Γλώσσες Αναπαράστασης Γνώσης στο Σημασιολογικό Ιστό

Γιώργος Στοΐλος
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Η/Υ
Εθνικό Μετσόβιο Πολυτεχνείο

1. Αναπαράσταση Γνώσης στο Σημασιολογικό Ιστό

Ο Σημασιολογικός Ιστός (Semantic Web) αποτελεί μια επέκταση του σημερινού Ιστού. Η επέκταση αυτή έχει ως σκοπό την αυτοματοποίηση των λειτουργιών και των εφαρμογών του διαδικτύου, όπως είναι οι μηχανές αναζήτησης και οι πράκτορες (agents). Η αυτοματοποίηση αυτή μπορεί να επιτευχθεί μόνον εφόσον η γνώση και η πληροφορία που υπάρχει αποθηκευμένη και δημοσιευμένη αυτή τη στιγμή στο σημερινό Παγκόσμιο Ιστό αποκτήσει τυπικό νόημα (formal meaning) και σημασιολογία (semantics) και δομηθεί με ένα τέτοιο τρόπο ώστε να γίνεται κατανοητή από τις μηχανές που την επεξεργάζονται (machine understandable). Από τη στιγμή που η πληροφορία θα είναι δομημένη με έναν σημασιολογικά πλούσιο τρόπο θα ενισχύεται ο διαμοιρασμός (sharing) και η επαναχρησιμοποίησή της (reusability) επιτυγχάνοντας τη *διαλειτουργικότητα (interoperability)* και *συνδεσιμότητα (interconnectivity) ετερογενών (heterogeneous)* συστημάτων και των εφαρμογών.

Προκειμένου η γνώση και η πληροφορία να περιγραφεί με έναν τυπικό (formal) τρόπο ο οποίος θα δηλώνει τη σημασία της πρέπει να χρησιμοποιήσουμε γλώσσες αναπαράστασης γνώσης. Προκειμένου όμως να χρησιμοποιήσουμε τέτοιες τεχνολογίες στο διαδίκτυο θα πρέπει να αναθεωρήσουμε και να τροποποιήσουμε κάποια από τα συστατικά τους. Πιο συγκεκριμένα, όπως είναι γνωστό, στο σημερινό Ιστό κατά ένα πολύ μεγάλο ποσοστό η πληροφορία δομείται με τη χρήση της γλώσσας XML. Έτσι λοιπόν αφενός στο Σημασιολογικό Ιστό πρέπει να περιγράψουμε γνώση με τη χρήση κάποιας γλώσσας αναπαράστασης γνώσης αλλά αφετέρου η σύνταξη της γλώσσας που θα χρησιμοποιήσουμε θα πρέπει να βασίζεται στη γλώσσα XML. Για το λόγο αυτό η W3C, η οποία είναι ο οργανισμός που ασχολείται με την ανάπτυξη και προτυποποίηση τεχνολογιών για τον Παγκόσμιο Ιστό, έχει αναπτύξει δύο γλώσσες αναπαράστασης γνώσης. Οι γλώσσες αυτές είναι η RDF(S) και η OWL.

Για ποιο λόγο όμως χρειαζόμαστε δύο γλώσσες αναπαράστασης γνώσης; Η απάντηση στο ερώτημα αυτό δίνεται αν κοιτάξουμε την αρχιτεκτονική του Σημασιολογικού Ιστού. Στην αρχιτεκτονική αυτή παρατηρούμε ότι ο Σημασιολογικός Ιστός αποτελείται από στρώματα (layers). Κάθε στρώμα υλοποιεί μια λειτουργικότητα (functionality), χρησιμοποιώντας και επεκτείνοντας τη λειτουργικότητα και τις τεχνολογίες που παρέχονται από τα χαμηλότερα στρώμα. Έτσι λοιπόν στα χαμηλά επίπεδα υλοποιούνται λειτουργίες οι οποίες είναι πολύ κοντά στον Παγκόσμιο Ιστό και στις μηχανές, όπως είναι οι τεχνολογίες που ασχολούνται με τον καθορισμό και την αναγνώριση των πόρων (URIs), ενώ καθώς ανεβαίνουμε στην ιεραρχία των επιπέδων διατρέχουμε επίπεδα τα οποία υλοποιούν λειτουργικότητες αναπαράστασης γνώσης, πολύπλοκης συλλογιστικής και εμπιστοσύνης πλησιάζοντας στην ανθρώπινη γνώση και σκέψη. Θα ασχοληθούμε με τα ακόλουθα επίπεδα:

1. **Επίπεδο μεταδεδομένων:** Στο επίπεδο αυτό εισάγεται μια πολύ βασική και απλή γλώσσα αναπαράστασης γνώσης για τον Παγκόσμιο Ιστό. Το μοντέλο της γλώσσας αυτής προσφέρει ουσιαστικά μόνο τη δυνατότητα δημιουργίας ισχυρισμών (assertions) για τα στοιχεία του διαδικτύου. Οι έννοιες που εισάγονται είναι αυτές του *πόρου (resource)* και τι *ιδιότητας (property),* οι οποίες και χρησιμοποιούνται για την περιγραφή μετα-πληροφορίας χωρίς να περιγράφουν κάποια περίπλοκη γνώση. Για παράδειγμα μπορούμε να περιγράψουμε κάποιον πόρο αποδίδοντάς του μια ή περισσότερες ιδιότητες με

τις τιμές που κατέχει για τις ιδιότητες αυτές. Η γλώσσα που υλοποιεί το επίπεδο αυτό είναι η RDF (Resource Description Framework) (Lassila, O., Swick, R., 1999).

2. **Επίπεδο σχήματος:** Στο επίπεδο αυτό εισάγονται κάποια επιπλέον βασικά στοιχεία για την περιγραφή γνώσης στο Σημασιολογικό Ιστό. Πιο συγκεκριμένα εισάγονται για πρώτη φορά οι έννοιες της *κλάσης (class)* και της ιεραρχίας κλάσεων και ιδιοτήτων. Για να οριστούν αυτές οι έννοιες χρησιμοποιείται η λειτουργικότητα του επιπέδου μεταδεδομένων. Η γλώσσα η οποία υλοποιεί το επίπεδο αυτό είναι η γλώσσα RDF-S (RDF-Schema) (Brickey, D., Guha, R.V., 2000).

3. **Λογικό επίπεδο:** Στο επίπεδο αυτό υλοποιούνται περισσότερο εκφραστικές γλώσσες αναπαράστασης γνώσης. Οι γλώσσες αυτές χρησιμοποιούν και επεκτείνουν τη λειτουργικότητα του επιπέδου σχήματος παρέχοντας περισσότερες εκφραστικές δυνατότητες. Η γλώσσα η οποία υλοποιεί τη λειτουργικότητα του επιπέδου αυτού είναι η OWL (Bechhofer et. al., 2004).

4. **Επίπεδο κανόνων:** Στο επίπεδο αυτό η λειτουργικότητα των γλωσσών του λογικού επιπέδου επεκτείνεται ακόμη περισσότερο παρέχοντας τη δυνατότητα καταγραφής κανόνων.

Στη συνέχεια θα μελετήσουμε τη σύνταξη και τη σημασιολογία των γλωσσών αναπαράστασης γνώσης RDF, RDF-S και OWL.

2. Οι γλώσσες RDF και RDF-S

4.2.1 Περιγραφή Μεταδεδομένων με την RDF

Η RDF είναι μια γλώσσα η οποία χρησιμοποιείται για την απλή περιγραφή *πόρων (resources)* του διαδικτύου. Η περιγραφή αυτή εστιάζεται κυρίως στην απόδοση μετα-πληροφορίας για τις οντότητες αυτές, όπως είναι η περιγραφή του τίτλου, του ονόματος, της ημερομηνίας δημιουργίας και άλλων χαρακτηριστικών κάποιου πόρου του διαδικτύου. Με την έννοια πόρος αναφερόμαστε σε οποιαδήποτε οντότητα του Παγκόσμιου ιστού, όπως είναι μια ιστοσελίδα, ένα τμήμα ή ένα σύνολο από ιστοσελίδες, ηλεκτρονικά αρχεία ή ακόμα και αντικείμενα τα οποία δεν είναι άμεσα διαθέσιμα στο διαδίκτυο, όπως είναι για παράδειγμα ένα βιβλίο.

Η RDF βασίζεται στην ιδέα ότι οι πόροι οι οποίοι πρέπει να περιγραφούν έχουν *ιδιότητες (properties)* οι οποίες έχουν συγκεκριμένη τιμή. Έτσι λοιπόν μια μετα-πληροφορία για ένα πόρο αποτελείται από μια ιδιότητα και την τιμή που έχει ο πόρος για την ιδιότητα αυτή. Μια έκφραση για έναν πόρο ονομάζεται RDF *πρόταση (sentence)*. Συνοψίζοντας λοιπόν, μια RDF πρόταση αποτελείται από μια *τριάδα (triple)* ενός *υποκειμένου (subject)*, *ιδιότητας (property)* και *αντικειμένου (object)*. Τη θέση του υποκειμένου καταλαμβάνει ο πόρος, τη θέση της ιδιότητας η ιδιότητα που του αποδίδουμε, ενώ τη θέση του αντικειμένου η τιμή που έχει ο πόρος αυτός για την ιδιότητα. Η τιμή αυτή μπορεί να είναι κάποιος άλλος πόρος ή κάποια τιμή δεδομένων. Συντακτικά οι προτάσεις αυτές δηλώνονται από μια διατεταγμένη τριάδα της μορφής, s p o. όπου τα s, p και ο αντιπροσωπεύουν το υποκείμενο, την ιδιότητα και το αντικείμενο, αντίστοιχα ενώ η τριάδα τερματίζεται με το σύμβολο της τελείας. Ένα σύνολο από τριάδες RDF μπορούμε να το αντιληφθούμε και ως έναν γράφο. Σε αυτόν τον γράφο τα αντικείμενα και τα υποκείμενα παίζουν το ρόλο των κόμβων ενώ οι ιδιότητες παίζουν το ρόλο των ακμών τους συνδέουν. Όπως αναφέραμε και στην εισαγωγή η RDF είναι μια γλώσσα αναπαράστασης γνώσης για το Σημασιολογικό Ιστό. Έτσι λοιπόν το πρότυπο της RDF καθορίζει και μια σύνταξη η οποία έχει σαν σκοπό οι RDF τριάδες να δομούνται με έναν τρόπο επεξεργάσιμο από υπολογιστικά συστήματα και εφαρμογές. Η σύνταξη αυτή δε θα μπορούσε να

βασίζεται σε άλλο πρότυπο παρά μόνο στην XML. Η σύνταξη αυτή αναφέρεται ως RDF/XML σύνταξη (Beckett 2003). Για λόγους γενικότητας η RDF χρησιμοποιεί URI references για να προσδιορίσει τις οντότητες οι οποίες βρίσκονται στη θέση του υποκειμένου, της ιδιότητας και του αντικειμένου. Ένα URI reference (URIRef) αποτελείται από ένα URI και από ένα προαιρετικό fragment identifier. Για παράδειγμα το URIRef http://www.example.org/index.html#section2 αποτελείται από το URI http://www.example.org/index.html και από τον fragment identifier Section2 τον οποίο διακρίνουμε από το URI με τη χρήση του συμβόλου #.

Ας δούμε τώρα πως μπορούμε να ορίσουμε μετα-πληροφορία σε διάφορους πόρους χρησιμοποιώντας τη γλώσσα RDF. Έστω ότι θέλουμε να περιγράψουμε τη γνώση ότι η ιστοσελίδα http://www.example.org/index.html έχει δημιουργό (creator) τον John Smith. Στην πρόταση αυτή μπορούμε να διακρίνουμε ένα υποκείμενο, τη σελίδα, ένα αντικείμενο τον John Smith και μια ιδιότητα, αυτή του δημιουργού της ιστοσελίδας. Αυτή η μετα-πληροφορία μπορεί να περιγραφεί σε μια τριάδα RDF ως εξής:

<http://.../index.html> <http://.../creator> <http://www.example.org/85730>.

όπου 85730 είναι ένα αντικείμενο που αντιπροσωπεύει τον John Smith. Παρατηρούμε ότι η RDF χρησιμοποιεί URIrefs όχι μόνο για να προσδιορίσει τον πότο τον οποίο και περιγράφουμε αλλά και την ιδιότητα και το αντικείμενο. Όπως γίνεται αντιληπτό ίσως είναι λίγο άβολο να γράφουμε συνέχεια ολόκληρα τα URI των οντοτήτων που χρησιμοποιούμε. Έτσι λοιπόν πολλές φορές χρησιμοποιούνται συντομεύσεις σε μορφή προθεμάτων (prefixes) τα οποία ονομάζονται QNames. Θα χρησιμοποιήσουμε τα QNames, rdf, rdfs, dc, ex, exterms και exdateid για να αναπαραστήσουμε τα URIs http://www.w3.org/1999/02/22-rdf-syntax-ns, http://www.w3.org/2000/01/ rdf-schema, http://purl.org/dc/elements/1.1/, http://www.example.org/, http://www.example.org/terms/, και http://www.example.org/date/, αντίστοιχα. Έτσι λοιπόν η παραπάνω τριάδα μπορεί να γραφτεί ως, ex:index.html dc:creator exstaff:85730.

Όπως είναι φυσικό αν επιθυμούμε μπορούμε να προσδώσουμε επιπλέον μετα-πληροφορίες για στον αρχικό μας πόρο. Για παράδειγμα μπορούμε να πούμε ότι η ημερομηνία δημιουργίας (creation-date) της ιστοσελίδας ήταν η 16η Αυγούστου του 1999 ή ότι η γλώσσα στην οποία είναι γραμμένη είναι τα Αγγλικά. Οι πληροφορίες αυτές μπορούν να περιγραφούν με τις ακόλουθες τριάδες:

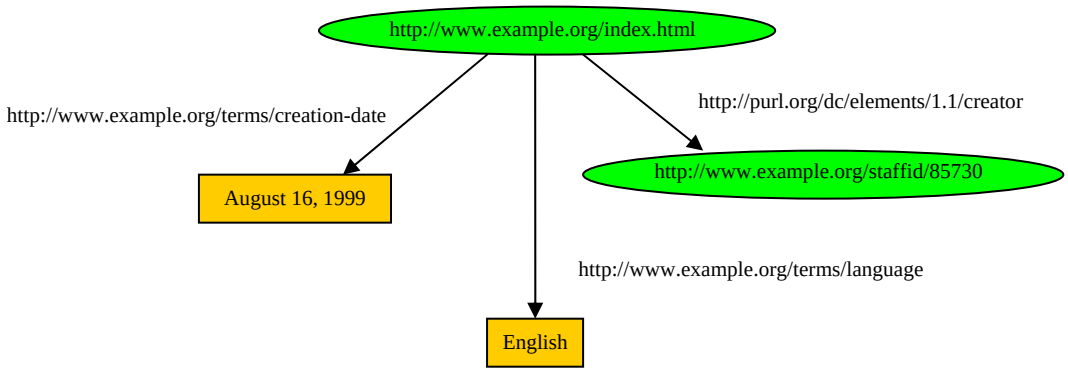
ex:index.html exterms:creation-date "August 16, 1999".
ex:index.html exterms:language "English".

Όπως αναφέραμε προηγουμένως οι τριάδες RDF μπορούν να αναπαρασταθούν σε μια μορφή γράφου. Για το παραπάνω σύνολο τριάδων που περιγράψαμε ο RDF γράφος που δημιουργείται φαίνεται στο Σχήμα 1.

Από το γράφο αυτόν φαίνεται ότι το αντικείμενο σε μια RDF πρόταση μπορεί να είναι είτε ένα URIref είτε σταθερές τιμές οι οποίες και ονομάζονται *λεκτικά (literals)*. Είναι σημαντικό να τονίσουμε ότι ένα λεκτικό απαγορεύεται να εμφανίζεται στη θέση του υποκειμένου ή της ιδιότητας. Ας δούμε τώρα τι μορφή έχει η γνώση μας αυτή σε RDF/XML σύνταξη. Η RDF/XML μορφή των παραπάνω προτάσεων είναι η ακόλουθη:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:exterms="http://www.example.org/terms/">
        <rdf:Description rdf:about="http://www.example.org/index.html">
                <exterms:creation-date>August 16, 1999</exterms:creation-date>
```

```
            <exterms:language>English</exterms:language>
            <dc:creator rdf:resource="http://www.example.org/staffid/85730"/>
        </rdf:Description>
</rdf:RDF>
```



Σχήμα 1 *Γράφος RDF προτάσεων.*

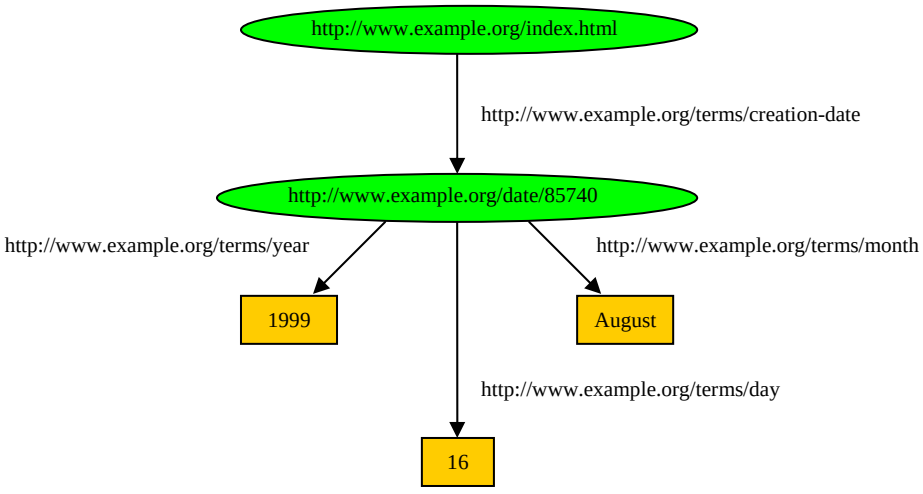
Ας εξηγήσουμε το παραπάνω RDF/XML κείμενο. Αρχικά παρατηρούμε ότι όπως σε όλα τα XML κείμενα έτσι και σε αυτό εισάγουμε τη δήλωση <?xml version="1.0"?> πριν από κάθε άλλο ορισμό. Εν συνεχεία το στοιχείο rdf:RDF δηλώνει ότι αυτό που ακολουθεί είναι ένα σύνολο από RDF προτάσεις. Μέσα στο στοιχείο αυτό δηλώνουμε επίσης και τις συντομεύσεις που θα χρησιμοποιήσουμε για τα URIrefs. Πιο συγκεκριμένα η συντόμευση rdf δηλώνει ότι οποιαδήποτε στοιχείο έχει ως πρόθεμα το rdf: αναφέρεται στο *χώρο ονομάτων (namespace)* που καθορίζεται από το URI http://www.w3.org/1999/02/22-rdf-syntax-ns#. Αντίστοιχα έχουμε ορίσει τα προθέματα dc και exterms. Στη συνέχεια ακολουθούν οι RDF/XML προτάσεις που δηλώνουν τις ματα-πληροφορίες που αναπαριστώνται από το γράφο του Σχήματος 1. Πιο συγκεκριμένα το στοιχείο rdf:Description υποδηλώνει την αρχή της περιγραφής ενός πόρου. Η δήλωση αυτή συνεχίζει περιγράφοντας σε ποιόν πόρο αναφερόμαστε χρησιμοποιώντας το στοιχείο rdf:about και το URIref του πόρου. Ο πόρος αυτός αντιστοιχεί στο υποκείμενο μιας τριάδας.

Στη συνέχεια περιγράφουμε τις ιδιότητες και τις τιμές τις οποίες ο πόρος αυτός έχει για τις ιδιότητες αυτές. Πιο συγκεκριμένα στην έκτη γραμμή χρησιμοποιήσαμε ως *ετικέτα (tag)* το στοιχείο exterms:create-date για να δηλώσουμε την ιδιότητα που κατέχει ο πόρος, ενώ ως περιεχόμενο της ετικέτας, δηλαδή αυτό το οποίο βρίσκεται ανάμεσα στην ετικέτα εκκίνησης (start-tag) <exterms:creation-date> και την ετικέτα τερματισμού (end-tag) </exterms:creation-date>, αποτελεί και την τιμή του πόρου για την ιδιότητα αυτή, δηλαδή το αντικείμενο της τριάδας. Με τον ίδιο ακριβώς τρόπο εργαζόμαστε και για την ιδιότητα που περιγράφει τη γλώσσα της ιστοσελίδας, όμως όπως παρατηρούμε η σύνταξη της τρίτης ιδιότητας είναι διαφορετική. Αυτό συμβαίνει γιατί η τιμή που είχαν οι προηγούμενες δύο ιδιότητες ήταν λεκτικά ενώ στην τρίτη ιδιότητα η τιμή είναι ένας άλλος πόρος. Για να ξεχωρίσουμε τις δύο αυτές περιπτώσεις, όπως είναι φυσικό χρησιμοποιούμε διαφορετική σύνταξη. Έτσι λοιπόν, στην περίπτωση αυτή χρησιμοποιούμε το QName dc:creator για να δηλώσουμε το όνομα της ιδιότητας όμως επιπλέον χρησιμοποιούμε το στοιχείο rdf:resource που παρέχει η γλώσσα RDF για να μπορέσουμε να δηλώσουμε το URIref του πόρου με τον οποίο συνδέετε ο πόρος τον οποίο και περιγράφουμε. Είναι σημαντικό στο σημείο αυτό να σημειώσουμε ότι ενώ μπορούμε να χρησιμοποιήσουμε τα QNames για την καταγραφή του ονόματος μιας ετικέτας, η χρήση τους στις τιμές των στοιχείων απαγορεύεται από το πρότυπο της RDF/XML (Beckett 2003). Το σύνολο των στοιχείων το οποίο παρέχει η γλώσσα RDF για την περιγραφή γνώσης αποτελεί το *λεξιλόγιο (vocabulary)* της γλώσσας.

Από το προηγούμενο παράδειγμα παρατηρούμε ότι η χρήση πόρων στη θέση των αντικειμένων μας δίνει τη δυνατότητα να δημιουργήσουμε αλυσίδες από τριάδες RDF όπου το αντικείμενο της μιας τριάδας εμφανίζεται ως υποκείμενο της άλλης. Αυτό το χαρακτηριστικό είναι αρκετά χρήσιμο καθώς μας δίνεται η δυνατότητα οι τιμές που αποδίδουμε στην ιδιότητα ενός πόρου να έχουν και αυτές με τη σειρά τους κάποια περιγραφή και άρα να αναπαριστούν μια περίπλοκη δομή. Για παράδειγμα στις προηγούμενες τριάδες αποδώσαμε ως τιμή της ιδιότητας ημερομηνία δημιουργίας το λεκτικό August 16, 1999. Ίσως μια καλύτερη προσέγγιση να ήταν η ημερομηνία αυτή να αναπαρασταθεί με κάποιο πόρο, ο οποίος θα ονομάζεται ημερομηνία και ο οποίος θα έχει τις ιδιότητες, μήνας, ημέρα και έτος οι οποίες στη συγκεκριμένη περίπτωση θα έχουν τις τιμές August, 16 και 1999, αντίστοιχα. Έτσι λοιπόν μπορούμε να κάνουμε τις παρακάτω δηλώσεις,

ex:index.html	exterms:creation-date	exdateid:85740.
exdateid:85740	exterms:month	August.
exdateid:85740	exterms:day	16.
exdateid:85740	exterms:year	1999.

Η μορφή του RDF γράφου για τις παραπάνω δηλώσεις φαίνεται στο Σχήμα 2. Αυτός ο τρόπος γραφής όμως συνεπάγεται ότι οποτεδήποτε χρειαζόμαστε να αναφερθούμε σε μια ενδιάμεση οντότητα, όπως είναι η ημερομηνία, πρέπει να εισάγουμε έναν καινούργιο πόρο για την οντότητα αυτή και να του αποδώσουμε κάποιο URIref. Στην περίπτωση που ο πόρος αυτός χρησιμοποιείται μόνο μια φορά χωρίς να κατέχει κάποια παραπάνω χρησιμότητα στη γνώση μας ίσως να μην επιθυμούμε να εισάγουμε ρητά τον πόρο αυτόν αποδίδοντάς του κάποιο URIref. Σε αυτήν την περίπτωση η RDF μας παρέχει την έννοια του *κενού κόμβου (blank node)* (Klyne, G., Carroll, J. 2003). Διαισθητικά η ύπαρξη ενός κενού κόμβου σε μια θέση υποδηλώνει την παρουσία κάποιας οντότητας στη θέση αυτή χωρίς όμως να δηλώνεται ρητά, μέσω κάποιου URIref, ποια είναι αυτή η οντότητα. Γραφικά ένας κενός κόμβος αναπαριστάται χρησιμοποιώντας το οβάλ σχήμα που χρησιμοποιούμε για τους πόρους, χωρίς όμως να αναγράφουμε κάποιο URIref.



Σχήμα 2 *Δημιουργία της δομής της ημερομηνίας.*

Όπως είναι προφανές ένα πολύπλοκο RDF κείμενο μπορεί να περιέχει πολλούς κενούς κόμβους. Σε αυτήν την περίπτωση είναι απαραίτητο να διακρίνουμε του κενούς κόμβους μεταξύ τους. Γι αυτό το λόγο και η RDF χρησιμοποιεί τους λεγόμενους *προσδιοριστές κενών κόμβων (blank node identifiers)*. Οι προσδιοριστές

κενών κόμβων έχουν συντακτικά τη μορφή _:name, η οποία υποδηλώνει την ύπαρξη ενός κενού κόμβου και ταυτόχρονα δίνει και ένα όνομα στον κόμβο αυτό. Για παράδειγμα, αν χρησιμοποιήσουμε κενό κόμβο για την αναπαράσταση της ημερομηνίας οι τριάδες μας θα έχουν την ακόλουθη μορφή,

ex:index.html	externs:creation-date	_:pagedate.
_:pagedate	externs:month	August.
_:pagedate	externs:day	16.
_:pagedate	externs:year	1999.

ενώ η RDF/XML μορφή των τριάδων αυτών είναι η ακόλουθη,

```
<rdf:Description rdf:about="http://www.example.org/index.html">
        <externs:creation-date rdf:parseType="Resource">
                <externs:month>August</externs:month>
                <externs:day>16</externs:day>
                <externs:year>1999</externs:year>
        </externs:creation-date>
</rdf:Description>
```

Η δήλωση rdf:parseType="Resource" χρησιμοποιείται για να υποδείξει ότι τα περιεχόμενα ενός στοιχείου πρέπει να ερμηνευτούν ως η περιγραφή ενός νέου κενού κόμβου. Η περιγραφή αυτού του κενού κόμβου γίνεται χωρίς να απαιτείται η δημιουργία μιας RDF περιγραφής με τη χρήση του στοιχείου rdf:Description, όπως είδαμε στο προηγούμενο παράδειγμα. Έτσι λοιπόν η δήλωση rdf:parseType="Resource" μας λέει ότι η τιμή της ιδιότητας externs:creation-date είναι ένας νέος κενός κόμβος και ότι οτιδήποτε ακολουθεί είναι οι τιμές του κενού κόμβου στις ιδιότητες που περιγράφονται. Αξίζει να αναφέρουμε στο σημείο αυτό ότι υπάρχουν και άλλοι τρόποι να αναπαραστήσουμε κενούς κόμβους στη σύνταξη RDF/XML. Πιο συγκεκριμένα μπορούμε να γράψουμε,

```
<rdf:Description rdf:about="http://www.example.org/index.html">
        <externs:creation-date rdf:nodeID="abc"/>
</rdf:Description>
<rdf:Description rdf:nodeID="abc">
                <externs:month>August</externs:month>
                <externs:day>16</externs:day>
                <externs:year>1999</externs:year>

</rdf:Description>
```

όπου το στοιχείο rdf:nodeID χρησιμοποιείται για να υποδείξει ότι αυτό που ακολουθεί είναι το όνομα ενός κενού κόμβου. Η μέθοδος αυτή έχει το πλεονέκτημα ότι αναθέτοντας ονόματα στους κενούς κόμβους μπορούμε να τους χρησιμοποιήσουμε και στις περιγραφές άλλων πόρων.

Στο προηγούμενο παράδειγμα είδαμε πως μπορούμε να δημιουργήσουμε τη δομή ημερομηνία από τα επιμέρους στοιχεία της ημέρας, του μήνα και του έτους. Όταν όμως δώσαμε τιμή στην ημέρα και στο μήνα χρησιμοποιήσαμε και για τις δύο αυτές ιδιότητες ένα λεκτικό το οποίο είναι ουσιαστικά μια συμβολοσειρά. Έτσι λοιπόν η τιμή "16" δεν αναπαριστά τον αριθμό δεκαέξι αλλά μια συμβολοσειρά της οποίας ο πρώτος χαρακτήρας είναι ο "1" και ο δεύτερος ο "6". Αυτού του είδους τα λεκτικά στην ορολογία της RDF ονομάζονται *απλά λεκτικά (plain literals)*. Ενώ στην περίπτωση του μήνα κάτι τέτοιο είναι αποδεκτό στην περίπτωση της ημέρας και του χρόνου θα ήταν φυσικό κάποιος να θέλει να αποδώσει ως τιμή των ιδιοτήτων αυτών

έναν αριθμό παρά μια συμβολοσειρά. Η κοινή πρακτική στις γλώσσες προγραμματισμού και στις βάσεις δεδομένων είναι η παροχή επιπρόσθετης πληροφορίας η οποία δηλώνει το πως πρέπει η τιμή που καθορίζουμε να ερμηνευτεί. Η πληροφορία αυτή δεν είναι άλλη παρά ο *τύπος δεδομένων (datatype)*, όπως για παράδειγμα οι τύποι δεδομένων δεκαδικός (decimal) και ακέραιος (integer). Τότε μια εφαρμογή η οποία καταλαβαίνει τους τύπους δεδομένων θα μπορεί να χειριστεί και την κάθε τιμή με τον κατάλληλο τρόπο. Για την παροχή της πληροφορίας αυτής η RDF παρέχει την έννοια των *δακτυλογραφημένων λεκτικών (typed literals)*.

Ένα *δακτυλογραφημένο λεκτικό* ορίζεται ως ένα ζεύγος μιας συμβολοσειράς και ενός URIref το οποίο προσδιορίζει τον τύπο δεδομένων της οντότητας που περιγράφεται από τη συμβολοσειρά. Για παράδειγμα χρησιμοποιώντας δακτυλογραφημένα λεκτικά μπορούμε να περιγράψουμε την ημερομηνία ως, _:pagedate exterms:year "1999"^^xsd:integer, όπου xsd είναι συντόμευση για το URI http://www.w3.org/2001/XMLSchema#. Βλέπουμε ότι χρησιμοποιούμε τη συμβολοσειρά «1999» για την περιγραφή του έτους, όμως επισυνάπτουμε επιπλέον την πληροφορία ότι αυτό το οποίο περιγράφεται από τη συμβολοσειρά είναι ένας ακέραιος. Από την άλλη χρησιμοποιώντας τύπους δεδομένων θα μπορούσαμε να μη δημιουργήσουμε έναν κενό κόμβο που να αναπαριστά την ημερομηνία αλλά θα μπορούσαμε να γράψουμε την τριάδα, ex:index.html exterms:creation-date "1999-08-16"^^xsd:date. Η τελευταία τριάδα σε RDF/XML σύνταξη γράφεται ως εξής,

```
<rdf:Description rdf:about="http://www.example.org/index.html">
        <exterms:creation-date rdf:datatype=
                "http://www.w3.org/2001/XMLSchema#date">1999-08-16
        </exterms:creation-date>
</rdf:Description>
```

Παρατηρούμε ότι χρησιμοποιούμε το στοιχείο rdf:datatype για να προσδιορίσουμε τον τύπο δεδομένων που πρέπει να χρησιμοποιηθεί για να ερμηνεύσουμε την τιμή που ακολουθεί. Όπως αναφέραμε προηγουμένως η χρήση συντομεύσεων στις τιμές ενός στοιχείου απαγορεύεται. Στην περίπτωση όμως των τύπων δεδομένων της XML μπορούμε να χρησιμοποιήσουμε τα στοιχεία *XML* (XML entities) για να βελτιώσουμε την αναγνωσιμότητα των κειμένων μας. Ο τρόπος για να το πετύχουμε αυτό είναι ο ακόλουθος. Αρχικά ορίζουμε τη συντόμευση xsd ως, <!DOCTYPE rdf:RDF [<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">]>. Στη συνέχεια μπορούμε να χρησιμοποιήσουμε τη συντόμευση αυτή γράφοντας &xsd;date αντί για ολόκληρο το URIref.

Μέχρι στιγμής είδαμε πως να χρησιμοποιούμε τις τριάδες RDF για να αποδίδουμε τιμές σε συγκεκριμένες ιδιότητες που κατέχει κάποιος πόρος του διαδικτύου. Αν θέλαμε να κάνουμε μια αντιστοίχηση με τις Περιγραφικές Λογικές, θα λέγαμε ότι μέχρι στιγμής έχουμε περιγράψει μόνο πως μπορούμε να δημιουργήσουμε ισχυρισμούς ρόλων. Πιο συγκεκριμένα μια τριάδα s p o. μπορεί να θεωρηθεί ως ένας ισχυρισμός της μορφής p(s,o), ο οποίος μας λέει ότι το άτομο s έχει την τιμή ο στην ιδιότητα (ρόλο) p. Το ερώτημα που τίθεται, λοιπόν, είναι αν η RDF προσφέρει κάποιο μηχανισμό για την περιγραφή σχέσεων στιγμιοτύπου ανάμεσα σε κάποιον πόρο και σε κάποια έννοια (κλάση). Η λειτουργικότητα αυτή παρέχεται στην RDF μέσω του στοιχείου rdf:type. Όταν χρησιμοποιούμε το στοιχείο rdf:type στη θέση της ιδιότητας μιας τριάδας RDF ουσιαστικά δηλώνουμε ότι αυτό το οποίο ακολουθεί στη θέση του αντικειμένου αποτελεί μια κλάση της οποίας μέλος αποτελεί ο πόρος που βρίσκεται στη θέση του υποκειμένου. Έτσι λοιπόν μπορούμε να πούμε ότι ο John Smith είναι κάποιος άνθρωπος γράφοντας την τριάδα, ex:85730 rdf:type exterms:Person. Η RDF/XML σύνταξη της τριάδα αυτής είναι ίδια με όσες έχουμε δει μέχρι στιγμής με τη μόνη διαφορά ότι στη θέση της ιδιότητας πρέπει να αναγραφεί το στοιχείο rdf:type. Το στοιχείο rdf:type όμως έχει και έναν εναλλακτικό

πιο συνοπτικό τρόπο σύνταξης στη μορφή RDF/XML. Αυτός δημιουργείται αφαιρώντας το στοιχείο rdf:type και αντικαθιστώντας το rdf:Description με το QName της κλάσης. Χρησιμοποιώντας αυτόν τον τρόπο γραφής στο προηγούμενο παράδειγμα έχουμε την ακόλουθη σύνταξη,

<exterms:Person rdf:about="ex:85730"/>

Όπως αναφέραμε και στο παράδειγμα της δημιουργίας της δομής της ημερομηνίας ένας πόρος που βρίσκεται στη θέση του αντικειμένου μπορεί σε μια άλλη τριάδα να εμφανιστεί στη θέση του αντικειμένου και έτσι μπορούμε να δημιουργήσουμε αλυσίδες από τριάδες RDF. Έτσι λοιπόν είναι καθ' όλα νόμιμο κάποιος να γράψει τις τριάδες ex:Harry rdf:type ex:Eagle και ex:Eagle rdf:type ex:EndangeredSpeciesList. Η πρώτη τριάδα μας δηλώνει ότι ο Harry είναι στιγμιότυπο της κλάσης των αετών, δηλαδή μέλος του συνόλου των αετών, ενώ η δεύτερη τριάδα μας δηλώνει ότι η κλάση των αετών είναι στιγμιότυπο της κλάσης που αναπαριστά τη λίστα των ειδών που απειλούνται προς εξαφάνιση. Αυτό σημαίνει ότι ο πόρος που αναπαριστά τους αετούς επενεργεί ταυτόχρονα και ως κλάση (έννοια) αλλά και ως άτομο. Η δυνατότητα δημιουργίας τέτοιων εκφράσεων ονομάζεται *μετα-μοντελοποίηση (metamodeling)* διότι η κλάση EndangeredSpeciesList είναι ουσιαστικά μια κλάση που επενεργεί πάνω σε μια άλλη κλάση, δηλαδή μια μετα-κλάση. Η RDF όμως προχωράει ένα βήμα παραπάνω επιτρέποντας σε έναν πόρο να χρησιμοποιηθεί σε οποιαδήποτε θέση μιας τριάδας. Προφανώς αυτή η λειτουργικότητα δεν προσφέρεται από τις ΠΛ, οι οποίες διαχωρίζουν αυστηρά το σύνολο των ατόμων, των κλάσεων (εννοιών) και το σύνολο των ιδιοτήτων (ρόλων).

Σε μερικές εφαρμογές χρειαζόμαστε τη δυνατότητα να μπορούμε να δημιουργούμε προτάσεις οι οποίες αναφέρονται σε άλλες προτάσεις RDF. Έτσι για παράδειγμα μπορούμε να περιγράψουμε το πότε δημιουργήθηκε κάποια τριάδα ή το ποιος τη δημιούργησε. Το λεξιλόγιο της RDF παρέχει τα στοιχεία για τη δημιουργία τέτοιου είδους προτάσεων. Τα στοιχεία αυτά είναι τα rdf:Statement, rdf:object, rdf:subject και rdf:predicate. Η διαδικασία δημιουργίας προτάσεων με τη χρήση του λεξιλογίου αυτού ονομάζεται *reification*. Χρησιμοποιώντας το λεξιλόγιο αυτό και εφαρμόζοντας τη διαδικασία του reification στην τριάδα ex:index.html exterms:creation-date exdateid:85740. προκύπτουν οι ακόλουθες προτάσεις RDF,

ex:triple100	rdf:type	rdf:Statement
ex:triple100	rdf:subject	ex:index.html
ex:triple100	rdf:predicate	exterms:creation-date
ex:triple100	rdf:object	exdateid:85740

Μπορούμε επίσης να περιγράψουμε πληροφορία όπως για παράδειγμα το ποιος δημιούργησε την πρόταση αυτή γράφοντας ex:triple100 dc:creator exstaff:85730. Η μορφή που έχουν οι προτάσεις αυτές σε RDF/XML σύνταξη είναι η ακόλουθη,

```
<rdf:Statement rdf:about="#triple100">
  <rdf:subject rdf:resource=" http://www.example.org/index.html"/>
  <rdf:predicate rdf:resource="http://www.example.com/terms/creation-date"/>
  <rdf:object rdf:resource=" http://www.example.org/85740" />
</rdf:Statement>
```

Ολοκληρώνοντας την παρουσίασή μας στην RDF αξίζει να σημειώσουμε ότι η RDF παρέχει επιπλέον στοιχεία για την περιγραφή συνόλων (bags), ακολουθιών (sequences), εναλλακτικών επιλογών (alternatives) και συλλογών (collections)

πόρων. Το RDF λεξιλόγιο που παρέχεται για τον ορισμό αυτών των δομών είναι τα στοιχεία, rdf:bag, rdf:seq και rdf:alt για τη δημιουργία συνόλων, ακολουθιών και εναλλακτικών επιλογών, αντίστοιχα και τα στοιχεία, rdf:first, rdf:rest και rdf:nil για τη δημιουργία συλλογών. Για παράδειγμα μπορούμε να ορίσουμε ένα σύνολο από φοιτητές χρησιμοποιώντας τον παρακάτω τμήμα δήλωσης RDF/XML,

```
<rdf:Description rdf:about="http://example.org/courses/6.001">
        <s:students>
                <rdf:Bag>
                <rdf:li rdf:resource="http://example.org/students/Amy"/>
                <rdf:li rdf:resource="http://example.org/students/Mohamed"/>
                </rdf:Bag>
        </s:students>
</rdf:Description>
```

Όπως παρατηρούμε από την παραπάνω σύνταξη εφόσον δε χρησιμοποιούμε κάποιο URIref στον ορισμό του συνόλου (Bag) τότε αυτό αναπαριστά έναν κενό κόμβο. Έτσι λοιπόν το στοιχείο s:students είναι μια ιδιότητα η οποία ως τιμή έχει έναν κενό κόμβο αυτόν που ορίζεται από το σύνολο. Ο αναγνώστης ο οποίος επιθυμεί περισσότερες πληροφορίες για τις δομές αυτές παραπέμπεται στο (Beckett 2003).

4.2.2 Απλή Αναπαράσταση με τη Γλώσσα RDF-S

Η RDF μας παρέχει τη δυνατότητα να δημιουργήσουμε απλές προτάσεις για τους πόρους τους οποίους θέλουμε να περιγράψουμε χρησιμοποιώντας ιδιότητες, τιμές και URIref για τον προσδιορισμό των συστατικών που συμμετέχουν σε μια πρόταση. Η RDF όμως δεν παρέχει δυνατότητα να ορίσουμε και να περιγράψουμε ένα επιπλέον λεξιλόγιο το οποίο πιθανόν να επιθυμούμε να χρησιμοποιήσουμε στις εφαρμογές μας. Πιο συγκεκριμένα δεν έχουμε τη δυνατότητα να ορίσουμε τις κλάσεις (έννοιες) οι οποίες εμφανίζονται σε μια εφαρμογή. Επιπρόσθετα, είναι φυσικό να επιθυμούμε την περιγραφή των κλάσεων και των ιδιοτήτων μας δηλώνοντας σχέσεις υπαγωγής ανάμεσά τους. Η γλώσσα η οποία παρέχει τη λειτουργικότητα αυτή είναι η RDF-S (Brickey, D., Guha, R.V., 2000). Ουσιαστικά η RDF-S παρέχει ένα επιπλέον λεξιλόγιο πάνω σε αυτό της RDF το οποίο περιλαμβάνει στοιχεία τα οποία προορίζονται στο να προσδώσουν την επιπρόσθετη αυτή λειτουργικότητα.

Η δυνατότητα ορισμού κλάσεων υλοποιείται μέσω του στοιχείου rdfs:Class του λεξιλογίου της RDF-S. Όπως είδαμε στην προηγούμενη ενότητα προκειμένου να δηλώσουμε ότι ένας πόρος είναι στιγμιότυπο μιας κλάσης χρησιμοποιήσαμε το στοιχείο rdf:type. Έτσι λοιπόν προκειμένου να δηλώσουμε μια νέα κλάση, έστω την ex:MotorVehicle, αυτό που έχουμε να κάνουμε είναι να δημιουργήσουμε την τριάδα ex:MotorVehicle rdf:type rdfs:Class, στην οποία η συντόμευση ex: αναπαριστά το URIref http://www.example.org/schemas/vehicles. Εν συνεχεία μπορούμε να δηλώσουμε ότι ο πόρος ex:Audi είναι μέλος της κλάσης ex:MotorVehicle γράφοντας, ex:Audi rdf:type ex:MotorVehicle.

Με αυτόν τον τρόπο μπορούμε να δημιουργήσουμε το δικό μας λεξιλόγιο, δηλαδή να ορίσουμε το σύνολο των κλάσεων που εμφανίζονται σε μια εφαρμογή μας. Για παράδειγμα μπορούμε επιπρόσθετα να κάνουμε τις δηλώσεις ex:Van rdf:type rdfs:Class. και ex:Truck rdf:type rdfs:Class. Παρατηρείστε ότι προκειμένου να δηλώσουμε μια κλάση πρέπει να κάνουμε χρήση της μετα-μοντελοποίησης. Αυτό γιατί η κλάση ex:Van ορίζεται ως στιγμιότυπο της μετα-κλάσης rdfs:Class. Αλλά ακόμα περισσότερο εφόσον το στοιχείο rdfs:Class είναι προφανώς και αυτό μια κλάση πρέπει να ισχύει, και όντως σε κάθε RDF-S αρχείο ισχύει, η τριάδα rdfs:Class rdf:type rdfs:Class (Hayes 2003)!

Από τη στιγμή που έχουμε ορίσει τις κλάσεις μας μπορεί να μας ενδιαφέρει να ορίσουμε σχέσεις υπαγωγής ανάμεσα σε αυτές. Στην RDF-S αυτό επιτυγχάνεται με τη χρήση του στοιχείου rdfs:subClassOf. Έτσι λοιπόν μπορούμε να δημιουργήσουμε τις ακόλουθες προτάσεις,

ex:Van rdfs:subClassOf ex:MotorVehicle

ex:Truck rdfs:subClassOf ex:MotorVehicle

Η RDF/XML σύνταξη των παραπάνω ορισμών είναι η ακόλουθη,

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xml:base="http://example.org/schemas/vehicles">

<rdfs:Class rdf:ID="MotorVehicle"/>

<rdfs:Class rdf:ID="Truck">
  <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdfs:Class>

<rdfs:Class rdf:ID="Van">
  <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdfs:Class>
```

Παρατηρείστε ότι στο παραπάνω παράδειγμα χρησιμοποιήσαμε την περιεκτική γραφή του στοιχείου rdf:type. Αυτό μπορούμε να το κάνουμε γιατί όπως είπαμε το στοιχείο rdfs:Class είναι και αυτό μια κλάση. Έτσι λοιπόν μπορούμε να αντικαταστήσουμε το στοιχείο rdf:Description με αυτό. Τέλος παρατηρούμε ότι χρησιμοποιήσαμε ένα νέο στοιχείο το οποίο δεν έχουμε συναντήσει μέχρι στιγμής. Το στοιχείο αυτό είναι το rdf:ID. Αυτό μας επιτρέπει να μειώσουμε τα ονόματα γράφοντας μόνο τους fragment identifiers. Το URI το οποίο μπαίνει ως πρόθεμα στον fragment identifier για να δημιουργηθεί το URIref του πόρου είναι το URI του αρχείου στο οποίο βρίσκονται οι δηλώσεις ή το URI που ορίζεται από το xml:base.

Εκτός από τη δυνατότητα περιγραφής κλάσεων, μέσω σχέσεων υπαγωγής, η RDF-S παρέχει επιπλέον τη δυνατότητα δημιουργίας περιγραφών για τις ιδιότητες που επιθυμούμε να χρησιμοποιήσουμε σε μια εφαρμογή μας. Πιο συγκεκριμένα μπορούμε να περιγράψουμε σχέσεις υπαγωγής ανάμεσα σε δύο ιδιότητες, αλλά επιπρόσθετα μπορούμε να ορίσουμε το πεδίο ορισμού και το πεδίο τιμών μιας σχέσης. Το πεδίο ορισμού μας δηλώνει το τι τύπου επιτρέπεται να είναι οι πόροι οι οποίοι βρίσκονται στη θέση του υποκειμένου μιας τριάδας ενώ το πεδίο τιμών μας λέει τι τύπου είναι οι πόροι οι οποίοι βρίσκονται στη θέση του αντικειμένου. Τα στοιχεία που παρέχονται από την RDF-S για την υλοποίηση της λειτουργικότητας αυτής είναι τα στοιχεία rdfs:subPropertyOf, rdfs:domain και rdfs:range.

Έστω για παράδειγμα ότι θέλουμε να εμπλουτίσουμε την προηγούμενη γνώση για τα οχήματα εισάγοντας και περιγράφοντας τις ιδιότητες ex:driver και ex:primaryDriver. Καταρχάς μπορούμε να δηλώσουμε ότι τα στοιχεία αυτά αντιπροσωπεύουν ιδιότητες του κόσμους μας. Οι δηλώσεις αυτές μπορούν να γίνουν με τις τριάδες ex:driver rdf:type rdf:Property. και ex:primaryDriver rdf:type rdf:Property. Παρατηρήστε ότι στις προηγούμενες τριάδες δε χρησιμοποιήσαμε κανένα στοιχείο του λεξιλογίου της RDF-S. Είναι πολύ φυσικό να απαιτήσουμε η ιδιότητα ex:primaryDriver είναι υπο-ιδιότητα της ιδιότητας ex:driver. Η πληροφορία αυτή μπορεί να περιγραφεί σε μια τριάδα χρησιμοποιώντας το λεξιλόγιο της RDF-S. Η τριάδα είναι η, ex:primaryDriver rdfs:subPropertyOf ex:driver. Τέλος όπως είναι

αντιληπτό η ιδιότητα ex:driver συνδέει ένα όχημα με έναν άνθρωπο. Η γνώση αυτή μπορεί να περιγραφεί με τις τριάδες, ex:driver rdfs:domain ex:MotoveVehicle και ex:driver rdfs:domain ex:Person. Η RDF/XML μορφή των παραπάνω προτάσεων είναι η ακόλουθη,

```
<rdf:Property rdf:ID="driver">
  <rdfs:domain rdf:resource="#MotorVehicle"/>
  <rdfs:range rdf:resource="#Person">
</rdf:Property>

<rdf:Property rdf:ID="primaryDriver">
  <rdfs:subPropertyOf rdf:resource="#driver"/>
</rdf:Property>
```

Τέλος η RDF-S μας παρέχει δύο στοιχεία τα οποία μπορούμε να χρησιμοποιήσουμε για να δημιουργήσουμε περιγραφές σε ελεύθερο κείμενο (free text) των πόρων των οποίων περιγράφουμε. Τα στοιχεία αυτά είναι το rdfs:comment και rdfs:label. Το στοιχείο rdfs:comment μπορεί να χρησιμοποιηθεί για να περιγραφθεί ένας πόρος χρησιμοποιώντας ελεύθερο κείμενο. Από την άλλη το στοιχείο rdfs:label μπορεί να χρησιμοποιηθεί για να δηλώσουμε ένα εναλλακτικό περισσότερο περιγραφικό όνομα για έναν πόρο μας. Τελειώνοντας να αναφέρουμε ότι για να αναφερθούμε στις γλώσσες RDF και RDF-S χρησιμοποιούμε πολλές φορές την ονομασία RDF(S).

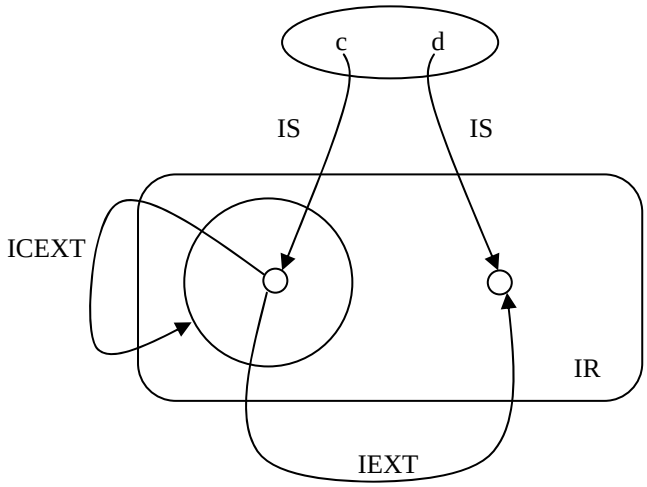
4.2.3 Σημασιολογία των RDF και RDF-S

Όπως κάθε γλώσσα αναπαράστασης γνώσης έτσι και για τις γλώσσες RDF και RDF-S ορίζεται τυπική σημασιολογία (formal semantics) η οποία αποσκοπεί στο να αποδώσει νόημα (σημασία) στο λεξιλόγιο των γλωσσών RDF και RDF-S αλλά και σημασία στις προτάσεις (τριάδες) RDF (Hayes 2003). Στην περίπτωση της σημασιολογίας των Περιγραφικών Λογικών, η κατάσταση ήταν σχετικά απλή και ξεκάθαρη. Υπενθυμίζουμε ότι μία ερμηνεία σε μια ΠΛ αντιστοιχεί ένα άτομο σε ένα αντικείμενο, μια έννοια σε ένα σύνολο αντικειμένων και έναν ρόλο (ιδιότητα) σε ένα σύνολο από ζευγάρια αντικειμένων του χώρου ερμηνείας. Στην περίπτωση όμως των RDF και RDF-S η κατάσταση είναι πολύ πιο περίπλοκη. Αυτό συμβαίνει λόγω των δυνατοτήτων της μετα-μοντελοποίησης που παρέχει η RDF. Υπενθυμίζουμε ότι στη μετα-μοντελοποίηση ένας πόρος μπορεί να παίξει τόσο το ρόλο μιας κλάσης, μιας ιδιότητας αλλά και ενός ατόμου. Αυτός ο πολυμορφικός χαρακτήρας των πόρων καταστεί δύσκολη την ερμηνεία τους καθώς σε μια τέτοια περίπτωση αφενός κάποιος πόρος θα πρέπει να ερμηνευτεί ως αντικείμενο του χώρου ερμηνείας αλλά επιπλέον θα πρέπει να ερμηνευτεί και ως σύνολο αντικειμένων ή ακόμα και σύνολο ζευγαριών αντικειμένων. Η θεωρία μοντέλων (model theory) της RDF είναι αρκετά περίπλοκη και δύσκολη. Έτσι λοιπόν η παρουσίασή μας θα εστιαστεί στις βασικές έννοιες και τα συστατικά της θεωρίας αυτής. Ο αναγνώστης μπορεί στη συνέχεια να απευθυνθεί στο (Hayes 2003) για μια πλήρη παρουσίαση της θεωρίας μοντέλων της RDF.

Όπως αναφέραμε οι πόροι σε ένα RDF αρχείο μπορούν να κατέχουν διπλό ρόλο. Για να μπορέσουμε να ξεπεράσουμε τη δυσκολία αυτή οι ερμηνείες των προτάσεων και του λεξιλογίου των γλωσσών RDF και RDF-S βασίζονται σε μια απλή αλλά ταυτόχρονα περιεκτική ιδέα. Οι RDF ερμηνείες πραγματοποιούνται σε δύο στάδια. Στο πρώτο στάδιο όλοι οι πόροι ανεξάρτητα με το αν αναπαριστούν άτομα, κλάσεις ή ιδιότητες ερμηνεύονται (αντιστοιχούνται) σε αντικείμενα ενός χώρου ερμηνείας. Στο δεύτερο στάδιο για κάθε ένα από τα αντικείμενα αυτά προσδίδουμε μια *επέκταση (extension)* η οποία ουσιαστικά τους αποδίδει το τελικό τους νόημα.

Έστω για παράδειγμα η τριάδα c rdf:type c η οποία δηλώνει ότι το c είναι στιγμιότυπο του εαυτού του. Αυτό σημαίνει ότι το c συμπεριφέρεται τόσο ως άτομο

όσο και ως κλάση. Σε πρώτο βήμα χρησιμοποιώντας μια συνάρτηση ερμηνείας IS, αντιστοιχίζουμε τον πόρο c σε ένα άτομο του χώρου ερμηνείας IR, δηλαδή δημιουργούμε την αντιστοίχηση c→IS(c) όπου IS(c)∈IR. Αυτό συνεπάγεται ότι όσον αφορά την εμφάνιση του c ως άτομο του έχουμε αποδώσει νόημα (ερμηνεία). Σε δεύτερο βήμα χρησιμοποιούμε μια δεύτερη συνάρτηση ερμηνείας, την ICEXT, η οποία αποδίδει μια επέκταση στο αντικείμενο IS(c). Εφόσον το c συμπεριφέρεται και ως κλάση η επέκταση αυτή θα αντιστοιχίσει στο αντικείμενο IS(c) ένα σύνολο αντικειμένων του χώρου ερμηνείας. Αναθέτουμε δηλαδή στο αντικείμενο IS(c) το σύνολο ICEXT(IS(c)). Έτσι λοιπόν στο δεύτερο αυτό βήμα προσδίδουμε νόημα και στην έκφανση του πόρου c ως κλάση. Με τον ίδιο τρόπο το c θα μπορούσε σε κάποια άλλη πρόταση RDF να εμφανίζεται ως ιδιότητα. Σε αυτήν την περίπτωση θα χρησιμοποιούσαμε μια άλλη συνάρτηση ερμηνείας, την IEXT, η οποία και θα αντιστοιχίσει στο αντικείμενο IS(c) ένα σύνολο από ζευγάρια αντικειμένων του χώρου ερμηνείας. Στο Σχήμα 3 φαίνεται σχηματικά ένα παράδειγμα μιας τέτοιας ερμηνείας για τις τριάδες c rdf:type c. και c c d. οι οποίες δηλώνουν ότι το c είναι στιγμιότυπο του εαυτού του και ότι επίσης συμπεριφέρεται ως ιδιότητα που συνδέει τον εαυτό της με κάποιον άλλο πόρο d.



Σχήμα 3 *Σύνθετη ερμηνεία της θεωρίας μοντέλων της RDF.*

Όπως φαίνεται από το σχήμα ο πόρος c έχει ερμηνευτεί τόσο ως αντικείμενο (IS(c)), κλάση η οποία και περιέχει ως μέλος της την ερμηνεία του c ως άτομο, δηλαδή ισχύει IS(c)∈ICEXT(IS(c)), αλλά και ως ιδιότητα η οποία περιέχει μια σύνδεση από το αντικείμενο IS(c) στο αντικείμενο IS(d), δηλαδή ισχύει (IS(c),IS(d))∈IEXT(IS(c)).

3. Εκφραστική Αναπαράσταση με την OWL

Στις προηγούμενες ενότητες είδαμε τις γλώσσες RDF και RDF-S. Είδαμε ότι το λεξιλόγιο των γλωσσών αυτών προσφέρει μια πολύ βασική εκφραστική δυνατότητα. Οι γλώσσες αυτές έχουν ουσιαστικά ως σκοπό να αποτελέσουν το θεμέλιο για πιο εκφραστικές γλώσσες ανώτερων επιπέδων, όπως είναι οι γλώσσες του λογικού επιπέδου. Μιλώντας με όρους Περιγραφικής Λογικής είδαμε ότι οι RDF και RDF-S παρέχουν τη δυνατότητα δημιουργίας σχέσεων υπαγωγής μεταξύ κλάσεων (δηλαδή εννοιών) και ιδιοτήτων (δηλαδή ρόλων) αλλά και τη δυνατότητα δημιουργίας ισχυρισμών. Επιπρόσθετα, η RDF-S παρέχει τη δυνατότητα καθορισμού του πεδίου ορισμού και του πεδίου τιμών μιας ιδιότητας με τη μορφή αξιωμάτων. Όπως θα δούμε στη συνέχεια τα δύο αυτά αξιώματα μπορούν να περιγραφούν ως δύο αξιώματα υπαγωγής στις ΠΛ.

Οι RDF και RDF-S, λοιπόν, δεν παρέχουν τη δυνατότητα να ορίσουμε κλάσεις από την ένωση, την τομή ή την άρνηση μιας ή και περισσοτέρων άλλων κλάσεων. Οι δυνατότητες αυτές και πολλές ακόμα παρέχονται από τις ΠΛ. Άρα οι ΠΛ είναι ένας καλός υποψήφιος για την δημιουργία μιας γλώσσας αναπαράστασης γνώσης για το λογικό επίπεδο. Όμως, όπως έχουμε αναφέρει ήδη μια τέτοια γλώσσα θα πρέπει να πληροί ορισμένες προϋποθέσεις για να μπορέσει να χρησιμοποιηθεί στο Σημασιολογικό Ιστό. Πιο συγκεκριμένα θα πρέπει να διαθέτει μια μορφή σύνταξης η οποία να συμβαδίζει με το πρότυπο XML. Επιπρόσθετα η γλώσσα αυτή θα πρέπει να είναι συμβατή με τις γλώσσες των χαμηλότερων επιπέδων, δηλαδή τις RDF και RDF-S, αλλά ακόμα να είναι κατανοητή και εύκολη στη χρήση. Η νέα γλώσσα η οποία έρχεται να υλοποιήσει τη λειτουργικότητα του λογικού επιπέδου είναι η OWL (Web Ontology Language) (Bechhofer et. al., 2004). Όπως θα δούμε στη συνέχεια, η OWL έχει πολύ μεγάλη σχέση με εκφραστικές ΠΛ όπως οι γλώσσες $\mathcal{SHOIN}(\mathbf{D})$ και $\mathcal{SHIF}(\mathbf{D})$.

Το πρότυπο της OWL καθορίζει ουσιαστικά τρεις υπογλώσσες αυξανόμενης εκφραστικής δυνατότητας. Οι γλώσσες αυτές είναι οι ακόλουθες:

- **OWL Lite:** Η γλώσσα αυτή απευθύνεται σε χρήστες οι οποίοι επιθυμούν να χρησιμοποιήσουν την OWL για την περιγραφή γνώσης σε εφαρμογές που δεν έχουν μεγάλες απαιτήσεις σε εκφραστικές δυνατότητες. Έτσι δίνεται η δυνατότητα ανάπτυξης εξειδικευμένων εργαλείων και μηχανισμών εξαγωγής συμπερασμάτων τα οποία αναμένεται να λειτουργούν ταχύτερα από εργαλεία τα οποία υλοποιούν περισσότερο εκφραστικές γλώσσες. Μιλώντας με όρους Περιγραφικών Λογικών θα λέγαμε ότι η γλώσσα παρέχει την ίδια εκφραστική δυνατότητα με τη γλώσσα $\mathcal{SHIF}(\mathbf{D})$.

- **OWL DL:** Η γλώσσα αυτή δίνει τη μέγιστη εκφραστική δυνατότητα που προσφέρεται από τη γλώσσα OWL χωρίς όμως να χάνονται οι καλές υπολογιστικές ιδιότητές της. Αυτό σημαίνει ότι η γλώσσα αυτή, σε αντίθεση με την τελευταία υπογλώσσα της OWL, είναι *αποφασίσιμη (decidable)*. Συγκριτικά με τις ΠΛ, η OWL DL παρέχει την ίδια εκφραστική δυνατότητα με τη γλώσσα $\mathcal{SHOIN}(\mathbf{D})$.

- **OWL Full:** Η γλώσσα αυτή προσφέρει το ίδιο λεξιλόγιο με τη γλώσσα OWL DL. Επιπρόσθετα όμως παρέχει τη συντακτική ελευθερία και τα χαρακτηριστικά της γλώσσας RDF και πιο συγκεκριμένα τη δυνατότητα μετα-μοντελοποίησης. Η γλώσσα αυτή είναι εμφανώς *μη-αποφασίσιμη (undecidable)* (Boris 2005).

Από τα παραπάνω γίνεται αντιληπτό ότι η μοναδική γλώσσα η οποία παρέχει συμβατότητα με το μοντέλο και τη σημασιολογία της RDF είναι η OWL Full. Από την άλλη όμως η μεγάλη εκφραστική δυνατότητά της την καταστεί μη-αποφασίσιμη και μέχρι σήμερα δεν είναι γνωστός κανένας αλγόριθμος εξαγωγής συμπερασμάτων γι αυτήν. Η μη-αποφασισιμότητα της OWL Full ανάγκασε τη ομάδα εργασίας τη OWL (OWL Working Group) να δημιουργήσει τις υπογλώσσες OWL Lite και OWL DL για τις οποίες βελτιστοποιημένοι αλγόριθμοι ήταν γνωστοί αλλά και υλοποιημένοι. Όπως είναι προφανές οι αλγόριθμοι αυτοί δεν είναι άλλοι παρά αλγόριθμοι εξαγωγής συμπερασμάτων για τις ΠΛ στις οποίες οι γλώσσες αυτές αντιστοιχούν.

Όπως είπαμε καθώς η OWL είναι μια γλώσσα αναπαράστασης γνώσης για το Σημασιολογικό Ιστό, πρέπει να διαθέτει μια μορφή σύνταξης που είναι συμβατή με την XML. Η σύνταξη αυτή δεν είναι άλλη από τη σύνταξη RDF/XML που είδαμε σε προηγούμενες ενότητες. Καθώς όμως η OWL παρέχει αρκετά εκφραστικούς κατασκευαστές και αξιώματα η σύνταξη αυτή γίνεται πολλές φορές αρκετά μεγάλη, περίπλοκη και με ελάχιστη διδακτική σημασία. Έτσι λοιπόν η OWL διαθέτει και μια άλλη μορφή σύνταξης η οποία αναφέρεται ως *αφηρημένη σύνταξη (abstract*

syntax). Στις επόμενες ενότητες θα χρησιμοποιήσουμε την αφηρημένη σύνταξη για να παρουσιάσουμε τις εκφραστικές δυνατότητες που παρέχει η γλώσσα OWL.

Αντίστοιχα με τις ΠΛ η OWL περιλαμβάνει ένα αλφάβητο το οποίο αποτελείται από *κλάσεις (classes)*, *ιδιότητες (properties)* και *άτομα (individuals)*. Οι κλάσεις της OWL αποτελούν ένα ανάλογο των εννοιών των Περιγραφικών Λογικών. Έτσι λοιπόν διασθητικά μια κλάση αναπαρηστά ένα σύνολο από αντικείμενα τα οποία έχουν κοινά χαρακτηριστικά, όπως για παράδειγμα η κλάση των ανθρώπων ή η κλάση των αυτοκινήτων. Όπως οι ΠΛ έτσι και η OWL περιλαμβάνει τόσο ατομικές κλάσεις όσο και *περιγραφές κλάσεων (class descriptions)*. Επιπρόσθετα, παρέχονται και *αξιώματα κλάσεων (class axioms)* όπως αξιώματα υπαγωγής, ισοδυναμίας αλλά και πολλά ακόμα που θα δούμε στην συνέχεια. Οι ιδιότητες είναι το αντίστοιχο των ρόλων στις ΠΛ, έτσι λοιπόν αναπαριστούν δυαδικές σχέσης, δηλαδή ζευγάρια αντικειμένων. Επιπρόσθετα η OWL προσφέρει τη δυνατότητα ορισμού αξιωμάτων ιδιοτήτων (property axioms), όπως αξιώματα μεταβατικών ρόλων αλλά και άλλα αξιώματα τα οποία δεν εμφανίζονται άμεσα παρά μόνο έμμεσα στις ΠΛ. Τέλος η OWL προσφέρει τη δυνατότητα ορισμού ισχυρισμών, οι οποίοι στην περίπτωση της OWL ονομάζονται *γεγονότα (facts)*.

4.3.1 Περιγραφές και Αξιώματα Κλάσεων

Η πρώτη δυνατότητα που μας παρέχει η OWL είναι η περιγραφή κλάσεων ως η *τομή (intersection)* μιας λίστας άλλων περιγραφών κλάσεων. Σε αφηρημένη σύνταξη μια τέτοια δήλωση έχει τη μορφή intersectionOf(C_1, C_2, …, C_n), όπου τα Ci, $1 \leq i \leq n$ είναι περιγραφές κλάσεων. Έτσι λοιπόν μια τέτοια έκφραση δηλώνει την κλάση η οποία είναι η τομή όλων των κλάσεων C_1, C_2, …, C_n. Η αντίστοιχη περιγραφή έννοιας στις ΠΛ θα ήταν η $C_1 \sqcap C_2 \sqcap \ldots \sqcap C_n$. Όπως και στην περίπτωση των ΠΛ μια τέτοια έκφραση απλώς δηλώνει την ύπαρξη μιας κλάσης χωρίς να της δίνει κάποιο όνομα. Αν θέλουμε να ονομάσουμε την κλάση αυτή θα πρέπει να χρησιμοποιήσουμε τα αξιώματα της OWL τα οποία θα δούμε αργότερα. Με παρόμοιο τρόπο η OWL μας προσφέρει τη δυνατότητα δημιουργίας περιγραφών κλάσεων από την *ένωση (union)* μιας λίστας περιγραφών κλάσεων ή το *συμπλήρωμα (complement)* μιας περιγραφής κλάσης. Στην πρώτη περίπτωση η αφηρημένη σύνταξη είναι η εξής, unionOf(C_1, C_2, …, C_n), ενώ στη δεύτερη περίπτωση η σύνταξη είναι η εξής, complementOf(C). Το συμπλήρωμα μιας κλάσης περιγραφής δηλώνει την κλάση η οποία περιλαμβάνει όλα τα αντικείμενα τα οποία δεν ανήκουν στην κλάση C.

Όπως οι ΠΛ έτσι και η OWL παρέχει περιγραφές κλάσεων μέσω περιορισμού του τύπου και του πλήθους των τιμών που μπορεί να πάρει μια ιδιότητα. Η μέθοδος αυτή ονομάζεται *περιορισμός ιδιότητας (property restriction)*. Σε αφηρημένη σύνταξη ένας περιορισμός ιδιότητας δηλώνεται με τη *λέξη κλειδί (keyword)* restriction. Στη συνέχεια δηλώνουμε την ιδιότητα στην οποία εφαρμόζουμε τον περιορισμό και τέλος δηλώνουμε τον περιορισμό που επιθυμούμε. Πιο συγκεκριμένα η OWL προσφέρει τον καθολικό και τον υπαρξιακό περιορισμό ιδιότητας. Οι περιορισμοί αυτοί δηλώνονται με τις λέξεις κλειδιά allValuesFrom και someValuesFrom ακολουθούμενες με από μια περιγραφή κλάσης. Για παράδειγμα αν θέλουμε να δηλώσουμε την κλάση των αντικειμένων τα οποία έχουν ένα παιδί το οποίο είναι Θηλυκό τότε σε αφηρημένη σύνταξη μπορούμε να γράψουμε, restriction(hasChild someValuesFrom(Female)). Παρόμοια, η κλάση των αντικειμένων που όλα τους τα παιδιά, αν έχουν, είναι είτε ψηλά είτε έξυπνα δηλώνεται ως, restriction(hasChild allValuesFrom(unionOf(Tall, Smart))). Η αντίστοιχη περιγραφή έννοιας στις ΠΛ είναι η $\forall$hasChild.(Tall$\sqcup$Smart). Επιπρόσθετα, η OWL παρέχει περιορισμούς πληθυκότητας. Όπως και οι ΠΛ η OWL προσφέρει τους τελεστές το-πολύ και το-λιγότερο όμως, επιπλέον, παρέχει και έναν άλλο τελεστή με τον οποίο μπορούμε να δηλώσουμε μια κλάση αντικειμένων περιγράφοντας την

ακριβή πληθυκότητα των μελών της σε μια ιδιότητα. Οι κατασκευαστές αυτοί δηλώνονται με τις λέξεις κλειδιά, minCardinality, maxCardinality και cardinality, αντίστοιχα, ακολουθούμενες από έναν φυσικό αριθμό ο οποίος δηλώνει την πληθυκότητα. Για παράδειγμα μπορούμε να δηλώσουμε την κλάση των αντικειμένων που περιέχουν το λιγότερο 4 παιδιά ή την κλάση των αντικειμένων που έχουν ακριβώς δύο αυτοκίνητα ως, restriction(hasChild minCardinality(4)) και restriction(hasCar cardinality(2)). Τέλος μπορούμε να περιγράψουμε κλάσεις ανάλογα με το αν συμμετέχουν σε μια σχέση με κάποιο συγκεκριμένο αντικείμενο. Σε αφηρημένη σύνταξη ο περιορισμός αυτός δηλώνεται με τη λέξη κλειδί value ακολουθούμενη από το όνομα του αντικειμένου. Για παράδειγμα, μπορούμε να περιγράψουμε την κλάση των αντικειμένων που συμμετέχουν με το αντικείμενο Αγγλία στη σχέση citizen-of. Διαισθητικά αυτό περιγράφει την κλάση που είναι κάτοικοι της Αγγλίας και η μορφή της σε αφηρημένη σύνταξη είναι η εξής, restriction(citizen-of value(England)). Παρατηρήστε ότι στην προκειμένη περίπτωση η λέξη England είναι ένα άτομο και όχι μια κλάση. Σε σύνταξη ΠΛ αυτό είναι ανάλογο με την ονοματική έννοια {England}, ενώ η παραπάνω κλάση αντιστοιχεί στην ΠΛ έννοια ∃citizen-of.{England}.

Τέλος μας παρέχεται η δυνατότητα ορισμού κλάσεων με *απαρίθμηση (enumeration)* των μελών τους. Ένα παράδειγμα είναι ο ορισμός της έννοιας (στην περίπτωση μας κλάσης) των ημερών της εβδομάδας, απαριθμώντας τα μέλη της, δηλαδή τη Δευτέρα, την Τρίτη κοκ. Στην OWL μια τέτοια κλάση μπορεί να οριστεί με τη εξής δήλωση σε αφηρημένη σύνταξη, oneof(o_1, o_2, …, o_n), όπου τα o_1, o_2, …, o_n είναι άτομα. Η δήλωση μιας τέτοιας έννοιας σε μια ΠΛ γλώσσα η οποία διαθέτει τον κατασκευαστή της ονοματικής έννοιας και τον κατασκευαστή της ένωσης έχει την εξής σύνταξη, {Δευτέρα}⊔{Τρίτη}⊔...⊔{Κυριακή}.

Ας δούμε τώρα τι μορφές αξιωμάτων κλάσεων μας προσφέρει η OWL. Όπως είναι αναμενόμενο η OWL μας δίνει τη δυνατότητα να ορίσουμε αξιώματα υπαγωγής και αξιώματα ισοδυναμίας. Η δυνατότητα περιγραφής σχέσεων υπαγωγής ήταν ήδη διαθέσιμη από την RDF-S μέσω του στοιχείου rdfs:subClassOf. Έτσι λοιπόν η OWL χρησιμοποιεί το στοιχείο αυτό για να ορίσει σχέσεις υπαγωγής ανάμεσα σε δύο κλάσεις. Η αφηρημένη σύνταξη των δηλώσεων αυτών είναι η subClassOf(C_1, C_2). Εκτός όμως από αυτήν την απλή μέθοδο δήλωσης μιας σχέσης υπαγωγής η OWL μας προσφέρει και μια επέκτασή της. Πιο συγκεκριμένα μπορούμε να δηλώσουμε μια σχέση υπαγωγής ανάμεσα σε μια κλάση και σε μια τομή από περιγραφές κλάσεων. Η δήλωση αυτή έχει τη μορφή Class(A partial C_1, C_2, …, C_n) ενώ σε ΠΛ η μορφή της είναι η εξής, $A \sqsubseteq C_1 \sqcap C_2 \sqcap \ldots \sqcap C_n$. Έτσι λοιπόν συνδυάζοντας περιγραφές κλάσεων και το παραπάνω αξίωμα μπορούμε να δημιουργήσουμε μια δήλωση της μορφής, Class(A partial unionOf(C, D), restriction(R allValuesFrom(F))) η οποία έχει την ίδια σημασία με την δήλωση $A \sqsubseteq (C \sqcup D) \sqcap \forall R.F$, σε σύνταξη ΠΛ.

Αντίθετα από την RDF και RDF-S και παρόμοια με τις ΠΛ η OWL μας προσφέρει τη δυνατότητα να ορίσουμε σχέσεις ισοδυναμίας ανάμεσα σε κλάσεις. Πιο συγκεκριμένα μας δίνεται η δυνατότητα να ορίσουμε ισοδυναμίες ανά δύο σε μια λίστα από n κλάσεις. Οι δηλώσεις αυτές έχουν τη μορφή EquivalentClasses(C_1, C_2, …, C_n) ενώ η σημασιολογία τους είναι η ίδια με τη σημασιολογία που προσφέρεται από τις δηλώσεις $C_i \equiv C_j$, $1 \leq i < j \leq n$ σε μια ΠΛ. Όπως και στην περίπτωση των σχέσεων υπαγωγής έτσι και τώρα έχουμε τη δυνατότητα να ορίσουμε μια κλάση ως ισοδύναμη με την τομή μιας λίστας κλάσεων. Η σύνταξη των αξιωμάτων αυτών είναι η ακόλουθη, Class(A complete C_1, C_2, …, C_n), ενώ η σημασιολογία του αξιώματος αυτού είναι ίδια με αυτή του αξιώματος $A \equiv C_1 \sqcap C_2 \sqcap \ldots \sqcap C_n$ στις ΠΛ. Επιπρόσθετα χρησιμοποιώντας την περιγραφή κλάσεων με απαρίθμηση μπορούμε να δηλώσουμε μια κλάση ως την απαρίθμηση κάποιων ατόμων. Η σύνταξη είναι η, EnumeratedClass(A complete o_1, o_2, …, o_n).

Τέλος η OWL μας προσφέρει τη δυνατότητα να δηλώσουμε ότι μια λίστα από κλάσεις είναι ξένες μεταξύ τους ανά δύο. Για παράδειγμα μπορούμε να πούμε ότι οι κλάσεις Αρσενικό και Θηλυκό ή οι κλάσεις Ψηλός και Κοντός είναι ξένες μεταξύ τους. Οι δηλώσεις αυτές σχηματίζονται με τη λέξη κλειδί DisjointClasses η οποία ακολουθείται από μια λίστα από κλάσεις. Για παράδειγμα θα μπορούσαμε να γράψουμε, DisjointClasses(Αρσενικό, Θηλυκό). Τελειώνοντας, να αναφέρουμε ότι η OWL μας προσφέρει δύο κλάσεις οι οποίες αντιστοιχούν στις ΠΛ έννοιες, $\top$ και $\bot$. Οι κλάσεις αυτές στην OWL αναφέρονται με τα ονόματα owl:Thing και owl:Nothing, αντίστοιχα.

4.3.2 Περιγραφές και Αξιώματα Ιδιοτήτων

Η OWL ξεχωρίζει ανάμεσα σε δύο είδη ιδιοτήτων. Το πρώτο ονομάζεται *ιδιότητες αντικειμένων (object properties)* ενώ το δεύτερο *ιδιότητες τύπων δεδομένων (datatype properties).* Οι ιδιότητες αντικειμένων χρησιμοποιούνται για να συνδέσουν δύο άτομα μεταξύ τους, όπως για παράδειγμα η ιδιότητα εχωΠατερα συνδέει ένα άτομο με ένα άλλο που αντιπροσωπεύει τον πατέρα του, ενώ οι ιδιότητες τύπων δεδομένων συνδέουν ένα άτομο με μια τιμή τύπου δεδομένων, όπως για παράδειγμα η ιδιότητα εχωΗλικια η οποία συνδέει ένα άτομο με μια τιμή που αντιπροσωπεύει την ηλικία του. Παρόμοια με τις ΠΛ η OWL προσφέρει ένα σύνολο από αξιώματα ιδιοτήτων. Στη συνέχεια παρουσιάζουμε τα αξιώματα αυτά.

Στην περίπτωση των ιδιοτήτων τιμών δεδομένων μπορούμε μόνο να ορίσουμε μια ιδιότητα ως, DatatypeProperty(R). Παρατηρήστε, ότι στις ιδιότητες τύπων δεδομένων δε μας δίνεται η δυνατότητα να ορίσουμε μια ιδιότητα ως την αντίστροφη μιας άλλης. Αυτό συμβαίνει γιατί η αντίστροφη ιδιότητα μιας ιδιότητας τύπων δεδομένων θα συνέδεε μια τιμή με ένα αντικείμενο το οποίο δεν επιτρέπεται.

Όπως και στις ΠΛ, μας δίνεται η δυνατότητα να ορίσουμε ένα αξίωμα υπαγωγής ιδιοτήτων. Οι ορισμοί αυτοί δίνονται από την αφηρημένη σύνταξη, SubPropertyOf(R S), όπου R και S είναι ιδιότητες αντικειμένων ή τύπων δεδομένων. Επιπλέον η OWL μας προσφέρει τη δυνατότητα να ορίσουμε μια ιδιότητα ως υπο-ιδιότητα μιας λίστας ιδιοτήτων. Η δήλωση αυτή γίνεται γράφοντας, ObjectProperty(R super(R_1) … super(R_n)), όπου οι σχέσεις R, R_1 … R_n είναι ιδιότητες αντικειμένων ή τύπων δεδομένων. Επιπρόσθετα μπορούμε να ορίσουμε μια ιδιότητα ως την αντίστροφη μιας άλλης γράφοντας, ObjectProperty(R inverseOf(S)), το οποίο δηλώνει ότι η ιδιότητα R είναι η αντίστροφη της S. Αυτό το αξίωμα σε ΠΛ αντιστοιχεί σε δύο αξιώματα της μορφής $S \sqsubseteq R^-$ και $R^- \sqsubseteq S$ τα οποία σε σημασιολογία Περιγραφικών Λογικών ερμηνεύονται ως, $\forall a,b \in \Delta^{\mathcal{I}}$ αν $S^{\mathcal{I}}(a,b)$ τότε $R^{\mathcal{I}}(b,a)$ και αντίστροφα. Αντίθετα με τις ΠΛ στην OWL μας δίνεται η δυνατότητα να ορίσουμε ισοδυναμία μεταξύ δύο ιδιοτήτων χρησιμοποιώντας τη σύνταξη EquivalentProperties(R S), όπου R και S είναι όπως πριν. Το αξίωμα αυτό δε μας προσφέρει ουσιαστικά κάποια μεγαλύτερη εκφραστική δυνατότητα σε σχέση με τις ΠΛ αφού είναι ισοδύναμο με δύο αξιώματα υπαγωγής της μορφής, $S \sqsubseteq R$ και $R \sqsubseteq S$. Στη συνέχεια θα γράφουμε $S \equiv R$ ως συντόμευση των δύο αυτών υπαγωγών. Επιπλέον, η OWL κληρονομεί τα αξιώματα ορισμού πεδίου ορισμού και συνόλου τιμών μιας ιδιότητας αντικειμένων ή τύπων δεδομένων. Η αφηρημένη σύνταξη για αυτά τα αξιώματα είναι η ObjectProperty(R domain(C)) και ObjectProperty(R range(C)), αντίστοιχα, όπου R είναι μια ιδιότητα αντικειμένων και C είναι μια κλάση. Στην περίπτωση των ιδιοτήτων τύπων δεδομένων η λέξη κλειδί ObjectProperty αντικαθίσταται με τη λέξη κλειδί DatatypeProperty και η κλάση C στην περίπτωση του ορισμού του πεδίου τιμών αντικαθίσταται με έναν τύπο δεδομένων. Πιο συγκεκριμένα ο ορισμός γίνεται γράφοντας, DatatypeProperty(T range(d)).

Επίσης η OWL μας προσφέρει μια σειρά από αξιώματα ιδιοτήτων κάποια από τα οποία δεν εμφανίζονται άμεσα, παρά μόνο έμμεσα, στις Περιγραφικές Λογικές. Πιο συγκεκριμένα μπορούμε να δηλώσουμε ότι μια ιδιότητα αντικειμένων είναι

συμμετρική (symmetric). Μια ιδιότητα R ονομάζεται συμμετρική ανν ∀a,b, R(a,b) συνεπάγεται ότι R(b,a). Η δήλωση αυτή γίνεται γράφοντας σε αφηρημένη σύνταξη, ObjectProperty(R symmetric). Οι ΠΛ δεν παρέχουν άμεσα μια τέτοια δήλωση όμως μπορούμε να δηλώσουμε ένα αξίωμα της μορφής $R \equiv R^-$ το οποίο ουσιαστικά αναγκάζει την R να ερμηνευτεί ως συμμετρική σε όλες τις ερμηνείες που ικανοποιούν το αξίωμα. Μπορούμε επιπλέον να δηλώσουμε ότι μια ιδιότητα αντικειμένων R είναι μεταβατική, γράφοντας, ObjectProperty(R transitive). Επιπρόσθετα, μπορούμε να δηλώσουμε μια ιδιότητα αντικειμένων ή τύπων δεδομένων ως *συναρτησιακή (functional)*. Μια σχέση ονομάζεται συναρτησιακή αν, ∀a,b,c R(a,b) και R(a,c) συνεπάγεται ότι b=c. Για παράδειγμα η σχέση εχωΦυσικοΠατερα είναι συναρτησιακή καθώς είναι αδύνατο κάποιος να έχει παραπάνω από έναν φυσικό πατέρα. Η δήλωση αυτή γίνεται γράφοντας, ObjectProperty(R functional). Αντίθετα από τα προηγούμενα, το αξίωμα αυτό μπορεί να γίνει και σε ιδιότητες τύπων δεδομένων. Σε αυτήν την περίπτωση το αξίωμα έχει τη μορφή DatatypeProperty(T functional), όπου T είναι μια ιδιότητα τύπων δεδομένων. Για παράδειγμα η ιδιότητα τύπων δεδομένων εχωΗλικια είναι συναρτησιακή. Εκ πρώτης όψης το αξίωμα αυτό φαίνεται να μας δίνει μεγαλύτερη εκφραστική δυνατότητα από αυτή των ΠΛ. Στην πραγματικότητα όμως ένα τέτοιο αξίωμα μπορεί σε μια ΠΛ να δηλωθεί από το αξίωμα υπαγωγής $\top \sqsubseteq \leq 1R$, το οποίο διαισθητικά δηλώνει ότι κάθε αντικείμενο του $\Delta^{\mathcal{I}}$ συμμετέχει στη σχέση R το πολύ με ένα άλλο αντικείμενο. Τέλος μας δίνεται η δυνατότητα να ορίσουμε μια σχέση ως *αντίστροφη συναρτησιακή (inverse functional).* Διαισθητικά το αξίωμα αυτό δηλώνει ότι το πολύ ένα αντικείμενο μπορεί να συνδέεται με ένα άλλο αντικείμενο, δηλαδή αν ισχύουν τα R(a,c) και R(b,c) τότε πρέπει να ισχύει και a=b.

4.3.3 Άτομα

Όπως οι ΠΛ έτσι και η OWL μας παρέχει τη δυνατότητα δημιουργίας ισχυρισμών ανάμεσα σε ένα άτομο και μια κλάση, ανάμεσα σε ένα ζεύγος ατόμων και μια ιδιότητα αντικειμένων ή ανάμεσα σε ένα ζεύγος ατόμου και τιμής και μια ιδιότητα τύπων δεδομένων. Σε αφηρημένη σύνταξη οι δηλώσεις αυτές γίνονται γράφοντας Individual(o type(C_1) ... type(C_n) value(R_1, o_1) … value(R_n, o_n)), όπου τα o, o_1 έως o_n είναι άτομα τα C_1 ... C_n είναι κλάσεις και τα R, R_1 έως και R_n είναι ιδιότητες. Στη περίπτωση όπου μια ιδιότητα R_i είναι ιδιότητα τύπων δεδομένων το άτομο o_i πρέπει να αντικατασταθεί από μια τιμή, δηλαδή κάποιο plain ή typed literal. Η OWL όμως προσφέρει και επιπλέον δυνατότητες στη δημιουργία ισχυρισμών. Πιο συγκεκριμένα μας δίνεται η δυνατότητα να ορίσουμε μια λίστα από άτομα ως *ταυτοδύναμα* γράφοντας SameIndividual(o_1 … o_n) ή ότι μια λίστα από άτομα είναι διαφορετικά μεταξύ τους γράφοντας DifferentIndividuals(o_1 … o_n).

4.3.4 OWL Lite, OWL DL και OWL Full

Το παραπάνω σύνολο περιγραφής και αξιωμάτων κλάσεων και ιδιοτήτων είναι το μέγιστο δυνατό που μας παρέχει η γλώσσα OWL. Οι υπογλώσσες που το προσφέρουν είναι η OWL DL και η OWL Full. Όπως είπαμε και στην εισαγωγή η διαφορά μεταξύ OWL DL και OWL Full είναι ότι η OWL Full προσφέρει επιπλέον τη δυνατότητα μεταμοντελοποίησης. Μπορούμε δηλαδή να χρησιμοποιήσουμε ένα στοιχείο A και ως κλάση και ως ιδιότητα αλλά και ως άτομο ταυτόχρονα μέσα σε μια βάση γνώσης. Για παράδειγμα μπορεί κάποιος να γράψει τη δήλωση Individual(A type(A)). Ακόμα περισσότερο στην OWL Full μπορεί κάποιος να χρησιμοποιήσει τα ίδια τα δομικά στοιχεία της γλώσσας σε οποιαδήποτε θέση επιθυμεί. Μπορούμε δηλαδή να γράψουμε μια έκφραση της μορφής SameIndividual(A unionOf), το οποίο δηλώνει ότι το στοιχείο A είναι το ίδιο με το στοιχείο unionOf και άρα μπορεί να χρησιμοποιηθεί για να δηλώνει την ένωση μιας λίστας κλάσεων. Αντίθετα η γλώσσα OWL DL διαχωρίζει αυστηρά το σύνολο των κλάσεων, των ιδιοτήτων και των ατόμων.

Αντίθετα από τις OWL DL και OWL Full η OWL Lite προσφέρει ένα πιο περιορισμένο σύνολο περιγραφών και αξιωμάτων κλάσεων. Αυτό είχε ως σκοπό τη δημιουργία μιας λιγότερο εκφραστικής υπογλώσσας στην οποία οι αλγόριθμοι εξαγωγής συμπερασμάτων θα είναι πολύ γρήγοροι. Έτσι λοιπόν στην OWL Lite κάποια από τα δομικά στοιχεία της OWL που είδαμε σε προηγούμενες ενότητες απαγορεύεται να χρησιμοποιηθούν. Πιο συγκεκριμένα η OWL Lite απαγορεύει την χρήση των λέξεων κλειδιών, owl:oneOf, owl:unionOf, owl:complementOf, owl:hasValue και owl:dishointWith. Επιπρόσθετα, οι περιορισμοί πληθυκότητας, minCardinality και maxCardinality περιορίζονται σε συναρτησιακούς περιορισμούς. Αυτό σημαίνει ότι οι βαθμοί πληθυκότητας μπορούν να είναι μόνο το 0 και το 1. Αν θέλαμε να μιλήσουμε με όρους Περιγραφικών Λογικών θα λέγαμε ότι η OWL Lite απαγορεύει τις ονοματικές έννοιες (nominlas), οι οποίες συμβολίζονται με το γράμμα $\mathcal{O}$, ενώ αντί για τον πλήρη περιορισμό πληθυκότητας ($\mathcal{N}$) επιτρέπει μόνο συναρτησιακούς περιορισμούς ($\mathcal{F}$). Σε αυτό το σημείο ο αναγνώστης είναι απολύτως φυσικό να υποθέσει ότι η OWL Lite απαγορεύει επίσης την περιγραφή κλάσεων με τη χρήση του συμπληρώματος ($\neg$), το οποίο και συμβολίζεται με $\mathcal{C}$, ή τη χρήση του κατασκευαστή της ένωσης ($\sqcup$), ο οποίος υπενθυμίζουμε ότι συμβολίζεται με το γράμμα ($\mathcal{U}$). Στην πραγματικότητα όμως κάτι τέτοιο δεν είναι αληθές. Πιο συγκεκριμένα το σύνολο των στοιχείων (κατασκευαστών) που προσφέρει η OWL Lite είναι τέτοιο ώστε αυτοί οι κατασκευαστές να μπορούν να προσομοιωθούν. Για παράδειγμα έστω η έννοια A η οποία σε σύνταξη ΠΛ ορίζεται από το αξίωμα ισοδυναμίας $A \equiv C \sqcap D$. Αν ο χρήστης θέλει να ορίσει την άρνηση της έννοιας A μπορεί να εργαστεί ως εξής. Ορίζει έναν νέο ρόλο, έστω τον forA. Στη συνέχεια ορίζει επιπλέον την έννοια A ως ισοδύναμη της έννοιας ≥ 1forA. Τέλος, ορίζει την έννοια negA ως, $negA \equiv \leq 0$forA. Η έννοια negA είναι το συμπλήρωμα της έννοιας, A. Αυτό συμβαίνει γιατί η έννοια ≥ 1forA είναι το συμπλήρωμα της έννοιας ≤ 0forA. Έτσι λοιπόν χρησιμοποιώντας κάθε φορά και έναν νέο ρόλο μπορούμε να ορίσουμε την άρνηση μιας έννοιας. Παρομοίως μπορούμε να εργαστούμε και για τον ορισμό της ένωσης δύο εννοιών. Συνοψίζοντας, η OWL Lite αντιστοιχεί στην ΠΛ $\mathcal{SHIF}(\mathbf{D})$ η οποία είναι μια αρκετά εκφραστική γλώσσα και άρα ο αρχικός σκοπός για τη δημιουργία μιας μη εκφραστικής γλώσσας απέτυχε.

4.3.5 RDF/XML Σύνταξη

Όπως αναφέραμε και στην εισαγωγή εφόσον η OWL είναι μια γλώσσα αναπαράστασης γνώσης για το Σημασιολογικό Ιστό εκτός από την αφηρημένη σύνταξη πρέπει να διαθέτει και RDF/XML σύνταξη. Η σύνταξη αυτή μας δίνει τη δυνατότητα να διαμορφώσουμε τη γνώση μας σε μια μορφή προσπελάσιμη από εφαρμογές του Παγκοσμίου Ιστού. Εφόσον η OWL διαθέτει αρκετά περίπλοκους κατασκευαστές η σύνταξη αυτή είναι πολλές φορές περίπλοκη. Στην ενότητα αυτή θα δούμε ορισμένα παραδείγματα της σύνταξης αυτής χωρίς να κάνουμε μια λεπτομερή παρουσίαση.

Στις περισσότερες περιπτώσεις η RDF/XML σύνταξη προκύπτει από την αντιστοίχηση των λέξεων κλειδιών της αφηρημένης σύνταξης της OWL σε στοιχεία της γλώσσας XML. Ας θεωρήσουμε για παράδειγμα τον ορισμό της κλάσης restriction(R allValuesFrom(C)), ο οποίος μας δηλώνει μια OWL κλάση εφαρμόζοντας έναν περιορισμό στην ιδιότητα R. Σε RDF/XML σύνταξη η λέξη κλειδί restriction αντιστοιχεί στο στοιχείο owl:Restriction, ενώ η λέξη κλειδί allValuesFrom αντιστοιχεί στο στοιχείο owl:allValuesFrom. Για να μπορέσουμε όμως να δηλώσουμε σε ποια ιδιότητα εφαρμόζουμε τον περιορισμό, με σκοπό τον ορισμό της κλάσης, χρειαζόμαστε ένα επιπλέον XML στοιχείο. Το στοιχείο αυτό είναι το στοιχείο owl:onProperty. Συνοψίζοντας η RDF/XML μορφή της παραπάνω δήλωσης είναι η ακόλουθη,

<owl:Restriction>

```
        <owl:onProperty rdf:resource="#R" />
        <owl:allValuesFrom rdf:resource="#C">
</owl:Restriction>
```

Παρομοίως, οι λέξεις κλειδιά Class και unionOf αντιστοιχούν στα στοιχεία owl:Class και owl:unionOf. Έτσι λοιπόν η δήλωση Class(A complete unionOf(C D)) αντιστοιχεί στη RDF/XML δήλωση,

```
<owl:Class rdf:ID="A">
        <owl:unionOf rdf:parseType="Collection">
                <owl:Class rdf:about="#C" />
                <owl:Class rdf:about="#D" />
        </owl:unionOf>
</owl:Class>
```

Έστω τώρα ότι σε αφηρημένη σύνταξη OWL κάνουμε τη δήλωση ObjectProperty(R domain(C) range(D) transitive symmetric) η οποία μας δηλώνει ότι το πεδίο ορισμού της ιδιότητας R είναι η έννοια C το πεδίο τιμών είναι η έννοια D και τέλος ότι η ιδιότητα R είναι μεταβατική και συμμετρική. Χρησιμοποιώντας τα στοιχεία XML που αντιστοιχούν στις παραπάνω λέξεις κλειδιά μπορούμε να κάνουμε τη δήλωση αυτή χρησιμοποιώντας τη σύνταξη RDF/XML. Η δήλωση αυτή είναι η ακόλουθη,

```
<owl:SymmetricProperty rdf:ID="R">
   <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#InverseFunctionalPr
operty"/>
   <rdfs:domain rdf:resource="#C"/>
   <rdfs:range rdf:resource="#D"/>
   <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#TransitiveProperty"/>
   <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</owl:SymmetricProperty>
```

Τέλος ας δούμε ένα αξίωμα ατόμων. Έστω η δήλωση Individual(a type(C) type(D) value(R, b)) η οποία μας δηλώνει ότι το άτομο a είναι στιγμιότυπο των κλάσεων C και D και συνδέεται με το άτομο b μέσω της σχέσης R. Η αντίστοιχη δήλωση σε RDF/XML σύνταξη είναι η ακόλουθη,

```
<C rdf:ID="o">
   <rdf:type rdf:resource="#D"/>
   <R rdf:resource="#b" />
</C>
```

Εφόσον θεωρητικά μια OWL οντολογία πρέπει να επεκτείνει μια RDF οντολογία, ένα OWL αρχείο εκκινεί με τις ίδιες ετικέτες με τις οποίες εκκινεί και ένα RDF αρχείο. Πιο συγκεκριμένα ένα OWL αρχείο ξεκινά με την ετικέτα <rdf:RDF>. Στη συνέχεια όμως προκειμένου να δηλώσουμε ότι αυτό που ακολουθεί είναι μια OWL οντολογία χρησιμοποιούμε την ετικέτα owl:Ontology. Η ετικέτα αυτή περιέχει το στοιχείο rdf:about το οποίο αποδίδει ένα όνομα για την οντολογία. Εκτός από τα στοιχεία τα οποία είδαμε και έχουν σχέση με περιγραφή και αξιώματα κλάσεων και ιδιοτήτων η OWL προσφέρει και μια σειρά από στοιχεία τα οποία χρησιμοποιούνται για την περιγραφή μεταπληροφορίας για την οντολογία. Πιο συγκεκριμένα το στοιχείο owl:priorVersion δηλώνει ένα URL στο οποίο βρίσκεται μια OWL οντολογία η οποία είναι προηγούμενη έκδοση της οντολογίας που δηλώνεται από το τρέχον αρχείο OWL. Επίσης το στοιχείο owl:versionInfo χρησιμοποιείται για να δηλώσει πληροφορίες για το τρέχον αρχείο όπως για παράδειγμα ημερομηνία και ώρα δημιουργίας και άλλα. Τέλος ένα στοιχείο της OWL το οποίο χρησιμοποιείται αρκετά αλλά επίσης έχει αποτελέσει αμφιλεγόμενο σημείο για τους ερευνητές όσον αφορά τη σημασιολογία του είναι το στοιχείο owl:imports. Με το στοιχείο αυτό μπορούμε να

εισάγουμε τους ορισμούς, κλάσεις, αξιώματα και ιδιότητες μιας άλλης οντολογίας στο τρέχον αρχείο.

4.3.6 Σημασιολογία

Η σημασιολογία της OWL περιγράφεται στο κείμενο (Patel-Schneider et. at., 2004) του προτύπου της OWL. Η σημασιολογία αυτή αποδίδεται απευθείας στα RDF/XML στοιχεία της OWL, όπως είναι το allValuesFrom και άλλα. Αυτό συνεπάγεται ότι προκειμένου να ερμηνευτούν κάποιες περίπλοκες και περιττές δομές που χρησιμοποιούνται στην RDF/XML σύνταξη της OWL η σημασιολογία αυτή περιπλέκεται χωρίς να υπάρχει κάποιος ιδιαίτερος λόγος. Για το λόγο αυτό συνήθως για να αποδώσουμε ερμηνεία στα δομικά στοιχεία, τα αξιώματα και τους κατασκευαστές της OWL, η οποία να έχει διδακτική και ουσιαστική σημασία εργαζόμαστε διαφορετικά. Ανάγουμε τους κατασκευαστές και τα δομικά στοιχεία της OWL σε εκφραστικές Περιγραφικές Λογικές. Η αναγωγή αυτή οφείλεται στους Horrocks και Patel-Schneider (2003). Εκτός από την παροχή σημασιολογίας στους κατασκευαστές και τα αξιώματα της OWL η αναγωγή αυτή μας προσφέρει ακόμα ένα πλεονέκτημα. Μπορούμε να ανάγουμε τα προβλήματα εξαγωγής συμπερασμάτων των OWL οντολογιών στα προβλήματα εξαγωγής συμπερασμάτων των Περιγραφικών Λογικών για τα οποία υπάρχουν υλοποιημένοι και βελτιστοποιημένοι αλγόριθμοι. Στο σχήμα 4 παρουσιάζουμε την αντιστοιχία ανάμεσα στην αφηρημένη σύνταξη των περιγραφών OWL κλάσεων και των περιγραφών εννοιών των Περιγραφικών Λογικών. Όπως παρατηρούμε, από την αντιστοιχία αυτή, προκύπτει και η σημασιολογία των δομικών στοιχείων της γλώσσας OWL.

Πίνακας 4 *Περιγραφές OWL κλάσεων*

Αφηρημένη Σύνταξη	**Σύνταξη ΠΛ**	**Σημασιολογία**
owl:Thing	$\top$	$\top^{\mathcal{I}}=\Delta^{\mathcal{I}}$
owl:Nothing	$\bot$	$\bot^{\mathcal{I}}=\varnothing$
intersectionOf(C_1, …, C_n)	$C_1\sqcap\ldots\sqcap C_n$	$(C_1\sqcap\ldots\sqcap C_n)^{\mathcal{I}}=C_1^{\mathcal{I}}\cap\ldots\cap C_n^{\mathcal{I}}$
unionOf(C_1, …, C_n)	$C_1\sqcup\ldots\sqcup C_n$	$(C_1\sqcup\ldots\sqcup C_n)^{\mathcal{I}}=C_1^{\mathcal{I}}\cup\ldots\cup C_n^{\mathcal{I}}$
complementOf(C)	$\neg C$	$(\neg C)^{\mathcal{I}}=\Delta^{\mathcal{I}}\setminus C^{\mathcal{I}}$
oneOf(o_1, …, o_n)	$\{o_1\}\sqcup\ldots\sqcup\{o_n\}$	$(\{o_1\}\sqcup\ldots\sqcup\{o_n\})^{\mathcal{I}}=\{o_1^{\mathcal{I}},\ldots,o_n^{\mathcal{I}}\}$
restriction(R someValuesFrom(C))	$\exists R.C$	$(\exists R.C)^{\mathcal{I}}=\{a^{\mathcal{I}} \mid \exists b^{\mathcal{I}}.\ (a^{\mathcal{I}},b^{\mathcal{I}})\in R^{\mathcal{I}}\ \&\ b^{\mathcal{I}}\in C^{\mathcal{I}}\}$
restriction(R allValuesFrom(C))	$\forall R.C$	$(\forall R.C)^{\mathcal{I}}=\{a^{\mathcal{I}} \mid \forall b^{\mathcal{I}}.\ (a^{\mathcal{I}},b^{\mathcal{I}})\in R^{\mathcal{I}}\rightarrow b^{\mathcal{I}}\in C^{\mathcal{I}}\}$
restriction(R hasValue(o))	$\exists R.\{o\}$	$(\exists R.\{o\})^{\mathcal{I}}=\{a^{\mathcal{I}} \mid (a^{\mathcal{I}},o^{\mathcal{I}})\in R^{\mathcal{I}}\}$
restriction(R minCardinality(n))	$\geq nR$	$(\geq nR)^{\mathcal{I}}=\{a^{\mathcal{I}} \mid \#\{b^{\mathcal{I}} \mid (a^{\mathcal{I}},b^{\mathcal{I}})\in R^{\mathcal{I}}\}\geq n\}$
restriction(R maxCardinality(n))	$\leq nR$	$(\leq nR)^{\mathcal{I}}=\{a^{\mathcal{I}} \mid \#\{b^{\mathcal{I}} \mid (a^{\mathcal{I}},b^{\mathcal{I}})\in R^{\mathcal{I}}\}\leq n\}$

restriction(R cardinality(n))	$\geq nR \sqcap \leq nR$	$(\geq nR \sqcap \leq nR)^{\mathcal{I}}=\{a^{\mathcal{I}} \mid \#\{b^{\mathcal{I}} \mid (a^{\mathcal{I}},b^{\mathcal{I}}) \in R^{\mathcal{I}}\}=n\}$

Παρόμοια αντιστοιχία μπορεί να οριστεί και στην περίπτωση των αξιωμάτων κλάσεων της γλώσσας OWL. Τα αξιώματα αυτά μπορούν να αντιστοιχηθούν σε αξιώματα υπαγωγής και ισοδυναμίας μεταξύ εννοιών μιας Περιγραφικής Λογικής. Οι αντιστοιχίες φαίνονται στον πίνακα 5 και στις περισσότερες των περιπτώσεων είναι αρκετά προφανής. Η μόνη περίπτωση που αξίζει να σχολιάσουμε είναι αυτή των ξένων κλάσεων. Από τη θεωρία συνόλων γνωρίζουμε ότι δύο σύνολα είναι ξένα μεταξύ τους αν η τομή τους είναι το κενό σύνολο. Αν μεταφερθούμε από το χώρο των συνόλων στον αφηρημένο χώρο των εννοιών τότε μπορούμε να ορίσουμε δύο έννοιες ως ξένες αν η τομή τους υπάγεται στην κενή έννοια. Αξίζει στο σημείο αυτό να επισημάνουμε ότι υπάρχει ένας εναλλακτικός τρόπος δήλωσης ότι δύο έννοιες είναι ξένες μεταξύ τους. Πιο συγκεκριμένα η έννοια C είναι ξένη με την έννοια D αν και μόνο αν $C \sqsubseteq \neg D$.

Πίνακας 5 *Αξιώματα OWL κλάσεων*

Αφηρημένη Σύνταξη	**Σύνταξη ΠΛ**	**Σημασιολογία**
Class(A partial C_1, …, C_n)	$A \sqsubseteq C_1 \sqcap \ldots \sqcap C_n$	$A^{\mathcal{I}} \subseteq C_1^{\mathcal{I}} \cap \ldots \cap C_n^{\mathcal{I}}$
Class(A complete C_1, …, C_n)	$A \equiv C_1 \sqcap \ldots \sqcap C_n$	$A^{\mathcal{I}} \subseteq C_1^{\mathcal{I}} \cap \ldots \cap C_n^{\mathcal{I}}$
EnumeratedClass(A o_1, …, o_n)	$A \equiv \{o_1\} \sqcup \ldots \sqcup \{o_n\}$	$(\{o_1\} \sqcup \ldots \sqcup \{o_n\})^{\mathcal{I}}=\{o_1^{\mathcal{I}},\ldots,o_n^{\mathcal{I}}\}$
SubClassOf(C_1, C_2)	$C_1 \sqsubseteq C_2$	$C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$
EquivalentClasses(C_1, …, C_n)	$C_1 \equiv \ldots \equiv C_n$	$C_1^{\mathcal{I}}=\ldots=C_n^{\mathcal{I}}$
DisjointClasses(C_1, …, C_n)	$C_i \sqcap C_j \sqsubseteq \bot$, $1 \leq i < j \leq n$	$C_i^{\mathcal{I}} \cap C_j^{\mathcal{I}} \subseteq \varnothing$

Στον πίνακα 6 παρουσιάζουμε την αντιστοιχία ανάμεσα στα αξιώματα ιδιοτήτων της OWL και αξιώματα ρόλων των Περιγραφικών Λογικών. Για ακόμα μια φορά παρατηρούμε ότι η OWL δεν παρέχει τίποτα περισσότερο από την εκφραστική δυνατότητα που μας παρέχει μια ΠΛ ακόμα και στις περιπτώσεις που κάτι τέτοιο δεν είναι άμεσα προφανές, όπως στην περίπτωση των συναρτησιακών ρόλων που εξηγήσαμε παραπάνω. Από τον πίνακα αυτό όμως παρατηρούμε και κάτι άλλο πολύ ενδιαφέρον. Τα αξιώματα ορισμού του πεδίου τιμών και του πεδίου ορισμού μιας ιδιότητας, τα οποία συναντήσαμε στη γλώσσα RDF-S, μπορούν και αυτά να κωδικοποιηθούν με τη χρήση αξιωμάτων Περιγραφικών Λογικών. Πιο συγκεκριμένα ένα αξίωμα ορισμού του πεδίου τιμών της μορφής ObjectProperty(R domain(C)) αντιστοιχεί στο αξίωμα υπαγωγής $\exists R.\top \sqsubseteq C$, ενώ ένα αξίωμα ορισμού του πεδίου τιμών της μορφής ObjectProperty(R range(C)) αντιστοιχεί στο αξίωμα υπαγωγής, $\top \sqsubseteq \forall R.C$. Διαισθητικά το πρώτο αξίωμα μας λέει ότι: «αν ένα αντικείμενο συνδέεται μέσω της σχέσης R με κάτι, τότε το αντικείμενο αυτό είναι τύπου C». Άρα αυτό το αξίωμα μας λέει ότι όλα τα άτομα τα οποία βρίσκονται στην πρώτη θέση ενός ζευγαριού (a,b) το οποίο ανήκει στη σχέση R θα είναι τύπου C. Το δεύτερο αξίωμα μας δηλώνει ότι «για όλα τα αντικείμενα του κόσμου, όποτε αυτά συνδέονται με κάποιο άλλο αντικείμενο μέσω της σχέσης R τότε το αντικείμενο αυτό είναι τύπου C». Άρα όλα τα άτομα τα οποία βρίσκονται στη δεύτερη θέση ενός ζευγαριού (a,b) το οποίο ανήκει στη σχέση R θα είναι τύπου C το οποίο αντιστοιχεί με τον ορισμό πεδίου τιμών.

Πίνακας 6 *Αξιώματα ιδιοτήτων αντικειμένων*

Αφηρημένη Σύνταξη	Σύνταξη ΠΛ	Σημασιολογία
SubPropertyOf(R_1, R_2)	$R \sqsubseteq S$	$R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$
EquivalentProperties(R_1, …, R_n)	$R_1 \equiv \ldots \equiv R_n$	$R_1^{\mathcal{I}} = \ldots = R_n^{\mathcal{I}}$
ObjectProperty(R super(R_1) … super(R_n)	$R \sqsubseteq R_i$	$R^{\mathcal{I}} \subseteq R_i^{\mathcal{I}}$
domain(C_1) … domain(C_n)	$\exists R.\top \sqsubseteq C_i$	$R^{\mathcal{I}} \subseteq C_i^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
range(C_1) … range(C_n)	$\top \sqsubseteq \forall R.C_i$	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times C_i^{\mathcal{I}}$
inverseOf(S)	$R \equiv S^-$	$R^{\mathcal{I}} = (S^-)^{\mathcal{I}}$
Symmetric	$R \equiv R^-$	$R^{\mathcal{I}} = (R^-)^{\mathcal{I}}$
Functional	$\top \sqsubseteq \leq 1 R$	$\forall a^{\mathcal{I}}.\#\{b \mid (a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}\} \leq 1$
InverseFunctional	$\top \sqsubseteq \leq 1 R^-$	$\forall a^{\mathcal{I}}.\#\{b \mid (a^{\mathcal{I}}, b^{\mathcal{I}}) \in (R^-)^{\mathcal{I}}\} \leq 1$
Transitive)	**Tr**(R)	$\{(a^{\mathcal{I}}, b^{\mathcal{I}}), (b^{\mathcal{I}}, c^{\mathcal{I}})\} \subseteq R^{\mathcal{I}} \rightarrow (a^{\mathcal{I}}, c^{\mathcal{I}}) \in R^{\mathcal{I}}$

Εν συνεχεία τα αξιώματα ατόμων παρουσιάζονται στον πίνακα 7.

Πίνακας 7 *Αξιώματα ατόμων*

Αφηρημένη Σύνταξη	Σύνταξη ΠΛ	Σημασιολογία
Individual(o type(C_1) … type(C_n)	$o:C_i$, $1 \leq i \leq n$	$o^{\mathcal{I}} \in C_i^{\mathcal{I}}$, $1 \leq i \leq n$
Value(R_1, o_1) … value(R_n, o_n)	$(o,o_i):C_i$, $1 \leq i \leq n$	$(o^{\mathcal{I}}, o_i^{\mathcal{I}}) \in C_i^{\mathcal{I}}$, $1 \leq i \leq n$
SameIndidivual(o_1, …, o_n)	$o_1 = \ldots = o_n$	$o_1^{\mathcal{I}} = \ldots = o_n^{\mathcal{I}}$
DifferentIndividuals(o_1, …, o_n)	$o_i \neq o_j$, $1 \leq i < j \leq n$	$o_i^{\mathcal{I}} \neq o_j^{\mathcal{I}}$, $1 \leq i < j \leq n$

Τέλος στον πίνακα 8 παρουσιάζουμε την αντιστοίχηση ανάμεσα στην αφηρημένη σύνταξη περιγραφών κλάσεων και αξιωμάτων ιδιοτήτων τα οποία σχετίζονται με τύπους δεδομένων. Έτσι λοιπόν η ιδιότητα T είναι μια ιδιότητα τύπων δεδομένων ενώ το σύμβολο d παριστάνει έναν τύπο δεδομένων.

Πίνακας 8 *Αξιώματα ιδιοτήτων τύπων δεδομένων*

Αφηρημένη Σύνταξη	Σύνταξη ΠΛ	Σημασιολογία
restriction(T someValuesFrom(d))	$\exists T.d$	$(\exists T.d)^{\mathcal{I}} = \{a^{\mathcal{I}} \mid \exists y.(a^{\mathcal{I}}, y) \in T^{\mathcal{I}} \& y \in d^{\mathbf{D}}\}$
restriction(T allValuesFrom(d))	$\forall T.d$	$(\forall T.d)^{\mathcal{I}} = \{a^{\mathcal{I}} \mid \forall y.(a^{\mathcal{I}}, y) \in T^{\mathcal{I}} \rightarrow y \in d^{\mathbf{D}}\}$

restriction(T minCardinality(n))	$\geq nT$	$(\geq nT)^{\mathcal{I}}=\{a^{\mathcal{I}} \mid \#\{y \mid (a^{\mathcal{I}},y)\in T^{\mathcal{I}}\}\geq n\}$
restriction(T maxCardinality(n))	$\leq nT$	$(\leq nT)^{\mathcal{I}}=\{a^{\mathcal{I}} \mid \#\{y \mid (a^{\mathcal{I}},y)\in T^{\mathcal{I}}\}\leq n\}$
DatatypeProperty(T super(T_1)…super(T_n)	$T\sqsubseteq T_i$	$T^{\mathcal{I}}\subseteq T_i^{\mathcal{I}}$
domain(C_1) … domain(C_n)	$\exists T.\top\sqsubseteq C_i$	$T^{\mathcal{I}}\subseteq C_i^{\mathcal{I}}\times\Delta_{\mathbf{D}}$
range(d_1) … range(d_n)	$\top\sqsubseteq\forall T.d_i$	$T^{\mathcal{I}}\subseteq\Delta_{\mathbf{D}}\times C_i^{\mathcal{I}}$
Functional)	$\top\sqsubseteq\leq 1T$	$\forall a^{\mathcal{I}}.\#\{y\mid(a^{\mathcal{I}},y)\in T^{\mathcal{I}}\}\leq 1$

Αναφορές

Beckett, D. (2003) RDF/XML Syntax Specification (Revised). World Wide Web Consortium, 10 October 2003 (work in progress). This version is http://www.w3.org/TR/2003/WD-rdf-syntax-grammar-20031010/. The latest version is http://www.w3.org/TR/rdf-syntax-grammar/

Bechhofer, S., van Harmelen F., Hendler, J., Horrocks, I., McGuinness., D.L., Patel-Schneider P.F., Stein, A.L., eds. (2004) OWL Web Ontology Language Reference. Available from http://www.w3.org/TR/owl-ref/.

Boris, M., (2005) On the Properties of Metamodeling in OWL. Proceedings of the 4th International Semantic Web Conference (ISWC 05)*, pp. 548-562 Galway, Ireland.*

Brickey, D., Guha, R.V. (2000) RDF Vocabulary Description Language 1.0: RDF Schema, W3C. Available at: http://www.w3.org/TR/2000/CR-rdf-schema-20000327.

Hayes, P. (2003) RDF Semantics. World Wide Web Consortium, 10 October 2003 (work in progress). This version is http://www.w3.org/TR/2003/WD-rdf-mt-20031010/. The latest version is http://www.w3.org/TR/rdf-mt/.

Horrocks, I., Patel-Schneider, P.F. (2003) Reducing OWL Entailment to Description Logic Satisfiability. Proceedings of the 2nd International Semantic Web Conference (ISWC '03).

Klyne, G., Carroll, J. (2003) Resource Description Framework (RDF): Concepts and Abstract Syntax. World Wide Web Consortium, (work in progress). This version is http://www.w3.org/TR/2003/WD-rdf-concepts-20031010/. The latest version is http://www.w3.org/TR/rdf-concepts/.

Lassila, O., Swick, R. (1999) W3C Resource Description Framework model and syntax specification. Available from http://www.w3.org/TR/RECrdf-syntax/

Patel-Schneider, P.F., Hayes, P., Horrocks, I. (2004) OWL Web Ontology Language Semantics and Abstract Syntax. W3C Recommendation available at, http://www.w3.org/TR/2004/REC-owl-semantics-20040210/

Διευθύνσεις URL

1. Η σελίδα για την RDF στην W3C: http://www.w3.org/RDF/
2. Η σελίδα για την OWL στην W3C: http://www.w3.org/OWL/
3. Το εργαλείο δημιουργίας οντολογιών OWL και RDF, Protégé: http://protege.stanford.edu/

Ακρωνύμια – Συντομεύσεις

RDF	Resource Description Framework
RDF-S	RDF-Schema
URI	Universal Resource Identifier
URIref	URI reference
OWL	Web Ontology Language

SWRL	Semantic Web Rule Language
RuleML	Rule Markup Language
RIF	Rule Interchange Format

Γλωσσάριο

abstract syntax	αφηρημένη σύνταξη
body	σώμα
complement	συμπλήρωμα
class	κλάση
datatype	τύπος δεδομένων
decidable	αποφασίσιμη
facts	γεγονότα
functional	συναρτησιακή
head	κεφαλή
heterogeneous	ετερογενές
individual	άτομο
interconnectivity	διασυνδεσιμότητα
interoperability	διαλειτουργικότητα
keyword	λέξη κλειδί
literal	λεκτικό
metamodeling	μετα-μοντελοποίηση
namespace	χώρος ονομάτων
predicate	κατηγόρημα
property	ιδιότητα
resource	πόρος
sentence	πρόταση
symmetric	συμμετρική
tag	ετικέτα
transitive	μεταβατικός
triple	τριάδα
undecidable	μη-αποφασίσιμή
union	ένωση
variable	μεταβλητή
vocabulary	λεξιλόγιο