμών. Το *RDF Schema* ορίζεται σε σχέση με το βασικό πληροφοριακό μοντέλο του RDF – μία δομή γράφου που περιγράφει πόρους και ιδιότητες. Όλα τα λεξιλόγια του RDF διαμοιράζονται κάποιες βασικές κοινές δομές, περιγράφουν κλάσεις πόρων και τύπους σχέσεων μεταξύ των ιδιοτήτων. Αυτός ο διαμοιρασμός κοινών κλάσεων και ιδιοτήτων επιτρέπει ένα καλύτερο συνδυασμό λεξιλογίων και απευθύνεται στην ανάγκη δημιουργίας μεταδεδομένων στα οποία οι προτάσεις μπορούν να επεξεργαστούν από πολλαπλά λεξιλόγια που διαχειρίζονται με ένα αποκεντριοποιημένο τρόπο από τις ανεξάρτητες κοινότητες.

Τα RDF σχήματα ορίζονται χρησιμοποιώντας τους τύπους που παρέχει το RDFS και βασίζονται στο μοντέλο δεδομένων του RDF. Επομένως, και τα σχήματα αποτελούν

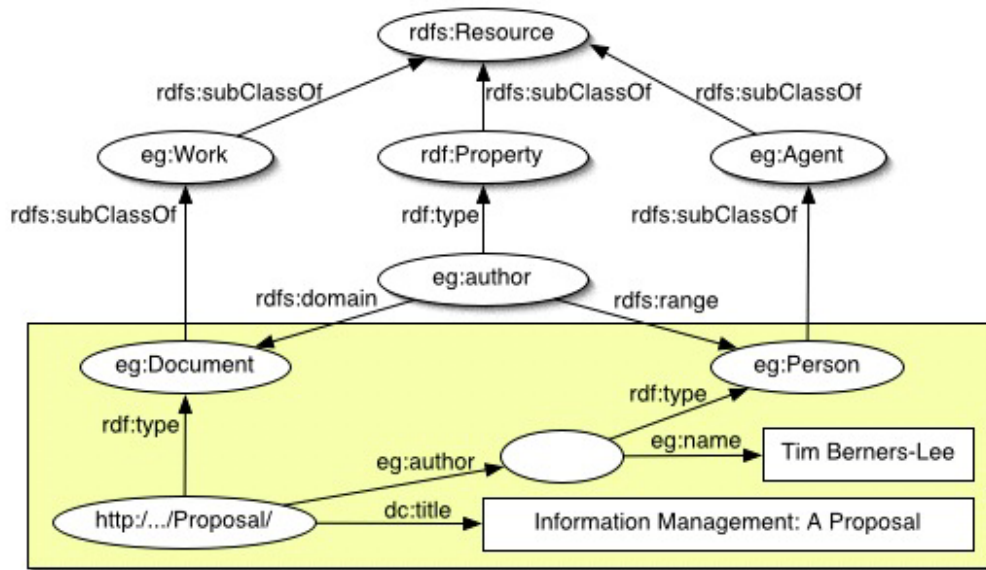
μεταδεδομένα. Το γεγονός αυτό παρέχει ευελιξία στις εφαρμογές καθώς αυτές δεν είναι απαραίτητο να γνωρίζουν εκ των προτέρων την σημασιολογία των σχημάτων στην οποία στηρίζονται τα μεταδεδομένα που αναλύουν αλλά μπορούν να την εξάγουν.

Οι κοινότητες περιγραφής πόρων έχουν την απαίτηση να μπορούν να λένε συγκεκριμένα πράγματα για συγκεκριμένες κατηγορίες πόρων. Για παράδειγμα, για την περιγραφή βιβλιογραφικών πόρων, είναι κοινά τα περιγραφικά χαρακτηριστικά: «συγγραφέας», «τίτλος» και «κατηγορία». *Η δήλωση των ιδιοτήτων (χαρακτηριστικών) και η σημασιολογία τους ορίζονται στο πλαίσιο του RDF σαν ένα RDF Schema.* Ένα RDF σχήμα δεν ορίζει μόνο τις ιδιότητες των πόρων (π.χ. τίτλος, συγγραφέας, κατηγορία, μέγεθος, χρώμα, κτλ.) αλλά μπορεί να ορίζει και τα είδη των πόρων που περιγράφονται (βιβλία, ιστοσελίδες, άνθρωποι, εταιρίες, κα).

Η προσέγγιση του λεξιλογίου περιγραφής (vocabulary description) στο RDF Schema επιτρέπει στους σχεδιαστές του λεξιλογίου να αναπαραστήσουν περιγραφές κλάσεων και ιδιοτήτων στον Παγκόσμιο Ιστό. Για παράδειγμα, με την περιγραφή τρόπων με τους οποίους συνδυασμοί κλάσεων, ιδιοτήτων και τιμών μπορούν να συνδυαστούν μεταξύ τους για ένα συγκεκριμένο σκοπό, ώστε το αποτέλεσμα να βγάζει νόημα.

Ας εξετάσουμε ένα παράδειγμα σε διαγραμματική μορφή, όπου γίνεται εμφανής η χρήση του λεξιλογίου του RDF Schema για την περιγραφή κλάσεων και ιδιοτήτων, καθώς και η σύνδεση με τα δεδομένα σε επίπεδο εφαρμογής.

Το παράδειγμα του σχήματος 2.10 δείχνει τον τρόπο με τον οποίο το RDF μπορεί να χρησιμοποιηθεί προκειμένου να περιγράψει αντικείμενα του πραγματικού κόσμου (έγγραφα, άτομα), τις κλάσεις στις οποίες ανήκουν (π.χ eg:Document, eg:Person), και τις ιδιότητες που χρησιμοποιούνται για να συσχετιστούν μέλη αυτών των κλάσεων – για το παρόν παράδειγμα η ιδιότητα eg:author. Μέσω της χρήσης του RDF Schema, μπορούν να περιγραφούν οι σχέσεις μεταξύ ιδιοτήτων του RDF (όπως η ιδιότητα eg:author) και κλάσεων πόρων. Στο παρόν παράδειγμα, χρησιμοποιείται το RDF Schema για να δηλώσει ότι η ιδιότητα eg:author συσχετίζει έγγραφα με άτομα. Επίσης, δηλώνεται ότι όλα τα έγγραφα θεωρούνται ως «εργασίες» (works) και όλα τα άτομα (people) ως πράκτορες (agents).

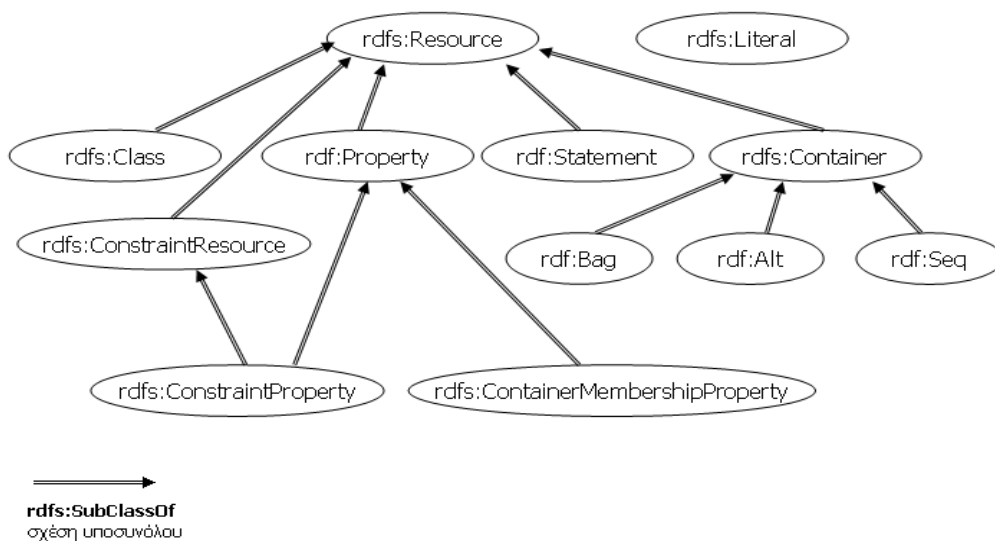


Σχήμα 2.10 Παράδειγμα RDFS σχήματος

Στις παρακάτω παραγράφους θα περιγραφεί το σύστημα τύπων που ορίζει η RDFS καθώς επίσης και οι πόροι που ορίζονται στο RDF&MS που περιγράφονται στο RDF σχήμα και χρησιμοποιούνται για τον ορισμό σχημάτων.

2.4.1 Βασικές Κλάσεις του RDF/S

Στην παράγραφο αυτή θα αναλυθούν οι κλάσεις που είτε ορίζονται είτε περιγράφονται από το RDF σχήμα. Οι κλάσεις αυτές σχετίζονται μεταξύ τους με σχέσεις υποσυνόλου/υπερσυνόλου. Η ιεραρχία των κλάσεων απεικονίζεται στο σχήμα 2.11.



Σχήμα 2.11 Ιεραρχία RDF/S Κλάσεων

Στην κορυφή της ιεραρχίας αυτής βρίσκεται η κλάση *Resource*, και χαμηλότερα σε αυτήν είναι οι κλάσεις : *Class*, *ConstraintResource*, *ConstraintProperty*, *Container*, *ContainerMembershipProperty*, *Literal*, *Property*, *Bag*, *Seq*, *Alt* και *Statement*.

Θα αναφερθούμε αναλυτικότερα σε καθεμία από αυτές:

rdfs:Resource

Οτιδήποτε μπορεί να περιγραφεί με τη βοήθεια RDF δηλώσεων αποτελεί μέρος της κλάσης αυτής. Η *rdfs:Resource* αντιστοιχεί στο πεδίο Πόροι.

rdfs:Class

Κάθε πόρος που αντιπροσωπεύει μία RDF κλάση πρέπει να δηλώνεται ως μέλος της *Rdfs:Class*, δηλαδή να αποδίδεται σε αυτόν η ιδιότητα *rdf:type*, με τιμή *rdfs:Class*. Η σημασιολογία του *rdfs:Class* είναι αντίστοιχη με αυτή των γενικών εννοιών *Τύπος*, *Κατηγορία* ή *Κλάση* στις Οντοκεντρικές Γλώσσες όπως η Java. Οτιδήποτε μπορεί να περιγραφεί με τη βοήθεια RDF δηλώσεων αποτελεί μέρος της κλάσης αυτής. Η *rdfs:Resource* αντιστοιχεί στο πεδίο Πόροι. Οι κλάσεις του RDF είναι δυνατόν να αναπαριστούν οτιδήποτε, όπως για παράδειγμα, βιβλία, ανθρώπους ή σελίδες του διαδικτύου.

rdfs:ConstraintResource

Η κλάση αυτή αποτελεί υποκλάση της *rdfs:Resource*. Μέλη της είναι τόσο οι κλάσεις όσο και οι ιδιότητες που χρησιμοποιούνται προκειμένου να εκφράσουν περιορισμούς. Αυτή η κλάση παρέχει στους επεξεργαστές RDF την δυνατότητα ελέγχου της συνέπειας των RDF μεταδεδομένων. Δεδομένης της μη ύπαρξης μηχανισμού που επιτρέπει την ανακάλυψη και τη σωστή ερμηνεία περιορισμών που δεν έχουν οριστεί στο RDFS, όπως για παράδειγμα η πληθικότητα μίας ιδιότητας. Όταν ένας επεξεργαστής RDF διαπιστώσει άγνωστα σε αυτόν μέρη της κλάσης *rdfs:ConstraintResource* τότε δεν είναι σε θέση να αποφανθεί για τη συνέπεια των μεταδεδομένων αυτών.

rdfs:ConstraintProperty

Η κλάση αυτή είναι υποκλάση τόσο της *rdfs:Property* όσο και της *rdfs:ConstraintResource*. Μέλη της είναι όλες οι ιδιότητες που δηλώνουν κάποιο περιορισμό. Οι ιδιότητες *rdfs:range* και *rdfs:domain* αποτελούν μέλη της κλάσης αυτής και περιορίζουν το πεδίο τιμών και το πεδίο ορισμού μίας ιδιότητας.

rdfs:Container

Η κλάση αυτή αντιπροσωπεύει το σύνολο των *συλλογών πόρων* (*containers*). Οι κλάσεις *rdf:Bag*, *rdf:Alt* και *rdf:Seq* αποτελούν υποκλάσεις της κλάσης αυτής.

rdfs:ContainerMembershipProperty

Αυτή η κλάση περιέχει τις μη αριθμημένες ιδιότητες *rdf:_1*, *rdf:_2*, *rdf:_3* ... οι οποίες δηλώνουν τα μέλη μίας συλλογής και αποτελεί υποκλάση της *rdfs:Property*.

rdfs:Literal

Η κλάση αυτή αντιπροσωπεύει το σύνολο *Literal* και περιέχει ατομικές τιμές όπως αλφαριθμητικά και ακεραίους.

Οι κλάσεις οι οποίες ορίζονται στο RDF M&S και περιγράφονται στο RDF Schema είναι οι παρακάτω:

rdf:Property

Η κλάση αυτή αναπαριστάνει το σύνολο των πόρων που είναι ιδιότητες.

rdf:Bag, rdf:Alt, rdf:Seq

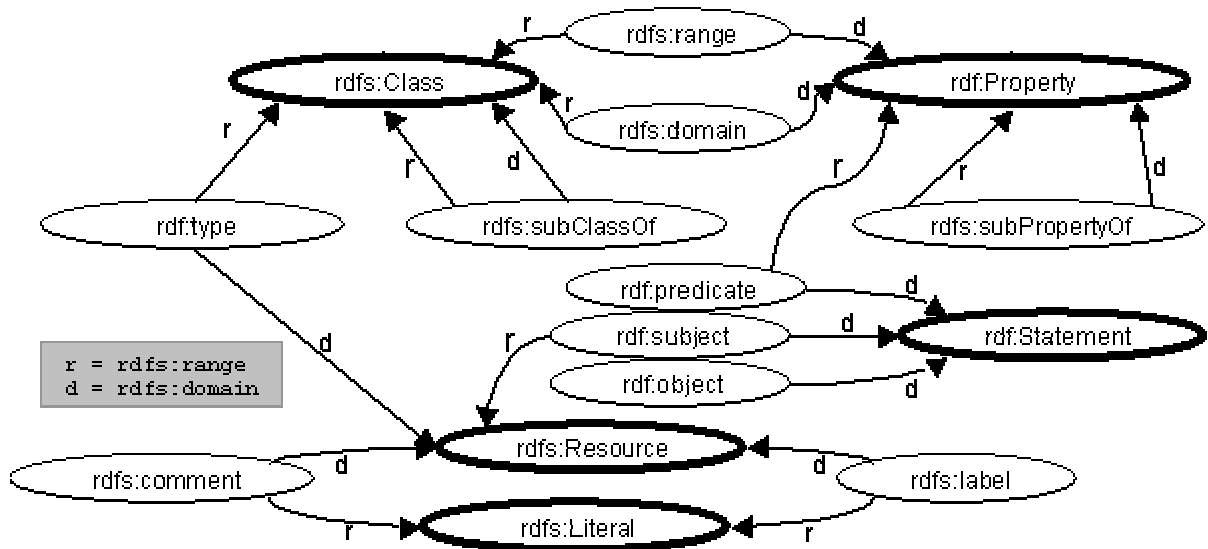
Οι κλάσεις αυτές αναπαριστούν τα σύνολα Bag, Alternative και Sequence αντίστοιχα και αποτελούν υποκλάσεις της Rdfs:Container.

Rdf:Statement

Αυτή η κλάση αναπαριστά το σύνολο των υποστασιοποιημένων δηλώσεων.

2.4.1 Βασικές RDF/S Ιδιότητες

Στην παράγραφο αυτή θα αναλυθούν οι βασικές ιδιότητες που είτε ορίζονται είτε περιγράφονται στο RDF σχήμα.



Σχήμα 2.12 Περιορισμοί που δηλώνονται στο RDF Σχήμα

Στο παραπάνω σχήμα (Σχήμα 2.12) απεικονίζονται οι περιορισμοί που δηλώνονται στο RDF σχήμα για τις ιδιότητες.

Rdfs:domain

Η κλάση αυτή αποτελεί μέλος της κλάσης *rdfs:ConstraintProperty* και χρησιμοποιείται για να ορίσει τις κλάσεις στις οποίες μπορεί να εφαρμοστεί μία ιδιότητα. Μία ιδιότητα μπορεί να έχει καμία, μία ή περισσότερες *rdfs:domain* ιδιότητες. Στην περίπτωση που δεν ορίζεται η ιδιότητα *rdfs:domain* για μία ιδιότητα, η ιδιότητα αυτή είναι δυνατόν να αποδοθεί σε οποιονδήποτε πόρο. Αν αποδίδονται παραπάνω από μία *rdfs:domain* ιδιότητες σε μία ιδιότητα τότε η ιδιότητα αυτή μπορεί να αποδοθεί σε οποιοδήποτε μέλος των κλάσεων που αποτελεί τιμή της *rdfs:domain*.

Το γεγονός ότι μια ιδιότητα μπορεί να έχει πολλαπλές `rdfs:domain` ιδιότητες οι οποίες είναι διασκορπισμένες σε διάφορα αρχεία δημιουργεί ασάφεια στη σημασιολογία των ιδιοτήτων. Έστω ότι σε ένα σχήμα ορίζεται η ιδιότητα *name*. Σε ένα σχήμα A ορίζεται σαν πεδίο ορισμού της ιδιότητας η κλάση `Person` και σε ένα σχήμα B η ιδιότητα `Company`. Το γεγονός ότι αποδίδεται διαφορετική τιμή στην ιδιότητα `rdfs:domain` σημαίνει ότι η σημασιολογία της ιδιότητας σε καθένα από τα παραπάνω σχήματα είναι διαφορετική. Στο σχήμα A, η ιδιότητα *name* δηλώνει το όνομα ενός ανθρώπου, ενώ στο σχήμα B, η ίδια ιδιότητα δηλώνει το όνομα μίας εταιρείας. Η σημασιολογία μιας ιδιότητας επεκτείνεται καθώς προστίθεται σε αυτήν μία καινούργια `rdfs:domain` ιδιότητα. Εφόσον η σημασιολογία μιας ιδιότητας μπορεί να επεκταθεί αυθαίρετα συμπεραίνουμε ότι και η σημασιολογία των μεταδεδομένων δεν είναι προκαθορισμένη.

Η δυνατότητα πολλαπλών `rdfs:domain` ιδιοτήτων, οι οποίες μπορεί να είναι διασκορπισμένες σε πολλά σχήματα, σε συνδυασμό με το γεγονός ότι το πεδίο ορισμού μιας ιδιότητας είναι η ένωση των τιμών της `rdfs:domain` δεν επιτρέπει την εξαγωγή συμπερασμάτων που αφορούν στην κλάση που ανήκει ένας πόρος στον οποίο αποδίδεται η ιδιότητα. Για παράδειγμα, αν έχουμε μία τριάδα $r(x,y)$ δεν μπορούμε να εξάγουμε την κλάση ή τις κλάσεις στις οποίες ανήκει ο πόρος x εφόσον το πεδίο ορισμού της ιδιότητας r δεν είναι συγκεκριμένο.

Αυτό συνεπάγεται ότι υπάρχει πρόβλημα με τις `rdfs:domain` ιδιότητες. Ωστόσο, δεν αναφέρεται στα κείμενα του RDF ο λόγος ύπαρξης ιδιοτήτων με πολλαπλές `rdfs:domain` ιδιότητες αντί για πολλαπλές διαφορετικές ιδιότητες.

Rdfs:range

Η ιδιότητα αυτή είναι μέλος της κλάσης `rdfs:ConstraintProperty` και χρησιμοποιείται για να περιορίσει το πεδίο τιμών των ιδιοτήτων. Η τιμή της `rdfs:range` είναι πάντα μία κλάση. Αν το πεδίο τιμών μιας ιδιότητας X ορίζεται να είναι η κλάση A τότε κάθε τιμή της ιδιότητας X πρέπει να ανήκει στην κλάση A . Μια ιδιότητα μπορεί να έχει το πολύ μία `rdfs:range` ιδιότητα. Σε περίπτωση που πρέπει να οριστεί μια ιδιότητα, για παράδειγμα *δημιουργός* που έχει σαν πεδίο τιμών δύο κλάσεις π.χ. *Ζωγράφος* και *Γλύπτης*, τότε αν υπάρχει κάποια ήδη ορισμένη υπερ-κλάση τους που είναι αποδεκτή σαν πεδίο τιμών, θέτουμε την κλάση αυτή σαν τιμή της `rdfs:range`. Διαφορετικά, δημιουργούμε μία υπερκλάση αυτών. Είναι δυνατόν σε μία κλάση να μην αποδοθεί καθόλου η ιδιότητα `rdfs:range`. Στην περίπτωση αυτή η ιδιότητα μπορεί να έχει οποιαδήποτε τιμή. Το πεδίο τιμών της δηλαδή είναι η ένωση των συνόλων `Pόροι` και `Literals`.

Παρατηρούμε από το σχήμα 2.12 ότι η τιμή της ιδιότητας `rdfs:range` για την ιδιότητα `rdfs:domain` είναι η κλάση `rdfs:Class`. Αυτός συνεπάγεται ότι μια ιδιότητα μπορεί να έχει ως πεδίο ορισμού μία οποιαδήποτε κλάση. Αλλά η δήλωση αυτή δεν είναι απόλυτα σωστή καθώς η κλάση `rdfs:Literal` που ανήκει στην κλάση `rdfs:Class` δεν μπορεί να είναι τιμή της ιδιότητας `rdfs:domain`.

Επίσης, η μη ρητή δήλωση του πεδίου τιμών μιας ιδιότητας στο πεδίο ορισμού της προκαλεί προβλήματα. Είναι δυνατόν σε σχήματα να οριστούν διαφορετικά πεδία τιμών για την ιδιότητα. Το γεγονός αυτό δημιουργεί ασάφεια στην σημασιολογία των μεταδεδομένων. Επίσης, το γεγονός ότι η ιδιότητα μπορεί να έχει σαν τιμές τόσο

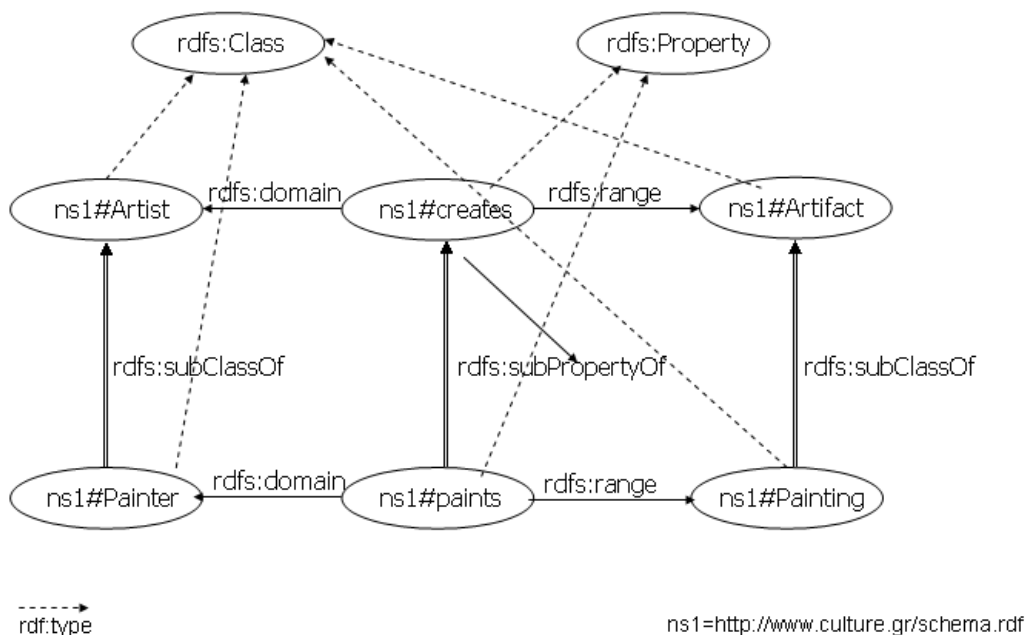
πόρους όσο και literals κάνει πιο πολύπλοκη την επεξεργασία των περιγραφών. Το βασικό πρόβλημα ωστόσο είναι ότι παραβιάζεται ο περιορισμός της μοναδικότητας της ιδιότητας `rdfs:range` δεδομένου ότι στο RDF δεν ορίζεται κοινή υπερκλάση των κλάσεων `rdfs:Resource` και `rdfs:Literal`.

Rdfs:type

Η ιδιότητα αυτή χρησιμοποιείται για να δείξει ότι ένας πόρος είναι μέλος μιας κλάσης. Ορίζεται όπως κάθε άλλη επώνυμη ιδιότητα σε ένα σχήμα RDFS, όπως φαίνεται στο σχήμα 2.12. Όταν ένας πόρος έχει ως τιμή της ιδιότητας `rdf:type` μια συγκεκριμένη κλάση, ο πόρος αυτός είναι στιγμιότυπο της κλάσης αυτής. Για τον ορισμό μιας συγκεκριμένης κλάσης χρησιμοποιούμε ως `rdf:type` τον τύπο `rdfs:Class`. Οι ιδιότητες έχουν επίσης έναν ορισμένο τύπο, τον `rdf:Property`, ενώ οι υποστασιοποιημένες δηλώσεις είναι του τύπου `rdf:Statement`. Ένας πόρος μπορεί επίσης να οριστεί ως αντικείμενο περισσοτέρων από μία κλάσεων, όχι αναγκαστικά συνδεδεμένων με τη σχέση εξειδίκευσης (πολλαπλή κατάταξη σε κλάσεις – multiple classification), δεν μπορεί όμως να οριστεί ταυτόχρονα και σαν κλάση και σαν ιδιότητα.

Rdfs:SubPropertyOf

Η ιδιότητα αυτή προσδιορίζει το γεγονός ότι μία ιδιότητα είναι υπο-ιδιότητα μιας άλλης. Χαρακτηριστικά αυτής είναι ότι είναι μεταβατική, δεν επιτρέπει το σχηματισμό κύκλων, ενώ μια ιδιότητα μπορεί να είναι υπο-ιδιότητα περισσοτέρων της μιας ιδιοτήτων. Αν για παράδειγμα (σχήμα 2.13) η ιδιότητα *paints* έχει οριστεί ως υπο-ιδιότητα της γενικότερης ιδιότητας *creates*, τότε αν σε ένα πόρο έχει οριστεί η ιδιότητα *paints*, πρέπει να θεωρήσουμε ότι έχει οριστεί και η *creates* με την ίδια τιμή. Αυτό πρακτικά σημαίνει ότι αν ζητήσουμε σε μία ερώτηση τις τιμές όλων των στιγμιότυπων της *creates*, θα πρέπει να πάρουμε και τις τιμές των στιγμιότυπων της ιδιότητας *paints*. Επίσης, αν π.χ. μια εφαρμογή δεν γνωρίζει τη σημασιολογία της *paints*, θα πρέπει τουλάχιστον να την χειριστεί όπως την *creates*, με δεδομένο ότι γνωρίζει τη σημασιολογία της δεύτερης.



Σχήμα 2.13 Η σχέση creates και η υποσχέση της paints

Rdfs:seeAlso

Δηλώνει ένα πόρο ο οποίος έχει πληροφορία για τον πόρο στον οποίο εφαρμόζεται η ιδιότητα `rdfs:SeeAlso`. Μπορεί να εξειδικευτεί χρησιμοποιώντας την ιδιότητα `rdfs:subPropertyOf` σε ιδιότητες που δηλώνουν το είδος της πληροφορίας που παρέχει ο πόρος. Εξειδίκευση της ιδιότητας `rdfs:SeeAlso` είναι η ιδιότητα `rdfs:isDefinedBy` που περιγράφεται παρακάτω. Το πεδίο τιμών και το πεδίο ορισμού της ιδιότητας `rdfs:SeeAlso` είναι κλάση `rdfs:Resource`.

Rdfs:isDefinedBy

Η ιδιότητα αυτή αποτελεί υπο-ιδιότητα της `rdfs:seeAlso`. Δηλώνει τον πόρο στον οποίο ορίζεται ο πόρος που εφαρμόζεται η ιδιότητα `rdfs:isDefinedBy` είναι κλάση `rdfs:Resource`. Η `rdfs:isDefinedBy` χρησιμοποιείται κυρίως για να δηλώσει το URI του RDF σχήματος στο οποίο ορίζεται μια ιδιότητα ή κλάση. Αν και συνήθως οι δηλώσεις χώρων ονοματοδοσίας XML που υπάρχουν σε ένα αρχείο δηλώνουν το URI στο οποίο ορίζονται οι σχήμα κατασκευές, υπάρχουν περιπτώσεις που απαιτείται επιπλέον πληροφορία. Για παράδειγμα, έστω η παρακάτω δήλωση: `<rdfs:subClassOf rdf:resource = "http://purl.org/dc/elements/1.0/Creator/">`, από την παραπάνω έκφραση δεν υποδεικνύεται που ορίζεται η ιδιότητα `Creator`. Σε τέτοιες περιπτώσεις χρησιμοποιείται η ιδιότητα `isDefinedBy` για να αναπαραστήσει ρητά αυτήν την πληροφορία. Η προσέγγιση αυτή είναι κατάλληλη και σε περιπτώσεις όπου το URI ενός χώρου ονοματοδοσίας (namespace) και τα στοιχεία που δηλώνονται σε αυτό δεν έχουν άμεση σχέση.

Rdfs:Comment και rdfs:label

Σε κάποιες περιπτώσεις είναι χρήσιμο να παρέχονται και αναλυτικές (λεκτικές) περιγραφές για πόρους με τρόπο που πιθανόν να μην είναι αξιοποιήσιμος από τον

υπολογιστή αλλά είναι κατανοητός από τον άνθρωπο. Για παράδειγμα, όταν κάποιος επιθυμεί να χρησιμοποιήσει ή να επεκτείνει ένα σχήμα, τέτοιες περιγραφές είναι χρήσιμες στην κατανόηση της σημασιολογίας του ορισμού των στοιχείων του υπάρχοντος σχήματος. Πιο συγκεκριμένα, η ιδιότητα `rdfs:label` χρησιμοποιείται για να αναθέσει ένα, ανθρώπινα κατανοητό, όνομα σε ένα πόρο ενώ, η `rdfs:comment` για μια πιο αναλυτική περιγραφή αυτού. Με δεδομένο ότι οι ιδιότητες στο RDF μπορούν να έχουν πολλαπλές τιμές, οι ιδιότητες αυτές μπορούν να αποτιμηθούν και σε διάφορες γλώσσες για να καλύψουν τις ανάγκες χρηστών διαφορετικών εθνικοτήτων.

Οι ιδιότητες που ορίζονται στο RDF M&S και περιγράφονται στο RDF σχήμα είναι:

rdf:predicate, rdf:subject, rdf:object

Οι παραπάνω αντιπροσωπεύουν τις ιδιότητες κατηγορημα (predicate), θέμα (subject) και αντικείμενο (object) και χρησιμοποιούνται για το σχηματισμό της υποστασιοποιημένης δήλωσης.

rdf:value

Χρησιμοποιείται για την αναπαράσταση σχέσεων πληθικότητας μεγαλύτερης του 2, για να δηλώσει την κύρια τιμή μιας ιδιότητας.

2.6 Αναπαράσταση RDF με τριάδες

Ένας άλλος τρόπος για την αναπαράσταση περιγραφών RDF είναι με την χρήση τριάδων. Η αναπαράσταση αυτή είναι απολύτως ανάλογη με αυτή του γράφου. Παρακάτω, παρουσιάζεται ένα παράδειγμα που αντιστοιχεί στο γράφο του σχήματος 2.1 (α):

ΚΑΤΗΓΟΡΗΜΑ (ΙΔΙΟΤΗΤΑ)	[HTTP://WWW.CULTURE.GR/SCHEMA.RDF#LAST_NAME]
Υποκείμενο (Πόρος)	[http://www.museum.gr/picasso.html]
Αντικείμενο (Literal)	“Picasso”

ενώ για το γράφο 2.1(β) η ισοδύναμη τριάδα θα ήταν:

ΚΑΤΗΓΟΡΗΜΑ (ΙΔΙΟΤΗΤΑ)	[HTTP://WWW.CULTURE.GR/SCHEMA.RDF#PAINTS]
Υποκείμενο (Πόρος)	[http://www.museum.gr/picasso.html]
Αντικείμενο (Πόρος)	[http://www.museum.gr/guernica]

Η αναπαράσταση αυτή μας επιτρέπει να αποδώσουμε μια σχεσιακή ερμηνεία σε ημιδομημένα δεδομένα και σχήματα (καθώς οι τριάδες είναι απλά μια υποπερίπτωση των πλειάδων).

2.7 Συντακτικό RDF/XML

Απαραίτητη προϋπόθεση για να είναι εφικτή η ανταλλαγή μεταδεδομένων RDF, μεταξύ των υπολογιστών ή των προγραμμάτων αποτελεί η ύπαρξη ενός συντακτικού. Για το σκοπό αυτό επιλέχθηκε ένα συντακτικό XML [BPS98], με δεδομένα τα πλεονεκτήματα της XML σχετικά με την δομή της και τη δυνατότητα ανάγνωσης της από ανθρώπους. Επίσης, η XML αποτελεί την ιδανική πληροφορία για την επίτευξη της καλύτερης δυνατής διαλειτουργικότητας, μεταξύ ανεξάρτητα ανεπτυγμένων εφαρμογών στον Παγκόσμιο Ιστό.

Το συντακτικό της RDF χρησιμοποιείται τον συμβολισμό των χώρων ονομάτων όπου τα πλήρη ονόματα προκύπτουν από τη συνένωση του προθέματος (URI χώρου ονομάτων) με το όνομα του αντίστοιχου στοιχείου. Συγκεκριμένα, για τους χώρους ονομάτων των ορισμών των RDF και RDF Schema θα χρησιμοποιούμε στη συνέχεια τα προθέματα `rdf:` και `rdfs:`, όπου

```

rdf = "http://www.w3c.org/1999/02/22-rdf-syntax-ns#"
rdfs= "http://www.w3c.org/2000/01/rdf-schema#"

```

Επίσης, όπου χρησιμοποιούνται άλλα προθέματα θα αναφέρονται σε άλλα σχήματα που έχουν κατασκευαστεί για μια συγκεκριμένη εφαρμογή.

2.7.1 Περιγραφές (Descriptions)

Ένα έγγραφο του RDF αποτελεί μια σειρά από *περιγραφές*. Κάθε περιγραφή αναφέρεται συνήθως σε ένα πόρο, και περιέχει μια σειρά από ιδιότητες του πόρου αυτού. Οι τιμές αυτών των ιδιοτήτων μπορεί να είναι είτε *literal* (λεκτικές) είτε άλλοι πόροι (η αναφορά σε αυτούς γίνεται μέσω του URI) είτε άλλες περιγραφές (η αναφορά γίνεται μέσω του πόρου που αυτές περιγράφουν).

Ως έγγραφο XML, τα μεταδεδομένα RDF περιέχονται σε ένα στοιχείο με το όνομα `rdf:RDF`. Αυτό το στοιχείο περιέχει μια σειρά από περιγραφές. Οι περιγραφές, ως στοιχείο του XML, έχουν δύο προαιρετικά γνωρίσματα, τα: `rdf:about` και `rdf:ID`.

- **rdf:about**

Το γνώρισμα αυτό χρησιμοποιείται για να αναφερθεί ή να περιγράψει ένα πόρο διαδικτύου στον οποίο αναφέρεται μέσω ενός σχετικού ή απόλυτου URI. Για παράδειγμα:

```

<rdf:Description rdf:about=
    "http://www.museum.gr/artists#picasso">
    <ns1:first_name> Pablo </ns1:first_name>
</rdf:Description>

```

- **rdf:ID**

Το γνώρισμα αυτό χρησιμοποιείται για να ορίσει ένα πόρο *περιγραφών*. Το URI του δημιουργούμενου πόρου αποτελείται από την συνένωση του URI του εγγράφου στο οποίο περιέχεται η περιγραφή με την τιμή του `rdf:ID`,

χρησιμοποιώντας και τον ειδικό χαρακτήρα/αναγνωριστικό τμήματος εγγράφου XML “#”

```
<rdf:Description rdf:ID = "foobar">
...
</rdf:Description>
```

Μια περιγραφή που δεν περιέχει κάποιο από τα παραπάνω γνωρίσματα αναφέρεται σε ένα *ανώνυμο πόρο*. Σε τέτοιες περιγραφές ανατίθενται (τυχαία) δημιουργούμενα URIs, τα οποία χρησιμοποιούνται για τις εσωτερικές αναφορές σε αυτά. Δεδομένου ότι αυτά τα URIs παράγονται τυχαία και δεν είναι μοναδικά αναγνωριστικά, δεν υπάρχει τρόπος αναφοράς σε τέτοιους πόρους από άλλες περιγραφές, εκτός από τα χαρακτηριστικά που τους δίνονται εσωτερικά.

```
<rdf:Description>
...
</rdf:Description>
```

Μία περιγραφή περιέχει μια σειρά από στοιχεία XML. Αυτά τα στοιχεία ερμηνεύονται ως ιδιότητες, των οποίων τα URIs προκύπτουν από την επέκταση των ονομάτων των στοιχείων, με τη συνένωση ονόματος και προθέματος. Αν ένα στοιχείο/ιδιότητα είναι άδειο, τότε πρέπει να έχει ορισμένο το γνώρισμα `rdf:resource`, η τιμή του οποίου είναι η ιδιότητα αυτή. Αλλιώς, αν η ιδιότητα έχει λεκτική τιμή, το αντίστοιχο στοιχείο XML περιέχει απλό κείμενο, το οποίο μπορεί να ακολουθείται από συντακτικό σήμανσης XML.

Ένα παράδειγμα εγγράφου RDF όπου επιδεικνύονται όλα αυτά τα χαρακτηριστικά είναι στο σχήμα 2.15 που αποτελεί συντακτική αναπαράσταση του σχήματος 2.15.

```
<?xml version="1.0" encoding = "UTF-16" ?>
<rdf:RDF xmlns:rdf = "http://www.w3c.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs = "http://www.w3c.org/2000/01/rdf-schema #"
xmlns:ns1 = "http://www.culture.gr/schema.rdf#"

  <rdf :Description rdf :about =
"http://www.museum.gr/artists#picasso" >

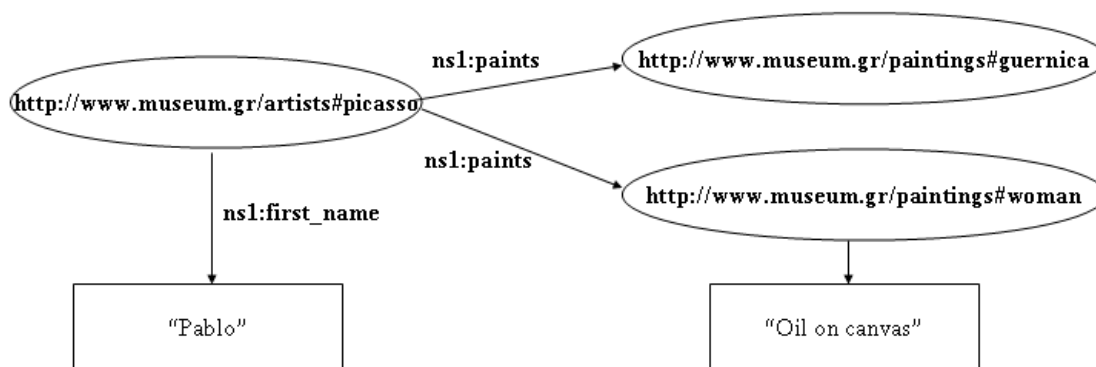
    <ns1:first_name> Pablo</ns1:first_name>

    <ns1:paints
      rdf:resource = http://www.museum.gr/paintings#guernica"/>

    <ns1:paints>
      <rdf:Description rdf:about =
        "http://www.museum.gr/paintings#woman">
        <ns1:has_material> Oil on canvas </ns1:has_material>
      </rdf:Description>
    </ns1:paints>

  </rdf :Description>
</rdf :RDF>
```

Σχήμα 2.14 Συντακτική αναπαράσταση ενός απλού RDF γράφου



Σχήμα 2.15 Ένα ενδεικτικό παράδειγμα γράφου περιγραφών RDF

Σε περίπτωση που δεν μπορούμε να διακρίνουμε αν το περιεχόμενο ενός στοιχείου /ιδιότητας είναι πόρος ή λεκτική τιμή (αμφισημία) μπορεί να χρησιμοποιηθεί το γνώρισμα `rdf:ParseType` με τιμή “Resource” ή “Literal” αντίστοιχα. Αυτό είναι ιδιαίτερα χρήσιμο στην περίπτωση που το περιεχόμενο περιέχει σημαδούρες αλλά θέλουμε και να αναλυθεί ως λεκτική τιμή και όχι ως περιγραφή RDF.

Οι προδιαγραφές που περιγράφονται στο [LS99] προτείνουν και δύο μετατροπές συντόμευσης στο παρακάτω συντακτικό, για να διευκολύνουν την «χειρονακτική» συγγραφή του αλλά και τον αυτόματο χειρισμό εγγράφων XML ως μεταδεδομένα RDF. Οι συντομώσεις αυτές είναι οι εξής:

- Το `rdf:Description` σε μια περιγραφή μπορεί να αντικατασταθεί από ένα URI. Σε αυτήν την περίπτωση το URI ερμηνεύεται ως η τιμή της ιδιότητας `rdf:type` για τον περιγραφόμενο πόρο. Για παράδειγμα η περιγραφή:

```

<rdf:Description rdf:about =
    "http://www.museum.gr/artists#picasso">

  <rdf:type   rdf:resource =
    "http://www.culture.gr/schema.rdf#Painter" />

  <ns1:first_name> Pablo   </ns1:first_name>

  <ns1:paints rdf:resource=
    "http://www.museum.gr/paintings#guernica" />

</rdf:Description >
  
```

όπου `ns1 = "http://www.culture.gr/schema.rdf#" ,` θα μπορούσε να γραφεί ως:

```

<ns1:Painter rdf:about =
    "http:// www.museum.gr/artists#picasso">
<ns1:first_name> Pablo </ns1:first_name>
<ns1:paints rdf:resource =
    "http://www.museum.gr/paintings#guernica" />
  
```

```
</ns1:Painter>
```

- Οι ιδιότητες με λεκτικές τιμές μπορούν να γραφούν ως γνωρίσματα του στοιχείου `rdf:Description` αντί ως περιεχόμενα του. Εφαρμόζοντας και αυτή τη σύντηξη στο παραπάνω παράδειγμα θα παίρναμε το παρακάτω έγγραφο του RDF:

```
<ns1:Painter
  rdf:about = "http:// www.museum.gr/artists#picasso"
  ns1:first_name= "Pablo" >
<ns1:paints rdf:resource =
  "http://www.museum.gr/paintings#guernica"/>
</ns1:Painter>
```

Το συντακτικό αυτό αρκεί για την αναπαράσταση των στοιχείων του μοντέλου.

2.7.2 Συλλογές

Οι συλλογές (`rdf:Bag`, `rdf:Alt`, `rdf:Seq`) μπορούν να χρησιμοποιηθούν σε όλα τα σημεία που μπορούν να χρησιμοποιηθούν και οι περιγραφές (`rdf:Description`). Αυτό οφείλεται στο γεγονός ότι οι περιγραφές αποτελούν συλλογές από ιδιότητες. Σε αντίθεση με τις περιγραφές, οι συλλογές δεν μπορούν να έχουν το γνώρισμα `rdf:about`, αλλά είτε έχουν το `rdf:ID` είτε είναι ανώνυμες.

Ειδικά όσον αφορά στην περιγραφή των μελών μιας συλλογής, δεν χρησιμοποιούνται οι σειριακές ιδιότητες `rdf:_1`, `rdf:_2` κλπ, που αποτελούν στιγμιότυπα της ιδιότητας `rdfs:ContainerMembershipProperty` – αλλά η «ειδική» ιδιότητα `rdf:li`, η οποία μπορεί να χειρίζεται κατάλληλα από εφαρμογές που αναγνωρίζουν το συντακτικό του RDF. Για παράδειγμα, η συλλογή του σχήματος 2.4 θα περιγραφόταν από το έγγραφο RDF του σχήματος 2.16 ενώ αυτή του σχήματος 2.5 από το έγγραφο του σχήματος 2.17.

```
<rdf:Bag  rdf:ID  ="hy350's students" >
  <rdf:li  rdf:resource = "students/Sofia"
  <rdf:li  rdf:resource = "students/John"
  <rdf:li  rdf:resource = "students/Paul"
  <rdf:li  rdf:resource = "students/Mary"
  <rdf:li  rdf:resource = "students/Peter"
</rdf:Bag>
```

Σχήμα 2.16 Συντακτική αναπαράσταση μιας συλλογής τύπου Bag

```
<rdf:Alt  rdf:ID  ="DistributionSite" >
  <rdf:li  rdf:resource = "ftp://ftp.x.org"
  <rdf:li  rdf:resource = "ftp://ftp.cs.purdue.edu"
  <rdf:li  rdf:resource = "ftp://ftp.eu.net"
</rdf:Alt>
```

Σχήμα 2.17 Συντακτική αναπαράσταση μιας συλλογής τύπου Alt

Για την κατανομημένη στα μέλη μιας συλλογής χρησιμοποιείται το `rdf:aboutEach` ως γνώρισμα του στοιχείου που αντιπροσωπεύει τη συλλογή.

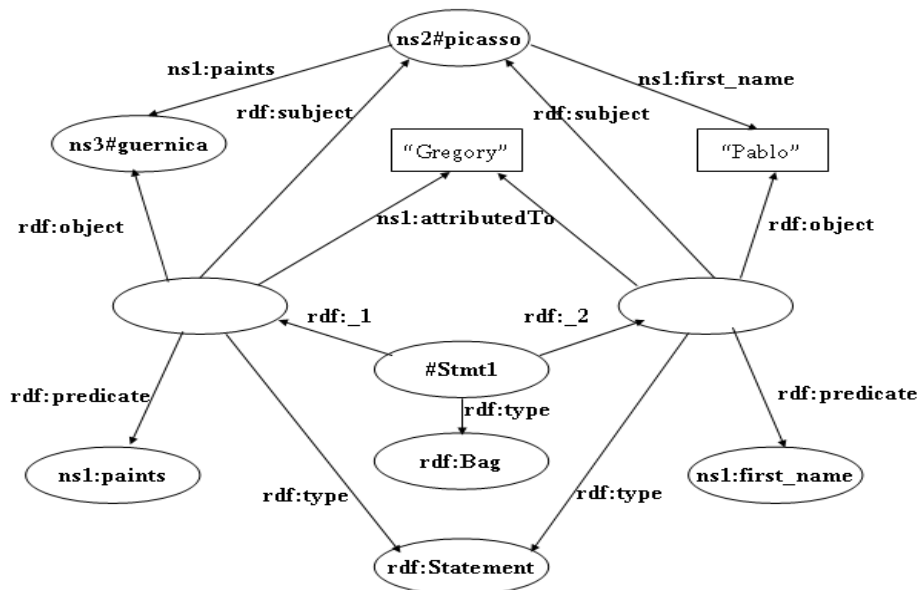
2.7.3 Υποστασιοποίηση

Το συντακτικό του RDF παρέχει ένα επιπλέον τρόπο έκφρασης προτάσεων που αναφέρονται σε άλλες προτάσεις, οι οποίες δεν είναι αναγκαστικά υποστασιοποιημένες προσθέτοντας ένα γνώρισμα `rdf:ID` στο στοιχείο ιδιότητα της πρότασης. Έτσι, μια άλλη πρόταση μπορεί να αναφέρεται σε αυτή χρησιμοποιώντας την τιμή αυτού του ID. Επίσης, μπορούμε να αναφερθούμε σε όλες τις προτάσεις που περιέχονται σε ένα στοιχείο `rdf:Description`, χρησιμοποιώντας το γνώρισμα `rdf:bagID` για να δώσουμε ένα αναγνωριστικό URI στο στοιχείο αυτό και όχι στον πόρο στον οποίο αναφέρεται η περιγραφή, όπως θα γινόταν με το `rdf:ID`. Τέλος, μπορούμε χρησιμοποιώντας για παράδειγμα το `rdf:aboutEach` με τιμή το αντίστοιχο `bagID`, να αναφερθούμε σε όλες τις προτάσεις που περιέχονται σε αυτήν την συλλογή/περιγραφή. Οι δυνατότητες αυτές φαίνονται στο παράδειγμα εγγράφου του σχήματος 2.17 που παριστάνεται από το γράφο του σχήματος 2.18.

```
<rdf :Description rdf :about =
  "http://www.museum.gr/artists#picasso" rdf:BagID= "Stmt1" >
  <ns1:first_name> Pablo</ns1:first_name>
  <ns1:paints
    rdf:resource = http://www.museum.gr/paintings#guernica"/>
</rdf :Description>

<rdf :Description rdf :about = "Stmt1" >
  <ns1:attributedTo>Gregory</ns1:attributedTo>
</rdf :Description>
```

Σχήμα 2.18 Συντακτική περιγραφή προτάσεων υψηλότερης τάξης, με υποστασιοποίηση



Σχήμα 2.19 Υποστασιοποίηση των προτάσεων που περιέχονται σε ένα στοιχείο περιγραφής

2.7.4 RDF vs. XML

Το RDF χρησιμοποιεί το συντακτικό της XML προκειμένου να σειριοποιηθεί σε αρχεία. Ωστόσο ίσως να μην είναι εμφανές γιατί δεν αρκεί η XML και μόνο για την αναπαράσταση των μεταδεδομένων του RDF. Η σύγχυση αυτή προκύπτει αν δούμε το RDF μόνο ως συντακτικό και όχι ως μοντέλο, που είναι στην πραγματικότητα, το οποίο είναι εντελώς διαφορετικό από αυτό της XML. Για να γίνει σαφέστερο, θα επιχειρήσουμε μια σύντομη ανάλυση του μοντέλου της XML και στη συνέχεια θα δείξουμε τις διαφορές του με αυτό του RDF.

Η XML προτείνει ένα μοντέλο δέντρου, με ετικέτες μόνο στους κόμβους (ονόματα στοιχείων –σημαδούρων). Επίσης, με τη χρήση XML References ([DM 00], [DJM00], [CD99]) είναι δυνατό ένα XML έγγραφο να περιγράψει ένα μοντέλο γράφου. Συγκεκριμένα, ένα μοντέλο δέντρου XML μπορεί να μεταφραστεί σε ένα γράφο σύμφωνο με ένα μοντέλο ημιδομημένων δεδομένων και αντίστροφα, καθώς τα μοντέλα ημιδομημένων δεδομένων είναι μοντέλα γράφων με ετικέτες μόνο στις ακμές. Οπότε, η αντιστοίχιση κόμβων με ακμές μπορεί να γίνει εύκολα. Στην περίπτωση των γράφων XML, κάτι αντίστοιχο δεν είναι πάντα δυνατό, ειδικά δεδομένου ότι η ίδια πληροφορία μπορεί να αναπαρασταθεί με πολλούς διαφορετικούς τρόπους στην XML. Η XML επιβάλλει περιορισμούς όσον αφορά στη σειρά των στοιχείων, όπως θα δούμε παρακάτω, ενώ δεν συμβαίνει το ίδιο και με τη σειρά των γνωρισμάτων ενώ στοιχείου. Το παραπάνω γεγονός περιπλέκει ακόμα περισσότερο την κατάσταση καθώς και τόσο τα στοιχεία όσο και τα γνωρίσματα αναπαριστούνται με τον ίδιο τρόπο σε ένα XML δέντρο.

Από την άλλη μεριά, το RDF είναι ένα μοντέλο κατευθυνόμενων γράφων, με ετικέτες τόσο στους κόμβους (πόροι που έχουν ταξινομηθεί σε μια κλάση) όσο και στις ακμές (επώνυμες ιδιότητες). Επομένως, δεν μπορεί να αναπαρασταθεί ούτε από το μοντέλο της XML ούτε από τα άλλα μοντέλα ημιδομημένων δεδομένων. Επίσης, το RDF προτείνει και κάποια άλλα στοιχεία που δεν εμφανίζονται σε άλλα μοντέλα γράφων, όπως είναι οι συλλογές και οι προτάσεις ανωτέρας τάξης.

Ένα άλλο σημαντικό χαρακτηριστικό του μοντέλου της XML είναι το γεγονός ότι θεωρεί σημαντική τη σειρά των στοιχείων σε ένα δέντρο / γράφο, καθώς επίσης και τον τρόπο που είναι εμφωλευμένα (nested) τα στοιχεία αυτά. Έτσι, τα δύο έγγραφα που φαίνονται στο σχήμα 2.20 είναι διαφορετικά επεξεργαζόμενα ως έγγραφα XML. Αυτή η επιλογή είναι λογική στην περίπτωση της XML καθώς αυτή είχε προταθεί με σκοπό την περιγραφή εγγράφων. Ωστόσο, σε εφαρμογές δεδομένων (π.χ. αποθήκευση και επερώτηση) η σειρά όχι μόνο δεν είναι σημαντική αλλά αποτελεί και σημαντικό παράγοντα υποβάθμισης των επιδόσεων τους. Για το λόγο αυτό, όλα τα συστήματα που χειρίζονται τέτοιου είδους δεδομένα δεν λαμβάνουν υπ' όψιν τη σειρά και το φώλιασμα. Εξάλλου, ακόμα και σε ειδικές λειτουργίες που παίζει ρόλο η σειρά (π.χ. ταξινόμηση των αποτελεσμάτων μιας επερώτησης), η επιβολή της σειράς στην αποθήκευση δεν εξασφαλίζει την ανάκτηση των αποτελεσμάτων με την ίδια σειρά. Κατά συνέπεια, ακόμα και τέτοιες λειτουργίες βελτιστοποιούνται καλύτερα από τα συστήματα και δεν έχει νόημα να αποτελούν μέρος της αναπαράστασης.

```
<Painter rdf:about =  
    "http://www.museum.gr/picasso.htm"  
    first_name = "Pablo" last_name= "Picasso" >  
    <paints rdf:resource = http://www.museum.gr/guernica.gif"/>  
</Painter>
```

```
<Painting rdf:about="http://www.museum.gr/guernica.gif" />
```

(α)

```
<Painter rdf:about =  
    "http://www.museum.gr/picasso.htm" >  
    <paints>  
        <Painting rdf:about="http://www.museum.gr/guernica.gif" />  
    </paints>  
    <first_name> Pablo </first_name>  
    <last_name> Picasso </last_name>  
</Painter>
```

(β)

Σχήμα 2.20 Διαφορετικές συντακτικές αναπαραστάσεις της ίδιας πληροφορίας σε XML

Η XML επίσης συνοδεύεται από μηχανισμούς ανάθεσης τύπων σε κόμβους. Κατ' αρχήν οι DTDs (Document Type Definitions) παρέχουν μηχανισμούς ορισμού της δομής εγγράφων XML. Δεδομένου ότι οι DTDs επίσης ορίζουν πεδία που μπορεί να έχει ένα στοιχείο, μπορούν επίσης να θεωρηθούν ως σχήματα, με περιορισμένη όμως εκφραστική ισχύ. Γι' αυτό προτάθηκε το XML Schema, το οποίο παρέχει πλήρες σύστημα τύπων και δυνατότητα ταξινόμησης σε κλάσεις, για τους κόμβους ενός δέντρου XML. Ωστόσο, στο παραπάνω δεν λαμβάνονται υπ' όψιν οι ιδιότητες ως γνωρίσματα ή σχέσεις μεταξύ των πόρων καθώς αναφέρονται σε διαφορετικό μοντέλο το οποίο έχει ετικέτες μόνο στις ακμές. Επίσης, πρέπει να σημειωθεί ότι [KS] οι σχέσεις επέκτασης και εκλέπτυνσης τύπων στοιχείων XML χρησιμοποιούνται καθαρά για συντακτική διευκόλυνση καθώς τα στιγμιότυπα μιας κλάσης δεν θεωρούνται στιγμιότυπα των υπερκλάσεων αυτής. Σκοπός τους είναι η χρησιμοποίηση υπαρχόντων ορισμών τύπων για τη δημιουργία νέων. Τέλος, στα σχήματα XML ένα στοιχείο δεν μπορεί να κατηγοριοποιηθεί σε περισσότερες από μία κλάσεις και για το λόγο αυτό τα μοντέλα αυτά είναι λιγότερο εκφραστικά από το RDF/RDFS, σε σχέση με τις ανάγκες της αναπαράστασης μας.

ΚΕΦΑΛΑΙΟ 3

Υποστασιοποίηση. Ορισμοί, θεωρήσεις και υπάρχουσες υλοποιήσεις

Στο προηγούμενο κεφάλαιο είδαμε ότι το RDF υποστηρίζει την έννοια της υποστασιοποίησης. Επίσης, είδαμε ότι τα RDF μεταδεδομένα μπορούν να αναπαρασταθούν ως κατευθυνόμενοι γράφοι με ετικέτες. Στο παρόν κεφάλαιο θα αναλύσουμε αρχικά την έννοια της υποστασιοποίησης και θα αναφέρουμε τις υπάρχουσες απόψεις όσον αφορά αυτήν. Επίσης, θα επιχειρήσουμε να επεκτείνουμε το μοντέλο γράφων του RDF σε ένα μοντέλο υπεργράφων το οποίο θα υποστηρίζει την υποστασιοποίηση. Τέλος, θα αναφερθούμε στα υπάρχοντα συστήματα που υποστηρίζουν την υποστασιοποίηση και στον τρόπο υποστήριξης αυτής.

3.1 Υποστασιοποίηση

Η υποστασιοποίηση (reification) ως έννοια είναι περισσότερο ένας φιλοσοφικός όρος παρά ένας τεχνικός όρος. Ως φιλοσοφικός όρος, υποστασιοποίηση είναι η μεταχείριση μιας αφηρημένης έννοιας ή ιδέας σαν να ήταν ένα αντικείμενο που έχει φυσική υπόσταση [G85]. Η υποστασιοποίηση ως τεχνική έχει την καταγωγή της στην τεχνητή νοημοσύνη και στην κυριολεξία σημαίνει «αντικειμενοποίηση». Ουσιαστικά, η τεχνική της υποστασιοποίησης μετατρέπει αυτά για τα οποία επιθυμούμε να μιλήσουμε σε «αντικείμενα», έτσι ώστε στη συνέχεια να υπάρχει η δυνατότητα να γίνουν ισχυρισμοί για αυτά. Στα πλαίσια του RDF, υποστασιοποίηση είναι η απόδοση ενός αναγνωριστικού σε μία πρόταση. Η πρόταση αυτή (υποστασιοποιημένη πρόταση) στη συνέχεια μπορεί να χρησιμοποιηθεί ως υποκείμενο ή ως αντικείμενο μίας άλλης πρότασης. Με αυτόν τον τρόπο, με τη βοήθεια της υποστασιοποίησης, μπορούν να φτιαχτούν RDF προτάσεις για άλλες RDF προτάσεις. Έτσι, για παράδειγμα η πρόταση “Ο Πέτρος πιστεύει ότι ο Παγκόσμιος Ιστός περιέχει ένα δισεκατομμύριο έγγραφα” μπορεί να αναπαρασταθεί σε μία RDF εφαρμογή μόνο αν αυτή υποστηρίζει την υποστασιοποίηση.

3.1.1 Μοντελοποίηση των Προτάσεων στο RDF

Το RDF, λοιπόν, μπορεί να χρησιμοποιηθεί για τη δημιουργία προτάσεων για άλλες προτάσεις. Οι προτάσεις αυτές αναφέρονται ως *προτάσεις υψηλότερης τάξεως*. Απαραίτητη προϋπόθεση, για να δημιουργηθεί μία πρόταση για μία άλλη πρόταση, είναι να δημιουργηθεί αρχικά ένας νέος πόρος για την πρόταση. Στη συνέχεια στον πόρο που αναπαριστά την πρόταση είναι δυνατόν να αποδοθούν ιδιότητες. Η διαδικασία απόδοσης ενός πόρου σε μία πρόταση ονομάζεται Υποστασιοποίηση (Reification).

Στο RDF M&S Specification [LS99] μόνο οι προτάσεις που έχουν εισαχθεί στη βάση μεταδεδομένων θεωρούνται αληθείς, ενώ αντίθετα θεωρείται ότι δεν γνωρίζουμε τίποτα για την αλήθεια των υποστασιοποιημένων προτάσεων.

Για παράδειγμα, η πρόταση:

“*Η Ora Lassila είναι ο δημιουργός (creator) της ιστοσελίδας <http://www.w3.org/Home/Lassila>”*

θα θεωρούνταν ως γεγονός από το RDF M&S Specification.

Ενώ αντίθετα, αν είχε λεχθεί η πρόταση

“*Ο Ralph Swick λέει ότι η Ora Lassila είναι ο δημιουργός της ιστοσελίδας <http://www.w3.org/Home/Lassila>”*

το RDF M&S Specification θα θεωρούσε ότι δεν έχει λεχθεί τίποτα για τον πόρο <http://www.w3.org/Home/Lassila>. Ως γεγονός θα θεωρούνταν το ότι, ο Ralph έχει δηλώσει αυτή την πρόταση. Προκειμένου να εκφραστεί αυτό το γεγονός στο RDF M&S Specification, πρέπει να μοντελοποιηθεί η αρχική πρόταση σαν ένας πόρος με τέσσερις ιδιότητες. Αυτή η διαδικασία τυπικά ονομάζεται *υποστασιοποίηση (reification)* στην κοινότητα της Αναπαράσταση Γνώσης (Knowledge Representation - KR). Ένα μοντέλο αυτής της πρότασης ονομάζεται μία *υποστασιοποιημένη πρόταση (reified statement)*.

Προκειμένου να μοντελοποιηθούν οι προτάσεις το RDF ορίζει τις παρακάτω ιδιότητες:

- subject** Η ιδιότητα *subject* προσδιορίζει τον πόρο που περιγράφεται από την μοντελοποιημένη πρόταση. Αυτό σημαίνει ότι η τιμή της ιδιότητας *subject* είναι ο πόρος για τον οποίο δημιουργήθηκε η αρχική πρόταση (στο παραπάνω παράδειγμα, <http://www.w3.org/Home/Lassila>).
- predicate** Η ιδιότητα *predicate* προσδιορίζει την αρχική ιδιότητα (property) σε μία μοντελοποιημένη πρόταση. Η τιμή της ιδιότητας *predicate* είναι ο πόρος εκείνος που αναπαριστά την συγκεκριμένη ιδιότητα στην αρχική πρόταση (στο παραπάνω παράδειγμα, η ιδιότητα *creator*).
- object** Η ιδιότητα *object* προσδιορίζει την τιμή της ιδιότητας στην μοντελοποιημένη πρόταση. Η τιμή της ιδιότητας *object* είναι το αντικείμενο της αρχικής πρότασης (στο παραπάνω παράδειγμα, "Ora Lassila").
- type** Η τιμή της ιδιότητας *type* περιγράφει τον τύπο του νέου πόρου. Όλες οι υποστασιοποιημένες προτάσεις είναι στιγμιότυπα του RDF:Statement, επομένως έχουν μία ιδιότητα *type* το αντικείμενο της οποίας είναι το RDF:Statement.

Ένας νέος πόρος με τις τέσσερις παραπάνω ιδιότητες αναπαριστά την αρχική πρόταση και μπορεί να χρησιμοποιηθεί τόσο σαν αντικείμενο όσο και σαν υποκείμενο άλλων προτάσεων. Ο πόρος με αυτές τις τέσσερις ιδιότητες δεν αποτελεί μία αντικατάσταση της αρχικής πρότασης, αλλά είναι ένα μοντέλο αυτής, σύμφωνα με το RDF. Μία πρόταση και η αντίστοιχη υποστασιοποιημένη πρόταση μπορούν να υπάρχουν ανεξάρτητα σε ένα RDF γράφο και οποιαδήποτε από αυτές μπορεί να είναι παρούσα χωρίς την άλλη. Ένας RDF γράφος λέγεται ότι περιέχει το γεγονός που δηλώνεται από την πρόταση αν και μόνο αν η πρόταση είναι παρούσα στο γράφο, ανεξάρτητα από το αν η αντίστοιχη υποστασιοποιημένη πρόταση είναι παρούσα.

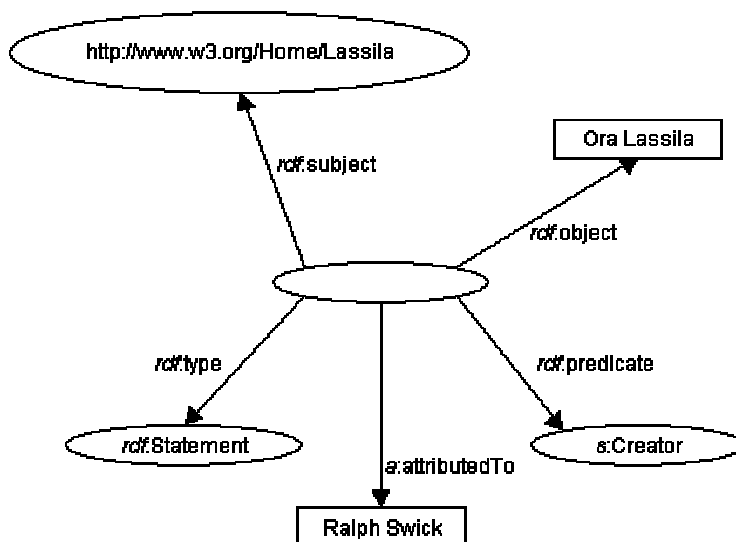
Για την μοντελοποίηση του παραπάνω παραδείγματος, μπορεί να προστεθεί μια ακόμα ιδιότητα στην υποστασιοποιημένη πρόταση, για παράδειγμα η ιδιότητα "attributedTo" με τιμή "Ralph Swick".

Η συντακτική αναπαράσταση της υποστασιοποιημένης πρότασης είναι η παρακάτω:

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:a="http://description.org/schema/">
  <rdf:Description>
    <rdf:subject resource="http://www.w3.org/Home/Lassila" />
    <rdf:predicate
      resource="http://description.org/schema/Creator"/>
    <rdf:object>Ora Lassila</rdf:object>
    <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement" />
    <a:attributedTo>Ralph Swick</a:attributedTo>
  </rdf:Description>
</rdf:RDF>
```

Σχήμα 3.1 Συντακτική αναπαράσταση Υποστασιοποιημένης Πρότασης

Η αναπαράσταση της υποστασιοποιημένης πρότασης με τη βοήθεια ενός γράφου RDF φαίνεται στο σχήμα 3.2.



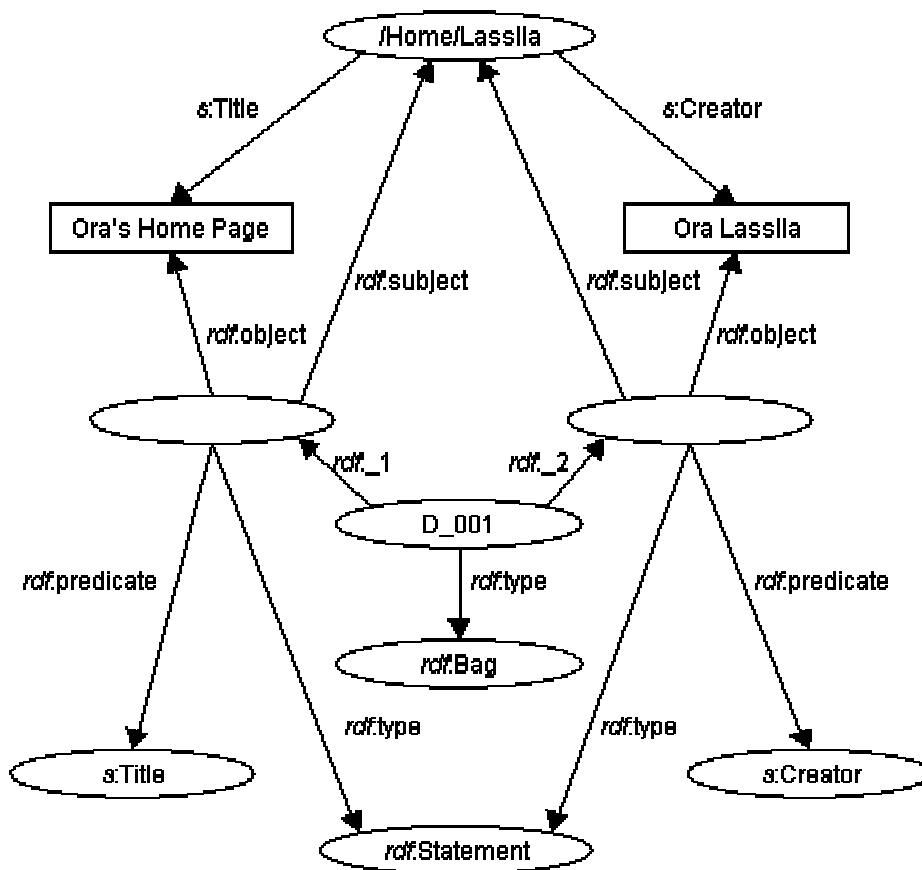
Σχήμα 3.2 Γράφος υποστασιοποιημένης πρότασης

Η υποστασιοποίηση χρειάζεται επίσης για την ομαδοποίηση προτάσεων που υποδηλώνονται από το στοιχείο Description. Το μοντέλο γράφου του RDF δεν απαιτεί μία ειδική δομή για τις Περιγραφές (Descriptions) αφού αυτές στην πραγματικότητα αποτελούν συλλογές προτάσεων, ένα πολυσύνολο (Bag) χρησιμοποιείται για να δηλώσει ότι ένα σύνολο προτάσεων προέρχεται από την ίδια συντακτικά Περιγραφή. Κάθε πρόταση σε μια Περιγραφή είναι υποστασιοποιημένη και κάθε μία από τις υποστασιοποιημένες προτάσεις είναι μέλος του πολυσυνόλου (Bag) που αναπαριστάται από αυτή την Περιγραφή.

Για παράδειγμα, το παρακάτω τμήμα RDF:

```
<rdf:RDF>
  <rdf:Description about="http://www.w3.org/Home/Lassila"
    bagID="D_001">
    <s:Creator>Ora Lassila</s:Creator>
    <s:Title>Ora's Home Page</s:Title>
  </rdf:Description>
</rdf:RDF>
```

αναπαριστάται στον RDF γράφο του σχήματος 3.3.



Σχήμα 3.3 Χρήση του Bag για την αναπαράσταση ομαδοποιημένων προτάσεων

3.1.2 Τυπικό Μοντέλο του RDF και Υποστασιοποίηση

Σύμφωνα με το RDF Specification [LS99] το μοντέλο δεδομένων του RDF ορίζεται τυπικά ως εξής:

1. Υπάρχει μια κλάση που ονομάζεται *Resources* (Πόροι).
2. Υπάρχει μια κλάση που ονομάζεται *Literals* (Λεκτικά).
3. Υπάρχει ένα υποκλάση των *Resources* που ονομάζεται *Properties*(Ιδιότητες).
4. Υπάρχει μια κλάση που ονομάζεται *Statements* (Προτάσεις), κάθε στοιχείο της οποίας είναι μία τριάδα της μορφής

{pred, sub, obj}

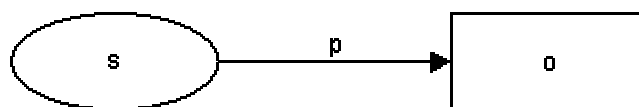
όπου

pred: είναι μια ιδιότητα (property – μέλος του *Properties*),

sub: είναι ένας πόρος (resource – μέλος του *Resources*), και

obj: είναι είτε ένας πόρος (resource) είτε ένα λεκτικό (literal - μέλος του *Literals*).

Μπορούμε να «δούμε» ένα σύνολο από προτάσεις (statements – μέλη του συνόλου *Statements*) σαν ένα κατευθυνόμενο γράφο με ετικέτες, όπου κάθε πόρος και λεκτικό (literal) είναι μία κορυφή, και η τριάδα {p, s, o} είναι μία ακμή από το s στο o, με την ετικέτα p. Ο γράφος αυτός απεικονίζεται στο σχήμα 3.4.



Σχήμα 3.4 Γράφος μίας απλής πρότασης (statement)

Ο παραπάνω γράφος μπορεί να διαβαστεί με διάφορους τρόπους, όπως

“Το o είναι η τιμή του p για το s”

ή (από δεξιά προς τα αριστερά)

“Το s έχει μία ιδιότητα p με τιμή o”

Για παράδειγμα, η πρόταση

“Η Ora Lassila είναι ο δημιουργός (creator) του πόρου <http://www.w3.org/Home/Lassila>”

μπορεί να αναπαρασταθεί γραφικά ως εξής:



Σχήμα 3.5 Γραφική αναπαράσταση απλής πρότασης

και η αντίστοιχη τριάδα, που αποτελεί μέλος του συνόλου *Statements*, θα είναι:

```
{creator, [http://www.w3.org/Home/Lassila], "Ora Lassila"}
```

Ο συμβολισμός $[I]$ δηλώνει ότι ο πόρος χαρακτηρίζεται από το μοναδικό αναγνωριστικό (URI) I και τα εισαγωγικά δηλώνουν ένα λεκτικό.

Με τη χρήση των τριάδων θα εξηγήσουμε πως γίνεται η υποστασιοποίηση των προτάσεων. Ας εξετάσουμε ένα παράδειγμα.

Δίνεται η παρακάτω πρόταση

```
{creator, [http://www.w3.org/Home/Lassila], "Ora Lassila"}
```

η υποστασιοποίηση αυτής μπορεί να εκφραστεί σαν ένας νέος πόρος X όπως φαίνεται παρακάτω:

```
{type, [X], [RDF:Statement]}  
{predicate, [X], [creator]}  
{subject, [X], [http://www.w3.org/Home/Lassila]}  
{object, [X], "Ora Lassila"}
```

Από τη μεριά ενός αναλυτή RDF (processor), τα γεγονότα δηλαδή, οι προτάσεις (statements) είναι τριάδες που είναι όλες μέλη της κλάσης *Statements*. Επομένως, η αρχική πρόταση παραμένει ένα γεγονός παρά το ότι υποστασιοποιείται από τη στιγμή που η τριάδα που αναπαριστά την αρχική πρόταση παραμένει μέλος του συνόλου *Statements*. Αυτό που έχει γίνει είναι ότι απλώς έχουν προστεθεί τέσσερις επιπλέον τριάδες.

Ο τυπικός ορισμός του type είναι ο εξής:

5. Υπάρχει ένα στοιχείο της κλάσης *Properties* γνωστό ως *RDF:type*.
6. Τα μέλη της κλάσης *Statements*, της μορφής $\{RDF:type, sub, obj\}$, πρέπει να ικανοποιούν τα ακόλουθα: τα *sub* και *obj* πρέπει να είναι μέλη της κλάσης *Resources*.

Ο τυπικός ορισμός του RDF Specification για την υποστασιοποίηση (reification) είναι ο εξής:

7. Υπάρχει ένα στοιχείο του συνόλου *Resources*, που δεν περιέχεται στο σύνολο *Properties*, γνωστό ως RDF:Statement.
8. Υπάρχουν τρία στοιχεία στο σύνολο *Properties* γνωστά σαν RDF:predicate, RDF:subject και RDF:object.
9. Η Υποστασιοποίηση (Reification) μίας τριάδας {pred, sub, obj} του συνόλου *Statements* είναι ένα στοιχείο *r* του συνόλου *Resources* που αναπαριστά την υποστασιοποιημένη τριάδα (reified triple) και τα στοιχεία s_1, s_2, s_3 , και s_4 του συνόλου *Statements* τα οποία είναι της μορφής:

$$\begin{aligned} s_1: & \{ \text{RDF:predicate, } r, \text{ pred} \} \\ s_2: & \{ \text{RDF:subject, } r, \text{ subj} \} \\ s_3: & \{ \text{RDF:object, } r, \text{ obj} \} \\ s_4: & \{ \text{RDF:type, } r, [\text{RDF:Statement}] \} \end{aligned}$$

Ο πόρος *r* στον παραπάνω ορισμό ονομάζεται η *υποστασιοποιημένη πρόταση (reified statement)*. Όταν ένας πόρος αναπαριστά μια υποστασιοποιημένη πρόταση, σημαίνει ότι έχει μία ιδιότητα RDF:type η τιμή της οποίας είναι RDF:Statement, οπότε αυτός ο πόρος πρέπει να έχει οπωσδήποτε ακριβώς μία ιδιότητα RDF:subject, ακριβώς μία ιδιότητα RDF:object, και ακριβώς μία ιδιότητα RDF:predicate.

3.2.3 Θεωρήσεις για την Υποστασιοποίηση

Οι παραπάνω θεωρήσεις αφορούσαν στο RDF Specification. Στην παράγραφο αυτή θα προτείνουμε κάποιες τροποποιήσεις στις θεωρήσεις αυτές λόγω της μη συμβατότητας τους με τη χρηστικότητα της υποστασιοποίησης του RDF όπως προτείνεται στην υπάρχουσα εργασία.

Ας εξετάσουμε τις προτεινόμενες αλλαγές στις προδιαγραφές του RDF όσον αφορά στην υποστασιοποίηση:

- Στο RDF Specification οι υποστασιοποιημένες προτάσεις μπορούν να υπάρχουν ανεξάρτητα από το αν η αντίστοιχη πρόταση υπάρχει ή όχι στο σχήμα. Στην εργασία αυτή θεωρούμε ότι δεν μπορεί να υπάρχει η υποστασιοποιημένη πρόταση αν προηγουμένως δεν έχει εισαχθεί η αντίστοιχη πρόταση. Συγκεκριμένα, μία πρόταση υποστασιοποιείται από τη στιγμή που αυτή θα χρησιμοποιηθεί ως υποκείμενο ή ως αντικείμενο μίας άλλης πρότασης. Η θεώρηση αυτή θα ήταν χρήσιμη για την περίπτωση εφαρμογών όπου έχουμε υποθετικό λογισμό. Για παράδειγμα, αν ισχύει μία πρόταση, τότε μπορούν να διατυπωθούν άλλες προτάσεις σχετικές με την ισχύουσα πρόταση.
- Μία υποστασιοποιημένη πρόταση θεωρείται από το RDF Specification ως μία πρόταση για την οποία δεν γνωρίζουμε τίποτα όσον αφορά στην αλήθεια αυτής. Αντίθετα, πάλι σύμφωνα με το RDF Specification, όλες οι προτάσεις

θεωρούνται ως αληθείς και αποδεκτές. Στην παρούσα εργασία, μία υποστασιοποιημένη πρόταση δεν είναι μη αληθής και αμφισβητούμενη, καθώς για να υποστασιοποιηθεί μία πρόταση, όπως θεωρήσαμε παραπάνω, θα πρέπει προηγουμένως να υπάρχει η κανονική πρόταση στο μοντέλο μας. Επομένως, μία τέτοια θεώρηση θα ήταν αντικρουόμενη με την προηγούμενη.

Η υποστασιοποίηση αποτελεί ένα μηχανισμό του RDF, ο οποίος μπορεί να εξυπηρετήσει στην ανάπτυξη εφαρμογών πρακτόρων όπου απαιτείται πολλές φορές η αναζήτηση πληροφορίας που έχει ειπωθεί από κάποιους συγκεκριμένους επιστημονικούς οργανισμούς, στην ανάπτυξη συστημάτων που μπορούν να αποθηκεύσουν πληροφορίες για την προέλευση των μεταδεδομένων ή συστημάτων αξιολόγησης πληροφοριών. Για παράδειγμα, ένα σύστημα αξιολόγησης πληροφοριών μπορεί να περιέχει την βαθμολόγηση από χρήστες προτάσεων όπως π.χ. “Το www.w3c.org/RDF-Specification είναι καλό για RDF πληροφορίες” η πρόταση αυτή μπορεί να υποστασιοποιηθεί και στην συνέχεια να αποδοθεί σε αυτήν βαθμολογία από το 1-10. Σε μία τέτοια εφαρμογή, ο περιορισμός που επιβάλλει το RDF Specification ότι όλες οι υποστασιοποιημένες προτάσεις είναι μη αληθείς δεν ευσταθεί, στην περίπτωση αυτή η πρόταση δεν είναι ούτε αληθής ούτε ψευδής και το ζητούμενο είναι η βαθμολόγηση αυτής της πρότασης. Επομένως σε τέτοιου είδους εφαρμογές, θεωρήσεις όπως αυτές του RDF Specification για την υποστασιοποίηση δεν έχουν ουσιαστική εφαρμογή.

Παρακάτω θέτονται οι περιορισμοί για την υποστασιοποίηση:

Περιορισμός 1

Μία πρόταση υποστασιοποιείται αν και μόνο αν η πρόταση αυτή υπάρχει ως υποκείμενο ή ως αντικείμενο μίας άλλης πρότασης.

Περιορισμός 2

Η ύπαρξη μίας υποστασιοποιημένης πρότασης δεν συνεπάγεται ότι η πρόταση αυτή δεν είναι αληθής.

3.2.3.1 Ιδιότητες Υποστασιοποίησης

Στην παραπάνω παράγραφο αναφέρθηκε ότι προκειμένου να υποστασιοποιηθεί μία πρόταση πρέπει προηγουμένως η πρόταση αυτή να αναφέρεται ως υποκείμενο ή ως αντικείμενο μίας άλλης πρότασης. Ωστόσο, δεν είναι δυνατόν όλες οι ιδιότητες (properties) να έχουν ως αντικείμενο ή υποκείμενο τους μία άλλη πρόταση. Για το σκοπό αυτό ορίζουμε μία ειδική υποκατηγορία των ιδιοτήτων (Properties) τις οποίες ονομάζουμε **Reifying Properties**, και οι οποίες έχουν ως υποκείμενο ή ως αντικείμενο τους μία πρόταση.

Ορισμός 1 (Reifying Property). *Οι Reifying Properties είναι ένα υποσύνολο των ιδιοτήτων του RDF, οι οποίες έχουν τα εξής χαρακτηριστικά:*

- {RDF:type, rdfs:range, [RDF:Statement]}

Η

- {RDF:type, rdfs:domain, [RDF:Statement]}

Τα Reifying Properties αποτελούν μία ειδική κατηγορία των Properties και μπορούν να δηλώνουν πίστη, άποψη, υπόθεση, υποψία και άλλα.

Για παράδειγμα, Reifying Property οι οποίες έχουν ως αντικείμενο μία υποστασιοποιημένη πρόταση ή ένα σύνολο εμφωλευμένων υποστασιοποιημένων προτάσεων, μπορεί να είναι π.χ. οι ιδιότητες said, believes, assumes, suspects, declared, mentioned.

Με αυτόν τον τρόπο μπορούν να αναπαρασταθούν στο RDF προτάσεις της μορφής

“Ο John είπε (said) ότι “Η Mary είναι δασκάλα””
ή

“Ο Melnik υποψιάζεται (suspects) ότι “Το RDF χρειάζεται την υποστασιοποίηση””.

Reifying properties με υποκείμενο μία υποστασιοποιημένη πρόταση ή ένα σύνολο εμφωλευμένων υποστασιοποιημένων δηλώσεων μπορεί να είναι για παράδειγμα, οι: wasDeclared, wasTold, wasToldAt, isBelievedBy, wasMentioned κ.α. ανάλογα με τις ανάγκες της εκάστοτε εφαρμογής.

Με τη βοήθεια των παραπάνω Reifying Properties μπορούν αποθηκευτούν με τη βοήθεια του RDF προτάσεις όπως οι:

“ “Ο Picasso ζωγράφησε(painted) την Guernica” ειπώθηκε (wasTold) από τον Nick”
ή

“ “ “Η Ora Lassila είναι ο δημιουργός(creator) του www.w3c.org/OraLassila ” πιστεύεται από (isBelievedBy) τον John” είπε (wasTold) η Ann”

Ωστόσο, δεν είναι δυνατόν ένα Reifying Property να έχει και ως υποκείμενο και ως αντικείμενο ταυτόχρονα μία υποστασιοποιημένη πρόταση ή ένα σύνολο εμφωλευμένων υποστασιοποιημένων δηλώσεων. Για τον λόγο αυτό θέτουμε τον παρακάτω περιορισμό.

Περιορισμός 3

Ένα Reifying Property δεν μπορεί να έχει ταυτόχρονα domain και range τα οποία να είναι του τύπου Statement.

3.3 Γράφοι και Υπεργράφοι στο RDF

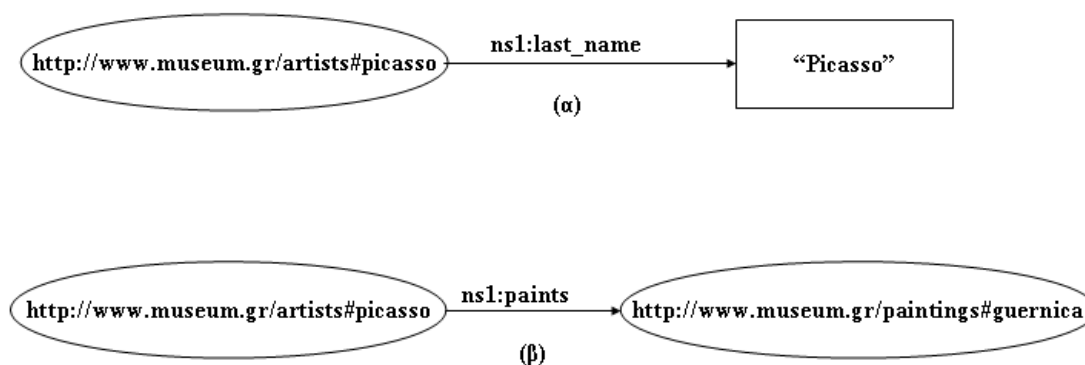
Ένας τρόπος αναπαράστασης των RDF περιγραφών είναι οι γράφοι (παρ.2.4) Οι γράφοι που χρησιμοποιεί το RDF είναι οι κατευθυνόμενοι γράφοι με ετικέτες. Αυτοί οι

γράφοι ονομάζονται και διαγράμματα κόμβων και ακμών (nodes and arcs diagrams). Οι κόμβοι αντιστοιχούν σε πόρους ή σε Literal ενώ οι ακμές σε ιδιότητες. Οι κόμβοι που αντιστοιχούν σε πόρους απεικονίζονται με ελλειψοειδή σχήματα, ενώ αυτοί που αντιστοιχούν σε Literal απεικονίζονται με παραλληλόγραμμα σχήματα. Δεδομένου ότι το πεδίο ορισμού μίας ιδιότητας είναι πάντα ένας πόρος ενώ το σύνολο τιμών είναι είτε ένας πόρος είτε ένα Literal, συνεπάγεται ότι μία ακμή στο γράφο θα ξεκινάει πάντα από μία έλλειψη και θα καταλήγει είτε σε μία έλλειψη είτε σε ένα παραλληλόγραμμα.

Στο σχήμα 3.6 απεικονίζεται μία RDF περιγραφή με τη βοήθεια ενός κατευθυνόμενου γράφου με ετικέτες.

Στο παράδειγμα αυτό, ο γράφος 3.6(α) δηλώνει ότι ο κόμβος *http://www.museum.gr/artists#picasso* έχει μία ιδιότητα, την *http://www.culture.gr/schema.rdf#last_name*, με τιμή “Picasso”.

Ο γράφος 3.6(β) δηλώνει ότι ο κόμβος *http://www.museum.gr/artists#picasso* έχει μία ιδιότητα *http://www.culture.gr-/schema.rdf#creates* με τιμή *http://www.museum.gr/paintings#Guernica*.



Σχήμα 3.6 Παραδείγματα RDF γράφων

Σε φυσική γλώσσα ο γράφος του σχήματος 3.6 (α) αναπαριστά τη φράση:

Το επώνυμο (ns1:last_name) του http://www.museum.gr/artists#picasso είναι "Picasso".

και ο γράφος του σχήματος 3.6(β) αναπαριστά τη φράση:

Ο δημιουργός (ns1:creator) της Guernica, http://www.museum.gr/paintings#Guernica είναι ο http://www.museum.gr/artists#picasso.

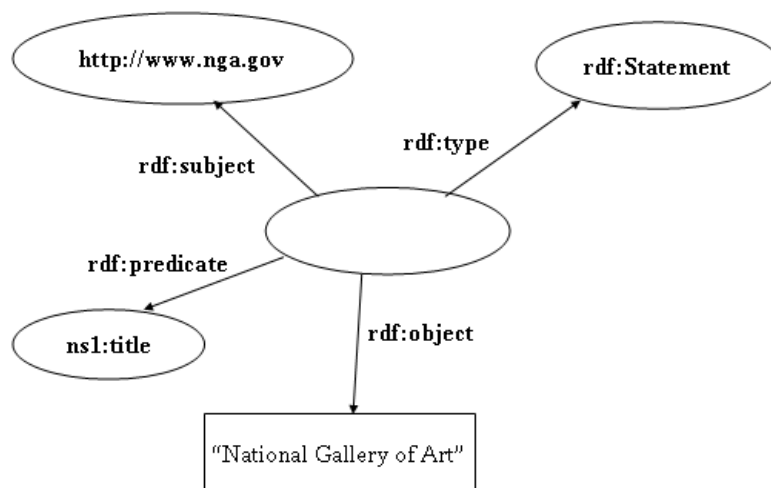
Η τριάδα *πόρος-ιδιότητα πόρου-τιμή* ονομάζεται πρόταση στην ορολογία της RDF. Τα τρία αυτά μέρη της πρότασης, αποτελούν αντίστοιχα το υποκείμενο, το κατηγορημα και το αντικείμενο της πρότασης.

Στο παραπάνω παράδειγμα χρησιμοποιήθηκε το πρόθεμα ns1 για τις ιδιότητες last_name και creator, στη θέση του πλήρους URI *http://www.culture.gr/schema.rdf#last_name* και *http://www.culture.gr/schema.rdf#creator* αντίστοιχα. Η συντομογραφία αυτή είναι εφικτή λόγω της δυνατότητας που παρέχει η XML για ορισμό προθεμάτων τα οποία υποδεικνύουν το έγγραφο στο οποίο έχει οριστεί το αντίστοιχο στοιχείο. Η δυνατότητα αυτή είναι ιδιαίτερα χρήσιμη στο RDF για τον ορισμό του σχήματος – λεξιλογίου (RDF Schema) στο οποίο έχει οριστεί η αντίστοιχη ιδιότητα. Ουσιαστικά ένα σχήμα ορίζει τις λέξεις – και τη σημασία αυτών - που μπορούν να χρησιμοποιηθούν για το σχηματισμό προτάσεων. Επομένως, το σχήμα είναι το αντίστοιχο ενός λεξικού που περιέχει όλες τις λέξεις που μπορούμε να χρησιμοποιήσουμε καθώς και τη σημασία τους.

Σύμφωνα με τα παραπάνω γίνεται σαφές ότι, απαραίτητη προϋπόθεση για την χρήση του RDF, στην ανταλλαγή και τον διαμοιρασμό πληροφορίας με σημασιολογικά σαφή τρόπο, είναι η ύπαρξη κοινώς αποδεκτών και σαφώς ορισμένων σχημάτων. Παρακάτω θα γίνει αναλυτική αναφορά στα βασικά στοιχεία ορισμού τέτοιων σχημάτων, οι πλήρεις προδιαγραφές των οποίων περιγράφονται στο [BG00].

Εκτός όμως των παραπάνω απλών προτάσεων που αναφέρθηκαν στην περίπτωση της υποστασιοποίησης πολλές φορές είναι απαραίτητη η αναπαράσταση πιο πολύπλοκων προτάσεων. Για παράδειγμα, η πρόταση:

“Ο τίτλος της σελίδας με URI <http://www.nga.org> είναι National Gallery Of Art”.
μοντελοποιείται ως εξής:



Σχήμα 3.7 Υποστασιοποιημένη Δήλωση

Όπως μπορούμε να παρατηρήσουμε και από το σχήμα 3.7, η αναπαράσταση των υποστασιοποιημένων προτάσεων με απλούς κατευθυνόμενους γράφους είναι πολύπλοκη και δυσνόητη και γίνεται ακόμα δυσκολότερη στην περίπτωση που επιθυμούμε να υποστασιοποιήσουμε όχι μόνο μία πρόταση αλλά ένα σύνολο προτάσεων οι οποίες έχουν ένα κοινό χαρακτηριστικό, όπως για παράδειγμα κοινή πηγή προέλευσης ή κοινή ημερομηνία στην οποία έχουν λεχθεί.

Ένα σύνολο προτάσεων αποτελεί ένα υπογράφο του γράφου των δεδομένων ο οποίος πρέπει να υποστασιοποιηθεί. Η έννοια του υπογράφου παραπέμπει στην θεωρία των υπεργράφων από την οποία δανειζόμαστε στοιχεία προκειμένου να προτείνουμε ένα μοντέλο υπεργράφων για το RDF το οποίο θα απλουστεύσει την γραφική αναπαράσταση των περιγραφών του RDF και ειδικά αυτή της υποστασιοποίησης.

Στην επόμενη παράγραφο θα γίνει περιγραφή της έννοιας του υπεργράφου και του υπογράφου και θα δοθούν οι ορισμοί αυτών προσαρμοσμένοι στο μοντέλο του RDF.

3.3.1 Μοντέλο Υπεργράφων

Υπεργράφος είναι ένας γράφος κάθε κόμβος του οποίου μπορεί να είναι είτε απλός κόμβος είτε ένας υπογράφος, δηλαδή τμήμα του γράφου που αποτελείται από κόμβους και ακμές διασυνδεδεμένες μεταξύ τους. Μία ακμή που συνδέει ένα κόμβο με ένα υπογράφο ονομάζεται υπερακμή.

Ορισμός 1. (Υπεργράφος) Ένας (κατευθυνόμενος) υπεργράφος είναι μία τριάδα $HG = (V, E_d, E_u)$ όπου:

- V είναι ένα σύνολο κόμβων
- E_d είναι ένα σύνολο κατευθυνόμενων υπερακμών
- E_u είναι ένα σύνολο μη-κατευθυνόμενων υπερακμών

Μία κατευθυνόμενη υπερακμή $(M, n) \in E_d$, όπου $M \subseteq V$ και $n \in V$ μπορεί να συνδέει

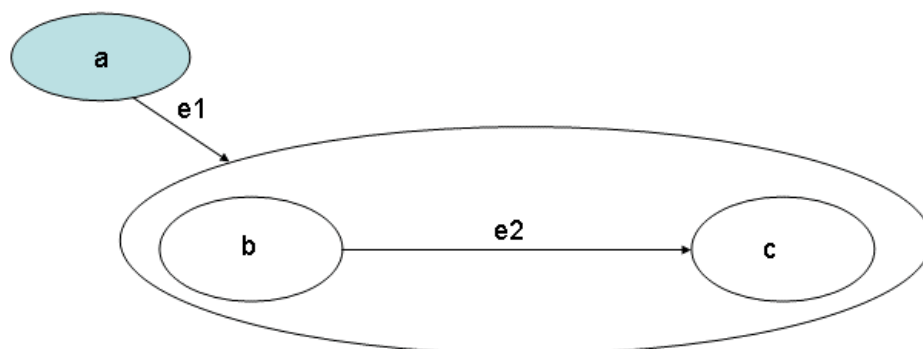
- ένα κόμβο του υπεργράφου με ένα άλλο κόμβο αυτού
- ένα κόμβο του υπεργράφου με ένα υπογράφο

αλλά,

ΔEN μπορεί να συνδέει ένα υπογράφο με ένα άλλο υπογράφο.

Μη-κατευθυνόμενη υπερακμή είναι απλά ένα υποσύνολο των κόμβων V και των κατευθυνόμενων υπερακμών E_d ενός υπεργράφου.

Παρακάτω εξετάζουμε μερικά παραδείγματα υπεργράφων.

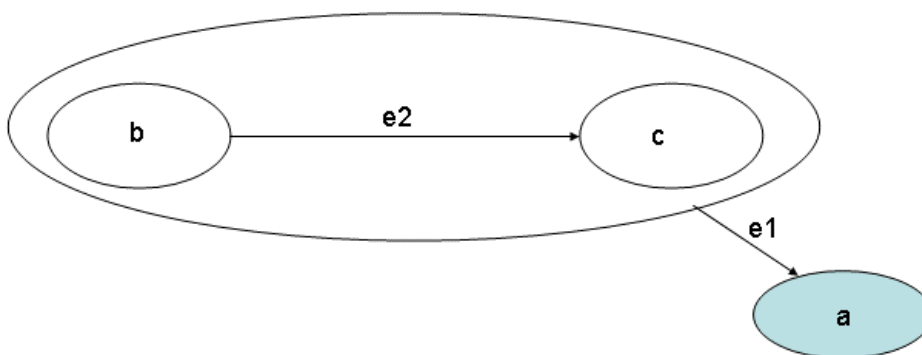


Σχήμα 3.8 Παράδειγμα υπεργράφου

Στο παράδειγμα του σχήματος 3.8 φαίνεται ένας υπεργράφος όπου:

- $V = (a, b, c)$
- $E_d = (e1, e2)$
- $E_u = (b, c, e2)$
- η κατευθυνόμενη υπερακμή με την ετικέτα $e1$ συνδέει τον κόμβο a , με τον υπογράφο $(b, c, e2)$ και τέλος
- η κατευθυνόμενη υπερακμή με την ετικέτα $e2$ συνδέει τον κόμβο b , με τον κόμβο c .

Παρακάτω εξετάζουμε ένα ακόμα παράδειγμα όπου μία κατευθυνόμενη υπερακμή ξεκινάει από ένα υπογράφο και καταλήγει σε ένα κόμβο.

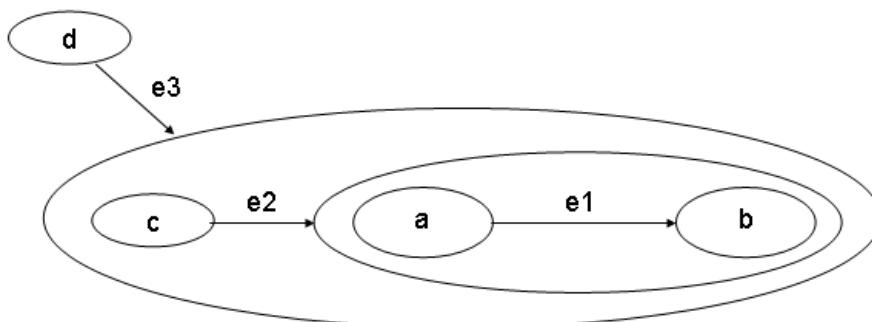


Σχήμα 3.9 Απλό Παράδειγμα υπεργράφου

Στο παράδειγμα του σχήματος 3.9 έχουμε ένα υπογράφο όπου:

- $V = (a, b, c)$
- $E_d = (e1, e2)$
- $E_u = (b, c, e2)$
- η κατευθυνόμενη υπερακμή με την ετικέτα $e1$ ξεκινάει από τον υπογράφο $(b, c, e2)$ και καταλήγει στον κόμβο a και τέλος
- η κατευθυνόμενη υπερακμή με την ετικέτα $e2$ συνδέει τον κόμβο b , με τον κόμβο c .

Το μοντέλο των υπεργράφων υποστηρίζει εμφωλευμένους γράφους σε θεωρητικά άπειρο βάθος. Για παράδειγμα στο σχήμα 3.10 έχουμε μία περίπτωση εμφωλευμένου υπεργράφου σε βάθος δυο.



Σχήμα 3.10 Παράδειγμα εμφωλευμένου υπεργράφου

Ορισμός 2. (Υπογράφος)

Έστω $G = (V, E_d, E_u)$ ένας υπεργράφος. Κάθε μη κατευθυνόμενη ακμή $eu \in E_u$ επάγει ένα υπογράφο του G , $G' = G = (V', E_d', E_u')$ τέτοιο ώστε

1. $V' = e_u$
2. $E_d' = \{e_d \mid (e_d = (M, n)) \in E_d \ \& \ (M \cup \{n\}) \subseteq e_u\}$ και
3. $E_u' = \{e_u' \mid (e_u' \in E_u \ \& \ e_u' \in e_u)\}$

Στο σχήμα 3.10 υπάρχουν δύο υπογράφοι οι

$$e_{u1} = (a, b, e1)$$

$$e_{u2} = (c, e2, e_{u1}) = (c, e2, a, b, e1)$$

Το μοντέλο των υπεργράφων προσφέρει πολλά πλεονεκτήματα στην γραφική αναπαράσταση της υποστασιοποίησης. Πρώτα από όλα, απλουστεύει σημαντικά τους γράφους των υποστασιοποιημένων προτάσεων. Επίσης, με τη βοήθεια των υπεργράφων κάνει εύκολη την αναπαράσταση προτάσεων της μορφής “Η Mary είπε ότι “Ο John είπε ότι “Ο Peter είναι ζωγράφος”””, δηλαδή εμφωλευμένων προτάσεων και αφηγηματικών δομών. Για παράδειγμα, η παραπάνω πρόταση μπορεί εύκολα να αναπαρασταθεί με ένα γράφο παρόμοιο με αυτό του σχήματος 3.10. Επίσης, λόγω της ύπαρξης της έννοιας των υπογράφων δίνεται η δυνατότητα να υποστασιοποιηθούν τμήματα ενός υπεργράφου και να αποδοθούν σε αυτά κάποιες κοινές ιδιότητες. Παραδείγματος χάριν, είναι δυνατόν να υποστασιοποιηθούν κάποιοι υπογράφοι και να αποδοθεί σε αυτούς η ιδιότητα ότι ειπώθηκαν από την Mary ή ότι τροποποιήθηκαν τελευταία φορά στις 07-06-2002. Μία λειτουργία η οποία θα είναι ιδιαίτερα χρήσιμη σε εφαρμογές πρακτόρων (agents) όπου πολλές φορές απαιτείται η ανταλλαγή πληροφοριών.

3.4 Χρήσεις της Υποστασιοποίησης

Η υποστασιοποίηση στον τομέα του RDF αποτελεί ένα από τα χαρακτηριστικά αυτού που έχουν δεχτεί την περισσότερη κριτική, καθώς μπορεί να αλλάξει την αξία «δεδομένων» που έχουν ήδη ισχυριστεί. Από την μία μεριά, η ομάδα του KR (Knowledge representation) ισχυρίζεται ότι η υποστασιοποίηση ίσως να είναι σημαντική αλλά υπάρχει ασάφεια στην διαχείριση της, καθώς ένα γεγονός μπορεί αρχικά να ισχυριστεί και στην συνέχεια να εμφανιστεί μία άλλη πρόταση η οποία να επηρεάσει την αλήθεια αυτού του γεγονότος. Επίσης, ισχυρίζεται ότι πρέπει να μεταχειρίζεται με προσοχή καθώς ξεφεύγει από τα πλαίσια της λογικής πρώτης τάξης. Από την άλλη μεριά, η ομάδα του W3C (και άλλοι), υποστηρίζουν ότι η υποστασιοποίηση επιτρέπει την παράθεση αποσπασμάτων και επομένως είναι οπωσδήποτε απαραίτητη σε ένα δυναμικό περιβάλλον όπως αυτό του Παγκόσμιου Ιστού, όπου δεν είναι ξεκάθαρο αν μία πηγή είναι αξία εμπιστοσύνης ή όχι.

Η εργασία αυτή ενστερνίζεται την τελευταία άποψη, και επιχειρεί να αποδείξει την αξία της υποστασιοποίησης, χωρίς να αρνείται ότι η αυτή αποτελεί μια έννοια η οποία

πρέπει να χειρίζεται με προσοχή. Η κύρια χρησιμότητα της υποστασιοποίησης είναι ότι παρέχει τη δυνατότητα ελέγχου της *αξιοπιστίας των μεταδεδομένων*. Ένας όρος που χρησιμοποιείται ευρύτατα τελευταία είναι αυτός του “Web Of Trust” [BKD+01], που αποτελεί μια προσπάθεια του W3C, ώστε τα δεδομένα του Παγκόσμιου Ιστού να γίνουν αξιόπιστα. Πράγματι, η αξιοπιστία των δεδομένων αποτελεί πλέον πολύ μεγάλη ανάγκη στο περιβάλλον του Παγκόσμιου Ιστού, όπου ο καθένας μπορεί να παραθέτει απόψεις και στοιχεία χωρίς έλεγχο. Με τη βοήθεια της υποστασιοποίησης, τα δεδομένα αποκτούν ταυτότητα και αξιοπιστία καθώς η υποστασιοποίηση παρέχει τη δυνατότητα αποθήκευσης πληροφοριών για την *προέλευση (provenance)* των μεταδεδομένων. Πράγματι, τα μεταδεδομένα μπορούν, χάρη στην υποστασιοποίηση, να συνοδεύονται από την πηγή προέλευσης τους, έτσι θεωρούνται αξιόπιστα μόνο αυτά που προέρχονται από κάποιες συγκεκριμένες πηγές, όπως για παράδειγμα ένας επιστημονικός οργανισμός. Αυτό είναι χρήσιμο, στην περίπτωση μίας εφαρμογής ενός πράκτορα (agent) που αναζητά πληροφορίες για ένα συγκεκριμένο θέμα στον Παγκόσμιο Ιστό και μπορεί να τεθεί ο περιορισμός οι πληροφορίες αυτές να προέρχονται μόνο από συγκεκριμένες πηγές. Έτσι, μία πρόταση ή ένα σύνολο προτάσεων μπορούν να συνοδεύονται από στοιχεία όπως ποια είναι η πηγή από την οποία προέρχονται ή σε ποια χρονική στιγμή λέχθηκαν. Με αυτόν τον τρόπο το RDF υποστηρίζει την αποθήκευση μεταδεδομένων για τα ίδια τα μεταδεδομένα. Το RDF λοιπόν αποκτά ένα ακόμα χαρακτηριστικό που του δίνει ένα σημαντικό πλεονέκτημα και το εφιστά χρησιμοποιήσιμο σε μοντέλα εμπιστοσύνης (trust models).

Το RDF υποστηρίζει μόνο πρώτης τάξεως προτάσεις. Ωστόσο, μέσω της υποστασιοποίησης, όπως αναφέρθηκε και παραπάνω, μπορούμε να «φτιάξουμε» προτάσεις για άλλες προτάσεις, τις λεγόμενες προτάσεις υψηλότερης τάξεως. Οι προτάσεις υψηλότερης τάξης δίνουν πολλές δυνατότητες. Πρώτα απ’ όλα επιτρέπουν την *έκφραση πεποιθήσεων ή απόψεων*. Για παράδειγμα, προτάσεις της μορφής «Ο Παύλος ισχυρίζεται ότι ο Picasso ζωγράφησε τη Guernica». Η δυνατότητα αυτή είναι πολύ χρήσιμη για συλλογή απόψεων σε επιστημονικά θέματα τα οποία παραμένουν ανοικτά και υπάρχουν απόψεις για αυτά οι οποίες δίστανται. Μία άλλη χρησιμότητα των προτάσεων υψηλότερης τάξεως είναι για την εκτίμηση ή την κατάταξη των πληροφοριών των μεταδεδομένων (rating). Για παράδειγμα, το RDF μπορεί να χρησιμοποιηθεί για να βαθμολογηθεί μία πρόταση (triple), όπως σε μία εφαρμογή όπου με σελιδοδείκτες του Παγκόσμιου Ιστού όπου οι χρήστες διαμοιράζονται τους σελιδοδείκτες και επίσης θέλουν να βαθμολογούν το κατά πόσο ένας σελιδοδείκτης είναι χρήσιμος, έτσι η πρόταση “Το www.w3c.org/rdf είναι ιστοσελίδα για το RDF” μπορεί να βαθμολογηθεί από ένα χρήστη με 10 ενώ από κάποιον άλλο με 6.

Επίσης, η υποστασιοποίηση κάνει εφικτή την αναπαράσταση αφηγηματικών δομών (narrative structures) σε ένα μοντέλο δεδομένων RDF. Για παράδειγμα, μπορεί να αναπαρασταθεί η πρόταση «Ο Πέτρος είπε ότι «ο Γιώργος είπε ότι «Η Σοφία πιστεύει ότι «ο Picasso ζωγράφησε την Guernica»»»». Το βάθος της αφηγηματικής δομής μπορεί να είναι θεωρητικά άπειρο. Η δυνατότητα υποστήριξης αυτής της δομής κάνει το RDF ακόμα πιο ελκυστικό για τη χρήση του σε πληθώρα εφαρμογών που απαιτούν τέτοιες δυνατότητες.

Τέλος, ένα τελευταίο αλλά εξίσου σημαντικό θέμα είναι αυτό των *εκδόσεων ενός RDF σχήματος (schema versioning)*, στο οποίο μπορεί να συμβάλει ουσιαστικά η υποστασιοποίηση. Ένα RDF σχήμα είναι απαραίτητο για την αποθήκευση των μεταδεδομένων. Ωστόσο, το σχήμα αυτό πολλές φορές είναι απαραίτητο να

τροποποιηθεί και οι τροποποιήσεις που υφίστανται πρέπει να αποθηκεύονται καθώς είναι πολύτιμες για τον διαχειριστή της βάσης των μεταδεδομένων είτε για την περίπτωση που επιθυμεί να επαναφέρει το RDF σχήμα σε μία προηγούμενη κατάσταση είτε για την διατήρηση του ιστορικού των αλλαγών τις οποίες έχει υποστεί το σχήμα μέχρι να περιέλθει στην τρέχουσα κατάσταση. Ένας λόγος παραπάνω για την αποθήκευση τέτοιων πληροφοριών είναι όταν υπάρχουν παραπάνω από ένας διαχειριστής της βάσης των μεταδεδομένων οπότε και πρέπει να κρατούνται πληροφορίες τόσο για το ποιος έκανε τις αλλαγές στο σχήμα όσο και πότε έγιναν αυτές. Στην περίπτωση αυτή έχουμε να κάνουμε με υποστασιοποίηση σε επίπεδο σχήματος. Βέβαια, η υποστασιοποίηση σε επίπεδο σχήματος δεν διαφέρει για παράδειγμα πολύ από αυτήν σε επίπεδο δεδομένων. Ένα παράδειγμα υποστασιοποίησης σε επίπεδο σχήματος είναι όταν θέλουμε να αναπαραστήσουμε την πληροφορία “‘Η κλάση Painter είναι υποκλάση της κλάσης Creator” ανήκει στην Version1” ή “Η ιδιότητα Paints είναι υποιδιότητα της ιδιότητας Creates” προστέθηκε από την Σοφία”.

Ωστόσο, παρά τα θετικά της υποστασιοποίησης, ελάχιστα είναι τα συστήματα εκείνα που την υποστηρίζουν (παρ. 3.5). Το μείζον πρόβλημα όσον αφορά στην υποστασιοποίηση είναι η πολυπλοκότητα αυτής και η δυσκολία υλοποίησης της. Γενικά, θεωρείται δύσκολη η υλοποίηση της υποστασιοποίησης προγραμματιστικά. Επίσης, η απαίτηση του W3C κάθε υποστασιοποιημένη πρόταση να αποτελείται από τέσσερα στοιχεία (rdf:subject, rdf:object, rdf:predicate, rdf:Statement) συνεπάγεται ότι στην περίπτωση που έχουμε να διαχειριστούμε μεγάλο όγκο δεδομένων, όπως συμβαίνει συχνά στον Παγκόσμιο Ιστό, η εφαρμογή μας να γίνεται πολύ απαιτητική σε χωρητικότητα.

Όπως γίνεται αντιληπτό η υποστασιοποίηση προσφέρει σημαντικά πλεονεκτήματα στο RDF καθώς με τη μετατροπή των προτάσεων (τριάδων) σε πολίτες πρώτης κατηγορίας είναι δυνατή η διατύπωση ισχυρισμών για τα μεταδεδομένα, η υποστήριξη ψηφιακών υπογραφών και γενικότερα η αύξηση της αξιοπιστίας των μεταδεδομένων. Με αυτό τον τρόπο το RDF εξυπηρετεί καλύτερα το στόχο του που είναι η μεταφορά και ο διαμοιρασμός πληροφοριών μεταξύ των εφαρμογών του Ιστού και παράλληλα παρέχει και την απαιτούμενη εμπιστοσύνη στα μεταδεδομένα.

3.5 Υπάρχουσες Υλοποιήσεις

3.5.1 Jena

Το Jena αναπτύχθηκε από την Εταιρία Hewlett-Packard. Είναι μία συλλογή από εργαλεία της RDF γραμμένα σε Java που περιλαμβάνουν: ένα Java API για μοντέλο/γράφο, ένα RDF Parser (που παρέχει υποστήριξη για το φιλτράρισμα N-Τριπλέτων), ένα σύστημα επερωτήσεων βασισμένο στο RDQL, υποστήριξη κλάσεων για οντολογίες της DAML+OIL και μόνιμη αποθήκευση σε BerkeleyDB και διάφορες άλλες υλοποιήσεις αποθήκευσης. Λόγω της αφαιρετικής αποθήκευσης, η Jena δίνει τη δυνατότητα ενσωμάτωσης νέων υποσυστημάτων αποθήκευσης. Για την διευκόλυνση της επερωτήσης, η Jena παρέχει προτασιο-κεντρικές μεθόδους για τη διαχείριση ενός RDF μοντέλου ως ένα σύνολο από RDF τριάδες και πόρο-κεντρικές μεθόδους για τη διαχείριση ενός RDF μοντέλου ως ένα σύνολο πόρων οι οποίοι έχουν ιδιότητες, καθώς και υποστήριξη για RDF πολυσύνολα (containers).

3.5.1.1 Υποστήριξη της Υποστασιοποίησης στο Jena

Το Jena παρέχει μία συντομευμένη μορφή για την υποστήριξη της υποστασιοποίησης. Μέχρι την έκδοση 1.5.0 το Jena δεν υποστήριζε ούτε καν αυτή την μορφή της μερικής υποστασιοποίησης. Η έκδοση 1.5.0 υποστηρίζει πλήρως το μοντέλο της υποστασιοποίησης στην σχεσιακή βάση δεδομένων.

3.5.1.2 Μοντέλο Υποστασιοποίησης στο Jena

Στο Jena η Πρόταση (Statement) είναι υπο-κλάση του Πόρου (Resource). Αυτό συνεπάγεται ότι οι προτάσεις μπορούν να κατασκευαστούν με άλλες προτάσεις ως υποκείμενο ή αντικείμενο αυτών και τέτοιες προτάσεις μπορούν να αποθηκευτούν ως στιγμιότυπα του Μοντέλου και να προσπελαύνονται με τη χρήση του Jena API.

Το παραπάνω αποτελεί μια συντομευμένη μορφή υποστασιοποίησης, που θεωρείται χρήσιμη για τον χρήστη καθώς επιτρέπει σε μια πρόταση να αναφερθεί σε μία άλλη πρόταση ως υποκείμενο ή ως αντικείμενο, χωρίς να απαιτείται αποκλειστικά η δημιουργία της τετράδας των υποστασιοποιημένων προτάσεων. Αντίθετα, οι προδιαγραφές της RDF επιτρέπουν μόνο την έννοια των κόμβων του γράφου αναπαριστώντας τις υποστασιοποιημένες προτάσεις μόνο μέσω αποκλειστικών τετράδων υποστασιοποίησης.

Σύμφωνα με τον ορισμό του Jena για την υποστασιοποίηση: *Υποστασιοποίηση είναι ο όρος που χρησιμοποιείται όταν μία πρόταση είναι το υποκείμενο ή το αντικείμενο μίας άλλης πρότασης.* Η υλοποίηση εξετάζει μια απλή άποψη. Μία πρόταση είναι μία υποκλάση του πόρου, και σαν τέτοια μπορεί να είναι το υποκείμενο ή το αντικείμενο μίας άλλης πρότασης. Επιπλέον, είναι δυνατόν μία πρόταση s_2 που είναι μέλος ενός μοντέλου m , να κάνει μία πρόταση σχετικά με μία άλλη πρόταση s_1 , χωρίς η s_1 να είναι μέλος του μοντέλου m .

Θεωρούμε την παρακάτω «τριάδα» το αντικείμενο της οποίας είναι μία άλλη τριάδα:

((foo, p, "value"), q, "annotation")

Αυτό το παράδειγμα μπορεί να κατασκευαστεί απευθείας στο Jena και να αποθηκευτεί εσωτερικά σε μία στενογραφημένη μορφή. Παρόλα αυτά, αυτή δεν είναι μία «νόμιμη» RDF τριάδα και μπορεί να αναπαρασταθεί στην RDF μόνο με τις παρακάτω τριπλέτες:

(#, rdf:type, rdf:Statement)

(#, rdf:subject, foo)

(#, rdf:predicate, p)

(#, rdf:object, "value")

(#, q, "annotation")

Όπου το # δείχνει ένα τοπικό αναγνωριστικό. Πράγματι αν το παραπάνω παράδειγμα κατασκευαστεί στο Jena και γραφεί στο RDF τότε το Jena θα χρησιμοποιήσει μία ετικέτα `rdf:ID` με ένα παραγόμενο τοπικό ID για να διασύνδεει την αναφερόμενη πρόταση αυτό που αναφέρεται σε αυτήν.

Έτσι από την μια μεριά η συντομευμένη μορφή της υποστασιοποίησης είναι ελκυστική. Φαίνεται να αποφεύγεται το επιπλέον κόστος της δημιουργίας τεσσάρων φαινομενικά περιττών προτάσεων και είναι βολικό προγραμματιστικά. Από την άλλη μεριά η ανάγκη συμμόρφωσης στις προδιαγραφές της RDF οδηγεί σε ασυμβατότητες. Αυτό το πρόβλημα έχει ως αποτέλεσμα την ασυνεπή υποστήριξη για την συντομευμένη υποστασιοποίηση με το Jena Interface. Υποστηρίζεται από το ModelMem και τους συγγραφείς της RDF/XML αλλά όχι από τους συγγραφείς της RDF/XML/ABBREV ή της N-TRIPLE, ούτε από τους αναγνώστες ή κάποιο από τα εμμένων αποθηκευτικά συστήματα.

Βάση της ανάδρασης των χρηστών ότι αυτό το χαρακτηριστικό είναι επιθυμητό, απαιτείται ένα ξεκαθάρισμα. Συγκεκριμένα, απαιτείται το σύστημα της σχεσιακής βάσης δεδομένων να υποστηρίζει αυτή τη μορφή και προτιμητέα με τέτοιο τρόπο ώστε να αποφευχθεί το επιπλέον κόστος της ρητής υποστασιοποίησης.

3.5.1.3 Υλοποίηση της σχεσιακής βάσης δεδομένων (RDB)

Για την υποστήριξη του συντομευμένου μοντέλου υποστασιοποίησης σε ένα σχεσιακό σύστημα βάσης δεδομένων έχουν γίνει οι παρακάτω αλλαγές στους πίνακες της σχεσιακής βάσης.

- σε όλες οι προτάσεις ανατίθεται ένα αναγνωριστικό βάσης (`database id`) στο οποίο μπορεί να γίνει αναφορά από άλλους πίνακες.
- Οι προτάσεις έχουν ένα πεδίο `is_reified`. Αν αυτό το flag έχει τεθεί τότε η πρόταση είναι παρούσα στο μοντέλο μόνο σε υποστασιοποιημένη μορφή και δεν θα εμφανιστεί στα αποτελέσματα της λειτουργίας της λίστας των Προτάσεων (`listStatements`).
- Οι προτάσεις που έχουν υποστασιοποιηθεί (γίνεται αναφορά σε αυτές από άλλες προτάσεις ή έχουν ένα συγκεκριμένο URI ανεξάρτητα από το αν γίνεται αναφορά σε αυτές ή όχι) έχουν ένα πεδίο εισόδου `reification` που είναι ο δείκτης μίας εισόδου `rdf_resources` για τον πόρο ο οποίος υποστασιοποιεί αυτή την πρόταση.

Ο πίνακας *Rdf_Statements* (*statement_id*, *subject*, *predicate*, *object*, *object_literal*, *model*, *reification*, *is_reified*) χρησιμοποιείται για να αναπαραστήσει τις υποστασιοποιημένες προτάσεις. Το πεδίο *statement_id*, αποτελεί το αναγνωριστικό της πρότασης και δεν μπορεί να είναι κενό (`null`). Το πεδίο *subject*, είναι το αναγνωριστικό (`id`) του πόρου που είναι υποκείμενο της πρότασης και δεν μπορεί να είναι κενό (`null`). Το πεδίο *predicate*, είναι το αναγνωριστικό (`id`) του πόρου ο οποίος αποτελεί το κατηγορούμενο της πρότασης και όπως με τα παραπάνω δεν μπορεί να είναι κενό. Το πεδίο *object*, είναι αντίστοιχα το αναγνωριστικό (`id`) του πόρου που αποτελεί το υποκείμενο της πρότασης. Το πεδίο *object_isliteral*, χρησιμοποιείται όταν το

αντικείμενο της πρότασης είναι λεκτικό (literal). Το πεδίο *model*, αντιστοιχεί στο αναγνωριστικό (id) του μοντέλου στο οποίο αναφέρεται η πρόταση. Το πεδίο *reification*, είναι δείκτης στην εγγραφή του πίνακα *Rdf_Resources*, δηλαδή στον πόρο που υποστασιοποιεί αυτή την πρόταση, και είναι κενό (null) αν η πρόταση δεν υποστασιοποιείται από κάποιον πόρο. Τέλος, το πεδίο *is_reified*, είναι αληθής μόνο αν η πρόταση υπάρχει μόνο σε υποστασιοποιημένη μορφή και στην πραγματικότητα δεν είναι παρούσα.

- Οι πόροι μπορεί να έχουν ένα επιπλέον πεδίο *statement* που χρησιμοποιείται όταν ο πόρος είναι η υποστασιοποίηση κάποιας πρότασης, αν το πεδίο αυτό δεν είναι κενό (null) περιέχει ένα δείκτη (index) στον πίνακα *rdf_statements*, που ανταποκρίνεται σε αυτόν τον πόρο.

Ο πίνακας ***Rdf_Resources*** (*id, namespace, statement, localname*). Το πεδίο *id* αποτελεί το αναγνωριστικό του πόρου. Το πεδίο *namespace*, αντιστοιχεί στον χώρο ονοματοδοσίας (namespace) του πόρου. Το πεδίο *statement* χρησιμοποιείται αν ο πόρος δείχνει σε μία υποστασιοποιημένη πρόταση, διαφορετικά είναι κενό. Τέλος, το πεδίο *localName* αντιστοιχεί στο τοπικό όνομα του πόρου.

- Ορίζονται επιπλέον οι παρακάτω συναρτήσεις: `getReifiedStatement`, `insertReifiedStatement`, `insertREsourceAsStatement`, `listReifiedStatements`, `assertReifiedStatements` και `setStatementReification`.

`getReifiedStatement` – Βρίσκει μία υποστασιοποιημένη πρόταση παίρνοντας σαν παράμετρο το id της υποστασιοποιημένης πρότασης

`insertReifiedStatement` – Εισάγει μία υποστασιοποιημένη πρόταση και συγκεκριμένα στον πίνακα `RDF_STATEMENT`

`insertResourceAsStatement` – Εισάγει ή αντιγράφει ένα πόρο ως πρόταση

`listReifiedStatements` – Επιστρέφει μια λίστα με όλες τις υποστασιοποιημένες προτάσεις

`assertReifiedStatements` – Αλλάζει το πεδίο *is_Reified* μίας πρότασης σε ψευδές ώστε να είναι ορατή αυτή, δηλ. να υπάρχει και στην πραγματικότητα και όχι μόνο ως υποστασιοποιημένη πρόταση.

`setStatementReification` – Αλλάζει το πεδίο *Reification* ώστε να δείχνει στον αντίστοιχο πόρο

Επιπλέον στα `supprt/layout configuration files` ορίζεται μία λογική ιδιότητα η `supportsJenaReification`, αν αυτή είναι αληθής τότε εσωτερικά στο σύστημα θα υλοποιηθούν οι παραπάνω επεκτάσεις των πινάκων έτσι ώστε να είναι δυνατή η υποστήριξη της υποστασιοποίησης. Αν η ψευδής, τότε το σύστημα επανέρχεται στις υποθέσεις που είχαν γίνει στην προ-υποστασιοποίησης έκδοση.

Μερικά θέματα του W3C's RDF Model and Syntax Specification απαιτούν προσεκτικό διάβασμα και ερμηνεία για να παραχθεί μία «σωστή» υλοποίηση. Έχουν προκύψει θέματα σχετικά με τους ανώνυμους πόρους (anonymous resources), την υποστασιοποίηση (reification) και τους RDF γράφους (RDF Graphs). Προκειμένου να υλοποιηθεί το Jena, ένα RDF API σε Java γίνονται κάποιες παραδοχές πάνω στις οποίες στηρίζεται η υλοποίηση αυτού.

3.5.2 Triple

Η γλώσσα TRIPLE είναι μία γλώσσα επερωτήσεων, εξαγωγής συμπερασμάτων και μετασχηματισμού που αναπτύχθηκε σαν μια ομαδική δουλειά από τον Stefan Decker (Stanford University Database Group) και τον Michael Sintek (DFKI GmbHKaiserslautern, Knowledge Management Department and Stanford University Database Group).

Η φύση της TRIPLE βασίζεται στην Λογική Horn και στην F-Logic και επιχειρεί να υποστηρίξει εφαρμογές που απαιτούν RDF συλλογιστική (reasoning) και μετασχηματισμούς. Για παράδειγμα, μπορεί να παρέχει μηχανισμούς επερώτησης πόρων του δικτύου (web resources) με ένα δηλωτικό τρόπο. Σε αντίθεση με πολλές άλλες γλώσσες επερώτησης της RDF, η TRIPLE επιτρέπει στη σημασιολογία των γλωσσών που στηρίζονται στην κορυφή της RDF, όπως το RDF Schema [BG00] και το Topic Maps, να ορίζεται με κανόνες, αντί να υποστηρίζει την ίδια λειτουργικότητα με μία ενσωματωμένη σημασιολογία. Σε εκείνες τις περιπτώσεις όπου ο ορισμός της σημασιολογίας της γλώσσας δεν είναι εύκολα δυνατός με κανόνες, όπως για παράδειγμα στην περίπτωση της DAML+OIL [CHH⁺01], η TRIPLE παρέχει πρόσβαση σε εξωτερικά προγράμματα, όπως για παράδειγμα σε description logics classifiers [CR]. Επομένως, η TRIPLE υποστηρίζει δύο διαφορετικά είδη επιπέδων: τις συντακτικές επεκτάσεις της Λογικής Horn για την υποστήριξη των βασικών δομών του RDF, όπως τους πόρους (resources) και τις προτάσεις και τμήματα για την σημασιολογική επέκταση του RDF, όπως το RDF Schema [BG00], το OIL και το DAML+OIL [CHH⁺01] που υλοποιούνται είτε απευθείας στην TRIPLE είτε μέσω αλληλεπίδρασης με εξωτερικά συλλογιστικά στοιχεία, όπως ένας DL ταξινομητής (classifier). Συγκεκριμένα, η TRIPLE παρέχει υποστήριξη για πόρους και χώρους ονομάτων (namespaces), συντμήσεις όπως για παράδειγμα `isa:=rdfSubClassOf`, μοντέλα (σύνολα από RDF προτάσεις), υποστασιοποίηση (reification) και κανόνες με τμήμα για εκφράσεις. Η TRIPLE επίσης παρέχει συναρτήσεις Skolem, οι οποίες όταν χρησιμοποιούνται σε κανόνες, μπορούν να χρησιμοποιηθούν για να μετατρέψουν ένα ή περισσότερα μοντέλα (π.χ. ένα σύνολο RDF προτάσεων) σε ένα νέο μοντέλο. Η δυνατότητα αυτή είναι ιδιαίτερα χρήσιμη για την αντιστοίχιση ή την ενσωμάτωση οντολογιών. Επιπλέον, αντί για ορισμούς υποκειμένου (subject), κατηγορούμενου (predicate) και αντικειμένου (object), η TRIPLE επιτρέπει την χρήση εκφράσεων μονοπατιού (path expressions). Για παράδειγμα, μπορούμε να ορίσουμε κανόνες (horn) οι οποίοι ψάχνουν για έγγραφα που έχουν ένα συγκεκριμένο θέμα. Η TRIPLE παρέχει μια ASCII σύνταξη που είναι αναγνώσιμη από τον άνθρωπο, καθώς και μία σύνταξη βασισμένη στο RDF που χρησιμεύει στην ανταλλαγή επερωτήσεων και κανόνων, όπως για παράδειγμα για την επικοινωνία μεταξύ των πρακτόρων (agents).

3.5.2.1 Χαρακτηριστικά της Triple

Στην παράγραφο αυτή θα αναφερθούμε στα κύρια χαρακτηριστικά της TRIPLE.

Χώροι Ονοματοδοσίας (Namespaces) και Πόροι (Resources)

Η TRIPLE παρέχει ειδική υποστήριξη για αναγνωριστικά χώρων ονοματοδοσίας και πόρων. Οι χώροι ονοματοδοσίας δηλώνονται μέσω δομών της μορφής *nsabbrev:=namespace*.

Για παράδειγμα,

```
rdf := "http://www.w3c.org/1999/02/22-rdf-syntax-ns#"
```

Οι πόροι γράφονται ως *nsabbrev:name* όπου *nsabbrev* είναι μία συντομογραφία του χώρου ονοματοδοσίας και *name* είναι το τοπικό όνομα του πόρου

Προτάσεις (Statements)

Μία RDF πρόταση (τριπλετα) - εμπνευσμένη από τη σύνταξη της F-Logic - γράφεται με τη μορφή

```
subject[predicate→object]
```

Πολλές προτάσεις που έχουν το ίδιο αντικείμενο μπορούν να συντημηθούν με τη μορφή του λεγόμενου “*modecules*”.

Για παράδειγμα:

```
Stefan[hasAge → 33; isMarried → yes;...]
```

Οι RDF προτάσεις μπορούν να είναι εμφωλευμένες, π.χ:

```
Stefan[hasAge → 33; isMarried → yes;...]
```

Υποστασιοποιημένες Προτάσεις (Reified Statements)

Όπως αναφέρθηκε και παραπάνω η TRIPLE υποστηρίζει την υποστασιοποίηση. Συγκεκριμένα, στην TRIPLE οι υποστασιοποιημένες προτάσεις (reified statements) γράφονται με τη μορφή <πρόταση> και μπορούν να χρησιμοποιηθούν εσωτερικά σε άλλες προτάσεις, επιτρέποντας με αυτό τον τρόπο προτάσεις της παρακάτω μορφής.

```
Stefan[believes→ <Ora[ isAuthorOf → homepage]>
```

3.6 Συμπεράσματα

Η υποστασιοποίηση είναι μία έννοια η οποία θεωρείται ιδιαίτερα “στριφνή” τόσο στην κατανόηση όσο και στην υλοποίηση της και για το λόγο αυτό αποφεύγεται η μελέτη αυτής και “εξοστρακίζεται” από τα συστήματα τα οποία υλοποιούν το RDF. Όλοι σχεδόν οι αναλυτές RDF συστημάτων – πλην των ελαχίστων εξαιρέσεων που αναφέρθηκαν παραπάνω - παραβλέπουν τον μηχανισμό της υποστασιοποίησης. Οι κύριοι λόγοι για τους οποίους συμβαίνει αυτό είναι ότι εισάγονται ζητήματα αλήθειας των υποστασιοποιημένων προτάσεων, καθώς θεωρείται ότι δεν γνωρίζουμε τίποτα

για την αξιοπιστία αυτών, αλλά και λόγω της πολυπλοκότητας των γράφων που αναπαριστούν τις υποστασιοποιημένες προτάσεις καθώς και της δυσκολίας της προγραμματιστικής υλοποίησης. Ωστόσο, η υποστασιοποίηση είναι ένας ιδιαίτερα χρήσιμος μηχανισμός, απαραίτητος σε πολλές κατηγορίες εφαρμογών και η υποστήριξη του από το RDF θα δώσει σε αυτό ώθηση έτσι ώστε να αντικαταστήσει άλλες γλώσσες όπως για παράδειγμα την KIF, η οποία αποτελεί μια γλώσσα ανταλλαγής μεταγνώσης μεταξύ πρακτόρων. Παρά την επιπλέον λειτουργικότητα που μπορεί να προσφέρει ο μηχανισμός της υποστασιοποίησης, δεν έχει γίνει καμμία θεωρητική μελέτη για αυτόν. Ακόμα και οι υπάρχουσες υλοποιήσεις δεν έχουν στηριχθεί σε ένα καλά καθορισμένο θεωρητικό πλαίσιο. Όσον αφορά στο Jena, έχουν γίνει κάποιες θεωρήσεις όπως για παράδειγμα ότι «μια υποστασιοποιημένη πρόταση θεωρείται ως ένας πόρος» αλλά δεν έχει γίνει κάποια ουσιαστική μελέτη της υποστασιοποίησης πριν την υλοποίηση αυτής, στο Triple δε, δεν έχει γίνει απολύτως καμμία μελέτη και απλά εφαρμόζεται ένας μηχανισμός για την αναπαράσταση των υποστασιοποιημένων προτάσεων.

Ωστόσο, αυτό που απαιτείται προκειμένου η υποστασιοποίηση να καταστεί χρήσιμη είναι αρχικά η τεκμηρίωση της προκειμένου να αποσαφηνιστούν τυχόν ασάφειες και μελανά σημεία. Επίσης, χρειάζεται ένας απλούστερος τρόπος αναπαράστασης των υποστασιοποιημένων προτάσεων, όπως αυτός που προτείνεται μέσω του μοντέλου των υπεργράφων και αποτελεί καινοτομία για το RDF καθώς δεν έχει προταθεί κάτι παρόμοιο για την αναπαράσταση των υποστασιοποιημένων προτάσεων. Η αναπαράσταση των υποστασιοποιημένων προτάσεων με τη βοήθεια του υπεργράφου αποτρέπει την επανάληψη αναπαράστασης των υποστασιοποιημένων προτάσεων κάνοντας απλούστερο και περισσότερο κατανοητό τον γράφο που αναπαριστά τα μεταδεδομένα, ωστόσο, το θέμα αυτό αγνοείται παντελώς στις υπάρχουσες υλοποιήσεις.

Προκειμένου, η υποστασιοποίηση να προσφέρει ουσιαστικά πρέπει να υποστηριχθεί τόσο η αποθήκευση όσο και η επερώτηση των υποστασιοποιημένων προτάσεων. Από τις υπάρχουσες υλοποιήσεις μόνο το Jena, παρέχει μηχανισμό αποθήκευσης και ανάκτησης των υποστασιοποιημένων προτάσεων, ενώ το Triple απλά παρέχει ένα μηχανισμό επερώτησης αυτών χωρίς να εξετάζει την μέθοδο αποθήκευσης τους. Όσον αφορά στην επερώτηση, στο Jena υπάρχει η δυνατότητα είτε ανάκτησης όλων των υποστασιοποιημένων προτάσεων, είτε μιας και μόνο υποστασιοποιημένης πρότασης της οποίας το αναγνωριστικό είναι γνωστό. Ωστόσο, μια τέτοια λειτουργικότητα δεν είναι αρκετή διότι αυτό που επιθυμείται σε πραγματικές εφαρμογές είναι η ανάκτηση πληροφοριών βάση κριτηρίων, δυνατότητα η οποία δεν προσφέρεται σε καμμία από τις υπάρχουσες υλοποιήσεις. Στην παρούσα εργασία μελετώνται και προτείνονται τρόποι ανάκτησης πληροφοριών υποστασιοποιημένων προτάσεων βάση κριτηρίων, όπως για παράδειγμα το αντικείμενο της υποστασιοποιημένης πρότασης ή το υποκείμενο αυτής.

Η εργασία μας επιχειρεί μια συνολική μελέτη της υποστασιοποίησης καθώς και να αποσαφηνίσει τυχόν αμφιλεγόμενα ζητήματα. Επίσης μελετά τον τρόπο υποστήριξης της υποστασιοποίησης σε ένα υπάρχον σύστημα και προτείνει όλα όσα είναι απαραίτητα για την υποστήριξη αυτής, στο στάδιο της συντακτικής και σημασιολογικής ανάλυσης, στην σχεσιακή βάση αποθήκευσης μεταδεδομένων και τέλος στην γλώσσα επερωτήσεων RQL.

ΚΕΦΑΛΑΙΟ 4

Προτεινόμενη Αναπαράσταση και Αποθήκευση RDF Μεταδεδομένων σε Σχεσιακά Συστήματα Διαχείρισης Βάσεων Δεδομένων με Υποστήριξη της Υποστασιοποίησης

Σε αυτό το κεφάλαιο θα περιγράψουμε τις απαιτούμενες αλλαγές στην υπάρχουσα σχεσιακή βάση δεδομένων για την αποθήκευση RDF μεταδεδομένων ώστε να είναι δυνατή η υποστήριξη της υποστασιοποίησης. Αρχικά, θα περιγράψουμε την RDF/XML σύνταξη και θα δώσουμε κάποια παραδείγματα. Στην συνέχεια θα παρουσιάσουμε το Συντακτικό Σημασιολογικό Αναλυτή RDF μεταδεδομένων (VRP). Και τέλος, θα παρουσιάσουμε την υπάρχουσα σχεσιακή αναπαράσταση καθώς και τις αλλαγές που προτείνονται σε αυτήν για την υποστήριξη της υποστασιοποίησης.

4.1 RDF/XML Σύνταξη - Παραδείγματα

Στην παράγραφο αυτή θα γίνει μια συνοπτική περιγραφή της RDF/XML σύνταξης η οποία ορίζεται στο κείμενο RDF M&S[LS99] για την αναπαράσταση και ανταλλαγή RDF μεταδεδομένων.

4.1.1 Γενικά Χαρακτηριστικά RDF/XML Σύνταξης

Για την καλύτερη δυνατή κατανόηση της RDF/XML Σύνταξης θα χρησιμοποιηθεί το παράδειγμα του σχήματος 4.1. Οι RDF περιγραφές ενός αρχείου περικλείονται στο στοιχείο *rdf:RDF*. Το στοιχείο *rdf:RDF* μπορεί να παραλείπεται σε περίπτωση που η εφαρμογή “γνωρίζει” ότι το αρχείο περιέχει RDF μεταδεδομένα. Μία RDF περιγραφή δηλώνεται με το στοιχείο *rdf:Description*. Το στοιχείο της RDF περιγραφής εφαρμόζεται συνήθως σε ένα πόρο και περιέχει μια σειρά από XML στοιχεία τα οποία αντιστοιχούν στις ιδιότητες που αποδίδονται στον πόρο. Το URI του πόρου στον οποίο αποδίδεται η περιγραφή τίθεται σαν τιμή του γνωρίσματος (attribute) *rdf:ID* ή του γνωρίσματος *rdf:about*. Τα παραπάνω γνωρίσματα ορίζονται μέσα στο στοιχείο *rdf:Description*. Το γνώρισμα *rdf:ID* χρησιμοποιείται όταν θέλουμε να ορίσουμε και/ή να περιγράψουμε ένα πόρο, ενώ το γνώρισμα *rdf:about* στην περίπτωση που θέλουμε να περιγράψουμε ένα ήδη ορισμένο πόρο.

Οι τιμές των ιδιοτήτων, δηλαδή των στοιχείων μιας περιγραφής, μπορεί να είναι είτε URIs δηλαδή πόροι, είτε λεκτικά (literals) είτε άλλες περιγραφές. Επομένως, μια περιγραφή ενός πόρου μπορεί να περιέχει και περιγραφές άλλων πόρων. Όσον αφορά στις δηλώσεις χώρων ονοματοδοσίας, των οποίων κλάσεις και ιδιότητες χρησιμοποιούνται στις RDF περιγραφές, τοποθετούνται συνήθως στην αρχή μέσα στο στοιχείο `rdf:RDF`.

Στην RDF/XML αναπαράσταση του σχήματος 4.1 περιγράφεται ο πόρος με URI `http://www.nga.gov/`. Η RDF περιγραφή δηλώνει ότι ο πόρος αυτός ανήκει στις κλάσεις (`rdf:type`) `http://www.odp.org/schema.rdf#ExternalPage` και `http://www.arts.org/schema.rf#Art_History`, έχει τίτλο *National Gallery Of Art* και δημιουργό (`odp:creator`) τον πόρο με URI `http://www.person.org/id123`. Ο πόρος με URI `http://www.person.org.id123` έχει όνομα (`odp:first_name`) John και επίθετο (`odp:last_name`) Miller.

```
1 <?xml version="1.0" ?>
2 <rdf:RDF
  xmlns:rdf="http://www.w3c.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdf="http://www.w3c.org/2000/01/rdf-schema#"
  xmlns:odp="http://www.odp.org/schema.rdf">
6 <rdf:Description rdf:about = "http://www.nga.gov">
7   <rdf:type rdf:resource="http://www.odp.org/schema.rdf#ExternalPage"/>
8   <rdf:type rdf:resource="http://www.arts.org/schema.rdf#Art_History"/>
9 <odp:title> National Gallery Of Art </odp:title>
10<odp:creator>
11<rdf:Description rdf:about="http://www.person.org/id123">
12<odp:first_name> John </odp:first_name>
13<odp:last_name> Miller </odp:larst_name>
14</rdf:Description>
15</odp:Creator>
16</rdf:Description>
17</rdf:RDF>
```

Σχήμα 4.1 RDF/XML Αναπαράσταση

Εκτός από την *αναλυτική σύνταξη* την οποία περιγράψαμε παραπάνω στο κείμενο RDF & MS [LS99] παρουσιάζεται και μια *συντομευμένη* RDF/XML σύνταξη. Στην εικόνα 4.2 απεικονίζονται οι RDF/XML περιγραφές του σχήματος 4.1 χρησιμοποιώντας την συντομευμένη σύνταξη. Για οικονομία χώρου παραλείπονται οι δηλώσεις χώρων ονοματοδοσίας και το στοιχείο `rdf:RDF`.

Τα παρακάτω αποτελούν δύο βασικά στοιχεία της σύνταξης:

- Το στοιχείο `rdf:Description` μπορεί να αντικατασταθεί από το URI μιας από τις κλάσεις που αποτελούν τιμές της ιδιότητας `rdf:type`, σε περίπτωση που μια περιγραφή περιέχει τουλάχιστον μια `rdf:type` ιδιότητα. Η αντικατάσταση αυτή είναι ισοδύναμη με την απόδοση της ιδιότητας `rdf:type` στον πόρο. Για παράδειγμα συγκρίνοντας το σχήμα 1 με το σχήμα

2 παρατηρούμε τι το στοιχείο *rdf:Description* έχει αντικατασταθεί από το στοιχείο *odp:ExternalPage*

- Όταν οι τιμές των ιδιοτήτων ενός πόρου είναι αλφαριθμητικά οι ιδιότητες αντί να αντιστοιχούν σε XML στοιχεία μπορούν να δηλωθούν σαν γνωρίσματα του στοιχείου *rdf:Description* στο οποίο περιγράφεται ο πόρος. Στο σχήμα 4.2 στην γραμμή 5 παρατηρούμε πως μετατρέπεται η περιγραφή του πόρου *http://www.person.org/id123* που περιέχεται στην εικόνα 4.1 στις γραμμές 11-14.

Στις RDF/XML περιγραφές μπορεί να περιέχεται ταυτόχρονα *αναλυτική* και *συντομευμένη* σύνταξη ταυτόχρονα. Επίσης, οι ετικέτες μπορεί να είναι είτε ετικέτες κόμβου είτε ετικέτες ακμής. Η RDF/XML σύνταξη επιτρέπει στον συντακτικό αναλυτή να διαφοροποιήσει τις ετικέτες.

```
<odp:ExternalPage rdf:about=»http://www.nga.gov/»>
<rdf :type rdf :resource= »http://www.arts.org/schema.rdf#Art_History »/>
<odp:title> National Gallery Of Art </odp:title>
<odp:creator>
<odp:Person rdf:about = «http://www.person.org/id123» odp:first_name=»John»
  odp:last_name= »Miller»>
</odp:creator>
</odp:ExternalPage >
```

Σχήμα 4.2 Συντομευμένη RDF/XML αναπαράσταση

4.1.2 RDF/XML αναπαράσταση σχημάτων

Στην ενότητα αυτή θα αναπαραστήσουμε RDF σχήματα με τη βοήθεια της RDF/XML σύνταξης. Στο σχήμα 4.3 απεικονίζεται ένα RDF Σχήμα στο οποίο ορίζεται μια ιεραρχία κλάσεων. Για την ακρίβεια ορίζονται οι κλάσεις *Arts*, *Art_History* και *Classical_Studies*. Πρέπει να τονιστεί ότι κάθε κλάση που ορίζουμε πρέπει να έχει μια ιδιότητα *rdf:type* με τιμή *rdfs#Class*. Οι κλάσεις *Art_History* και *Classical_Studies* ορίζονται ως υποκλάσεις της κλάσης *Arts*. Επίσης στην κλάση *Arts_History* αποδίδεται η ιδιότητα *isDefinedBy* με τιμή τον πόρο με URI *http://www.arts.org/schema.rdf* . Παρατηρούμε ότι το στοιχείο *rdf:Description* αντικαθίστανται από το στοιχείο *rdfs#Class* και ότι στις περιγραφές των κλάσεων χρησιμοποιείται το γνώρισμα *rdf:ID* εφόσον θεωρούμε ότι οι κλάσεις ορίζονται στο αρχείο που περιέχονται οι RDF περιγραφές. Τέλος, παρατηρούμε ότι το URI ενός πόρου μπορεί να είναι σχετικό.

```
1 <?xml version=»1.0»?>
2 <rdf:RDF>
  xmlns:rdf=»http://www.w3c.org/1999/02/22-rdf-syntax-ns#»
  xmlns:rdfs=»http://www.w3c.org/2000/01/rdf-schema#» >
5 <rdfs:Class rdf:ID=»Arts» />
6 <rdfs:Class rdf:ID=»Art_History»>
```

```
7 <rdfs:SubClassOf rdf:resource=#Arts>/>
8 <rdfs:isDefinedBy rdf:resource=http://www.arts.org/schema.rdf>/>
9 </rdfs:Class>
10 <rdfs:Class rdf:ID=»Classical_Studies»>
11 <rdfs:SubClassOf rdf:resource=#Arts>/>
12 </rdfs:Class>
13</rdf:RDF>
```

Σχήμα 4.3 RDF Σχήμα – Ιεραρχία Κλάσεων

Ένα άλλο παράδειγμα στο οποίο ορίζονται κλάσεις και ιδιότητες φαίνεται στο σχήμα 4.4. Κάθε ιδιότητα πρέπει να έχει την ιδιότητα `rdf:type` με τιμή `rdf#Property`.

```
<?xml version=»1.0»?>
<rdf:RDF
  xmlns:rdf=»http://www.w3c.org/1999/02/22-rdf-syntax-ns#»
  xmlns:rdfs=»http://www.w3c.org/2000/01/rdf-schema#» >
  <rdfs:Class rdf:ID=»ExternalPage»>
  <rdfs:Class rdf:ID=»Person»>
  <rdf:Property rdf:ID=»title»>
    <rdfs:subPropertyOf rdf:resource=»http://www.dc.org/schema.rdf#title»>
    <rdfs:domain rdf:resource=#ExternalPage>/
    <rdfs:range rdf:resource=»http://www.w3c.org/2000/01/rdf-schema#Literal»/
  </rdf:Property>

  <rdf:Property rdf:ID=»creator»>
    <rdfs:subPropertyOf rdf:resource=»http://www.dc.org/schema.rdf#creator»>
    <rdfs:domain rdf:resource=#ExternalPage>/
    <rdfs:range rdf:resource=#Person>/
  </rdf:Property>

  <rdf:Property rdf:ID=»last_name»>
    <rdfs:domain rdf:resource=#Person>/
    <rdfs:range rdf:resource=» http://www.w3c.org/2000/01/rdf-schema#Literal »/
  </rdf:Property>

  <rdf:Property rdf:ID=»first_name»>
    <rdfs:domain rdf:resource=#Person>/
    <rdfs:range rdf:resource=» http://www.w3c.org/2000/01/rdf-schema#Literal »/
  </rdf:Property>
</rdf:RDF>
```

Σχήμα 4.4 RDF Σχήμα για την Περιγραφή Σελίδων

4.1.3 RDF/XML αναπαράσταση υποστασιοποιημένων δηλώσεων

Στην ενότητα αυτή θα περιγραφεί η RDF/XML αναπαράσταση υποστασιοποιημένων δηλώσεων. Η RDF/XML αναπαράσταση της υποστασιοποιημένης δήλωσης “*Ο τίτλος(odp#title) της σελίδας http://www.nga.gov/ είναι National Gallery Of Art*” φαίνεται στο Σχήμα 4.5. Πρέπει να παρατηρήσουμε ότι ο πόρος που αναπαριστά το μοντέλο της δήλωσης είναι ανώνυμος. Προκειμένου να χρησιμοποιήσουμε ένα πόρο σε μια άλλη περιγραφή, πρέπει να αποδώσουμε στον πόρο αυτόν ένα URI. Αυτό είναι δυνατό με την προσθήκη στο στοιχείο *rdf:Description* του γνώρισματος *rdf:ID* με τιμή το όνομα του πόρου π.χ. *rdf:ID=»stat1*». Στην παρακάτω αναπαράσταση εκτός από τις τέσσερις βασικές ιδιότητες που πρέπει να έχει μια υποστασιοποιημένη δήλωση έχει προστεθεί και η ιδιότητα *s:said*.

```
<rdf:Description>
  <rdf:subject rdf:resource=»http://www.nga.gov»>
  <rdf:predicate rdf:resource=»http://www.odp.org/schema.rdf#title»>
  <rdf:object > National Gallery Of Art </rdf:object>
  <rdf:type rdf:resource=»http://www.w3c.org/1999/02/22-rdf-syntax-
ns#Statement»/>
  <s:said>Sofia </s:said>
</rdf:Description>
```

Σχήμα 4.5 RDF Αναπαράσταση Υποστασιοποιημένης Πρότασης

Στην περίπτωση που η δήλωση που πρόκειται να υποστασιοποιηθεί περιέχεται στα μεταδεδομένα, μπορούμε να δημιουργήσουμε την υποστασιοποιημένη δήλωση ως εξής: Προσθέτουμε το γνώρισμα *rdf:ID* στο στοιχείο που αντιπροσωπεύει την ιδιότητα της δήλωσης με τιμή το όνομα του πόρου που θα αντιστοιχεί στην υποστασιοποιημένη δήλωση. Για παράδειγμα, αν στα μεταδεδομένα υπήρχε η δήλωση “*Ο τίτλος(odp#title) της σελίδας http://www.nga.gov/ είναι National Gallery Of Art*”, προσθέτοντας στο στοιχείο *odp:title* το γνώρισμα *rdf:ID* ορίζουμε το URI της υποστασιοποιημένης δήλωσης.

```
<odp:ExternalPage rdf:about=»http://www.nga.gov/»>
  <odp :title rdf :ID=»stat1»> National Gallery Of Art </odp:title>
</odp:ExternalPage>
```

Με τη βοήθεια της υποστασιοποίησης είναι δυνατή η δημιουργία εμφωλευμένων προτάσεων. Για παράδειγμα, η πρόταση “*Ο Gregory είπε ότι “Ο τίτλος της σελίδας http://www.nga.gov/ είναι National Gallery Of Art”*” μπορεί να αναπαρασταθεί ως εξής:

```
<Description rdf:about=»Gregory»>
  <s:said rdf:resource =»stat1» />
</Description>
```

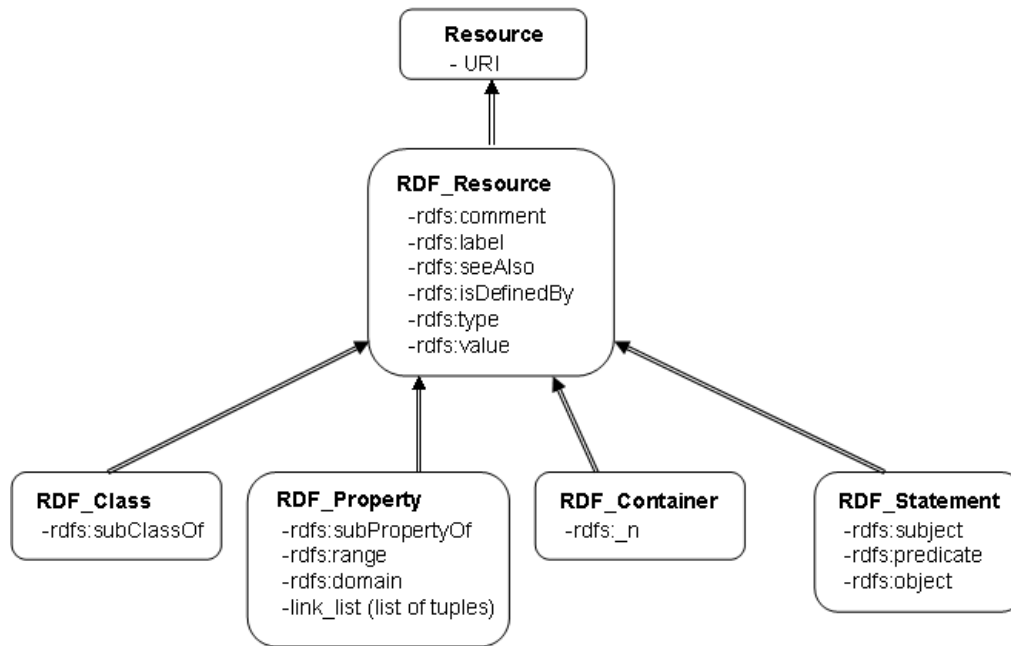
4.2 Συντακτικός Σημασιολογικός Αναλυτής RDF μεταδεδομένων (VRP)

Ο Συντακτικός Σημασιολογικός Αναλυτής RDF (Validating RDF Parser) [A99] είναι ένα εργαλείο κατάλληλο για την ανάλυση (parsing), τον έλεγχο της συνέπειας και την επεξεργασία των RDF σχημάτων και των περιγραφών πόρων. Ο Συντακτικός Αναλυτής (Parser) αναλύει συντακτικά τις προτάσεις ενός δοσμένου RDF/XML αρχείου σύμφωνα με τις Προδιαγραφές του RDF Model & Syntax. Ο Σημασιολογικός Αναλυτής ελέγχει αν οι προτάσεις που περιέχονται τόσο στο RDF Σχήμα όσο και στους περιγραφείς πόρων ικανοποιούν τους σημασιολογικούς περιορισμούς που απορρέουν από τις προδιαγραφές του RDF Schema Specification (RDFS). Σε αντίθεση με άλλους διαθέσιμους RDF συντακτικούς αναλυτές, ο VRP στηρίζεται στα εργαλεία CUP και Jflex, τα οποία είναι όμοια με τα εργαλεία YACC και LEX, και χρησιμοποιούνται για τη δημιουργία συντακτικών αναλυτών σε Java. Το Σημασιολογικό τμήμα του VRP στηρίζεται στην αναπαράσταση αντικειμένων, διαχωρίζοντας το RDF Σχήμα από τα στιγμιότυπα αυτού. Ο VRP έχει χρησιμοποιηθεί επιτυχώς σε ένα μεγάλο αριθμό RDF Σχημάτων υπαρχόντων εφαρμογών του Σημασιολογικού Ιστού (Semantic Web).

Ο VRP υποστηρίζει ενσωμάτωση του RDF σε XML ή HTML έγγραφα, τύπους Δεδομένων XML Σχημάτων (XML Schema Data Types) και το σύνολο των χαρακτήρων του Unicode. Επίσης, παρέχει διάφορες παραμέτρους για εκσφαλμάτωση (debugging) και σειριοποίηση (serialization) υπό τη μορφή τριπλέτων ή γράφων καθώς και πλήρη στατιστικά (χρόνου, χαρακτηριστικών) των σημασιολογικά ορθών σχημάτων και περιγραφών πόρων. Τέλος, είναι εύκολο να χρησιμοποιηθεί τόσο σαν αυτόνομη εφαρμογή (Γραφική Διεπιφάνεια Χρήσης - GUI) όσο και να ενσωματωθεί σε άλλες εφαρμογές.

4.2.1 Μοντέλο Αντικειμένων του VRP

Το μοντέλο αντικειμένων του VRP περιλαμβάνει την ιεραρχία κλάσεων που εμφανίζεται στο Σχήμα 4.5. Στην ρίζα της ιεραρχίας βρίσκεται η κλάση *Resource*. Η κλάση *RDF_Resource* αποτελεί μία άμεση υποκλάση αυτής. Οι κλάσεις *RDF_Class*, *RDF_Property*, *RDF_Statements* και *RDF_Container* είναι άμεσες υποκλάσεις της κλάσης *RDF_Resource*.



Σχήμα 4.5 Ιεραρχία Κλάσεων του Μοντέλου VRP

Παρακάτω, περιγράφονται οι κλάσεις του Μοντέλου VRP και δίνονται τα στιγμιότυπα αντικειμένων αυτών των κλάσεων:

Resource: Τα στιγμιότυπα αυτής της κλάσης αναπαριστούν πόρους στους οποίους δεν έχει αποδοθεί κάποια από τις προκαθορισμένες RDF/S ιδιότητες, όπως για παράδειγμα, οι `rdf:type`, `rdfs:comment`, `rdfs:label`, `rdfs:seeAlso`, `rdfs:isDefinedBy`. Σε αντίθεση τα στιγμιότυπα του **RDF_Resource** αναπαριστούν πόρους στους οποίους δεν έχει αποδοθεί καμία από τις προαναφερθείσες RDF/S ιδιότητες. Για παράδειγμα, το αντικείμενο το οποίο αναπαριστά τον πόρο με URI <http://www.artchive.com/rembrandt/abraham.jpg> ανήκει στην κλάση **RDF_Resource**.

RDF_Class: Τα στιγμιότυπα αυτής της κλάσης αναπαριστούν τις κλάσεις RDF. Για παράδειγμα, το αντικείμενο που αναπαριστά την κλάση *Painting* ανήκει σε αυτήν την κλάση.

RDF_Property: Τα στιγμιότυπα αυτής της κλάσης αναπαριστούν τις ιδιότητες του RDF (RDF properties). Για παράδειγμα, τα αντικείμενα που αναπαριστούν τις ιδιότητες *paints* και *maxCardinality* ανήκουν σε αυτήν την κλάση.

RDF_Container: Τα στιγμιότυπα αυτής της κλάσης αναπαριστούν τα πολυσύνολα του RDF (RDF containers).

RDF_Statement: Τα στιγμιότυπα αυτής της κλάσης αναπαριστούν τις υποστασιοποιημένες δηλώσεις.

RDF_Resource@4487	
URI	http://www.archive.com/rembrandt/abraham.jpg
rdf:type	ns1 #Painting ns2 #ExtResource

RDF_Property@4467	
URI	ns1 #paints
rdf:type	rdf #Property
rdfs:domain	ns1 #creates
rdfs:range	ns1 #Painting
link_list	www.culture.net/picasso132- http://www.museum.es/woman.qti www.culture.net/Rembrandt- http://archive.com/rembrandt/abra- ham.jpg

RDF_Class@4455	
URI	ns1 #Painting
rdf:type	rdfs #RealWorldObject
rdfs:subClassOf	ns1 #Artifact

Σχήμα 4.6. Το Μοντέλο Αντικειμένων του VRP

Το μοντέλο αντικειμένων αναπαριστά όλη την πληροφορία του RDF/XML αρχείου το οποίο δέχεται ως είσοδο ο VRP. Πιο συγκεκριμένα, ένα αντικείμενο, το οποίο ανήκει σε μία από τις παραπάνω κλάσεις της Java, δημιουργείται για κάθε πόρο που υπάρχει σε μία αναλυτική περιγραφή. Η υπόλοιπη πληροφορία αποθηκεύεται ως τιμές γνωρισμάτων του δημιουργούμενου αντικειμένου. Στο μοντέλο του VRP η διάκριση μεταξύ του σχήματος και των δεδομένων είναι ρητή.

4.2.3 Σημασιολογικός Έλεγχος της Βάσης Δεδομένων

Ο Σημασιολογικός Αναλυτής (Validator) ελέγχει την συνέπεια των αναλυόμενων περιγραφών. Η κλάση η οποία υλοποιεί την λειτουργικότητα της σημασιολογίας είναι ο DBValidator. Ο Σημασιολογικός Αναλυτής παίρνει ως είσοδο το μοντέλο αντικειμένων που παράγεται από τον VRP. Η διαδικασία της σημασιολογικής ανάλυσης βασίζεται τόσο στις περιγραφές του αρχείου εισόδου (π.χ. το μοντέλο αντικειμένων του VRP) όσο και στις πληροφορίες οι οποίες αποθηκεύονται στην βάση δεδομένων. Σε αντίθεση με άλλους συντακτικούς αναλυτές ο Σημασιολογικός Αναλυτής της Βάσης δεν συνδέεται με απομακρυσμένους ή τοπικούς χώρους ονοματοδοσίας (namespaces) για να φέρει επιπλέον πληροφορία για τους εξωτερικούς πόρους – οι οποίοι συνήθως είναι τα σχήματα – που βρίσκονται στο αρχείο εισόδου. Ο Σημασιολογικός Αναλυτής της Βάσης φέρνει τέτοιου είδους πληροφορίες μόνο από την βάση δεδομένων. Η διαδικασία του σημασιολογικού ελέγχου αποτελείται από δύο φάσεις: τον σημασιολογικό έλεγχο του σχήματος και των δεδομένων.

4.2.3.1 Σημασιολογικός Έλεγχος Σχήματος

Όσον αφορά στον σημασιολογικό έλεγχο του σχήματος, κατά την φόρτωση ελέγχεται αν οι περιορισμοί του Αλφαβήτου (Alphabet), του Μετα-σχήματος (Meta-Schema) και του Σχήματος (Schema) ικανοποιούνται. Για να ελεγχθούν οι παραπάνω περιορισμοί, χρησιμοποιούνται μερικές από τις μεθόδους σημασιολογικού ελέγχου

του VPR. Πιο συγκεκριμένα, χρησιμοποιούνται μέθοδοι που ελέγχουν αν υπάρχουν κύκλοι στην ιεραρχία των υποκλάσεων και των υποιδιοτήτων. Επιπλέον, χρησιμοποιείται η μέθοδος που υλοποιεί την κληρονομικότητα του πεδίου ορισμού και του συνόλου τιμών (domain/range). Η μέθοδος αυτή κληρονομεί σε μία ιδιότητα το πεδίο ορισμού/σύνολο τιμών της υπερ-ιδιότητας αυτής σε περίπτωση που δεν ορίζεται πεδίο ορισμού/σύνολο τιμών για την ιδιότητα αυτή. Επιπλέον, κατά τη φόρτωση του σχήματος ορίζονται μέθοδοι για τον έλεγχο των υπολοίπων περιορισμών. Η μέθοδος *type_check* ελέγχει αν κάθε πόρος του σχήματος έχει τον σωστό τύπο. Για παράδειγμα, ελέγχει αν οι πόροι του σχήματος που συσχετίζονται με την σχέση υποκλάσης (subClassOf) είναι και οι δύο κλάσεις ή μετακλάσεις (classes/metaclasses), ή αν ο πόρος που χρησιμοποιείται σαν πεδίο ορισμού μίας ιδιότητας είναι κλάση ή μετακλάση. Η μέθοδος *subpropertyDomainRange_check* ελέγχει αν το πεδίο ορισμού/σύνολο τιμών μιας ιδιότητας είναι υποσύνολο του πεδίου ορισμού/συνόλου τιμών της υπερ-ιδιότητας αυτής. Τέλος, οι μέθοδοι *uniqueDefinedDomain* και *uniqueDefinedRange* ελέγχουν αν είναι μοναδικά ορισμένα το πεδίο ορισμού και το σύνολο τιμών μίας ιδιότητας.

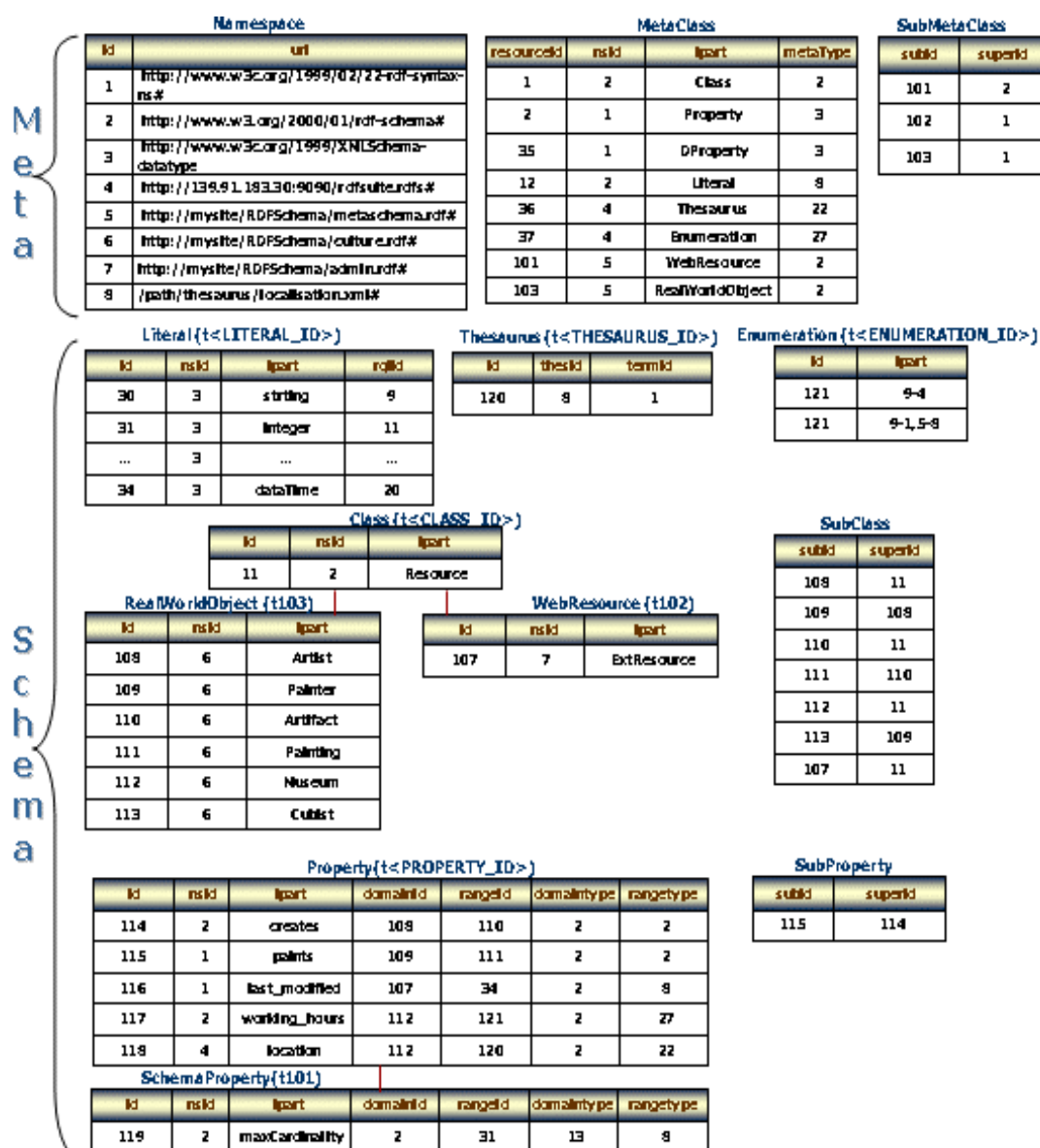
4.3 Υπάρχον Μοντέλο Σχεσιακής Αναπαράστασης RDF Μεταδεδομένων

Σε αυτή την ενότητα θα περιγράψουμε την υπάρχουσα σχεσιακή αναπαράσταση του RDF Σχήματος καθώς και των περιγραφών πόρων. Στην υπάρχουσα αναπαράσταση οι RDF κλάσεις και ιδιότητες αποθηκεύονται σαν σχήμα της βάσης. Επομένως, το σχήμα της βάσης μεταβάλλεται καθώς αποθηκεύονται νέα σχήματα στη βάση. Οι RDF περιγραφές για τα δεδομένα εμπλουτίζουν το παραπάνω σχήμα.

Το κύριο μοντέλο του RDF/S αναπαριστάνεται από έξι πίνακες (Σχήμα 4.7), τους *MetaClass*, *Class*, *Property*, *SubMetaClass*, *SubClass*, *SubProperty* οι οποίοι περιέχουν τις μετα-κλάσεις, τις κλάσεις και τις ιεραρχίες των ιδιοτήτων που ορίζονται στα RDF σχήματα. Επίσης, υπάρχει ένας πίνακας που ονομάζεται *NameSpace* στον οποίο αποθηκεύονται οι χώροι ονοματοδοσίας (namespaces) των RDF σχημάτων που αποθηκεύονται στην σχεσιακή βάση. Επιπλέον, ο πίνακας *Literal* χρησιμοποιείται για την αποθήκευση των Λεκτικών (*Literal*), όπως *string*, *integer*, *dateTime*.

Το κύριο χαρακτηριστικό της αναπαράστασης είναι ο διαχωρισμός του RDF Σχήματος από τις πληροφορίες των δεδομένων, επίσης, η διάκριση μεταξύ μονότιμων (unary) και δίτιμων σχέσεων στις οποίες αποθηκεύονται τα στιγμιότυπα των κλάσεων και των ιδιοτήτων αντίστοιχα. Πιο συγκεκριμένα, οι πίνακες των κλάσεων αποθηκεύουν τα URIs των πόρων ενώ οι πίνακες των ιδιοτήτων αποθηκεύουν τους κόμβους αφετηρίας και προορισμού της ιδιότητας. Τέλος, οι ταξινομητικές σχέσεις μεταξύ των ετικετών του σχήματος (schema labels) αποθηκεύονται στους πίνακες *SubMetaClass*, *SubClass* και *SubProperty*, ενώ τα αντίστοιχα στιγμιότυπα των πινάκων συνδέονται μέσω της σχέσης *subtable*.

Τέλος, πρέπει να σημειωθεί ότι ανατίθεται ένα ακέραιο αναγνωριστικό σε κάθε μετακλάση, κλάση και ιδιότητα. Τα ακέραια αναγνωριστικά που αποδίδονται είναι αυξητικά.



Σχήμα 4.7 Σχεσιακή Αναπαράσταση RDF Σχημάτων

4.3.1 Περιγραφή Πινάκων

Στην ενότητα αυτή θα περιγραφούν λεπτομερώς οι κύριοι πίνακες για την αναπαράσταση του σχήματος καθώς και η δομή των πινάκων που δημιουργούνται για την αποθήκευση των στιγμιότυπων των μετακλάσεων, των κλάσεων και των ιδιοτήτων. Οι πίνακες διακρίνονται σε πίνακες μέτα, σχήματος (schema) και δεδομένων ανάλογα με τον τύπο της πληροφορίας που αποθηκεύουν (π.χ., πληροφορίες μετασχημάτων, σχημάτων ή δεδομένων). Τα χαρακτηριστικά για τα οποία δημιουργείται πίνακας περιεχομένων είναι με **έντονη γραμματοσειρά**.

Ο πίνακας *Namespace(id, uri)* περιέχει πληροφορία για τους χώρους ονοματοδοσίας των αποθηκευμένων σχημάτων καθώς και τα ονόματα των αρχείων των θησαυρών που είναι αποθηκευμένοι στην σχεσιακή βάση. Αποτελείται από δύο πεδία. Το πεδίο *id* έχει σαν ένα εσωτερικό ακέραιο αναγνωριστικό που του αποδίδεται, ενώ το πεδίο *uri* έχει σαν τιμή το URI του χώρου ονοματοδοσίας (namespace) ή το όνομα του αρχείου ενός θησαυρού.

1. Πίνακες Μέτα

Ο πίνακας *MetaClass(resourceid, nsid, lpart, metatype)* περιέχει τις RDF/S μετακλάσεις όπως, π.χ., *rdfs:Class*, *rdf:Property*, *rdfs:Literal* καθώς και τις μετακλάσεις του Thesaurus και του Enumeration και όλες τις μετακλάσεις που ορίζονται στα αποθηκευμένα σχήματα. Πιο συγκεκριμένα ο πίνακας αυτός αποτελείται από τέσσερα πεδία. Το πεδίο *resourceid*, το οποίο έχει σαν τιμή το ακέραιο αναγνωριστικό (κωδικός) που αποδίδεται στην μετακλάση. Το πεδίο *nsid* το οποίο έχει σαν τιμή το ακέραιο αναγνωριστικό του χώρου ονοματοδοσίας στο οποίο ορίζεται η μετακλάση, το πεδίο *lpart* το οποίο έχει σαν τιμή το τοπικό όνομα της μετακλάσης (π.χ. *RealWorldObject*) και τέλος το πεδίο *metatype* το οποίο είναι ένα ακέραιο αναγνωριστικό που δηλώνει τον τύπο της μετακλάσης, π.χ., αν πρόκειται για μια μετακλάση κλάσεων ή για μια μετακλάση ιδιοτήτων.

Ο πίνακας *SubMetaClass(subid, superid)* περιέχει τις σχέσεις υποκλάσης (*subclassOf*) μεταξύ των μετακλάσεων. Ο πίνακας αυτός αποτελείται από δύο πεδία τα *subid* και τα *superid* κανένα από τα οποία δεν μπορεί να έχει κενή τιμή (*null*). Το πεδίο *subid* έχει σαν τιμή το ακέραιο αναγνωριστικό της υπο-μετακλάσης ενώ το πεδίο *superid* παίρνει σαν τιμή το ακέραιο αναγνωριστικό της υπερ-μετακλάσης.

2. Πίνακες Σχημάτων

Ο πίνακας *Class(id, nsid, lpart)* περιέχει το URI και τον κωδικό των κλάσεων που έχουν καταχωρηθεί στην βάση. Ο πίνακας αυτός περιέχει τρία πεδία. Το πεδίο *id* έχει σαν τιμή το εσωτερικό ακέραιο αναγνωριστικό που αποδίδεται στην κλάση. Το πεδίο *nsid* που έχει σαν τιμή το αναγνωριστικό του χώρου ονοματοδοσίας στον οποίο ορίζεται η κλάση και τέλος το πεδίο *lpart* που αποτελεί το τοπικό όνομα της κλάσης (π.χ. *Artist*).

Ο πίνακας *Property(id, nsid, lname, domainid, rangeid, domaintype, rangetype)* που περιέχει όλες τις ιδιότητες που έχουν καταχωρηθεί στην βάση. Αποτελείται από τα πεδία *id*, *nsid*, *lname*, *domainid* και *rangeid*, *domaintype* και *rangetype*. Τα τρία πρώτα πεδία είναι αντίστοιχα του πίνακα *Class*. Το πεδίο *domainid* έχει σαν τιμή τον κωδικό της κλάσης που αποτελεί το πεδίο ορισμού της ιδιότητας ενώ το πεδίο *rangeid* έχει σαν τιμή τον κωδικό της κλάσης αποτελεί το πεδίο τιμών της ιδιότητας. Τέλος, τα πεδία *domaintype* και *rangetype* δηλώνουν αντίστοιχα τον τύπο του πεδίου ορισμού, π.χ αν το πεδίο ορισμού (*domain*) είναι μία κλάση ή μία μετακλάση και τον τύπο του συνόλου τιμών, π.χ. αν το σύνολο τιμών είναι μία κλάση, μία μετακλάση, ένα λεκτικός τύπος, ένας θησαυρός ή ένας απαριθμητικός τύπος.

Ο πίνακας **Data_Property** που περιέχει όλες τις ιδιότητες ιδιότητες των οποίων ούτε το πεδίο ορισμού ούτε το σύνολο τιμών είναι μετακλάση. Η πληροφορία που αποθηκεύεται σε αυτόν τον πίνακα είναι πλεονάζουσα καθώς η ίδια πληροφορία περιέχεται ήδη στον πίνακα *Property*, παρόλα αυτά η πληροφορία αυτή είναι χρήσιμη για την επερώτηση των RDF περιγραφών. Αποτελείται από τα πεδία *id*, *nsid*, *lname*, *domainid* και *rangeid*, *domaintype* και *rangetype* αντίστοιχα με αυτά του πίνακα *Property* καθώς η δομή αυτού του πίνακα είναι ακριβώς ίδια με τη δομή του πίνακα *Property*.

Οι πίνακες **SubClass(subid, superid)** και **SubProperty(subid, superid)** οι οποίοι αναπαριστούν τις ιεραρχίες των κλάσεων και των ιδιοτήτων αντίστοιχα. Ο πίνακας *SubClass* περιέχει τα πεδία *subid* και *superid*. Για κάθε σχέση υποκλάσης που δηλώνεται μεταξύ δύο κλάσεων στο πεδίο *subid* καταχωρείται ο κωδικός της υποκλάσης και στο πεδίο *superid* ο κωδικός της υπερκλάσης. Αντίστοιχα, ισχύουν για τον πίνακα *SubProperty*.

Ο πίνακας **Literal(id, nsid, lpart, rqlid)** ο οποίος περιέχει τους λεκτικούς τύπους του RDF, όπως π.χ. string, float, dateTime. Ο πίνακας αυτός αποτελείται από τέσσερα πεδία. Το πεδίο *id* το οποίο είναι το εσωτερικό ακεραίο αναγνωριστικό που ανατίθεται στο λεκτικό τύπο. Το πεδίο *nsid* που έχει σαν τιμή το αναγνωριστικό του χώρου ονοματοδοσίας στον οποίο ορίζεται το literal και το πεδίο *lpart* που αποτελεί το τοπικό όνομα του literal και τέλος το πεδίο *rqlid* που αποτελεί ακεραίο αναγνωριστικό που ανατίθεται στον λεκτικό τύπο από την RQL.

Ο πίνακας **Thesaurus** ο οποίος περιέχει τους τύπους των θησαυρών. Ένας τύπος θησαυρού αποτελείται από ένα όνομα (αρχείου) θησαυρού που αποθηκεύεται στον πίνακα NameSpace και ένα όρο θησαυρού. Οι ιδιότητες που έχουν σαν πεδίο ορισμού ένα τύπο θησαυρού μπορούν να έχουν ως τιμές τους όρους θησαυρού που είναι πιο συγκεκριμένες από τον τύπο του θησαυρού.Ο πίνακας αυτός αποτελείται από τρία πεδία. Το πεδίο *id* το οποίο είναι το εσωτερικό ακεραίο αναγνωριστικό που ανατίθεται στο τύπο του θησαυρού. Το πεδίο *thesid* που έχει σαν τιμή το ακεραίο αναγνωριστικό που ανατίθεται στον θησαυρό του και το πεδίο *termid* το οποίο είναι το ακεραίο αναγνωριστικό που ανατίθεται στον όρο του θησαυρού.

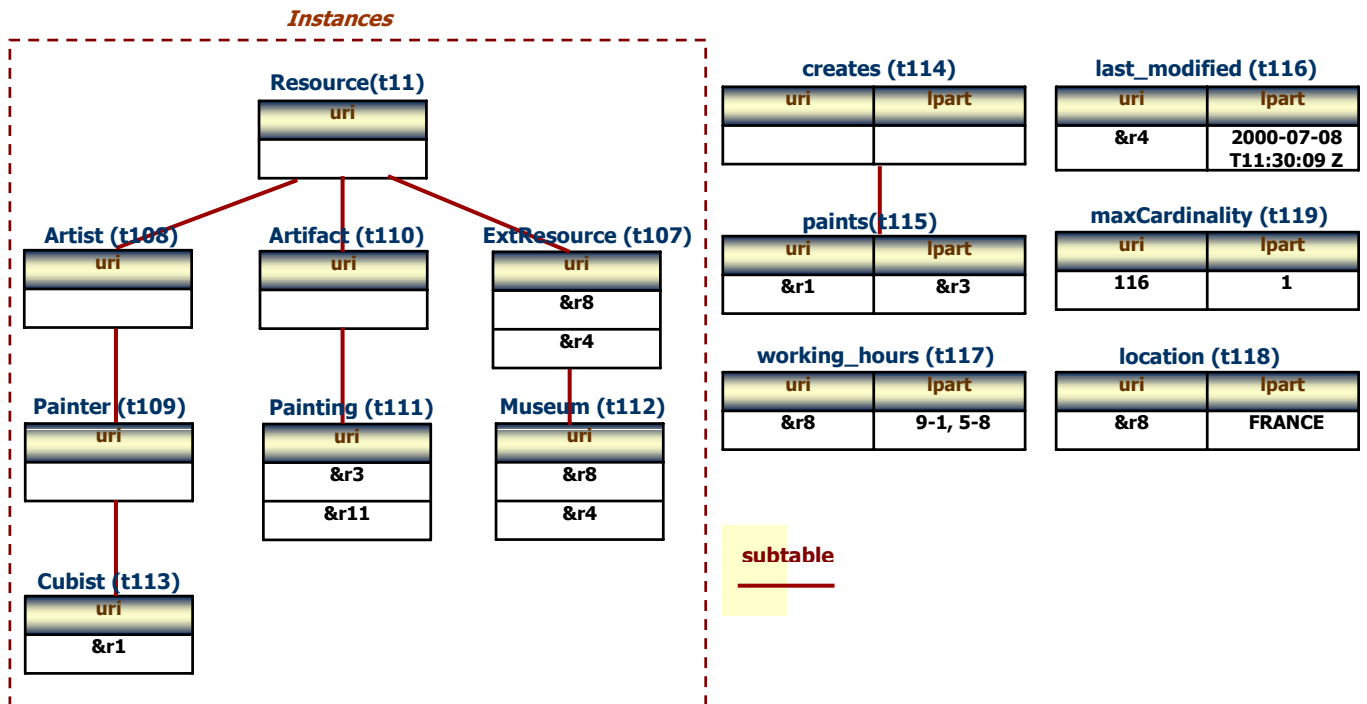
Ο πίνακας **Enumeration(id, lpart)** ο οποίος περιέχει τις τιμές των τύπων απαρίθμησης. Ο πίνακας αυτός αποτελείται από τα πεδία *id* και *lpart*. Το πεδίο *id* έχει σαν τιμή ένα εσωτερικό ακεραίο αναγνωριστικό που ανατίθεται στον τύπο απαρίθμησης και το πεδίο *enumname* το οποίο είναι μια τιμή του τύπου απαρίθμησης.

Ο πίνακας **Generator** αποτελείται μόνο από ένα πεδίο το *id* το οποίο είναι το μέγιστο εσωτερικό ακεραίο αναγνωριστικό που έχει ανατεθεί σε μία μετακλάση, κλάση, ιδιότητα, τύπο θησαυρού, τύπο απαρίθμησης και λεκτικό τύπο.

Η βάση δεδομένων περιέχει επίσης τους παρακάτω πίνακες:

Πίνακες Μετακλάσεων (MetaClass tables): Για κάθε μετακλάση που αποθηκεύεται στην βάση δεδομένων δημιουργείται ένας πίνακας **t<metaclass_id>** ο οποίος περιέχει τις κλάσεις ή τις ιδιότητες που ταξινομούνται κάτω από την μετακλάση. Το

`metaclass_id` είναι το αναγνωριστικό που ανατίθεται στην συγκεκριμένη μετακλάση. Αυτός ο πίνακας δηλώνεται ως υποπίνακας των πινάκων που αναπαριστούν τις υπερκλάσεις. Ανάλογα με το αν η μετακλάση είναι υποκλάση του `rdfs:Class` ή του `rdf:Property` η δομή του πίνακα είναι όμοια με τη δομή του πίνακα `t<CLASS_ID>` ή του πίνακα `t<PROPERTY_ID>` αντίστοιχα.



Σχήμα 4.8 Πίνακες δεδομένων

Πίνακες Δεδομένων

- Πίνακες Κλάσεων (Class tables):** Για κάθε κλάση που αποθηκεύεται στην σχεσιακή βάση, δημιουργείται ένας πίνακας με όνομα `t<class_id>` ο οποίος περιέχει τα URIs τα οποία είναι κατηγοριοποιημένα σε αυτήν την κλάση. Ο πίνακας δηλώνεται ως υποπίνακας των πινάκων που αναπαριστούν τις υπερκλάσεις αυτού. Επομένως, οι πίνακες των κλάσεων κατασκευάζουν μια ιεραρχία η ρίζα των οποίων είναι ο πίνακας `rdfs:Resource`.
- Πίνακες Ιδιοτήτων (Property tables):** Για κάθε ιδιότητα που αποθηκεύεται στην σχεσιακή βάση δημιουργείται ένας πίνακας `t<property_id>(att0 VARCHAR, att1)` ο οποίος αποθηκεύει τις ιδιότητες. Το `property_id` είναι ένα αναγνωριστικό που ανατίθεται σε μία συγκεκριμένη ιδιότητα. Ο πίνακας δηλώνεται ως ένας υποπίνακας των πινάκων που αναπαριστούν τις υπερ-ιδιότητες αυτού. Ο πίνακας αποτελείται από τα πεδία `uri` και `lpart`. Το πεδίο `uri` αναπαριστά το URI του πόρου στον οποίο εφαρμόζεται η ιδιότητα (property) Ενώ το πεδίο `lpart` αναπαριστά την τιμή της ιδιότητας. Ο τύπος του πεδίου αυτού ποικίλει ανάλογα με το πεδίο τιμών της ιδιότητας αυτής:

- Αν το πεδίο τιμών είναι μία κλάση, μία συμβολοσειρά XML datatype, ένα enumeration ή ένας θησαυρός (thesaurus) τότε το *lpart* είναι VARCHAR
 - Αν το πεδίο τιμών είναι ένας ακέραιος XML datatype τότε το *lpart* είναι ένας ακέραιος
 - Αν το πεδίο τιμών είναι ένα float XML datatype τότε το *lpart* είναι ένας float
 - Αν το πεδίο τιμών είναι μία δυαδικός (Boolean) XML datatype τότε το *lpart* είναι ένας boolean
 - Αν το πεδίο τιμών είναι dateTime XML datatype τότε το *lpart* είναι ένα datetime
 - Αν το πεδίο τιμών είναι μία μετακλάση (metaclass) τότε το *lpart* είναι ένας ακέραιος. Το πεδίο *lpart* αποθηκεύει το ακέραιο αναγνωριστικό της κλάσης ή της ιδιότητας η οποία είναι η τιμή αυτής της ιδιότητας
- **Πίνακες Θησαυρών:** Όταν αποθηκεύεται ένας θησαυρός στην βάση δεδομένων, δημιουργούνται οι παρακάτω πίνακες:
- Ο πίνακας **thes<thesaurus_id>** ο οποίος περιέχει τους όρους του θησαυρού και αποτελείται από τρία πεδία, τα Ένα ακέραιο αναγνωριστικό ανατίθεται σε κάθε όρο του θησαυρού.
att0: Περιέχει το ακέραιο αναγνωριστικό που ανατίθεται στον όρο του θησαυρού
att1: Περιέχει τον όρο
att2: Περιέχει την τιμή του γνωρίσματος DescID που μπορεί να αποδοθεί στον θησαυρό στο XML αρχείο αυτού
 - **narrow<thesaurus_id>(att0 INTEGER, att1 INTEGER):** Αποθηκεύει τις σχέσεις του τύπου *NarrowerTerms* ή *PartOfTerms* που υπάρχουν μεταξύ των όρων του θησαυρού.
att0: Περιέχει το αναγνωριστικό του στενότερου τύπου
att1: Περιέχει το αναγνωριστικό του ευρύτερου τύπου

Πρέπει να σημειωθεί ότι, το όνομα ενός πίνακα προκύπτει από το γράμμα <t> (το πρώτο γράμμα της λέξης “table”) και το ακέραιο αναγνωριστικό της μετακλάσης, της κλάσης ή της ιδιότητας που αναπαριστά αυτός ο πίνακας. Για παράδειγμα το όνομα του πίνακα που αντιστοιχεί στην κλάση με αναγνωριστικό 15 είναι *t15*. Γενικά, τα ονόματα των πινάκων έχουν τη μορφή **t(able)+ αναγνωριστικό** μετακλάσης/κλάσης/ιδιότητας.

4.4.2 Τροποποιήσεις στην Σχεσιακή Βάση Αποθήκευσης RDF Μεταδεδομένων

Παραπάνω είδαμε ότι κατά την φάση της Συντακτικής και Σημασιολογική Ανάλυσης δημιουργούνται τα αντικείμενα του VRP τα οποία αναπαριστούν τις πληροφορίες του σχήματος και των δεδομένων. Ωστόσο, αν και κατά την φάση της Συντακτικής Ανάλυσης δημιουργούνται τα VRP αντικείμενα για τις υποστασιοποιημένες προτάσεις, η υπάρχουσα υλοποίηση της Σχεσιακής Βάσης Μεταδεδομένων δεν

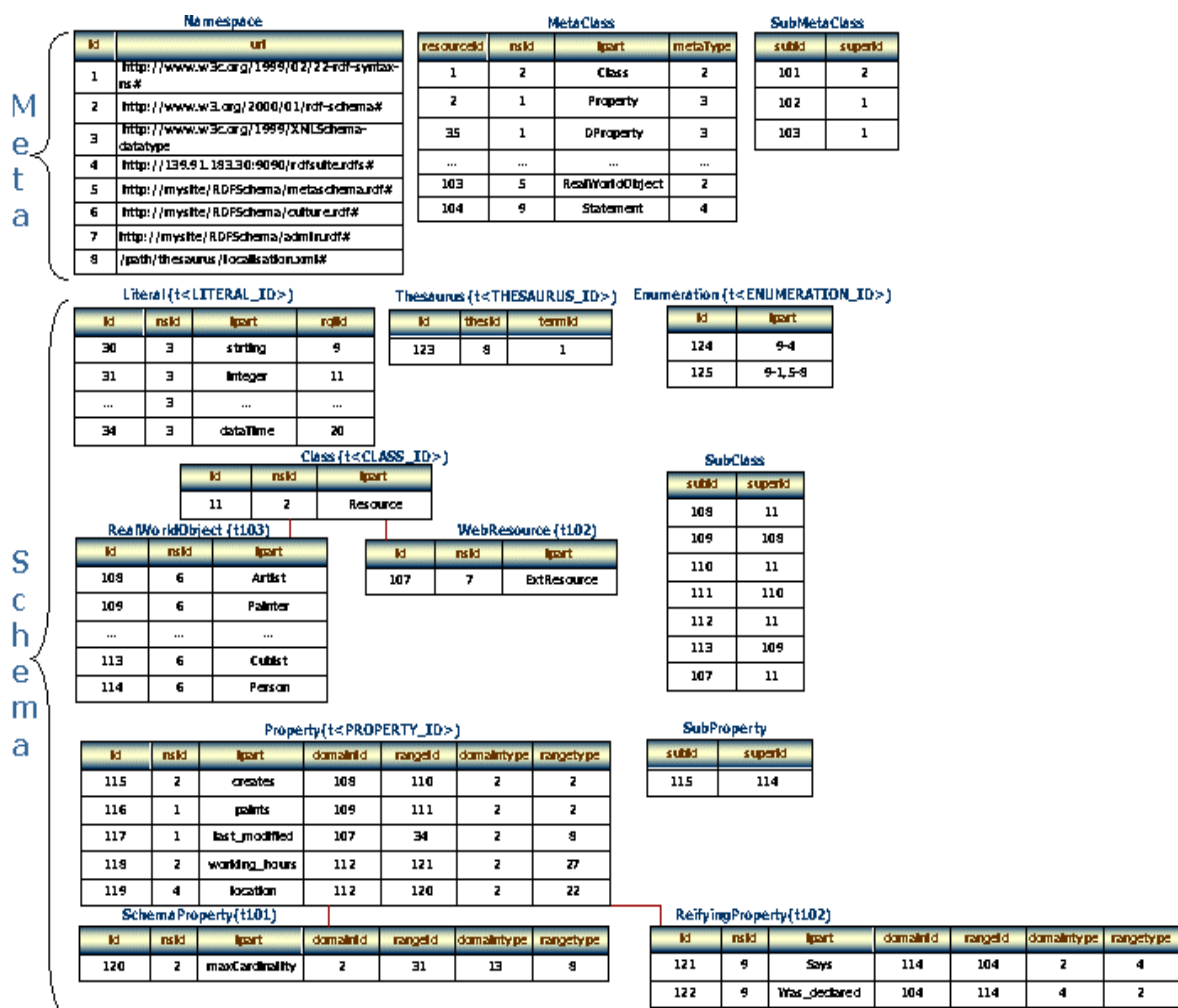
υποστηρίζει την αποθήκευσή τους. Στην παράγραφο αυτή θα εξετάσουμε εκείνες τις αλλαγές που πρέπει να γίνουν σε αυτήν ώστε να υποστηριχθεί και η αποθήκευση των υποστασιοποιημένων προτάσεων.

1. Πίνακες Μέτα

Γενικά οι πίνακες Μέτα δεν υφίστανται καμμία αλλαγή, ωστόσο πρέπει να προστεθεί στον πίνακα *MetaClass* (Σχ. 4.14), μια πλειάδα με τοπικό όνομα (*lpart*) *Statement*, και ένας μετα-τύπος για αυτήν. Η προσθήκη αυτή γίνεται εξαιτίας της θεώρησης μας ότι η *Statement*, είναι μια μετακλάση και είναι απαραίτητη επειδή σε επίπεδο σχήματος υπάρχουν ιδιότητες οι οποίες έχουν πεδίο ορισμού ή σύνολο τιμών του τύπου *Statement*. Ο πίνακας *Property*, που περιέχει τις ιδιότητες, αποτελείται από τα πεδία (*id*, *domain*, *range*, *d_type*, *r_type*). Τα πεδία *d_type* και *r_type* έχουν ως τιμή τον τύπο του πεδίου ορισμού και τον τύπο του συνόλου τιμών της ιδιότητας αντίστοιχα. Σε περίπτωση που μια ιδιότητα έχει για παράδειγμα σύνολο τιμών το οποίο είναι μια υποστασιοποιημένη πρόταση, πρέπει να μπει ως τιμή της *r_type* το *Statement*.

2. Πίνακες Σχήματος

Όπως αναφέρθηκε και στο Κεφάλαιο 3, θεωρούμε μια υποκατηγορία των ιδιοτήτων (*properties*), στην οποία υπάγονται όλες εκείνες οι ιδιότητες με πεδίο ορισμού ή σύνολο τιμών του τύπου *Statement*. Οι ιδιότητες αυτές ονομάζονται *Reifying Properties* και αποτελούν μια υποκλάση της κλάσης *Property*. Για τον λόγο αυτό, σε επίπεδο σχήματος, δημιουργείται ένας νέος πίνακας με όνομα *ReifyingProperty*, ο οποίος περιέχει όλες τις ιδιότητες της κατηγορίας αυτής (Σχήμα 4.9)



Σχήμα 4.9 Σχεσιακή Αναπαράσταση με Υποστασιοποίηση

3. Πίνακες Δεδομένων

Όσον αφορά στο επίπεδο δεδομένων πρέπει να προστεθεί ένας νέος πίνακας, ο *Statement*. Στον πίνακα αυτόν θα περιέχονται όλες οι υποστασιοποιημένες προτάσεις. Τα πεδία τα οποία θα συνιστούν τον πίνακα αυτόν είναι τα εξής:

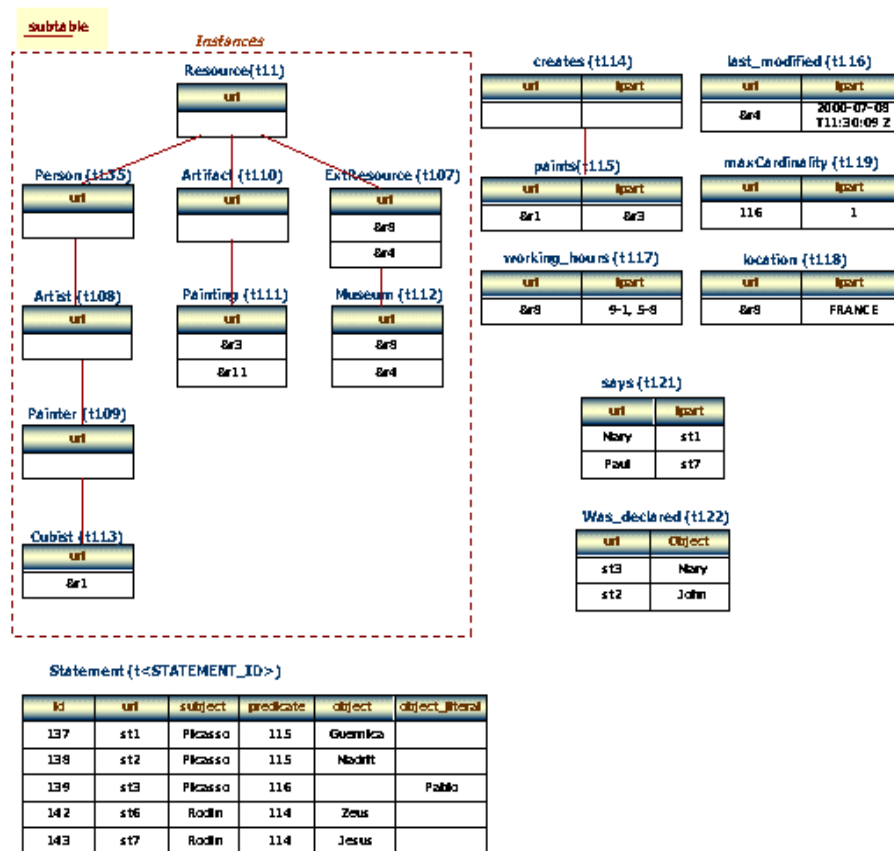
- *id*, το πεδίο αυτό αποτελεί ένα εσωτερικό αθέρατο αναγνωριστικό της υποστασιοποιημένης δήλωσης
- *uri*, το πεδίο αυτό αποτελεί το αναγνωριστικό της υποστασιοποιημένης πρότασης, το οποίο είναι ένα αλφαριθμητικό, όπως π.χ. *st1_statement*. Το αναγνωριστικό αυτό αποτελεί το αναγνωριστικό το οποίο αποδίδεται στον

πόρο της υποστασιοποιημένης πρότασης έτσι ώστε να μπορεί να γίνει αναφορά αργότερα σε αυτήν και να χρησιμοποιηθεί ως υποκείμενο ή ως αντικείμενο άλλων προτάσεων.

- *sub*, το πεδίο αυτό αποτελεί το υποκείμενο της υποστασιοποιημένης δήλωσης και αντιστοιχεί στην ιδιότητα *rdf:subject*. Η τιμή του πεδίου *sub* είναι το *urί* του πόρου που αποτελεί το υποκείμενο της υποστασιοποιημένης δήλωσης. Το υποκείμενο της υποστασιοποιημένης δήλωσης μπορεί να είναι ένας χώρος ονοματοδοσίας, μία μετακλάση, μία κλάση, μία ιδιότητα ή το στιγμιότυπο μιας κλάσης. Τέλος, το πεδίο αυτό μπορεί να είναι και μία υποστασιοποιημένη δήλωση. Το πεδίο αυτό δεν μπορεί να είναι κενό (null).
- *pred*, το πεδίο αυτό αποτελεί την ιδιότητα της υποστασιοποιημένης πρότασης και αντιστοιχεί στην ιδιότητα *rdf:predicate*. Η τιμή του πεδίου *pred* είναι ένας δείκτης στον κωδικό της ιδιότητας. Το πεδίο αυτό δεν μπορεί να είναι κενό (null).
- *object*, το πεδίο αυτό αποτελεί το αντικείμενο της υποστασιοποιημένης δήλωσης και αντιστοιχεί στην ιδιότητα *rdf:object*. Αντίστοιχα με το πεδίο *sub* αντιστοιχεί στο *urί* του αντικειμένου το οποίο μπορεί να είναι: ένας χώρος ονοματοδοσίας, μία μετακλάση, μία κλάση, μία ιδιότητα ή το στιγμιότυπο μιας κλάσης. Τέλος, το πεδίο αυτό μπορεί να είναι και μία υποστασιοποιημένη δήλωση. Το πεδίο αυτό είναι κενό μόνο στην περίπτωση που το αντικείμενο της πρότασης είναι ένα literal.
- *object_literal*, το πεδίο αυτό αποτελεί το αντικείμενο της υποστασιοποιημένης δήλωσης, στην περίπτωση που αυτό είναι literal. Στην αντίθετη περίπτωση είναι κενό (null).

Η παραπάνω υλοποίηση μας εξασφαλίζει ότι κάθε πρόταση έχει ένα μοναδικό αναγνωριστικό. Επομένως, μπορεί να χρησιμοποιηθεί σαν υποκείμενο ή σαν αντικείμενο μίας άλλης πρότασης, γεγονός που μας επιτρέπει να έχουμε φωλιασμένες υποστασιοποιημένες προτάσεις, π.χ. ο Γιώργος είπε ότι η Μαρία είπε ότι ο Γιάννης είπε ότι ο Πικάσο ζωγράφισε την Guernica. Αυτή η δυνατότητα είναι χρήσιμη στην περίπτωση ανταλλαγής πληροφοριών μεταξύ Agents.

Η αναπαράσταση των υποστασιοποιημένων δηλώσεων στους πίνακες των δεδομένων φαίνεται στο σχήμα 4.10.



Σχήμα 4.10 Αναπαράσταση Υποστασιοποιημένων Δεδομένων

4.4.2.1 Υποστασιοποίηση RDF Σχήματος

Χωρίς να υποστεί τροποποιήσεις το σχήμα της σχεσιακής βάσης που αναφέρθηκε παραπάνω μπορούμε να αποθηκεύσουμε πληροφορίες για τις Εκδόσεις των RDF Σχημάτων. Πράγματι, για την αποθήκευση των πληροφοριών του σχήματος χρησιμοποιούμε τον πίνακα Statement, στον οποίο πλην των υποστασιοποιημένων προτάσεων που αναπαριστούν προτάσεις της μορφής «Ο Picasso ζωγράφισε την Guernica», αποθηκεύουμε και όλα τα στοιχεία του σχήματος σε μορφή υποστασιοποιημένων προτάσεων.

Με τον όρο στοιχεία του σχήματος εννοούμε: *τους χώρους ονοματοδοσίας (namespaces), τις μετακλάσεις (metaclass), τις σχέσεις υπομετακλάσεων (submetaclass), τις κλάσεις (class), τις ιδιότητες (properties), τις σχέσεις υποκλάσεις subclass), τις σχέσεις υπο-ιδιοτήτων (subproperties) καθώς και τα literals.*

Κατά την φόρτωση του σχήματος, δεν δημιουργούνται μόνο οι πίνακες MetaClass, Class, Property κτλ., αλλά παράλληλα αποθηκεύονται όλες οι πληροφορίες αυτές και στον πίνακα Statement. Για παράδειγμα, η πληροφορία του σχήματος, «Το Painter είναι μια κλάση», αποθηκεύεται στον πίνακα Statement, με `rdf:subject` το `Painter`,

rd:predicate το rdf:type και rdf:object το Class, και παράλληλα αποδίδεται στην πρόταση αυτή ένα μοναδικό uri, το οποίο μπορεί να χρησιμοποιηθεί σαν υποκείμενο ή σαν αντικείμενο μιας ιδιότητας, π.χ. της has_version. Με αυτόν τον τρόπο δίνεται η δυνατότητα, να αποθηκευτούν πληροφορίες για όλα τα στοιχεία του σχήματος.

Ίσως κάποιος θα μπορούσε να ισχυριστεί κάποιος ότι το κόστος της παραπάνω μεθόδου είναι πολύ μεγάλο καθώς η βάση θα είχε πολύ μεγαλύτερες απαιτήσεις σε αποθηκευτικό χώρο. Επίσης, η καθυστέρηση κατά την «φόρτωση» του RDF σχήματος θα διπλασιαζόταν καθώς θα έπρεπε η ίδια πληροφορία να αποθηκευτεί τόσο στους πίνακες του σχήματος όσο και στον πίνακα Statement.

Ωστόσο, τα μειονεκτήματα αυτά δείχνουν λιγότερο σημαντικά αν αναλογιστούμε τα οφέλη που μας προσφέρει η αναπαράσταση αυτή. Πρώτα από όλα χαρακτηρίζεται από ομοιομορφία, όπως θα δούμε και στο επόμενο Κεφάλαιο 5, η επερώτηση, τόσο για τις υποστασιοποιημένες προτάσεις δεδομένων όσο και για τα υποστασιοποιημένα στοιχεία του σχήματος γίνεται με τον ίδιο τρόπο. Η παραπάνω δυνατότητα κάνει την εφαρμογή πιο προσιτή για τον χρήστη, καθώς δεν χρειάζεται να μάθει κάτι καινούργιο. Ένα άλλο σημαντικό πλεονέκτημα, της παραπάνω σχεδίασης είναι ότι υπάρχει η δυνατότητα να αποθηκευτεί οποιουδήποτε τύπου πληροφορία απαιτεί ο διαχειριστής του συστήματος σχετικά με τα στοιχεία του RDF σχήματος.

ΚΕΦΑΛΑΙΟ 5

Υποστήριξη της Υποστασιοποίησης στην Γλώσσα Επερωτήσεων RQL

Σκοπός της ενότητας αυτής είναι η παρουσίαση των απαιτούμενων τροποποιήσεων στην υπάρχουσα γλώσσα RQL για την υποστήριξη επερωτήσεων υποστασιοποιημένων δηλώσεων. Προκειμένου να εξασφαλιστεί η πληρέστερη παρουσίαση αυτών, θα περιγραφεί αρχικά το τυπικό μοντέλο στο οποίο στηρίζεται το RDF/S και η RQL και θα γίνει αναφορά στην τυπική ερμηνεία των βασικών RQL επερωτήσεων. Στη συνέχεια, θα παρουσιαστούν κάποια παραδείγματα επερωτήσεων με υποστασιοποίηση και ο προτεινόμενος τρόπος υποστήριξης της υποστασιοποίησης στην RQL.

5.1 Το τυπικό μοντέλο για το RDF/S

Στην ενότητα αυτή θα παρουσιάσουμε ένα μοντέλο δεδομένων με γράφους το οποίο επιχειρεί να «γεφυρώσει» τις ιδιαιτερότητες των προτύπων RDF M&S [LS99] και RDF Schema [BG02].

Έστω τα παρακάτω μη πεπερασμένα και ανά δύο ξένα μεταξύ τους σύνολα συμβόλων:

- Ονόματα Μετακλάσεων, $\mathcal{M} = \{m_1, m_2, m_3, \dots\}$
- Ονόματα Κλάσεων, $\mathcal{C} = \{c_1, c_2, c_3, \dots\}$
- Ονόματα Τύπων Ιδιοτήτων, $\mathcal{P} = \{p_1, p_2, p_3, \dots\}$
- Αναγνωριστικά Πόρων, $\mathcal{U} = \{u_1, u_2, u_3, \dots\}$
- Τιμές Πολυσυνόλων, $\mathcal{O} = \{o_1, o_2, o_3, \dots\}$
- Ατομικοί Τύποι, \mathcal{L} με ατομικές τιμές όπως *string*, *integer*, *dateTime* κλπ.

Το σύνολο \mathcal{M} περιλαμβάνει τα ονόματα *rdfs:Class* και *rdfs:Property*. Το *rdfs:Class* περιλαμβάνει τη ρίζα της ιεραρχίας των μετακλάσεων (\mathcal{M}_c) τα στιγμιότυπα των οποίων είναι κλάσεις, ενώ το *rdfs:Property* περιλαμβάνει τη ρίζα της ιεραρχίας των

μετακλάσεων (\mathcal{M}_p) τα στιγμιότυπα των οποίων είναι ιδιότητες. Το σύνολο C περιέχει την κλάση *rdfs:Resource* η οποία είναι η ρίζα της ιεραρχίας των κλάσεων των οντοτήτων. Το RDF/S υποστηρίζει επίσης δομημένες τιμές οι οποίες ονομάζονται πολυσύνολα (επίπεδο δεδομένων). Το σύνολο \mathcal{P} , εκτός από τα ονόματα των ιδιοτήτων, περιλαμβάνει επίσης αριθμητικές ετικέτες $\{1,2,3,\dots\}$ που χρησιμοποιούνται ως ονόματα ιδιοτήτων για τα μέλη των τιμών των πολυσυνόλων. Το σύνολο όλων των τιμών των πολυσυνόλων δηλώνεται με το O . Κάθε τιμή ενός πολυσυνόλου RDF/S είναι μοναδικά αναγνωρίσιμη από ένα URI και μπορεί να είναι το στιγμιότυπο ενός και μόνο ενός τύπου στο Bt , του *rdf:Bag*, *rdf:Seq* ή του *rdf:Alt*. Επιπλέον, το πεδίο ορισμού κάθε ατομικού τύπου t στο Lt δηλώνεται ως $\text{dom}(t)$, ενώ το \mathcal{L} αναπαριστά το σύνολο $\cup_{t \in Lt} \text{dom}(t)$. Στην πραγματικότητα, το Lt αναπαριστά το σύνολο όλων των βασικών τύπων δεδομένων της XML που χρησιμοποιούνται από ένα RDF/S Σχήμα.

Προκειμένου να αναπαραστήσουμε και τις υποστασιοποιημένες δηλώσεις ορίζουμε ένα επιπλέον σύνολο το *Statement* το οποίο περιλαμβάνει τριάδες του τύπου $\langle \text{rdf:subject}, \text{rdf:predicate}, \text{rdf:object} \rangle$. Όπου, $\text{rdf:predicate} \in \mathcal{P}$, $\text{rdf:subject} \in \text{domain}(\mathcal{P})$ και $\text{rdf:object} \in \text{range}(\mathcal{P})$.

Κάθε RDF σχήμα χρησιμοποιεί ένα πεπερασμένο σύνολο ονομάτων μετακλάσεων $M \subseteq \mathcal{M}$, ονομάτων κλάσεων $C \subseteq C$ και ονομάτων ιδιοτήτων $P \subseteq \mathcal{P}$, καθώς και το σύνολο των τύπων ονομάτων Lt, Bt και το σύνολο *rdf:Statement*. Τα ονόματα των ιδιοτήτων ορίζονται με τη βοήθεια των μετακλάσεων, των κλάσεων, των ατομικών αναγνωριστικών ή των τύπων ονομάτων των πολυσυνόλων και του *Statement*. Για κάθε $p \in P$, το $\text{domain}(p) \in M \cup C \cup \text{Statement}$ και το $\text{range}(p) \in M \cup C \cup Bt \cup Lt \cup \text{Statement}$.

Ορισμός 1: Ένα RDF σχήμα είναι μία οκτάδα $RS = (V_s, E_u, E_d, \psi_1, \psi_2, \lambda, \prec, N)$, όπου

$N = M \cup C \cup P \cup Bt \cup Lt \cup \text{Statement}$:

- V_s το σύνολο των κόμβων του σχήματος,
- E_u το σύνολο μη κατευθυνόμενων ακμών του σχήματος,
- E_d το σύνολο κατευθυνόμενων ακμών του σχήματος
- ψ_1 είναι μια συνάρτηση πρόσπτωσης $\psi_1: E_d \rightarrow V_s \times E_u$
- ψ_2 είναι μια συνάρτηση πρόσπτωσης $\psi_2: E_d \rightarrow E_u \times V_s$
- λ είναι μια συνάρτηση απόδοσης ετικετών $\lambda: V_s \cup E_u \cup E_d \rightarrow 2^M$ και
- \prec είναι μια αυστηρή μερική διάταξη τέτοια ώστε:
 - *Class* είναι η ρίζα της ιεραρχίας μετακλάσεων κλάσεων
 - *Property* είναι η ρίζα της ιεραρχίας μετακλάσεων ιδιοτήτων
 - *Resource* είναι η ρίζα της ιεραρχίας κλάσεων

Οι κόμβοι και οι ακμές ενός σχήματος ορίζονται μοναδικά από τα ονόματα τους στο N . Η συνάρτηση πρόσπτωσης ψ περιέχει το πεδίο ορισμού (*rdfs:domain*) και το σύνολο τιμών (*rdfs:range*) των ιδιοτήτων. Η συνάρτηση απόδοσης ετικετών λ συσχετίζει τα ονόματα των κλάσεων και των ιδιοτήτων με έναν από τους τύπους T που παρουσιάστηκαν παραπάνω.

5.1.1 Το Σύστημα Τύπων

Το σύστημα τύπων αποτελεί βασικό συστατικό στοιχείο του τυπικού μοντέλου του RDF/S το οποίο ενστερνιζόμαστε και επιτρέπει την αναπαράσταση των RDF δηλώσεων ως ατομικές ή σύνθετες τιμές δεδομένων. Μια άλλη συνεισφορά του συστήματος τύπων είναι ότι δίνει τη δυνατότητα στην *RQL*, η οποία θα περιγραφεί παρακάτω, στηριζόμενη σε αυτό να υποστηρίξει σύνθετες επερωτήσεις οι οποίες δημιουργούνται με συναρτησιακό τρόπο.

Ωστόσο, προκειμένου να υιοθετηθεί το σύστημα τύπων πρέπει να ληφθούν υπ' όψιν ένα σύνολο παραμέτρων. Πρώτα απ' όλα τα RDFS σχήματα, σε αντίθεση με τα σχήματα του οντοκεντρικού μοντέλου δεδομένων, δεν επιβάλλουν ένα αυστηρό σύστημα τύπων για τις περιγραφές των μεταδεδομένων, αυτό συνεπάγεται ότι ένας πόρος μπορεί να περιγραφεί λεκτικά με χρήση ιδιοτήτων οι οποίες συνδέονται καταχρηστικά με κλάσεις. Μία άλλη παράμετρος η οποία λαμβάνεται υπ' όψιν είναι ότι τα RDFS σχήματα επιτρέπουν για τον ίδιο πόρο πλέον της μίας περιγραφής. Επομένως, ένας πόρος είναι δυνατόν να ταξινομηθεί ταυτόχρονα σε πολλαπλές κλάσεις οι οποίες δεν συσχετίζονται αναγκαστικά με σχέσεις υποκλάσης. Επίσης, τα RDF/S σχήματα μπορούν εύκολα να επεκταθούν προς χάριν των αναγκών περιγραφών συγκεκριμένων κοινοτήτων. Ενώ τα RDF δεδομένα μπορεί να είναι ατομικές τιμές, αναγνωριστικά πόρων (URIs), ονόματα μετακλάσεων, κλάσεων, ιδιοτήτων, πολυσύνολα ή υποστασιοποιημένες προτάσεις.

Με βάση τις παραπάνω ιδιαιτερότητες του RDF/S, το σύστημα τύπων που υιοθετείται είναι το εξής:

$$\tau = \tau_{\mathcal{M}_c} \mid \tau_{\mathcal{M}_p} \mid \tau_C \mid \tau_{\mathcal{P}} [\tau, \tau] \mid \tau_U \mid \tau_L \mid \tau_s \mid \{\tau\} \mid [1:\tau_1, 2:\tau_2, \dots, n:\tau_n] \mid (1:\tau_1 + 2:\tau_2 + \dots + n:\tau_n)$$

όπου:

- $\tau_{\mathcal{M}_c}$ είναι μία μετακλάση κλάσεων,
- $\tau_{\mathcal{M}_p}$ είναι μία μετακλάση ιδιοτήτων,
- τ_C είναι μία κλάση
- $\tau_{\mathcal{P}} [\tau, \tau]$ είναι μία ιδιότητα,
- τ_U είναι ο τύπος για των αναγνωριστικών πόρων, ο οποίος περιλαμβάνει και τα αναγνωριστικά των χώρων ονοματοδοσίας,
- τ_L είναι ένας ατομικός τύπος ο οποίος ανήκει στο L , άρα $\tau_L = \{\text{string} \mid \text{integer} \mid \text{boolean} \mid \text{dateTime} \mid \dots\}$,
- τ_s είναι ο τύπος των υποστασιοποιημένων δηλώσεων,
- $[.]$ είναι ο τύπος Bag και τέλος,
- $(.)$ είναι ο τύπος Alternative.

Το σύνολο όλων των ονομάτων τύπων συμβολίζεται με T .

Το σύστημα αυτό δίνει τη δυνατότητα χειρισμού των κλάσεων και των ιδιοτήτων των σχημάτων RDFS ως αυθύπαρκτες οντότητες. Σε αντίθεση με τα παραδοσιακά οντοκεντρικά μοντέλα δεδομένων, οι κλάσεις και οι ιδιότητες του RDF (π.χ. σχέσεις και γνωρίσματα των πόρων) μεταφράζονται αντίστοιχα σε μονότιμες σχέσεις του τύπου $\{t_U\}$ και σε δίτιμες σχέσεις του τύπου $\{[t_U, t_U]\}$ (στην περίπτωση των σχέσεων)

ή του τύπου $\{[t_U, t_L]\}$ (στην περίπτωση των γνωρισμάτων). Επιπλέον, οι ιδιότητες των οποίων το πεδίο ορισμού (domain) και το σύνολο τιμών (range) είναι μία μετακλάση μπορεί να μεταφραστεί ως εξής: $\{[(t_C + t_P), (t_C + t_P)]\}$, ωστόσο στην προηγούμενη σχέση το πεδίο ορισμού δεν είναι δυνατόν να είναι ταυτόχρονα και κλάση και σχέση. Τέλος, η ανάθεση ενός πεπερασμένου συνόλου πόρων, του τύπου t_U , σε κάθε όνομα κλάσης, του τύπου t_C , είναι δυνατή μέσω μίας *συνάρτησης πληθυσμού κλάσης (class population function)*.

Ορισμός 2: Μια *συνάρτηση πληθυσμού κλάσης*, $\pi_c: C \rightarrow 2^U$, αναθέτει ένα πεπερασμένο σύνολο αναγνωριστικών πόρων σε μια κλάση του σχήματος, έτσι ώστε:

- για κάθε $c, c' \in C$, $c \prec c' \Rightarrow u \in \pi_c \Rightarrow u \notin \pi_{c'}$

Με τον ίδιο τρόπο, ορίζοντας τις *συναρτήσεις πληθυσμού* $\pi_{M_c}: M_c \rightarrow 2^C$ και $\pi_{M_p}: M_p \rightarrow 2^P$ είναι δυνατή η σύλληψη στιγμιοτύπων των μετακλάσεων με ονόματα κλάσεων και ιδιοτήτων. Οι συναρτήσεις πληθυσμού είναι το αντίθετο από τις συναρτήσεις απόδοσης ετικετών λ που χρησιμοποιούνται σε κάθε επίπεδο αφαίρεσης. Οι συναρτήσεις αυτές είναι *μερικές (partial)* εξαιτίας του ότι είναι δυνατόν να υπάρχουν (μετα)κλάσεις χωρίς πληθυσμό. Το σύνολο όλων των τιμών που μπορούν να κατασκευαστούν από τα ονόματα των κλάσεων ή των ιδιοτήτων, των αναγνωριστικών πόρων (URIs), των ατομικών τύπων και των υποστασιοποιημένων προτάσεων (Statement) που χρησιμοποιούν το σύστημα τύπων δηλώνεται με V και την *συνάρτηση μετάφρασης (interpretation function)* $[[\cdot]]$.

Ορισμός 3: Δοσμένης μιας *συνάρτησης πληθυσμού* π ονομάτων του (μετα)σχήματος, η *συνάρτηση μετάφρασης* $[[\cdot]]$ ορίζεται ως εξής:

- για τους ατομικούς τύπους: $[[\tau_L]] = \text{dom}(\tau_L)$
- για τους τύπους πόρων: $[[\tau_U]] = u \in U$
- για τους τύπους κλάσεων: $[[\tau_C]] = \{v \mid v \in \pi(c)\} \cup \{[[c']] \mid c' \prec c\}$
- για τους τύπους ιδιοτήτων: $[[\tau_P[\tau, \tau']]] = \{[v_1, v_2] \mid v_1 \in [[\tau]], v_2 \in [[\tau']]\} \cup \{[[p']] \mid p' \prec p\}$
- για τους τύπους μετακλάσεων: $[[\tau_M]] = \{v \mid v \in \pi(m)\} \cup \{[[m']] \mid m' \prec m\}$
- για τον τύπο *Bag*: $[[\{\tau\}]] = \{[v_1, v_2, \dots, v_j] \mid j > 0, \text{ για κάθε } i \in [1..j], v_i \in [[\tau]]\}$
- για τον τύπο *Alt*: $[[(1:\tau_1 + 2:\tau_2 + \dots + n:\tau_n)]]$ = $\{i:v_i \mid \text{για κάθε } i \in [1..n], v_i \in [[\tau_i]]\}$
- για τον τύπο *Statement*: $[[\tau_s]] = \{[v_1, p, v_2] \mid p \in P, v_1 \in [[\text{domain}(p)]]], v_2 \in [[\text{range}(p)]]\}$

Με βάση το σύστημα τύπων T και τις τιμές V , ορίζουμε τυπικά την *περιγραφή του σχήματος* και της *βάσης*.

Ορισμός 4 Ένα σχήμα περιγραφής S είναι μια δυάδα $S = (RS, \sigma)$, όπου $RS = (V_S, E_w, E_d, \psi_1, \psi_2, \lambda, \prec, N)$ είναι ένας έγκυρος γράφος RDF/S σχήματος και σ είναι μια συνάρτηση τύπου $\sigma: N \rightarrow T$.

Η συνάρτηση τύπου σ , συσχετίζει τα ονόματα των (μετα)κλάσεων με τους τύπους των (μετα)κλάσεων και τα ονόματα των ιδιοτήτων με τους τύπους των ιδιοτήτων.

Η συνάρτηση αποτίμησης (valuation function) συσχετίζει τους κόμβους και τις ακμές των RDF προτάσεων με μία από τις τιμές του V . Η συνάρτηση απόδοσης ονομάτων (ετικετών) (labeling function) εμπεριέχει την δήλωση `rdf:type`, η οποία συνδέει το γράφο το γράφο δεδομένων του RDF με το γράφο του RDF σχήματος. Πιο συγκεκριμένα, η συνάρτηση απόδοσης ετικετών επιστρέφει το όνομα μιας ή περισσοτέρων (μετα)κλάσεων η οποία μπορεί να ορίζεται σε αρκετές καλά ορισμένες ιεραρχίες ονομάτων. Σε αντίθεση με τα παραδοσιακά αντικειμενοστρεφή μοντέλα, όλα τα ονόματα κλάσεων που είναι κόμβοι πόρων έχουν ένα μοναδικό τύπο t_C . Τέλος, οι ατομικοί κόμβοι οι οποίοι έχουν τιμές literal ανήκουν στην μετάφραση συγκεκριμένων τύπων όπως *string*, *integer*, *date*, κτλ.

Ορισμός 6 Μία βάση περιγραφών RDF, με ένα σχήμα RS , είναι μια οκτάδα $RD = (V_D, E_{UD}, E_{DD}, \psi_1, \psi_2, \lambda, \prec, N)$, όπου

- V_D ένα σύνολο από κόμβους,
- E_{UD} ένα σύνολο από μη κατευθυνόμενες ακμές,
- E_{DD} ένα σύνολο από κατευθυνόμενες ακμές
- ψ_1 είναι η συνάρτηση πρόσπτωσης $\psi_1: E_d \rightarrow V_D \times E_{UD}$
- ψ_2 είναι η συνάρτηση πρόσπτωσης $\psi_2: E_d \rightarrow E_{UD} \times V_D$
- λ είναι μια συνάρτηση απόδοσης ετικετών $\lambda: V_D \cup E_{UD} \cup E_{DD} \rightarrow 2^{N \cup T}$

5.2 Η Γλώσσα Επερωτήσεων Μεταδεδομένων RDF

Στην ενότητα αυτή θα παρουσιάσουμε την *RQL* (*RDF Query Language*), μια γλώσσα επερωτήσεων για μεταδεδομένα RDF, η οποία δίνει τη δυνατότητα επερώτησης και σχημάτων RDF ως ημιδομημένα δεδομένα. Η *RQL* βασίζεται σε μία συναρτησιακή προσέγγιση (του τύπου OQL [CB97]), που επιτρέπει σύνθεση συναρτήσεων και τελεστών σάρωσης (iterators). Η *RQL* υποστηρίζει γενικευμένες εκφράσεις μονοπατιών, απόδοση τιμών ετικετών τόσο σε κόμβους (π.χ. κλάσεις) όσο και σε ακμές (π.χ. ιδιότητες). Τέλος, παρέχει ένα συγκεκριμένο σύνολο από προκαθορισμένους αλγεβρικούς τελεστές, οι οποίοι ονομάζονται φίλτρα (filters) και μπορούν να χρησιμοποιηθούν για τον σχηματισμό επερωτήσεων.

Η *RQL* στηρίζεται στο μοντέλο γράφων το οποίο παρουσιάστηκε στην προηγούμενη ενότητα, το οποίο περιλαμβάνει διάφορες μορφές μεταδεδομένων RDF, όπως ατομικές τιμές, αναγνωριστικά πόρων και σχέσεις, είτε μοναδιαίες μεταξύ κλάσεων είτε δυαδικές για τις ιδιότητες. Βασικό χαρακτηριστικό της *RQL* είναι το σύστημα τύπων το οποίο παρουσιάστηκε σε προηγούμενη ενότητα και ορίζει ένα σύνολο βασικών επερωτήσεων και τελεστών σάρωσης που μπορούν να συνδυαστούν με συναρτησιακό τρόπο για τη δημιουργία επερωτήσεων. Η *RQL* “κρύβει” τη διάκριση

μεταξύ ατομικών τιμών, αναγνωριστικών πόρων και ετικετών κλάσεων ή ιδιοτήτων με σκοπό την ομοιόμορφη επερώτηση κόμβων και ακμών σε μεταδεδομένα ή σχήματα RDF.

Η RQL περιέχει τόσο απλές επερωτήσεις για την επερώτηση ονομάτων τόσο σε επίπεδο γράφων σχημάτων όσο και σε επίπεδο γράφων δεδομένων. Για παράδειγμα, οι βασικές επερωτήσεις:

- **Class**
- **Property**

επιστρέφουν τα ονόματα όλων των κλάσεων και όλων των ιδιοτήτων αντίστοιχα που ορίζονται στο σχήμα.

Ενώ, η επίσης απλή επερώτηση **Artist** επιστρέφει όλα τα στιγμιότυπα της κλάσης αυτής.

Οι βασικές επερωτήσεις των κλάσεων, των ιδιοτήτων καθώς και οι επερωτήσεις για το πεδίο τιμών και το σύνολο τιμών των ιδιοτήτων φαίνονται στον πίνακα 5.3. Στον πίνακα αυτόν παρουσιάζεται επίσης η τυπική ερμηνεία αυτών.

Έκφραση	Ερμηνεία
subClassOf(c)	$\{c' \in C \mid c' \prec c\}$
subClassOf^(c)	$\{c' \in C \mid c' \prec c, \text{δεν υπάρχει } c'' \in C, c'' \prec c \prec \text{ και } c' \prec c''\}$
subPropertyOf(c)	$\{p' \in P \mid p' \prec p\}$
subPropertyOf^(c)	$\{p' \in P \mid p' \prec p, \text{δεν υπάρχει } p'' \in P, p'' \prec p \text{ και } p' \prec p''\}$
P	$\{ \{ p \} \}$
^p	$\{ \{ p \} \} - \cup p' \prec p (\{ \{ p' \} \})$
col1 intersect col2	$\{ \{ col1 \} \} \cap \{ \{ col2 \} \}$
col1 union col2	$\{ \{ col1 \} \} \cup \{ \{ col2 \} \}$
col1 minus col2	$\{ \{ col1 \} \} \setminus \{ \{ col2 \} \}$
count(col)	$n, \text{ if } \exists i1, \dots, in, (vi1, \dots, vin, (j,i1 (j(in, (j, vj)(\{ \{ col \} \}$
Seq [n]	$vn, \text{ if } 1 (n (1, (n, vn)(\{ \{ seq \} \}$

Πίνακας 5.1: Τυπική ερμηνεία των βασικών RQL ερωτήσεων

Η RQL υποστηρίζει επίσης γενικευμένες εκφράσεις μονοπατιών με μεταβλητές σε ετικέτες κόμβων (κλάσεων) και ακμών (ιδιοτήτων).

Για παράδειγμα, στην επερώτηση

```
SELECT X, Y
FROM {X}creates{Y}
```

ορίζονται οι μεταβλητές στις ετικέτες κλάσεων X και Y στο μονοπάτι των δεδομένων. Η επερώτηση αυτή επιστρέφει όλες τις δυάδες των δημιουργών και των δημιουργημάτων αυτών.

Επίσης, στην RQL επιτρέπεται ο ορισμός μεταβλητών στο μονοπάτι του σχήματος, όπως για παράδειγμα στην επερώτηση

```
SELECT $X, $Y
FROM {$X}creates{$Y}
```

η οποία επιστρέφει τις κλάσεις που μπορούν να εμφανιστούν σαν πεδίο ορισμού και σύνολο τιμών της ιδιότητας creates.

Στον Πίνακα 5.4 παρουσιάζεται η τυπική ερμηνεία των βασικότερων εκφράσεων μονοπατιών της RQL, όπου γίνεται διάκριση σε μονοπάτια δεδομένων, σχήματος και μικτά μονοπάτια.

	Έκφραση Μονοπατιού	Ερμηνεία
Μονοπάτι Δεδομένων	$c\{X\}$ $\{X\}p\{Y\}$ $\{X\}@P\{Y\}$ $\$X\{Y\}$	$\{v \mid v \in [[c]]\}$ $\{ \langle v_1, v_2 \rangle \mid \langle v_1, v_2 \rangle \in [[p]] \}$ $\{ \langle v_1, p, v_2 \rangle \mid p \in P, \langle v_1, v_2 \rangle \in [[p]] \}$ $\{ \langle c, v \rangle \mid c \in C, v \in [[c]] \}$
Μονοπάτι Σχήματος	Class $\{X\}$ $\$X$ Property $\{P\}$ $@P$ $c\{\$C\}$ $\{\$X\}p\{\$Y\}$ $\$X\{\$Y\}$ $\{\$X\}@P\{\$Y\}$	$\{c \mid c \in C\}$ $\{c \mid c \in C\}$ $\{p \mid p \in P\}$ $\{p \mid p \in P\}$ $\{c' \mid c' \in C, c' \leq c\}$ $\{ \langle c_1, c_2 \rangle \mid c_1, c_2 \in C, c_1 \leq \text{domain}(p), c_2 \leq \text{range}(p) \}$ $\{ \langle c_1, c_2 \rangle \mid c_1, c_2 \in C, c_2 \leq c_1 \}$ $\{ \langle c_1, p, c_2 \rangle \mid p \in P, c_1, c_2 \in C, c_1 \leq \text{domain}(p), c_2 \leq \text{range}(p) \}$
Μικτό Μονοπάτι	$c\{X;\$C\}$ $\{X;\$Z\}p\{Y;\$W\}$ $p\{Y;\$W\}$ $\{X\}p\{Y;\$W\}$ $\{X\}p\{\$W\}$ $\{\$Z\}p\{Y;\$W\}$ $\{X;\$Z\}@P\{Y;\$W\}$	$\{ \langle v, c' \rangle \mid c' \in C, c' \leq c, v \in [[c]] \}$ $\{ \langle v_1, c_1, v_2, c_2 \rangle \mid c_1, c_2 \in C, c_1 \leq \text{domain}(p), v_1 \in [[c_1]], c_2 \leq \text{range}(p), v_2 \in [[c_2]], \langle v_1, v_2 \rangle \in [[p]] \}$ $\{ \langle v_2, c_2 \rangle \mid c_2 \in C, c_2 \leq \text{range}(p), v_2 \in [[c_2]], \langle v_1, v_2 \rangle \in [[p]] \}$ $\{ \langle v_1, v_2, c_2 \rangle \mid c_2 \in C, c_2 \leq \text{range}(p), v_2 \in [[c_2]], \langle v_1, v_2 \rangle \in [[p]] \}$ $\{ \langle v_1, c_2 \rangle \mid c_2 \in C, c_2 \leq \text{range}(p), v_2 \in [[c_2]], \langle v_1, v_2 \rangle \in [[p]] \}$ $\{ \langle c_1, v_2, c_2 \rangle \mid c_1, c_2 \in C, c_1 \leq \text{domain}(p), v_1 \in [[c_1]], c_2 \leq \text{range}(p), v_2 \in [[c_2]], \langle v_1, v_2 \rangle \in [[p]] \}$ $\{ \langle v_1, c_1, p, v_2, c_2 \rangle \mid p \in P, c_1, c_2 \in C, c_1 \leq \text{domain}(p), v_1 \in [[c_1]], c_2 \leq \text{range}(p), v_2 \in [[c_2]], \langle v_1, v_2 \rangle \in [[p]] \}$

Πίνακας 5.2: Τυπική ερμηνεία των βασικών εκφράσεων μονοπατιών της RQL

Όπως φαίνεται και από τα παραδείγματα που αναφέρθηκαν παραπάνω, οι εκφράσεις μονοπατιών της RQL χρησιμοποιούν τη δομή **SELECT-FROM-WHERE**. Ο όρος **SELECT** ορίζει τη λίστα των μεταβλητών, οι τιμές των οποίων θα εμφανιστούν στα αποτελέσματα. Ο όρος **FROM** περιέχει τις εκφράσεις μονοπατιών που ορίζονται, και οι οποίες καθορίζουν τα τμήματα του RDF/S γράφου που θα λάβουν μέρος στην αποτίμηση της επερώτησης. Ο όρος **WHERE**, ο οποίος είναι προαιρετικός, ορίζει τα κριτήρια φιλτραρίσματος της πληροφορίας βάση του όρου FROM και περιορίζει το πλήθος των αποτελεσμάτων. Τέλος, ο όρος USING NAMESPACE είναι δυνατόν να χρησιμοποιηθεί. Ο όρος αυτός χρησιμοποιείται για τη δήλωση μεταβλητών που αντιστοιχούν σε προθέματα χώρων ονοματοδοσίας. Τα προθέματα των χώρων επερώτησης τα οποία έχουν δηλωθεί στον όρο USING NAMESPACE μπορούν στη συνέχεια να χρησιμοποιηθούν στην 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.

Το σύστημα τύπων εξασφαλίζει τη δυνατότητα σύνθεσης πολύπλοκων επερωτήσεων με χρήση βασικών εκφράσεων επερώτησης. Πιο συγκεκριμένα, για να είναι έγκυρη μία έκφραση επερώτησης, οι τελεστές 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 τελεστές τύπου ιδιότητας (τP), αλλιώς επιστρέφεται ένα σφάλμα τύπου. Κάθε ετικέτα που χρησιμοποιείται συνοδεύεται από τον τύπο της, ο οποίος εξασφαλίζει τη συναρτησιακή σύνθεση των εκφράσεων επερώτησης.

5.3 Υποστήριξη Επερωτήσεων Υποστασιοποίησης στην RQL

Αν και η RQL υποστηρίζει επερωτήσεις τόσο σε επίπεδο σχήματος όσο και σε επίπεδο δεδομένων καθώς και δήλωση μεταβλητών στο μονοπάτι του σχήματος ωστόσο δεν υποστηρίζει επερωτήσεις για βάσεις μεταδεδομένων οι οποίες περιέχουν υποστασιοποιημένες προτάσεις. Στόχος της παραγράφου αυτής είναι η περιγραφή των προσθηκών που πρέπει να γίνουν στην RQL ώστε αυτή να υποστηρίζει την επερώτηση υποστασιοποιημένων προτάσεων. Η υποστήριξη της λειτουργίας αυτής είναι ζωτικής σημασίας για εφαρμογές στις οποίες είναι η τα μεταδεδομένα πρέπει να χαρακτηρίζονται από αξιοπιστία. Επίσης, είναι πολύ χρήσιμη για επερωτήσεις των εκδόσεων σχημάτων, όπου για κάθε τμήμα του σχήματος – μετακλάση, κλάση, ιδιότητα, υπο-κλάση, υπο-ιδιότητα – διατηρείται πληροφορία σχετικά με το πότε τροποποιήθηκε και από ποιόν. Μια άλλη περίπτωση την οποία θα εξετάσουμε στην οποία είναι χρήσιμη η υποστασιοποίηση είναι η υποστήριξη τριαδικών σχέσεων από το RDF.

Προκειμένου να υποστηριχθεί η επερώτηση υποστασιοποιημένων προτάσεων πρέπει να τροποποιηθεί το σύστημα τύπων έτσι ώστε να περιλαμβάνει και ένα τύπο για τις υποστασιοποιημένες προτάσεις. Για το σκοπό αυτό θεωρούμε έναν επιπλέον τύπο, τον

Statement Type $[[\tau_s]] = \{(v_1, v_2, v_3) \mid v_2.\text{typeof} = \text{Property}, v_1.\text{typeof} = \text{domain}(v_2), v_3.\text{typeof} = \text{range}(v_2)\}$

Επίσης, ορίζουμε τις συναρτήσεις
 subject(s)
 predicate(s)
 object(s)

οι οποίες έχουν πεδίο ορισμού μια υποστασιοποιημένη πρόταση και επιστρέφουν αντίστοιχα το υποκείμενο, το κατηγορούμενο και το αντικείμενο της υποστασιοποιημένης πρότασης.

Ορίζουμε, επίσης, την επερώτηση ReifProperty, η οποία επιστρέφει όλες τις ιδιότητες οι οποίες έχουν ως αντικείμενο ή ως υποκείμενο του τύπου Statement, δηλαδή μια υποστασιοποιημένη πρόταση. Η παραπάνω συνάρτηση ουσιαστικά εκτελεί την παρακάτω επερώτηση:

```
((selectP
from Property {P}
where domain(P)= Statement)
```

```
union
(select P
from Property {P}
where range(P)= Statement))
```

και επιστρέφει ένα πίνακα με όλες τις ιδιότητες που ικανοποιούν αυτές τις ιδιότητες.

Τέλος, ορίζουμε δύο ακόμα επερωτήσεις τις SubReifProperty και ObjReifProperty. Η επερωτηση SubReifProperty επιστρέφει όλες τις ιδιότητες οι οποίες έχουν ως υποκείμενο μια υποστασιοποιημένη πρόταση. Κατά την επερωτηση αυτή εκτελείται ουσιαστικά το παρακάτω ερώτημα.

```
select P
from Property {P}
where range(P)= Statement
```

Αντίστοιχα, η επερωτηση ObjReifProperty, επιστρέφει όλες εκείνες τις ιδιότητες που έχουν ως αντικείμενο μια υποστασιοποιημένη πρόταση. Το ερώτημα που εκτελείται σε αυτήν την περίπτωση είναι:

```
select P
from Property {P}
where domain(P)= Statement
```

Όπως μπορούμε να παρατηρήσουμε τα αποτελέσματα της επερωτησης ReifProperty είναι η ένωση των αποτελεσμάτων των επερωτήσεων SubReifProperty και ObjReifProperty. Επομένως,

$ReifProperty \equiv SubReifProperty \cup ObjReifProperty$

Μια ακόμα επερωτηση την οποία πρέπει να ορίσουμε είναι η:

Statement

η οποία επιστρέφει όλες τις υποστασιοποιημένες προτάσεις οι οποίες περιέχονται στην βάση μεταδεδομένων.

Στην περίπτωση ενός συστήματος πολλών πρακτόρων (multi-agent), όπου ενδιαφερόμαστε για την πηγή προέλευσης των πληροφοριών πρέπει να υποστηριχτούν επερωτήσεις όπως οι παρακάτω:

Επερωτηση 1:

Τι πιστεύει (believes) ο πόρος που το URI του είναι "John" ;

```
select Y
from {X}believes {Y}
where X = "John"
```

όπου το Y είναι του τύπου Statement καθώς το σύνολο τιμών της ιδιότητας Y είναι οι υποστασιοποιημένες δηλώσεις. Τα αποτελέσματα της παραπάνω επερώτησης είναι τριπλέτες διότι μια υποστασιοποιημένη πρόταση αποτελείται από τρία στοιχεία, το υποκείμενο, το κατηγορούμενο και το αντικείμενο.

Επερώτηση 2:

Τι πιστεύει (believes) ο πόρος με URI "John" σχετικά με το RDF;

```
(select Y
from {X}believes{Y}
where X = "John" and subject(Y) = "RDF" )
union
(select Y
from {X}believes{Y}
where X = "John" and object(Y) = "RDF")
```

Η ιδιότητα believes έχει σαν πεδίο ορισμού μία κλάση και σύνολο τιμών του τύπου Statement. Στην παραπάνω επερώτηση θέτονται οι περιορισμοί το υποκείμενο ιδιότητας να έχει υπί το οποίο να ταιριάζει με το "John" και είτε το υποκείμενο (subject(Y)) είτε το αντικείμενο (object(Y)) των υποστασιοποιημένων δηλώσεων να ταιριάζει με το "RDF". Τέλος, η επερώτηση αυτή επιστρέφει το Y, το οποίο αποτελείται από τις τριάδες, υποκείμενο, κατηγορούμενο, αντικείμενο οι οποίες πληρούν αυτές τις προδιαγραφές.

Επερώτηση 3:

Ποια είναι τα υποκείμενα και τα κατηγορούμενα των υποστασιοποιημένων προτάσεων που πιστεύει (believes) ο John;

```
select subject(Y), predicate(Y)
from {X}believes{Y}
where X = "John"
```

Ο μόνος έλεγχος που γίνεται στην παραπάνω πρόταση είναι αν το υποκείμενο της ιδιότητας believes ταιριάζει με το "John" και επιστρέφεται μόνο το υποκείμενο και το κατηγορούμενο των υποστασιοποιημένων προτάσεων.

Επερώτηση 4:

Να βρεθούν όλα όσα έχει πιστεύει, ισχυρίζεται, δηλώνει (believes, claims, declares) καθώς και όλα όσα έχουν ειπωθεί, ισχυριστεί (was_told_by, was_claimed_by) από τον John ;

```
(select Y
from {X}@P{Y}
where X = "John" and @P in ObjReifProperty)
union
(select X
```

```
from {X}@P{Y}
where Y = "John" and @P in SubReifProperty)
```

Στην παραπάνω επερώτηση βρίσκονται όλες οι ιδιότητες οι οποίες ανήκουν στις Reifying Properties, δηλαδή αυτές που έχουν είτε πεδίο ορισμού είτε σύνολο τιμών του τύπου Statement. Σε αυτές τις ιδιότητες που έχουν πεδίο ορισμού Statement (SubReifProperty), ελέγχεται αν το αντικείμενο τους ταιριάζει με το "John" ενώ σε εκείνες που έχουν σύνολο τιμών του τύπου Statement (ObjReifProperty) αν το υποκείμενο τους ταιριάζει με το "John".

Επερώτηση 5:

Να βρεθούν όλες οι απόψεις (believes, claims, declares, was_told_by κτλ.) του John για το RDF;

```
(select Y
from {X}@P{Y}
where X = "John" and P in ObjReifProperty and subject(Y) = "RDF")
union
(select Y
from {X}@P{Y}
where X = "John" and P in ObjReifProperty and object(Y) = "RDF")
(select Y
from {X}@P{Y}
where Y = "John" and P in SubReifProperty and subject(Y) = "RDF")
union
(select Y
from {X}@P{Y}
where Y = "John" and P in SubReifProperty and object(Y) = "RDF")
```

Η παραπάνω επερώτηση μοιάζει με την προηγούμενη αλλά περιέχει ένα επιπλέον έλεγχο. Ελέγχεται αν το υποκείμενο ή το αντικείμενο των υποστασιοποιημένων προτάσεων ταιριάζει με το "RDF".

Επερώτηση 6:

Ποια είναι η πηγή της πληροφορίας (Picasso, paints, Guernica);

```
(select X, @P
from {X}@P{Y}
where (@P in ObjReifProperty) and (subject(Y) = "Picasso") and (object(Y) =
"Guernica") and (predicate(Y) = "paints"))
union
(select Y, @P
from {X}@P{Y}
where (@P in SubReifProperty) and (subject(X) = "Picasso") and (object(X) =
"Guernica") and (predicate(X) = "paints"))
```

Στην επερώτηση αυτή ψάχνουμε για τις ιδιότητες οι οποίες έχουν πεδίο ορισμού ή σύνολο τιμών του τύπου Statement και στην συνέχεια ελέγχουμε αν οι

υποστασιοποιημένες προτάσεις αυτών των ιδιοτήτων έχουν υποκείμενο που ταιριάζει στο “Picasso”, κατηγορούμενο που ταιριάζει στο “Paints” και αντικείμενο που ταιριάζει στο “Guernica”.

Επερώτηση 7:

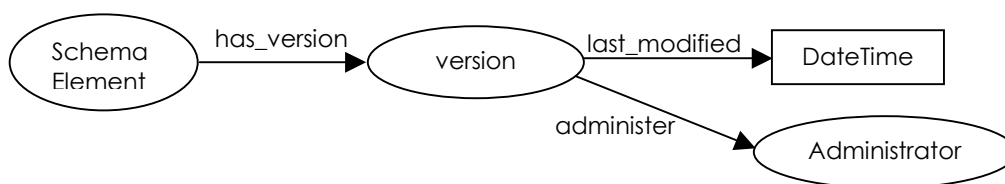
Να βρεθούν όλες οι υποστασιοποιημένες προτάσεις που αφορούν στο *RDF*;

```
select S
from Statement(S)
where subject(S) = "RDF" or object(S) = "RDF"
```

Στην επερώτηση αυτή ελέγχουμε ποιες από τις υποστασιοποιημένες προτάσεις έχουν υποκείμενο που ταιριάζει με το “RDF” ή αντικείμενο που ταιριάζει με το “RDF”.

Μια άλλη εφαρμογή στην οποία είναι χρήσιμη η υποστασιοποίηση είναι αυτή του Schema Versioning. Παρακάτω εξετάζουμε κάποιες επερωτήσεις οι οποίες απαιτούνται σε αυτού του τύπου τις εφαρμογές.

Έχουμε το εξής σχήμα:



Προκειμένου να υλοποιήσουμε το Schema Versioning κάνουμε τις εξής υποθέσεις:

- Στοιχεία του σχήματος θεωρούνται
 - Οι μετακλάσεις
 - Οι κλάσεις
 - Οι ιδιότητες
 - Το πεδίο ορισμού μιας ιδιότητας
 - Το σύνολο τιμών μιας ιδιότητας
 - Οι σχέσεις υποκλάσης
 - Οι σχέσεις υποιδιότητας
- Στο σχήμα μπορούν να γίνονται μόνο προσθήκες.
- Κάθε έκδοση του σχήματος αποτελείται από τα στοιχεία που προστέθηκαν σε αυτήν συν όλα τα στοιχεία της προηγούμενης έκδοσης. Για παράδειγμα, η Έκδοση 2 του σχήματος περιλαμβάνει όλα τα στοιχεία της Έκδοσης 1 καθώς και όλα τα στοιχεία που ανήκουν στην Έκδοση 2.

Ας εξετάσουμε μερικές επερωτήσεις που μπορούν να γίνουν σε μια εφαρμογή Schema Versioning.

Επερώτηση 8:

Να δοθεί όλο το σχήμα και η έκδοση στην οποία προστέθηκε κάθε στοιχείο του σχήματος.

```
Select subject(X), predicate(X), object(X), Y
from {X}has_version{Y}
```

Η ιδιότητα `has_version` έχει πεδίο ορισμού του τύπου `Statement` και σύνολο τιμών την κλάση `Version`. Επομένως, η παραπάνω επερώτηση επιστρέφει όλες τις υποστασιοποιημένες προτάσεις που αναπαριστούν το σχήμα καθώς και την έκδοση στην οποία προστέθηκε κάθε πρόταση.

Επερώτηση 9:

Να βρεθούν τα στοιχεία που προστέθηκαν κατά την Έκδοση 2 (*Version2*) του σχήματος.

```
Select subject(X), predicate(X), object(X)
from {X}has_version{Y}
where Y = "Version2"
```

Η παραπάνω επερώτηση μοιάζει με την προηγούμενη με μόνη διαφορά ότι έχει προστεθεί ο περιορισμός *where Y = "Version2"* ο οποίος περιορίζει τα αποτελέσματα στις προτάσεις εκείνες οι οποίες έχουν προστεθεί κατά την *Version2*.

Επερώτηση 10:

Να βρεθούν όλα τα στοιχεία της Έκδοσης 1 (*Version1*) και της Έκδοσης 2 (*Version2*) του σχήματος.

```
(Select subject(X), predicate(X), object(X)
from {X}has_version{Y}
where Y = "Version2")
union
(Select subject(X), predicate(X), object(X)
from {X}has_version{Y}
where Y = "Version1"
)
```

Η επερώτηση αυτή επιστρέφει όλες τις προτάσεις οι οποίες ανήκουν είτε στην *Version1* είτε στην *Version2*.

Επερώτηση 11:

Να βρεθούν όλα τα στοιχεία που έχουν προστεθεί στο σχήμα μέχρι τις 01-01-2003.

```
Select subject(X), predicate(X), object(X)
from {X}has_version{Y}.last_modified{Z}
where Z<= 2003-01-01
```

Στην επερώτηση αυτή επιστρέφονται μόνο οι προτάσεις αυτές που έχουν προστεθεί στο σχήμα πριν από τις 01-01-2003.

Επερώτηση 12:

Να βρεθούν όλα τα στοιχεία του σχήματος που αφορούν στην κλάση Artist και η αντίστοιχη έκδοση κάθε στοιχείου.

```
Select subject(X), predicate(X), object(X), Y
from {X}has_version{Y}
where subject(X) = "Artist" or object(X) = "Artist"
```

Η παραπάνω επερώτηση επιστρέφει όλες εκείνες τις προτάσεις οι οποίες έχουν είτε σαν υποκείμενο είτε σαν κατηγορούμενο το "Artist".

Επερώτηση 13:

Να βρεθούν όλες οι αλλαγές που έχουν γίνει στην σχέση creates και η έκδοση που έγινε κάθε αλλαγή.

```
Select subject(X), predicate(X), object(X), Y
from {X}has_version{Y}
where subject(X) = "creates" or object(X) = "creates"
```

Η επερώτηση αυτή μοιάζει με την παραπάνω αλλά αντί για "Artist" ελέγχει για το "creates".

Επερώτηση 14:

Να βρεθούν όλες οι αλλαγές που έχουν γίνει στην σχέση creates μετά τις 01-01-2003.

```
Select subject(X), predicate(X), object(X), Y
from {X}has_version{Y}.last_modified{Z}
where ((subject(X) = "creates") or (object(X) = "creates")) and Z>=2003-01-01
```

Στην επερώτηση αυτή επιστρέφονται όλες οι προτάσεις που έχουν ως υποκείμενο ή ως αντικείμενο το creates και επίσης έχουν τροποποιηθεί μετά τις 01-01-2003.

Επερώτηση 15:

Να βρεθεί ποιος έχει κάνει την προσθήκη (*Painter*, *rdf:type*, *Class*).

```
Select subject(X), predicate(X), object(X), Z
from {X}has_version{Y}.administer{Z}
where (subject(X) = "Painter") and (object(X) = "Class") and (object(X) = "rdf:type")
```

Η επερώτηση αυτή επιστρέφει την υποστασιοποιημένη πρόταση που έχει σαν στοιχεία τα "*Painter*", "*isA*", "*Class*" καθώς και το uri αυτού που έχει κάνει την προσθήκη αυτής της πρότασης.

Τέλος, μια άλλη ενδιαφέρουσα εφαρμογή είναι η υποστήριξη των τρυαδικών σχέσεων. Πιθανές επερωτήσεις οι οποίες θα ήταν απαραίτητο να υποστηριχθούν είναι οι εξής:

Επερώτηση 16:

Να δοθούν όλα τα στιγμιότυπα της τριαδικής σχέσης *supply*.

```
select subject(X), object(X), Y
from {X}supply{Y}
```

Η επερώτηση αυτή επιστρέφει όλες τριάδες <προμηθευτής, υποκατάστημα, προϊόν>. Το πεδίο ορισμού της σχέσης *supply* είναι μια υποστασιοποιημένη πρόταση, η οποία περιέχει την πληροφορία του προμηθευτή και του υποκαταστήματος που προμηθεύει αυτός. Επομένως, το *subject(X)* επιστρέφει το όνομα του προμηθευτή ενώ το *object(X)* το όνομα του υποκαταστήματος που προμηθεύει ο συγκεκριμένος προμηθευτής. Τέλος, η ιδιότητα *supply* έχει ως σύνολο τιμών την κλάση προϊόντα. Επομένως, το *Y* επιστρέφει το προϊόν το οποίο προμηθεύει ο συγκεκριμένος προμηθευτής το υποκατάστημα.

Επερώτηση 17:

Να δοθούν όλα τα στιγμιότυπα της τριαδικής σχέσης *supply* που σχετίζονται με το uri το οποίο περιέχει το "*John*".

```
select subject(X), object(X), Y
from {X}supply{Y}
where (subject(X) = "John" ) or (object(X) = "John") or Y = "John"
```

Η παραπάνω επερώτηση επιστρέφει εκείνα τα στιγμιότυπα της τριαδικής σχέσης *supply* στα οποία ένα τουλάχιστον μέρος της τριάδας περιέχει στο uri αυτού το "*John*".

Επερώτηση 18:

Να βρεθεί ποιος προμηθευτής προμηθεύει ποια υποκαταστήματα με το *product1*.

```
select subject(X), object(X)
from {X}supply{Y}
where Y = "product1"
```

Στην επερώτηση αυτή, ελέγχεται ποιοι προμηθευτές και ποια υποκαταστήματα προμηθεύονται με το *product1*. Το Y αναπαριστά την κλάση των αντικειμένων και για το λόγο αυτό θέτουμε σαν κριτήριο το where Y = "product1".

Επερώτηση 19:

Να βρεθεί ποιος προμηθεύει του *Department1* με το *product2*.

```
select subject(X), object(X), Y
from {X}supply{Y}
where object(X) = "Department1" and Y = "product2"
```

Τέλος, στην παραπάνω επερώτηση θέτουμε δύο κριτήρια, το object(X) να περιέχει το "*Department1*" και το Y να περιέχει το "*product2*". Οπότε η επερώτηση αυτή επιστρέφει όλες εκείνες τις τριάδες στις οποίες το προϊόν είναι το *product2* και το υποκατάστημα είναι το *Department1*.

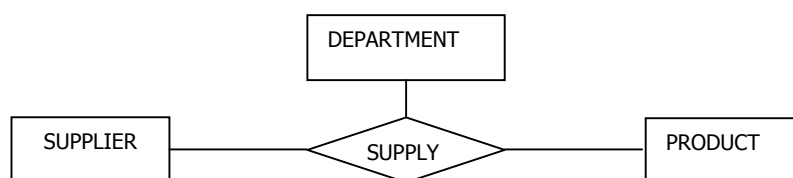
ΚΕΦΑΛΑΙΟ 6

Σενάρια Χρήσης Υποστασιοποίησης

Στα προηγούμενα κεφάλαια έγινε μια μελέτη της υποστήριξης της υποστασιοποίησης στο μοντέλο του RDF/S, τόσο σε επίπεδο αποθήκευσης των μεταδεδομένων σε σχεσιακή βάση όσο και σε επίπεδο επερώτησης αυτών. Στο παρόν κεφάλαιο θα εξετάσουμε συγκεκριμένα σενάρια χρήσης της υποστασιοποίησης, όπως για παράδειγμα την αναπαράσταση μη-δυναδικών σχέσεων και την επικοινωνία μεταξύ πρακτόρων με τη βοήθεια του RDF.

6.1 Αναπαράσταση Μη-Δυναδικών Σχέσεων

Το μοντέλο δεδομένων της RDF υποστηρίζει μόνο δυναδικές σχέσεις, δηλαδή, μία πρόταση προσδιορίζει μία σχέση μεταξύ δύο πόρων (resources). Στα παρακάτω παραδείγματα θα προτείνουμε ένα τρόπο για την αναπαράσταση σχέσεων μεγαλύτερου βαθμού (arity) στην RDF, με χρήση της υποστασιοποίησης. Ας εξετάσουμε την παρακάτω τριαδική σχέση:



Σχήμα 6.1 Τριαδική σχέση

Στο παραπάνω μοντέλο Οντοτήτων-Συσχετίσεων έχουμε την τριαδική σχέση – «Προμηθεύει» (SUPPLY) μεταξύ του Προμηθευτή (SUPPLIER), του Προϊόντος (PRODUCT) και του Υποκαταστήματος (DEPARTMENT).

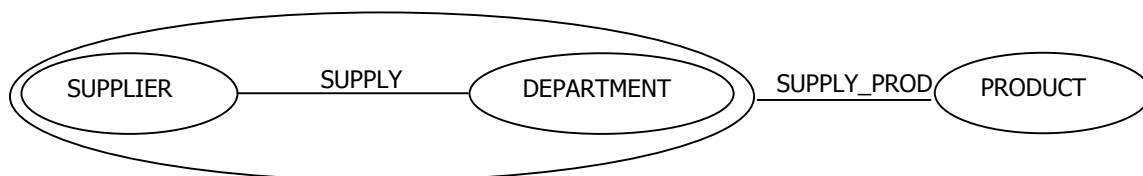
Η παραπάνω τριαδική σχέση δεν μπορεί να αναπαράσταθει στο μοντέλο δεδομένων της RDF. Οι απλοί γράφοι που μπορούμε να έχουμε στην RDF είναι οι παρακάτω:



Σχήμα 6.2 Αναπαράσταση τριαδικής σχέσης με δύο δυναδικές

Ωστόσο, αυτό που επιθυμούμε να αναπαραστήσουμε με την παραπάνω τριαδική σχέση είναι η συσχέτιση μεταξύ αυτών των τριών οντοτήτων, δηλ. ό,τι ο προμηθευτής προμηθεύει το συγκεκριμένο υποκατάστημα με κάποια συγκεκριμένα προϊόντα.

Η αναπαράσταση των τριαδικών σχέσεων μπορεί να γίνει με τη βοήθεια της υποστασιοποίησης όπως φαίνεται στο παρακάτω σχήμα.



Σχήμα 6.3 Αναπαράσταση τριαδικής σχέση με υποστασιοποίηση

Όσον αφορά στο ποιά από τις δύο δυαδικές σχέσεις που συνιστούν την τριαδική θα υποστασιοποιηθεί εξαρτάται από την περίπτωση. Θεωρούμε ότι η επιλογή της σχέσης που θα υποστασιοποιηθεί μπορεί να γίνει είτε από τον χρήστη είτε από το σύστημα. Πάντως και στις δύο περιπτώσεις υπάρχει συμμετρία, δηλαδή το νόημα είναι το ίδιο ανεξάρτητα από το ποιά από τις δύο δυαδικές σχέσεις θα υποστασιοποιηθεί τελικά.

Επίσης, πρέπει να τονιστεί ότι ο λόγος χρήσης της υποστασιοποίησης οφείλεται στο γεγονός ότι η τριαδική σχέση υφίστανται σαν μία ενότητα διαφορετικά θα μπορούσαμε να «σπάσουμε» την τριαδική σχέση σε δύο δυαδικές. Ωστόσο, αν κάναμε κάτι τέτοιο θα χανόταν το νόημα της τριαδικής σχέσης κάτι που διατηρείται με τη βοήθεια της υποστασιοποίησης.

Ένα πλεονέκτημα που προσφέρει η υποστασιοποίηση είναι ότι δεν έχουμε επανάληψη πληροφορίας. Για να γίνει σαφές τι εννοούμε ας δούμε το παράδειγμα που ο SUPPLIER s1 προμηθεύει το DEPARTMENT d1 με τα PRODUCTS p1,p2,p3. Η πληροφορία ότι ο s1 προμηθεύει το d1 αποθηκεύεται στην υποστασιοποιημένη πρόταση με αναγνωριστικό id1. Το μόνο που απαιτείται προκειμένου να αναπαρασταθεί η τριαδική σχέση είναι η σύνδεση του αναγνωριστικού id1 της υποστασιοποιημένης πρότασης με τα p1,p2 και p3.

ID	S	P	O
id1	s1	supply	d1

Σχέση SUPPLY_PR

S	O
id1	p1
id1	p2
id1	p3

Σε RDF μπορεί να γραφεί ως εξής:

```
<rdf:Description rdf:ID="d1">
  <rdf:type rdf:resource="&rdf;Statement"/>
  <rdf:subject rdf:resource="#s1"/>
  <rdf:object="d1"/>
  <rdf:predicate rdf:resource="#supply"/>
</rdf:Description>

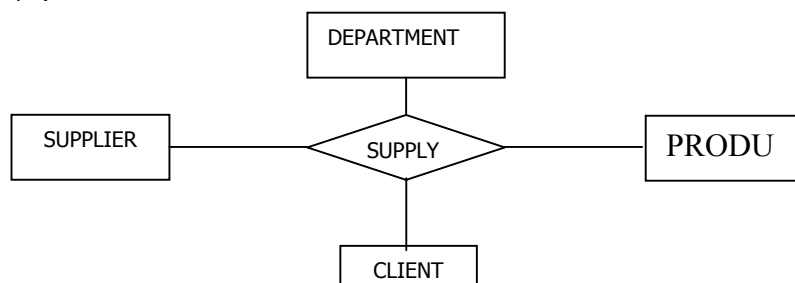
<rdf:Description rdf:about="#id1">
  <reif:supply_prod rdf:Resource="#p1" />
</rdf:Description>

<rdf:Description rdf:about="#id1">
  <reif:supply_prod rdf:Resource="#p2" />
</rdf:Description>

<rdf:Description rdf:about="#id1">
  <reif:supply_prod rdf:Resource="#p3" />
</rdf:Description>
```

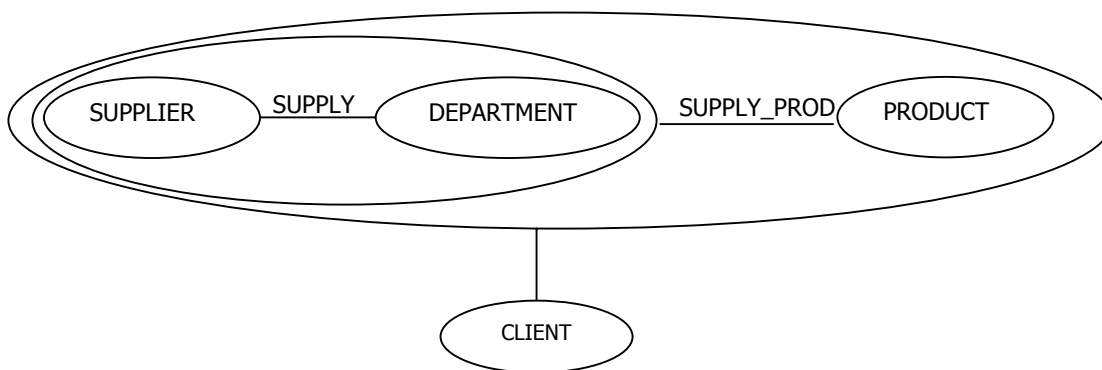
Με τη βοήθεια της υποστασιοποίησης μπορούμε να εκφράσουμε όχι μόνο τριαδικές σχέσεις αλλά και τετραδικές, πενταδικές και γενικεύοντας n -αδικές σχέσεις.

Πράγματι, αν η υποστασιοποίηση εφαρμοστεί αναδρομικά μπορεί να εκφράσει σχέσεις οποιουδήποτε βαθμού. Ας εξετάσουμε την περίπτωση μίας τετραδικής σχέσης. Θεωρούμε μια επέκταση του παραπάνω παραδείγματος όπου εκτός από των άλλων πρέπει να κρατάμε και την πληροφορία ποιος υπάλληλος του προμηθευτή έκανε την προμήθεια.



Σχήμα 6.4 Παράδειγμα τετραδικής σχέσης

Η παραπάνω τετραδική σχέση μπορεί να αναπαρασταθεί με τον RDF γράφο του παρακάτω Σχήματος:



Σχήμα 6.5 Αναπαράσταση τετραδικής σχέσης με υποστασιοποίηση

Επομένως με τη βοήθεια της υποστασιοποίησης μπορούμε να εκφράσουμε σχέσεις οποιουδήποτε βαθμού στην RDF επεκτείνοντας τις δυνατότητες της.

6.2 Επικοινωνία μεταξύ Agents

6.2.1 Εισαγωγή

Ο όρος «agent» - ή «πράκτορας» στα ελληνικά - είναι δύσκολο να οριστεί. Οι πράκτορες συχνά περιγράφονται σαν οντότητες με χαρακτηριστικά που θεωρούνται χρήσιμα για ένα συγκεκριμένο πεδίο. Παίζουν πολύ σημαντικό ρόλο, τόσο στην βελτίωση της εμπειρίας των χρηστών σχετικά με την οργάνωση της πληροφορίας, όσο και στην εκτέλεση “βαρετών” εργασιών ή στην συγχώνευση δεδομένων από πολλές διαφορετικές πηγές.

Στο παρελθόν, τα προγράμματα που συνάθροιζαν δεδομένα από πολλαπλές πηγές, όπως τα προγράμματα συγχώνευσης ηλεκτρονικής αλληλογραφίας ή διαχείρισης σχέσεων πελατών, έπρεπε να μπορούν να μιλούν πολυάριθμα πρωτόκολλα με διαφορετική δομή για να μπορούν να εξάγουν αποτελέσματα. Με μία πλούσια ύλη πληροφοριών η δυνατότητα αυτοματοποίησης γίνεται σημαντική και καλούνται οι πράκτορες που χρησιμοποιούν μία μόνο ενοποιημένη αφαίρεση να λύσουν τέτοιου είδους προβλήματα.

Οι πράκτορες μπορούν να βοηθήσουν τους χρήστες σε περιπτώσεις υπερφόρτωσης πληροφοριών με την εξαγωγή πληροφορίας-κλειδί (key information) από μηνύματα ηλεκτρονικού ταχυδρομείου ή άλλα έγγραφα και να την παρουσιάσουν στον χρήστη με περιλήψεις.

Μία άλλη χρησιμότητα των πρακτόρων είναι η αυτόματη συντήρηση συλλογών. Οι σύγχρονοι αλγόριθμοι ανάκτησης πληροφορίας είναι ικανοί να ομαδοποιούν τα έγγραφα βάση της ομοιότητας τους ή άλλων κριτηρίων (metrics). Προηγούμενη εργασία έχει δείξει ότι αυτή η αυτόματη ταξινόμηση είναι πολύ χρήσιμη σε πολλές περιπτώσεις. Επιπλέον, οι χρήστες μπορούν να φτιάχνουν συλλογές εκτελώντας ένα ερώτημα. Ένας πράκτορας, εξοπλισμένος με τις προδιαγραφές για τις οποίες ψάχνει ο

χρήστης, μπορεί να δημιουργήσει μία συλλογή από τα αποτελέσματα του ερωτήματος και να παρακολουθεί για νέα δεδομένα που εισέρχονται στο σύστημα που ταιριάζουν με το ερώτημα. Για παράδειγμα, οι πράκτορες μπορούν να φιλτράρουν αυτόματα την ηλεκτρονική αλληλογραφία του χρήστη, για έγγραφα που εμφανίζονται να ταιριάζουν με μία ή περισσότερες συλλογές που έχουν καθοριστεί από τον χρήστη, όπως “Website Project” ή “Γράμματα από τη Μαμά”.

Επομένως οι πράκτορες μπορούν να ανακτούν και επεξεργάζονται αυτόματα πληροφορία από διάφορες πηγές, όπως από ηλεκτρονική αλληλογραφία, ημερολόγια, τον Παγκόσμιο Ιστό, κτλ.

6.2.2 Αλληλεπίδραση μεταξύ Πρακτόρων

Οι πράκτορες πάρα πολλές φορές δεν είναι αυτόνομοι, προκειμένου να εξάγουν τις ζητούμενες πληροφορίες πρέπει να αλληλεπιδράσουν με άλλους πράκτορες και να ανταλλάξουν πληροφορίες. Η αλληλεπίδραση είναι ένα από τα πιο σημαντικά χαρακτηριστικά ενός πράκτορα. Οι πράκτορες αλληλεπιδρούν περιοδικά για να μοιραστούν πληροφορία και για να εκτελέσουν ενέργειες προκειμένου να επιτύχουν τους στόχους τους. Αυτό συμβαίνει στα συστήματα Multi-agent, δηλ. στα συστήματα τα οποία αποτελούνται από πολλούς πράκτορες που μπορούν να επικοινωνούν μεταξύ τους.

Απαραίτητη προϋπόθεση για να «επικοινωνήσουν» οι πράκτορες είναι να μιλούν την ίδια «γλώσσα», και επειδή μιλάμε για πράκτορες αυτό μεταφράζεται σε μία κοινή οντολογία. Η οντολογία αυτή μπορεί να εκφραστεί σε είτε σε RDF είτε σε κάποια άλλη γλώσσα που περιγράφει οντολογίες. Η κοινή οντολογία ωστόσο δεν είναι η μόνη προϋπόθεση για να επιτευχθεί η αλληλεπίδραση μεταξύ των πρακτόρων. Συνολικά απαιτούνται τρία βασικά στοιχεία [F01], [Y01] για την αλληλεπίδραση αυτή:

1. Μία κοινή γλώσσα επικοινωνίας πρακτόρων (agent communication language) και ένα πρωτόκολλο
2. Ένα κοινό format για το περιεχόμενο της επικοινωνίας και
3. Μία κοινή οντολογία.

Ας εξετάσουμε αναλυτικότερα καθένα από αυτά.

1. Γλώσσα Επικοινωνίας Πρακτόρων

Η Γλώσσα Επικοινωνίας Πρακτόρων (Agent Communication Language) ή ACL για συντομία, όπως γίνεται αντιληπτό είναι η γλώσσα στην οποία «μιλούν» οι πράκτορες για να ανταλλάξουν πληροφορίες μεταξύ τους, να θέσουν ερωτήματα ο ένας στον άλλο καθώς και να επιτελέσουν ενέργειες που απαιτούνται στα πλαίσια της αλληλεπίδρασης μεταξύ των πρακτόρων.

Γενικά, μία ACL παρέχει στους πράκτορες τα μέσα για την ανταλλαγή πληροφορίας και γνώσης.

Τα βασικά χαρακτηριστικά μίας ACL είναι ότι

- μπορεί να διαχειρίζεται προτάσεις (propositions), κανόνες και πράξεις αντί για απλά αντικείμενα τα οποία δεν έχουν καμία σημασιολογία
- ένα ACL μήνυμα περιγράφει μία επιθυμητή κατάσταση σε μία δηλωτική γλώσσα, αντί για μία διαδικασία ή μέθοδο.

Οι περισσότερο δημοφιλείς ACLs είναι οι KQML και η FIPA ACL.

Εδώ θα εξετάσουμε αναλυτικότερα την KQML. KQML, είναι το ακρώνυμο του Knowledge Query and Manipulation Language [FFM+], που χρησιμοποιείται τόσο σαν ένα message format όσο και σαν πρωτόκολλο διαχείρισης για να υποστηρίξει run-time γνώση που διαμοιράζεται μεταξύ των agents.

Η KQML είναι μία υψηλού επιπέδου, μηνυματο-κεντρική (message-oriented) γλώσσα επικοινωνίας και ένα πρωτόκολλο για την ανταλλαγή πληροφορίας ανεξάρτητα από την σύνταξη του περιεχομένου και την χρησιμοποιούμενη οντολογία. Επομένως είναι ανεξάρτητη του μηχανισμού μεταφοράς (TCP/IP, SMTP), ανεξάρτητη από τη γλώσσα περιεχομένου (content language – KIF, SQL, Prolog) και ανεξάρτητη από την οντολογία του περιεχομένου.

Αυτή η γλώσσα μπορεί να θεωρηθεί ότι αποτελείται από τρία επίπεδα: (1) ένα επίπεδο επικοινωνίας, (2) ένα επίπεδο μηνύματος, και (3) ένα επίπεδο περιεχομένου.

(1) Το **επίπεδο περιεχομένου** περιέχει το ακριβές περιεχόμενο του μηνύματος στην γλώσσα αναπαράστασης του προγράμματος. Η KQML μπορεί να μεταφέρει οποιαδήποτε γλώσσα αναπαράστασης, συμπεριλαμβανομένων τόσο των γλωσσών που εκφράζονται σαν συμβολοσειρές ASCII όσο και αυτών που εκφράζονται με δυαδικούς χαρακτήρες. Κάθε KQML υλοποίηση αγνοεί το τμήμα του περιεχομένου του μηνύματος, εκτός από το ότι αποφασίζει το σημείο στο οποίο τελειώνει.

(2) Το **επίπεδο επικοινωνίας** κωδικοποιεί ένα σύνολο από χαρακτηριστικά στο μήνυμα που περιγράφουν τις παραμέτρους επικοινωνίας σε ένα χαμηλότερο επίπεδο, όπως την ταυτότητα του αποστολέα και του παραλήπτη, και ένα μοναδικό αναγνωριστικό που σχετίζεται με την επικοινωνία.

(3) Το **επίπεδο μηνύματος**, το οποίο κωδικοποιεί ένα μήνυμα το οποίο κάποια εφαρμογή επιθυμεί να μεταβιβάσει σε κάποια άλλη, και το οποίο αποτελεί το βασικό τμήμα της KQML. Αυτό το επίπεδο αποφασίζει τα είδη των αλληλεπιδράσεων που μπορεί να έχει κάποιος με έναν agent που «μιλάει» σε KQML. Η βασική λειτουργία του επιπέδου αυτού είναι να αναγνωρίζει το πρωτόκολλο δικτύου με το οποίο παραδίδει το μήνυμα και να παρέχει μία πράξη λόγου (speech act) ή performative το οποίο ο αποστολέας αποδίδει στο περιεχόμενο. Αυτή η πράξη του λόγου (speech act) δηλώνει αν το μήνυμα είναι ισχυρισμός, ερώτηση, εντολή ή κάποιο άλλο το σύνολο των performatives. Επιπλέον, αφού το περιεχόμενο είναι αδιαφανές στην KQML, το επίπεδο μηνύματος περιλαμβάνει επίσης κάποια προαιρετικά χαρακτηριστικά που περιγράφουν την γλώσσα περιεχομένου, την οντολογία που χρησιμοποιείται και μερικούς τύπους περιγραφής του περιεχομένου, όπως ένα περιγραφητή που καθορίζει ένα θέμα στην οντολογία. Αυτά τα χαρακτηριστικά δίνουν τη

δυνατότητα στις εφαρμογές της KQML να αναλύουν, να δρομολογούν και να παραδίδουν σωστά μηνύματα των οποίων το περιεχόμενο είναι απρόσιτο.

Για να γίνουν κατανοητά όσα αναφέρθηκαν παραπάνω παρατίθεται το format ενός KQML μηνύματος.

```
(register
  :sender          agentA
  :receiver        agentB
  :reply-with      message2
  :language        common_language
  :ontology        common_ontology
  :content         `(ServiceProvision
Manufacturing:TaskDecomposition) ' '
)
```

Σχήμα 6.6. Παράδειγμα KQML μηνύματος

Το μήνυμα όπως παρατηρούμε ξεκινά με την λέξη ``register'', η οποία είναι η πράξη που σκοπεύει το μήνυμα. Το υπόλοιπο μήνυμα περιέχει λέξεις-κλειδιά που χρειάζονται για το μήνυμα και τα επίπεδα επικοινωνίας. Οι λέξεις-κλειδιά που χρησιμοποιούνται στα μηνύματα της KQML ορίζονται ως εξής:

- **sender**: ο πράκτορας που στέλνει το μήνυμα.
- **receiver**: ο πράκτορας που λαμβάνει το μήνυμα.
- **from**: ο γνήσιος αποστολέας, χρησιμοποιείται όταν ένα μήνυμα στέλνεται χρησιμοποιώντας ενδιάμεσους πράκτορες.
- **to**: τελικός παραλήπτης, χρησιμοποιείται όταν ένα μήνυμα στέλνεται χρησιμοποιώντας ενδιάμεσους πράκτορες.
- **in-reply-to**: αναγνωριστικό του μηνύματος που προκάλεσε την αποστολή του μηνύματος που περιέχει την απάντηση.
- **reply-with**: αναγνωριστικό που θα χρησιμοποιηθεί από το μήνυμα που θα απαντήσει στο μήνυμα αυτό.
- **language**: γλώσσα για τη διερμηνευση της πληροφορίας που βρίσκεται στο πεδίο content αυτού του μηνύματος.
- **ontology**: αναγνωρίζει την οντολογία για να διερμηνεύσει την πληροφορία που βρίσκεται στο πεδίο content του μηνύματος.
- **content**: πληροφορία context-specific που περιγράφει τις ιδιαιτερότητες του μηνύματος.

2. Τύπος περιεχομένου επικοινωνίας (meta-knowledge)

Ο τύπος (format) του περιεχομένου επικοινωνίας ή πιο κοινά η μετα-γνώση μπορεί να αναπαρασταθεί με διάφορες γλώσσες. Το RDF μπορεί να αντεπεξέλθει στις ανάγκες αναπαράστασης της μεταγνώσης όπως θα δούμε παρακάτω. Μια άλλη γλώσσα που είναι η πλέον χρησιμοποιούμενη στους πράκτορες για αναπαράσταση περιεχομένου επικοινωνίας είναι η KIF.

Παρακάτω θα εξεταστούν δύο σενάρια επικοινωνίας μεταξύ πρακτόρων και θα εξεταστούν οι γλώσσες RDF και KIF.

3. Κοινή Οντολογία

Μία οντολογία είναι μία σαφής παρουσίαση της αντίληψης [Grubber, 1993]. Ο όρος αυτός έχει δανειστεί από την φιλοσοφία, όπου μία Οντολογία είναι μία συστηματική περιγραφή της ύπαρξης.

Για τα συστήματα Τεχνητής Νοημοσύνης, ότι «υπάρχει» είναι αυτό που μπορεί να αναπαρασταθεί. Όταν η γνώση ενός πεδίου αναπαριστάται με ένα δηλωτικό τυπικό τρόπο, το σύνολο των αντικειμένων που μπορούν να αναπαρασταθούν ονομάζεται τομέας σύμπαν της ομιλίας (universe of discourse). Αυτό το σύνολο των αντικειμένων και οι περιγραφόμενες σχέσεις μεταξύ αυτών, απεικονίζονται σε ένα τυπικό λεξικό με το οποίο ένα πρόγραμμα βασισμένο στη γνώση μπορεί να αναπαραστήσει την γνώση. Επομένως, στο πεδίο της Τεχνητής Νοημοσύνης, η οντολογία μπορεί να περιγραφεί σαν ένα πρόγραμμα ορίζοντας ένα σύνολο όρων αναπαράστασης. Σε μία τέτοια οντολογία, οι ορισμοί συσχετίζουν τα ονόματα των οντοτήτων στο σύμπαν της ομιλίας (universe of discourse) (π.χ. classes, relations, functions ή άλλα αντικείμενα) με κείμενο, που μπορεί να διαβαστεί από τον άνθρωπο, το οποίο περιγράφει τι σημαίνουν τα ονόματα και τα τυπικά αξιώματα που περιορίζουν την διερμηνεία και την σωστή χρήση αυτών των όρων. Τυπικά, μία οντολογία είναι μία πρόταση μίας θεωρίας λογικής [ONT].

Χρησιμοποιούμε κοινές οντολογίες προκειμένου να περιγράψουμε οντολογικές δεσμεύσεις για ένα σύνολο από πράκτορες έτσι ώστε να μπορούν να επικοινωνούν μεταξύ τους σε ένα πεδίο ομιλίας (discourse) χωρίς απαραίτητα να λειτουργούν πάνω σε μια γενικά διαμοιραζόμενη θεωρία. Λέμε ότι ένας πράκτορας **δεσμεύεται (commits)** σε μία οντολογία αν οι ενέργειες του είναι συνεπείς με τους ορισμούς της οντολογίας. Η ιδέα των οντολογικών δεσμεύσεων βασίζεται στην οπτική του Επιπέδου Γνώσεων (Knowledge Level perspective) (Newell, 1982). Το Επίπεδο Γνώσης είναι ένα επίπεδο περιγραφής της γνώσης ενός πράκτορα το οποίο είναι ανεξάρτητο από την αναπαράσταση σε επίπεδο συμβόλων που χρησιμοποιείται εσωτερικά από τον πράκτορα.

Η γνώση αποδίδεται στους πράκτορες παρακολουθώντας τις ενέργειες τους. Ένας πράκτορας “γνωρίζει” κάτι αν ενεργεί σαν να είχε την πληροφορία και ενεργεί λογικά για να επιτύχει τους στόχους του. Οι «ενέργειες» των πράκτορων – συμπεριλαμβανομένων των εξυπηρετητών βάσεων γνώσης και των συστημάτων βάσεων γνώσης – μπορούν να ιδωθούν μέσω ενός tell και ask λειτουργικού interface (Levesque, 1984) όπου ένας πελάτης αλληλεπιδρά με ένα πράκτορα κάνοντας λογικούς ισχυρισμούς (tell) και θέτοντας ερωτήματα (queries).

Στην πραγματικότητα, μία κοινή οντολογία ορίζει το λεξιλόγιο με το οποίο ανταλλάσσονται ερωτήματα και ισχυρισμοί μεταξύ των πρακτόρων. Οι οντολογικές δεσμεύσεις είναι συμφωνίες να χρησιμοποιούν το λεξιλόγιο που διαμοιράζονται, με ένα λογικά συναφή και συνεπή τρόπο. Οι πράκτορες που χρησιμοποιούν το ίδιο λεξιλόγιο δεν χρειάζεται να έχουν την ίδια βάση γνώσεων, καθένας ξέρει πράγματα που δεν τα ξέρει ο άλλος, και ένας πράκτορας που δεσμεύεται σε μία οντολογία δεν είναι υποχρεωμένος να απαντάει όλα τα ερωτήματα που μπορούν να σχηματιστούν από το λεξικό που διαμοιράζονται.

Συνεπώς, προκειμένου να μπορεί να υπάρξει επικοινωνία μεταξύ των πρακτόρων πρέπει να διαμοιράζονται μία κοινή οντολογία. Μία δέσμευση για κοινή οντολογία είναι μία εγγύηση συνέπειας, αλλά όχι πληρότητας, σχετικά με τα ερωτήματα και τους ισχυρισμούς, χρησιμοποιώντας το λεξιλόγιο που ορίζεται στην οντολογία.

Οι οντολογίες ορίζονται σαν σχήματα προδιαγραφών (specification schemes) για την περιγραφή εννοιών και σχέσεων μεταξύ αυτών στο πεδίο μιας συνομιλίας. Είναι σημαντικό ότι οι πράκτορες όχι μόνο έχουν οντολογίες για να αντιλαμβάνονται ένα πεδίο, αλλά επίσης έχουν οντολογίες με παρόμοιες δομές. Τέτοιες οντολογίες, όταν υπάρχουν, ονομάζονται **κοινές οντολογίες (common ontologies)**.

Όταν οι agents που αλληλεπιδρούν έχουν δεσμευτεί σε μία κοινή οντολογία, αναμένεται ότι θα χρησιμοποιήσουν αυτή την οντολογία για να διερμηνεύσουν την μεταξύ τους επικοινωνία. Με τον τρόπο αυτόν οδηγούνται σε αμοιβαία κατανόηση και σε προβλέψιμες συμπεριφορές.

Η **Ontolingua** αναφέρεται συχνά στην βιβλιογραφία σαν ένα σύστημα που παρέχει ένα λεξιλόγιο για τον ορισμό επαναχρησιμοποιήσιμων, φορητών και διαμοιραζόμενων οντολογιών. Οι ορισμοί της Ontolingua περιγράφονται χρησιμοποιώντας συντακτικό και σημασιολογία παρόμοια με αυτή της **Knowledge Interchange Format** γνωστή και ως KIF, που είναι ένα format για την τυποποίηση σχημάτων αναπαράστασης γνώσης βασισμένο στη λογική πρώτης τάξης.

Οι οντολογίες σαν ένας μηχανισμός προδιαγραφών

Ένα σύνολο γνώσης που έχει αναπαρασταθεί τυπικά βασίζεται σε μία *αντίληψη (conceptualization)*: τα αντικείμενα, τις έννοιες, και τις άλλες οντότητες που υποθέτουμε ότι υπάρχουν σε μία περιοχή ενδιαφέροντος και τις σχέσεις που υπάρχουν μεταξύ αυτών (Genesereth & Nilsson, 1987). Μία αντίληψη είναι μία αφηρημένη, απλοποιημένη άποψη του κόσμου που επιθυμούμε να αναπαραστήσουμε για κάποιο σκοπό. Κάθε βάση γνώσεων, σύστημα βασισμένο στη γνώση, ή πράκτορα που λειτουργεί σε επίπεδο γνώσης δεσμεύεται με κάποια αντίληψη (conceptualization), άμεσα ή έμμεσα.

Προδιαγραφές Συστήματος Πολλαπλών Πρακτόρων

Έστω ότι έχουμε πράκτορες που θέλουν να επικοινωνούν μεταξύ τους και να ανταλλάζουν πληροφορίες.

Οι πράκτορες “τρέχουν” σε συστήματα τα οποία χρησιμοποιούν RDF βάσεις δεδομένων τόσο για τον ορισμό της ιεραρχίας των κλάσεων και των ιδιοτήτων καθώς και για τον καθορισμό των περιορισμών όσον αφορά στο πεδίο ορισμού και το σύνολο τιμών των ιδιοτήτων.

Οι πράκτορες θέλουν να κάνουν ερωτήσεις και να παίρνουν απαντήσεις τόσο από άλλους πράκτορες όσο και από την δική τους βάση δεδομένων. Για το λόγο αυτό πρέπει να χρησιμοποιήσουμε μία απλή γλώσσα επερωτήσεων για το RDF, την λεγόμενη RQL.

Προκειμένου να επιτευχθεί η επικοινωνία μεταξύ των πρακτόρων πρέπει προηγουμένως να υπάρξει συμφωνία μεταξύ αυτών όσον αφορά στα

1. πρωτόκολλο για την επικοινωνία της μεταγνώσης (KQML, FIPA) ή Agent Communication Language (ACL)
2. τύπος μεταγνώσης (KIF, MCF, RDF)
3. πρωτόκολλο για την μεταφορά δεδομένων (HTTP, FTP, ODBC) και τέλος
4. τύπος επερωτήσεων στα δεδομένα (SQL, OQL, RQL).

Στην περίπτωση μας για την επικοινωνία μεταξύ των πρακτόρων θα χρησιμοποιηθούν τα παρακάτω:

1. KQML
2. RDF
3. HTTP
4. RQL

1. Ως πρωτόκολλο για την επικοινωνία της μεταγνώσης θα χρησιμοποιηθεί η KQML.

Παρακάτω παρατίθενται κάποια παραδείγματα μηνυμάτων σε KQML:

```
(ask-one
  :sender      peter
  :content    ( select P
               from {X}first_name{Y}, {W}paints{P}
               where Y = "Pablo" and X=W
  :receiver   paul
  :reply-with Museum-info
  :language   RDF
  :ontology   MUSEUM)

(tell
  :sender      paul
  :content    (<RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
syntax-ns#">
              <rdf:Bag>
                <rdf:li rdf:type="resource"
rdf:resource="http://www.museum.es/woman.qti" />
              </rdf:Bag>
            </RDF>
  )
  :receiver   peter
  :in-reply-to Museum-info
  :language   RDF
  :ontology   MUSEUM)
```

Για την περίπτωση μας θα χρησιμοποιούμε μόνο ερωτήματα των δύο παραπάνω τύπων, δηλ. ερώτηση (ask) και ισχυρισμός ή απάντηση (tell).

Στην περίπτωση που έχουμε ερώτηση (ask) το περιεχόμενο του content θα είναι ένα KIF ή μια RQL επερωτήσ, ενώ όταν έχουμε ισχυρισμό (tell) το content θα είναι γραμμένο σε KIF ή RDF.

2. Όσον αφορά στον τύπο της μεταγνώσης θα εξετάσουμε δύο περιπτώσεις

- το RDF

- την KIF (Knowledge Interchange Format), η οποία είναι μία ευρέως χρησιμοποιούμενη γλώσσα.
3. Το πρωτόκολλο μεταφοράς δεδομένων που θα χρησιμοποιηθεί είναι το HTTP.
 4. Και τέλος οι επερωτήσεις θα γίνονται στην γλώσσα RQL, η οποία αποτελεί μία γλώσσα επερώτησης του RDF.

Έστω ότι έχουμε δύο πράκτορες, τους Agent1 και Agent2, οι οποίοι ανταλλάσσουν πληροφορίες μεταξύ τους.

Θα εξετάσουμε δύο σενάρια: στην πρώτη περίπτωση το format της μετα-γνώσης θα είναι σε RDF και στην δεύτερη περίπτωση το format της μετα-γνώσης θα είναι σε KIF.

1^η Περίπτωση: Μεταγνώση σε RDF

Στην περίπτωση αυτή η Content Language που χρησιμοποιεί η ACL είναι το RDF. Προκειμένου να είναι εφικτή και αποτελεσματική η επικοινωνία μεταξύ των agent πρέπει να διαμοιράζονται μία κοινή οντολογία πάνω στην οποία θα ανταλλάζουν πληροφορίες. Για το σκοπό αυτό θα θεωρήσουμε ότι υπάρχει μια κοινή οντολογία ορισμένη σε RDF.

Επομένως, οι πράκτορες θα έχουν ένα κοινό σχήμα και θα μπορούν να ανταλλάσσουν πληροφορίες για τα δεδομένα του σχήματος.

Θα εξετάσουμε δύο περιπτώσεις:

1. Ο agent ρωτάει ένα συγκεκριμένο agent
2. Ο agent ρωτάει όλους τους agents του συστήματος, δηλ. κάνει broadcast ένα μήνυμα.

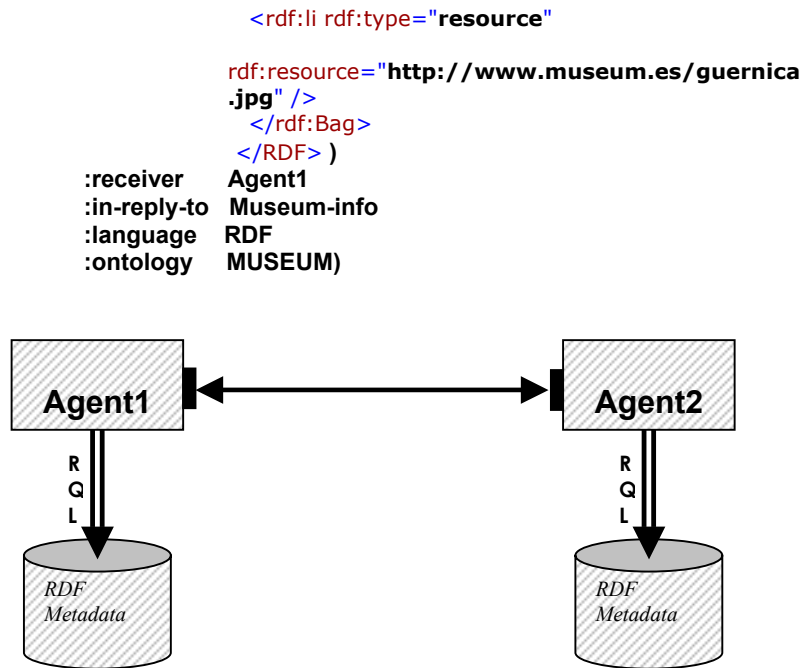
1. Ο agent ρωτάει ένα συγκεκριμένο agent

Έστω ότι ο Agent1 ρωτάει τον Agent2 «ποιοι πίνακες έχουν ζωγραφιστεί από ένα ζωγράφο που έχει μικρό όνομα Pablo».

```
(ask-one
  :sender Agent1
  :content ( select P
             from {X}paints{P}
             where X = "Picasso" )
  :receiver Agent2
  :reply-with Museum-info
  :language RDF
  :ontology MUSEUM)
```

Ο Agent2 απαντάει τον Agent1:

```
(tell
  :sender Agent2
  :content (<RDF xmlns:rdf="http://www.w3.org/1999/02/22-
rdf-syntax-ns#">
           <rdf:Bag>
```



Σχήμα 6.7. Επικοινωνία μεταξύ Πρακτόρων

Το πρόβλημα που προκύπτει στην περίπτωση αυτή είναι ότι θέλουμε να αποθηκεύσουμε στη Βάση1 ότι τα έργα που έχουν ζωγραφιστεί από κάποιον που ονομάζεται Picasso είναι η Guernica σύμφωνα με τον Agent2. Αλλά αυτή η πληροφορία αυτή δεν μπορεί να αποθηκευτεί με τις υπάρχουσες δυνατότητες της RDF. Προκειμένου να αποθηκευτεί η παραπάνω πληροφορία πρέπει η τριπλέτα <Picasso – paints - Guernica> να υποστασιοποιηθεί, δηλ. να αποδοθεί σε αυτήν ένα αναγνωριστικό. Στην συνέχεια το αναγνωριστικό αυτό θα χρησιμοποιηθεί σαν υποκείμενο (ή αντικείμενο) μίας ιδιότητας π.χ της «assumes» η οποία θα αναφέρει ποιος ισχυρίζεται την πρόταση.

Έστω ότι ο Agent1 λαμβάνει την πληροφορία <Picasso – panits – Guernica > από τον Agent2.

Η πληροφορία αυτή δεν μπορεί να αποθηκευτεί ως έχει στη βάση γνώσεων του Agent1. Αρχικά εισάγεται η τριπλέτα <Picasso – panits – Guernica > στην Βαση1. Στην συνέχεια αποδίδεται στην τριπλέτα αυτή ένα αναγνωριστικό έστω το id1 και τέλος εισάγεται στην Βάση1 μία επιπλέον τριπλέτα η <Agent2 – assumes – id1>.

Για τη σχέση Paints

id	s	o
id1	Picasso	Guernica

Για τη σχέση assumes

Agent	TripID
Agent2	id1

2. Ο agent ρωτάει όλους τους agents του συστήματος, δηλ. κάνει broadcast ένα μήνυμα

Έστω ότι ο Agent1 στέλνει προς όλους τους πράκτορες το ερώτημα «ποιοι πίνακες έχουν ζωγραφιστεί από ένα ζωγράφο που έχει μικρό όνομα Pablo».

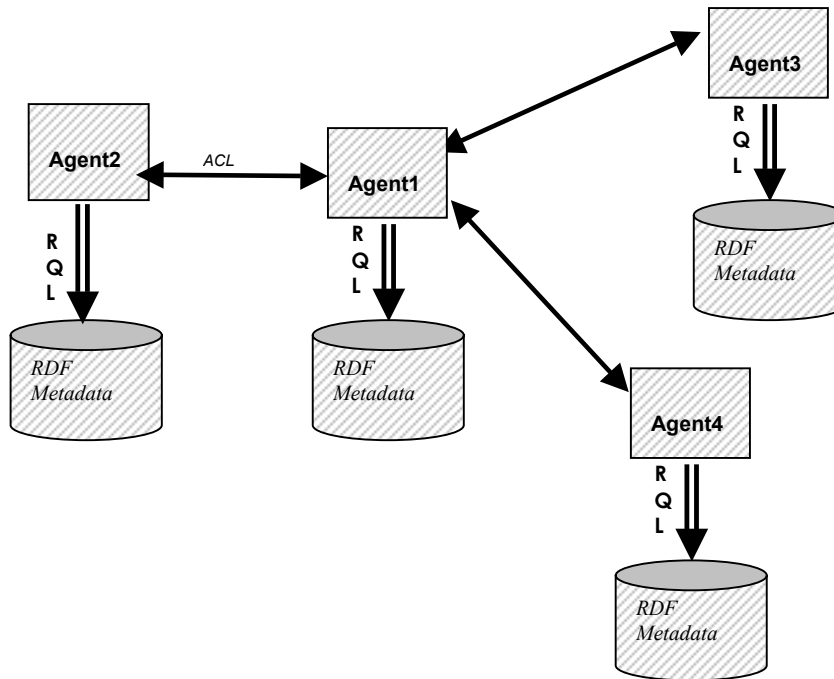
```
(ask-all
  :sender Agent1
  :content ( select P
            from {X}paints{P}
            where X = "Picasso" )
  :reply-with Museum-info
  :language RDF
  :ontology MUSEUM)
```

Στο ερώτημα του Agent1 απαντούν οι Agent2, Agent3 και Agent4.

```
(tell
  :sender Agent2
  :content (<RDF xmlns:rdf="http://www.w3.org/1999/02/22-
rdf-syntax-ns#">
          <rdf:Bag>
            <rdf:li rdf:type="resource"
rdf:resource="http://www.museum.es/guernica
.jpg" />
          </rdf:Bag>
        </RDF> )
  :receiver Agent1
  :in-reply-to Museum-info
  :language RDF
  :ontology MUSEUM)
```

```
(tell
  :sender Agent3
  :content (<RDF xmlns:rdf="http://www.w3.org/1999/02/22-
rdf-syntax-ns#">
          <rdf:Bag>
            <rdf:li rdf:type="resource"
rdf:resource="http://www.museum.es/toledo.jp
g" />
          </rdf:Bag>
        </RDF> )
  :receiver Agent1
  :in-reply-to Museum-info
  :language RDF
  :ontology MUSEUM)
```

```
(tell
  :sender Agent4
  :content (<RDF xmlns:rdf="http://www.w3.org/1999/02/22-
rdf-syntax-ns#">
          <rdf:Bag>
            <rdf:li rdf:type="resource"
rdf:resource="http://www.museum.es/woman.j
pg" />
          </rdf:Bag>
        </RDF> )
  :receiver Agent1
  :in-reply-to Museum-info
  :language RDF
  :ontology MUSEUM)
```



Σχήμα 6.8. Παράδειγμα Πολλαπλών Πρακτόρων

Και στην περίπτωση αυτή στην Βάση1 πρέπει να αποθηκευτούν τόσο οι πληροφορίες του ποιοι πίνακες ζωγραφίστηκαν από τον Picasso όσο και ποιος πράκτορας ισχυρίζεται την κάθε πληροφορία. Απαραίτητη προϋπόθεση για να μην υπάρξει απώλεια πληροφορίας είναι η χρήση της υποστασιοποίησης. Η διαδικασία που ακολουθούμε είναι η ίδια με την προηγούμενη περίπτωση.

Για τη σχέση Paints έχουμε

id	s	o
id1	Picasso	Guernica
id3	Picasso	Toledo
id2	Picasso	Woman

Για τη σχέση assumes

Agent	TripID
Agent2	id1
Agent3	id2
Agent4	id3

Επίσης υποστασιοποίηση θα μπορούσε να οριστεί και σε επίπεδο agent. Αυτό σημαίνει ότι κάθε agent στη βάση γνώσεων του υποστασιοποιεί όλες όχι μόνο τις

πληροφορίες που προέρχονται από άλλες πηγές, αλλά και αυτές τις οποίες γνωρίζει από μόνος του.

Άρα, αν ρωτήσουμε τον agent1 «τι γνωρίζεις εσύ για τους πίνακες του Picasso;» θα μας επιστρέψει τις πληροφορίες που έχουν εισαχθεί στην βάση γνώσεων αυτού και δεν προέρχονται από άλλες πηγές όπως για παράδειγμα από τους agent2, agent3 και agent4.

Προκειμένου να είμαστε σε θέση να εξάγουμε γρήγορα την παραπάνω πληροφορία πρέπει να υποστασιοποιήσουμε και τις γνώσεις που έχει ο ίδιος ο Agent. Προκειμένου να γίνει αυτό πρέπει να αποδώσουμε σε κάθε τριάδα - που συνιστά το συστατικό στοιχείο της γνώσης - ένα μοναδικό αναγνωριστικό. Στη συνέχεια συσχετίζουμε το μοναδικό αυτό αναγνωριστικό με τον Agent1, με τον ίδιο ακριβώς τρόπο που είχε γίνει η συσχέτιση παραπάνω για τους Agent2, Agent3 και Agent4.

Ας δούμε ένα παράδειγμα προκειμένου να γίνουν περισσότερο κατανοητά όσα αναφέρθηκαν παραπάνω.

Έστω ότι ο Agent1 έχει στην βάση γνώσεων του την πληροφορία ότι ο Picasso ζωγράφισε τους πίνακες Spain και Toledo. Επίσης, έχει ανακτήσει την πληροφορία από τους Agent2, Agent3 και Agent4 ότι ο Picasso έχει ζωγραφίσει επίσης τους πίνακες Guernica, Toledo και Woman αντίστοιχα.

Σε όλες αυτές τις πληροφορίες αποδίδεται ένα μοναδικό αναγνωριστικό, όπως φαίνεται παρακάτω στον παρακάτω πίνακα.

Επομένως, για τη σχέση Paints έχουμε

id	s	o
id1	Picasso	Guernica
id3	Picasso	Toledo
id2	Picasso	Woman
id4	Picasso	Spain

Ενώ στην σχέση assumes προσθέτονται δύο επιπλέον εγγραφές:

Agent	TripID
Agent2	id1
Agent3	id2
Agent4	id3
Agent1	id3
Agent1	id4

Όταν ο agent1 ερωτηθεί τι γνωρίζει ο ίδιος για τους πίνακες που έχει ζωγραφίσει ο Picasso, η απάντηση που θα δώσει θα είναι μόνο οι **Toledo** και **Spain**.

2^η Περίπτωση: Μεταγνώση σε KIF

Στην περίπτωση αυτή η Context Language που χρησιμοποιείται είναι η KIF. Θεωρούμε ένα mediator μεταξύ του Agent και των RDF Metadata που μετατρέπει την πληροφορία που υπάρχει σε KIF σε RDF δεδομένα.

Στην περίπτωση αυτή προκύπτει το εξής πρόβλημα: η KIF έχει πολύ μεγαλύτερη εκφραστικότητα σε σχέση με το RDF.

Συγκεκριμένα, ενώ η KIF υποστηρίζει την υποστασιοποίηση σε αντίθεση με το RDF. Για παράδειγμα η έκφραση “Steve Beliefs X” όπου X μία πρόταση έχει την εξής αναπαράσταση σε KIF :

```
(hold true  
(has-belief X steve))
```

```
(believes steve  
'(material moon stilton))
```

ενώ δεν μπορεί να αναπαρασταθεί με την υπάρχουσα εκφραστικότητα σε RDF.

Όπως μπορεί να γίνει αντιληπτό έχουμε απώλεια πληροφορίας. Για το λόγο αυτό είναι αναγκαίο να εμπλουτίσουμε το RDF.

Βέβαια, θα μπορούσε να ισχυριστεί κάποιος ότι δεν είναι απαραίτητος ο εμπλουτισμός του RDF καθώς στην περίπτωση που έχουμε απευθείας επικοινωνία μεταξύ των πρακτόρων σχετικά με τις οντολογίες η χρήση του RDF σαν content language δεν δημιουργεί πρόβλημα. (Εκτός του ότι δεν μπορούμε να κρατάμε στη RDF βάση μας την πληροφορία ποιος είπε τι και δεν κάνουμε διάκριση μεταξύ των πληροφοριών που λαμβάνουμε από τους agent και αυτών που είχαμε από πριν στη βάση μας).

Αλλά, αν θέλουμε να χρησιμοποιηθεί το RDF σε πραγματικές εφαρμογές multi-agent συστημάτων πρέπει να υποστηρίζει όλα εκείνα που υποστηρίζουν και οι υπόλοιπες context languages.

Μάλιστα προς το παρόν το RDF, λόγω της ελλιπής εκφραστικότητας της, αποφεύγεται να χρησιμοποιείται σαν content language.

Χρήσης Υποστασιοποίησης σε θέματα πίστης των πρακτόρων

Όταν χρησιμοποιούνται πολλαπλοί πράκτορες για να παράγουν την ίδια πληροφορία ανακύπτουν θέματα όπως πως να αντιμετωπιστούν οι συγκρούσεις. Για παράδειγμα, αν ένας πράκτορας είναι επιφορτισμένος με την αποστολή να αποφασίζει την ημέρα παράδοσης ενός εγγράφου χρησιμοποιώντας επεξεργασία σε φυσική γλώσσα και ένας άλλος πράκτορας κάνει το ίδιο αλλά εξάγοντας την first date του εγγράφου, ποια από τις δύο θα πιστέψουμε αν δεν είναι ίδιες; Σε περιπτώσεις όπως αυτή, είναι σημαντικό η πληροφορία να περιέχει ετικέτες με συγγραφικά μετα-δεδομένα έτσι ώστε ο χρήστης να μπορεί να διαλέξει τη σωστή πρόταση.

Για να επιτύχουμε το παραπάνω, συζητάμε ένα μέρος της οντολογίας του συστήματος που χρησιμοποιείται για να περιγράψει τα χαρακτηριστικά των ίδιων των πραγματικών προτάσεων, όπως για παράδειγμα ποιος τις ισχυρίστηκε και πότε τις ισχυρίστηκε.

Δεδομένου ότι απαιτούνται μόνο τρεις τιμές, μία για το subject, μία για το object και μία για το predicate, για να περιγραφούν οι προτάσεις στο μοντέλο μας, είναι δυνατόν

να δοθεί ένα αναγνωριστικό σε κάθε πρόταση και στη συνέχεια να γίνει ένας ισχυρισμός για τον συγγραφέα και για την ώρα δημιουργίας της αρχικής πρότασης.

Στην πραγματικότητα, το μοντέλο του RDF καθορίζει ότι προκειμένου να δημιουργηθούν προτάσεις για προτάσεις, η πρόταση για την οποία γίνονται οι αναφορές πρέπει να υποστασιοποιηθεί σε ένα πόρο και να ανατεθεί σε αυτήν ένα URI, και οι προτάσεις που αναφέρονται σε αυτήν μπορούν να χρησιμοποιήσουν τον υποστασιοποιημένο πόρο σαν υποκείμενο ή σαν αντικείμενο. Με αυτή τη χρήση της υποστασιοποίησης μας ανακύπτει ένα «ευφύες» θέμα που αφορά στο RDF. Σε ένα RDF έγγραφο, υποτίθεται ότι όλες οι προτάσεις που εισάγονται από τον συγγραφέα είναι αληθείς. Προκειμένου να φτιάξουμε μία πρόταση σχετικά με μία άλλη πρόταση που ο συγγραφέας δεν πιστεύει απαραίτητα ότι είναι αληθής, η πρόταση στην οποία θα αναφερθούμε πρέπει να είναι υποχρεωτικά σε υποστασιοποιημένη μορφή. Επομένως, ο συγγραφέας δεσμεύει ένα όνομα σε μία συγκεκριμένη πρόταση με ένα συγκεκριμένο υποκείμενο, αντικείμενο και κατηγορούμενο, αλλά δεν κάνει τον ισχυρισμό ότι η πρόταση είναι αληθής, το μόνο που κάνει είναι ότι ισχυρίζεται κάποιες άλλες ιδιότητες σχετικά με την πρόταση που χρησιμοποιεί αυτό το όνομα.

Η διατήρηση διαφόρων επιπέδων αξιοπιστίας είναι σημαντικό σε ένα σύστημα που περιέχει προτάσεις που γίνονται από αναρίθμητους ανεξάρτητους agents, καθώς και πληροφορίες από συναδέλφους, φίλους και πελάτες του χρήστη. Προκειμένου να διατηρήσουμε μεταδεδομένα για τις ίδιες τις προτάσεις σε μία αποθήκη πληροφοριών, μία λύση είναι να γίνει η αποθήκη πληροφοριών ένα «ουδέτερο μέλος», που θα καταγράφει ποιος είτε τι και πότε ειπώθηκαν αυτά, αλλά χωρίς να γίνονται ισχυρισμοί για την αλήθεια αυτών. Αυτό μπορεί να επιτευχθεί με το να γίνονται όλες οι προτάσεις από μέλη άλλα από αυτά στα οποία η βάση πληροφοριών έχει υποστασιοποιήσει την πληροφορία. Από τη στιγμή που έχουμε ένα σύστημα για την καταγραφή μεταδεδομένων για προτάσεις, μπορούν να εξεταστούν θέματα ανάκλησης, άρνησης και λήξης των ισχυρισμών, π.χ. προτάσεις που έχουν ισχυριστεί από συγκεκριμένα μέλη. Θεωρήστε για παράδειγμα ένα πράκτορα που είναι υπεύθυνος για την παραγωγή του τίτλου ιδιοκτησίας για ιστοσελίδες. Μερικές ιστοσελίδες, όπως αυτές που τα περιεχόμενα τους ενημερώνονται καθημερινά, έχουν τίτλους που αλλάζουν συνεχώς. Συχνά οι χρήστες θέλουν να μπορούν να βρύνουν σελίδες βασισμένοι στο τι θυμούνται σχετικά με τη σελίδα. Μία προσέγγιση για την διαχείριση των συνεχών μεταβολών της πληροφορίας, είναι να επιτρέπεται στους agent να διαγράφουν τον προηγούμενο ισχυρισμό και να τον αντικαθιστούν με μία ενημερωμένη έκδοση. Παρόλα αυτά θα ήταν πολύ αποτελεσματικό να μπορεί ο χρήστης να θέτει ερωτήματα της μορφής, «Δείξε μου όλες τις ιστοσελίδες που είχαν τίτλο *Tips for Maintaining Your Car* κάποια χρονική στιγμή στο παρελθόν». Με το να επιτρέπεται στους χρήστες να ανακτούν τέτοιες προτάσεις, μπορούν να θέτουν ερωτήματα για την ανάκτηση παρελθούσας ή παλιάς (obsolete) πληροφορίας επειδή αυτή η πληροφορία δεν έχει διαγραφεί.

Επιπλέον αυτό το σύστημα επιτρέπει στους χρήστες να αγνοούν ένα ισχυρισμό που γίνεται από ένα πράκτορα με το να αρνούνται τον ισχυρισμό, ενώ διατηρεί τον αρνούμενο ισχυρισμό για μελλοντική χρήση.

ΚΕΦΑΛΑΙΟ 7

Επίλογος

Στην εργασία αυτή παρουσιάσαμε την υποστήριξη της υποστασιοποίησης στο RDF/S. Αρχικά, παρουσιάστηκε η έννοια της υποστασιοποίησης και περιγράφηκαν οι λόγοι για τους οποίους θα ήταν χρήσιμη η υποστήριξη της υποστασιοποίησης σε εφαρμογές RDF. Προκειμένου να εκφραστεί η έννοια της υποστασιοποίησης στο RDF ορίσαμε την έννοια του υπεργράφου, ο οποίος σε αντίθεση με τον απλό κατευθυνόμενο γράφο μπορεί να αναπαραστήσει τις υποστασιοποιημένες προτάσεις με απλούστερο τρόπο. Με βάση το μοντέλο των υπεργράφων επαναπροσδιορίσαμε τους ορισμούς του RDF σχήματος, ώστε να ανταποκρίνονται στις υπάρχουσες ανάγκες για υποστήριξη της υποστασιοποίησης. Επίσης, προτείναμε την τροποποίηση της σχεσιακής αναπαράστασης των RDF μεταδεδομένων, ώστε να αποθηκεύονται σε αυτήν και οι τριάδες οι οποίες αναπαριστούν τις υποστασιοποιημένες δηλώσεις. Για το σκοπό αυτό έγιναν αλλαγές τόσο στους πίνακες της σχεσιακής βάσης μεταδεδομένων σε επίπεδο μετασχήματος, όσο και σε επίπεδο δεδομένων. Τέλος, σε μια προσπάθεια σφαιρικής υποστήριξης της υποστασιοποίησης στο μοντέλο του RDF, παρουσιάστηκε η δηλωτική γλώσσα επερώτησης RQL, η οποία είναι κατάλληλη για επερώτηση μεταδεδομένων και σχημάτων RDF για πύλες κοινοτήτων διαδικτύου. Στη συνέχεια, προτείναμε τον εμπλουτισμό του βασικού συστήματος τύπων της RQL, το οποίο είναι απαραίτητο για την βελτιστοποίηση της αποθήκευσης και της επερώτησης, ούτως ώστε να υπάρχει τύπος και για τις υποστασιοποιημένες δηλώσεις. Τέλος, προτάθηκαν συναρτήσεις οι οποίες πρέπει να ενσωματωθούν στην γλώσσα επερώτησης RQL, ώστε να είναι δυνατή η επερώτηση των υποστασιοποιημένων δηλώσεων.

Η υποστήριξη της υποστασιοποίησης στο RDF/S και στην RQL, σχεδιάστηκε προκειμένου να καταστήσει το RDF πιο ελκυστικό για χρήση σε εφαρμογές στις οποίες η αξιοπιστία και η πηγή των μεταδεδομένων παίζουν πρωταρχικό ρόλο. Μια χαρακτηριστική περίπτωση τέτοιου είδους εφαρμογών είναι για πράκτορες (agents) οι οποίοι ανταλλάσσουν πληροφορίες και απαιτείται να είναι γνωστή η πηγή των πληροφοριών που λαμβάνουν και αποστέλλουν. Το RDF με την υπάρχουσα λειτουργικότητα του δεν μπορεί να χρησιμοποιηθεί για εφαρμογές αυτού του είδους, ωστόσο με την ενσωμάτωση της υποστασιοποίησης σε αυτό τόσο κατά την αποθήκευση των μεταδεδομένων όσο και κατά την επερώτηση αυτών, θα επέτρεπε τέτοιου είδους χρήση. Επίσης, στην εργασία αυτή έγινε μια προσπάθεια να υποστηριχθούν οι Εκδόσεις RDF Σχημάτων (RDF Schema Versioning), με την βοήθεια της υποστασιοποίησης. Τέλος, μια άλλη χρήση της υποστασιοποίησης που παρουσιάζεται είναι αυτή της υποστήριξης τριαδικών σχέσεων, καθώς το RDF δεν υποστηρίζει σχέσεις αυτού του τύπου.

Προκειμένου να αξιολογηθεί η υπάρχουσα εργασία θα προτείναμε την υλοποίηση της σχεσιακής βάσης μεταδεδομένων της μελέτης αυτής, η οποία θα επιτρέπε την αποθήκευση των υποστασιοποιημένων προτάσεων. Η αξιολόγηση της αποδοτικότητας της προτεινόμενης σχεδίασης θα ήταν εφικτή με την υλοποίηση πειραμάτων και την εισαγωγή τόσο μεγάλου όγκου μεταδεδομένων όσο και υποστασιοποιημένων προτάσεων. Μια μελέτη της συμπεριφοράς της σχεσιακής αναπαράστασης μεταδεδομένων θα ήταν πολύ απαραίτητη, ούτως ώστε να ελεγχθεί η απόδοση του συστήματος – καθώς η υποστασιοποίηση έχει κατηγορηθεί κατά καιρούς ότι θα επιβράδυνε κατά πολύ την ταχύτητα και τις επιδόσεις ενός τέτοιου συστήματος. Ο εμπλουτισμός στη συνέχεια τους συστήματος τύπων της RQL και των συναρτήσεων αυτής θα αποτελούσε ένα πολύ ενδιαφέρον θέμα προς υλοποίηση και μελέτη. Επίσης, η πιθανή προσθήκη επιπλέον τελεστών για την υποστήριξη επερωτήσεων υποστασιοποίησης θα ήταν ένα θέμα προς μελέτη. Μετά την υποστήριξη RQL επερωτήσεων για την υποστασιοποίηση θα ήταν απαραίτητος και ο έλεγχος της απόδοσης των επερωτήσεων και πιθανές τροποποιήσεις ώστε να βελτιστοποιηθεί η απόδοση του συστήματος.

Το θέμα των υπεργράφων παραμένει ένα πολύ ενδιαφέρον ζήτημα καθώς δίνει δυνατότητες απλούστερης αναπαράστασης των RDF σχημάτων, και παραμένει ένα θέμα ανοιχτό προς μελέτη ούτως ώστε να γίνει αποδεκτό από το RDF.

Ένα άλλο θέμα το οποίο ανακύπτει είναι η χρήση του RDF και της RQL σε συστήματα πρακτόρων, και η αντικατάσταση παραδοσιακών γλωσσών ανταλλαγής μεταδεδομένων όπως είναι η KIF από το RDF. Η προσθήκη επιπλέον λειτουργικότητας στο RDF, όπως αυτή της υποστασιοποίησης, θα το καταστήσει πολύ περισσότερο δημοφιλές.

Τέλος, οι Εκδόσεις RDF Σχημάτων είναι ένα ζήτημα που παρουσιάζει πολύ μεγάλο επιστημονικό ενδιαφέρον και θα ήταν πολύ χρήσιμη η μελέτη αυτού σε σχέση με το RDF και την υποστασιοποίηση και η υλοποίηση ενός τέτοιου συστήματος.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [BG00] Dan Brickley and R.V.Guha. *Resource Description Framework (RDF) Schema Specification 1.0*. W3C Candidate Recommendation, March 2000. Available at <http://www.w3c.org/TR/rdf-schema>.
- [BG02] D. Brickley, R.V. Guha. *RDF Vocabulary Description Language 1.0: RDF Schema*. W3C Workig Draft. April 30, 2002. Available at <http://www.w3.org/TR/rdf-schema/>
- [BHL99] T. Bray, D. Hollander, and A. Layman. *Namespaces in XML*. W3C Recommendation. January 14, 1999.
- [BKD+01] J. Broekstra, M. Klein, S. Decker, D. Fensel, F. Harmelen, and I. Horrocks. *Enabling knowledge representation on the Web by extending RDF Schema*. Linkoping University Electronic Press, Sweden. February 2, 2001. Available at <http://www.ep.liu.se/ea/cis/2001/001/>.
- [BM00] D. Brickley and L. Miller. *Rdf, sql and the semantic web - a case study*. Available at ilrt.org/discovery/2000/10/swsql/.
- [BPS98] Tim Bray, Jean Paoli, and C.M.Sperberg-McQueen. *Extensible Markup Language (XML) 1.0*. W3C Recommendation, 10 February 1998. Available at <http://www.w3c.org/TR/REC-xml>.
- [Bri01] Brian McBride. *Jena: Implementing the RDF Model and Syntax Specification*.
- [BWT+] Luis Botelho, Steven Willmott, Tianning, Zhang Jonathan Dale. *Review of Content Languages Suitable for Agent-Agent Communication*.
- [CB97] R.G.G. Cattell and D. Barry. *The Object Database Standard ODMG 2.0*. Morgan Kaufmann, 1997.
- [CD99] J.Clark and S. DeRose. *XML Path Language (XPath)*. W3C Recommendation, November 1999. Available at <http://www.w3c.org/TR/xpath>

- [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>.
- [CR] C. Cumby, and D. Roth. *Learning with Feature Description Logics*. Department of Computer Science University of Illinois.
- [DC] *Dublin Core Metadata Initiative*. Available at <http://purl.oclc.org/dc>.
- [DJM00] S. DeRose, R. Daniel Jr., and E. Maler. *XML Pointer Language (XPointer)*. W3C Candidate Recommendation, July 2000. Available at <http://www.w3c.org/TR/xptr>.
- [DMO+00] S. DeRose, E. Maler, D. Orchard, and B. Trafford. *XML Linking Language (XLink)*. W3C Candidate Recommendation, July 2000. Available at <http://www.w3c.org/TR/xlink>.
- [FFM+] T. Finin, R. Fritzon, D. McKay, and R. McEntire. *KQML as an Agent Communication Language*. Computer Science Department and University of Maryland Baltimore County.
- [FK99] D. Florescu and D. Kossmann. *A performance evaluation of alternative mapping schemes for storing xml data in a relational database*. Technical Report 3680, INRIA Rocquencourt, France, 1999.
- [F01] Roberto A. Flores-Mendez. *Towards a Standardization of Multi-Agent System Frameworks*. ACM, 21-Jan-01. Available at www.acm.org/crossroads/xrds5-4/multiagent.html
- [G85] Joseph Gabel. *False Consciousness: An Essay on Reification*. Harpercollins, 1985. Available at <http://www.philosophy-pages.com/dy/r.htm#reif>
- [GB97] R.V. Guha and Tim Bray. *Meta Content Framework using XML*. June 1997. Available at <http://www.w3c.org/TR/NOTE-MCF-XML/>.
- [H02] P. Hayes. “*RDF Model Theory*”. W3C Working Draft. April 29, 2002. Available at <http://www.w3.org/TR/rdf-mt>
- [HKQ] David Huynh, David Karger, Dennis Quan. *Haystack: A Platform for Creating, Organizing and Visualizing Information Using RDF*.
- [JAD] *Jade and Ontologies*. Available at <http://www.ryerson.ca/~dgrimsha/courses/cps720/JADEOntology.html>

- [K2002] Klyne Graham. “*Circumstance, provenance and partial knowledge*”. 13 March 2002
- [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.
- [KS] G.M Kuper and J. Simeon. *Subsumption for XML Types*. To appear in the International Conference on Database Theory (ICDT 2001), January 2001, London, UK.
- [L] J. Liljegen. *Description of an RDF database implementation*. Available at WWW-DB.stanford.edu/~melnik/rdf/db-jonas.html.
- [La] Ora Lassila. *RDF Metadata and Agent Architectures*. Nokia Research Center
- [LFP] Yannis Lambrou, Tim Finin, and Yun Peng. *Agent Communication Languages: The Current Landscape*. University of Maryland, Baltimore Country.
- [LLJ96] C. Lagoze, C.A. Lynch, and R. Daniel Jr. *The Warwick Framework: A Container Architecture for Aggregating Sets Of Metadata*. Technical Report 1593, Cornell University, June 1996. Available at <http://cstr.cs.cornell.edu/Dienst/UI/2.0/Describe/ncstrl.cornell/TR-1593>.
- [LS99] O.Lassila and R.Swick. *Resource Description Framework (RDF) Model and Syntax Specification*. W3C Recommendation, February 1999. Available at <http://www.w3c.org/TR/REC-rdf-syntax>.
- [M] S. Melnik. *Storing RDF in a relational database*. Available at <http://www-db.stanford.edu/~melnik/rdf/db.html>.
- [MD01] Sergey Melnik, Stefan Decker. *Representing Order in RDF*. Database Group, Stanford University. January 2001.
- [NDW01] W. Nejdl, H. Dhraief, and M. Wolpers, *O-telos-rdf: a Resource Description Format with Enhanced Meta-Modeling Functionalities Based on O-telos*. In Workshop on Knowledge Markup and Semantic Annotation at the 1st International Conference on Knowledge Capture (K-CAP 2001), Victoria, BC., Canada. 2001.
- [O] Available at <http://www.ontopia.net/topicmaps/materials/tmrdfoildaml.html>
- [O00] Ora Lassila. *RDF Metadata and Agent Architectures*.2000

- [ONT] Available at <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html#1>.
- [OS1999] Ora Lassila, Ralph R. Swick. “*Resource Description Framework (RDF) Model and Syntax Specification*”. W3C Recommendation 22 February 1999
- [SD01] Michael Sintek, Stefan Decker. “*TRIPLE --- An RDF Query, Inference and Transformation Language*”. October 2001
- [Y01] G. Michael Youngblood. *Web Hunting: Design of a Simple Intelligent Web Search Agent*. ACN, March 2001. Available at <http://www.acm.org/crossroads/xrds5-4/webhunting.html>