Πανεπιστήμιο Κρήτης

Σχολή Θετικών Επιστημών

Τμήμα Επιστήμης Υπολογιστών

Αποθήκευση Μεταδεδομένων RDF για Πύλες Κοινοτήτων Διαδικτύου

Σοφία Μ. Αλεξάκη

Μεταπτυχιακή Εργασία

Ηράκλειο, Νοέμβριος 2000

Πανεπιστήμιο Κρήτης
Σχολή Θετικών Επιστημών
Τμήμα Επιστήμης Υπολογιστών

Αποθήκευση Μεταδεδομένων RDF για Πύλες Κοινοτήτων Διαδικτύου

Εργασία που υποβλήθηκε από την
Σοφία Μ. Αλεξάκη
ως μερική εκπλήρωση των απαιτήσεων για την απόκτηση
ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΔΙΠΛΩΜΑΤΟΣ ΕΙΔΙΚΕΥΣΗΣ

Συγγραφέας:

_____
Σοφία Μ. Αλεξάκη
Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

Εισηγητική Επιτροπή:

_____
Δημήτριος Πλεξουσάκης
Επίκουρος Καθηγητής, Επόπτης

_____
Πάνος Κωνσταντόπουλος
Καθηγητής, Μέλος

_____
Ευάγγελος Μαρκάτος
Επίκουρος Καθηγητής, Μέλος

_____
Βασίλης Χριστοφίδης
Ερευνητής Γ΄, Ι.Π. Ι.Τ.Ε, Επιβλέπων
Δεκτή:

_____
Πάνος Κωνσταντόπουλος, Καθηγητής
Πρόεδρος Επιτροπής Μεταπτυχιακών Σπουδών

Ηράκλειο, Νοέμβριος 2000

Στον πατέρα μου

Αποθήκευση Μεταδεδομένων RDF για Πύλες Κοινοτήτων Διαδικτύου

Σοφία Μ. Αλεξάκη

Μεταπτυχιακή Εργασία

Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

Περίληψη

Τα τελευταία χρόνια υπάρχει αυξημένο ενδιαφέρον για τις πύλες κοινοτήτων διαδικτύου (π.χ. εταιρικές, ηλεκτρονικού εμπορίου). Αναπόσπαστο μέρος μιας πύλης αποτελεί ο κατάλογος γνώσης στον οποίο αποθηκεύονται ποικίλα είδη μεταδεδομένων (π.χ. ταξινόμησης, διαχείρισης, αξιολόγησης περιεχομένου, ασφάλειας κτλ.) για τους πληροφοριακούς πόρους που υπάρχουν στην κοινότητα (π.χ. έγγραφα, δεδομένα). Ο κατάλογος γνώσης υποστηρίζει την οργάνωση του περιεχομένου με διαφορετικούς τρόπους, χρησιμοποιώντας διάφορες οντολογίες και λεξιλόγια διαθέσιμα σε μια κοινότητα. Μέσα σ' αυτό το πλαίσιο οι βασικές λειτουργίες που υποστηρίζονται από μια πύλη είναι επερώτηση των πληροφοριακών πόρων με σημασιολογικά κατανοητό τρόπο, προσαρμογή περιεχομένου στον εκάστοτε χρήστη και ολοκλήρωση ετερογενών πηγών. Ο μεγάλος όγκος του καταλόγου γνώσης (π.χ. Open Directory: 180 Mbytes για κατηγορίες και 700 Mbytes για περιγραφές) καθιστά τη διαχείριση μεταδεδομένων με τη χρήση συστημάτων βάσεων δεδομένων ένα ιδιαίτερα ενδιαφέρον ζήτημα. Ένα πρότυπο που έχει προταθεί πρόσφατα για την περιγραφή πόρων και το οποίο καλύπτει τις ανάγκες μεταδεδομένων για πύλες κοινοτήτων διαδικτύου είναι το RDF.

Αντικείμενο της παρούσας εργασίας είναι η αναπαράσταση και αποθήκευση RDF μεταδεδομένων σε συστήματα οντοκεντρικών σχεσιακών βάσεων δεδομένων. Οι ιδιαιτερότητες του RDF σε σχέση με παραδοσιακά μοντέλα δεδομένων καθιστούν την αναπαράσταση του σε συστήματα βάσεων δεδομένων αρκετά πολύπλοκη. Προκειμένου να καθορίσουμε τους σημασιολογικούς περιορισμούς που πρέπει να πληρούν τα μεταδεδομένα σε μια κοινότητα διαδικτύου αλλά και για να πετύχουμε την αποτελεσματική αποθήκευση τους μελετούμε μια θεωρητική θεμελίωση του RDF

χρησιμοποιώντας την γλώσσα παράστασης γνώσης Telos. Το μοντέλο αναπαράστασης που προτείνουμε υποστηρίζει την εκφραστικότητα και ελευθερία περιγραφών που παρέχει το RDF και τη δυνατότητα πολλαπλής ταξινόμησης. Επίσης λαμβάνει υπόψη την ιδιαιτερότητα του RDF γράφου που έχει ετικέτες τόσο στις ακμές όσο και στους κόμβους, παρέχει δυνατότητα βελτιστοποίησης της επεξεργασίας των ιεραρχιών κλάσεων και ιδιοτήτων και συμβάλλει στην δημιουργία αποδοτικών επερωτήσεων. Ένα ιδιαίτερο χαρακτηριστικό του συστήματος που έχει υλοποιηθεί είναι ότι καθιστά δυνατή την βαθμιαία φόρτωση των δεδομένων RDF και σχημάτων στη βάση, κάτι ιδιαίτερα σημαντικό για μεγάλους όγκους RDF μεταδεδομένων.

Επόπτης: Δημήτριος Πλεξουσάκης
Επίκουρος Καθηγητής Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

Storage of RDF Metadata for Community Web Portals

Sofia M. Alexaki

Master of Science Thesis

Computer Science Department
University Of Crete

Abstract

An increasing interest in Community Web Portals (corporate, e-marketplace etc.) can be observed over the last few years. The core Portal component is the Knowledge Catalog holding different types of metadata (e.g. cataloguing, content rating) associated with different types of portal resources (e.g. documents, data) that are available to the community members. The Knowlegde Catalog supports the organization of the content of the portal in a multitude of ways by employing ontologies and thesauri. Fundamental services of a Portal, such as querying in semantically meaningful way, personalization and integration of heterogeneous sources, are based on the knowledge Catalog. The expected large volume of the knowledge catalog (for instance, the Open Directory Portal of Netscape comprises around 180M of Subject Topics and 700M of indexed URIs) makes the management of metadata using database management systems a challenging topic. RDF, a W3C recommendation is by now a standard framework for the description of resources.

The topic of this thesis is the representation and the efficient storage of RDF metadata in object-relational database management systems (ORDBMS). Due to the peculiarities of RDF in comparison to conventional data model (e.g. properties are first class objects, optional and multivalued properties, specialization between properties), its representation in ORDBMS is fairly complex. In order to define the semantic constraints which metadata of a community should conform to and to achieve its effective storage, we study the theoretical underpinnings of the problem using the knowledge representation language Telos. Our general model is expressive enough to represent the majority of RDF primitives as well as multiple instantiation. It also considers that RDF graphs have labels on both nodes and edges, optimizes processing of class/property hierarchies and ensures better query performance. A distinctive feature of our system is

the fact that it enables incremental loading of RDF descriptions and schemas, which is crucial for handling the large volumes of RDF metadata.

Supervisor: Dimitris Plaxousakis
Assistant Professor of Computer Science
University of Crete

Ευχαριστίες

Στο σημείο αυτό θα ήθελα να ευχαριστήσω θερμά τον επόπτη καθηγητή μου κ. Δημήτρη Πλεξουσάκη για την πολύτιμη βοήθεια, καθοδήγηση και για την άψογη συνεργασία μας σε όλη την διάρκεια της παρούσας εργασίας.

Ιδιαίτερα θα ήθελα να ευχαριστήσω τον επιβλέποντα της εργασίας μου κ. Βασίλη Χριστοφίδη για την καθοδήγηση του, τον χρόνο που μου αφιέρωσε και τις γνώσεις που απέκτησα κοντά του κατά την διάρκεια συνεργασίας μας. Οφείλω να αναγνωρίσω ότι η συνεχής υποστήριξη και βοήθειά του ήταν καθοριστική για την ολοκλήρωση της εργασίας μου.

Επίσης θα ήθελα να ευχαριστήσω τον Γρηγόρη Καρβουναράκη και τον Karsten Tolle καθώς η συνεργασία μας στα πλαίσια του προγράμματος, του οποίου τμήμα αποτέλεσε αυτή η εργασία, ήταν πολύ εποικοδομητική.

Θα πρέπει να ευχαριστήσω τα μέλη της ομάδας Πληροφοριακών Συστημάτων και Τεχνολογίας Λογισμικού του Ινστιτούτου Πληροφορικής και ιδιαίτερα τον Χρήστο Γεωργή για την βοήθεια που μου πρόσφεραν.

Θα ήθελα ακόμα να ευχαριστήσω όλους τους φίλους μου και ιδιαίτερα τον Άρη Νικολογιάννη για την αμέριστη συμπαράσταση, κατανόηση και φιλία τους σε μια πολύ δύσκολη για μένα περίοδο.

Το μεγαλύτερο όμως ευχαριστώ το χρωστώ στους γονείς μου Ματθαίο και Αθηνά και στην αδερφή μου Ελευθερία για την αγάπη τους, τις θυσίες που έχουν κάνει για μένα, την εμπιστοσύνη που μου δείχνουν και την δύναμη που μου δίνουν.

Περιεχόμενα

Κατάλογος Εικόνων

Κατάλογος Πινάκων

Κεφάλαιο 1

Εισαγωγή

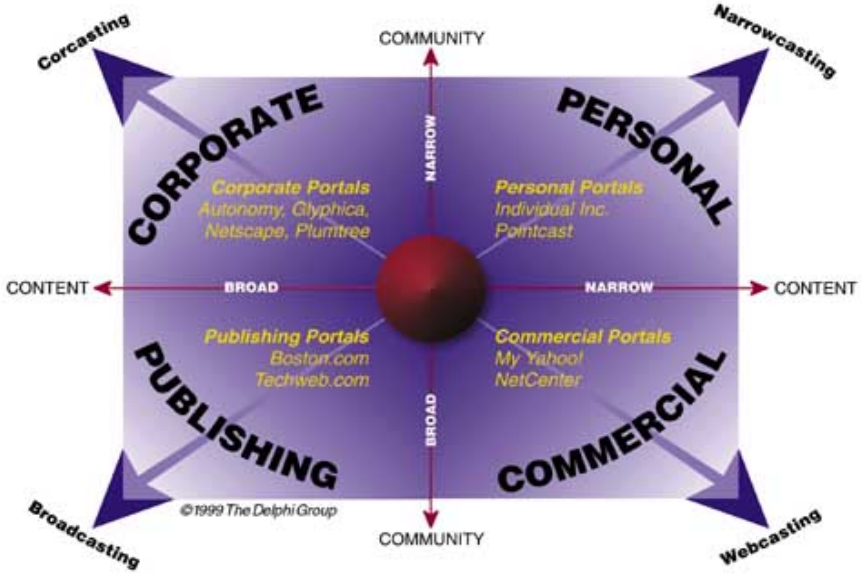
Ο όρος *πύλη* (portal) είναι ένας νέος όρος που χρησιμοποιείται ευρέως τα τελευταία χρόνια. Δεν υπάρχει σαφής ορισμός για το τι είναι μια πύλη και τι είδους υπηρεσίες πρέπει να προσφέρει. Γενικά, μπορεί να χαρακτηριστεί σαν ένα σημείο που ολοκληρώνει πληροφορία, εφαρμογές και ανθρώπους και προσφέρει εύκολη, περιεκτική και χρήσιμη προσπέλαση [S99a]. Το βασικό χαρακτηριστικό μιας πύλης είναι ότι οργανώνει την πληροφορία χρησιμοποιώντας ένα κατάλογο που συνήθως έχει δεντρική μορφή και επιτρέπει την αναζήτηση πληροφορίας μέσω της πλοήγησης πάνω σ΄ αυτόν τον κατάλογο. Θα πρέπει να τονιστεί ότι μια πύλη οργανώνει την πληροφορία. Αν και μπορεί να 'φιλοξενεί' πληροφορία που προσφέρεται από προμηθευτές πληροφορίας (content providers) δεν αποτελεί η ίδια βασικό προμηθευτή.

Παράλληλα μια πύλη συχνά παρέχει την λειτουργικότητα που προσφέρουν οι μηχανές αναζήτησης. Δηλαδή την δυνατότητα επερώτησης δίνοντας λέξεις κλειδιά τα οποία μπορούν να συσχετιστούν μεταξύ τους με δυαδικές εκφράσεις π.χ. and, or. Επίσης μπορεί να παρέχει υπηρεσίες, όπως δωρεάν ηλεκτρονική διεύθυνση ή δυνατότητα επικοινωνίας μέσω του διαδικτύου με άλλους χρήστες (π.χ. chat rooms). Μια ακόμα υπηρεσία που μπορεί να προσφέρει μια πύλη είναι η προσαρμογή τόσο της μορφής όσο και του περιεχομένου των σελίδων στις προτιμήσεις του χρήστη (customization). Για παράδειγμα, είναι δυνατόν να τροποποιηθεί η αρχική σελίδα της πύλης που εμφανίζεται στον χρήστη και να προστεθεί σ' αυτήν η πρόγνωση του καιρού ή τα αθλητικά νέα. Ένα αντιπροσωπευτικό παράδειγμα πύλης είναι το Yahoo![1].

[1] http://www.yahoo.com/

1.1 Κατηγοριοποίηση Πυλών

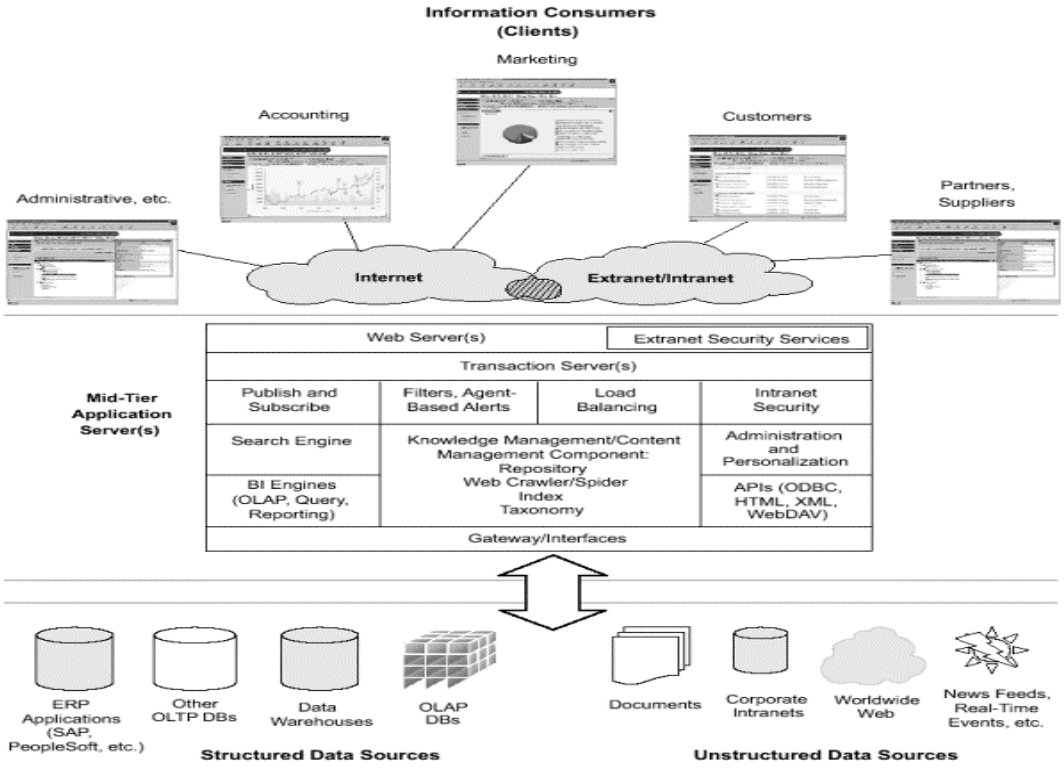
Κατηγοριοποίηση των πυλών μπορεί να γίνει με βάση την *ποικιλία του περιεχομένου τους* καθώς και το *εύρος της κοινότητας* που απευθύνονται [Delphi99] (βλέπε εικόνα 1.1). Οι **Εκδοτικές (Publishing)** πύλες (π.χ. Boston.com, Techweb.com) απευθύνονται σε μεγάλες κοινότητες με ποικίλα ενδιαφέροντα. Παρέχουν κυρίως δυνατότητα επερώτησης με λέξεις κλειδιά και απαιτείται ελάχιστη αλληλεπίδραση από την μεριά του χρήστη. Όλοι οι μεγάλοι προμηθευτές πληροφορίας (π.χ., CNN, ABC News) ανήκουν στην παραπάνω κατηγορία. Οι **Εμπορικές (Commercial)** πύλες (π.χ. My Yahoo!, NetCenter) απευθύνονται σε πολλαπλές και ποικίλες κοινότητες, όπως και οι Εκδοτικές, επιπλέον όμως προσφέρουν την δυνατότητα προσαρμογής του περιεχομένου στα ενδιαφέροντα και τις προτιμήσεις του χρήστη. Οι πύλες αυτές είναι οι πιο δημοφιλείς για τις on-line κοινότητες. Οι **Προσωπικές (Personal)** πύλες (π.χ. Individual Inc.) παρέχουν περιορισμένο περιεχόμενο όμως προσαρμόζονται σε μεγάλο βαθμό στις προτιμήσεις του κάθε χρήστη. Για παράδειγμα, μια λειτουργία που μπορούν να παρέχουν είναι η παρουσίαση στην αρχική σελίδα των ειδήσεων που αναφέρονται σε κατηγορίες που έχει επιλέξει ο χρήστης ή ακόμα και την αποστολή τους στην ηλεκτρονική διεύθυνση του χρήστη.



Εικόνα 1.1. Κατηγοριοποίηση Πυλών.

Τέλος, οι **Εταιρικές (Corporate)** πύλες [FA99] περιέχουν πλούσιο περιεχόμενο για ένα συγκεκριμένο πεδίο/εταιρεία. Το περιεχόμενο τους είναι ευρύτερο από το περιεχόμενο των Εμπορικών πυλών εφόσον υπάρχει μεγάλη ποικιλία και πολυπλοκότητα

στην πληροφορία που διαχειρίζονται. Σκοπός των πυλών αυτής της κατηγορίας είναι να παρέχουν πρόσβαση, αλληλεπίδραση και διανομή της γνώσης που υπάρχει στην εταιρεία. Σύμφωνα με τον ορισμό που δίνεται στην αναφορά της Merill Lynch [ST98] οι Εταιρικές πύλες είναι εφαρμογές που δίνουν την δυνατότητα στις εταιρείες να χρησιμοποιήσουν την πληροφορία που είναι αποθηκευμένη είτε μέσα στην εταιρεία είτε εξωτερικά και παρέχουν στους χρήστες ένα περιβάλλον που μπορεί να προσαρμόζεται στις ανάγκες τους και παίζει καθοριστικό ρόλο στη λήψη των αποφάσεων. Είναι συνδυασμός από εφαρμογές που συγκεντρώνουν, διαχειρίζονται, αναλύουν και διανέμουν πληροφορία μέσα και έξω από την εταιρεία.



Εικόνα 1.2. Πύλη Γνώσης.

Παράλληλα οι πύλες κατηγοριοποιούνται σε πύλες *δεδομένων* (data), *πληροφορίας* (information), *συνεργατικές* (collaborative) και *γνώσης* (knowledge) (βλέπε εικόνα 1.2) [W99a], [T99], [S99b]. Οι πύλες δεδομένων διαχειρίζονται κυρίως δομημένα δεδομένα τα οποία αποθηκεύονται σε βάσεις δεδομένων. Οι πύλες πληροφορίας διαχειρίζονται αδόμητα δεδομένα όπως ηλεκτρονικά μηνύματα, απλά κείμενα, εικόνες. Οι συνεργατικές πύλες προσφέρουν δυνατότητες αλληλεπίδρασης μεταξύ των χρηστών της κοινότητας.

Τέλος, οι πύλες γνώσης παρέχουν τις δυνατότητες που προσφέρουν οι παραπάνω κατηγορίες πυλών και επιπλέον παρέχουν ένα **κατάλογο γνώσης** (knowledge catalog). Ο κατάλογος γνώσης είναι μια 'αποθήκη' μεταδεδομένων και υποστηρίζει την οργάνωση του περιεχομένου της πύλης με διαφορετικούς τρόπους, καθώς αποθηκεύει μεταδεδομένα διαφορετικών τύπων για τους πληροφοριακούς πόρους της πύλης (π.χ. έγγραφα, τμήματα εγγράφων, δεδομένα. Μερικοί ενδεικτικοί τύποι μεταδεδομένων που περιέχονται στον κατάλογο γνώσης είναι τα μεταδεδομένα ταξινόμησης (cataloguing), διαχείρισης, αξιολόγησης περιεχομένου και ασφάλειας πρόσβασης.

1.2 Υπηρεσίες που παρέχονται από μια πύλη

Στην συνέχεια θα περιγράψουμε τις βασικές υπηρεσίες που ενδέχεται να υποστηρίζει μια πύλη. Οι υπηρεσίες μπορούν να χωριστούν στις παρακάτω κατηγορίες:

- *Υπηρεσίες πληροφορίας:* Αναφέρονται στην ολοκλήρωση της πληροφορίας που βρίσκεται σε ετερογενείς πηγές πληροφορίας όπως βάσεις δεδομένων, συστήματα διαχείρισης εγγράφων, δομημένα έγγραφα, απλά κείμενα και στην παρουσίαση της στον χρήστη με ενιαίο τρόπο. Ακόμα αναφέρονται στην παροχή τρόπων επερώτησης της πληροφορίας, στην υποστήριξη ποικίλων τύπων αρχείων όπως ήχο, εικόνα, κείμενο, στην διαχείριση εγγράφων και στους μηχανισμούς εξόρυξης της πληροφορίας.

- *Υπηρεσίες εφαρμογής***:** Δίνουν την δυνατότητα ομοιόμορφης προσπέλασης τόσο σε δεδομένα που είναι προσπελάσιμα μέσω του αναδιφητή όσο και σε δεδομένα που δεν είναι άμεσα προσπελάσιμα, όπως δεδομένα αποθηκευμένα σε βάσεις δεδομένων.

- *Υπηρεσίες συνεργατικότητας:* Υποστηρίζουν επικοινωνία πραγματικού χρόνου μεταξύ των χρηστών της πύλης. Δίνουν την δυνατότητα διαλόγου μεταξύ των χρηστών, χρήσης κοινών εγγράφων, υπηρεσίες ηλεκτρονικού ταχυδρομείου κτλ.

- *Υπηρεσίες πρόσβασης και ολοκλήρωσης:* Παρέχουν ταξινόμηση της πληροφορίας καθώς και ολοκλήρωση περιεχομένου που είναι απαραίτητη για την δημιουργία δυναμικών εγγράφων που ανταποκρίνονται στις απαιτήσεις του χρήστη. Επίσης παρέχουν την δυνατότητα προσαρμογής του περιεχομένου στις προτιμήσεις του χρήστη (personalization) και προσφέρουν υπηρεσίες ασφάλειας.

- *Υπηρεσίες παρουσίασης:* Παρέχουν διεπαφές χρήσης που διευκολύνουν τους χρήστες και μπορούν να προσαρμοστούν στις προτιμήσεις τους.

- *Υπηρεσίες διαχείρισης:* Περιλαμβάνουν υπηρεσίες διαχείρισης της πύλης, όπως ενημέρωση των χρηστών όταν συμβαίνουν αλλαγές στην πύλη (π.χ. προσθήκη νέων εφαρμογών ή περιεχομένου), δυνατότητα καταχώρησης χρηστών κτλ.

Η υλοποίηση ενός μεγάλου μέρους των παραπάνω υπηρεσιών, όπως επερώτηση και μάλιστα με σημασιολογικά κατανοητό τρόπο πληροφοριακών πόρων, προσαρμογή περιεχομένου στον εκάστοτε χρήστη (personalization), ολοκλήρωση ετερογενών πηγών, βασίζεται στον κατάλογο γνώσης. Να σημειώσουμε ότι το μεγαλύτερο μέρος των υπηρεσιών που περιγράφτηκαν συνήθως παρέχονται από ***Πύλες Κοινοτήτων Διαδικτύου*** (π.χ. εταιρικές) οι οποίες απευθύνονται σε συγκεκριμένες κοινότητες χρηστών και όχι σε όλους τους χρήστες του διαδικτύου. Αυτό οφείλεται στο γεγονός στις παραπάνω πύλες υπάρχουν πλουσιότερες περιγραφές για τους πληροφοριακούς πόρους.

1.3 Διαχείριση Καταλόγου Γνώσης

Με βάση όσα έχουν αναφερθεί είναι φανερό ότι η διαχείριση του καταλόγου γνώσης αποτελεί βασικό ζήτημα για την ανάπτυξη μιας πύλης. Ο κατάλογος γνώσης προκειμένου να υποστηρίξει το πλήθος των παραπάνω λειτουργιών είναι ιδιαίτερα σύνθετος. Υποστηρίζει συσχετίσεις μεταξύ κατηγοριών, πολλαπλή ταξινόμηση, ημιδομημένες περιγραφές κτλ. Για την μοντελοποίηση του καταλόγου γνώσης προτείνουμε την χρήση των W3C πρότυπων για την περιγραφή πόρων, συγκεκριμένα των προτύπων RDF και RDFS. Να σημειώσουμε ότι τα υπάρχοντα εμπορικά συστήματα για την ανάπτυξη πυλών για κοινότητες διαδικτύου (π.χ. Epicentric[2], Plumtree[3], Autonomy[4]) καθώς επίσης και τα εμπορικά συστήματα αποθήκευσης μεταδεδομένων που έχουν υλοποιηθεί π.χ. Microsoft Repository [M00a] δεν χρησιμοποιούν τα πρότυπα του παγκόσμιου ιστού για την αναπαράσταση των μεταδεδομένων.

Επίσης προκειμένου να χειριστούμε τους μεγάλους όγκους μεταδεδομένων του καταλόγου γνώσης χρησιμοποιούμε Συστήματα Διαχείρισης Σχεσιακών Βάσεων Δεδομένων (ΣΔΣΒΔ) για την αποθήκευση των μεταδεδομένων. Ένα παράδειγμα καταλόγου γνώσης είναι ο κατάλογος του Open Directory (ODP)[5]. Αποτελείται από 180 Mbytes για περιγραφές κατηγοριών και 700 Mbytes για περιγραφές σελίδων.

[2] www.epicentric.com
[3] www.plumtree.com
[4] www.autonomy.com
[5] www.dmoz.org

1.4 Πλαίσιο Περιγραφής Πόρων (Resource Description Framework)

Για την μοντελοποίηση του καταλόγου γνώσης επιλέξαμε το Πλαίσιο Περιγραφής Πόρων (Resource Description Framework, RDF). Πρόκειται για ένα πρότυπο που έχει προταθεί από το W3C για να διευκολύνει την αναπαράσταση και ανταλλαγή μεταδεδομένων. Το ευέλικτο μοντέλο δεδομένων του RDF έχει συμβάλλει στην υιοθέτηση του από πολλούς προμηθευτές περιεχομένου (content providers) όπως ABCNews, CNN, Time Inc, New Media και πύλες του διαδικτύου όπως το Open Directory, CNET[6], XMLNews[7]. Επιπλέον η τελευταία έκδοση (6)[8] του Netscape χρησιμοποιεί το RDF για να αποθηκεύσει πληροφορία για τον χρήστη, όπως τις πιο προσφιλείς σελίδες, τις πιο πρόσφατες σελίδες που έχει 'επισκεπτεί' καθώς και για να οργανώνει τα προσωπικά του ηλεκτρονικά μηνύματα με βάση την σημασιολογία τους.

Στην συνέχεια θα δείξουμε ότι το RDF μπορεί καλύψει τις απαιτήσεις των μεταδεδομένων που υπάρχουν σε κοινότητες διαδικτύου. Στο RDF οτιδήποτε (π.χ. μια σελίδα του διαδικτύου, ένας δικτυακός τόπος, ένα άτομο, ένα XML στοιχείο) θεωρείται πόρος και του αποδίδεται ένα URI. Οι περιγραφές που υπάρχουν για τους πόρους αναπαριστώνται μέσω ιδιοτήτων που μπορούν να αποδοθούν στους πόρους. Συγκεκριμένα, το RDF ορίζει ένα μοντέλο γράφων όπου οι πόροι αποτελούν τους κόμβος του γράφου και οι ιδιότητες τις ακμές του. Επομένως με το RDF είναι δυνατόν να αναπαρασταθούν οι περιγραφές που αποδίδονται σε κάθε είδους πληροφοριακό πόρο μιας κοινότητας (π.χ. μια σελίδα του διαδικτύου, ένα ηλεκτρονικό μήνυμα, ένα αρχείο pdf) και μάλιστα να αποδοθούν σ' αυτούς διαφορετικοί τύποι μεταδεδομένων.

Το RDF παρέχει την δυνατότητα ορισμού σχημάτων τα οποία επιτρέπουν την ερμηνεία των μεταδεδομένων. Τα RDF σχήματα ορίζουν τις δυνατές/επιτρεπτές ετικέτες για τους κόμβους και τις ακμές ενός γράφου περιγραφών RDF. Οι RDF ιδιότητες είναι προαιρετικές και πλειότιμες, επομένως οι ετερογενείς περιγραφές που υπάρχουν για τους πόρους της πύλης μπορούν να αναπαρασταθούν σαν μεταδεδομένα RDF.

Στους πληροφοριακούς πόρους μιας πύλης συχνά αποδίδονται μεταδεδομένα που περιγράφουν τον πόρο από διαφορετικές όψεις. Για παράδειγμα ας θεωρήσουμε μια σελίδα του διαδικτύου που απεικονίζει ένα πίνακα ζωγραφικής. Σ' αυτήν την σελίδα θα αποδοθούν περιγραφές που αναφέρονται στην φύση της σαν σελίδα π.χ. η ημερομηνία

[6] home.cnet.com
[7] www.xmlnews.org
[8] www.netscape.com

τελευταίας τροποποίησης, το θέμα της σελίδας. Παράλληλα θα αποδοθούν περιγραφές που σχετίζονται με τον πίνακα π.χ. ο δημιουργός του πίνακα, η χρονολογία κατασκευής. Το γεγονός ότι στο RDF ένας πόρος μπορεί να είναι μέλος πολλαπλών κλάσεων καθιστά δυνατή την περιγραφή του από διαφορετικές όψεις.

Τέλος, το RDF καθιστά δυνατή την εξειδίκευση τόσο των κλάσεων όσο και των ιδιοτήτων συμβάλλοντας στην δημιουργία υπο-κοινοτήτων. Επίσης καθιστά δυνατή την περιγραφή σχημάτων διευκολύνοντας την επαναχρησιμοποίηση σχημάτων μεταξύ διαφορετικών κοινοτήτων.

1.5 Αντικείμενο της εργασίας

Σκοπός της παρούσας εργασίας είναι η αναπαράσταση και αποθήκευση μεταδεδομένων RDF σε συστήματα σχεσιακών βάσεων δεδομένων. Οι ιδιαιτερότητες του RDF σε σχέση με παραδοσιακά μοντέλα δεδομένων, όπως η αντιμετώπιση των ιδιοτήτων σαν αυτόνομες οντότητες (first-class objects) που προσδιορίζονται μόνο από το όνομα τους, η δυνατότητα πολλαπλής ταξινόμησης, η δυνατότητα δημιουργίας σχέσεων εξειδίκευσης μεταξύ ιδιοτήτων και το γεγονός ότι οι ιδιότητες είναι προαιρετικές και πλειότιμες, καθιστά αρκετά πολύπλοκη την αναπαράσταση του σε ΣΔΣΒΔ.

Αρχικά μελετήσαμε το RDF και εντοπίσαμε μια σειρά προβλημάτων που προκύπτουν εξαιτίας της ελευθερίας που παρέχει το RDF στον ορισμό και την περιγραφή σχημάτων. Λαμβάνοντας υπόψη τα προβλήματα που δημιουργούνται προτείνουμε ένα *περιορισμό* του RDF τον οποίο θεμελιώνουμε θεωρητικά χρησιμοποιώντας στην γλώσσα παράστασης γνώσης Telos [MBJK90]. Η θεμελίωση/περιορισμός του RDF είναι απαραίτητη για τον σαφή καθορισμό των σημασιολογικών περιορισμών που πρέπει να πληρούν τα μεταδεδομένα RDF καθώς επίσης και για την αποτελεσματική αποθήκευση και στην συνέχεια την επερώτηση τους.

Στην συνέχεια προτείναμε ένα γενικό μοντέλο αναπαράστασης των μεταδεδομένων RDF σε οντοκεντρικές σχεσιακές βάσεις δεδομένων το οποίο υποστηρίζει την εκφραστικότητα και ελευθερία περιγραφών που παρέχει το RDF. Επίσης λαμβάνει υπόψη του την ιδιαιτερότητα του γράφου RDF που έχει ετικέτες τόσο στις ακμές όσο και στους κόμβους και παρέχει δυνατότητα βελτιστοποίησης της επεξεργασία των ιεραρχιών κλάσεων και ιδιοτήτων, θεωρώντας ότι στις κλάσεις και στις ιδιότητες αποδίδονται κωδικοί που βασίζονται στην θέση τους στην ιεραρχία. Επιπλέον προτείναμε

μια σειρά παραλλαγών στο βασικό αυτό μοντέλο οι οποίες βασίζονται σε υποθέσεις που κάνουμε για τα σχήματα RDF και οι οποίες μπορούν μειώσουν τον αριθμό των πινάκων που δημιουργούνται στην βάση αλλά και τον αριθμό των συζεύξεων (joins) που απαιτούνται για την επερώτηση βάσεων περιγραφών RDF.

Το σύστημα που υλοποιήσαμε για την αποθήκευση των RDF μεταδεδομένων αποτελείται από δύο βασικά μέρη λογισμικού. Το πρώτο μέρος, ο *Εκτεταμένος Σημασιολογικός Αναλυτής*, ελέγχει την συνέπεια των μεταδεδομένων RDF. Το δεύτερο μέρος υλοποιεί την διαδικασία φορτώματος των μεταδεδομένων RDF στην βάση. Το σύστημα μας βασίζεται στο εργαλείο VRP που εκτελεί την συντακτική και σημασιολογική ανάλυση RDF περιγραφών. Ένα ιδιαίτερο χαρακτηριστικό του συστήματος είναι ότι καθιστά δυνατή την *βαθμιαία φόρτωση* των RDF δεδομένων και σχημάτων στη βάση. Αυτή η λειτουργικότητα είναι ιδιαίτερα χρήσιμη δεδομένου του μεγάλου όγκου μεταδεδομένων RDF όπως π.χ. ο κατάλογος γνώσης του Open Directory.

1.6 Οργάνωση της εργασίας

Στο δεύτερο κεφάλαιο μετά από μια σύντομη παρουσίαση του RDF παρουσιάζουμε ένα θεωρητικό μοντέλο για το RDF που βασίζεται στην γλώσσα παράστασης Telos.

Στο τρίτο κεφάλαιο αναλύονται και αξιολογούνται οι υπάρχουσες προσεγγίσεις στην αποθήκευση γράφων XML και ιδιαίτερα ημιδομημένων δεδομένων RDF.

Στο τέταρτο κεφάλαιο περιγράφεται μια γενική αναπαράσταση μεταδεδομένων RDF σε σχεσιακές βάσεις δεδομένων. Επίσης παρουσιάζονται παραλλαγές σε αυτή την γενική αναπαράσταση που βασίζονται σε υποθέσεις που μπορούμε να κάνουμε για τα εκάστοτε σχήματα RDF και έχουν ως σκοπό την αποδοτικότερη αποθήκευση των μεταδεδομένων. Στην συνέχεια παρουσιάζεται η αρχιτεκτονική του συστήματος αποθήκευσης μεταδεδομένων RDF που έχουμε υλοποιήσει.

Στο πέμπτο κεφάλαιο παρουσιάζεται μια γενική ανασκόπηση και αναφέρονται κάποια συμπεράσματα. Τέλος παρουσιάζουμε τις μελλοντικές επεκτάσεις του συστήματος και θέματα που πρέπει να μελετηθούν.

Κεφάλαιο 2

Παρουσίαση και Θεμελίωση της RDF/S

Σ' αυτό το κεφάλαιο θα παρουσιαστεί το Πλαίσιο Περιγραφής Πόρων (Resource Description Framework – RDF). Το RDF διευκολύνει την αναπαράσταση, ανταλλαγή και επαναχρησιμοποίηση μεταδεδομένων. Έχει προταθεί από το World Wide Web Consortium και περιγράφεται στα κείμενα RDF Model and Syntax Specification (RDF M&S) [LS99] και RDF Schema Specification (RDFS) [BG00]. Καθιστά δυνατή την διαλειτουργικότητα των μεταδεδομένων παρέχοντας μηχανισμούς που υποστηρίζουν κοινές συμβάσεις για την *σημασιολογία*, την *σύνταξη* και την *δομή* τους. Συγκεκριμένα, ορίζει ένα μοντέλο γράφων για την αναπαράσταση των μεταδεδομένων. Η σύνταξη που χρησιμοποιεί για την ανταλλαγή και επεξεργασία τους είναι η XML. Επίσης ορίζει μια γλώσσα περιγραφής σχημάτων (σύνολα κλάσεων και ιδιοτήτων), την RDFS η οποία συμβάλλει στην απόδοση σημασιολογίας και δομής στα μεταδεδομένα.

2.1 Υπόβαθρο του RDF

Το RDF είναι αποτέλεσμα της συνεργασίας ενός πλήθους κοινοτήτων που χρειάζονταν μία ευέλικτη και επεκτάσιμη αρχιτεκτονική για μεταδεδομένα, ικανή να εφαρμοστεί σε ένα ευρύ και ετερογενή χώρο, όπως ο παγκόσμιος ιστός. Ξεκίνησε σαν επέκταση του μηχανισμού Platform for Internet Content Selection (PICS) [KMRT96]. Πρόκειται για ένα γενικό μηχανισμό για την αναπαράσταση και μεταφορά πληροφορίας από ένα εξυπηρετητή (server) σε πελάτες (clients). Η πληροφορία αυτή αξιολογεί το περιεχόμενο των σελίδων του παγκόσμιου ιστού. Παραδείγματα τέτοιου είδους πληροφορίας είναι ότι η σελίδα περιέχει βία ή ότι έχει γραφτεί από κάποιο αξιόλογο ερευνητή. Κάθε οργανισμός ή άτομο μπορεί να αξιολογεί το περιεχόμενο των σελίδων

ανάλογα με τις ανάγκες του χρησιμοποιώντας αυθαίρετες τιμές. Δηλαδή δεν υπάρχουν κοινά αποδεκτά κριτήρια ή προκαθορισμένες τιμές. Παράλληλα οι χρήστες, για παράδειγμα οι γονείς, θα πρέπει να ρυθμίζουν τους αναδιφητές (browsers), έτσι ώστε να απορρίπτουν τις σελίδες που δεν ταιριάζουν στα κριτήρια τους. Η δημιουργία του μηχανισμού αυτού ήταν συνέπεια της ανάγκης επιβολής περιορισμών και αξιολόγησης του περιεχομένου του παγκόσμιου ιστού.

Παράλληλα, το RDF έχει επηρεαστεί από την περιοχή των δομημένων εγγράφων και ιδιαίτερα από την XML. Ακόμα έχει αντλήσει στοιχεία από τις περιοχές της αναπαράστασης γνώσης και των ψηφιακών βιβλιοθηκών. Άλλες προτάσεις που έχουν γίνει για μεταδεδομένα πληροφοριακών πόρων όπως το Dublin Core [DC], η αρχιτεκτονική Warwick Framework [LLD96] και η αρχιτεκτονική Metadata Content Framework [GB97] έχουν ασκήσει επιρροή στο σχεδιασμό του RDF.

2.2 Στόχοι-Χαρακτηριστικά του RDF

Ο βασικός σκοπός του RDF είναι να ορίσει ένα γενικό μηχανισμό περιγραφής 'πόρων' (resources) που θα παρέχει δια-λειτουργικότητα μεταξύ εφαρμογών που ανταλλάσσουν μεταδεδομένα. Η δια-λειτουργικότητα αφορά την σύνταξη, την δομή αλλά κυρίως την σημασιολογία των μεταδεδομένων. Καθιστά δηλαδή δυνατή την δημιουργία μεταδεδομένων με σημασιολογία κατανοητή από τις μηχανές (machine readable semantics). Παράλληλα το RDF δεν ορίζει εκ των προτέρων την σημασιολογία ενός πεδίου εφαρμογής, όπως συμβαίνει για παράδειγμα στο Dublin Core [DC]. Στο Dublin Core ορίζεται ένα σύνολο ιδιοτήτων π.χ. *Δημιουργός* (Creator), *Θέμα* (Subject) που μπορούν να χρησιμοποιηθούν για περιγραφή πόρων του διαδικτύου και να διευκολύνουν στην ανάκτησή τους. Αντίθετα μπορεί να εφαρμοστεί και να περιγράψει οποιοδήποτε πεδίο εφαρμογής. Παρέχει την δυνατότητα ορισμού σχημάτων σε οποιοδήποτε κοινότητα ή ακόμα και σε άτομα. Παρακάτω αναφέρονται μερικά άλλα γενικά χαρακτηριστικά του RDF [B98].

- **Εύκολη ανταλλαγή/μεταφορά:** Το απλό τριαδικό μοντέλο του RDF καθώς και η XML σύνταξη που χρησιμοποιείται για την δήλωση RDF περιγραφών συντελούν στην δημιουργία μεταδεδομένων που μπορούν εύκολα να διαβαστούν/κατανοηθούν από τους ανθρώπους και (κυρίως) τις μηχανές.

- **Επαναχρησιμοποίηση και Επεκτασιμότητα σχημάτων:** Ένας από τους βασικούς στόχους του RDF είναι να συμβάλλει στην επαναχρησιμοποίηση και επέκταση

σχημάτων που ορίζονται στις διάφορες κοινότητες. Για το σκοπό αυτό καθιστά εφικτή την δημιουργία μεταδεδομένων που βασίζονται σε πολλαπλά σχήματα. Για παράδειγμα, για την περιγραφή ενός πόρου μπορούν να χρησιμοποιηθούν ιδιότητες που ορίζονται στο Dublin Core και σε πλήθος άλλων σχημάτων. Επίσης παρέχει δυνατότητα εξέλιξης των υπαρχόντων σχημάτων και δημιουργίας σχημάτων βασισμένα σε ήδη υπάρχοντα. Για παράδειγμα, θα μπορούσε να δημιουργηθεί ένα σχήμα που θα ήταν εξειδίκευση του Dublin Core.

- **Δυνατότητα περιγραφής ιδιοτήτων και κλάσεων.** Είναι χρήσιμο να μπορούμε να ανακτήσουμε ιδιότητες και κλάσεις που καλύπτουν τις ανάγκες της εφαρμογής μας και έχουν οριστεί σε υπάρχοντα σχήματα . Γι' αυτό το σκοπό το RDF καθιστά δυνατή την δημιουργία μεταδεδομένων και για το σχήμα. Για παράδειγμα, μπορούμε να ρωτήσουμε αν υπάρχει μια ιδιότητα ορισμένη σε κάποιο σχήμα RDF που περιγράφει το *δημιουργό* μιας σελίδας του παγκόσμιου ιστού.

- **Δυνατότητα περιγραφής των ίδιων των μεταδεδομένων.** Για να διαπιστωθεί η αξιοπιστία των περιγραφών – κάτι ιδιαίτερα σημαντικό για τον παγκόσμιο ιστό όπου οποιοσδήποτε μπορεί να δημιουργήσει μεταδεδομένα – απαιτείται η ύπαρξη μεταδεδομένων για τα μεταδεδομένα των πληροφοριακών πόρων. Έστω για παράδειγμα η περιγραφή «*Ο τίτλος (title) της σελίδας με URI http://www.nga.gov/ είναι National Gallery of Art*». Είναι χρήσιμο να ξέρουμε σε ποιον αποδίδεται αυτή η περιγραφή.

2.3 Μοντέλο Δεδομένων RDF

Το μοντέλο δεδομένων του RDF είναι ένα πολύ γενικό μοντέλο για την περιγραφή των πληροφοριακών πόρων (resources). Στους πόρους αποδίδονται ιδιότητες. Οι ιδιότητες αντιστοιχούν είτε σε *σχέσεις* μεταξύ πόρων είτε σε *γνωρίσματα* του πόρου που αποδίδονται. Συγκεκριμένα, το μοντέλο δεδομένων του RDF αποτελείται από τρεις βασικούς τύπους δεδομένων: *Πόρους* (Resources), *Ιδιότητες* (Properties) και *Δηλώσεις* (Statements).

Πόρος ονομάζεται οτιδήποτε μπορεί να περιγραφεί με RDF εκφράσεις. Μπορεί να είναι μια σελίδα του διαδικτύου π.χ. http://www.w3.org/Overview.html, ένα τμήμα μιας σελίδας π.χ. ένα συγκεκριμένο στοιχείο ενός XML εγγράφου ή ακόμα και ένα σύνολο σελίδων. Πόροι ονομάζονται και τα αντικείμενα τα οποία δεν είναι προσπελάσιμα μέσω του διαδικτύου, όπως για παράδειγμα ένα τυπωμένο βιβλίο ή ένας πίνακας ζωγραφικής. Σε κάθε πόρο αποδίδεται ένα μοναδικό URI (Uniform Resource Identifier) [LFIM98] στο

οποίο μπορεί να προστεθεί και μια άγκυρα (anchor ids). Για παράδειγμα το URI http://www.ics.forth.gr/proj/isst/RDF αποδίδεται στην σελίδα με την αντίστοιχη ηλεκτρονική διεύθυνση και το URI http://www.ics.forth.gr/RDF/Dublin_Core.rdf#Title αποδίδεται στην ιδιότητα *Title* που ορίζεται στο αρχείο http://www…/Dublin_Core.rdf. Τα URIs αποτελούν τα αναγνωριστικά των πόρων.

Ιδιότητα, όπως αναφέρθηκε και παραπάνω, είναι είτε ένα γνώρισμα που χρησιμοποιείται για να περιγράψει ένα πόρο ή μια σχέση μεταξύ πόρων. Μια ιδιότητα έχει καθορισμένη σημασιολογία και πιθανόν έχει κάποιο πεδίο ορισμού, πεδίο τιμών καθώς και συσχετίσεις με άλλες ιδιότητες. Οι ιδιότητες είναι και αυτές πόροι, άρα έχουν μοναδικά URIs και είναι δυνατόν να περιγράφονται από RDF εκφράσεις.

Δήλωση ονομάζεται η τριάδα *πόρος – ιδιότητα – τιμή ιδιότητας*. Τα τρία αυτά μέρη μίας δήλωσης ονομάζονται θέμα (subject), κατηγόρημα (predicate) και αντικείμενο (object). Το αντικείμενο μίας δήλωσης μπορεί να είναι είτε πόρος, είτε ένα Literal. Ο πόρος μπορεί και αυτός με την σειρά του να έχει άλλες ιδιότητες. Ο τύπος Literal συμπεριλαμβάνει τα αλφαριθμητικά και τους άλλους ατομικούς (primitive) τύπους δεδομένων που ορίζονται στο XML Schema [BM00].

Συνεπώς, το μοντέλο δεδομένων του RDF βασίζεται στην δημιουργία τριάδων (πόρος – ιδιότητα – τιμή ιδιότητας). Ένα σύνολο ιδιοτήτων που αποδίδονται στον ίδιο πόρο ονομάζεται *περιγραφή* του πόρου αυτού. Το απλό τριαδικό μοντέλο του RDF συντελεί στην δημιουργία μεταδεδομένων με σημασιολογία κατανοητή από τις μηχανές.

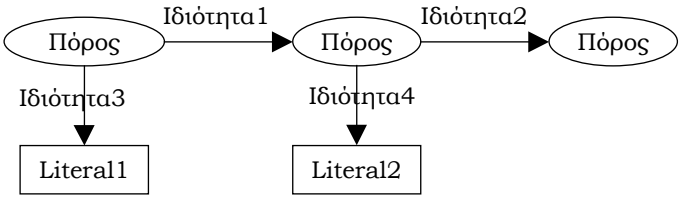
Για την αναπαράσταση των RDF περιγραφών μπορούν να χρησιμοποιηθούν τρία διαφορετικά εργαλεία:

- Κατευθυνόμενοι γράφοι με ετικέτες οι οποίοι εκφράζουν την σημασιολογία των RDF περιγραφών.
- Μοντέλο δηλώσεων-τριάδων που επίσης εκφράζει την σημασιολογία των περιγραφών.
- RDF/XML έγγραφα τα οποία αναφέρονται στην σύνταξη των περιγραφών. Η σύνταξη τους περιγράφεται στο RDF M&S [LS99].

Στην συνέχεια θα παρουσιάσουμε μερικές RDF περιγραφές και θα χρησιμοποιήσουμε κατευθυνόμενους γράφους για την αναπαράσταση τους δεδομένου ότι εκφράζουν καθαρότερα την σημασιολογία των RDF περιγραφών.

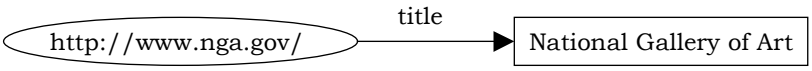
2.3.1 RDF Γράφοι

Οι RDF γράφοι είναι κατευθυνόμενοι γράφοι με ετικέτες. Οι κόμβοι αντιστοιχούν στους πόρους ή σε Literal και οι ακμές στις ιδιότητες. Οι κόμβοι που αντιστοιχούν σε πόρους απεικονίζονται με ελλείψεις ενώ οι κόμβοι που αντιστοιχούν σε literal απεικονίζονται με παραλληλόγραμμα. Τα αναγνωριστικά τόσο των κόμβων όσο και των ακμών είναι τα URIs. Δεδομένου ότι ένας πόρος μπορεί να εμφανίζεται σε πολλές περιγραφές, η μοναδικότητα των URIs κάνει εφικτό τον συνδυασμό μεταδεδομένων που αναφέρονται στον ίδιο πόρο και βρίσκονται σε διαφορετικά αρχεία ή σε διαφορετικές περιγραφές. Στην εικόνα 2.1 απεικονίζεται μια γενική RDF περιγραφή.



Εικόνα 2.1. Γενικός RDF γράφος.

Στην συνέχεια θα αναπαραστήσουμε συγκεκριμένα RDF μεταδεδομένα. Έστω η παρακάτω πρόταση: *"Ο τίτλος (title) της σελίδας με URI http://www.nga.gov/ είναι National Gallery of Art.* Ο γράφος που αντιστοιχεί σ' αυτήν την πρόταση είναι ο παρακάτω:

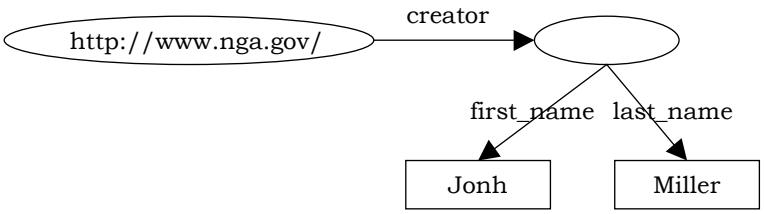


Εικόνα 2.2. RDF γράφος με τιμή ιδιότητας ένα ατομικό τύπο.

Ο παραπάνω γράφος μπορεί επίσης να ερμηνευτεί και ως εξής: "*Η σελίδα http://www.nga.gov/ έχει τίτλο National Gallery of Art*". Γενικά μια δήλωση ερμηνεύεται ως εξής: <θέμα> έχει <κατηγόρημα> <αντικείμενο>.

Έστω η παρακάτω πρόταση *"Το άτομο με όνομα John και επίθετο Miller είναι ο δημιουργός (creator) της σελίδας http://www.nga.gov/"*. Ο γράφος που αντιστοιχεί σ' αυτήν την πρόταση απεικονίζεται στην εικόνα 2.3. Για να μπορέσουμε να περιγράψουμε το άτομο με όνομα Pablo και επίθετο Picasso πρέπει να εισάγουμε ένα καινούριο πόρο (και όχι literal) ώστε να μπορούμε να του αποδώσουμε ιδιότητες. Το αναγνωριστικό του πόρου αυτού είναι άγνωστο (anonymous resource) γι' αυτό χρησιμοποιείται μια κενή

έλλειψη για την αναπαράστασή του. Σε επίπεδο εφαρμογής θα πρέπει να αποδοθεί κάποιο μοναδικό αναγνωριστικό σε όλους τους ανώνυμους πόρους ώστε να μπορούν να επεξεργαστούν αλλά και να διαφοροποιηθούν μεταξύ τους.



Εικόνα 2.3. RDF γράφος με τιμή ιδιότητας ένα πόρο.

Στην συνέχεια θα περιγράψουμε ένα σύνολο πόρων (συμπεριλαμβανομένων και ιδιοτήτων) που ορίζονται στο μοντέλο δεδομένων του RDF επειδή η χρήση τους προβλέπεται να είναι ευρεία. Στις επόμενες παραγράφους οι πόροι αυτοί απεικονίζονται με μαύρα πλάγια γράμματα. Επιτρέπουν την αναπαράσταση πιο πολύπλοκων εκφράσεων. Παρατηρούμε ότι για την αναπαράσταση των εκφράσεων αυτών χρησιμοποιείται το τριαδικό μοντέλο του RDF που περιγράψαμε.

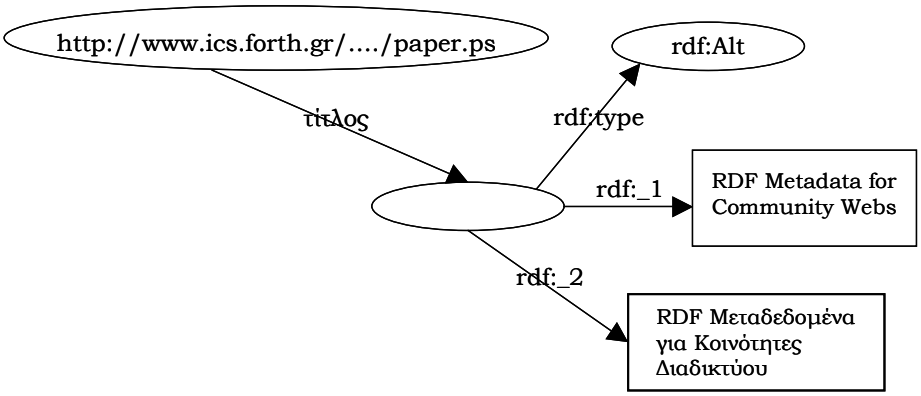
2.3.2 RDF Συλλογές

Οι τιμές μιας ιδιότητας συχνά χρειάζεται να είναι μια συλλογή από πόρους ή αλφαριθμητικά π.χ. όταν θέλουμε να αναφέρουμε του συγγραφείς μιας δημοσίευσης. Για το σκοπό αυτό ορίζονται οι RDF συλλογές (RDF Containers) στο RDF μοντέλο δεδομένων. Πρόκειται για μια ειδική κατηγορία πόρων οι οποίοι περιέχουν συλλογές από πόρους ή literals. Ορίζονται 3 είδη RDF συλλογών: *Bag*, *Sequence* and *Alternative*.

Bag καλείται ένα πολυσύνολο από πόρους ή literals. Χρησιμοποιείται για να δηλώσει ότι μία ιδιότητα έχει πολλαπλές τιμές και ότι η σειρά των τιμών αυτών δεν έχει σημασία. Ένα πολυσύνολο μπορεί να περιέχει τους μαθητές μίας τάξης εφόσον δεν μας ενδιαφέρει η σειρά. Τα πολυσύνολα μπορούν να έχουν περιέχουν πολλές φορές τα ίδια μέλη.

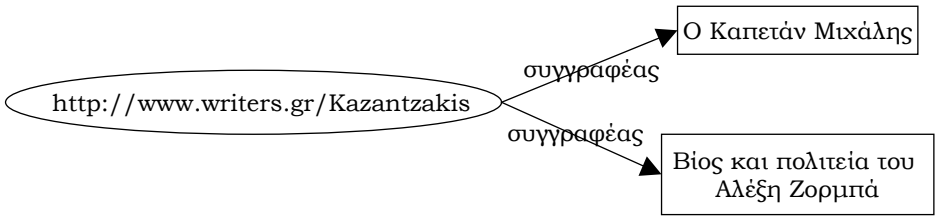
Sequence (ακολουθία) καλείται μια λίστα από πόρους ή literals. Χρησιμοποιείται για να δηλώσει ότι μία ιδιότητα έχει πολλαπλές τιμές και ότι η σειρά των τιμών αυτών είναι σημαντική. Για παράδειγμα μπορούμε να δημιουργήσουμε μια ακολουθία που να περιέχει τους μήνες του χρόνου εφόσον σ' αυτήν την περίπτωση έχει σημασία η σειρά. Οι ακολουθίες μπορούν να περιέχουν πολλές φορές τα ίδια μέλη.

Alternative (παραλλαγή) καλείται μια συλλογή από πόρους ή literals. Τα μέλη της συλλογής αντιπροσωπεύουν εναλλακτικές τιμές για την τιμή μιας ιδιότητας. Για παράδειγμα μια παραλλαγή μπορεί να περιέχει τον τίτλο ενός βιβλίου μεταφρασμένο σε διάφορες γλώσσες ή τις διαφορετικές διευθύνσεις από τις οποίες ένας πληροφοριακός πόρος είναι δυνατόν να ανακτηθεί. Εφαρμογές οι οποίες διαχειρίζονται παραλλαγές 'γνωρίζουν' ότι μπορούν να επιλέξουν οποιαδήποτε από τα μέλη της λίστας σαν τιμή της ιδιότητας. Το πρώτο μέλος μίας παραλλαγής αποτελεί την προκαθορισμένη τιμή, αλλά βέβαια μπορεί να επιλεχτεί οποιαδήποτε άλλο μέλος της παραλλαγής σαν τιμή της ιδιότητας.



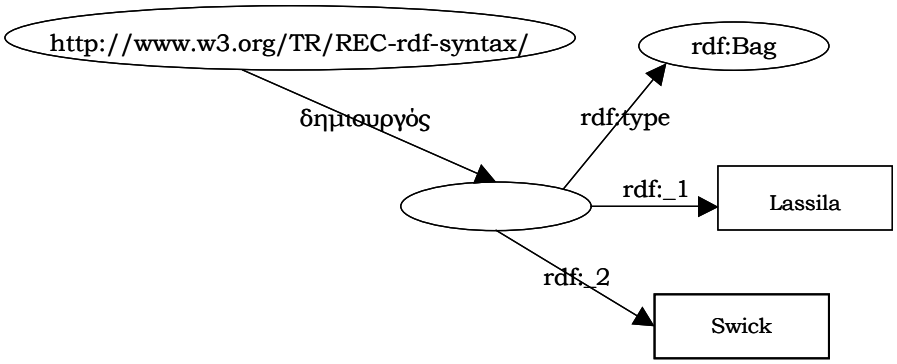
Εικόνα 2.4. RDF παραλλαγή.

Στην εικόνα 2.4 απεικονίζεται μια παραλλαγή που έχει σαν μέλη της τον τίτλο μίας δημοσίευσης μεταφρασμένο σε δύο γλώσσες. Παρατηρούμε ότι για την αναπαράσταση μιας συλλογής δημιουργούμε ένα νέο πόρο ο οποίος πρέπει να ανήκει σε ένα από τα 3 είδη συλλογών που περιγράφτηκαν. Στην συγκεκριμένη περίπτωση ανήκει τύπο *παραλλαγή* (rdf:Alt). Η ιδιότητα ***rdf:type*** χρησιμοποιείται για να δηλώσει την κλάση που ανήκει ένας πόρος. Τα μέλη μιας συλλογής αποδίδονται στον πόρο που αντιπροσωπεύει την συλλογή με ένα σύνολο από ιδιότητες ***_1, _2, _3*** … οι οποίες έχουν οριστεί γι΄ αυτόν τον σκοπό. Μια συλλογή μπορεί να έχει και άλλες ιδιότητες που την περιγράφουν όπως οποιοδήποτε άλλος πόρος.



Εικόνα 2.5. RDF γράφος με πλειότιμες ιδιότητες.

Η λειτουργικότητα τόσο των ακολουθιών όσο και των παραλλαγών είναι εμφανής. Από τα παραπάνω όμως δεν είναι φανερή η διαφορά ανάμεσα σε πολυσύνολα και επαναλαμβανόμενες/πλειότιμες ιδιότητες (χρήση μιας ιδιότητας πολλαπλές φορές). Οι επαναλαμβανόμενες ιδιότητες χρησιμοποιούνται όταν δεν υπάρχει σχέση μεταξύ των διαφορετικών τιμών της ιδιότητας. Έστω ότι έχουμε την πρόταση: *"Ο Καζαντζάκης έχει γράψει τα βιβλία ο Καπετάν Μιχάλης και Βίος και πολιτεία του Αλέξη Ζορμπά"*. Τα παραπάνω βιβλία δεν έχουν κάποια ιδιαίτερη σχέση, εκτός ότι έχουν γραφτεί από τον ίδιο συγγραφέα. Η αναπαράσταση αυτής της πρότασης απεικονίζεται στην εικόνα 2.5.

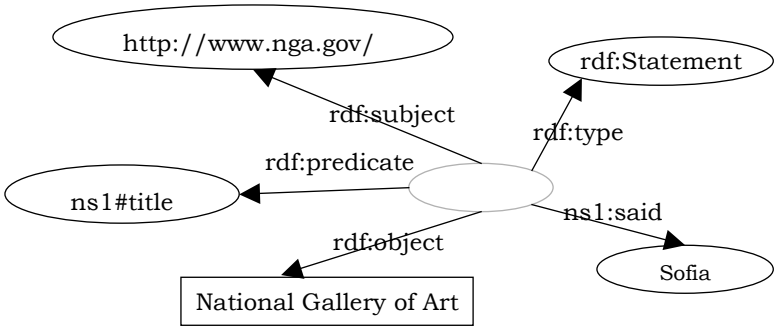


Εικόνα 2.6. RDF γράφος με τιμή ιδιότητας ένα πολυσύνολο.

Τα πολυσύνολα χρησιμοποιούνται όταν υπάρχει ιδιαίτερη σημασιολογία μεταξύ των τιμών της ιδιότητας. Έστω η πρόταση *"Lassila και Swick έχουν γράψει το RDF M&S"*. Για να δείξουμε ότι το κείμενο είναι αποτέλεσμα ομαδικής δουλειάς και όχι του καθ' ενός ξεχωριστά χρησιμοποιούμε την ιδιότητα *δημιουργός* με τιμή ένα πολυσύνολο (εικόνα 2.6).

2.3.3 Υποστασιοποιημένες Δηλώσεις (Reified Statements)

Μια από τις βασικές δυνατότητες που παρέχει το μοντέλο δεδομένων του RDF είναι περιγραφή των ίδιων των μεταδεδομένων. Για να περιγράψουμε μια δήλωση πρέπει να δημιουργήσουμε την υποστασιοποιημένη δήλωσή της. Αυτή η διαδικασία στην περιοχή της αναπαράστασης γνώσης ονομάζεται υποστασιοποίηση (reification). Μια υποστασιοποιημένη δήλωση είναι ένας νέος πόρος με τέσσερις ιδιότητες ***rdf:subject, rdf:predicate, rdf:object*** και ***rdf:type***. Η τιμή της ιδιότητας rdf:subject είναι το αντίστοιχο θέμα (subject) της δήλωσης που υποστασιοποιείται. Το ίδιο ισχύει και για τις τιμές των ιδιοτήτων predicate και object. Η τιμή της ιδιότητας *rdf:type* για κάθε πόρο που αντιστοιχεί σε υποστασιοποιημένη δήλωση είναι η κλάση rdf:Statement. Η υποστασιοποιημένη δήλωση που αντιστοιχεί στην πρόταση *"Ο τίτλος (title) της σελίδας με URI http://www.nga.gov/ είναι National Gallery of Art"* απεικονίζεται στη εικόνα 2.7 (Να σημειώσουμε ότι στην εικόνα 2.7 στην υποστασιοποιημένη δήλωση αποδίδεται και η ιδιότητα ns1:said.)



Εικόνα 2.7. Υποστασιοποιημένη δήλωση.

Ο πόρος που αντιπροσωπεύει την υποστασιοποιημένη δήλωση μπορεί να χρησιμοποιηθεί τόσο σαν θέμα όσο και σαν αντικείμενο μίας άλλης δήλωσης. Η αναπαράσταση της *δήλωση "Η Σοφία λέει ότι ο τίτλος (title) της σελίδας με URI http://www.nga.gov/ είναι National Gallery of Art"* παρουσιάζεται στην εικόνα 2.7. Η υποστασιοποιημένη δήλωση δεν υποκαθιστά την δήλωση. Σε ένα γράφο μπορεί να υπάρχει κάθε μια από αυτές ή και οι δύο. Πρέπει να τονιστεί ότι ένας RDF γράφος εκφράζει ένα γεγονός μόνο όταν η δήλωση που το εκφράζει περιέχεται στο γράφο ανεξάρτητα αν υπάρχει ή όχι η υποστασιοποιημένη δήλωση.

2.4 Γλώσσα Περιγραφής RDF Σχημάτων (RDFS)

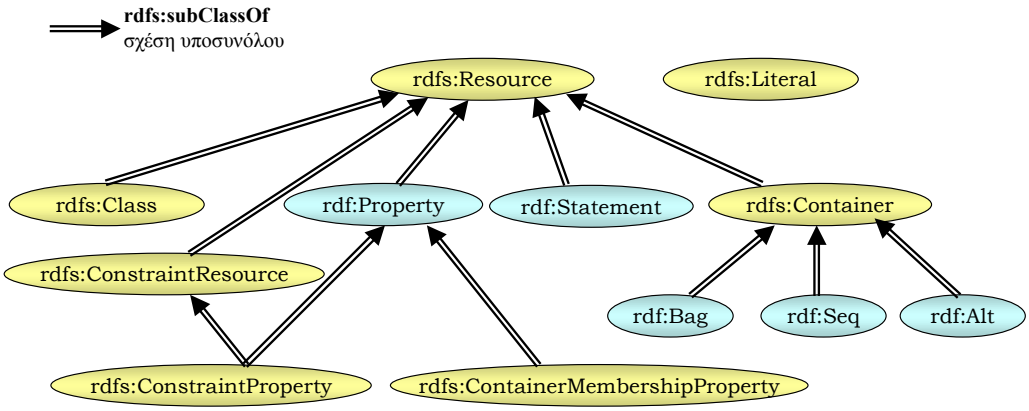
Το μοντέλο δεδομένων του RDF είναι ένα απλό μοντέλο για την περιγραφή των πόρων. Δεν παρέχει μηχανισμούς για την δήλωση και περιγραφή των ιδιοτήτων και κλάσεων που χρησιμοποιούνται στις RDF περιγραφές. Για το σκοπό αυτό έχει προταθεί μια γλώσσα ορισμού σχημάτων (συνόλων κλάσεων και ιδιοτήτων), η RDF Schema Specification Language (RDFS). Η RDFS παρέχει ένα σύνολο από βασικούς τύπους (κλάσεις και ιδιότητες) που χρησιμοποιούνται για την δημιουργία των RDF σχημάτων. Για παράδειγμα, ορίζει ιδιότητες που εκφράζουν περιορισμούς για το πεδίο ορισμού και το πεδίο τιμών των RDF ιδιοτήτων. Καθώς επίσης και πόρους (ιδιότητες) που δηλώνουν σχέσεις μεταξύ κλάσεων ή μεταξύ ιδιοτήτων. Τα RDF σχήματα ορίζονται χρησιμοποιώντας τους τύπους που παρέχει η RDFS και βασίζονται στο RDF μοντέλο δεδομένων. Τα σχήματα δηλαδή είναι και αυτά RDF μεταδεδομένα. Το γεγονός αυτό παρέχει ευελιξία στις εφαρμογές δεδομένου ότι δεν χρειάζεται να γνωρίζουν εκ των προτέρων την σημασιολογία των σχημάτων στα οποία στηρίζονται τα μεταδεδομένα που αναλύουν αλλά μπορούν να την εξάγουν. Τα RDF σχήματα παρέχουν δυνατότητα ερμηνείας των μεταδεδομένων και έμμεσα επηρεάζουν την δομή των μεταδεδομένων λόγω των περιορισμών που υπάρχουν στο πεδίο τιμών και ορισμού των ιδιοτήτων (rdfs:domain, rdfs:range). Στην συνέχεια θα περιγράψουμε το σύστημα τύπων που ορίζει η RDFS καθώς επίσης και πόρους που ορίζονται στο RDF M&S και περιγράφονται στο RDF σχήμα[9] και χρησιμοποιούνται για τον ορισμό σχημάτων.

2.4.1 Βασικές RDF/S Κλάσεις

Παρακάτω αναλύονται οι βασικές κλάσεις που είτε ορίζονται είτε περιγράφονται στο RDF σχήμα. Οι κλάσεις σχετίζονται μεταξύ τους με σχέσεις υποσυνόλου/υπερσυνόλου. Η ιεραρχία κλάσεων που δημιουργείται απεικονίζεται στην εικόνα 2.8. Κορυφή της ιεραρχίας είναι η κλάση rdfs:Resource.

Rdfs:Resource: Μέλη της κλάσης αυτής είναι οτιδήποτε μπορεί να περιγραφτεί με RDF δηλώσεις. Η rdfs:Resource αντιστοιχεί στο σύνολο Πόροι που περιγράφτηκε παραπάνω.

[9] Στο RDF σχήμα ορίζονται οι κατασκευές της RDFS και περιγράφονται πόροι που έχουν οριστεί στο κείμενο RDF M&S.



Εικόνα 2.8. Ιεραρχία RDF/S κλάσεων.

Rdfs:Class: Έχει αντίστοιχη σημασιολογία με τις γενικές έννοιες *Τύπος*, *Κατηγορία* ή *Κλάση* στις οντο-κεντρικές γλώσσες όπως η Java. Κάθε πόρος που αντιπροσωπεύει μια RDF κλάση πρέπει να δηλώνεται μέλος της rdfs:Class, δηλαδή να του αποδίδεται η ιδιότητα *rdf:type* με τιμή *rdfs:Class*. Οι RDF κλάσεις μπορούν να αναπαριστάνουν σχεδόν οτιδήποτε π.χ. σελίδες του διαδικτύου, ανθρώπους, βιβλία κτλ.

Rdfs:ConstraintResource: Η κλάση αυτή είναι υποκλάση της rdfs:Resource. Μέλη της είναι οι κλάσεις και ιδιότητες που χρησιμοποιούνται για να εκφράζουν περιορισμούς. Η κλάση αυτή δίνει την δυνατότητα στους RDF επεξεργαστές να κρίνουν αν είναι σε θέση να ελέγξουν την συνέπεια των RDF μεταδεδομένων. Δεδομένου ότι δεν υπάρχει μηχανισμός που να επιτρέπει την ανακάλυψη και την σωστή ερμηνεία περιορισμών που δεν έχουν οριστεί στην RDFS, όπως η πληθικότητα (cardinality) μιας ιδιότητας, όταν ένας RDF επεξεργαστής διαπιστώσει ότι στα μεταδεδομένα που αναλύει υπάρχουν άγνωστα γι΄ αυτόν μέλη της κλάσης rdfs:ConstraintResource τότε δεν είναι σε θέση να αποφανθεί για την συνέπεια τους.

Rdfs:ConstraintProperty: Είναι υποκλάση της rdfs:ConstraintResource και της rdf:Property. Μέλη της είναι όλες οι ιδιότητες που δηλώνουν κάποιο περιορισμό. Οι ιδιότητες *rdfs:range* και *rdfs:domain* που θα περιγραφούν παρακάτω και περιορίζουν το πεδίο τιμών και το πεδίο ορισμού μιας ιδιότητας ανήκουν στη κλάση αυτή.

Rdfs:Container: Η κλάση αυτή αντιπροσωπεύει το σύνολο των *συλλογών πόρων* (containers). Υποκλάσεις της είναι οι κλάσεις rdf:Bag, rdf:Seq και rdf:Alt.

Rdfs:ContainerMembershipProperty: Περιέχει τις μη αριθμήσιμες ιδιότητες rdf:_1, rdf:_2… οι οποίες δηλώνουν τα μέλη μιας συλλογής. Είναι υποκλάση της rdf:Property.

Rdfs:Literal: Αντιπροσωπεύει το σύνολο Literal. Περιέχει ατομικές τιμές όπως αλφαριθμητικά, ακεραίους.

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

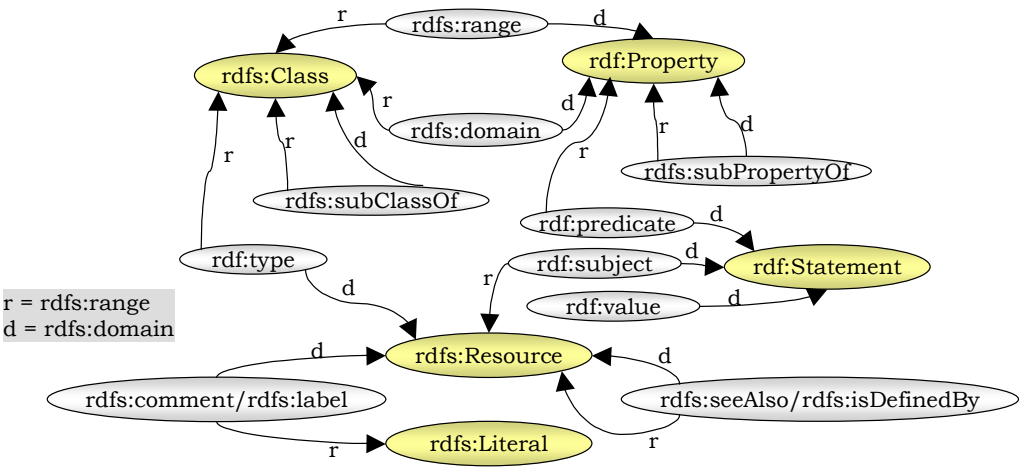
rdf:Property: Αναπαριστάνει το υποσύνολο των πόρων που είναι ιδιότητες.

Rdf:Bag rdf:Seq και **rdf:Alt:** Αναπαριστάνουν τα σύνολα Bag, Sequences και Alternative. Είναι υποκλάσεις της rdfs:Container.

Rdf:Statement: Αναπαριστάνει το σύνολο των υποστασιοποιημένων δηλώσεων.

2.4.2 Βασικές RDF/S Ιδιότητες

Παρακάτω αναλύονται οι βασικές ιδιότητες που είτε ορίζονται είτε περιγράφονται στο RDF σχήμα. Στην εικόνα 2.9 απεικονίζονται οι περιορισμοί που δηλώνονται στο RDF Σχήμα για τις ιδιότητες.



Εικόνα 2.9. Περιορισμοί που δηλώνονται στο RDF Σχήμα.

Rdf:type: Δηλώνει ότι ένας πόρος είναι μέλος μιας κλάσης. Στο RDF δεν υπάρχει η έννοια της αυστηρής κληρονομικότητας των ιδιοτήτων όπως συμβαίνει σε οντο-κεντρικά συστήματα. Αντίθετα οι κλάσεις δεν επιβάλλουν καμιά δομή στα μέλη τους. Ένας πόρος μπορεί να είναι μέλος πολλαπλών κλάσεων. Στο κείμενο της RDFS αναφέρεται ότι η τιμή της ιδιότητας rdf:type είναι ένας πόρος που πρέπει να είναι μέλος της rdfs:Class, δηλαδή οποιαδήποτε RDF κλάση. Αυτό όμως δεν είναι απόλυτα σωστό δεδομένου ότι η rdfs:Literal είναι κλάση αλλά δεν μπορεί να αποτελεί τιμή της ιδιότητας rdf:type. Οι κλάσεις που ορίζονται στην RDFS, άλλα και οι κλάσεις που ορίζονται στα RDF σχήματα

έχουν την ιδιότητα rdf:type με τιμή rdfs:Class. Αντίστοιχα, η ιδιότητα rdf:type αποδίδεται τόσο στις ιδιότητες που ορίζονται στην RDFS όσο και στις απλές ιδιότητες με τιμή rdf:Property.

Ένα χαρακτηριστικό του RDF το οποίο δεν δηλώνεται ρητά είναι η δυνατότητα που παρέχει για δημιουργία μετα-σχημάτων. Το πεδίο ορισμού της ιδιότητας rdf:type είναι η κλάση rdfs:Resource. Αυτό σημαίνει ότι η ιδιότητα rdf:type αποδίδεται και σε κλάσεις, δηλαδή μια κλάση μπορεί να είναι μέλος μιας άλλης κλάσης. Συνεπώς, μπορούν να δημιουργηθούν μετα-σχήματα απεριόριστων επιπέδων. Η ανάγκη ύπαρξης μεταδεδομένων πολλαπλών επιπέδων αναλύεται στην δημοσίευση [KG97]. Για παράδειγμα, σε ένα σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων υπάρχουν δεδομένα τριών επιπέδων, τα απλά δεδομένα, οι πίνακες της εφαρμογής π.χ. *Σελίδα*, *Δημιουργός* και τέλος πίνακες όπως Table, Relations, Keys που αποτελούν το μετα-μοντέλο της βάσης. Το RDF παρόλο που παρέχει την δυνατότητα μοντελοποίησης πολλαπλών επιπέδων δεν εξηγεί πως γίνεται ο διαχωρισμός των επιπέδων και τα είδη των σχέσεων που μπορούν να δημιουργηθούν ανάμεσα τους.

Rdfs:subClassOf: Δηλώνει σχέση υποσυνόλου-υπερσυνόλου μεταξύ κλάσεων. Η ιδιότητα rdfs:subClassOf είναι μεταβατική. Αν μια κλάση Α είναι υποκλάση μιας κλάσης Β η οποία είναι υποκλάση της Γ τότε και η κλάση Α είναι υποκλάση της Γ. Συνεπώς και οι πόροι που είναι μέλη της κλάσης Α θα είναι και μέλη της κλάσης Γ. Τόσος ο πόρος στον οποίο εφαρμόζεται η ιδιότητα rdfs:subClassOf όσο και η τιμή της ιδιότητας πρέπει να είναι κλάσεις. Μια κλάση μπορεί να είναι υποκλάση πολλών κλάσεων. Δεν μπορεί όμως να δηλωθεί σαν υποκλάση του εαυτού της ή κάποιας από τις υποκλάσεις της. Δεν επιτρέπεται δηλαδή δημιουργία κύκλων στην ιεραρχία των υποκλάσεων.

Rdfs:domain: Είναι μέλος της κλάσης rdfs:ConstraintProperty και χρησιμοποιείται για να ορίσει τις κλάσεις στις οποίες μπορεί να εφαρμοστεί μια ιδιότητα. Μια ιδιότητα μπορεί να έχει μηδέν, μία ή περισσότερες rdfs:domain ιδιότητες. Στην περίπτωση που δεν ορίζεται η ιδιότητα rdfs:domain για μια ιδιότητα, τότε η ιδιότητα μπορεί να αποδοθεί σε οποιοδήποτε πόρο. Αν αποδίδονται παραπάνω από μία rdfs:domain ιδιότητες σε μια ιδιότητα τότε αυτή μπορεί να εφαρμοστεί σε οποιοδήποτε μέλος των κλάσεων που αποτελούν τιμές της rdfs:domain.

Πρόβλημα 1: Πολλαπλές rdfs:domain ιδιότητες: Το γεγονός ότι μια ιδιότητα μπορεί να έχει πολλαπλές rdfs:domain ιδιότητες οι οποίες είναι διασκορπισμένες σε διάφορα αρχεία δημιουργεί ασάφεια στην σημασιολογία των ιδιοτήτων. Έστω ότι σε ένα σχήμα ορίζεται η ιδιότητα *name*. Σε ένα σχήμα Α ορίζεται σαν πεδίο ορισμού της ιδιότητας

name η κλάση *Person* και σε ένα σχήμα Β η κλάση *Company*. Το γεγονός ότι αποδίδεται διαφορετική τιμή στην ιδιότητα rdfs:domain σημαίνει ότι η σημασιολογία της ιδιότητας σε κάθε ένα από τα παραπάνω σχήματα είναι διαφορετική. Στο σχήμα Α η ιδιότητα *name* δηλώνει το όνομα ενός ανθρώπου ενώ στο σχήμα Β το όνομα μιας εταιρείας. Η σημασιολογία μιας ιδιότητας επεκτείνεται καθώς προστίθεται σ' αυτήν μια καινούρια 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 το γεγονός ότι το πεδίο ορισμού μιας ιδιότητας είναι η ένωση των τιμών της rdfs:domain δεν επιτρέπει την εξαγωγή συμπερασμάτων (inferencing) που αφορούν την κλάση που ανήκει ένας πόρος στον οποίο αποδίδεται η ιδιότητα. Για παράδειγμα αν έχουμε την τριάδα p(x,y) δεν μπορούμε να εξάγουμε την κλάση/κλάσεις που ανήκει ο πόρος x εφόσον το πεδίο ορισμού της ιδιότητας p δεν είναι συγκεκριμένο.

Στα κείμενα του RDF δεν αναφέρεται ποιος είναι ο λόγος να υπάρχει μια ιδιότητα με πολλαπλές rdfs:domain ιδιότητες αντί για πολλαπλές διαφορετικές ιδιότητες. Μια πιθανή εξήγηση είναι ότι αν δημιουργούνταν πολλαπλές διαφορετικές ιδιότητες θα χάνονταν κάποια σχέση ή κοινή σημασιολογία που υπάρχει μεταξύ των ιδιοτήτων. Στην ενότητα 2.5.2.2 θα δείξουμε πως μπορεί να αναπαρασταθεί η σχέση αυτή χωρίς να χρησιμοποιούνται πολλαπλές rdfs:domain 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 I ορίζεται (με την ιδιότητα rdfs:range) να είναι η κλάση Α τότε κάθε τιμή της ιδιότητας I πρέπει να ανήκει στην κλάση Α. Μια ιδιότητα μπορεί να έχει το πολύ μια rdfs:range ιδιότητα. Σε περίπτωση που πρέπει να οριστεί μια ιδιότητα π.χ. 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 μια υπερ-κλάση τους. Είναι δυνατόν σε μια 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 Πόροι (Resources) και Literals.

Πρόβλημα 2: Η τιμή της ιδιότητας rdfs:range για την ιδιότητα rdfs:domain είναι η κλάση rdfs:Class (εικόνα 2.9). Αυτό σημαίνει ότι μια ιδιότητα μπορεί να έχει σαν πεδίο ορισμού οποιαδήποτε κλάση. Όπως και στην περίπτωση της ιδιότητας rdf:type η δήλωση αυτή δε είναι απόλυτα σωστή. Η κλάση rdfs:Literal που ανήκει στην κλάση rdfs:Class δεν μπορεί να είναι τιμή της ιδιότητας rdfs:domain.

Πρόβλημα 3: Μη ρητή δήλωση πεδίου τιμών στο σημείο ορισμού: Η μη ρητή δήλωση του πεδίου τιμών μιας ιδιότητας στο σημείο ορισμού της προκαλεί προβλήματα. Είναι δυνατόν σε σχήματα (διαφορετικά από αυτό όπου ορίζεται η ιδιότητα) να οριστούν διαφορετικά πεδία τιμών για την ιδιότητα. Το γεγονός αυτό δημιουργεί ασάφεια στην σημασιολογία των μεταδεδομένων (βλέπε πρόβλημα 2). Επίσης το γεγονός ότι η ιδιότητα μπορεί να έχει σαν τιμές τόσο πόρους όσο και literals κάνει πιο πολύπλοκη την επεξεργασία (αποθήκευση, επερώτηση) των περιγραφών. Το βασικό όμως 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.

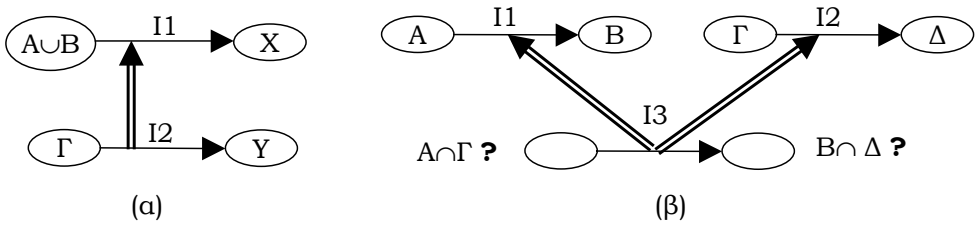
Η μοναδικότητα της ιδιότητας rdfs:range έχει συζητηθεί πολύ τελευταία[10]. Αναφέρεται ότι η σημασιολογία της rdfs:range πρέπει να είναι η εξής: Δεδομένου μιας δήλωσης *p(y,x)* και γνωρίζοντας μια ιδιότητα rdfs:range της ιδιότητας p *rdfs:range(p,s)* εξάγουμε ότι η τιμή της ιδιότητας p ανήκει στην κλάση s δηλαδή *rdf:type(x,s)*. Η παραπάνω ερμηνεία της rdfs:range δεν επιβάλει την μοναδικότητα της rdfs:range. Επιτρέπει την ύπαρξη πολλαπλών rdfs:range ιδιοτήτων και πεδίο τιμών της ιδιότητας θεωρείται η τομή των τιμών της rdfs:range. Η θεώρηση ότι το πεδίο τιμών μιας ιδιότητας είναι η τομή των τιμών (κλάσεων) της rdfs:range ευνοεί την εξαγωγή συμπερασμάτων. Μάλιστα τα συμπεράσματα αυτά δεν μεταβάλλονται όταν προκύπτει μια νέα τιμή της ιδιότητας rdfs:range. Έστω η ιδιότητα *creates* με πεδίο ορισμού την κλάση *Painting*. Για όλες τις δηλώσεις *creates(x,y)* μπορούμε να συμπεράνουμε ότι το x ανήκει στην κλάση *Painting*. Αν στην συνέχεια προκύψει μια νέα rdfs:range τιμή π.χ. *Sculpture* το πεδίο τιμών της ιδιότητας *creates* θα είναι η τομή των *Painting* και *Sculpture*. Τα συμπεράσματα που είχαν εξαχθεί εξακολουθούν να ισχύουν. Όμως η παραπάνω θεώρηση δημιουργεί πρόβλημα στον έλεγχο της εγκυρότητας των μεταδεδομένων. Για παράδειγμα οι δηλώσεις *p(x,y)* που πριν ήταν έγκυρες εφόσον τα y ανήκουν στην κλάση *Painting* τώρα είναι άκυρες αν τα y δεν ανήκουν και στην κλάση *Sculpture*. Πρέπει δηλαδή να ελεγχθεί ξανά η εγκυρότητα όλων των δηλώσεων p(x,y).

[10] http://lists.w3.org/Archives/Public/www-rdf-interest/2000Sep/0107.html

rdfs:subPropertyOf: Δηλώνει σχέση εξειδίκευσης μεταξύ ιδιοτήτων. Μια ιδιότητα μπορεί να είναι εξειδίκευση καμιάς, μιας ή περισσοτέρων ιδιοτήτων. Αν μια ιδιότητα I1 είναι υποιδιότητα μιας πιο γενικής ιδιότητας I2 και σε ένα πόρο A αποδίδεται η ιδιότητα I1 με τιμή B, τότε ο πόρος A έχει και την ιδιότητα I2 με τιμή B. Για παράδειγμα, η ιδιότητα *ζωγράφος* ορίζεται σαν υποιδιότητα της ιδιότητας *δημιουργός*. Αν η Guernica έχει *ζωγράφο* τον Picasso, τότε θα έχει και *δημιουργό* τον Picasso. Τόσο ο πόρος στον οποίο εφαρμόζεται η ιδιότητα rdfs:subPropertyOf όσο και η τιμή της ιδιότητας πρέπει να είναι ιδιότητες. Μια ιδιότητα δεν μπορεί να δηλωθεί σαν υποιδιότητα του εαυτού της ή κάποιας από τις υποιδιότητες της.

Πρόβλημα 4: Απλή εξειδίκευση ιδιοτήτων: Στα κείμενα του RDF δεν αναφέρεται κάποιος περιορισμός που πρέπει να ισχύει για το πεδίο τιμών και ορισμού μιας υποιδιότητας. Όμως για να είναι μια υποιδιότητα υποσύνολο της υπερ-ιδιότητάς της επιβάλλεται να ισχύει η σχέση υποσυνόλου μεταξύ των πεδίων ορισμού και τιμών τους αντίστοιχα [ASC97].

Αναφέραμε παραπάνω ότι μια ιδιότητα μπορεί να έχει πολλαπλές rdfs:domain ιδιότητες. Έστω η ιδιότητα I1 με πεδίο ορισμού την ένωση των κλάσεων A και B (εικόνα 2.10(α)). Η υποιδιότητα I2 έχει πεδίο ορισμού την κλάση Γ. Αν η κλάση Γ είναι υποκλάση είτε της A είτε/και της B τότε ο περιορισμός που επιβάλλαμε πληρείται. Το ίδιο ισχύει και όταν η Γ είναι υποκλάση της ένωσης των A και B (όχι της κάθε μιας χωριστά). Στο RDF δεν ορίζεται η έννοια της ένωσης δύο κλάσεων. Άρα η παραπάνω περίπτωση δεν μπορεί να εκφραστεί στο RDF οπότε θα θεωρούνταν μη έγκυρη. Η παραπάνω περίπτωση δείχνει ένα ακόμα πρόβλημα που δημιουργεί η δυνατότητα απόδοσης πολλαπλών rdfs:domain ιδιοτήτων.



Εικόνα 2.10.(α) Σχέση μεταξύ πεδίων ορισμού/τιμών ιδιότητας- υποιδιότητας. (β) Κληρονομικότητα πεδίου ορισμού /τιμών για ιδιότητες με πολλαπλές υπερ-ιδιότητες.

Επίσης στο RDF δεν γίνεται αναφορά για το θέμα της κληρονομικότητας των πεδίων ορισμού και τιμών μιας ιδιότητας στις υποιδιότητές της. Σε περίπτωση που το

πεδίο ορισμού/τιμών μιας υποιδιότητας δεν ορίζεται είναι λογικό να κληρονομεί το αντίστοιχο της υπερ-ιδιότητας.

Πρόβλημα 5: Πολλαπλή εξειδίκευση ιδιοτήτων: Όταν μια ιδιότητα είναι υποιδιότητα 2 ή περισσοτέρων ιδιοτήτων και αυτές δεν συνδέονται μεταξύ τους με την σχέση subPropertyOf τότε το πεδίο ορισμού/τιμών της (αν δεν έχουν οριστεί) θα πρέπει να είναι η τομή των πεδίων ορισμού/τιμών των υπερ-ιδιοτήτων (βλέπε εικόνα 2.10 (β)). Στο RDF όμως δε ορίζεται η τομή δύο κλάσεων. Αν οι κλάσεις είναι ίδιες τότε η τομή τους είναι η ίδια η κλάση. Αν δύο κλάσεις συνδέονται με σχέση υποσυνόλου (subClassOf) τότε η τομή τους θα είναι η υποκλάση. Διαφορετικά θα μπορούσαμε να δεχτούμε σαν πεδίο ορισμού/τιμών της ιδιότητας τις κοινές υποκλάσεις των πεδίων ορισμού/τιμών των υπερ-ιδιοτήτων της.

Rdfs:seeAlso: Δηλώνει ένα πόρο (π.χ. ένα έγγραφο) ο οποίος περιέχει 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 rdsf: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: Είναι υποιδιότητα της rdfs:seeAlso. Δηλώνει τον πόρο στον οποίο ορίζεται ο πόρος που εφαρμόζεται η ιδιότητα rdfs:isDefinedBy. Το πεδίο τιμών και το πεδίο ορισμού της ιδιότητας 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 (βλέπε 2.4.3.1) που υπάρχουν σε ένα αρχείο δηλώνουν το URI στο οποίο ορίζονται οι σχήμα κατασκευές, υπάρχουν περιπτώσεις που χρειάζεται 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 namespace και τα στοιχεία που δηλώνονται σ' αυτό δεν έχουν άμεση σχέση.

Rdfs:comment και rdfs:label: Οι ιδιότητες rdfs:comment και rdfs:label παρέχουν περιγραφή για τους πόρους σε μορφή κειμένου. Συμβάλουν στην μεγαλύτερη κατανόηση της σημασιολογίας των πόρων από τον άνθρωπο. Χρησιμοποιούνται ιδιαίτερα για την περιγραφή σχήμα κατασκευών για να διευκολύνουν τους χρήστες στην γρήγορη

κατανόηση νέων σχημάτων και στη διαπίστωση της καταλληλότητας αυτών για τις ανάγκες τους. Συγκεκριμένα, η ιδιότητα *rdfs:comment* παρέχει δυνατότητα απόδοσης σχολίων. Η ιδιότητα *rdfs:label* αποδίδει στους πόρους ονόματα κατανοητά από τους χρήστες. Οι περιγραφές μπορούν να αναπαρασταθούν σε πολλές γλώσσες χρησιμοποιώντας το XML κατηγόρημα xml:lang.

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

rdf:predicate, rdf:subject, rdf:object: Αντιπροσωπεύουν τις ιδιότητες κατηγόρημα (predicate), θέμα (subject) και αντικείμενο (object) και χρησιμοποιούνται για να σχηματίσουν την υποστασιοποιημένη δήλωση.

Rdf:value: Χρησιμοποιείται στην αναπαράσταση σχέσεων πληθικότητας μεγαλύτερης από 2, για να δηλώσει την κύρια τιμή μιας ιδιότητας (βλέπε RDF M&S).

2.4.3 Χώροι ονοματοδοσίας XML και RDF

Σ' αυτήν την ενότητα θα περιγράψουμε τον μηχανισμό ονοματοδοσίας XML και θα αναφέρουμε τα προβλήματα εγκυρότητας που προκύπτουν λόγω των συσχετίσεων που υπάρχουν μεταξύ σχημάτων ορισμένων σε διαφορετικούς χώρους ονοματοδοσίας XML.

2.4.3.1 Ο Μηχανισμός Ονοματοδοσίας XML

Ο βασικός μηχανισμός που χρησιμοποιεί το RDF για τον προσδιορισμό των RDF σχημάτων και την χρήση αυτών για την δημιουργία RDF μεταδεδομένων είναι ο μηχανισμός ονοματοδοσίας XML [BHL99]. Ένας χώρος ονοματοδοσίας είναι μια συλλογή από τύπους στοιχείων (elements types) και/ή γνωρίσματα (attributes names) στην οποία αποδίδεται ένα μοναδικό URI. Ένα RDF σχήμα ορίζεται σε ένα χώρο ονοματοδοσίας και προσδιορίζεται από το URI του χώρου αυτού. Για παράδειγμα στο RDF σχήμα αποδίδεται το URI *http://www.w3.org/2000/01/rdf-schema#*. Οι πόροι που ορίζονται σε ένα σχήμα έχουν μοναδικά ονόματα που προκύπτουν συνδυάζοντας το URI του σχήματος με τα τοπικά ονόματα των πόρων. Είναι προφανές ότι το RDF αγνοεί πιο έξυπνους μηχανισμούς ονοματοδοσίας σε βάσεις πληροφοριών όπως αυτοί που στηρίζονται στην έννοια των πλαισίων συμφραζομένων (context) [TC97].

Κατά την δημιουργία RDF μεταδεδομένων προκειμένου να χρησιμοποιήσουμε τις ιδιότητες ή τις κλάσεις που ορίζονται σε κάποιο σχήμα, δημιουργούμε μια δήλωση με την οποία σχετίζεται ένας κωδικός/πρόθεμα (π.χ., rdfs) με το URI ενός σχήματος (π.χ.,

http://www.w3.org/2000/01/rdf-schema#). Οι κλάσεις και οι ιδιότητες του σχήματος αυτού αναφέρονται μέσα στα RDF μεταδεδομένα συνδυάζοντας το πρόθεμα με το τοπικό όνομα της ιδιότητας (π.χ, rdfs:Class). Η δήλωση δηλαδή ενός χώρου ονοματοδοσίας επιτρέπει την αναφορά στις ιδιότητες και κλάσεις που ορίζονται σ' αυτόν.

2.4.3.2 Σχέση μεταξύ σχημάτων ορισμένων σε διαφορετικούς χώρους ονοματοδοσίας XML.

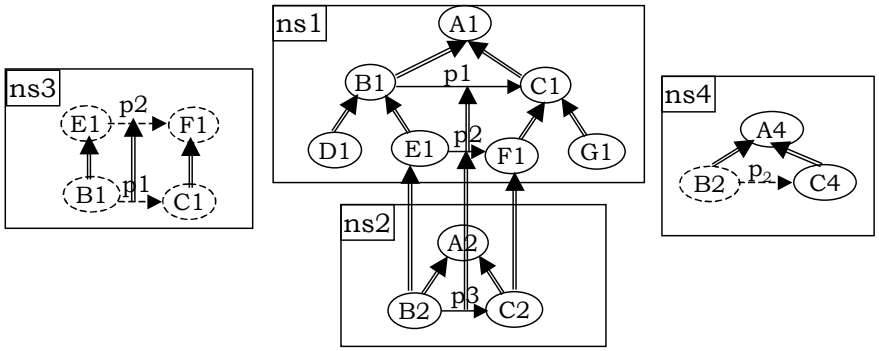
Στην υπο-ενότητα αυτή θα περιγράψουμε τους τρόπους με τους οποίους μπορούν να συσχετιστούν ιδιότητες και κλάσεις που ορίζονται σε διαφορετικά σχήματα και τα προβλήματα εγκυρότητας που προκύπτουν. Στην εικόνα 2.11 απεικονίζονται 4 RDF σχήματα. Οι κλάσεις ή οι ιδιότητες που περιγράφονται σε ένα σχήμα αλλά δεν ορίζονται σ' αυτό παριστάνονται με διακεκομμένες γραμμές. Παρατηρούμε ότι μεταξύ σχημάτων μπορούν να δημιουργηθούν τα εξής είδη συσχετίσεων:

- Σχέσεις υποσυνόλου μεταξύ κλάσεων/ιδιοτήτων που ορίζονται σε διαφορετικά σχήματα. Στην εικόνα παρατηρούμε ότι οι κλάσεις του σχήματος ns2, B2 και C2 δηλώνονται σαν υποκλάσεις των κλάσεων E1 και F1 αντίστοιχα που ορίζονται στο σχήμα ns1.

- Σχέσεις υποσυνόλου μεταξύ κλάσεων/ιδιοτήτων μπορούν να δηλωθούν σε διαφορετικό σχήμα από τα σχήματα που ορίζονται οι κλάσεις/ιδιότητες που συσχετίζονται. Στην εικόνα 2.11 παρατηρούμε ότι στο σχήμα ns3 ορίζονται σχέσεις υποσυνόλου μεταξύ των κλάσεων B1, E1 και των ιδιοτήτων p1, p2 ενώ οι κλάσεις και οι ιδιότητες ορίζονται σε διαφορετικό σχήμα (ns1).

- Οι κλάσεις που δηλώνονται σαν πεδίο ορισμού και πεδίο τιμών μιας ιδιότητας μπορεί να ορίζονται σε διαφορετικό σχήμα από αυτό που ορίζεται η ιδιότητα.

- Μια ιδιότητα rdfs:range ή rdfs:domain μπορεί να αποδοθεί σε μια ιδιότητα σε οποιοδήποτε σχήμα. Δηλαδή το πεδίο ορισμού και τιμών μιας ιδιότητας μπορεί να καθοριστεί σε οποιοδήποτε σχήμα. Στο σχήμα ns4 παρατηρούμε ότι στην ιδιότητα p2 η οποία ορίζεται στο σχήμα ns2 αποδίδεται η ιδιότητα rdfs:range με τιμή C4 και η ιδιότητα rdfs:domain με τιμή B2.



Εικόνα 2.11. Συσχετίσεις μεταξύ RDF σχημάτων.

Το RDF παρέχει απόλυτη ελευθερία στην απόδοση περιγραφών σε κλάσεις και ιδιότητες σε διαφορετικούς χώρους ονοματοδοσίας από αυτούς που ορίζονται. Το πρόβλημα που δημιουργείται είναι το εξής:

Πρόβλημα 6 – Εγκυρότητα σχημάτων σε πολλαπλούς χώρους ονοματοδοσίας: Η ένωση δύο εγκύρων RDF σχημάτων δεν είναι πάντα ένα έγκυρο RDF σχήμα.

Για παράδειγμα στην εικόνα 2.11 τόσο το σχήμα ns1 όσο και το ns4 είναι έγκυρα. Η ένωση όμως των σχημάτων δεν είναι ένα έγκυρο σχήμα δεδομένου ότι παραβιάζεται ο περιορισμός της μοναδικότητας της rdfs:range ιδιότητας εφόσον στην ιδιότητα p2 αποδίδονται δύο rdfs:range ιδιότητες με τιμές F1 και C4. Αντίστοιχα συμβαίνει και για την ένωση των σχημάτων ns1 και ns3. Στην περίπτωση αυτή δημιουργείται κύκλος στην ιεραρχία κλάσεων και στην ιεραρχία ιδιοτήτων. Στο σχήμα ns1 η κλάση E1 δηλώνεται σαν υποκλάση της B1 ενώ στο σχήμα ns3 το αντίστροφο.

Με βάση την παραπάνω διαπίστωση προκύπτει το εξής θέμα: *Πώς ελέγχεται η συνέπεια μεταδεδομένων που στηρίζονται σε πολλαπλά σχήματα*;

– Αρκεί να είναι συνεπή τα μεταδεδομένα καθώς και το μέρος του κάθε σχήματος που αναφέρονται τα μεταδεδομένα.

– Πρέπει να είναι συνεπή τα μεταδεδομένα καθώς και κάθε RDF σχήμα ξεχωριστά;

– Πρέπει να είναι συνεπή τόσο τα μεταδεδομένα όσο και η ένωση των σχημάτων;

Το RDF δεν καθορίζει πως πρέπει να γίνεται ο έλεγχος της εγκυρότητας των μεταδεδομένων. Επίσης δεν καθορίζει τη σημασιολογία της δήλωσης ενός χώρου ονοματοδοσίας σε ένα άλλο χώρο. Το γεγονός ότι η εισαγωγή ενός χώρου ονοματοδοσίας σε ένα άλλο δεν είναι μεταβατική, εφόσον πάντα πρέπει να εισάγουμε τους χώρους ονοματοδοσίας rdf και rdfs σε ένα αρχείο RDF περιγραφών, υποδηλώνει ότι

του αποδίδει σημασιολογία **αναφοράς**. Στην προσέγγιση μας αποδίδουμε σημασιολογία **αντιγραφής**.

Έλεγχος συνέπειας RDF μεταδεδομένων: Θεωρούμε ότι ένα σύνολο RDF περιγραφών είναι συνεπές όταν τόσο οι περιγραφές όσο και η ένωση των σχημάτων (ή έστω των μερών των σχημάτων) που χρησιμοποιούνται στις περιγραφές πληρούν τους σημασιολογικούς περιορισμούς της RDF/S.

2.5 Θεμελίωση της RDF/S με την χρήση μοντέλων αναπαράστασης γνώσης

Αυτή την στιγμή υπάρχουν δύο προσεγγίσεις όσον αφορά την RDF/S. Η πρώτη προσέγγιση αναφέρεται στην *επέκταση* της RDF/S. Σκοπός, άλλωστε, της RDF/S γλώσσας είναι να αποτελέσει την βάση πάνω στην οποία οι διάφορες 'κοινότητες' θα προσθέσουν τις δικές τους κατασκευές για την περιγραφή των πεδίων τους, και όχι μια 'πλήρη' γλώσσα περιγραφής σχημάτων. Για παράδειγμα, στην δημοσίευση [BKD+00] παρουσιάζεται η γλώσσα OIL [HFB+00] – μια γλώσσα για ορισμό και ανταλλαγή οντολογιών - σαν επέκταση του RDF σχήματος. Παράλληλα στην δημοσίευση [SEMD00] επεκτείνεται η RDFS ώστε να υποστηρίξει την μοντελοποίηση αξιωμάτων (ontological axioms) τα οποία είναι απαραίτητα για τον ορισμό οντολογιών.

Η δεύτερη προσέγγιση αναφέρεται στον *περιορισμό ή θεμελίωση* της RDF/S [CDH00], [B00], [W99] [C00] και [CK00]. Στην κατηγορία αυτή ανήκει η δική μας προσέγγιση η οποία βασίζεται στην γλώσσα παράστασης γνώσης Telos. Πιστεύουμε ότι πριν μιλήσουμε για επεκτάσεις είναι καλό να θεμελιώσουμε θεωρητικά την RDF/S. Άλλωστε αυτό είναι απαραίτητο για τον έλεγχο της εγκυρότητας και την αποτελεσματική αποθήκευση των RDF περιγραφών και στην συνέχεια την επερώτηση τους.

2.5.1 Συνοπτική παρουσίαση της γλώσσας Telos

Η Telos [MBJK90] είναι μια γλώσσα παράστασης γνώσης που υποστηρίζει ένα οντοκεντρικό μοντέλο δεδομένων. Προσφέρει τους παρακάτω εκφραστικούς μηχανισμούς: μη φραγμένη ιεραρχία ταξινόμησης, πολλαπλή και αυστηρή κληρονομικότητα και πλειότιμα γνωρίσματα, τα οποία με την σειρά τους μπορεί να έχουν και αυτά δικά τους γνωρίσματα. Επίσης παρέχει μηχανισμούς εξαγωγής συμπερασμάτων και χρονικής λογικής που δεν θα χρησιμοποιηθούν για την ανάλυση της RDF/S.

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

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

- Ονοματοδοσία: Κάθε οντότητα έχει ένα εσωτερικό, παραγόμενο από το σύστημα αναγνωριστικό. Επιπλέον ο χρήστης έχει την δυνατότητα να ονομάσει ο ίδιος μια οντότητα με ένα λογικό όνομα. Το λογικό όνομα ενός γνωρίσματος είναι της μορφής x.y όπου x είναι το λογικό όνομα της οντότητας της οποίας αποτελεί γνώρισμα και y είναι το όνομα του ίδιου.

- Ταξινόμηση: Με το μηχανισμό αυτό μια ατομική οντότητα περιγράφεται σα μέλος μιας κλάσης, της οποίας κληρονομεί τα γνωρίσματα. Μια κλάση είναι και αυτή με την σειρά της μια οντότητα, άρα μπορεί να είναι περίπτωση μιας άλλης κλάσης. Έτσι δημιουργείται μια μη φραγμένη ακολουθία από κλάσεις. Στο κατώτερο επίπεδο (Token) τοποθετούνται οι ατομικές οντότητες. Κατόπιν υπάρχουν οι απλές κλάσεις (S_Class), μετά οι μετα-κλάσεις (M1_Class) κοκ. Όλες οι οντότητες που περιγράφονται από την Telos ταξινομούνται στην κλάση του συστήματος Object. Υποκλάσεις της Object είναι οι κλάσεις Individual και Attribute. Στην κλάση Individual ταξινομούνται οι οντότητες, οι κλάσεις από οντότητες, οι κλάσεις από κλάσεις από οντότητες κοκ. Στην κλάση Attribute ταξινομούνται οι σχέσεις που έχουν οι οντότητες μεταξύ τους, οι κλάσεις σχέσεων, οι κλάσεις από κλάσεις σχέσεων κοκ. Οι κλάσεις (σχέσεων ή οντοτήτων) που ορίζει ο χρήστης ταξινομούνται και στη κλάση του συστήματος Class. Επίσης υπάρχουν και οι πρωτογενείς τιμές Integer, Real, String. Μια οντότητα μπορεί να ανήκει σε παραπάνω από μια κλάσεις.

- Απόδοση γνωρίσματος: Με το μηχανισμό αυτό αποδίδονται γνωρίσματα στις οντότητες. Τα γνωρίσματα μπορούν να θεωρηθούν σαν σχέσεις μεταξύ οντοτήτων μια και έχουν ένα πεδίο ορισμού και ένα πεδίο τιμών. Κάθε γνώρισμα μπορεί να έχει παραπάνω από μια ή και καμία τιμή. Επίσης μπορεί να έχει και αυτό γνωρίσματα.

- Περιορισμός στην ταξινόμηση γνωρισμάτων: Αν ένα γνώρισμα είναι περίπτωση μιας κατηγορίας γνωρισμάτων, τότε το πεδίο ορισμού και το πεδίο τιμών του πρέπει να

είναι περιπτώσεις των πεδίων ορισμού και τιμών της κλάσης γνωρισμάτων στην οποία ανήκει.

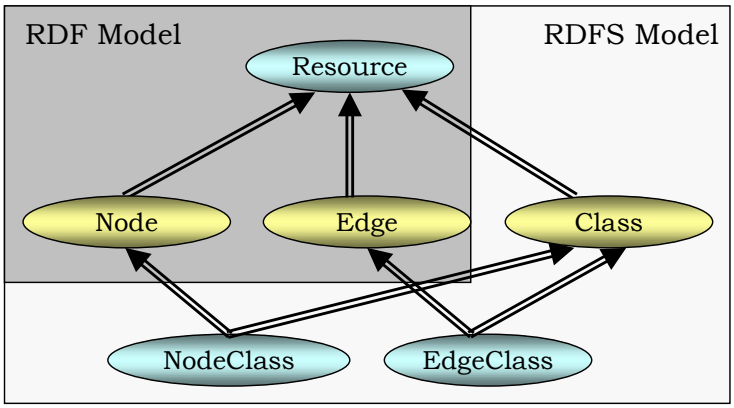
- Γενίκευση (αντίστροφο: Εξειδίκευση): Ο μηχανισμός αυτός ισχύει μεταξύ κλάσεων. Ορίζει μια σχέση υποσυνόλου μεταξύ των κλάσεων που ονομάζεται isA. Αν A isA B, τότε η A ονομάζεται υποκλάση της B. Η A κληρονομεί όλα τα γνωρίσματα της B και είτε έχει επιπλέον γνωρίσματα είτε περιορίζει το σύνολο τιμών των γνωρισμάτων που κληρονομεί από την B. Μια κλάση μπορεί να έχει πάνω άπω μία υπερκλάσεις.

- Περιορισμός στις υποκλάσεις γνωρισμάτων: Έστω AC1 και AC2 κατηγορίες γνωρισμάτων. Αν AC1 isA AC2 τότε τα πεδία ορισμού και τιμών της AC1 είναι υποκλάσεις των αντίστοιχων της AC2.

2.5.2 Θεμελίωση της RDF/S χρησιμοποιώντας την γλώσσα παράστασης Telos

2.5.2.1 Ανάλυση RDF/S Μοντέλου δεδομένων

Το χαρακτηριστικό του RDF μοντέλου που το διαφοροποιεί από τα παραδοσιακά μοντέλα δεδομένων είναι ότι οι ιδιότητες είναι αυτόνομες οντότητες (fisrt-class objects) και προσδιορίζονται μόνο από το όνομα τους (URI). Καθένας μπορεί να δημιουργήσει μια ιδιότητα π.χ. *τίτλος* χωρίς απαραίτητα να υπάρχουν περιορισμοί στο πεδίο ορισμού και τιμών της. Οι ιδιότητες δηλαδή δεν ορίζονται με βάση τις κλάσεις. Σε αντίθεση με τα οντοκεντρικά συστήματα όπου όλη η πληροφορία για μια οντότητα κρατείται μαζί και οι ιδιότητες που μπορεί να έχει καθορίζονται άμεσα από τον τύπο της, το RDF κάνει πραγματικότητα την έκφραση "*οτιδήποτε μπορεί να πει οτιδήποτε για οτιδήποτε*"[LCS99]. Οποιαδήποτε στιγμή μπορεί να προστεθεί μια ιδιότητα με πεδίο ορισμού οποιαδήποτε κλάση. Άρα δεν υπάρχει προκαθορισμένος αριθμός ιδιοτήτων που μπορούν να αποδοθούν στους πόρους. Οι RDF ιδιότητες είναι προαιρετικές και πλειότιμες. Το RDF μοντέλο δεδομένων μοιάζει αρκετά σε μοντέλα ημι-δομημένων δεδομένων [ABS99].



Εικόνα 2.12. RDF/S Μοντέλο.

Στην εικόνα 2.12 παρουσιάζουμε μια διαφορετική όψη του βασικού RDF/S μοντέλου δεδομένων η οποία κάνει πιο ξεκάθαρη την διαφοροποίηση μεταξύ των πόρων. Οι *πόροι* (*Resources*) διαφοροποιούνται με βάση την φύση τους σε *κόμβους* (*Nodes*) και σε *ακμές* (*Edges*). Επίσης διαφοροποιούνται ανάλογα με το αν παριστάνουν συγκεκριμένες ή αφηρημένες έννοιες σε *tokens* και *κλάσεις* (*Classes).*

Πριν αναλύσουμε τις παραπάνω κατηγορίες πρέπει να σημειώσουμε ότι στο RDF οι ιδιότητες δεν ορίζονται σαν κλάσεις. Το γεγονός ότι μια RDF ιδιότητα μπορεί να ερμηνευτεί σαν το σύνολο των ζευγαριών (subject, object) που ενώνονται με ακμή με το όνομα της ιδιότητας σε ένα RDF γράφο σημαίνει ότι οι RDF ιδιότητες είναι κλάσεις ιδιοτήτων. Γι' αυτό άλλωστε μπορεί να εφαρμοστεί και η ιδιότητα rdfs:subPropertyOf, η οποία δηλώνει σχέση υποσύνολου/υπερσυνόλου, μεταξύ ιδιοτήτων. Τα ζευγάρια (subject, object) ονομάζονται περιπτώσεις της RDF ιδιότητας.

Η κατηγορία *Κόμβοι* περιλαμβάνει το σύνολο των οντοτήτων. Μια οντότητα μπορεί να είναι συγκεκριμένη, όπως μια σελίδα του παγκόσμιου ιστού π.χ *http://www.nga.gov/* ή αφηρημένη όπως μια RDF κλάση π.χ. *ExternalPage*.

Η κατηγορία *Ακμές* περιέχει το σύνολο των RDF ιδιοτήτων που αντιπροσωπεύουν δυαδικές σχέσεις μεταξύ αφηρημένων κόμβων (π.χ. η RDF ιδιότητα *creator* η οποία ορίζεται μεταξύ των κλάσεων *ExternalPage* και *Person)* καθώς και το σύνολο των περιπτώσεων των RDF ιδιοτήτων (π.χ. η ιδιότητα *creator* που αποδίδεται στον πόρο *http://www.nga.gov/* με τιμή ένα συγκεκριμένο πόρο).

Στην εικόνα 2.12 με τιρκουάζ χρώμα χρωματίζονται οι κλάσεις που ορίζονται στην RDF/S ενώ με κίτρινο οι κλάσεις που εμείς έχουμε προσθέσει για να

αναπαραστήσουμε την παραπάνω διαφοροποίηση. Για να αναπαραστήσουμε την ερμηνεία που έχουμε αποδώσει στις RDF ιδιότητες, εισάγουμε στο μοντέλο μας την έννοια *Class*. Η κλάση *Class* αντιπροσωπεύει τόσο τις κλάσεις οντοτήτων και όσο και τις κλάσεις ιδιοτήτων. Υποκλάσεις της είναι οι κλάσεις *NodeClass* και *EdgeClass*. Η κλάση NodeClass αντιστοιχεί στην κλάση rdfs:Class. Η κλάση *EdgeClass* που είναι υποκλάση των *Class* και *Property* περιέχει τις RDF ιδιότητες και αντιστοιχεί στην κλάση *rdf:Property*.

2.5.2.2 Αναπαράσταση RDF/S μοντέλου στην Telos

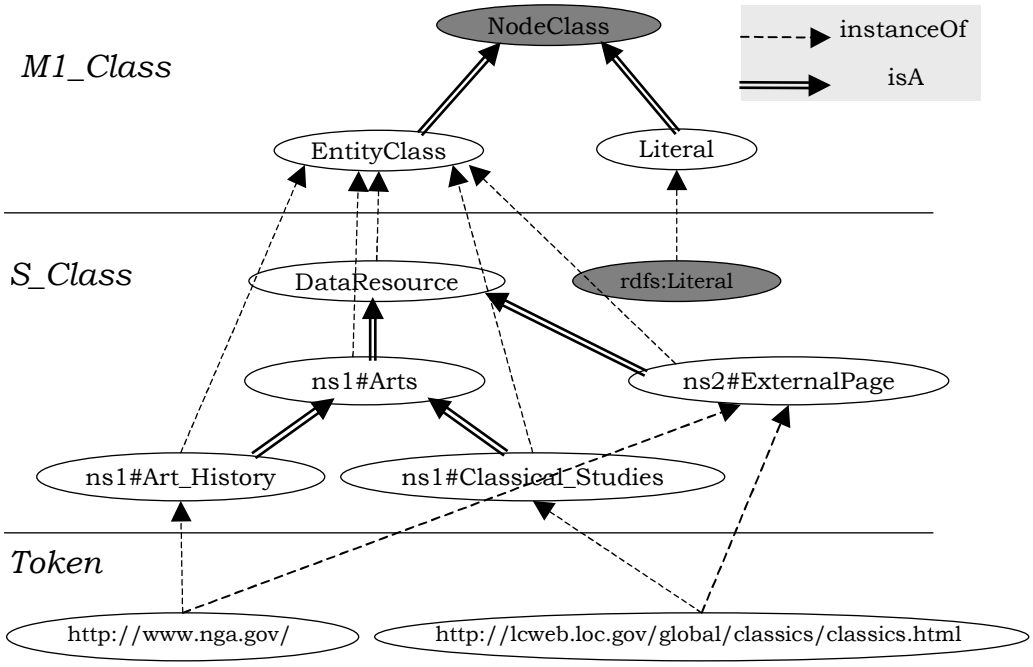
Στην ενότητα αυτή θα αναπαραστήσουμε τις βασικές RDF/S κατασκευές στην Telos. Παράλληλα θα αναφέρουμε τους περιορισμούς που έχουμε επιβάλλει και τις παραδοχές που έχουμε κάνει.

Αναπαράσταση RDF κλάσεων: Το μετα-μοντέλο που δημιουργείται για την αναπαράσταση των RDF κλάσεων απεικονίζεται στο πάνω μέρος της εικόνας 2.13. Στην κορυφή της ιεραρχίας βρίσκεται η κλάση *NodeClass*. Υποκλάσεις της NodeClass είναι οι κλάσεις *EntityClass* και *Literal*. Η κλάση *EntityClass* περιέχει τις κλάσεις που ορίζονται από τους χρήστες. Η κλάση Literal περιέχει την κλάση rdfs:Literal και τους ατομικούς τύπους που ορίζονται στο XML Schema. Θα πρέπει να τονιστεί ότι οι κλάσεις *EntityClass* και *Literal* δεν έχουν κοινά μέλη. Ο λόγος που εισαγάγαμε την κλάση *EntityClass* είναι για να δηλώσουμε καθαρά ότι μια RDF ιδιότητα μπορεί να έχει σαν πεδίο ορισμού οποιαδήποτε κλάση εκτός την κλάση rdfs:Literal (*Πρόβλημα 2*).

Οι ιδιότητες rdf:type και rdfs:subClassOf έχουν την ίδια σημασιολογία με τους μηχανισμούς ταξινόμησης (instanceOf) και γενίκευσης (isA) που υποστηρίζονται από την Telos και στο μοντέλο μας αντιπροσωπεύονται από αυτούς. Στη εικόνα 12 αναπαριστάνεται στην Telos μια ιεραρχία κλάσεων καθώς και μέλη των κλάσεων αυτών. Κάθε RDF κλάση δηλώνεται μέλος της *EntityClass* μέσω του μηχανισμού ταξινόμησης (instanceOf). Τα μέλη μιας κλάσης αποδίδονται σ' αυτή και πάλι μέσω του μηχανισμού ταξινόμησης. Τέλος, οι κλάσεις οργανώνονται σε ιεραρχίες μέσω του μηχανισμού γενίκευσης. Στο μοντέλο μας κορυφή της ιεραρχίας των κλάσεων είναι κλάση *DataResource*. Σε αντίθεση με την RDFS όπου η κορυφή της ιεραρχίας είναι η κλάση rdfs:Resource. Στην Telos δεν μπορούμε να δημιουργήσουμε μια κλάση με την σημασιολογία της rdfs:Resource. Μια κλάση δηλαδή που να έχει σαν μέλη της οντότητες διαφορετικών επιπέδων όπως απλούς πόρους (*http://www.nga.gov/*) και κλάσεις (ExternalPage). Μόνο η ω-κλάση Resource (ή Object στην Telos) έχει αυτήν την

ιδιότητα. Η κλάση *DataResource* είναι υποσύνολο της rdfs:Resource και περιέχει το σύνολο των απλών οντοτήτων (πόρων που δεν είναι σχήμα κλάσεις ή ιδιότητες). Κάθε κλάση έχει σαν υπερ-κλάση την *DataResource*. Όμως λόγω της μεταβατικής ιδιότητας του μηχανισμού γενίκευσης, μια κλάση δηλώνεται άμεσα σαν υποκλάση της DataResource μόνο όταν δεν έχει υπερ-κλάσεις. Στην εικόνα 2.13 απεικονίζονται οι παρακάτω RDF περιγραφές:

Ορίζουμε την κλάση Arts και την κλάση ExternalPage. Οι κλάσεις Art_History και Classical_Studies είναι υποκλάσεις της Arts. Η σελίδα http://www.nga.gov/ ανήκει στην κλάση Art_History και στην κλάση ExternalPage. Η σελίδα http://lcweb.loc.gov/global/classics/classics.html ανήκει στην κλάση Classical_Studies και στην κλάση ExternalPage.



Εικόνα 2.13. Αναπαράσταση RDF κλάσεων και μελών τους στην Telos.

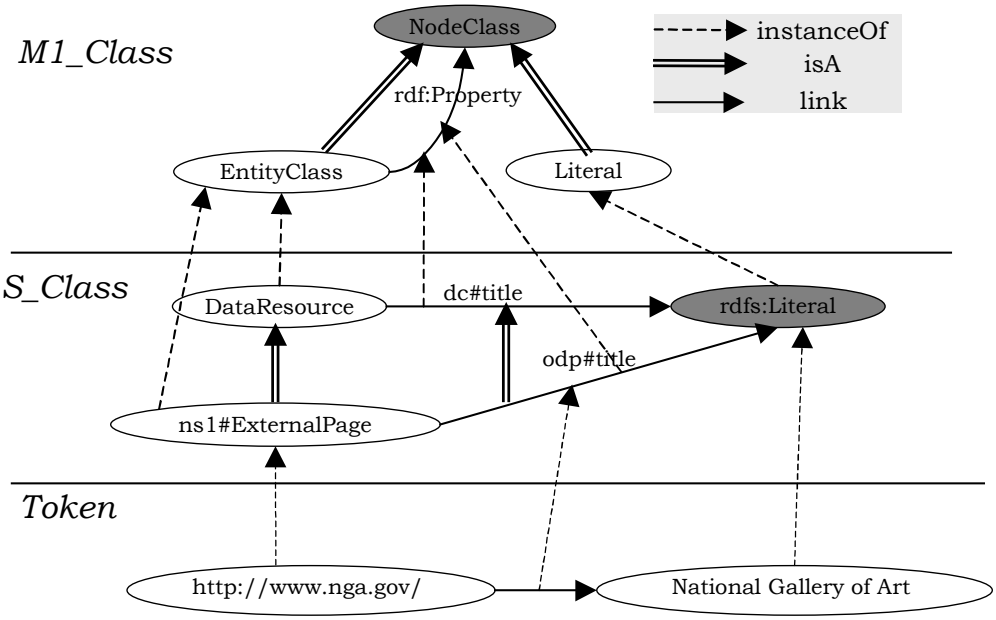
Αναπαράσταση RDF ιδιοτήτων: Στην γλώσσα RDFS κάποιες από τις βασικές ιδιότητες, όπως rdfs:range, rdfs:domain, rdf:type και rdfs:subClassOf, παρουσιάζονται τόσο σαν δομικά στοιχεία για τον ορισμό των κατασκευών της RDF/S (συμπεριλαμβανομένων και των ίδιων) όσο και σαν απλές RDF ιδιότητες που ορίζονται στο RDF σχήμα. Το γεγονός αυτό δυσκολεύει την κατανόηση της RDFS [NWC00]. Στην αναπαράσταση που παρουσιάζουμε στην συνέχεια, οι παραπάνω βασικές ιδιότητες

μοντελοποιούνται διαφορετικά από τις ‘συνηθισμένες’ ιδιότητες που ορίζονται από τους χρήστες.

Στο μοντέλο μας η κλάση rdf:Property αναπαριστάται από την κλάση γνωρισμάτων rdf:Property που βρίσκεται στο επίπεδο M1. Το πεδίο ορισμού της είναι η κλάση *EntityClass* και το πεδίο τιμών της η κλάση rdfs:Class. Περιέχει τις RDF ιδιότητες που ορίζονται από τους χρήστες. Δεν περιέχει τις ιδιότητες που ορίζονται στην RDFS. Οι βασικές ιδιότητες της RDF/S (type, subClassOf, subPropertyOf, range, domain) αντιστοιχούνται σε μηχανισμούς της γλώσσας Telos ενώ οι ιδιότητες comment, label, seeAlso, isDefinedBy μοντελοποιούνται διαφορετικά (εικόνα 2.15).

Οι RDF ιδιότητες αντιστοιχούν σε κλάσεις γνωρισμάτων και είναι μέλη της κλάσης γνωρισμάτων rdf:Property. Το πεδίο ορισμού μιας ιδιότητας μπορεί να είναι οποιαδήποτε S_Class εκτός από την rdfs:Literal, ενώ το πεδίο τιμών συμπεριλαμβάνει και την κλάση rdfs:Literal. Αν το πεδίο ορισμού/τιμών μιας ιδιότητας είναι η κλάση rdfs:Resource τότε ορίζουμε σαν πεδίο ορισμού/τιμών την κλάση DataResource. Άρα στο μοντέλο μας οι ιδιότητες που ορίζονται από το χρήστη μπορούν να περιγράψουν μόνο ‘απλούς’ πόρους. Αν το πεδίο ορισμού της ιδιότητας δεν ορίζεται θεωρούμε ως πεδίο ορισμού την κλάση *DataResource.* Η ιδιότητα rdfs:subPropertyOf αντιστοιχίζεται στο μηχανισμό εξειδίκευσης (isA) ανάμεσα σε κλάσεις γνωρισμάτων. Ο μηχανισμός αυτός ικανοποιεί τον περιορισμό που θέσαμε για το πεδίο ορισμού και τιμών μιας υποιδιότητας. Στην εικόνα 2.14 απεικονίζονται οι παρακάτω RDF περιγραφές:

Ορίζουμε την ιδιότητα odp#title. Η ιδιότητα odp#title είναι υποιδιότητα της ιδιότητας dc#title (Dublin Core ιδιότητα). Το πεδίο ορισμού της ιδιότητας ns1#title είναι η κλάση ExternalPage και το πεδίο τιμών η κλάση rdfs:Literal. Το πεδίο ορισμού της κλάσης dc#title είναι η κλάση rdfs:Resource και το πεδίο τιμών η κλάση rdfs:Literal. Ο τίτλος (odp#title) της σελίδας http://www.nga.gov/ είναι National Gallery of Art.



Εικόνα 2.14. Αναπαράσταση RDF ιδιοτήτων στην Telos.

Στην συνέχεια θα παραθέσουμε περιορισμούς που εισαγάγουμε στο μοντέλο μας σε σχέση με το RDF.

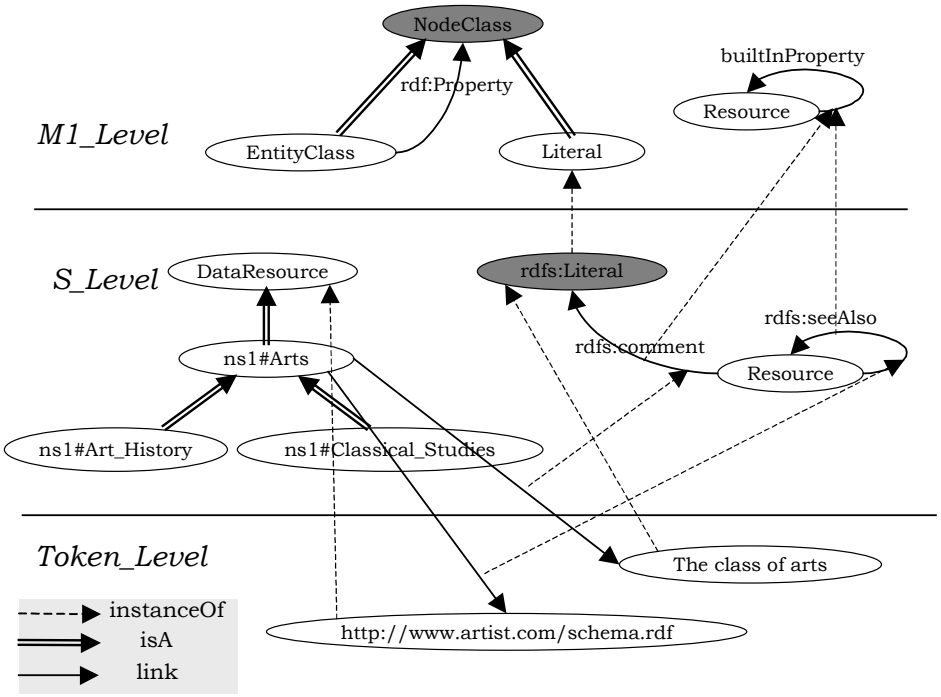
Περιορισμός 1: Επιτρέπουμε σε μια ιδιότητα να έχει μόνο μια rdfs:domain ιδιότητα η οποία πρέπει να ορίζεται στο αρχείο που ορίζεται και η ιδιότητα στην οποία αποδίδεται. Οι περιορισμοί προκύπτουν εξαιτίας των προβλημάτων (Πρόβλημα 1) που δημιουργούνται λόγω της απόδοσης πολλαπλών rdfs:domain ιδιοτήτων σε μια ιδιότητα και του *προβλήματος 5*. Η παραπάνω παραδοχή δεν είναι περιοριστική. Θα δείξουμε με ένα παράδειγμα πως μπορεί να αναπαρασταθεί με διαφορετικό τρόπο μια ιδιότητα με πολλαπλές rdfs:domain ιδιότητες ενώ ταυτόχρονα διατηρούμε την κοινή σημασιολογία. Έστω η ιδιότητα *name* με πεδίο ορισμού τις κλάσεις *Person* και *Company*. Η αναπαράσταση που προτείνουμε είναι η εξής: Δημιουργούμε δύο ιδιότητες *name1* και *name2* με πεδίο ορισμού τις κλάσεις *Person* και *Company* αντίστοιχα. Για να διατηρήσουμε την κοινή σημασιολογία των ιδιοτήτων δημιουργούμε την ιδιότητα *name* η οποία θα είναι υπερ-κλάση των ιδιοτήτων *name1* και *name2*.

Περιορισμός 2: Το πεδίο ορισμού μιας ιδιότητας πρέπει να δηλώνεται στον ορισμό της (βλέπε πρόβλημα 3).

Περιορισμός: Οι σχέσεις υπερκλάσης που αποδίδονται σε μια κλάση πρέπει να δηλώνονται στον χώρο ονοματοδοσίας που ορίζεται η κλάση. Το ίδιο ισχύει και για τις

ιδιότητες. Ο περιορισμός αυτός προκύπτει εξαιτίας των ασυνεπειών που δημιουργούνται κατά την ένωση έγκυρων σχημάτων (πρόβλημα 6).

Διαφορά RDF ιδιοτήτων και Telos γνωρισμάτων: Όπως αναφέραμε η Telos επιτρέπει την δημιουργία κλάσεων γνωρισμάτων με πεδίο ορισμού κλάσεις γνωρισμάτων. Το RDF, παρόλο που βασίζεται στην έννοια της *ιδιότητας,* δεν παρέχει την δυνατότητα ορισμού ιδιοτήτων με πεδίο ορισμού άλλες ιδιότητες. Παρέχει μόνο δυνατότητα απόδοσης ιδιοτήτων σε RDF ιδιότητες. Για παράδειγμα, μπορεί να οριστεί μια ιδιότητα π.χ. inverseProperty με πεδίο ορισμού την κλάση rdf:Property η οποία στην συνέχεια μπορεί να αποδοθεί σε οποιαδήποτε RDF ιδιότητα.

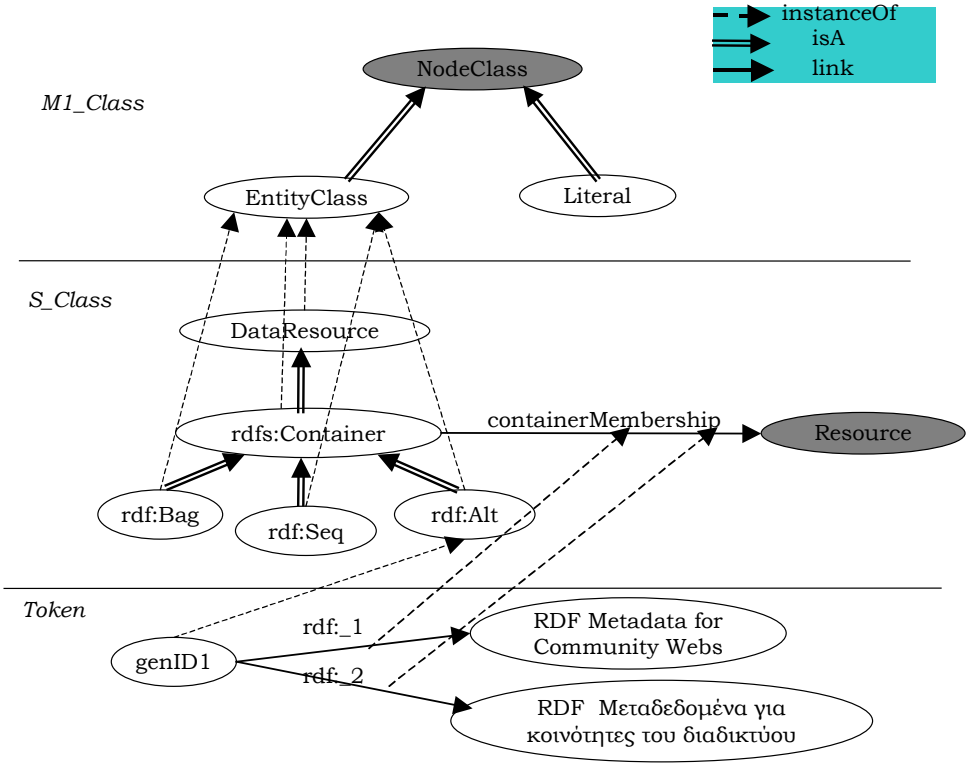


Εικόνα 2.15. Αναπαράσταση RDF/S ιδιοτήτων.

Αναπαράσταση προκαθορισμένων rdf/s ιδιοτήτων: Οι προκαθορισμένες ιδιότητες rdfs:comment, rdfs:label, rdfs:seeAlso και rdfs:isDefinedBy χρησιμοποιούνται για την περιγραφή κυρίως σχήμα κατασκευών, είναι δυνατόν όμως να αποδοθούν και σε 'απλούς' πόρους. Για την αναπαράσταση τους δημιουργούμε σε M1 επίπεδο την κλάση γνωρισμάτων builtInProperty (εικόνα 2.15). Οι παραπάνω ιδιότητες αποτελούν περιπτώσεις της builtInProperty. Το πεδίο ορισμού τους είναι η ω-κλάση Resource έτσι ώστε να μπορούν να αποδοθούν σε διαφορετικές κατηγορίες αντικειμένων (απλούς

πόρους και σχήμα κατασκευές). Το πεδίο τιμών των rdfs:comment και rdfs:label είναι η κλάση rdfs:Literal και των rdfs:seeAlso και rdfs:isDefinedBy η κλάση Resource.

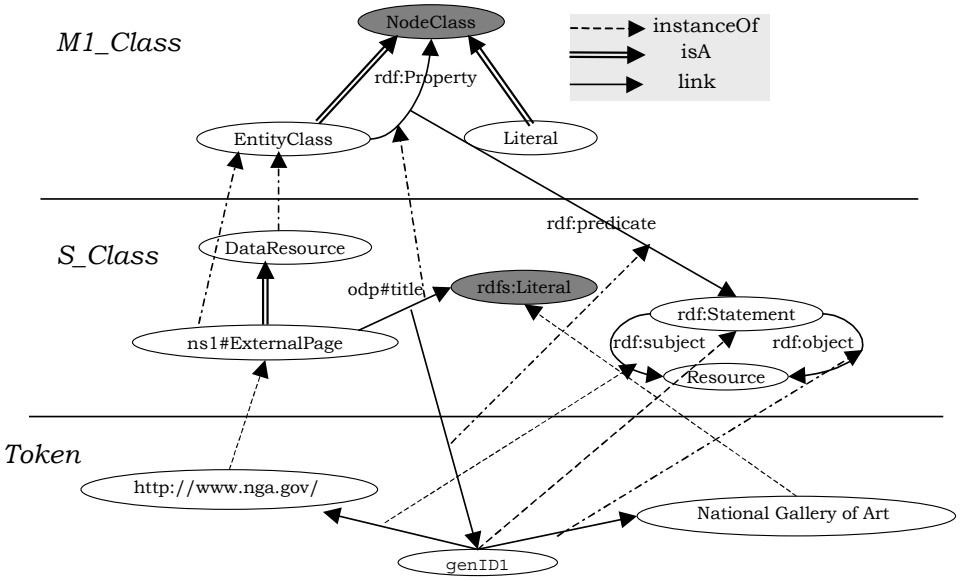
Στο μοντέλο της RDF/S που παρουσιάσαμε παραπάνω η κλάση rdfs:Resource δεν αναπαριστάται. Αντίθετα παρουσιάζονται τα υποσύνολα της NodeClass, rdf:Property, builtInProperty και DataResource. Παράλληλα οι ιδιότητες της RDF/S δεν είναι περιπτώσεις της rdf:Property.



Εικόνα 2.16. Αναπαράσταση RDF συλλογών.

Αναπαράσταση RDF συλλογών: Για την αναπαράσταση των συλλογών εισάγουμε στο μοντέλο μας την κλάση rdfs:Container η οποία ορίζεται υποκλάση της DataResource και τις κλάσεις rdf:Bag, rdf:Seq και rdf:Alt που είναι υποκλάσεις της rdfs:Container. Οι ιδιότητες rdf:_1, rdf:_2, .. που δηλώνουν τα μέρη ενός container είναι δύσκολο να αναπαρασταθούν στο μοντέλο μας σαν κλάσεις γνωρισμάτων (όπως και σε οποιοδήποτε άλλο σχήμα) επειδή είναι μη αριθμήσιμες. Για την αναπαράσταση των μελών των συλλογών εισαγάγουμε την κλάση γνωρισμάτων containerMembership σε επίπεδο S_Class με πεδίο τιμών την κλάση Resource. Τα βέλη που αναπαριστάνουν τα

περιεχόμενα μιας συλλογής θα είναι περιπτώσεις της κλάσης γνωρισμάτων containerMembership (βλέπε εικόνα 2.16).



Εικόνα 2.17. Αναπαράσταση υποστασιοποιημένων δηλώσεων

Αναπαράσταση Υποστασιοποιημένων δηλώσεων: Η διάκριση μεταξύ των δηλώσεων και των αντίστοιχων υποστασιοποιημένων δηλώσεων πρέπει να είναι ξεκάθαρη. Όπως αναφέραμε παραπάνω ένας RDF γράφος εκφράζει ένα γεγονός μόνο όταν η δήλωση που το εκφράζει περιέχεται στο γράφο. Γι' αυτό το λόγο μια υποστασιοποιημένη δήλωση δεν μοντελοποιείται σαν ένα σύνδεσμος μεταξύ των κόμβων αφετηρίας και προορισμού, όπως μια δήλωση. Η αναπαράσταση που επιλέξαμε απεικονίζεται στην εικόνα 2.17 Για κάθε υποστασιοποιημένη δήλωση δημιουργούμε μια οντότητα που ανήκει στην κλάση rdf:Statement και έχει γνωρίσματα τύπου rdf:subject και rdf:object. Οι τιμές των γνωρισμάτων rdf:subject και rdf:object μπορεί να είναι οποιοδήποτε πόρος (ή και αλφαριθμητικό για την ιδιότητα rdf:object). Επίσης στην οντότητα καταλήγει ένα γνώρισμα τύπου rdf:predicate το οποίο ξεκινάει από την κλάση γνωρισμάτων που παριστάνει την ιδιότητα που υπάρχει στην υποστασιοποιημένη δήλωση (βλέπε εικόνα 2.17).

Στην συνέχεια παραθέτουμε τους περιορισμούς που ορίζονται στην RDF/S τόσο για το σχήμα όσο και για τα δεδομένα. Επίσης παραθέτουμε και τους περιορισμούς που εμείς προσθέτουμε. Όταν πληρούνται οι παρα κάτω επιπλέον περιορισμοί η ένωση έγκυρων σχημάτων RDF είναι πάντα ένα έγκυρο σχήμα RDF.

Βασικοί Περιορισμοί RDF/S για δεδομένα
Ένας πόρος μπορεί να ανήκει το πολύ σε μια από τις κλάσεις rdfs:Class, rdf:Property, rdf:Bag, rdf:Alt, rdf:Seq, rdf:Statement και rdfs:Literal.
Για κάθε δήλωση (pred, sub, obj) ο πόρος pred πρέπει να ανήκει στην κλάση rdf:Property.
Η τιμή μιας ιδιότητας με πεδίο τιμών την κλάση A πρέπει να ανήκει στην κλάση A ή σε κάποια υποκλάση της.
Μια ιδιότητα μπορεί να αποδοθεί μόνο σε πόρους που ανήκουν σε μια από τις κλάσεις που αποτελούν το πεδίο ορισμού της ιδιότητας, δηλαδή στην ένωση των τιμών της ιδιότητας rdfs:domain.
Σε μια υποστασιοποιημένη δήλωση πρέπει να έχουν αποδοθεί ακριβώς μια φορά οι ιδιότητες rdf:predicate, rdf:subject και rdf:object.

Βασικοί Περιορισμοί RDF/S για σχήμα
Δεν επιτρέπεται να δημιουργηθεί κύκλος στην ιεραρχία των κλάσεων.
Δεν επιτρέπεται να δημιουργηθεί κύκλος στην ιεραρχία των ιδιοτήτων.
Σε μια ιδιότητα μπορεί να αποδοθεί το πολύ μια rdfs:range ιδιότητα.
Η ιδιότητα rdfs:subClassOf εφαρμόζεται μεταξύ κλάσεων.
Η ιδιότητα rdfs:subPropertyOf εφαρμόζεται μεταξύ ιδιοτήτων.
Η τιμή της ιδιότητας rdfs:range είναι μια κλάση.
Η τιμή της ιδιότητας rdfs:domain είναι μια κλάση (εκτός την κλάση rdfs:Literal).
Η τιμή της ιδιότητας rdf:type είναι μια κλάση.
Έστω δύο ιδιότητες p1 και p2 και η ιδιότητα p1 αποτελεί υποιδιότητα της p2. Το πεδίο

ορισμού/τιμών της ιδιότητας p1 πρέπει να είναι υποσύνολο ή να ταυτίζεται με το πεδίο ορισμού/τιμών της ιδιότητας p2. (Αυτός ο περιορισμός δεν δηλώνεται ρητά στο RDF/S).

Επιπλέον Περιορισμοί για σχήμα
Σε μια ιδιότητα μπορεί να αποδοθεί μια και μόνο μια rdfs:domain ιδιότητα και μάλιστα στο χώρο ονοματοδοσίας που ορίζεται και η ιδιότητα..
Το πεδίο τιμών μιας ιδιότητας πρέπει να ορίζεται στο χώρο ονοματοδοσίας που ορίζεται και η ιδιότητα.
Οι σχέσεις υπερκλάσης που αποδίδονται σε μια κλάση πρέπει να δηλώνονται στον χώρο ονοματοδοσίας που ορίζεται η κλάση.
Οι σχέσεις υπερ-ιδιότητας που αποδίδονται σε μια ιδιότητα πρέπει να δηλώνονται στο χώρο ονοματοδοσίας που ορίζεται η ιδιότητα.

Εικόνα 2.18. Σημασιολογικοί Περιορισμοί..

Κεφάλαιο 3

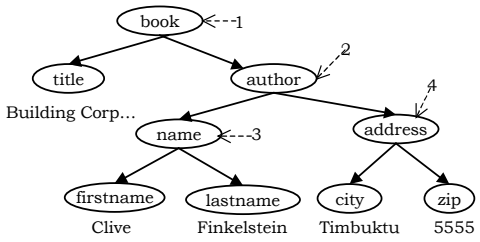
Αποθήκευση XML και RDF ημιδομημένων δεδομένων: Υπάρχουσες προσεγγίσεις

Στο προηγούμενο κεφάλαιο είδαμε ότι τα RDF δεδομένα μπορούν να αναπαρασταθούν σαν κατευθυνόμενοι γράφοι με ετικέτες. Στο παρόν κεφάλαιο θα αναλύσουμε διάφορες προσεγγίσεις που υπάρχουν για την αποθήκευση γράφων. Συγκεκριμένα, θα παρουσιάσουμε τις προσεγγίσεις που έχουν προταθεί για την αποθήκευση XML δεδομένων τα οποία, όπως και τα RDF δεδομένα, μπορούν να αναπαρασταθούν σαν γράφοι με ετικέτες. Επιπλέον θα παρουσιάσουμε τις κυριότερες προσπάθειες που έχουν γίνει για την αποθήκευση των RDF δεδομένων καθώς και συστήματα που έχουν υλοποιηθεί.

3.1 Αποθήκευση XML ημιδομημένων δεδομένων

Το μοντέλο που χρησιμοποιείται για την αναπαράσταση των XML δεδομένων είναι κατευθυνόμενοι γράφοι με ετικέτες στους κόμβους. Οι κόμβοι του γράφου αντιστοιχούν σε XML στοιχεία ή γνωρίσματα. Οι ακμές αντιπροσωπεύουν σχέσεις πατέρα-παιδιού που υπάρχουν ανάμεσα σε στοιχεία ή ανάμεσα σε στοιχείο και γνώρισμα. Στην εικόνα 3.1 απεικονίζονται XML δεδομένα και η αναπαράσταση τους με την μορφή γράφων.

```
<book>
    <title>Building Corp. Portals with XML</title>
    <author>
  <name>
      <firstname>Clive<firstname>
      <lastname>Finkelstein <lastname>
  <name>
  <address>
       <city>Timbuktu</city>
       <zip>5555</zip>
  </address>
      </author>
    <book>
```



Εικόνα 3.1. XML δεδομένα και γράφος.

3.1.1 Αποθήκευση σε αρχεία

Η πιο απλή προσέγγιση για την αποθήκευση XML δεδομένων είναι η αποθήκευση τους σαν απλά αρχεία κειμένου (ASCII αρχεία). Στην συνέχεια για την επεξεργασία των αποθηκευμένων XML δεδομένων απαιτείται η χρήση ενός XML συντακτικού αναλυτή ο οποίος αναπαριστά τα XML δεδομένα σε δεντρική μορφή (ή σε μορφή γράφων) στην κύρια μνήμη και παρέχει πρόσβαση σ' αυτά δομή μέσω ενός συνόλου μεθόδων (π.χ. Document Object Model [DOM99]). Το πλεονέκτημα αυτής της προσέγγισης είναι ότι μπορεί εύκολα να υλοποιηθεί χωρίς να απαιτείται κάποιο σύστημα διαχείρισης βάσεων δεδομένων ή διαχειριστής αντικειμένων. Όμως έχει σημαντικά μειονεκτήματα. Η ανάλυση (parsing) των XML αρχείων, η οποία μπορεί να είναι αρκετά χρονοβόρα ιδιαίτερα όταν τα αρχεία είναι μεγάλα, απαιτείται κάθε φορά που εκτελείται κάποια μορφή επεξεργασίας στο αρχείο π.χ. επερώτηση. Το δεύτερο μειονέκτημα είναι ότι το δέντρο που παράγει ο συντακτικός αναλυτής πρέπει να κρατηθεί ολόκληρο στην μνήμη κατά την επεξεργασία του. Ένα ακόμα μειονέκτημα είναι ότι τα αρχεία είναι δύσκολο να ενημερωθούν όπως επίσης και το γεγονός ότι είναι δύσκολο να δημιουργηθούν δείκτες.

Η παραπάνω προσέγγιση έχει πολύ μεγάλους χρόνους απόκρισης και δεν μπορεί να εφαρμοστεί σε συστήματα όπου ο χρόνος, κυρίως για την επερώτηση, είναι κρίσιμος παράγοντας. Για το σκοπό αυτό έχει προταθεί ένα σύνολο διαφορετικών προσεγγίσεων όπου για την αποθήκευση των XML γράφων χρησιμοποιούνται είτε σχεσιακές βάσεις δεδομένων, είτε διαχειριστές αντικειμένων. Στις αναπαραστάσεις αυτές το σχήμα του

εκάστοτε συστήματος αποθήκευσης μπορεί να βασίζεται είτε στα υπάρχοντα XML σχήματα [TBMM00] ή Ορισμούς Τύπων Εγγράφων (DTDs) [STH+99], είτε στα XML δεδομένα [FK99], [SKWW00], [DFS99] είτε τέλος σε ένα γενικό μοντέλο αναπαράστασης γράφων π.χ. *προσέγγιση ακμής*. Στην συνέχεια θα παρουσιάσουμε τις διαφορετικές προσεγγίσεις που έχουν προταθεί για την αναπαράσταση των XML γράφων.

3.1.2 Βασικά μοντέλα σχεσιακής αναπαράστασης

3.1.2.1 Προσέγγιση Ακμής

Η *προσέγγιση ακμής* (edge approach) η οποία παρουσιάζεται στην δημοσίευση [FK99] είναι η απλούστερη προσέγγιση για την αποθήκευση XML γράφων σε σχεσιακές βάσεις δεδομένων. Αντιστοιχεί σε μια γενική αναπαράσταση XML δεδομένων όπου το σχήμα που δημιουργείται είναι ανεξάρτητο της δομής των XML δεδομένων. Στην προσέγγιση αυτή ο XML γράφος αποθηκεύεται σε ένα μοναδικό πίνακα, τον πίνακα *Edge*. Σε κάθε κόμβο (XML στοιχείο) του γράφου το οποίο περιέχει κάποιο άλλο στοιχείο ή γνώρισμα (δηλαδή δεν περιέχει μόνο απλό κείμενο) αποδίδεται ένα αναγνωριστικό (βλέπε εικόνα. 3.1). Μια εγγραφή του πίνακα *Edge* αντιπροσωπεύει μια ακμή του γράφου και περιέχει το αναγνωριστικό του κόμβου αρχής, το αναγνωριστικό κόμβου προορισμού ή το κείμενο αν ο κόμβος προορισμού περιέχει μόνο κείμενο, την ετικέτα (tag) του στοιχείου προορισμού και ένα πεδίο (ordinal) που δείχνει την σειρά του στοιχείου/γνωρίσματος σε σχέση με τα άλλα χαρακτηριστικά του κόμβου αρχής. Στην εικόνα 3.2 απεικονίζεται ο πίνακας Edge που αντιστοιχεί σε ένα μέρος του γράφου της εικόνας 3.1.

sourceid	ordinal	tag	targetid	data
-	1	book	1	
1	1	title		Building Corp…
1	2	author	2	
2	1	name	3	
2	2	address	4	
3	1	firstname		Clive
3	2	lastname		Finkelstein

Εικόνα 3.2. Προσέγγιση ακμής: πίνακας Edge

Οι δείκτες που προτείνεται να δημιουργηθούν στον πίνακα Edge είναι οι εξής: ένας δείκτης στο πεδίο *sourceid* και ένας δείκτης στο συνδυασμό των πεδίων *{tag, targetid}*. Ο πρώτος δείκτης είναι χρήσιμος για την επαναδημιουργία ενός αντικειμένου

όταν δίνεται το αναγνωριστικό του. Ο δείκτης στα πεδία *{tag, targetid}* διευκολύνει την διάσχιση του γράφου δίνοντας συνθήκες επιλογής (backward traversal). Για παράδειγμα, χρησιμοποιείται για να απαντηθεί η ερώτηση *"Βρες τα αντικείμενα που έχουν επίθετο (lastname) Finkelstein"*.

3.1.2.2 Προσέγγιση γνωρίσματος

Στην *προσέγγιση γνωρίσματος* (Attribute approach) [FK99] δημιουργείται ένας καινούριος πίνακας για κάθε στοιχείο ή γνώρισμα που υπάρχει στα XML δεδομένα. Η προσέγγιση αυτή προκύπτει από την οριζόντια διαίρεση του πίνακα *Edge* με βάση το πεδίο *tag*. Κάθε πίνακας που δημιουργείται έχει τα πεδία (*sourceid, ordinal, targeteid, data*). Στην εικόνα 3.3 απεικονίζονται δύο από τους πίνακες που δημιουργούνται για τα XML δεδομένα του παραδείγματος μας. Όσον αφορά τους δείκτες προτείνεται να κατασκευαστεί ένας δείκτης στο πεδίο *sourceid* και ένας δείκτης στο πεδίο *targetid* για τους ίδιους λόγους με αυτούς που αναφέρθηκαν παραπάνω.

title

sourceid	ordinal	targetid	data
1	1		Building Corp...

author

sourceid	ordinal	targetid	data
1	2	2	

Εικόνα 3.3. Πίνακες προσέγγισης γνωρίσματος.

3.1.2.3 Σύγκριση προσέγγισης γνωρίσματος και προσέγγισης ακμής.

Στην δημοσίευση [FK99] γίνεται σύγκριση των δύο παραπάνω προσεγγίσεων. Εξετάζεται το μέγεθος της βάσης που απαιτείται σε κάθε περίπτωση, ο χρόνος που απαιτείται για την φόρτωση των δεδομένων, ο χρόνος που απαιτείται για την εκτέλεση ενός συνόλου επερωτήσεων που θεωρούνται ως σημείο αναφοράς και τέλος ο χρόνος που απαιτείται για διάφορα είδη ενημέρωσης ενός XML αρχείου. Τα αποτελέσματα που παρατίθενται έχουν ληφθεί για μέγεθος XML δεδομένων ίσο με **80** MB και ο συνολικός αριθμός διαφορετικών γνωρίσματων/στοιχείων είναι **20**.

Μέγεθος Βάσης: Διαπιστώθηκε ότι η προσέγγιση *γνωρίσματος* απαιτεί λιγότερο χώρο. Αυτό οφείλεται στο γεγονός ότι στην προσέγγιση *ακμής* τα ονόματα των στοιχείων και γνωρισμάτων αποθηκεύονται στην βάση όσες φορές εμφανίζονται και στα XML δεδομένα. Αντίθετα στην προσέγγιση *γνωρίσματος* αποθηκεύονται μόνο μια φορά σαν ονόματα των πινάκων. Ο συνολικός χώρος που απαιτείται για την αποθήκευση των XML δεδομένων (συμπεριλαμβανομένων και δεικτών) και στις δύο προσεγγίσεις είναι

μεγαλύτερος 2-3 φορές από τον χώρο που καταλαμβάνουν τα XML δεδομένα. Οι δείκτες καταλαμβάνουν πάνω από το 40% του συνολικού χώρου. Ο πίνακας 3.1 δείχνει το μέγεθος της βάση για XML δεδομένα μεγέθους 80 MB.

	Ακμής	**Γνωρίσματος**
Μέγεθος Δεδομένων (MB)	122	105
Μέγεθος Δεικτών (MB)	86	71
Συνολικό Μέγεθος (MB)	***208***	***176***

Πίνακας 3.1. Μέγεθος βάσης για XML δεδομένα μεγέθους 80 MB.

Χρόνος Φόρτωσης: Ο χρόνος που απαιτείται για την ανάλυση (parsing), την αποθήκευση των δεδομένων στην βάση και την δημιουργία των δεικτών είναι περίπου *48* λεπτά για την προσέγγιση *ακμής* και *42* λεπτά για την προσέγγιση *γνωρίσματος,* είναι δηλαδή ελαφρά καλύτερος στην προσέγγιση *γνωρίσματος*. Αυτό οφείλεται στο γεγονός ότι τα ονόματα των στοιχείων και γνωρισμάτων αποθηκεύονται στην βάση μόνο μια φορά.

Χρόνος απόκρισης στις επερωτήσεις: Όσον αφορά την επερώτηση, στις περισσότερες περιπτώσεις η προσέγγιση *γνωρίσματος* δίνει καλύτερα αποτελέσματα. Αυτό οφείλεται στο γεγονός ότι στην προσέγγιση *ακμής* οι συζεύξεις στον μοναδικό και πιθανότατα πολύ μεγάλο πίνακα *Edge* είναι πολύ ακριβές. Όπως επίσης και στο γεγονός ότι τα περισσότερα δεδομένα του πίνακα *Edge,* τον οποίο επεξεργαζόμαστε σε κάθε επερώτηση, είναι άσχετα με την επερώτηση. Υπάρχουν βέβαια και περιπτώσεις όπου η προσέγγιση *ακμής* δίνει καλύτερα αποτελέσματα. Επερωτήσεις που αφορούν την επαναδημιουργία ενός αντικειμένου, δηλαδή την εύρεση των στοιχείων και γνωρισμάτων του, εκτελούνται γρηγορότερα στην προσέγγιση *ακμής* δεδομένου ότι απαιτείται η επερώτηση σε ένα μόνο πίνακα. Ενώ στην προσέγγιση *γνωρίσματος* πρέπει πρώτα να γίνουν επερωτήσεις στο σχήμα για την εύρεση των πινάκων, επερώτηση αυτών και στην συνέχεια ένωση των αποτελεσμάτων.

Το μειονέκτημα της προσέγγισης *γνωρίσματος* είναι η δημιουργία πολλαπλών πινάκων η οποία πιθανόν να προκαλεί σπατάλη χώρου. Επίσης υπάρχουν συστήματα βάσεων δεδομένων που επιτρέπουν την δημιουργία μικρού αριθμού πινάκων.

3.1.3 Δύο εναλλακτικές σχεσιακές αναπαραστάσεις

3.1.3.1 Παραλλαγή της προσέγγισης γνωρίσματος (XML Monet)

Μια παραλλαγή της προσέγγισης γνωρίσματος παρουσιάζεται στην δημοσίευση [SKWW00]. Στην προσέγγιση αυτή για κάθε μονοπάτι από την ρίζα του XML γράφου δημιουργείται ένας πίνακας δύο πεδίων. Το όνομα του πίνακα αντιστοιχεί στο όνομα του μονοπατιού που αντιπροσωπεύει. Για παράδειγμα, δύο από τους πίνακες που δημιουργούνται για τον XML γράφο της εικόνας 3.1 είναι οι πίνακες *book.title* και *book.author* (εικόνα 3.4).

book.title

source	target
1	Building Corp…

book.author

source	tagret
1	2

Εικόνα 3.4. Δυαδικοί πίνακες που αντιστοιχούν σε μονοπάτια στην προσ. XML Monet.

Η βασική διαφορά της προσέγγισης αυτής από την προσέγγιση *γνωρίσματος* έγκειται στο γεγονός ότι στην προσέγγιση *γνωρίσματος* όλα τα στοιχεία/γνωρίσματα με τα ίδιο όνομα αντιστοιχίζονται στον ίδιο πίνακα ανεξάρτητα από το μονοπάτι που βρίσκονται. Για παράδειγμα, έστω ότι στον XML γράφο υπάρχουν τα μονοπάτια *book.title* και *magazine.title.* Για την αναπαράσταση των XML δεδομένων, στην προσέγγιση *γνωρίσματος* θα δημιουργηθεί ένας μόνο πίνακας με όνομα *title*, ενώ σ΄ αυτήν την προσέγγιση θα δημιουργηθούν δύο πίνακες, οι πίνακες *book.title* και *magazine.title.*

Στον πίνακα 3.2 παρατίθενται στατιστικά στοιχεία τα οποία παρουσιάζονται στην δημοσίευση [SKWW00] για το μέγεθος της βάσης σε σχέση με το μέγεθος των αρχικών δεδομένων, τον αριθμό των πινάκων που δημιουργούνται και το χρόνο που απαιτείται για την ανάλυση (parsing) και αποθήκευση των δεδομένων στην βάση. Παρατηρούμε ότι στην πρώτη περίπτωση το μέγεθος της βάσης είναι μικρότερο από το αρχικό αρχείο. Αυτό οφείλεται αρχικά στο γεγονός ότι τα ονόματα των γνωρισμάτων/στοιχείων αποθηκεύονται μόνο μια φορά σαν ονόματα των πινάκων και επίσης στο γεγονός ότι στο σύστημα Monet[11] το οποίο χρησιμοποιείται για την αποθήκευση των δεδομένων ένα αλφαριθμητικό αποθηκεύεται μόνο μια φορά.

[11] http://www.cwi.nl/~monet

Μέγεθος XML Δεδομένων	Μέγεθος XML Monet βάσης	#Πινάκων	Συνολ. Χρόνος Φόρτωσης
46.6 MB	44.2 MB	187	30.4 s
7.9 MB	8.2 MB	95	4.5 s
56.1 MB	95.6 MB	2587	56.6 s

Πίνακας 3.2. Μέγεθος βάσης και χρόνος φόρτωσης στην προσέγγιση XML Monet

Μεταβάλλοντας το μέγεθος των XML δεδομένων διαπιστώθηκε ότι το μέγεθος της βάσης και ο χρόνος φόρτωσης των δεδομένων αυξάνεται γραμμικά σε σχέση με το μέγεθος των XML δεδομένων. Αντίστοιχα παρατηρείται ότι ο χρόνος απόκρισης των επερωτήσεων είναι γραμμικός σε σχέση με το μέγεθος της βάσης.

3.1.3.2 Παραλλαγή της προσέγγισης ακμής (Προσέγγιση SYU)

Στην δημοσίευση [SYU99] παρουσιάζεται μια ακόμα προσέγγιση για την αποθήκευση XML δέντρων σε σχεσιακές βάσεις δεδομένων. Το σχήμα της βάσης αποτελείται από τέσσερις πίνακες. Οι πίνακες *Element*, *Attribute*, *Text* παριστάνουν τα διαφορετικά είδη κόμβων που υπάρχουν στο δέντρο. Ο πίνακας *Path* περιέχει τα μονοπάτια που υπάρχουν στο XML δέντρο. Να σημειώσουμε ότι στην προσέγγιση αυτή σε κάθε στοιχείο, γνώρισμα ή απλό κείμενο που βρίσκεται στον XML γράφο αποδίδεται ένα ζευγάρι τιμών (x,y) που δηλώνει την θέση του στον γράφο. Με βάση το ζευγάρι τιμών μπορεί να διαπιστωθεί αν υπάρχει σχέση προγόνου-απογόνου μεταξύ των κόμβων. Ο πίνακας *Element* περιέχει πληροφορία για τα στοιχεία, συγκεκριμένα περιέχει το αναγνωριστικό του αρχείου που βρίσκεται το στοιχείο, το μονοπάτι, την σειρά που έχει σε σχέση με τα άλλα παιδιά του ίδιου πατέρα (index, reindex) και την θέση του στον XML γράφο. Ο πίνακας *Attribute* περιέχει πληροφορία για τους κόμβους που αντιστοιχούν σε γνωρίσματα, περιέχει το αναγνωριστικό του εγγράφου, το μονοπάτι, την τιμή του γνωρίσματος και την θέση του στο XML γράφο. Τέλος, ο πίνακας Text περιέχει πληροφορία για το απλό κείμενο, συγκεκριμένα περιέχει το αναγνωριστικό του εγγράφου, το μονοπάτι, το κείμενο και την θέση του κειμένου στο δέντρο.

Path

pathexp	pathID
/book	1
/book/title	2
/book/author	3

Element

docID	pathID	index	reindex	pos
1	1	0	-1	(x, y)
1	2	0	-1	(x2, y2)
1	3	0	-1	(x2, y2)

Attribute

docID	pathID	attvalue	pos

Text

docID	pathID	value	pos
1	2	Building Corp…	(x3, y3)

Εικόνα 3.5 Σχεσιακό σχήμα προσέγγισης SYU.

Για την εκτέλεση των επερωτήσεων χρησιμοποιείται τόσο η θέση των κόμβων στο γράφο όσο και ταίριασμα προτύπου βασισμένο στα μονοπάτια που υπάρχουν στον πίνακα *path*. Η κωδικοποίηση που έχει γίνει για την θέση των κόμβων η οποία επιτρέπει την εύρεση σχέσεων ιεραρχίας (προγόνου – απογόνου) είναι απαραίτητη για την αποδοτική απόκριση επερωτήσεων που αναφέρονται σε διάσχιση μονοπατιών.

3.1.3.3 Σύγκριση των δύο παραλλαγών

Στην δημοσίευση [SKWW00] γίνεται σύγκριση των προσεγγίσεων XML Monet και SYU. Τα αποτελέσματα που έχουν ληφθεί επιβεβαιώνουν το γεγονός ότι ο τεμαχισμός των XML δεδομένων σε πολλαπλούς πίνακες όπως γίνεται στην προσέγγιση XML Monet επιταχύνει κατά ένα μεγάλο παράγοντα την εκτέλεση των επερωτήσεων εξαιτίας του ότι μειώνεται ο όγκος των δεδομένων που πρέπει να επεξεργαστούν κατά τις επερωτήσεις. Στη προσέγγιση *SYU* (όπως και στην προσέγγιση *ακμής*) όλα τα δεδομένα πρέπει να επεξεργαστούν σε κάθε επερώτηση.

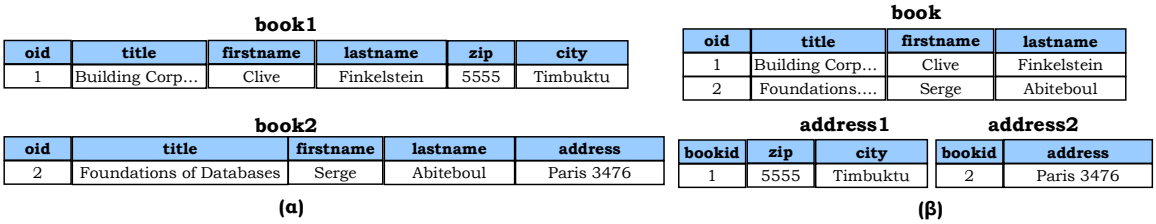
3.1.4 Χρήση XML δομών στην σχεσιακή αναπαράσταση

3.1.4.1 Αυτόματη εξαγωγή σχημάτων

Στην προσέγγιση *STORED* [DFS99] η σχεσιακή αναπαράσταση της βάσης εξάγεται αυτόματα με βάση τα υπάρχοντα XML δεδομένα, αξιοποιώντας την πιθανή κανονικότητα που υπάρχει σ' αυτά.

Ο αλγόριθμος που χρησιμοποιείται για την εξαγωγή του σχήματος έχει σαν παραμέτρους εισόδου το μέγιστο αριθμό των πινάκων που μπορούν να σχηματιστούν, το μέγιστο αριθμό πεδίων που μπορεί να έχει ένας πίνακας και το μέγιστο χώρο που μπορούν να καταλαμβάνουν τα δεδομένα στην βάση. Επίσης έχει σαν είσοδο ένα αριθμό

c που δηλώνει πόσες φορές πρέπει να εμφανιστεί ένα γνώρισμα σε ένα αντικείμενο για να δημιουργηθεί ένας πίνακας γι' αυτό το γνώρισμα. Για παράδειγμα, αν το c τίθεται ίσο με 3 τότε αν ένα γνώρισμα π.χ. address εμφανίζεται σε κάποιο αντικείμενο το πολύ μέχρι δύο φορές τότε το γνώρισμα θα αποθηκεύεται σαν δύο διαφορετικά πεδία ενός πίνακα, ενώ αν εμφανίζεται πάνω από δύο φορές θα δημιουργείται ένας ξεχωριστός πίνακας π.χ. address. Τέλος, δέχεται σαν παράμετρο έναν αριθμό που δηλώνει το ποσοστό των δεδομένων που πρέπει να αποθηκεύονται στην βάση. Μπορεί δηλαδή ένα ποσοστό δεδομένων να μην μπορεί να αποθηκευτεί σύμφωνα με το σχεσιακό σχήμα που θα εξαχθεί. Παράλληλα ο αλγόριθμος μπορεί να λάβει υπόψη του ένα συγκεκριμένο σύνολο επερωτήσεων και το σχήμα που τελικά θα εξάγει να βελτιστοποιεί την απόκριση των επερωτήσεων. Ο αλγόριθμος δίνει σαν αποτέλεσμα ένα σύνολο από πίνακες που αποτελούν το σχεσιακό σχήμα, τα πεδία των πινάκων και τα πεδία που πρέπει απαραίτητα να έχει ένα αντικείμενο για να καταχωρηθεί σε κάποιον από τους πίνακες.



Εικόνα 3.6. Δύο πιθανά σχεσιακά σχήματα που εξάγονται από την προσ. STORED.

Έστω ότι εκτός από τα XML δεδομένα της εικόνας 3.1 όπου περιγράφεται ένα βιβλίο υπάρχει και η περιγραφή ενός άλλου βιβλίου όπου η διεύθυνση του συγγραφέα (address) δεν περιέχει άλλα στοιχεία αλλά είναι ένα απλό αλφαριθμητικό π.χ. "Paris 3476". Δύο πιθανά σχεσιακά σχήματα για την αναπαράσταση των παραπάνω XML δεδομένων απεικονίζονται στην εικόνα 3.6. Το πρώτο σχήμα αποτελείται από τους πίνακες *book1* και *book2*. Στον πίνακα *book2* αποθηκεύονται τα βιβλία στα οποία η διεύθυνση (address) είναι αλφαριθμητικό. Στον πίνακα *book1* τα βιβλία στα οποία η διεύθυνση αναπαριστάνεται σαν σύνθετο αντικείμενο. Το δεύτερο σχήμα αποτελείται από τον πίνακα *book* όπου αποθηκεύονται όλα τα βιβλία και τους πίνακες *address1* και *address2* που αποθηκεύουν τα διαφορετικά είδη διευθύνσεων. Βάση των παραμέτρων που έχουν καθοριστεί και των δεδομένων που υπάρχουν ο αλγόριθμος προτείνει τελικά το βέλτιστο σχήμα.

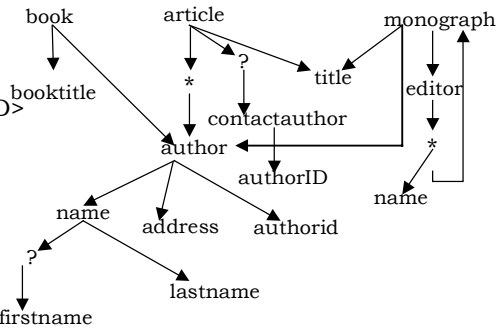
Η παραπάνω μέθοδος δουλεύει ικανοποιητικά μόνο όταν υπάρχει αρκετή ομοιομορφία στα δεδομένα. Δεδομένα που δεν είχαν ληφθεί υπόψη κατά την δημιουργία

του σχήματος ίσως να μην μπορούν να αποθηκευτούν στο επιλεγμένο σχήμα. Για παράδειγμα πρόβλημα προκύπτει αν προστεθεί ένας συγγραφέας με το γνώρισμα *τηλέφωνο* (*phone*). Για το σκοπό αυτό για την αποθήκευση των δεδομένων εκτός από την σχεσιακή βάση χρησιμοποιείται και ένα σύστημα αποθήκευσης ημιδομημένων δεδομένων όπου αποθηκεύονται τα δεδομένα που δεν προσαρμόζονται στο υπάρχον σχεσιακό σχήμα.

3.1.4.2 Αξιοποίηση XML σχημάτων και Ορισμών Τύπων Εγγράφων

Στην δημοσίευση [STH+99] παρουσιάζεται μια διαφορετική προσέγγιση στην εξαγωγή σχεσιακού σχήματος για την αναπαράσταση XML δεδομένων. Το σχεσιακό σχήμα εξάγεται με βάση τα XML σχήματα ή Ορισμούς Τύπων Εγγράφων, σε αντίθεση με προσεγγίσεις που έχουμε αναφέρει μέχρι τώρα στις οποίες το σχήμα εξάγεται από τα XML δεδομένα (π.χ. προσέγγιση *γνωρίσματος)* ή είναι ανεξάρτητο από αυτά (π.χ. προσέγγιση *ακμής* και *SYU).* Η προσέγγιση αυτή, σε αντίθεση με το STORED που μπορεί να χειριστεί οποιαδήποτε δεδομένα, βασίζεται στην υπόθεση ότι τα XML δεδομένα βασίζονται σε γνωστά XML σχήματα ή DTDs που έχουν ληφθεί υπόψη στην δημιουργία του σχεσιακού σχήματος.

```
<!ELEMENT book (booktitle, author)
<!ELEMENT article (title, author*, contactauthor)>
<!ELEMENT contactauthor EMPTY>
<!ATTLIST contactauthor authorID IDREF IMPLIED>
<!ELEMENT monograph (title, author, editor)>
<!ELEMENT editor (monograph*)>
<!ATTLIST editor name CDATA #REQUIRED>
<!ELEMENT author (name, address)>
<!ATTLIST author id ID #REQUIRED>
<!ELEMENT name (firstname?, lastname)>
```



Εικόνα 3.7. Ορισμός τύπου εγγράφου και γράφος του.

Στην συνέχεια θα παρουσιάσουμε συνοπτικά τις τρεις παραλλαγές, *Basic, Shared* και *Hybrid* που χρησιμοποιούνται για την εξαγωγή του σχεσιακού σχήματος.

book (bookID: integer, book.booktitle : string, book.author.name.firstname: string, book.author.name.lastname: string, book.author.address: string, author.authorid: string)
booktitle (booktitleID: integer, booktitle: string)
article (articleID: integer, article.contactauthor.authorid: string, article.title: string)
article.author (article.authorID: integer, article.author.parentID: integer, article.author.name.firstname: string, article.author.name.lastname: string, article.author.address: string, article.author.authorid: string)
contactauthor (contactauthorID: integer, contactauthor.authorid: string)
title (titleID: integer, title: string)
monograph (monographID: integer, monograph.parentID: integer, monograph.title: string, monograph.editor.name: string,monograph.author.name.firstname: string, monograph.author.name.lastname: string, monograph.author.address: string, monograph.author.authorid: string)
editor (editorID: integer, editor.parentID: integer, editor.name: string)
editor.monograph (editor.monographID: integer, editor.monograph.parentID: integer, editor.monograph.title: string,editor.monograph.author.name.firstname: string, editor.monograph.author.name.lastname: string, editor.monograph.author.address: string, editor.monograph.author.authorid: string)
author (authorID: integer, author.name.firstname: string, author.name.lastname: string, author.address: string, author.authorid: string)
name (nameID: integer, name.firstname: string, name.lastname: string)
firstname (firstnameID: integer, firstname: string)
lastname (lastnameID: integer, lastname: string)
address (addressID: integer, address: string)

Εικόνα 3.8. Σχεσιακό σχήμα με βάση την παραλλαγή Basic.

Στην παραλλαγή *Basic* για κάθε στοιχείο που υπάρχει στο DTD (ή XML σχήμα) δημιουργείται ένας πίνακας. Τα πεδία κάθε πίνακα είναι όσο το δυνατόν περισσότεροι απόγονοι του στοιχείου. Συγκεκριμένα είναι όλοι οι απόγονοι (στοιχεία-γνωρίσματα) που δεν έχουν το τελεστή * και δεν σχηματίζουν αναδρομή π.χ. *monograph*. Για παράδειγμα ο πίνακας που θα δημιουργηθεί για το στοιχείο *author* θα έχει σαν πεδία τα *firstname, lastname, address* και *authorid*. Παράλληλα πίνακες θα δημιουργηθούν και για τα στοιχεία *firstname, lastname* και *address* εφόσον για κάθε στοιχείο σχηματίζεται ένας πίνακας. Στην εικόνα 3.8 φαίνεται το σχεσιακό σχήμα που εξάγεται για το DTD της εικόνας 3.7. Επίσης για κάθε στοιχείο που μπορεί να εμφανίζεται πολλαπλές φορές, δηλαδή έχει τον τελεστή *, δημιουργείται ένας διαφορετικός πίνακας. Για παράδειγμα, όταν δημιουργούμε τον πίνακα *article* επειδή ο αριθμός των συγγραφέων μπορεί να είναι μεγαλύτερος του ένα, δημιουργούμε ένα νέο πίνακα *article.author*. Παρατηρούμε στο σχεσιακό σχήμα της εικόνας ότι έχουν δημιουργηθεί δύο πίνακες για το στοιχείο *author*,

οι πίνακες *article.author* και *author*. Ο δεύτερος πίνακας δημιουργείται επειδή όπως αναφέραμε για κάθε στοιχείο δημιουργείται ένας πίνακας. Επίσης πίνακας δημιουργείται όταν υπάρχει αναδρομή π.χ. μεταξύ *editor-monograph* όπου δημιουργείται ο πίνακας *editor.monograph*. Δεν θα αναλύσουμε περαιτέρω την παραλλαγή αυτή εφόσον, όπως είναι ήδη φανερό, παράγει πάρα πολλούς πίνακες.

Στην *Shared* κάθε στοιχείο αποθηκεύεται σε ένα μόνο πίνακα. Η βασική ιδέα στην παραλλαγή αυτή είναι να αναγνωριστούν οι κόμβοι που στην Basic αποθηκεύονται σε πολλαπλούς πίνακες όπως τα στοιχεία firstname, lastname και address. Πίνακες δημιουργούνται μόνο για τα στοιχεία τα οποία στον DTD γράφο υπάρχουν πάνω από ένα βέλη που να δείχνουν σ' αυτά π.χ. *author* όπως επίσης και για τα στοιχεία από τα οποία ξεκινάει ο γράφος π.χ. *book*, *article*. Επίσης, όπως και στην Basic, δημιουργούνται καινούριοι πίνακες για τα στοιχεία που στον γράφο ακολουθούν το τελεστή * ή σχηματίζουν κύκλο. Τα πεδία ενός πίνακα που αντιστοιχεί σε ένα στοιχείο X είναι όλοι οι κόμβοι που βρίσκονται κάτω από τον κόμβο αυτόν και δεν έχουν δημιουργηθεί πίνακες γι' αυτούς. Στην εικόνα 3.9 απεικονίζεται το σχήμα που δημιουργείται για το DTD της εικόνας 3.7. Παρατηρούμε ότι ο αριθμός των πινάκων που δημιουργούνται είναι πολύ μικρότερος στην προσέγγιση Shared.

book (bookID:integer, book.booktitle.isroot:boolean, book.booktitle:string)
article (articleID: integer, article.contactauthor.isroot: boolean,
article.contactauthor.authorid: string)
monograph (monographID: integer,monograph.parentID: integer, monograph.parentCODE: integer, monograph.editor.isroot: boolean, monograph.editor.name: string)
title (titleID: integer, title.parentID: integer, title.parentCODE: integer, title: string)
author (authorID: integer, author.parentID: integer, author.parentCODE: integer,
author.name.isroot: boolean,author.name.firstname.isroot: :boolean,
author.name.firstname: string, author.name.lastname.isroot:
boolean,author.name.lastname: string, author.address.isroot: boolean, author.address:
string, author.authorid: string)

Εικόνα 3.9. Σχεσιακό σχήμα με βάση την παραλλαγή Shared.

Συγκρίνοντας τις δύο παραπάνω προσεγγίσεις με βάση την απόκριση στις επερωτήσεις, παρατηρείται ότι για μια επερώτηση που αναφέρεται στους συγγραφείς στην προσέγγιση Shared χρειάζεται να επεξεργαστεί ένα μόνο πίνακα (author) ενώ στην προσέγγιση Basic 5 πίνακες. Όμως η προσέγγιση Shared απαιτεί περισσότερες συζεύξεις στην εκτέλεση των επερωτήσεων. Η *Hybrid* είναι μια παραλλαγή των *Shared* και *Basic* η

οποία μειώνει τον αριθμό των συζεύξεων και ταυτόχρονα δεν δημιουργεί πολύ μεγάλο αριθμό πινάκων όπως η Basic.

Η *Hybrid* μοιάζει με την *Shared*. Η βασική διαφορά τους είναι ότι στη *Hybrid* πεδία των πινάκων γίνονται και τα στοιχεία για τα οποία ο αριθμός ακμών που δείχνει σ' αυτά είναι μεγαλύτερος του ένα, αρκεί στον DTD γράφο να μην ακολουθούν μετά το σύμβολο * ή να μην σχηματίζουν αναδρομή (βλέπε πίνακα book στις δύο προσεγγίσεις). Το σχεσιακό σχήμα για τον γράφο του DTD της εικόνας 3.7 απεικονίζεται στην εικόνα 3.10.

book (bookID: integer, book.booktitle.isroot: boolean, book.booktitle : string, author.name.firstname: string, author.name.lastname: string, author.address: string, author.authorid: string)
article (articleID: integer, article.contactauthor.isroot: boolean, article.contactauthor.authorid: string, article.title.isroot: boolean, article.title: string)
monograph (monographID: integer, monograph.parentID: integer, monograph.parentCODE: integer, monograph.title: string, monograph.editor.isroot: boolean, monograph.editor.name: string, author.name.firstname: string, author.name.lastname: string, author.address: string, author.authorid: string)
author (authorID: integer, author.parentID: integer, author.parentCODE: integer, author.name.isroot: boolean,author.name.firstname.isroot: boolean, author.name.firstname: string, author.name.lastname.isroot: boolean, author.name.lastname: string, author.address.isroot: boolean, author.address: string, author.authorid: string)

Εικόνα 3.10. Σχεσιακό σχήμα με βάση την παραλλαγή Hybrid.

Για την αξιολόγηση των τριών παραλλαγών στην δημοσίευση [STH+99] υπολογίζεται ο μέσος αριθμός SQL joins που απαιτούνται για την αποτίμηση εκφράσεων μονοπατιού (path expressions) μήκους N. Η μετρική αυτή χρησιμοποιείται επειδή οι εκφράσεις μονοπατιού είναι το βασικότερο στοιχείο των γλωσσών που έχουν προταθεί για την επερώτηση ημιδομημένων δεδομένων. Στις συγκρίσεις δεν συμμετέχει η προσέγγιση *Basic* εξαιτίας του μεγάλου αριθμού πινάκων που δημιουργεί. Τα αποτελέσματα δείχνουν ότι για τον υπολογισμό των εκφράσεων μονοπατιού η *Hybrid* απαιτεί λιγότερες συζεύξεις για την εκτέλεση μιας SQL επερώτησης αλλά δημιουργεί περισσότερες SQL επερωτήσεις. Το συμπέρασμα είναι ότι δεν είναι φανερό εκ' των προτέρων ποια από τις δύο προσεγγίσεις απαιτεί συνολικά λιγότερες συζεύξεις, αλλά εξαρτάται από την δομή του Ορισμού Τύπου Εγγράφου.

3.1.5 Αποθήκευση XML δεδομένων σε διαχειριστές αντικειμένων

3.1.5.1 Προσέγγιση SM Object

Η προσέγγιση SM Object [TWCZ00] χρησιμοποιεί σαν σύστημα αποθήκευσης το σύστημα Shore[12] το οποίο είναι ένας διαχειριστής αντικειμένων[13] (object manager). Για κάθε XML αρχείο δημιουργείται ένα μόνο αντικείμενο το οποίο καλείται αντικείμενο-αρχείου (εικόνα 3.11). Αν δημιουργούνταν ένα αντικείμενο για κάθε στοιχείο θα απαιτούνταν μεγάλος χώρος αποθήκευσης δεδομένου ότι τα στοιχεία είναι συνήθως μικρά. Τα στοιχεία αντιστοιχίζονται σε μέρη αντικειμένου (lightweight αντικείμενα – l-αντικείμενα). Ένα αντικείμενο αρχείου αποτελείται από το σύνολο των l-αντικειμένων. Η μορφή ενός l-αντικειμένου είναι η εξής:

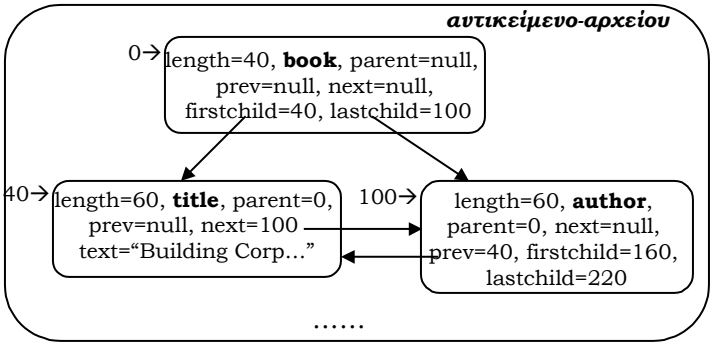
length	flag	tag	parent	prev	next	opt_child	opt_attr	opt_text

Η μετατόπιση/θέση (offset) του l-αντικείμενου μέσα στο αντικείμενο αρχείου χρησιμοποιείται σαν αναγνωριστικό (ele_id). Το πεδίο length δηλώνει το συνολικό μήκος ενός l-αντικειμένου. Το πεδίο tag δηλώνει το όνομα του XML στοιχείου. Το πεδίο parent περιέχει το αναγνωριστικό του l-αντικειμένου που αποτελεί τον πατέρα του στην ιεραρχία. Τα πεδία prev και next δείχνουν τα διπλανά στοιχεία. Τα πεδία child, attr και text είναι προαιρετικά. Το πεδίο child δείχνει το πρώτο και το τελευταίο παιδί ενός l-αντικειμένου. Το πεδίο attr περιέχει τα ζευγάρια γνώρισμα-τιμή του XML στοιχείου. Το κείμενο τοποθετείται στο πεδίο text αν το κείμενο είναι το μόνο παιδί του στοιχείου.

Οι δείκτες που δημιουργούνται είναι οι εξής: Ένας B-tree δείκτης ο οποίος αντιστοιχεί το συνδυασμό (tag, text) στο αναγνωριστικό του στοιχείου. Χρησιμοποιείται για την ανάκτηση των στοιχείων με ένα συγκεκριμένο tag. Ένας δεύτερος δείκτης αντιστοιχεί το συνδυασμό (parent, tag) στο αναγνωριστικό του στοιχείου. Είναι χρήσιμος για την ανάκτηση των παιδιών ενός στοιχείου με μια συγκεκριμένη ετικέτα (tag).

[12] http://www.cs.wisc.edu/shore/alpha/overview.html

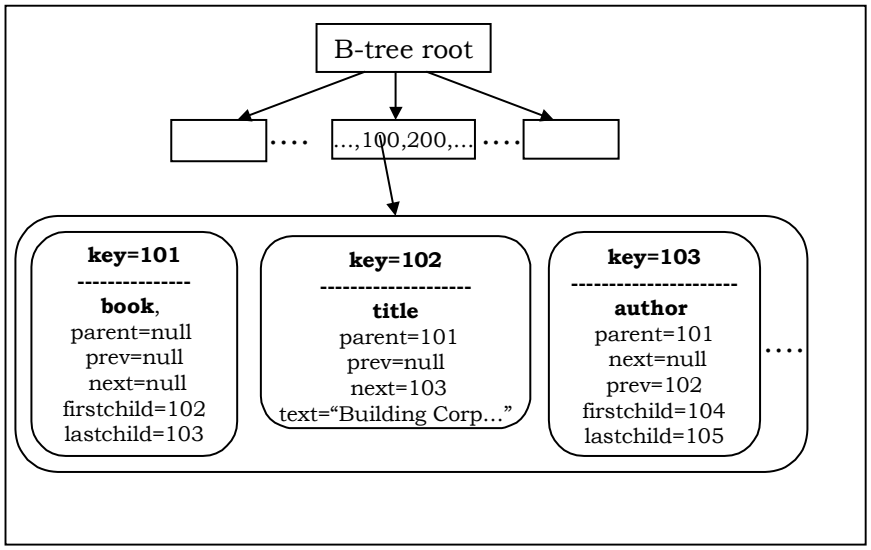
[13] Ο διαχειριστής αντικειμένων διαχειρίζεται την αποθήκευση αντικείμενων σε φυσικό επίπεδο. Αποτελεί την βάση για την δημιουργία συστημάτων διαχείρισης βάσεων δεδομένων.



Εικόνα 3.11. Αντικείμενο-αρχείου.

3.1.5.2 Προσέγγιση B-tree

Η παραπάνω προσέγγιση παρουσιάζει προβλήματα στην ενημέρωση των αντικειμένων. Όταν τα l-αντικείμενα ενημερώνονται και αυξάνεται ο χώρος που απαιτείται για την αποθήκευσή τους τότε μαρκάρονται σαν άκυρα και τα νέα l-αντικείμενα προσαρτώνται στο τέλος του αντικειμένου-αρχείου. Έτσι δημιουργείται σπατάλη χώρου, το αντικείμενο χωρίζεται σε κομμάτια και επίσης πρέπει να ενημερωθούν τα l-αντικείμενα που δείχνουν σ' αυτό. Στην B-Tree προσέγγιση (εικόνα 3.12), η οποία είναι επέκταση της προηγούμενης, αποδίδεται στα l-αντικείμενα ένα αναγνωριστικό που δεν εξαρτάται από την αποθήκευση τους σε φυσικό επίπεδο. Στα l-αντικείμενα που αντιστοιχούν στα στοιχεία ενός XML αρχείου αποδίδονται διαδοχικά αναγνωριστικά. Τα αναγνωριστικά αυτά αποτελούν τα κλειδιά του B-tree που σχηματίζεται. Όταν ενημερώνεται ένα l-αντικείμενο δεν χρειάζεται να ενημερωθούν τα l-αντικείμενα που δείχνουν σ' αυτό εφόσον τα λογικά αναγνωριστικά είναι αμετάβλητα. Επίσης εφόσον ο κώδικας του B-tree διαχειρίζεται το χώρο στα φύλλα δεν χρειάζεται να μετακινηθούν τα αντικείμενα όταν μεγαλώνουν. Η προσέγγιση όμως αυτή έχει επιπλέον επιβάρυνση εφόσον πρέπει να ψάχνουμε μέσω του B-tree. Εφόσον όμως το l-αντικείμενο που ψάχνουμε είναι πολύ πιθανό βρίσκεται στο ίδιο φύλλο ή σε διπλανό η συνολική επιβάρυνση δεν είναι μεγάλη. Το θετικό είναι ότι ο χρόνος αναζήτησης είναι σταθερά αλγοριθμικός, εφόσον το δέντρο είναι ισορροπημένο. Οι δείκτες που χρησιμοποιούνται είναι ίδιοι με την προηγούμενη προσέγγιση.



Εικόνα 3.12. B-tree προσέγγιση.

3.1.6 Συνολική Σύγκριση προσεγγίσεων

Στην συνέχεια θα αναφέρουμε σύγκριση που έχει γίνει στην δημοσίευση [TWCZ00] ανάμεσα στις 2 τελευταίες προσεγγίσεις, στην προσέγγιση ακμής καθώς επίσης και την περίπτωση όπου τα δεδομένα αποθηκεύονται σε απλά ASCII αρχεία.

Μέγεθος Βάσης: Το μέγεθος του χώρου αποθήκευσης που απαιτείται για κάθε μια από τις παραπάνω προσεγγίσεις απεικονίζεται στο πίνακα 3.3. Το αρχικό μέγεθος των δεδομένων είναι 65 MB. Ο πίνακας δείχνει ξεχωριστά το μέγεθος που απαιτείται για την αποθήκευση των δεδομένων και την αποθήκευση του δείκτη που δημιουργείται στο συνδυασμό (tag, text). Παρατηρούμε ότι το μέγεθος της βάσης είναι περίπου 3-4 φορές μεγαλύτερο από το αρχικό μέγεθος του XML αρχείου. Η προσέγγιση *ακμής* απαιτεί περισσότερο χώρο για την αποθήκευση των XML δεδομένων.

	Ακμής	**SM Object**	**B-tree**	**ASCII**
Μέγεθος δεδομένων (MB)	127	98	101	65
Μέγεθος Δείκτη (MB)	109	114	94	
Συνολικό Μέγεθος (MB)	***236***	***212***	***195***	***65***

Πίνακας 3.3. Μέγεθος βάσης για XML δεδομένα μεγέθους 65 MB.

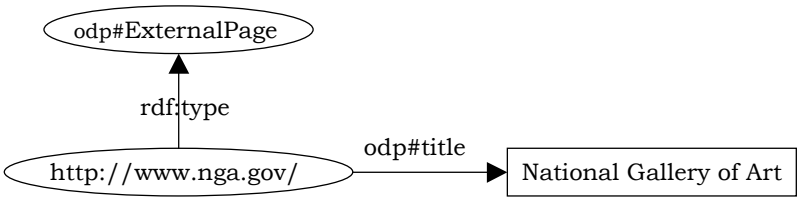
Χρόνος απόκρισης στις επερωτήσεις: Για την σύγκριση της απόδοσης των παραπάνω μεθόδων μετρήθηκαν τα αποτελέσματα σε ένα σύνολο επερωτήσεων. Τα αποτελέσματα

δείχνουν ότι η αποθήκευση των XML αρχείων σε ASCII αρχεία εξαιτίας των προβλημάτων που αναφέραμε έχει πολύ μεγάλους χρόνους απόκρισης στις επερωτήσεις. Επομένως η λύση αυτή θα μπορούσε να υιοθετηθεί μόνο σε περιπτώσεις που δεν ενδιαφέρει ο χρόνος απόκρισης.

Επίσης τα αποτελέσματα δείχνουν ότι όταν χρησιμοποιείται ένας διαχειριστής αντικειμένων για την αποθήκευση XML αρχείων η απόκριση σε όλους τους τύπους επερωτήσεων είναι πολύ γρηγορότερη σε σχέση με την απόκριση των επερωτήσεων σε σχεσιακά συστήματα διαχείρισης βάσεων δεδομένων. Συγκεκριμένα, η προσέγγιση *ακμής* είναι συνήθως 2 με 5 φορές αργότερη από τις άλλες προσεγγίσεις εξαιτίας του επιπλέον χρόνου που απαιτείται λόγω του ενδιάμεσου επιπέδου (query interface) της σχεσιακής βάσης μέσω του οποίου αποθηκεύονται τα δεδομένα ή εκτελούνται οι επερωτήσεις. Τα συστήματα σχεσιακών βάσεων δεδομένων έχουν το πλεονέκτημα ότι είναι επεκτάσιμα (scalable) και μπορούν να τρέχουν σε πολλές πλατφόρμες λογισμικού και υλικού (portable) εξαιτίας του πρότυπου SQL που υποστηρίζουν.

3.2 Σχεσιακή αναπαράσταση RDF ημιδομημένων δεδομένων

Στην ενότητα αυτή θα αναλύσουμε τις προσεγγίσεις που έχουν προταθεί για την αναπαράσταση των RDF γράφων. Στο σημείο αυτό πρέπει να τονιστεί ότι υπάρχει μια ιδιαιτερότητα στον RDF γράφο σε σχέση με τον XML γράφο. Ένας RDF γράφος έχει ετικέτες και στους κόμβους και στις ακμές. Η σημασιολογία των ετικετών τόσο των κόμβων όσο και των ακμών καθορίζεται από τα RDF σχήματα. Για παράδειγμα στον απλό RDF γράφο της εικόνας 3.13 η ετικέτα του κόμβου με URI *http://www.nga.gov/* είναι odp#ExternalPage ενώ της ακμής odp#title. Η παραπάνω διαφοροποίηση απαιτεί τροποποίηση των προσεγγίσεων που έχουν προταθεί για την αποθήκευση των XML δεδομένων έτσι ώστε να μπορέσουμε να επιτύχουμε αποτελεσματική αποθήκευση αλλά και αποδοτική επερώτηση των RDF μεταδεδομένων.



Εικόνα 3.13. RDF γράφος.

Στις κυριότερες προσπάθειες που έχουν γίνει για την 'μόνιμη' (persistent) αποθήκευση των RDF περιγραφών και στην συνέχεια την επερώτηση τους επιχειρείται η αποθήκευση τους σε σχεσιακές βάσεις δεδομένων [M00b]. Όλες οι σχεσιακές αναπαραστάσεις βασίζονται στην δημιουργία ενός πίνακα στον οποίο αποθηκεύονται οι τριάδες (p, s, o) στις όποιες αναλύεται ο RDF γράφος. Η αναπαράσταση αυτή θυμίζει την προσέγγιση *ακμής*. Παρακάτω θα περιγράψουμε τις διαφοροποιήσεις στις σχεσιακές αναπαραστάσεις που έχουν προταθεί.

3.2.1 Σχεσιακή αναπαράσταση RDF δεδομένων

1η προσέγγιση – Μοναδικός πίνακας

Η πιο απλή προσέγγιση [M00b] η οποία έχει προταθεί για την σχεσιακή αναπαράσταση RDF δεδομένων αποτελείται από ένα πίνακα, τον πίνακα *Triple* ο οποίος έχει τα πεδία *property*, *resource* και *value* που αντιστοιχούν στα τρία μέρη μιας RDF τριάδας (εικόνα 3.14). Το πεδίο *hint* δηλώνει αν η τριάδα υπάρχει στον RDF γράφο ή αν υπάρχει η υποστασιοποιημένη δήλωση. Ο πίνακας *Triple* θυμίζει τον πίνακα *Edge* της προσέγγισης ακμής. Στην περίπτωση βέβαια των RDF δεδομένων δεν μας ενδιαφέρει η σειρά με την οποία αναφέρονται τα στοιχεία/γνωρίσματα (δηλαδή οι ιδιότητες) ενός πόρου γι' αυτό και στον πίνακα *Triples* δεν υπάρχει το πεδίο ordinal.

Triple

property: varchar(255)	resource: varchar(255)	value: blob	hint: char(1)
odp#title	*http://www.nga.gov/*	*National Gallery of Art*	n
rdf:type	*http://www.nga.gov/*	odp#ExternalPage	n

Εικόνα 3.14. Σχεσ. Αναπαράσταση RDF δεδομένων με ένα μοναδικό πίνακα.

2η προσέγγιση

Σε μια δεύτερη προσέγγιση [L00b] που έχει προταθεί οι δυο βασικοί πίνακες που δημιουργούνται είναι οι πίνακες *resources* και *statements* οι οποίοι απεικονίζονται στην εικόνα 3.15. Ο πίνακας *resources* περιέχει τόσο τους πόρους όσο και τα literals. Το πεδίο *uri* έχει σαν τιμή το URI του πόρου ενώ το πεδίο *value* έχει σαν τιμή το αλφαριθμητικό (literal), προφανώς μόνο ένα από δύο τα πεδία έχει τιμή σε κάθε εγγραφή. Το πεδίο *id* περιέχει το ακέραιο αναγνωριστικό που αποδίδεται τόσο στους πόρους όσο και στα literals. Το πεδίο *lang* δηλώνει την γλώσσα που έχουν γραφτεί τα αλφαριθμητικά. Να σημειώσουμε ότι η κοινή αναπαράσταση πόρων και literals αντιβαίνει με το RDF όπου υπάρχει σαφής διαχωρισμός τους. Ο πίνακας *statements,* όπως στην προηγούμενη προσέγγιση ο πίνακας *Triples* περιέχει τις τριάδες. Όμως αντί για τα URIs των πόρων και τα Literals χρησιμοποιούνται τα αναγνωριστικά τους. Το πεδίο *fact* στον πίνακα *statements* δηλώνει αν στον RDF γράφο υπάρχει η δήλωση ή η υποστασιοποιημένη δήλωση. Σε κάθε τριάδα (pred, sub, obj) που αποθηκεύεται στον πίνακα *statements* αποδίδεται ένα id. Έτσι ώστε να μπορούν να αναπαρασταθούν οι περιγραφές που αποδίδονται σε τριάδες.

resources

id:int	uri: text	value:text	lang:text
1	odp#title		en
2	*http://www.nga.gov/*		en
3		*National Gallery of Art*	en
4	rdf:type		en
5	odp#ExternalPage		en

statements

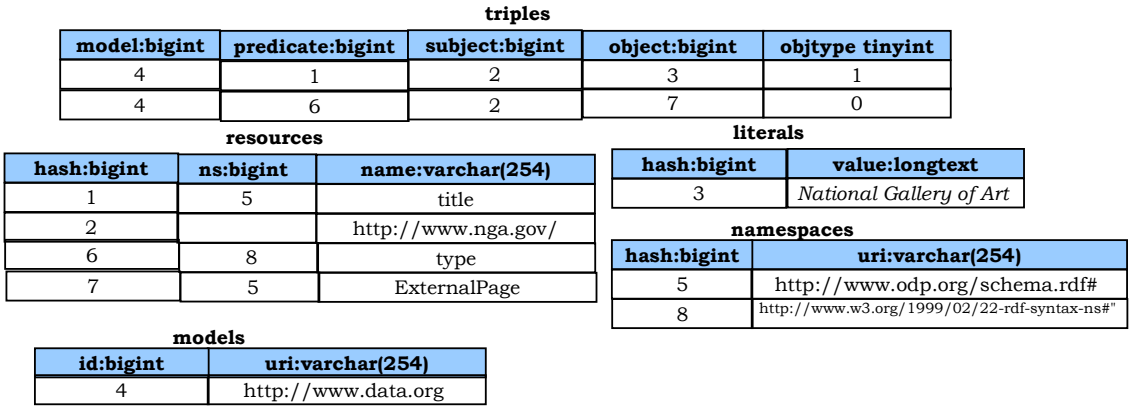
id:int	pred:int	sub:int	obj:int	fact:int
6	*1*	*2*	3	0
7	*4*	*2*	5	0

Εικόνα 3.15. Σχεσ. Αναπαράσταση RDF δεδομένων με απόδοση κωδικών στους πόρους και literals.

3η προσέγγιση

Η τρίτη σχεσιακή αναπαράσταση [M00b] που προτείνεται αποτελείται από τους πίνακες *triples, resources, namespaces, literals* και *models*. Ο πίνακας *triples* αντιστοιχεί στον πίνακα *statements* της παραπάνω αναπαράστασης. Επιπλέον στην αναπαράσταση αυτή εισάγεται η έννοια του συνόλου RDF περιγραφών (model). Για παράδειγμα, ένα σύνολο RDF περιγραφών αποτελούν οι περιγραφές που περιέχονται σε ένα συγκεκριμένο

αρχείο. Κάθε δήλωση έχει και το αναγνωριστικό του συνόλου περιγραφών στο οποίο περιέχεται. Τα σύνολα RDF περιγραφών αποθηκεύονται στον πίνακα models. Τα URIs των πόρων αποθηκεύονται στον πίνακα resources ενώ τα literals στον πίνακα literals. Τα URIs των πόρων χωρίζονται στο αναγνωριστικό που αποδίδεται στο χώρο ονοματοδοσίας (namespace) και στο όνομα.



Εικόνα 3.16. Σχεσ. Αναπαράσταση RDF δεδομένων

Όσον αφορά τα ευρετήρια, στον πίνακα *triples* δημιουργούνται τρία ευρετήρια, το πρώτο στο συνδυασμό των πεδίων (*subject*, *predicate*), το δεύτερο στο πεδίο *model* και το τρίτο στο συνδυασμό των πεδίων (*object*, *predicate*). Το πεδίο *hash* στους πίνακες *resources*, *literals* και *namespaces* ορίζεται σαν πρωτεύον κλειδί όπως επίσης και το πεδίο *id* στον πίνακα *models*.

3.2.1.1 Σύγκριση προσεγγίσεων

Στις προσεγγίσεις 2 και 3 γίνεται απόδοση αναγνωριστικών στους πόρους και τα literals. Αυτό έχει σαν αποτέλεσμα την μείωση του χώρου που απαιτείται για την αποθήκευση των RDF μεταδεδομένων. Ταυτόχρονα όμως αυξάνεται ο αριθμός των συζεύξεων που απαιτούνται για την εκτέλεση των επερωτήσεων.

3.2.2 Συστήματα αποθήκευσης RDF δεδομένων

Προς το παρόν έχουν υλοποιηθεί ελάχιστα συστήματα για την ‘μόνιμη’ αποθήκευση των RDF περιγραφών και στην συνέχεια την επερώτηση τους. Ένα από τα συστήματα αποθήκευσης που έχουν υλοποιηθεί είναι το ***RDF Data Store*** [RDS00] το

οποίο αποτελεί ένα από τα βασικά κομμάτια λογισμικού που έχει υλοποιηθεί στα πλαίσια του έργου *DESIRE* [DESIRE]. Για την αποθήκευση των μεταδεδομένων χρησιμοποιείται είτε ο διαχειριστής αντικειμένων BerkeleyDB[14], είτε η σχεσιακή βάση δεδομένων mySQL[15]. Η σχεσιακή αναπαράσταση που χρησιμοποιείται μοιάζει με την πρώτη αναπαράσταση που περιγράφτηκε. Δηλαδή οι RDF περιγραφές αποθηκεύονται με την μορφή τριάδων. Ορίζεται ένα σύνολο απλών συναρτήσεων (APIs) μέσω των οποίων οι εφαρμογές μπορούν να αποθηκεύσουν τριάδες, να σβήσουν τριάδες από το σύστημα αποθήκευσης ή να ανακτήσουν δεδομένα. Οι επερωτήσεις που επιτρέπονται είναι στο επίπεδο των τριάδων. Για παράδειγμα "Δώσε τις ιδιότητες του πόρου r και τις τιμές τους." Η "Ανάκτησε τους πόρους που έχουν την ιδιότητα p με τιμή x.".

Η **rdfDB** [G00] είναι ένα σύστημα αποθήκευσης RDF περιγραφών. Οι περιγραφές αποθηκεύονται στην βάση δεδομένων με την μορφή τριάδων. Παρέχει μια διεπαφή χρήσης που υποστηρίζει εντολές που μοιάζουν με τις SQL εντολές και επιτρέπουν την προσθήκη, διαγραφή και επερώτηση των τριάδων. Έχει υλοποιηθεί στην γλώσσα προγραμματισμού C πάνω από το διαχειριστή αντικειμένων Berkeley DB. Τα δεδομένα μπορούν να αποθηκευτούν στην βάση είτε δίνοντας την διεύθυνση του αρχείου που περιέχει τις RDF/XML περιγραφές, είτε προσθέτοντας τριάδες χρησιμοποιώντας την διεπαφή χρήσης. Για παράδειγμα η εντολή *insert into [database_name] (arc source target*) αποθηκεύει στην βάση την τριάδα *(arc source target).* Η σύνταξη που παρέχεται για τις επερωτήσεις έχει την μορφή "s*elect [variable1, variable2, ...] from {database} where [constraint1, constraint2, ...]".* Για παράδειγμα μια επερώτηση είναι η εξής: *"select ?x from db where (odp#title ?x 'National Gallery of Art')"* η οποία επιστρέφει τους πόρους που έχουν την ιδιότητα *odp#title* με τιμή *National Gallery of Art.* Όπως φαίνεται στην παραπάνω επερώτηση οι μεταβλητές αρχίζουν με το χαρακτήρα ?. Οι περιορισμοί (constraints) έχουν την μορφή *(arc source target*) όπου οποιοδήποτε από τα μέλη της τριάδας μπορεί να είναι μεταβλητή ή πόρος. Το πεδίο *target* μπορεί επίσης να είναι αλφαριθμητικό ή ακέραιος. Εκτός από την δυνατότητα φόρτωσης ενός αρχείου παρέχει και την δυνατότητα διαγραφής όλων των τριάδων που έχουν φορτωθεί στην βάση από ένα συγκεκριμένο αρχείο.

[14] http://www.sleepycat.com/
[15] http://www.mysql.com/

Τέλος, αποθήκευση RDF περιγραφών σε σχεσιακές βάσεις δεδομένων έχει υλοποιηθεί ή υλοποιείται στα συστήματα REDFOOT[16], RedLand[17] και Wraf [Wraf00]. Σε όλα τα παραπάνω συστήματα τα δεδομένα αποθηκεύονται με την μορφή τριάδων.

3.3 Συμπεράσματα

Από τα παραπάνω έγινε αντιληπτό ότι ένας διαχειριστής αντικειμένων έχει πολύ καλύτερες επιδόσεις στην απόκριση των επερωτήσεων σε σύγκριση με τα συστήματα σχεσιακών βάσεων δεδομένων. Όμως μια τέτοια λύση είναι άμεσα συνδεδεμένη με το συγκεκριμένο διαχειριστή αντικειμένων. Παράλληλα διαπιστώθηκε ότι η προσέγγιση *γνωρίσματος* σε σύγκριση με την προσέγγιση *ακμής* που έχει χρησιμοποιηθεί μέχρι τώρα για την αποθήκευση RDF γράφων έχει πολύ καλύτερους χρόνους απόκρισης στους περισσότερους τύπους επερωτήσεων. Η προσέγγιση STORED δεν θα μπορούσε να χρησιμοποιηθεί για την αποθήκευση RDF μεταδεδομένων εφόσον απαιτεί κανονικότητα στα δεδομένα κάτι που αντιβαίνει με το RDF. Το ίδιο ισχύει και για τις παραλλαγές Basic, Shared και Hybrid εφόσον θεωρούν ότι το σχεσιακό σχήμα είναι σταθερό και τα δεδομένα 'υπακούουν' σε σχήματα που είχαν ληφθεί υπόψη κατά την δημιουργία του σχεσιακού σχήματος. Βέβαια οι παραπάνω παραλλαγές όπως και η προσέγγιση STORED απαιτούν συνήθως λιγότερες συζεύξεις μεταξύ των πινάκων για την απόκριση των επερωτήσεων.

[16] http://www.oasis-open.org/cover/redfoot090README.txt
[17] Available at http://www.redland.opensource.ac.uk/

Κεφάλαιο 4

Αναπαράσταση και Αποθήκευση RDF μεταδεδομένων σε Σχεσιακά Συστήματα Διαχείρισης Βάσεων Δεδομένων

Σ' αυτό το κεφάλαιο αυτό θα περιγράψουμε το σύστημα που έχουμε υλοποιήσει για την αποθήκευση RDF μεταδεδομένων σε σχεσιακές βάσεις δεδομένων. Συγκεκριμένα, θα αναλύσουμε την σχεσιακή αναπαράσταση που προτείνουμε και θα περιγράψουμε την αρχιτεκτονική του συστήματος που έχουμε υλοποιήσει για την φόρτωση των RDF μεταδεδομένων. Επίσης θα παρουσιάσουμε συνοπτικά τον Συντακτικό Σημασιολογικό Αναλυτή RDF μεταδεδομένων (VRP) ο οποίος δέχεται σαν είσοδο τις RDF/XML περιγραφές τις αναλύει και τις αποθηκεύει στην κύρια μνήμη με βάση ένα εσωτερικό στο οποίο βασίζεται η διαδικασία φόρτωσης στην βάση δεδομένων.

4.1 RDF/XML Σύνταξη – Παραδείγματα

Σ' αυτήν την παράγραφο θα περιγράψουμε συνοπτικά την RDF/XML σύνταξη που ορίζεται στο κείμενο RDF M&S [LS99] για την αναπαράσταση και ανταλλαγή των RDF μεταδεδομένων. Υποθέτουμε ότι ο αναγνώστης γνωρίζει την XML [BPS98].

4.1.1 Γενικά χαρακτηριστικά RDF/XML σύνταξης

Για να γίνει κατανοητή η RDF/XML σύνταξη που θα περιγράψουμε στην συνέχεια θα κάνουμε αναφορές στα RDF μεταδεδομένα της εικόνας 4.1. Οι RDF περιγραφές ενός αρχείου περικλείονται στο στοιχείο *rdf:RDF* (γραμμή 2). Το στοιχείο *rdf:RDF* μπορεί να παραλείπεται αν η εφαρμογή 'γνωρίζει' ότι το αρχείο περιέχει RDF μεταδεδομένα. Μια RDF περιγραφή δηλώνεται με το στοιχείο *rdf:Description* (γραμμές

6 και 11). Συνήθως εφαρμόζεται σε ένα πόρο και περιέχει μια σειρά από XML στοιχεία που 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* (γραμμές 6 και 10) όταν θέλουμε να περιγράψουμε ένα ήδη ορισμένο πόρο.

Οι τιμές των ιδιοτήτων (δηλαδή των στοιχείων) που περιέχονται σε μια περιγραφή μπορεί να είναι είτε URIs δηλαδή πόροι (γραμμή 7), είτε literals (γραμμές 9, 12, 13) ή ακόμα και άλλες περιγραφές (γραμμές 11-14). Δηλαδή μια περιγραφή κάποιου πόρου μπορεί να περιέχει περιγραφές και άλλων πόρων.

Οι δηλώσεις των χώρων ονοματοδοσίας, των οποίων κλάσεις και ιδιότητες χρησιμοποιούνται στις RDF περιγραφές, τοποθετούνται συνήθως στην αρχή μέσα στο στοιχείο rdf:RDF (γραμμές 3-5).

Στην RDF/XML αναπαράσταση της εικόνας 2.1 περιγράφεται ο πόρος με URI *http://www.nga.gov/*. Η αναπαράσταση δηλώνει ότι ο πόρος ανήκει στις κλάσεις *http://www.odp.org/schema.rdf#ExternalPage* και http://www.arts.org/schema.rdf#»Art_History (γραμμές 7 και 8), έχει τίτλο (odp:title) *National Gallery of Art* (γραμμή 9) και δημιουργό (odp:creator) τον πόρο με URI *http://www.person.org/id123 (*γραμμές 10-15). Ο πόρος με URI http://www.person.org/id123 έχει όνομα (odp:first_name) John και επίθετο (odp:last_name) Miller (γραμμές 12-13).

```
1 <?xml version=»1.0»?>
2 <rdf:RDF
xmlns:rdf=»http://www.w3.org/1999/02/22-rdf-syntax-ns#»
xmlns:rdfs=»http://www.w3.org/2000/01/rdf-schema#»
xmlns:odp=»http://www.odp.org/schema.rdf#»>
<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/>
<odp:title>National Gallery of Art</odp:title>
<odp:creator>
<rdf:Description rdf:about=»http://www.person.org/id123»>
<odp:first_name>John<odp:first_name/>
<odp:last_name>Miller<odp:last_name/>
```

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

Εικόνα 4.1. RDF/XML αναπαράσταση.

```
<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>
```

Εικόνα 4.2. Συντομευμένη RDF/XML αναπαράσταση

Παράλληλα με την *αναλυτική* σύνταξη που έχουμε περιγράψει μέχρι τώρα στο κείμενο RDF M&S [LS99] παρουσιάζεται και μια *συντομευμένη* RDF/XML σύνταξη. Στην εικόνα 4.2 απεικονίζονται οι RDF/XML περιγραφές της εικόνας 4.1 χρησιμοποιώντας την συντομευμένη σύνταξη. Για λόγους χώρου παραλείπονται οι δηλώσεις των χώρων ονοματοδοσίας καθώς και το *rdf:RDF* στοιχείο. Δυο βασικά σημεία της σύνταξης αυτής είναι τα παρακάτω:

- Αν μια περιγραφή περιέχει (τουλάχιστον) μια *rdf:type* ιδιότητα τότε το στοιχείο *rdf:Description* μπορεί να αντικατασταθεί από το URI μιας από τις κλάσεις που αποτελούν τιμές της ιδιότητας *rdf:type*. Η αντικατάσταση αυτή είναι ισοδύναμη με την απόδοση της *rdf:type* ιδιότητας στον πόρο. Για παράδειγμα, συγκρίνοντας στην εικόνα 1 (γραμμές 6-7) και την εικόνα 2 (γραμμή 1) παρατηρούμε ότι το στοιχείο *rdf:Description* έχει αντικατασταθεί από το στοιχείο *odp:ExternalPage*.

- Όταν οι τιμές των ιδιοτήτων ενός πόρου είναι αλφαριθμητικά οι ιδιότητες αντί να αντιστοιχούνται σε XML στοιχεία μπορούν να δηλωθούν σαν γνωρίσματα του στοιχείου rdf:Description στο οποίο περιγράφεται ο πόρος. Στην εικόνα 2 στην γραμμή 5 παρατηρούμε πως μετατρέπεται η περιγραφή του πόρου http://www.person.org/id123 που περιέχεται στην εικόνα 1 στις γραμμές 11-14.

Στις RDF/XML περιγραφές μπορεί να περιέχεται *αναλυτική* και *συντομευμένη* σύνταξη ταυτόχρονα. Το γεγονός αυτό κάνει την διαδικασία της συντακτικής ανάλυσης

πιο πολύπλοκη. Επίσης παρατηρούμε ότι οι ετικέτες (XML tag) μπορεί να είναι είτε ετικέτες κόμβου (εικόνα 2 γραμμή 1) είτε ετικέτες ακμής. Η RDF/XML σύνταξη επιτρέπει στον συντακτικό αναλυτή να διαφοροποιήσει τις ετικέτες.

4.1.2 RDF/XML αναπαράσταση σχημάτων

Σε αυτήν την ενότητα θα αναπαραστήσουμε RDF σχήματα με την RDF/XML σύνταξη. Στην εικόνα 4.3 απεικονίζεται ένα RDF σχήμα στο οποίο ορίζεται μια ιεραρχία κλάσεων. Συγκεκριμένα ορίζονται οι κλάσεις *Arts* (γραμμή 5)*, Art_History* (γραμμή 6) και *Classical_Studies* (γραμμή 10). Να υπενθυμίσουμε ότι κάθε κλάση που ορίζουμε πρέπει να έχει την ιδιότητα *rdf:type* τιμή *rdfs#Class*. Οι κλάσεις *Art_History* και *Classical_Studies* ορίζονται σαν υποκλάσεις της *Arts* (γραμμές 7 και 11). Επίσης στην κλάση *Art_History* αποδίδεται η ιδιότητα *isDefinedBy* με τιμή τον πόρο με URI http://www.arts.org/schema.rdf (γραμμή 8). Παρατηρούμε ότι το στοιχείο *rdf:Description* αντικαθιστάται από το στοιχείο *rdfs#Class* (συντομευμένη σύνταξη) και ότι στις περιγραφές των κλάσεων χρησιμοποιείται το γνώρισμα *rdf:ID* εφόσον θεωρούμε ότι οι κλάσεις ορίζονται στο αρχείο που περιέχονται οι RDF περιγραφές. Επίσης παρατηρούμε ότι το URI ενός πόρου μπορεί να είναι σχετικό (π.χ. #Arts γραμμή 7).

```
1<?xml version=»1.0»?>
2<rdf:RDF
xmlns:rdf=»http://www.w3.org/1999/02/22-rdf-syntax-ns#»
xmlns:rdfs=»http://www.w3.org/2000/01/rdf-schema#»>
<rdfs:Class rdf:ID=»Arts»/>
<rdfs:Class rdf:ID=»Art_History»>
<rdfs:subClassOf rdf:resource=»#Arts»/>
<rdfs:isDefinedBy rdf:resource= «http://www.arts.org/schema.rdf»/>
</rdfs:Class>
<rdfs:Class rdf:ID=»Classical_Studies»>
<rdfs:subClassOf rdf:resource=»#Arts»/>
</rdfs:Class>
13</rdf:RDF>
```

Εικόνα 4.3. RDF σχήμα: Ιεραρχία κλάσεων.

Στην εικόνα 4.4 αναπαριστάνεται ένα δεύτερο RDF σχήμα στο οποίο ορίζονται κλάσεις και ιδιότητες. Κάθε ιδιότητα πρέπει να έχει την ιδιότητα rdf:type με τιμή rdf#Property.

```
<?xml version=»1.0»?>
<rdf:RDF
      xmlns:rdf=»http://www.w3.org/1999/02/22-rdf-syntax-ns#»
      xmlns:rdfs=»http://www.w3.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.w3.org/2000/01/rdf-schema#Literal»/>
     </rdf:Property>
<rdf:Property  rdf:ID=»creator»>
<rdfs:subPropertyOfrdf: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.w3.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.w3.org/2000/01/rdf-schema#Literal»/>
    </rdf:Property>
</rdf:RDF>
```

Εικόνα 4.4. RDF σχήμα για την περιγραφή σελίδων

4.1.3 RDF/XML αναπαράσταση υποστασιοποιημένων δηλώσεων και συλλογών

Στην ενότητα αυτή θα περιγράψουμε την RDF/XML αναπαράσταση υποστασιοποιημένων δηλώσεων και συλλογών. Η RDF/XML αναπαράσταση της υποστασιοποιημένης δήλωσης *"Ο τίτλος (odp#title) της σελίδας http://www.nga.gov/ είναι National Gallery of Art"* παρατίθεται στην συνέχεια. Παρατηρούμε ότι ο πόρος που αναπαριστά το μοντέλο της δήλωσης είναι ανώνυμος (anonymous resource). Για να αποδώσουμε ένα URI στον πόρο, αυτό απαιτείται στη περίπτωση που θέλουμε να τον

χρησιμοποιήσουμε και σε άλλη περιγραφή, θα πρέπει στο στοιχείο *rdf:Description* να προσθέσουμε το γνώρισμα *rdf:ID* με τιμή το όνομα του πόρου π.χ. *rdf:ID=»stat1»*. Στην παρακάτω αναπαράσταση εκτός από τις 4 βασικές ιδιότητες που πρέπει να έχει μια υποστασιοποιημένη δήλωση έχουμε προσθέσει και την 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.w3.org/1999/02/22-rdf-syntax-ns#Statement»
/>
    <s:said>Sofia</s:said>
 </rdf:Description>
```

Εικόνα 4.5. RDF/XML αναπαράσταση υποστασιοποιημένων δηλώσεων.

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 ότι *"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>
```

Για την αναπαράσταση των συλλογών χρησιμοποιούνται, αντί του στοιχείου *rdf:Description*, τα στοιχεία *rdf:Bag*, *rdf:Seq* και *rdf:Alt*. Τα παραπάνω στοιχεία είναι δυνατόν είτε να έχουν το γνώρισμα *rdf:ID*, είτε να μην προσδιορίζουν το URI του πόρου που περιγράφουν. Δεν μπορούν να περιέχουν το γνώρισμα *rdf:about*. Οι ιδιότητες που δηλώνουν τα μέρη της συλλογής αναπαριστάνονται είτε με τα ονόματα τους rdf:_1, rdf:_2… είτε από το στοιχείο rdf:li. Για παράδειγμα η RDF/XML αναπαράσταση που αντιστοιχεί στην παραλλαγή της εικόνας 2.4 είναι η παρακάτω:

```
    <rdf:Alt>
      <rdf:li>RDF Metadata for Community Webs</rdf:li>
      <rdf:li>RDF μεταδεδομένα για Κοινότητες του Διαδικτύου</rdf:li>
```

```
</rdf:Alt>
```

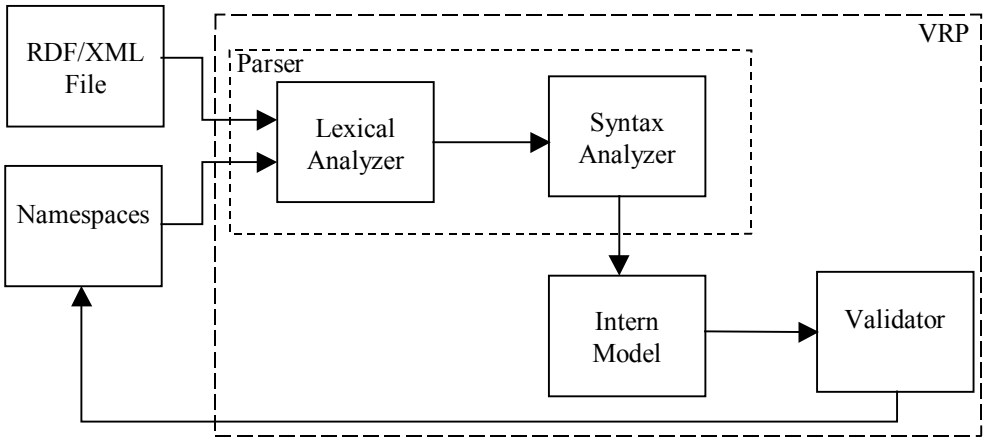
Εικόνα 4.6. RDF/XML αναπαράσταση συλλογών.

4.2 Συντακτικός Σημασιολογικός Αναλυτής RDF μεταδεδομένων (VRP)

Ο Συντακτικός Σημασιολογικός Αναλυτής RDF μεταδεδομένων (VRP) είναι ένα εργαλείο κατάλληλο για την ανάλυση (parsing) RDF/XML αρχείων και τον έλεγχο της συνέπειας τους. Σε αντίθεση με άλλους RDF συντακτικούς αναλυτές π.χ. SiRPAC [S00], ο VRP βασίζεται στα εργαλεία CUP [S96] και Jflex [Jflex] (όμοια με τα εργαλεία YACC/LEX) τα οποία χρησιμοποιούνται για την δημιουργία συντακτικών αναλυτών σε Java. Έτσι μπορεί να λειτουργήσει αυτόνομα. Δεν απαιτούνται για την λειτουργία του άλλα προγράμματα π.χ., XML συντακτικοί αναλυτές. Παράλληλα μπορούν εύκολα να γίνουν τροποποιήσεις σε περίπτωση που γίνουν αλλαγές στην RDF/XML σύνταξη.

4.2.1 Αρχιτεκτονική VRP

Στην εικόνα 4.7 απεικονίζεται η αρχιτεκτονική του VRP. Ο VRP δέχεται σαν είσοδο είτε RDF/XML αρχεία είτε html ή XML αρχεία τα οποία περιέχουν RDF/XML περιγραφές. Τα RDF μεταδεδομένα που αναλύει μπορεί να βασίζονται σε πολλαπλά σχήματα. Ο VRP αποτελείται από δύο βασικά μέρη τον συντακτικό αναλυτή (parser) και τον σημασιολογικό αναλυτή. Ο συντακτικός αναλυτής ελέγχει αν το αρχείο που αναλύει ‘υπακούει’ στην RDF/XML σύνταξη που ορίζεται στο RDF M&S. Η έξοδος του συντακτικού αναλυτή είναι το σύνολο τριάδων *(κατηγόρημα, θέμα, αντικείμενο)* του προς ανάλυση αρχείου. Οι τριάδες που εξάγονται από τον συντακτικό αναλυτή οργανώνονται με βάση το εσωτερικό μοντέλο του VRP (βλέπε υπο-ενότητα 4.2.2) και αποθηκεύονται προσωρινά στην κύρια μνήμη. Ο σημασιολογικός αναλυτής ελέγχει την συνέπεια του RDF/XML αρχείου.



Εικόνα 4.7. Αρχιτεκτονική Συντακτικού Σημασιολογικού Αναλυτή RDF μεταδεδομένων.

4.2.2 Εσωτερικό Μοντέλο VRP

Το εσωτερικό μοντέλο του VRP επιτρέπει την οργάνωση των RDF περιγραφών και το διαχωρισμό των μεταδεδομένων από το σχήμα. Αποτελείται από μια ιεραρχία Java κλάσεων (εικόνα 4.8). Συγκεκριμένα, αποτελείται από τις κλάσεις *Resource*, *RDF_Resource*, *RDF_Class*, *RDF_Property*, *RDF_Container* και *RDF_Statement*. Για κάθε πόρο που εμφανίζεται στις RDF περιγραφές που επεξεργαζόμαστε δημιουργείται ένα ακριβώς Java αντικείμενο το οποίο ανήκει σε μια από τις παραπάνω κλάσεις. Στην συνέχεια θα παρουσιάσουμε τις κλάσεις του εσωτερικού μοντέλου του VRP δίνοντας και παραδείγματα αντικειμένων που ανήκουν σ' αυτές (εικόνα 4.9). Τα αντικείμενα δημιουργούνται με βάση τις RDF/XML περιγραφές που απεικονίζονται στην εικόνα 4.1.

Resource: Η κλάση Resource βρίσκεται στην κορυφή της ιεραρχίας. Άμεσα[18] μέλη της κλάσης αυτής είναι τα αντικείμενα που αντιστοιχούν σε πόρους στους οποίους δεν έχουν αποδοθεί οι προκαθορισμένες RDF/S ιδιότητες π.χ, rdf:type, rdfs:seeAlso, rdfs:comment. Δηλαδή τα άμεσα μέλη της κλάσης είναι κυρίως αντικείμενα που αντιπροσωπεύουν πόρους που δεν τους έχει αποδοθεί η rdf:type ιδιότητα, άρα δεν ανήκουν σε κάποια κλάση. Το αντικείμενο που αντιστοιχεί στον πόρο με URI http://www.odp.org/schema.rdf ανήκει στην κλάση Resource. Το αντικείμενο με τα γνωρίσματά του απεικονίζεται στην εικόνα 4.9.

[18] Άμεσα μέλη μιας κλάσης θεωρούμε τα αντικείμενα που δηλώνονται σαν περιπτώσεις της κλάσης. Έμμεσα μέλη μιας κλάσης είναι τα μέλη των υποκλάσεων της.

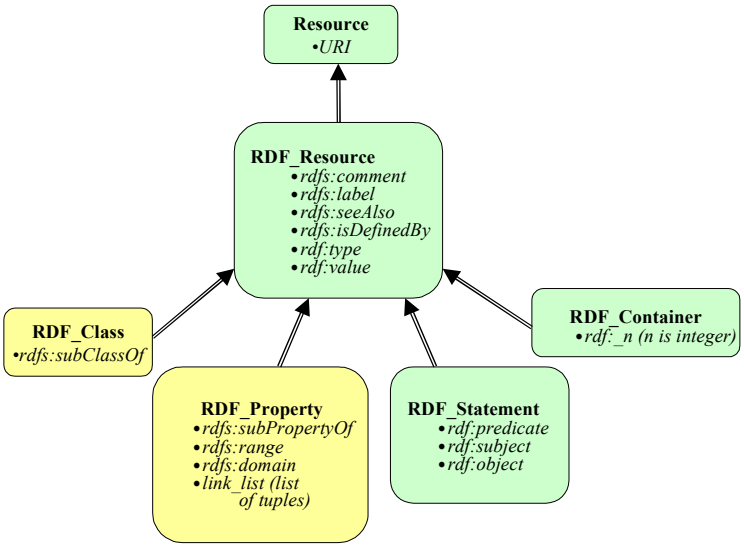
RDF_Resource: Περιέχει τα αντικείμενα που αντιστοιχούν σε πόρους στους οποίους έχει αποδοθεί κάποια από τις προκαθορισμένες RDF/S ιδιότητες. Για παράδειγμα, το αντικείμενο που αντιστοιχεί στον πόρο με URI http://www.nga.gov/ ανήκει στην κλάση RDF_Resource (εικόνα 4.9)

RDF_Class: Περιέχει τα αντικείμενα που αντιστοιχούν σε RDF κλάσεις. Το αντικείμενο που αντιστοιχεί στον πόρο art#Art_History ανήκει στην κλάση RDF_Class (εικόνα 4.9).

RDF_Property: Περιέχει τα αντικείμενα που αντιστοιχούν σε RDF ιδιότητες. Το αντικείμενο που αντιστοιχεί στον πόρο με URI odp#title ανήκει στην κλάση RDF_Property (εικόνα 4.9). Οι περιπτώσεις μιας RDF ιδιότητας δηλαδή όλα τα βέλη που σε ένα RDF γράφο έχουν το όνομα της ιδιότητας, αποτελούν τιμές του γνωρίσματος link_list και αποθηκεύονται σαν ζευγάρια (κόμβος αρχής, κόμβος προορισμού).

RDF_Container: Περιέχει τα αντικείμενα που ανήκουν κάποιο από τα τρία είδη συλλογών. Για παράδειγμα το αντικείμενο που αντιστοιχεί στην συλλογή της εικόνας 4.6 ανήκει στην κλάση RDF_Container (εικόνα 4.9).

RDF_Statement: Περιέχει τα αντικείμενα που αντιπροσωπεύουν υποστασιοποιημένες δηλώσεις.



Εικόνα 4.8. Ιεραρχία κλάσεων του VRP μοντέλου.

Resource@3456	
URI	http://www.odp.org/schema.rdf

RDF_Resource@4487	
URI	http://www.nga.gov/
rdf:type	odp#ExternalPage, art#Art_History

RDF_Container@4455	
URI	ns1#genID1
rdf:type	rdf#Alt
rdf:_n	RDF Metadata for… RDF μεταδεδομένα για…

RDF_Class@4455	
URI	art#Art_History
rdf:type	rdfs#Class
rdfs:subClassOf	art#Arts
rdfs:isDefinedBy	http://www.odp.org/schema.rdf

RDF_Property@4567	
URI	odp#title
rdf:type	rdf#Property
rdfs:subPropertyOf	dc#title
rdfs:domain	odp#ExternalPage
rdfs:range	rdfs#Literal
link_list	http://www.nga.gov/, National Gallery of Art ……

Εικόνα 4.9. Στιγμιότυπα των κλάσεων του VRP μοντέλου.

Τα γνωρίσματα *URI, rdf:predicate, rdf:subject* και *rdf:object* πρέπει να έχουν ακριβώς μια τιμή. Το γνώρισμα *rdfs:range* μπορεί να έχει το πολύ μια τιμή. Τα υπόλοιπα γνωρίσματα των κλάσεων του μοντέλου του VRP είναι πλειότιμα. Οι περιορισμοί που προσθέτουμε απαιτούν τόσο το γνώρισμα *rdfs:range* όσο και το γνώρισμα *rdfs:domain* να έχουν μια ακριβώς τιμή.

URI	Object reference
art#Art_History	RDF_Class@4455
odp#title	RDF_Property@5678
http://www.nga.gov/	RDF_Resource@448
http://www.arts.org/schema.rdf	Resource@3456

Πίνακας 4.1. VRP Hashmap.

Να τονίσουμε εδώ ότι οι τιμές των γνωρισμάτων είναι αλφαριθμητικά και όχι αναφορές αντικειμένων. Για παράδειγμα η τιμή του γνωρίσματος rdfs:domain για το αντικείμενο που αντιστοιχεί στην ιδιότητα με URI odp#title είναι το αλφαριθμητικό odp#ExternalPage, και όχι η αναφορά του αντικειμένου με URI ns2#ExternalPage. Δηλαδή τα αντικείμενα συνδέονται μέσω αναφορών τιμών (value references) χρησιμοποιώντας τα URIs των αντικειμένων. Η αντιστοίχηση μεταξύ των URIs των αντικειμένων και των αναφορών τους επιτυγχάνεται μέσω ενός Hashmap. Στον πίνακα

4.1 παρουσιάζεται ένα μέρος του Hashmap που σχηματίζεται για τις RDF περιγραφές της εικόνας 4.1. Ο λόγος για τον οποίο δεν χρησιμοποιήθηκαν αναφορές αντικειμένων είναι ότι δεν γνωρίζουμε εκ' των προτέρων την πιο στενή κλάση στην οποία ανήκει ένα αντικείμενο, καθώς οι περιγραφές για τους πόρους μπορεί να βρίσκονται διασκορπισμένες στο αρχείο. Νέα πληροφορία για το αντικείμενο μπορεί να το ταξινομήσει σε κάποια κλάση στην ιεραρχία του VRP μοντέλου πιο κάτω από την τωρινή. Συνεπώς μια νέα αναφορά αντικειμένου πρέπει να δημιουργηθεί και άρα θα πρέπει να ενημερωθούν όλες οι αναφορές αντικειμένων που δείχνουν σ' αυτό το αντικείμενο.

Στην συνέχεια θα παραθέσουμε ένα παράδειγμα που εξηγεί πως από τις τριάδες που παράγονται από τον συντακτικό αναλυτή σχηματίζεται το εσωτερικό VRP μοντέλο. Έστω ότι ο συντακτικός αναλυτής δίνει την *τριάδα (rdf#type, http://www.nga.gov/, odp#ExternalPage)*. Σε περίπτωση που δεν έχουν ήδη δημιουργηθεί αντικείμενα με URIs http://www.nga.gov/ και odp#ExternalPage δημιουργεί τα 2 παρακάτω αντικείμενα. Το αντικείμενο με URI http://www.nga.gov/ το οποίο ανήκει στην κλάση RDF_Resource εφόσον έχει την ιδιότητα *rdf:type*. Το αντικείμενο με URI odp#ExternalPage το οποίο ανήκει στην κλάση RDF_Class δεδομένου ότι ο πόρος odp#ExternalPage πρέπει να είναι κλάση για να μπορεί να έχει μέλη άλλους πόρους. Επόμενες τριάδες μπορεί να προσθέσουν πληροφορία σ' αυτά τα αντικείμενα.

RDF_Resource@4487	
URI	http://www.nga.gov/
rdf:type	odp#ExternalPage

RDF_Class@4455	
URI	art#Art_History

Το μοντέλο του VRP επιτρέπει την οργάνωση των RDF περιγραφών και το διαχωρισμό των μεταδεδομένων από το σχήμα, σε αντίθεση με άλλους RDF συντακτικούς αναλυτές οι οποίοι απλώς δημιουργούν ένα σύνολο από τριάδες. Η ιεραρχική οργάνωση των μεταδεδομένων καθιστά σαφή την σημασιολογία τους. Έτσι διευκολύνεται τόσο ο έλεγχος της συνέπειας των RDF περιγραφών από τον VRP όσο και η επεξεργασία των RDF περιγραφών από άλλες εφαρμογές οι οποίες συνεργάζονται με το VRP, όπως εργαλεία για αποθήκευση RDF μεταδεδομένων σε συστήματα βάσεων δεδομένων.

4.2.3 Σημασιολογικός έλεγχος

Το βασικό χαρακτηριστικό του VRP είναι η δυνατότητα που παρέχει για τον έλεγχο της εγκυρότητας ενός RDF/XML αρχείου. Το σύνολο των σημασιολογικών περιορισμών που ελέγχονται από τον VRP αντιστοιχεί στους βασικούς RDF/S περιορισμούς για σχήμα και δεδομένα που απεικονίζεται στην εικόνα 2.18. Οι σημασιολογικοί έλεγχοι που εκτελούνται από τον VRP βασίζονται στο εσωτερικό μοντέλο που έχει δημιουργηθεί.

Στην συνέχεια αναλύεται η διαδικασία που ακολουθεί ο VRP για ελέγξει την εγκυρότητα ενός RDF/XML αρχείου. Αρχικά αναλύει το RDF/XML αρχείο και δημιουργεί το VRP μοντέλο που αντιστοιχεί στις RDF περιγραφές που αναλύονται. Στην συνέχεια συνδέεται στους χώρους ονοματοδοσίας όπου ορίζονται τα σχήματα που χρησιμοποιούνται στις περιγραφές και τα αναλύει. Από τις παραπάνω περιγραφές στο μοντέλο που έχει δημιουργηθεί από την ανάλυση του αρχικού αρχείου προσθέτει μόνο την παρακάτω πληροφορία:

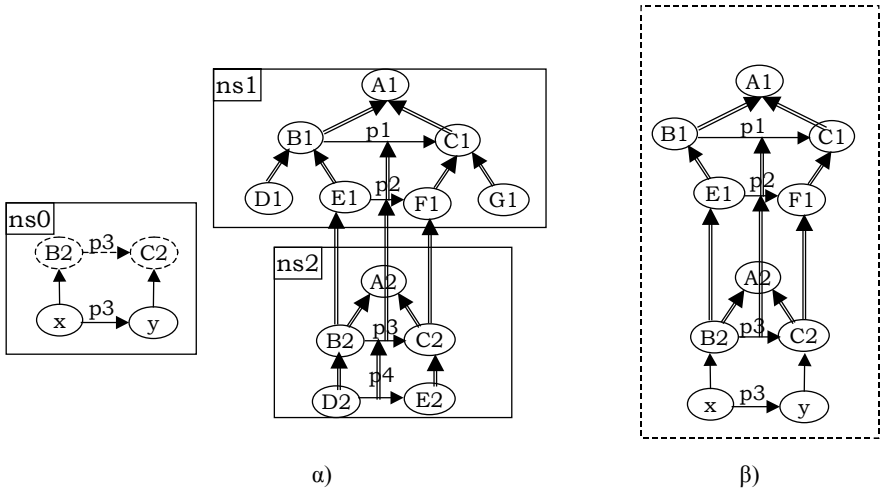
Πληροφορία για τα αντικείμενα που αντιστοιχούν σε κλάσεις που χρησιμοποιούνται στα αρχικά μεταδεδομένα (π.χ. υποκλάσεις) καθώς και τα αντικείμενα που αντιστοιχούν στις κλάσεις της ιεραρχίας κλάσεων πάνω από αυτές.

Πληροφορία για τα αντικείμενα που αντιστοιχούν σε ιδιότητες που χρησιμοποιούνται στα αρχικά μεταδεδομένα (π.χ. πεδίο ορισμού τους) καθώς και τα αντικείμενα που αντιστοιχούν σε ιδιότητες της ιεραρχίας ιδιοτήτων πάνω από αυτές. Παράλληλα προσθέτει τα RDF_Class αντικείμενα που αντιστοιχούν στα πεδία ορισμού και τιμών των ιδιοτήτων και τα αντικείμενα που αντιστοιχούν στις κλάσεις της ιεραρχίας κλάσεων πάνω από αυτές.

Να τονίσουμε ότι αν κάποιο από τα παραπάνω αντικείμενα βρίσκεται σε διαφορετικό χώρο ονοματοδοσίας π.χ. όταν η υπερκλάση μια κλάσης ορίζεται σε διαφορετικό χώρο ονοματοδοσίας από την κλάση, ο VRP θα αναλύσει και το σχήμα που ορίζεται η υπερκλάση και θα εισάγει πληροφορία στο VRP μοντέλο με αντίστοιχη λογική με προηγουμένως.

Έστω ότι ο VRP αναλύει τα RDF μεταδεδομένα του σχήματος ns0 της εικόνας 4.10α. Στα μεταδεδομένα χρησιμοποιούνται οι κλάσεις B2 και C2 και η ιδιότητα p3 που ορίζονται στο σχήμα ns2. Το αρχικό μοντέλο του VRP αποτελείται από τα αντικείμενα που αντιστοιχούν στους πόρους x, y, στις κλάσεις B2 και C2 και στην ιδιότητα p3. Στην συνέχεια ο VRP προσθέτει και τα αντικείμενα που αντιστοιχούν στις κλάσεις E1, B1, A1,

A2 οι οποίες βρίσκονται πάνω από την B2 στην ιεραρχία κλάσεων καθώς επίσης και τις κλάσεις F1 και C1 που βρίσκονται πάνω από την B2 στην ιεραρχία κλάσεων. Επίσης προσθέτει και τις ιδιότητες p2 και p1 που βρίσκονται πάνω από την p3 στην ιεραρχία ιδιοτήτων. Στην εικόνα 4.10β απεικονίζεται το σύνολο των RDF περιγραφών που θα ελέγξει ο VRP αν πληρούν τις περιγραφές για να αποφανθεί για την συνέπεια του αρχείου ns0. Το αρχείο θα είναι συνεπές μόνο αν το συνολικό εσωτερικό VRP μοντέλο πληροί όλους τους περιορισμούς. Δηλαδή σύμφωνα με τον VRP ένα αρχείο συνεπές όταν τόσο τα μεταδεδομένα όσο και η ένωση των μερών των σχημάτων που σχετίζονται μ' αυτά είναι συνεπής.



α) β)

Εικόνα 4.10. Σύνολο RDF περιγραφών για έλεγχο συνέπειας.

4.3 Λογικό Μοντέλο Σχεσιακής Αναπαράστασης RDF μεταδεδομένων

Σε αυτήν την ενότητα θα περιγράψουμε ένα γενικό μοντέλο αναπαράστασης RDF μεταδεδομένων σε οντοκεντρικές σχεσιακές βάσεις δεδομένων. Στην υπο-ενότητα 4.3.5 προτείνουμε κάποιες παραλλαγές στο γενικό αυτό μοντέλο με στόχο την αποτελεσματικότερη αποθήκευση στηριζόμενοι σε επιπλέον υποθέσεις για τα σχήματα που δεν προσφέρονται από το βασικό RDF/S μοντέλο. Να σημειώσουμε ότι επιλέξαμε ένα σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων για την αποθήκευση των RDF δεδομένων αντί ένα διαχειριστή αντικειμένων που όπως είδαμε έχει καλύτερες επιδόσεις ώστε το σύστημα μας να μπορεί να τρέχει σε πολλές πλατφόρμες λογισμικού και υλικού (portable).

Η σχεσιακή αναπαράσταση που προτείνουμε υποστηρίζει την ελευθερία περιγραφών που παρέχει το RDF με εξαίρεση τους περιορισμούς που έχουμε προσθέσει για τα είδη των επεκτάσεων που επιτρέπουμε μεταξύ σχημάτων (βλέπε εικόνα 2.18). Καταρχήν, υποστηρίζουμε την αποθήκευση μεταδεδομένων που βασίζονται σε πολλαπλά σχήματα. Η προσθήκη μιας κλάσης στην βάση δεν απαιτεί τον προσδιορισμό των ιδιοτήτων που μπορούν να αποδοθούν σ' αυτήν. Μια ιδιότητα μπορεί να προστεθεί οποιαδήποτε στιγμή στη βάση και να έχει σαν πεδίο ορισμού και τιμών οποιαδήποτε κλάση. Επίσης υποστηρίζουμε αποτελεσματική αποθήκευση και επερώτηση των σχέσεων υποκλάσης και υποιδιότητας. Επιπλέον οι ιδιότητες που μπορούν να αποδοθούν σε ένα πόρο δεν καθορίζονται κατά την διάρκεια εισαγωγής του στην βάση δεδομένων ούτε δεσμεύεται χώρος για την αποθήκευση τους. Αντίθετα ο αριθμός των ιδιοτήτων μπορεί να μεταβάλλεται συνεχώς καθώς νέα σχήματα, συγκεκριμένα ιδιότητες, προσθέτονται στην βάση. Ανά πάσα στιγμή μπορεί να αποδοθεί σε ένα πόρο μια ιδιότητα. Τέλος, χειριζόμαστε αποτελεσματικά το γεγονός ότι οι ιδιότητες μπορεί να είναι προαιρετικές, μοναδικές ή πλειότιμες.

Στην αναπαράσταση που προτείνουμε οι RDF κλάσεις και ιδιότητες αποθηκεύονται σαν σχήμα της βάσης. Συνεπώς το σχήμα της βάσης μεταβάλλεται καθώς νέα RDF σχήματα αποθηκεύονται στην βάση. Οι RDF περιγραφές για δεδομένα εμπλουτίζουν το παραπάνω σχήμα. Η προσέγγιση μας μοιάζει με την προσέγγιση *γνωρίσματος* (ενότητα 3.1.2.2). Εμείς όμως λαμβάνουμε υπόψη την ιδιαιτερότητα του RDF γράφου όπου οι ετικέτες μπορεί να είναι είτε ετικέτες κόμβου είτε ετικέτες ακμής.

Το αρχικό σχήμα της βάσης δεδομένων αποτελείται από τους πίνακες *Class, Property, SubClass, SubProperty, Namespace, Type, Bag, Seq, Alt και Statement.* Στην συνέχεια θα αναλύσουμε τους παραπάνω πίνακες και θα περιγράψουμε αναλυτικά πως αποθηκεύονται στην βάση τα RDF σχήματα και μεταδεδομένα.

4.3.1 Αναπαράσταση RDF σχημάτων

Οι πίνακες *Class* και *Property* αποθηκεύουν πληροφορία για τις RDF κλάσεις και ιδιότητες αντίστοιχα. Οι πίνακες *SubClass* και *SubProperty* αποθηκεύουν πληροφορία για τις ιεραρχίες κλάσεων και ιδιοτήτων. Οι παραπάνω τέσσερις πίνακες ουσιαστικά αποτελούν το μετα-μοντέλο της βάσης δεδομένων.

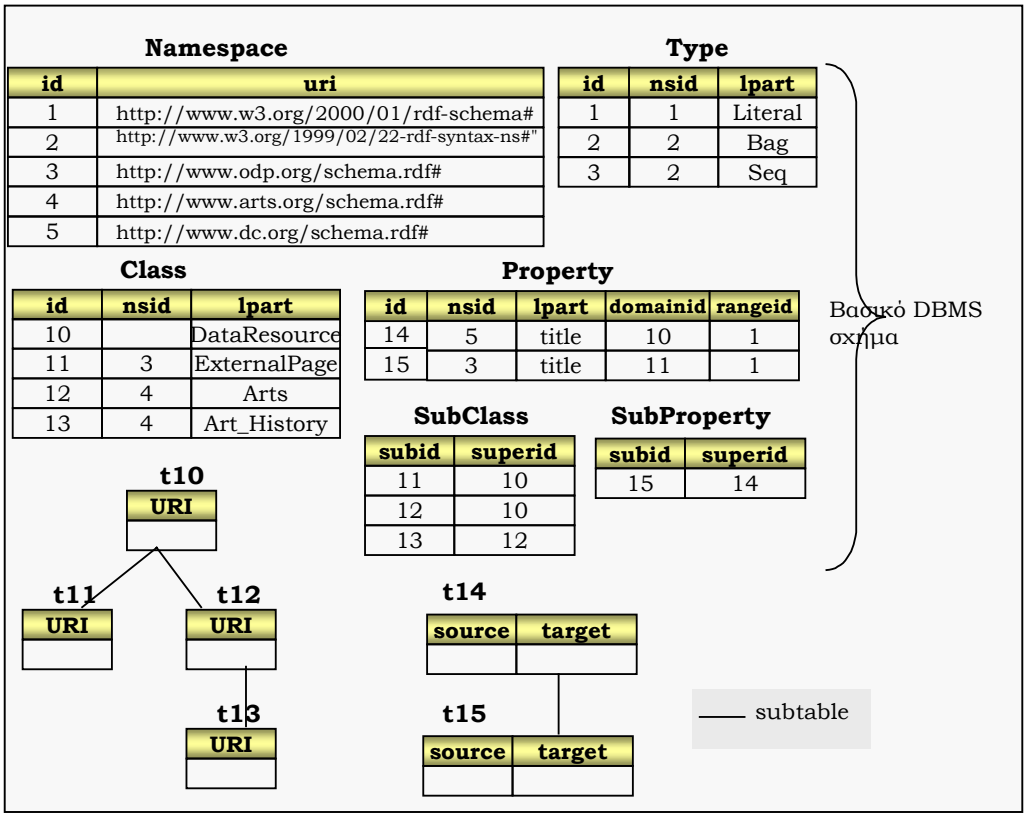
Ο πίνακας *Namespace* (εικόνα 4.11) περιέχει πληροφορία για τους χώρους ονοματοδοσίας στους οποίους ορίζονται σχήμα κατασκευές που έχουν αποθηκευτεί στην βάση δεδομένων. Αποτελείται από δύο πεδία. Το πεδίο *uri* έχει σαν τιμή το URI του

χώρου ονοματοδοσίας, ενώ το πεδίο *id* έχει σαν τιμή το ακέραιο αναγνωριστικό που του αποδίδεται. Ο βασικός λόγος που εισαγάγαμε τον πίνακα Namespace είναι για να μειώσουμε το χώρο αποθήκευσης που απαιτείται τόσο για τις κλάσεις όσο και για τις ιδιότητες. Αυτό φαίνεται καθαρά παρακάτω που αναλύονται οι πίνακες Class και Property όπου αντί να καταχωρείται ολόκληρο το URI της κλάσης ή ιδιότητας αντίστοιχα καταχωρείται το αναγνωριστικό του χώρου ονοματοδοσίας και το τοπικό όνομα. Η οικονομία χώρου που επιτυγχάνουμε με την παραπάνω αναπαράσταση είναι σημαντική δεδομένου του μεγάλου όγκου κλάσεων και ιδιοτήτων που αναμένεται να αποθηκεύονται στην βάση δεδομένων εξαιτίας των πολλαπλών και πιθανόν σύνθετων RDF σχημάτων (π.χ. σχήματα που προκύπτουν από ολοκλήρωση οντολογιών και θησαυρών όρων).

Ένα ιδιαίτερο χαρακτηριστικό της αναπαράστασης που προτείνουμε είναι ότι αποδίδουμε κωδικούς στις κλάσεις και στις ιδιότητες. Ο βασικός λόγος που τους εισάγουμε είναι ότι αποδίδοντας στις κλάσεις και στις ιδιότητες κωδικούς οι οποίοι βασίζονται στην θέση τους στην ιεραρχία κλάσεων και ιδιοτήτων αντίστοιχα μπορούμε στην συνέχεια να διατρέξουμε και να επερωτήσουμε τις ιεραρχίες πολύ αποδοτικά. Στην μεταπτυχιακή εργασία [L00a] παρουσιάζονται διάφοροι τρόποι κωδικοποίησης ιεραρχιών που θα μπορούσαμε να εφαρμόσουμε. Να σημειώσουμε όμως ότι μπορούν να εφαρμοστούν μόνο για ιεραρχίες μονής κληρονομικότητας. Προς το παρόν όπως φαίνεται και στην εικόνα 4.11 οι κωδικοί που δίνουμε είναι απλώς αυξητικοί.

Ο πίνακας *Class* περιέχει το URI και τον κωδικό των κλάσεων που έχουν καταχωρηθεί στην βάση (εικόνα 4.11). Συγκεκριμένα έχει τρία πεδία, το πεδίο *id* έχει σαν τιμή τον κωδικό που αποδίδεται στην κλάση, το πεδίο *nsid* έχει σαν τιμή το αναγνωριστικό του χώρου ονοματοδοσίας που ορίζεται η κλάση και το πεδίο *lpart* έχει σαν τιμή το τοπικό όνομα της κλάσης.

Ο πίνακας *Property* περιέχει το αναγνωριστικό, το URI, το πεδίο ορισμού και το πεδίο τιμών των ιδιοτήτων που έχουν καταχωρηθεί στην βάση (εικόνα 4.11). Τα τρία πρώτα πεδία είναι αντίστοιχα μ' αυτά 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 το κωδικό της κλάσης που αποτελεί το πεδίο τιμών.



Εικόνα 4.11. Σχεσιακή αναπαράσταση RDF σχημάτων.

Οι πίνακες *SubClass* και *SubProperty* αναπαριστάνουν τις ιεραρχίες κλάσεων και ιδιοτήτων αντίστοιχα. Ο πίνακας *SubClass* έχει 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 και για τον πίνακα *SubProperty*. Σε περίπτωση που μια κλάση δεν έχει υπερκλάσεις την θεωρούμε υποκλάση της κλάσης DataResource και άρα καταχωρείται στον πίνακα SubClass μια εγγραφή με τον κωδικό της κλάσης και τον κωδικό της κλάσης DataResource.

Ο πίνακας *Type* περιέχει τους κωδικούς που αποδίδονται στους βασικούς τύπους/κλάσεις της RDF/S, όπως rdfs#Literal, rdf#Bag.

Στην εικόνα 4.11 παρατηρούμε ότι οι κωδικοί για τους χώρους ονοματοδοσίας μπορεί να συμπίπτουν με τους κωδικούς που αποδίδονται στις κλάσεις και ιδιότητες δεδομένου ότι έχουν διαφορετική χρησιμότητα και δεν πρόκειται στο πεδίο ενός πίνακα να υπάρχουν κωδικοί και από τις δύο κατηγορίες.

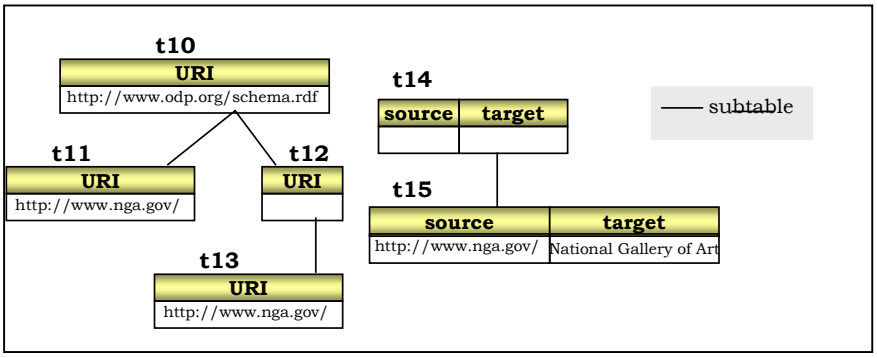
Το αρχικό σχήμα που παρουσιάσαμε εμπλουτίζεται καθώς προσθέτονται RDF σχήματα στην βάση. Συγκεκριμένα, για κάθε νέα κλάση ή ιδιότητα που αποθηκεύεται στην βάση δημιουργείται ένας νέος πίνακας. Οι πίνακες που αντιστοιχούν στις κλάσεις (πίνακες-κλάσεων) έχουν μόνο ένα πεδίο, το *URI* ενώ οι πίνακες που αντιστοιχούν σε ιδιότητες (πίνακες-ιδιοτήτων) έχουν δύο πεδία, το *source* και *target*. Οι πίνακες που δημιουργούνται για μια κλάση ή ιδιότητα δηλώνονται σαν υποπίνακες των πινάκων που αντιστοιχούν σε υπερκλάσεις τους ή σε υποιδιότητες αντίστοιχα. Κορυφή της ιεραρχίας των πινάκων που αντιστοιχούν σε κλάσεις είναι ο πίνακας που αντιστοιχεί στην κλάση *DataResource*.

Το όνομα ενός πίνακα προκύπτει από το αναγνωριστικό της κλάσης/ιδιότητας. Για παράδειγμα το όνομα του πίνακα που αντιστοιχεί στην κλάση με αναγνωριστικό 11 είναι *t11*. Τα ονόματα που προκύπτουν είναι ευέλικτα λόγω του μικρού μήκους τους. Στα περισσότερα σχεσιακά συστήματα βάσεων δεδομένων τα ονόματα των πινάκων πρέπει να αρχίζουν με κάποιο γράμμα. Γι΄ αυτό το λόγο τα ονόματα των πινάκων έχουν την μορφή **t**(able) + ***αναγνωριστικό*** κλάσης/ιδιότητας.

Τα ευρετήρια που έχουμε κατασκευάσει στους παραπάνω πίνακες θα περιγραφούν στην υπο-ενότητα 4.5.1.

4.3.2 Αναπαράσταση RDF περιγραφών πληροφοριακών πόρων

Οι RDF περιγραφές για δεδομένα αποθηκεύονται στους πίνακες που αντιστοιχούν στις ιδιότητες και κλάσεις (εικόνα 4.12). Δηλώσεις που αναφέρονται στην κλάση που ανήκει ένας πόρος, δηλαδή έχουν την ιδιότητα rdf:type, έχουν σαν αποτέλεσμα την καταχώρηση μιας εγγραφής με το URI του πόρου στον πίνακα που αντιστοιχεί στην κλάση. Για τις δηλώσεις με οποιαδήποτε άλλη ιδιότητα καταχωρείται μια εγγραφή στον πίνακα που αντιστοιχεί στην ιδιότητα. Στο πεδίο *source* καταχωρείται το θέμα (subject) της δήλωσης ενώ στο πεδίο target το αντικείμενο (object) της δήλωσης.



Εικόνα 4.12. Αναπαράσταση RDF περιγραφών για δεδομένα.

4.3.3 Αναπαράσταση RDF συλλογών

Η αναπαράσταση των RDF συλλογών παρουσιάζει ιδιαιτερότητα γι' αυτό και αναλύεται χωριστά. Οι ιδιότητες rdf:_1, rdf_2… που δηλώνουν τα μέλη μιας συλλογής είναι μη αριθμήσιμες όποτε δεν μπορούμε να τις χειριστούμε όπως τις υπόλοιπες RDF ιδιότητες δηλαδή δεν μπορούμε να δημιουργήσουμε για κάθε ιδιότητα ένα πίνακα. Να σημειώσουμε επίσης ότι οι RDF συλλογές είναι δυνατόν να περιέχουν μόνο πόρους και literals. Για παράδειγμα δεν μπορεί να οριστεί μια συλλογή από ακέραιους ή ημερομηνίες.

Bag

bagid	member	URI	string	int	date
bag1	1			2	
bag1	2				26/08/78
bag1	3			5	
bag2	1	http://www.nga.gov/			
bag2	2	http://www.w3c.org/			

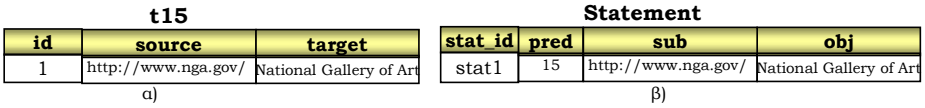
Εικόνα 4.13. Σχεσιακή αναπαράσταση RDF συλλογών.

Η αναπαράσταση που προτείνουμε στην συνέχεια για τις RDF συλλογές επιτρέπει την αποθήκευση συλλογών είτε με ετερογενή είτε ομογενή μέλη και επεκτείνει (εξειδικεύει) τους τύπους που μπορεί να έχουν τα μέλη μιας συλλογής. Συγκεκριμένα τα μέλη μιας συλλογής μπορεί να είναι URIs (resources), αλφαριθμητικά, ακέραιοι, ημερομηνίες κτλ. Για την αποθήκευση των τριών ειδών συλλόγων δημιουργούνται οι πίνακες *Bag*, *Seq* και *Alt*. Οι παραπάνω πίνακες έχουν τα εξής πεδία: το πεδίο *cont_id* όπου καταχωρείται το *URI* της συλλογής, το πεδίο *membershipProperty* όπου καταχωρείται ένας αριθμός που δείχνει την διάταξη του μέλους της συλλογής και τα

πεδία *URI*, *string, integer, date* σε κάποιο από τα οποία καταχωρείται το μέλος της συλλογής ανάλογα με τον τύπο του. Στην εικόνα 4.13 απεικονίζεται ο πίνακας *Bag* που περιέχει δύο συλλογές. Το πολυσύνολο bag1 έχει ετερογενή μέλη που ανήκουν στους τύπους int και date. Το πολυσύνολο bag2 περιέχει μόνο URIs (πόρους).

4.3.4 Αναπαράσταση υποστασιοποιημένων δηλώσεων

Για την αναπαράσταση των υποστασιοποιημένων δηλώσεων επεκτείνουμε τους πίνακες-ιδιοτήτων προσθέτοντας το πεδίο *id.* Το id είναι ένα ακέραιο αναγνωριστικό και θα αποτελεί το αναγνωριστικό της υποστασιοποιημένης δήλωσης που αντιστοιχεί στην δήλωση που αποδίδεται το id. Η παραπάνω αναπαράσταση προϋποθέτει ότι και η αντίστοιχη δήλωση έχει καταχωρηθεί στην βάση. Διαφορετικά, οι υποστασιοποιημένες δηλώσεις καταχωρούνται στον πίνακα *Statement* με πεδία *stat_id, pred, sub* και *obj*. Το πεδίο *stat_id* έχει σαν τιμή το URI της υποστασιοποιημένης δήλωσης, τα πεδία *pred, sub* και *obj* αντιστοιχούν στις ιδιότητες *rdf:subject*, *rdf:object* και *rdf:predicate.* Η τιμή του πεδίου *pred* είναι ο κωδικός της ιδιότητας. Στην εικόνα 4.14 απεικονίζεται η αναπαράσταση της υποστασιοποιημένης δήλωσης της εικόνας 2.7 για τις δύο παραπάνω περιπτώσεις.



Εικόνα 4.14. Σχεσιακή αναπαράσταση υποστασιοποιημένων δηλώσεων.

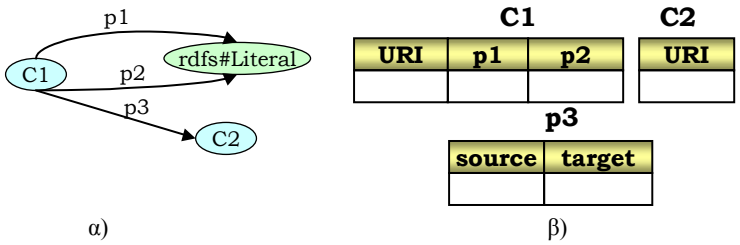
4.3.5 Παραλλαγές στην προτεινόμενη σχεσιακή αναπαράσταση

Ένα μειονέκτημα του γενικού μοντέλου για την αναπαράσταση RDF μεταδεδομένων που έχουμε περιγράψει είναι η πληθώρα των πινάκων που δημιουργούνται όταν τα RDF σχήματα αποτελούνται από μεγάλο αριθμό κλάσεων και ιδιοτήτων. Το γεγονός ότι σε πολλά συστήματα σχεσιακών βάσεων δεδομένων υπάρχει ανώτατο όριο στον αριθμό των πινάκων που μπορούν να δημιουργηθούν κάνει την παραπάνω προσέγγιση μη εφαρμόσιμη σε κάποιες περιπτώσεις. Παράλληλα η δημιουργία μεγάλου αριθμού πινάκων προκαλεί σπατάλη χώρου και αυξάνει τον αριθμό των συζεύξεων που απαιτούνται για την επερώτηση RDF βάσεων περιγραφών. Στην συνέχεια θα προτείνουμε παραλλαγές στην γενική αναπαράσταση που προτείναμε έτσι ώστε να εξαλείψουμε το παραπάνω πρόβλημα. Οι παραλλαγές αυτές βασίζονται σε επιπλέον υποθέσεις που μπορούμε να κάνουμε για τα εκάστοτε RDF σχήματα.

4.3.5.1 Ιδιότητες-Γνωρίσματα και κλάσεις RDF

Όπως έχουμε αναφέρει οι RDF ιδιότητες αντιστοιχούν είτε σε γνωρίσματα (ιδιότητες-γνωρίσματα) είτε σε σχέσεις μεταξύ πόρων (ιδιότητες-σχέσεις). Στην παραλλαγή που προτείνουμε οι ιδιότητες-γνωρίσματα αναπαριστάνονται σαν ***πεδία*** πινάκων-κλάσεων και όχι σαν πίνακες όπως στην γενική αναπαράσταση. Συνεπώς οι πίνακες-κλάσεων εκτός από το πεδίο *URI* έχουν σαν πεδία και τα ονόματα των ιδιοτήτων-γνωρισμάτων που έχουν σαν πεδίο ορισμού την συγκεκριμένη κλάση. Οι πίνακες που αναπαριστάνουν ιδιότητες-σχέσεις παραμένουν αναλλοίωτοι.

Έστω ένα RDF σχήμα που αποτελείται από τις κλάσεις C1, C2 και τις ιδιότητες p1 και p2 και p3 με πεδίο ορισμού την κλάση C1. Το πεδίο τιμών των ιδιοτήτων p1 και p2 είναι η κλάση rdfs:Literal και της ιδιότητας p3 η κλάση C2 (εικόνα 4.15α). Το σχήμα της σχεσιακής βάσης που αντιστοιχεί στο παραπάνω RDF σχήμα απεικονίζεται στην εικόνα 4.15β. Να σημειώσουμε ότι στην εικόνα δεν παρουσιάζονται οι πίνακες *Class* και *Property*.



Εικόνα 4.15. Παραλλαγή σχεσιακής αναπαράστασης βασισμένη στις ιδιότητες-γνωρίσματα.

Υπόθεση για τα RDF σχήματα: Η παραλλαγή αυτή στηρίζεται σε δύο υποθέσεις που κάνουμε για τα RDF σχήματα. Η πρώτη υπόθεση είναι ότι οι ιδιότητες-γνωρίσματα είναι ***μονότιμες***. Η δεύτερη υπόθεση είναι ότι στα RDF σχήματα ***δεν υπάρχουν σχέσεις εξειδίκευσης*** μεταξύ ιδιοτήτων-γνωρισμάτων. Αν κρίνουμε από τα υπάρχοντα RDF σχήματα όπου σπάνια χρησιμοποιείται η σχέση εξειδίκευσης μεταξύ ιδιοτήτων-γνωρισμάτων η παραπάνω υπόθεση δεν είναι περιοριστική.

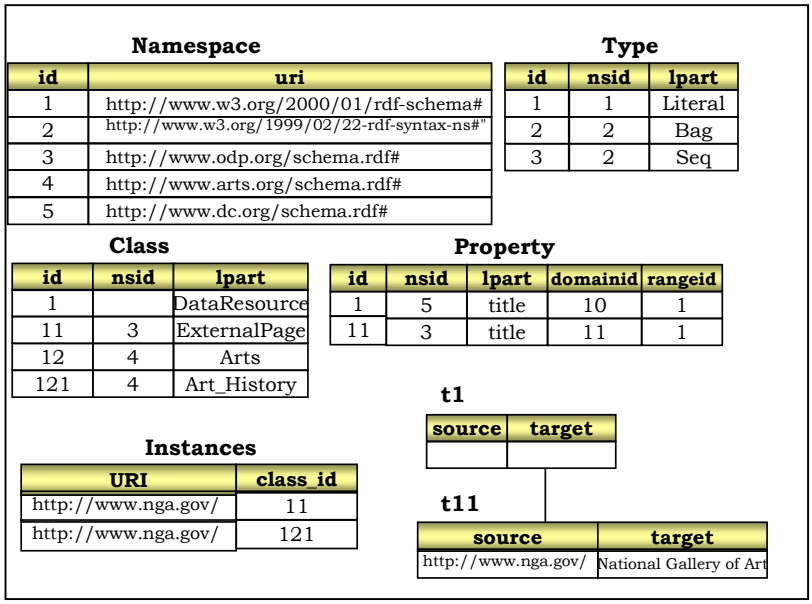
Μια από τις βασικές απαιτήσεις που πρέπει να πληροί το σύστημα αποθήκευσης RDF μεταδεδομένων είναι η δυνατότητα προσθήκης μιας ιδιότητας οποιαδήποτε στιγμή. Η δυνατότητα αυτή παρέχεται και από την παρούσα παραλλαγή. Μπορούμε να προσθέσουμε στο σύστημα μια ιδιότητα-γνώρισμα οποιαδήποτε στιγμή δεδομένου ότι σήμερα τα περισσότερα σχεσιακά συστήματα βάσεων δεδομένων υποστηρίζουν την προσθήκη ενός πεδίου σε ένα πίνακα μετά την δημιουργία του.

Η παραλλαγή που περιγράψαμε προκαλεί μικρότερη διάσπαση στο σχήμα της βάσης η οποία επιφέρει μείωση του αριθμού των πινάκων αλλά και μείωση του αριθμού των συζεύξεων που απαιτούνται κατά την επερώτηση.

4.3.5.2 Παραλλαγή για RDF σχήματα με εικονικές κλάσεις ή ιδιότητες

Η παραλλαγή που προτείνουμε στην συνέχεια βασίζεται στην παρακάτω υπόθεση για τα RDF σχήματα:

Υπόθεση για τα RDF σχήματα: α) Τα RDF σχήματα αποτελούνται από ιεραρχίες κλάσεων/ιδιοτήτων μεγάλου βάθους. β) Υπάρχουν πολλές κλάσεις/ιδιότητες RDF στις οποίες είτε ταξινομούνται ελάχιστοι πόροι είτε δεν ταξινομείται κανένας πόρος. Να σημειώσουμε ότι μια μεγάλη κατηγορία των σχημάτων που υπάρχουν σήμερα έχει τα παραπάνω χαρακτηριστικά.



Εικόνα 4.16. Παραλλαγή για μεγάλου βάθους ιεραρχίες και εικονικές κλάσεις.

Είναι φανερό ότι η δημιουργία πινάκων για κλάσεις/ιδιότητες με ελάχιστα ή μηδενικά μέλη προκαλεί αδικαιολόγητη αύξηση του αριθμού των πινάκων. Για σχήματα με τα παραπάνω χαρακτηριστικά προτείνουμε μια παραλλαγή στην γενική αναπαράσταση με τις παρακάτω διαφοροποιήσεις (εικόνα 4.16).

Αποδίδονται κωδικοί στις κλάσεις/ιδιότητες που βασίζονται στην θέση τους στην ιεραρχία κλάσεων/ιδιοτήτων. Η επεξεργασία των ιεραρχιών θα βασίζεται στους κωδικούς των κλάσεων/ιδιότητων. Άρα ο πίνακας SubClass/SubProperty καταργείται.

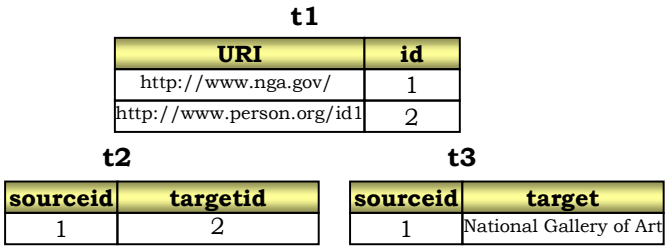
Δεν δημιουργείται ένας πίνακας για κάθε κλάση. Πληροφορία που αναφέρεται στις κλάσεις που ανήκουν οι πόροι καταχωρείται σε ένα νέο πίνακα, τον πίνακα ***Instances.*** Ο πίνακας *Instances* έχει τα πεδία *URI* και *class_id*. Στο πρώτο πεδίο του πίνακα καταχωρείται το URI του πόρου και στο δεύτερο πεδίο ο κωδικός της κλάσης που ανήκει. Προφανώς πολλαπλές εγγραφές του πίνακα Instances μπορεί να έχουν την ίδια τιμή για το πεδίο URI εφόσον ένα πόρος μπορεί να ανήκει σε πολλές κλάσεις. Αντίστοιχη αναπαράσταση μπορούμε να κάνουμε και για τις ιδιότητες και τις περιπτώσεις τους.

Οι πίνακες Namespace, Class, Property και Type παραμένουν αναλλοίωτοι.

4.3.5.3 Απόδοση αναγνωριστικών στους απλούς πόρους

Στην παραλλαγή που προτείνουμε σ' αυτήν την παράγραφο αποδίδουμε ακέραια αναγνωριστικά στους απλούς πόρους. Η μοναδική αλλαγή που προκαλεί η παραλλαγή αυτή στην γενική αναπαράσταση που έχουμε προτείνει είναι ότι τροποποιούνται οι πίνακες-κλάσεων και οι πίνακες-ιδιοτήτων (βλέπε εικόνα 4.17). Συγκεκριμένα, στους πίνακες-κλάσεων προστίθεται το πεδίο *id* που έχει σαν τιμή το αναγνωριστικό που αποδίδεται στον πόρο. Οι πίνακες-ιδιοτήτων εξακολουθούν να έχουν δύο πεδία όμως τιμές των πεδίων είναι τώρα τα αναγνωριστικά των κόμβων αρχής και προορισμού και όχι τα URIs τους (εικόνα 4.17 πίνακας t2). Αν το πεδίο τιμών της ιδιότητας είναι η κλάση rdfs:Literal το πεδίο *target* θα έχει σαν τιμές αλφαριθμητικά και όχι ακέραιους (εικόνα 4.17 πίνακας t3).

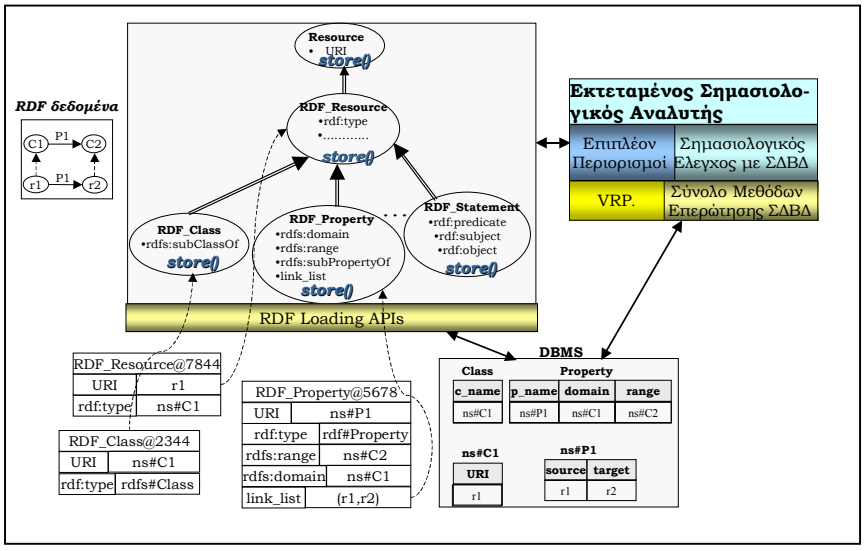
Η αναπαράσταση αυτή απαιτεί μικρότερο χώρο αποθήκευσης. Το ζήτημα όμως είναι κατά πόσο μειώνει τον χρόνο απόκρισης των επερωτήσεων που εκτελούνται στην βάση. Για το σκοπό αυτό πρέπει να μετρηθεί η απόδοση σε διαφορετικούς τύπους επερωτήσεων.



Εικόνα 4.17. Παραλλαγή 3: Απόδοση αναγνωριστικών στους πόρους.

4.4 Αρχιτεκτονική Συστήματος Αποθήκευσης RDF μεταδεδομένων

Στην ενότητα αυτή θα περιγραφεί η αρχιτεκτονική του συστήματος που έχουμε υλοποιήσει για την αποθήκευση RDF μεταδεδομένων σε ένα οντοκεντρικό σχεσιακό σύστημα βάσεων δεδομένων (εικόνα 4.18). Το σύστημα μας αποτελείται από δύο βασικά μέρη λογισμικού. Το πρώτο μέρος ο *Εκτεταμένος Σημασιολογικός Αναλυτής* ελέγχει την συνέπεια των RDF μεταδεδομένων. Το δεύτερο μέρος υλοποιεί την διαδικασία φορτώματος των RDF μεταδεδομένων στην βάση. Στην συνέχεια θα περιγράψουμε αναλυτικά τα δύο κομμάτια.



Εικόνα 4.18. Αρχιτεκτονική συστήματος αποθήκευσης RDF μεταδεδομένων.

4.4.1 Εκτεταμένος Σημασιολογικός Αναλυτής

Η πρώτη λειτουργία του Εκτεταμένου Σημασιολογικού Αναλυτή (ΕΣΑ) είναι να ελέγχει ένα σύνολο περιορισμών που αναφέρονται στο σχήμα. Δύο από τους περιορισμούς που ελέγχει είναι η μοναδικότητα της *rdfs:domain* ιδιότητας και ο υποχρεωτικός καθορισμός πεδίου ορισμού και τιμών στις ιδιότητες, πρόκειται για τους περιορισμούς που έχουμε προσθέσει (βλέπε εικόνα 2.18). Για να ελέγξει αν πληρούνται οι παραπάνω περιορισμοί βασίζεται τόσο στο εσωτερικό μοντέλο του VRP όσο και στην βάση δεδομένων. Για παράδειγμα για να διαπιστώσει αν μια ιδιότητα έχει σαν πεδίο ορισμού ακριβώς μια κλάση, αρχικά ελέγχει αν το RDF_Property αντικείμενο που αντιστοιχεί στην ιδιότητα έχει πολλαπλές τιμές για το γνώρισμα rdfs:domain. Αν η ιδιότητα έχει μόνο μια τιμή για το γνώρισμα rdfs:domain, τότε σε περίπτωση που η

κλάση είναι καταχωρημένη στην βάση ελέγχει αν το πεδίο ορισμού της ιδιότητας της βάσης συμπίπτει με την τιμή της rdfs:domain. Οι επερωτήσεις στην βάση δεδομένων υλοποιούνται από ένα σύνολο μεθόδων (RDF Querying APIs).

Δύο ακόμα περιορισμοί που ελέγχει ο *ΕΣΑ* είναι η μοναδικότητα της ιδιότητας *rdfs:range* και η έλλειψη κύκλων στην ιεραρχία κλάσεων και ιδιοτήτων. Παρόλο που οι περιορισμοί αυτοί ελέγχονται από τον VRP αποδεικνύεται ότι υπάρχουν περιπτώσεις όπου ο συνδυασμός των προς ανάλυση περιγραφών και της καταχωρημένης πληροφορίας στην βάση να οδηγήσει σε παραβίαση των παραπάνω περιορισμών. Αυτό οφείλεται στο πρόβλημα ότι η ένωση δύο εγκύρων RDF σχημάτων δεν είναι ένα έγκυρο RDF σχήμα. Σε συνδυασμό με το γεγονός ότι ο VRP δεν είναι δυνατόν να 'γνωρίζει' όλα τα αρχεία στα οποία βρίσκονται περιγραφές για τις ιδιότητες και κλάσεις που αναλύει.

Η δεύτερη λειτουργία του *ΕΣΑ* είναι να ελέγχει την συνέπεια RDF περιγραφών δεδομένων βασιζόμενος στο σχήμα που είναι καταχωρημένο στην βάση δεδομένων. Δεν συνδέεται δηλαδή στους χώρους ονοματοδοσίας όπου ορίζονται τα σχήματα όπως ο VRP. Προφανώς πρέπει να έχει ήδη αποθηκευτεί στη βάση η σχήμα πληροφορία που χρησιμοποιείται στις περιγραφές. Η ανάγκη για την παραπάνω λειτουργία προέκυψε από το γεγονός ότι συχνά τα RDF σχήματα είναι πολύ μεγάλα. Συνεπώς η ανάλυση των σχημάτων και η αποθήκευση τους στην κύρια μνήμη έχει μεγάλες απαιτήσεις σε μνήμη. Λαμβάνοντας υπόψη και το γεγονός ότι τα μεταδεδομένα μπορεί να βασίζονται σε πολλαπλά σχήματα καταλαβαίνουμε ότι η ποσότητα μνήμης που απαιτείται συνολικά για τον έλεγχο της εγκυρότητας RDF περιγραφών μπορεί να είναι πολύ μεγάλη. Αυτή η προσέγγιση είναι αποδοτική όταν έχουμε μεγάλα σχήματα και σχετικά μικρό όγκο RDF περιγραφών. Αν ο όγκος των RDF περιγραφών είναι πολύ μεγάλος τότε η επικοινωνία με την βάση δεδομένων πιθανόν να επιφέρει αρκετή καθυστέρηση.

4.4.2 Φόρτωση RDF μεταδεδομένων στην βάση δεδομένων

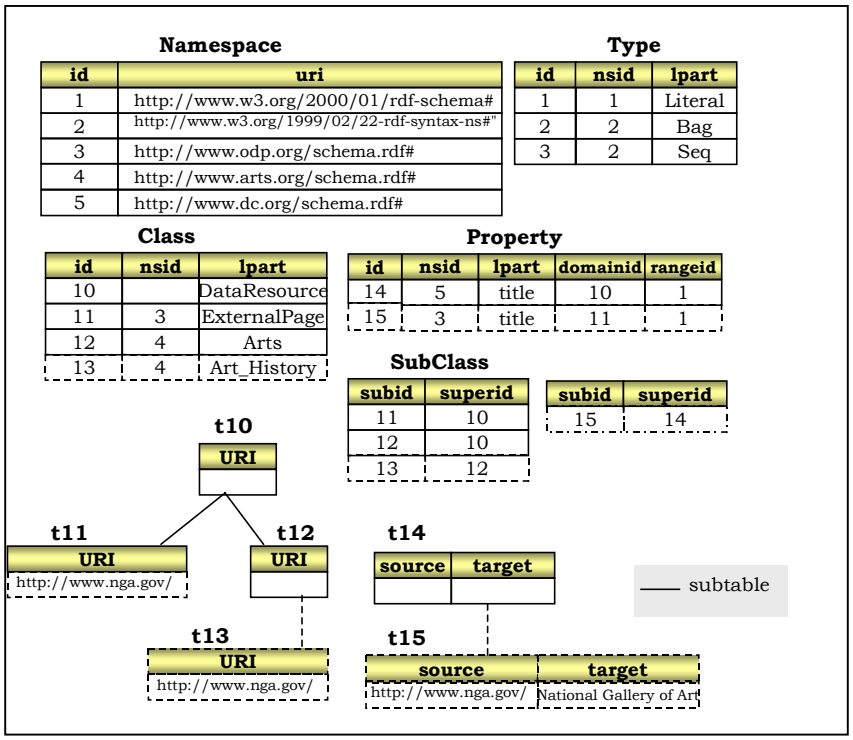
Το δεύτερο μέρος του συστήματος υλοποιεί την αποθήκευση των RDF περιγραφών στην βάση σύμφωνα με το γενικό μοντέλο αναπαράστασης που αναλύσαμε παραπάνω. Επίσης έχουμε υλοποιήσει και την παραλλαγή όπου στους πόρους αποδίδονται ακέραια αναγνωριστικά.

Για την αποθήκευση των RDF περιγραφών έχει δημιουργηθεί ένα σύνολο μεθόδων οι οποίες ορίζονται μέσα στις κλάσεις του μοντέλου του VRP. Για κάθε γνώρισμα των κλάσεων του VRP μοντέλου δημιουργείται μια μέθοδος που αποθηκεύει τις τιμές του γνωρίσματος στην βάση. Για παράδειγμα στην κλάση RDF_Resource

ορίζεται η μέθοδος *storetype()* η οποία αναφέρεται στο γνώρισμα *rdf:type* και αποθηκεύει στην βάση πληροφορία που αφορά τους τύπους του αντικειμένου. Όταν καλείται η μέθοδος *storetype()* για ένα αντικείμενο το URI αντικειμένου καταχωρείται στους πίνακες που αντιστοιχούν στις κλάσεις που αποτελούν τιμές του γνωρίσματος rdf:type. Η μέθοδος *storePropertyInstances()* ορίζεται στην κλάση RDF_Property. Όταν καλείται η μέθοδος *storePropertyInstances()* για ένα αντικείμενο της κλάσης RDF_Property αποθηκεύονται στον πίνακα της βάσης που αντιστοιχεί στην ιδιότητα όλες οι περιπτώσεις της ιδιότητας. Οι μέθοδοι κάθε κλάσης ομαδοποιούνται και σχηματίζονται πιο σύνθετες μέθοδοι αποθήκευσης (ονομάζονται store()) οι οποίες ορίζονται στην ίδια κλάση και τελικά καλούνται για την αποθήκευση των RDF περιγραφών. Λίστα των μεθόδων που έχουν υλοποιηθεί για την αποθήκευση των RDF περιγραφών στην βάση δεδομένων καθώς και για τις επερωτήσεις στη βάση δεδομένων που απαιτούνται τόσο κατά την φάση της αποθήκευσης όσο και κατά τη φάση των ελέγχων για την εγκυρότητα των περιγραφών περιέχονται στο παράρτημα Α.

Στην συνέχεια θα περιγράψουμε την διαδικασία αποθήκευσης των RDF/XML περιγράφων στην βάση δεδομένων. Ο αλγόριθμος που εφαρμόζουμε αποτελείται από δύο φάσεις. Στην πρώτη φάση αποθηκεύονται στην βάση τα αντικείμενα των κλάσεων RDF_Class και RDF_Property. Κατά την φάση αυτή ενημερώνεται το σχήμα της βάσης καθώς για κάθε νέα κλάση ή ιδιότητα που προστίθεται στην βάση δημιουργείται ένας πίνακας. Στην δεύτερη φάση προσθέτονται εγγραφές στους πίνακες της βάσης.

Ας δούμε με ένα παράδειγμα τι επιδράσεις προκαλεί η προσθήκη ενός αντικειμένου των κλάσεων RDF_Class, RDF_Property και RDF_Resource στην βάση δεδομένων. Αρχικά στην βάση έχουν αποθηκευτεί δύο αντικείμενα της κλάσης RDF_Class που αντιστοιχούν στις κλάσεις art#Arts και odp#ExternalPage και ένα αντικείμενο της κλάσης RDF_Property που αντιστοιχεί στην ιδιότητα dc#title. Στην εικόνα 4.19 απεικονίζεται η κατάσταση της βάσης (με διακεκομμένες γραμμές απεικονίζονται οι εγγραφές ή οι πίνακες που θα προστεθούν στην συνέχεια).



Εικόνα 4.19. Περιγραφή αποθήκευσης μεταδεδομένων στην βάση.

Προσθήκη του αντικειμένου *RDF_Class@4455* (εικόνα 9) που αντιστοιχεί στην κλάση art#Art_History θα είχε σαν αποτέλεσμα την προσθήκη μιας εγγραφής στον πίνακα Class με id ίσο με το ακέραιο αναγνωριστικό που αποδίδεται στην κλάση, nsid ίσο με το id που έχει αποδοθεί στο χώρο ονοματοδοσίας με URI http://www.arts.org/schema.rdf και lpart ίσο με Art_History, την προσθήκη μιας εγγραφής στον πίνακα SubClass με τιμές για τα πεδία subid και superid τα αναγνωριστικά που έχουν αποδοθεί στις κλάσεις art#Art_History και art#Arts και την δημιουργία του πίνακα t13 ο οποίος δηλώνεται σαν υποπίνακας του πίνακα t12.

Προσθήκη του αντικειμένου RDF_Property@4567 (εικόνα 4.19) που αντιστοιχεί στην ιδιότητα odp#title θα είχε σαν αποτέλεσμα την προσθήκη μιας εγγραφής στον πίνακα Property, την προσθήκη μιας εγγραφής στον πίνακα SubProperty με τιμές για τα πεδία subid και superid τα αναγνωριστικά που έχουν αποδοθεί στις ιδιότητες odp#title και dc#title και την δημιουργία πίνακα t15 όπου θα αποθηκεύονται οι περιπτώσεις της ιδιότητας odp#title. Ο πίνακας t15 δηλώνεται σαν υποπίνακας του πίνακα t14 που αντιστοιχεί στην ιδιότητα dc#title.

Προσθήκη του αντικειμένου RDF_Resource@4487 (εικόνα 9) που αντιστοιχεί στον πόρο http://www.nga.gov/ θα είχε σαν αποτέλεσμα την προσθήκη μιας εγγραφής με το URI του πόρου στον πίνακα t11 και μιας εγγραφής στον πίνακα t13.

4.4.2.1 Πως χειριζόμαστε τους ανώνυμους πόρους;

Στην παράγραφο αυτή θα αναφέρουμε πως χειριζόμαστε τους ανώνυμους πόρους. Ο VRP σε κάθε ανώνυμο πόρο αποδίδει ένα URI. Το URI είναι συνδυασμός του ονόματος του αρχείου που περιέχεται ο ανώνυμος πόρος και ενός αναγνωριστικού που αυξάνεται κατά ένα για κάθε ανώνυμο πόρο. Για παράδειγμα στον πρώτο ανώνυμο πόρο που συναντάται σε ένα αρχείο αποδίδεται το URI *<όνομα αρχείου>#genID1*.

Τα παραπάνω URIs δεν μπορούν να χρησιμοποιηθούν σαν αναγνωριστικά των πόρων της βάσης και ο λόγος είναι ότι τα URIs που αποδίδονται στους ανώνυμους πόρους ενός αρχείου από τον VRP μπορεί να μεταβληθούν αν τροποποιηθεί το αρχείο. Ας θεωρήσουμε ένα αρχείο στο οποίο ορίζεται μια συλλογή, στον ανώνυμο πόρο που αντιπροσωπεύει την συλλογή αποδίδεται το URI genID1. Έστω ότι το αρχείο τροποποιείται και προστίθεται μια ακόμα συλλογή. Ο VRP αποδίδει το URI genID1 στην νέα συλλογή και το URI genID2 στην παλιότερη. Άρα το URI που αντιστοιχεί στον πόρο μεταβάλλεται.

Για να αποφύγουμε την ύπαρξη διαφορετικών ανώνυμων πόρων με το ίδιο URI τροποποιούμε το URI των ανώνυμων πόρων που αποδίδεται από τον VRP. Τα URIs που αποδίδουμε στους ανώνυμους πόρους είναι της μορφής “id” + <I>. Ο I είναι ένας καθολικός μετρητής για όλη την βάση ο οποίος αυξάνεται κατά ένα για κάθε ανώνυμο πόρο.

4.4.2.2 Πως χειριζόμαστε τις Υποστασιοποιημένες δηλώσεις;

Μια ιδιαίτερη κατηγορία ανώνυμων πόρων είναι οι υποστασιοποιημένες δηλώσεις. Ένα από τα θέματα που έχει συζητηθεί ιδιαίτερα στην λίστα του RDF [rdf-interest][19] είναι κατά πόσο σε υποστασιοποιημένες δηλωσεις με τις ίδιες τιμές για τις ιδιότητες *rdf:predicate*, *rdf:subject* και *rdf:object* πρέπει να αποδίδεται το ίδιο URI [B99]. Η επικρατέστερη άποψη ήταν ότι πρέπει να έχουν το ίδιο URI, δηλαδή αποτελούν ένα μοναδικό πόρο. Στην υλοποίηση μας υποστηρίζουμε την παραπάνω άποψη.

[19] http://lists.w3.org/Archives/Public/www-rdf-interest

Η ιδιαιτερότητα του συστήματος αποθήκευσης που έχουμε υλοποιήσει είναι ότι επιτρέπει την ***βαθμιαία φόρτωση σχήματος και δεδομένων***. Η παραπάνω λειτουργικότητα κρίνεται απαραίτητη εξαιτίας α) των μεγάλων σχημάτων RDF β) των μεγάλων όγκων μεταδεδομένων και γ) της χρήσης πολλαπλών σχημάτων στην δημιουργία των μεταδεδομένων. Παραπάνω αναφέραμε ότι ο έλεγχος της εγκυρότητας των μεταδεδομένων μπορεί να γίνει με βάση τα σχήματα RDF που είναι αποθηκευμένα στην βάση δεδομένων. Αυτό σημαίνει ότι δεν χρειάζεται να ελεγχθεί η εγκυρότητα των σχημάτων που χρησιμοποιούνται στα μεταδεδομένα και επίσης ότι μόνο οι κλάσεις και οι ιδιότητες που χρησιμοποιούνται στα μεταδεδομένα χρειάζεται να κρατούνται στην μνήμη.

Επίσης επιτρέπεται η φόρτωση μεγάλων σχημάτων RDF (π.χ. κατηγορίες Open Directory) – τα οποία θα ήταν αδύνατο να φορτωθούν – χωρίζοντας τα σε επιμέρους μικρότερα σχήματα. Στην περίπτωση αυτή στη μνήμη χρειάζεται να κρατείται μόνο το μέρος του σχήματος το οποίο πρόκειται να φορτωθεί και οι κλάσεις και ιδιότητες άλλων σχημάτων με τις οποίες το RDF σχήμα συσχετίζεται. Να σημειώσουμε εδώ ότι η βαθμιαία φόρτωση σχήματος στην βάση δεδομένων γίνεται αποτελεσματικότερη αν γνωρίζουμε ότι τα RDF σχήματα που αποθηκεύονται πληρούν τους επιπλέον περιορισμούς που έχουμε θέσει (εικόνα 2.18). Αυτό οφείλεται στο γεγονός ότι μπορούμε να παραλείψουμε τους σημασιολογικούς ελέγχους που εκτελούνται από τον Εκτεταμένο Σημασιολογικό Αναλυτή εφόσον η ένωση των σχημάτων θα είναι ένα έγκυρο σχήμα.

4.4.3 Υλοποίηση Συστήματος

Το σύστημα μας έχει υλοποιηθεί στην γλώσσα προγραμματισμού Java (έκδοση 1.2). Για την αποθήκευση των μεταδεδομένων χρησιμοποιήθηκε η PostgreSQL[20], ένα οντοκεντρικό σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων. Η επικοινωνία με την βάση δεδομένων γίνεται μέσω της διεπιφάνειας προγραματισμού εφαρμογών για επικοινωνία με βάσεις δεδομένων της Java (JDBC API 2.0). Δεδομένου ότι το σύστημα μας έχει υλοποιηθεί στην Java καθιστά δυνατή την λειτουργία του σε πολλαπλές πλατφόρμες λογισμικού και υλικού (portable). Παράλληλα, για την αποθήκευση των δεδομένων μπορεί να χρησιμοποιηθεί, χωρίς να τροποποιηθεί η εφαρμογή μας, οποιαδήποτε άλλη οντοκεντρική σχεσιακή βάση δεδομένων η οποία υποστηρίζεται από την διεπιφάνεια προγραμματισμού εφαρμογών JDBC (π.χ. Oracle).

[20] http://www.postgresql.org/index.html

4.4.3.1 PostgreSQL

Η PostgreSQL συγκαταλέγεται στις οντοκεντρικές σχεσιακές (object-relational) βάσεις δεδομένων. Υποστηρίζει την γλώσσα επερωτήσεων για σχεσιακές βάσεις δεδομένων, SQL. Παράλληλα όμως υποστηρίζει και οντο-κεντρικά χαρακτηριστικά όπως είναι η έννοια της κλάσης και της κληρονομικότητας. Η έννοια *κλάση* αντιστοιχεί στην έννοια του *πίνακα* στις σχεσιακές βάσεις, όμως οι κλάσεις στην PostgreSQL έχουν επιπλέον χαρακτηριστικά? το κυριότερο είναι η κληρονομικότητα. Μια κλάση είναι μια συλλογή από αντικείμενα. Κάθε αντικείμενο μιας συλλογής έχει τα ίδια γνωρίσματα και κάθε γνώρισμα έχει ένα συγκεκριμένο τύπο. Μια κλάση μπορεί να κληρονομεί από μία ή περισσότερες κλάσεις. Για παράδειγμα έστω ότι ορίζουμε την κλάση/πίνακα *Καλλιτέχνης(όνομα, επίθετο)* και την κλάση *Ζωγράφος (πίνακας)* που κληρονομεί από την κλάση Καλλιτέχνης. Τα μέλη της κλάσης Ζωγράφος έχουν τα γνωρίσματα *όνομα, επίθετο* και *ζωγράφος*. Στην ιεραρχία των κλάσεων δεν επιτρέπεται η δημιουργία κύκλου. Με κατάλληλες SQL επερωτήσεις στην κλάση *Καλλιτέχνης* μπορούμε να ανακτήσουμε είτε μόνο τα άμεσα μέλη της κλάσης είτε και τα μέλη των υποκλάσεων της.

4.4.4 Πειράματα – Αποτελέσματα

Στην ενότητα αυτή θα παρουσιάσουμε κάποια προκαταρκτικά αποτελέσματα που έχουμε λάβει για τις επιδόσεις του συστήματος και αναφέρονται στο χώρο αποθήκευσης και το χρόνο φόρτωσης τόσο σχημάτων όσο και δεδομένων. Για τα πειράματα μας χρησιμοποιήθηκε ο κατάλογος και οι περιγραφές του Open Directory.

Κατά την εκτέλεση των πειραμάτων μας ο εξυπηρετητής της βάσης όσο και η εφαρμογή μας έτρεχαν σε μηχάνημα ULTRA 60 στα 450 MHz και κύρια μνήμη 250 MBytes.

Οι πίνακες 4.2 και 4.3 απεικονίζουν το μέγεθος του αρχείου που πρόκειται να αποθηκευτεί στην βάση, τον αριθμό των τριάδων (δηλώσεων) που περιέχονται σ' αυτό, το χρόνο αποθήκευσης του στη βάση δεδομένων, τον χώρο που καταλαμβάνει στην βάση δεδομένων και το συνολικό μέγεθος της βάσης δεδομένων μετά το φόρτωμα του εκάστοτε αρχείου.

Μέγεθος αρχείου (MB)	**Αριθμός Τριάδων**	**Χρόνος αποθήκευσης (sec)**	**Χώρος αποθήκευσης (MB)**	**Συνολικό Μέγεθος Βάσης (MB)**
kids.rdf(0.1)	1202	6	1834	3263
news.rdf(0,11)	1432	9	1987	5250
shopping.rdf (0.5)	6362	81	8634	13884
health.rdf (0.55)	5918	119	8135	22019
games.rdf (0.75)	9214	249	12648	34667
home.rdf (0.95)	9516	337	13424	48091
computers.rdf (0.98)	11648	527	16032	64123
recr.rdf (1.3)	14432	827	20048	84171
business.rdf (1,3)	13158	921	18336	102507
refer.rdf(1,6)	12584	1030	18144	120651
sports.rdf(1,65)	20430	2022	28176	148827
science(1,73)	17032	2020	24032	172859
society.rdf(3,25)	31636	4569	44472	217331
arts.rdf(3.7)	48664	8851	66944	284275

Πίνακας 4.2. Στατιστικά στοιχεία γιατο χώρο και χρόνο αποθήκευσης σχημάτων.

Μέγεθος αρχείου (MB)	**Αριθμός Τριάδων**	**Χρόνος αποθήκευ-σης (sec)**	**Χώρος αποθήκευσης (MB)**	**Συνολικό Μέγεθος Βάσης (MB) start size:741817**
kids.rdf (2.7)	23448	132	6536	748353
news.rdf (22)	182993	480	28776	777665
health.rdf(18.5)	132239	601	37168	815225
games.rdf(13)	107547	1869	44168	860617
home.rdf(10)	83461	681	36736	900041
computers.rdf(33)	262366	1297	68178	972161

Πίνακας 4..3. Στατιστικά στοιχεία για το χώρο και χρόνο αποθήκευσης δεδομένων.

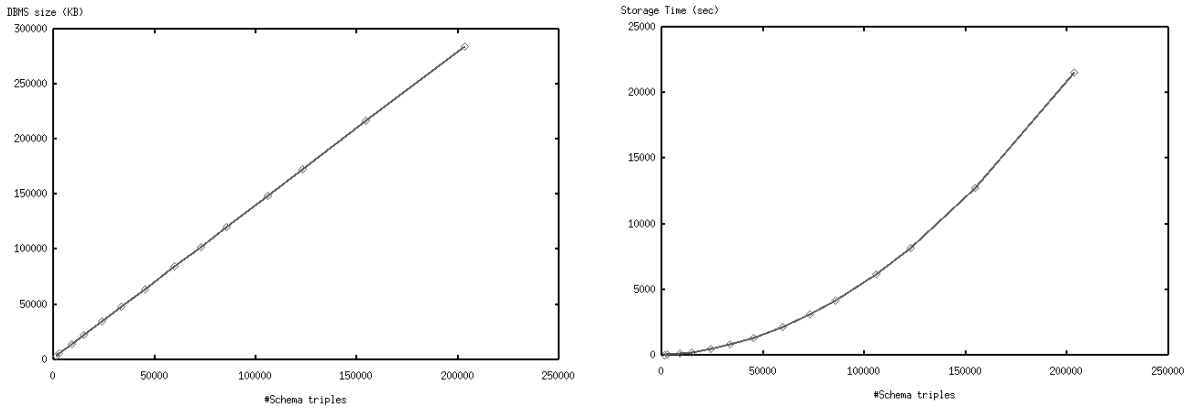
Μέγεθος Βάσης: Στην εικόνα 4.20 παρατηρούμε ότι όταν φορτώνουμε RDF σχήματα το μέγεθος της βάσης αυξάνεται γραμμικά με το μέγεθος των τριάδων που προστίθενται. Ο χώρος που απαιτείται για την αποθήκευση σχημάτων είναι περίπου 14 φορές μεγαλύτερος από το μέγεθος του αρχείου που περιέχει το σχήμα. Αυτό οφείλεται στην πληθώρα των πινάκων (248236) που δημιουργούνται κατά την φόρτωση σχήματος σε συνδυασμό με το γεγονός ότι η βάση δεδομένων για κάθε πίνακα που δημιουργείται αποθηκεύει πληροφορία π.χ. για τα γνωρίσματα του τις τιμές τους, τους δείκτες που κατασκευάζονται. Να σημειώσουμε ότι το 65% του συνολικού χώρου δαπανείται για την αποθήκευση των γνωρισμάτων των πινάκων.

Στην εικόνα 4.21 παρατηρούμε ότι ο χώρος που καταλαμβάνουν τα δεδομένα δεν αυξάνεται γραμμικά με το μέγεθος των τριάδων. Παρατηρούμε ότι όσο μικραίνει ο λόγος *αριθμός κλάσεων/αριθμό URIs* τόσο μικραίνει και ο χώρος που απαιτείται. Ο χώρος που καταλαμβάνουν τα δεδομένα στην βάση είναι 1-3 φορές μεγαλύτερος από το μέγεθος του αρχείου.

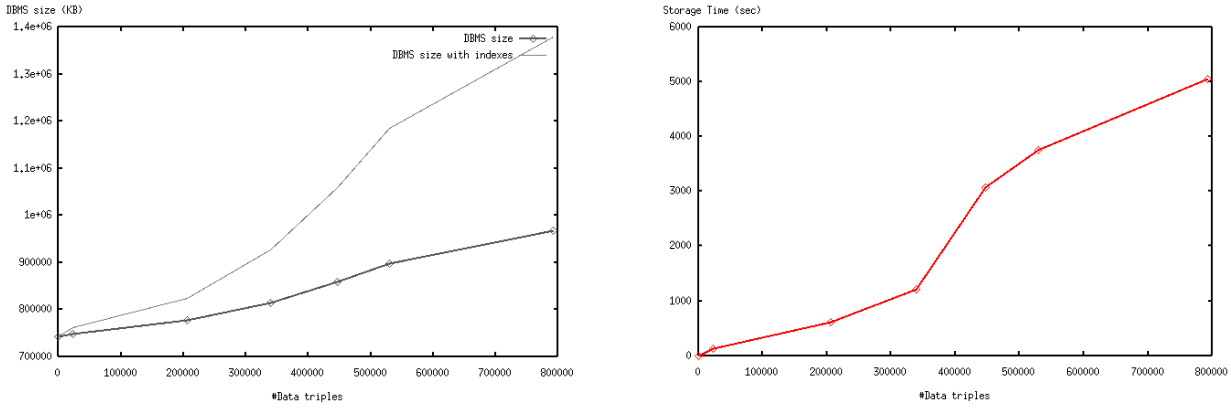
Στην εικόνα 4.21 απεικονίζεται και το μέγεθος της βάσης όταν προστίθενται δείκτες σε όλα τα γνωρίσματα των πινάκων ιδιοτήτων και κλάσεων Παρατηρούμε ότι η γραφική παράσταση έχει την ίδια μορφή με την γραφική παράσταση για τα δεδομένα . Ο χώρος που καταλαμβάνουν τα δεδομένα στην βάση όταν προστίθενται δείκτες είναι περίπου 6 φορές μεγαλύτερος από το μέγεθος του αρχείου.

Χρόνος αποθήκευσης: Στην εικόνα 4.20 παρατηρούμε ότι ο χρόνος που απαιτείται για την αποθήκευση σχήματος αυξάνεται καθώς το μέγεθος της βάσης μεγαλώνει. Αυτό οφείλεται στους ελέγχους που πρέπει να εκτελεστούν στην βάση, όπως να ελέγξει αν ένας πίνακας έχει ήδη δημιουργηθεί ή να ενημερώσει του δείκτες που κατασκευάζονται στους πίνακες που αποθηκεύουν τα γνωρίσματα των πινάκων της βάσης, οι οποίοι απαιτούν περισσότερο χρόνο καθώς το μέγεθος της βάσης μεγαλώνει.

Ο χρόνος αποθήκευσης των δεδομένων, όπως και ο χώρος, εξαρτάται από τον αριθμό των κλάσεων που ταξινομούνται τα URIs. Όσο μικραίνει ο *λόγος αριθμός κλάσεων/αριθμός URIs* τόσο μικραίνει και ο χρόνος αποθήκευσης εφόσον μειώνεται ο χρόνος αναζήτησης της βάσης για την εύρεση των πινάκων που θα καταχωρηθούν τα δεδομένα .



Εικόνα 4.20.Γραφική παράσταση για το χώρο και χρόνο αποθήκευσης RDF σχημάτων σε σχέση με τον αριθμό των τριάδων.



Εικόνα 4.21. Γραφική παράσταση για το χώρο και χρόνο αποθήκευσης δεδομένων σε σχέση με τον αριθμό των τριάδων.

4.5 Φυσικό Μοντέλο σχεσιακής αναπαράστασης RDF μεταδεδομένων

4.5.1 Δείκτες

Στην συνέχεια θα αναφέρουμε τους δείκτες που κατασκευάζονται έτσι ώστε να επιταχύνουμε την διαδικασία αποθήκευσης των δεδομένων στην βάση. Στον πίνακα NameSpace δημιουργείται ένας δείκτης στο πεδίο uri. Ο δείκτης χρησιμοποιείται όταν ανακτούμε το αναγνωριστικό που αποδίδεται σε κάποιο χώρο ονοματοδοσίας. Η επερώτηση αυτή εκτελείται όταν αποθηκεύουμε στην βάση κλάσεις και ιδιότητες ή όταν ρωτάμε για την ύπαρξη τους. Σε κάθε ένα από τους πίνακες Class και Property δημιουργούμε ένα δείκτη στο πεδίο lpart. Οι δείκτες αυτοί επιταχύνουν την εύρεση των κλάσεων και των ιδιοτήτων. Στους πίνακες *SubClass* και *SubProperty* δημιουργούνται δείκτες στο πεδίο *subid* ώστε να επιταχυνθεί η ανάκτηση των υπερκλάσεων ή των υπεριδιοτήτων. Στους πίνακες που αντιστοιχούν σε κλάσεις δημιουργούμε δείκτη στο πεδίο uri. Ο δείκτης αυτός επιταχύνει τους ελέγχους που γίνονται για το αν ο πόρος έχει αποθηκευτεί στην βάση. Στην εναλλακτική σχεσιακή αναπαράσταση που προτείναμε ο δείκτης αυτός χρησιμοποιείται και για την εύκολη ανάκτηση των αναγνωριστικών των πόρων. Επίσης σε κάθε πίνακα που αντιστοιχεί σε ιδιότητα δημιουργούμε δείκτη σε κάποιο από τα πεδία του έτσι ώστε να διευκολύνουμε το έλεγχο για το αν μια εγγραφή έχει καταχωρηθεί στην βάση. Οι δείκτες στους πίνακες ιδιοτήτων μπορούν να παραληφθούν αν γνωρίζουμε ότι στα δεδομένα που επεξεργαζόμαστε δεν αποδίδεται σε ένα πόρο πολλές φορές μια ιδιότητα με την ίδια τιμή.

Η Postgresql υποστηρίζει τρία είδη δεικτών B-tree, Hash και Rtree. Ένας B-tree δείκτης είναι πιθανόν να χρησιμοποιηθεί για την εκτέλεση μιας ερώτησης όταν το πεδίο στο οποίο έχει δημιουργηθεί ο δείκτης περιέχεται στην επερώτηση και εφαρμόζεται σ' αυτό οποιοσδήποτε από τους παρακάτω τελεστές σύγκρισης <, <=, =, >=, >. Αντίθετα ένας Hash δείκτης είναι πιθανόν να χρησιμοποιηθεί μόνο όταν στο πεδίο εφαρμόζεται ο τελεστής ισότητας =. Ο δείκτης Rtree είναι κατάλληλος για δυσδιάστατα δεδομένα, π.χ. πολύγονα. Όλοι οι παραπάνω δείκτες που δημιουργούμε είναι B-tree. Για τις περισσότερες παραπάνω περιπτώσεις θα μπορούσαν να χρησιμοποιηθούν και Hash δείκτες εφόσον στα πεδία που δεικτοδοτούνται εφαρμόζεται τελεστής ισότητας, επειδή όμως στην Postgresql ο δείκτης B-tree είναι πιο σταθερός και υποστηρίζει και παράλληλη ενημέρωση, (ο δείκτης Hash δεν υποστηρίζει) επιλέξαμε τον δείκτη Hash. Όμως για τους δείκτες που δημιουργούνται στα πεδία των πινάκων που αντιστοιχούν σε

ιδιότητες όπως επίσης και στα πεδία των πινάκων που αντιστοιχούν σε κλάσεις ενδείκνυται να χρησιμοποιηθούν B-tree δείκτες δεδομένου ότι επιτρέπουν διατεταγμένη σάρωση των πινάκων. Η δυνατότητα αυτή μπορεί να κάνει πιο αποδοτικές τις συζεύξεις μεταξύ των πινάκων.

Κεφάλαιο 5

Επίλογος

Στην εργασία αυτή προτείναμε μια θεωρητική θεμελίωση του RDF χρησιμοποιώντας την γλώσσα παράστασης γνώσης Telos.

Επίσης προτείναμε μια γενική αναπαράσταση των RDF μεταδεδομένων σε οντοκεντρικές σχεσιακές βάσεις δεδομένων. Η αναπαράσταση αυτή υποστηρίζει όλη την βασική εκφραστική δύναμη του RDF με εξαίρεση τους περιορισμούς που έχουμε θέσει για την εγκυρότητα της ένωσης πολλαπλών σχημάτων RDF. Σε σχέση με τις υπόλοιπες αναπαραστάσεις που έχουν προταθεί μέχρι τώρα για την αποθήκευση RDF μεταδεδομένων οι οποίες χρησιμοποιούν ένα μοναδικό πίνακα για την αποθήκευση σχήματος και δεδομένων, η προσέγγιση μας καθιστά την διαδικασία φορτώματος πιο πολύπλοκη και χρονοβόρα. Το σημαντικό όμως είναι ότι στην προσέγγιση μας οι επερωτήσεις έχουν ταχύτερους χρόνους απόκρισης.

Επιπλέον προτείνουμε μια σειρά από τροποποιήσεις στην γενική αναπαράσταση οι οποίες βασίζονται σε υποθέσεις για τα σχήματα που αποσκοπούν στο να μειώσουν τον αριθμό των πινάκων που δημιουργούνται καθώς και το αριθμό των συζεύξεων (joins) μεταξύ των πινάκων.

Τέλος, η δυνατότητα *βαθμιαίας φόρτωσης* RDF δεδομένων και σχημάτων στη βάση κρίθηκε απαραίτητη εξαιτίας του μεγάλου όγκου RDF μεταδεδομένων που υπάρχουν *στις πύλες κοινοτήτων διαδικτύου* π.χ. κατάλογος ODP. Η δυνατότητα αυτή αποτελεί και ένα ιδιαίτερο χαρακτηριστικό του συστήματος που έχει υλοποιηθεί.

5.1 Μελλοντικές Κατευθύνσεις

Από τις παραλλαγές που προτάθηκαν η πιο σημαντική είναι η κωδικοποίηση κλάσεων/ιδιοτήτων με βάση την θέση τους στην ιεραρχία κλάσεων/ιδιοτήτων αντίστοιχα. Μια τέτοια κωδικοποίηση αναμένεται να βελτιώσει σε μεγάλο βαθμό το χρόνο που απαιτείται για την επεξεργασία των ιεραρχιών. Στην τωρινή υλοποίηση ήδη χρησιμοποιούνται κωδικοί οι οποίοι είναι αυξητικοί, άρα η προσθήκη αυτή μπορεί εύκολα να ενσωματωθεί στο σύστημα μας. Στην εργασία [L00a] προτείνεται ένα σύνολο κωδικοποιήσεων ιεραρχιών οι οποίες όμως μπορούν να εφαρμοστούν για μονής κληρονομικότητας ιεραρχίες.

Για την αξιολόγηση του συστήματος μας απαιτείται να πραγματοποιηθούν συστηματικά πειράματα χρησιμοποιώντας τις παραλλαγές της γενικής αναπαράστασης άλλα και διαφορετικούς αλγόριθμους κωδικοποίησης ιεραρχιών κλάσεων και ιδιοτήτων

Ένα θέμα που δεν έχουμε θίξει σ' αυτήν την εργασία είναι η ενημέρωση των RDF μεταδεδομένων. Το RDF δεν παρέχει την δυνατότητα ενημέρωσης των μεταδεδομένων καθώς η XML σύνταξη με την οποία αυτά αναπαριστάνονται δεν υποστηρίζει την ενημέρωση. Θεωρούμε ότι πρέπει να υποστηρίξουμε ενημέρωση μόνο για τα δεδομένα και όχι για το σχήμα. Όπως προτείνεται άλλωστε και στο RDF, όταν απαιτείται η μεταβολή κάποιου σχήματος (π.χ. όταν προκύπτει νέα πληροφορία για το πεδίο εφαρμογής που περιγράφει είτε όταν απαιτείται περισσότερη εξειδίκευση) θα πρέπει να δημιουργείται ένα νέο σχήμα (με διαφορετικό δηλαδή URI) και όχι να μεταβάλλεται το προηγούμενο. Με την παραπάνω σύμβαση αποφεύγονται προβλήματα ασυνέπειας που πιθανόν θα προέκυπταν σε μεταδεδομένα που βασίζονται σε σχήματα που μεταβάλλονται. Η ενημέρωση των δεδομένων μπορεί ενσωματωθεί στην υπάρχουσα XML σύνταξη. Αρκεί να οριστεί ένα σύνολο κλάσεων και ιδιοτήτων οι οποίες θα υποστηρίζουν την ενημέρωση των δεδομένων και οι οποίες πρέπει θα έχουν ειδική διαχείριση από το σύστημα αποθήκευσης.

Καταλήγοντας, να σημειώσουμε ότι το σύστημα που έχουμε προτείνει καλύπτει τις ανάγκες αποθήκευσης των μεταδεδομένων *πυλών κοινοτήτων διαδικτύου*.

Παράρτημα Α

Μέθοδοι Αποθήκευσης και Επερώτησης

RDF_Class

Μέθοδοι Επερώτησης

- boolean **isStoredClass**()

Ελέγχει αν η κλάση έχει αποθηκευτεί στην βάση. Επιστρέφει true ή false.

- boolean **checkStoredsubClassOf():**

Ελέγχει αν οι υπερκλάσεις της κλάσης συμπίπτουν με αυτές που έχουν καταχωρηθεί στην βάση.

Μέθοδοι Αποθήκευσης

- void **storesubClassOf()**:

Στον πίνακα SubClass καταχωρείται πληροφορία για τις σχέσεις υποσύνολου της κλάσης. Σε περίπτωση που δεν υπάρχει καμία υπερ-κλάση τότε καταχωρείται στον πίνακα SubClass η πληροφορία ότι είναι υποκλάση της *DataResource*.

- void **createTable()**

Δημιουργεί τον πίνακα που αντιστοιχεί στην κλάση. Αν η κλάση έχει υπερκλάσεις, ο πίνακας που δημιουργείται δηλώνεται σαν υποπίνακας των πινάκων που αντιστοιχούν στις υπερ-κλάσεις της. Αν ο πίνακας δεν έχει υπερ-κλάσεις τότε κληρονομεί από την κλάση *DataResource.*

- void **storeClass**(int id):

Αποθηκεύει την κλάση στο πίνακα Class.

- void **store()**:

Αποθηκεύει τις RDF/S ιδιότητες της κλάσης.

RDF_Property

Μέθοδοι Επερώτησης

- boolean **isStoredProperty():**

Ελέγχει αν η κλάση έχει αποθηκευτεί στην βάση.

- boolean **israngeDefined()**

Ελέγχει αν το πεδίο τιμών της ιδιότητας είναι ορισμένο.

- boolean **isdomainUnique()**:

Ελέγχει αν η ιδιότητα έχει μοναδική rdfs:domain ιδιότητα.

- int[] **getPropertyDR()**:

Επιστρέφει τους κωδικούς που έχουν αποδοθεί στο πεδίο ορισμού και τιμών της ιδιότητας.

- boolean **hasType(String res, int type)***:*

Ελέγχει αν ο πόρος με URI res ανήκει στην κλάση με κωδικό type.

- boolean **checkStoredsubPropertyOf():**

Ελέγχει αν οι υπερ-ιδιότητες της ιδιότητα συμπίπτουν με αυτές που έχουν καταχωρηθεί στην βάση.

- boolean **checkRange(String to, int rng_id):**

Ελέγχει αν ο κόμβος προορισμού της ιδιότητας ανήκει στο πεδίο τιμών της ιδιότητας.

- boolean **checkDomain(String from, int dom_id)*:***

Ελέγχει αν ο κόμβος αφετηρίας της ιδιότητας ανήκει στο πεδίο ορισμού της ιδιότητας.

Μέθοδοι Αποθήκευσης

- void **storeProperty(int id):**

Αποθηκεύει την ιδιότητα στον πίνακα Property.

- void **storesubPropertyOf():**

Στον πίνακα SubProperty καταχωρείται πληροφορία για τις σχέσεις υπο-ιδιότητας της ιδιότητας.

- void **createTable():**

Δημιουργεί τον πίνακα που αντιστοιχεί στην ιδιότητα. Αn η ιδιότητα έχει υπερ-ιδιότητες ο πίνακας που δημιουργείται δηλώνεται σαν υποπίνακας των πινάκων που αντιστοιχούν στις υπερ-ιδιότητες του.

- void **storelink(String from, String to, boolean isToLit):**

Αποθηκεύει το σύνδεσμο from-to (περίπτωση ιδιότητας) στον πίνακα που αντιστοιχεί στην ιδιότητα. Αποθηκεύονται είτε τα URIs, είτε οι κωδικοί των πόρων.

- public void **store():**

Αποθηκεύει τις RDF/S ιδιότητες της ιδιότητας καθώς και τις περιπτώσεις της ιδιότητας.

RDF_Resource:

Μέθοδοι Επερώτησης

- String **getUriNs(String uri):**

Επιστρέφει το namespace που ορίζεται ο πόρος.

- int **getNsId(String ns)**:

Επιστρέφει το αναγνωριστικό που έχει αποδοθεί στο χώρο ονοματοδοσίας.

- void **assignID(String s):**

Μεταβάλλει το URI των ανώνυμων πόρων.

- boolean **isStoredUnClassifiedResource():**

Ελέγχει αν ο πόρος έχει καταχωρηθεί σε πίνακα διαφορετικό από τον πίνακα που αντιστοιχεί στην κλάση DataResource.

Μέθοδοι Αποθήκευσης

- public void **storetype():**

Αποθηκεύει τον υποκείμενο πόρο (είτε μόνο το URI του πόρου, είτε και το id) στους πίνακες που αντιστοιχούν στις κλάσεις που ανήκει ο πόρος.

- void **storecomment()/storelabel()**:

Αποθηκεύει τις περιπτώσεις της ιδιότητας rdfs:comment/rdfs:label για τον συγκεκριμένο πόρο.

- void **storeseeAlso()/storeisDefinedBy**:

Αποθηκεύει τις περιπτώσεις της ιδιότητας rdfs:seeAlso/rdfs:isDefinedBy για τον συγκεκριμένο πόρο.

- public void **store():**

Αποθηκεύει τις RDF/S ιδιότητες του πόρου.

Resource

Μέθοδοι Επερώτησης

- boolean **isStoredResource():**

Ελέγχει αν πόρος έχει αποθηκευτεί στην βάση.

Μέθοδοι Αποθήκευσης

- public void **store():**

Αποθηκεύει τον πόρο στον πίνακα DataResource (αν δεν έχει αποθηκευτεί σε κάποιο υποπίνακα του).

RDF_Container:

Μέθοδοι Αποθήκευσης

- public void **storeContainerMembershipProperty():**

Αποθηκεύει την συλλογή στον κατάλληλο πίνακα (Bag, Seq, Alt)

- public void **store():**

Αποθηκεύει τις RDF/S ιδιότητες της συλλογής.

RDF_Statement:

Μέθοδοι Αποθήκευσης

- public void **storeStatement()**:

Αποθηκεύει την υποστασιοποιημένη δήλωση στον πίνακα Statement.

- public void **store():**

Αποθηκεύει τις RDF/S ιδιότητες της υποστασιοποιημένης δήλωσης.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[ABS99] S. Abiteboul, P. Buneman, and D. Suciu. *Data on the Web: From Relations to Semistructured Data and XML*. Morgan Kaufmann, 1999.

[ASC97] A. Analyti, N. Spyratos and P. Constantopoulos. *On the Definition of Semantic Network Semantics.* Technical Report FORTH-ICS/TR-187, 45 pages, February 1997.

[B98] Tim Bray. *RDF and Metadata*. June 1998. Available at http://www.xml.com/xml/pub/98/06/rdf.html.

[B00] Harold Boley. *Relationships Between Logic Programming and RDF*. Revised from Version in: Proc. PRIIA 2000, in conjunction with PRICAI 2000, Melbourne, August. 2000. Available at http://www.dfki.uni-kl.de/~boley/rdfphtml/.

[BG00] Dan Brickley and R.V. Guha. *Resource Description Framework (RDF) Schema Specification 1.0.* W3C Candidate Recommendation, March 2000. Avalilable at http://www.w3.org/TR/rdf-schema.

[BHL99] Tim Bray, Dave Hollander, and Andrew Layman. *Namespaces in XML.* W3C Recommendation, Jan 1999. Available at http://www.w3.org/TR/REC-xml-names.

[BKD+00] J. Broekstra, M. Klein, S. Decker, D. Fensel, and I. Horrocks. *Adding formal semantics to the Web: building on top of RDF Schema.* Available at http://www.cs.vu.nl/~jbroeks/papers/extending-rdfs.pdf.

[BM00] Paul V. Biron and Ashok Malhotra. *XML Schema Part 2: Datatypes*. W3C Working Draft, 07 April 2000. Available at http://www.w3.org/TR/xmlschema-2/.

[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.w3.org/TR/REC-xml.

[C00] Pierre-Antoine Champin. *RDF Tutorial*. June 2000. Available at http://www710.univ-lyon1.fr/~champin/rdf-tutorial/.

[CDH00] O. Corby, R.Dieng, and C. Hebert. *A Conceptual Graph Model for W3C Resource Description Framework*. To appear in Proc. Of the 8th International Conference on Conceptual Structures Logical, Linguistic, and Computational Issues, Springer-Verlag, Darmstadt, Germany, August 2000, Springer-Verlag.

[CK00] Wolfram Conen and Reinhold Klapsing. *A Logical Interpretation of RDF*. Discussion Paper October 2000. Available at http://nestroy.wi-inf.uni-essen.de/rdf/logical_interpretation/.

[DC] *Dublin Core Metadata Initiative*. Available at http://purl.oclc.org/dc/.

[Delphi99] The Delphi Group Releases Model for Characterizing Portal Markets. San Diego, CA – Delphi's International Knowledge Management Summit March 29, 1999. Available at http://wwd.delphigroup.com/pressreleases/1999-PR/03291999IKMSPortalModel.html

[DESIRE] DESIRE *Project: Development of a European Service for Information on Research and Education*. 1998-2000. Available at http://www.desire.org/.

[DFS99] A. Deutsch, M. F. Fernandez, and D. Suciu. *Storing Semistructured Data with STORED*. In Proceedings of the ACM SIGMOD International Conference on Management of Data, pages 431{442, Philadelphia, PA, USA, 1999. Available at http://www.research.att.com/~suciu/strudel/external/files/_F1652672656.ps.

[DOM99] *Document Object Model (DOM*). Available at http://www.w3.org/DOM/.

[FA99] C. Finkelstein and P. Aiken. *Building Corporate Portals with XML*. McGaw-Hill, 1999.

[FK99] Daniela Florescu and Donald Kossmann. *A Performance Evaluation of Alternative Mapping Schemes for Storing XML Data in a Relational Database.* Rapport de Recherche No. 3680 INRIA, Rocquencourt, France, May 1999. Available at http://www-caravel.inria.fr/dataFiles/GFSS00.ps.

[GB97] R.V. Guha and Tim Bray. *Meta Content Framework using XML.* June 1997. Available at http://www.w3.org/TR/NOTE-MCF-XML/.

[G00] R.V. Guha. *RdfDB*. Available at http://web1.guha.com/rdfdb/.

[HH00] Jeff Heflin and James Hendler. *Dynamic Ontologies on the Web*. Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000). AAAI/MIT Press, Menlo Park, CA, 2000. Pp. 443-449. Available at http://www.cs.umd.edu/projects/plus/SHOE/pubs/#aaai2000.

[HFB+ 00] Horrocks, D. Fensel, J. Broekstra, S. Decker, M. Erdmann, C. Goble, F. Van Harmelen, M. Klein, S. Staab, and R. Studer. OIL: *The Ontology Inference Layer*. Technical report, University of Manchester / Vrije Univer-siteit Amsterdam, 2000. Available at http://www.ontoknowledge.org/oil/.

KG97] Brigitte Kerherve and Olivier Gerbe. *Models for Metadata or Metamodels for Data?*. 2nd IEEE Metadata Conference, Silver Spring, September 1997.

[KMRT96] Tim Krauskopf, Jim Miller, Paul Resnick, and Win Treese. *PICS Label Distribution Label Syntax and Communication Protocols*. W3C Recommendation, October 1996. Available at http://www.w3.org/TR/REC-PICS-labels-961031.

[L00a] Sandrine Lafois. *Interrogation de donnees structurees en abre Application a C-WEB*. Master Thesis, DEA Systems Informatiques Repatris, Paris VI, 2000.

[L00b] Jonas Liljegren. *Description of an rdf database implementation*. Available at http://WWW-DB.Stanford.EDU/~melnik/rdf/db-jonas.html.

[LCS99] Tim Berners-Lee, Dan Connolly, and Ralph R. Swick. *Web Architecture: Describing and Exchanging Data.* W3C Note, June 1999. Available at http://www.w3.org/1999/04/WebData.

[LFIM98] Tim Berners-Lee, R. Fielding, U.C. Irvine, and L. Masinter. *Uniform resource identifiers (uri): Generic syntax.* RFC 2396, August 1998. Available at http://www.ietf.org/rfc/rfc2396.txt.

[LLD96] C. Lagoze, C. A. Lynch, and R. Daniel. *The Warwick Framework: A Container Architecture for Aggregating Sets of Metadata*. June 2 1996. Available at

http://cs-tr.cs.cornell.edu:80/Dienst/UI/2.0/Describe/ncstrl.cornell/TR96-1593.

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

[M98] Eric Miller. *An Introduction to the Resource Description Framework*. D-Lib Magazine. May 1998. Available at http://www.dlib.org/dlib/may98/miller/05miller.html.

[M00a] Microsoft Corporation. *Using the Microsoft Repository*. 2000. Available at
http://www.sqlmag.com/Articles/Index.cfm?ArticleID=8029&Key=Data%20Warehousing.

[M00b] Sergey Melnik. *Storing RDF in a relational database*. Available at http://WWW-DB.Stanford.EDU/~melnik/rdf/db.html.

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

[NWC00] Wolfgang Nejdl, Martin Wolpers, and Christian Capelle. *The RDF Schema Specification Revisited*. In J. Ebert et al. (eds.), Modelle und Modellierungssprachen in Informatik und Wirtschaftsinformatik, Modellierung 2000, St. Goar, April 5-7, 2000, Foelbach Verlag, Koblenz, 2000.

[PICS] W3C PICS The Platform for Content Selection Home Page.

Available at http://www.w3.org/PICS.

[RDF APIs] GINF: http://www –db.stanford.edu/~melnik/rdf/api.html

RADIX: http://www.mailbase.ac.uk/lists/rdf-dev/1999-06/0002.html

Netscape Communicator: http://lxr.mozilla.org/seamonkey/source/rdf/base/idl

RDF for java: http://www.alphaworks.ibm.com/formula/rdfxml.

[RDS00] *RDF Data Store: triplestore*. http://www.desire.org/toolkit/RDFds.html http://www.bized.ac.uk/test/dan/2000/08/metaconf/rudolf-perl/Docs/rdfTripleStoreDoc.htm.

[SEMD00] Steffen Staab, Michael Erdmann, Alexander Maedche, and Stefan Decker. *An Extensible Approach for Modeling Ontologies in RDF(S)*. ECDL 2000 Workshop on the Semantic Web.Available at http://www.aifb.uni-karlsruhe.de/~sst/Research/Publications/onto-rdfs.pdf.

[S99a] Avi Saha. *Application Framework for e-business: Portals*. IBM Software Strategy, November 1999. Available at http://www-4.ibm.com/software/developer/library/portals/index.html.

[S99b] Eric Severson. *Enterprise Information Portals and XML*. IBM Corporation, July 1999. Available at http://edms.solutions.ibm.com/portal.pdf.

[SKWW00] Schmidt, M. Kersten, M. Windhouwer and Florian Waas. *EfficientRelational Storage and Retrieval of XML Documents. WebDB2000*. Available at http://dbms.uni-muenster.de/events/webdb2000/PAPERS/3c.ps.

[ST98] C. Shilakes and J.Tylman. *Enterprise Information Portals*. Merrill Lynch, Inc., New York, NY, November 16, 1998.

[STH+99] J. Shanmugasundaram, K. Tufte, G. He, C. Zhang, D. DeWitt, J. Naughton, *Relational Databases for Querying XML Documents: Limitations and Opportunities*. VLDB Conference, September 1999. Available at http://www.cs.wisc.edu/~jai/papers/RdbmsForXML.pdf

[SYU99] T. Shimura, M. Yoshikawa, and S. Uemura. *Storage and Retrieval of XML Documents Using Object-Relational Databases*. In Database and Expert Systems Applications, pages 206{217. Springer, 1999. Available at http://www.eecs.umich.edu/~krunapon/research/xml/refs/dexa99.pdf.

[T99] Daniel Tkach. *Knowledge Portals*. IBM 1999. Available at http://www-4.ibm.com/software/data/km/advances/kportals.html.

[TBMM00] H. Thompson, D. Beech, M. Maloney and N. Mendelsohn. *XML Schema Part 1: Structures*. W3C Candidate Recommendation, October 2000. Available at http://www.w3.org/TR/2000/CR-xmlschema-1-20001024/.

[TC97] M. Theodorakis and P. Constantopoulos. *On Context-based Naming in Information Bases*. Proc. 2nd IFCIS International Conference on Cooperative Information Systems (CoopIS'97), Charleston, South Carolina, USA, June 24-27, 1997 pp.140-149. Available at http://www.ics.forth.gr/~etheodor/ijcis97/html/paper.html

[TWCZ00] F. Tian, D. DeWitt, J. Chan and C. Zhang. *The Design and Performance Evaluation of Alternative XML Storage Strategies*. Technical Report, CS Dept., University of Wiscosin, 2000. Available at http://www.cs.wisc.edu/niagara/papers/vldb00XML.pdf.

[W99a] Sarah Roberts-Witt. *Making Sense of Portal Pandemonium.* 1999. Available at http://department.stthomas.edu/msejourn/V10-1/art-Roberts-Witt.htm.

[W99b] Fang Wei. *F-logic and Implementation of Internet Metadata*. Diploma Thesis, December 1999. Available at http://www.informatik.uni-freiburg.de/~fwei/.

[Wraf00] *Web Resource Application Framework*: Wraf. Available at http://www.uxn.nu/wraf/.