

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΪΟΥ  
ΤΜΗΜΑ ΜΑΘΗΜΑΤΙΚΩΝ

ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΣΠΟΥΔΕΣ ΣΤΑ ΜΑΘΗΜΑΤΙΚΑ

---

# Οι Οντολογίες ως Εργαλείο των Εννοιολογικών Τεχνολογιών

---

Συγγραφέας:  
Παναγιώτης ΤΟΥΡΛΑΣ

Επιβλέπων:  
Παναγιώτης ΝΑΣΤΟΥ

27 Φεβρουαρίου 2021





Τριμελής Επιτροπή  
Βασίλειος Μεταφυσής, Καθηγητής  
Ιωάννης Σταματίου, Καθηγητής  
Παναγιώτης Νάστου, Επίκουρος Καθηγητής

Αφιερωμένη στη σύζυγό μου Όλγα,  
που με στήριξε με όλες της τις δυνάμεις.

Παναγιώτης Τούρλας,  
Σάμος 2021.

Ευχαριστώ τον επιβλέποντα καθηγητή μου, κύριο Νάστου, για το ενδιαφέρον και την εμπιστοσύνη που μου έδειξε ώστε να φέρω εις πέρας τη παρούσα μεταπτυχιακή εργασία.



# Περιεχόμενα

<b>1</b>	<b>Εννοιολογικές Τεχνολογίες</b>	<b>5</b>
1.1	Ο Παγκόσμιος Ιστός ως ιστός αρχείων . . . . .	5
1.2	Πληροφορία και γνώση στον Παγκόσμιο Ιστό . . . . .	6
1.3	Αρχιτεκτονική του Εννοιολογικού Ιστού . . . . .	8
1.3.1	Κωδικοποίηση κειμένου και συντακτικοί κανόνες . . . . .	9
1.3.2	Ανταλλαγή δεδομένων και ταξινομίες . . . . .	10
1.3.3	Ανάζητηση πληροφοριών σε δεδομένα RDF . . . . .	15
<b>2</b>	<b>Οντολογίες</b>	<b>19</b>
2.1	Περιγραφική Λογική . . . . .	20
2.1.1	RBox . . . . .	20
2.1.2	TBox . . . . .	21
2.1.3	ABox . . . . .	22
2.1.4	Βάσεις Γνώσης . . . . .	23
2.2	Η δομή μιας οντολογίας . . . . .	24
2.2.1	Άτομα . . . . .	24
2.2.2	Κλάσεις . . . . .	24
2.2.3	Σχέσεις . . . . .	25
2.3	Η γλώσσα OWL2 . . . . .	25
2.3.1	Κλάσεις στην OWL2 . . . . .	25
2.3.2	Μεταβατικότητα και ανακλαστικότητα . . . . .	26
2.3.3	Ξένες κλάσεις . . . . .	27

2.3.4	Ορισμός κλάσεων μέσω απαρίθμησης . . . . .	28
2.3.5	Σχέσεις στην OWL2 . . . . .	29
2.3.6	Σχέσεις ατόμων - δεδομένων . . . . .	29
2.3.7	Ισότητα ατόμων . . . . .	30
2.3.8	Ενώσεις, τομές και συμπληρώματα . . . . .	31
2.3.9	Υπαρξιακοί και καθολικοί περιορισμοί . . . . .	32
2.3.10	Χαρακτηριστικά σχέσεων . . . . .	33
2.4	Μια οντολογία για φωτογραφικές μηχανές . . . . .	34
2.4.1	Συνολική δομή . . . . .	35
2.4.2	Κλάσεις . . . . .	36
2.4.3	Ιδιότητες τύπων δεδομένων . . . . .	38
2.4.4	Ιδιότητες αντικειμένων . . . . .	39
2.4.5	Στιγμιότυπα της οντολογίας . . . . .	40
<b>3</b>	<b>Διαχείριση Γνώσης και Οντολογίες</b>	<b>43</b>
3.1	Εικονικές Επιχειρήσεις . . . . .	43
3.2	Μηχανισμοί Ελέγχου Πρόσβασης . . . . .	43
3.3	Μοντέλο Διαχείρισης Γνώσης . . . . .	44
3.3.1	Επίπεδα Οργάνωσης Γνώσης . . . . .	46
3.3.2	Λεξιλόγιο του μοντέλου . . . . .	49
3.3.3	Σχέσεις του μοντέλου . . . . .	49
3.3.4	Πολιτική Ελέγχου Πρόσβασης Γνώσης . . . . .	52



# Κατάλογος Σχημάτων

1.1	Η παραγωγή γνώσης από δεδομένα στον Π.Ι. . . . .	7
1.2	Αρχιτεκτονική του Εννοιολογικού Ιστού . . . . .	9
1.3	Γράφος μιας ταξινόμιας σε RDFS . . . . .	13
2.1	Κλάσεις, ιδιότητες και σχέσεις της οντολογίας φωτογραφικών μηχανών	35
3.1	Αρχιτεκτονική του RBAC . . . . .	44
3.2	Σχεδιασμός και παραγωγή σε μια εικονική επιχείρηση . . . . .	46
3.3	Επίπεδα οργάνωσης γνώσης σε μια εικονική επιχείρηση . . . . .	48
3.4	Διάδοση εξουσιοδότησης μέσω ισοδύναμων εννοιών . . . . .	51
3.5	Διάδοση εξουσιοδότησης μέσω υποκλάσεων . . . . .	51
3.6	Διαχωρισμός VEACP/EMACPs . . . . .	52



# Εισαγωγή

Οι Εννοιολογικές Τεχνολογίες είναι ένα σύνολο από προγραμματιστικά εργαλεία με στόχο να επιτρέψει στους υπολογιστές να επεξεργάζονται τόσο τα διαθέσιμα δεδομένα όσο και το σημασιολογικό τους περιεχόμενο. Στον πυρήνα αυτών των εργαλείων συναντάμε τις οντολογίες, περιγραφές ενός πεδίου ενδιαφέροντος οι οποίες:

- παρέχουν μια αυστηρή λογική θεμελίωση για τη διαδικασία περιγραφής
- επιτρέπουν την παραγωγή νέας γνώσης από την ήδη υπάρχουσα με αλγοριθμικό τρόπο.

Το πρώτο πεδίο ανάπτυξης και εφαρμογής των Εννοιολογικών Τεχνολογιών προέκυψε στις αρχές της δεκαετίας του 2000 και ήταν ο λεγόμενος Εννοιολογικός Ιστός. Ουσιαστικά αποτελούσε μια προσπάθεια διεύρυνσης του Παγκόσμιου Ιστού ώστε να μπορούμε να συνδέσουμε δεδομένα με βάση την έννοια που καλύπτουν και ταυτόχρονα οι συνδέσεις να είναι προσβάσιμες από υπολογιστές. Μια τέτοια μετάβαση θα άλλαζε δραστικά τη μορφή του Παγκόσμιου Ιστού καθώς θα επέτρεπε την αυτοματοποίηση διαδικασιών που μέχρι πρότινος απαιτούσαν ανθρώπινη παρέμβαση. Κλασικό παράδειγμα αποτελεί το [1], όπου περιγράφεται πως θα μπορούσε ένας “προσωπικός ηλεκτρονικός βοηθός” να κανονίσει μια συνάντηση εκ μέρους μας σε συνεννόηση με άλλους “βοηθούς” τρίτων.

Από τα υποθετικά σενάρια των προηγούμενων δεκαετιών έχουμε φτάσει στις σημερινές εφαρμογές ευρείας κλίμακας όπως η Siri<sup>1</sup> και ο Google Assistant<sup>2</sup> που επιτρέπουν την αλληλεπίδραση μέσω της ομιλίας και μπορούν να “κατανοήσουν” τα αιτήματα ενός χρήστη.

Επιπλέον όμως, οι Εννοιολογικές Τεχνολογίες βρίσκουν απήχηση και στην βιομηχανία καθώς επιτρέπουν την αποτελεσματικότερη οργάνωση και κυρίως, τη μεταφορά γνώσης. Αυτό τις καθιστά ένα εργαλείο κλειδί για τις σύγχρονες γραμμές παραγωγής αφού συχνά απαιτείται η σύνθεση ενός τελικού προϊόντος από διάφορα επιμέρους κομμάτια και ανεξάρτητες διαδικασίες. Συνεπώς ο πλήρης συντονισμός όλων των εμπλεκομένων αποκτά μεγάλη σημασία.

---

<sup>1</sup><https://www.apple.com/siri/>

<sup>2</sup><https://assistant.google.com/>

Ένα ακόμα πεδίο με σημαντικές εφαρμογές αποτελούν οι Επιστήμες Υγείας. Αυτό οφείλεται στον τεράστιο όγκο δεδομένων που είναι διαθέσιμος και αυξάνεται συνεχώς. Ενδεικτικά [2], στις Η.Π.Α. ήδη από το 2010 ο όγκος δεδομένων σχετικά με το σύστημα υγείας έφτανε τα 150 exabytes<sup>1</sup>. Δυστυχώς όμως τα δεδομένα δεν ισοδυναμούν με γνώση. Παρόλα αυτά το χάσμα μπορεί να γεφυρωθεί από οντολογίες, όπως για παράδειγμα η SNOMED CT [3]. Έτσι οι επαγγελματίες της υγείας μπορούν να έχουν ένα κοινό σημείο αναφοράς σχετικά με την ορολογία που χρησιμοποιούν και, το πιο σημαντικό, κοινές ερμηνείες της ορολογίας αυτής.

## Δομή της εργασίας

Η παρούσα εργασία χωρίζεται σε τρία κεφάλαια.

Στο πρώτο κεφάλαιο παρουσιάζονται τα βασικά εργαλεία των Εννοιολογικών Τεχνολογιών [4] [5] μέσα από την εφαρμογή τους στον Εννοιολογικό Ιστό. Συγκεκριμένα δίνεται μια περιγραφή:

- των URI, τα οποία μας επιτρέπουν να δίνουμε μοναδικά ονόματα σε αντικείμενα και οντότητες του Εννοιολογικού Ιστού
- του μοντέλου RDF [6] για την έκφραση σχέσεων και ιδιοτήτων μέσα από τριάδες της μορφής “υποκείμενο-κατηγορημα-αντικείμενο”
- της γλώσσας SPARQL [7] για την αναζήτηση πληροφοριών σε βάσεις δεδομένων που αποθηκεύονται τριάδες RDF
- της γλώσσας RDFS [8] η οποία ορίζει ένα λεξιλόγιο για την περιγραφή κλάσεων και σχέσεων ανάμεσα σε αυτές
- της γλώσσας OWL [9] η οποία βασίζεται στην Περιγραφική Λογική για τη δημιουργία οντολογιών και των κανόνων που διέπουν τη σημασιολογία τους

Στο δεύτερο κεφάλαιο παρουσιάζεται:

- Η Περιγραφική Λογική *SROIQ* [10], ένα υποσύνολο της Πρωτοβάθμιας Λογικής που μας επιτρέπει να περιγράψουμε λεπτομερώς τις σχέσεις ανάμεσα σε αντικείμενα και τις ιδιότητές τους.
- Η γλώσσα OWL και συγκεκριμένα η πιο πρόσφατη εκδοχή της, OWL2 η οποία δανείζεται τους μηχανισμούς περιγραφής της *SROIQ*. Μέσω της OWL2 μπορούμε να κατασκευάζουμε οντολογίες και επιπλέον να παράγουμε νέα γνώση αλγοριθμικά από τις περιγραφές μας.

<sup>1</sup> 1 exabyte = 10<sup>18</sup> bytes

Τέλος, το τρίτο κεφάλαιο ασχολείται με την εφαρμογή των οντολογιών στο περιβάλλον μιας εικονικής επιχείρησης. Συγκεκριμένα, παρουσιάζεται ένας μηχανισμός εξουσιοδότησης ο οποίος βασίζεται στους τύπους δεδομένων, τους ρόλους των μελών και τις δραστηριότητες σχεδιασμού και παραγωγής προκειμένου να καθορίσει τις δυνατότητες διαμοιρασμού γνώσης ανάμεσα στα μέλη.



# Κεφάλαιο 1

## Εννοιολογικές Τεχνολογίες

Οι Εννοιολογικές Τεχνολογίες αποτελούνται από μια σειρά εργαλείων που διευρύνουν τις δυνατότητες του Παγκόσμιου Ιστού (World Wide Web). Μέσα από τη χρήση τους επιδιώκεται η σύνδεση διαφορετικών ειδών δεδομένων με βάση τις έννοιες που περιγράφουν. Για να καταλάβουμε τι σημαίνει αυτό, θα πρέπει αρχικά να κατανοήσουμε τα δομικά μέρη του Παγκόσμιου Ιστού.

### 1.1 Ο Παγκόσμιος Ιστός ως ιστός αρχείων

Ο Παγκόσμιος Ιστός (Π.Ι. εν συντομία) είναι ένα εξαιρετικά επιτυχημένο εργαλείο που βασίστηκε σε μια απλή ιδέα: την πρόσβαση σε αρχεία που περιέχουν συνδέσμους σε άλλα αρχεία και επιτρέπουν την απευθείας μετάβαση σε αυτά. Η ιδέα αυτή, γνωστή ως *υπερκείμενο*<sup>1</sup> ήταν ήδη γνωστή από την δεκαετία του 1960 αλλά οι πραγματικές της δυνατότητες αξιοποιήθηκαν το 1989 από τον Sir Tim Berners-Lee με τη δημιουργία της πρώτης ιστοσελίδας<sup>2</sup> του Π.Ι. Ένα λεπτό σημείο που θα πρέπει να αναφερθεί εδώ είναι η διάκριση μεταξύ Διαδικτύου και Π.Ι.: το Διαδίκτυο αποτελεί ένα παγκόσμιο δίκτυο συσκευών και υποδικτύων που βασίζονται στα πρωτόκολλα TCP/IP<sup>3</sup> για την μεταξύ τους επικοινωνία ενώ ο Π.Ι. συνδυάζει την υποδομή του Διαδικτύου με μια σειρά από άλλες τεχνολογίες διασύνδεσης:

- το Σύστημα Ονοματοδοσίας Διαδικτύου (γνωστό και ως Domain Name System ή DNS<sup>4</sup>) για τον συσχετισμό των domain names με τις διευθύνσεις IP.
- το πρωτόκολλο HTTP<sup>5</sup> (HyperText Transfer Protocol) το οποίο καθορίζει τον

---

<sup>1</sup><https://en.wikipedia.org/wiki/Hypertext#History>

<sup>2</sup><http://info.cern.ch/hypertext/WWW/TheProject.html>

<sup>3</sup><https://datatracker.ietf.org/doc/html/rfc1180>

<sup>4</sup><https://datatracker.ietf.org/doc/html/rfc1034>

<sup>5</sup><https://www.ietf.org/rfc/rfc2616.html>

τρόπο επικοινωνίας μεταξύ ενός διακομιστή και ενός πελάτη στον Π.Ι.

- τους Ενιαίους Εντοπιστές Πόρων (γνωστούς και ως Uniform Resource Locators ή URLs<sup>1</sup>), που επιτρέπουν την ταυτοποίηση αρχείων με μοναδικό τρόπο στον Π.Ι.

**Παρατήρηση 1.1.1** Τα URLs αποτελούν υποσύνολο των Ενιαίων Αναγνωριστικών Πόρων (γνωστοί και ως Uniform Resource Identifiers ή URIs<sup>2</sup>). Τα URIs είναι ένα σημαντικό εργαλείο των Εννοιολογικών Τεχνολογιών καθώς χρησιμοποιούνται για να περιγράψουν όχι μόνο τοποθεσίες ενός δικτύου αλλά και τοποθεσίες του πραγματικού κόσμου, ανθρώπους, αντικείμενα ή ακόμα και αφηρημένες έννοιες.

## 1.2 Πληροφορία και γνώση στον Παγκόσμιο Ιστό

Ο Π.Ι. ουσιαστικά υλοποιεί ένα κανάλι μεταφοράς δεδομένων μεταξύ ανθρώπων. Εννοούμε δηλαδή ότι ο δημιουργός μιας σελίδας τη “δημοσιεύει” σε μια τοποθεσία την οποία μπορεί να επισκεφθεί οποιοσδήποτε τρίτος έχει στη διάθεσή του έναν απλό φυλλομετρητή (ή αλλιώς web browser). Αυτό το μοντέλο επικοινωνίας ονομάζεται *αρχιτεκτονική πελάτη/διακομιστή*<sup>3</sup>. Εδώ ο ρόλος των τεχνολογικών υποδομών είναι να καταστήσουν δυνατή την πρόσβαση σε μια τοποθεσία και να παρουσιάσουν τα δεδομένα της. Πέρα από αυτό, κάθε διαδικασία κατανόησης και παραγωγής γνώσης γίνεται από τον άνθρωπο-αποδέκτη (βλ. Σχήμα 1.1).

Με παρόμοιο τρόπο θα μπορούσαμε να πούμε ότι δουλεύει και ένας πίνακας ανακοινώσεων στον πραγματικό κόσμο.

**Παράδειγμα 1.2.1** Έστω ότι κάποιος αναρτά μια αγγελία πώλησης μιας ηλεκτρικής κιθάρας στον πίνακα ανακοινώσεων του διαδρόμου ενός πανεπιστημίου. Τότε η αγγελία αυτή είναι αυτομάτως διαθέσιμη σε οποιονδήποτε γνωρίζει την τοποθεσία του πίνακα στο πανεπιστήμιο και μπορεί να διαβάσει τη γλώσσα στην οποία είναι γραμμένη η αγγελία.

Στο παράδειγμα 1.2.1 κάνουμε μια έμμεση παραδοχή: αγνοούμε την ανάγκη να δώσουμε μια περιγραφή της κιθάρας σαν μουσικό όργανο καθώς τη θεωρούμε κοινή γνώση. Τι θα γινόταν όμως αν στη θέση της κιθάρας είχαμε ένα didgeridoo<sup>4</sup>;

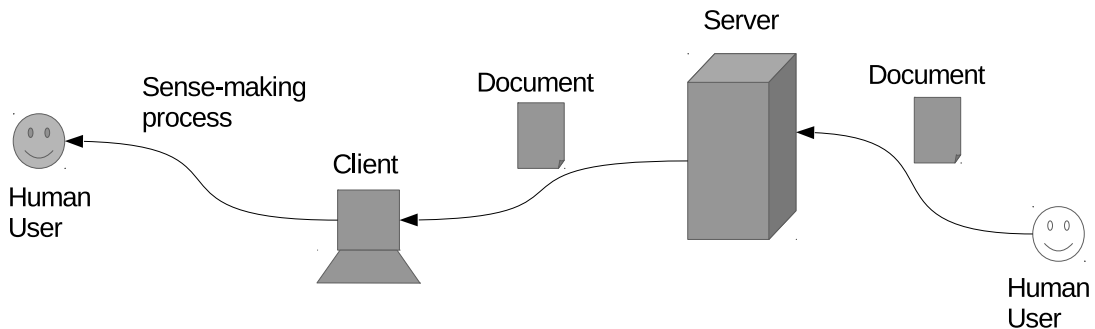
<sup>1</sup><https://url.spec.whatwg.org/#concept-url>

<sup>2</sup>[https://www.w3.org/Addressing/URL/URI\\_Overview.html](https://www.w3.org/Addressing/URL/URI_Overview.html)

<sup>3</sup>[https://en.wikipedia.org/wiki/Client-server\\_model](https://en.wikipedia.org/wiki/Client-server_model)

<sup>4</sup><https://en.wikipedia.org/wiki/Didgeridoo>





Σχήμα 1.1: Η παραγωγή γνώσης από δεδομένα στον Π.Ι.

Σίγουρα θα δυσκόλευε την πώληση το γεγονός ότι είναι λίγοι οι υποψήφιοι αγοραστές που πιθανώς να γνωρίζουν κάποια πράγματα για αυτό το εξωτικό μουσικό όργανο. Θα μπορούσαμε όμως να βελτιώσουμε την αγγελία μας συνοδεύοντας την και από μια σελίδα με ιστορικές πληροφορίες και το πως μπορεί να παίζει κανείς μουσική με ένα didgeridoo. Τότε ο αναγνώστης της αγγελίας μας αποκτά μια σφαιρικότερη εικόνα του τι είναι αυτό το μουσικό όργανο. Κατά κάποιο τρόπο, μπορούμε να πούμε ότι η αγγελία μας δεν μεταφέρει απλώς δεδομένα όπως το όνομα ενός οργάνου και την τιμή του αλλά εσωκλείει και "γνώση" σχετική με το αντικείμενο που μας ενδιαφέρει.

Επιστρέφοντας στον κόσμο του Διαδικτύου, τον ίδιο στόχο πετυχαίνουμε με τις Εννοιολογικές Τεχνολογίες. Εμπλουτίζουμε τα διαθέσιμα δεδομένα κατάλληλα ώστε να παρουσιάζουν ευρύτερες έννοιες. Μια μεγάλη διαφορά όμως είναι μια επιπλέον επιδίωξή μας: Τα δεδομένα δεν θέλουμε απλά να παρουσιάζουν και έννοιες, θέλουμε ταυτόχρονα να είναι επεξεργάσιμα από μηχανές. Έτσι η διαδικασία παραγωγής γνώσης μπορεί, σε κάποιο βαθμό, να αυτοματοποιηθεί.

**Παρατήρηση 1.2.1** Μιλάμε για μερική αυτοματοποίηση της παραγωγής γνώσης για τους εξής λόγους:

- Η φυσική γλώσσα έχει τόσο μεγάλες εκφραστικές δυνατότητες που μπορούν να οδηγήσουν σε παράδοξα και σε μη-αποφάνσιμα προβλήματα.<sup>1</sup> Άρα μια

<sup>1</sup>Αποφάνσιμο (decidable) ονομάζεται ένα πρόβλημα για το οποίο υπάρχει αλγόριθμος που οδηγεί πάντα σε μια σωστή απάντηση, είτε θετική ή αρνητική.

απόπειρα να κατανοήσουμε πλήρως αλγοριθμικά την φυσική γλώσσα είναι εκ προοιμίου καταδικασμένη.

- Στη φυσική γλώσσα υπάρχει (συνήθως) μια κοινή γνώση μεταξύ των συνομιλητών, η οποία και καθορίζει τα συμφραζόμενα. Συμπληρώνουμε δηλαδή τα νοηματικά κενά των πληροφοριών που δεχόμαστε έτσι ώστε να μπορούμε να τις κατανοήσουμε. Η διαδικασία αυτή συμβαίνει στο μυαλό του κάθε συνομιλητή δυναμικά και μπορεί να οδηγήσει σε διαφορετική κατανόηση.

Οι παραπάνω παράγοντες οδηγούν στην ανάγκη για κοινά εργαλεία και πρότυπα προκειμένου να υλοποιήσουμε τις Εννοιολογικές Τεχνολογίες. Ένα πρώτο πεδίο εφαρμογής που θα εξετάσουμε είναι ο Εννοιολογικός Ιστός.

**Παρατήρηση 1.2.2** Υπάρχει διάκριση ανάμεσα στους όρους “Εννοιολογικές Τεχνολογίες” και “Εννοιολογικός Ιστός”. Ο Εννοιολογικός Ιστός δεν αποτελεί την μοναδική εφαρμογή αυτών των τεχνολογιών. Άλλα παραδείγματα εφαρμογών συναντώνται:

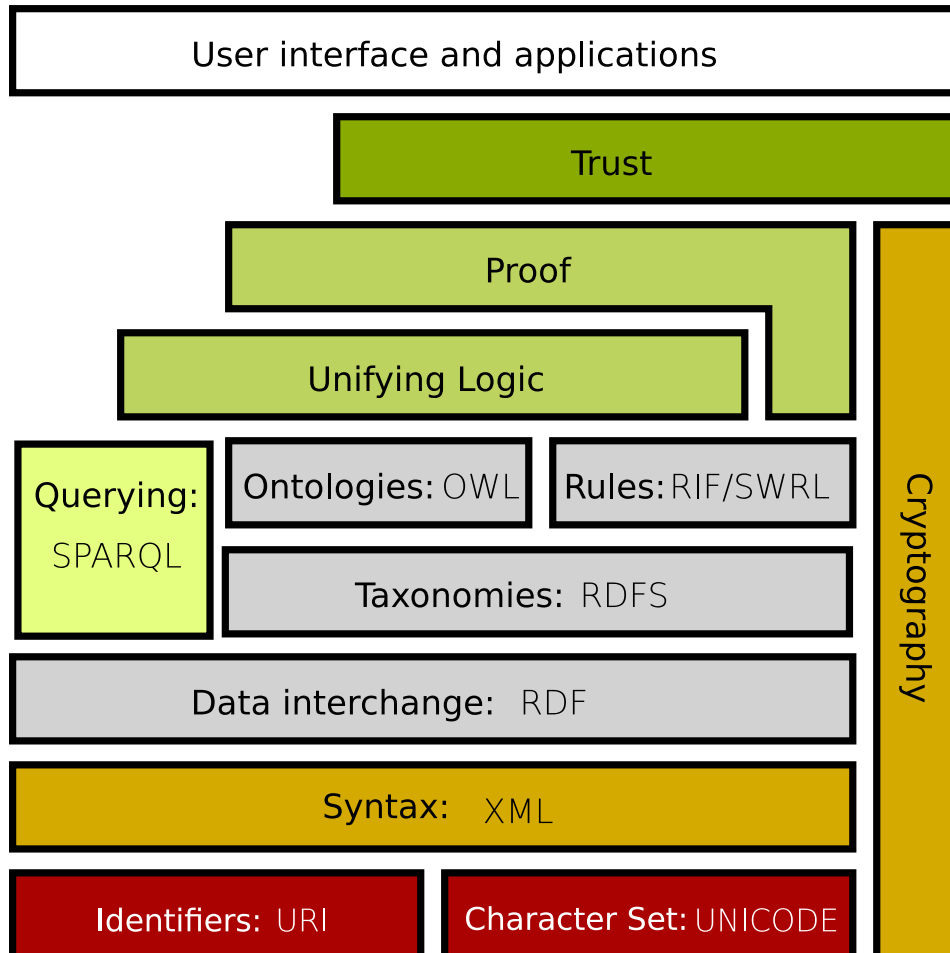
- στη Βιομηχανία, όπου οι Εννοιολογικές Τεχνολογίες μπορούν να χρησιμοποιηθούν για να καθοριστεί ένα πεδίο γνώσης με σαφή τρόπο για όλους τους ενδιαφερόμενους
- στις Εννοιολογικές Βάσεις Δεδομένων<sup>1</sup>, οι οποίες επιτρέπουν την αποθήκευση των σχέσεων που συνδέουν τα δεδομένα και τη σύνθεση νέων σχέσεων για την εξαγωγή γνώσης.

### 1.3 Αρχιτεκτονική του Εννοιολογικού Ιστού

Ο Εννοιολογικός Ιστός αποτελεί μια πολυεπίπεδη αρχιτεκτονική. Ξεκινώντας από τα χαμηλότερα στρώματα που καθορίζουν τους συντακτικούς κανόνες και την κατάλληλη κωδικοποίηση (UTF-8, URI, XML), περνάμε σε αυτά που καθορίζουν την ανταλλαγή δεδομένων (RDF) και την βασική σημασιολογική τους ταξινόμηση (RDFS). Έτσι έχουμε μια υποδομή πάνω στην οποία μπορούμε να επεκτείνουμε την σημασιολογική ταξινόμηση με τρόπο που να επιτρέπει τις αλγοριθμικές εφαρμογές (OWL), να ορίζουμε σύνολα κανόνων (RIF) και να αναζητούμε πληροφορίες (SPARQL). Στο τέλος τοποθετούμε εργαλεία που ενοποιούν τα κατώτερα στρώματα, μας παρέχουν τη δυνατότητα να πιστοποιούμε τις πηγές των δεδομένων και καθορίζουν τον βαθμό αξιοπιστίας τους. Πρέπει να σημειωθεί επιπλέον και η χρήση εργαλείων της Κρυπτογραφίας όπως το πρωτόκολλο HTTPS και οι ψηφιακές υπογραφές προκειμένου να εξασφαλίζεται η ασφαλής επικοινωνία και η ακεραιότητα των δεδομένων.

<sup>1</sup><https://skr3.nlm.nih.gov/SemMedDB/>

Η σύνθεση της αρχιτεκτονικής που περιγράψαμε φαίνεται στο σχήμα 1.2. Στη συνέχεια θα δώσουμε μια συνοπτική περιγραφή για κάθε ένα από τα στρώματα που αναφέραμε.



Σχήμα 1.2: Αρχιτεκτονική του Εννοιολογικού Ιστού

(πηγή: [https://commons.wikimedia.org/wiki/File:Semantic\\_web\\_stack.svg](https://commons.wikimedia.org/wiki/File:Semantic_web_stack.svg))

#### 1.3.1 Κωδικοποίηση κειμένου και συντακτικοί κανόνες

Η κωδικοποίηση UTF-8 βασίζεται στο πρότυπο Unicode<sup>1</sup> το οποίο προσφέρει 143,859 χαρακτήρες. Το βασικότερο πλεονέκτημα της UTF-8 είναι η ευρύτατη υποστήριξη που έχει παγκοσμίως, διευκολύνοντας τη διασυνδεσιμότητα στον Εννοιολογικό Ιστό. Τα URIs (βλ. και Παρατήρηση 1.1.1) επιτρέπουν να περιγράψουμε

<sup>1</sup><https://home.unicode.org/>

με μοναδικό τρόπο οτιδήποτε. Έτσι γίνεται ευκολότερος ο συνδυασμός δεδομένων από διαφορετικές πηγές.

Οι βασικοί κανόνες σύνταξης προκύπτουν από την γλώσσα σήμανσης XML (Extensible Markup Language)<sup>1</sup> και περιλαμβάνουν:

- Ετικέτες (tags) της μορφής `<X>`, `</X>` και `<X/>`
- Στοιχεία (elements) που αποτελούνται από ετικέτες και κάποιο περιεχόμενο όπως `<greeting>Hello, world!</greeting>` ή ετικέτες χωρίς περιεχόμενο όπως `<greet/>`.
- Χαρακτηριστικά (attributes) που συνοδεύουν μια ετικέτα όπως `<greeting wave=true>Hello!</greeting>` ή `<greet name=John/>`

### 1.3.2 Ανταλλαγή δεδομένων και ταξινομίες

Το μοντέλο RDF επιτρέπει τον ορισμό οντοτήτων στον Παγκόσμιο Ιστό καθώς και των σχέσεων και των ιδιοτήτων τους. Βασίζεται σε τριάδες της μορφής 'υποκείμενο - κατηγορία - αντικείμενο' (subject - predicate - object). Στη συνέχεια παρουσιάζεται το συντακτικό RDF/XML, το οποίο βασίζεται στους κανόνες σύνταξης της XML για την αναπαράσταση τριάδων. Επιπλέον αξιοποιεί τα URIs για τις ονομασίες οντοτήτων.

**Παρατήρηση 1.3.1** Η μορφή RDF/XML είναι η πρώτη που δημιουργήθηκε για το μοντέλο RDF αλλά όχι η μοναδική. Σήμερα υπάρχουν διάφορες εναλλακτικές μορφές όπως:

- Turtle
- N-Triples
- N-Quads
- JSON-LD
- RDF/JSON

Στόχος τους είναι μια αναπαράσταση πιο φιλική προς τον άνθρωπο, η μείωση των χαρακτήρων που απαιτούνται για μια πλήρη αναπαράσταση αλλά και οι δυνατότητες αποθήκευσης σε βάσεις ειδικά σχεδιασμένες για RDF τριάδες (triplestores<sup>2</sup>)

<sup>1</sup><https://www.w3.org/TR/xml/>

<sup>2</sup><https://en.wikipedia.org/wiki/Triplestore>

Στη συνέχεια παρουσιάζεται ένα απλό παράδειγμα που ξεκινά από μια δήλωση σε φυσική γλώσσα, την τριάδα στην οποία αντιστοιχεί και την έκφρασή της σε RDF/XML.

**Παράδειγμα 1.3.1** Έστω η ακόλουθη δήλωση σε φυσική γλώσσα (για ευκολία θα χρησιμοποιήσουμε την αγγλική γλώσσα):

*http://www.example.org/index.html has a creation date whose value is August 16, 1999.*

Αρχικά ορίζουμε κάποια προθέματα τα οποία μας επιτρέπουν να γράψουμε URIs σε συντομευμένη μορφή και να καθορίσουμε τους τύπους δεδομένων που χρησιμοποιούνται.

```
1 prefix "rdf:"
2 namespace URI: http://www.w3.org/1999/02/22-rdf-syntax-ns#
3
4 prefix "ex:"
5 namespace URI: http://www.example.org/
6
7 prefix "extterms:"
8 namespace URI: http://www.example.org/terms/
9
10 prefix "xsd:"
11 namespace URI: http://www.w3.org/2001/XMLSchema#
```

Στη συνέχεια κατασκευάζουμε μια τριάδα από τα δεδομένα μας:

*ex:index.html extterms:creation-date "1999-08-16"^^xsd:date*

Το υποκείμενο (ιστοσελίδα) φέρει μια ιδιότητα (ημερομηνία δημιουργίας) με μια τιμή (1999-08-16). Επίσης ο τύπος αυτής της τιμής (ημερομηνία) καθορίζεται ξεχωριστά (*^^xsd:date*). Η τελική μορφή σε RDF/XML είναι:

```
1 <?xml version="1.0"?>
2 <rdf:RDF
3   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4   xmlns:extterms="http://www.example.org/terms/">
5     <rdf:Description
6       rdf:about="http://www.example.org/index.html">
7       <extterms:creation-date
8         rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
9         1999-08-16
10      </extterms:creation-date>
11    </rdf:Description>
12 </rdf:RDF>
```

Τα βασικότερα σημεία της υλοποίησης είναι:

- Γραμμές 3 και 4 : Ορίζονται τα namespaces που θα χρησιμοποιήσουμε.
- Γραμμή 5: Ξεκινά ο ορισμός μιας τριάδας με την ετικέτα `<rdf:Description>`
- Γραμμή 6: Ορίζεται το υποκείμενο της τριάδας με την ετικέτα `<rdf:about>`
- Γραμμή 7: Ξεκινά ο ορισμός του κατηγορήματος της τριάδας με την ετικέτα `<exterms:creation-date>`
- Γραμμή 8: Το χαρακτηριστικό `<rdf:datatype>` καθορίζει τον τύπο του στοιχείου εντός της ετικέτας.

Η χρήση του μοντέλου RDF μας επιτρέπει να συνδέουμε δεδομένα μέσα από τριάδες. Αυτές οι συνδέσεις όμως είναι ανεπαρκείς χωρίς κοινό σημασιολογικό περιεχόμενο για όλους τους χρήστες. Προκειμένου να επιτευχθεί κάτι τέτοιο, χρησιμοποιούμε την γλώσσα RDFS. Η γλώσσα αυτή εφοδιάζει το μοντέλο RDF με ένα λεξιλόγιο κατάλληλο για να ορίσουμε ταξινομίες, δηλαδή να ομαδοποιήσουμε οντότητες με κοινά χαρακτηριστικά. Τα βασικά στοιχεία της RDFS αποτυπώνονται στους παρακάτω ορισμούς:

**Ορισμός 1.3.1** Με τον όρο κλάση αναφερόμαστε σε οποιαδήποτε οντότητα περιγράφεται από τριάδες RDF. Τα μέλη μιας κλάσης ονομάζονται στιγμιότυπα και φέρουν την ιδιότητα `rdf:type`. Επιπλέον ορίζουμε ως `rdfs:Class` την κλάση όλων των RDFS κλάσεων.

**Παρατήρηση 1.3.2** Κατά σύμβαση, οι κλάσεις θα ξεκινάνε με κεφαλαίο γράμμα ενώ οι ιδιότητες με μικρό.

**Ορισμός 1.3.2** Αν μια κλάση *X* είναι υποκλάση της *X* τότε κάθε στιγμιότυπο της *X* είναι και στιγμιότυπο της *X*. Η σχέση αυτή δηλώνεται μέσα από την ιδιότητα `rdfs:subClassOf`.

**Ορισμός 1.3.3** Η κλάση `rdf:Property` αποτελεί την κλάση όλων των ιδιοτήτων σε RDF και είναι στιγμιότυπο της `rdfs:Class`.

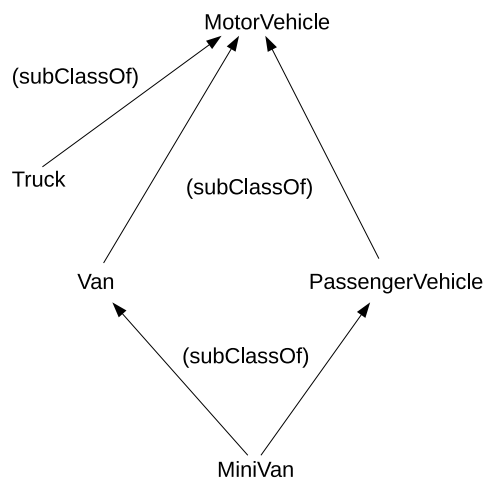
**Ορισμός 1.3.4** Το πεδίο ορισμού *rdfs:domain* καθορίζει την κλάση ενός υποκειμένου σε μια τριάδα RDF. Αποτελεί στιγμότυπο της *rdfs:Property*. Μια τριάδα της μορφής *P rdfs:domain C* σημαίνει ότι κάθε υποκείμενο που προηγείται της *P* σε μια δοσμένη τριάδα, είναι στιγμότυπο της *C*.

**Ορισμός 1.3.5** Το πεδίο τιμών *rdfs:range* καθορίζει την κλάση ενός αντικειμένου σε μια τριάδα RDF. Αποτελεί στιγμότυπο της *rdfs:Property*. Μια τριάδα της μορφής *P rdfs:range C* σημαίνει ότι κάθε αντικείμενο που έπεται της *P* σε μια δοσμένη τριάδα, είναι στιγμότυπο της *C*.

Σε αντίθεση με την έννοια της κλάσης όπως τη συναντάμε στον Αντικειμενοστραφή Προγραμματισμό, μια κλάση στην RDFS δεν ορίζεται από τις ιδιότητες των στιγμιότυπων της. Οι ιδιότητες είναι αυτές που ορίζονται με βάση τις κλάσεις που αφορούν. Το όφελος αυτής της προσέγγισης έγκειται στο ότι μπορούν να οριστούν νέες ιδιότητες με τα ίδια πεδία ορισμού και τιμών χωρίς να χρειάζεται τροποποίηση αυτών που ήδη υπάρχουν. Αυτό κάνει ευκολότερη την προσθήκη νέων περιγραφών. Η διαφορά μπορεί να γίνει καλύτερα αντιληπτή μέσα από το παράδειγμα που ακολουθεί.

**Παράδειγμα 1.3.2** Έστω η ιδιότητα *Author* με πεδίο ορισμού την κλάση *Document* και πεδίο τιμών την κλάση *Person*. Το αντικειμενοστραφές αντίστοιχο θα ήταν μια κλάση *Book* με μια ιδιότητα *Author* τύπου *Person*.

Στη συνέχεια παρουσιάζεται μια απλή ταξινόμια οχημάτων. Τα προθέματα των URI βασίζονται σε όσα έχουν οριστεί στο παράδειγμα 1.3.1 με μια επιπλέον προσθήκη για το πρόθεμα *rdfs*.



Σχήμα 1.3: Γράφος μιας ταξινόμιας σε RDFS

**Παράδειγμα 1.3.3** Ορίζουμε την ακόλουθη ταξινόμια σε μορφή τριάδων:

```

1 prefix "rdfs:"
2 namespace URI: http://www.w3.org/2000/01/rdf-schema#
3
4 ex:MotorVehicle      rdfs:type          rdfs:Class
5 ex:PassengerVehicle  rdfs:type          rdfs:Class
6 ex:Van               rdfs:type          rdfs:Class
7 ex:Truck             rdfs:type          rdfs:Class
8 ex:MiniVan          rdfs:type          rdfs:Class
9
10 ex:PassengerVehicle  rdfs:subClassOf    ex:MotorVehicle
11 ex:Van               rdfs:subClassOf    ex:MotorVehicle
12 ex:Truck             rdfs:subClassOf    ex:MotorVehicle
13
14 ex:MiniVan           rdfs:subClassOf    ex:Van
15 ex:MiniVan           rdfs:subClassOf    ex:PassengerVehicle

```

Η παραπάνω ταξινόμια μπορεί να γραφτεί σε μορφή RDF/XML ως εξής:

```

1 <?xml version="1.0"?>
2 <rdf:RDF
3   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
5   xml:base="http://example.org/schemas/vehicles">
6
7   <rdf:Description rdf:about="MotorVehicle">
8     <rdf:type
9       rdf:resource="http://www.w3.org/rdf-schema#Class"/>
10  </rdf:Description>
11
12  <rdf:Description rdf:about="PassengerVehicle">
13    <rdf:type
14      rdf:resource="http://www.w3.org/rdf-schema#Class"/>
15    <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
16  </rdf:Description>
17
18  <rdf:Description rdf:about="Truck">
19    <rdf:type
20      rdf:resource="http://www.w3.org/rdf-schema#Class"/>
21    <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
22  </rdf:Description>
23
24  <rdf:Description rdf:about="Van">
25    <rdf:type
26      rdf:resource="http://www.w3.org/rdf-schema#Class"/>
27    <rdfs:subClassOf rdf:resource="#MotorVehicle"/>

```



```
28 </rdf:Description>
29
30 <rdf:Description rdf:about="MiniVan">
31     <rdf:type
32         rdf:resource="http://www.w3.org/rdf-schema#Class"/>
33     <rdfs:subClassOf rdf:resource="#Van"/>
34     <rdfs:subClassOf rdf:resource="#PassengerVehicle"/>
35 </rdf:Description>
36
37 </rdf:RDF>
```

### 1.3.3 Ανάζητηση πληροφοριών σε δεδομένα RDF

Η γλώσσα SPARQL μας επιτρέπει να διατυπώνουμε ερωτήματα σε βάσεις δεδομένων που χρησιμοποιούν το μοντέλο RDF. Το κυριότερο πλεονέκτημά της είναι ότι μας βοηθά να εκμεταλλευόμαστε τις σχέσεις που υπάρχουν μεταξύ των διαθέσιμων δεδομένων. Έτσι ένα ερώτημα μπορεί να επιστρέψει τόσο άμεσα αποτελέσματα, όπως δηλαδή θα συνέβαινε σε μια σχεσιακή βάση δεδομένων, αλλά και έμμεσα αποτελέσματα τα οποία προκύπτουν από δεδομένα με κοινό σημασιολογικό περιεχόμενο.

Τα ερωτήματα σε SPARQL περιέχουν μοτίβα τριάδων. Τα μοτίβα αυτά μοιάζουν με τριάδες RDF με τη διαφορά ότι υποκείμενο, κατηγορημα και αντικείμενο μπορεί να είναι μεταβλητές. Στο παρακάτω ορισμό παρουσιάζονται τα βασικά μέρη ενός ερωτήματος SPARQL.

**Ορισμός 1.3.6** *Οι μεταβλητές της γλώσσας SPARQL συμβολίζονται με το πρόθεμα "?" ακολουθούμενο από το όνομα της μεταβλητής. Η λέξη-κλειδί SELECT χρησιμοποιείται για τον καθορισμό των μεταβλητών ενός ερωτήματος ενώ η λέξη-κλειδί WHERE χρησιμοποιείται για τον καθορισμό του μοτίβου που αναζητούμε. Με την λέξη-κλειδί PREFIX μπορούμε να ορίσουμε προθέματα για τις RDF τριάδες ενός ερωτήματος.*

Ακολουθεί ένα απλό παράδειγμα ενός ερωτήματος SPARQL και μιας τριάδας RDF που το ικανοποιεί.

**Παράδειγμα 1.3.4** Έστω η τριάδα:

```
ex:book ex:terms:title "a book about SPARQL"
```

Το ερώτημα

```

1 PREFIX ex: <http://www.example.org/67>
2 PREFIX exterm:s: <http://www.example.org/terms>
3
4 SELECT ?title
5 WHERE
6 {
7     ex:book exterm:s:title ?title
8 }

```

ικανοποιείται από την δοσμένη τριάδα ενώ το αποτέλεσμα που επιστρέφεται έχει τη μορφή

```

1 -----
2 | title
3 =====
4 | "a book about SPARQL"
5 -----

```

Όταν ορίζουμε ένα μοτίβο σε SPARQL τα αποδεκτά αποτελέσματα το ικανοποιούν πλήρως. Μπορούμε να δούμε τι σημαίνει αυτό στο εξής παράδειγμα.

### Παράδειγμα 1.3.5 Έστω οι εξής τριάδες RDF:

```

1 ex:a      exterm:s:name      "John Doe"
2 ex:a      exterm:s:email     "jd@example.com"
3 ex:b      exterm:s:name      "Mike Brown"
4 ex:b      exterm:s:email     "mike@brown.org"
5 ex:c      exterm:s:email     "dbook@example.org"

```

Τότε το ερώτημα

```

1 PREFIX ex: <http://www.example.org/>
2 PREFIX exterm:s: <http://www.example.org/terms>
3
4 SELECT ?name ?email
5 WHERE
6 {
7     ?x exterm:s:name ?name
8     ?x exterm:s:email ?email
9 }

```

μας επιστρέφει τα ζεύγη

```

1 -----
2 | name | email
3 =====
4 | "John Doe" | "jd@example.com"

```

```
5 | "Mike Brown" | "mike@brown.org"
6 -----
```

Η τριάδα `ex:c exterm:email "dbook@example.org"` απουσιάζει από τα αποτελέσματα καθώς δεν υπάρχει αντίστοιχη τριάδα `ex:c exterm:name "..."` και συνεπώς δεν ικανοποιεί πλήρως το μοτίβο του ερωτήματος.

Το σημαντικότερο χαρακτηριστικό της γλώσσας SPARQL είναι η δυνατότητα να παίρνουμε αποτελέσματα που δεν είναι απευθείας καταχωρημένα στη βάση αλλά προκύπτουν από τη δομή των δεδομένων αξιοποιώντας την RDFS.

#### **Παράδειγμα 1.3.6** Έστω οι εξής τριάδες RDF:

```
1 prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
2 prefix ex: <http://www.example.org/>
3 prefix exterm: <http://www.example.org/terms>
4 prefix dc: <http://purl.org/dc/elements/1.1/>
5
6 ex:composer rdfs:subPropertyOf dc:creator
7 ex:photographer rdfs:subPropertyOf dc:creator
8
9 ex:file02924 ex:fileName "9894397.mp3"
10 ex:file02924 ex:composer "Charles Ives"
11
12 ex:file74395 ex:fileName "884930.mp3"
13 ex:file74395 ex:composer "Eric Satie"
14
15 ex:file69383 ex:fileName "119933.jpg"
16 ex:file69383 ex:photographer "Diane Arbus"
```

#### *Τότε το ερώτημα*

```
1 PREFIX ex: <http://www.example.org/>
2 PREFIX exterm: <http://www.example.org/terms>
3 PREFIX dc: <http://purl.org/dc/elements/1.1/>
4
5 SELECT ?filename ?creator
6 WHERE
7 {
8     ?resource ex:filename ?filename
9     ?resource dc:creator ?creator
10 }
```

#### *μας επιστρέφει τα ζεύγη*

```
1 filename | creator
```

```
2 =====
3 "119933.mp3" | "Diane Arbus"
4 "9894397.mp3" | "Charles Ives"
5 "884930.mp3" | "Eric Satie"
```

Βλέπουμε ότι ενώ καμία τριάδα δεν φέρει την ιδιότητα `dc:creator`, έχουν οριστεί δυο υποκλάσεις της, η `ex:composer` και η `ex:photographer`. Το ερώτημα του παραδείγματος μας επιτρέπει να χρησιμοποιήσουμε αυτές τις σχέσεις παίρνοντας αποτελέσματα που καλύπτονται από μια κοινή έννοια.

Έχοντας παρουσιάσει τα εργαλεία με τα οποία μπορούμε να ορίσουμε απλές ταξινομίες μπορούμε να προχωρήσουμε στις οντολογίες, οι οποίες μας παρέχουν ένα αυστηρό λογικό υπόβαθρο για να θέτουμε αξιώματα. Έτσι γίνεται εφικτό να αποτυπώνουμε με πιο λεπτομερή τρόπο τις ιδιότητες των κλάσεων και των σχέσεων που ορίζουμε.

## Κεφάλαιο 2

### Οντολογίες

Οι οντολογίες αποτελούν μια μετεξέλιξη των εννοιολογικών γράφων [11] και είναι το εργαλείο με τις μεγαλύτερες εκφραστικές δυνατότητες στις Εννοιολογικές Τεχνολογίες. Σύμφωνα με τον Gruber [12], μια οντολογία αποτελεί έναν *τυπικό και σαφή προσδιορισμό μιας κοινής σύλληψης*. Από τον παραπάνω ορισμό μπορούμε να εξάγουμε τα κύρια χαρακτηριστικά μιας οντολογίας:

- Τυπικότητα, η οποία μας επιτρέπει να ορίζουμε έννοιες με τρόπο που τις καθιστά επεξεργάσιμες από μηχανές. Επιπλέον μας επιτρέπει να επιβεβαιώσουμε την εγκυρότητα μιας έκφρασης αλγοριθμικά, βασιζόμενοι μόνο στη δομή της.
- Σαφήνεια, που σημαίνει ότι αποδίδουμε το επιθυμητό νόημα χωρίς την ανάγκη για επιπλέον πληροφορίες.
- Κοινή σύλληψη, καθώς μια οντολογία έχει αξία εφόσον περιγράφει έννοιες αποδεκτές από όλους τους χρήστες της.

Οι οντολογίες φέρουν ένα αυστηρό λογικό υπόβαθρο. Αυτό είναι αναγκαίο προκειμένου να περιγράφουν ένα πεδίο ενδιαφέροντος με ακρίβεια και να εξασφαλίζεται ότι οι αλγόριθμοι για τον χειρισμό τους συμπεριφέρονται με προβλέψιμο και ενιαίο τρόπο. Μια ακόμα σημαντική δυνατότητα των οντολογιών είναι το γεγονός ότι μας επιτρέπουν να παράγουμε νέα γνώση αυτόματα με τους λεγόμενους *reasoners*<sup>1</sup>, μια κατηγορία λογισμικού που βρίσκει τις λογικές συνέπειες των αξιωμάτων μιας οντολογίας. Είναι προφανές ότι για να φέρει πρακτική αξία, η διαδικασία αυτή πρέπει να τερματίζει πάντοτε και σε πολυωνυμικό χρόνο. Κάτι τέτοιο είναι εφικτό αν περιοριστούμε σε ένα υποσύνολο της Πρωτοβάθμιας Λογικής, γνωστό ως Περιγραφική Λογική.

Στο κεφάλαιο αυτό θα παρουσιαστεί η Περιγραφική Λογική *SRQIQ* και η γλώσσα

---

<sup>1</sup>[https://en.wikipedia.org/wiki/Semantic\\_reasoner](https://en.wikipedia.org/wiki/Semantic_reasoner)

OWL (Web Ontology Language), η οποία αποτελεί την πρόταση του W3C (World Wide Web Consortium) για τη δημιουργία οντολογιών.

## 2.1 Περιγραφική Λογική

Η Περιγραφική Λογική είναι ένας φορμαλισμός αναπαράστασης της γνώσης με κυριότερο χαρακτηριστικό το γεγονός ότι τα προβλήματα λογικής συνεπαγωγής που μπορούμε να εκφράσουμε σε αυτήν είναι αποφάνσιμα. Δεν πρόκειται για μια κλειστή γλώσσα αλλά για μια συλλογή “διαλέκτων” με διαφορετικές εκφραστικές δυνατότητες οι οποίες μπορούν να συνθέσουν μια γλώσσα με τα επιθυμητά χαρακτηριστικά. Η γλώσσα OWL βασίζεται στην Περιγραφική Λογική *SROIQ* της οποίας τα βασικά στοιχεία παρουσιάζονται στη συνέχεια.

**Ορισμός 2.1.1** *Ορίζουμε τα εξής, ξένα μεταξύ τους, σύνολα:*

- Το σύνολο  $N_I$  των ονομάτων ατόμων που συμβολίζουν οντότητες.
- Το σύνολο  $N_C$  των ονομάτων εννοιών που αναφέρονται σε κλάσεις οντοτήτων.
- Το σύνολο  $N_R$  των ονομάτων ιδιοτήτων, δηλαδή των διμελών σχέσεων μεταξύ ατόμων ή κλάσεων.

Το σύνολο  $\{N_I, N_C, N_R\}$  ονομάζεται *λεξιλόγιο*.

Έχοντας στη διάθεσή μας ένα λεξιλόγιο, μπορούμε να προχωρήσουμε στον ορισμό των εννοιών RBox, TBox και ABox μιας βάσης γνώσης που προκύπτει από μια Περιγραφική Λογική.

### 2.1.1 RBox

Ένα RBox είναι ένα σύνολο που περιγράφει το πως συνδέονται οι ιδιότητες που έχουμε ορίσει. Για τον τυπικό του ορισμό θα χρειαστούμε κάποια προαπαιτούμενα.

**Ορισμός 2.1.2** *Ορίζουμε ως ιδιότητα:*

- την καθολική ιδιότητα  $u$ , δηλαδή την διμελή σχέση που συνδέει οποιαδήποτε δυο άτομα
- κάθε στοιχείο  $r, r^- \in N_R$ , όπου με  $r^-$  συμβολίζουμε την αντίστροφη μιας δοσμένης ιδιότητας

**Ορισμός 2.1.3** Ένα αξίωμα εγκλεισμού ιδιότητας είναι μια δήλωση της μορφής  $r_1 \circ r_2 \dots \circ r_n \sqsubseteq r$ , όπου τα  $r_1, r_2, \dots, r_n, r$  είναι ιδιότητες και με  $\circ$  συμβολίζουμε την σύνθεση. Στην περίπτωση όπου  $n = 1$  έχουμε απλό εγκλεισμό ιδιότητας, δηλαδή  $r \sqsubseteq s$ .

Ένα απλό παράδειγμα τέτοιου εγκλεισμού είναι το εξής:

$$\text{fatherOf} \sqsubseteq \text{childOf}^-$$

Δηλαδή η αντίστροφη της ιδιότητας  $\text{childOf}$  (η ιδιότητα του γονέα) συμπεριλαμβάνει την ιδιότητα του πατέρα.

**Ορισμός 2.1.4** Ένα πεπερασμένο σύνολο από αξιώματα εγκλεισμού ιδιοτήτων ονομάζεται ιεραρχία ιδιοτήτων.

**Ορισμός 2.1.5** Ένα χαρακτηριστικό ιδιότητας είναι μια δήλωση της μορφής:

- $\text{Ref}(r)$ , για ανακλαστικές σχέσεις
- $\text{Asy}(r)$ , για μη συμμετρικές σχέσεις
- $\text{Dis}(r, s)$ , για ξένες μεταξύ τους σχέσεις

Συγκεκριμένα, η  $\text{Dis}(r, s)$  ισχύει αν για κάθε ζεύγος ατόμων που ισχύει η  $r$  δεν ισχύει η  $s$ .

**Ορισμός 2.1.6** Ένα  $R\text{Box } \mathcal{R}$  είναι η ένωση μιας ιεραρχίας ιδιοτήτων και πεπερασμένων χαρακτηριστικών ιδιοτήτων.

## 2.1.2 TBox

Για τον τυπικό ορισμό ενός TBox θα χρειαστούμε τους παρακάτω ορισμούς.

**Ορισμός 2.1.7** Ορίζουμε ως έννοιες (εννοώντας τις κλάσεις οντοτήτων):

- κάθε όνομα  $C \in N_C$
- τα σύμβολα  $\perp, \top$  για την κενή έννοια (δηλαδή την κλάση που δεν έχει κανένα στιγμιότυπο) και την καθολική έννοια (δηλαδή την έννοια που περιλαμβάνει όλες τις άλλες), αντίστοιχα

- κάθε έκφραση της μορφής  $\{a_1, a_2, \dots, a_n\}$  όπου  $\{a_i\} \subseteq N_I$  (ονομαστικές έννοιες)

Επιπλέον,

- αν  $C, D$  είναι έννοιες, τότε είναι και οι  $\neg C$  (άρνηση),  $C \sqcap D$  (τομή) και  $C \sqcup D$  (ένωση)
- αν  $r$  είναι μια ιδιότητα και  $C$  μια έννοια, τότε οι  $\exists r.C$  (υπαρξιακή ποσοτικοποίηση) και  $\forall r.C$  (καθολική ποσοτικοποίηση) είναι επίσης έννοιες
- αν  $r$  είναι μια ιδιότητα,  $n \in \mathbf{N}$  και  $C$  μια έννοια, τότε οι εκφράσεις  $\exists r.\text{Self}$  (περιορισμός σε αυτοαναφορικές σχέσεις),  $\geq nr.C$  (ελάχιστος περιορισμός) και  $\leq nr.C$  (μέγιστος περιορισμός) είναι επίσης έννοιες

**Ορισμός 2.1.8** Ορίζουμε  $\mathbb{C}$  να είναι το σύνολο όλων των εννοιών που προέρχονται από ένα σύνολο ονομάτων εννοιών  $N_C$ .

**Ορισμός 2.1.9** Έστω  $C, D \in \mathbb{C}$ . Η έκφραση  $C \sqsubseteq D$  ονομάζεται αξίωμα εγκλεισμού έννοιας και σημαίνει ότι η έννοια  $D$  συμπεριλαμβάνει την έννοια  $C$ .

Μπορούμε έτσι να δώσουμε τον ορισμό του TBox.

**Ορισμός 2.1.10** Έστω  $\mathcal{T}$  ένα πεπερασμένο σύνολο αξιωμάτων εγκλεισμού εννοιών. Το  $\mathcal{T}$  ονομάζεται TBox της SROIQ.

### 2.1.3 ABox

Μέσω των RBox και TBox μπορούμε να εκφράσουμε χαρακτηριστικά των ιδιοτήτων και των κλάσεων αντίστοιχα. Για τα χαρακτηριστικά ατόμων θα χρειαστούμε του ακόλουθους ορισμούς.

**Ορισμός 2.1.11** Έστω  $a$  ένα άτομο και  $C$  μια έννοια. Τότε με  $C(a)$  θα συμβολίζουμε έναν εννοιολογικό ισχυρισμό, δηλαδή το γεγονός ότι το άτομο  $a$  είναι στιγμιότυπο της έννοιας  $C$ .

**Ορισμός 2.1.12** Έστω  $a, b$  άτομα και  $r$  μια ιδιότητα. Τότε με  $r(a, b)$  θα συμβολίζουμε τον ισχυρισμό ιδιότητας, δηλαδή το γεγονός ότι το άτομο  $a$  συνδέεται με το άτομο  $b$  μέσω της σχέσης  $r$ . Αντίστοιχα, με  $\neg r(a, b)$  θα συμβολίζουμε την άρνηση της  $r$ .



**Ορισμός 2.1.13** Έστω  $a, b$  άτομα. Τότε με  $a \approx b$  θα συμβολίζουμε τον ισχυρισμό ατομικής ισότητας, δηλαδή το γεγονός ότι δυο ονόματα αφορούν την ίδια οντότητα.

**Παρατήρηση 2.1.1** Η ατομική ισότητα μπορεί να φανεί χρήσιμη όταν συνδυάζουμε γνώση για ένα πεδίο ενδιαφέροντος από διαφορετικές πηγές. Αν γαι παράδειγμα συνδυάζαμε δυο οντολογίες για τα ζώα της Αφρικής και στη μια συναντούσαμε το είδος *Acinonyx jubatus* ενώ στην άλλη το είδος *Acinonyx venator* θα ίσχυε ότι *Acinonyx venator*  $\approx$  *Acinonyx jubatus* καθώς και τα δυο αποτελούν ονόματα του τσιτάχ.

Οι παραπάνω ορισμοί συνθέτουν τους λεγόμενους ατομικούς ισχυρισμούς. Βάσει αυτών, μπορούμε να δώσουμε και τον ορισμό του ABox.

**Ορισμός 2.1.14** Ένα πεπερασμένο σύνολο  $\mathcal{A}$  από ατομικούς ισχυρισμούς ονομάζεται ABox. Επιπλέον, το  $\mathcal{A}$  ονομάζεται εκτατικά ανηγμένο αν περιέχει μόνο ονόματα εννοιών και ονόματα ιδιοτήτων.

## 2.1.4 Βάσεις Γνώσης

Έχοντας καλύψει όλα τα δομικά χαρακτηριστικά της γλώσσας *SROIQ* μπορούμε να ορίσουμε τις βάσεις γνώσης που μπορούμε να κατασκευάσουμε.

**Ορισμός 2.1.15** Μια *SROIQ* βάση γνώσης  $\mathcal{KB}$  αποτελεί την ένωση ενός ενός RBox  $\mathcal{R}$ , ενός ενός TBox  $\mathcal{T}$  και ενός ενός ABox  $\mathcal{A}$ . Τα στοιχεία της βάσης γνώσης ονομάζονται αξιώματα. Ακόμα, συμβολίζουμε με  $N_I(\mathcal{KB})$ ,  $N_C(\mathcal{KB})$  και  $N_R(\mathcal{KB})$  τα ονόματα ατόμων, εννοιών και ιδιοτήτων της  $\mathcal{KB}$ , αντίστοιχα.

Ένα απλό παράδειγμα μιας *SROIQ* βάσης γνώσης είναι το εξής.

**Παράδειγμα 2.1.1** Έστω η βάση γνώσης  $\mathcal{KB}$ :

RBox  $\mathcal{R}$

- $\text{owns} \sqsubseteq \text{caresFor} \Rightarrow \text{"If somebody owns something, they care for it."}$

TBox  $\mathcal{T}$

- $\text{Healthy} \sqsubseteq \neg \text{Dead} \Rightarrow \text{"Healthy beings are not dead."}$
- $\text{Cat} \sqsubseteq \text{Alive} \sqcup \text{Dead} \Rightarrow \text{"Every cat is dead or alive."}$

- $\text{HappyCatOwner} \sqsubseteq \exists \text{owns.Cat} \sqcap \forall \text{caresFor.Healthy} \Rightarrow \text{"A happy cat owner owns a cat and all beings he cares for are healthy."}$

$\text{ABox } \mathcal{A}$

- $\text{HappyCatOwner}(\text{schrödinger}) \Rightarrow \text{"Schrödinger is happy cat owner."}$

## 2.2 Η δομή μιας οντολογίας

Ανεξάρτητα από το πεδίο εφαρμογής και την γλώσσα που επιλέγεται, όλες οι οντολογίες μοιράζονται κάποια βασικά μέρη. Αυτό αποτελεί απόρροια της στενής σχέσης οντολογιών και Περιγραφικής Λογικής, από την οποία αντλούνται αυτούσιοι οι μηχανισμοί περιγραφής ιδιοτήτων. Οι όποιες διαφορές και προσθήκες στις οντολογίες απλά εξασφαλίζουν την προσβασιμότητα από μηχανές. Στη συνέχεια παρουσιάζονται τα άτομα, οι κλάσεις και οι σχέσεις, αναπόσπαστα κομμάτια κάθε οντολογίας.

### 2.2.1 Άτομα

Με τον όρο *άτομα* εννοούμε οποιαδήποτε οντότητα θέλουμε να συμπεριλάβουμε στην περιγραφή μας, όπως για παράδειγμα 'Γιάννης', 'ρολόι' ή 'Αθήνα'. Τα άτομα μιας οντολογίας αντιστοιχίζονται στο σύνολο  $N_I$  των ονομάτων ατόμων της Περιγραφικής Λογικής. Τα άτομα χρησιμεύουν ως 'ετικέτες' που ξεχωρίζουν τα αντικείμενα και δεν επηρεάζουν την λογική δομή μιας οντολογίας.

Συχνά όμως διαφορετικά άτομα μπορεί να αντιστοιχίζονται στην ίδια οντότητα του πραγματικού κόσμου, χωρίς να γνωρίζουμε την αντιστοιχία αυτή. Αυτός είναι και ο λόγος που στις οντολογίες υιοθετούμε την υπόθεση του 'ανοικτού σύμπαντος'<sup>1</sup>, όπου μια δήλωση μπορεί να είναι αληθής ακόμα και αν δεν το γνωρίζουμε. Αντίθετα, η υπόθεση του 'κλειστού σύμπαντος' συνεπάγεται ότι έχουμε πλήρη γνώση κάθε αληθούς δήλωσης άρα κάθε άλλη δήλωση είναι λανθασμένη. Η υπόθεση του ανοικτού σύμπαντος είναι θεμελιώδης για τις οντολογίες καθώς χωρίς αυτή θα ήταν αδύνατο να εξάγουμε νέα γνώση. Εκτός αυτού, εκφράζει το γεγονός ότι οι οντολογίες συχνά χρησιμοποιούνται για την περιγραφή ενός διαρκώς μεταβαλλόμενου πεδίου γνώσης όπου είναι αδύνατο να έχουμε πλήρη εποπτεία.

### 2.2.2 Κλάσεις

Οι κλάσεις αποτελούν συλλογές αντικειμένων. Οι συλλογές αυτές προκύπτουν είτε μέσα από την απαρίθμηση των μελών τους (δηλαδή μέσω ενός *εκτατικού* ορισμού)

<sup>1</sup> [https://en.wikipedia.org/wiki/Open-world\\_assumption](https://en.wikipedia.org/wiki/Open-world_assumption)

είτε μέσα από κάποιες ιδιότητες που οφείλουν να έχουν τα μέλη μιας κλάσης (δηλαδή μέσω ενός *εντατικού* ορισμού) Οι κλάσεις αντιστοιχίζονται στο σύνολο  $N_C$  των ονομάτων εννοιών της Περιγραφικής Λογικής. Ακόμα, οι κλάσεις μας επιτρέπουν τη δημιουργία ιεραρχικών δομών μέσα από εγκλεισμούς, αποτελώντας μια επέκταση των δυνατοτήτων που συναντάμε στην γλώσσα RDFS. Στις οντολογίες η έννοια της κλάσης αντιμετωπίζεται όπως και στις ταξινομίες (βλ. παράδειγμα 1.3.3). Μια κλάση δηλαδή δεν ορίζεται από τις ιδιότητες των στιγμιοτύπων της αλλά οι ιδιότητες είναι αυτές που ορίζονται με βάση τις κλάσεις που αφορούν.

### 2.2.3 Σχέσεις

Με τον όρο σχέσεις εννοούμε τις διμελείς σχέσεις μεταξύ οντοτήτων, στις οποίες αποδίδουν ιδιότητες. Οι σχέσεις σε μια οντολογία είναι το αντίστοιχο του συνόλου  $N_R$  των ονομάτων ιδιοτήτων στην Περιγραφική Λογική. Όμοια με τις ιεραρχίες κλάσεων, μπορούμε να κατασκευάσουμε ιεραρχίες ιδιοτήτων. Επιπλέον μπορούμε να προσδώσουμε στις σχέσεις αυτές τυπικά χαρακτηριστικά των διμελών σχέσεων όπως συμμετρία ή ανακλαστικότητα αλλά και να εισάγουμε καθολικούς ή υπαρξιακούς περιορισμούς στο πως συνδέονται τα μέλη μιας σχέσης. Τέλος, σημειώνεται ότι η ανακλαστικότητα αποτελεί το μέσο για να ορίσουμε αυτοαναφορικές ιδιότητες, περιοριζόμενη όμως από τους κανόνες της Περιγραφικής Λογικής.

## 2.3 Η γλώσσα OWL2

Η γλώσσα OWL αποτελεί το πιο διαδεδομένο εργαλείο για την κατασκευή οντολογιών. Βασίζεται στα κατώτερα στρώματα των Εννοιολογικών Τεχνολογιών (URI, XML και RDFS, βλ. σχήμα 1.2) για τον καθορισμό της συντακτικής της δομής ενώ η Περιγραφική Λογική δίνει στη γλώσσα την τυπική σημασιολογία και τους κανόνες συνεπαγωγής. Από το 2009 η δεύτερη έκδοση της OWL, γνωστή και ως OWL2, είναι διαθέσιμη. Η έκδοση αυτή βασίζεται στην Περιγραφική Λογική *SROIQ* και φέρει τη μέγιστη δυνατή εκφραστικότητα διατηρώντας την πολυωνυμική πολυπλοκότητα και την αποφανσιμότητα. Στη συνέχεια παρουσιάζονται τα δομικά στοιχεία της OWL2.

### 2.3.1 Κλάσεις στην OWL2

Οι κλάσεις στην OWL2 ακολουθούν τη λογική της RDFS δίνοντας τη δυνατότητα οργάνωσης μέσω υποκλάσεων και δηλώνοντας τις συλλογές ατόμων με κοινά χαρακτηριστικά. Τα άτομα μπορούν να ανήκουν ταυτόχρονα σε πολλές διαφορετικές κλάσεις. Μπορούμε να δηλώσουμε τις κλάσεις στις οποίες ανήκει ένα άτομο μέσα από περιγραφές της μορφής:

```
<(class) rdf:about=(individual)/>
```

Στην περίπτωση που θέλουμε να δηλώσουμε την υποκλάση μιας δοσμένης κλάσης χρησιμοποιούμε μια περιγραφή της μορφής:

```
<owl:Class rdf:about=(child class)>
    <rdfs:subClassOf rdf:resource=(parent class)/>
</owl:Class>
```

**Παράδειγμα 2.3.1** Έστω ότι θέλουμε να δώσουμε μια περιγραφή σε OWL2 για το άτομο με το όνομα "Mary". Θα μπορούσαμε αρχικά να δηλώσουμε ότι ανήκει στις κλάσεις "Person" και "Woman":

```
<Person rdf:about="Mary"/>
<Woman rdf:about="Mary"/>
```

Μπορούμε ακόμα να διατυπώσουμε τη σχέση που συνδέει αυτές τις κλάσεις μέσα από την δήλωση:

```
<owl:Class rdf:about="Woman">
    <rdfs:subClassOf rdf:resource="Person"/>
</owl:Class>
```

Μια τέτοια δήλωση επιτρέπει σε έναν *reasoner* να εξαγάγει το συμπέρασμα ότι κάθε άτομο που ανήκει στην κλάση "Woman" ανήκει και στην κλάση "Person".

### 2.3.2 Μεταβατικότητα και ανακλαστικότητα

Μέσα από τον ορισμό υποκλάσεων μπορούμε να δημιουργούμε αλυσίδες που ξεκινούν από μια ειδική έννοια καταλήγοντας σε μια γενικότερη. Αυτό οφείλεται στην *μεταβατικότητα* των υποκλάσεων. Επεκτείνοντας το παράδειγμα 2.3.1, η δήλωση

```
<owl:Class rdf:about="Mother">
    <rdfs:subClassOf rdf:resource="Woman"/>
</owl:Class>
```

σημαίνει ότι ένα άτομο που ανήκει στην κλάση "Mother", ανήκει στην κλάση "Person".

Μια ακόμα ιδιότητα των υποκλάσεων είναι η *ανακλαστικότητα*, δηλαδή το ότι κάθε κλάση αποτελεί υποκλάση του εαυτού της. Αυτό μπορεί να αιτιολογηθεί εύκολα αν σκεφτούμε τις κλάσεις σαν σύνολα και συνεπώς κάθε σύνολο ως υποσύνολο του εαυτού του.

Στις περιγραφές μας μπορεί συχνά να χρειαστεί να ορίσουμε διαφορετικές κλάσεις με τα ίδια όμως μέλη. Αυτό αντικατοπτρίζει το γεγονός ότι μια έννοια μπορεί να

είναι γνωστή με διαφορετικά ονόματα. Αυτού του είδους τη σημασιολογική ισότητα μπορούμε να την δείξουμε μέσα από μια δήλωση της μορφής

```
<owl:Class rdf:about=(class A)>
  <owl:equivalentClass rdf:resource=(class B) />
</owl:Class>
```

και ένα απλό παράδειγμα εφαρμογής της είναι η ισοδυναμία των κλάσεων "Person" και "Human" :

```
<owl:Class rdf:about="Person">
  <owl:equivalentClass rdf:resource="Human" />
</owl:Class>
```

**Παρατήρηση 2.3.1** Μπορούμε να δούμε την ισότητα κλάσεων ως ισότητα συνόλων. Έτσι η προηγούμενη δήλωση γίνεται ισοδύναμη με το ζεύγος δηλώσεων:

```
<owl:Class rdf:about=(class A)>
  <rdfs:subClassOf rdf:resource=(class B) />
</owl:Class>
<owl:Class rdf:about=(class B)>
  <rdfs:subClassOf rdf:resource=(class A) />
</owl:Class>
```

### 2.3.3 Ξένες κλάσεις

Υπάρχουν περιπτώσεις όπου ένα άτομο είναι αδύνατο να ανήκει ταυτόχρονα σε δυο δοσμένες κλάσεις. Μπορούμε να καθορίσουμε κλάσεις που είναι ξένες μεταξύ τους μέσα από μια δήλωση της μορφής:

```
<owl:AllDisjointClasses>
  <owl:members rdf:parseType="Collection">
    <owl:Class rdf:about=(class A) />
    <owl:Class rdf:about=(class B) />
  </owl:members>
</owl:AllDisjointClasses>
```

Πρέπει να σημειωθεί ότι μια τέτοια δήλωση μας δίνει πληροφορίες και για τις υποκλάσεις που σχετίζονται με τις δοσμένες κλάσεις. Για παράδειγμα, η δήλωση

```
<owl:AllDisjointClasses>
  <owl:members rdf:parseType="Collection">
    <owl:Class rdf:about="Woman" />
    <owl:Class rdf:about="Man" />
  </owl:members>
</owl:AllDisjointClasses>
```

```

    </owl:members>
</owl:AllDisjointClasses>

```

σημαίνει επιπλέον ότι και οι κλάσεις "Mother" και "Man" είναι ξένες μεταξύ τους.

### 2.3.4 Ορισμός κλάσεων μέσω απαρίθμησης

Η OWL2 διαθέτει τη δυνατότητα ορισμού των λεγόμενων *κλειστών κλάσεων*, οι οποίες προκύπτουν από την απαρίθμηση των μελών τους. Οι κλάσεις αυτές έχουν τη μορφή:

```

<owl:Class rdf:about=(class name)>
  <owl:equivalentClass>
    <owl:Class>
      <owl:oneOf rdf:parseType="Collection">
        <rdf:Description rdf:about=(individual 1) />
        <rdf:Description rdf:about=(individual 2) />
        ...
        <rdf:Description rdf:about=(individual N) />
      </owl:oneOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

```

**Παρατήρηση 2.3.2** *Μια σημαντική ιδιότητα των κλειστών κλάσεων είναι η μη επεκτασιμότητά τους. Αυτό συνεπάγεται ότι κάθε νέο στιγμότυπο της κλάσης ανάγεται σε κάποιο από τα στοιχεία που συνθέτουν την κλάση. Για παράδειγμα, στην περίπτωση της κλάσης*

```

<owl:Class rdf:about="MyBirthdayGuests">
  <owl:equivalentClass>
    <owl:Class>
      <owl:oneOf rdf:parseType="Collection">
        <rdf:Description rdf:about="Bill" />
        <rdf:Description rdf:about="John" />
        <rdf:Description rdf:about="Mary" />
      </owl:oneOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

```

*αν δηλώσουμε ότι το άτομο "Bob" είναι μέλος της κλάσης "MyBirthdayGuests" τότε αναγκαστικά "Bob"  $\approx$  "Bill", "Bob"  $\approx$  "John" ή "Bob"  $\approx$  "Mary".*

### 2.3.5 Σχέσεις στην OWL2

Η περιγραφή σχέσεων μεταξύ ατόμων βασίζεται στην ίδια σύνταξη που χρησιμοποιεί το μοντέλο RDF και έχει τη μορφή:

```
<rdf:Description rdf:about=(subject)>
    <(predicate) rdf:resource=(object) />
</rdf:Description>
```

Μια επιπλέον δυνατότητα της OWL2 είναι η δήλωση ότι δύο άτομα *δεν* συνδέονται από μια δοσμένη ιδιότητα:

```
<owl:NegativePropertyAssertion>
    <owl:sourceIndividual rdf:resource=(subject) />
    <owl:assertionProperty rdf:resource=(predicate) />
    <owl:targetIndividual rdf:resource=(object) />
</owl:NegativePropertyAssertion>
```

Δηλώσεις αυτού του τύπου είναι σημαντικές σε μια οντολογία καθώς η υπόθεση του ανοικτού σύμπαντος σημαίνει ότι οποιαδήποτε δήλωση μπορεί να είναι αληθής, ακόμα και κάτι που δεν γνωρίζουμε. Επομένως πρέπει να δηλώσουμε ρητά ότι μια πρόταση είναι ψευδής προκειμένου να μην ισχύει.

Όμοια με τις υποκλάσεις, μπορούμε να ορίσουμε και υπο-ιδιότητες. Οι δηλώσεις αυτές έχουν τη μορφή:

```
<owl:ObjectProperty rdf:about=(property A)>
    <rdfs:subPropertyOf rdf:resource=(property B) />
</owl:ObjectProperty>
```

Μπορούμε ακόμα να ορίσουμε τα πεδία ορισμού και τιμών μιας ιδιότητας:

```
<owl:ObjectProperty rdf:about=(property)>
    <rdfs:domain rdf:resource=(class A) />
    <rdfs:range rdf:resource=(class B) />
</owl:ObjectProperty>
```

**Παρατήρηση 2.3.3** Τα πεδία ορισμού και τιμών είναι εντελώς όμοια με τις αντίστοιχες έννοιες στην RDFS, με κάποιες προσθήκες που υποδηλώνουν ότι χρησιμοποιούνται εντός μιας οντολογίας. Με τον τρόπο αυτό ένας *reasoner* μπορεί να συμπεράνει μέσα από μια ιδιότητα τις κλάσεις των ατόμων που σχετίζει αυτή.

### 2.3.6 Σχέσεις ατόμων - δεδομένων

Μέχρι στιγμής έχουμε δει παραδείγματα για το πως μπορούμε να συνδέσουμε μεταξύ τους άτομα. Μπορούμε όμως να συνδέσουμε και ένα άτομο με κάποια

συγκεκριμένη τιμή. Οι τύποι δεδομένων που μπορούμε να χρησιμοποιήσουμε προέρχονται από την γλώσσα XML Schema<sup>1</sup> ενώ δηλώσεις αυτού του τύπου έχουν τη μορφή:

```
<(class) rdf:about=(individual)>
  <(property) rdf:datatype=(prefix) (datatype)>(value)</(property)>
</(class)>
```

όπου `prefix` είναι το σταθερό πρόθεμα `"http://www.w3.org/2001/XMLSchema#"`. Ο συνδυασμός προθέματος και τύπου δεδομένων μας δίνει ένα URI από μια προκαθορισμένη λίστα τύπων. Ένα προφανές παράδειγμα εφαρμογής αυτής της δυνατότητας είναι η αντιστοίχιση ενός ατόμου σε έναν ακέραιο αριθμό (εν προκειμένω, η ηλικία του):

```
<Person rdf:about="John">
  <hasAge rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">
    51
  </hasAge>
</Person>
```

### 2.3.7 Ισότητα ατόμων

Έχουμε κάνει ήδη λόγο για την υπόθεση του ανοικτού σύμπαντος στις οντολογίες, ως αναγκαία για την μοντελοποίηση ενός πεδίου ενδιαφέροντος όπου η γνώση μας μπορεί να είναι αποσπασματική. Μια ακόμα χρήσιμη υπόθεση στις οντολογίες (ή για την ακρίβεια η αντίστροφή της) είναι αυτή των *μοναδικών ονομάτων*. Υποθέτοντας ότι η αντιστοιχία ανάμεσα στο σύνολο ονομάτων ατόμων και τις οντότητες που περιγράφουν είναι 1:1 μας εμποδίζει να περιγράψουμε την ίδια έννοια με διαφορετικά ονόματα. Η ανάγκη για κάτι τέτοιο όμως συναντάται πολύ συχνά σε πραγματικές εφαρμογές που συνδυάζουν δεδομένα από διαφορετικές πηγές.

Προκειμένου να αντιμετωπιστεί η ύπαρξη διαφορετικών ονομάτων για το ίδιο άτομο, μπορούμε να κατασκευάσουμε δηλώσεις της μορφής:

```
<rdf:Description rdf:about=(individual A)>
  <owl:sameAs rdf:resource=(individual B)/>
</rdf:Description>
```

ενώ με όμοιο τρόπο μπορούμε να δηλώσουμε ότι δυο ονόματα αναφέρονται σε διαφορετικά άτομα:

```
<rdf:Description rdf:about=(individual A)>
  <owl:differentFrom rdf:resource=(individual B)/>
</rdf:Description>
```

<sup>1</sup><https://www.w3.org/TR/xmlschema11-2/>



### 2.3.8 Ενώσεις, τομές και συμπληρώματα

Στην OWL2 έχουμε τη δυνατότητα να ορίσουμε κλάσεις και ιδιότητες που βασίζονται σε υπάρχουσες δηλώσεις, τις οποίες μπορούμε να συνδυάσουμε με εργαλεία της θεωρίας συνόλων όπως την τομή, την ένωση και το συμπλήρωμα δοσμένων κλάσεων. Δηλώσεις αυτού του τύπου έχουν τις εξής μορφές:

- Τομή

```
<owl:Class rdf:about=(new class)>
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Class rdf:about=(existing class)/>
        <owl:Class rdf:about=(existing class)/>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>
```

- Ένωση

```
<owl:Class rdf:about=(new class)>
  <owl:equivalentClass>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about=(existing class)/>
        <owl:Class rdf:about=(existing class)/>
      </owl:unionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>
```

- Συμπλήρωμα

```
<owl:Class rdf:about=(new class)>
  <owl:equivalentClass>
    <owl:Class>
      <owl:complementOf rdf:resource=(existing class)/>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>
```

Οι παραπάνω τελεστές δεν χρησιμοποιούνται μόνο ξεχωριστά αλλά και σε εμφωλευμένη μορφή. Για παράδειγμα, η κλάση "ChildlessPerson" μπορεί να οριστεί μέσω των "Person" και "Parent":

```
<owl:Class rdf:about="ChildlessPerson">
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Class rdf:about="Person"/>
        <owl:Class>
          <owl:complementOf rdf:resource="Parent"/>
        </owl:Class>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>
```

### 2.3.9 Υπαρξιακοί και καθολικοί περιορισμοί

Οι υπαρξιακοί και οι καθολικοί περιορισμοί ιδιοτήτων χρησιμοποιούνται στις οντολογίες με τρόπο ανάλογο με αυτό των ποσοδεικτών στις λογικές προτάσεις. Η πρώτη περίπτωση είναι αυτή της *υπαρξιακής ποσοτικοποίησης*, μέσω της οποίας ορίζουμε μια κλάση ως ένα σύνολο ατόμων με την εξής ιδιότητα: κάθε άτομο της κλάσης συνδέεται με τουλάχιστον ένα άτομο μιας δοσμένης κλάσης (διαφορετικής ή μη) μέσω της ιδιότητας για την οποία εφαρμόζουμε τον περιορισμό. Η υπαρξιακή ποσοτικοποίηση γίνεται μέσω δηλώσεων της μορφής:

```
<owl:Class rdf:about=(class)>
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource=(property) />
      <owl:someValuesFrom rdf:resource=(class) />
    </owl:Restriction>
  </owl:equivalentClass>
</owl:Class>
```

Στην περίπτωση της *καθολικής ποσοτικοποίησης*, ορίζουμε κλάσεις με παρόμοιο τρόπο αλλά η απαίτηση για ένα τουλάχιστον άτομο μιας δοσμένης κλάσης μετατρέπεται στην απαίτηση όλα τα άτομα να ανήκουν σε μια συγκεκριμένη κλάση. Η καθολική ποσοτικοποίηση γίνεται μέσω δηλώσεων της μορφής:

```
<owl:Class rdf:about=(class)>
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource=(property) />
      <owl:allValuesFrom rdf:resource=(class) />
    </owl:Restriction>
  </owl:equivalentClass>
</owl:Class>
```

Εκτός από την υπαρξιακή και την καθολική ποσοτικοποίηση, μπορούμε να καθορίσουμε και το πλήθος ατόμων που σχετίζονται μέσω μιας σχέσης με ένα υποκείμενο. Συγκεκριμένα, μπορούμε να ορίσουμε το μέγιστο ή το ελάχιστο επιτρεπόμενο πλήθος ατόμων αλλά και τον ακριβή αριθμό μέσω δηλώσεων της μορφής:

```
<rdf:Description rdf:about=(individual)>
  <rdf:type>
    <owl:Restriction>
      <owl:(cardinality restriction)
        rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">
          (value)
      </owl:(cardinality restriction)>
    <owl:onProperty rdf:resource=(property) />
    <owl:onClass rdf:resource=(class) />
  </owl:Restriction>
</rdf:type>
</rdf:Description>
```

όπου το πεδίο "cardinality restriction" αντιστοιχεί σε μια από τις τιμές:

- `maxQualifiedCardinality`, για τον προσδιορισμό μέγιστου πλήθους
- `minQualifiedCardinality`, για τον προσδιορισμό ελάχιστου πλήθους
- `qualifiedCardinality`, για τον προσδιορισμό ακριβούς πλήθους

### 2.3.10 Χαρακτηριστικά σχέσεων

Οι σχέσεις που ορίζουμε στην OWL2 μπορούν να εμπλουτιστούν με διάφορα χαρακτηριστικά που προέρχονται απευθείας από την *SROIQ* και φανερώνουν τις δυνατότητες λεπτομερούς περιγραφής στις οντολογίες. Παραθέτουμε μια λίστα των χαρακτηριστικών αυτών συνοδευόμενα από τις αντίστοιχες δηλώσεις:

- αντίστροφη μιας δοσμένης σχέσης
 

```
<owl:ObjectProperty rdf:about=(new property)>
  <owl:inverseOf rdf:resource=(existing property) />
</owl:ObjectProperty>
```
- συμμετρική σχέση αν το A συνδέεται με το B, τότε και το B συνδέεται με το A)
 

```
<owl:SymmetricProperty rdf:about=(property) />
```

- αντισυμμετρική σχέση (αν το A συνδέεται με το B, τότε το B δεν συνδέεται με το A)

```
<owl:AsymmetricProperty rdf:about=(property) />
```

- ξένες σχέσεις (δύο άτομα δεν μπορούν να συνδέονται από τις δυο σχέσεις ταυτόχρονα)

```
<rdf:Description rdf:about=(property A)>  
  <owl:propertyDisjointWith rdf:resource=(property B) />  
</rdf:Description>
```

- ανακλαστική σχέση (το υποκείμενο και το αντικείμενο μιας σχέσης μπορεί να είναι το ίδιο άτομο)

```
<owl:ReflexiveProperty rdf:about=(property) />
```

- μη ανακλαστική σχέση (το υποκείμενο και το αντικείμενο μιας σχέσης δεν μπορεί να είναι το ίδιο άτομο)

```
<owl:IrreflexiveProperty rdf:about=(property) />
```

- μεταβατική σχέση (αν το A συνδέεται με το B και το B με το Γ, τότε και το A συνδέεται με το Γ)

```
<owl:TransitiveProperty rdf:about=(property) />
```

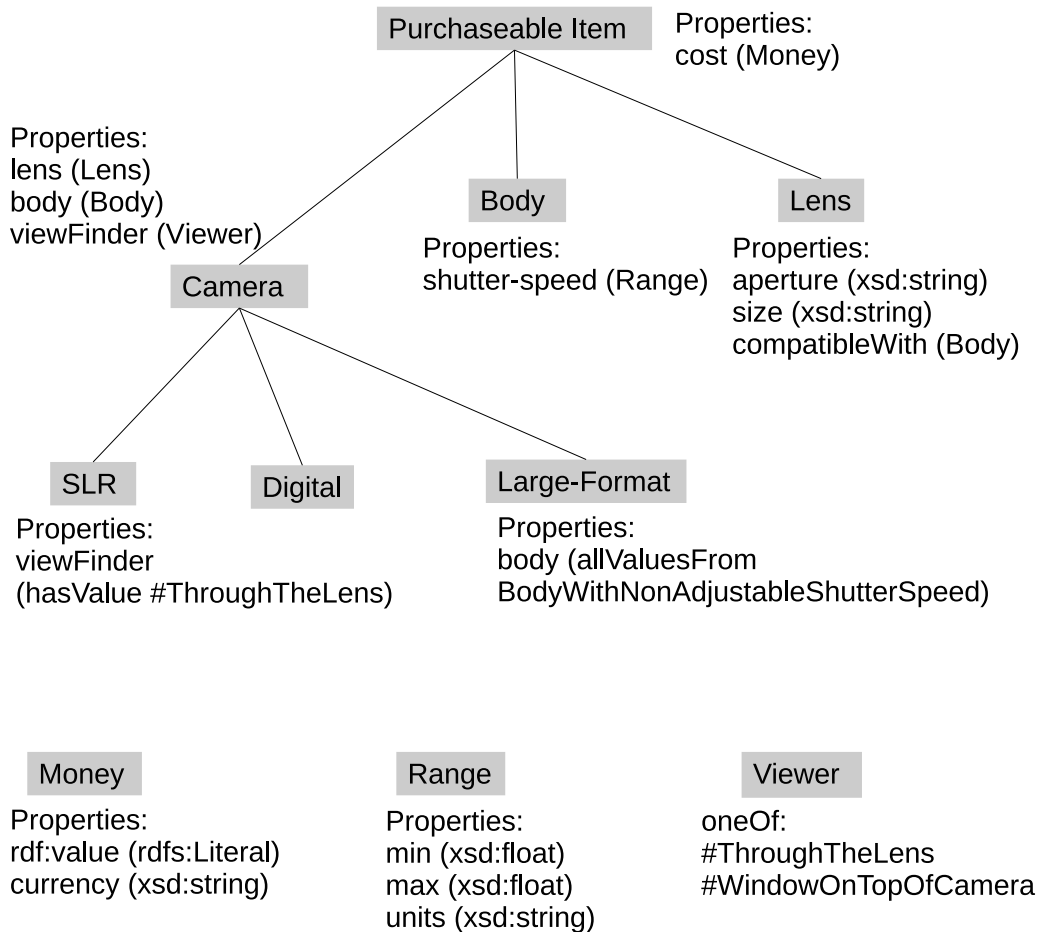
## 2.4 Μια οντολογία για φωτογραφικές μηχανές

Μέσα από μια ολοκληρωμένη οντολογία είναι δυνατή η καλύτερη κατανόηση του πως μπορούμε να συνταιριάζουμε τα εργαλεία της γλώσσας OWL2 στις περιγραφές μας. Στην ενότητα αυτή δίνονται τα βήματα για τη σύνθεση μιας οντολογίας για φωτογραφικές μηχανές [13]. Η βασική ιδέα του παραδείγματος είναι η περιγραφή ενός προϊόντος μέσα από τα τεχνικά στοιχεία και την τιμή του.

Σχετικά με τα τεχνικά στοιχεία, οι φωτογραφικές μηχανές αποτελούνται από το σώμα (body), τον φακό (lens) και το σκόπευτρο (viewfinder). Οι βασικές ιδιότητες που συμπεριλαμβάνονται στην οντολογία είναι η ταχύτητα κλείστρου (shutter speed), το διάφραγμα (aperture) και το εστιακό μήκος (focal length). Οι τύποι μηχανών που συμπεριλαμβάνονται είναι SLR (single-lens reflex), digital και large-format.

Αρχικά, στο σχήμα 2.1 παρουσιάζονται διαγραμματικά οι κλάσεις, οι ιδιότητες και οι μεταξύ τους σχέσεις που συνθέτουν το πεδίο ενδιαφέροντος. Η κλάση στην κορυφή του διαγράμματος αποτελεί την γενικότερη έννοια και με αυτήν συνδέονται διάφορες υποκλάσεις. Επιπλέον υπάρχουν κλάσεις ανεξάρτητες από την ταξινόμια

αυτή, όπως φαίνεται στο κάτω μέρος του διαγράμματος. Οι διάφορες ιδιότητες που ορίζονται μας δίνουν χαρακτηριστικά των κλάσεων ή τους τύπους δεδομένων που χρησιμοποιούνται. Τέλος, δίνονται δυο σχέσεις ισοδυναμίας που μας επιτρέπουν τη χρήση διαφορετικών όρων για την ίδια έννοια.



Equivalence Relationships: aperture  $\approx$  f-stop, size  $\approx$  focal-length

Σχήμα 2.1: Κλάσεις, ιδιότητες και σχέσεις της οντολογίας φωτογραφικών μηχανών

### 2.4.1 Συνολική δομή

Για την ευκολότερη κατανόηση της δομής της οντολογίας, δίνεται ξεχωριστά η συνολική της δομή σε RDF/XML και οι λίστες των κλάσεων και ιδιοτήτων.

```

<?xml version="1.0" encoding="UTF-8"?>
<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:owl="http://www.w3.org/2002/07/owl#"
xmlns="http://www.xfront.com/owl/ontologies/camera/#"
xmlns:camera="http://www.xfront.com/owl/ontologies/camera/#"
xml:base="http://www.xfront.com/owl/ontologies/camera/">

<!-- OWL meta-data for the document -->
<owl:Ontology rdf:about="">
  <rdfs:comment>
    Camera OWL Ontology
    Author: Roger L. Costello
  </rdfs:comment>
</owl:Ontology>

<!-- ***** CLASS DEFINITIONS *****-->

<owl:Class rdf:about="class name">
  <!-- class details, if any -->
</owl:Class>

<!-- ***** PROPERTY DEFINITIONS ***** -->

<owl:DatatypeProperty rdf:about="property name">
  <!-- property details -->
</owl:DatatypeProperty>

<owl:ObjectProperty rdf:about="property name">
  <!-- property details -->
</owl:ObjectProperty>

</rdf:RDF>
```

### 2.4.2 Κλάσεις

```
<owl:Class rdf:about="Money">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#
    Thing" />
</owl:Class>

<owl:Class rdf:about="Range">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#
    Thing" />
</owl:Class>
```

## 2.4. ΜΙΑ ΟΝΤΟΛΟΓΙΑ ΓΙΑ ΦΩΤΟΓΡΑΦΙΚΕΣ ΜΗΧΑΝΕΣ

---

```
<owl:Class rdf:about="PurchaseableItem">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#
    Thing" />
</owl:Class>

<owl:Class rdf:about="Camera">
  <rdfs:subClassOf rdf:resource="#PurchaseableItem" />
</owl:Class>

<owl:Class rdf:about="Window">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#
    Thing" />
</owl:Class>

<!-- notice that these two draw from a different namespace -->
<camera:Window rdf:about="ThroughTheLens" />
<camera:Window rdf:about="WindowOnTopOfCamera" />

<owl:Class rdf:about="Body">
  <rdfs:subClassOf rdf:resource="#PurchaseableItem" />
</owl:Class>

<owl:Class rdf:about="Lens">
  <rdfs:subClassOf rdf:resource="#PurchaseableItem" />
</owl:Class>

<owl:Class rdf:about="Digital">
  <rdfs:subClassOf rdf:resource="#Camera" />
</owl:Class>

<owl:Class rdf:about="Viewer">
  <owl:oneOf rdf:parseType="Collection">
    <camera:Window rdf:about="#ThroughTheLens" />
    <camera:Window rdf:about="#WindowOnTopOfCamera" />
  </owl:oneOf>
</owl:Class>

<owl:Class rdf:about="BodyWithNonAdjustableShutterSpeed">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#Body" />
    <owl:Restriction>
      <owl:onProperty rdf:resource="#shutter-speed" />
      <owl:cardinality>0</owl:cardinality>
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>
```

```
<owl:Class rdf:about="SLR">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#Camera" />
    <owl:Restriction>
      <owl:onProperty rdf:resource="#viewFinder" />
      <owl:hasValue rdf:resource="#ThroughTheLens" />
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>

<owl:Class rdf:about="Large-Format">
  <rdfs:subClassOf rdf:resource="#Camera" />
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#body" />
      <owl:allValuesFrom rdf:resource="#
        BodyWithNonAdjustableShutterSpeed" />
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

### 2.4.3 Ιδιότητες τύπων δεδομένων

```
<owl:DatatypeProperty rdf:about="currency">
  <rdfs:domain rdf:resource="#Money" />
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#
    string" />
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="min">
  <rdfs:domain rdf:resource="#Range" />
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#
    float" />
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="max">
  <rdfs:domain rdf:resource="#Range" />
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#
    float" />
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="units">
  <rdfs:domain rdf:resource="#Range" />
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#
    string" />
```



```
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="focal-length">
  <owl:equivalentProperty rdf:resource="#size" />
  <rdfs:domain rdf:resource="#Lens" />
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#
    string" />
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="f-stop">
  <owl:equivalentProperty rdf:resource="#aperture" />
  <rdfs:domain rdf:resource="#Lens" />
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#
    string" />
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="size">
  <rdfs:domain rdf:resource="#Lens" />
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#
    string" />
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="aperture">
  <rdfs:domain rdf:resource="#Lens" />
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#
    string" />
</owl:DatatypeProperty>
```

### 2.4.4 Ιδιότητες αντικειμένων

```
<owl:ObjectProperty rdf:about="cost">
  <rdfs:domain rdf:resource="#PurchaseableItem" />
  <rdfs:range rdf:resource="#Money" />
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="part" />

<owl:ObjectProperty rdf:about="lens">
  <rdfs:subPropertyOf rdf:resource="#part" />
  <rdfs:domain rdf:resource="#Camera" />
  <rdfs:range rdf:resource="#Lens" />
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="body">
  <rdfs:subPropertyOf rdf:resource="#part" />
  <rdfs:domain rdf:resource="#Camera" />
```

```
<rdfs:range rdf:resource="#Body" />
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="viewFinder">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
    FunctionalProperty" />
  <rdfs:domain rdf:resource="#Camera" />
  <rdfs:range rdf:resource="#Viewer" />
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="compatibleWith">
  <rdfs:domain rdf:resource="#Lens" />
  <rdfs:range rdf:resource="#Body" />
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="shutter-speed">
  <rdfs:domain rdf:resource="#Body" />
  <rdfs:range rdf:resource="#Range" />
</owl:ObjectProperty>
```

### 2.4.5 Στιγμιότυπα της οντολογίας

Η πρακτική αξία της οντολογίας που κατασκευάσαμε προκύπτει όταν επιχειρήσουμε να περιγράψουμε μοντέλα φωτογραφικών μηχανών. Το λεξιλόγιο που έχουμε στη διάθεσή μας καλύπτει πλέον την τεχνική ορολογία με τυποποιημένο τρόπο και εφόσον η οντολογία είναι διαθέσιμη προς τρίτους, επιτρέπει τον διαμοιρασμό της γνώσης. Το παρακάτω παράδειγμα δείχνει ένα τέτοιο στιγμιότυπο:

```
<SLR rdf:about="Olympus-OM-10">
  <lens>
    <Lens>
      <focal-length>75-300mm zoom</focal-length>
      <f-stop>4.5-5.6</f-stop>
    </Lens>
  </lens>
  <body>
    <Body>
      <shutter-speed rdf:parseType="Resource">
        <min>0.002</min>
        <max>1.0</max>
        <units>seconds</units>
      </shutter-speed>
    </Body>
  </body>
  <cost rdf:parseType="Resource">
    <rdf:value>325</rdf:value>
```

#### 2.4. ΜΙΑ ΟΝΤΟΛΟΓΙΑ ΓΙΑ ΦΩΤΟΓΡΑΦΙΚΕΣ ΜΗΧΑΝΕΣ

---

```
<currency>USD</currency>  
</cost>  
</SLR>
```



## Κεφάλαιο 3

# Διαχείριση Γνώσης και Οντολογίες

Στο κεφάλαιο αυτό παρουσιάζεται ένα μοντέλο διαχείρισης γνώσης [14] που βασίζεται σε τρεις οντολογίες προκειμένου να εξουσιοδοτήσει κατάλληλα τα μέλη μιας εικονικής επιχείρησης.

### 3.1 Εικονικές Επιχειρήσεις

Με τον όρο εικονική επιχείρηση εννοούμε μια προσωρινή συνεργασία επιχειρήσεων με διαφορετικά (αλλά συμπληρωματικά μεταξύ τους) πεδία δραστηριοτήτων. Η συνεργασία αυτή δεν πραγματοποιείται μέσω φυσικής παρουσίας αλλά μέσω εργαλείων της Πληροφορικής που επιτρέπουν τη δικτύωση απομακρυσμένων ατόμων ή ομάδων. Σκοπός μιας τέτοιας συνεργασίας είναι η καλύτερη εκμετάλλευση μιας επιχειρηματικής ευκαιρίας μέσα από τον διαμοιρασμό πληροφοριών και πόρων.

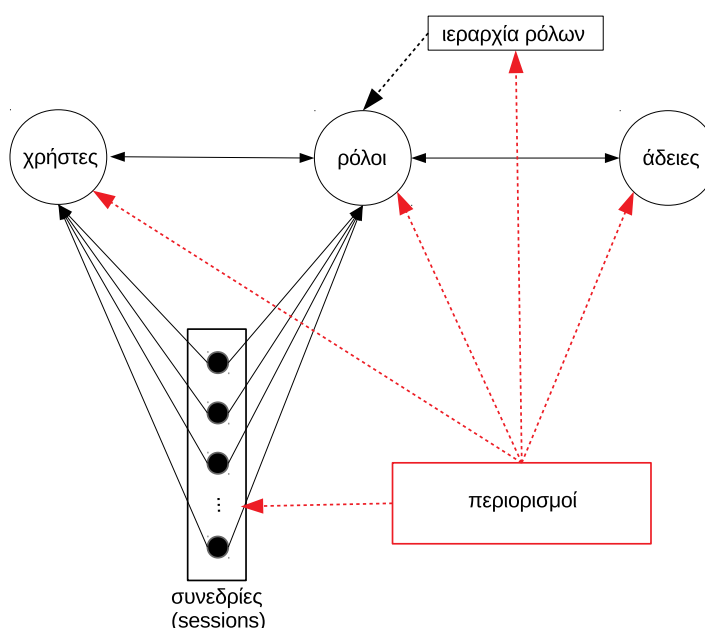
Η αποτελεσματικότητα μιας εικονικής επιχείρησης εξαρτάται σε μεγάλο βαθμό από τις μεθόδους και τα εργαλεία που χρησιμοποιούνται για τη διαχείριση της γνώσης καθώς επιτρέπουν την δημιουργία σχέσεων εμπιστοσύνης μεταξύ των εμπλεκομένων. Το ίδιο σημαντική είναι και η ασφάλεια που παρέχει η σωστή διαχείριση της γνώσης, ειδικά στο περιβάλλον μιας εικονικής επιχείρησης που εξ ορισμού βασίζεται στο Διαδίκτυο για την επικοινωνία των μελών μιας εικονικής επιχείρησης.

### 3.2 Μηχανισμοί Ελέγχου Πρόσβασης

Μέσα από έναν μηχανισμό ελέγχου πρόσβασης καθορίζεται το *πότε, πως και ποιός* μπορεί να χρησιμοποιήσει έναν πόρο. Μια απλή υλοποίηση για έναν τέτοιο μηχανισμό είναι το μοντέλο ACM (access control matrix ή πίνακας ελέγχου πρόσβασης). Ουσιαστικά πρόκειται για έναν πίνακα όπου κάθε γραμμή αντιστοιχεί σε έναν χρή-

στη και κάθε στήλη σε έναν διαθέσιμο πόρο. Οι τιμές αυτού του πίνακα είναι που καθορίζουν την πρόσβαση. Το μοντέλο αυτό είναι απλό και διαισθητικό αλλά δύσκολο να εφαρμοστεί σε μεγάλη κλίμακα. Αυτό διότι η εξουσιοδότηση γίνεται για κάθε χρήστη ξεχωριστά, κάτι που κάνει δύσκολη την συντήρηση και την ενημέρωση του συστήματος.

Ένας βελτιωμένος μηχανισμός ελέγχου πρόσβασης είναι το RBAC (role-based access control ή έλεγχος πρόσβασης βάσει ρόλων), το οποίο προσφέρει ένα κατάλληλο πλαίσιο αφαίρεσης μέσω ρόλων.



Σχήμα 3.1: Αρχιτεκτονική του RBAC

Οι ρόλοι αυτοί μπορούν να αποδοθούν στους χρήστες και συνοδεύονται από προκαθορισμένα δικαιώματα πρόσβασης, λύνοντας έτσι το πρόβλημα της συντήρησης μιας λίστας αδειών για κάθε χρήστη ξεχωριστά. Επιπλέον, οι ρόλοι αυτοί συνθέτουν μια ιεραρχία στην οποία ένας 'υψηλότερος' ρόλος κληρονομεί όλα τα δικαιώματα πρόσβασης των 'χαμηλότερων' ρόλων.

### 3.3 Μοντέλο Διαχείρισης Γνώσης

Για την ανάπτυξη ενός προϊόντος σε μια εικονική επιχείρηση, απαιτείται συχνά ο συνδυασμός γνώσης από διαφορετικές πηγές. Επιπλέον, είναι αναγκαίος ο καθορισμός των κριτηρίων που πρέπει να πληρούνται προκειμένου να παραχωρείται πρόσβαση σε αποθηκευμένη γνώση. Ο λόγος για αυτό είναι το γεγονός ότι η γνώση αυτή συχνά προστατεύεται ως πνευματική ιδιοκτησία και η εταιρία στην οποία ανή-

κει θέλει να ελέγχει τις συνθήκες διαμοιρασμού της. Στο μοντέλο που παρουσιάζεται στη συνέχεια, η δυνατότητα διαμοιρασμού καθορίζεται από τις εξής παραμέτρους:

- ταυτότητα του χρήστη που ζητά πρόσβαση
- ρόλος που έχει αναληφθεί από τον χρήστη εντός του περιβάλλοντος της εικονικής επιχείρησης
- είδος των διεργασιών που βασίζονται στην γνώση που θα διαμοιραστεί
- χρονική περίοδος στη διάρκεια ζωής της εικονικής επιχείρησης

Στο παράδειγμα που ακολουθεί, παρουσιάζεται ένα σενάριο όπου απαιτείται ο διαμοιρασμός γνώσης κατά τη διαδικασία παραγωγής ενός προϊόντος.

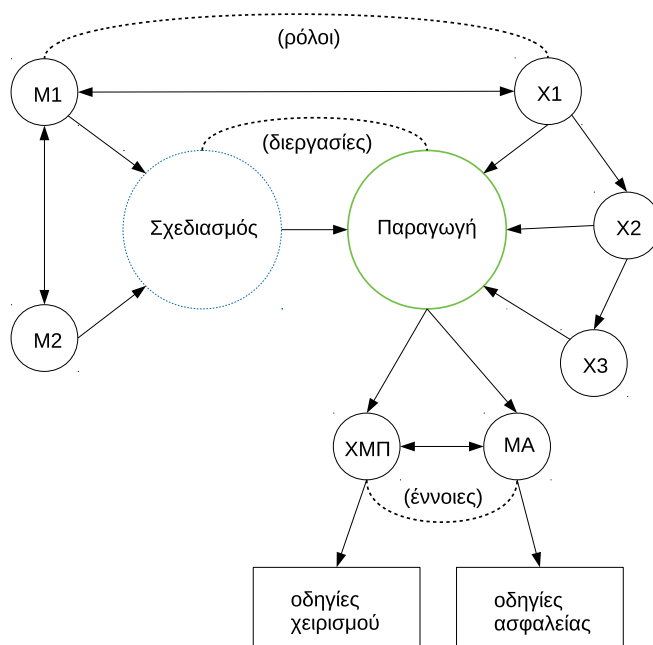
**Παράδειγμα 3.3.1** Υποθέτουμε ότι σε μια εικονική επιχείρηση η παραγωγή ενός προϊόντος χωρίζεται στα στάδια του σχεδιασμού της διαδικασίας παραγωγής και της παραγωγής μέσα από την χρήση των κατάλληλων μηχανημάτων. Οι διεργασίες αυτές διεκπεραιώνονται μέσα από τους εξής ρόλους:

- Μηχανικός 1 (M1)
- Μηχανικός 2 (M2)
- Χειριστής 1 (X1)
- Χειριστής 2 (X2)
- Χειριστής 3 (X3)

Συγκεκριμένα, οι ρόλοι M1 και M2 εκτελούν σε πρώτο χρόνο τον σχεδιασμό της διαδικασίας παραγωγής ανταλλάσσοντας μεταξύ τους την αναγκαία γνώση. Ταυτόχρονα, στις απαιτήσεις του ρόλου M1 είναι η μεταφορά της διαδικασίας που σχεδιάστηκε στον ρόλο X1 και από αυτόν στους X2 και X3 για την ανάληψη της διαδικασίας. Η διαδικασία παραγωγής συνδέεται άμεσα με τις εξής έννοιες:

- χειρισμός των μηχανημάτων παραγωγής (ΧΜΠ)
- μέτρα ασφαλείας κατά τον χειρισμό (ΜΑ)

Τέλος, η γνώση που καλύπτει τις έννοιες αυτές μπορεί να αποκτηθεί μέσα από τα αντίστοιχα εγχειρίδια οδηγιών.



Σχήμα 3.2: Σχεδιασμός και παραγωγή σε μια εικονική επιχείρηση

Στο παράδειγμα του σχήματος 3.2 οι διακεκομμένες ακμές υποδεικνύουν τους διαφορετικούς τρόπους διαμοιρασμού γνώσης. Παρατηρούμε ότι η ανάγκη διαμοιρασμού υπάρχει μεταξύ:

- **ρόλων**, καθώς οι διάφοροι ρόλοι παρουσιάζουν διαφορετικές εξειδικεύσεις
- **διεργασιών**, που συχνά αποτελούν μια αλυσίδα όπου το αποτέλεσμα της μιας αποτελεί το θεμέλιο της άλλης
- **εννοιών**, όπου η κατάλληλη οργάνωση μπορεί να διευκολύνει την αναζήτηση πληροφοριών

### 3.3.1 Επίπεδα Οργάνωσης Γνώσης

Το σύνολο της διαθέσιμης γνώσης σε μια εικονική επιχείρηση οργανώνεται σε τρία επίπεδα. Στο υψηλότερο επίπεδο, το λεγόμενο *επίπεδο εννοιών*, συναντάμε τρεις οντολογίες που αντιστοιχούν σε τρεις διαφορετικούς τομείς μιας εικονικής επιχείρησης:

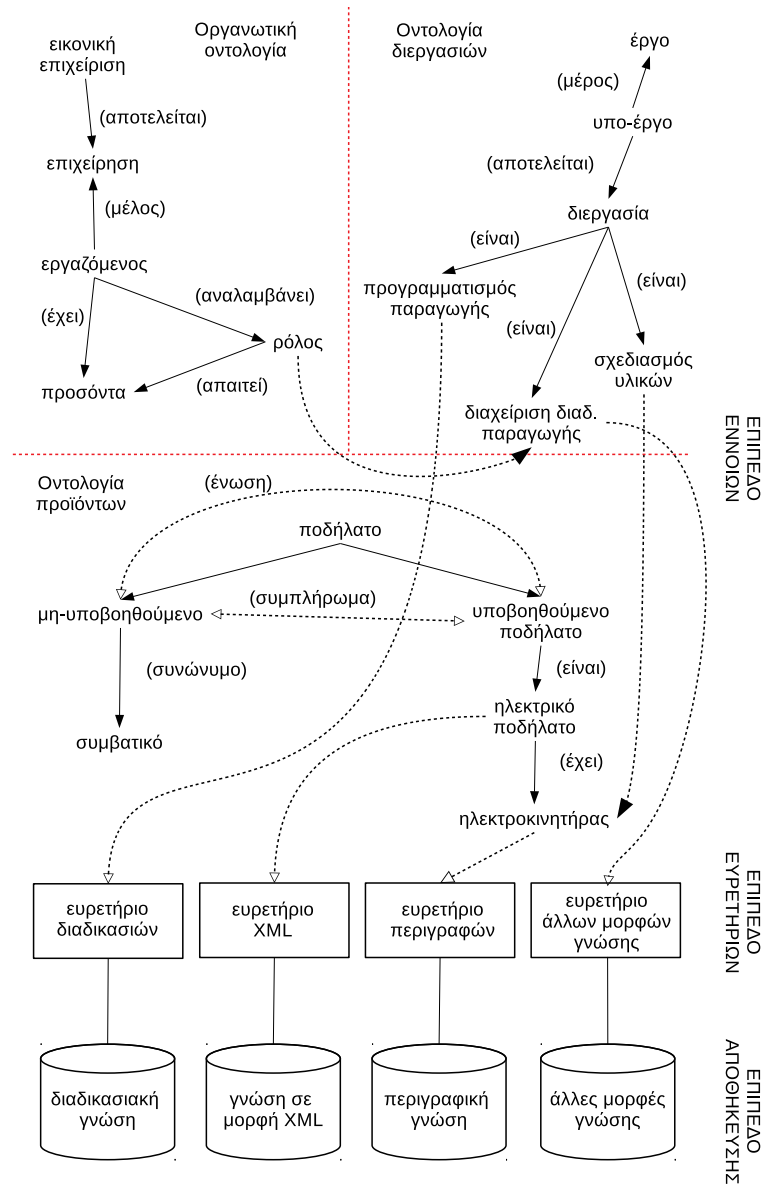


- την **οντολογία προϊόντων**, η οποία αναπαριστά τις σχέσεις μεταξύ των προϊόντων της εικονικής επιχείρησης
- την **οργανωτική οντολογία**, η οποία αναπαριστά τις σχέσεις μεταξύ ρόλων στο περιβάλλον μιας εικονικής επιχείρησης
- την **οντολογία διεργασιών**, η οποία αναπαριστά τις σχέσεις ανάμεσα σε διαδικασίες που αφορούν την παραγωγή προϊόντων

Οι έννοιες που απαρτίζουν τις οντολογίες αυτές αποκτούν πρακτική αξία όταν αντιστοιχίζονται σε πραγματικά δεδομένα. Αυτό γίνεται εμφανές στο παράδειγμα 3.2, όπου οι σχετικές με την παραγωγή έννοιες αντιστοιχίζονται σε εγχειρίδια οδηγιών. Γενικεύοντας, η συλλογή αναλυτικών περιγραφών των εννοιών που έχουν να κάνουν με την οργάνωση και τις δραστηριότητες της εικονικής επιχείρησης γίνεται στο λεγόμενο *επίπεδο αποθήκευσης*. Παραδείγματα μέσω των οποίων για το σκοπό αυτό είναι οι σχεσιακές βάσεις δεδομένων, συλλογές αρχείων XML, τεχνικά σχεδιαγράμματα κ.α. Συνοπτικά, στο επίπεδο αποθήκευσης συναντάμε τους εξής τύπους γνώσης:

- περιγραφική γνώση ("know-what")
- διαδικασιακή γνώση ("know-how")
- γνώση αιτίας-αποτελέσματος ("know-why")

Η σύνδεση των δυο επιπέδων γίνεται μέσω του *επιπέδου ευρετηρίου* στο οποίο γίνεται αντιστοίχιση των εννοιών με τις αναλυτικές περιγραφές του επιπέδου αποθήκευσης. Μια σχηματική περιγραφή των παραπάνω αποτελεί το σχήμα 3.3.



Σχήμα 3.3: Επίπεδα οργάνωσης γνώσης σε μια εικονική επιχείρηση

### 3.3.2 Λεξιλόγιο του μοντέλου

Η λίστα όρων που ακολουθεί μας δίνει τα βασικά στοιχεία που συνθέτουν την έννοια της εικονικής επιχείρησης σε μια τυποποιημένη μορφή :

- **εικονική επιχείρηση** (Virtual Enterprise - **VE**): Η ομάδα επιχειρήσεων που συνεργάζεται.
- **έργο** (Project - **P**): Η βασική μονάδα δραστηριοτήτων μιας εικονικής επιχείρησης. Μπορεί να χωριστεί σε υπο-έργα εφόσον απαιτείται.
- **μέλος επιχείρησης** (Enterprise Member - **EM**): Ένα μέλος επιχείρησης ανήκει σε μια από τις επιχειρήσεις που συνθέτουν την VE. Καθώς μια επιχείρηση μπορεί να συμμετέχει σε διαφορετικές VE ταυτόχρονα, είναι δυνατό να υπάρχουν σχέσεις συνεργασίας αλλά και ανταγωνιστικές σχέσεις μεταξύ των μελών μιας επιχείρησης.
- **ρόλος εικονικής επιχείρησης** (Virtual Enterprise Role - **VER**): Οι ρόλοι εικονικής επιχείρησης καλύπτουν τις διεργασίες που πραγματοποιούνται εντός μιας εικονικής επιχείρησης και μπορεί να αντιστοιχίζονται σε εσωτερικούς ρόλους μιας επιχείρησης.
- **διεργασία** (Functional Task - **FT**): Η διεργασία αποτελεί την βασική μονάδα ενός έργου. Τα έργα βασίζονται σε διεργασίες σε χρονολογική σειρά και η εκτέλεσή τους γίνεται από τους αντίστοιχους ρόλους.
- **άδεια γνώσης** (Knowledge Permission - **KP**): Η άδεια αυτή αφορά τη δυνατότητα ανάγνωσης και τροποποίησης (read/write permission) γνώσης.

### 3.3.3 Σχέσεις του μοντέλου

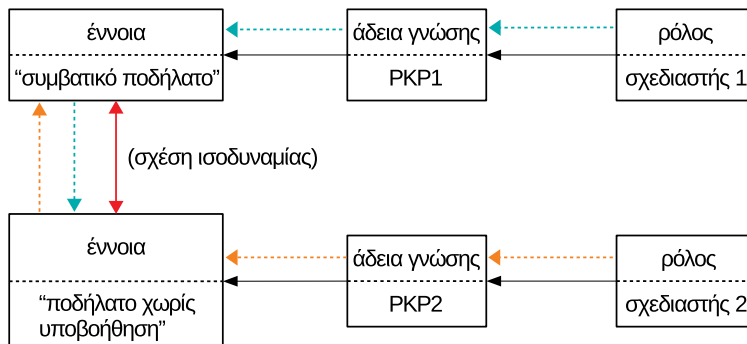
Η λίστα σχέσεων που ακολουθεί παρουσιάζει τον τρόπο οργάνωσης και λειτουργίας μιας εικονικής επιχείρησης μέσω του μοντέλου διαχείρισης γνώσης. Συγκεκριμένα μας δίνει μια εικόνα για το πως δημιουργούνται και ανατίθενται ρόλοι, πως οργάνωνεται η διαδικασία περάτωσης ενός έργου και αντλείται γνώση από προηγούμενα ολοκληρωμένα έργα αλλά και πως οι διάφοροι ρόλοι σχετίζονται μεταξύ τους.

- **Ανάθεση**: Οι σχέσεις αυτού του τύπου βασίζονται στο λεξιλόγιο της υποενότητας 3.3.2 και αποτελούν μια επέκταση του μοντέλου RBAC. Παραδείγματα αναθέσεων είναι οι σχέσεις
  - U-R-A (user-role assignment)
  - VE-EM-A (Virtual Enterprise - Enterprise Member assignment)

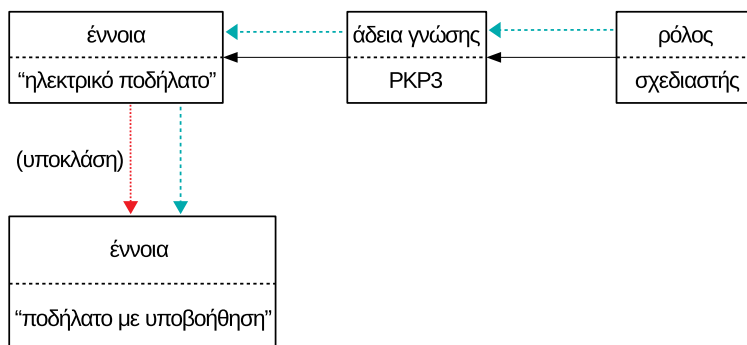
- VER-FT-A (Virtual Enterprise Role - Functional Task assignment)
- VE-P-A (Virtual Enterprise - Project assignment)
- **Σχέσεις έργων:** Καθορίζουν τη δυνατότητα μεταφοράς γνώσης από έργο σε έργο και χωρίζονται σε:
  - Σχέσεις υποσυνόλων (για παράδειγμα ένα κύριο έργο μπορεί να χωριστεί σε πολλά ανεξάρτητα υπο-έργα).
  - Σχέσεις εκδόσεων, οι οποίες συνδέουν τις διαφορετικές εκδόσεις ενός έργου στο πέρασμα του χρόνου (για παράδειγμα μια μεταγενέστερη έκδοση έχει πρόσβαση σε πληροφορίες που αφορούν μια προγενέστερη έκδοση).
  - Σχέσεις αναφοράς, οι οποίες συνδέουν ένα κύριο έργο με ένα έργο αναφοράς από το οποίο αντλεί γνώση.
  - Σχέσεις διεργασιών, οι οποίες δημιουργούν μια ακολουθία υπο-έργων και καθορίζουν το χρονικό σημείο που επιτρέπεται ο διαμοιρασμός γνώσης μεταξύ τους.
  - Σχέσεις αποκλεισμού, οι οποίες υποδηλώνουν ξένα μεταξύ τους έργα. Σε αυτήν την περίπτωση δεν είναι δυνατός ο διαμοιρασμός γνώσης. Έτσι διατηρείται η ασφάλεια των πόρων και των πληροφοριών ανάμεσα σε χρήστες ή επιχειρήσεις με ανταγωνιστικές σχέσεις.
- **Σχέσεις συνεργασίας ρόλων:** Περιγράφουν το πως συνδέονται οι διάφοροι ρόλοι μιας εικονικής επιχείρησης και χωρίζονται σε:
  - Συνεργασίες, όπου οι σχετιζόμενοι ρόλοι εκτελούν μια διεργασία από κοινού και φέρουν τα ίδια δικαιώματα πρόσβασης σε γνώση για τη διεργασία αυτή.
  - Σχέσεις εξάρτησης, όπου οι σχετιζόμενοι ρόλοι εκτελούν διαφορετικές αλλά γύρω από το ίδιο αντικείμενο διεργασίες και το αποτέλεσμα μιας διεργασίας αποτελεί μέρος μιας άλλης.
  - Σχέσεις ανεξαρτησίας, όπου οι σχετιζόμενοι ρόλοι εκτελούν άσχετες μεταξύ τους διεργασίες και ο διαμοιρασμός γνώσης δεν επιτρέπεται.
- **Σχέσεις γνώσης:** Οι σχέσεις γνώσης επιτρέπουν την σύνδεση μεταξύ εννοιών και την σύνδεση εννοιών με λεπτομερείς περιγραφές, όπως ορίστηκε στα επίπεδα αναπαράστασης της γνώσης. Οι σχέσεις που χρησιμοποιούνται εδώ προέρχονται απευθείας από την γλώσσα OWL και είναι οι εξής:
  - εγκλεισμός ( $\sqsubseteq$ )
  - ισότητα ( $\approx$ )
  - σχέση μέρους ("partOf" relationship)

- τομή (  $\sqcap$  )
- ένωση (  $\sqcup$  )
- συμπλήρωμα (  $\neg$  )

Οι σχέσεις αυτές επιτρέπουν την δυναμική επέκταση της πρόσβασης σε νέα γνώση, καθώς οι μεταβολές τους αντιστοιχούν σε μεταβολές των δυνατοτήτων πρόσβασης ενός ρόλου. Το χαρακτηριστικό αυτό ονομάζεται **διάδοση εξουσιοδότησης** και αποτελεί ένα από τα σημαντικότερα χαρακτηριστικά του μοντέλου διαχείρισης γνώσης που παρουσιάζεται. Στα σχήματα 3.4 και 3.5 φαίνεται πως η διάδοση εξουσιοδότησης μπορεί να εφαρμοστεί σε υποκλάσεις και ισοδύναμες έννοιες μιας οντολογίας. Οι μαύρες ακμές υποδεικνύουν τη γνώση στην οποία έχει απευθείας πρόσβαση ένας ρόλος ενώ οι χρωματισμένες ακμές δείχνουν πως η πρόσβαση επεκτείνεται έμμεσα σε νέες έννοιες.



Σχήμα 3.4: Διάδοση εξουσιοδότησης μέσω ισοδύναμων εννοιών



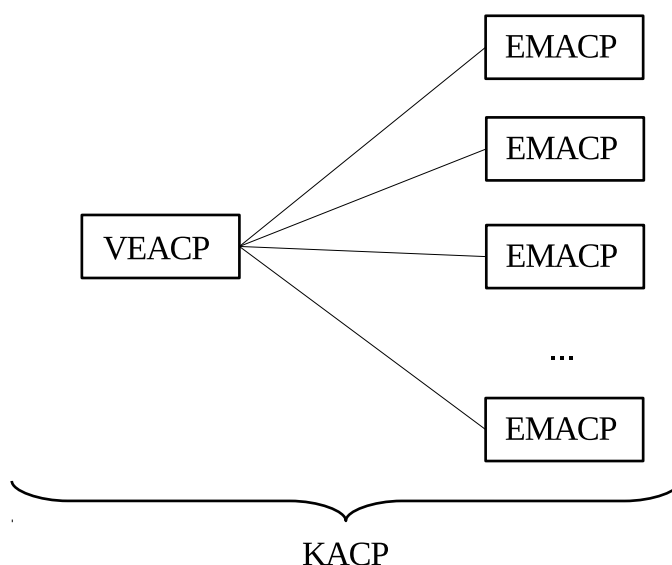
Σχήμα 3.5: Διάδοση εξουσιοδότησης μέσω υποκλάσεων

### 3.3.4 Πολιτική Ελέγχου Πρόσβασης Γνώσης

Έχοντας δημιουργήσει μια κατάλληλη υποδομή για την οργάνωση και την διαχείριση γνώσης μπορούμε πλέον να διαμορφώσουμε και το αντίστοιχο πλαίσιο έκφρασης κανόνων εξουσιοδότησης ή αλλιώς μια *πολιτική ελέγχου πρόσβασης γνώσης* (**KACP** - Knowledge Access Control Policy). Ουσιαστικά, πρόκειται για ένα σύνολο κανόνων με κοινή μορφή των οποίων η ισχύς μεταβάλλεται δυναμικά ανάλογα με τις συνθήκες, όπως για παράδειγμα η χρονική στιγμή στη διάρκεια ζωής ενός έργου ή μια μεταβολή στις οντολογίες που συνθέτουν το επίπεδο εννοιών.

Μια KACP καλύπτει τόσο την εικονική επιχείρηση συνολικά όσο και τις επιμέρους επιχειρήσεις μέσα από διαφορετικά αλλά συνδεδεμένα σύνολα κανόνων:

- Πολιτική ελέγχου πρόσβασης εικονικής επιχείρησης (**VEACP** - Virtual Enterprise Access Control Policy): Ένα σύνολο κανόνων που αφορά την πρόσβαση και τον διαμοιρασμό συνολικά στο περιβάλλον της επιχείρησης.
- Πολιτικές ελέγχου πρόσβασης μελών (**EMACPs** - Enterprise Members Access Control Policies): Σύνολα κανόνων που αφορούν το κάθε μέλος της εικονικής επιχείρησης ξεχωριστά. Οι EMACPs έπονται της VEACP και οφείλουν να μην έρχονται σε σύγκρουση με αυτή.



Σχήμα 3.6: Διαχωρισμός VEACP/EMACPs

Κάθε κανόνας που ανήκει σε μια KACP έχει συγκεκριμένα δομικά μέρη:

- **Στόχος** (<Target/>) : το σύνολο των οντοτήτων που καλύπτει ο κανόνας. Ένας στόχος χωρίζεται περαιτέρω στα εξής σύνολα:

- ένα σύνολο υποκειμένων (<SubjectSet/>), δηλαδή τα άτομα ή τις ομάδες ατόμων που αφορά
  - ένα σύνολο επιτρεπόμενων ενεργειών (<ActionSet/>), όπως "read" ή "write"
  - το σύνολο των σχετικών με τον κανόνα εννοιών (<KnowledgeSet/>)
  - το περιβάλλον που συνθέτουν διάφορες παράμετροι σχετικές με την εξουσιοδότηση (<Environment/>)
- **Όροι** (<Condition/>): Ένα σύνολο από αναγκαίες συνθήκες για την εφαρμογή του κανόνα.
  - **Αποτέλεσμα** (<Effect/>): Το αποτέλεσμα της εφαρμογής του κανόνα όταν ικανοποιούνται οι αναγκαίες συνθήκες.

Στην περίπτωση που κάποιοι κανόνες πρέπει να εφαρμοστούν ταυτόχρονα ένας αλγόριθμος συνδυασμού (<RuleCombiningAlgorithm/>) προσφέρει τις εξής επιλογές:

- επιτρέπει την παράκαμψη ενός κανόνα από κάποιον άλλο
- απαγορεύει κάθε παράκαμψη κανόνων
- επιτρέπει την εφαρμογή του πρώτου σε χρονολογική σειρά κανόνα

Ένα παράδειγμα σύνθεσης μιας πολιτικής ελέγχου πρόσβασης γνώσης βρίσκεται στο παράρτημα, με θέμα την συναρμολόγηση ενός ποδηλάτου και την κατασκευή των κατάλληλων εξαρτημάτων στο περιβάλλον μιας εικονικής επιχείρησης.

## Παράρτημα

### Κατάλογος χαρακτηριστικών, κλάσεων και ιδιοτήτων RDF

1	RDF ATTRIBUTES
2	-----
3	rdf:about
4	defines the resource being described
5	rdf:Description
6	container for the description of a resource
7	rdf:resource
8	a resource to identify a property
9	rdf:datatype
10	data type of an element
11	rdf:ID
12	the ID of an element
13	rdf:li
14	list
15	rdf:_n
16	defines a node
17	rdf:nodeID
18	the ID of an element node
19	rdf:parseType
20	defines how an element should be parsed
21	rdf:RDF
22	the root of an RDF document
1	RDF CLASSES
2	-----
3	rdf:XMLLiteral
4	the class of XML literal values
5	rdf:Property
6	the class of properties
7	rdf:Statement
8	the class of RDF statements
9	rdf:Alt, rdf:Bag, rdf:Seq
10	containers of alternatives, unordered containers
11	and ordered containers (rdfs:Container is a
12	super-class of the three)
13	rdf:List
14	the class of RDF Lists
15	rdf:nil
16	an instance of rdf:List representing the empty list
1	RDF PROPERTIES
2	-----



```
3 rdf:type
4     an instance of rdf:Property used to state
5     that a resource is an instance of a class
6 rdf:first
7     the first item in the subject RDF list
8 rdf:rest
9     the rest of the subject RDF list after
10    rdf:first
11 rdf:value
12    idiomatic property used for structured values
13 rdf:subject
14    the subject of the RDF statement
15 rdf:predicate
16    the predicate of the RDF statement
17 rdf:object
18    the object of the RDF statement
```

#### Κατάλογος κλάσεων και ιδιοτήτων RDFS

```
1 RDFS CLASSES
2 -----
3 rdfs:Resource
4     represents the class of everything,
5     all things described by RDF are resources
6 rdfs:Class
7     declares a resource as a class for other resources
8     the definition of rdfs:Class is recursive as
9     rdfs:Class is the class of classes and so it
10    is an instance of itself
11 rdfs:Literal
12    literal values such as strings and integers,
13    may be plain or typed.
14 rdfs:Datatype
15    the class of datatypes, both an instance and a
16    subclass of rdfs:Class, each instance of
17    rdfs:Datatype is a subclass of rdfs:Literal
```

```
1 RDFS PROPERTIES
2 -----
3 Properties are instances of the class rdf:Property
4 and describe a relation between subject resources
5 and object resources. When used as such a property
6 is a predicate.
7
8 rdfs:domain
```

```
9         declares the class of the subject in a triple
10        whose predicate is that property
11 rdfs:range
12        declares the class or datatype of the object
13        in a triple whose predicate is that property
14 rdfs:subClassOf
15        allows declaration of hierarchies of classes
16        which support inheritance of a property
17        domain and range from a class to its subclasses
18 rdfs:subPropertyOf
19        an instance of rdf:Property that is used to
20        state that all resources related by one
21        property are also related by another
22 rdfs:label
23        an instance of rdf:Property that may be used
24        to provide a human-readable version
25        of a resource's name
26 rdfs:comment
27        an instance of rdf:Property that may be used
28        to provide a human-readable description of
29        a resource
30 rdfs:seeAlso
31        an instance of rdf:Property that is used to
32        indicate a resource that might provide
33        additional information about the subject
34        resource
35 rdfs:isDefinedBy
36        an instance of rdf:Property that is used
37        to indicate a resource defining the subject
38        resource and may be used to indicate an RDF
39        vocabulary in which a resource is described
```

Η πλήρης περιγραφή των RDF(S) είναι διαθέσιμη στην τοποθεσία:

<https://www.w3.org/TR/rdf-schema/>

# Bibliography

- [1] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific American*, pp. 29–37, May 2001.
- [2] J. Andreu-Perez, C. Poon, R. Merrifield, S. Wong, and G.-Z. Yang, "Big data for health," *IEEE journal of biomedical and health informatics*, vol. 19, 07 2015.
- [3] S. B. Bhattacharyya, *Introduction to SNOMED CT*. Springer, Singapore, 2016.
- [4] J. Domingue, D. Fensel, and J. A. Hendler, eds., *Handbook of Semantic Web Technologies*, ch. 1, pp. 3–41. Springer, 2011.
- [5] S. Omerovic, V. Milutinovic, and S. Tomazic, "Concepts, ontologies, and knowledge representation," 01 2001.
- [6] F. Manola and E. Miller, "Rdf primer," 2004.
- [7] S. Steve Harris and A. Seaborne, "Sparql 1.1 query language," 2013.
- [8] D. Brickley and R. V. Guha, "Rdf schema 1.1," 2014.
- [9] P. Hitzler, M. Krötzsch, B. Parsia, P. F. Patel-Schneider, and S. Rudolph, "Owl 2 web ontology language primer (second edition)," 2012.
- [10] I. Horrocks, O. Kutz, and U. Sattler, "The even more irresistible sroiq," in *Proceedings of the Tenth International Conference on Principles of Knowledge Representation and Reasoning*, KR'06, pp. 57–67, AAAI Press, 2006.
- [11] J. Sowa, "Conceptual graphs for representing conceptual structures," 2009.
- [12] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowledge Acquisition*, vol. 5, no. 2, pp. 199–220, 1993.
- [13] R. L. Costello, "Camera ontology," 2003.
- [14] T.-Y. Chen, "Knowledge sharing in virtual enterprises via an ontology-based access control approach," *Computers in Industry*, vol. 59, no. 5, pp. 502–519, 2008.