



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ

«Διδακτικής της Τεχνολογίας & Ψηφιακών Συστημάτων»

Κατεύθυνση : Ηλεκτρονική Μάθηση

Semantic Web Tutorial

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

ΣΟΦΙΑΣ ΜΑΥΡΟΓΙΑΝΝΗ

Επιβλέπων: Συμεών Ρετάλης

Επίκουρος Καθηγητής Πανεπιστημίου Πειραιά

Αθήνα, Μάρτιος 2007

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ

«Διδακτικής της Τεχνολογίας & Ψηφιακών Συστημάτων»

Κατεύθυνση : Ηλεκτρονική Μάθηση

Semantic Web Tutorial

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

ΣΟΦΙΑΣ ΜΑΥΡΟΓΙΑΝΝΗ

Επιβλέπων: Συμεών Ρετάλης

Επίκουρος Καθηγητής Πανεπιστημίου Πειραιά

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 30^η Μαρτίου 2007:

.....
Συμεών Ρετάλης	N.M. Σγούρος	A. Κανάτας
Επίκουρος Καθηγητής	Αναπληρωτής Καθηγητής	Επίκουρος Καθηγητής
Πανεπιστημίου Πειραιά	Πανεπιστημίου Πειραιά	Πανεπιστημίου Πειραιά

Αθήνα, Μάρτιος 2007

**Στους γονείς μου, Γιάννη και Βιβή
και στην αδερφή μου Μαρίνα**

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΙΑ

Περίληψη

Ο σημασιολογικός Ιστός αποτελεί μία εξελιγμένη μορφή του σημερινού ιστού που επιτρέπει την καλύτερη επεξεργασία των πληροφοριών για τον Ιστό από τον υπολογιστή, με τη δόμηση των εγγράφων που γράφονται για τον Ιστό κατά τέτοιο τρόπο ώστε η πληροφορία που εμπεριέχουν να γίνεται κατανοητή από τις μηχανές. Έτσι, οι πληροφορίες μπορούν να χρησιμοποιηθούν για τη δημιουργία πιο σύνθετων εφαρμογών και σφαιρικών βάσεων δεδομένων.

Ο Σημασιολογικός Ιστός αναμένεται να διαθέσει προς επεξεργασία στους χρήστες τεράστιες ποσότητες πληροφοριακών πόρων (π.χ δεδομένα, έγγραφα, εικόνες, προγράμματα) σε συνδυασμό με διάφορα είδη περιγραφικής πληροφορίας (μεταδεδομένα). Ανώτερος στόχος της όλης προσπάθειας είναι η ποιότητα των υπηρεσιών, η οποία συνίσταται κυρίως στην βελτιωμένη αναζήτηση, την εκτέλεση σύνθετων διεργασιών μέσω του Διαδικτύου και στην εξατομίκευση της πληροφορίας σύμφωνα με τις ανάγκες του εκάστοτε χρήστη. Οι γλώσσες οντολογίας, όπως η OWL, η οποία επεκτείνει την RDF, παρέχουν μια τεχνολογική βάση για να επιτρέψουν την υλοποίηση του σημασιολογικού Ιστού.

Το συγκεκριμένο tutorial εξηγεί όλα τα παραπάνω, χρησιμοποιώντας τη βασική θεωρία και τα κατάλληλα παραδείγματα.

Λέξεις κλειδιά: Σημασιολογικός Ιστός, μεταδεδομένα, οντολογίες, RDF, OWL, πράκτορες.

Abstract

Semantic Web is a concept that enables better machine processing of information found on the Web, by structuring documents written for the Web in such a way that they become understandable by machines. This can be used for creating more complex applications (intelligent browsers, more advanced Web agents), global databases with the data from the Web, reuse of information, etc.

The Semantic Web is expected to provide users with enormous quantities of informative resources (i.e. data, documents, pictures, programs) in combination with various types of descriptive information (metadata). The final objective of all this effort is the quality of services, which regards mainly the improved search, the implementation of complex activities through the Internet and the individualisation of information according to the user's needs. The ontology languages used, such as OWL, which extends RDF, provide a technological base in order to allow the implementation of the Semantic Web.

This tutorial explains all the above, using both the basic theory and the appropriate examples.

Keywords: Semantic Web, metadata, ontologies, RDF, OWL, agents.

Περιεχόμενα

1.	Σημασιολογικός Ιστός	10
1.1.	Υπάρχοντα εργαλεία εκμάθησης του SW	10
1.2.	Semantic Web Tutorial	11
1.3.	Σκοπός της διπλωματικής εργασίας.....	12
1.3.1.	Σημασιολογικός ιστός και e-applications.....	13
1.3.2.	Χρησιμότητα του SW Tutorial	18
1.4.	Δομή του tutorial	19
1.5.	Σε ποιον απευθύνεται.....	20
2.	Semantic Web Tutorial	22
2.1.	Εκπαιδευτική φιλοσοφία του Tutorial	22
2.1.1.	Μάθηση μέσω της ανάλυσης περιπτώσεων και on-line εκπαίδευση..	23
2.1.2.	Βήματα μιας ανάλυσης περιπτώσεων	24
2.1.3.	Πλατφόρμες συνεργασίας.....	29
2.1.4.	On-line ανάλυση περιπτώσεων.....	30
2.2.	Περιγραφή του SW Tutorial.....	30
2.2.1.	Αρχική σελίδα.....	31
2.2.2.	Semantic Web	34
2.2.3.	Ontologies.....	36
2.2.4.	Resource Description Framework.....	39
2.2.5.	Web Ontology Language (OWL)	40
2.2.6.	Protégé.....	41
2.2.7.	Εφαρμογή του Protégé	42
2.2.8.	Altova SemanticWorks.....	46
2.2.9.	Agents.....	52
3.	Εισαγωγή στο SW	54
3.1.	Η ανάγκη για semantic στο Web	54
3.2.	Τι είναι ο σημασιολογικός ιστός και γιατί ενδείκνυται	56
3.3.	Χαρακτηριστικά του SW	58
3.3.1.	Σήμανση και ανάκτηση πληροφοριών	58
3.3.2.	Μεταδεδομένα.	59
3.3.3.	Δεικτοδότηση.....	59
3.3.4.	Μια τεράστια διαλειτουργική βάση δεδομένων	59
3.3.5.	Αυτόματη ανάκτηση δεδομένων.....	60
3.3.6.	Υπηρεσίες βασισμένες στο WEB	60
3.3.7.	Ανακάλυψη των υπηρεσιών	61
3.3.8.	Οι ευφυείς πράκτορες.....	61
3.4.	Πρακτική του SW	62
3.4.1.	SW και γνώση.....	62
3.4.2.	B2B Ηλεκτρονικό εμπόριο	62
3.4.3.	Σημασιολογικός Ιστός και επιχείρηση – Προτερήματα	63
3.4.4.	Λήψη αποφάσεων	65
3.4.5.	Η επιχειρησιακή ανάπτυξη.....	66
3.4.6.	Μεταβίβαση πληροφοριών και γνώσης	67
3.4.7.	Διοίκηση και αυτοματοποίηση	67
3.5.	Βασικές έννοιες του SW	68
3.5.1.	Οι τεχνολογίες Semantic Web	68
3.5.2.	Αρχιτεκτονική του SW	72
3.5.3	Software Agents.....	77
3.6.	SW παράδειγμα	77

3.7.	Εφαρμογές.....	78
4.	Ταξονομίες - Οντολογίες.....	79
4.1.	Η ανάγκη για ιεράρχηση των data.....	79
4.2.	Χρησιμότητα των taxonomies.....	83
4.3.	Το φάσμα οντολογίας.....	87
4.3.1.	Ορισμός.....	87
4.3.2.	Παράδειγμα: Thesaurus.....	89
4.4.	Οντολογίες και SW.....	95
4.4.1.	Τι είναι οντολογία.....	95
4.4.2.	Η σύνταξη, η δομή, η σημασιολογία.....	97
4.5.	Παράδειγμα.....	104
4.5.1.	Ιεράρχηση.....	104
4.5.2.	Κατάλογοι, ιεραρχίες, και δέντρα.....	105
4.6.	Βασικές έννοιες.....	106
4.6.1.	Κλάσεις, υποκλάσεις και instances.....	106
4.6.2.	Πολλαπλές κλάσεις.....	107
4.6.3.	Κλάσεις ως σύνολα.....	109
4.7.	Ονοματολογία.....	109
4.7.1.	Ονόματα και προσδιοριστικά.....	109
4.8.	Ιδιότητες.....	110
4.9.	Κατασκευή οντολογίας.....	111
4.9.1.	Τι πρέπει να λάβουμε υπόψη.....	111
4.9.2.	Βήματα για την κατασκευή οντολογίας.....	115
4.10.	Παράδειγμα.....	119
4.11.	Γλώσσες για οντολογίες.....	121
4.11.1.	RDFS.....	121
4.11.2.	Η γλώσσα RDF.....	121
4.11.3.	OWL.....	122
5.	RDF (Resource Description Framework).....	123
5.1.	Γιατί χρειάζεται το RDF.....	123
5.2.	Τι είναι το RDF.....	126
5.3.	Άλλα χαρακτηριστικά RDF.....	129
5.3.1.	Container.....	129
5.3.2.	Reification.....	131
5.4.	Ορολογία RDF.....	133
5.4.1.	Τι περιλαμβάνει.....	133
5.5.	Παραλληλισμός: RDF και συμβατικές βάσεις δεδομένων.....	137
5.6.	Οι βασικές έννοιες.....	140
5.6.1.	Πόροι.....	140
5.6.2.	Οι ιδιότητες.....	141
5.6.3.	Οι δηλώσεις.....	141
5.7.	Τρεις μορφές μιας δήλωσης.....	141
5.8.	Reification.....	143
5.9.	Datatypes.....	144
5.10.	RDF: Σύνταξη βασισμένη σε XML.....	145
5.10.1.	Το rdf:resource.....	147
5.10.2.	Εμφωλευμένες περιγραφές.....	149
5.10.3.	Το στοιχείο rdf:type.....	149
5.10.4.	Τα στοιχεία containers.....	150
5.10.5.	Reification.....	154

5.10.6.	RDF Properties.....	155
5.10.7.	Οι ιδιότητες (properties) ως πόροι.....	155
5.11.	Ονόματα, ετικέτες και προσδιοριστικά.....	156
5.11.1.	Ιδιότητες των δηλώσεων.....	157
5.12.	Απεικόνιση των statements με RDF γράφους.....	157
5.12.1.	Πόροι με πολλές δηλώσεις.....	158
5.12.2.	Ανώνυμοι κόμβοι.....	159
5.12.3.	Οι πόροι ως αντικείμενα δηλώσεων.....	160
5.12.4.	Container κόμβοι.....	162
5.12.5.	Γραφική αναπαράσταση των ιδιοτήτων των δηλώσεων.....	163
5.13.	Παράδειγμα: Webscripiter: fusing information.....	164
5.14.	Τι είναι το σχήμα RDF.....	166
5.14.1.	Παραλληλισμός UML Class diagram με RDFS.....	167
5.14.2.	Τα βασικά συστατικά του σχήματος RDF είναι τα εξής:.....	168
5.14.3.	Παράδειγμα Griffith University.....	172
5.15.	RDF και στρώματα RDFS.....	175
5.15.1.	RDF Schema: Η γλώσσα.....	176
5.15.2.	Παράδειγμα: University.....	180
6.	OWL (Web Ontology Language).....	182
6.1.	Οι περιορισμοί της εκφραστικής δύναμης των RDF και RDF Schema.....	182
6.2.	Συμβατότητα της OWL με τα RDF/RDFS.....	183
6.3.	Τρία είδη OWL.....	184
6.3.1.	OWL Full.....	184
6.3.2.	OWL DL.....	184
6.3.3.	OWL Lite.....	185
6.4.	Η γλώσσα OWL.....	187
6.4.1.	Σύνταξη OWL.....	187
6.4.2.	Header.....	188
6.4.3.	Class Elements.....	189
6.4.4.	Property Elements.....	189
6.4.5.	Property Restrictions.....	191
6.4.6.	Special Properties.....	194
6.4.7.	Boolean Combinations.....	194
6.4.8.	Enumerations.....	196
6.4.9.	Instances.....	196
6.4.10.	Data Types.....	198
6.4.11.	Πληροφορίες για την έκδοση.....	198
6.5.	Παράδειγμα.....	199
7.	Agents.....	203
7.1.	Ευφυείς πράκτορες: τι είναι.....	203
7.2.	Βασικοί τύποι δεδομένων.....	204
7.3.	Αρχιτεκτονικές με στρώματα.....	204
7.4.	Αλληλεπιδράσεις πρακτόρων.....	205
7.5.	Η εξέλιξη των πρακτόρων.....	207
7.6.	Πλαίσια και πρότυπα.....	207
8.	Επίλογος-Συμπεράσματα.....	210
8.1.	Συμπεράσματα.....	210
8.2.	Το μέλλον.....	210
8.3.	Σύνοψη.....	211
8.4.	Μελλοντικές επεκτάσεις.....	213

9. Βιβλιογραφία.....214

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΑΙΑ

1. Σημασιολογικός Ιστός

1.1. Υπάρχοντα εργαλεία εκμάθησης του SW

Αν κάποιος ενδιαφερθεί και αναζητήσει υλικό σχετικά με το Semantic Web, σίγουρα θα βρει αρκετές πληροφορίες σε έντυπη μορφή, καθώς πληθώρα βιβλίων έχουν γραφτεί σχετικά με το διαδίκτυο και τη μορφή που αναμένεται να λάβει αυτό στο άμεσο μέλλον.

Πρώτος ο Tim Berners-Lee, ο διευθυντής του World Wide Web Consortium, ο οποίος και συνέλαβε την ιδέα, αναφέρθηκε στο μέλλον του σημερινού ιστού και έχει γράψει ήδη αρκετά βιβλία σχετικά, που περιλαμβάνουν αναλυτική περιγραφή της νέας μορφής του διαδικτύου, της αρχιτεκτονικής του και των τρόπων αναπαράστασης της γνώσης που θα χρησιμοποιηθούν. Πληθώρα από tutorials και συγγράμματα υπάρχουν σήμερα όσον αφορά στο όραμα του Σημασιολογικού Ιστού, τη χρησιμότητά του για την καλύτερη οργάνωση και διαχείριση της πληροφορίας, και τη συμβολή του στην καλύτερη και αποδοτικότερη λειτουργία ολόκληρων επιχειρήσεων.

Επιπλέον, υπάρχουν πολλοί ιστοχώροι που ασχολούνται με το θέμα και προσφέρουν, εκτός από πληροφορίες, και τα απαραίτητα εργαλεία για την υλοποίηση του σημασιολογικού ιστού, καθώς και συγκεκριμένα παραδείγματα ως προς τη χρήση του για τη λήψη αποφάσεων, τη διαχείριση της γνώσης και την ανάπτυξη των επιχειρήσεων.

Αναφέρουμε ενδεικτικά τα εξής: www.w3.org, www.semanticweb.org, http://en.wikipedia.org/wiki/Semantic_Web, <http://protégé.stanford.edu>, http://www.altova.com/products/semanticworks/semantic_web_rdf_owl_editor.html, κλπ.

1.2. Semantic Web Tutorial

Όλες οι πληροφορίες που διατίθενται στο διαδίκτυο είναι απευθείας προσβάσιμες από οποιονδήποτε. Ωστόσο, στον Ιστό σήμερα υπάρχουν πάρα πολλές πηγές με τεράστιες ποσότητες πληροφορίας και ποικίλης ποιότητας. Κάποιες πληροφορίες ισχύουν, μερικές ίσως είναι ξεπερασμένες, κάποιες λανθασμένες, και πολλές από αυτές αναφέρονται σε περιττές λεπτομέρειες. Καθένας που επιθυμεί να μάθει κάτι για τον σημασιολογικό Ιστό, ή που επιθυμεί να οργανώσει μια σειρά μαθημάτων για το σημασιολογικό Ιστό, βρίσκεται αντιμέτωπος με αυτά τα προβλήματα. Αυτό το εργαλείο προορίζεται να βοηθήσει. Ένα εγχειρίδιο πρέπει να είναι εκλεκτικό στα θέματα που καλύπτει, να μπορεί, έως κάποιο βαθμό, να λάβει υπόψη τις ανάγκες του χρήστη και φυσικά να επικεντρωθεί στις θεμελιώδεις πτυχές του αντικείμενου, που εύλογα αναμένεται να παραμείνουν χρήσιμες για κάποιο χρονικό διάστημα στο μέλλον.

Η χρησιμότητα του συγκεκριμένου εργαλείου, έναντι των εντύπων και των on-line εγχειριδίων που διατίθενται στο Web, έγκειται στην παρουσίαση της θεωρίας που αφορά στον σημασιολογικό Ιστό, σχετικά με την περιγραφή της μορφής του, της χρησιμοποιούμενης αρχιτεκτονικής και των εργαλείων που πρέπει να χρησιμοποιηθούν, σε συνδυασμό όμως με τα αντίστοιχα παραδείγματα, την συγκεκριμένη μεθοδολογία που πρέπει να ακολουθηθεί σε κάθε περίπτωση αλλά και την αξιολόγηση της βαθύτερης κατανόησης αυτών μέσα από την πρακτική εξάσκηση του ενδιαφερόμενου, καθώς σε κάθε κεφάλαιο παρατίθενται κάποιες προτεινόμενες ασκήσεις.

Επιπλέον, η παρουσίαση του περιεχομένου κάθε κεφαλαίου συγκεντρωτικά, ανάλογα με το αν πρόκειται για θεωρία, άσκηση ή παράδειγμα, είναι αρκετά βοηθητική για τον εκάστοτε χρήστη, ώστε να επικεντρωθεί στις θεμελιώδεις πτυχές κάθε φορά, ανάλογα με τις ανάγκες και τις τυχόν ελλείψεις που μπορεί ο ίδιος να έχει, αλλά και τις όποιες απορίες μπορεί να προκύψουν κατά τη διάρκεια υλοποίησης κάποιου έργου.

Τέλος, η επίδειξη χρήσης συγκεκριμένου λογισμικού με λεπτομερή περιγραφή των ενεργειών και των βημάτων που ακολουθούνται, επιτρέπει στη συνέχεια στον εκπαιδευόμενο, και παρακινεί θα λέγαμε πολύ περισσότερο από οποιονδήποτε άλλο τρόπο, να πειραματιστεί μόνος του πάνω στο συγκεκριμένο αντικείμενο και έτσι να κάνει κτήμα του τις γνώσεις που τα παρασχέθηκαν με τη βοήθεια του tutorial.

Με τη βοήθεια του συγκεκριμένου εργαλείου επικεντρωνόμαστε στις κύριες ιδέες και τεχνικές και παρέχουμε αρκετή λεπτομέρεια για να επιτρέψουμε στον ενδιαφερόμενο να εμπεδώσει το υλικό δημιουργικά και να είναι στη συνέχεια σε θέση να υλοποιήσει εύκολα και σε ικανοποιητικό βαθμό δικές του εφαρμογές.

1.3. Σκοπός της διπλωματικής εργασίας

Η συγκεκριμένη εργασία αποτελεί ένα tutorial για τον Σημασιολογικό Ιστό, το οποίο διακρίνεται από ξεχωριστά κεφάλαια, σύμφωνα με το περιεχόμενο που κρίνεται απαραίτητο να παρουσιαστεί και τοποθετημένα σε τέτοια σειρά, που επιτρέπει την κατανόηση του περιεχομένου του καθενός, εφόσον έχουν καλυφθεί τα προαπαιτούμενα στις ενότητες που προηγούνται. Οι στόχοι που καλύπτει στο σύνολό του είναι οι εξής:

- Να εξηγήσει το όραμα για τον Σημασιολογικό Ιστό όσον αφορά στη δημιουργία δεδομένων που θα είναι κατανοητά από τον ίδιο τον υπολογιστή.
- Να τονίσει τη χρησιμότητά του, γιατί χρειαζόμαστε το SW και πώς θα διευκολυνθούμε όταν υλοποιηθεί αυτή η μορφή του Web.
- Να παρουσιάσει και να περιγράψει τις τεχνολογίες που χρησιμοποιούνται για την υλοποίηση του Semantic Web, οι οποίες μπορούν να χρησιμοποιηθούν και ευρύτερα.
- Να βοηθήσει το χρήστη να κατανοήσει τις καινούριες έννοιες μέσα από έτοιμα παραδείγματα.
- Να συμβάλλει στη βαθύτερη κατανόηση από την πλευρά του χρήστη των νέων τεχνολογιών που χρησιμοποιούνται προς ανάπτυξη του σημασιολογικού ιστού, μέσα από προτεινόμενες ασκήσεις.
- Να παρακινήσει τον εκπαιδευόμενο να πειραματιστεί ο ίδιος με προτεινόμενα λογισμικά, ώστε να οικειοποιηθεί τις γνώσεις που απέκτησε.

Η εργασία έχει χωριστεί σε ενότητες, καθμία από τις οποίες είναι εξέχουσας σημασίας για την κατανόηση του όρου «Semantic Web», τόσο σε θεωρητικό επίπεδο, όσο και σε επίπεδο πρακτικής εφαρμογής. Στόχος της παρούσας εργασίας είναι να αναπτύξει μία μεθοδολογία για την εκμάθηση της έννοιας του σημασιολογικού

ιστού, τη σύνθεση και τον εμπλουτισμό της γνώσης που αφορά στο συγκεκριμένο αντικείμενο. Η μεθοδολογία αυτή υλοποιείται ακολουθώντας μία συγκεκριμένη δομή, μέσα από την οποία παρουσιάζονται τα βασικά σημεία και στη συνέχεια αναλύονται όσο το δυνατό λεπτομερέστερα, ακολουθώντας τη λογική της μετάβασης από το γενικότερο προς το ειδικότερο, έτσι ώστε να μην προκύπτουν δυσκολίες στην κατανόηση, εφόσον ο χρήστης θα έχει καλύψει γνωστικά όσα προαπαιτούνται για να προχωρήσει στο επόμενο στάδιο.

Το συγκεκριμένο tutorial αποτελεί επιπλέον ένα εργαλείο e-learning, σε επίπεδο ασύγχρονης μάθησης, καθώς ο εκπαιδευτής, έχοντας προετοιμάσει το εκπαιδευτικό υλικό, το παρέχει στον εκπαιδευόμενο, ο οποίος με τη σειρά του μπορεί να πάρει αυτό το εκπαιδευτικό υλικό και να εκπαιδευτεί μόνος του, οπουδήποτε και αν βρίσκεται και σε όποια χρονική στιγμή επιλέξει ο ίδιος. Επιπλέον, παρέχεται η δυνατότητα επικοινωνίας με τον διδάσκοντα, για την επίλυση τυχόν αποριών που ενδέχεται να προκύψουν από την πλευρά του εκπαιδευόμενου.

1.3.1. Σημσιολογικός ιστός και e-applications

Η πρωτεύουσα σημασία της παρούσας εργασίας σχετικά με τον Σημσιολογικό Ιστό τονίζεται μέσα από την κυρίαρχη θέση του τελευταίου σε όλο και περισσότερες εκφάνσεις της καθημερινής μας ζωής. Ενδεικτικά αναφέρουμε κάποιους από τους βασικότερους τομείς, στα πλαίσια των οποίων βρίσκει εφαρμογή το Semantic Web.

1.3.1.1 Ηλεκτρονική διακυβέρνηση

Η αποτελεσματικότητα οποιωνδήποτε κυβερνήσεων απεικονίζεται με την αποδοτικότητα, τις καλύτερες υπηρεσίες στους πολίτες και τις βελτιωμένες κυβερνητικές διαδικασίες της. Πολλές κυβερνήσεις επομένως έχουν αρχίσει τις επιθετικές εκστρατείες για να αυξήσουν απέραντα τον αριθμό των on-line αλληλεπιδράσεων με τους πολίτες και για να παρέχουν μεγάλη ποσότητα on-line πληροφοριών και γνώσης σε αυτούς.

Για να ενθαρρυνθεί η χρήση όλων αυτών των πληροφοριών και της γνώσης, μια κυβέρνηση πρέπει να είναι σε θέση να παρέχει έναν κατάλληλο τρόπο για τους πολίτες της, ώστε να έχουν πρόσβαση και να λαμβάνουν τις πληροφορίες και τη γνώση που επιθυμούν.

Ως εκ τούτου, για να χειριστεί την οργάνωση της γνώσης και την πρόσβαση σε αυτήν με έναν πιο εφικτό τρόπο, μια κυβέρνηση πρέπει να προσθέσει κάποια λογική, μια σημασιολογική δομή, για να οργανώσει τις πληροφορίες που παρέχει. Οι σύνδεσμοι υπερ-κειμένου μπορούν να χρησιμοποιηθούν για να παρέχουν τις πληροφορίες στους πολίτες με μια σχετικά λίγη προσπάθεια. Εντούτοις, η οργάνωση των συνδέσμων με έναν εύχρηστο τρόπο είναι μια μεγάλη πρόκληση για τους υπεύθυνους για την ανάπτυξη ιστοχώρων, και οι σύνδεσμοι υπερ-κειμένου στη συμβατική HTML δεν έχουν μια ευπροσδιόριστη έννοια.

Μια προσέγγιση για να λυθεί αυτό το πρόβλημα είναι η ανάπτυξη ενός σημασιολογικού Ιστού. Ένας τέτοιος Ιστός μπορεί να χρησιμοποιηθεί για να αναπτύξει μια αποτελεσματικότερη και διαφανή e-διακυβέρνηση. Οι υπεύθυνοι για την ανάπτυξη ιστοχώρων θα χρησιμοποιήσουν στις ιστοσελίδες σημασιολογική σήμανση, σημασιολογικούς συνδέσμους και μεταδεδομένα, ώστε να επιτραπεί στις μηχανές να ακολουθήσουν τους συνδέσμους και να διευκολύνουν την συμπλήρωση της γνώσης και της πληροφορίας από πολλές διαφορετικές πηγές.

Η τεχνολογία του σημασιολογικού Ιστού έχει χρησιμοποιηθεί για τη διαχείριση της γνώσης στην ηλεκτρονική διακυβέρνηση, με τις οντολογίες να παρέχουν ένα καλά ορισμένο μέσο ώστε οι χρήστες να προσεγγίσουν τη γνώση αποτελεσματικά. Παραδείγματος χάρη, ο Klischewski (2003) προώθησε ένα πρόγραμμα στο πανεπιστήμιο του Αμβούργου για να εφαρμοστεί η τεχνολογία του σημασιολογικού Ιστού και να επιτρέψει το contextualisation DiBIS (www.dibis.de), μια υπηρεσία πληροφοριών Ιστού για τους πολίτες του Αμβούργου. Στα πλαίσια ενός άλλου προγράμματος, αναπτύχθηκε η SmartGov οντολογία για να παρέχει στη δημόσια αρχή μια αποθήκη βασικών γνώσεων για τις υπηρεσίες κυβερνητικής συναλλαγής.

1.3.1.2 Ηλεκτρονικό εμπόριο B2C

Το ηλεκτρονικό εμπόριο της μορφής business-to-consumer είναι κυρίαρχο στη χρήση του Web. Ένα χαρακτηριστικό σενάριο περιλαμβάνει έναν χρήστη που επισκέπτεται ένα ή περισσότερα on-line καταστήματα, εξετάζει τις προσφορές τους, επιλέγει τα προϊόντα. Πιο συγκεκριμένα, ένας χρήστης θα συνέλεγε τις πληροφορίες για τις τιμές, τους όρους, και την κατάσταση (όπως η διαθεσιμότητα) και θα προχωρούσε έπειτα να επιλέξει την καλύτερη προσφορά. Αλλά το εγχειρίδιο που κοιτάζει συνήθως βιαστικά απαιτεί πολύ χρόνο για να πραγματοποιηθεί κάτι τέτοιο.

Χαρακτηριστικά, ένας χρήστης θα επισκεφτεί το πολύ ένα ή σχετικά λίγα on-line καταστήματα πριν λάβει μια απόφαση.

Προς διευκόλυνση της παραπάνω διαδικασίας, πράκτορες λογισμικού που επισκέπτονται διάφορα καταστήματα, εξάγουν το προϊόν και τις πληροφορίες για τις τιμές και συντάσσουν μια έκθεση αγοράς. Οι πληροφορίες εξάγονται από την on-line περιοχή καταστημάτων μέσω της αναζήτησης με βάση τη λέξη - κλειδί και άλλων μέσων της ανάλυσης κειμένων. Αυτή η διαδικασία βασίζεται σε συγκεκριμένες υποθέσεις (όπως για παράδειγμα, η τιμή υποδεικνύεται από την τιμή της λέξης που ακολουθείται από έναν θετικό αριθμό και από το σύμβολο \$). Αυτή η ευρετική προσέγγιση είναι ωστόσο επιρρεπής σε λάθη και δεν είναι πάντα δεδομένο ότι θα λειτουργήσει σωστά. Λόγω αυτών των δυσκολιών, μόνο περιορισμένες πληροφορίες εξάγονται. Παραδειγματος χάριν, οι δαπάνες μεταφοράς, οι χρόνοι παράδοσης, οι τυχόν απαγορεύσεις στη χώρα προορισμού, το επίπεδο ασφάλειας είναι χαρακτηριστικά που δε γίνονται γνωστά. Αλλά όλοι αυτοί οι παράγοντες μπορούν να είναι σημαντικοί για τη λήψη απόφασης από την πλευρά του χρήστη.

Το SemanticWeb θα επιτρέψει την ανάπτυξη των πρακτόρων λογισμικού οι οποίοι έχουν τη δυνατότητα να ερμηνεύουν τις πληροφορίες για τα προϊόντα και τους όρους της κάθε υπηρεσίας.

- Η τιμολόγηση και οι πληροφορίες προϊόντων θα εξαχθούν σωστά, και οι πολιτικές παράδοσης και μυστικότητας θα ερμηνευθούν και θα συγκριθούν με τις απαιτήσεις των χρηστών.

- Οι πρόσθετες πληροφορίες για τη φήμη των on-line καταστημάτων θα ανακτηθούν από άλλες πηγές, παραδειγματος χάριν, ανεξάρτητες αντιπροσωπείες εκτίμησης ή καταναλωτικούς οργανισμούς.

- Οι πράκτορες θα είναι σε θέση να προβούν σε αυτοματοποιημένες διαπραγματεύσεις, εκ μέρους του αγοραστή, με τους αντίστοιχους πράκτορες των καταστημάτων.

1.3.1.3 E-health

Οι αναζητήσεις της πληροφορίας μέσω των διακριτών keywords από τις κλασσικές μηχανές αναζήτησης, δεν είναι ικανές να αποδώσουν το αντιληπτικό και σημασιολογικό περιεχόμενο της ερώτησης του ενδιαφερομένου. Λόγω, της μεγάλης σημασίας των εφαρμογών της ιατρικής στην ποιότητα της ανθρώπινης ζωής, οι

διάφοροι οργανισμοί παροχής υγείας επισπεύσανε τις ερευνητικές προσπάθειες σχετικά με το σημασιολογικό ιστό. Θα μπορούσε να ειπωθεί ότι το πεδίο της ιατρικής είναι ένας από τους πρωτοπόρους όσον αφορά στην εφαρμογή και αξιοποίηση της ιδέας του σημασιολογικού ιστού [10,11]. Μία από τις σημαντικότερες μηχανές αναζήτησης είναι η PubMed που αναπτύχθηκε από την NLM και αφορά στην αναζήτηση βιβλιογραφικών εγγραφών από το ιατρικό πεδίο.

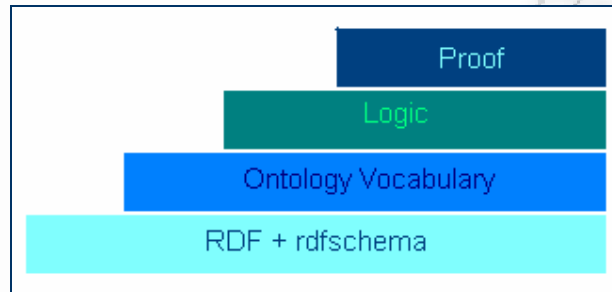
Τα διεθνή δίκτυα περίθαλψης ασθενών τα τελευταία χρόνια έχουν ως στόχο την διευκόλυνση του διαμοιρασμού της πληροφορίας που σχετίζεται με την περίθαλψη και την υγεία, αλλά και την κάλυψη των ιατρικών αναγκών για πληθυσμούς που βρίσκονται σε κίνηση (mobile population). Στόχος της δημιουργίας του **Virtual Electronic HealthCare Record (EHCR)** είναι ένα περιβάλλον στο οποίο θα παρέχεται ολοκληρωμένη και συνεχής περίθαλψη στους ασθενείς όπου και αν βρίσκονται.

Ένας τέτοιος φάκελος υγείας θεωρείται το κλειδί για τις υπηρεσίες που είναι εξουσιοδοτημένες από ένα Healthcare Information Infrastructure (HII) και έχουν κύριο προσανατολισμό τον χρήστη. Ετερογενή αυτόνομα συστήματα έχουν σχεδιαστεί ώστε να υποστηρίζουν συγκεκριμένου τύπου πληροφορίες καθώς και τμήματα του ηλεκτρονικού φακέλου υγείας. Για να επιτευχθεί η συνένωση των κατανεμημένων τμημάτων του ηλεκτρονικού φακέλου υγείας απαιτείται άμεση επικοινωνία των ετερογενών συστημάτων ώστε η πληροφορία σε ιατρικά θέματα να γίνεται μέσω μηνυμάτων με standard μορφή. Βέβαια, επειδή η περιεκτική ιατρική πληροφορία για ασθενείς είναι δύσκολο να αποκτηθεί, κρίνεται απαραίτητη η ένταξη των κατανεμημένων ετερογενών ηλεκτρικών φακέλων υγείας σε ένα ενιαίο Virtual Electronic HealthCare Record (EHCR). Η πληροφορία αυτή βρίσκεται on-line μέσω ενιαίας διεπαφής σε ένα περιβάλλον οπτικοποίησης.

Όμως για να πραγματοποιηθεί το Virtual Electronic HealthCare Record (EHCR) πρέπει να ελέγχονται πλήρως όλες οι πηγές πληροφοριών σε επίπεδο μεταδεδομένων αλλά παράλληλα να διατηρείται η αυτονομία των συστημάτων την στιγμή που αποθηκεύονται τα δεδομένα. Συνεπώς, κρίνεται απαραίτητη η ενοποίηση σε σημασιολογικό επίπεδο.

1.3.1.4 E-learning

Η κύρια ευθύνη του Σημασιολογικού Ιστού είναι μπορεί να συλλαμβάνει και να εκφράζει έννοιες και σημασίες. Προκειμένου να επιτύχει τον ανωτέρω στόχο, ο Σημασιολογικός Ιστός είναι σχεδιασμένος με αρχιτεκτονική διαφόρων επιπέδων.



Τα επίπεδα του Σημασιολογικού Ιστού

Η παραπάνω αρχιτεκτονική φαίνεται ότι ικανοποιεί τις απαιτήσεις του e-Learning [3]. Ως ιδέα λειτουργεί ως εξής: η οντολογία παρέχει το υπόβαθρο για να μπορούν να γίνουν σημασιολογικές επερωτήσεις και πλοήγηση στα δεδομένα που περιγράφουν οι RDF και XML, με τη βοήθεια εργαλείων μάθησης. Οι δύο έννοιες ταιριάζουν απόλυτα. Στην πραγματικότητα, ο Σημασιολογικός Ιστός μπορεί να χρησιμοποιηθεί ως πλατφόρμα υλοποίησης ενός συστήματος e-Learning και να ικανοποιεί τις απαιτήσεις του.

Μέσω του Σημασιολογικού Ιστού, το υλικό προς μάθηση διανέμεται και μπορεί να συνδεθεί με οντολογίες, για να διαμορφώσει εξειδικευμένο για το χρήστη περιβάλλον, σχετικό με τα ενδιαφέροντά του. Η δυνατότητα αυτή εξυπηρετεί την απαίτηση για διανομή πληροφορίας ατομικά στον εκπαιδευόμενο, με ύλη που εκείνος ορίζει και που αυτή ανταποκρίνεται στις εκάστοτε ανάγκες του. Επιπλέον, οι πράκτορες του Ιστού δίνουν τη δυνατότητα για επερωτήσεις που αφορούν σε ετερόκλητα τμήματα της πληροφορίας. Αυτή, δηλαδή, δεν παρέχεται με «γραμμικό» τρόπο σε συνέχειες, αλλά η πρόσβαση γίνεται σε διάφορα τμήματά της. Η δυνατότητα αυτή σε ένα περιβάλλον e-Learning προσφέρει το πλεονέκτημα της γρήγορης και επικεντρωμένης στην ουσία μάθησης και δεδομένου ότι αυτή θα ανανεώνεται συνεχώς, ο εκπαιδευόμενος επιτυγχάνει συνεχή πρόοδο. Τα παραπάνω αναδεικνύουν την αλληλεπίδραση του χρήστη με τον Ιστό, ως βασική προϋπόθεση λειτουργίας του e-Learning, που σημαίνει ότι ο τρόπος λειτουργίας δεν είναι

κεντρικοποιημένος, αλλά κατανεμημένος. Ο ακόλουθος πίνακας συνοψίζει τις σχέσεις αυτές:

Απαιτήσεις	e-Learning	Σημασιολογικός Ιστός
Διανομή γνώσεων	Ο μαθητευόμενος αποφασίζει για την ύλη του (pull)	Το υλικό συγκεντρώνεται κατόπιν επερωτήσεων
Ανταπόκριση	Ανταποκρίνεται στις ανάγκες που εντοπίζει	Το υλικό παρέχεται μέσω προφίλ χρήστη
Πορεία	Μη γραμμική, επιτρέπει απ' ευθείας πρόσβαση σε οποιοδήποτε κομμάτι γνώσης	Πλοήγηση στις πληροφορίες του ενδιαφέροντος του χρήστη
Πρόοδος	Συνεχής, η μάθηση δε σταματάει ποτέ	Η βάση των πληροφοριών ανανεώνεται
Αρχή	Κατανεμημένη, διδάσκων και διδασκόμενος συναποφασίζουν	Αναγκαία η αλληλεπίδραση χρήστη-συστήματος
Εξατομίκευση	Προσωποποιημένη, σχεδιασμένη για τις ανάγκες κάθε μαθητευόμενου	Ο χρήστης αναζητεί πληροφορίες και η οντολογία συνδέει το πληροφοριακό υλικό με τις συγκεκριμένες αναζητήσεις του
Προσαρμοστικότητα	Δυναμική, το περιεχόμενο της γνώσης αλλάζει σύμφωνα με την εμπειρία του μαθητευόμενου	Όλα τα παραπάνω

e-Learning & Σημασιολογικός Ιστός

1.3.2. Χρησιμότητα του SW Tutorial

Ένας εκπαιδευόμενος που έχει ολοκληρώσει επιτυχώς το tutorial θα είναι σε θέση να:

- συζητήσει και να εξηγήσει τις βασικές αρχές του σημασιολογικού ιστού
- συζητήσει και να αναλύσει ό,τι αφορά στις γλώσσες για τις οντολογίες RDF και OWL.
- γνωρίζει τη δομή και τις αρχές που διέπουν τη διαδικασία ανάπτυξης μίας οντολογίας και να κρίνει σωστά τις υπάρχουσες εφαρμογές για το Semantic Web
- κατασκευάσει σωστά οντολογίες κάνοντας χρήση του εργαλείου Protégé, δικαιολογώντας και αξιολογώντας τον τρόπο σχεδιασμού τους.

1.4. Δομή του tutorial

Η παρούσα εργασία έχει χωριστεί σε πέντε ενότητες, όσον αφορά στην υλοποίησή της.

Στην πρώτη ενότητα, το “*Semantic Web*”, γίνεται μια εκτενής αναφορά στην ανάγκη που έχει προκύψει πλέον για την εισαγωγή της σημασιολογίας στον Ιστό, παρουσιάζοντας τις ελλείψεις και τα προβλήματα που χαρακτηρίζουν σήμερα το WWW. Παρέχεται ένας ορισμός του σημασιολογικού ιστού και προβάλλονται οι δυσκολίες που θα προκύψουν μέσα από την εφαρμογή του, ενώ ακολουθεί μια περιγραφή της αρχιτεκτονικής του Semantic Web και μία σύντομη αναφορά στις τεχνολογίες που χρησιμοποιούνται για την υλοποίησή του.

Στη δεύτερη ενότητα, που καλείται “*Ontologies*”, παρουσιάζεται η ανάγκη για ταξινόμηση, ο ορισμός της ταξινόμησης και η αναλυτική περιγραφή της μέσα από συγκεκριμένα παραδείγματα, ενώ στη συνέχεια γίνεται μετάβαση στην έννοια της οντολογίας, της σημασίας που αυτή έχει για τον σημασιολογικό ιστό και περιγραφή των χαρακτηριστικών της. Τέλος, παρατίθεται συγκεκριμένη και λεπτομερής μεθοδολογία για την κατασκευή μίας οντολογίας. Και στο σημείο αυτό παρατίθενται αντιπροσωπευτικά παραδείγματα και εφαρμογές οντολογιών. Εδώ προτείνονται και συγκεκριμένες ασκήσεις για τον εκπαιδευόμενο, ώστε να εμπεδώσει τις έννοιες που έμαθε και να προχωρήσει στο επόμενο στάδιο.

Στην τρίτη ενότητα, “*RDF*”, περιγράφεται το Resource Description Framework, η σημασία του για τον σημασιολογικό ιστό, η ορολογία, οι βασικές έννοιες και τα χαρακτηριστικά του και παρατίθενται τα αντίστοιχα παραδείγματα σε κάθε περίπτωση. Στο ίδιο κεφάλαιο παρουσιάζεται και το σχήμα RDF, με την ορολογία που αυτό χρησιμοποιεί και αντιπροσωπευτικά παραδείγματα. Και πάλι προτείνονται συγκεκριμένες ασκήσεις και δραστηριότητες που αφορούν στο συγκεκριμένο αντικείμενο.

Στην τέταρτη ενότητα, “*OWL*”, γίνεται εκτενής αναφορά και περιγραφή της Web Ontology Language, αφού πρώτα έχει προηγηθεί η σημασία της, η οποία προκύπτει από τους περιορισμούς που διακρίνουν την RDF που έχει ήδη παρουσιαστεί. Παρατίθενται τα χαρακτηριστικά της συγκεκριμένης γλώσσας, οι κανόνες σύνταξής της, συγκεκριμένα παραδείγματα και προτεινόμενες ασκήσεις. Στην συγκεκριμένη ενότητα παρουσιάζεται ένα ολοκληρωμένο παράδειγμα κατασκευής μίας οντολογίας κάνοντας χρήση του εργαλείου Protégé, σε μορφή βίντεο, έτσι ώστε ο χρήστης να δει

και σε εφαρμογή την ανάπτυξη της οντολογίας και να είναι σε θέση να προβεί και ο ίδιος σε ανάλογη δραστηριότητα.

Στην πέμπτη ενότητα, "Agents", περιγράφονται οι πράκτορες και η σημασία τους για τον σημασιολογικό ιστό, καθώς και οι αλληλεπιδράσεις αυτών μέσα στο περιβάλλον του Semantic Web.

1.5. Σε ποιον απευθύνεται

Ουσιαστικά, οποιοσδήποτε ασχολείται με λογισμικό και ενδιαφέρεται για τη μελλοντική μορφή του World Wide Web θα βρει χρήσιμο το συγκεκριμένο tutorial, το οποίο συμβάλλει σε ικανοποιητικό βαθμό στην ουσιαστική κατανόηση των βασικών χαρακτηριστικών του οράματος του σημασιολογικού ιστού. Δεν πρόκειται για ένα εγχειρίδιο προγραμματισμού, αλλά στρέφεται γύρω από την παρουσίαση και αναλυτική περιγραφή των θεμελιωδών εννοιών του Semantic Web. Έτσι, τόσο φοιτητές εκπαιδευτικών ιδρυμάτων (σε προπτυχιακό ή μεταπτυχιακό επίπεδο) όσο και οι ίδιοι οι εκπαιδευτικοί, που για ερευνητικούς ή παιδαγωγικούς σκοπούς ενδιαφέρονται για το διαδίκτυο και την εξέλιξή του, θα ωφεληθούν μελετώντας το συγκεκριμένο tutorial, το οποίο μπορεί να αποτελέσει τόσο ένα προσωπικό εργαλείο εκμάθησης όσο και έναν οδηγό για μία σειρά μαθημάτων πανεπιστημιακού επιπέδου σε μορφή e-learning.

Ωστόσο, η τάση προς το σημασιολογικό Ιστό είναι βασισμένη στο στόχο της σημασιολογικής διαλειτουργικότητας, η οποία επιτρέπει την ανεξαρτησία εφαρμογών, τις βελτιωμένες τεχνικές αναζήτησης, και την κατάληξη σε συμπεράσματα από την πλευρά των ίδιων των υπολογιστών. Οι ίδιες οι επιχειρήσεις λοιπόν δεν αγνοούν τις τάσεις αυτές, καθώς έχουν επιπτώσεις ή βελτιώνουν σημαντικές εφαρμογές. Ο σημασιολογικός Ιστός μπορεί να υλοποιηθεί σε χιλιάδες πιλοτικά έργα, σε διαφορετικές βιομηχανίες, όπως την επιστήμη, τη δικαιοσύνη, την ιατρική, και τη χρηματοδότηση βιβλιοθηκών. Έτσι, πολλές επιχειρήσεις που επιθυμούν να χρησιμοποιήσουν προϊόντα σημασιολογικού Ιστού καλούνται να διεισδύσουν στη νέα αυτή μορφή του Web.

Η χρήση του συγκεκριμένου εργαλείου από τις παραπάνω διαφορετικές ομάδες χρηστών υπαγορεύει την ανάγκη για ανάπτυξη μίας εφαρμογής εύχρηστης, τόσο από γνώστες της τεχνολογίας, όσο και από χρήστες που πιθανότατα δεν έχουν

αποκτήσει κάποια εξοικείωση μέχρι στιγμής. Ένα τέτοιο περιβάλλον, όντας παράλληλα ευχάριστο και ευκολονόητο, θα είναι εφικτό να κερδίσει το ενδιαφέρον των χρηστών και να προάγει την παραγωγικότητά τους.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΑΙΑΣ

2. Semantic Web Tutorial

2.1. Εκπαιδευτική φιλοσοφία του Tutorial

Η μάθηση μέσω της ανάλυσης περιπτώσεων (**Case-based Learning**) [6] είναι μια δημοφιλής στρατηγική διδασκαλίας στην τριτοβάθμια εκπαίδευση, ιδιαίτερα στα πτυχιακά προγράμματα σε επαγγελματικό επίπεδο, όπως η διοίκηση, η εκπαίδευση, η ιατρική, η νομική και οι πολιτικές μελέτες. Αυτή η στρατηγική επιτρέπει στους σπουδαστές να εφαρμόσουν τις ιδέες και τις διάφορες θεωρίες σε πραγματικές καταστάσεις επίλυσης προβλημάτων. Υπάρχει άμεση σχέση μεταξύ του on-line μαθησιακού περιβάλλοντος και του on-line επαγγελματικού κόσμου, οι εκπαιδευτικοί έχουν νέες ευκαιρίες να καθοδηγήσουν τους εκπαιδευόμενους μέσω της ανάλυσης μελετών περιπτώσεων, συμβάλλοντας έτσι στη βελτίωση της on-line μάθησης.

Μια μελέτη περιπτώσεων είναι μια καλά διερευνημένη ιστορία. Κάθε ιστορία περιλαμβάνει ένα δίλημμα και οι εκπαιδευόμενοι καλούνται να το επιλύσουν με τέτοιο τρόπο μέσα από τον οποίο αναδεικνύεται η επίτευξη συγκεκριμένων μαθησιακών στόχων. Πιο συγκεκριμένα, σε μία μελέτη περιπτώσεων παρέχεται ένα στιγμιότυπο ενός οργανισμού σε μια συγκεκριμένη χρονική στιγμή. Οι περιπτώσεις προετοιμάζονται αφού έχει πραγματοποιηθεί μία εις βάθος έρευνα καθώς και συνεντεύξεις με τους βασικούς φορείς. Οι καλύτερες μελέτες ενσωματώνουν ποσοτικές και ποιοτικές πληροφορίες με διαφορετικές προοπτικές για τις διαφορετικές πλευρές της περίπτωσης. Οι μελέτες περιπτώσεων που αντιπροσωπεύουν πολλές επιστήμες είναι τώρα προσβάσιμες μέσω διαδικτύου.

Σήμερα, οι επαγγελματίες πρέπει όλο και περισσότερο να συνεργαστούν για να επιλύσουν τα διάφορα προβλήματα. Η ειδίκευση σε έναν συγκεκριμένο τομέα δεν είναι αρκετή. Κάποιος πρέπει να είναι σε θέση να κατανοήσει από κάθε πλευρά τα σύνθετα ζητήματα, τις εσωτερικές ή εξωτερικές επιπτώσεις στην επιχείρηση. Και είναι πολύ δυσκολότερο να διδάξει κανείς την κρίση, την ισορροπία των διάφορων

προοπτικών, την ομαδική εργασία και την επικοινωνία που απαιτείται, ώστε να είναι αναγνωριστέοι κάποιος και να πετύχει στο σημερινό περιβάλλον εργασίας.

Η μάθηση μέσα από την εμπειρία ενθαρρύνει τους εκπαιδευόμενους να γεφυρώσουν τη θεωρία με την πρακτική για την εφαρμογή της θεωρίας σε πρακτικό επίπεδο. Η ανάλυση περιπτώσεων παρέχει ένα είδος εμπειρικής ευκαιρίας, καθώς επιτρέπει τον πειραματισμό, χωρίς την ανησυχία για τον αντίκτυπο σε πραγματικές οργανώσεις ή πελάτες. Προωθεί μια πολύπλευρη προσέγγιση των συστημάτων και το σημαντικότερο, προσφέρει τη δυνατότητα να δει ο εκπαιδευόμενος ότι δεν υπάρχει καμία εύκολη απάντηση και ότι συχνά δεν υπάρχει ούτε η «σωστή» απάντηση.

Η ανάλυση περιπτώσεων μπορεί να εφαρμοστεί για την εκπαίδευση μεμονωμένων σπουδαστών, ομάδων ή και σε ολόκληρη τάξη, για να προωθήσει την επικοινωνία των σπουδαστών, ενώ όταν χρησιμοποιείται στην on-line εκπαίδευση συντελεί στην ανάπτυξη δεξιοτήτων όσον αφορά στις νέες τεχνολογίες. Αυτές οι δεξιότητες είναι πολύτιμες στους σπουδαστές που προετοιμάζονται για ειδικευση σχεδόν σε οποιοδήποτε τομέα.

2.1.1. Μάθηση μέσω της ανάλυσης περιπτώσεων και on-line εκπαίδευση

Στα πλαίσια της on-line εκπαίδευσης, η ανάλυση περιπτώσεων μπορεί να χρησιμοποιηθεί για να επιτύχει τους ακόλουθους στόχους:

- *Ανάπτυξη των on-line ερευνητικών δεξιοτήτων:* Σε μία μελέτη περιπτώσεων περιγράφεται μία ιστορία που πραγματοποιείται μέσα σε ένα δεδομένο χρονικό πλαίσιο. Οι σπουδαστές μπορούν να διευρύνουν το βαθμό στον οποίο έχουν κατανοήσει την ιστορία μέσω της έρευνας παλαιότερων αντίστοιχων περιπτώσεων περισσότερων τρεχουσών εξελίξεων.
- *Ανάπτυξη της ικανότητας των εκπαιδευομένων για εργασία σε εικονικές ομάδες.* Οι αναθέσεις ομάδας απαιτούν τη συμφωνία μεταξύ των σπουδαστών όσον αφορά σε κανόνες σχετικά με τη συχνότητα, την ποιότητα και τα μέσα επικοινωνίας. Ζητήματα όπως ο δίκαιος διαμοιρασμός του φόρτου εργασίας, η συμφωνία της μορφής εργασίας και των προσδοκιών για ένα πρόγραμμα πρέπει να επιλυθούν προκειμένου να ολοκληρωθεί η ανάθεση. Η εμπειρία σε αυτές τις on-line διαπραγματεύσεις και η επίλυση προβλημάτων θα

συμβάλλει στη ανάπτυξη της ικανότητας συνεργασίας των σπουδαστών "εικονικά".

- *Εισαγωγή των εκπαιδευομένων στις νέες τεχνολογίες.* Οι αναθέσεις τέτοιου είδους εργασιών μπορούν να έχουν ως σκοπό να εισαγάγουν τους σπουδαστές σε μέσα επικοινωνίας που δεν έχουν ξαναχρησιμοποιηθεί σε τάξη.
- *Η ανάπτυξη on-line μαθησιακών κοινοτήτων:* Η συζήτηση των διαφόρων απόψεων και των διαφορετικών συμπερασμάτων μπορεί να αναζωογονήσει την αλληλεπίδραση τάξεων. Με τη χρησιμοποίηση ποικίλων διαλογικών τεχνικών, η διαδικασία της ανάλυσης περιπτώσεων μπορεί να δημιουργήσει μία δυναμική κοινότητα σε μία τάξη.

Στη συνέχεια παραθέτουμε τα βήματα της ανάλυσης περιπτώσεων καθώς και συγκεκριμένα παραδείγματα των on-line στρατηγικών διδασκαλίας για κάθε στάδιο, τα οποία παρατίθενται σε μορφή δραστηριοτήτων. Φυσικά ένας εκπαιδευτικός θα επέλεγε και θα προσαρμόζε τις εκπαιδευτικές δραστηριότητες προς την εκπλήρωση των στόχων των μαθημάτων του, μέσα στα χρονικά πλαίσια που απαιτείται και με βάση το επίπεδο της τάξης του.

2.1.2. Βήματα μιας ανάλυσης περιπτώσεων

Η ανάλυση περιπτώσεων ακολουθεί μία σειρά βασικών ερωτήσεων και τα αποτελέσματα αναλύονται ώστε να βρεθεί λύση για το πρόβλημα που κωδικοποιείται σε κάθε περίπτωση. Τα βασικά βήματα περιλαμβάνουν:

- Προσδιορισμός του προβλήματος
- Σχεδιασμός της απαιτούμενης έρευνας
- Διεξαγωγή της έρευνας
- Προσδιορισμός και αξιολόγηση των εναλλακτικών λύσεων
- Επιλογή της καλύτερης πρότασης λύσης
- Παρουσίαση των αποτελεσμάτων

Κάποια από αυτά τα βήματα ή και όλα μπορούμε να τα αντιστοιχίσουμε σε on-line εκπαιδευτικές δραστηριότητες.

i) **Προσδιορισμός του προβλήματος.**

Το πρώτο βήμα για τους εκπαιδευόμενους είναι να αξιολογήσουν την περίπτωση και να καθορίσουν το κεντρικό πρόβλημά της. Πρέπει να προσδιορίσουν τα ζητήματα που παρουσιάζονται, τις προοπτικές που λαμβάνονται και τον πρωταγωνιστή της ιστορίας, να ερευνήσουν τα στοιχεία και άλλες ενισχυτικές πληροφορίες. Ένα ενιαίο πρόβλημα μπορεί να είναι προφανές, βασισμένο σε αυτήν την αρχική ανάγνωση. Εάν προκύπτουν πολλαπλά προβλήματα, πρέπει να δοθούν προτεραιότητες ή να ταξινομηθούν σε κατηγορίες.

Σύμφωνα με τον E. P. Corey, ο σαφής καθορισμός του προβλήματος θα συμβάλλει στα εξής:

- Να προσδιοριστούν τα άμεσα ζητήματα και να καθοριστούν με τέτοιο τρόπο, ώστε να αναζητηθούν στη συνέχεια απαντήσεις προς συγκεκριμένη κατεύθυνση.
- Να συσχετιστούν τα συγκεκριμένα θέματα με ευρύτερα οργανωτικά και ιστορικά ζητήματα.
- Να εξεταστούν τα προβλήματα από την προοπτική ενός ατόμου και να αναγνωριστούν οι ευθύνες και η δράση αυτού.

Η καλή προετοιμασία είναι βασική για τη συμμετοχή των εκπαιδευομένων στα επόμενα βήματα της ανάλυσης περιπτώσεων. Ο εκπαιδευτικός πρέπει να φροντίσει για τον καθορισμό του προβλήματος σε ατομικό επίπεδο από την πλευρά των εκπαιδευομένων, για να εξασφαλιστεί έτσι ότι όλοι έχουν μελετήσει προσεκτικά την συγκεκριμένη μελέτη. Κατόπιν, όταν κάθε σπουδαστής έχει καθορίσει με σαφήνεια το πρόβλημα, μπορούν να πραγματοποιηθούν αναθέσεις σε ομάδες.

Μια επιλογή, ιδιαίτερα όταν μία περίπτωση περιλαμβάνει διάφορα αντικρουόμενα θέματα, είναι να ομαδοποιηθεί η τάξη με βάση τον προσδιορισμό των προβλημάτων που έχει πραγματοποιήσει ο καθένας. Για παράδειγμα, κάποιοι εκπαιδευόμενοι ίσως είχαν επιλέξει τις πολιτικές πρόσληψης του διευθυντή, ενώ άλλοι επέλεξαν την πολιτική του ανθρώπινου δυναμικού ή την πρόσφατη αναδιοργάνωση της επιχείρησης ως άμεσο ζήτημα στην μελετηθείσα περίπτωση. Κάθε ομάδα θα μπορούσε να ακολουθήσει λοιπόν την αντίστοιχη επιλογή. Αργότερα, οι ομάδες μπορούν να συγκρίνουν και να αντιπαραβάλουν τις προτάσεις τους σε επίπεδο τάξης.

Όταν οι εκπαιδευόμενοι δεν έχουν εμπειρία σε on-line ομαδικά προγράμματα, ο εκπαιδευτικός, σε αυτό το σημείο, μπορεί να πραγματοποιήσει μία εισαγωγή στα βασικά στοιχεία των εικονικών ομάδων. Οι εκπαιδευτικοί μπορούν να ορίσουν τα συγκεκριμένα προγράμματα και τους ρόλους των ομάδων, ή ενδέχεται να ζητηθεί από τους εκπαιδευόμενους να πραγματοποιήσουν μια συμφωνία όσον αφορά στους κανόνες της ομάδας, για την ομαλή και αποδοτική επικοινωνία και συνεργασία τους.

Εάν ο εκπαιδευτικός προτιμά να συνεχίσει με ολόκληρη την τάξη να επεξεργάζεται τον ίδιο προβληματισμό, όλοι οι ορισμοί θα μπορούσαν να παρουσιαστούν στο στάδιο αυτό. Κατόπιν η κάθε ομάδα θα μπορούσε να θέσει τις δικές της προτεραιότητες και να ορίσει το πρόβλημα από κοινού. Ως εναλλακτική λύση, κάθε σπουδαστής μπορεί να παρουσιάσει τα διακριθέντα προβλήματα, ώστε να ακολουθήσει διαδικασία ψηφοφορίας σε επίπεδο τάξης, και το πρόβλημα που θα επιλεγεί με αυτό τον τρόπο θα αποτελέσει το αντικείμενο ανάλυσης από την τάξη.

Οποιαδήποτε προσέγγιση χρησιμοποιείται, ολοκληρώνοντας αυτό το βήμα, ένα ή περισσότερα προβλήματα καθορίζονται σαφώς και διαμορφώνονται οι ομάδες.

ii) Σχεδιασμός της έρευνας.

Το επόμενο βήμα είναι να αρθρωθούν μία έως τρεις ερωτήσεις κλειδιά και να οριστεί ο τρόπος διεξαγωγής της έρευνας. Οι εκπαιδευόμενοι επαναπροσδιορίζουν προσεκτικά τα στοιχεία που παρουσιάζονται στην συγκεκριμένη περίπτωση που μελετούν, καθώς και τα όποια δεδομένα υπάρχουν, για να εντοπίσουν τυχόν κενά. Τα είδη των ερωτήσεων που καλούνται να θέσουν σε αυτό το σημείο περιλαμβάνουν τα εξής:

- Κατά πόσον ενδείκνυται να ληφθούν υπόψη ιστορικοί παράγοντες όσον αφορά στην επιχείρηση, τη βιομηχανία ή τον σχετικό τομέα, για να διαμορφωθεί ένα πλαίσιο για το τρέχον πρόβλημα; Πρέπει οικονομικές, κοινωνικές ή άλλες τάσεις να αποτελέσουν αντικείμενο συζήτησης;
- Ποιες σύγχρονες τάσεις, είτε εσωτερικές είτε εξωτερικές, θα μπορούσαν να οδηγήσουν ή να αποτελέσουν εμπόδιο στην επίλυση του προβλήματος; Τέτοιες τάσεις θα μπορούσαν να είναι δημογραφικές ή σχετικές με το εργατικό δυναμικό, ανθρωπίνους ή οικονομικούς πόρους, ρυθμιστικές αλλαγές ή τεχνολογικές καινοτομίες.

- Πρέπει να ληφθούν υπόψη ζητήματα που αφορούν στην ηγεσία, τη φιλοσοφία της επιχείρησης ή τη διοίκηση και τη λήψη αποφάσεων;

Με την απαίτηση από την πλευρά των εκπαιδευομένων να ακολουθήσουν την on-line έρευνα για αυτό το βήμα, οι εκπαιδευτικοί δίνουν πρόσθετες δυνατότητες μάθησης. Ανάλογα με το επίπεδο της τάξης και την εμπειρία των σπουδαστών, ο εκπαιδευτικός παρέχει μαθήματα ή συμπληρωματικό υλικό για το πώς πρέπει να πραγματοποιήσουν την on-line έρευνα και στη συνέχεια την αξιολόγηση των πόρων που εντοπίζουν. Στην περίπτωση όπου οι σπουδαστές διεξάγουν την πρώτη τους έρευνα, γίνονται στο σημείο αυτό προτάσεις για τον τρόπο διεξαγωγής και για πραγματοποίηση συνεντεύξεων.

Όταν η τάξη μελετά στο σύνολό της το ίδιο πρόβλημα, ο εκπαιδευτικός μπορεί να ορίσει και να αναθέσει τα διαφορετικά μέρη της έρευνας σε κάθε ομάδα. Ανεξάρτητα από την στρατηγική που ακολουθείται, στο τέλος του συγκεκριμένου βήματος, διάφορες ερωτήσεις που στηρίζονται στο πρόβλημα το οποίο έχει καθοριστεί στο βήμα 1, πρέπει να δηλωθούν, μαζί με ένα σχέδιο που θα αποβλέπει στην εύρεση απάντησής τους.

iii) Διεξαγωγή της έρευνας

Σε αυτή τη φάση, πραγματοποιείται η έρευνα που έχει προγραμματιστεί στο βήμα 2. Εάν οι ομάδες εργάζονται εκτός του φόρουμ της τάξης, σε αυτό το στάδιο μπορεί να τους ζητηθεί να εκθέσουν μια σύνοψη της έρευνάς τους μέχρι αυτή τη φάση, σε συνδυασμό με την ανταλλαγή των πηγών που έχει χρησιμοποιήσει η κάθε ομάδα.

Ένα χαρακτηριστικό της on-line έρευνας είναι ο εντοπισμός των πόρων αλλά και η αξιολόγησή τους. Υπάρχουν πολλά πολύτιμα εργαλεία που μπορούν να βοηθήσουν τους εκπαιδευόμενους και στις δύο αυτές δραστηριότητες και στο στάδιο όπου βρισκόμαστε κάτι τέτοιο είναι μία καλή πρόταση. Τα συγκεκριμένα εργαλεία είναι σημαντικά επειδή η ποσότητα της on-line πληροφορίας μπορεί να είναι συντριπτική και η ποιότητα ποικίλλει επίσης.

Οι εκπαιδευτικοί μπορούν να συμβάλουν προτείνοντας πρόσθετες δραστηριότητες για να επεκτείνουν την έρευνα των σπουδαστών. Παραδείγματος χάρη, στο φόρουμ συζήτησης διάφορες αντιπαραβαλλόμενες πηγές πληροφοριών μπορούν να προταθούν από τις διάφορες ομάδες, οι οποίες ίσως περιλαμβάνουν on-line άρθρα δημοσιεύσεων, κυβερνητικά έγγραφα ή εκθέσεις που έχουν εντοπιστεί στον ιστοχώρο

της επιχείρησης που μελετάται, ή ακόμη και βίντεο από λόγους και συνεντεύξεις. Η προτεινόμενη δραστηριότητα μπορεί να απαιτεί από ένα άτομο ή τις ομάδες των σπουδαστών να επιλέξουν έναν ή περισσότερους πόρους που αναλύουν, και έπειτα να συνοψίσουν τους τρόπους με τους οποίους αυτοί οι πόροι σχετίζονται με το συγκεκριμένο πρόβλημα που μελετούν.

Οι αναθέσεις μπορεί να περιλαμβάνουν την διεξαγωγή ερευνών ή συνεντεύξεων από πεπειραμένες πηγές. Ένας φιλοξενούμενος μπορεί να περάσει από συνέντευξη σε μια εικονική τάξη, ή να κληθεί να συμμετέχει σε μια ασύγχρονη σύνοδο ερωταποκρίσεων σε ένα φόρουμ συζήτησης. Αυτές οι δυνατότητες πραγματοποίησης on-line έρευνας θα είναι ιδιαίτερα ευεργετικές για τους εκπαιδευόμενους. Στον τέλος του βήματος αυτού, οι σπουδαστές, οι ομάδες ή/και η τάξη θα έχουν τις πληροφορίες που πρέπει ώστε να συστήσουν συγκεκριμένες λύσεις για το πρόβλημα.

iv) Προσδιορισμός και αξιολόγηση των εναλλακτικών λύσεων.

Σε αυτό το βήμα οι σπουδαστές διατυπώνουν και αξιολογούν τις εναλλακτικές λύσεις για το καθορισμένο πρόβλημα. Με βάση τα στοιχεία που διαθέτουν και τα αποτελέσματα από την έρευνα έχει διεξαχθεί, οι σπουδαστές ή οι ομάδες προτείνουν δύο ή περισσότερες εναλλακτικές λύσεις σε πρακτικό επίπεδο. Όταν προκύψουν αρκετές εναλλακτικές λύσεις, οι σπουδαστές ή οι ομάδες μπορούν να παρέχουν τη λογική με την οποία προτείνουν τις λύσεις τους και να χρησιμοποιήσουν την on-line ψηφοφορία για να περιορίσουν τον αριθμό σε δύο ή τρεις λύσεις για αξιολόγηση.

Στη συνέχεια, οι σπουδαστές θα αξιολογήσουν αυστηρά, θα συγκρίνουν και θα αντιπαραβάλουν αυτές τις εναλλακτικές συστάσεις. Οι εξωτερικοί και εσωτερικοί παράγοντες, οι απαιτήσεις για την υλοποίηση, οι επιπτώσεις και άλλοι παράγοντες πρέπει να ληφθούν υπόψη. Στο στάδιο αυτό λοιπόν, οι σπουδαστές ή οι ομάδες μελετούν ο ένας τις προτάσεις του άλλου.

Η διαδικασία αξιολόγησης μπορεί επίσης να λάβει τη μορφή ενός παιχνιδιού, με τα άτομα ή τις ομάδες να έχουν τους ρόλους των βασικών παικτών που περιγράφουν την εκάστοτε πρόταση από τη δική τους οπτική γωνία. Μια άλλη επιλογή είναι η συζήτηση μεταξύ εκείνων που παρουσιάζουν τις προτάσεις και εκείνων που ρωτούν τη λογική που τις διακρίνει.

Στο τέλος του βήματος αυτού, διάφορες εναλλακτικές λύσεις έχουν αξιολογηθεί προσεκτικά και είναι πλέον στιγμή να επιλεγεί η καλύτερη.

ν) **Επιλογή της καλύτερης πρότασης.**

Σε αυτό το βήμα, οι σπουδαστές ή οι ομάδες συστήνουν την καλύτερη λύση για το πρόβλημα. Είναι απαραίτητο να υποστηρίξουν και να δικαιολογήσουν την πρότασή τους, αναφέροντας τα αποτελέσματα της έρευνας και άλλες ουσιαστικές πηγές.

Η έκβαση από το συγκεκριμένο βήμα είναι μια καλά ορισμένη πρόταση για το αρχικό πρόβλημα που καθορίστηκε στο βήμα 1.

νι) **Παρουσίαση συμπερασμάτων.**

Οι σπουδαστές ή οι ομάδες ολοκληρώνουν την ανάλυση περιπτώσεων με μία παρουσίαση των συμπερασμάτων, συμπεριλαμβάνοντας την έρευνα, την ανάλυση και τις προτάσεις.

Ανάλογα με την τάξη και το αντικείμενο της περίπτωσης, οι σπουδαστές θα κληθούν να περιγράψουν τις αλλαγές στις μορφές διαχείρισης ή ηγεσίας, το επιχειρησιακό πρότυπο, την οργανωτική δομή, τα προϊόντα ή τις υπηρεσίες, τις πολιτικές ή τα συστήματα για να εφαρμόσουν ανάλογα με την περίπτωση τις προτάσεις τους. Τα συμπεράσματα για το case μπορούν να παρουσιαστούν υπό μορφή εγγράφου, καθώς και ως ηλεκτρονικό χαρτοφυλάκιο με τα αρχεία για κάθε βήμα, ή και ως παρουσίαση PowerPoint.

Τέλος, οι εκπαιδευόμενοι μπορεί επίσης να κληθούν να σχολιάσουν τις διαδικασίες της εικονικής συνεργασίας, της έρευνας και της αλληλεπίδρασης, υποβάλλοντας σχετική έκθεση, με μια αξιολόγηση της αποτελεσματικότητας της συμφωνίας που πραγματοποιήθηκε μεταξύ των ομάδων στο βήμα 1.

2.1.3. Πλατφόρμες συνεργασίας

Η ανάλυση περιπτώσεων περιλαμβάνει σε μεγάλο βαθμό τη συνεργασία και μερικές εφαρμογές διδακτικών υλικών είναι πιο αποτελεσματικές από άλλες σε επίπεδο ομάδας. Μαθήματα πανεπιστημιακού επιπέδου παραδίδονται σε on-line σύνδεση χρησιμοποιώντας πλατφόρμες όπως BlackBoard, WebCT ή LearningSpace. Το διδακτικό υλικό έχει τις επιλογές παράδοσης σε πραγματικό χρόνο, σε επίπεδο σύγχρονης επικοινωνίας στις εικονικές τάξεις καθώς επίσης και οποτεδήποτε, σε επίπεδο ασύγχρονης επικοινωνίας στα φόρουμ συζήτησης. Οι εκπαιδευτικοί μπορούν να οργανώσουν χωριστά φόρουμ για τις διάφορες ομάδες, ενώ αν

διδάσκουν σε συγκεκριμένα περιβάλλοντα μπορούν να εκμεταλλευθούν τα free ή trial εργαλεία για τη σύγχρονη επικοινωνία, τις έρευνες, τις ψηφοφορίες ή άλλα διαλογικά χαρακτηριστικά γνωρίσματα που συμπληρώνουν το διδακτικό υλικό και ενισχύουν την on-line επικοινωνία των εκπαιδευομένων και τη συνεργατική μάθηση.

2.1.4. On-line ανάλυση περιπτώσεων

Η ανάλυση περιπτώσεων μπορεί να χρησιμοποιηθεί και να συμβάλλει στην καλλιέργεια και την αναπύξη επαγγελματικών, ακαδημαϊκών και τεχνολογικών δεξιοτήτων. Ο εκπαιδευτικός μπορεί να επεξεργαστεί τις εκπαιδευτικές δραστηριότητες που αφορούν στους στόχους των μαθημάτων του και είναι προσαρμοσμένες με τις ανάγκες και τα ενδιαφέροντα των σπουδαστών. Στην πορεία, οι εκπαιδευτικοί ίσως διακρίνουν τους on-line σπουδαστές τους να κατέχονται από μεγαλύτερο ενδιαφέρον για να συμμετέχουν στις τάξεις. Με τη δυνατότητα για αλληλεπίδραση μεταξύ εκπαιδευτικού-σπουδαστή, σπουδαστών μεταξύ τους και σπουδαστή-φιλοξενούμενου, ενισχύεται η ευκαιρία για απόκτηση ουσιαστικών μαθησιακών εμπειριών.

2.2. Περιγραφή του SW Tutorial

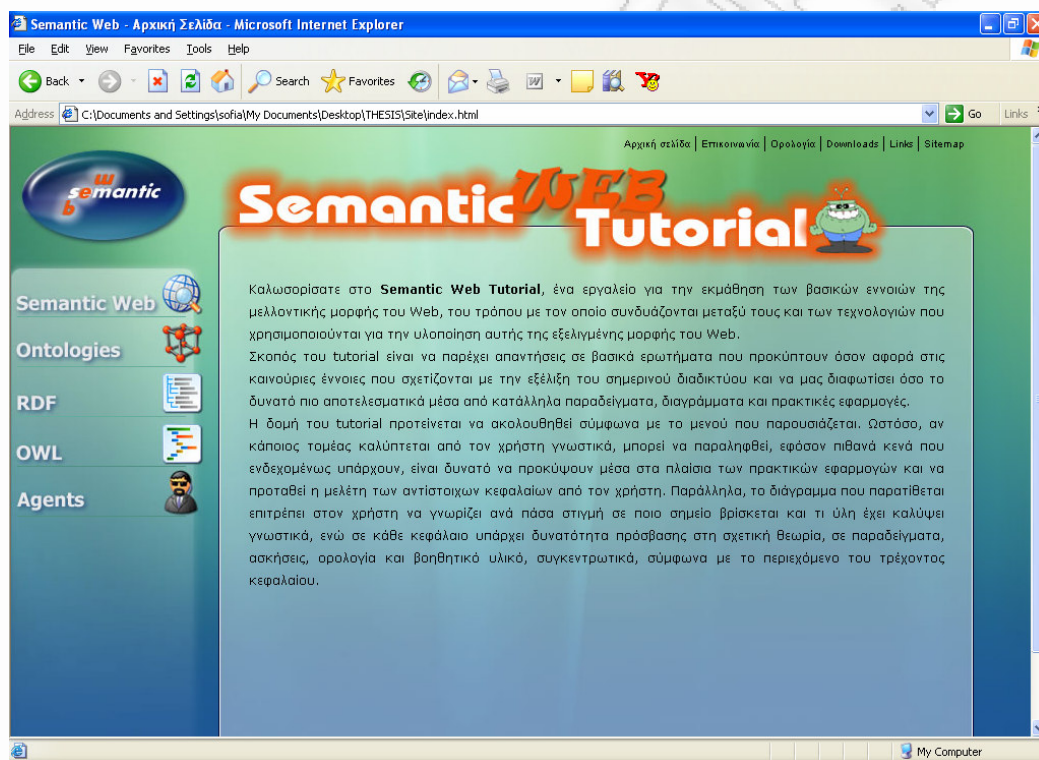
Το Semantic Web Tutorial είναι ένα εργαλείο για την εκμάθηση των βασικών εννοιών του σημασιολογικού ιστού και την κατανόηση της λειτουργίας του μέσα από την θεωρητική ανάλυση αλλά και την πρακτική εφαρμογή. Ακολουθεί τη φιλοσοφία που περιγράφηκε παραπάνω, καθώς παρουσιάζονται συγκεκριμένες περιπτώσεις για να περιγραφεί κάθε φορά το προς μελέτη αντικείμενο, και πιο χαρακτηριστικά, περιγράφεται πλήρως μία ολοκληρωμένη πρακτική εφαρμογή, η οποία μπορεί να αποτελέσει πρότυπο τόσο για την αντιμετώπιση ανάλογων περιπτώσεων, όσο και για την προσπέραση τυχόν εμποδίων που μπορεί να παρουσιαστούν στα πλαίσια μίας υλοποίησης. Στα εισαγωγικά και θεωρητικά θέματα αρχικά θα ακολουθηθεί ένα περισσότερο behavior-ιστικό μοντέλο. Μετά από κάθε κομμάτι θεωρίας θα ακολουθούν ερωτήσεις που θα τονίζουν τα βασικότερα σημεία της.

Στα πρακτικά θέματα θα ακολουθηθεί η μέθοδος drill & practice, η οποία θα συνοδεύεται από το κατάλληλο υλικό (επίδειξη χρήσης λογισμικών) καθώς και σενάρια πειραματισμού με τα διάφορα λογισμικά (costructiv-ιστικό μοντέλο).

Στη συνέχεια ακολουθεί μία μικρή περιγραφή της δομής και λειτουργίας του tutorial, με αντιπροσωπευτικές εικόνες από τα βασικά σημεία του.

2.2.1. Αρχική σελίδα

Στην αρχική σελίδα του tutorial γίνεται μια εισαγωγή στο θέμα το οποίο αποτελεί αντικείμενο του συγκεκριμένου εργαλείου, με μια σύντομη περιγραφή των δυνατοτήτων που παρέχονται στον χρήστη. Η πρώτη αυτή σελίδα έχει την ακόλουθη μορφή:



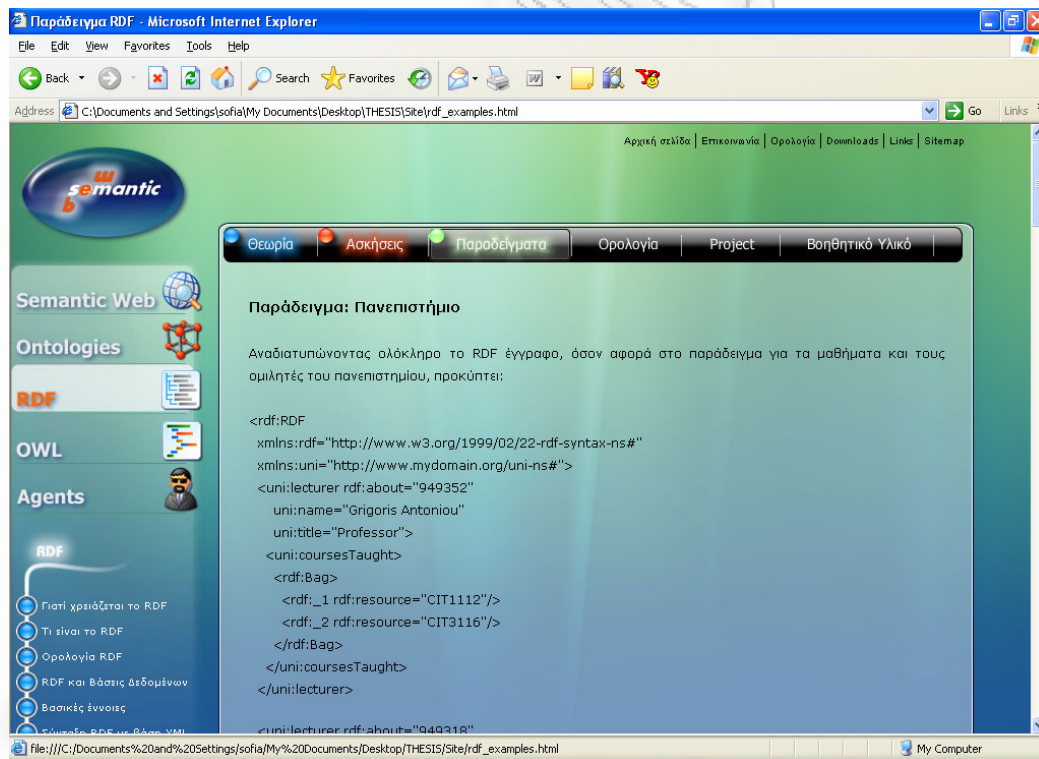
Η πρώτη σελίδα του κάθε κεφαλαίου είναι εισαγωγική, καθώς αναφέρεται στο περιεχόμενό του. Εκτενής ανάλυση ακολουθεί στα υποκεφάλαια, τα οποία και είναι εμφανή στο αριστερό τμήμα της κάθε σελίδας, ώστε να έχει τη δυνατότητα ο χρήστης να γνωρίζει ανά πάσα στιγμή τι ύλη έχει καλύψει και να μπορεί να μεταβεί χωρίς δυσκολία στο υποκεφάλαιο που τον ενδιαφέρει.

Στο σημείο αυτό πρέπει να τονίσουμε ότι σε κάθε κεφάλαιο, στο πάνω τμήμα της σελίδας υπάρχει μία μπάρα με το εξής μενού: «Θεωρία-Ασκήσεις-Παραδείγματα-Ορολογία-Project-Βοηθητικό Υλικό». Πρόκειται για μία διευκόλυνση του χρήστη, ο

οποίος έχει έτσι τη δυνατότητα να εντοπίζει συγκεντρωτικά τα βασικά σημεία της κάθε ενότητας, ανάλογα με το περιεχόμενό τους, αν δηλαδή πρόκειται για θεωρητικό κομμάτι, άσκηση ή παράδειγμα, αλλά ακόμη και την όποια ορολογία σχετίζεται με την συγκεκριμένη ενότητα και το υλικό που προαπαιτείται να γνωρίζει για την κατανόηση του κεφαλαίου.

Αν για παράδειγμα λοιπόν, πατήσουμε στο tab «Θεωρία» του πρώτου κεφαλαίου, θα οδηγηθούμε σε μια μικρή αναφορά στα βασικά σημεία της θεωρίας του Semantic Web.

Κατά αντίστοιχο τρόπο, αν μεταβούμε στο κεφάλαιο που αναφέρεται στο RDF και επιλέξουμε το tab «Παραδείγματα», έχουμε τη δυνατότητα να μελετήσουμε συγκεντρωτικά τα χαρακτηριστικά παραδείγματα που συναντήσαμε στην συγκεκριμένη ενότητα, και τα οποία θα συμβάλλουν στην βαθύτερη κατανόηση της θεωρίας. Η σελίδα που προκύπτει σε αυτή την περίπτωση είναι η εξής:



Επιλέγοντας στη συνέχεια το tab «Ορολογία» έχουμε πρόσβαση στις βασικές έννοιες του κεφαλαίου RDF, που συνοδεύονται από έναν περιγραφικό ορισμό. Η αντίστοιχη σελίδα είναι η εξής:



Η ίδια λογική ως προς τη δομή ακολουθείται σε όλες τις ενότητες του tutorial.

Αξιοσημείωτο είναι το γεγονός ότι στη δομή της κάθε ενότητας σε μορφή υποκεφαλαίων που παρουσιάζεται στο αριστερό τμήμα της κάθε σελίδας, τα υποκεφάλαια χαρακτηρίζονται από ένα συγκεκριμένο χρώμα, το οποίο και είναι ενδεικτικό όσον αφορά στο περιεχόμενό τους, αν δηλαδή πρόκειται για θεωρία (οπότε συναντούμε το γαλάζιο χρώμα), άσκηση (οπότε συναντούμε το κόκκινο χρώμα) ή παράδειγμα (οπότε συναντούμε το πράσινο χρώμα). Άλλωστε, τα χρώματα αυτά χαρακτηρίζουν και τη μπάρα που ήδη αναφέραμε στο πάνω μέρος, επισημαίνοντας τη θεωρία, τις ασκήσεις και τα παραδείγματα. Επιπλέον, στη συγκεκριμένη μπάρα υπάρχει η επιλογή «Βοηθητικό υλικό» όπου παρατίθεται για παράδειγμα μία παρουσίαση της XML, η οποία είναι απαραίτητη για την κατανόηση της RDF που ακολουθεί τη λογική της.

Τέλος, στο πάνω δεξί μέρος της εφαρμογής υπάρχει ένα μενού που μας επιτρέπει να μεταβούμε στην αρχική σελίδα, στη σελίδα που εμφανίζονται τα links, τα downloads καθώς και στη σελίδα μέσω της οποίας υπάρχει δυνατότητα επικοινωνίας ενδεχομένως με έναν πιθανό επιβλέποντα, και τέλος στο Sitemap του tutorial. Σημειώνουμε εδώ πως όσον αφορά στην ορολογία που σχετίζεται γενικά με το

Semantic Web (καθώς αυτό είναι που μας ενδιαφέρει στο συγκεκριμένο σημείο στο οποίο αναφερόμαστε), πρόκειται για μια πιο γενική αντιμετώπιση συγκριτικά με την ορολογία που περιλαμβάνεται σε κάθε κεφάλαιο χωριστά και η οποία είναι πιο ειδικευμένη, εφόσον προσαρμόζεται στα δεδομένα της αντίστοιχης ενότητας.

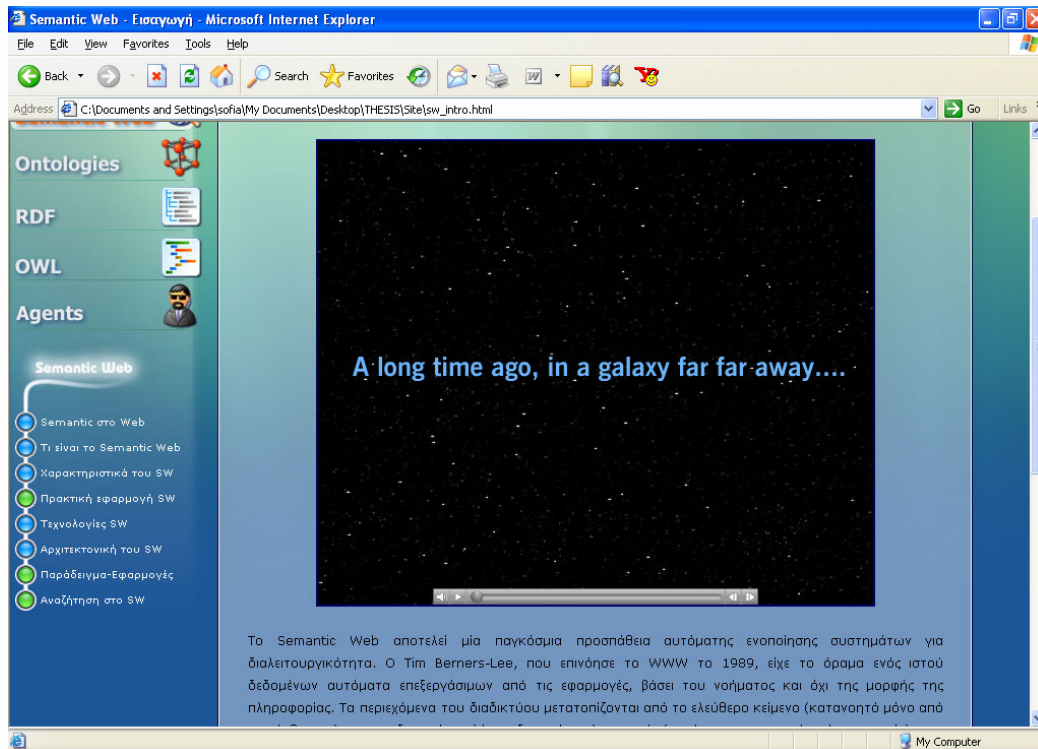
2.2.2. Semantic Web

Ο χρήστης ξεκινάει ακολουθώντας τη δομή του tutorial, από το πρώτο δηλαδή κεφάλαιο, το επονομαζόμενο “Semantic Web”, στο οποίο και περιγράφεται η ανάγκη μέσα από την οποία προέκυψε το όραμα του σημασιολογικού ιστού καθώς και οι βασικές έννοιες που τον αποτελούν. Στόχος του συγκεκριμένου κεφαλαίου είναι:

- η κατανόηση της σημασίας του σημασιολογικού ιστού: ο παγκόσμιος ιστός, στη μορφή που είναι σήμερα, περιέχει πλήθος πληροφοριών οι οποίες παρουσιάζονται σε πολυάριθμους δικτυακούς τόπους, σε διάφορες γλώσσες, με διαφορετικές ορολογίες. Για να μπορέσουμε να χειριστούμε την πληροφορία αυτή με μηχανικό τρόπο, πρέπει να την ομογενοποιήσουμε σε δομή και συντακτικό, ώστε να υποστηριχθεί η αποδοτική αναζήτηση, επεξεργασία και ενοποίηση δεδομένων.

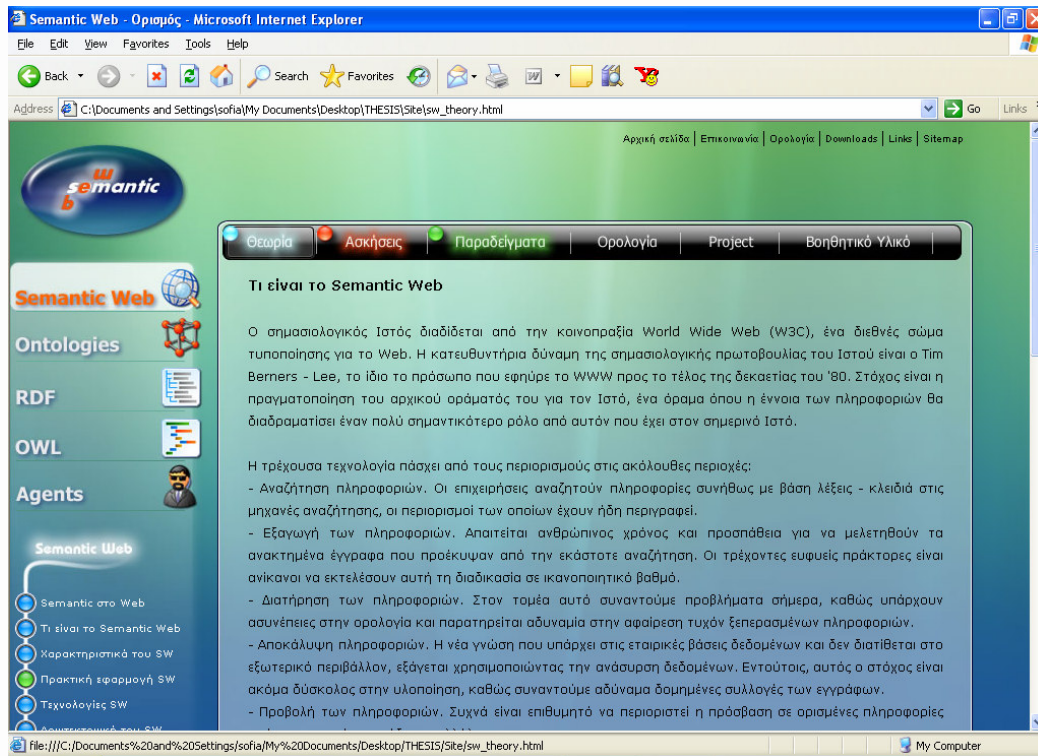
- η εισαγωγή στις θεμελιώδεις έννοιες που αποτελούν το Semantic Web, τα βασικά χαρακτηριστικά του και τις τεχνολογίες που πρέπει κανείς να γνωρίσει ώστε να κατανοήσει πώς μπορεί να χειριστεί και να υλοποιήσει αυτή τη μορφή του Web.

Η εισαγωγική σελίδα του πρώτου αυτού κεφαλαίου λοιπόν έχει την ακόλουθη μορφή:



Εδώ πρέπει να τονίσουμε πως πριν από τη θεωρητική εισαγωγή στο αντικείμενο του κεφαλαίου, παρουσιάζεται σε μορφή flash, ένα σενάριο που εισάγει το χρήστη στην έννοια του σημασιολογικού ιστού. Στόχος μας είναι η πρόκληση του ενδιαφέροντος του εκπαιδευόμενου, καθώς ένα τέτοιο tutorial ενδείκνυται για χρήση με εκπαιδευτικό σκοπό.

Προς διευκόλυνση του χρήστη, τα βασικά σημεία κάθε υποενότητας είναι συγκεντρωμένα και προσβάσιμα μέσω του tab «Θεωρία», όπως ακριβώς συμβαίνει και με τα όποια παραδείγματα αντιστοιχούν στην κάθε περίπτωση, τα οποία και πάλι είναι θεωρητικής φύσεως (λόγω του αντικειμένου της συγκεκριμένης ενότητας στην οποία αναφερόμαστε). Η αντιστοιχη σελίδα έχει ως εξής:



Τις υποενότητες του συγκεκριμένου κεφαλαίου ακολουθεί μία σειρά από ερωτήσεις, μέσα από τις οποίες τονίζονται τα βασικότερα σημεία όσων έχουν περιγραφεί, έτσι ώστε ο χρήστης να έχει τη δυνατότητα να ανατρέξει στις έννοιες που ενδεχομένως έχει προσπεράσει ή δεν έχει εμπεδώσει. Ακολουθούμε λοιπόν εδώ, όπως και στη συνέχεια όταν πρόκειται για τα αντίστοιχα τμήματα της κάθε ενότητας που ακολουθεί, το συμπεριφοριστικό μοντέλο παιδαγωγίσης, καθώς εστιάζουμε στην παρατηρήσιμη συμπεριφορά του ατόμου.

2.2.3. Ontologies

Στη δεύτερη αυτή ενότητα του tutorial αναφερόμαστε στις έννοιες της ταξονομίας και της οντολογίας. Στόχος είναι η κατανόηση της σημασίας των δύο εννοιών και για αυτό χρησιμοποιούμε συγκεκριμένα παραδείγματα. Μέσα από αυτά είναι προφανής η ανάγκη για ιεράρχηση που συναντούμε σε όλο το φάσμα των δραστηριοτήτων μας σήμερα και κατά συνέπεια η ύπαρξη της έννοιας της ταξονομίας από τη μία, ενώ από την άλλη ορίζεται η οντολογία, η οποία κυριαρχεί στον σημασιολογικό ιστό, καθώς με βάση αυτή πραγματοποιείται η δόμηση όλης της διακινούμενης πληροφορίας, ώστε αυτή να είναι επεξεργάσιμη και κατανοητή από όλους.

Στο σημείο αυτό πρέπει να τονίσουμε πως δεν παρατίθενται απλά οι ορισμοί των δύο εννοιών, αλλά τα παραδείγματα που παρουσιάζονται είναι τέτοιας μορφής που επιτρέπουν την προσπάθεια γενίκευσης εκ μέρους των χρηστών, των αντιστοιχιών και της συνεχούς σύνδεσης των επιμέρους θεμάτων, ώστε να προκύψει με αυτό τον τρόπο άμεσα η σημασία των εννοιών για την κατασκευή της επόμενης μορφής του σημερινού παγκόσμιου ιστού. Ενδεικτικά παραθέτουμε τη σελίδα που αφορά στο παράδειγμα με το λεξικό, μέσα από το οποίο γίνεται προφανής η έννοια της ταξινόμησης που ήδη όλοι μας χρησιμοποιούμε, χωρίς να το έχουμε πρακτικά επιδώσει, σε πολλές εκφάνσεις της καθημερινής μας ζωής:

The screenshot shows a Microsoft Internet Explorer browser window displaying a web page. The address bar shows the path: C:\Documents and Settings\sofia\My Documents\Desktop\THEISIS\Site\taxonomy_thesaurus.html. The page has a navigation menu at the top with tabs: Θεωρία, Ασκήσεις, Παράδειγματα (selected), Ορολογία, Project, and Βοηθητικό Υλικό. On the left, there is a sidebar with 'Semantic Web' and 'Ontologies' sections. The main content area is titled 'Παράδειγμα: Thesaurus' and contains the following text:

Λεξικό
 Πρόκειται για ένα ελεγχόμενο λεξιλόγιο, το οποίο είναι οργανμένο με μία ορισμένη τάξη και δομή έτσι ώστε να υφίστανται συγκεκριμένες συσχετίσεις, ιεραρχίας και ισοδυναμίας, μεταξύ των διαφόρων όρων που παρατίθενται σαφώς και προσδιορίζονται από συγκεκριμένους, τυποποιημένους δείκτες. Ένα λεξικό έχει ως στόχο να διευκολυνθεί η ανάκτηση των εγγράφων και να επιτευχθεί η συνέπεια στην ευετηρίαση των καταγεγραμμένων εγγράφων.

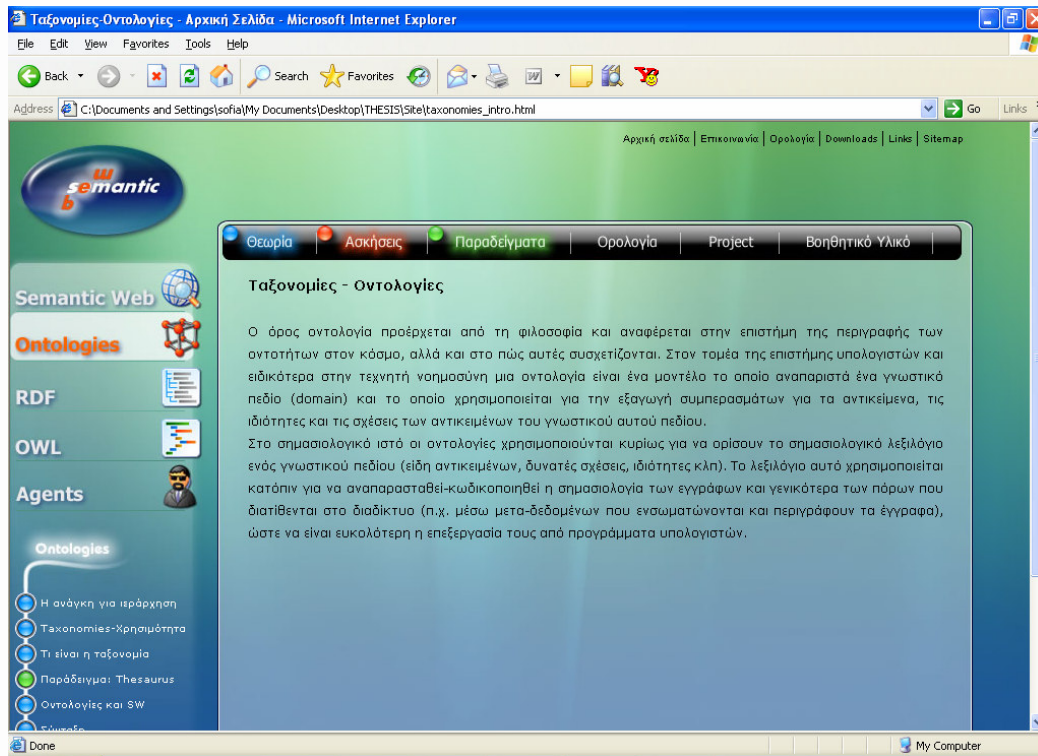
Οι συσχετίσεις μεταξύ των όρων μπορούν να ταξινομηθούν σε τέσσερις κατηγορίες:

- Equivalence
- homographic
- Hierarchical
- Associative

Στον ακόλουθο πίνακα παρατίθενται αυτές οι συσχετίσεις και παρουσιάζονται και σχετικοί ορισμοί και παραδείγματα.

SEMANTIC RELATION	DEFINITION	EXAMPLE
Synonymy Similar to Equivalent Used for	A term X has nearly the same meaning as a term Y.	"Report" is a synonym for "document"
Homonymy Spelled the same Homographic	A term X is spelled the same way as a term Y, which has a different	The "tank," which is a military vehicle, is a homonym for the "tank," which is a receptacle for

Η αρχική σελίδα του κεφαλαίου είναι της μορφής που ακολουθεί:



Και πάλι παρουσιάζονται τα σημεία κλειδιά της ενότητας συγκεντρωμένα και διαχωρισμένα σύμφωνα με το περιεχόμενό τους. Αναλύονται τα βασικά χαρακτηριστικά που αποτελούν μια οντολογία (κλάσεις, υποκλάσεις, instances) αλλά γίνεται και μια περιγραφή της μεθοδολογίας που πρέπει να ακολουθησει κανείς για την κατασκευή μίας ολοκληρωμένης οντολογίας. Στο βοηθητικό υλικό της συγκεκριμένης ενότητας, ο χρήστης μπορεί να μεταβεί μέσω υπερσυνδέσμων σε σελίδες όπου παρουσιάζονται έτοιμες οντολογίες, ώστε να γίνει αντιληπτή η φιλοσοφία και η λογική δόμησής τους.

Ωστόσο, η πρακτική εξάσκηση του εκπαιδευομένου πάνω στο συγκεκριμένο αντικείμενο περιλαμβάνεται σε επόμενο κεφάλαιο, καθώς έως τώρα δεν έχει γίνει λόγος, παρά μόνο μία μικρή αναφορά σε μία από τις υποενότητες του συγκεκριμένου κεφαλαίου, για τις γλώσσες που χρησιμοποιούνται για τη δημιουργία των οντολογιών και οι οποίες είναι φυσικά απαραίτητες, μαζί με τους κανόνες που ακολουθούν, για την κατασκευή μίας ορθά δομημένης οντολογίας.

Επομένως, και σε αυτό το κεφάλαιο παρατίθεται μία σειρά από ερωτήσεις προκειμένου να προσδιοριστεί ο βαθμός στον οποίο έχουν γίνει αντιληπτά τα βασικά σημεία που αφορούν στις οντολογίες, ώστε να είναι εύκολη η μετάβαση του εκπαιδευομένου στις επόμενες ενότητες, όπου παρουσιάζονται αναλυτικά οι γλώσσες

για τις οντολογίες, με τελικό στόχο την πρακτική εφαρμογή, εφόσον προηγουμένως έχουν υιοθετηθεί όλες οι απαραίτητες έννοιες για την ικανοποιητική απόδοση του εκπαιδευόμενου, στο στάδιο χρήσης του κατάλληλου εργαλείου προς την ολοκλήρωση του tutorial.

2.2.4. Resource Description Framework

Στην επόμενη ενότητα επικεντρωνόμαστε στη γλώσσα RDF. Αρχικά επισημαίνεται η διαφορά μεταξύ XML και RDF, καθώς η πρώτη δεν είναι σε θέση να καλύψει τις ανάγκες του σημασιολογικού ιστού, όπου η σημασιολογία των δεδομένων είναι το κύριο χαρακτηριστικό που μας ενδιαφέρει.

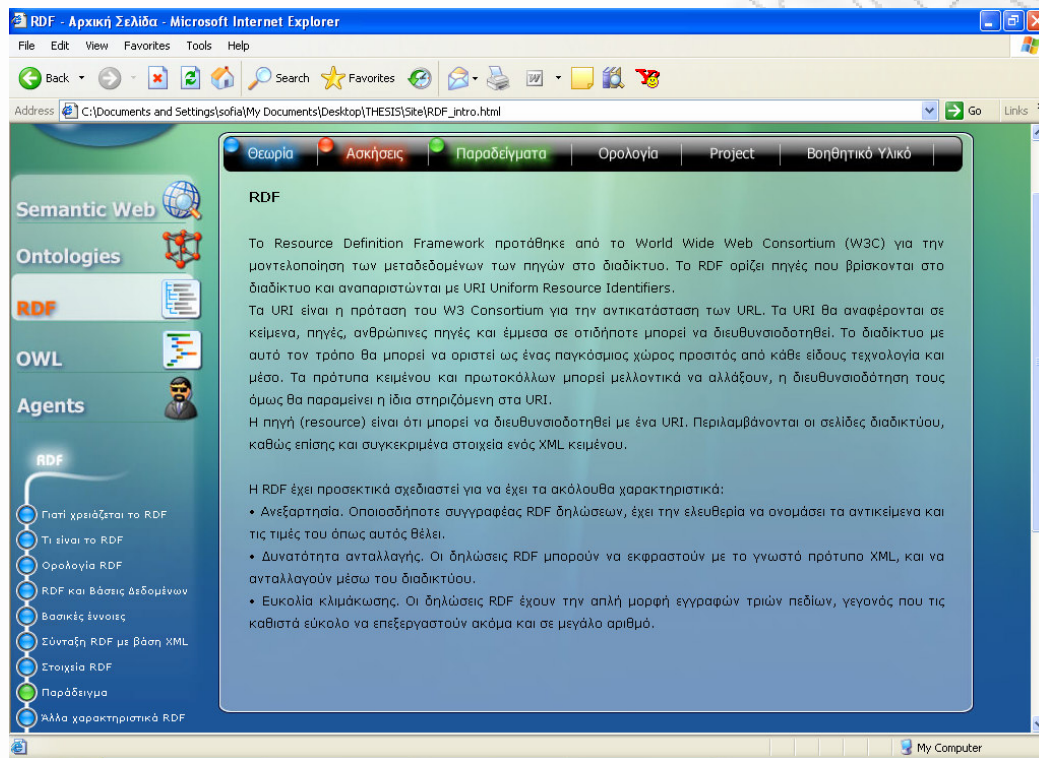
Συγκεκριμένα παραδείγματα σε μορφή XML αναδεικνύουν τις ελλείψεις της συγκεκριμένης γλώσσας και μέσα από αυτές προκύπτουν οι απαιτήσεις για μία νέα, πιο εξελιγμένη γλώσσα από την XML, η οποία έχει ήδη οριστεί ως Resource Description Framework (RDF).

Στόχος της συγκεκριμένης ενότητας είναι η εισαγωγή στη γλώσσα RDF, η παρουσίαση των βασικών στοιχείων και της ορολογίας της και η κατανόηση αυτών μέσα από λεπτομερή παραδείγματα, είτε σε μορφή κώδικα, ακολουθώντας την XML σύνταξη, είτε σε μορφή γραφικών παραστάσεων και διαγραμμάτων, μέσα από τα οποία γίνονται σαφή βασικά σημεία της σύνταξης και των κανόνων της RDF. Με τη βοήθεια αυτής της ενότητας, ο εκπαιδευόμενος θα γνωρίσει πώς ορίζεται η σημασιολογία που εγκλείεται σε πληροφοριακούς πόρους με τυπικό, διαλειτουργικό και ανθρωπίνως αναγνώσιμο τρόπο μέσω ενός μηχανισμού για την περιγραφή πληροφοριακών πόρων, ο οποίος δεν κάνει καμία υπόθεση για τη φύση του συγκεκριμένου πεδίου εφαρμογής ή τη δομή του εγγράφου που περιέχει την πληροφορία. Εκτός από τη διαλειτουργικότητα μεταξύ συστημάτων, το RDF στοχεύει στην επαναχρησιμοποίηση, το διαμοιρασμό και την επεκτασιμότητα των ορισμών μεταδεδομένων και κατά συνέπεια, στην αυτόματη επεξεργασία πόρων που ανταλλάσσονται μέσω του Διαδικτύου.

Επιπλέον, στην ενότητα αυτή παρουσιάζεται το σχήμα RDF, καθώς κρίνεται απαραίτητο για την ορθή κατασκευή λεξιλογίων RDF. Αφού λοιπόν πραγματοποιηθεί η εισαγωγή στην έννοια του RDFS και η κατάλληλη θεωρητική περιγραφή του, ακολουθεί αναλυτική περιγραφή της ορολογίας που χρησιμοποιεί, με συγκεκριμένα και αντιπροσωπευτικά παραδείγματα.

Και σε αυτή την ενότητα προτείνονται συγκεκριμένες ασκήσεις, μέσα από τις οποίες τονίζονται τα βασικά σημεία που πρέπει κανείς να λάβει υπόψη για να εμποδώσει το περιεχόμενο της γλώσσας και να είναι στη συνέχεια σε θέση να υλοποιήσει και σε πρακτικό επίπεδο δικές του εφαρμογές κάνοντας χρήση της RDF.

Παραθέτουμε ενδεικτικά την αρχική σελίδα της ενότητας όπως και για τις προηγούμενες:



Και πάλι υπάρχει η δυνατότητα επιλογής μεταξύ της θεωρίας, των ασκήσεων, των παραδειγμάτων και της ορολογίας που αφορούν στο σύνολο του κεφαλαίου, μέσω του αντίστοιχου tab στο πάνω μέρος της σελίδας.

2.2.5. Web Ontology Language (OWL)

Την ενότητα που αφορά στη γλώσσα RDF ακολουθεί η αναφορά στη γλώσσα OWL (Web Ontology Language) με την σχετική ανάλυση και περιγραφή που απαιτείται.

Σε πρώτο στάδιο τονίζεται η διαφορά της από την RDF, καθώς η OWL επεκτείνει τις δυνατότητες που παρέχει η RDF και είναι η γλώσσα που χρησιμοποιούμε για τον ορισμό και την κατασκευή των οντολογιών μας στη συνέχεια.

Αφού λοιπόν περιγραφεί με λόγια η υπεροχή της OWL έναντι της RDF, ακολουθεί η ανάλυση των τριών ειδών στα οποία διακρίνεται η συγκεκριμένη γλώσσα και εν συνεχεία περιγράφονται αναλυτικά και με βοηθητικά παραδείγματα τα χαρακτηριστικά της, όσον αφορά στη σύνταξη και τα στοιχεία που χρησιμοποιεί. Όπως σε κάθε ενότητα, εκτός από τα επιμέρους παραδείγματα που παρατίθενται μέσα στα πλαίσια των διαφόρων υποενοτήτων, ανάλογα με το αντικείμενο καθεμιάς από αυτές, τελικά παρουσιάζεται ένα ολοκληρωμένο και όσο το δυνατόν πιο αντιπροσωπευτικό παράδειγμα.

Κατόπιν, σε αυτό το κεφάλαιο, και εφόσον έχουν προηγηθεί όλα τα παραπάνω, οπότε ο εκπαιδευόμενος έχει κατανοήσει όλες τις βασικές έννοιες που του έχουν παρουσιαστεί και περιγραφεί, θα λέγαμε ότι περνάμε στο κονστрукτιβιστικό παιδαγωγικό μοντέλο. Πιο συγκεκριμένα, ακολουθεί η επίδειξη χρήσης συγκεκριμένου λογισμικού προς την κατασκευή μίας οντολογίας, σύμφωνα με τα όσα έχουμε ήδη αναφερθεί, τόσο στο συγκεκριμένο κεφάλαιο, όσο και στα προηγούμενα. Το εργαλείο που παρουσιάζεται είναι το Protégé, το οποίο περιγράφεται αναλυτικότερα στη συνέχεια. Την παρουσίαση αυτή ακολουθεί πρακτική εφαρμογή από την πλευρά του εκπαιδευομένου, ο οποίος καλείται να πειραματιστεί δημιουργώντας τις δικές του οντολογίες.

2.2.6. Protégé

Αυτό που θα ήταν αξιοσημείωτο εδώ, είναι η δυνατότητα που παρέχεται στον χρήστη να παρακολουθήσει αναλυτικά τη δημιουργία μίας οντολογίας με χρήση του εργαλείου Protégé, σε γλώσσα OWL.

Το Protégé είναι μια open-source πλατφόρμα που παρέχει σε μια αυξανόμενη κοινότητα χρηστών μία σειρά εργαλείων για την κατασκευή προτύπων και εφαρμογών με βάση τις οντολογίες. Στον πυρήνα του, το Protégé υλοποιεί ένα πλούσιο σύνολο δομών αναπαράστασης της γνώσης, που υποστηρίζουν τη δημιουργία, την απεικόνιση και το χειρισμό των οντολογιών σύμφωνα με διάφορα σχήματα αντιπροσώπευσης. Μπορεί να προσαρμοστεί έτσι ώστε να παρέχει στο χρήστη τη δυνατότητα για δημιουργία προτύπων γνώσης και την εισαγωγή νέων δεδομένων.

Ο Protégé-OWL editor επιτρέπει στους χρήστες:

- Να φορτώσουν και να σώσουν την OWL και τις οντολογίες RDF.

- Να κατασκευάσουν και να απεικονίσουν τις κλάσεις και τις ιδιότητες.
- Να καθορίσουν τα λογικά χαρακτηριστικά των κλάσεων ως OWL εκφράσεις.

Υπάρχουν διάφορα χαρακτηριστικά γνωρίσματα που διακρίνουν το Protégé και το κάνουν να ξεχωρίζει από άλλα εργαλεία δημιουργίας βάσεων γνώσεων. Κανένα άλλο εργαλείο εκτός από το Protégé δεν έχει όλα τα ακόλουθα χαρακτηριστικά γνωρίσματα:

- Διασθητικό και εύχρηστο γραφικό interface
- Εξελισιμότητα: Η βάση δεδομένων του Protégé φορτώνει τα frames μόνο μετά από απαίτηση του χρήστη και χρησιμοποιεί την αποθήκευση στην cache μνήμη όταν απαιτείται. Δεν υπάρχει ουσιαστικά καμία επιδείνωση στην απόδοση καθώς δημιουργούμε όλο και περισσότερα frames, από μερικά έως και αρκετές χιλιάδες. Η κοινότητα χρηστών του Protégé έχει δημιουργήσει μέχρι τώρα βάσεις γνώσεων που αποτελούνται από 150.000 frames.
- Plug-in αρχιτεκτονική: Μπορούμε εύκολα να επεκτείνουμε το Protégé με διάφορα Plug-ins, ανάλογα με τον τομέα που μας ενδιαφέρει. Κάποιες προτάσεις είναι οι ακόλουθες:
 - Μικρά user-interface components που ταιριάζουν και μπορούν να χρησιμοποιθούν στις φόρμες του Protégé.
 - Back-end plug-ins που χρησιμοποιούν τους δικούς μας μηχανισμούς αποθήκευσης.
 - Νέες εφαρμογές που συνδέονται περίπλοκα με μια βάση γνώσεων ως ένα Protégé tab.

2.2.7. Εφαρμογή του Protégé

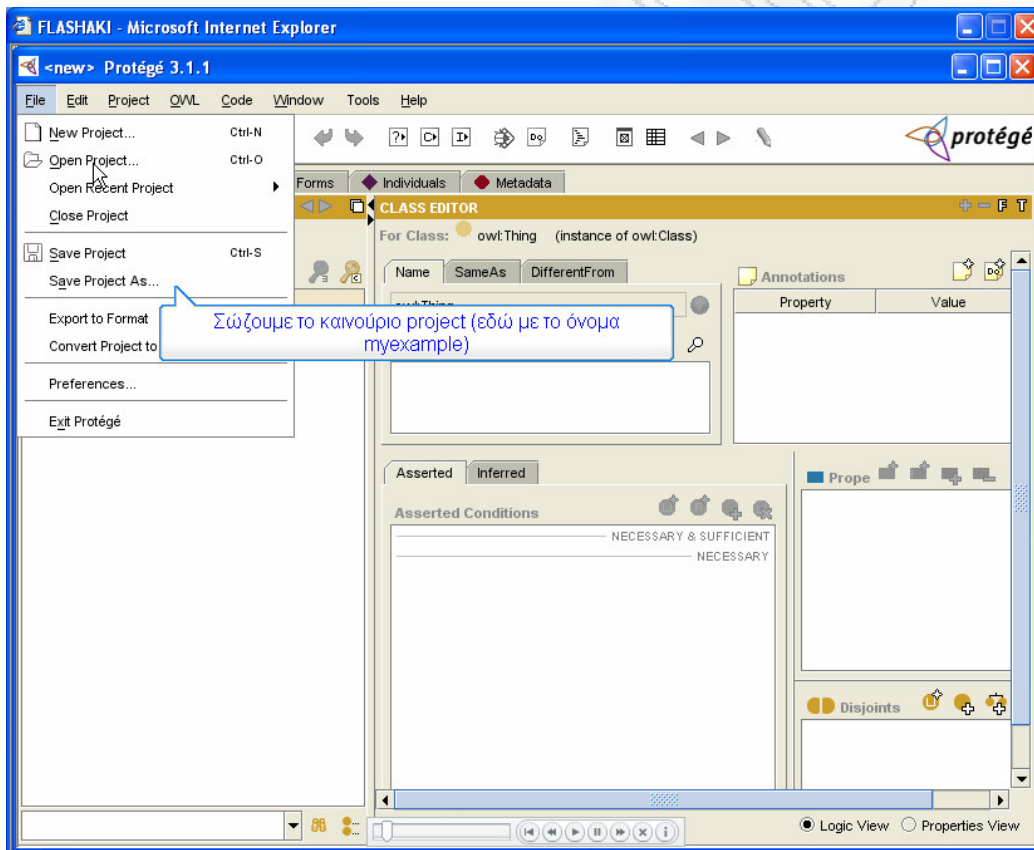
Η παραδειγματική υλοποίηση μίας οντολογίας πραγματοποιήθηκε με τη χρήση του εργαλείου που ήδη περιγράψαμε και παρατίθεται σε μορφή βίντεο.

Το αντικείμενο της οντολογίας μας είναι η διαφήμιση, όσον αφορά στο περιεχόμενο, το μέσο μετάδοσης και το προσωπικό που συντελεί στη δημιουργία μίας διαφήμισης. Με βάση λοιπόν το συγκεκριμένο θέμα και με τη βοήθεια του Protégé, δημιουργούμε τις κατάλληλες κλάσεις, αποδίδουμε σε αυτές τις ιδιότητες που μας ενδιαφέρουν, ορίζουμε τις μεταξύ τους συσχετίσεις και τους περιορισμούς των ιδιοτήτων και τέλος δημιουργούμε κάποια instances.

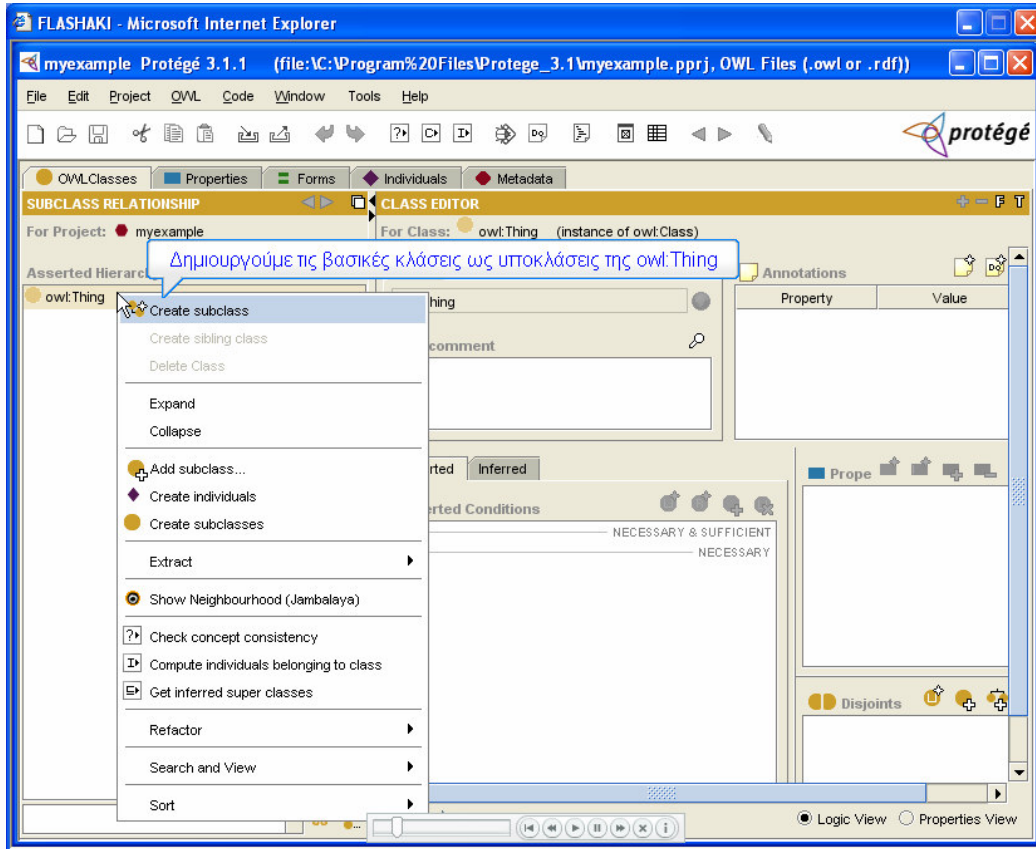
Το συγκεκριμένο βίντεο έχει χωριστεί σε ενότητες, σύμφωνα πάντα με τη λογική που πρέπει κανείς να ακολουθήσει ώστε να σχηματίσει μια ολοκληρωμένη οντολογία. Σε κάθε βήμα, κατά τη δημιουργία της οντολογίας, υπάρχει επεξήγηση όσον αφορά στις επιλογές μας, ενώ έχουμε ανά πάσα στιγμή τη δυνατότητα να μεταβούμε σε προηγούμενο ή επόμενο σημείο της περιγραφής, μέσα από το κατάλληλο μενού που περιλαμβάνεται στο βίντεο.

Ενδεικτικά παραθέτουμε στη συνέχεια κάποιες εικόνες.

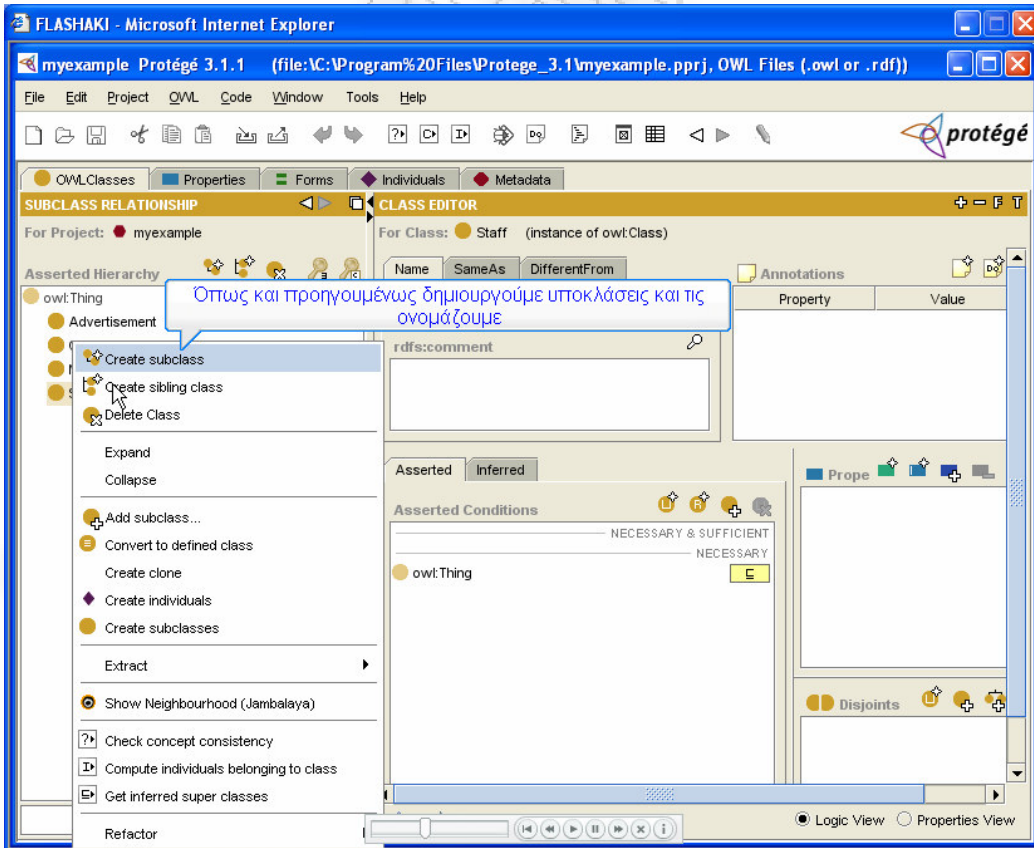
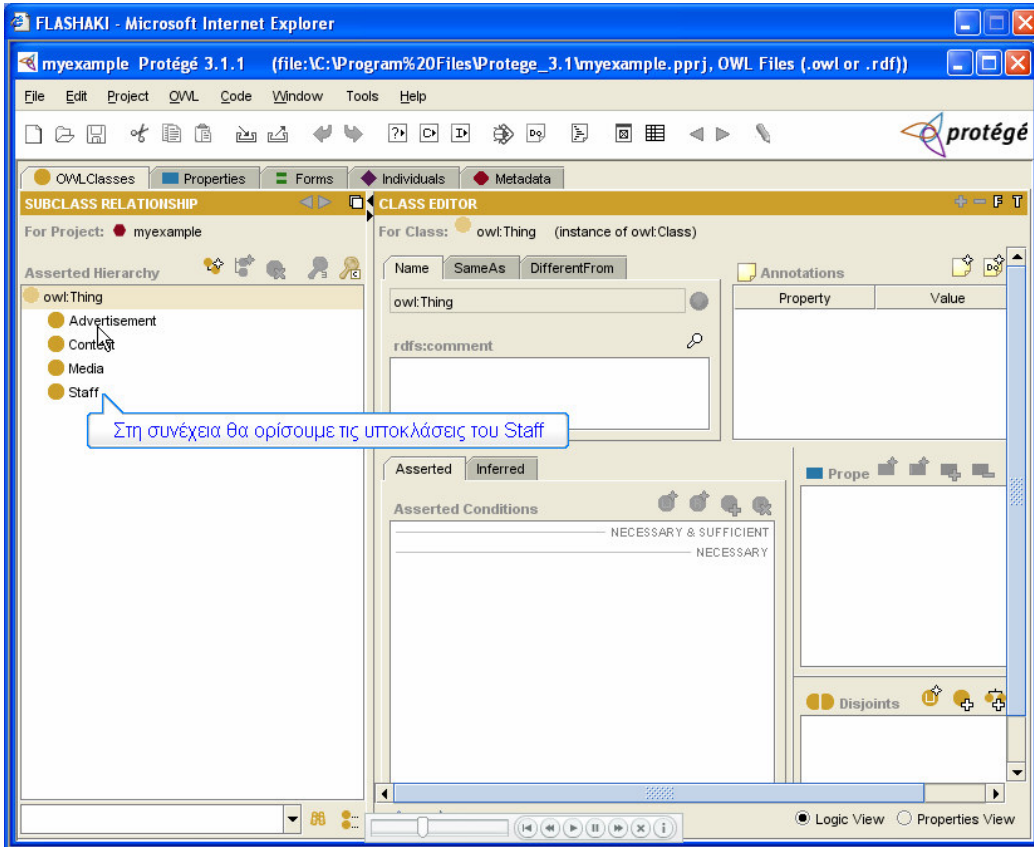
Αρχικά δημιουργούμε ένα καινούριο project και το σώζουμε:



Στη συνέχεια δημιουργούμε τις κλάσεις της οντολογίας μας, σύμφωνα με τις βασικές έννοιες που έχουμε ορίσει και μας ενδιαφέρουν.



Και σε κάθε κλάση, ανάλογα αν χρειάζεται ή όχι, ορίζουμε υποκλάσεις:



Με ανάλογο τρόπο περιγράφονται όλα τα βήματα που ακολουθούμε για τη δημιουργία της οντολογίας (κλάσεις, υποκλάσεις, ιδιότητες, χαρακτηριστικά και περιορισμοί ιδιοτήτων, instances), με λεπτομερή και αναλυτική περιγραφή των επιλογών μας.

Στο τέλος του κεφαλαίου προτείνονται συγκεκριμένες ασκήσεις, με διαφορετικό περιεχόμενο, ώστε ο εκπαιδευόμενος να εξασκηθεί και να αποκτήσει κάποια άνεση στη χρήση του συγκεκριμένου λογισμικού, προς την κατασκευή μίας ορθής και σωστά δομημένης οντολογίας. Στο σημείο αυτό πρέπει να τονίσουμε πως πάντα υπάρχει η δυνατότητα για μελέτη έτοιμων οντολογιών μέσω των κατάλληλων υπερσυνδέσμων, στους οποίους έχει πρόσβαση από την αντίστοιχη σελίδα.

Το τελικό αποτέλεσμα της οντολογίας που κατασκευάσαμε μπορούμε να το δούμε σε γραφική αναπαράσταση κάνοντας χρήση ενός άλλου εργαλείου για οντολογίες, του *Altova SemanticWorks 2007*. Μία περιγραφή του συγκεκριμένου λογισμικού ακολουθεί, ενώ στη συνέχεια παρατίθενται και αντιπροσωπευτικές εικόνες των κλάσεων που κατασκευάσαμε με τη βοήθεια του *Protégé*.

2.2.8. Altova SemanticWorks

2.2.8.1 Περιγραφή

Το *Altova SemanticWorks* είναι ένα εργαλείο το οποίο μας επιτρέπει να δημιουργήσουμε γραφικά και να εκδώσουμε τα έγγραφα RDF, τα λεξιλόγια RDFS, και τις οντολογίες σε OWL με τον πλήρη έλεγχο της σύνταξης. Είναι ένα εύχρηστο εργαλείο το οποίο μας δίνει τη δυνατότητα για:

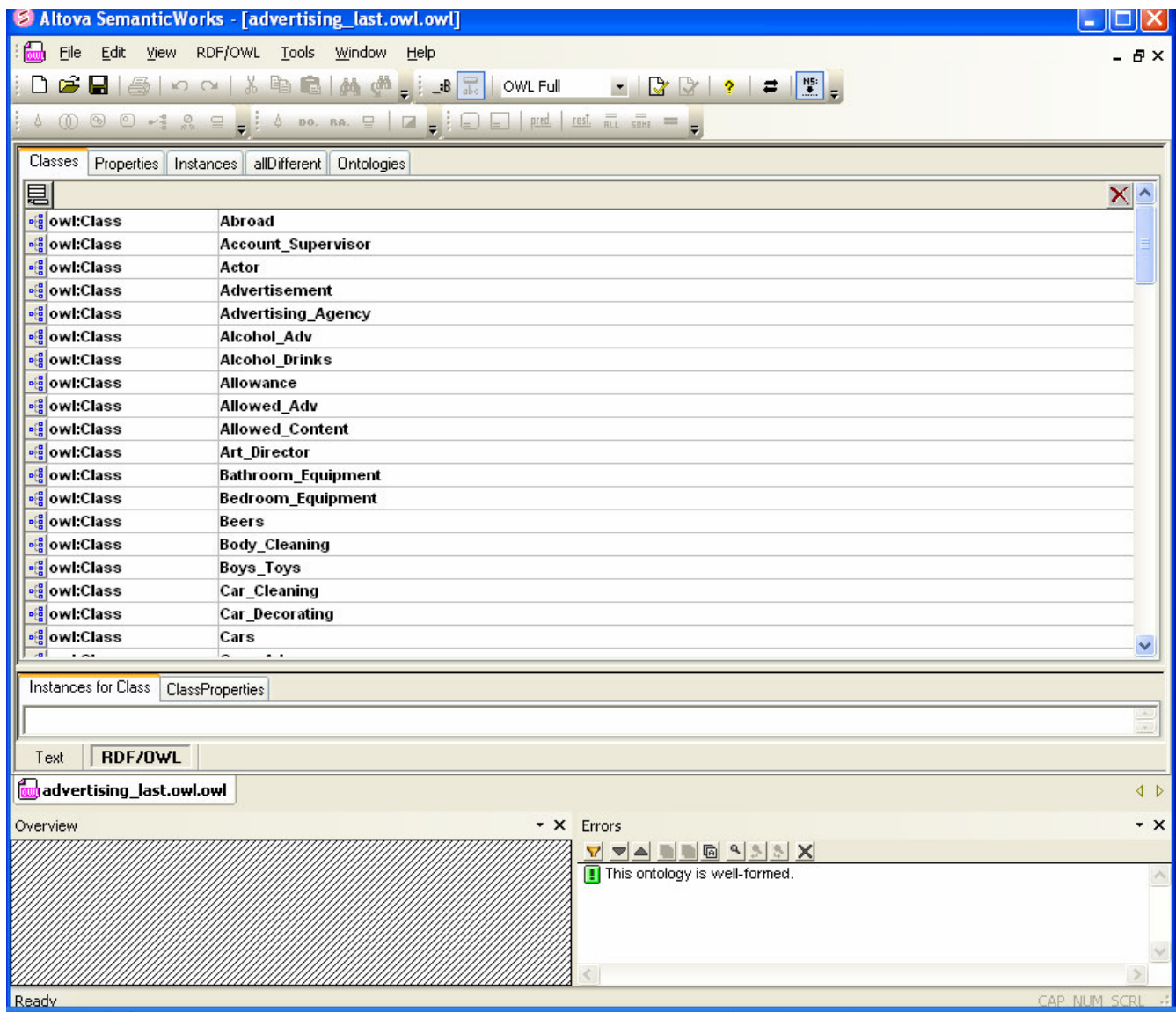
- Οπτική δημιουργία και επεξεργασία αρχείων που είναι γραμμένα σε σχήμα RDF, RDF (RDFS), OWL Lite, OWL DL και OWL Full, καθώς χρησιμοποιούν ένα εύχρηστο οπτικό interface και επιτρέπουν τη λειτουργία drag-and-drop.
- Έλεγχο της σύνταξης ώστε να εξασφαλίσουμε την προσαρμογή με τις RDF/XML προδιαγραφές.
- Έκδοση των σχημάτων RDF/XML έχοντας πρόσβαση στο οπτικό σχέδιο RDF/OWL.
- Εκτόπωση των γραφικών αναπαραστάσεων RDF και OWL που μπορούν να αποτελέσουν υλικό για τις εφαρμογές του Σημασιολογικού Ιστού.

Ανά πάσα στιγμή μπορούμε να μεταβούμε από το γραφικό τμήμα της RDF/OWL στην κειμενική απόδοση, για να δούμε πώς το έγγραφο μας χτίζεται σε RDF/XML, ενώ η αυτόματη παραγωγή του κώδικα που προκύπτει από τις εφαρμογές μας μας επιτρέπει να πειραματιστούμε με τις έννοιες του Σημασιολογικού Ιστού χωρίς να πρέπει να γραφεί ο περιπλοκος κώδικας.

2.2.8.2 Εφαρμογή

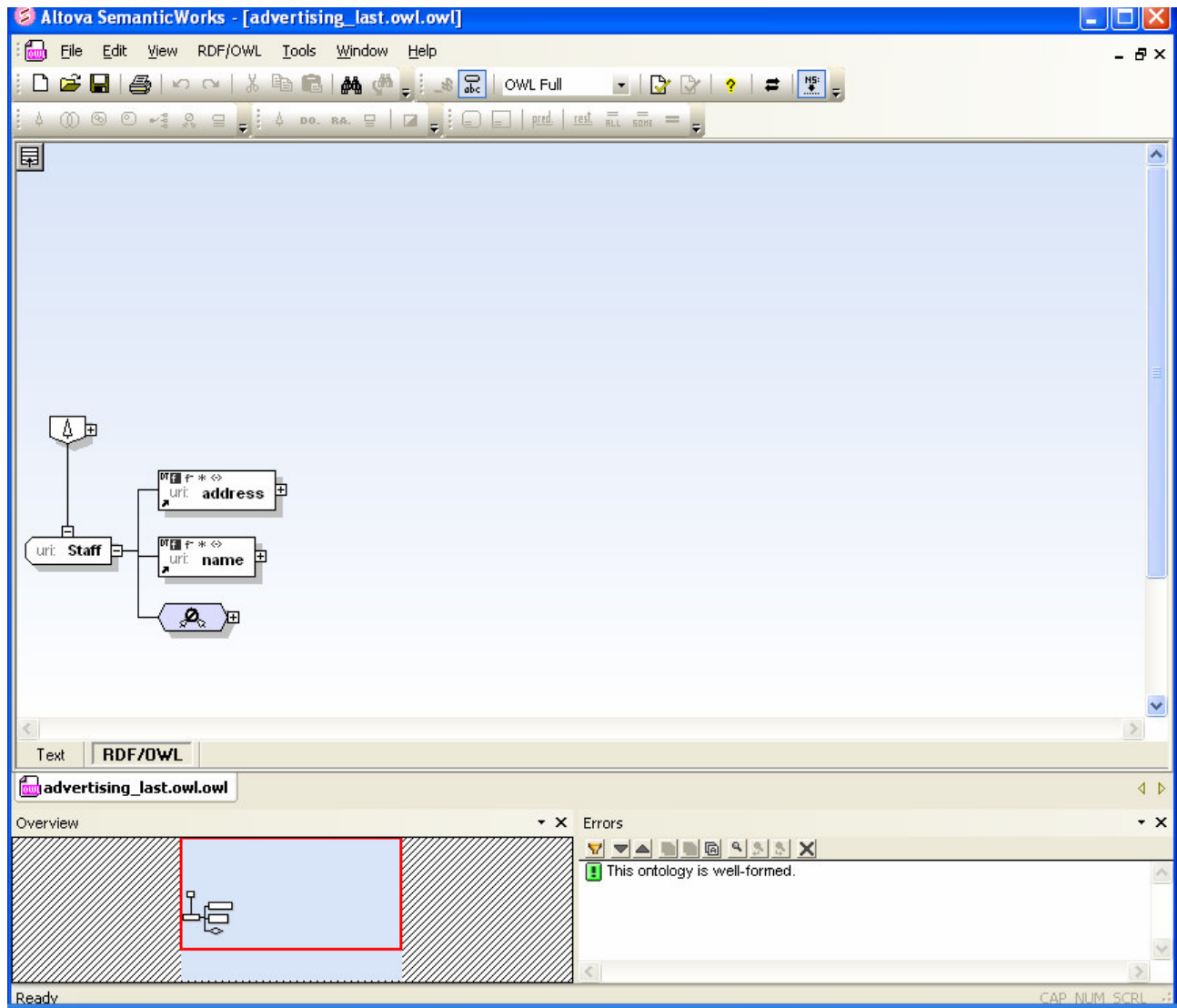
Εισάγοντας την οντολογία που κατασκευάσαμε, με τη βοήθεια του εργαλείου Protégé, στο SemanticWorks, μπορούμε να δούμε γραφικά τις διάφορες κλάσεις και τις μεταξύ τους συσχετίσεις, αλλά ταυτόχρονα έχουμε τη δυνατότητα να ελέγξουμε την ορθότητα της οντολογίας μας.

Ανοίγοντας την εφαρμογή μας μέσα από το SemanticWorks, μπορούμε να δούμε τις διάφορες κλάσεις διατεταγμένες σε αλφαβητική σειρά:

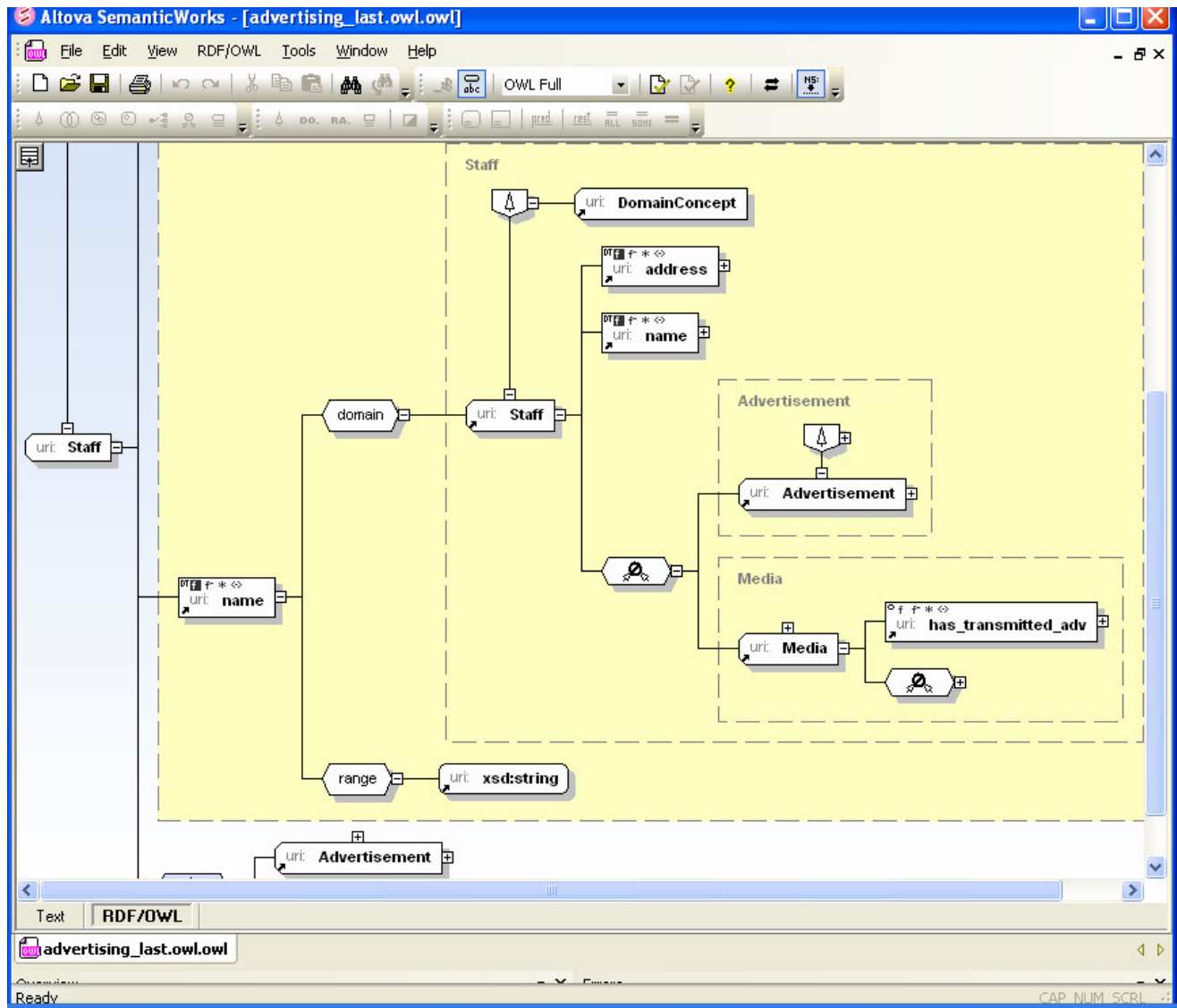


Στο παράθυρο που εμφανίζεται κάτω δεξιά στην οθόνη μας, μπορούμε να διακρίνουμε την πιστοποίηση για την ορθότητα της οντολογίας μέσω της φράσης "This ontology is well formed".

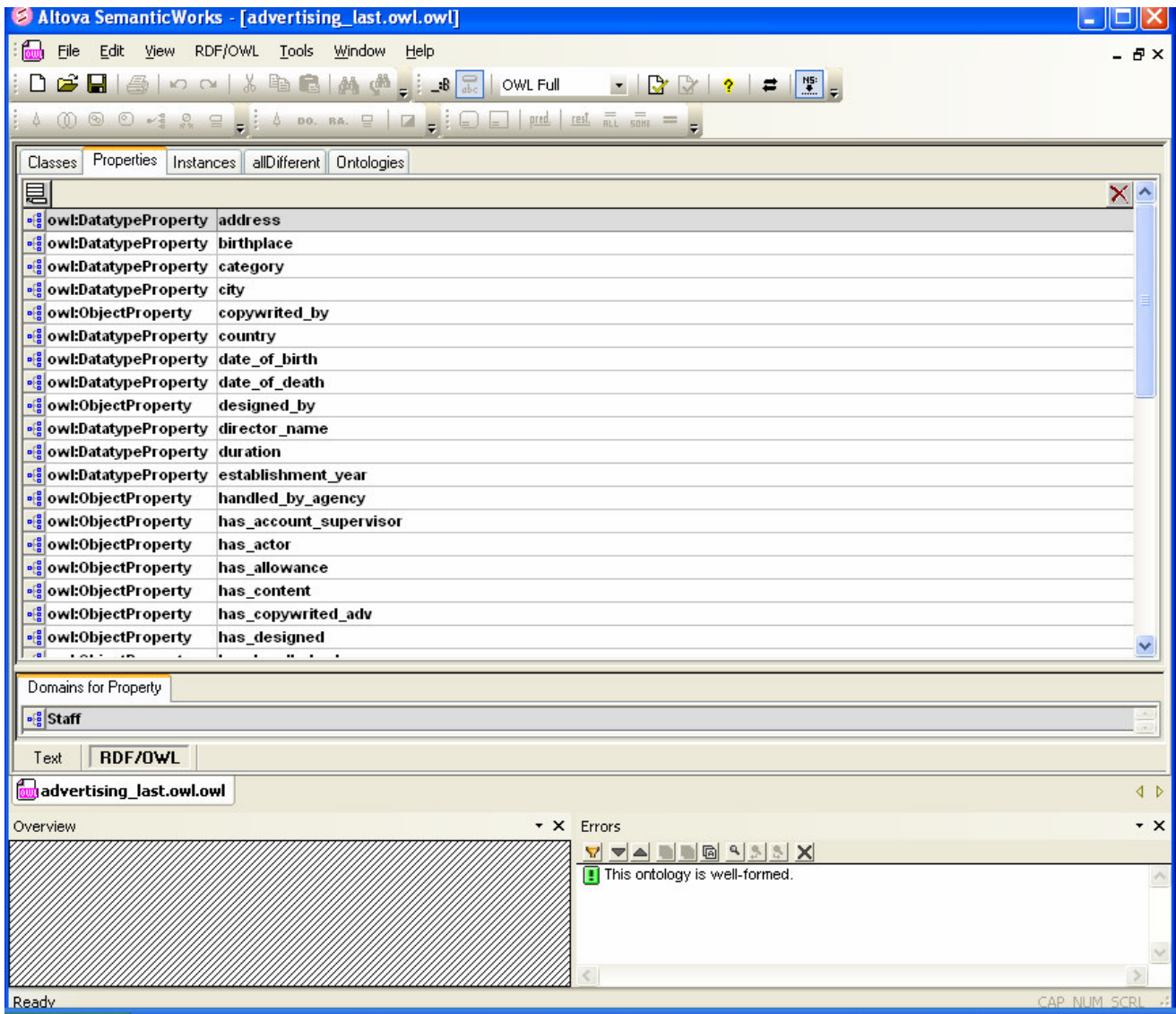
Επιλέγοντας κάποια από τις κλάσεις που έχουμε κατασκευάσει και πατώντας το εικονίδιο στα αριστερά της, μπορούμε να δούμε τη γραφική απεικόνισή της. Για παράδειγμα, επιλέγοντας την κλάση "Staff" προκύπτει η εξής εικόνα:



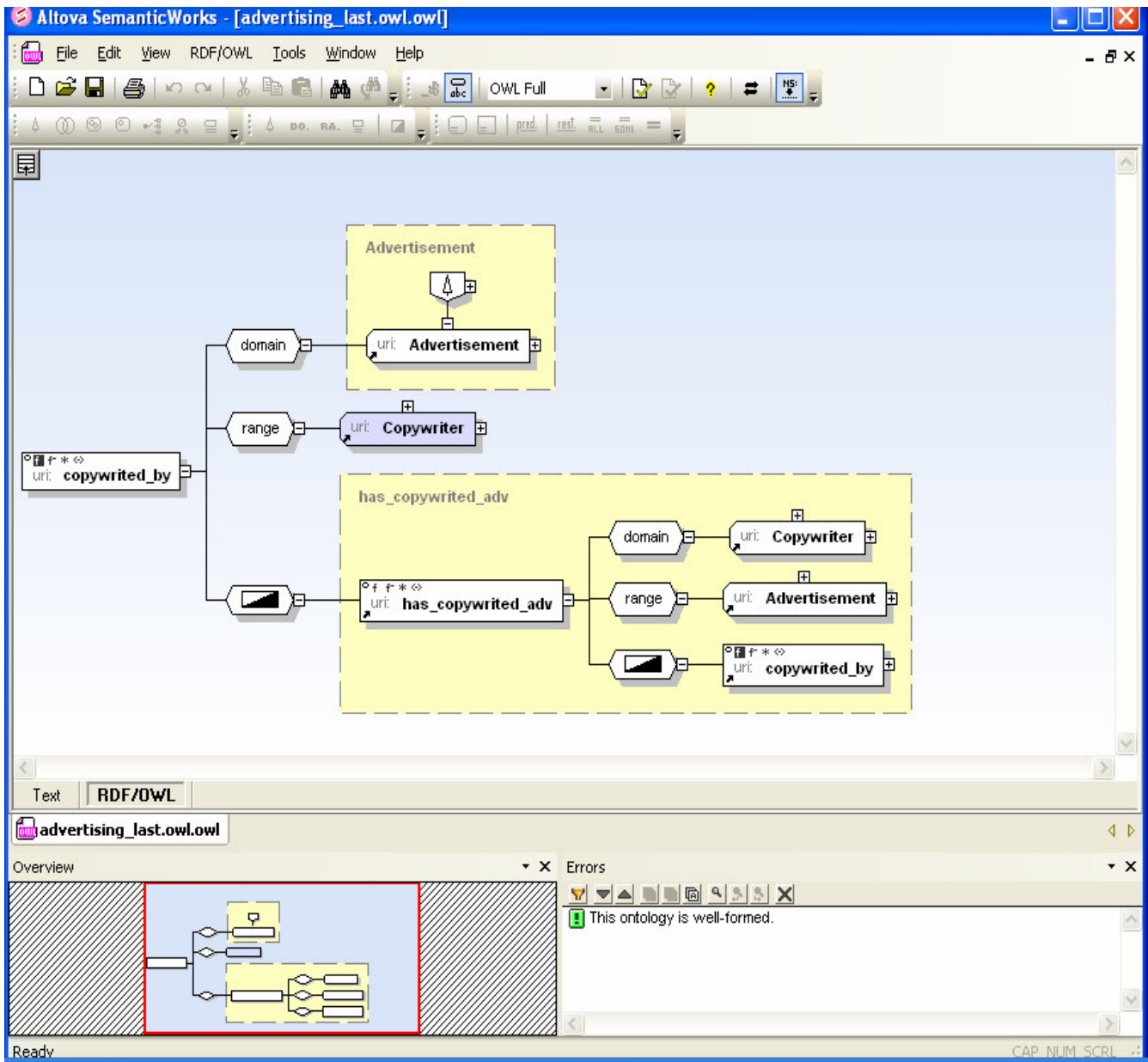
Αν επιλέξουμε να επεκτείνουμε τις ιδιότητες που έχουν αποδοθεί στην συγκεκριμένη κλάση, πατώντας στο "+" καθενιάς από αυτές, προκύπτουν οι συσχετίσεις που έχουμε ορίσει με άλλες κλάσεις, είτε πρόκειται για υπερκλάσεις είτε για υποκλάσεις, τα domain και range των ιδιοτήτων, τυχόν υπο-ιδιότητες που μπορεί να υπάρχουν, ακόμη και τα πεδία ορισμού αυτών ή και οι περιορισμοί που έχουν οριστεί κατά την ανάπτυξη της οντολογίας. Εξάλλου, τις ίδιες ακριβώς δυνατότητες, ως προς τη δημιουργία της οντολογίας, μας παρέχει και το συγκεκριμένο εργαλείο. Μία εικόνα ενδεικτική είναι αυτή που ακολουθεί, έχοντας επεκτείνει την ιδιότητα "name" που ανήκει στην κλάση "Staff":



Αντίστοιχες εικόνες θα προκύψουν αν επιλέξουμε το tab με τα properties και επεκτείνουμε στη συνέχεια κάποια από αυτά. Η πρώτη εικόνα που βλέπουμε επιλέγοντας το properties είναι όλες οι ιδιότητες τοποθετημένες σε αλφαβητική σειρά:



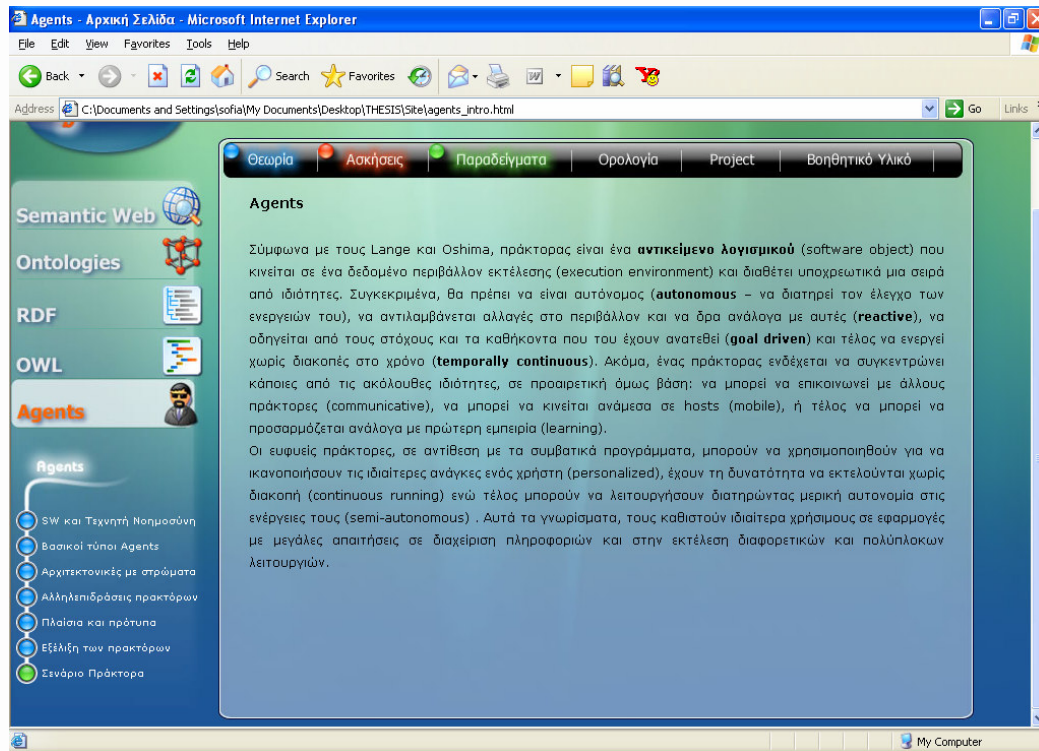
Επιλέγοντας στη συνέχεια κάποια από τις ιδιότητες και επεκτείνοντας τη, μπορούμε να δούμε όλες τις συσχετίσεις που αφορούν στην συγκεκριμένη:



2.2.9. Agents

Το τελευταίο αυτό κεφάλαιο αναφέρεται στους πράκτορες (agents). Αφού οριστεί τι ακριβώς εννοούμε με την έννοια πράκτορας λογισμικού, περιγράφεται η χρησιμότητα τέτοιων προγραμμάτων στο σημασιολογικό διαδίκτυο, αναφέρονται τα διαφορετικά είδη στα οποία διακρίνονται οι agents και γίνεται ακόμη πιο σαφής χρησιμότητά τους μέσα από τα παραδείγματα που παρουσιάζονται και εδώ, όπως και στις υπόλοιπες ενότητες.

Η αρχική σελίδα του κεφαλαίου είναι η ακόλουθη:



Και πάλι οι υποενότητες διακρίνονται στο αριστερό τμήμα της σελίδας. Αυτό που έχει μεγαλύτερη βαρύτητα εδώ είναι το παράδειγμα που πρατίζεται, το οποίο αποσαφηνίζει τη λειτουργία ενός agent στο περιβάλλον του Semantic Web, ενώ μέσα από αυτό τονίζεται παράλληλα και η χρησιμότητα του ίδιου του ιστού, όταν πλέον θα έχει λάβει τη μορφή αυτή και θα λειτουργεί επαρκώς και αποδοτικά, προς όφελος του χρήστη.

3. Εισαγωγή στο SW

3.1. Η ανάγκη για semantic στο Web

Το διαδίκτυο [8], όπως το γνωρίζουμε σήμερα, είναι ένα διαρκώς αυξανόμενο μέσο, αφού ο αριθμός των διαδικτυακών τόπων μεγαλώνει ακόμα εκθετικά χάρη στην ευκολία και την ελευθερία που παρέχεται, σχεδόν στον καθένα, να προσθέσει νέα δεδομένα και ιστοσελίδες. Αυτή η ελευθερία παρουσιάζει σίγουρα αρκετά πλεονεκτήματα, ωστόσο συνεπάγεται και έναν αριθμό μειονεκτημάτων, αφού υπάρχει ελάχιστος ή καθόλου έλεγχος του περιεχομένου των ιστοσελίδων. Το γεγονός αυτό οδηγεί σε προβλήματα όπως η ακρίβεια των δεδομένων, η νομιμότητα (ζητήματα πνευματικών δικαιωμάτων) και η αισθητική. Σε ένα περισσότερο υψηλό επίπεδο σημειώνεται η έλλειψη ενός συστήματος δεικτοδότησης στον ιστό, η οποία καθιστά, μερικές φορές, αδύνατη την εύρεση της επιθυμητής πληροφορίας, ακόμα και αν αυτή είναι δημόσια διαθέσιμη στον ιστό.

Οι μηχανές αναζήτησης [7] που χρησιμοποιούν λέξεις κλειδιά, όπως οι AltaVista, Yahoo και Google, είναι τα κύρια εργαλεία που χρησιμοποιούνται σήμερα. Είναι σαφές ότι το Web δεν θα είχε την τεράστια επιτυχία που έχει αν δεν υπήρχαν οι μηχανές αναζήτησης.

Υπάρχουν σοβαρά προβλήματα που συνδέονται με τη χρήση των μηχανών αναζήτησης, τα οποία μπορούμε να συνοψίσουμε στα εξής:

- Υψηλή ανταπόκριση, χαμηλή ακρίβεια. Ακόμα κι αν οι βασικές σχετικές σελίδες ανακτώνται, έχουν μικρή χρησιμότητα εάν άλλα 28.758 άσχετα με το θέμα έγγραφα ανακτήθηκαν επίσης.
- Χαμηλή ή καμία ανταπόκριση. Συχνά συμβαίνει να μην παίρνουμε οποιαδήποτε απάντηση για το αίτημά μας, ή να μην ανακτώνται οι σημαντικές και σχετικές σελίδες. Αν και η χαμηλή ανταπόκριση είναι ένα λιγότερο συχνό πρόβλημα με τις τρέχουσες μηχανές αναζήτησης, δεν παύει να εμφανίζεται.

- Τα αποτελέσματα είναι ιδιαίτερα ευαίσθητα στο λεξιλόγιο. Συχνά οι αρχικές λέξεις κλειδιά δεν επιστρέφουν τα αποτελέσματα που επιθυμούμε. Σε αυτές τις περιπτώσεις τα έγγραφα που μας ενδιαφέρουν χρησιμοποιούν διαφορετική ορολογία από αυτήν που έχουμε χρησιμοποιήσει στην αρχική μας αναζήτηση. Αυτό δεν είναι ικανοποιητικό καθώς, σημασιολογικά, παρόμοιες ερωτήσεις θα έπρεπε να επιστρέφουν παρόμοια αποτελέσματα.

- Τα αποτελέσματα είναι μονές ιστοσελίδες. Εάν χρειαζόμαστε πληροφορίες που είναι εξαπλωμένες σε διάφορα έγγραφα, πρέπει να θέσουμε διάφορες ερωτήσεις για να συλλέξουμε τα σχετικά με το ζητούμενο έγγραφο, και έπειτα πρέπει συγκεντρώσουμε τις όποιες πληροφορίες.

Συμπερασματικά, παρά τις βελτιώσεις στην τεχνολογία των μηχανών αναζήτησης, οι δυσκολίες παραμένουν ουσιαστικά οι ίδιες. Φαίνεται ότι το μέγεθος του περιεχομένου του Web ξεπερνά την τεχνολογική πρόοδο. Αλλά ακόμα κι αν μια αναζήτηση είναι επιτυχής, είναι ο χρήστης που πρέπει να μελετήσει τα επλεγμένα έγγραφα για να εξαγάγει τις πληροφορίες που τον ενδιαφέρουν. Δηλαδή δεν υπάρχει πολλή υποστήριξη για την ανάκτηση των πληροφοριών, μια πολύ χρονοβόρα δραστηριότητα. Επομένως, ο όρος *ανάκτηση πληροφοριών*, που χρησιμοποιείται σε συνδυασμό με τις μηχανές αναζήτησης, είναι κάπως παραπλανητικός.

Επιπλέον, οι περισσότερες πληροφορίες είναι διαθέσιμες σήμερα σε μια αδύναμη δομημένη μορφή, όπως για παράδειγμα το κείμενο, τον ήχο, το βίντεο. Η τρέχουσα τεχνολογία πάσχει από τους περιορισμούς στις ακόλουθες περιοχές:

- Αναζήτηση πληροφοριών. Οι επιχειρήσεις αναζητούν πληροφορίες συνήθως με βάση λέξεις - κλειδιά στις μηχανές αναζήτησης, οι περιορισμοί των οποίων έχουν ήδη περιγραφεί.
- Εξαγωγή των πληροφοριών. Απαιτείται ανθρώπινος χρόνος και προσπάθεια για να μελετηθούν τα ανακτημένα έγγραφα που προέκυψαν από την εκάστοτε αναζήτηση. Οι τρέχοντες ευφυείς πράκτορες είναι ανίκανοι να εκτελέσουν αυτή τη διαδικασία σε ικανοποιητικό βαθμό.
- Διατήρηση των πληροφοριών. Στον τομέα αυτό συναντούμε προβλήματα σήμερα, καθώς υπάρχουν ασυνέπειες στην ορολογία και παρατηρείται αδυναμία στην αφαίρεση τυχόν ξεπερασμένων πληροφοριών.

- Αποκάλυψη πληροφοριών. Η νέα γνώση που υπάρχει στις εταιρικές βάσεις δεδομένων και δεν διατίθεται στο εξωτερικό περιβάλλον, εξάγεται χρησιμοποιώντας την ανάσχυση δεδομένων. Εντούτοις, αυτός ο στόχος είναι ακόμα δύσκολος στην υλοποίηση, καθώς συναντούμε αδύναμα δομημένες συλλογές των εγγράφων.
- Προβολή των πληροφοριών. Συχνά είναι επιθυμητό να περιοριστεί η πρόσβαση σε ορισμένες πληροφορίες από συγκεκριμένες ομάδες υπαλλήλων.

Μια εναλλακτική προσέγγιση είναι η αναπαράσταση του περιεχομένου του Web σε μια μορφή που είναι ευκολότερα κατανοητή από τους υπολογιστές και η χρήση ευφυών τεχνικών, έτσι ώστε να εκμεταλλευθούμε αυτές τις αναπαραστάσεις. Αναφερόμαστε σε αυτό το σημείο στην μετάβαση από τον παγκόσμιο ιστό στον Σημασιολογικό Ιστό. Είναι σημαντικό να γίνει κατανοητό ότι το Semantic Web δεν θα είναι μια νέα σφαιρική λεωφόρος της πληροφορικής παράλληλη στο υπάρχον World Wide Web, αλλά αντ' αυτού θα εξελιχθεί βαθμιαία από το υπάρχον Web.

3.2. Τι είναι ο σημασιολογικός ιστός και γιατί ενδείκνυται

Ο σημασιολογικός Ιστός διαδίδεται από την κοινοπραξία World Wide Web (W3C), ένα διεθνές σώμα τυποποίησης για το Web. Η κατευθυντήρια δύναμη της πρωτοβουλίας του σημασιολογικού Ιστού είναι ο Tim Berners - Lee, το ίδιο το πρόσωπο που εφηύρε το WWW προς το τέλος της δεκαετίας του '80. Στόχος είναι η πραγματοποίηση του αρχικού οράματός του για τον Ιστό, ένα όραμα όπου η έννοια των πληροφοριών θα διαδραματίσει έναν πολύ σημαντικότερο ρόλο από αυτόν που έχει στον σημερινό Ιστό.

Το σημασιολογικό δίκτυο αποτελεί μια απάντηση στην αύξηση των απαιτήσεων του χρήστη, αφού ο κύριος σκοπός του είναι η δημιουργία ενός περιβάλλοντος όπου όλες οι πληροφορίες θα είναι δυνατό να γίνουν κατανοητές από τα εργαλεία που χρησιμοποιούνται από το χρήστη. Από τη στιγμή που τα δεδομένα γίνονται διαθέσιμα με τέτοιο τρόπο έτσι ώστε να γίνονται κατανοητά από τις μηχανές, τότε μπορούν να θεωρηθούν ως πληροφορία χρήσιμη για το χρήστη.

Ο σημασιολογικός Ιστός δεν είναι μόνο το World Wide Web [9]. Αντιπροσωπεύει ένα σύνολο τεχνολογιών που θα λειτουργήσει εξίσου καλά στα εσωτερικά εταιρικά intranets. Αυτό είναι ανάλογο με τις υπηρεσίες Ιστού που αντιπροσωπεύουν τις

υπηρεσίες όχι μόνο σε ολόκληρο το Διαδίκτυο αλλά και μέσα στο ενδοδίκτυο μιας εταιρίας. Έτσι, ο σημασιολογικός Ιστός θα επιλύσει διάφορα βασικά προβλήματα που αντιμετωπίζουν οι τρέχουσες αρχιτεκτονικές τεχνολογίες πληροφοριών. Ενδεικτικά αναφέρουμε τα εξής:

- Η **υπερφόρτωση πληροφοριών** είναι το προφανέστερο πρόβλημα που χρήζει λύσης. Φυσικά, το πρόβλημα έχει αυξηθεί με τη διάδοση του Διαδικτύου, του ηλεκτρονικού ταχυδρομείου, και τώρα του στιγμιαίου μηνύματος. Δυστυχώς, η προκατάληψή μας ως προς την παραγωγή ενάντια στην επαναχρησιμοποίηση της γνώσης έχει σαν αποτέλεσμα τη διόγκωση του συγκεκριμένου προβλήματος.

- Το **σύστημα stoveripe**

Το Stoveripe είναι ένα σύστημα όπου όλα τα συστατικά είναι συνδεδεμένα έτσι ώστε να εργάζονται μόνο από κοινού. Επομένως, οι πληροφορίες ρέουν μόνο στο stoveripe και δεν μπορούν να μοιραστούν σε άλλα συστήματα που τα χρειάζονται. Για παράδειγμα, ο πελάτης μπορεί μόνο να επικοινωνήσει με συγκεκριμένο υλικό - λογισμικό που καταλαβαίνει μόνο μια ενιαία βάση δεδομένων με ένα συγκεκριμένο σχήμα.

Η διάσπαση των stoveripe συστημάτων πρέπει να εμφανιστεί σε όλες τις σειρές των αρχιτεκτονικών επιχειρηματικών πληροφοριών. Εντούτοις, οι τεχνολογίες σημασιολογικού Ιστού θα είναι αποτελεσματικότερες στη διάσπαση των συστημάτων βάσεων δεδομένων.

- Η **ανεπιτυχής συγκέντρωση δεδομένων** (συνδυασμός πλήθους πληροφοριών από διαφορετικές πηγές) είναι ένα επαναλαμβανόμενο πρόβλημα σε διάφορους τομείς.

Όσον αφορά στον σημασιολογικό Ιστό, η λέξη σημασιολογικός υποδεικνύει ότι η έννοια των στοιχείων μπορεί να ανακαλυφθεί, όχι μόνο από τους ανθρώπους, αλλά και από τους υπολογιστές. Αντίθετα, η σημασία των πληροφοριών στον Ιστό σήμερα είναι προσιτή στους χρήστες που διαβάζουν ιστοσελίδες και τις ετικέτες των συνδέσμων υπερ-κειμένου, και από προγραμματιστές που γράφουν το εξειδικευμένο λογισμικό για να δουλέψουν με δεδομένα. Η φράση «σημασιολογικός Ιστός» αντιπροσωπεύει ένα όραμα στο οποίο ο υπολογιστής, το λογισμικό, αλλά και οι άνθρωποι μπορούν να βρουν, να διαβάσουν, να καταλάβουν και να

χρησιμοποιήσουν στοιχεία από το World Wide Web για να πετύχουν βασικούς στόχους για τους χρήστες.

3.3. Χαρακτηριστικά του SW

Είναι προφανές ότι ο καθένας έχει μία δική του άποψη, μια διαφορετική εικόνα στο μυαλό του, σχετικά με το τι ακριβώς είναι ο σημασιολογικός ιστός. Ωστόσο, διακρίνουμε σε όλες τις απόψεις κάποια κοινά χαρακτηριστικά, όπως τα εξής:

- Σήμανση και ανάκτηση πληροφοριών
- Μεταδεδομένα
- Δεικτοδότηση
- Ο Ιστός ως μεγάλη, διαλειτουργική βάση δεδομένων
- Αυτόματη ανάκτηση δεδομένων
- Υπηρεσίες βασισμένες στο WEB
- Ανακάλυψη των υπηρεσιών
- Ευφυείς πράκτορες λογισμικού

3.3.1. Σήμανση και ανάκτηση πληροφοριών

Ο καθένας αναζητά πληροφορίες με διάφορους τρόπους. Οι βιβλιοθήκες διαθέτουν καταλόγους, ενώ πλέον, σε πολλές περιπτώσεις, επιτρέπουν και την ηλεκτρονική αναζήτηση. Οι μηχανές αναζήτησης είναι ζωτικής σημασίας για τον Ιστό. Για να βρει κανείς πληροφορίες, μια σημασιολογική προσέγγιση του Ιστού θα επέτρεπε να ξεπεραστεί η λέξη-κλειδί και να πραγματοποιηθεί η αναζήτηση ανάλογα με τις έννοιες και τις κατηγορίες.

Οι πληροφορίες είναι διαδεδομένες σε όλο τον Ιστό. Τα περισσότερα συστήματα που χρησιμοποιούν την αναζήτηση πληροφοριών με βάση τη σημασιολογία τους διατηρούν τις ιεραρχίες των εννοιών που χρησιμοποιούν και προσπαθούν να εντοπίσουν τις έννοιες αυτές στα ανακτώμενα έγγραφα. Για να προχωρήσουμε ένα βήμα παραπέρα, πρέπει τα έγγραφα να δηλώνουν με κάποιο τρόπο το λεξιλόγιο και το σύνολο των εννοιών που χρησιμοποιούν και να προσπαθήσουν να εντοπίσουν πού χρησιμοποιούνται μετά από αναζήτηση.

3.3.2. Μεταδεδομένα.

Τα δεδομένα για άλλα δεδομένα ονομάζονται μεταδεδομένα. Για παράδειγμα, ο αριθμός ISBN και το όνομα του συγγραφέα είναι μεταδεδομένα για ένα μυθιστόρημα. Οι τύποι στοιχείων που περιγράφουν τα στοιχεία σε μια βάση δεδομένων εμπίπτουν επίσης στην κατηγορία των μεταδεδομένων. Είναι ακόμα δυνατό να υπάρξουν μετα-μεταδεδομένα (μια δήλωση για την προέλευση ενός κομματιού των μεταδεδομένων μπορεί να θεωρηθεί δεδομένο για τα μεταδεδομένα, ή μετα-μεταδεδομένα). Υπό μια έννοια, τα μεταδεδομένα είναι δεδομένα. Η διάκριση βρίσκεται στην προοριζόμενη χρήση των δεδομένων και στο θέμα των μεταδεδομένων. Τα μεταδεδομένα είναι που θα χρησιμοποιηθούν για αναζητήσεις και ανάκτηση πληροφοριών.

3.3.3. Δεικτοδότηση

Στον κόσμο των φυσικών εγγράφων (όπως τα βιβλία), ο αναγνώστης κρατά σημειώσεις και σχόλια στο περιθώριο, υπογραμμίζει και δίνει έμφαση σε συγκεκριμένα σημεία, συρράπτει νέα στοιχεία στις εκθέσεις, και προσθέτει σκέψεις και ιδέες σε εκείνες των αρχικών συντακτών. Οι γλώσσες σήμανσης, όπως η XML θα έπρεπε ίσως να επιτρέπουν ανάλογες διαδικασίες, αλλά δεν είναι τόσο εύκολο να μεταφέρονται οι σημειώσεις του κάθε χρήστη σε διαφορετική εφαρμογή και διαφορετικό υπολογιστή. Οι ιστοχώροι της μορφής *wiki*- προσπαθούν να υλοποιήσουν κάτι τέτοιο, αλλά αυτή η διαδικασία καλύπτει στο ελάχιστο αυτό που θα επιθυμούσαμε.

Επειδή οι σχολιασμοί πρέπει να είναι διαμοιράσιμοι, και επειδή η έννοια των διαφορετικών τύπων σχολιασμών πρέπει να γίνει κατανοητή ευρέως, η υποστήριξη της δυνατότητας για εκτενείς σχολιασμούς αποτελεί χαρακτηριστικό του σημασιολογικού Ιστού.

3.3.4. Μια τεράστια διαλειτουργική βάση δεδομένων

Σήμερα είναι σύνθηρες να λαμβάνουμε πληροφορίες από μια βάση δεδομένων. Αυτές οι βάσεις δεδομένων είναι γενικά χωριστές και όχι εύκολα χρησιμοποιήσιμες, δεδομένου ότι υπάρχουν και άλλες, συγχωνευμένες πηγές δεδομένων, και πολλές πληροφορίες βρίσκονται έξω από τις βάσεις δεδομένων. Με τον σημασιολογικό ιστό,

θα υπάρχουν τρόποι ώστε να ενοποιηθεί η περιγραφή και η ανάκτηση των αποθηκευμένων στοιχείων, που επιτρέπουν σε ένα μεγάλο μέρος του Ιστού να θεωρηθεί μέρος μιας μεγάλης εικονικής βάσης δεδομένων.

Ας θεωρήσουμε έναν αθλητικό ερευνητή που ψάχνει τα στοιχεία μπέιζμπολ. Υπάρχουν διάφορες on-line βάσεις δεδομένων μπέιζμπολ: Ο σημαντικότερος ιστοχώρος ένωσης μπέιζμπολ είναι μόνο ένας από πολλούς. Αλλά εάν ο ερευνητής μας θέλει να βρει τις στατιστικές απόδοσης για τον Stan Musial, του οποίου η σταδιοδρομία διήρκεσε από τη δεκαετία του '40 έως τη δεκαετία του '60, δεν μπορεί να λάβει τα στοιχεία για ολόκληρη την περίοδο με ένα αμοιβαία συμβατό σχήμα. Τουλάχιστον για τις στατιστικές μπέιζμπολ, υπάρχει κοινή συμφωνία για τους ορισμούς των σημαντικότερων στατιστικών. Εάν ο Ιστός λειτουργήσει ως μια διαλειτουργική βάση δεδομένων, ο ερευνητής θα μπορέσει να πάρει τα στοιχεία από όλα τα σημαντικά site, και το λογισμικό του ερευνητή θα είναι σε θέση είτε να επιδείξει όλα τα στοιχεία μαζί είτε να συνδυάσει αυτόματα τα στοιχεία από, για παράδειγμα, τα σημαντικότερα sites.

3.3.5. Αυτόματη ανάκτηση δεδομένων

Ένα τμήμα λογισμικού, με βάση την αναζήτηση που του έχει ανατεθεί, καθορίζει ποια στοιχεία χρειάζεται, από πού και πώς μπορεί να τα πάρει. Χρησιμοποιώντας το παράδειγμα του μπέιζμπολ, υποθέστε ότι ο ερευνητής μας πρέπει να εντοπίσει τις σωστές ιστοσελίδες, να τις φορτώσει, και έπειτα με κάποιον τρόπο να πάρει τα στοιχεία και να τα οργανώσει. Αυτό είναι δύσκολο να πραγματοποιηθεί και απαιτεί συχνά πολύ χρόνο. Με τον σημασιολογικό Ιστό, η μορφοποίηση των δεδομένων και ο τρόπος πρόσβασης σε αυτά θα περιγραφόταν με έναν τρόπο που θα επέτρεπε στον υπολογιστή να πάρει και να χρησιμοποιήσει τα στοιχεία αυτόματα.

3.3.6. Υπηρεσίες βασισμένες στο WEB

Μία υπηρεσία είναι μια συμπεριφορά που παρέχει ένα όφελος (πραγματοποίηση κρατήσεων, χρονικός προσδιορισμός προγράμματος, ενημέρωση για τιμές, πραγματοποίηση παραγγελιών και άλλα). Αν επιθυμεί κάποιος να αγοράσει ένα προϊόν το οποίο είναι εύθραστο για παράδειγμα, όπως λουλούδια ή τρόφιμα, μόλις επιλέξει το προϊόν, πρέπει να σιγουρευτεί ότι η παράδοσή του θα ταιριάζει με το υπόλοιπο πρόγραμμά του. Η τιμή, οι όροι αγοράς, οι ώρες παράδοσης, και το

πρόγραμμά του μπορούν όλα να θεωρηθούν ως υπηρεσίες που πρέπει να ενεργοποιηθούν και να συντονιστούν. Στον σημασιολογικό Ιστό, όλες αυτές οι υπηρεσίες θα δημοσίευαν τα κατανοητά στον υπολογιστή στοιχεία που σχετίζονται με τις συγκεκριμένες υπηρεσίες, και τα οποία θα επέτρεπαν στη συνέχεια να κάνει όλο τον συντονισμό.

3.3.7. Ανακάλυψη των υπηρεσιών

Για να χρησιμοποιήσουμε τις διάφορες υπηρεσίες, εμείς (και ειδικά το λογισμικό μας) πρέπει να είναι σε θέση να τις εντοπίσει και να μάθει πώς μπορεί να τις επικαλεσθεί. Η προφανέστερη προσέγγιση θα ήταν να δημιουργηθούν κατάλογοι των υπηρεσιών με τυποποιημένες μεθόδους προσπέλασης. Οι υπηρεσίες θα περιγράφονταν με τυποποιημένη ορολογία, και οι πληροφορίες για το πώς να έχει κανείς πρόσβαση σε αυτές και οι διαθέσιμες πληροφορίες θα κωδικοποιούνταν με τυποποιημένους τρόπους.

Ας αντιπαραθέσουμε το παράδειγμά μας με μια φυσική βιβλιοθήκη. Αφού χρησιμοποιήσουμε τον κατάλογο καρτών που διαθέτει η βιβλιοθήκη ή την ηλεκτρονική έκδοσή του, και εξοικειωθούμε με τις ταξινομήσεις, θα μάθουμε πώς να βρίσκουμε τα βιβλία στα ράφια. Εδώ, οι τυποποιημένες μέθοδοι προσπέλασης είναι το γνωστό σύστημα ταξινόμησης και η φυσική τακτοποίηση των βιβλίων στη βιβλιοθήκη.

Μια πιο προηγμένη προσέγγιση θα ήταν να σταλούν τα αιτήματα αναζητήσεων με βάση τις υπηρεσίες που απαιτούνται, και οι προσφερόμενες υπηρεσίες να περιγράψουν τις δυνατότητές τους κατά τέτοιο τρόπο, ώστε ο εν δυνάμει χρήστης να μπορεί να συμπεράνει τις ικανότητές τους και να ξεκινήσει μια συνομιλία για να εντοπίσει οποιεσδήποτε αβέβαιες ή αδιευκρίνιστες πληροφορίες. Επιστρέφοντας στο παράδειγμα της βιβλιοθήκης, θα ήταν σα να είχαμε προσλάβει έναν πεπειραμένο βιβλιοθηκάριο ο οποίος θα μας ενημέρωνε για τα σωστά βιβλία που ψάχνουμε αλλά και για τον τρόπο με τον οποίο θα κατανοήσουμε τις πληροφορίες που αναφέρονται σε αυτά τα βιβλία.

3.3.8. Οι ευφυείς πράκτορες

Ένας πράκτορας είναι κάποιος ή κάτι που ενεργεί εξ ονόματός μας. Ένας πράκτορας λογισμικού θα ενεργούσε με έναν κάπως αυτόνομο τρόπο, επικοινωνώντας με

άλλους πράκτορες λογισμικού (οι οποίοι ενδεχομένως έχουν κάποια ειδικευση) για να ανακαλύψει τις υπηρεσίες, τα προϊόντα, ή τις πληροφορίες αντί για εμάς. Για παράδειγμα, ένας ειδικευμένος πράκτορας ξέρει πώς να αγοράσει τα εισιτήρια αερογραμμών και να κάνει τις κρατήσεις. Ένας άλλος πράκτορας θα εκτελέσει τις απαραίτητες διαδικασίες, θα περάσει τα αποτελέσματα πίσω στον δικό μας πράκτορα, ο οποίος θα μας ειδοποιήσει για το αποτέλεσμα. Είναι σαφές ότι ένα δίκτυο αλληλεπιδρώντων πρακτόρων θα έπρεπε να είναι σε θέση να περιγράψει τους στόχους του χρησιμοποιώντας τα καθιερωμένα λεξιλόγια, για να ανακαλύψει τις υπηρεσίες και τις πηγές πληροφοριών και για να χρησιμοποιήσει πολλές από τις δυνατότητες που περιγράψαμε παραπάνω.

3.4. Πρακτική του SW

3.4.1. SW και γνώση

Ο στόχος του Semantic Web είναι να αναπτυχθούν πιο προηγμένα συστήματα διαχείρισης γνώσης [7]:

- Η γνώση θα οργανωθεί σε εννοιολογικά τμήματα σύμφωνα με τη σημασία της.
- Τα αυτοματοποιημένα εργαλεία θα υποστηρίξουν τη διατήρηση της γνώσης, μέσα από τον έλεγχο για τις ασυνέπειες και την εξαγωγή της νέας γνώσης.
- Η αναζήτηση με βάση τη λέξη - κλειδί θα αντικατασταθεί από τη λογική της ερώτησης - απάντησης: η ζητούμενη γνώση θα ανακτηθεί, θα εξαχθεί, και θα παρουσιαστεί με έναν τρόπο φιλικό προς τον χρήστη.
- Θα υποστηριχθεί η αναζήτηση απάντησης στην τεθείσα ερώτηση σε περισσότερα του ενός έγγραφα.
- Ο καθορισμός της πρόσβασης σε ορισμένες πληροφορίες από ορισμένους χρήστες (ακόμη και τμήματα των εγγράφων) θα είναι δυνατός.

3.4.2. B2B Ηλεκτρονικό εμπόριο

Οι επιχειρήσεις έχουν ανταλλάξει τα στοιχεία τους χρησιμοποιώντας την προσέγγιση ηλεκτρονικής ανταλλαγής δεδομένων (Electronic Data Interchange). Εντούτοις αυτή η τεχνολογία είναι περίπλοκη και κατανοητή μόνο από τους εμπειρογνώμονες. Κάθε

B2B επικοινωνία απαιτεί ξεχωριστό προγραμματισμό, με αποτέλεσμα τέτοιες επικοινωνίες να είναι δαπανηρές.

Το Διαδίκτυο εμφανίζεται να είναι μια ιδανική υποδομή για την ενδοεπιχειρησιακή επικοινωνία. Οι επιχειρήσεις έχουν εξετάσει όλο και περισσότερο τις λύσεις που στηρίζονται στο Internet, και νέα επιχειρησιακά πρότυπα, όπως οι B2B πύλες, έχουν προκύψει. Ακόμα, το B2B ηλεκτρονικό εμπόριο συναντά δυσκολίες λόγω της έλλειψης προτύπων. Η HTML (γλώσσα σήμανσης υπερκειμένων) δεν είναι δυνατό να υποστηρίξει τις προδιαγεγραμμένες δραστηριότητες αποτελεσματικά: δεν παρέχει ούτε τη δομή ούτε τη σημασιολογία των πληροφοριών. Τα νέα πρότυπα XML είναι μια μεγάλη βελτίωση αλλά μπορούν ακόμα να υποστηρίξουν τις επικοινωνίες μόνο σε περιπτώσεις όπου υπάρχει a priori συμφωνία για το λεξιλόγιο που χρησιμοποιείται και για την έννοιά του.

Με την υλοποίηση του Semantic Web οι επιχειρήσεις θα προχωρήσουν σε συνεργασίες αντιμετωπίζοντας τις όποιες διαφορές στην ορολογία, καθώς θα χρησιμοποιηθούν τυποποιημένα αφηρημένα πρότυπα για κάθε τομέα, και τα στοιχεία θα ανταλλαχθούν χρησιμοποιώντας υπηρεσίες μετάφρασης. Η δημοπρασία, οι διαπραγματεύσεις, και η σύνταξη των συμβάσεων θα πραγματοποιούνται αυτόματα (ή ημί-αυτόματα) από τους πράκτορες λογισμικού.

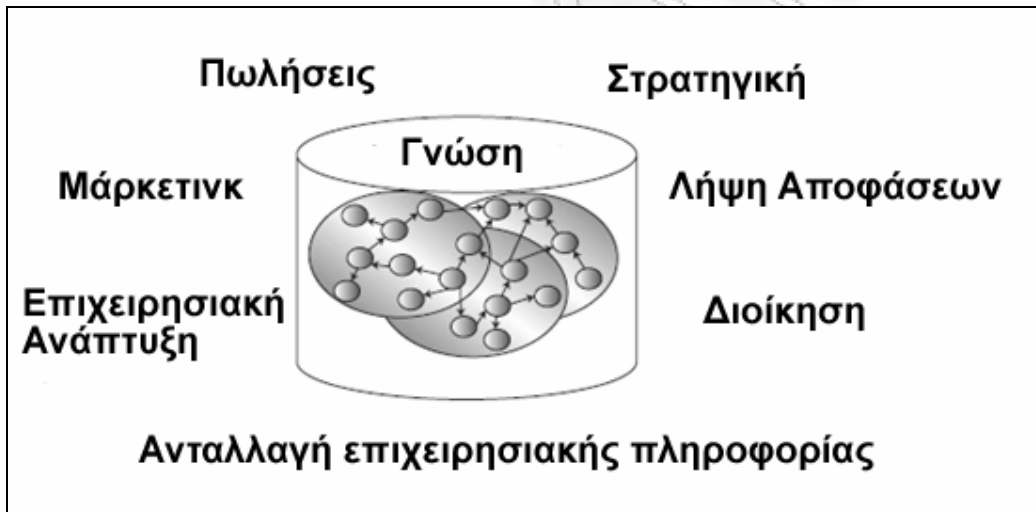
3.4.3. Σημασιολογικός Ιστός και επιχείρηση - Προτερήματα

Η γνώση είναι δύναμη. Σήμερα, που πνιγόμαστε μέσα σε ένα πλεόνασμα πληροφοριών, συνειδητοποιούμε ότι πρέπει να είμαστε σε θέση να διακρίνουμε τις σωστές πληροφορίες για να λάβουμε τις σωστές αποφάσεις. “Η επιχείρηση που έχει τις καλύτερες πληροφορίες, γνωρίζει πού να τις εντοπίσει όταν χρειαστεί και μπορεί να τις χρησιμοποιήσει το συντομότερο δυνατό, είναι εκείνη που τελικά κερδίζει” είναι η επικρατούσα λογική στην εποχή μας.

Επίσης έχουμε συνειδητοποιήσει ότι η γνώση (η εφαρμογή των στοιχείων), όχι τα ακατέργαστα στοιχεία, είναι σημαντικότερη. Αυτή η επιχείρηση θα εκμεταλλευτεί πλήρως τους πόρους που διαθέτει και θα αποκτήσει ανταγωνιστικό πλεονέκτημα. Η διαχείριση γνώσης είναι το κλειδί.

Ποιος δεν θέλει την καλύτερη γνώση; Ποιος δεν θέλει τις καλές πληροφορίες; Οι παραδοσιακές τεχνικές διαχείρισης γνώσης έχουν αντιμετωπίσει νέες προκλήσεις από το σημερινό Διαδίκτυο: η υπερφόρτωση πληροφοριών, η ανεπάρκεια της λέξης

κλειδιού που χρησιμοποιείται στην αναζήτηση, η έλλειψη έγκυρων πληροφοριών. Ο σημασιολογικός Ιστός είναι εκείνος που μπορεί να δομήσει το χάος της υπάρχουσας πληροφορίας. Εμείς, για να λάβουμε τη γνώση, πρέπει, εκμεταλλευόμενοι τις τεχνολογίες που διατίθενται, να είμαστε σε θέση να χρησιμοποιήσουμε την κατάλληλη σήμανση στις πληροφορίες που διαθέτουμε, έτσι ώστε αυτές να μπορούν να γίνουν κατανοητές από τον υπολογιστή. Όταν ανακαλύπτουμε τις νέες πληροφορίες, πρέπει να έχουμε την απόδειξη ότι μπορούμε πράγματι να τις εμπιστευθούμε και στη συνέχεια πρέπει να είμαστε σε θέση να τις συσχετίσουμε με όποια πληροφορία ήδη διαθέτουμε. Τέλος, χρειαζόμαστε τα εργαλεία για να εκμεταλλευθούμε αυτήν την νέα γνώση. Όλα τα παραπάνω είναι μερικές από τις βασικές έννοιες του σημασιολογικού Ιστού.



Το παραπάνω σχήμα παρέχει μια εικόνα για το πώς μία εταιρία μπορεί να περιστραφεί γύρω από τον εταιρικό σημασιολογικό Ιστό, ο οποίος αντανακλά ουσιαστικά κάθε τμήμα της επιχείρησης. Χρησιμοποιώντας σήμανση για την οργάνωση της διατιθέμενης πληροφορίας, υπάρχει η δυνατότητα αυτή η πληροφορία να οδηγήσει σε γνώση και να συμβάλλει έτσι στη λήψη των σωστών αποφάσεων από την πλευρά των εργαζομένων. Υπάρχει το ενδεχόμενο μία επιχείρηση να διαθέτει πολλές πληροφορίες που δεν χρησιμοποιούνται. Εάν πρόκειται για μεγάλη επιχείρηση, είναι πιθανό να υπάρχουν έργα που ενδεχομένως μπορούν να χρησιμοποιήσουν γνώση που αποτελεί προϊόν κάποιων άλλων, που η ίδια εταιρία είχε αναλάβει παλαιότερα, αποφεύγοντας έτσι τη διπλή προσπάθεια. Κάνοντας χρήση ειδικών προγραμμάτων, θα ήταν εφικτό παλαιότερα παθήματα να

αποτελέσουν μελλοντικά μαθήματα, να παρέχουν τις ανταγωνιστικές πληροφορίες νοημοσύνης και να σώσουν πολύ χρόνο και εργασία. Μία εταιρική βάση γνώσεων που θα μπορούσε να αναζητηθεί και να αναλυθεί από τους πράκτορες λογισμικού, θα επέτρεπε την ύπαρξη εφαρμογών βασισμένων στο WEB, που κερδίζουν πολύ χρόνο και χρήμα. Ακολουθούν μερικά χαρακτηριστικά παραδείγματα.

3.4.4. Λήψη αποφάσεων

Το να διαθέτει κανείς οργανωμένη γνώση, και όχι απλά δεδομένα, συμβάλλει στη λήψη καλύτερων αποφάσεων. Όλες οι επιχειρήσεις αποτελούνται από τμήματα και ομάδες εργασίας οι οποίες διαθέτουν προγράμματα που αποτελούν πηγές πληροφοριών. Ο συνδυασμός των πληροφοριών αυτών των επιμέρους ομάδων και η κατανόηση των μεταξύ τους συσχετίσεων αποτελούν το βασικό μοχλό για τη δημιουργική δράση της εταιρίας.

Το απλούστερο παράδειγμα για να κατανοήσουμε τα παραπάνω είναι η διαδικασία σύνταξης αναφορών. Κάθε υπάλληλος γράφει μια αναφορά, ο διευθυντής παίρνει όλες τις αναφορές και τις συνδυάζει παράγοντας μια αναφορά προγράμματος. Ο διευθυντής τμήματος του διευθυντή προγράμματος λαμβάνει τη αναφορά προγράμματος και δημιουργεί μια αναφορά τμήματος. Τέλος, ο προϊστάμενός του τελευταίου συντάσσει τις αναφορές τμήματος ώστε να προκύψει μια περίληψη η οποία και παραδίδεται στον πρόεδρο της επιχείρησης. Κατά τη διάρκεια αυτής της διαδικασίας, οι πληροφορίες φιλτράρονται έτσι ώστε το τελικό προϊόν να είναι μια κατανοητή έκθεση που χρησιμοποιείται για να ληφθούν αποφάσεις. Δυστυχώς, οι σημαντικές πληροφορίες αρκετά συχνά δεν συμπεριλαμβάνονται στην τελική έκθεση. Αυτό που θα διευκόλυνε τα πράγματα θα ήταν η δημιουργία σημασιολογικών συστημάτων υποστήριξης απόφασης (Decision Support Systems) που εστιάζουν στην ανάλυση και την αλληλεπίδραση πρακτόρων λογισμικού μεταξύ του τελικού χρήστη και του υπολογιστή για τη λήψη απόφασης, προκειμένου να κατορθώσει ο τελικός χρήστης να λάβει τις κατάλληλες αποφάσεις.

Ακόμη και χωρίς συστήματα υποστήριξης απόφασης, οι πράκτορες λογισμικού μπορούν να ελέγξουν τις γνώσεις μας και να παρέχουν τις κατάλληλες προειδοποιήσεις.

3.4.5. Η επιχειρησιακή ανάπτυξη

Είναι ιδιαίτερα σημαντικό για το προσωπικό μίας επιχείρησης να διαθέτει τις πιο πρόσφατες πληροφορίες που θα μπορούσαν να βοηθήσουν την επιχείρηση να διακριθεί.

Ας υποθέσουμε ένα σενάριο όπου ο πωλητής είναι σε μια συνεδρίαση με έναν πιθανό πελάτη. Κατά τη διάρκεια της συζήτησης, ο πωλητής ανακαλύπτει ότι ο πελάτης ενδιαφέρεται πολύ για ένα ορισμένο θέμα. Ο πιθανός πελάτης δηλώνει πως ενδιαφέρεται για τη μίσθωση μιας επιχείρησης ώστε να κατασκευαστεί ένα on-line σύστημα ηλεκτρονικού εμπορίου που χρησιμοποιεί τη βιομετρική ταυτοποίηση. Εάν ο πωλητής είναι σε θέση να φθάσει στην εταιρική βάση γνώσεων γρήγορα, μπορεί να είναι σε θέση να βρει τις σημαντικές πληροφορίες που εκμεταλλεύονται την ευκαιρία. Χρησιμοποιώντας την εταιρική βάση γνώσεων, ο πωλητής θα μπορούσε γρήγορα να αποκριθεί με τέτοιο τρόπο ώστε να πετύχει πολύ εύκολα τον σκοπό του. Λόγω του σημασιολογικού Ιστού που λειτουργεί στην επιχείρηση, δίνεται ευκολότερα η δυνατότητα στην τελευταία να αναλάβει όλο και περισσότερα έργα.

Οι ανταγωνιστικές προτάσεις θα μπορούσαν να είναι μια άλλη σημαντική χρήση του σημασιολογικού Ιστού μίας επιχείρησης. Η επιχείρηση που διαθέτει περισσότερες πληροφορίες σχετικά με τους υποψήφιους πελάτες, το συγκεκριμένο έργο που θα αναλάβει και τις απαιτήσεις που χρειάζεται να καλυφθούν είναι εκείνη που θα βγει κερδισμένη. Επιπλέον, μία βάση δεδομένων όπου παλαιότερες αναφορές και προτάσεις είναι οργανωμένες και συσχετισμένες μεταξύ τους, αποτελεί μία πολύ καλή πηγή πληροφοριών που θα είναι ιδιαίτερα χρήσιμη. Εισάγοντας πληροφορίες στη βάση δεδομένων, ένας πράκτορας λογισμικού δύναται να αναλύει αυτές τις πληροφορίες και να καταλήγει σε συμπεράσματα που υπό άλλες συνθήκες δεν θα γίνονταν αντιληπτά.

Η διαχείριση σχέσης πελατών (CRM) επιτρέπει τη συνεργασία μεταξύ των συνεργατών, των πελατών, και των υπαλλήλων με την παροχή των σχετικών, εξατομικευμένων πληροφοριών από ποικίλες πηγές στοιχείων μέσα σε μία επιχείρηση. Αυτές οι λύσεις έχουν διαδραματίσει σημαντικότατο ρόλο ώστε να διατηρηθεί η εμπιστοσύνη από την πλευρά των πελατών, αλλά ένα εμπόδιο στη δημιουργία μιας τέτοιας λύσης είναι η ταχύτητα στην ενσωμάτωση των πηγών δεδομένων, καθώς επίσης και η δυνατότητα παραλληλισμού των πληροφοριών στα διάφορα τμήματα της επιχείρησης.

Η χρήση των τεχνολογιών που σχετίζονται με τον Σημαιολογικό ιστό στις επιχειρήσεις μπορεί να δημιουργήσει μια εξυπνότερη λύση CRM. Οι εμπειρογνώμονες βιομηχανίας ηλεκτρονικού εμπορίου θεωρούν ότι ο σηματολογικός Ιστός μπορεί να χρησιμοποιηθεί στο ηλεκτρονικό εμπόριο. Σε ένα τέτοιο περιβάλλον, οι επιχειρήσεις έρχονται σε επαφή με τους πιθανούς επιχειρησιακούς συνεργάτες ή τους πελάτες. Παραδοσιακά, αυτή η διαδικασία υλοποιείται από τους μισθωμένους μεσίτες, και πολλοί έχουν προτείνει μια υπηρεσία που χειρίζεται τις διαφημιστικές υπηρεσίες. Οι εμπειρογνώμονες υποστηρίζουν ότι μόνο οι τεχνολογίες σηματολογικού Ιστού μπορούν αρκετά να καλύψουν αυτές τις απαιτήσεις, και θεωρούν ότι ο σηματολογικός Ιστός μπορεί να αυτοματοποιήσει τις διαδικασίες.

Οι ευκαιρίες για τις επιχειρήσεις με τη χρήση των σηματολογικών τεχνολογιών του Ιστού είναι απεριόριστες.

3.4.6. Μεταβίβαση πληροφοριών και γνώσης

Η μεταβίβαση πληροφοριών και γνώσης είναι κυρίαρχη σε οποιαδήποτε επιχείρηση, αλλά δεδομένου ότι οι περισσότερες επιχειρήσεις συνεχώς επεκτείνονται και συλλέγουν περισσότερες πληροφορίες, αυτό είναι μια σημαντική διαδικασία, καθώς τα χάσματα επικοινωνίας είναι αναπόφευκτα. Με λίγη προσπάθεια, μια εταιρική βάση γνώσεων θα μπορούσε τουλάχιστον να περιλαμβάνει μία κατάσταση με την περιγραφή των προγραμμάτων και των αρμοδιοτήτων κάθε ομάδας εργασίας. Η χρησιμοποίηση των υπηρεσιών σηματολογικού Ιστού μπορεί να επιτρέψει κάτι τέτοιο.

3.4.7. Διοίκηση και αυτοματοποίηση

Μια παρενέργεια της κατοχής μιας τέτοιας βάσης δεδομένων είναι η δυνατότητα των προγραμμάτων λογισμικού για αυτοματοποίηση των διοικητικών εργασιών. Η κράτηση για την πραγματοποίηση ενός ταξιδιού, για παράδειγμα, είναι ένα παράδειγμα όπου ο σηματολογικός Ιστός και οι υπηρεσίες του μπόρεσαν να βοηθήσουν να καταστήσουν έναν επίπονο στόχο εύκολο. Ο καθένας έχει τις δικές του, προσωπικές προτιμήσεις ταξιδιού και πρέπει να λάβει υπόψη διάφορα στοιχεία. Η περάτωση των ρυθμίσεων μπορεί απαιτήσει αρκετό χρόνο. Ευτυχώς, με τις υπηρεσίες του σηματολογικού Ιστού, ένα μεγάλο μέρος της εργασίας αυτής θα

μπορούσε να ολοκληρωθεί με μια αυτοματοποιημένη διαδικασία. Εάν υπήρχαν οι προσωπικές προτιμήσεις ταξιδιού στην εταιρική βάση δεδομένων, ο πράκτορας ταξιδιού θα μπορούσε να επιλέξει τις ρυθμίσεις, να συγκρίνει, να αντιπαραβάλλει, να αξιολογήσει τις επιλογές και να σας παρουσιάσει έναν κατάλογο καλύτερων επιλογών. Οι υπηρεσίες του σημασιολογικού Ιστού έχουν τη δυνατότητα να αυτοματοποιήσουν τους σύνθετους στόχους της επιχείρησης.

3.5. Βασικές έννοιες του SW

3.5.1. Οι τεχνολογίες Semantic Web

Τα σενάρια που περιγράφονται παραπάνω δεν αποτελούν επιστημονική φαντασία και δεν απαιτούν επαναστατική επιστημονική πρόοδο για να επιτευχθούν. Μπορούμε εύλογα να υποστηρίξουμε ότι πρόκειται για πρόκληση από μηχανική και τεχνολογική πλευρά και όχι επιστημονική. Σήμερα, οι μέγιστες ανάγκες εντοπίζονται στους τομείς της ολοκλήρωσης, της τυποποίησης, της ανάπτυξης των εργαλείων, και της υιοθέτησης από τους χρήστες. Αλλά, φυσικά, η περαιτέρω τεχνολογική πρόοδος θα οδηγήσει σε ένα πιο προηγμένο Semantic Web από αυτό που μπορεί, σε γενικές γραμμές, να επιτευχθεί σήμερα.

Για την επίτευξη των λειτουργιών που περιγράφηκαν παραπάνω αναφέρουμε τα εξής ως απαραίτητα εργαλεία:

3.5.1.1 Μεταδεδομένα

Σήμερα, το περιεχόμενο του Ιστού είναι μορφοποιημένο έτσι, ώστε να είναι κατανοητό στους ανθρώπους και όχι στα προγράμματα. Η HTML είναι η κυρίαρχη γλώσσα στις ιστοσελίδες. Ένα τμήμα μιας τυπικής ιστοσελίδας ενός γιατρού θα έμοιαζε με το παρακάτω:

```
<h1>Agilitas Therapy Centre</h1>
```

```
Welcome to the home page of the Agilitas Therapy Centre.
```

```
Do you feel pain? Have you had an injury? Let our staff
```

```
Lisa Davenport, Kelly Townsend (our lovely secretary)
```

and Steve Matthews take care of your body and soul.

<h2>Consultation hours</h2>

Mon 11am - 7pm

Tue 11am - 7pm

Wed 3pm - 7pm

Thu 11am - 7pm

Fri 11am - 3pm<p>

But note that we do not offer consultation

during the weeks of the

State Of Origin games.

Για το χρήστη, οι πληροφορίες παρουσιάζονται με ικανοποιητικό τρόπο, αλλά τα μηχανήματα θα έχουν τα προβλήματά τους. Οι αναζητήσεις με βάση λέξεις-κλειδιά θα προσδιορίσουν τις ώρες θεραπείας. Ένας ευφυής πράκτορας ίσως να είναι σε θέση να προσδιορίσει και το προσωπικό του κέντρου. Αλλά θα έχει πρόβλημα να διακρίνει τον θεράποντα ιατρό από τη γραμματέα, και ακόμη μεγαλύτερο πρόβλημα να βρει τις ακριβείς ώρες διαβούλευσης.

Η σημασιολογική προσέγγιση του Ιστού στην επίλυση αυτών των προβλημάτων δεν είναι η ανάπτυξη των υπερευφών πρακτόρων. Αντ' αυτού, προτείνει να αντιμετωπίσει το πρόβλημα από την πλευρά ιστοσελίδων, χρησιμοποιώντας πιο κατάλληλες γλώσσες. Οι σελίδες θα μπορούσαν, εκτός από τις πληροφορίες μορφοποίησης που στόχευαν στην παραγωγή ενός εγγράφου κατανοητό στους χρήστες, να περιέχουν και πληροφορίες σχετικά με το περιεχόμενό τους. Στο παράδειγμά μας, θα μπορούσαν να υπάρχουν πληροφορίες όπως:

<company>

<reatmentOffered>Physiotherapy</reatmentOffered>

<companyName>Agilitas Physiotherapy Centre</companyName>

<staff>

<therapist>Lisa Davenport</therapist>

<therapist>Steve Matthews</therapist>

<secretary>Kelly Townsend</secretary>

</staff>

</company>

Αυτή η αναπαράσταση είναι πολύ ευκολότερα επεξεργάσιμη από υπολογιστές. Ο όρος *μεταδεδομένα* αναφέρεται σε τέτοιου είδους πληροφορίες: δεδομένα για τα δεδομένα. Τα μεταδεδομένα συλλαμβάνουν μέρος της έννοιας των στοιχείων, εξ'ού και ο όρος *σημασιολογικός* στον σημασιολογικό Ιστό. Στο παραδείγμα μας, παραπάνω, δεν φάνηκε να υπάρχει κανένα εμπόδιο στην πρόσβαση στις πληροφορίες της ιστοσελίδας: οι λεπτομέρειες θεραπειάς, τα ραντεβού, οι τιμές και οι περιγραφές των προϊόντων, φάνηκε να μπορούν να ανακτηθούν άμεσα από το υπάρχον περιεχόμενο του Ιστού. Αλλά κάτι τέτοιο δεν είναι εφικτό αν δεν χρησιμοποιηθούν μεταδεδομένα που μπορούν να είναι επεξεργάσιμα από τον υπολογιστή.

Οι πρώτοι χρήστες του Internet αποφάσισαν να υιοθετήσουν την HTML επειδή είχε καθιερωθεί ως πρότυπο και ανέμεναν τα οφέλη που θα προέκυπταν από την πρόωρη υιοθέτησή της. Άλλοι ακολούθησαν όταν περισσότερα και καλύτερα εργαλεία Ιστού διατέθηκαν. Και σύντομα η HTML ήταν παγκοσμίως αποδεκτό πρότυπο.

Κατά τον ίδιο τρόπο υιοθετήθηκε η XML. Αν και δεν επαρκεί για την πραγματοποίηση του σημασιολογικού οράματος του Ιστού, η XML είναι ένα σημαντικό πρώτο βήμα. Οι πρώτοι χρήστες, ίσως κάποιοι μεγάλοι οργανισμοί που τους ενδιέφερε η διαχείριση της γνώσης και το B2B ηλεκτρονικό εμπόριο, θα υιοθετήσουν τα XML και RDF, τα τρέχοντα W3C πρότυπα σχετικά με τον σημασιολογικό Ιστό. Σιγά σιγά όλο και περισσότεροι προμηθευτές εργαλείων και τελικοί χρήστες θα υιοθετούν τη νέα τεχνολογία. Αυτό θα αποτελέσει ένα αποφασιστικό βήμα όσον αφορά στον σημασιολογικό Ιστό, αλλά είναι επίσης μια πρόκληση. Άλλωστε, η μέγιστη τρέχουσα πρόκληση δεν είναι επιστημονική αλλά περισσότερο τεχνολογική.

3.5.1.2 Οντολογίες

Ο όρος οντολογία προέρχεται από τη φιλοσοφία [1,7], όπου χρησιμοποιείται ως όνομα ενός τομέα της φιλοσοφίας, σχετικά με τη μελέτη της φύσης της ύπαρξης, τον κλάδο της μεταφυσικής που αφορά στον προσδιορισμό των ειδών που πραγματικά υπάρχουν και την περιγραφή τους. Για παράδειγμα, η παρατήρηση ότι ο κόσμος αποτελείται από συγκεκριμένα αντικείμενα που μπορούν να ομαδοποιηθούν σε κατηγορίες σύμφωνα με κάποιες κοινές ιδιότητες είναι μια χαρακτηριστική οντολογική δέσμευση.

Ωστόσο, στα πιο πρόσφατα έτη, η οντολογία έχει γίνει μια από τις πολλές λέξεις που χρησιμοποιούνται στην πληροφορική και στις οποίες έχει αποδοθεί ένα συγκεκριμένο τεχνικό νόημα, διαφορετικό από το αρχικό. Σύμφωνα με τους T.R. Gruber και P. Studer: “Μια οντολογία είναι μια ρητή και επίσημη περιγραφή ενός τομέα”. Χαρακτηριστικά, μια οντολογία αποτελείται από έναν πεπερασμένο κατάλογο όρων και τις σχέσεις μεταξύ αυτών των όρων. Οι όροι δείχνουν τις σημαντικές έννοιες (κλάσεις αντικειμένων) του τομέα. Παραδείγματος χάριν, σε μια πανεπιστημιακή ρύθμιση, τα μέλη προσωπικού, οι σπουδαστές, οι σειρές μαθημάτων, τα θέατρα διάλεξης, και οι επιστήμες είναι μερικές σημαντικές έννοιες.

Οι σχέσεις περιλαμβάνουν ιεραρχίες κατηγοριών. Μια ιεραρχία διευκρινίζει ότι μια κατηγορία C είναι υποκατηγορία μιας άλλης κατηγορίας C' εάν κάθε αντικείμενο μέσα στη C περιλαμβάνεται επίσης στην C' .

Εκτός από τις σχέσεις υποκατηγοριών, οι οντολογίες μπορούν να περιλάβουν πληροφορίες όπως:

- ιδιότητες (ο X διδάσκει Y)
- περιορισμούς (μόνο τα μέλη σχολής μπορούν να διδάξουν τα μαθήματα)
- δηλώσεις disjointness (το μάθημα και το γενικό προσωπικό είναι disjoint)
- περιγραφή των λογικών σχέσεων μεταξύ των αντικειμένων (κάθε τμήμα πρέπει να περιλάβει τουλάχιστον δέκα μέλη σχολής)

Στα πλαίσια του Ιστού, οι οντολογίες παρέχουν έναν κοινό τρόπο κατανόησης ενός τομέα. Μια τέτοια κοινή κατανόηση είναι απαραίτητη για να υπερνικήσει τις διαφορές στην ορολογία.

Δύο εφαρμογές ίσως να χρησιμοποιήσουν τον ίδιο όρο με διαφορετική έννοια. Στο πανεπιστήμιο A , ο όρος *course* μπορεί να αναφέρεται σε μία κατεύθυνση (όπως η πληροφορική), ενώ στο πανεπιστημιακό B μπορεί να αναφέρεται σε ένα συγκεκριμένο μάθημα (τεχνολογία υπολογιστών). Τέτοιες διαφορές μπορούν να υπερνικηθούν με την περιγραφή της ιδιαίτερης ορολογίας του συγκεκριμένου τομέα σε μια κοινή οντολογία ή με τον καθορισμό των αντιστοιχίσεων μεταξύ των οντολογιών. Σε καθεμία περίπτωση, είναι εύκολο να δει κανείς ότι οι οντολογίες υποστηρίζουν τη σημασιολογική διαλειτουργικότητα. Οι οντολογίες είναι χρήσιμες για την οργάνωση των ιστοχώρων. Σε πολλές ιστοσελίδες παρουσιάζονται σήμερα στην αριστερή πλευρά τα κορυφαία επίπεδα μιας ιεραρχίας όρων. Ο χρήστης μπορεί να ανοίξει έναν από αυτούς τους όρους για να εμφανιστούν οι υποκατηγορίες.

Επίσης, οι οντολογίες είναι χρήσιμες όσον αφορά στην ακρίβεια των αναζητήσεων στον Ιστό. Οι μηχανές αναζήτησης μπορούν να ψάξουν τις σελίδες που αναφέρονται σε μια ακριβή έννοια σε μια οντολογία αντί της συλλογής όλων των σελίδων στις οποίες ορισμένες, γενικά διαφορετικές λέξεις-κλειδιά εμφανίζονται. Κατ' αυτό τον τρόπο, οι διαφορές στην ορολογία μεταξύ ιστοσελίδας και της τεθείσας ερώτησης μπορούν να υπερνικηθούν. Επιπλέον, οι αναζητήσεις στον Ιστό μπορούν να εκμεταλλευτούν τη γενίκευση/ειδίκευση των πληροφοριών. Εάν μια ερώτηση αποτύχει στην εύρεση οποιωνδήποτε σχετικών εγγράφων, η μηχανή αναζήτησης μπορεί να προτείνει στο χρήστη μια γενικότερη ερώτηση. Ή εάν πάρα πολλές απαντήσεις ανακτώνται, η μηχανή αναζήτησης μπορεί να προτείνει στο χρήστη ορισμένες από αυτές, που είναι πιο ειδικευμένες.

Στην τεχνητή νοημοσύνη (AI) υπάρχει μια μακροχρόνια παράδοση ανάπτυξης και χρησιμοποίησης των γλωσσών οντολογίας. Αυτή τη στιγμή, οι σημαντικότερες γλώσσες οντολογίας είναι οι ακόλουθες:

- η XML παρέχει μια σύνταξη για τα δομημένα έγγραφα αλλά δεν επιβάλλει κανέναν σημασιολογικό περιορισμό στην έννοια αυτών των εγγράφων
- Το σχήμα XML είναι μια γλώσσα για τον καθορισμό της δομής των εγγράφων XML
- η RDF είναι ένα πρότυπο για τα αντικείμενα ("πόρους") και τις σχέσεις μεταξύ τους, παρέχει μια απλή σημασιολογία και για την αναπαράστασή τους χρησιμοποιείται σύνταξη XML
- Το σχήμα RDF είναι μια γλώσσα περιγραφής λεξιλογίου για την περιγραφή των ιδιοτήτων και των κατηγοριών των πόρων RDF, με μια σημασιολογία για τη γενίκευση των ιεραρχιών των ιδιοτήτων και των κατηγοριών
- Η OWL είναι μια πλουσιότερη γλώσσα περιγραφής λεξιλογίου για την περιγραφή των ιδιοτήτων και των κατηγοριών, όπως οι σχέσεις μεταξύ των κατηγοριών (π.χ., disjointness), το πλήθος των στοιχείων ενός συνόλου (π.χ. "ακριβώς ένας"), την ισότητα, την πλουσιότερη περιγραφή των ιδιοτήτων, τις χαρακτηριστικές ιδιότητες (π.χ. συμμετρία), και τις απαριθμημένες κατηγορίες.

3.5.2. Αρχιτεκτονική του SW

3.5.2.1 Το συντακτικό: XML + XML Schema.

Η προτεινόμενη αρχιτεκτονική χρησιμοποιεί την γλώσσα XML (Extensible Markup Language) -*Επεκτάσιμη γλώσσα σήμανσης*-. Η XML παρέχει στους χρήστες την δυνατότητα να δημιουργήσουν κείμενα με απεριόριστα πολύπλοκη δομή και συντακτικό. Έτσι είναι δυνατό να δομηθούν οι πληροφορίες που περιέχονται στα κείμενα με αποτέλεσμα να είναι πιο εύκολα επεξεργάσιμα από τους υπολογιστές.

Για παράδειγμα η παρακάτω πρόταση που είναι ένα απλό κείμενο:

“Η Θεανώ ήπιε το γάλα”

μπορεί να αποδοθεί σε μορφή XML όπως παρακάτω:

```
<sentence>
  <person>
    Η Θεανώ
  </person>
  ήπιε
  <drink>
    το γάλα
  </drink>
</sentence>
```

Τα δομικά μέρη ενός XML κειμένου είναι τα *Στοιχεία (Elements)*. Ένα στοιχείο περιλαμβάνεται ανάμεσα στις *ετικέτες αρχής και τέλους (opening and closing tags)*. Η ετικέτα αρχής περιέχει το όνομα του Στοιχείου ανάμεσα στα σύμβολα “<” και “>” (Π.χ. <sentence>, <drink>) ενώ στην ετικέτα τέλους προστίθεται και το σύμβολο “/” μετά το “<”. (Π.χ </sentence>, </drink>).

Ο συντάκτης ενός XML κειμένου είναι ελεύθερος να ορίσει την δομή του κειμένου του και τα *Στοιχεία* που θα χρησιμοποιήσει. Ένα XML κείμενο λέγεται ότι είναι *Γραμματικά Ορθό* όταν τηρεί τους λεξικογραφικούς κανόνες της γλώσσας XML. (Π.χ Ένα στοιχείο που αρχίζει μετά από ένα άλλο, πρέπει και να τελειώνει πριν. Με άλλα λόγια πρέπει να “φωλιάζει” το ένα μέσα στο άλλο).

Επιπλέον όμως, το πρότυπο XML συμπληρώνεται από το πρότυπο **XML Schema (σχήμα)**. Το XML σχήμα είναι μια γλώσσα στην οποία μπορούμε να γράφουμε “λεξικά” και “γραμματικές” για XML έγγραφα. Ένα XML σχήμα ορίζει τα επιτρεπόμενα στοιχεία, τις ιδιότητές τους καθώς και τον τρόπο με τον οποίο μπορούν να συνδυάζονται ή να διατάσσονται μεταξύ τους μέσα στο XML κείμενο. Με άλλα λόγια αποτελεί το “συντακτικό” του XML κειμένου. Στο παράδειγμά μας το XML

σχήμα θα μπορούσε να δηλώσει τα στοιχεία “person”, “drink” και “sentence”, καθώς επίσης και το ότι το στοιχείο “sentence” περιέχει τα άλλα δύο. Όταν ένα XML κείμενο ακολουθεί τους κανόνες κάποιου XML σχήματος τότε ονομάζεται Έγκυρο (Valid).

Ένα κείμενο σε XML δεν περιέχει αυτόματα κάποιο νόημα: κανένα πρόγραμμα δεν μπορεί να καταλάβει τη σημασία της πιο πάνω πρότασης. Όμως ακόμη και ένα απλό πρόγραμμα συντακτικής ανάλυσης (Parser) μπορεί να αναγνωρίσει στην παραπάνω πρόταση ότι η φράση “Η Θεανώ” είναι ένα στοιχείο τύπου “person” και η φράση “το γάλα” είναι ένα στοιχείο τύπου “drink”.

3.5.2.2 Το μοντέλο δεδομένων RDF

Το Resource Description Framework (RDF) - Περιβάλλον Περιγραφής Πόρων - είναι το πρότυπο που υιοθετήθηκε από το W3C για την περιγραφή πληροφοριακών πόρων και γενικότερα για την αναπαράσταση γνώσης στο περιβάλλον του Διαδικτύου.

Για το RDF, πόρος είναι οτιδήποτε για το οποίο θέλουμε να δηλώσουμε κάτι ή αλλιώς κάτι το οποίο θέλουμε να περιγράψουμε. Πόρος μπορεί να είναι μια σελίδα στο Διαδίκτυο ή ένα δικτυακός τόπος ή φυσικά αντικείμενα ή έννοιες ή και μία λέξη: οτιδήποτε. Δεν έχει σημασία αν ο πόρος είναι προσπελάσιμος στο Διαδίκτυο ή όχι.

Οι πόροι δηλώνονται με τα Universal Resource Identifiers (URI) -Καθολικό Αναγνωριστικό Πόρων-. Το URI είναι κάτι γενικότερο από το γνωστό μας URL του Διαδικτύου. Το URI δεν μας λέει κατ’ ανάγκη πως θα προσπελάσουμε κάτι: ακόμη περισσότερο, δεν δηλώνει απαραίτητα κάτι προσπελάσιμο στο διαδίκτυο. Το URI είναι απλώς ένα παγκόσμιο, μοναδικό και μόνιμο αναγνωριστικό - ταυτότητα του πόρου που περιγράφει.

Το RDF είναι ένα απλό μοντέλο δεδομένων: όλες οι προτάσεις - δεδομένα του RDF αποτελούνται από μια τριπλέτα της μορφής: [Υποκείμενο, Σχέση, Αντικείμενο]. Στην πραγματικότητα, οι περισσότερες φυσικές γλώσσες χρησιμοποιούν αυτή την δομή και η συντριπτική πλειοψηφία της γνώσης που θέλουμε να αναπαραστήσουμε στους υπολογιστές μπορεί να παρασταθεί με αυτή την μορφή. Οι προτάσεις RDF παριστάνονται συνήθως ως κατευθυνόμενοι γράφοι με ετικέτες. Έτσι η πρόταση του προηγούμενου παραδείγματος μπορεί να παρασταθεί ως:

<http://definitions.semanticweb.gr/actions/drink>



Στο περιβάλλον RDF, το Υποκείμενο και η Σχέση που αποτελούν την τριπλέτα δηλώνονται με ένα URI . Το αντικείμενο μπορεί να δηλώνεται με ένα URI ή και μπορεί να είναι ένα αλφαριθμητικό - λέξη.

Τέλος, το πρότυπο RDF ορίζει επίσης το XML συντακτικό: τον τρόπο δηλαδή με τον οποίο οι προτάσεις RDF εκφράζονται ως κείμενα XML. Το γράφημα του παραδείγματος μας γράφεται σε μορφή RDF/XML όπως παρακάτω:

```
<rdf:RDF>
  <rdf:Description about="http://definitions.semanticweb.gr/persons#theano"
    "http://definitions.semanticweb.gr/actions/drink =
    http://definitions.semanticweb.gr/drinks#milk "
</rdf:RDF>
```

3.5.2.3 Το σημασιολογικό επίπεδο: Οντολογίες, RDF Schema + DAML

Η γλώσσα RDF είναι ένα μοντέλο δεδομένων. Υπάρχει όμως και η ανάγκη να περιγράψουμε τα χαρακτηριστικά των αντικειμένων (πόρων) για τα οποία κάνουμε δηλώσεις. Τον ρόλο αυτό παίζει το RDF Schema. Το RDF σχήμα ορίζει ένα λεξικό (εκφρασμένο επίσης σε RDF) για να εκφράζονται οι κατηγορίες (κλάσεις) των πόρων, οι πόροι, οι ιδιότητες τους και οι σχέσεις μεταξύ τους.

Επέκταση του σχήματος RDF αποτελούν οι Οντολογίες: Στα πλαίσια του SW, Οντολογία είναι μια αυστηρή (formal) περιγραφή των αντικειμένων-πόρων και των σχέσεων μεταξύ τους. Οι Οντολογίες επιτρέπουν σε μια κοινότητα χρηστών να έχουν κοινή ονοματολογία και κοινή αντίληψη για τα αντικείμενα-πόρους τα οποία δηλώνουν ή χρησιμοποιούν. Επίσης οι Οντολογίες επιτρέπουν στους χρήστες να περιγράψουν τον "κόσμο" που χρησιμοποιούν (που τους αφορά) με ένα τέτοιο τρόπο, ώστε οι υπολογιστές να μπορούν να επεξεργαστούν και να "συνδέσουν" δεδομένα από διαφορετικούς κόσμους.

Η δυσκολία στις οντολογίες βρίσκεται στο ότι οι κοινότητες χρηστών με κοινά ενδιαφέροντα θα πρέπει να συμφωνήσουν στην οντολογική περιγραφή του πεδίου ενδιαφέροντός τους.

3.5.2.4 Λογική και Απόδειξη

Το επόμενο επίπεδο στην αρχιτεκτονική του SW είναι η χρήση κάποιου μηχανισμού εξαγωγής συμπερασμάτων κυρίως με την χρήση της λογικής. Οι Οντολογίες μπορούν να περιέχουν κανόνες επαγωγής έτσι ώστε ένα πρόγραμμα να μπορεί να “παράγει” νέα γνώση εφαρμόζοντας τους κανόνες αυτούς στα υπάρχοντα δεδομένα.

Στην οντολογία του παραδείγματός μας ένας κανόνας μπορεί να είναι ο παρακάτω:

“Αν ένα πρόσωπο ήπιε γάλα τότε δεν διψάει”.

Με βάση αυτό τον κανόνα και την παραπάνω δήλωση ένα πρόγραμμα μπορεί να συνάγει ότι

“Η Θεανώ δεν διψάει”

Οι υπάρχουσες γλώσσες λογικού προγραμματισμού (π.χ. Prolog) προσφέρουν ένα καλό δείγμα των δυνατοτήτων που θα μπορούσαν να υλοποιηθούν. Μαζί με τη δυνατότητα λογικών επαγωγών θα πρέπει να υπάρχει και η δυνατότητα απόδειξης (απόδειξη εδώ με την έννοια της καταγραφής) της επαγωγής: Μια γλώσσα καταγραφής επαγωγών μπορεί να χρησιμοποιηθεί ώστε τα δεδομένα και οι κανόνες που χρησιμοποιήθηκαν στην επαγωγή να μπορούν να μεταφερθούν και να επιβεβαιωθούν από άλλα συστήματα.

3.5.2.5 Πιστοποίηση (Trust)

Επειδή το Διαδίκτυο δίνει την δυνατότητα σε οποιονδήποτε να δηλώνει οτιδήποτε, είναι απαραίτητο να υπάρχει κάποιος μηχανισμός πιστοποίησης της ορθότητας των δηλώσεων καθώς επίσης και της ακεραιότητας των δεδομένων. Ένας τέτοιος μηχανισμός είναι οι ψηφιακές υπογραφές (Digital Signatures). Οι ψηφιακές υπογραφές είναι κρυπτογραφημένα μπλοκ δεδομένων που συνοδεύουν τα πραγματικά δεδομένα και τις σελίδες του Διαδικτύου και πιστοποιούν ότι τα

δεδομένα αυτά προέρχονται από αξιόπιστη πηγή. Οποιοδήποτε πρόγραμμα χρησιμοποιεί ή διαβάζει δεδομένα από το SW θα πρέπει να ελέγχει πρώτα την ακρίβεια των δεδομένων αυτών.

3.5.3 Software Agents

Τα προγράμματα-πράκτορες (Software Agents) είναι προγράμματα που εκτελούν κάποια λειτουργία (συνήθως χωρίς άμεση επίβλεψη από τον χρήστη) και παράγουν αποτελέσματα με το πέρας της εκτέλεσης αυτής. Συνήθως, τα προγράμματα-πράκτορες περιδιαβαίνουν το Διαδίκτυο και επεξεργάζονται τις πληροφορίες που βρίσκουν στις σελίδες που επισκέπτονται.

Προγράμματα-πράκτορες χρησιμοποιούνται ήδη για λειτουργίες όπως ανεύρεση, ταξινόμηση και επιλογή δεδομένων: π.χ. σύγκριση τιμών του ίδιου προϊόντος σε πολλά ηλεκτρονικά καταστήματα ή ειδοποίηση για την εμφάνιση νέου περιεχομένου σε δικτυακούς τόπους ειδήσεων και ενημέρωσης. Οι τεχνολογίες του SW θα οδηγήσουν σε μια έκρηξη στον αριθμό και τις δυνατότητες των προγραμμάτων αυτών, καθώς το περιεχόμενο του Web θα γίνεται όλο και πιο προσιτό και εύκολα επεξεργάσιμο από τα προγράμματα-πράκτορες.

3.6. SW παράδειγμα

Έστω για παράδειγμα ότι ένας σύμβουλος λογισμικού λαμβάνει ένα νέο πρόγραμμα. Πρόκειται να δημιουργήσει μια σειρά υπηρεσιών Ιστού (Web Services) κάνοντας χρήση SOAP, για έναν από τους μεγαλύτερους πελάτες του. Αρχικά λοιπόν, πρέπει να μάθει κάποια πράγματα σχετικά με το SOAP, χρησιμοποιώντας μία μηχανή αναζήτησης. Δυστυχώς, τα αποτελέσματα που παρουσιάζονται δεν είναι ιδιαίτερα χρήσιμα, καθώς πρόκειται για λίστες σχετικά με απορρυπαντικά πιάτων, σαπουνία προσώπου, ακόμη και σαπουνόπερες που αναμιγνύονται στα αποτελέσματα. Μόνο μετά από το κατάλληλο «κοσκίνισμα» και την ανάγνωση μέσω των συνδέσμων, είναι δυνατό να βρεθούν οι πληροφορίες που απαιτούνται. Λόγω των διαφορετικών σημασιολογικών εννοιών της λέξης "soap", τα αποτελέσματα είναι ποικίλα, και απαιτείται πολλή εργασία για να βρεθούν οι πληροφορίες που απαιτούνται. Εντούτοις, στον σημασιολογικό Ιστό, θα μπορούσαμε να χρησιμοποιήσουμε έναν σημασιολογικό πράκτορα για να ψάξουμε για το "SOAP", όπου ο όρος SOAP είναι

ένας τύπος προδιαγραφής τεχνολογίας που χρησιμοποιείται στις υπηρεσίες Ιστού. Αυτή τη φορά, τα αποτελέσματα της αναζήτησής μας θα είναι σχετικά. Ο σημασιολογικός πράκτορας Ιστού μπορεί επίσης να ψάξει το εταιρικό δίκτυό μας για την προδιαγραφή SOAP και να ανακαλύψει εάν οι συνάδελφοί μας έχουν αναλάβει παρόμοια προγράμματα ή έχουν πραγματοποιήσει κάποια σχετική έρευνα.

Με βάση τις σημασιολογικές πληροφορίες, ο πράκτοράς μας θα μας παρουσιάζει επίσης έναν κατάλογο με σχετικές τεχνολογίες. Γνωρίζοντας ότι οι WSDL, XML, και URI είναι όλες οι τεχνολογίες οι σχετικές με το SOAP, θα πραγματοποιήσουμε κάποια έρευνα πριν αρχίσουμε το πρόγραμμά μας. Οπλισμένοι με τις πληροφορίες που επιστρέφονται από το σημασιολογικό πράκτορα Ιστού, θα διαβάσουμε τις σχετικές προδιαγραφές τεχνολογίας.

Ο σημασιολογικός πράκτορας Ιστού δεν περιλαμβάνει την τεχνητή νοημοσύνη - μάλλον, στηρίζεται στα δομημένα σύνολα πληροφοριών και συμπεράσματος που του επιτρέπουν "να καταλάβει" τη σχέση μεταξύ των διαφορετικών πηγών δεδομένων. Ο υπολογιστής δεν καταλαβαίνει πραγματικά τις πληροφορίες με τον τρόπο που ένας άνθρωπος μπορεί, αλλά έχει αρκετές πληροφορίες για να πραγματοποιήσει τις λογικές συνδέσεις και να λάβει αποφάσεις.

3.7. Εφαρμογές

Ένα πεδίο εφαρμογής του SW είναι οι δικτυακές υπηρεσίες: (Web services). Δικτυακές υπηρεσίες είναι εφαρμογές υπολογιστών που είναι προσπελάσιμες μέσω του Διαδικτύου (π.χ. η δυνατότητα κράτησης και έκδοσης εισιτηρίου μέσω κάποιας σελίδας στο Διαδίκτυο). Το SW θα οδηγήσει στην ανάπτυξη γλωσσών και προτύπων για την περιγραφή τέτοιων υπηρεσιών. Μέσω αυτών των γλωσσών, τα προγράμματα πράκτορες θα είναι σε θέση να ανευρίσκουν αυτόματα τέτοιες υπηρεσίες, να καταλαβαίνουν τις δυνατότητες και τους περιορισμούς τους και τελικά να κάνουν χρήση αυτών των υπηρεσιών για να εκπληρώσουν τον στόχο τους.

Τα πλεονεκτήματα του SW θα επεκταθούν σταδιακά και στον φυσικό κόσμο μέσω συσκευών που θα συνδέονται στο Διαδίκτυο (π.χ. οικιακές συσκευές, κινητά τηλέφωνα, αυτοκίνητα κλπ). Όλο και περισσότερες δραστηριότητες της καθημερινής μας ζωής θα γίνονται απλούστερες και ευκολότερες χάρη στις δυνατότητες αυτοματισμού του νέου Διαδικτύου.

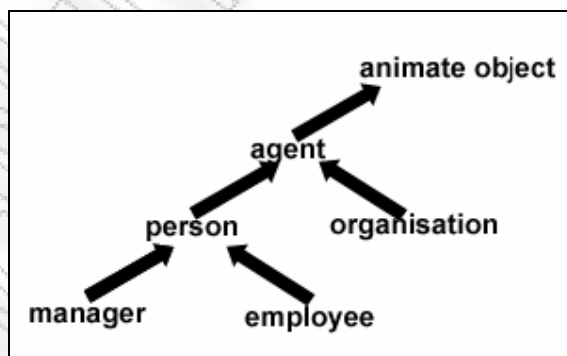
4. Ταξονομίες - Οντολογίες

4.1. Η ανάγκη για ιεράρχηση των data

Το πρώτο βήμα για την υλοποίηση του σημασιολογικού Ιστού και τη χρησιμοποίηση των υπηρεσιών του είναι η έκφραση μιας ταξονομίας σε μορφή κατανοητή από τον υπολογιστή. Αλλά τι είναι μια ταξονομία;

Στην πραγματικότητα, ο κόσμος δεν θα μπορούσε να υπάρξει χωρίς ταξονομίες, δεδομένου ότι το έχουμε στη φύση μας, ως ανθρώπινα όντα, να ταξινομούμε. Η ταξονομία είναι ακριβώς αυτό [4,7]: μια ταξινόμηση υπό μορφή ιεραρχίας. Μια ιεραρχία είναι απλά μια δομή σε μορφή δέντρου. Όπως ένα δέντρο, η ταξονομία έχει μια ρίζα και τους κλάδους. Κάθε διακλαδισμένο σημείο καλείται κόμβος. Οι δύο βασικές ιδέες για μια ταξονομία είναι ότι είναι μια ταξινόμηση και ένα δέντρο.

Στον τομέα της τεχνολογίας πληροφοριών μια ταξονομία ορίζεται ως εξής: Η ταξινόμηση των οντοτήτων πληροφοριών υπό μορφή ιεραρχίας, σύμφωνα με τις θεωρούμενες σχέσεις που οι οντότητες αυτές αντιπροσωπεύουν στον πραγματικό κόσμο. Μία ταξονομία απεικονίζεται συνήθως με τη ρίζα της στην κορυφή, όπως στο ακόλουθο σχήμα.



Κάθε κόμβος της ταξονομίας -συμπεριλαμβανομένης της ρίζας- είναι μια οντότητα πληροφοριών που αντιπροσωπεύει μια πραγματική οντότητα. Κάθε σύνδεση μεταξύ των κόμβων αντιπροσωπεύει μια ειδική σχέση αποκαλούμενη *"is the subclassification"*

of “ (εάν το βέλος της σύνδεσης δείχνει επάνω προς τον κόμβο γονέων) ή “is the superclassification of” (εάν το βέλος της σύνδεσης δείχνει κάτω στον κόμβο παιδιών). Πιθανώς, η ορολογία με την οποία εξοικειωθήκαμε, όπως χρησιμοποιείται στον αντικειμενοστρεφή προγραμματισμό, είναι τα αντίστοιχα “is subclass of” και “is superclass of”.

Μια κλάση είναι μια γενική οντότητα. Στο παραπάνω σχήμα, τα παραδείγματα περιλαμβάνουν την κλάση Person, τις υποκατηγορίες της Employee και Manager, και την superclass της Agent.

Δεδομένου ότι διασχίζουμε την ταξινόμηση προς τη ρίζα που βρίσκεται στην κορυφή, οι οντότητες γίνονται γενικότερες. Καθώς πηγαίνουμε προς τα κάτω στην ταξινόμηση, προς τα φύλλα στο κατώτατο σημείο, οι οντότητες γίνονται πιο εξειδικευμένες. Ο Agent, παραδείγματος χάρη, είναι γενικότερος από το Person, το οποίο είναι με τη σειρά του γενικότερο από το Employee. Αυτό το είδος συστήματος ταξινόμησης καλείται ταξινόμηση γενίκευσης/ειδίκευσης.

Οι ταξινόμιες ενδείκνυνται για να ταξινομήσουν τις οντότητες πληροφοριών σημασιολογικά, δηλαδή βοηθούν στη καθιέρωση μιας απλής σημασιολογίας (η σημασιολογία εδώ ακριβώς σημαίνει "σημασία" ή ένα είδος μεταδεδομένων) για κάποιες πληροφορίες.

Μια ταξινόμηση είναι μια σημασιολογική ιεραρχία στην οποία οι οντότητες πληροφοριών συσχετίζονται είτε μέσω μιας σχέσης subclassification (ή subclass of) είτε μέσω μιας σχέσης superclassification of (ή superclass of). Κάθε οντότητα πληροφοριών διακρίνεται από μια ιδιαίτερη ιδιότητα (distinguishing property) που την καθιστά μοναδική ως υποκατηγορία της οντότητας γονέων της. Στο σχήμα που είδαμε, η ιδιότητα που διακρίνει μια συγκεκριμένη υποκατηγορία στο πιο υψηλό επίπεδο (πιο κοντά στη ρίζα) είναι πιθανώς ένα μεγάλο σύνολο ιδιοτήτων.

Εξετάζοντας τη διάκριση μεταξύ του θηλαστικού και του ερπετού κάτω από την κλάση Vertebrata που είναι ο γονεάς τους (στο σχήμα που φαίνεται στη συνέχεια, μια διαστιγμένη γραμμή μεταξύ Mammalia και Diapsida δείχνουν ότι είναι στο ίδιο επίπεδο, αποτελώντας και οι δύο subclassifications του Vertebrata). Αν και τα θηλαστικά και τα ερπετά έχουν τέσσερα πόδια (κοινές ιδιότητες), τα θηλαστικά είναι warm-blooded και τα ερπετά είναι cold-blooded. Η ιδιότητα αυτή μπορεί να θεωρηθεί τουλάχιστον μια από τις ιδιότητες που διακρίνει τα θηλαστικά από τα ερπετά. Θα μπορούσαν να υπάρξουν και άλλες. Μια άλλη ιδιαίτερη ιδιότητα μεταξύ των θηλαστικών και των ερπετών είναι η ιδιότητα egg-laying. Αν και υπάρχουν

εξαιρέσεις, σε γενικές γραμμές, τα θηλαστικά δε γεννούν αυγά, σε αντίθεση με τα ερπετά.

Kingdom: Animalia
Phylum: Chordata
Subphylum: Vertebrata
Class: Mammalia
Subclass: Theria
Infraclass: Eutheria
Order: Primates
Suborder: Anthropoidea
Superfamily: Hominoidea
Family: Hominidae
Genus: Homo
Species: Sapiens
Class: Diapsida (Reptiles, Dinosaurs, Birds)

Αυτό που πρέπει να μας μείνει στο σημείο αυτό, είναι ότι για κάθε υποκλάση πρέπει να υπάρχει μία ιδιότητα που αποτελεί ιδιαίτερο χαρακτηριστικό της, το οποίο και της επιτρέπει να διακρίνεται από τις υπόλοιπες υποκλάσεις μιας κλάσης. Επιπλέον, η ιδιομορφία που παρατηρούμε είναι ότι ο βαθμός λεπτομέρειας της ιδιότητας αυξάνεται καθώς πηγαίνουμε προς τα κάτω στην ταξονομία.

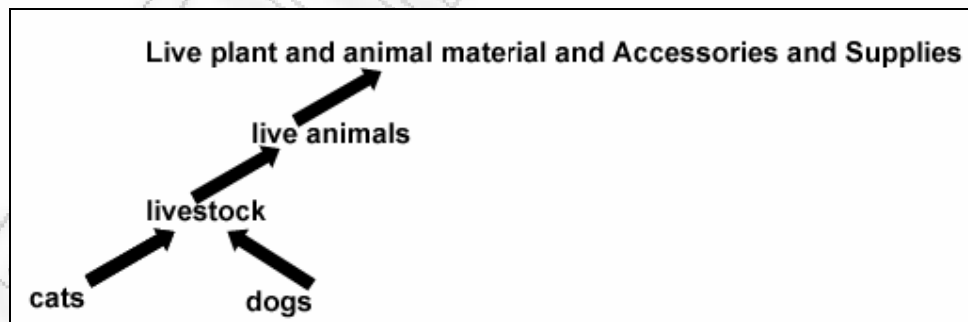
Ας εξετάσουμε ένα παράδειγμα: Είναι ένας διευθυντής υπάλληλος; Θα έπρεπε ένας διευθυντής και ένας υπάλληλος να διαχωρίζονται στο ίδιο επίπεδο στην ταξονομία, ως υποκατηγορίες του Person; Δεν είναι ένας διευθυντής υπάλληλος επίσης; Έτσι, δεν θα έπρεπε διευθυντής και κάποια άλλη οντότητα πληροφοριών (την οποία ονομάζουμε X προς το παρόν) να θεωρηθούν ως υποκατηγορίες του Employee; Ίσως η διάκριση πρέπει να είναι μεταξύ του manager και του nonmanager. Αλλά ίσως αυτές οι διακρίσεις είναι κάπως ανακριβείς. Ίσως το Person είναι μια κλάση οντότητας πληροφοριών, αλλά ο manager και ο employee δεν είναι πραγματικά υποκατηγορίες του Person. Αντ' αυτού, είναι διαφορετικοί ρόλοι (μια διαφορετική σχέση) που οποιοδήποτε δεδομένο πρόσωπο μπορεί να έχει. Αυτή η τελευταία άποψη περιπλέκει την εικόνα, φυσικά, αλλά μπορεί να είναι ακριβέστερη εάν η πρόθεσή μας είναι να διαμορφώσουμε τον πραγματικό κόσμο όσο το δυνατόν με μεγαλύτερη σημασιολογική ακρίβεια. Τελικά, ένας διευθυντής είναι επίσης υπάλληλος ενός οργανισμού και ο τελευταίος έχει επίσης υπαλλήλους που δεν είναι διευθυντές.

Αυτή η ιδέα είναι παρόμοια με ένα subdirectory σε έναν κατάλογο αρχείων: Ένα subdirectory είναι ένα αρχείο που περιέχει άλλα αρχεία. Φυσικά, σε αυτήν την τελευταία περίπτωση, το subdirectory είναι κάτι περισσότερο από μια συλλογή αρχείων. Τα αρχεία είναι μέρος του subdirectory. Και αυτό που πρέπει να προσέξουμε εδώ είναι ότι η σχέση *είναι μέρος του (is part of)* μπορεί να αποτελέσει μια ταξινόμηση, η οποία θα ήταν μια ταξινόμηση συνάθροισης (σε αντιδιαστολή με μια ταξινόμηση γενίκευσης/ειδίκευσης, το πρώτο είδος ταξινόμησης που εξετάσαμε παραπάνω).

Ο πίνακας που ακολουθεί παρουσιάζει ένα τμήμα της πιο γνωστής ταξινόμησης που χρησιμοποιείται στο ηλεκτρονικό εμπόριο, σύμφωνα με το παγκόσμιο πρότυπο για την ταξινόμηση προϊόντων και υπηρεσιών (UNSPSC, <http://www.ecma.org>).

SEGMENT	FAMILY	CLASS	COMMODITY	TITLE
10	00	00	00	Live Plant and Animal Material and Accessories and Supplies
10	10	00	00	Live Animals
10	10	15	00	Livestock
10	10	15	01	Cats
10	10	15	02	Dogs

Αν και αυτή η ταξινόμηση επιδεικνύεται με το συνοπτικό αυτό σχήμα, μπορούμε να την παρουσιάσουμε με ένα δέντρο, όπως στο επόμενο σχήμα.



Οι ταξινόμησης είναι καλές για την ταξινόμηση των οντοτήτων μας. Είναι ένας τρόπος δόμησης και χαρακτηρισμού των μεταδεδομένων της πληροφορίας που μας ενδιαφέρει. Επειδή οι ταξινόμησης είναι δέντρα, μερικές φορές υπάρχουν περιττές πληροφορίες σε μια ταξινόμηση. Γιατί; Επειδή υπάρχει μόνο ένας κόμβος γονέων για

κάθε κόμβο παιδιών, μπορούμε μερικές φορές να έχουμε διπλούς κόμβους παιδιών κάτω από τους διαφορετικούς γονείς. Παραδείγματος χάρη, εάν είχαμε τις υποκλάσεις Manager και Employee που τοποθετήθηκαν κάτω από το Person, όπως στο προηγούμενο παράδειγμα, όλοι οι managers θα τοποθετούνταν κάτω και από τους δύο κόμβους, δεδομένου ότι είναι και Managers και Employees. Επομένως έχει σημασία ο τρόπος με τον οποίο η ταξινόμια είναι δομημένη. Οι οντολογίες χρησιμοποιούν τις ταξονομίες για τη δόμησή τους, αλλά προσθέτουν επιπλέον συσχετίσεις, ιδιότητες, τιμές ιδιοτήτων.

4.2. Χρησιμότητα των taxonomies

Ας αναλογιστούμε μια αναζήτηση στο διαδίκτυο. Χρησιμοποιούμε μια μηχανή αναζήτησης για να προσπαθήσουμε να βρούμε τα θέματα που μας ενδιαφέρουν, με βάση τις λέξεις κλειδιά. Μερικές φορές αναζητούμε προϊόντα και υπηρεσίες που θα επιθυμούμε να αγοράσουμε. Σε άλλες περιπτώσεις θα θέλαμε κάποιος άλλος να εντοπίσουν προϊόντα και υπηρεσίες που εμείς ως επιχείρηση παρέχουμε. Και στις δύο περιπτώσεις ωστόσο, αν ο ενδιαφερόμενος δεν εντοπίσει αυτό που του παρέχουμε, θα το αναζητήσει κάπου αλλού, με αποτέλεσμα, ακόμη και αν το δικό μας προϊόν είναι ακριβώς αυτό που ψάχνει, εφόσον δεν έχει κατορθώσει να το εντοπίσει, θα απευθυνθεί κάπου αλλού και τελικά δεν θα το αποκτήσει από εμάς.

Η πιο κοινή χρήση των ταξονομιών είναι η αναζήτηση πληροφοριών, ειδικά όταν έχουμε μόνο μια γενική ιδέα γι' αυτό που αναζητούμε. Οποιοσδήποτε έχει επισκεφθεί μία δημόσια βιβλιοθήκη έχει χρησιμοποιήσει το Dewey δεκαδικό σύστημα. Οι κορυφαίες κατηγορίες (οι ρίζες του δέντρου) του συστήματος (<http://www.loc.gov/cds/lcsh.html>) είναι 10 πολύ γενικές κατηγορίες των πιθανών θεμάτων των βιβλίων, με άλλα λόγια, 10 τρόποι να κατηγοριοποιήσουμε το περιεχόμενο της γνώσης, όπως ο ακόλουθος πίνακας παρουσιάζει.

CODE	DESCRIPTION
000	Generalities
100	Philosophy and psychology
200	Religion
300	Social sciences
400	Language
500	Natural sciences and mathematics
600	Technology (Applied sciences)
700	The arts
800	Literature and rhetoric
900	Geography and history

Κάθε μία από αυτές τις κατηγορίες ρίζας έχει πολύ λεπτομερέστερη επεξεργασία του περιεχομένου από κάτω της. Ο παρακάτω πίνακας παρουσιάζει ένα παράδειγμα: Κατηγορία 500, φυσικές επιστήμες και μαθηματικά.

CODE	DESCRIPTION	CODE	DESCRIPTION
500	Natural sciences and mathematics	550	Earth sciences
501	Philosophy and theory	551	Geology, hydrology, meteorology
502	Miscellany	552	Petrology
503	Dictionaries and encyclopedias	553	Economic geology
504	Not assigned or no longer used	554	Earth sciences of Europe
505	Serial publications	555	Earth sciences of Asia
506	Organizations and management	556	Earth sciences of Africa
507	Education, research, related topics	557	Earth sciences of North America
508	Natural history	558	Earth sciences of South America
509	Historical, areas, persons treatment	559	Earth sciences of other areas

510	Mathematics	560	Paleontology Paleozoology
511	General principles	561	Paleobotany
512	Algebra and number theory	562	Fossil invertebrates
513	Arithmetic	563	Fossil primitive phyla
514	Topology	564	Fossil Mollusca and Molluscoidea
515	Analysis	565	Other fossil invertebrates
516	Geometry	566	Fossil Vertebrata (Fossil Craniata)
517	Not assigned or no longer used	567	Fossil cold-blooded vertebrates
518	Not assigned or no longer used	568	Fossil Aves (Fossil birds)
519	Probabilities and applied mathematics	569	Fossil Mammalia
520	Astronomy and allied sciences	570	Life sciences
521	Celestial mechanics	571	Not assigned or no longer used
522	Techniques, equipment, materials	572	Human races
523	Specific celestial bodies and phenomena	573	Physical anthropology
524	Not assigned or no longer used	574	Biology
525	Earth (Astronomical geography)	575	Evolution and genetics
526	Mathematical geography	576	Microbiology
527	Celestial navigation	577	General nature of life
528	Ephemerides	578	Microscopy in biology
529	Chronology	579	Collection and preservation
530	Physics	580	Botanical sciences
531	Classical mechanics Solid mechanics	581	Botany

532	Fluid mechanics Liquid mechanics	582	Spermatophyta (Seed-bearing plants)
533	Gas mechanics	583	Dicotyledones
534	Sound and related vibrations	584	Monocotyledones
535	Light and parafotic phenomena	585	Gymnospermae (Pinophyta)
536	Heat	586	Cryptogamia (Seedless plants)
537	Electricity and electronics	587	Pteridophyta (Vascular cryptograms)
538	Magnetism	588	Bryophyta
539	Modern physics	589	Thallobionta and Prokaryotae
540	Chemistry and allied sciences	590	Zoological sciences
541	Physical and theoretical chemistry	591	Zoology
542	Techniques, equipment, materials	592	Invertebrates
543	Analytical chemistry	593	Protozoa, Echinodermata, related phyla
544	Qualitative analysis	594	Mollusca and Molluscoidea
545	Quantitative analysis	595	Other invertebrates
546	Inorganic chemistry	596	Vertebrata (Craniata, Vertebrates)
547	Organic chemistry	597	Cold-blooded vertebrates: Fishes
548	Crystallography	598	Aves (Birds)
549	Mineralogy	599	Mammalia (Mammals)

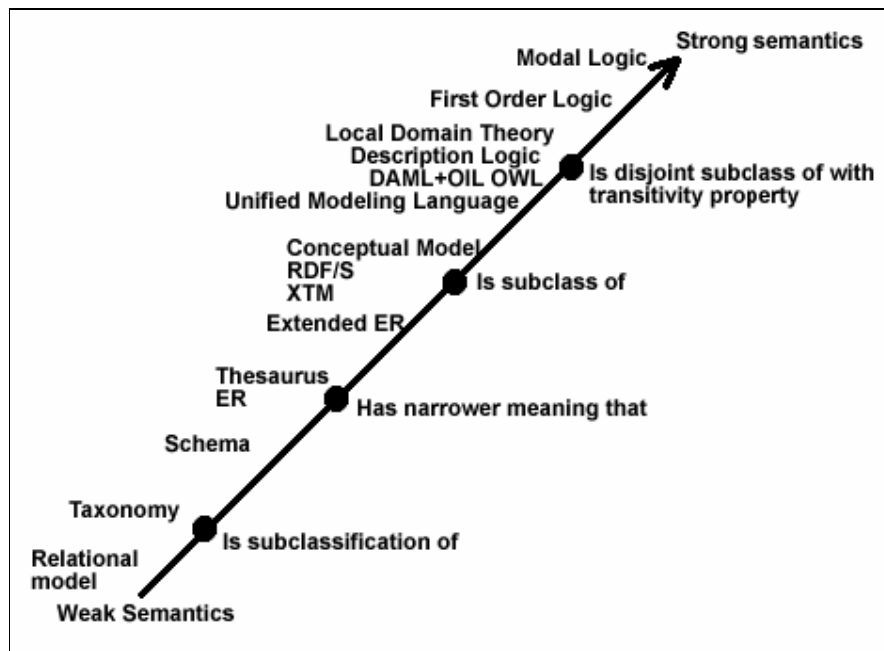
Εάν ψάχναμε ένα βιβλίο για δεινοσαύρους, θα κοιτάζαμε πιθανώς κάτω από την κατηγορία 567 ("απολιθωμένα cold-blooded σπονδυλωτά") εφόσον οι δεινόσαυροι είναι ερπετά ("cold-blooded"), ή ενδεχομένως κάτω από την κατηγορία 568 ("απολιθωμένα Aves: απολιθωμένα πουλιά") εφόσον οι δεινόσαυροι είναι πουλιά, ή ενδεχομένως κάτω από τη γενικότερη κατηγορία 566 ("απολιθωμένος σπόνδυλος: απολιθωμένο Craniata") εάν ξέραμε ότι οι δεινόσαυροι έχουν σπονδυλικές στήλες ή εάν ξέραμε ότι απολιθωμένο Craniata σήμαινε "τα ζώα που έχουν κρανία." Και εάν ξέραμε το τελευταίο, πιθανώς θα ξέραμε επίσης ότι τα ζώα που έχουν κρανία έχουν και σπονδυλική στήλη.

Όταν κοιτάζουμε τις κίτρινες σελίδες ενός τηλεφωνικού καταλόγου, βλέπουμε κάτω από τον τίτλο "Αυτοκίνητα" πολλούς υπότιτλους ή κατηγορίες: Εξαρτήματα, έμποροι αυτοκινήτων, ενοικιαζόμενα αυτοκίνητα, επισκευή αυτοκινήτων, και λοιπά. Αυτά αποτελούν μια απλή ταξινόμηση. Οι κίτρινες σελίδες είναι μια τεράστια ταξινόμηση. Διατάσσεται αλφαβητικά για να παρέχει βοήθεια σε κάποιον που ψάχνει προϊόντα ή υπηρεσίες, αλλά η αρχική λειτουργία της ως ταξινόμηση είναι η ταξινόμηση του διαθέσιμου περιεχομένου. Οι ταξινόμιες Yahoo και Google

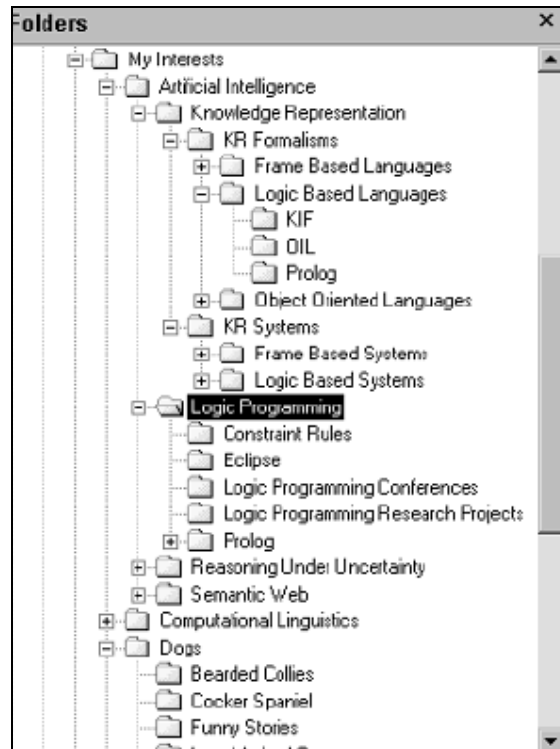
ενεργούν με τον ίδιο σχεδόν τρόπο: Βοηθούν έναν χρήστη που ψάχνει, ταξινομώντας το περιεχόμενο όσο το δυνατό πιο ρεαλιστικά.

4.3. Το φάσμα οντολογίας

4.3.1. Ορισμός



Το χαμηλό αριστερό σημείο του παραπάνω σχήματος υποδεικνύει μια ταξινόμηση. Σε μια ταξινόμηση, η σημασιολογία της σχέσης μεταξύ ενός γονέα και ενός κόμβου παιδιών είναι σχετικά απροσδιόριστη ή όχι πλήρως ορισμένη. Σε μερικές περιπτώσεις, πρόκειται για τη σχέση subclass of, ή την is part of ή σε άλλες περιπτώσεις είναι εντελώς απροσδιόριστη. Εάν εξετάσουμε τη δομή των αρχείων του υπολογιστή μας, η σχέση μεταξύ οποιουδήποτε δεδομένου καταλόγου και ενός από τα συγκεκριμένα subdirectories του είναι αυθαίρετη. Για παράδειγμα, έστω ότι δημιουργούμε ένα subdirectory για να κρατήσουμε ένα ειδικό υποσύνολο (μια υποκατηγορία) του αρχείου "έγγραφα". Δημιουργούμε λοιπόν ένα subdirectory για να τοποθετήσουμε τα έγγραφα που αντιπροσωπεύουν τις διασκέψεις και τα εργαστήρια, αφού εξετάσουμε το γενικό περιεχόμενο του συνολικού directory. Το επόμενο σχήμα επιδεικνύει ένα υποσύνολο ενός directory (Microsoft Windows 2000).



Παρατηρούμε ότι κάτω από το subdirectory Logic Programming είναι τοποθετημένα τα subdirectories των logic programming systems, (Prolog και Eclipse, υποκατηγορίες του Logic Programming), καθώς επίσης και τα subdirectories Logic Programming Conferences και Logic Programming Research Projects, τα οποία συσχετίζονται με το γενικότερο Directory.

Στη συνέχεια αναφέρουμε ως παράδειγμα το λεξικό, το οποίο περιλαμβάνει τις συσχετίσεις μεταξύ των όρων (λέξεις ή φράσεις) που αποτελούν μία ταξινόμηση. Κατά συνέπεια, ένα λεξικό είναι μία ταξινόμηση που περιλαμβάνει επίσης και τις σημασιολογικές σχέσεις των όρων.

Μία ταξινόμηση, όπως ένα λεξικό ή μια οντολογία, είναι ένας τρόπος δόμησης των στοιχείων μας και απόδοσης σε αυτά μιας απλής σημασιολογίας. Στον Ιστό, οι ταξινόμιες μπορούν να χρησιμοποιηθούν για να βοηθήσουν τους χρήστες να βρουν προϊόντα και υπηρεσίες. Οι ταξινόμιες μπορούν επίσης να μας βοηθήσουν να ταξινομήσουμε τα ενδιαφέροντά μας (εάν περιλαμβάνουν προϊόντα και υπηρεσίες). Επειδή οι ταξινόμιες αναφέρονται στην ταξινόμηση του περιεχομένου (σημασιολογία ή σημασία), επιτρέπουν στις μηχανές αναζήτησης και σε άλλες εφαρμογές που χρησιμοποιούν τις ταξινόμιες να βρουν άμεσα τις οντότητες πληροφοριών, πολύ γρηγορότερα και με πολύ μεγαλύτερη ακρίβεια.

4.3.2. Παράδειγμα: Thesaurus

Πρόκειται για ένα ελεγχόμενο λεξιλόγιο, το οποίο είναι οργανωμένο με μία ορισμένη τάξη και δομή έτσι ώστε να υφίστανται συγκεκριμένες συσχετίσεις, ιεραρχίας και ισοδυναμίας, μεταξύ των διαφόρων όρων που παρατίθενται σαφώς και προσδιορίζονται από συγκεκριμένους, τυποποιημένους δείκτες.

Ένα λεξικό έχει ως στόχο να διευκολυνθεί η ανάκτηση των εγγράφων και να επιτευχθεί η συνέπεια στην ευρετηρίαση των καταγεγραμμένων εγγράφων.

Οι συσχετίσεις μεταξύ των όρων μπορούν να ταξινομηθούν σε τέσσερις κατηγορίες:

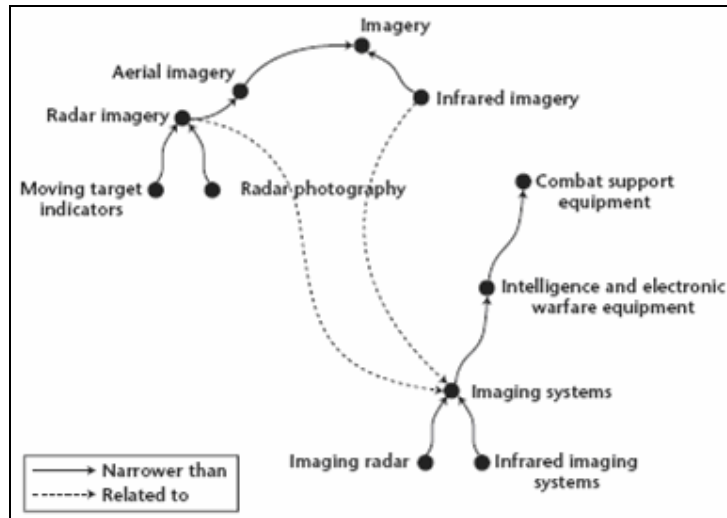
- Equivalence
- homographic
- Hierarchical
- Associative

Στον ακόλουθο πίνακα παρατίθενται αυτές οι συσχετίσεις και παρουσιάζονται και σχετικοί ορισμοί και παραδείγματα.

Semantic Relation	Definition	Example
<i>Synonym</i> Similar to Equivalent Used for	A term X has nearly the same meaning as a term Y	"Report" is a synonym for "document"
<i>Homonym</i> Spelled the same Homographic	A term X is spelled the same way as a term Y, which has a different meaning	The "tank", which is a military vehicle, is a homonym for the "tank", which is a receptacle for holding liquids.
<i>Broader than</i> (Hierarchic: parent of)	A term X is broader in meaning than a term Y	"Organisation" has a broader meaning than "financial institution"
<i>Narrower Than</i> (Hierarchic: child of)	A term X is narrower in meaning than a term Y	"Financial institution" has a narrower meaning than "organisation".
<i>Associated</i> (Associative Related)	A term X is associated with a term Y, i.e., there is some unspecified relationship between the two	A "nail" is associated with a "hammer".

Ένα λεξικό χρησιμοποιείται χαρακτηριστικά για να συνδέσει την έννοια ενός όρου στην έννοια ενός άλλου όρου.

Στο σχήμα που ακολουθεί, η έννοια "radar imagery" είναι στενότερη από την έννοια "aerial imagery" που είναι με τη σειρά της στενότερη από το "imagery" και σχετίζεται με την έννοια "imaging systems".



Κατά συνέπεια, ένα λεξικό είναι ελεγχόμενο λεξιλόγιο με σκοπό να υποστηρίξει την ανάκτηση πληροφοριών με την καθοδήγηση τόσο του προσώπου που ορίζουν τα μεταδεδομένα όσο και του ερευνητή που πραγματοποιεί την αναζήτηση των όρων, να επιλέξουν δηλαδή τους ίδιους όρους για την ίδια έννοια.

Για παράδειγμα, ένα λεξικό που προσαρμόζεται στον ISO 2788 υποστηρίζει τη πλοήγηση και την επιλογή όρων με την παρουσίαση σχέσεων μεταξύ των όρων που είναι κοντά στη σημασία. Ένα λεξικό εξασφαλίζει ότι:

- οι έννοιες περιγράφονται με έναν συνεπή τρόπο
- έμπειροι χρήστες είναι σε θέση να ορίσουν εύκολα τις αναζητήσεις τους για να εντοπίσουν τις πληροφορίες που χρειάζονται
- οι χρήστες δεν πρέπει απαραίτητα να είναι εξοικειωμένοι με την τεχνική ή τοπική ορολογία.

Αν και οι περισσότεροι άνθρωποι εξοικειώνονται με το Thesaurus³ Roget, ένας κοινός πόρος διαθέσιμος στους τεχνικούς είναι το λεξικό WordNet (WordNet: <http://www.cogsci.princeton.edu/~wn/>).

Το WordNet σχεδιάστηκε σύμφωνα με τις ψυχολinguιστικές θεωρίες της ανθρώπινης λεξικολογικής μνήμης. Στο WordNet οι όροι μπορούν να αναζητηθούν αμφίδρομα on-line ή να μεταφορτωθούν και να χρησιμοποιηθούν από τους

υπεύθυνους για την ανάπτυξη λογισμικού που επιθυμούν να ενσωματώσουν την thesaural γνώση στις εφαρμογές τους. Στο WordNet, σε μια λέξη δίνονται ορισμοί (οι ευδιάκριτοι ορισμοί για μια μεμονωμένη λέξη καλούνται συνήθως έννοιες της λέξης), καθώς μια δεδομένη λέξη, όπως η λέξη bank, έχει πολλαπλές έννοιες. Η ύπαρξη πολλών εννοιών μιας λέξης καλείται πολυσημία ("πολλαπλές έννοιες") στη γλωσσολογία. Μια λέξη (ή κοινή φράση) σε WordNet έχει επίσης τις πληροφορίες που σχετίζονται τυπικά με αυτήν, σύμφωνα με το ISO 2788.

- synonyms (συνώνυμα) - εκείνοι οι κόμβοι στην ταξονομία που συνδέονται με μία συσχέτιση "ίδιο με" (mean the same as) μέσα στην ταξονομία
- hypernyms - εκείνοι οι κόμβοι που συνδέονται με μία συσχέτιση "ευρύτερο από" (broader than) στην ταξονομία
- hyponyms - εκείνοι οι κόμβοι που συνδέονται με μία συσχέτιση "στενότερο από" (narrower than) στην ταξονομία

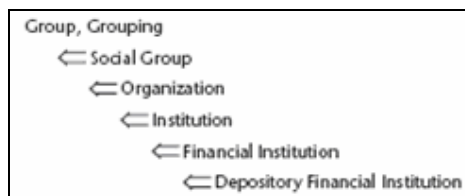
Στο WordNet, παραδείγματος χάριν, η λέξη "bank" συνδέεται με τις ακόλουθες έννοιες και hypernymic πληροφορίες:

<i>1. depository financial institution, bank, banking concern, banking company—(a financial institution that accepts deposits and channels the money into lending activities; "he cashed a check at the bank"; "that bank holds the mortgage on my home")</i>
<i>2. bank—(sloping land (especially the slope beside a body of water); "they pulled the canoe up on the bank"; "he sat on the bank of the river and watched the currents")</i>
<i>3. bank—(a supply or stock held in reserve for future use (especially in emergencies)</i>
<i>4. bank, bank building—(a building in which commercial banking is transacted; "the bank is on the corner of Nassau and Witherspoon")</i>
<i>5. bank—(an arrangement of similar objects in a row or in tiers; "he operated a bank of switches")</i>
<i>6. savings bank, coin bank, money box, bank—(a container (usually with a slot in the top) for keeping money at home; "the coin bank was empty")</i>

7. bank – (a long ridge or pile; “a huge bank of earth”)
8. bank – (the funds held by a gambling house or the dealer in some gambling games; “he tried to break the bank at Monte Carlo”)
9. bank, cant, camber – (a slope in the turn of a road or track; the outside is higher than the inside in order to reduce the effects of centrifugal force)
10. bank – (a flight maneuver; aircraft tips laterally about its longitudinal axis (especially in turning); “the plane went into a steep bank”)

Στο WordNet, οι διάφορες σχέσεις υφίστανται μεταξύ συνωνύμων, ή synsets. Αυτό σημαίνει ότι οι ευδιάκριτες λέξεις ή οι φράσεις, οι όροι, οι οποίοι είναι συνώνυμοι κατά προσέγγιση στο ίδιο επίπεδο, συγκεντρώνονται ως σύνολο. Το synset επομένως ενεργεί ως μία έννοια που στέκεται πίσω από τον όρο και αντιπροσωπεύει τη διανοητική κοινοποίηση του όρου. Έτσι, διακρίνουμε τον όρο από την έννοια στη γνωστική ή σημασιολογική αναπαράσταση. Ένας όρος είναι η αναπαράσταση της σημασίας που αυτός αντιπροσωπεύει μέσω μίας ετικέτας ή μιας σειράς χαρακτήρων. Η ελλοχέουσα σημασία είναι η έννοια και οι ιδιότητες, οι τιμές ιδιοτήτων, και οι συσχετίσεις με άλλες έννοιες.

Το επόμενο σχήμα επιδεικνύει την ταξονομική δομή για την πρώτη σημασία της λέξης “bank”.



Όπως μπορούμε να δούμε στη συνέχεια, οι πρώτες τρεις από τις δέκα συνολικά έννοιες της λέξης bank επιδεικνύονται, μαζί με το hypernymic καθεμιάς από αυτές. Κάθε => δείχνει τον άμεσο γονέα του προηγούμενου όρου (μια αύξηση στην εγκοπή χρησιμοποιείται επίσης για να δείξει την ενίσχυση ενός επιπέδου στην ταξονομική δομή).

Word Sense and Hypernymic Taxomic Representation

Sense 1: depository financial institution, bank, banking concern, banking company – (a financial institution that accepts deposits and channels the money into lending)

<p>activities; "he cashed a check at the bank"; "that bank holds the mortgage on my home")</p> <p>=> financial institution, financial organization, financial organisation – (an institution (public or private) that collects funds (from the public or other institutions) and invests them in financial assets)</p> <p>=> institution, establishment – (an organization founded and united for a specific purpose)</p> <p>=> organization, organisation – (a group of people who work together)</p> <p>=> social group – (people sharing some social relation)</p> <p>=> group, grouping – (any number of entities (members) considered as a unit)</p>
<p>Sense 2: bank – (sloping land (especially the slope beside a body of water); "they pulled the canoe up on the bank"; "he sat on the bank of the river and watched the currents")</p> <p>=> slope, incline, side – (an elevated geological formation; "he climbed the steep slope"; "the house was built on the side of the mountain")</p> <p>=> geological formation, geology, formation – (the geological features of the earth)</p> <p>=> natural object – (an object occurring naturally; not made by man)</p> <p>=> object, physical object – (a tangible and visible entity; an entity that can cast a shadow; "it was full of rackets, balls, and other objects")</p> <p>=> entity, physical thing – (that which is perceived or known or inferred to have its own physical existence (living or nonliving))</p>
<p>Sense 3: bank – (a supply or stock held in reserve for future use (especially in emergencies))</p>

Παραδείγματος χάρη, ας εξετάσουμε την πρώτη έννοια της λέξης αυτής: η έννοια της τράπεζας ως χώρος οικονομικών καταθέσεων έχει αμέσως ως γονέα τον οικονομικό οργανισμό, ο οποίος έχει στη συνέχεια ως γονέα τον οργανισμό, και τα λοιπά. Ένας άλλος τρόπος να εξετάσει κανείς αυτήν την ταξονομική δομή είναι με τη "ρίζα" του ταξονομικού δέντρου στην κορυφή, και σημειώνουμε ότι έχουμε αντιστρέψει το βέλος για να δείξει ότι ο γονέας είναι επάνω από το παιδί.

Το επόμενο βήμα προς την αύξηση της σημασιολογικής αφθονίας είναι το εννοιολογικό πρότυπο. Καθώς κινούμαστε από τις ταξονομίες και τα λεξικά προς τα εννοιολογικά πρότυπα και τις λογικές θεωρίες, είμαστε όλο και περισσότερο στη σφαίρα των οντολογιών. Αν και το πλαίσιο μας καλείται φάσμα οντολογίας, για να δείξει ότι ολόκληρο το οντολογικό ή σημασιολογικό πλαίσιο περιλαμβάνει μια σειρά δυνατοτήτων σε προοδευτική σειρά, το πάνω δεξί μισό αντιπροσωπεύει καταλληλότερα την κοινή έννοια της οντολογίας.

Τι είναι ένα εννοιολογικό πρότυπο; Ένα εννοιολογικό πρότυπο είναι ένα πρότυπο μιας θεματικής περιοχής ή ενός γνωστικού τομέα, που αντιπροσωπεύει τις βασικές οντότητες του τομέα, τις σχέσεις μεταξύ των οντοτήτων, τις ιδιότητες και τις τιμές

ιδιοτήτων, των οντοτήτων και τις σχέσεις, και μερικές φορές κανόνες που συνδέουν τις οντότητες, σχέσεις, και τις ιδιότητες με τους πιο περίπλοκους τρόπους.

Σε ένα εννοιολογικό πρότυπο, είναι δυνατό να καθοριστεί και να εκφραστεί η σχέση «υποκατηγορίας» μεταξύ μιας κατηγορίας γονέων και μιας κατηγορίας παιδιών. Οι αντικειμενοστρεφείς γλώσσες προγραμματισμού, όπως η UML, είναι αρκετά πλούσιες να εκφράσουν τη σημασιολογία της σχέσης υποκατηγορίας μεταξύ δύο δεδομένων κλάσεων.

Επίσης, σημαντικό είναι οι ορισμοί μιας κλάσης, superclass, και μιας subclass να διευκρινίζονται καλά σημασιολογικά στο επίπεδο μεταδεδομένων, έτσι ώστε οι κλάσεις σε επίπεδο αντικειμένων, όπως τα Person και Employee, να μπορούν να διευκρινιστούν καλά σημασιολογικά.

Το επίπεδο αντικειμένων μας ενδιαφέρει, καθώς είναι το επίπεδο στο οποίο κατασκευάζουμε τα πρότυπά μας για τα διάφορα συστήματα και τους διάφορους τομείς. Το επίπεδο μεταδεδομένων είναι το επίπεδο που καθορίζει τις κλάσεις, τις σχέσεις και τις ιδιότητες που θα χρησιμοποιήσουμε στο επίπεδο αντικειμένων για να καθορίσουμε τα πρότυπά μας.

Το επίπεδο μεταδεδομένων είναι συχνά το επίπεδο όπου η εννοιολογική γλώσσα διαμόρφωσης (όπως η UML) καθορίζεται. Η έννοια του επιπέδου meta και του επιπέδου αντικειμένων μπορεί να συγχέονται, αλλά θα γίνουν πιο εκτενείς αναφορές όταν εξετάσουμε τις οντολογίες.

Το μοντέλο οντοτήτων - συσχετίσεων (ER) που χρησιμοποιείται για να καθορίσουμε ένα εννοιολογικό σχήμα για μια βάση δεδομένων θεωρείται επίσης εννοιολογική γλώσσα διαμόρφωσης. Όταν κάποιος σχεδιάζει μια βάση δεδομένων, πρώτα δημιουργεί ένα εννοιολογικό σχήμα, το οποίο μειώνει σε ένα λογικό σχήμα, και αυτό πάλι το μειώνει τελικά σε ένα φυσικό σχήμα. Αυτά τα σχήματα αντιπροσωπεύουν τα επίπεδα αφαίρεσης: από το ανθρώπινο εννοιολογικό επίπεδο στη βάση δεδομένων με γραμμές και στήλες και τελικά στους πραγματικούς πίνακες, τις στήλες και τα κλειδιά.

4.4. Οντολογίες και SW

4.4.1. Τι είναι οντολογία

Ορισμοί

Ακριβώς τι είναι μια οντολογία; [7] Μια οντολογία καθορίζει τις κοινές λέξεις και τις έννοιες που χρησιμοποιούνται για να περιγράψουν και να αναπαραστήσουν έναν τομέα γνώσης. Εάν αναζητήσουμε τον ορισμό της λέξης οντολογία στο λεξικό, θα βρούμε τον ακόλουθο (online: <http://www.m-w.com/>):

1. Ένας κλάδος της μεταφυσικής σχετικά με τη φύση και τις σχέσεις της ύπαρξης
2. Μια ιδιαίτερη θεωρία για τη φύση της ύπαρξης ή τα είδη των υπάρξεων

Αυτός ορισμός δείχνει ότι ο όρος έχει τις ρίζες του στη φιλοσοφία - συγκεκριμένα, ένα μέρος της μεταφυσικής που είναι η συστηματική μελέτη των αρχών που κρύβονται κάτω από ένα ιδιαίτερο θέμα, συχνότερα η φύση της ύπαρξης και η φύση της εμπειρίας.

Η ΤΠ προσφέρει τους ακόλουθους ορισμούς.

- Μια οντολογία καθορίζει τις κοινές λέξεις και έννοιες που χρησιμοποιούνται για να περιγράψουν και να αναπαραστήσουν έναν τομέα γνώσης
- Μια οντολογία είναι ένα προϊόν εφαρμοσμένης μηχανικής που αποτελείται από "ένα συγκεκριμένο λεξιλόγιο που χρησιμοποιείται για να περιγράψει την πραγματικότητα, συν ένα σύνολο ρητών υποθέσεων σχετικά με την προοριζόμενη σημασία αυτού του λεξιλογίου".

Ένας τομέας είναι μια περιοχή γνώσης. Μερικά παραδείγματα τέτοιων τομέων γνώσης είναι: ιατρική, επισκευή αυτοκινήτων, οικονομικός σχεδιασμός, σχεδίαση μηχανών, επιχειρησιακή διαχείριση, φυσική, κλωστοϋφαντουργικά προϊόντα. Η περιγραφή ενός τομέα της γνώσης είναι η πράξη της έκφρασης, είτε σε γραπτό είτε σε προφορικό λόγο, των σημαντικών σημείων για έναν συγκεκριμένο τομέα γνώσης. Παραδείγματος χάριν, στην περιγραφή της επισκευής αυτοκινήτων, θα μιλούσαμε πιθανώς για τα εξής:

- τα είδη αυτοκινήτων (sedan, stationwagons, σπορ αυτοκίνητα, αυτοκίνητα πολυτελείας)

- οι τύποι μηχανών (που αντιστοιχούν ίσως στους τύπους καυσίμων: βενζίνη, diesel, electric-powered, hybrid)
- συγκεκριμένες μηχανές (παραδείγματος χάριν, μια 1995-96 V-6 Ford Taurus 244/4.0 Aerostar Automatic με Block Casting # 95TM-AB, Head Casting 95TM)
- οι κατασκευαστές (Ford, General Motors, Chevrolet, Nissan, Honda, VOLVO, VOLKSWAGEN, Saab, Hyundai και λοιπά)
- τα μέρη των αυτοκινήτων (μηχανές, συστήματα φρένων, συστήματα ψύξης, ηλεκτρικά συστήματα, αναστολή, σώμα και λοιπά) και τις ιδιότητές τους (μια μηχανή έχει 4 ..6 ..8, ή 12 κυλίνδρους, τα φρένα έχουν διαφορετικές συνθέσεις όπως το semimetallic ή μη σιδηρούχο υλικό)

Κατά την περιγραφή ενός τομέα γνώσης, περιγράφουμε τα σημαντικά στοιχεία του τομέα, τις ιδιότητές τους, και τις σχέσεις μεταξύ τους. Εάν επρόκειτο να διαμορφώσουμε την περιγραφή μας (επειδή, για παράδειγμα, γράφουμε ένα βιβλίο για την επισκευή αυτοκινήτων), μπορούμε ακόμη και να περιλάβουμε κανόνες για τον συγκεκριμένο τομέα, όπως ο ακόλουθος κανόνας διαγνώσεων, ο οποίος διευκρινίζει πώς να καθορίσει κανείς αυτό που κάνει λάθος σε ένα αυτοκινητικό σύστημα προκειμένου να το επισκευάσει: Εάν το αυτοκίνητο δεν παίρνει μπρος, πρέπει να ελέγξει και να καθαρίσει τις συνδέσεις μπαταριών.

Επομένως, μια περιγραφή είναι ή μπορεί να είναι μια οντολογία. Περιλαμβάνει τα εξής είδη εννοιών:

- κλάσεις (γενικά πράγματα) σχετικά με το αντικείμενο που μας ενδιαφέρει
- οι σχέσεις μεταξύ εκείνων των κλάσεων
- οι ιδιότητες (και τιμές ιδιοτήτων)
- οι λειτουργίες και οι διαδικασίες που σχετίζονται με τις κλάσεις
- οι περιορισμοί και οι κανόνες που αναφέρονται στις κλάσεις

Εκτός από την απλή περιγραφή ενός τομέα γνώσης, πρέπει επίσης να αναπαραστήσουμε αυτή την περιγραφή, το οποίο σημαίνει ότι κωδικοποιούμε την περιγραφή με έναν τρόπο που επιτρέπει σε κάποιον τρίτο να τη χρησιμοποιήσει.

Μια περιγραφή αποτελείται από λέξεις και φράσεις σε μια φυσική γλώσσα, δηλαδή λεξιλόγιο/ορολογία και προτάσεις που συνδυάζουν τις ορολογίες για να εκφράσουν τις σχέσεις μεταξύ των όρων (θα χρησιμοποιήσουμε το “λεξιλόγιο” και την “ορολογία” εδώ ως ισοδύναμα). Έτσι, αναπαράσταση σημαίνει ότι απεικονίζουμε

την περιγραφή χρησιμοποιώντας όρους και προτάσεις. Καθορίζουμε τους όρους και έπειτα συνδυάζουμε εκείνους τους καθορισμένους όρους με τέτοιο τρόπο που διαμορφώνουν τη σημασία του τομέα γνώσης στον οποίο αναφέρονται.

Στην τεχνολογία πληροφοριών, εντούτοις, χρησιμοποιούμε μια ελαφρώς πιο περίπλοκη έννοια: αναπαριστούμε, προκειμένου να χρησιμοποιηθεί η περιγραφή στην τεχνολογία πληροφοριών, δηλαδή δημιουργούμε ένα πρότυπο ώστε το λογισμικό να είναι σε θέση να το χρησιμοποιήσει. Αναπαριστούμε τις κλάσεις, τις σχέσεις, τις ιδιότητες και τους κανόνες για τον εκάστοτε τομέα γνώσης. Χρησιμοποιούμε τους όρους της φυσικής γλώσσας περιγραφής, όπως ετικέτες, για τις βασικές έννοιες - που είναι, η έννοια του τομέα της γνώσης που αποτελείται από τις κλάσεις, τις ιδιότητες, και τις σχέσεις. Χαρακτηριστικά, αναπαριστούμε ή κωδικοποιούμε την οντολογία σε μια γλώσσα λογικής, αντιπροσώπευσης της γνώσης παρά σε μια φυσική γλώσσα, επειδή θέλουμε να αναπαραστήσουμε την περιγραφή μας όσο το δυνατόν πιο σαφώς. Η φυσική γλώσσα μπορεί να είναι πολύ διαφορούμενη.

Θέλουμε επίσης να καταστήσουμε την έννοιά της διαθέσιμη για να κάνουμε χρήση της τεχνολογίας πληροφοριών. Η αναπαράσταση πρέπει λοιπόν αναφέρεται στον προσδιορισμό των όρων (το λεξιλόγιο που ενεργεί όπως οι ετικέτες για τις έννοιες), των εννοιών και των σχέσεών τους.

4.4.2. Η σύνταξη, η δομή, η σημασιολογία

Στο σημείο αυτό καλούμαστε να καλύψουμε γνωστικά τα εξής [9]:

- τα κρίσιμα ζητήματα στη δημιουργία των οντολογιών: αυτό που καθιστά μια οντολογία καλύτερη από άλλη, και ποια χαρακτηριστικά γνωρίσματα παρέχουν οι οντολογίες και πώς τα παρέχουν - ενώ άλλα πρότυπα στο φάσμα οντολογίας δεν μπορούν
- η σημασία των οντολογιών από την πλευρά ενός διευθυντή τεχνολογίας πληροφοριών που πρέπει να εξετάσει τις νέες τεχνολογίες σημασιολογικού Ιστού για την ενσωμάτωση στα συστήματα και τις πρακτικές της υποδομής της επιχείρησής του και τον αντίκτυπό τους στις στρατηγικές πληροφοριών για το μέλλον.

4.4.2.1 Σύνταξη

Η σύνταξη προσδιορίζεται συνήθως από τη μορφή, το σχήμα και τη δομή, και έτσι θα την απεικονίσουμε εδώ. Κάθε γλώσσα έχει μια σύνταξη και μια σημασιολογία: COBOL, FORTRAN, C, Java, αγγλικά, κινέζικα, SQL, XML, RDF, και OWL. Μια γλώσσα μπορεί να θεωρηθεί το επίσημο σύστημα που έχει ένα αλφάβητο ή ένα λεξιλόγιο καθορισμένο (ή και τα δύο), ένα σύνολο κανόνων για πώς το αλφάβητο και το λεξιλόγιο μπορούν να συνδυαστούν σε προτάσεις, και μια σημασιολογία για το αλφάβητο, το λεξιλόγιο, και τις δηλώσεις/τις προτάσεις.

Η γλώσσα προγραμματισμού, ακριβώς όπως μια φυσική γλώσσα όπως τα αγγλικά, έχει μια επίσημη σύνταξη. Ας αναλογιστούμε τις γλώσσες προγραμματισμού C ή Java. Όταν μαθαίνουμε C, ένα μεγάλο μέρος αυτού που μαθαίνουμε είναι η σύνταξη της γλώσσας προγραμματισμού C - για παράδειγμα, η δήλωση `#include` προηγείται οποιωνδήποτε άλλων δηλώσεων, ένα `for loop` έχει έναν ορισμένο αριθμό συστατικών, τα οποία πρέπει να διατάσσονται σε μία συγκεκριμένη σειρά. Επιπλέον, τα συστατικά αυτά πρέπει να υπακούουν σε ορισμένους απλούς περιορισμούς, όπως για παράδειγμα, εάν μια μεταβλητή δηλώνεται να είναι τύπου ακέραιου αριθμού, κατόπιν η μόνη πιθανή τιμή για εκείνη την μεταβλητή είναι ένας ακέραιος αριθμός. Έτσι, η σύνταξη αφορά στη σειρά και στη μορφή.

Εάν το πρόγραμμα είναι συντακτικά σωστό, κατόπιν ο μεταγλωττιστής - που αναλύει αρχικά και επιβεβαιώνει τη σύνταξη - δεν θα παράγει μηνύματα προειδοποίησης ή λάθους. Εφόσον όλα τα συντακτικά λάθη (λάθη στη μορφοποίηση, τη σειρά, και την απλή δακτυλογράφηση) διορθωθούν, κατόπιν ο μεταγλωττιστής θα συνεχίσει το υπόλοιπο της εργασίας του, δημιουργώντας τους πίνακες συμβόλων, τους μετασχηματισμούς και τις βελτιστοποιήσεις και θα αρχίσει να εξετάζει τη στοιχειώδη σημασιολογία του προγράμματος.

Ο μεταγλωττιστής δεν ξέρει πώς σκοπεύουμε εμείς να διαμορφώσουμε την ιδιαίτερη λογική ροή του προγράμματος, και για το λόγο αυτό αυτά τα είδη σημασιολογικών λαθών προγραμματισμού καλούνται μερικές φορές λογικά λάθη. Μέχρι σήμερα, είναι απαραίτητος ο προγραμματιστής ή ένας εξαιρετικά έξυπνος μεταγλωττιστής (ίσως οδηγούμενος από ένα έμπειρο σύστημα που χρησιμοποιεί τη γνώση) για να ανιχνεύσει και να διορθώσει τα σημασιολογικά λάθη, επειδή, γενικά, οι υπολογιστές και το λογισμικό τους δεν είναι σε θέση για κάτι τέτοιο.

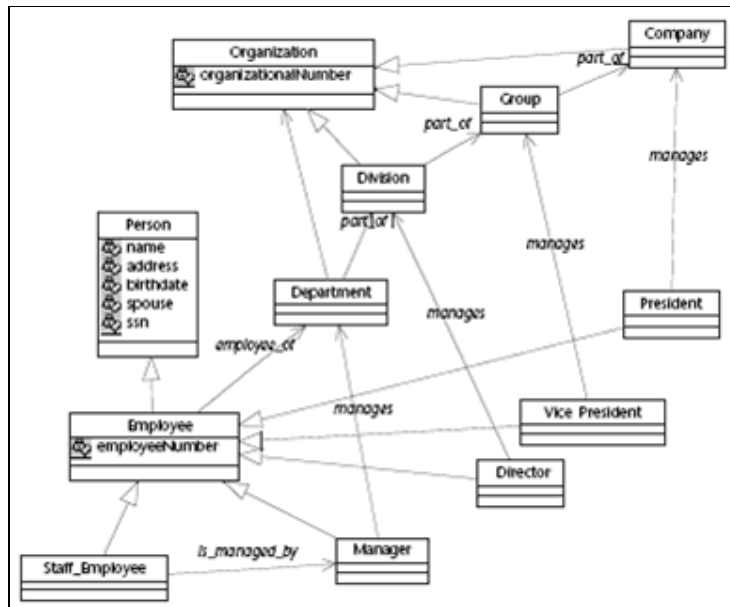
Στον κόσμο του Ιστού, εξετάζουμε χαρακτηριστικά τα έγγραφα, και επομένως η σύνταξη των εγγράφων, μαζί με τις γλώσσες σήμανσης που δομούν τα έγγραφα και τα στοιχεία, αποτελούν το αρχικό ενδιαφέρον μας. Η σύνταξη των εγγράφων περιλαμβάνει τις σειρές των χαρακτήρων από κάποιο αλφάβητο (για το κείμενο) ή κάποιο σύνολο καθορισμένων δυαδικών κωδικοποιήσεων (για τη γραφική παράσταση, το βίντεο, και τα λουπά). Τα συντακτικά σύμβολα όμως δεν έχουν νόημα εκτός αν τους δίνεται μια σημασιολογική ερμηνεία (με άλλα λόγια, μία αντιστοίχιση με τα αντικείμενα ενός προτύπου όπου η σημασία τους αναπαρίσταται). Η σημασιολογική ερμηνεία είναι η σημασιολογία: Ερμηνεύει τα συντακτικά σύμβολα όσον αφορά στην προοριζόμενη έννοιά τους. Η XML για παράδειγμα έχει μια σύνταξη. Ένα έγγραφο που είναι χαρακτηρισμένο σε XML είναι είτε σωστό είτε όχι συντακτικά, όσον αφορά στη σύνταξη XML. Αυτό σημαίνει ότι ορισμένα στοιχεία πρέπει να εμφανιστούν σε μια ορισμένη σειρά, οι ετικέτες XML πρέπει οριοθετηθούν και λουπά.

4.4.2.2 Δομή

Η σύνταξη αφορά στη σειρά και τη μορφή, αλλά επίσης και στη δομή. Οι βάσεις δεδομένων, τα αντικείμενα του Ιστού, τα αντικείμενα στον επερχόμενο σημασιολογικό Ιστό, τα πρότυπα, και οι οντολογίες απαιτούν δομή. Τα πρότυπα απαιτούν γενικά δομή, έναν τρόπο οργάνωσης των στοιχείων του προτύπου. Ένα σχήμα βάσεων δεδομένων, παραδείγματος χάριν, είναι πρώτιστα ένας τρόπος για να περιγραφεί και να οριστεί η δομή μιας βάσης δεδομένων.

Τα στοιχεία της βάσης δεδομένων- πίνακες, στήλες, σειρές, και τιμές - απαιτούμε να ακολουθούν τη δομή του σχήματος, τον τρόπο που πρέπει να οργανώνονται και ορισμένα από αυτά να περιλαμβάνονται μέσα σε άλλα. Αυτή η προδιαγραφή επιβάλλεται χαρακτηριστικά από το σύστημα διαχείρισης βάσεων δεδομένων (ΠΔΒΔ (πρόγραμμα διαχείρισης βάσεων δεδομένων)), το οποίο μπορούμε να σκεφτούμε, τουλάχιστον μερικώς, ως compiler ή validator των στοιχείων όσον αφορά στο σχήμα βάσεων δεδομένων. Το ΠΔΒΔ κάνει πολύ περισσότερα από αυτό, εντούτοις. Αναλύει τα αρχεία XML (έλεγχει αν ακολουθείται η συντακτική προδιαγραφή της δεδομένης έκδοσης XML) και έπειτα τα επικυρώνει σύμφωνα με ένα συγκεκριμένο σχήμα, βασισμένο στη γλώσσα σχημάτων XML.

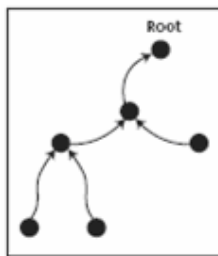
Τα εννοιολογικά πρότυπα, όπως εκείνα που γράφονται σε UML, αναφέρονται επίσης στη δομή. Η δομή στα εννοιολογικά πρότυπα απεικονίζεται μερικώς στις ιεραρχίες κληρονομιάς της σχέσης *subclass*: μια κλάση αποτελεί υποκλάση μιας άλλης κλάσης. Η δομή απεικονίζεται επίσης στη σχέση *part of*: μια κλάση είναι μέρος μιας άλλης. Η δομή απεικονίζεται επίσης σε άλλες αυθαίρετες σχέσεις. Σε ένα πρότυπο UML μιας εφαρμογής του ανθρώπινου δυναμικού, παραδείγματος χάριν, δύο κλάσεις μπορεί να συσχετίζονται μέσω της σχέσης *employee of* (παρόμοια με τη σχέση σε ένα εννοιολογικό σχήμα βάσεων δεδομένων, το οποίο κατασκευάζεται συνήθως σύμφωνα με ένα μοντέλο οντοτήτων-συσχετίσεων), όπως στο ακόλουθο σχήμα.



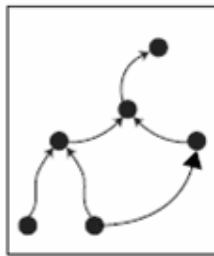
Η δομή μπορεί χαρακτηριστικά να αντιπροσωπευθεί από μια αναπαράσταση, χρησιμοποιώντας έναν κόμβο και μια άκρη ή μία σύνδεση (τα τελευταία μπορεί να είναι κατευθυνόμενα (ένα βέλος δείχνει στον κόμβο όπου κατευθύνεται η σχέση)). Η γενική μελέτη τέτοιων προτύπων node-and-end καλείται θεωρία γραφικών παραστάσεων, όπου μια γραφική παράσταση είναι μια πιο περίπλοκη δομή δεδομένων από ένα δέντρο, που είναι μια απλούστερη ιεραρχική δομή. Ένας γράφος είναι πιο περίπλοκος από ένα δέντρο, καθώς πρόκειται για ένα δέντρο με κατευθυνόμενες ή μη συνδέσεις, που ενώνουν αυθαίρετα τους κόμβους, ενώ ένα δέντρο είναι μια δομή δεδομένων που έχει απλά άκρα ή συνδέσεις (κλάδοι), έναν διακεκριμένο κόμβο αποκαλούμενο ρίζα, και από τη ρίζα υπάρχει μια μοναδική πορεία προς κάθε κόμβο.

Η κύρια διαφορά μεταξύ μιας γραφικής παράστασης και ενός δέντρου είναι ότι μια γραφική παράσταση μπορεί να έχει πολλαπλές πορείες προς τους κόμβους. Μια

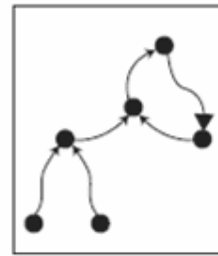
κατευθυνόμενη γραφική παράσταση είναι μια γραφική παράσταση στην οποία η άκρη κατευθύνεται από έναν κόμβο σε άλλο (σκεφτείτε μια σχέση όπως *father-of*, όπου η άκρη από τον John στο Harry δηλώνει ότι ο John είναι πατέρας του Harry). Μια μη κατευθυνόμενη γραφική παράσταση σημαίνει ότι δεν υπάρχει κανένα βέλος, αλλά μόνο μια απλούστερη άκρη (σκεφτείτε τη σχέση *friend-of* μεταξύ John και της Sue: ο John είναι φίλος της Sue και η Sue είναι φίλη του John). Μια γραφική παράσταση χωρίς κύκλους (συνδέσεις μεταξύ ενός κόμβου παιδιών και ενός από τους προγόνους του) καλείται ακυκλική γραφική παράσταση. Μια γραφική παράσταση με κύκλους (συνδέσεις μεταξύ ενός κόμβου παιδιών και ενός από τους γονείς) καλείται κυκλική γραφική παράσταση. Οι κατευθυνόμενες γραφικές παραστάσεις με κύκλους καλούνται κατευθυνόμενες κυκλικές γραφικές παραστάσεις, ενώ οι κατευθυνόμενες γραφικές παραστάσεις χωρίς κύκλους καλούνται κατευθυνόμενες ακυκλικές γραφικές παραστάσεις (DAGs), και είναι η δομή δεδομένων που χρησιμοποιείται για τις περισσότερες σύνθετες δομές, όπως τα ER, UML, και τα πρότυπα οντολογίας.



Δέντρο



Κατευθυνόμενος ακυκλικός γράφος



Κατευθυνόμενος κυκλικός γράφος

Η δομή η ίδια, αν και σημαντική, δεν είναι κρίσιμα καθοριστική ή ο χαρακτηριστικός παράγοντας για τα πρότυπα. Η σημασιολογική ερμηνεία όμως είναι. Η δομή είναι μια πλευρά του βαθμού σημασιολογικής ερμηνείας που απαιτείται. Η γνώση (όπως κωδικοποιείται στις οντολογίες, παραδείγματος χάριν) είναι η σχετικά σύνθετη συμβολική αναπαράσταση κάποιας πτυχής του κόσμου (ό,τι καλούμε θεματικές περιοχές, τομείς, και ότι επεκτείνεται πέρα από αυτά).

4.4.2.3 Σημασιολογία

Η σημασιολογική ερμηνεία είναι η χαρτογράφηση μεταξύ κάποιου δομημένου υποσυνόλου στοιχείων και ενός προτύπου κάποιου συνόλου αντικειμένων σε έναν

τομέα γνώσης, όσον αφορά στην προοριζόμενη σημασία εκείνων των αντικειμένων και στις σχέσεις μεταξύ εκείνων των αντικειμένων.

Χαρακτηριστικά, το πρότυπο βρίσκεται στο μυαλό του ανθρώπου. Ως άνθρωποι "καταλαβαίνουμε" τη σημασιολογία, το οποίο σημαίνει ότι αντιπροσωπεύουμε συμβολικά τον κόσμο, τα αντικείμενα του κόσμου, και τις σχέσεις μεταξύ εκείνων των αντικειμένων. Έχουμε τη σημασιολογία του κόσμου υπόψη μας, δομημένη και ερμηνευμένη. Όταν βλέπουμε ένα κείμενο, βλέπουμε τα σύμβολα σε μια σελίδα και τα ερμηνεύουμε όσον αφορά στο τι σημαίνουν σύμφωνα με το διανοητικό πρότυπό μας, δηλαδή παρέχουμε τη σημασιολογία (σημασία). Εάν επιθυμούμε να βοηθήσουμε στη διάδοση της γνώσης που ενσωματώνεται σε ένα έγγραφο, καθιστούμε εκείνο το έγγραφο διαθέσιμο σε τρίτους, αναμένοντας από αυτούς ότι με τη βοήθεια του δικού τους σημασιολογικού διερμηνέα (τα διανοητικά πρότυπά τους) θα κατανοήσουν τα σύμβολα στις σελίδες των εγγράφων. Έτσι, δεν υπάρχει καμία γνώση σε ένα έγγραφο αν δεν ερμηνεύεται η σημασιολογία του. Αν επιθυμούμε, εντούτοις, ο υπολογιστής να συμβάλλει στη διάδοση της γνώσης που ενσωματώνεται σε ένα έγγραφο - σύμφωνα με τον σημασιολογικό Ιστό - είναι ανάγκη τουλάχιστον να αυτοματοποιηθεί η διαδικασία σημασιολογικής ερμηνείας. Πρέπει να περιγράψουμε και να αναπαραστήσουμε με έναν τρόπο κατανοητό στον υπολογιστή τα διανοητικά μας πρότυπα, για συγκεκριμένους τομείς γνώσης. Οι οντολογίες παρέχουν σε μας αυτή τη δυνατότητα. Αυτό είναι ένα μεγάλο μέρος αυτού που ο σημασιολογικός Ιστός αποτελεί. Το λογισμικό του μέλλοντος (συμπεριλαμβανομένων των ευφών πρακτόρων, των υπηρεσιών Ιστού, και τα λοιπά) θα είναι σε θέση να χρησιμοποιήσει τη γνώση που κωδικοποιείται στις οντολογίες, που ερμηνεύουν σημασιολογικά τα έγγραφα του Ιστού και τα αντικείμενά μας.

Οι οντολογίες προσπαθούν να περιορίσουν τα πιθανά επίσημα πρότυπα της ερμηνείας (σημασιολογίας) των λεξιλογίων σε ένα σύνολο εννοιών που εμείς ορίζουμε. Κανένα άλλο πρότυπο με περιορισμένη σημασιολογία - ταξονομίες, σχήματα βάσεων δεδομένων, λεξικά - δεν κάνουν κάτι τέτοιο, αλλά υποθέτουν ότι οι άνθρωποι θα εξετάσουν τα "λεξιλόγια" και θα παράσχουν μαγικά τη σημασιολογία μέσω του ενσωματωμένου ανθρώπινου σημασιολογικού διερμηνέα: του μυαλού τους που χρησιμοποιεί τα διανοητικά του πρότυπα.

Με τις οντολογίες σκοπός είναι να μετατοπιστεί το "σημασιολογικό ερμηνευτικό φορτίο" στον υπολογιστή, ο οποίος τελικά θα φτάσει στο επίπεδο του ανθρώπινου χρήστη, ως προς τη σημασιολογία, και δεν θα είναι ο άνθρωπος εκείνος που θα

πρέπει να κατέβει στο επίπεδο του υπολογιστή. Γι' αυτό, για παράδειγμα, δεν προγραμματίζουμε ακόμα σε assembly. Η τεχνολογία λογισμικού και η πληροφορική έχουν εξελίξει τις υψηλότερες επιπέδους γλώσσες, που πολύ περισσότερο ευθυγραμμίζονται με το ανθρώπινο σημασιολογικό/εννοιολογικό επίπεδο. Οι οντολογιστές θέλουν να προχωρήσουν ακόμη περισσότερο.

Με την έκφραση *σημασιολογική ερμηνεία από τον υπολογιστή*, εννοούμε ότι με τη δόμηση (και τους περιορισμούς) που θέτουμε στα σύμβολα που χρησιμοποιούμε, σε μια λογική, αξιωματική γλώσσα (δηλ. μια γλώσσα αναπαράστασης γνώσης), ο υπολογιστής θα συμπεράνει μέσω μιας διαδικασίας (που και πάλι κατασκευάζεται από τον άνθρωπο σύμφωνα με λογικές αρχές) κατά προσέγγιση ό,τι ένας άνθρωπος θα μπορούσε να συμπεράνει σε αντίστοιχες περιστάσεις.

Αυτό σημαίνει ότι με δεδομένο ένα επίσημο λεξιλόγιο-αλφάβητο, τους όρους/τα σύμβολα (λογικά και μη), και τις δηλώσεις/τις εκφράσεις (και, φυσικά, τους κανόνες με βάση τους οποίους διαμορφώνει κανείς εκφράσεις από όρους) – θα θέλαμε τα επίσημα πρότυπα ερμηνείας που συσχετίζονται με τα σύμβολα και τις εκφράσεις (δηλ. η σημασιολογία) να προσεγγίσουν εκείνα τα πρότυπα που ο ίδιος ο άνθρωπος θα προσδιόριζε μόνος του (δηλ. κοντά στην ανθρώπινη σύλληψη εκείνου του τομέα γνώσης).

Ένας τρόπος απεικόνισης αυτών των σχέσεων φαίνεται στον ακόλουθο πίνακα:

Σύνταξη	Σημασιολογία
zDLKSH	String constant
162728	Integer Constant
IcountForLoop	IntegerType Variable
X	Variable
4+3	Addition (IntegerType Constant, IntegerType Constant)
Not (X Or Y)	Negation Boolean Type (BooleanType Variable InclusiveOr BooleanType Variable)

Η σχέση μεταξύ ενός αλφάβητου και οι κανόνες ανάπτυξης λέξεων σύμφωνα με το συγκεκριμένο αλφάβητο χαρτογραφούνται (αντιστοιχίζονται) στα επίσημα αντικείμενα του σημασιολογικού προτύπου για το οποίο εκείνα τα σύμβολα και οι συντακτικοί κανόνες για αυτά έχουν μια συγκεκριμένη ή αποτελούμενη έννοια. Από την συντακτική πλευρά έχουμε τα σύμβολα και από την σημασιολογική πλευρά

έχουμε τους κανόνες. Επιπλέον, έχουμε τους κανόνες χαρτογράφησης (αντιστοίχισης) των όσων κατασκευάζουμε από την συντακτική πλευρά σε όσα κατασκευάζουμε από την σημασιολογική πλευρά.

Το σημαντικό είναι ότι έχουμε καθορίσει μια γλώσσα προδιαγραφών που παραπέμπει σε εκείνα τα σημασιολογικά αντικείμενα που θέλουμε εκείνη η γλώσσα και τα κατασκευάσματά της να αναφέρεται (δηλ. να σημαίνει).

Το παραπάνω σχήμα επιδεικνύει την αντιστοίχιση μεταξύ των συντακτικών αντικειμένων και μιας απλής σημασιολογίας για αυτά τα αντικείμενα. Πιο συγκεκριμένα, τα συντακτικά αντικείμενα αντιστοιχίζονται σε μια περιγραφική στενογραφία για τη σημασιολογία: "zDLKFL" είναι μια σταθερά από χαρακτήρες, "4+3" είναι μια πράξη πρόσθεσης και λοιπά.

Η σημασιολογία στο επίπεδο του υπολογιστή είναι σε πρωτόγονο στάδιο, απλή, συγκριτικά με τη σύνθετη, πλούσια σημασιολογία των ανθρώπων, καθώς ο υπολογιστής δε γνωρίζει και δεν μπορεί να απεικονίσει, τα πάντα είναι bits. Αλλά με το σχεδιασμό ενός λογικού συστήματος αναπαράστασης γνώσης (μιας γλώσσα) και των οντολογιών (εκφράσεις αυτού που οι άνθρωποι θέλουν να αναπαραστήσουν για τον κόσμο, τις οντότητές του, και τις σχέσεις μεταξύ εκείνων των οντοτήτων), και με τον υπολογιστή να μπορεί να καταλήξει σε συμπεράσματα που είναι εξαιρετικά κοντά σε όσα ο ανθρώπινος νους θα συμπεράνε σε αντίστοιχες περιστάσεις (ισχυρισμοί, γεγονότα και τα λοιπά), θα έχουμε κατορθώσει τα συστήματά μας να είναι εφοδιασμένα με σημασιολογικές απαντήσεις στο επίπεδο του ανθρώπινου μυαλού. Θα έχουμε επομένως έναν εν λειτουργία σημασιολογικό Ιστό.

4.5. Παράδειγμα

4.5.1. Ιεράρχηση

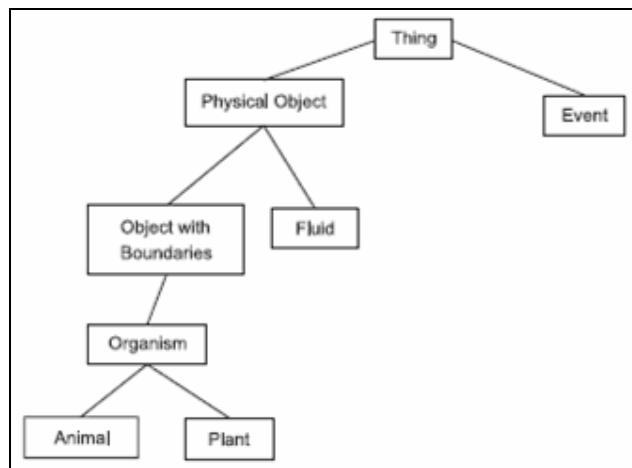
Η ιεράρχηση που καθημερινά κάνουμε τόσο εύκολα μπορεί να χαρακτηριστεί άτυπη. Περισσότερη τυπικότητα χρειάζεται όταν απαιτείται ακρίβεια, όταν πρόκειται για επιστημονική εργασία και για συστήματα υπολογιστών όπου το ένα πρέπει να κατανοήσει τις πληροφορίες του άλλου. Τα επίσημα συστήματα ταξινόμησης χρειάζονται ένα σχέδιο για την οργάνωσή τους. Το απλούστερο είδος οργάνωσης είναι ο ιεραρχικός κατάλογος.

4.5.2. Κατάλογοι, ιεραρχίες, και δέντρα

Εάν μας ζητηθεί να γράψουμε ένα μενού, το πιθανότερο είναι πως θα δημιουργήσουμε έναν ιεραρχικό κατάλογο. Για παράδειγμα, ένα μενού γευμάτων που έχει ακριβώς δύο επίπεδα στην ιεραρχία του είναι το εξής:

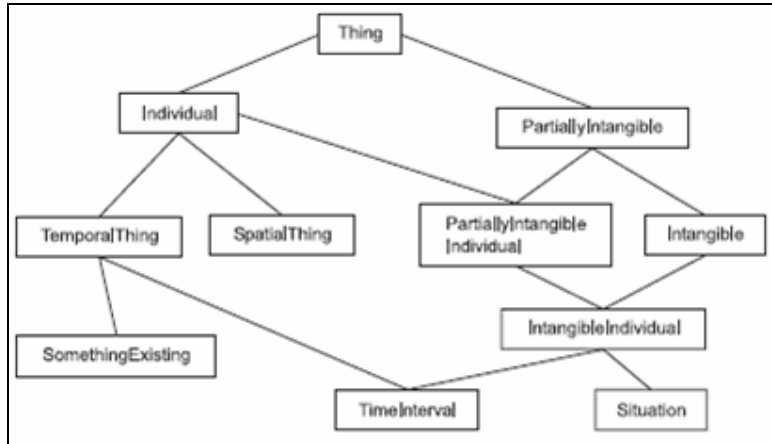
Dinner
 Chicken
 Potatoes
 Asparagus
Dessert
 Ice Cream
 Flan
Beverages
 Water
 White Wine

Αυτό το παράδειγμα, ωστόσο, δεν είναι πραγματικά μια ταξινόμηση - το κοτόπουλο δεν είναι κάποιο γεύμα, αλλά αποτελεί μέρος ενός γεύματος. Είναι φυσικό να προσπαθήσει κανείς να ταξινομήσει τα πράγματα με μια ιεραρχία. Σε μια ακριβέστερη ιεραρχία ταξινόμησης, που καλείται δέντρο, κάθε όρος αποτελεί ένα είδος αυτού που περιγράφει ο όρος που αποτελεί τον πατέρα του στο συγκεκριμένο δέντρο. Το ακόλουθο σχήμα παρουσιάζει ένα παράδειγμα.



Σε ένα τέτοιο δέντρο ταξινόμησης, η διακλάδωση είναι βασισμένη στις διακρίσεις που μπορούν να γίνουν μεταξύ των κλάσεων, όπως το *Animal* έναντι του *Plant*. Φυσικά, οι διακρίσεις πρέπει να είναι σαφείς. Στα δέντρα, μια γενικότερη έννοια τοποθετείται ψηλότερα στο διάγραμμα από μια πιο συγκεκριμένη. Ο όρος που βρίσκεται ακριβώς επάνω από κάποιον άλλο καλείται γονέας του. Το παράδειγμα που παρουσιάζεται εδώ είναι ένα χαρακτηριστικό είδος σχεδίου ταξινόμησης.

Εντούτοις, ο κόσμος είναι σύνθετος, και μια ιεραρχία είναι συχνά πάρα πολύ απλή για να συλλάβει την ουσία της σχέσης μεταξύ των διαφόρων στοιχείων. Παραδείγματος χάριν, οι οντότητες μπορούν να πάρουν ιδιότητες από περισσότερους από έναν γονείς. Στο σχήμα που παραθέτουμε στη συνέχεια, απεικονίζεται ένα μικρό μέρος της οντολογίας Cyc [12].

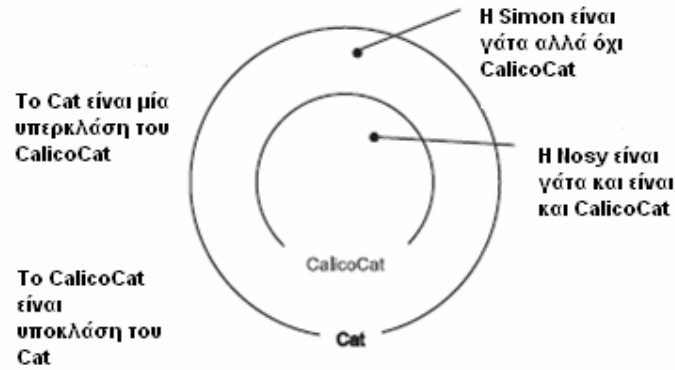


Το Cyc είναι ένα εξαιρετικά περίπλοκο σημασιολογικό σύστημα που στόχο έχει να συλλάβει τη γνώση, έτσι ώστε ένας υπολογιστής να μπορεί να εκτελέσει διάφορους συλλογισμούς.

4.6. Βασικές έννοιες

4.6.1. Κλάσεις, υποκλάσεις και instances

Υποθέτουμε ότι έχουμε ορίσει μία κλάση: *cat*, η οποία μπορεί να έχει πραγματικά instances. Από τις γάτες που αποτελούν τα instances αυτής της κλάσης, μερικές μπορούν να είναι γάτες θηλυκές, λευκές, μαύρες κλπ. Είμαστε ελεύθεροι να ορίσουμε μια νέα κλάση, αποκαλούμενη *CalicoCat*, με κάποιο ιδιαίτερο χαρακτηριστικό όσον αφορά στο τρίχωμα του συγκεκριμένου είδους. Όλες οι γάτες μας ανήκουν σε αυτή τη νέα κατηγορία (εξ ορισμού). Αλλά ανήκουν ακόμα στην κλάση *cat* επίσης. Το *CalicoCat* καλείται υποκλάση του *cat*, η οποία είναι superclass της *CalicoCat*. Το παρακάτω σχήμα επεξηγεί αυτές τις σχέσεις.



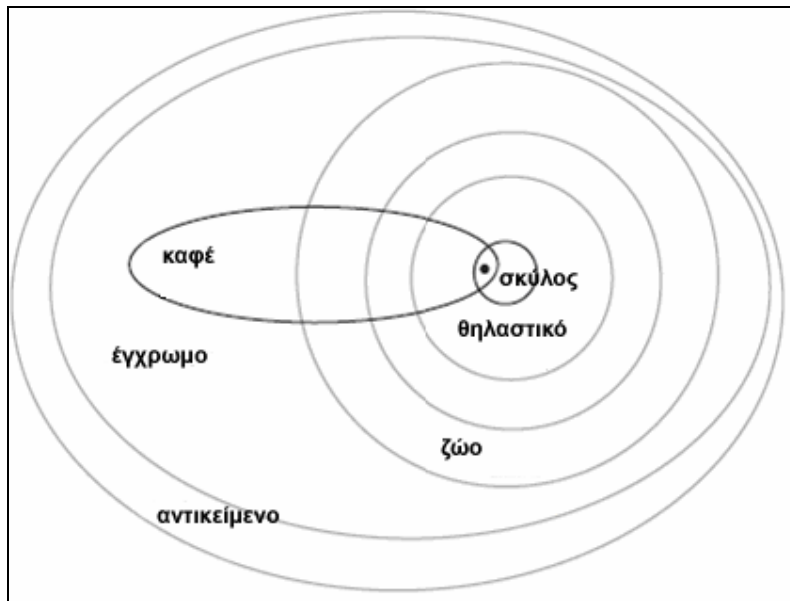
Μπορούμε να κάνουμε λόγο για κλάσεις και υποκλάσεις, υποκλάσεις και υπερκλάσεις, types και subtypes, subtypes και supertypes, που υφίστανται μεταξύ των συσχετισμένων κλάσεων ή μεταξύ δύο παρακείμενων επιπέδων σε μια ιεραρχία ταξινόμησης.

Ένα συγκεκριμένο, μεμονωμένο πράγμα, όπως μια συγκεκριμένη γάτα, καλείται instance της κλάσης του. Αυτή όμως δεν είναι μια μοναδική σχέση, επειδή η συγκεκριμένη γάτα είναι επίσης instance της superclass της κλάσης cat στο σχέδιο ταξινόμησης. Σύμφωνα με το παραπάνω σχήμα, το instance Nosal ανήκει τόσο στην κλάση CalicoCat, όσο και στην κλάση Cat (και θα ήταν επίσης instance του “θηλαστικού”, εάν είχαμε καθορίσει μια τέτοια κλάση).

4.6.2. Πολλαπλές κλάσεις

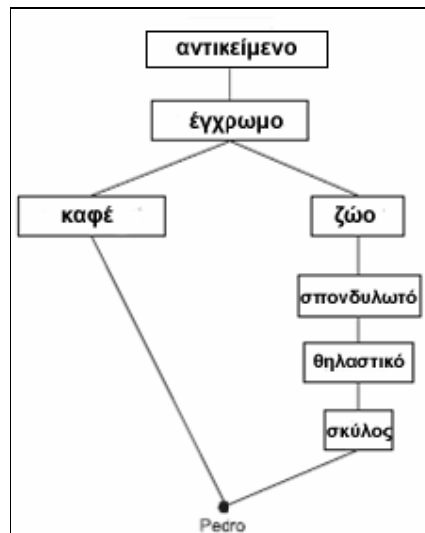
Ένα instance μπορεί να ανήκει σε περισσότερα του ενός συστήματα ταξινόμησης. Η συγκεκριμένη γάτα που αναφέραμε παραπάνω, θα μπορούσε επίσης να τοποθετηθεί στην κλάση CityResidents, που περιλαμβάνει τα όντα που ζουν σε μια πόλη. Τα διαφορετικά συστήματα ταξινόμησης δεν χρειάζεται να έχουν σχέση το ένα με το άλλο.

Τα επόμενα δύο σχήματα επεξηγούν τα συστήματα πολλαπλάσιας ταξινόμησης που ισχύουν για έναν καφέ σκύλο.



Σύμφωνα με το ένα σύστημα ταξινόμησης, το αντικείμενο είναι ένας σκύλος, ενώ σύμφωνα με το άλλο είναι ένα καφέ αντικείμενο. Τα «σκύλος» και «καφέ» έχουν κοινή μία μικρή περιοχή, καθώς κάποια καφέ αντικείμενα είναι σκύλοι, ενώ κάποιοι σκύλοι είναι καφέ.

Στο ακόλουθο σχήμα βλέπουμε μία άλλη ταξινόμηση ενός αντικειμένου (του σκύλου που ονομάζεται Pedro).



4.6.3. Κλάσεις ως σύνολα

Υπάρχουν πολλοί τρόποι να καθοριστούν οι κλάσεις, αλλά οι περισσότερες τρέχουσες γλώσσες οντολογίας για τον Ιστό βασίζονται στα σύνολα. Μια κλάση θεωρείται ένα σύνολο στοιχείων που συγκεντρώνονται με σκοπό την ταξινόμηση. Ένα σύνολο είναι μια μαθηματική δομή που αναπαριστά μια συλλογή αντικειμένων, κανένα από τα οποία δεν έχει αντίγραφο και στην οποία συλλογή τα στοιχεία δεν διατάσσονται σε κάποια σειρά.

Σύμφωνα με τα σύνολα, μια κλάση είναι το σύνολο του οποίου μέλη είναι ακριβώς εκείνα τα instances που ικανοποιούν τα κριτήρια της κλάσης. Μια υποκλάση είναι ένα υποσύνολο. Αυτό καθιστά σαφές ότι τα μέλη της υποκλάσης έχουν όλα τα ίδια χαρακτηριστικά της κλάσης των γονέων τους, επειδή τα μέλη της υποκλάσης είναι (εξ ορισμού) επίσης μέλη της κλάσης γονέων. Η γλώσσα οντολογίας του Ιστού (OWL) έχει έναν λίγο διαφορετικό ορισμό για την κλάση. Η διαφορά έγκειται στο ότι η κλάση στην περίπτωση αυτή δεν περιλαμβάνει instances, αλλά συσχετίζεται με αυτά.

4.7. Ονοματολογία

Μια οντολογία έχει να κάνει με τα ονόματα των κλάσεων και το πώς συσχετίζονται το ένα με το άλλο. Τα ονόματα αντιπροσωπεύουν τις έννοιες, κλάσεις και κατηγορίες. Τα διάφορα στοιχεία έχουν διαφορετικές συσχετίσεις το ένα με το άλλο (μια γάτα είναι υποκλάση του *animal*, ένας βράχος δεν είναι), επίσης έχουν ιδιότητες, και εκείνες οι ιδιότητες έχουν ονόματα επίσης. Μια οντολογία διευκρινίζει τα ονόματα, τις ιδιότητες που μας ενδιαφέρουν και τις σχέσεις μεταξύ τους, μαζί με τους περιορισμούς που πρέπει να χρησιμοποιούνται κατάλληλα.

Ένα αυτοκίνητο, παραδείγματος χάρη, μπορεί να έχει τέσσερις ρόδες (ή ενδεχομένως τρεις, ανάλογα με τον ορισμό που έχουμε δώσει), αλλά σίγουρα όχι λιγότερες. Γενικά, τα ονόματα χρησιμοποιούνται για να διευκρινίσουν τις έννοιες.

4.7.1. Ονόματα και προσδιοριστικά

Για να κατασκευάσουμε οντολογίες, χρειαζόμαστε ένα σύστημα προσδιοριστικών για να αναφερθούμε στους τύπους, τις κλάσεις και τις ιδιότητες που σκοπεύουμε να δημιουργήσουμε. Εμείς χρησιμοποιούμε τα ονόματα, αλλά τα ονόματα δεν είναι

μοναδικά προσδιοριστικά. Η RDF έχει υιοθετήσει το σχέδιο του προσδιορισμού των στοιχείων με μοναδικές αναφορές URI, είτε τα προσδιορισμένα στοιχεία είναι προσπελάσιμα από τον Ιστό είτε όχι. Τα URIs χρησιμοποιούνται ως προσδιοριστικά ακόμα και όταν δεν υποδεικνύουν κάποιον πραγματικό πόρο που είναι προσβάσιμος μέσα από ένα δίκτυο.

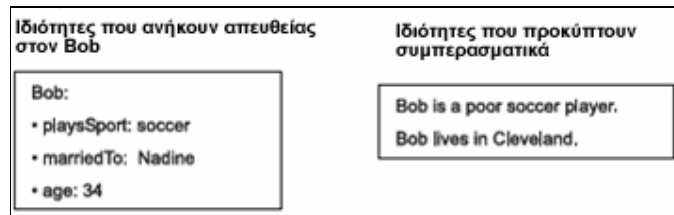
4.8. Ιδιότητες

Εκτός από τις κατηγορίες ταξινόμησης σε μια οντολογία πρέπει να προβλέψουμε και τις ιδιότητες. Οι ιδιότητες περιγράφονται συχνά με τη μορφή του ζευγαριού {όνομα, τιμή} ή κάποια ισοδύναμη. Ένα παράδειγμα είναι {χρώμα, κόκκινο}. Στο πρότυπο γραφικών παραστάσεων σε RDF, η τιμή μιας ιδιότητας αναπαρίσταται ως κόμβος που συνδέεται με τον κάτοχο της ιδιότητας με ένα τόξο ή μια άκρη που ονομάζει την ιδιότητα. Ο τύπος της ιδιότητας (όπως το χρώμα) καλείται συχνά κατηγορημα, που υπονοεί μια γενικότερη σχέση από την ιδιότητα απλώς. Ένα κατηγορημα αναφέρεται σε οποιαδήποτε δυαδική σχέση μεταξύ δύο στοιχείων. Ιδιότητες όπως το χρώμα και το βάρος είναι συγκεκριμένοι τύποι δυαδικών σχέσεων.

Υπάρχουν άλλες διαφορετικές προσεγγίσεις στην αναπαράσταση των ιδιοτήτων αλλά είναι συνήθως, λίγο πολύ, ισοδύναμες από λογική άποψη. Στις αντικειμενοστρεφείς γλώσσες προγραμματισμού, τα αντικείμενα έχουν τις ιδιότητες ως τμήμα των ορισμών τους. Οι τιμές τους ορίζονται όταν εκτελείται το πρόγραμμα. Εάν μια σφαίρα είναι κόκκινη, για παράδειγμα, το κόκκινο χρώμα "ανήκει" στη σφαίρα από μια άποψη. Όταν η γνώση αναπαρίσταται σε ένα πλήθος υπολογιστών, στο αντικείμενο που αντιπροσωπεύει τη σφαίρα θα οριζόταν μια ιδιότητα χρώματος της οποίας η τιμή θα ήταν "κόκκινο".

Αλλά στην πραγματική ζωή, τα αντικείμενα μπορούν επίσης να έχουν ιδιότητες που δεν τους "ανήκουν". Ας πάρουμε για παράδειγμα μια ιστορία σε ένα περιοδικό, σύμφωνα με την οποία: "Robert Smith is a poor soccer player." Μπορούμε να γράψουμε αυτήν την αξίωση ως εξής: {" Robert Smith", has-soccer-ability, "poor" }. Αυτή η αξίωση έχει οριστεί για τον Smith από έναν τρίτο και μπορεί να μην έχει καμία σχέση με την εγγενή φύση του. Στο σημασιολογικό Ιστό, όπου ο στόχος είναι να ειπωθούν τα πάντα για τα πάντα, οι περισσότερες ιδιότητες θα βεβαιωθούν με ανάλογο τρόπο από τρίτους. Οι ιδιότητες δεν χρειάζεται να ανήκουν στο

υποκειμένο. Σε RDF παρέχεται αυτή η δυνατότητα. Το σχήμα που ακολουθεί επεξηγεί αυτήν την διάκριση.



Όταν μια ιδιότητα δεν είναι ένα έμφυτο μέρος του ορισμού ενός πράγματος, αλλά αντ' αυτού βεβαιώνεται χωριστά, είναι δυνατό για την ιδιότητα αυτή να εφαρμοστεί με έναν "ακατάλληλο" τρόπο. Παραδείγματος χάρη, ας θεωρήσουμε ότι η ιδιότητα "shows-time" έχει τιμή "5:00μ.μ." και ότι αναφέρεται σε ένα ρολόι. Δεν μπορεί να έχει τιμή "πράσινο", ούτε θα είχε κάποιο νόημα ως ιδιότητα ενός θαλάσσιου κύματος. Οι οντολογίες χρησιμοποιούνται για τον προσδιορισμό του ποιες ιδιότητες και ποιοι τύποι τιμών είναι κατάλληλοι με τη βοήθεια διάφορων περιορισμών.

4.9. Κατασκευή οντολογίας

4.9.1. Τι πρέπει να λάβουμε υπόψη

Μία από τις σημαντικότερες λειτουργίες μιας συγκεκριμένης οντολογίας είναι ο καθορισμός ενός συνόλου κλάσεων που καλύπτουν όλες μαζί μια γνωστική περιοχή [1,4]. Παραδείγματος χάριν, μια οντολογία για τις πωλήσεις θα πρέπει να καλύψει τους πελάτες, τις εντολές αγοράς, τις παραλαβές πωλήσεων, τα στοιχεία καταλόγων και τα λοιπά. Πολλές υπάρχουσες οντολογίες ήδη καλύπτουν ποικίλες περιοχές και πολύ περισσότερες θα υπάρξουν στο μέλλον. Οι συγκεκριμένες οντολογίες πρέπει να κατασκευαστούν με τα γνωστά λεξιλόγια και κανόνες κατασκευής.

4.9.1.1 Πλαίσια

Ένα πλαίσιο θα παράσχει μια σύνταξη, ένα λεξιλόγιο, και μερικούς προκαθορισμένους όρους. Θα μπορούσαμε σχεδόν να πούμε ότι ένα πλαίσιο

οντολογίας είναι μια οντολογία για την κατασκευή των οντολογιών. Μια οντολογία θα καθορίσει γενικά τις κλάσεις ή κατηγορίες, τους όρους, και τις σχέσεις. Θα καθορίσει ποιες κλάσεις μπορούν να χρησιμοποιηθούν με ποιες σχέσεις. Μπορεί επίσης να καθορίσει τους τύπους των δεδομένων και τους περιορισμούς στις χρήσεις των κλάσεων και των ιδιοτήτων. Οι οντολογίες δεν είναι συστήματα λογικής αλλά μπορούν να διευκρινίσουν τους κανόνες για να υποστηριχθούν ορισμένα λογικά συμπεράσματα. Παραδείγματος χάρη, ένα σύστημα οντολογίας ή ένα πλαίσιο μπορεί να παρέχει έναν τυποποιημένο τρόπο να διευκρινιστεί ότι δύο κλάσεις είναι ισοδύναμες. Ένας λογικός reasoner που συναντά ένα Instance μιας από τις κλάσεις θα ήξερε ότι όλα τα χαρακτηριστικά της ισοδύναμης κλάσης ισχύουν επίσης.

4.9.1.2 Σχεδιασμός οντολογιών

Η δημιουργία μια καλής ποιότητας οντολογίας, που είναι καλά δομημένη, είναι απαλλαγμένη από τις αντιφάσεις και εκφράζει την πρόθεση των σχεδιαστών της δεν είναι εύκολη. Φυσικά, ο καθένας μπορεί να παράγει ένα λεξιλόγιο πολύ γρήγορα. Για παράδειγμα:

```
Living Thing
Plant
Animal
    Wild
    Pet
        Dog
        Cat
    Healthy
    Sick
```

Ας σταματήσουμε καλύτερα εδώ, καθώς έχουμε ήδη προβλήματα. Για παράδειγμα, μια γάτα ή ένας σκύλος μπορεί να είναι άγριο ζώο, αλλά μπορεί επίσης να είναι και κατοικίδιο. Αλλά πιθανώς δεν θέλουμε να επαναλάβουμε τις γάτες και τα σκυλιά κάτω από το *wild*. Επιπλέον, τα *wild*, *pet*, *healthy* και *sick* δεν είναι οργανισμοί. Περιγράφουν ορισμένες ιδιότητες ή καταστάσεις των οργανισμών, και δεν ανήκουν σε μια ιεραρχία ζωντανών οργανισμών. Μπορεί να ανήκουν κάπου αλλού στην οντολογία.

Ο σχεδιαστής μιας καλής οντολογίας πρέπει να μπορεί να αντιληφθεί και να αρθρώσει τις ελλοχεύουσες ιδέες, και έχοντας καλή γνώση της σύνταξης της γλώσσας

οντολογίας θα μπορέσει να διαμορφώσει το σωστό πρότυπο. Οποιαδήποτε από αυτές τις δεξιότητες είναι δύσκολο να αποκτηθεί, πόσο μάλλον όλες μαζί. Επιπλέον, μια οντολογία είναι συνήθως ένας συμβιβασμός μεταξύ διαφορετικών αναγκών, και συχνά πολλοί εμπλέκονται στο σχεδιασμό της.

4.9.1.3 Τι άλλο πρέπει να λάβουμε υπόψη

Ποικίλες άλλες πρακτικές εκτιμήσεις εμφανίζονται όταν χρησιμοποιούμε τις οντολογίες σε μεγαλύτερη κλίμακα. Θα θελήσουμε, για παράδειγμα, να χρησιμοποιήσουμε τα μέρη περισσότερων από μία οντολογιών και, όπως οτιδήποτε κατασκευάζεται αλλάζει κατά τη διάρκεια του χρόνου μέσω μιας διαδικασίας εξέλιξης, θα πρέπει να είμαστε σε θέση να προσαρμοστούμε στις αλλαγές κάθε έκδοσης. Οι συγχωνευμένες οντολογίες μερικές φορές μπορεί να χρησιμοποιήσουν τους όρους ή τις κλάσεις από μια οντολογία μαζί με μερικές από άλλη.

Υποθέστε, παραδείγματος χάρη, ότι εσείς έχετε έναν πράκτορα λογισμικού για να κάνει τις ρυθμίσεις του ταξιδιού σας. Ο πράκτορας πρέπει να καταλάβει τις έννοιες και τους όρους των επιχειρήσεων αερογραμμών, και επίσης, για παράδειγμα, των ξενοδοχείων. Όμως, οι αερογραμμές και τα ξενοδοχεία ίσως να χρησιμοποιούν διαφορετικούς όρους και σχήματα για τους χρόνους και τις ημερομηνίες. Ο πράκτορας δεν είναι απαραίτητο να ξέρει τα πάντα, αλλά τόσα ώστε να πραγματοποιήσει μία κράτηση στο σωστό χρόνο για τους σωστούς ανθρώπους. Το σχήμα που ακολουθεί απεικονίζει ένα πιθανό ζευγάρι των οντολογιών με το οποίο ο πράκτοράς σας θα πρέπει να εργαστεί.

Έννοιες οντολογίας αεροπορικής εταιρίας

Fight
Fight number
Price
Person
passenger
Passenger List
Time
scheduled arrival time
actual arrival time

Έννοιες οντολογίας ξενοδοχείου

Amenities
Occupancy
Person
guest
Reservation
Room
Room type
Room schedule
Tariff
Time
check-in time
checkout time
scheduled check-in time
scheduled checkout time

Ο συνδυασμός μιας οντολογίας με μια άλλη καλείται μερικές φορές συγχώνευση. Ένας άλλος όρος είναι το *import* - μια οντολογία μπορεί να εισαχθεί σε μία άλλη. Αυτή η διαδικασία ηχεί απλή αλλά μπορεί να είναι δυσνόητη. Η προφανής ερώτηση είναι εάν η συγχώνευση μπορεί τελικά να επιτευχθεί. Εάν οι δύο οντολογίες κατασκευάστηκαν χρησιμοποιώντας το ίδιο σύστημα (όπως την OWL, παραδείγματος χάρη), μια συγχώνευση θα είναι συνήθως τεχνικά εφικτή. Εάν όμως κατασκευάστηκαν με διαφορετικό, δεν είναι σίγουρο εάν θα μπορούσαν να συγχωνευθούν, ακόμα και σε γενικές γραμμές. Κάποιος θα πρέπει να δημιουργήσει τις αντιστοιχίες τους σε μια άλλη μορφή, αλλά εάν χρησιμοποιούν ασυμβίβαστες σχεδιαστικές αρχές, αυτό μπορεί να μην είναι δυνατό, ή μπορεί να απαιτήσει κάποιες πληροφορίες να αγνοηθούν ή να προστεθούν. Ακόμα κι αν δύο οντολογίες μπορούν να συγχωνευθούν, μπορεί να μην είναι συνεπείς. Όταν υπάρχουν ασυνέπειες, μια από τις ακόλουθες εκβάσεις θα συμβεί πιθανώς: Διαφορετικά συμπεράσματα θα μπορούσαν να συναχθούν από μια μηχανή συλλογισμού, ανάλογα με τις δηλώσεις που ήταν γνωστές στο σύστημα και το σχέδιο του reasoner ή οι συγκρουόμενες κλάσεις δεν θα έχουν ποτέ κάποιο instance (δεδομένου ότι κανένα instance δεν μπορεί να ικανοποιήσει τους συγκρουόμενους όρους) ή ο reasoner θα ανακαλύψει τη σύγκρουση και θα σταματήσει τη διαδικασία που εκτελεί.

Ένα παράδειγμα θα ήταν μια κλάση που αντιπροσωπεύει ένα αυτοκίνητο, το οποίο σε μια οντολογία είναι περιορισμένο στην κατοχή τεσσάρων τροχών, ενώ σε άλλη επιτρέπεται να έχει τρεις καθώς επίσης και τέσσερις τροχούς. Τι θα έπρεπε να συμβεί όταν το σύστημα συναντήσει ένα τρίτροχο όχημα που ανήκει στην κλάση αυτοκίνητο σύμφωνα με τη μια οντολογία αλλά όχι σύμφωνα με την άλλη;

Στην πράξη, το σύστημα θα ήταν σε θέση πιθανώς να διακρίνει δύο τέτοιες κλάσεις, καθώς προέρχονται από διαφορετικές οντολογίες και επομένως θα έχουν διαφορετικά προσδιοριστικά. Αλλά είναι επίσης πιθανό ότι το σύστημα θα ήξερε ότι οι δύο κλάσεις προορίζονταν να αντιπροσωπεύσουν το ίδιο είδος. Αυτό θα μπορούσε να συμβεί επειδή κάποιος κατασκεύασε έναν χάρτη μεταξύ των δύο οντολογιών, ή επειδή το ίδιο το σύστημα ήταν σε θέση να το ανακαλύψει.

Ο ασφαλέστερος τρόπος να αποτραπεί αυτό το είδος σύγκρουσης θα ήταν να αποφύγει τις κλάσεις που αναπαριστούν τα ίδια πράγματα. Αλλά αυτό θα μπορούσε να βεβαιωθεί μόνο εάν οι συγχωνευμένες οντολογίες κάλυπταν απολύτως διαφορετικές περιοχές.

4.9.1.4 Δέσμευση

Όταν αποφασίζουμε να χρησιμοποιήσουμε μια συγκεκριμένη οντολογία, δεχόμαστε όλους τους ισχυρισμούς και τις επιπτώσεις που αυτοί συνεπάγονται όσον αφορά στη συγκεκριμένη οντολογία. Στην περίπτωση αυτή λέμε ότι κάνουμε *commit* στην οντολογία που μας ενδιαφέρει. Στο παράδειγμα του προηγούμενου σχήματος, ο πράκτορας που προσπαθεί να κάνει τις ρυθμίσεις ταξιδιού κάνει *commit* στην οντολογία αεροπορικού ταξιδιού.

4.9.1.5 Οι αλλαγές

Στη καθημερινή ζωή, η σημασία και η χρήση των λέξεων αλλάζουν και εξαρτώνται συχνά από τα συμφραζόμενα. Κατά τον ίδιο τρόπο αλλάζουν και οι οντολογίες. Όροι μπορεί να προστεθούν ή να αφαιρεθούν, οι ορισμοί των ήδη υπάρχοντων όρων μπορεί επίσης να αλλάξουν ή να διευρυνθούν. Επομένως, είναι επιθυμητό για μια οντολογία να είναι σε θέση να δηλώσει την έκδοσή της. Αλλά ακόμα κι αν μπορεί, υπάρχουν ακόμα σύνθετα ζητήματα που εξετάζονται.

Ας θεωρήσουμε ότι μια οντολογία που χρησιμοποιούμε αλλάζει σε μια νέα έκδοση. Κατά συνέπεια, θα μπορούσαν να αλλάξουν τα συμπεράσματα στα οποία έχουμε καταλήξει, τα σημαντικά μέρη της βάσης γνώσεών μας, ή να έχει επιπτώσεις στις επικοινωνίες με τους πράκτορες ή τους επιχειρησιακούς συνεργάτες. Φυσικά, ο σχεδιαστής οντολογίας θα προσπαθήσει να κάνει τις αλλαγές με τέτοιο τρόπο ώστε να αποφευχθούν αυτά τα προβλήματα, αλλά αυτό δεν είναι πάντα εφικτό.

4.9.2. Βήματα για την κατασκευή οντολογίας

1. Προσδιορισμός του **scope**
2. Επαναχρησιμοποίηση
3. Καθορισμός των όρων
4. Προσδιορισμός της ταξινόμιας
5. Προσδιορισμός ιδιοτήτων
6. Προσδιορισμός **facets**
7. Προσδιορισμός των *instances*
8. Έλεγχος για τυχόν ασυνέπειες

4.9.2.1 Προσδιορισμός του scope

Ο σχεδιασμός μιας οντολογίας σχετίζεται με τον προσδιορισμό του συνόλου των δεδομένων και της δομής τους έτσι ώστε να μπορούν να χρησιμοποιηθούν από διάφορα προγράμματα. Μία οντολογία δηλαδή αποτελεί ένα μοντέλο (πρότυπο) για έναν συγκεκριμένο τομέα και για έναν συγκεκριμένο σκοπό. Κατά συνέπεια δεν υπάρχει σωστή ή λανθασμένη οντολογία, αλλά για κάθε τομέα που περιγράφουμε μπορεί να υπάρχουν διάφορες εναλλακτικές. Οι διαφορές μεταξύ τους έχουν άμεση σχέση με τη χρήση για την οποία προορίζεται η οντολογία και για τυχόν επεκτάσεις της που πρόκειται στο μέλλον να υλοποιηθούν. Τα βασικά ερωτήματα που καλούμαστε να απαντήσουμε είναι τα εξής: Ποιον τομέα θα καλύψει η συγκεκριμένη οντολογία; Ποια θα είναι η χρήση της; Σε ποια ερωτήματα πρέπει να μας παρέχει απαντήσεις; Ποιος θα κάνει χρήση και συντήρηση της οντολογίας;

4.9.2.2 Επαναχρησιμοποίηση

Με την εξάπλωση του Semantic Web οι διάφορες οντολογίες θα γίνουν όλο και περισσότερο επαναχρησιμοποιήσιμες. Ήδη, αν επιχειρήσουμε να κατασκευάσουμε μία οντολογία, σίγουρα θα βρούμε κάποια σχετική με το αντικείμενο που μας ενδιαφέρει, την οποία μπορούμε να λάβουμε υπόψη για την έναρξη της δικής μας εργασίας.

4.9.2.3 Καθορισμός των όρων

Ένα πρώτο βήμα στον σχεδιασμό μιας οντολογίας είναι η απλή καταγραφή, χωρίς κάποια ιδιαίτερη δομή, των όρων που μας ενδιαφέρουν και στους οποίους πρέπει να αναφέρεται η συγκεκριμένη οντολογία. Τυπικά, πρόκειται για ουσιαστικά όσον αφορά στις κλάσεις που θα αποτελέσουν την οντολογία, και ρήματα (ή ρηματικές εκφράσεις) όσον αφορά στις ιδιότητες (όπως για παράδειγμα *is part of*, *has component*).

4.9.2.4 Προσδιορισμός της ταξονομίας

Τον καθορισμό των σχετικών όρων πρέπει να ακολουθήσει η οργάνωση αυτών σε μια ιεραρχική ταξινόμια. Είναι σημαντικό στο σημείο αυτό να βεβαιωθούμε ότι η ιεραρχία είναι όντως ταξονομική, με την έννοια ότι αν μία κλάση A είναι υποκλάση μιας κλάσης B, τότε κάθε instance της A αποτελεί instance της B.

4.9.2.5 Προσδιορισμός των ιδιοτήτων

Το συγκεκριμένο βήμα συνήθως πραγματοποιείται παράλληλα με το προηγούμενο, καθώς είναι λογικό να οργανώσει κανείς τις ιδιότητες που συσχετίζουν τις κλάσεις, ενώ οργανώνει σε ιεραρχία τις κλάσεις της οντολογίας.

Σημειώνουμε εδώ, για παράδειγμα, ότι σύμφωνα με τη σημασιολογία της ιδιότητας *subclass of*, απαιτείται, όταν μία κλάση A αποτελεί υποκλάση της B, τότε κάθε δήλωση που αναφέρεται σε μια ιδιότητα και ισχύει για τα instances της B, πρέπει να ισχύει και για κάθε instance της A. Δεδομένης αυτής της κληρονομικότητας, είναι λογικό να ορίζουμε ιδιότητες για κλάσεις που ιεραρχικά είναι τοποθετημένες υψηλότερα, καθώς είναι πιο γενικές.

Κατά τον ορισμό ιδιοτήτων για τις διάφορες κλάσεις, ορίζουμε τα *domain* και *range* αυτών. Στο σημείο αυτό πρέπει να προσέξουμε το εξής: Είτε ορίζουμε τα *domain* και *range* όσο το δυνατό πιο γενικά μπορούμε, ώστε να μπορούν να χρησιμοποιηθούν από όλες τις υποκλάσεις που υφίστανται, είτε τα ορίζουμε όσο το δυνατό πιο συγκεκριμένα, κάτι που θα μας επιτρέψει να εντοπίσουμε ασυνέπειες στην οντολογία που οφείλονται σε τυχόν παραβιάσεις.

4.9.2.6 Προσδιορισμός των περιορισμών

Αφού υλοποιηθούν όλα τα παραπάνω βήματα, η οντολογία θα απαιτεί μόνο την εκφραστικότητα που παρέχει το RDF Schema. Ο εμπλουτισμός λοιπόν των ιδιοτήτων, που έχουν ήδη οριστεί, με περιορισμούς σημαίνει τα εξής:

- *Πληθυστικότητα*: Για όσο το δυνατό περισσότερες ιδιότητες μπορούμε, καθορίζουμε τον αριθμό των διαφορετικών τιμών που επιτρέπεται να έχουν. Συχνά συναντούμε τα εξής: “at least one value” (*required properties*), “at most one value” (*single-valued properties*)
- *Απαιτούμενες τιμές*: Συχνά μία κλάση καθορίζεται από τις συγκεκριμένες τιμές που έχει μία ιδιότητά της, και τέτοιες τιμές ορίζονται στην OWL κανοντας

χρήση της *owl:hasValue*. Υπάρχει περίπτωση επίσης μία ιδιότητα να παίρνει τιμές από μία συγκεκριμένη κλάση (*owl:someValuesFrom*)

- *Χαρακτηριστικά σχέσεων*: πρόκειται για τα εξής χαρακτηριστικά: συμμετρία, μεταβατικότητα, αντίθετες ιδιότητες και λειτουργικές (functional) τιμές.

Αφού υλοποιηθεί και αυτό το βήμα, μπορούμε να ελέγξουμε την οντολογία για τυχόν ασυνέπειες (πάντα μετά από το τελευταίο αυτό βήμα, καθώς το RDF Schema δεν παρέχει αυτή τη δυνατότητα). Ασυνέπειες που πολλές φορές συναντούμε αναφέρονται σε ασυμβατότητα των ορισμών των domain και range για μεταβατικές, συμμετρικές και αντίθετες ιδιότητες, στην πληθυκότητα των τιμών των ιδιοτήτων και σε απαιτήσεις που μπορεί να έχουμε για κάποιες τιμές ιδιοτήτων, οι οποίες όμως ενδεχομένως να συγκρούονται με τους περιορισμούς που έχουν τεθεί στα domain και range αυτών.

4.9.2.7 Προσδιορισμός των instances

Ο αριθμός των instances σε γενικές γραμμές ξεπερνάει κατά πολύ τον αριθμό των κλάσεων μιας οντολογίας. Οι οντολογίες ποικίλουν σε μέγεθος, από εκατοντάδες κλάσεις έως και δεκάδες χιλιάδες κλάσεις, ενώ το πλήθος των instances ποικίλει από εκατοντάδες σε εκατοντάδες χιλιάδες. Αυτός είναι και ο λόγος για τον οποίο ο καθορισμός των instances δεν πραγματοποιείται manually, αλλά αυτά λαμβάνονται από βάσεις δεδομένων.

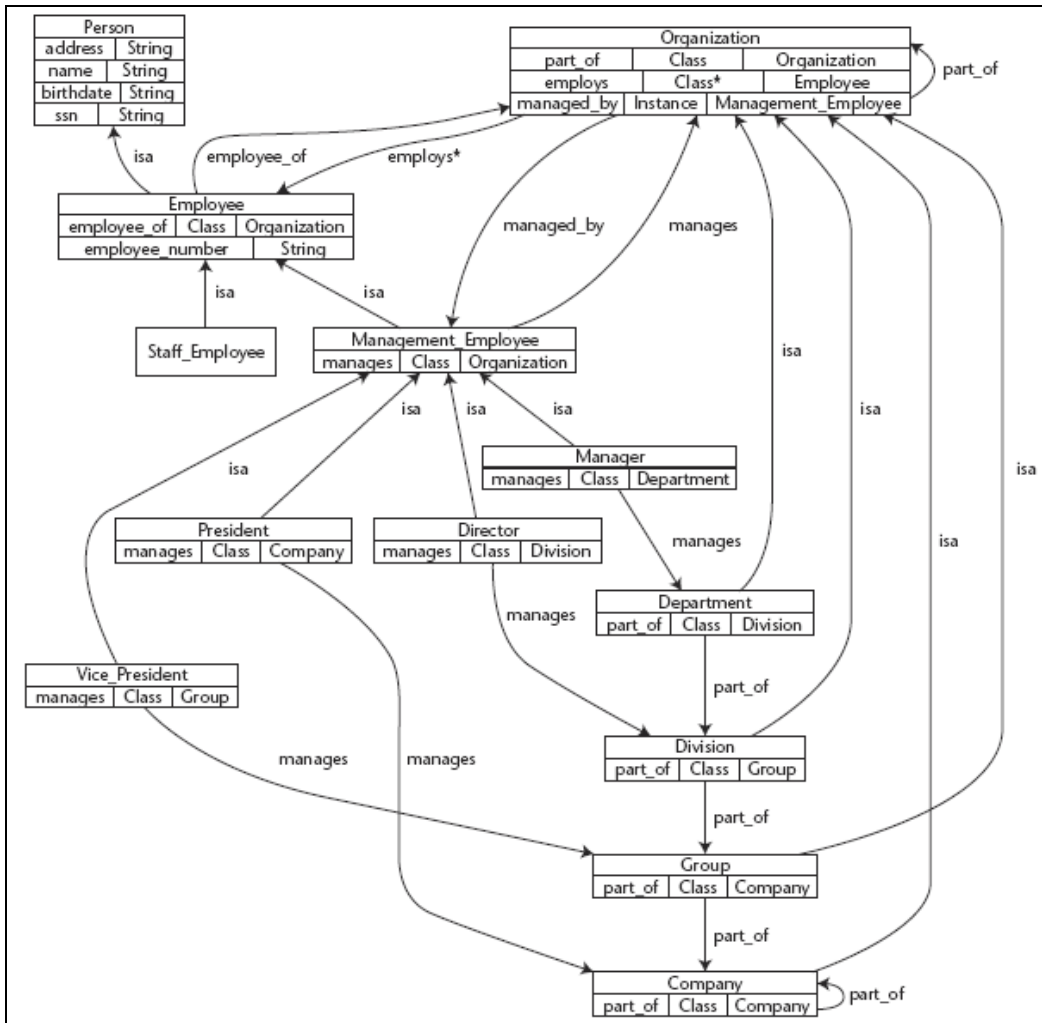
4.9.2.8 Έλεγχος για ασυνέπειες

Ένα σημαντικό πλεονέκτημα της χρήσης OWL έναντι RDF Schema είναι η δυνατότητα ελέγχου των ασυνεπειών της οντολογίας της ίδιας, και των instances που έχουν οριστεί.

Παραδείγματα ασυνεπειών είναι συχνά, όπως ήδη αναφέραμε, ασυμβατότητα των ορισμών των domain και range για μεταβατικές, συμμετρικές και αντίθετες ιδιότητες, η πληθυκότητα των τιμών των ιδιοτήτων έναντι απαιτήσεων που μπορεί να έχουμε για κάποιες τιμές ιδιοτήτων, οι οποίες όμως ενδεχομένως να συγκρούονται με τους περιορισμούς που έχουν τεθεί στα domain και range αυτών.

4.10. Παράδειγμα

Το σχήμα που ακολουθεί παρουσιάζει μια απλή οντολογία του ανθρώπινου δυναμικού μιας επιχείρησης, όπως έχει δημιουργηθεί με το εργαλείο Protégé (<http://protege.stanford.edu>).



Παρατηρούμε πως υπάρχουν κλάσεις όπως: Person, Organization, Employee, τα οποία σε μια οντολογία καλούνται έννοιες (*concepts*). Αυτές οι έννοιες και οι σχέσεις μεταξύ τους υλοποιούνται συνήθως ως κλάσεις, σχέσεις, ιδιότητες, και τιμές (των ιδιοτήτων). Έτσι το σχήμα απεικονίζει πρώτιστα τις έννοιες των σημαντικών οντοτήτων του τομέα που περιγράφει, οι οποίες υλοποιούνται ως κλάσεις, όπως είναι οι Person, Organization και Employee. Επίσης απεικονίζονται οι σχέσεις μεταξύ αυτών των οντοτήτων, όπως το employee_of, managed_by και manages. Τέλος,

απεικονίζονται ιδιότητες, όπως address, name, birthdate, κάτω από την κλάση Person. Αυτές οι ιδιότητες έχουν είτε συγκεκριμένες τιμές είτε, συχνότερα, τιμές από κάποια άλλη κλάση (σε αυτή την περίπτωση οι τιμές που παίρνουν έχουν κάποιο συγκεκριμένο range). Η τιμή για την ιδιότητα employee_of, μια ιδιότητα της κλάσης Employee, παραδείγματος χάρη, προέρχεται από την κλάση Organization. Με τον όρο range εννοούμε ότι οι μόνες πιθανές τιμές για οποιαδήποτε instances της ιδιότητας employee_of που καθορίζεται για την κλάση Employee πρέπει να προέλθουν από την κλάση Organization.

Αμέσως βλέπουμε ότι μια οντολογία προσπαθεί να συλλάβει την έννοια (την επονομαζόμενη σημασιολογία) μιας ιδιαίτερης θεματικής περιοχής ή του τομέα της γνώσης που αντιστοιχεί σε αυτό που κάποιος γνωρίζει για εκείνη την περιοχή. Μια οντολογία χαρακτηρίζει επίσης εκείνη τη σημασία από την άποψη των εννοιών και των σχέσεών τους. Επιπλέον, μια οντολογία αντιπροσωπεύεται συχνά ως κλάσεις, σχέσεις, ιδιότητες και τιμές. Το σχήμα είναι μια γραφική αναπαράσταση μιας απλής οντολογίας που προσπαθεί να απεικονίσει το ανθρώπινο δυναμικό (πρόσωπο, υπάλληλος, οργάνωση), τις υποκατηγορίες του (υπάλληλος προσωπικού, διοικητικός υπάλληλος, επιχείρηση, ομάδα, τμήμα), τις ιδιότητές τους, και τις σχέσεις μεταξύ των εννοιών.

Αυτό που σημειώνουμε εδώ είναι το εξής: Δεν υπάρχει καμία λογική διαφορά μεταξύ μιας γραφικής και μιας κειμενικής απόδοσης μιας οντολογίας (ή οποιουδήποτε άλλου προτύπου απεικόνισης). Αυτό το γεγονός είναι σημαντικό, καθώς μια οντολογία αναπαρίσταται από μια γλώσσα αναπαράστασης γνώσης (όπως μια σημασιολογική γλώσσα Ιστού όπως RDF/S, DAML+OIL, OWL). Επιπλέον, τέτοιες γλώσσες οντολογίας βασίζονται σε μια ιδιαίτερη λογική, η οποία είναι η ίδια με μια γλώσσα με σύνταξη και σημασιολογία. Μερικές φορές, επομένως, καλούμε τη γλώσσα στην οποία η οντολογία αναπαρίσταται, μια γλώσσα βασισμένη στη λογική. Οι γλώσσες οντολογίας υποστηρίζονται από μια αυστηρή λογική, η οποία με αυτόν τον τρόπο κάνει την οντολογία κατανοητή από τον υπολογιστή, εννοώντας ότι η σημασιολογία του προτύπου είναι σημασιολογικά ερμηνεύσιμη από τον υπολογιστή, δηλαδή ο υπολογιστής και το λογισμικό του μπορούν να ερμηνεύσουν τη σημασιολογία του προτύπου - χωρίς άμεση ανθρώπινη συμμετοχή. Το λογισμικό που υποστηρίζεται από τις οντολογίες φτάνει στο νοητικό επίπεδο του ανθρώπινου νου, έτσι ώστε οι άνθρωποι δεν είναι απαραίτητο να κινηθούν προς το επίπεδο των

υπολογιστών. Αυτό είναι ένα εξαιρετικά σημαντικό σημείο, και υπογραμμίζει την αξία των οντολογιών.

4.11. Γλώσσες για οντολογίες

4.11.1. RDFS

Η RDFS είναι η γλώσσα RDF που χρησιμοποιείται για την περιγραφή των οντολογιών. Στην έκδοση του 1999, RDFS ήταν τα αρχικά για το Resource Description Framework Schema Specification. Εκείνη η έκδοση RDFS δεν εκδόθηκε ποτέ, αλλά χρησιμοποιήθηκε τελικά. Τώρα καλείται γλώσσα περιγραφής λεξιλογίου RDF (RDF Vocabulary Description Language), αλλά είναι ακόμα γνωστό ως RDFS και δεν έχει αλλάξει πολύ. Οι περισσότερες από τις αλλαγές ήταν στην τεκμηρίωση και την επίσημη ερμηνεία. Η RDFS είναι αρκετά σταθερή στη μορφή της, και χρησιμοποιείται από διάφορα άλλα γλωσσικά προγράμματα οντολογίας.

4.11.2. Η γλώσσα RDF

Η RDF σχεδιάζεται για να κάνουμε δηλώσεις για τους πόρους. Ένας πόρος αντιπροσωπεύει μια έννοια και του δίνεται ένα μοναδικό URI για να την προσδιορίσει. Μια δήλωση αποτελείται από τρία μέρη: ένα υποκείμενο, ένα κατηγορημα (ή ιδιότητα), και ένα αντικείμενο, το οποίο είναι η τιμή μιας ιδιότητας. Λόγω των τριών αυτών τμημάτων, μια δήλωση RDF καλείται συχνά τριπλέτα (triple). Μια δήλωση κάνει έναν ισχυρισμό για το υποκείμενό της, όπως αυτό: {Bob, livesIn, Cleveland }. Αυτή η δήλωση, πιθανώς για ένα πρόσωπο που ονομάζεται "Bob", λέει ότι ο Bob έχει μια ιδιότητα "livesIn" του οποίου τιμή είναι "Cleveland". Κάθε ένα από αυτά τα τρία στοιχεία αντιστοιχεί σε ένα URI. "Bob", παραδείγματος χάριν, πραγματικά είναι αυτό το URI: <http://www.example.com/people#bob>.

Μια συλλογή δηλώσεων ή τριπλετών καλείται γραφική παράσταση, επειδή μπορεί να αναπαρασταθεί ως συλλογή κόμβων (οι πόροι και οι κυριολεκτικές τιμές) που συνδέονται με γραμμές (οι ιδιότητες). Οποιοσδήποτε δεδομένος πόρος μπορεί να αποτελέσει το αντικείμενο περισσότερων από μιας δηλώσεων. Οτιδήποτε μπορεί να ειπωθεί σε RDFS δηλώνεται στη γλώσσα RDF, δηλαδή σε μορφή RDF δηλώσεων. Κάθε δήλωση RDFS είναι επομένως μια δήλωση RDF. Η διαφορά μεταξύ RDF και

RDFS είναι ότι οι όροι RDFS δεν έχουν καμία ιδιαίτερη σημασία σε RDF. Η σημασία των όρων RDFS καθορίζεται από τα W3C RDFS έγγραφα. Αυτό σημαίνει ότι ένας απλός RDF επεξεργαστής, που δεν είναι προγραμματισμένος να γνωρίζει RDFS, θα δημιουργήσει τη σωστή τριπλέτα όταν επεξεργάζεται μια δήλωση RDFS, αλλά δεν θα ξέρει τι να κάνει στη συνέχεια.

4.11.3. OWL

Συχνά, κατά τον ορισμό κλάσεων και ιδιοτήτων, προκύπτουν διάφορες ανάγκες, όπως να ορίσουμε την πληθυκότητα κάποιων ιδιοτήτων (πχ. "ένα αυτοκίνητο δεν μπορεί να έχει παραπάνω από τέσσερις τροχούς"), τη δυνατότητα για εναλλακτικές τιμές (πχ. "ένα αμάξι μπορεί να διαθέτει CD player"), τον συνδυασμό κλάσεων ("οι συμμετέχοντες είναι μέλη του συλλόγου και της ομάδας ποδοσφαίρου") και γενικότερα τον ακριβέστερο σχεδιασμό της οντολογίας.

Η OWL (Web Ontology Language) παρέχει ένα πιο ικανό γλωσσολογικό πλαίσιο από την RDF.

5. RDF (Resource Description Framework)

5.1. Γιατί χρειάζεται το RDF

Η XML είναι μια καθολική μεταγλώσσα για τον προσδιορισμό της σήμανσης. Παρέχει ένα ομοιόμορφο πλαίσιο, και ένα σύνολο εργαλείων για την ανταλλαγή των στοιχείων και των μεταδεδομένων μεταξύ των εφαρμογών. Εντούτοις, η XML δεν παρέχει οποιαδήποτε μέσα για τη σημασιολογία (σημασία) των δεδομένων. Παραδείγματος χάρη, δεν υπάρχει καμία έννοια η οποία συνδέεται με την “ενθυλάκωση” των ετικετών. Κάθε εφαρμογή ξεχωριστά αναλαμβάνει να ερμηνεύσει τη διαδικασία της ενθυλάκωσης. Ας υποθέσουμε ότι θέλουμε να εκφράσουμε το ακόλουθο γεγονός:

Kostas Kalogirou is a lecturer of Discrete Mathematics.

Υπάρχουν διάφοροι τρόποι αυτήν την πρόταση σε XML. Τρεις δυνατότητες είναι:

```
<course name="Discrete Mathematics">  
  <lecturer>Kostas Kalogirou</lecturer>  
</course>
```

```
<lecturer name="Kostas Kalogirou">  
  <teaches>Discrete Mathematics</teaches>  
</lecturer>
```

```
<teachingOffering>  
  <lecturer>Kostas Kalogirou</lecturer>  
  <course>Discrete Mathematics</course>  
</teachingOffering>
```

Οι πρώτες δύο αναπαραστάσεις περιλαμβάνουν ουσιαστικά αντίθετες ενθυλακώσεις, αν και αντιπροσωπεύουν τις ίδιες πληροφορίες. Έτσι δεν υπάρχει κανένας τυποποιημένος τρόπος για την ερμηνεία της ενθυλάκωσης των ετικετών.

Αν και η RDF καλείται συχνά "γλώσσα", είναι ουσιαστικά ένα μοντέλο δεδομένων. Η βασική δομική μονάδα της είναι μια τριπλέτα αντικείμενο-ιδιότητα-τιμή, αποκαλούμενη δήλωση [2,7,9]. Η προηγούμενη πρόταση για τον Kalogirou είναι μια τέτοια δήλωση. Φυσικά, ένα αφηρημένο πρότυπο δεδομένων χρειάζεται μια συγκεκριμένη σύνταξη προκειμένου να αντιπροσωπευθεί και να διαβιβαστεί, και σε RDF έχει δοθεί μια σύνταξη σε XML. Κατά συνέπεια, κληρονομεί τα οφέλη που συνδέονται με την XML. Εντούτοις, είναι σημαντικό να γίνει κατανοητό ότι άλλες συντακτικές αντιπροσωπεύσεις RDF, βασισμένες σε XML, είναι επίσης δυνατές. Η σύνταξη που βασίζεται σε XML δεν είναι ένα απαραίτητο συστατικό του προτύπου RDF.

Η RDF είναι ανεξάρτητη από τον τομέα, δεδομένου ότι καμία υπόθεση για κάποιο ιδιαίτερο τομέα δεν γίνεται [17]. Εξαρτάται από τους χρήστες να καθοριστεί η ορολογία τους σε μια γλώσσα σχημάτων αποκαλούμενη «σχήμα RDF» (RDFS). Το όνομα «σχήμα RDF» θεωρείται ανεπιτυχής επιλογή, καθώς υπονοεί ότι το σχήμα RDF έχει μια παρόμοια σχέση με την RDF όπως το σχήμα XML έχει με την XML, αλλά στην πραγματικότητα αυτό δεν ισχύει. Το σχήμα XML ορίζει τη δομή των εγγράφων XML, ενώ το σχήμα RDF καθορίζει το λεξιλόγιο που χρησιμοποιείται στα πρότυπα δεδομένων RDF. Σε RDFS μπορούμε να καθορίσουμε το λεξιλόγιο, να διευκρινίσουμε ποιες ιδιότητες ισχύουν για ποια είδη αντικειμένων και ποιες τιμές μπορούν να πάρουν και να περιγράψουν τις σχέσεις μεταξύ των αντικειμένων. Παραδείγματος χάριν, μπορούμε να γράψουμε

Lecturer is a subclass of academic staff member.

Αυτή η πρόταση σημαίνει ότι όλοι οι ομιλητές είναι επίσης μέλη του ακαδημαϊκού προσωπικού. Είναι σημαντικό να γίνει κατανοητό ότι υπάρχει μια καθορισμένη έννοια που συνδέεται με την έκφραση "is a subclass of", η οποία δεν ερμηνεύεται από κάθε εφαρμογή ξεχωριστά, αλλά είναι δεδομένη για όλο το λογισμικό επεξεργασίας RDF. Μέσω του καθορισμού της σημασιολογίας ορισμένων συστατικών, τα RDF/RDFS μας επιτρέπουν να μοντελοποιήσουμε κάποιους τομείς γνώσης.

Επεξηγούμε τη σημασία του σχήματος RDF με ένα παράδειγμα. Εξετάστε τα ακόλουθα στοιχεία XML:

```
<academicStaffMember>Dimitris Antoniou</academicStaffMember>
```

```
<professor>Michael Maher</professor>
```

```
<course name="Discrete Mathematics">  
  <isTaughtBy>Kostas Kalogirou</isTaughtBy>  
</course>
```

Ας υποθέσουμε ότι θέλουμε να συλλέξουμε όλα τα μέλη ακαδημαϊκού προσωπικού. Μια έκφραση σε Xpath είναι

```
//academicStaffMember
```

Το αποτέλεσμα είναι μόνο ο Dimitris Antoniou, το οποίο ενώ είναι σωστό από την άποψη της XML, σημασιολογικά δεν είναι ικανοποιητικό. Εμείς θα είχαμε περιλάβει επίσης τους Michael Maher και Kostas Kalogirou στην απάντηση, καθώς όλοι οι καθηγητές είναι μέλη ακαδημαϊκού προσωπικού (δηλαδή ο καθηγητής είναι μια υποκατηγορία του academicStaffMember).

Οι σειρές μαθημάτων διδάσκονται μόνο από τα μέλη ακαδημαϊκού προσωπικού. Αυτό το είδος πληροφοριών χρησιμοποιεί το σημασιολογικό πρότυπο της ιδιαίτερης περιοχής, και δεν μπορεί να αντιπροσωπευθεί σε XML ή σε RDF αλλά είναι χαρακτηριστικό της γνώσης που γράφεται στο σχήμα RDF. Κατά συνέπεια το RDFS καθιστά τις σημασιολογικές πληροφορίες κατανοητές και προσιτές στον υπολογιστή, σύμφωνα με το όραμα του σημασιολογικού Ιστού.

Για να διαδραματίσει το ρόλο της περιγραφής των δεδομένων και των μεταδεδομένων, η RDF (και άλλες πιθανές γλώσσες) πρέπει να διαθέτουν τις ακόλουθες δυνατότητες:

- Να περιγράφει τα περισσότερα είδη δεδομένων που θα είναι διαθέσιμα.
- Να περιγράφει το δομικό σχέδιο των συνόλων των δεδομένων.
- Να περιγράφει τις σχέσεις μεταξύ τμημάτων των δεδομένων.

Προς αυτή την κατεύθυνση, η RDF χρησιμοποιεί ένα απλό πρότυπο δεδομένων. Βασικά, υπάρχουν οι πόροι και οι δηλώσεις που μπορούν να γίνουν για εκείνους τους πόρους. Μια ενιαία δήλωση συνδέει δύο πόρους. Αυτές οι δηλώσεις είναι όπως

οι απλές προτάσεις που έχουν μια δομή υποκείμενο-ρήμα-αντικείμενο, όπως οι "Ο Billy ζει στο Σικάγο." Ο Billy είναι το υποκείμενο, ζει είναι το ρήμα, και το Σικάγο είναι το αντικείμενο. Αυτό, εν συντομία, είναι το πρότυπο δεδομένων. Φυσικά, οι περιπλοκές προκύπτουν όταν το πρότυπο προσαρμόζεται στις πρακτικές εφαρμογές, αλλά γενικά η RDF χρησιμοποιείται για τις απλές δηλώσεις που περιγράφουν τις πληροφορίες για συγκεκριμένα θέματα.

5.2. Τι είναι το RDF

Το πλαίσιο περιγραφής πόρων (Resource Description Framework) είναι μια γλώσσα περιγραφής πόρων βασισμένη στην XML [2,7]. Ενώ ο ορισμός "του πόρου" μπορεί να είναι αρκετά ευρύς, θα θεωρήσουμε ως πόρο ένα ηλεκτρονικό αρχείο διαθέσιμο μέσω του Ιστού. Ένας τέτοιος πόρος προσεγγίζεται μέσω ενός ομοιόμορφου εντοπιστή των πόρων (URL). Ενώ τα έγγραφα XML συνδέουν τα μεταδεδομένα με τα τμήματα ενός εγγράφου, μια χρήση του RDF είναι η δημιουργία μεταδεδομένων για ολόκληρο το έγγραφο ως αυτόνομη οντότητα. Με άλλα λόγια, αντί να χαρακτηρίσει τα εσωτερικά μέρη ενός εγγράφου, η RDF παρέχει μεταδεδομένα για τα χαρακτηριστικά ενός εγγράφου, όπως το συγγραφέα, την ημερομηνία δημιουργίας και τον τύπο. Μια ιδιαίτερα καλή χρήση της RDF είναι η περιγραφή πόρων οι οποίοι είναι "αδιαφανείς" όπως οι εικόνες ή τα ακουστικά αρχεία.

Ακολουθεί μια εφαρμογή, η οποία χρησιμοποιεί RDF για να περιγράψει μία εικόνα. Η εφαρμογή φορτώνει την εικόνα στη σωστή πλευρά και επιτρέπει την εισαγωγή δεδομένων σε μια μορφή στην αριστερή πλευρά. Τα panels στην αριστερή πλευρά επιτρέπουν να φορτωθούν RDF σχήματα για να περιγράψουν την εικόνα (τα βασικά είναι τα στοιχεία Dublin Core (www.dublincore.org) και ένα τεχνικό σχήμα με μεταδεδομένα σχετικά με τη φωτογραφική μηχανή που χρησιμοποιείται). Εκτός από την ενσωμάτωση των μεταδεδομένων στη φωτογραφία, μπορεί να γίνει και εξαγωγή των σχολιασμών RDF σε ένα εξωτερικό αρχείο, όπως φαίνεται παρακάτω:

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#"
  xmlns:s0="http://www.w3.org/2000/PhotoRDF/dc-1-0#"
  xmlns:s1="http://sophia.inria.fr/~enerbonn/rdfpiclang#"
  xmlns:s2="http://www.w3.org/2000/PhotoRDF/technical-1-0#">
```

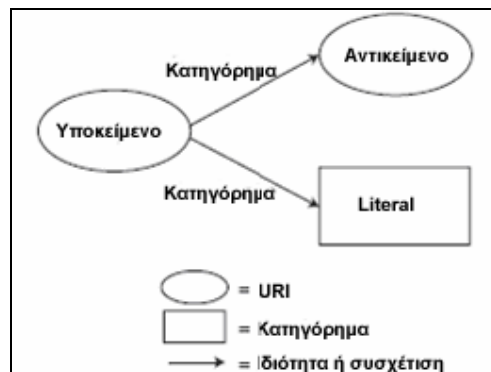
```

<rdf:Description
  rdf:about="http://www.c2i2.com/~budstv/images/shop1.jpg">
  <s0:relation>part-of Store Front</s0:relation>
  <s0:type>image</s0:type>
  <s0:format>image/jpeg</s0:format>
  <s1:xml:lang>en</s1:xml:lang>
  <s0:description>Buddy Belden's work bench for
  TV/VCR repair</s0:description>
  <s2:camera>Kodak EasyShare</s2:camera>
  <s0:title>TV Shop repair bench</s0:title>
</rdf:Description>
</rdf:RDF>

```

Το πρώτο πράγμα που παρατηρούμε είναι η χρήση των namespaces σε όλα τα στοιχεία. Αρχικά δηλώνονται τέσσερα namespaces. Διευκρινίζεται ότι το παρόν έγγραφο είναι ένα έγγραφο RDF. Ένα έγγραφο RDF περιέχει μία ή περισσότερες "περιγραφές" πόρων. Μια περιγραφή είναι ένα σύνολο δηλώσεων για έναν πόρο. Η ιδιότητα `rdf:about` αναφέρεται στον πόρο που περιγράφεται. Η ιδιότητα `rdf:about` δείχνει το URL ενός JPEG, της ονομαζόμενης εικόνας `shop1.jpg`. Η ιδιότητα `rdf:about` είναι πολύ σημαντική για την κατανόηση της RDF επειδή όλοι οι πόροι που περιγράφονται σε RDF πρέπει να αναφέρονται σε ένα URI. Τα παιδιά του στοιχείου `description` αποτελούν όλα ιδιότητες του πόρου που περιγράφεται. Δύο ιδιότητες είναι **bolded**, οι τιμές των οποίων αποθηκεύονται ως περιεχόμενο. Εν περιλήψει, αναδεικνύεται μία σύνταξη όπου περιγράφουμε έναν πόρο, τις ιδιότητες των πόρων, και τις τιμές των ιδιοτήτων.

Το πρότυπο RDF καλείται συχνά "τριπλέτα" επειδή έχει τρία μέρη, τα οποία περιγράφονται ως γραμματικά μέρη μιας πρότασης: υποκείμενο, κατηγορημα και αντικείμενο. Το ακόλουθο σχήμα επιδεικνύει τα στοιχεία του προτύπου και τα σύμβολα που συνδέονται με τα στοιχεία όταν τα παριστάνουμε γραφικά.



Τα βασικά στοιχεία RDF είναι τα ακόλουθα:

Υποκείμενο: Στη γραμματική, αυτό είναι το ουσιαστικό ή η ονομαστική φράση που υποδεικνύει αυτόν που ενεργεί. Στην πρόταση "η επιχείρηση πωλεί μπαταρίες," το υποκείμενο είναι "η επιχείρηση." Στη λογική, είναι ο όρος για τον οποίο κάτι βεβαιώνεται. Σε RDF, είναι ο πόρος που περιγράφεται από το κατηγορημα και το αντικείμενο, τα οποία ακολουθούν. Επομένως, σε RDF, θέλουμε ένα URI για να αντιπροσωπεύσουμε τη μοναδική έννοια "επιχείρηση", όπως <http://www.business.org/ontology/#company>, για να δείξουμε ότι αναφερόμαστε σε μια ιδιόκτητη επιχείρηση.

Ένας πόρος RDF αντιπροσωπεύει είτε τους ηλεκτρονικούς πόρους, όπως τα αρχεία, είτε τις έννοιες, όπως "το πρόσωπο." Πρόκειται για "κάτι που έχει ταυτότητα."

Κατηγορημα: Στη γραμματική, αυτό είναι το μέρος μιας πρότασης που τροποποιεί το θέμα και περιλαμβάνει τη φράση ρήματος. Επιστρέφοντας στην πρότασή μας "η επιχείρηση πωλεί τις μπαταρίες," το κατηγορημα είναι η φράση "πωλεί τις μπαταρίες." Με άλλα λόγια, το κατηγορημα μας λέει κάτι για το υποκείμενο. Σε RDF, ένα κατηγορημα είναι μια σχέση μεταξύ του υποκειμένου και του αντικειμένου. Κατά συνέπεια, σε RDF, θα καθορίζαμε ένα μοναδικό URI για το "πωλεί" όπως "http://www.business.org/ontology/#sells".

Αντικείμενο: Στη γραμματική, αυτό είναι ένα ουσιαστικό όπου ενεργεί το ρήμα. Επιστρέφοντας στην πρότασή μας "η επιχείρηση πωλεί τις μπαταρίες," το αντικείμενο είναι το ουσιαστικό "μπαταρίες." Στη λογική, ένα αντικείμενο ενεργεί στο κατηγορημα. Σε RDF, ένα αντικείμενο είναι είτε ένας πόρος, στον οποίο αναφέρεται το κατηγορημα, είτε μια κυριολεκτική τιμή. Στο παράδειγμά μας, θα καθορίζαμε ένα μοναδικό URI για "τις μπαταρίες" όπως "http://www.business.org/ontology/#batteries".

Δήλωση: Σε RDF, είναι ο συνδυασμός των προηγούμενων τριών στοιχείων, υποκείμενο, κατηγορημα, και αντικείμενο (ως ενιαία μονάδα). Το αντικείμενο μπορεί να αντιπροσωπευθεί από έναν πόρο ή από μια κυριολεκτική τιμή.

Πρέπει να τονίσουμε ότι οι πόροι σε RDF πρέπει να προσδιορίζονται από τον πόρο IDs, που είναι URIs. Αυτό είναι σημαντικό έτσι ώστε μια μοναδική έννοια να μπορεί να προσδιοριστεί σαφώς μέσω μιας συνολικά μοναδικής ταυτότητας. Αυτό είναι μια

βασική διαφορά μεταξύ της σημασιολογίας και της σύνταξης. Η συντακτική έννοια των λέξεων είναι συχνά διαφορούμενη. Αντιστοιχίζοντας ένα μοναδικό URI σε μία έννοια, μπορούμε πάντα να αναφερθούμε στην έννοια αυτή με την σωστή σημασία της κάθε φορά.

5.3. Άλλα χαρακτηριστικά RDF

Τα παρακάτω RDF γνωρίσματα επιτρέπουν την αύξηση του composure των δηλώσεων. Δύο κύριες κατηγορίες χαρακτηριστικών γνωρισμάτων επιτρέπουν κάτι τέτοιο: το πρότυπο του **container** και το πρότυπο **reification** (πραγματοποίηση δηλώσεων για δηλώσεις). Το πρότυπο container χρησιμοποιείται σε ομάδες πόρων ή τιμών, ενώ το reification επιτρέπεται στις υψηλότερου επιπέδου δηλώσεις, έτσι ώστε να λάβουμε γνώση για άλλες δηλώσεις. Και τα δύο χαρακτηριστικά γνωρίσματα προσθέτουν κάποια πολυπλοκότητα. Ακολουθούν δύο βασικά παραδείγματα για την κατανόησή τους.

5.3.1. Container

Χρειαζόμαστε ένα container για να διαμορφώσουμε την πρόταση "Τα άτομα στη συνεδρίαση ήταν ο Joe, ο Bob, η Susan, και ο Ralph." Για να εκφράσουμε αυτή την πρόταση σε RDF, δημιουργούμε ένα container, αποκαλούμενο bag, για τα αντικείμενα στη δήλωση, όπως φαίνεται στη συνέχεια.

```
<rdf:RDF
  xmlns:ex='http://www.example.org/sample#'
  xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
  <rdf:Description rdf:about='ex:meeting'>
    <ex:attendees>
      <rdf:Bag rdf:ID="people">
        <rdf:li rdf:resource='ex:Joe' />
        <rdf:li rdf:resource='ex:Bob' />
        <rdf:li rdf:resource='ex:Susan' />
        <rdf:li rdf:resource='ex:Ralph' />
      </rdf:Bag>
    </ex:attendees>
  </rdf:Description>
</rdf:RDF>
```

RDF bag container

Βλέπουμε ένα στοιχείο `rdf:Description` (ένα υποκείμενο), ένα κατηγορημα (συμμετέχοντες), και ένα αντικείμενο, το οποίο είναι μια συλλογή πόρων (`bag`), όπου σε κάθε στοιχείο της τελευταίας γίνεται αναφορά από ένα στοιχείο `"rdf:li"` ή `"list item"`.

Σύμφωνα με τα παραπάνω, η "συνεδρίαση" "παρακολουθείται" από "τους ανθρώπους" και οι άνθρωποι είναι ένας τύπος `bag`, τα μέλη του οποίου χαρακτηρίζονται ως μέλη με ένα `rdf: _ #` κατηγορημα.

Τα RDF containers είναι διαφορετικά από τα XML containers δεδομένου ότι είναι πιο ξεκάθαρα. Είναι η ίδια περίπτωση με τις σχέσεις μεταξύ των στοιχείων, οι οποίες είναι επίσης υπονοούμενες σε XML, ενώ τέτοιες σχέσεις είναι ξεκάθαρες σε RDF. Αυτή η ξεκάθαρη διαμόρφωση των containers και των σχέσεων είναι μια προσπάθεια να αφαιρεθεί η ασάφεια από τα πρότυπά μας έτσι ώστε οι υπολογιστές να μπορούν να ενεργήσουν στη θέση των χρηστών.

Τρεις τύποι RDF containers είναι διαθέσιμοι:

- **Bag**: ένα στοιχείο `rdf:bag` χρησιμοποιείται για να δείξει μια άτακτη συλλογή. Τα αντίγραφα επιτρέπονται στη συλλογή. Ένα παράδειγμα που υποδεικνύει το πότε θα έπρεπε να χρησιμοποιηθεί αυτός ο τύπος θα ήταν όταν όλα τα μέλη της συλλογής υποβάλλονται σε επεξεργασία χωρίς ανησυχία για τη σειρά που ακολουθείται.
- **Sequence** (ακολουθία): ένα στοιχείο `rdf:seq` χρησιμοποιείται για να δείξει μια διατεταγμένη συλλογή (μια "ακολουθία" στοιχείων). Τα αντίγραφα επιτρέπονται στη συλλογή. Ένας λόγος για να χρησιμοποιηθεί μια ακολουθία θα ήταν να συντηρηθεί η αλφαβητική σειρά των στοιχείων. Ένα άλλο παράδειγμα θα ήταν να υποβληθούν σε επεξεργασία τα στοιχεία σύμφωνα με τη σειρά με την οποία προστέθηκαν στο έγγραφο.
- **Alternate** (εναλλαγή): ένα στοιχείο `rdf:alt` χρησιμοποιείται για να δείξει μια επιλογή μεταξύ πολλαπλών τιμών ή πόρων. Αυτό σε XML αναφέρεται ως επιλογή. Μερικά παραδείγματα θα ήταν μια επιλογή της μορφής μιας εικόνας (JPEG, GIF, bmp) ή οποτεδήποτε επιθυμούμε να περιορίσουμε μια τιμή σε ένα περιορισμένο σύνολο τιμών.

Τώρα που έχουμε προσθέσει την ιδέα των "συλλογών" στις δηλώσεις μας, χρειαζόμαστε έναν τρόπο για να κάνουμε τις δηλώσεις είτε για τη συλλογή είτε για τα μεμονωμένα μέλη της συλλογής. Μπορούμε να κάνουμε τις δηλώσεις για τη

συλλογή με την ιδιότητα rdf:ID στο container. Η παραγωγή των δηλώσεων για τα μεμονωμένα μέλη είναι η ίδια με την παραγωγή οποιασδήποτε άλλης δήλωσης, αναφέροντας απλά τον πόρο στη συλλογή ως το αντικείμενο της δήλωσής μας.

5.3.2. Reification

Ενώ τα containers έχουν επιπτώσεις στη διαμόρφωση μιας ενιαίας δήλωσης (παραδείγματος χάριν, ένα αντικείμενο που γίνεται μια συλλογή από τιμές), το reification μας επιτρέπει να μεταχειριστούμε μια δήλωση ως αντικείμενο μιας άλλης δήλωσης. Αυτό αναφέρεται συχνά ως "παραγωγή δηλώσεων για τις δηλώσεις" και καλείται reification. Ακολουθεί ένα απλό παράδειγμα του reification.

```
@prefix : <http://example.org/onto#>.
@prefix earl: <http://www.w3.org/2001/03/earl/0.95#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix dc: <http://purl.org/dc/elements/1.1/>.
```

```
:Jane earl:asserts
  [ rdf:subject :MyPage;
    rdf:predicate earl:passes;
    rdf:object "Accessibility Tests" ];
earl:email <mailto:Jane@example.org>;
earl:name "Jane Jones".
```

```
:MyPage
  a earl:WebContent;
  dc:creator <http://example.org/onto/person/Mary/>.
```

Παράδειγμα reification.

Σύμφωνα με το παράδειγμα αυτό, η Jane έχει εξετάσει τις ιστοσελίδες της Mary και βεβαιώνει ότι περνά τα τεστ. Το τμήμα σχετικά με το reification είναι η πρόταση με το υποκείμενο, το κατηγορημα, και τα μέρη αντικειμένου που είναι το αντικείμενο του "βεβαιώνει." Το ίδιο παράδειγμα σε RDF έχει ως εξής:

```
<rdf:RDF
  xmlns:dc='http://purl.org/dc/elements/1.1/'
  xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
  xmlns:earl='http://www.w3.org/2001/03/earl/0.95#'>
  <rdf:Description rdf:about='http://example.org/onto#Jane'>
    <earl:asserts rdf:parseType='Resource'>
```

```

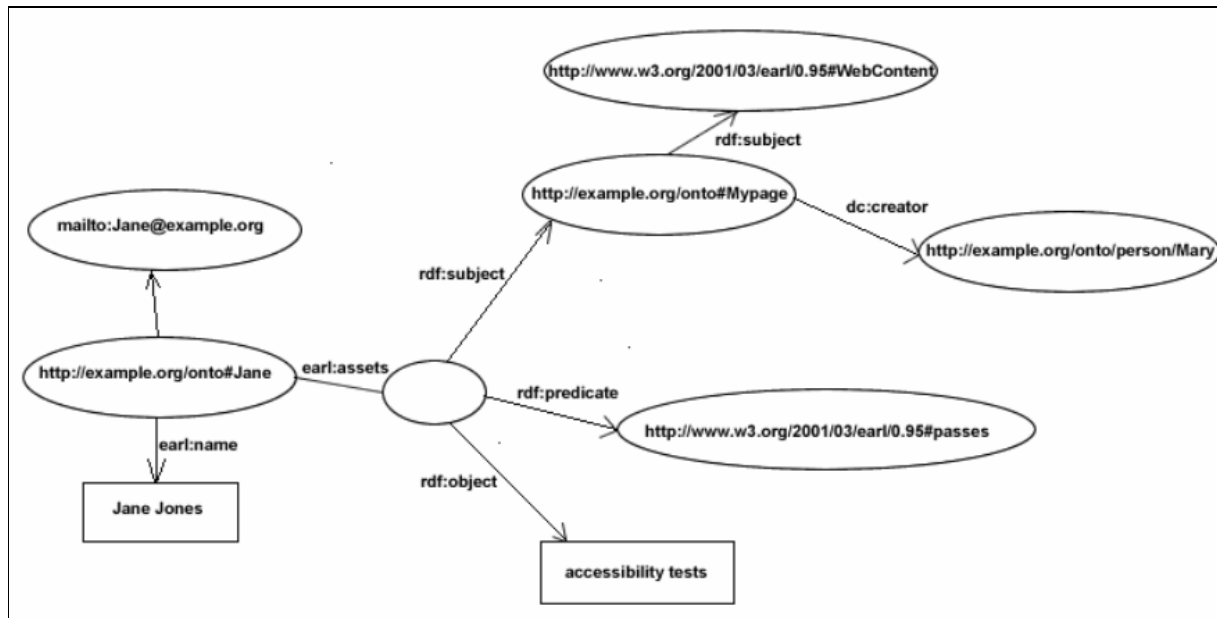
<rdf:subject>
  <earl:WebContent
    rdf:about='http://example.org/onto#MyPage'>
    <dc:creator
      rdf:resource='http://example.org/onto/person/Mary
    /' />
  </earl:WebContent>
</rdf:subject>
<rdf:predicate>
  rdf:resource='http://www.w3.org/2001/03/earl/0.95#passes' />
<rdf:object>Accessibility Tests</rdf:object>
</earl:asserts>
<earl:email rdf:resource='mailto:Jane@example.org' />
<earl:name>Jane Jones</earl:name>
</rdf:Description>
</rdf:RDF>

```

Παράδειγμα RDF reification

Η μέθοδος reification για δηλώσεις σε RDF αναφέρεται στη διαμόρφωση της δήλωσης ως πόρου διευκρινίζοντας το υποκείμενο, το κατηγορημα, το αντικείμενο και τον τύπο της δήλωσης. Στη συνέχεια, μπορούν να γίνουν δηλώσεις για τη διαμορφωμένη δήλωση. Ωστόσο, το reification δεν μπορεί να εφαρμοστεί για όλους τους τύπους δεδομένων. Είναι ευκολότερο να αντιμετωπιστούν οι δηλώσεις ως γεγονότα.

Το παρακάτω σχήμα παριστάνει τη γραφική παράσταση μιας μορφοποιημένης πρότασης. Η πρόταση αντιμετωπίζεται ως ενιαία οντότητα μέσω ενός ανώνυμου κόμβου. Ο ανώνυμος κόμβος είναι δίπλα σε ένα στοιχείο Description χωρίς ιδιότητα rdf:about. Η ιδιότητα rdf:parseType στον κώδικα σημαίνει ότι το περιεχόμενο του στοιχείου αναλύεται παρόμοια με ένα νέο στοιχείο Description.



Κατά γενική ομολογία, η μέθοδος reification δεν είναι απλή. Μια εφαρμογή κοινής λογικής του reification αφορά σε σχολιασμούς της εργασίας άλλων ανθρώπων. Οι σχολιασμοί, από τη φύση τους, είναι δηλώσεις για τις δηλώσεις κάποιου άλλου. Έτσι, σαφώς, το reification σχετίζεται με κάτι τέτοιο.

Στην πραγματικότητα, μερικές τρέχουσες εφαρμογές του σημασιολογικού Ιστού αποβάλλουν ρητά το reification από τις βάσεις γνώσεών τους για να μειώσουν την πολυπλοκότητα. Η πολυπλοκότητα βλάπτει την υιοθέτηση, και η υιοθέτηση RDF από τους επικρατόντες υπεύθυνους για την ανάπτυξη είναι σημαντικά πιο αργή από άλλες W3C τεχνολογίες.

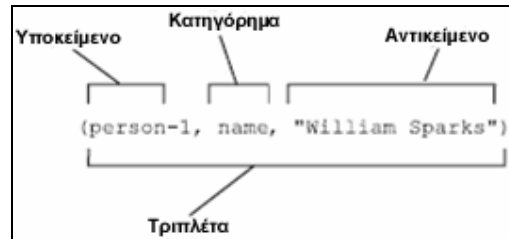
5.4. Ορολογία RDF

5.4.1. Τι περιλαμβάνει

5.4.1.1 Ορολογία

Μία δήλωση καλείται μερικές φορές τριπλέτα επειδή, όπως ήδη ορίσαμε, αποτελείται από τρία μέρη. Το υποκείμενο μιας δήλωσης καλείται subject (υποκείμενο), το αντίστοιχο ενός ρήματος καλείται predicate (κατηγορημα), και το υπόλοιπο μέρος

καλείται object (αντικείμενο). Το ακόλουθο διάγραμμα απεικονίζει τη δομή μιας τριπλέτας RDF:



Η τιμή του κατηγορήματος μπορεί να είναι μια απλή τιμή, όπως ένας συνηθισμένος αριθμός ή μια σειρά χαρακτήρων όπως το “William Sparks” στο διάγραμμα. Τέτοιες τιμές καλούνται literals. Επίσης, μπορεί να είναι είτε κυριολεκτική είτε ένας άλλος πόρος, ανάλογα με την περίπτωση. Στην RDF υπάρχει τρόπος να προσδιορίσουμε εάν μια κυριολεκτική τιμή πρέπει να ακολουθεί έναν συγκεκριμένο τύπο δεδομένων, που σημαίνει ότι προορίζεται να είναι, παραδείγματος χάρη, ένας ακέραιος αριθμός ή ένα τμήμα XML.

5.4.1.2 Προσδιορισμός των resources

Η RDF, για να είναι ευρέως χρησιμοποιήσιμη, πέρα από τον Ιστό, πρέπει να είναι σε θέση να προσδιορίσει τα όσα περιγράφει κατά τρόπο τυποποιημένο, ευρέως διαδεδομένο, έτσι ώστε το πλήθος των συστημάτων στον Ιστό να μπορεί επίσης να αναφερθεί σε αυτά. Τα παλαιότερα συστήματα, όπως οι συμβατικές βάσεις δεδομένων, δεν έχουν κανέναν τυποποιημένο τρόπο να προσδιορίσουν τα θέματά τους στα συστήματα και στα δίκτυα.

Για τον προσδιορισμό των πόρων - για τους οποίους η RDF κάνει δηλώσεις- χρησιμοποιούνται όπως είπαμε URI (ομοίομορφο προσδιοριστικό των πόρων) αναφορές. Ένα URI μπορεί να χρησιμοποιηθεί για να προσδιορίσει μια έννοια, κάτι απτό, ή ένα κομμάτι των στοιχείων που μπορούν να ανακτηθούν πέρα από ένα δίκτυο. Μια αναφορά URI είναι ένα URI συν τους προαιρετικούς χαρακτήρες, όπως το αποκαλούμενο *fragment* προσδιοριστικό (το μέρος που ακολουθεί το σύμβολο # μετά από ένα URI, ενδεχομένως). Η RDF διαθέτει κανόνες για το πώς να κατασκευάσει τα συσχετιζόμενα URIs, έτσι ώστε να μπορούν να χρησιμοποιηθούν βολικά όταν ανταλλάσσεται ένα στοιχείο RDF.

Τα ονόματα δεν είναι μοναδικά - υπάρχουν πολλά άτομα με κοινό ονοματεπώνυμο, αλλά δεν είναι το ίδιο πρόσωπο ακριβώς επειδή έχουν το ίδιο όνομα. Αντιθέτως, κάτι

δεδομένο μπορεί να έχει πολλά ονόματα. Για να προσδιορίσουμε έναν πόρο, ένα URI είναι συνδεδεμένο με τον πόρο αυτό, και με κανέναν άλλο πόρο. Πώς αυτό πραγματοποιείται δεν διευκρινίζεται και δεν μπορεί πιθανώς να διευκρινιστεί γενικά. Ένας τρόπος είναι να γραφτεί η συσχέτιση σε ένα έγγραφο, όπως μια προδιαγραφή. Εντούτοις, υπάρχει πάντα η δυνατότητα η περιγραφή στο έγγραφο να μην είναι αρκετά σαφής ώστε να προσδιορίσει πλήρως αυτό που περιγράφεται. Και εάν πρόκειται για μια έννοια αντί ενός απτού πράγματος, η σημασία της έννοιας δεν μπορεί να καταστεί σαφής αρκετά. Φυσικά, αυτό το πρόβλημα δεν είναι μοναδικό σε RDF ή στον σημασιολογικό Ιστό.

Μπορούμε επίσης να προσδιορίσουμε έναν πόρο από τις ιδιότητες και τις σχέσεις του με άλλους πόρους. Μία ομάδα ποδοσφαιρού, για παράδειγμα, που είχε μόνο έναν γενικό διευθυντή κατά τη διάρκεια μίας συγκεκριμένης εποχής, εφόσον ήταν γνωστό ότι ένα πρόσωπο ήταν ο γενικός διευθυντής της κατά τη διάρκεια του συγκεκριμένου χρονικού διαστήματος, εκείνο το άτομο θα προσδιοριζόταν μεμονωμένα ακόμα κι αν το όνομά του δεν αναφερόταν. Προφανώς, μερικές ιδιότητες και σχέσεις είναι πιο χρήσιμες για την ταυτότητα από άλλες.

Με οποιονδήποτε τρόπο και αν καθιερώνεται η ταυτότητα ενός πόρου, η RDF χρησιμοποιεί την συγκεκριμένη αναφορά URI ως μοναδική σειρά χαρακτήρων, ανεξαρτήτως αν αυτή η τελευταία είναι αναγνώσιμη από τον χρήστη ή όχι. Όταν η συσχέτιση ενός URI με τον προσδιορισμένο πόρο της περιγράφεται σε ένα έγγραφο, με οποιονδήποτε τρόπο, μια ερώτηση προκύπτει φυσικά: Πώς μπορεί αυτό το έγγραφο να βρεθεί; Η RDF δεν παρέχει καμία τυποποιημένη μέθοδο.

Κάποιοι έχουν συμφωνήσει σε μια άτυπη σύμβαση ότι το URI πρέπει να είναι υπό μορφή HTTP URL που μπορεί να χρησιμοποιηθεί για να ανακτήσει το έγγραφο καθορισμού από έναν ιστοχώρο. Όταν αυτό γίνεται, είναι σημαντικό να είναι σαφές το γεγονός ότι το URL που χρησιμοποιείται για να προσδιορίσει έναν πόρο δεν δείχνει ότι το ίδιο το περιγραφικό έγγραφο είναι ο ίδιος ο πόρος. Το URI, ως προσδιοριστικό, είναι ακριβώς μια αυθαίρετη σειρά χαρακτήρων, και οποιαδήποτε χρησιμότητα που μπορεί να έχει για τη λήψη ενός εγγράφου δεν ορίζεται σε RDF.

Ακολουθεί ένα παράδειγμα για να καταστήσουμε τα παραπάνω πιο σαφή. Η σύσταση RDF περιέχει μια έννοια αποκαλούμενη δήλωση. Ο τύπος ενός πόρου, που είναι μια δήλωση RDF, προσδιορίζεται χρησιμοποιώντας το URI του:

<http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement>

Σε αυτό το URI βρίσκουμε ένα έγγραφο, το οποίο περιέχει το ακόλουθο RDF:

```
<!--
```

```
This is the RDF Schema for the RDF data model as described in  
The Resource Description Framework (RDF) Model and Syntax  
Specification http://www.w3.org/TR/REC-rdf-syntax -->
```

```
<s:Class rdf:ID="Statement"  
  s:comment="A triple consisting of a predicate, a subject,  
  and an object." />  
<s:Class rdf:ID="Property"  
  s:comment="A name of a property, defining specific meaning  
  for the property" />
```

Το συγκεκριμένο URI δεν είναι απαραίτητο να δείξει ένα έγγραφο. Το γεγονός ότι το κάνει είναι χρήσιμο (σε αυτήν την περίπτωση, η "# statement", θα ανάγκαζε το τονισμένο τμήμα να επιδειχθεί σε έναν browser που καταλαβαίνει XML).

5.4.1.3 Οι ανώνυμοι πόροι

Η RDF δεν απαιτεί κάθε πόρος σε μια δήλωση να προσδιορίζεται με μια αναφορά URI. Ένας πόρος χωρίς μια προσδιορίζουσα αναφορά URI είναι ανάλογος με την αναφορά σε ένα πρόσωπο όπως "εκείνος ο άνδρας" ή "μια γυναίκα με ένα κόκκινο αυτοκίνητο." Δεν θα ήταν βολικό να παραστήσουμε σύνθετα σύνολα στοιχείων με απλές τριπλέτες RDF χωρίς τη χρήση των ανώνυμων πόρων, επειδή θα αναγκαζόμασταν να εφεύρουμε περιττά προσδιοριστικά. Τα περισσότερα προγράμματα υπολογιστών δημιουργούν εσωτερικά τα προσδιοριστικά τους για αυτούς τους πόρους, αλλά αυτά δεν είναι απαραίτητο να γίνουν γνωστά έξω από την εφαρμογή. Οι ανώνυμοι πόροι καλούνται επίσης κενοί κόμβοι (blank nodes) ή b-nodes.

5.5. Παραλληλισμός: RDF και συμβατικές βάσεις δεδομένων

Το πρότυπο RDF με τους πόρους και τις τριπλέτες μπορεί να φανεί πολύ διαφορετικό από το πρότυπο που χρησιμοποιείται στις συμβατικές σχεσιακές βάσεις δεδομένων, αλλά η διαφορά δεν είναι τόσο μεγάλη όσο φαίνεται αρχικά. Στις συμβατικές βάσεις δεδομένων, τα δεδομένα οργανώνονται συνήθως σε πίνακες όπως αυτός που ακολουθεί.

Όνομα	Τηλέφωνο	Email	Πόλη	Περιοχή
William Sparks	423 344-4452	wsparks@mail.com	Springfiels	MA
Constance Tim	321 299-9887	csim@mailto.com	Chicago	IL

Κάθε σειρά του πίνακα έχει την ίδια δομή με κάθε άλλη, και η σειρά αντιπροσωπεύει μια ιδιαίτερη σχέση μεταξύ όλων των στοιχείων εκείνης της σειράς. Κάθε στήλη περιέχει ένα γεγονός για ένα ιδιαίτερο πρόσωπο, και κάθε σειρά αντιπροσωπεύει ένα από εκείνα τα πρόσωπα.

Μια συλλογή των στοιχείων που ανήκουν μαζί καλείται συχνά tuple. Ακριβέστερα, tuple είναι ένας διαταγμένος κατάλογος τιμών. Αν και τα στοιχεία σε μια σειρά μιας σχεσιακής βάσης δεδομένων δεν διατάσσονται σε γενικές γραμμές, στην πράξη διατάσσονται έτσι ώστε να μπορούμε να ξέρουμε ποιο στοιχείο αντιστοιχεί σε ποια στήλη. Με άλλα λόγια, η πρώτη σειρά του πίνακα μπορεί να αντιπροσωπευθεί από το tuple ("William Sparks", "435 555-6789", "wsparks@mailto.com", "Springfield", "MA"), εφ' όσον ξέρουμε ότι το πρώτο στοιχείο του tuple είναι το όνομα, το δεύτερο είναι ο τηλεφωνικός αριθμός και ούτω καθ'εξής.

Σε έναν καλά σχεδιασμένο πίνακα, όλα τα γεγονότα σε μια σειρά εξαρτώνται από ένα στοιχείο, αποκαλούμενο πρωτεύον κλειδί, το οποίο πρέπει να είναι μοναδικό μέσα σε έναν πίνακα. Ο προφανής υποψήφιος είναι εδώ το όνομα του προσώπου. Εάν επρόκειτο να βάλουμε αυτό το στοιχείο στις λέξεις, θα διάβαζε κάτι παρεμφερές: Υπάρχει ένα πρόσωπο του οποίου το όνομα είναι William Sparks, ο τηλεφωνικός αριθμός είναι 435 555-6789, η διεύθυνση ηλεκτρονικού ταχυδρομείου είναι wsparks@mailto.com, ζει στο Σπρίνγκφιλντ, και του οποίου ο κρατικός κώδικας είναι MA.

Ωστόσο, μερικά πράγματα δεν είναι σωστά. Αρχικά, είπαμε ότι κάθε στοιχείο πρέπει να εξαρτηθεί μόνο από το πρωτεύον κλειδί, αλλά τα πεδία της πόλεως και του

κράτους δεν είναι απολύτως ανεξάρτητα. Στην πραγματικότητα, πρέπει πιθανώς να υπάρξει ένα πεδίο διευθύνσεων, το οποίο θα περιλαμβάνει την πόλη και το κράτος. Αυτές οι λεπτομέρειες, αν και σημαντικές στο σχέδιο βάσεων δεδομένων, δεν πειράζουν εδώ.

Ας εξετάσουμε ένα άλλο σημείο. Το όνομα ενός προσώπου μπορεί να αλλάξει, και μπορούν να υπάρξουν διαφορετικές μορφές του ίδιου ονόματος και, φυσικά, περισσότερα από ένα πρόσωπα μπορούν να έχουν το ίδιο όνομα. Έτσι, θα ήταν καλύτερο να προσδιοριστεί ένα πρόσωπο που χρησιμοποιεί κάτι πιο οριστικό, όπως έναν αυθαίρετο αλλά μοναδικό αριθμό ταυτότητας, ή έναν εθνικό αριθμό αναγνώρισης. Αυτός ο αριθμός θα αντιπροσώπευε το πρόσωπο. Ο πίνακας που ακολουθεί παρουσιάζει αυτήν την αλλαγή.

ID	Όνομα	Τηλέφωνο	Email	Πόλη	Περιοχή
person-1	William Sparks	423 344-4452	wsparks@mailt.com	Springfiels	MA
person-2	Constance Tim	321 299-9887	csim@mailto.com	Chicago	IL

Τώρα λοιπόν έχουμε: Πρόσωπο-1 έχει το όνομα William Sparks, αριθμό τηλεφώνου 423 344-4452... Η διαφορά μεταξύ των δύο εκδόσεων βρίσκεται στην ταυτότητα του προσώπου. Η έκδοση 1 απαριθμεί τα στοιχεία για κάποιο πρόσωπο που συμβαίνει να έχει το όνομα William Sparks. Η έκδοση 2 απαριθμεί τα στοιχεία συγκεκριμένα για το πρόσωπο-1. Στην έκδοση 1, θα έχουμε πρόβλημα εάν το όνομα του προσώπου αλλάξει, δεδομένου ότι οποιαδήποτε στοιχεία αναφέρονται στο αρχικό όνομα δεν θα ανήκουν σε κάποιον. Βέβαια, και οι δύο εκδόσεις μπορούν να αντιπροσωπευθούν σε RDF, η οποία έχει μόνο πόρους και τριπλέτες. Για να παραστήσουμε αυτό το στοιχείο σε τριπλέτα, πρέπει να απλοποιήσουμε τον πίνακα σε όσο το δυνατό μικρότερα κομμάτια.

Μια σειρά ενός πίνακα δύο-στηλών μπορεί αποτελέσει μία τριπλέτα. Μπορούμε να μετατρέψουμε τον πίνακά μας σε σύνολο πινάκων δύο στηλών, όπως φαίνεται στη συνέχεια.

Person ID	Όνομα
person-1	William Sparks
person-2	Constance Tim

Person ID	Τηλέφωνο
person-1	423 344-4452

Αυτό είναι νόμιμο εφ' όσον τα γεγονότα (οι στήλες του αρχικού πίνακα) είναι αληθινά ανεξάρτητα το ένα από το άλλο. Σύμφωνα με το κλασικό μοτίβο: (πρόσωπο) - (τύπος του γεγονότος) - (τιμή του γεγονότος), σε αυτήν την περίπτωση, ο τύπος γεγονότος είναι "όνομα" για τον πρώτο πίνακα και "τηλέφωνο" για τον δεύτερο.

Γενικά, φυσικά, το αριστερό στοιχείο μπορεί να είναι οτιδήποτε, όχι μόνο ένα πρόσωπο. Σε RDF, όπως έχουμε αναφέρει, έχουμε πόρους. Έτσι, μπορούμε να γράψουμε μία τριπλέτα RDF όπως (πόρος) - (τύπος του γεγονότος) - (τιμή του γεγονότος), όπως μια σειρά του πίνακα δύο-στηλών αντιπροσωπεύει. Παίρνει μια συλλογή για να αντιπροσωπεύσει το περιεχόμενο μιας βάσης δεδομένων. Τα γεγονότα σε οποιοδήποτε καλά σχεδιασμένο σύνολο πινάκων μπορούν να ξαναγραφούν υπό μορφή τριπλέτας RDF. Θα επιθυμούσαμε να μετατρέψουμε το πρωτεύον κλειδί σε μια αναφορά URI, για να προσδιορίσει το υποκείμενο σύμφωνα με την RDF, αν και σε μερικές περιπτώσεις θα ήταν εφικτό να χρησιμοποιήσουμε ανώνυμους κόμβους για να αντιπροσωπεύσουμε τις σειρές.

Σε αυτή την περίπτωση - εάν η RDF και οι συγγενικοί πίνακες είναι υπό κάποια έννοια ισοδύναμα - τότε ποιος ο λόγος να χρησιμοποιούμε RDF; Η βάση δεδομένων θα έχει πιθανώς καλύτερη απόδοση εάν τα δεδομένα είναι σε κανονική μορφή, αλλά με την RDF δεν είναι απαραίτητο να ισχύει κάτι τέτοιο. Μπορούμε πάντα να προσθέσουμε νέες τριπλέτες που αντιπροσωπεύουν νέες πληροφορίες. Εάν έχουν διαφορετικό κατηγορήμα (ο τύπος στοιχείων που αντιπροσωπεύεται από το όνομα μιας στήλης), δεν θα αρμόσουν σε έναν υπάρχοντα πίνακα. Αυτό θα προκαλούσε μερικά προβλήματα για μια συμβατική βάση δεδομένων αλλά κανένα για δεδομένα σε μορφή RDF. Έτσι, η RDF έχει το πλεονέκτημα της ευελιξίας. Επιπλέον, μπορούμε να κάνουμε δηλώσεις για τα κατηγορήματα καθώς επίσης και δηλώσεις για τις τιμές των ιδιοτήτων, επειδή οι τύποι κατηγορήματος είναι επίσης πόροι σε RDF. Με άλλα λόγια, η RDF μπορεί να περιγράψει πολλές από τις ιδιότητες των δομών δεδομένων.

Είναι όμως μία τριπλέτα ικανή να αντιπροσωπεύσει όλα τα δεδομένα που μας ενδιαφέρουν; Ας εξετάσουμε την περίπτωση μιας συμβατικής βάσης δεδομένων. Όλα τα στοιχεία σε μια σειρά ανήκουν κανονικά μαζί, ενώ η οργάνωσή τους σε τριπλέτες φαίνεται να χάνει αυτή την σύνδεση. Άλλωστε, ένας πόρος μπορεί να έχει εκατοντάδες συσχετιζόμενα γεγονότα, αλλά μόνο πέντε από αυτά μπορεί να ανήκουν σε μια δεδομένη σειρά ενός δεδομένου πίνακα. Μερικές γλώσσες

προγραμματισμού, συμπεριλαμβανομένης της Prolog, που σχεδιάζεται ειδικά για το λογικό προγραμματισμό, μπορούν να χρησιμοποιήσουν tuples οποιουδήποτε μήκους για να συλλάβουν τα γεγονότα και τις ιδιότητες. Γιατί να περιορίσουμε την RDF σε τριπλέτες λοιπόν;

Ο λόγος είναι η απλότητα: Οι τριπλέτες είναι μικρότερες και απλούστερες. Οι δομές δεδομένων στα πλαίσια των προγραμμάτων μπορούν να είναι απλούστερες, επειδή το μέγεθός τους θα είναι πάντα το ίδιο. Οι δομές δεδομένων είναι σχετικά εύκολα προσβάσιμες και επεξεργάσιμες. Το αποτέλεσμα μπορεί μερικές φορές να είναι πρόσθετη πολυπλοκότητα. Για να συλλάβουμε την ουσία μιας σειράς σε έναν πίνακα βάσης δεδομένων, χρειαζόμαστε έναν πόρο για να συλλέξουμε όλες τις μεμονωμένες τριπλέτες και να δηλώσουμε ότι όλες αυτές ανήκουν στη σειρά. Αυτή η διαδικασία μπορεί να μην είναι εύκολη. Ευτυχώς, δεν είναι πάντα απαραίτητο να προσδιοριστούν οι σειρές που περιείχαν αρχικά τα γεγονότα που έχουν αποσυντεθεί σε μεμονωμένες τριπλέτες.

5.6. Οι βασικές έννοιες

Οι θεμελιώδεις έννοιες RDF είναι οι πόροι, οι ιδιότητες και οι δηλώσεις.

5.6.1. Πόροι

Μπορούμε να σκεφτούμε ως πόρο ένα αντικείμενο, ένα "πράγμα" για το οποίο θέλουμε να μιλήσουμε. Οι πόροι μπορούν να είναι συγγραφείς, βιβλία, εκδότες, θέσεις, άνθρωποι, ξενοδοχεία, δωμάτια, ερωτήσεις αναζήτησης, και τα λοιπά. Κάθε πόρος έχει ένα URI, ένα καθολικό προσδιοριστικό πόρων. Ένα URI μπορεί να είναι ένα URL (ενοποιημένος εντοπιστής των πόρων, ή διεύθυνση Ιστού) ή κάποιο άλλο είδος μοναδικού προσδιοριστικού. (υπενθυμίζουμε ότι ένα προσδιοριστικό δεν επιτρέπει απαραίτητως την πρόσβαση σε έναν πόρο).

Τα URIs έχουν καθοριστεί όχι μόνο για ιστοθέσεις αλλά και για διαφορετικά αντικείμενα όπως αριθμούς τηλεφώνου, αριθμούς ISBN και γεωγραφικές θέσεις. Γενικά, υποθέτουμε ότι ένα URI είναι το προσδιοριστικό ενός πόρου Ιστού.

5.6.2. Οι ιδιότητες

Οι ιδιότητες είναι ένα ειδικό είδος πόρων που περιγράφουν τις σχέσεις μεταξύ των πόρων, παραδείγματος χάριν "έχει γραφτεί από", "ηλικία", "τίτλος", και τα λοιπά. Οι ιδιότητες σε RDF προσδιορίζονται επίσης από URIs (και στην πράξη από URLs). Αυτή η ιδέα της χρησιμοποίησης URIs για να προσδιορίσει "τα πράγματα" και τις σχέσεις μεταξύ τους είναι αρκετά σημαντική. Αυτή η επιλογή μας οδηγεί σε ένα σφαιρικό, παγκόσμιο, μοναδικό σχήμα. Η χρήση ενός τέτοιου σχήματος περιορίζει πολύ το πρόβλημα ομωνύμων.

5.6.3. Οι δηλώσεις

Οι δηλώσεις βεβαιώνουν τις ιδιότητες των πόρων. Μια δήλωση είναι μία τριπλέτα της μορφής αντικείμενο - κατηγορήμα - τιμή, που αποτελείται από έναν πόρο, μια ιδιότητα και μια τιμή. Οι τιμές μπορούν είτε να είναι πόροι είτε αλφαριθμητικά.

5.7. Τρεις μορφές μιας δήλωσης

Ένα παράδειγμα μιας δήλωσης είναι

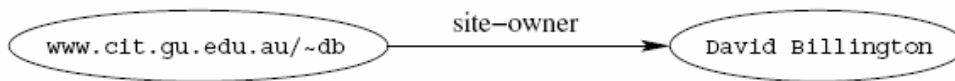
David Billington is the owner of the Web page <http://www.cit.gu.edu.au/~db>.

Ο απλούστερος τρόπος για αυτήν τη δήλωση είναι να χρησιμοποιηθεί ο ορισμός και να εξεταστεί η τριπλέτα

(“David Billington”, <http://www.mydomain.org/site-owner>, <http://www.cit.gu.edu.au/~db>).

Μπορούμε να αναλογιστούμε αυτή την τριπλέτα (x, P, y) ως έναν λογικό τύπο $P(x, y)$, όπου το δυαδικό κατηγορήμα P σχετίζει το αντικείμενο x με το αντικείμενο y . Στην πραγματικότητα, η RDF προσφέρει μόνο δυαδικά κατηγορήματα (ιδιότητες).

Μια δεύτερη άποψη είναι η γραφική αναπαράσταση. Το σχήμα που ακολουθεί παρουσιάζει τη γραφική παράσταση που αντιστοιχεί στην προηγούμενη δήλωση.

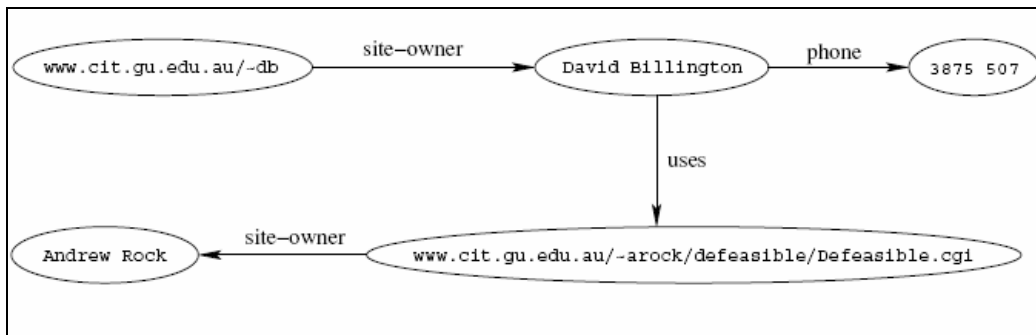


Είναι μια κατευθυνόμενη γραφική παράσταση με τους επονομαζόμενους κόμβους και τα τόξα να κατευθύνονται από τον πόρο (το υποκείμενο της δήλωσης) στην τιμή (το αντικείμενο της δήλωσης). Αυτό το είδος γραφικής παράστασης είναι γνωστό κοινότητα στην τεχνητής νοημοσύνης ως semantic net.

Όπως είπαμε ήδη, η τιμή μιας δήλωσης μπορεί να είναι ένας πόρος, επομένως, μπορεί να συνδεθεί με άλλους πόρους. Ας εξετάσουμε τις ακόλουθες τριπλέτες:

(<http://www.cit.gu.edu.au/~db>, <http://www.mydomain.org/siteowner>,
 " David Billington ")
 ("David Billington", <http://www.mydomain.org/phone>, "3875507")
 (" David Billington ", <http://www.mydomain.org/uses>,
<http://www.cit.gu.edu.au/~arock/defeasible/Defeasible.cgi>)
 (" www.cit.gu.edu.au/~arock/defeasible/Defeasible.cgi",
<http://www.mydomain.org/site-owner>, "Andrew Rock")

Η γραφική αναπαράσταση έχει ως εξής:



Οι γράφοι είναι ένα ισχυρό εργαλείο για την ανθρώπινη κατανόηση. Αλλά το όραμα του σημασιολογικού Ιστού απαιτεί τα δεδομένα να είναι κατανοητά και επεξεργάσιμα από τον υπολογιστή.

Επομένως, υπάρχει μια τρίτη δυνατότητα αντιπροσώπευσης βασισμένη σε XML. Σύμφωνα με αυτήν την δυνατότητα, ένα έγγραφο RDF αντιπροσωπεύεται από ένα στοιχείο XML με την ετικέτα `rdf:RDF`. Το περιεχόμενο αυτού του στοιχείου είναι διάφορες περιγραφές, οι οποίες χρησιμοποιούν τις ετικέτες `rdf:Description`. Κάθε περιγραφή κάνει μια δήλωση για έναν πόρο, ο οποίος προσδιορίζεται με έναν από τρεις διαφορετικούς τρόπους:

- ένα κατηγορημα about, το οποίο παραπέμπει σε έναν υπάρχοντα πόρο
- ένα ID κατηγορημα, που δημιουργεί έναν νέο πόρο
- χωρίς κάποιο όνομα, δημιουργώντας έναν ανώνυμο πόρο

Ακολουθεί η αναπαράσταση της πρώτης δήλωσής μας:

```
<?xml version="1.0" encoding="UTF-16"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:mydomain="http://www.mydomain.org/my-rdf-ns">

  <rdf:Description rdf:about="http://www.cit.gu.edu.au/~db">
    <mydomain:site-owner>
      David Billington
    </mydomain:site-owner>
  </rdf:Description>
</rdf:RDF>
```

Η πρώτη γραμμή διευκρινίζει ότι χρησιμοποιούμε XML. Στα ακόλουθα παραδείγματα παραλείπουμε αυτήν την γραμμή, αλλά λαμβάνουμε υπόψη ότι πρέπει να είναι παρούσα σε οποιοδήποτε έγγραφο RDF με XML σύνταξη. Το στοιχείο `rdf:Description` κάνει μια δήλωση για τον πόρο

<http://www.cit.gu.edu.au/~db>.

Μέσα στην περιγραφή η ιδιότητα χρησιμοποιείται ως ετικέτα, και το περιεχόμενο είναι η τιμή της ιδιότητας.

Οι περιγραφές δίνονται με μια ορισμένη σειρά, με άλλα λόγια, η σύνταξη XML επιβάλλει *serialization*. Η σειρά των περιγραφών (ή των πόρων) δεν είναι σημαντική σύμφωνα με το αφηρημένο πρότυπο RDF. Αυτό πάλι δείχνει ότι το πρότυπο γραφικών παραστάσεων είναι το πραγματικό πρότυπο στοιχείων RDF και ότι η XML είναι ακριβώς μια πιθανή σειριακή αντιπροσώπευση της γραφικής παράστασης.

5.8. Reification

Σε RDF είναι δυνατό να γίνουν δηλώσεις για άλλες δηλώσεις, όπως

Dimitris believes that David Billington is the creator of the Web page <http://www.cit.gu.edu.au/~db>.

Αυτό το είδος δήλωσης μπορεί να χρησιμοποιηθεί για να περιγράψει την εμπιστοσύνη σε άλλες δηλώσεις, το οποίο είναι σημαντικό σε μερικά είδη εφαρμογών. Η λύση είναι να οριστεί ένα μοναδικό προσδιοριστικό σε κάθε δήλωση, η οποία μπορεί να χρησιμοποιηθεί για να αναφερθούμε στη δήλωση. Η RDF επιτρέπει κάτι τέτοιο χρησιμοποιώντας το μηχανισμό reification. Η βασική ιδέα είναι να εισαχθεί ένα βοηθητικό αντικείμενο, για παράδειγμα, *belief1*, και να συσχετισθεί με κάθε ένα από τα τρία μέρη της αρχικής δήλωσης μέσω των ιδιοτήτων *υποκείμενο*, *κατηγορήμα* και *αντικείμενο*.

Στο προηγούμενο παράδειγμα το υποκείμενο του *belief1* θα ήταν ο David Billington, το κατηγορήμα θα ήταν *ο δημιουργός*, και το αντικείμενο <http://www.cit.gu.edu.au/~db>. Αυτή η προσέγγιση είναι απαραίτητη επειδή σε RDF υπάρχουν μόνο τριπλέτες, επομένως δεν μπορούμε να προσθέσουμε ένα προσδιοριστικό άμεσα σε μία τριπλέτα.

5.9. Datatypes

Ας αναλογιστούμε τον αριθμό τηλεφώνου "3875507". Ένα πρόγραμμα που διαβάζει σύμφωνα με το πρότυπο RDF δεν μπορεί να ξέρει εάν τα "3875507" πρόκειται να ερμηνευθούν ως ακέραιος αριθμός ή ως αλφαριθμητικό, ή πράγματι εάν είναι ένας ακέραιος αριθμός, εάν αυτός είναι στο δεκαδικό ή οκταδικό σύστημα. Ένα πρόγραμμα μπορεί μόνο να ξέρει πώς να ερμηνεύσει αυτόν τον πόρο εάν στην εφαρμογή δίνονται ρητά οι πληροφορίες ότι προορίζεται να αντιπροσωπεύσει έναν αριθμό, και ακριβώς ποιος αριθμός είναι αυτός. Η κοινή πρακτική στον προγραμματισμό των γλωσσών ή των συστημάτων βάσεων δεδομένων είναι να παρασχεθεί αυτό το είδος πληροφοριών συνδέοντας έναν τύπο στοιχείων με τον αριθμό τηλεφώνου, που σε αυτήν την περίπτωση θα είναι ένας τύπος στοιχείων όπως ένας δεκαδικός ή ακέραιος αριθμός. Σε RDF, τα *literals* χρησιμοποιούνται για να παρέχουν αυτό το είδος πληροφοριών.

Έτσι, θα μπορούσαμε να περιγράψουμε την ηλικία του Kostas Kalogirou ως έναν ακέραιο αριθμό, ο αριθμός 27, χρησιμοποιώντας την τριπλέτα:

("Kostas Kalogirou", <http://www.mydomain.org/age>,
"27"^^<http://www.w3.org/2001/XMLSchema#integer>)

Αυτό το παράδειγμα παρουσιάζει τα εξής: τη χρήση της ^^-σημείωσης για να δείξει τον τύπο του στοιχείου που μας ενδιαφέρει, και τη χρήση των τύπων δεδομένων που προκαθορίζονται από XML.

Η χρήση οποιουδήποτε εξωτερικά καθορισμένου σχεδίου δακτυλογράφησης στοιχείων επιτρέπεται στα έγγραφα RDF, αλλά στην πράξη, το ευρύτερα χρησιμοποιημένο σχέδιο δακτυλογράφησης στοιχείων θα είναι το XML σχήμα. Το σχήμα XML προκαθορίζει μια μεγάλη σειρά τύπων δεδομένων, συμπεριλαμβανομένων των Booleans, των ακέραιων αριθμών και floating-point αριθμών, των χρόνων και των ημερομηνιών.

5.10. RDF: Σύνταξη βασισμένη σε XML

Ένα έγγραφο RDF αποτελείται από ένα στοιχείο `rdf:RDF`, το περιεχόμενο του οποίου είναι διάφορες περιγραφές. Παραδείγματος χάρη, εξετάζοντας τον τομέα των πανεπιστημιακών μαθημάτων και των ομιλητών στο πανεπιστήμιο Griffith το έτος 2001:

```
<!DOCTYPE owl [  
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">  
>  
<rdf:RDF  
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"  
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"  
  xmlns:uni="http://www.mydomain.org/uni-ns#">  
  
  <rdf:Description rdf:about="949352">  
    <uni:name> Dimitris Petridis</uni:name>  
    <uni:title>Professor</uni:title>  
  </rdf:Description>  
  
  <rdf:Description rdf:about="949318">  
    <uni:name>Kostas Kalogirou</uni:name>  
    <uni:title>Associate Professor</uni:title>  
    <uni:age rdf:datatype="&xsd;integer">27</uni:age>  
  </rdf:Description>  
  
  <rdf:Description rdf:about="949111">
```

```

    <uni:name>Michael Maher</uni:name>
    <uni:title>Professor</uni:title>
</rdf:Description>

<rdf:Description rdf:about="CIT1111">
  <uni:courseName>Discrete Mathematics</uni:courseName>
  <uni:isTaughtBy>Kostas Kalogirou</uni:isTaughtBy>
</rdf:Description>

<rdf:Description rdf:about="CIT1112">
  <uni:courseName>Concrete Mathematics</uni:courseName>
  <uni:isTaughtBy>Dimitris Petridis</uni:isTaughtBy>
</rdf:Description>
<rdf:Description rdf:about="CIT2112">
  <uni:courseName>Programming III</uni:courseName>
  <uni:isTaughtBy>Michael Maher</uni:isTaughtBy>
</rdf:Description>

<rdf:Description rdf:about="CIT3112">
  <uni:courseName>Theory of Computation</uni:courseName>
  <uni:isTaughtBy>Kostas Kalogirou</uni:isTaughtBy>
</rdf:Description>

<rdf:Description rdf:about="CIT3116">
  <uni:courseName>Knowledge Representation</uni:courseName>
  <uni:isTaughtBy>Dimitris Petridis</uni:isTaughtBy>
</rdf:Description>
</rdf:RDF>

```

Ο μηχανισμός namespace XML χρησιμοποιείται, αλλά με έναν επεκταθέντα τρόπο. Τα XML namespaces χρησιμοποιούνται μόνο για λόγους αποσαφήνισης. Σε RDF, τα εξωτερικά namespaces αναμένονται να είναι έγγραφα RDF που προσδιορίζουν τους πόρους, οι οποίοι χρησιμοποιούνται έπειτα στο έγγραφο RDF που γίνεται imported. Αυτός ο μηχανισμός επιτρέπει την επαναχρησιμοποίηση των πόρων από άλλους που μπορούν να αποφασίσουν να παρεμβάλουν πρόσθετα χαρακτηριστικά γνωρίσματα σε αυτούς τους πόρους.

Επιπλέον, η ιδιότητα `rdf:about` του στοιχείου `rdf:Description` χρησιμοποιείται ως ισοδύναμη έννοια με αυτή ενός ID attribute, αλλά χρησιμοποιείται συχνά για να ξεκαθαρίσει ότι το αντικείμενο για το οποίο γίνεται μια δήλωση "ήδη έχει καθοριστεί" αλλού. Ένα σύνολο δηλώσεων RDF απλά διαμορφώνει μια μεγάλη γραφική παράσταση, η οποία συσχετίζει έννοιες με άλλες έννοιες μέσω των ιδιοτήτων, και δεν υπάρχει περίπτωση να οριστεί κάτι σε μια θέση και να αναφερθούμε σε αυτό αλλού. Στην πραγματικότητα το προηγούμενο παράδειγμα

είναι σε ένα βαθμό παραπλανητικό. Εάν θελήσουμε να είμαστε απολύτως σωστοί, πρέπει να αντικαταστήσουμε όλα τα περιστατικά και τις ταυτότητες προσωπικού, όπως 949352 και CIT3112, με αναφορές σε εξωτερικά namespaces, όπως για παράδειγμα:

```
<rdf:Description  
rdf:about="http://www.mydomain.org/uni-ns/#CIT3112">
```

Το περιεχόμενο των στοιχείων rdf:Description καλείται “στοιχεία ιδιοτήτων”. Για παράδειγμα, στην περιγραφή

```
<rdf:Description rdf:about="CIT3116">  
  <uni:courseName>Knowledge Representation</uni:courseName>  
  <uni:isTaughtBy>Dimitris Petridis</uni:isTaughtBy>  
</rdf:Description>
```

τα στοιχεία uni:courseName και uni:isTaughtBy, και τα δύο, καθορίζουν τα ζευγάρια ιδιότητα-αξία για το CIT3116. Η προηγούμενη περιγραφή αντιστοιχεί σε δύο δηλώσεις RDF.

Τέλος, η ιδιότητα rdf:datatype="xsd:integer" χρησιμοποιείται για να δείξει τον τύπο των στοιχείων της τιμής της ιδιότητας “ηλικία”. Ακόμα κι αν η ιδιότητα ηλικία έχει καθοριστεί για να έχει "xsd:integer" ως τιμή της, πρέπει να υποδεικνύεται ο τύπος της τιμής αυτής της ιδιότητας, κάθε φορά που χρησιμοποιείται. Αυτό πρόκειται να εξασφαλίσει ότι ένας επεξεργαστής RDF μπορεί να ορίσει το σωστό τύπο της τιμής της ιδιότητας, ακόμα κι αν δεν έχει δει τον αντίστοιχο καθορισμό σχημάτων RDF πριν.

5.10.1. Το rdf:resource

Το προηγούμενο παράδειγμα δεν ήταν ικανοποιητικό από μια άποψη: οι σχέσεις μεταξύ των μαθημάτων και των ομιλητών δεν καθορίστηκαν τυπικά αλλά σιωπηρά μέσω της χρήσης του ίδιου ονόματος. Για τον υπολογιστή όμως, η χρήση του ίδιου ονόματος μπορεί να είναι απλώς μια σύμπτωση: για παράδειγμα, ο Kostas Kalogirou που διδάσκει CIT3112 δεν μπορεί να είναι το ίδιο πρόσωπο με εκείνον που έχει ταυτότητα 949318 ο οποίος συμβαίνει να ονομάζεται Kostas Kalogirou. Αυτό που χρειαζόμαστε είναι μια επίσημη προδιαγραφή του γεγονότος ότι, παραδείγματος χάριν, ο δάσκαλος CIT1111 είναι το μέλος προσωπικού με τον αριθμό 949318, το

όνομα του οποίου είναι Kostas Kalogirou. Μπορούμε να πετύχουμε κάτι τέτοιο χρησιμοποιώντας μια ιδιότητα rdf:resource:

```
<rdf:Description rdf:about="CIT1111">  
  <uni:courseName>Discrete Mathematics</uni:courseName>  
  <uni:isTaughtBy rdf:resource="949318"/>  
</rdf:Description>
```

```
<rdf:Description rdf:about="949318">  
  <uni:name> Kostas Kalogirou </uni:name>  
  <uni:title>Associate Professor</uni:title>  
</rdf:Description>
```

Σε περίπτωση που είχαμε καθορίσει τον πόρο του εργαζομένου με αριθμό ID 949318 στο έγγραφο RDF που χρησιμοποιεί τις ιδιότητες ταυτότητας αντί του κατηγορήματος *about*, θα έπρεπε να χρησιμοποιήσουμε το σύμβολο # μπροστά από το 949318 στην τιμή του rdf:resource:

```
<rdf:Description rdf:about="CIT1111">  
  <uni:courseName>Discrete Mathematics</uni:courseName>  
  <uni:isTaughtBy rdf:resource="#949318"/>  
</rdf:Description>
```

```
<rdf:Description rdf:ID="#949318">  
  <uni:name> Kostas Kalogirou </uni:name>  
  <uni:title>Associate Professor</uni:title>  
</rdf:Description>
```

Το ίδιο ισχύει για τους εξωτερικά καθορισμένους πόρους: Για παράδειγμα, αναφερόμαστε στον εξωτερικά καθορισμένο πόρο CIT1111 μέσω του

<http://www.mydomain.org/uni-ns/#CIT1111>

ως τιμή του rdf:about, όπου www.mydomain.org/uni-NS είναι το URI όπου βρίσκεται ο ορισμός του CIT1111. Με άλλα λόγια, μια περιγραφή με ένα ID ορίζει ένα URI, το οποίο μπορεί να χρησιμοποιηθεί για να παραπέμψει στην συγκεκριμένη περιγραφή.

5.10.2. Εμφωλευμένες περιγραφές

Υπάρχει περίπτωση να ορίσουμε περιγραφές μέσα σε άλλες περιγραφές. Για παράδειγμα, μπορούμε να αντικαταστήσουμε τις περιγραφές του προηγούμενου παραδείγματος με την ακόλουθη περιγραφή:

```
<rdf:Description rdf:about="CIT1111">
  <uni:courseName>Discrete Mathematics</uni:courseName>
  <uni:isTaughtBy>
    <rdf:Description rdf:about="949318">
      <uni:name>Kostas Kalogirou</uni:name>
      <uni:title>Associate Professor</uni:title>
    </rdf:Description>
  </uni:isTaughtBy>
</rdf:Description>
```

Άλλα μαθήματα, όπως το CIT3112, εξακολουθούν να μπορούν να αναφερθούν στο νέο πόρο 949318. Με άλλα λόγια, αν και μια περιγραφή μπορεί να καθοριστεί μέσα σε μια άλλη περιγραφή, το πεδίο της είναι σφαιρικό.

5.10.3. Το στοιχείο `rdf:type`

Στα παραδείγματά μας μέχρι τώρα, οι περιγραφές εμπίπτουν σε δύο κατηγορίες: μαθήματα και ομιλητές. Αυτό το γεγονός είναι σαφές σε εμάς, αλλά δεν είναι επίσημα γνωστό, με αποτέλεσμα να μην είναι προσιτό στον υπολογιστή. Σε RDF, ωστόσο, είναι δυνατό να γίνουν τέτοιες δηλώσεις που χρησιμοποιούν το στοιχείο `rdf:type`. Για παράδειγμα:

```
<rdf:Description rdf:about="CIT1111">
  <rdf:type rdf:resource="&uni;course"/>
  <uni:courseName>Discrete Mathematics</uni:courseName>
  <uni:isTaughtBy rdf:resource="949318"/>
</rdf:Description>

<rdf:Description rdf:about="949318">
  <rdf:type rdf:resource="&uni;lecturer"/>
  <uni:name>Kostas Kalogirou</uni:name>
  <uni:title>Associate Professor</uni:title>
</rdf:Description>
```

Το `rdf:type` μας επιτρέπει να δομήσουμε το έγγραφο RDF.

5.10.4. Τα στοιχεία containers

Όπως ήδη αναφέραμε, τα containers χρησιμοποιούνται για να συλλέξουμε διάφορους πόρους ή ιδιότητες για τα οποία θέλουμε να κάνουμε μια δήλωση ως σύνολο. Για παράδειγμα, εάν θέλαμε να μιλήσουμε για τα μαθήματα που παραδίδει ένας συγκεκριμένος καθηγητής, τρεις τύποι containers είναι διαθέσιμοι σε RDF:

- `rdf:Bag`, το οποίο μπορεί να περιέχει πολλαπλά περιστατικά, τα οποία δεν είναι διατεταγμένα σε μία σειρά. Χαρακτηριστικό παράδειγμα είναι τα έγγραφα που αποτελούν ένα αρχείο, περίπτωση στην οποία δεν επιβάλλεται μία συγκεκριμένη σειρά.
- `rdf:Seq` ένα διαταγμένο container, το οποίο μπορεί να περιέχει τα πολλαπλά περιστατικά. Χαρακτηριστικά παραδείγματα είναι οι ενότητες μιας σειράς μαθημάτων, ένας αλφαριθμητικός τηλεφωνικός κατάλογος - παραδείγματα όπου απαιτείται μία συγκεκριμένη σειρά.
- `rdf:Alt` είναι ένα σύνολο εναλλακτικών λύσεων. Ως χαρακτηριστικό παράδειγμα αναφέρουμε τις μεταφράσεις ενός εγγράφου στις διάφορες γλώσσες.

Το περιεχόμενο των δεδομένων των containers είναι στοιχεία που ονομάζονται `rdf:_1`, `rdf:_2`, και λοιπά. Αναδιατυπώνοντας ολόκληρο το RDF έγγραφό μας προκύπτει:

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:uni="http://www.mydomain.org/uni-ns#">

  <uni:lecturer rdf:about="949352"
    uni:name="Dimitris Petridis"
    uni:title="Professor">
    <uni:coursesTaught>
      <rdf:Bag>
        <rdf:_1 rdf:resource="CIT1112"/>
        <rdf:_2 rdf:resource="CIT3116"/>
      </rdf:Bag>
    </uni:coursesTaught>
  </uni:lecturer>

  <uni:lecturer rdf:about="949318"
    uni:name="Kostas Kalogirou"
    uni:title="Associate Professor">
    <uni:coursesTaught>
      <rdf:Bag>
        <rdf:_1 rdf:resource="CIT1111"/>
      </rdf:Bag>
    </uni:coursesTaught>
  </uni:lecturer>
</rdf:RDF>
```

```

    <rdf:_2 rdf:resource="CIT3112"/>
  </rdf:Bag>
</uni:coursesTaught>
</uni:lecturer>

<uni:lecturer rdf:about="949111"
  uni:name="Michael Maher"
  uni:title="Professor">
  <uni:coursesTaught rdf:resource="CIT2112"/>
</uni:lecturer>

<uni:course rdf:about="CIT1111"
  uni:courseName="Discrete Mathematics">
  <uni:isTaughtBy rdf:resource="949318"/>
</uni:course>

<uni:course rdf:about="CIT1112"
  uni:courseName="Concrete Mathematics">
  <uni:isTaughtBy rdf:resource="949352"/>
</uni:course>

<uni:course rdf:about="CIT2112"
  uni:courseName="Programming III">
  <uni:isTaughtBy rdf:resource="949111"/>
</uni:course>

<uni:course rdf:about="CIT3112"
  uni:courseName="Theory of Computation">
  <uni:isTaughtBy rdf:resource="949318"/>
</uni:course>

<uni:course rdf:about="CIT3116"
  uni:courseName="Knowledge Representation">
  <uni:isTaughtBy rdf:resource="949352"/>
</uni:course>

</rdf:RDF>

```

Αντι `rdf:_1`, `rdf:_2` .. είναι δυνατό να γραφτεί `rdf:li`. Χρησιμοποιούμε αυτήν την συντακτική παραλλαγή στο ακόλουθο παράδειγμα. Υποθέτουμε ότι η σειρά μαθημάτων CIT1111 διδάσκεται είτε τον Δημήτρη Πετριδη είτε από τον Κώστα Καλογήρου:

```

<uni:course rdf:about="CIT1111"
  uni:courseName="Discrete Mathematics">
  <uni:lecturer>
  <rdf:Alt>

```

```

    <rdf:li rdf:resource="949352"/>
    <rdf:li rdf:resource="949318"/>
  </rdf:Alt>
</uni:lecturer>
</uni:course>

```

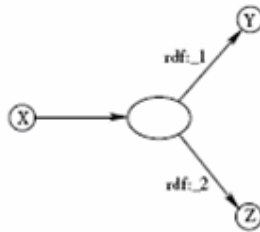
Τα containers έχουν μια προαιρετική ID ιδιότητα, με την οποία μπορεί να γίνει αναφορά σε αυτά:

```

<uni:lecturer rdf:about="949318"
  uni:name="Kostas Kalogirou"
  uni:title="Associate Professor">
  <uni:coursesTaught>
    <rdf:Bag rdf:ID="DBcourses">
      <rdf:_1 rdf:resource="CIT1111"/>
      <rdf:_2 rdf:resource="CIT3112"/>
    </rdf:Bag>
  </uni:coursesTaught>
</uni:lecturer>

```

Μια χαρακτηριστική εφαρμογή των containers είναι η αντιπροσώπευση των κατηγορημάτων με περισσότερα από δύο χαρακτηριστικά. Στο παράδειγμα $referee(X, Y, Z)$, όπου το X είναι ο διαιτητής ενός παιχνιδιού σκακιού μεταξύ των παικτών Y και Z, μία λύση είναι να διακριθεί ο διαιτητής X από τους φορείς Y και Z. Η γραφική παράσταση φαίνεται στο παρακάτω σχήμα.



Η λύση σύμφωνα με την XML σύνταξη είναι της μορφής:

```

<referee rdf:about="..#X">
  <players>
    <rdf:Bag>
      <rdf:li rdf:resource="..#Y"/>
      <rdf:li rdf:resource="..#Z"/>
    </rdf:Bag>
  </players>
</referee>

```


Το `rdf:Bag` καθορίζει έναν ανώνυμο βοηθητικό πόρο. Επιλέξαμε να χρησιμοποιήσουμε το `Bag` επειδή υποθέσαμε ότι δε γίνεται καμία διάκριση μεταξύ των φορέων. Εάν είχε σημασία η σειρά, για παράδειγμα ότι ο πρώτος φορέας έχει τα λευκά και ο δεύτερος τα μαύρα, θα χρησιμοποιούσαμε μια ακολουθία αντ' αυτού.

Ένας περιορισμός των `containers` είναι ότι δεν υπάρχει κανένας τρόπος να θεωρήσουμε ότι "αυτά είναι όλα τα μέλη του `container`", διότι, ενώ μια γραφική παράσταση μπορεί να περιγράψει μερικά από τα μέλη, δεν υπάρχει κανένας τρόπος να αποκλειστεί η δυνατότητα να υπάρχει μια άλλη γραφική παράσταση κάπου, η οποία να περιγράφει πρόσθετα μέλη. Η RDF επιτρέπει την περιγραφή των ομάδων που περιέχουν μόνο διευκρινισμένα μέλη, υπό μορφή συλλογών RDF. Μια συλλογή RDF είναι ομάδα στοιχείων που αναπαρίστανται ως λίστα στη γραφική παράσταση RDF. Αυτή η λίστα κατασκευάζεται χρησιμοποιώντας ένα προκαθορισμένο λεξιλόγιο συλλογής που αποτελείται από το προκαθορισμένο τύπο `rdf:List`, τις προκαθορισμένες ιδιότητες `rdf:first` και `rdf:rest`, και το προκαθορισμένο πόρο `rdf:nil`. Έτσι, μας επιτρέπεται να γράψουμε:

```
<rdf:Description rdf:about="CIT2112">
  <uni:isTaughtBy>
    <rdf:List>
      <rdf:first>
        <rdf:Description rdf:about="949111"/>
      </rdf:first>
      <rdf:rest>
        <rdf:List>
          <rdf:first>
            <rdf:Description rdf:about="949352"/>
          </rdf:first>
          <rdf:rest>
            <rdf:List>
              <rdf:first>
                <rdf:Description rdf:about="949318"/>
              </rdf:first>
              <rdf:rest>
                <rdf:Description rdf:about="&rdf:nil"/>
              </rdf:rest>
            </rdf:List>
          </rdf:rest>
        </rdf:List>
      </rdf:rest>
    </rdf:List>
  </uni:isTaughtBy>
</rdf:Description>
```

Αυτό δηλώνει ότι το CIT2112 διδάσκεται από τους δασκάλους που προσδιορίζονται ως πόροι 949111 ..949352, και 949318, και κανέναν άλλο (το οποίο υποδεικνύεται από το σύμβολο nil). Μία πιο σύντομη σύνταξη έχει καθοριστεί, χρησιμοποιώντας την τιμή "Collection" για το rdf:parseType:

```
<rdf:Description rdf:about="CIT2112">
  <uni:isTaughtBy rdf:parseType="Collection">
    <rdf:Description rdf:about="949111"/>
    <rdf:Description rdf:about="949352"/>
    <rdf:Description rdf:about="949318"/>
  </uni:isTaughtBy>
</rdf:Description>
```

5.10.5. Reification

Κάποιες φορές επιθυμούμε να κάνουμε δηλώσεις για άλλες δηλώσεις, όπως ήδη αναφέραμε παραπάνω. Θα πρέπει λοιπόν να είμαστε σε θέση να αναφερθούμε σε μια δήλωση χρησιμοποιώντας ένα προσδιοριστικό. Η RDF επιτρέπει μια τέτοια αναφορά μέσω ενός μηχανισμού reification που μετατρέπει μια δήλωση σε πόρο. Για παράδειγμα, η περιγραφή

```
<rdf:Description rdf:about="949352">
  <uni:name>Dimitris Petridis</uni:name>
</rdf:Description>
```

γίνεται ως εξής:

```
<rdf:Statement rdf:about="StatementAbout949352">
  <rdf:subject rdf:resource="949352"/>
  <rdf:predicate rdf:resource="&uni:name"/>
  <rdf:object>Dimitris Petridis</rdf:object>
</rdf:Statement>
```

Σημειώνουμε εδώ ότι τα rdf:subject, rdf:predicate, και rdf:object μας επιτρέπουν την πρόσβαση στα επιμέρους μέρη μιας δήλωσης.

Το ID της δήλωσης μπορεί να χρησιμοποιηθεί για την αναφορά σε αυτήν, όπως μπορεί να γίνει για οποιαδήποτε περιγραφή. Μπορούμε είτε να γράψουμε ένα

rdf:Description εάν δεν θέλουμε να μιλήσουμε άλλο για τη συγκεκριμένη δήλωση, είτε ένα rdf:Statement εάν επιθυμούμε να αναφερθούμε σε αυτήν.

Εάν περισσότερες από μία ιδιότητες περιλαμβάνονται σε μία περιγραφή, τότε αυτές αντιστοιχούν σε περισσότερες από μια δηλώσεις. Αυτές οι δηλώσεις μπορούν είτε να τοποθετηθούν σε ένα Bag και να θεωρηθούν ως μία οντότητα, ή μπορούμε να κάνουμε reify χωριστά.

5.10.6. RDF Properties

Μια δήλωση συνδέει έναν πόρο με την τιμή μιας ιδιότητας, ενώ το κατηγορημα, ή ο τύπος της ιδιότητας, δείχνει τον τύπο της συσχέτισης, και το αντικείμενο αντιπροσωπεύει την τιμή που ορίζεται.

Τα κατηγορήματα σε RDF είναι επίσης πόροι και μπορούν να αποτελέσουν το αντικείμενο των δηλώσεων.

5.10.7. Οι ιδιότητες (properties) ως πόροι

Δεδομένου ότι ένα κατηγορημα RDF (ή ιδιότητα) είναι επίσης ένα είδος πόρου, αυτό προσδιορίζεται από μια αναφορά URI. Για παράδειγμα, μια τυποποιημένη ιδιότητα RDF, σύμφωνα με το μοντέλο RDF είναι:

<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>

Μια ιδιότητα που προσδιορίζεται από αυτήν την αναφορά URI χρησιμοποιείται για να δηλώσει τον τύπο ενός πόρου. Αν θεωρήσουμε ως υποκείμενο ένα πρόσωπο, τότε ο τύπος του είναι <http://description.org/schema/Person>. Ξέρουμε ότι η ιδιότητα στη δήλωση είναι η ιδιότητα *type* επειδή το προσδιοριστικό του δίνεται ως <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>. Τώρα, δεδομένου ότι μια ιδιότητα είναι ένας πόρος, μπορεί επίσης να αποτελέσει το αντικείμενο δηλώσεων.

Τέτοιες δηλώσεις διευκρινίζουν πώς το κατηγορημα πρόκειται να χρησιμοποιηθεί, ή πώς μπορούμε να επιβάλλουμε περιορισμούς σε αυτό.

5.11. Ονόματα, ετικέτες και προσδιοριστικά

Συχνά, τα ονόματα και οι ετικέτες ενδέχεται να αντικατασταθούν με ονόματα που υποδεικνύουν τη σημασία τους από μόνα τους (όπως το Person αντί για το <http://description.org/schema/Person>). Θα μπορούσαμε να δημιουργήσουμε μία ολόκληρη βάση δεδομένων χρησιμοποιώντας το πρότυπο RDF με τριπλέτες, χωρίς τη χρησιμοποίηση URIs, αλλά μόνο κάνοντας χρήση απλών ονομάτων. Ωστόσο, αν κάποιος άλλος θα ήθελε να χρησιμοποιήσει τη συγκεκριμένη βάση, είναι πιθανό να μην μπορούσε να κατανοήσει τα ονόματα και τις ιδιότητες που εμείς θα είχαμε ορίσει, με αποτέλεσμα να χρειάζεται να χρησιμοποιήσουμε πιο γενικά προσδιοριστικά. Στο σημείο αυτό είναι που κάνουν την εμφάνισή τους οι αναφορές URI.

Η δημιουργία διαμοιράσιμων λεξιλογίων και προσδιοριστικών είναι χαρακτηριστικό της οντολογίας, και οι περισσότερες οντολογίες που προορίζονται για χρήση στον ιστό κάνουν χρήση URIs, ακριβώς όπως η RDF. Με σκοπό την πλήρη κατανόηση, θα χρησιμοποιούμε συνηθισμένες λέξεις και ονόματα ως URI αναφορές, αλλά πρέπει να θυμόμαστε ότι στην πραγματική πρακτική, οι URI αναφορές χρησιμοποιούνται για να προσδιορίσουν τους πόρους, κάτι που αποτελεί έναν τρόπο να κατασταθεί πιθανό για τα στοιχεία RDF να διανεμηθούν πέρα από τον Ιστό, το οποίο είναι απαραίτητο για τον σημασιολογικό Ιστό.

Εάν ένας πόρος προσδιορίζεται από μια αναφορά URI, τι θα έπρεπε να επιδείξει μια εφαρμογή για να τον αναπαραστήσει; Οι χαρακτηριστικές αναφορές URI δεν είναι πολύ αναγνώσιμες. Είναι καλύτερο να χρησιμοποιηθεί μια αναγνώσιμη ετικέτα (*label*), η οποία είναι γενικά το συνηθισμένο όνομα για τον πόρο. Η ίδια η ετικέτα είναι ακόμα μια ιδιότητα του RDF πόρου. Έχει καθοριστεί μια τυποποιημένη ετικέτα, με πλήρες URI το <http://www.w3.org/2000/01/rdf-schema#label>, ή `rdfs:label` για συντομία (η χρησιμοποίηση προθεμάτων όπως *rdfs:* για να αναπαραστήσουμε το κοινό μέρος ενός URI είναι ένας άλλος τρόπος να κατασταθεί ένα URI πιο αναγνώσιμο, μια πρακτική σχετική πολύ με τη χρήση των namespaces σε XML). Παρατηρούμε ότι ο πόρος δεν προσδιορίζεται από το όνομά του, αν και το σωστό λογισμικό θα μπορούσε να πραγματοποιήσει μια ερώτηση και να ανακτήσει όλους τους πόρους με τη συγκεκριμένη ετικέτα. Φυσικά, θα μπορούσαμε επίσης να καθορίσουμε μια ιδιότητα *name* για να συμπληρώσουμε την ετικέτα.

5.11.1. Ιδιότητες των δηλώσεων

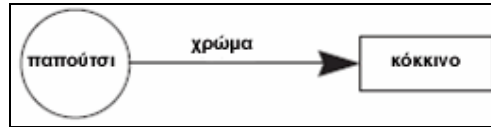
Εστώ ότι θέλουμε να μιλήσουμε για μία δήλωση και να πούμε, για παράδειγμα, ποιος την δημοσίευσε και πού ή να δηλώσουμε ότι διαφωνούμε με αυτήν. Οι δηλώσεις έχουν και αυτές τις ιδιότητές τους. Αλλά αυτό δεν είναι απλό σε RDF. Μια δήλωση δεν είναι πόρος και δεν μπορεί έτσι να αποτελέσει το υποκείμενο μιας άλλης δήλωσης. Σε RDF αυτός ο περιορισμός αντιμετωπίζεται με τον εξής τρόπο. Υπάρχει ο ιδιαίτερος τύπος πόρου που ήδη αναφέραμε, με το προσδιοριστικό:

<http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement>

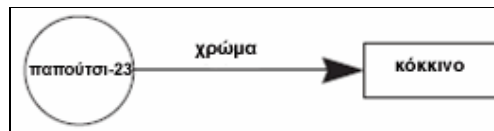
Ένας τέτοιος πόρος καλείται `rdf:Statement` και έχει τέσσερις ιδιότητες: `rdf:subject`, `rdf:predicate`, `rdf:object`, και `rdf:type` (η αξία του οποίου είναι `rdf:Statement`). Ένα `rdf:Statement` δηλώνει ότι υπάρχει μια δήλωση με τα συγκεκριμένα υποκείμενο, κατηγορημα και αντικείμενο. Το παράξενο είναι ότι ένα τέτοιο `rdf:Statement` δεν μπορεί να προσδιοριστεί με καμία τριπλέτα, ακόμα κι αν υπάρχει κάποια που έχει το ίδιο υποκείμενο, κατηγορημα, και αντικείμενο. Δεν μπορούμε να κάνουμε δηλώσεις για οποιαδήποτε πραγματική τριπλέτα, αλλά για κάτι που είναι τύπου `rdf:Statement` και έχει τις ίδιες τιμές για κάθε ένα από τα τρία μέρη της συγκεκριμένης δήλωσης.

5.12. Απεικόνιση των statements με RDF γράφους

Οι κατευθυνόμενοι γράφοι ενδείκνυνται για την αναπαράσταση των δηλώσεων RDF. Μια γραφική παράσταση είναι μια συλλογή κόμβων που συνδέονται με γραμμές (μερικές φορές αποκαλούμενες τόξα). Σε μια γραφική παράσταση με labels, οι γραμμές έχουν ετικέτες, ενώ σε μια κατευθυνόμενη γραφική παράσταση, οι γραμμές δείχνουν από τον έναν κόμβο άλλο. Στις γραφικές παραστάσεις που αντιπροσωπεύουν μια συλλογή δηλώσεων RDF, ένας κόμβος αντιπροσωπεύει έναν πόρο ή μια κυριολεκτική τιμή, και ένα βέλος αντιπροσωπεύει ένα κατηγορημα. Το παρακάτω σχήμα παρουσιάζει μια γραφική παράσταση με labels για το χρώμα ενός παπουτσιού. Αυτή η γραφική παράσταση είναι εξαιρετικά απλή, αλλά ακόμα κι έτσι είναι εύκολο να παρατηρηθεί ότι η ετικέτα "παπούτσι" είναι διφορούμενη. Μιλάμε γενικά για ένα παπούτσι, ή για ένα συγκεκριμένο παπούτσι;



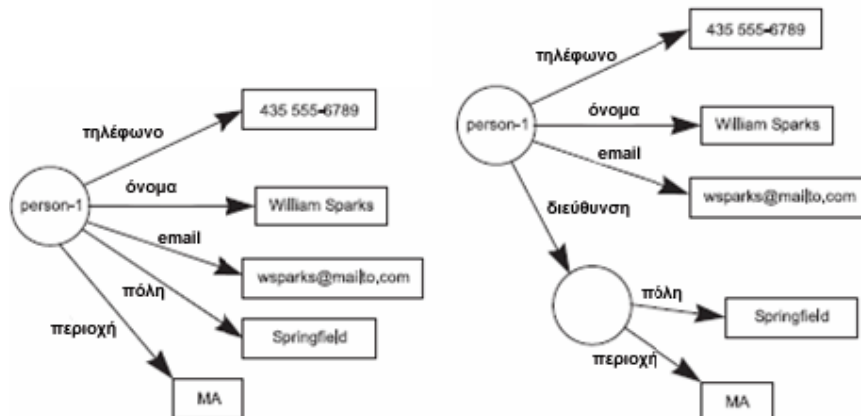
Σε RDF, δεν μπορούμε να μιλήσουμε για όλα τα παπούτσια γενικά, αλλά για συγκεκριμένους πόρους. Μπορούμε να κάνουμε δηλώσεις για την κλάση των παπουτσιών, που αντιπροσωπεύει μια συλλογή αντικειμένων τα οποία μοιράζονται τις ιδιότητες που τα κάνουν " παπούτσια," αλλά δεν είναι το ίδιο με το να κάνουμε δηλώσεις γενικά για παπούτσια. Ας δώσουμε σε ένα παπούτσι ένα προσδιοριστικό αντί ενός γενικού ονόματος, όπως φαίνεται στο σχήμα που ακολουθεί (στην πράξη, τα προσδιοριστικά για το παπούτσι και για την ιδιότητα χρώμα θα ήταν αναφορές URI).



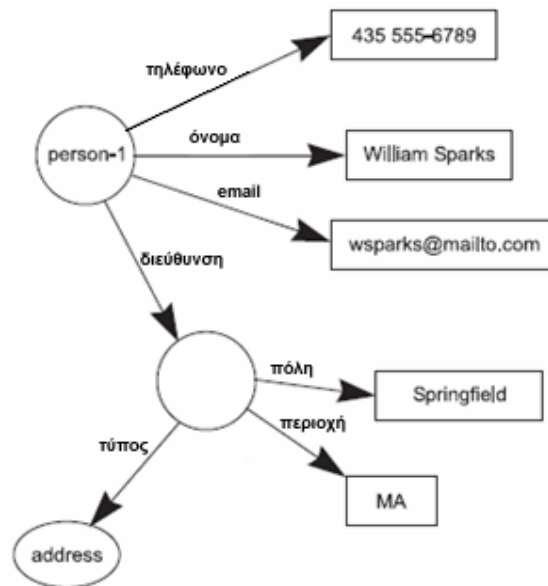
Στην απεικόνιση των RDF γράφων, χρησιμοποιούμε έναν κύκλο ή οβάλ για να αναπαραστήσουμε έναν πόρο και ένα ορθογώνιο για να αναπαραστήσουμε μια κυριολεκτική τιμή, όπως ένα string. Σε αυτό το παράδειγμα, η τιμή της ιδιότητας είναι ένα string που αποτελείται από τους χαρακτήρες r, e, d.

5.12.1. Πόροι με πολλές δηλώσεις

Ας εξετάσουμε μια γραφική παράσταση που απεικονίζει αρκετές δηλώσεις για τον ίδιο πόρο.



Παρατηρούμε ότι τα «πόλη» και «περιοχή» δεν συσχετίζονται μεταξύ τους, αν και ξέρουμε ότι στην πραγματικότητα δεν είναι έτσι. Στο δεύτερο σχήμα αυτό το πρόβλημα αντιμετωπίζεται, με έναν τρόπο που επεξηγεί ένα άλλο σημαντικό χαρακτηριστικό γνώρισμα του RDF: τον ανώνυμο κόμβο. Τώρα το person-1 έχει μια διεύθυνση, η οποία συνδέεται με μια πόλη και μία περιοχή. Αλλά πώς ξέρουμε ότι ο νέος κόμβος είναι ένας πόρος τύπου «διεύθυνση»; Είτε θα μπορούσαμε να το συμπεράνουμε, εάν ξέραμε ότι το αντικείμενο μιας ιδιότητας «διεύθυνση» πρέπει να είναι τύπου «διεύθυνση» (ένα σχήμα ή μια οντολογία μπορεί να μας πει), ή μπορούμε ρητά να το δηλώσουμε με τον τύπο του νέου κόμβου, όπως παρουσιάζεται στο επόμενο σχήμα.



Παρατηρούμε ότι το αντικείμενο της ιδιότητας «τύπος» είναι ένα οβάλ - ένας πόρος αντί απλά μιας ετικέτας με κείμενο. Ο τύπος ενός πόρου είναι ο ίδιος ένα είδος πόρου.

5.12.2. Ανώνυμοι κόμβοι

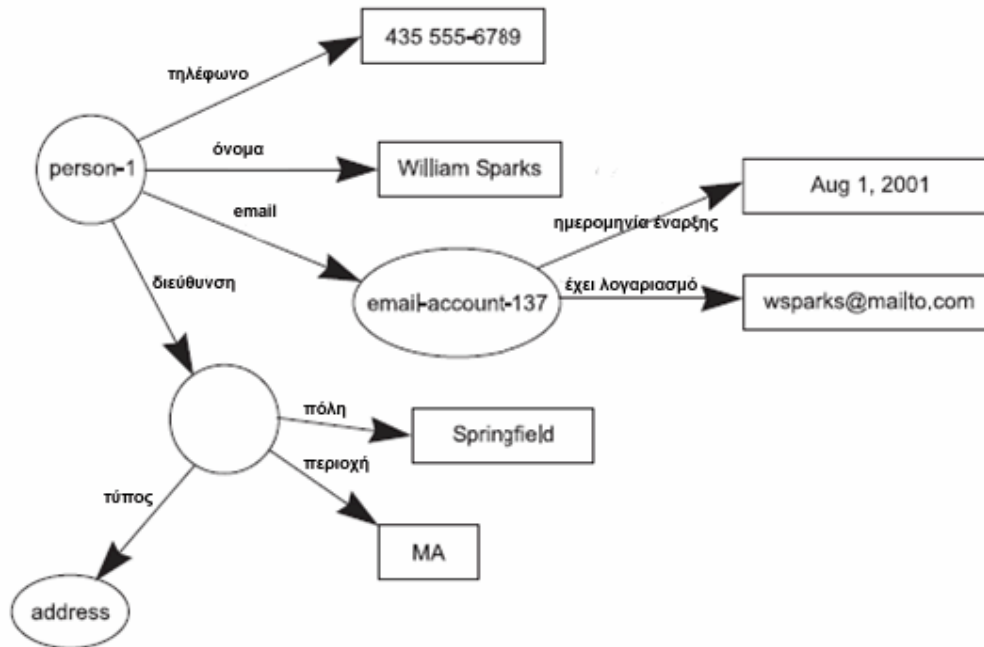
Για τον ανώνυμο κόμβο δεν υπάρχει προσδιοριστικό. Στη γραφική παράσταση παρατηρούμε: "person-1 has an address that is something whose type is address, whose city is 'Springfield', and whose state is 'MA'." Αυτό το "something" είναι γνωστό από τις ιδιότητές του, και όχι από μια ταυτότητα. Ο κόμβος είναι ένα κατασκεύασμα που αντιπροσωπεύει τον τρόπο που αποφασίσαμε να ομαδοποιήσουμε τα γεγονότα.

Τέτοια φαινόμενα τα συναντούμε συχνά στους RDF γράφους. Τέτοιοι κόμβοι καλούνται ανώνυμοι κόμβοι, ή μερικές φορές κενοί κόμβοι (b-nodes). Μια σχεσιακή βάση δεδομένων περιλαμβάνει τους εξαρτώμενους και ανεξάρτητους πίνακες. Ένας εξαρτώμενος πίνακας εξαρτάται από έναν ή περισσότερους άλλους πίνακες για την ύπαρξη και τη σημασία του. Ένας ανώνυμος κόμβος RDF είναι κάπως ανάλογος. Το πρόγραμμα επεξεργασίας RDF μπορεί να αντιστοιχίσει ένα εσωτερικό προσδιοριστικό κόμβων στον ανώνυμο κόμβο για πρακτικούς λόγους, αλλά εκείνο το προσδιοριστικό δεν έχει καμία χρησιμότητα σε έναν άλλο επεξεργαστή που λαμβάνει τη γραφική παράσταση. Η κατάσταση του κόμβου στη γραφική παράσταση του δίνει τη θεμελιώδη ταυτότητά του.

5.12.3. Οι πόροι ως αντικείμενα δηλώσεων

Υποθέτουμε ότι θέλουμε να κάνουμε μια δήλωση για το email του person-1. Ίσως το δημιούργησαν την 1η Αυγούστου, 2001. Θα επιθυμούσαμε να δημιουργήσουμε μία τριπλέτα με το email ως υποκείμενο, αλλά δεν μπορούμε επειδή το υποκείμενο πρέπει να είναι πόρος. Τον περιορισμό αυτόν μπορούμε να τον αντιμετωπίσουμε είτε με τη χρησιμοποίηση ενός ανώνυμου κόμβου είτε με τη δημιουργία ενός νέου πόρου. Η προσέγγιση που επιλέγουμε μπορεί να εξαρτηθεί από τον τρόπο με τον οποίο οι πληροφορίες οργανώνονται όταν τις λαμβάνουμε.

Το ακόλουθο σχήμα παρουσιάζει μια γραφική παράσταση για τη δεύτερη προσέγγιση.



Φαίνεται αρκετά απλό, αλλά εκθέτει μια νέα πτυχή της επεξεργασίας RDF. Στο προηγούμενο σχήμα, μπορούμε να βρούμε το email απλά, εάν καταλάβουμε την έννοια της ιδιότητας που ονομάζεται email. Αλλά στο τελευταίο σχήμα, δεν υπάρχει κάπου διεύθυνση email - τουλάχιστον δεν υπάρχει ένας τέτοιος επονομαζόμενος κόμβος. Για να εντοπίσουμε την πραγματική διεύθυνση email, πρέπει να βρούμε έναν πόρο τύπου email-account (για να απλουστεύσουμε τη γραφική παράσταση δεν συμπεριελάβαμε μια δήλωση type για τον πόρο account). Αναζητούμε μια ιδιότητα που ονομάζεται «έχει λογαριασμό», και κατόπιν λαμβάνουμε την τιμή της.

Η συγκεκριμένη διαδικασία μπορεί να είναι προφανής, καθώς μπορούμε να συμπεράνουμε τι σημαίνει κάθε οντότητα από την ετικέτα της. Ένας υπολογιστής, ωστόσο, θα δει τις ετικέτες και τους τύπους ως αυθαίρετα strings, και πρέπει να συμπεράνει από τους τύπους κατηγορήματος και αντικειμένου πώς να λάβει τη διεύθυνση email. Για να γίνει αυτό, πρέπει να είναι σε ισχύ κανόνες για εκείνους τους τύπους, μαζί με κανόνες για τα όποια συμπεράσματα. Οι κανόνες για τους τύπους ορίζονται από το RDF Schema, τους τύπους των δεδομένων και την οντολογία που χρησιμοποιούμε. Οι κανόνες για τα συμπεράσματα σχετίζονται με τη λογική.

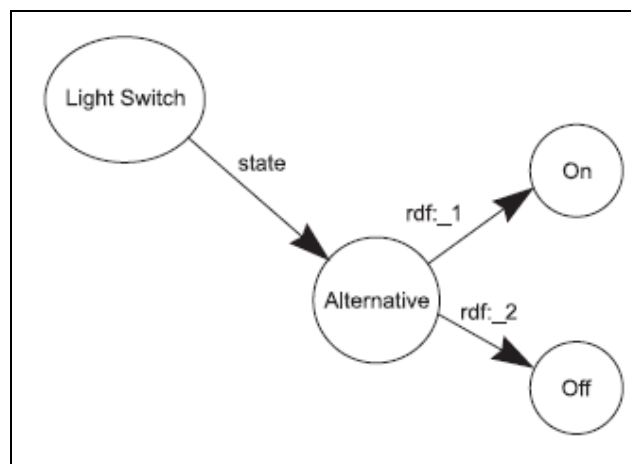
5.12.4. Container κόμβοι

Σύμφωνα με όσα έχουμε αναφέρει, η RDF καθορίζει τρεις ειδικούς τύπους πόρων για συλλογές, και ένα σύνολο ειδικών ιδιοτήτων για τη συσχέτιση της συλλογής με τα μέλη της:

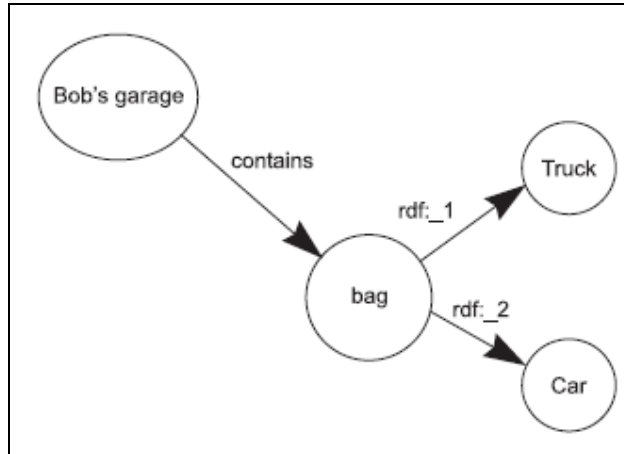
- *Bag* - περιλαμβάνει οποιονδήποτε αριθμό πόρων χωρίς συγκεκριμένη σειρά (τα αντίγραφα επιτρέπονται)
- *Sequence* - περιλαμβάνει μια διατεταγμένη συλλογή πόρων
- *Alternative* - διευκρινίζει ότι οποιοδήποτε από τα μέλη της μπορεί να επιλεγεί (παραδείγματος χάριν, ένας διακόπτης μπορεί να είναι σε θέση on ή off)

Τα τόξα από συλλογή κόμβων προς τα μέλη της ονομάζονται με ετικέτες όπως `rdf:_1`, `rdf:_2` και λοιπά.

Το ακόλουθο σχήμα παρουσιάζει το τρίτο είδος, το alternative container.



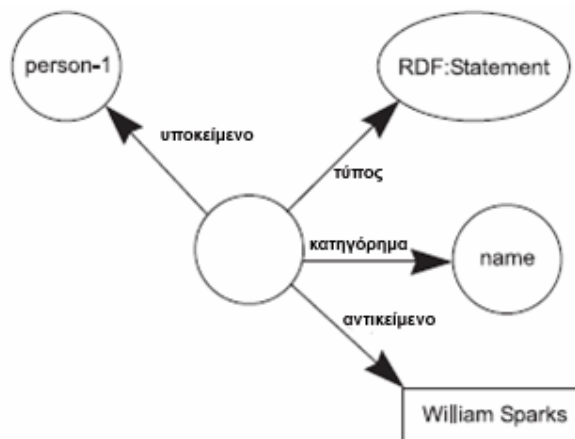
Στο επόμενο σχήμα, στις ετικέτες διαβάζουμε "truck" και "car" για λόγους απλότητας. Στην πράξη θα αντιπροσώπευαν ένα συγκεκριμένο αυτοκίνητο και ένα συγκεκριμένο φορτηγό.



Οι υπό πρόσφατες συστάσεις RDF προβλέπουν επίσης έναν τύπο *collection*. Αυτός ο τύπος έχει μια ενδιαφέρουσα συμπεριφορά που είναι διαφορετική από τους προηγούμενους τρεις τύπους containers, στους οποίους μπορεί κανείς να προσθέσει ένα ακόμη στοιχείο, απλά προσθέτοντας μία τριπλέτα στο container. Αλλά μια συλλογή κατασκευάζεται μία φορά και στη συνέχεια δεν μπορεί να δεχτεί άλλα μέλη.

5.12.5. Γραφική αναπαράσταση των ιδιοτήτων των δηλώσεων

Το `rdf:Statement` είναι ένα ειδικό είδος πόρου που μπορεί να χρησιμοποιηθεί για να αναφερθούμε σε μία τριπλέτα RDF. Στο πρότυπο RDF, μία συνηθισμένη τριπλέτα δεν είναι ένας πόρος, έτσι δεν μπορεί να αποτελέσει το υποκείμενο μιας άλλης δήλωσης. Αλλά μπορούμε να κάνουμε το σύνολο των δηλώσεων που παρουσιάζονται στο παρακάτω σχήμα.



Αυτή η γραφική παράσταση λέει ότι "υπάρχει κάτι του οποίου ο τύπος είναι RDF:Statement. Έχει το "person-1" ως υποκείμενο, το "name" ως κατηγορημα, και το "William Sparks" ως αντικείμενο.

(στο σημείο αυτό υπενθυμίζουμε ότι η χρήση RDF: εδώ είναι μια στενογραφία για το πολύ μεγαλύτερο σε μήκος *uri* - το γεγονός ότι χρησιμοποιούμε κεφαλαία αντί πεζά τονίζει το γεγονός ότι αυτά τα προθέματα είναι αυθαίρετα εκτός αν χρησιμοποιούνται με συνέπεια). Θα μπορούσαμε να δώσουμε σε αυτόν τον κόμβο, ο οποίος είναι τύπου RDF:Statement, ένα URI εάν θέλαμε, αλλά καλύτερα να τον αφήσουμε κενό. Μπορούμε να χρησιμοποιήσουμε αυτόν το νέο κόμβο ως υποκείμενο άλλων δηλώσεων. Ένας επεξεργαστής RDF μπορεί να συμπεράνει ότι αυτές οι δηλώσεις ισχύουν για την αρχική τριπλέτα εφόσον θα γίνει το ταίριασμα του υποκειμένου, του αντικειμένου και του κατηγορήματος.

Αυτή η διαδικασία καλείται *reifying* όπως έχουμε ήδη ορίσει, όσον αφορά στην αρχική δήλωση (φτιάχνουμε κάτι από κάτι άλλο). Σε αυτήν την περίπτωση, έχουμε φτιάξει έναν πόρο από μια δήλωση κατά κάποιον τρόπο. Ο πόρος `rdf:Statement` δε βεβαιώνει το γεγονός που βεβαιώνεται από την αρχική δήλωση - μόνο η αρχική δήλωση μπορεί να το κάνει αυτό. Στην πραγματικότητα, η δήλωση που περιγράφεται από το `rdf:Statement` δεν είναι απαραίτητο να υπάρχει. Αυτή η κατάσταση είναι όπως η φράση, "σκέφτομαι να εκφράσω την επιθυμία να πάω σε ένα εστιατόριο."

Μπορούμε να το πούμε ακόμα κι αν δεν εννοούμε ότι πραγματικά θέλουμε να πάμε σε ένα εστιατόριο. Η δήλωση "reified" μπορεί να μην αντικαταστήσει τη δήλωση που αντιπροσωπεύει, αλλά παρέχει έναν περιορισμένο τρόπο για να μιλήσουμε για εκείνη την αρχική δήλωση (μια δήλωση με τις ίδιες ιδιότητες πάντα, δεδομένου ότι δεν μπορούμε να προσδιορίσουμε μια οποιαδήποτε συγκεκριμένη δήλωση).

5.13. Παράδειγμα: Webscripeter: fusing information

Το WebScripeter: (www.isi.edu/webscripeter) είναι ένα σύστημα που αναπτύσσεται στο πανεπιστήμιο νότιας Καλιφόρνιας που χρησιμοποιεί DAML (γλώσσα σήμανσης πρακτόρων DARPA) για να χαρακτηρίσει ιστοσελίδες, έτσι ώστε οι πληροφορίες για αυτές να μπορούν να εξαχθούν με έναν ομοιόμορφο τρόπο και να συνδυαστούν με

τις πληροφορίες από άλλες ιστοσελίδες. Η DAML είναι μια γλώσσα που χρησιμοποιεί RDF για να καθορίσει τις οντολογίες ή τα λεξιλόγια.

Το WebScripeter είναι ένα εργαλείο που επιτρέπει στους συνηθισμένους χρήστες εύκολα και γρήγορα να συγκεντρώνουν πληροφορίες από τις πολλαπλές, ετερογενείς πηγές του Ιστού.

Ακολουθεί ένα παράδειγμα από μία ιστοσελίδα (το όνομα και ο τηλεφωνικός αριθμός έχουν αλλάξει). Τα μεταδεδομένα πρέπει να είναι σαφή:

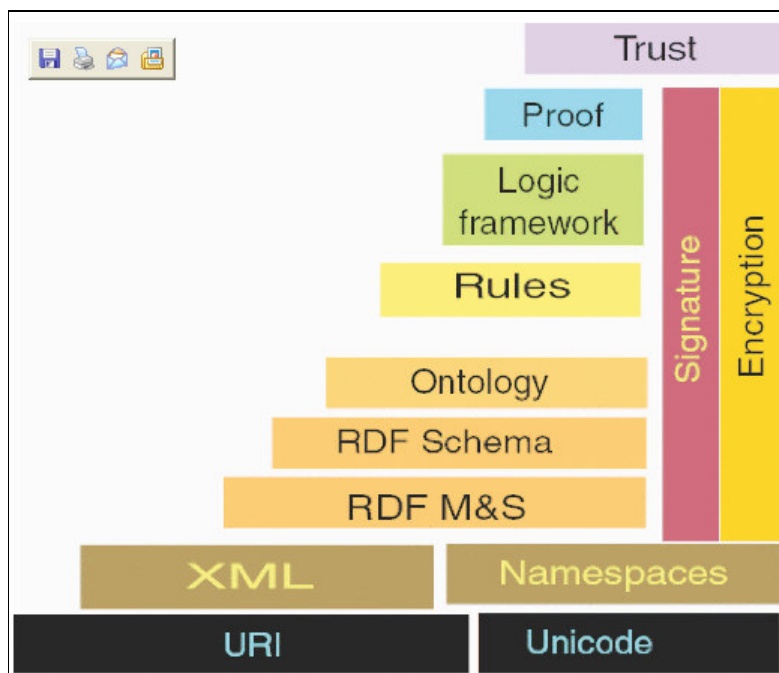
```
<WebScripeterReportRow rdf:ID="5">
  <First>William</First>
  <Last>Sparks</Last>
  <Role>researcher</Role>
  <Role_No>5</Role_No>
  <Role_Label>Researcher</Role_Label>
  <Prefix>sparks</Prefix>
  <Office>1231</Office>
  <Phone>435-555-6789</Phone>
  <Addendum>http://www.isi.edu/~sparks/sparks.daml</Addendum>
  <Interests>user interfaces development, XML/Database
  application, Distributed Resource Allocation and
  scheduling</Interests>
  <Home_Page>http://www.isi.edu/~sparks</Home_Page>
  <Project_Nicknames>CAMERA</Project_Nicknames>
  <Project_Nicknames>CACE-UI-IPT</Project_Nicknames>
  <Project_Pages>http://www.isi.edu/camera</Project_Pages>
  <Project_Pages>
  http://www.xfaster.com/xfast.nsf/pages/useript?opendocument
  </Project_Pages>
  <Thumbnail>
  http://www.isi.edu/webscripeter/sparks-thumb-2001-02-27.jpg
  </Thumbnail>
  <Medium>
  http://www.isi.edu/webscripeter/sparks-medium-2001-02-27.jpg
  </Medium>
  <Full>
  http://www.isi.edu/webscripeter/sparks-full-2001-02-27.jpg
  </Full>
</WebScripeterReportRow>
```

Όλα τα ονόματα στοιχείων, όπως τα First, Last και Role, έχουν καθοριστεί χρησιμοποιώντας DAML και είναι διαθέσιμα σε οποιαδήποτε εφαρμογή WebScripeter. Μια εφαρμογή που μπορεί να αναλύσει RDF και να καταλάβει DAML μπορεί να χρησιμοποιήσει αυτές τις πληροφορίες από πολλές απόψεις.

Παραδείγματος χάρη, εάν ο ιδιοκτήτης αυτής της σελίδας, William Sparks, περιέλαβε τη σήμανση WebScrippter στη σελίδα του, μια εφαρμογή θα μπορούσε να την εξαγάγει και να την χρησιμοποιήσει για να χτίσει έναν πίνακα με τα μέλη της σχολής και τα επαγγελματικά και προσωπικά ενδιαφέροντά τους.

5.14. Τι είναι το σχήμα RDF

Το σχήμα RDF είναι η γλώσσα που τοποθετείται ένα στρώμα πάνω από το RDF. Αυτή η προσέγγιση με τη μορφή στρωμάτων επιδεικνύεται στο σχήμα:



<http://www.nowpublishers.com>

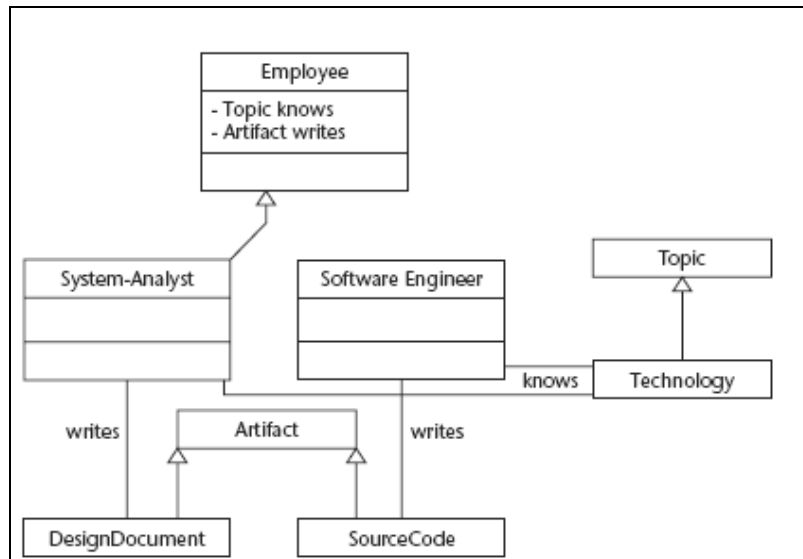
Στη βάση του σωρού είναι οι έννοιες του καθολικού προσδιορισμού (URI) και του Unicode. Επάνω από εκείνες τις έννοιες, βάζουμε τη σύνταξη XML σε στρώσεις (στοιχεία, ιδιότητες) και namespaces για να αποφύγουμε τις συγκρούσεις λεξιλογίου. Πάνω από την XML είναι οι ισχυρισμοί του προτύπου RDF και της σύνταξης που συζητήσαμε, σύμφωνα με τη λογική της τριπλέτας. Εάν χρησιμοποιήσουμε την τριπλέτα για να παραστήσουμε μια κλάση, ιδιότητα κλάσης και τιμή, μπορούμε να δημιουργήσουμε τις ιεραρχίες κλάσεων για την ταξινόμηση και την περιγραφή των αντικειμένων. Αυτός είναι και ο στόχος του σχήματος RDF. Επάνω από το σχήμα RDF έχουμε τις οντολογίες (μια ταξινόμηση είναι μια οντολογία). Επάνω από τις

οντολογίες, μπορούμε να προσθέσουμε τους κανόνες λογικής για τις οντολογίες μας. Επιπλέον, το στρώμα κανόνων παρέχει έναν τυποποιημένο τρόπο για να θέτει ερωτήματα και να φιλτράρει το RDF. Το στρώμα κανόνων είναι είδος μιας ικανότητας "εισαγωγικής λογικής", ενώ το πλαίσιο λογικής θα είναι "προηγμένη λογική." Το πλαίσιο λογικής επιτρέπει στις επίσημες αποδείξεις λογικής να διαμοιράζονται. Τελικά, με τέτοιες αποδείξεις, ένα στρώμα εμπιστοσύνης μπορεί να καθιερωθεί για την εμπιστοσύνη μεταξύ εφαρμογών. Αυτός ο "Ιστός της εμπιστοσύνης" αποτελεί τον τρίτο και τελικό Ιστό στο τριμερές όραμα του Berners Lee (συνεργάσιμος Ιστός, σημασιολογικός Ιστός, Ιστός της εμπιστοσύνης).

Το σχήμα RDF είναι ένα απλό σύνολο τυποποιημένων πόρων RDF και ιδιοτήτων για να επιτρέψει στους ανθρώπους να δημιουργήσουν τα λεξιλόγια RDF. Το πρότυπο στοιχείων που εκφράζεται από το σχήμα RDF είναι το ίδιο πρότυπο στοιχείων που χρησιμοποιείται από τις αντικειμενοστρεφείς γλώσσες προγραμματισμού, όπως η Java. Το πρότυπο στοιχείων για το σχήμα RDF μας επιτρέπει να δημιουργήσουμε τις κλάσεις στοιχείων. Μία κλάση ορίζεται ως μια ομάδα πραγμάτων με κοινά χαρακτηριστικά. Στον αντικειμενοστρεφή προγραμματισμό (OOP), μια κλάση ορίζεται ως ένα πρότυπο ή ένα σχεδιάγραμμα για ένα αντικείμενο που αποτελείται από χαρακτηριστικά (επίσης αποκαλούμενα data members) και συμπεριφορές (επίσης αποκαλούμενες μεθόδους). Ένα αντικείμενο είναι μια απεικόνιση μιας κλάσης. Οι αντικειμενοστρεφείς γλώσσες επιτρέπουν επίσης στις κλάσεις να κληρονομήσουν τα χαρακτηριστικά και τις συμπεριφορές από μια κλάση γονέων (επίσης αποκαλούμενη super class).

5.14.1. Παραλληλισμός UML Class diagram με RDFS

Η βιομηχανία λογισμικού έχει τυποήσει πρόσφατα την αποκαλούμενη Unified Modeling Language (UML) για να διαμορφώσει τις ιεραρχίες κλάσης. Στο παρακάτω σχήμα απεικονίζεται ένα διάγραμμα UML που παριστάνει δύο τύπους υπαλλήλων και τον τρόπο με τον οποίο σχετίζονται με τα θέματα που γράφουν και ξέρουν.



Το σχήμα χρησιμοποιεί διάφορα σύμβολα UML για να δείξει τις έννοιες της κλάσης, της κληρονομικότητας και της ένωσης. Το ορθογώνιο με τρία τμήματα είναι το σύμβολο για μια κλάση. Τα τρία τμήματα είναι για το όνομα κλάσης, τις ιδιότητες κλάσης (μέσο τμήμα), και τις συμπεριφορές ή τις μεθόδους κλάσης (κατώτατο τμήμα). Το σχήμα RDF χρησιμοποιεί μόνο τα πρώτα δύο μέρη μιας κλάσης, δεδομένου ότι χρησιμοποιείται για την απεικόνιση δεδομένων και όχι για συμπεριφορές προγραμματιστικές. Η κληρονομικότητα αναφέρεται στο γεγονός κληρονομιάς από μια υποκατηγορία των χαρακτηριστικών μίας superclass. Το βέλος από την υποκατηγορία στην superclass το υποδεικνύει. Η κληρονομικότητα καλείται συχνά "isa," όπως "a software engineer *is a(n)* employee."

5.14.2. Τα βασικά συστατικά του σχήματος RDF είναι τα εξής:

- `rdfs:Class`. Ένα στοιχείο που καθορίζει μια ομάδα σχετικών πραγμάτων που μοιράζονται ένα σύνολο ιδιοτήτων. Αυτό είναι συνώνυμο με την έννοια του τύπου ή της κατηγορίας. Σε συνδυασμό με τα `rdf:Property`, `rdfs:range`, και `rdfs:domain` κατατάσσει τις ιδιότητες στις κλάσεις. Απαιτεί ένα URI ως προσδιοριστικό που θα οριστεί στην ιδιότητα `rdf:about`.
- `rdfs:label`. Μια ιδιότητα που καθορίζει μια κατανοήσιμη από τον άνθρωπο ετικέτα για την κλάση. Αυτό είναι σημαντικό για να επιδειχθεί το όνομα της κλάσης στις εφαρμογές ακόμα κι αν το επίσημο μοναδικό προσδιοριστικό για την κλάση είναι το URI στην ιδιότητα `rdf:about`.

- `rdfs:subclassOf`. Ένα στοιχείο που διευκρινίζει ότι μια κλάση είναι μια ειδίκευση μιας υπάρχουσας κλάσης. Αυτό ακολουθεί το ίδιο πρότυπο με τη βιολογική κληρονομιά, όπου μια κατηγορία παιδιών μπορεί να κληρονομήσει τις ιδιότητες μιας κατηγορίας γονέων. Η ιδέα της ειδίκευσης είναι ότι μια υποκατηγορία προσθέτει μερικά μοναδικά χαρακτηριστικά σε μια γενική έννοια. Επομένως, η μετάβαση προς τα κάτω στην ιεραρχία κλάσης αναφέρεται ως ειδίκευση, ενώ η άνοδος στην ιεραρχία κλάσης αναφέρεται ως γενίκευση.
- `rdf:Property`. Ένα στοιχείο που καθορίζει μια ιδιότητα μιας κατηγορίας και την ποικιλία των τιμών που μπορεί να αντιπροσωπεύσει. Αυτό χρησιμοποιείται από κοινού με τις ιδιότητες `rdfs:domain` και `rdfs:range`. Είναι σημαντικό να γίνει κατανοητή μια βασική διαφορά μεταξύ της διαμόρφωσης των κλάσεων σε RDFS εναντίον της διαμόρφωσης των κλάσεων στον αντικειμενοστρεφή προγραμματισμό. Στο RDFS υιοθετείται μια από κάτω προς τα επάνω μέθοδος στη διαμόρφωση κλάσης, ενώ στον OOP υιοθετείται μια από επάνω προς τα κάτω μέθοδος. Σε OOP, καθορίζουμε μία κλάση και όλα όσα περιέχει. Σε RDFS, καθορίζουμε τις ιδιότητες και δηλώνουμε σε ποια κλάση ανήκουν. Έτσι, σε OOP πηγαίνουμε κάτω από την κλάση στις ιδιότητες. Σε RDFS, ανεβαίνουμε από τις ιδιότητες στην κλάση.
- `rdfs:domain`. Αυτή η ιδιότητα καθορίζει σε ποια κλάση ανήκει μία ιδιότητα. Η τιμή της ιδιότητας πρέπει να είναι μια ήδη καθορισμένη κλάση.
- `rdf:type`. Μια τυποποιημένη ιδιότητα που καθορίζει ότι ένα υποκείμενο RDF είναι ενός τύπου που καθορίζεται σε ένα σχήμα RDF. Παραδείγματος χάρη, θα μπορούσαμε να πούμε ότι ένα πρόσωπο με την ταυτότητα 865 είναι ένας τύπος υπαλλήλου που περιγράφεται ως εξής:

```

<rdf:Description rdf:about=
"http://www.mybiz.com/staff/ID/865">
  <rdf:type rdf:resource = "&example_chp5;Employee">

```
- `rdfs:subPropertyof`. Μια ιδιότητα που δηλώνει ότι η ιδιότητα που αποτελεί το αντικείμενο της δήλωσης είναι ένα `subproperty` μιας άλλης υπάρχουσας ιδιότητας. Αυτό το χαρακτηριστικό γνώρισμα υπερβαίνει πραγματικά τις κοινές γλώσσες OOP όπως την Java και τη C# οι οποίες επιτρέπουν μόνο κληρονομιά κλάσης. Ένα παράδειγμα αυτού θα ήταν να κληρονομηθεί μια

ιδιότητα αποκαλούμενη "Σαββατοκόριακο" που θα ήταν ένα subPropertyOf του "εβδομάδα."

- rdfs:seeAlso. Μια ιδιότητα που επιτρέπει να αναφερθούμε σε έναν πόρο που μπορεί να παρέχει τις πρόσθετες πληροφορίες RDF για τον τρέχοντα πόρο
- rdfs:isDefinedBy. Μια ιδιότητα που καθορίζει το namespace ενός θέματος. Πρόκειται για ένα subPropertyOf του rdfs:seeAlso.
- rdfs:comment. Μια ιδιότητα για παροχή πρόσθετων περιγραφικών πληροφοριών, για να εξηγήσει τις κλάσεις και τις ιδιότητες σε άλλους χρήστες του σχήματος. Όπως στον προγραμματισμό, τα καλά σχόλια είναι ουσιαστικά για την κατανόηση και την ενθάρρυνση υιοθέτησης.
- rdfs:Literal. Μια ιδιότητα που αντιπροσωπεύει μια σταθερή τιμή η οποία αντιπροσωπεύεται ως σειρά χαρακτήρων.
- rdfs:XMLLiteral. Μια ιδιότητα που αντιπροσωπεύει μια σταθερή τιμή που είναι καλοσηματισμένη XML. Αυτό επιτρέπει στην XML να ενσωματωθεί εύκολα σε RDF.

Εκτός από τις κλάσεις και τις ιδιότητες που περιγράφονται, το σχήμα RDF περιγράφει τις κλάσεις και τις ιδιότητες για τις έννοιες RDF των containers και του reification. Για τα containers, το σχήμα RDF καθορίζει rdfs:Container, rdf:Bag, rdf:Seq, rdf:Alt, rdfs:member, και rdfs:ContainerMembershipProperty. Για το reification, το σχήμα RDF καθορίζει rdf:Statement, rdf:subject, rdf:predicate, και rdf:object. Στη συνέχεια ακολουθεί το RDF Schema για το UML διάγραμμα των δύο τύπων υπαλλήλων που αναδείξαμε παραπάνω:

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<!DOCTYPE rdf:RDF [
  <!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
  <!ENTITY example_chp5
'http://protege.stanford.edu/example-chp5#'>
  <!ENTITY rdfs 'http://www.w3.org/TR/1999/PR-rdf-schema-19990303#'>
]>
<rdf:RDF xmlns:rdf="&rdf;"
  xmlns:example_chp5="&example_chp5;"
  xmlns:rdfs="&rdfs;">
<rdfs:Class rdf:about="&example_chp5;Artifacts"
  rdfs:label="Artifacts">
  <rdfs:subClassOf rdf:resource="&rdfs;Resource"/>
</rdfs:Class>
<rdfs:Class rdf:about="&example_chp5;DesignDocument"
  rdfs:label="DesignDocument">
```

```

    <rdfs:subClassOf rdf:resource="&example_chp5;Artifacts"/>
  </rdfs:Class>
  <rdfs:Class rdf:about="&example_chp5;Employee"
    rdfs:label="Employee">
    <rdfs:subClassOf rdf:resource="&rdfs;Resource"/>
  </rdfs:Class>
  <rdfs:Class rdf:about="&example_chp5;Software-Engineer"
    rdfs:label="Software-Engineer">
    <rdfs:subClassOf rdf:resource="&example_chp5;Employee"/>
  </rdfs:Class>

  <!-- Classes SourceCode, System-Analyst, Technology, and Topic omitted
  for brevity. They are similar to the above Classes. -->

  <rdf:Property rdf:about="&example_chp5;knows"
    rdfs:label="knows">
    <rdfs:domain rdf:resource="&example_chp5;Employee"/>
    <rdfs:range rdf:resource="&example_chp5;Topic"/>
  </rdf:Property>
  <rdf:Property rdf:about="&example_chp5;writes"
    rdfs:label="writes">
    <rdfs:range rdf:resource="&example_chp5;Artifacts"/>
    <rdfs:domain rdf:resource="&example_chp5;Employee"/>
  </rdf:Property>
</rdf:RDF>

```

RDF Schema για το UML διάγραμμα των δύο τύπων υπαλλήλων

Παρατηρούμε ότι οι κλάσεις του σχήματος RDF δεν καθορίζονται χρησιμοποιώντας `rdf:type` ή `rdf:about`. Αντ' αυτού, χρησιμοποιείται μια σύντμηση αποκαλούμενη "typed node element."

Παρακάτω παρατίθεται ένα έγγραφο RDF που παράγεται από το Protégé, προσαρμομένο στο σχήμα RDF που προηγείται.

```

<?xml version='1.0' encoding='ISO-8859-1'?>
<!DOCTYPE rdf:RDF [
  <!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
  <!ENTITY example_chp5 'http://protege.stanford.edu/example-chp5#'>
  <!ENTITY rdfs 'http://www.w3.org/TR/1999/PR-rdf-schema-19990303#'>
]>
<rdf:RDF xmlns:rdf="&rdfs;"
  xmlns:example_chp5="&example_chp5;"
  xmlns:rdfs="&rdfs;">
  <example_chp5:SourceCode rdf:about="&example_chp5;example-chp5_00015"
    example_chp5:name="stuff.java"
    rdfs:label="example-chp5_00015"/>
  <example_chp5:System-Analyst rdf:about="&example_chp5;examplechp5_
00016"
    example_chp5:name="Jane Jones"
    rdfs:label="example-chp5_00016">

```

```

<example_chp5:writes rdf:resource="&example_chp5;example-chp5_00017"/>
</example_chp5:System-Analyst>
<example_chp5:DesignDocument rdf:about="&example_chp5;examplechp5_
00017"
  example_chp5:name="system.sdd"
  rdfs:label="example-chp5_00017" />
<example_chp5:Software-Engineer rdf:about="&example_chp5;examplechp5_
00018"
  example_chp5:name="John Doe"
  rdfs:label="example-chp5_00018">
  <example_chp5:writes rdf:resource="&example_chp5;example-chp5_00015" />
</example_chp5:Software-Engineer>
</rdf:RDF>

```

Έγγραφο RDF

5.14.3. Παράδειγμα Griffith University

5.14.3.1 Κλάσεις και ιδιότητες

Όσον αφορά στο παράδειγμά μας σχετικά με τα μαθήματα και τις διαλέξεις του πανεπιστημίου Griffith, αρχικά πρέπει να γίνουμε συγκεκριμένοι ως προς τα θέματα για τα οποία θα μιλήσουμε ακριβώς. Εδώ, κάνουμε μία σαφή διάκριση: θα αναφερθούμε σε συγκεκριμένους καθηγητές, όπως ο Kostas Kalogirou , και συγκεκριμένα μαθήματα, όπως τα Διακριτά Μαθηματικά. Τα συγκεκριμένα βέβαια έχουν ήδη υλοποιηθεί σε RDF, αλλά η διαφορά έγκειται στο ότι στην περίπτωση του RDF αναφερόμαστε σε μεμονωμένα αντικείμενα (μαθήματα), ενώ στην περίπτωση του RDFS κάνουμε λόγο για κλάσεις που προσδιορίζουν τους τύπους των αντικειμένων.

Οι κλάσεις είναι σύνολα στοιχείων, instances των οποίων αποτελούν τα μεμονωμένα αντικείμενα. Έχουμε ήδη προσδιορίσει τη σχέση μεταξύ κλάσεων και instances σε RDF με το `rdf:type`.

Μία πολύ σημαντική χρήση των κλάσεων είναι για την επιβολή περιορισμών σε ό,τι μπορεί να έχουμε δηλώσει σε ένα έγγραφο RDF χρησιμοποιώντας το RDF Schema. Για παράδειγμα, δεν επιτρέπονται εκφράσεις όπως: “Τα Διακριτά Μαθηματικά διδάσκονται από την Πληροφορική”, καθώς τα μαθήματα διδάσκονται αποκλειστικά και μόνο από καθηγητές. Αυτό σημαίνει ότι πρέπει να ορίσουμε έναν περιορισμό στην ιδιότητα “is taught by”, δηλαδή στο range της συγκεκριμένης ιδιότητας.

5.14.3.2 Ιεραρχίες κλάσεων και κληρονομικότητα

Ας υποθέσουμε ότι διαθέτουμε τις εξής κλάσεις:

staff members *assistant professors*
academic staff members *administrative staff members*
professors *technical support staff members*
associate professors

Οι συγκεκριμένες κλάσεις δεν είναι ασυσχέτιστες μεταξύ τους, καθώς, για παράδειγμα, ο professor είναι πάντα και academic staff member. Αυτό σημαίνει ότι ο professor είναι υποκλάση του academic staff member, ή διαφορετικά, ότι ο academic staff member είναι υπερκλάση του professor. Κατά αυτό τον τρόπο προκύπτει η ιεραρχία του σχήματος:



Γενικότερα, μία κλάση A είναι υποκλάση της B αν κάθε instance της A είναι instance και της B, χωρίς να απαιτείται κάποια αυστηρή ιεραρχία ωστόσο. Δεν είναι απαραίτητο δηλαδή στο RDF Schema, ένα γράφημα υποκλάσεων να είναι ένα δέντρο. Μία κλάση μπορεί να έχει πολλές υπερκλάσεις.

Η ιεραρχική οργάνωση των κλάσεων έχει ιδιαίτερη πρακτική σημασία, όπως διευκρινίζουμε παρακάτω. Ας θεωρήσουμε ότι ισχύει ο εξής περιορισμός (range restriction):

Courses must be taught by academic staff members only.

Έστω επίσης ότι ο Michael Maher είναι καθηγητής. Τότε, σύμφωνα με τον παραπάνω περιορισμό, δεν επιτρέπεται σε αυτόν να διδάσκει μαθήματα, καθώς δεν υπάρχει καμία δήλωση που να ορίζει αυτόν ως academic staff member. Αυτό λοιπόν που πραγματοποιείται στο RDF Schema είναι ότι ο Michael Maher κληρονομεί τη δυνατότητα να διδάξει από την κλάση academic staff member.

Με αυτό τον τρόπο, το RDF Schema ορίζει τη σημασιολογία του *is subclass of*, ώστε αυτή να ισχύει και να χρησιμοποιείται από κάθε RDF επεξεργαστή, και να μη χρειάζεται κάθε εφαρμογή να δίνει δική της ερμηνεία. Λόγω αυτών των σημασιολογικών ορισμών, το RDFS είναι μία γλώσσα που χρησιμοποιείται για τον προσδιορισμό της σημασιολογίας διαφόρων τομέων. Είναι με άλλα λόγια μία γλώσσα για οντολογίες, σε σχετικά απλή μορφή.

Οι κλάσεις, η κληρονομικότητα και οι ιδιότητες είναι γνωστές από τον αντικειμενοστραφή προγραμματισμό, και όσο και αν υπάρχουν ομοιότητες, υπάρχουν και διαφορές.

Στο αντικειμενοστραφή προγραμματισμό, η κλάση ενός αντικειμένου ορίζει τις ιδιότητες που σχετίζονται με αυτό και για να προσθέσουμε μία καινούρια ιδιότητα στην κλάση πρέπει να την τροποποιήσουμε.

Αντίθετα, στο RDFS, οι ιδιότητες ορίζονται σφαιρικά, εννοώντας ότι δεν αποτελούν τμήμα του ορισμού μιας κλάσης, επιτρέποντας έτσι να ορίσουμε ιδιότητες για μία κλάση χωρίς να τη αλλάξουμε.

Το πλεονέκτημα στην περίπτωση αυτή είναι ότι μπορούμε να χρησιμοποιήσουμε ήδη ορισμένες κλάσεις και να τις προσαρμόσουμε στις απαιτήσεις μας ορίζοντας επιπρόσθετες ιδιότητες.

5.14.3.3 Ιεραρχίες ιδιοτήτων

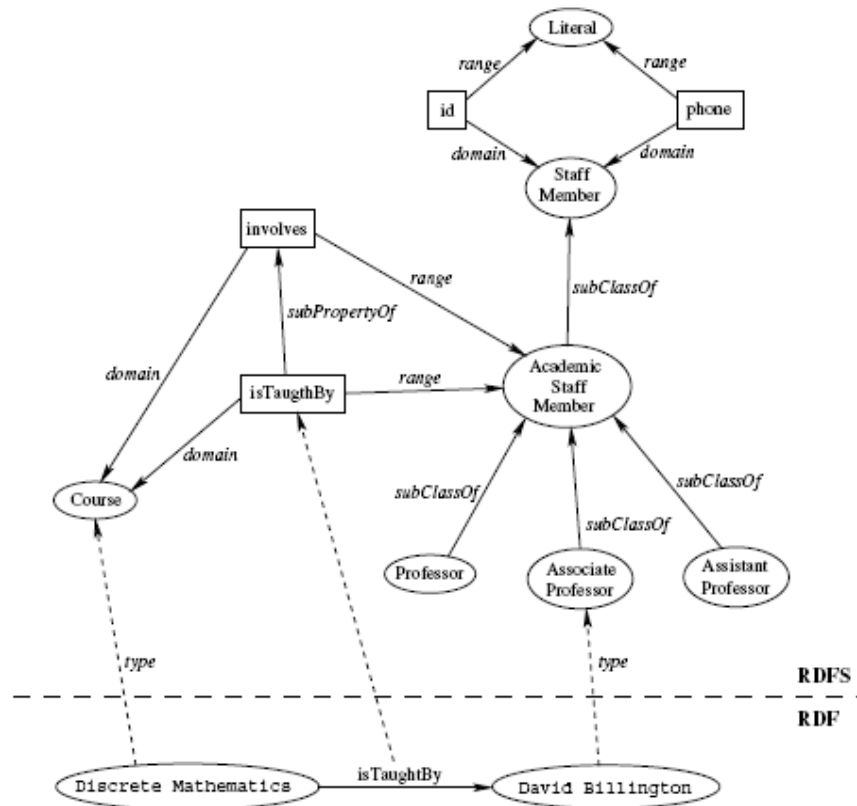
Ιεραρχικές σχέσεις μπορούν να οριστούν όπως είδαμε μεταξύ κλάσεων. Κάτι αντίστοιχο μπορεί να πραγματοποιηθεί και για τις ιδιότητες. Για παράδειγμα η ιδιότητα “is taught by” αποτελεί subproperty της “involves”. Έτσι, αν ένα μάθημα *a* διδάσκεται από το ακαδημαϊκό μέλος *c*, τότε το *a* λέμε ότι involves το *c*, χωρίς να ισχύει απαραίτητα το αντίθετο. Ενδέχεται για παράδειγμα ο *c* να διορθώνει τα γραπτά που αφορούν στο συγκεκριμένο μάθημα χωρίς να το διδάσκει.

5.15. RDF και στρώματα RDFS

Ας θεωρήσουμε τη δήλωση RDF:

Discrete Mathematics is taught by Kostas Kalogirou.

Οι κλάσεις που σχετίζονται με την συγκεκριμένη δήλωση αφορούν σε *lecturers*, *academic staff members*, *staff members*, *first-year courses*, και ιδιότητες όπως *is taught by*, *involves*, *phone*, *employee id*. Η διαστρωμάτωση των RDF και RDFS για το συγκεκριμένο παράδειγμα φαίνονται στο ακόλουθο σχήμα, όπου τα τετράγωνα αντιπροσωπεύουν ιδιότητες, οι ελλείψεις πάνω από τη διαχωριστική γραμμή κλάσεις και κάτω από αυτήν instances. Το συγκεκριμένο γράφημα είναι γραμμένο σε RDF Schema, με τα εξής συστατικά του: *subClassOf*, *Class*, *Property*, *subPropertyOf*, *Resource* και λοιπά.



5.15.1. RDF Schema: Η γλώσσα

Το RDF Schema παρέχει έναν πρότυπο τρόπο για την έκφραση πληροφοριών. Η πρώτη απόφαση που καλούμαστε να πάρουμε είναι ποια γλώσσα θα χρησιμοποιήσουμε. Με δεδομένο ότι το RDF Schema χρησιμοποιεί resources και properties, είναι προφανές ότι μπορούμε να χρησιμοποιήσουμε την ίδια την RDF. Η επιλογή μας αυτή μπορεί να δικαιολογηθεί και με βάση το προηγούμενο σχήμα, όπου ουσιαστικά απεικονίζουμε έναν γράφο με labels ο οποίος μπορεί να κωδικοποιηθεί σε RDF. Άλλωστε η RDF μας επιτρέπει να κάνουμε δηλώσεις για οποιοδήποτε resource και καθετί που έχει URI μπορεί να αποτελέσει resource. Έτσι, αν επιθυμούμε να ορίσουμε την κλάση lecturer ως υποκλάση του academic staff member μπορούμε:

1. να ορίσουμε τα resources lecturer, academic staff member και subClass of
2. να ορίσουμε ότι το subClass of αποτελεί ιδιότητα
3. να γράψουμε την τριπλέτα (subClass of, lecturer, academicStaffMember)

Όλα αυτά τα βήματα επιτρέπονται στην RDF. Έτσι, ένα έγγραφο RDFS είναι απλά ένα έγγραφο RDF, και χρησιμοποιούμε σύνταξη RDF βασισμένη σε XML.

Ακολουθούν τα βασικά στοιχεία του RDF Schema.

5.15.1.1 Κύριες κλάσεις

Οι βασικές κλάσεις είναι:

rdfs:Resource, η κλάση για όλα τα resources.

rdfs:Class, η κλάση για όλες τις κλάσεις.

rdfs:Literal, η κλάση για όλα τα strings. Τα literals αποτελούν τον μόνο τύπο σε RDF/RDFS.

rdf:Property, η κλάση για όλες τις ιδιότητες.

rdf:Statement, η κλάση για όλες τις reified δηλώσεις.

Για παράδειγμα, η κλάση lecturer μπορεί να προσδιοριστεί ως εξής:

```
<rdfs:Class rdf:ID="lecturer">
```


...
</rdfs:Class>

5.15.1.2 Βασικά properties για τον προσδιορισμό των συσχετίσεων

rdf:type, το οποίο συσχετίζει ένα resource στην κλάση του. Το resource δηλώνεται ως instance αυτής της κλάσης.

rdfs:subClassOf, το οποίο συσχετίζει μία κλάση με μία από τις υπερκλάσεις της. Όλα τα instances της κλάσης αποτελούν Instances και της υπερκλάσης (σημειώνουμε ότι μία κλάση μπορεί να αποτελεί υποκλάση περισσότερων από μίας κλάσης). Για παράδειγμα η κλάση femaleProfessor μπορεί να είναι υποκλάση τόσο της female όσο και της professor.

rdfs:subPropertyOf, συσχετίζει ένα property σε ένα από τα superproperties αυτού. Για παράδειγμα, η δήλωση ότι όλοι οι lecturers είναι staff members:

```
<rdfs:Class rdf:about="lecturer">  
<rdfs:subClassOf rdf:resource="staffMember"/>  
</rdfs:Class>
```

Σημειώνουμε ότι τα rdfs:subClassOf και rdfs:subPropertyOf είναι μεταβατικά εξ ορισμού. Επίσης, το rdfs:Class είναι υποκλάση του rdfs:Resource (κάθε κλάση είναι ένα resource), και το rdfs:Resource αποτελεί instance του rdfs:Class (το rdfs:Resource είναι η κλάση για όλα τα resources, άρα είναι και αυτό κλάση!). Για τον ίδιο λόγο, κάθε κλάση αποτελεί instance του rdfs:Class.

5.15.1.3 Βασικά properties για τους περιορισμούς των properties

rdfs:domain, που ορίζει το domain ενός property P , δηλαδή την κλάση εκείνων των resources που μπορεί να αποτελέσουν υποκείμενο σε μία τριπλέτα με κατηγορημα P . Αν το domain δεν έχει προσδιοριστεί, τότε κάθε resource μπορεί να είναι υποκείμενο.

rdfs:range, που ορίζει το range μιας ιδιότητας P , δηλαδή, την κλάση εκείνων των resources που εμφανίζονται ως τιμές σε μία τριπλέτα με κατηγορημα P .

Για παράδειγμα, η δήλωση ότι το τηλέφωνο αφορά σε staff members μόνο και ότι η τιμή του είναι literal:

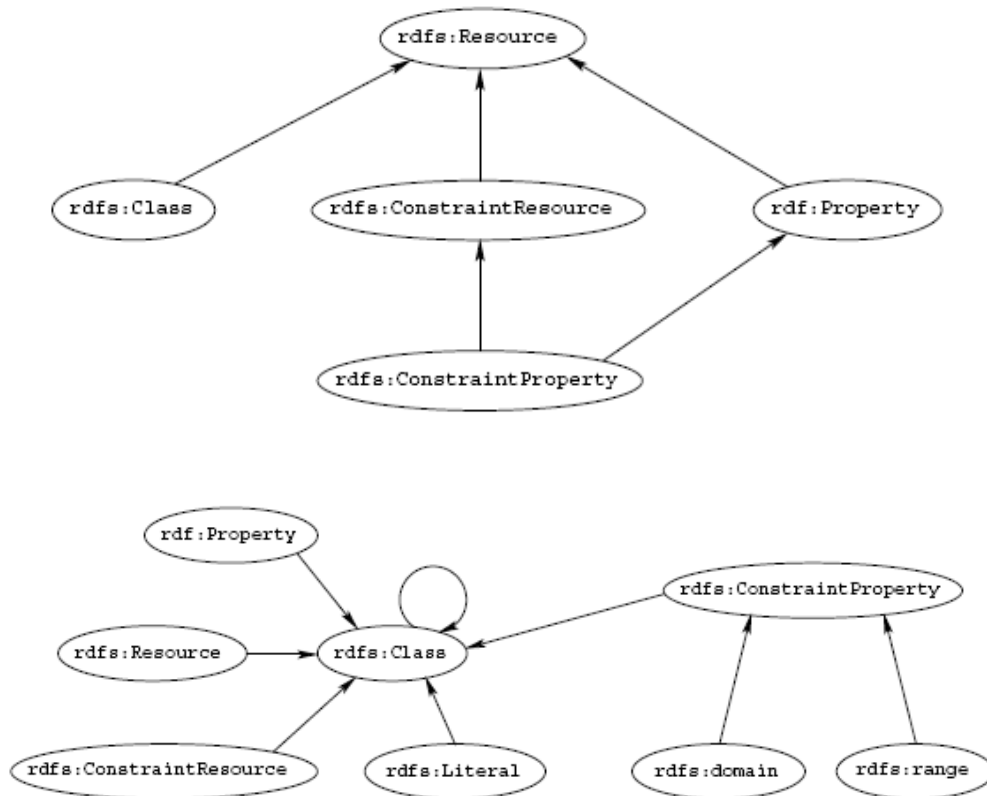
```
<rdf:Property rdf:ID="phone">  
<rdfs:domain rdf:resource="#staffMember"/>  
<rdfs:range rdf:resource="&rdf:Literal"/>  
</rdf:Property>
```

Στο RDF Schema υπάρχουν επίσης τα:

rdfs:ConstraintResource, η κλάση για τους περιορισμούς

rdfs:ConstraintProperty, που περιλαμβάνει όλα τα properties που προσδιορίζουν τους περιορισμούς. Έχει μόνο δύο instances, *rdfs:domain* και *rdfs:range*. Είναι υποκλάση του *rdfs:ConstraintResource* και *rdf:Property*.

Τα παρακάτω σχήματα παρουσιάζουν τις συσχετίσεις των διαφόρων πρότυπων στοιχείων.



5.15.1.4 Χρήσιμα Properties για Reification

rdf:subject, που συσχετίζει μία reified δήλωση με το υποκείμενό της
rdf:predicate, που συσχετίζει μία reified δήλωση με το κατηγορημά της
rdf:object, που συσχετίζει μία reified δήλωση με το αντικείμενό της

5.15.1.5 Container κλάσεις

rdf:Bag, η κλάση των bags

rdf:Seq, η κλάση των sequences

rdf:Alt, η κλάση των alternatives.

rdfs:Container, που αποτελεί superclass όλων των κλάσεων container, συμπεριλαμβανομένων και των τριών παραπάνω

.

5.15.1.6 Επιπλέον χρήσιμα Properties

Ένα resource μπορεί να οριστεί και να περιγραφεί σε πολλά σημεία του Web. Τα ακόλουθα properties μας επιτρέπουν να ορίσουμε links σε αυτές τις διευθύνσεις:

rdfs:seeAlso συσχετίζει ένα resource σε κάποιο άλλο που το επεξηγεί

rdfs:isDefinedBy αποτελεί subproperty του *rdfs:seeAlso* και συσχετίζει ένα resource στην τοποθεσία όπου βρίσκεται ο ορισμός του (τυπικά ένα RDF Schema).

Συχνά είναι χρήσιμο να παρέχουμε περισσότερες πληροφορίες για τους αναγνώστες, το οποίο μπορεί να υλοποιηθεί με τα ακόλουθα properties:

rdfs:comment: Σχόλια που συνοδεύουν ένα resource.

rdfs:label: Ένα label (όνομα) συνδέεται με ένα resource, το οποίο ενδεχομένως να αποτελεί και το όνομα του κόμβου στη γραφική αναπαράσταση του εγγράφου RDF.

5.15.2. Παράδειγμα: University

Αναφερόμαστε στα μαθήματα και στους lecturers και παρουσιάζουμε ένα μοντέλο που μας βοηθά να αποκτήσουμε μια εικόνα του συγκεκριμένου τομέα, δηλαδή μια οντολογία.

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

  <rdfs:Class rdf:ID="lecturer">
    <rdfs:comment>
      The class of lecturers
      All lecturers are academic staff members.
    </rdfs:comment>
    <rdfs:subClassOf rdf:resource="#academicStaffMember"/>
  </rdfs:Class>

  <rdfs:Class rdf:ID="academicStaffMember">
    <rdfs:comment>
      The class of academic staff members
    </rdfs:comment>
    <rdfs:subClassOf rdf:resource="#staffMember"/>
  </rdfs:Class>

  <rdfs:Class rdf:ID="staffMember">
    <rdfs:comment>The class of staff members</rdfs:comment>
  </rdfs:Class>

  <rdfs:Class rdf:ID="course">
    <rdfs:comment>The class of courses</rdfs:comment>
  </rdfs:Class>

  <rdf:Property rdf:ID="involves">
    <rdfs:comment>
      It relates only courses to lecturers.
    </rdfs:comment>
    <rdfs:domain rdf:resource="#course"/>
    <rdfs:range rdf:resource="#lecturer"/>
  </rdf:Property>

  <rdf:Property rdf:ID="isTaughtBy">
    <rdfs:comment>
      Inherits its domain ("course") and range ("lecturer")
      from its superproperty "involves"
    </rdfs:comment>
    <rdfs:subPropertyOf rdf:resource="#involves"/>
  </rdf:Property>
```

```
<rdf:Property rdf:ID="phone">
  <rdfs:comment>
    It is a property of staff members
    and takes literals as values.
  </rdfs:comment>
  <rdfs:domain rdf:resource="#staffMember"/>
  <rdfs:range rdf:resource="&rdf;Literal"/>
</rdf:Property>
</rdf:RDF>
```

6. OWL (Web Ontology Language)

6.1. Οι περιορισμοί της εκφραστικής δύναμης των RDF και RDF Schema

Τα RDF και RDFS επιτρέπουν την αναπαράσταση κάποιας οντολογικής γνώσης, όσον αφορά στην οργάνωση των λεξιλογίων σε ιεραρχίες: σχέσεις υποκατηγοριών (subproperty), περιορισμοί domain και range, instances κλάσεων [7,13]. Εντούτοις, διάφορα άλλα χαρακτηριστικά γνωρίσματα λείπουν, όπως για παράδειγμα:

- Ο τοπικός χαρακτήρας των ιδιοτήτων

Το `rdfs:range` καθορίζει το range μιας ιδιότητας, του *eats* για παράδειγμα, για όλες τις κλάσεις. Κατά συνέπεια, στο RDFS δεν μπορούμε να δηλώσουμε τους περιορισμούς range που ισχύουν για μερικές κλάσεις μόνο. Για παράδειγμα, δεν μπορούμε να πούμε ότι οι αγελάδες τρώνε μόνο φυτό, ενώ άλλα ζώα μπορούν να φάνε κρέας, επίσης.

- Disjointness των κλάσεων.

Μερικές φορές επιθυμούμε να πούμε ότι οι δύο κλάσεις είναι disjoint. Παραδείγματος χάρη, το αρσενικό και το θηλυκό είναι μεταξύ τους disjoint. Αλλά στο RDFS μπορούμε μόνο να δηλώσουμε σχέσεις υποκλάσεων, π.χ. το female είναι μια υποκατηγορία του person.

- Boolean συνδυασμοί κλάσεων.

Μερικές φορές επιθυμούμε να δημιουργήσουμε νέες κλάσεις που προκύπτουν από τον συνδυασμό άλλων, χρησιμοποιώντας την ένωση, τη διατομή και το συμπλήρωμα. Παραδείγματος χάρη, θέλουμε να ορίσουμε το person ως την disjoint ένωση των κλάσεων male και female. Το σχήμα RDF δεν επιτρέπει τέτοιους ορισμούς.

- Περιορισμοί ως προς το πλήθος των στοιχείων ενός συνόλου.

Μερικές φορές επιθυμούμε να τοποθετήσουμε περιορισμούς ως προς το πόσες ευδιάκριτες τιμές μια ιδιότητα μπορεί ή πρέπει να πάρει. Παραδείγματος χάρη, θα επιθυμούσαμε να πούμε ότι ένα πρόσωπο έχει ακριβώς δύο γονείς, ή ότι μια σειρά μαθημάτων διδάσκεται από τουλάχιστον έναν καθηγητή. Πάλι, τέτοιοι περιορισμοί είναι αδύνατο να εκφραστούν στο σχήμα RDF.

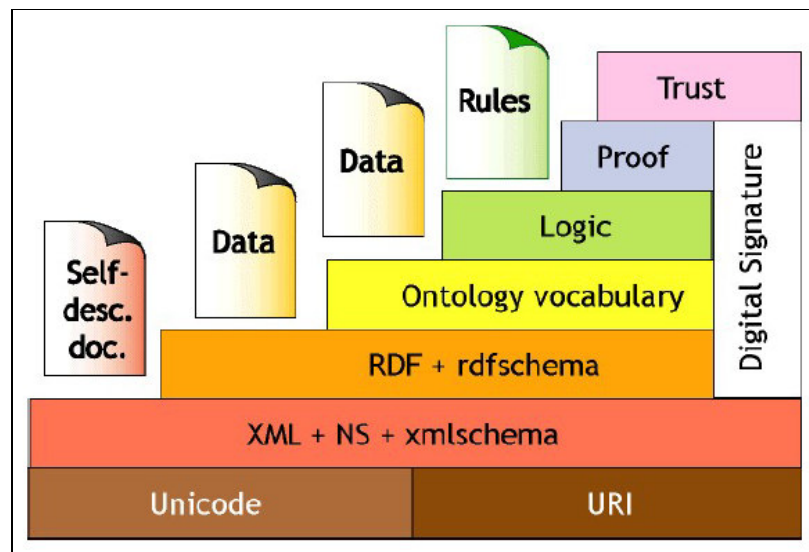
- Ειδικά χαρακτηριστικά των ιδιοτήτων.

Μερικές φορές είναι χρήσιμο να ειπωθεί ότι μια ιδιότητα είναι μεταβατική (όπως "μεγαλύτερο από"), μοναδική (όπως "is mother of"), ή το αντίστροφο μιας άλλης ιδιότητας (όπως "eats" και "is eaten by").

Κατά συνέπεια χρειαζόμαστε μια γλώσσα οντολογίας που είναι πλουσιότερη από το σχήμα RDF, μια γλώσσα που προσφέρει αυτά τα χαρακτηριστικά γνωρίσματα και περισσότερα, ενώ ταυτόχρονα να παρέχει και την κατάλληλη υποστήριξη όσον αφορά στο reasoning.

6.2. Συμβατότητα της OWL με τα RDF/RDFS

Η OWL θα ήταν μια επέκταση του σχήματος RDF, υπό την έννοια ότι θα χρησιμοποιούσε την έννοια των RDF κλάσεων και ιδιοτήτων (rdfs:Class, rdfs:subClassOf, κ.λπ....) και θα προσέθετε γλωσσικά πρότυπα για να υποστηρίξει την πλουσιότερη εκφραστικότητα που απαιτείται [7]. Μια τέτοια επέκταση του σχήματος RDF θα ήταν επίσης σύμφωνη με τη βαλμένη σε στρώσεις αρχιτεκτονική του σημασιολογικού Ιστού (βλ. σχήμα).



Δυστυχώς, η απλή επέκταση του RDFS θα λειτουργούσε αρνητικά ως προς την εκφραστική δύναμη και το αποδοτικό reasoning. Το σχήμα RDF διαθέτει μερικούς πολύ ισχυρούς τύπους, αλλά δομές όπως οι `rdfs:Class` (η κλάση όλων των κλάσεων) και `rdf:Property` (η κλάση όλων των ιδιοτήτων) είναι πολύ εκφραστικές και θα οδηγούσαν σε ανεξέλεγκτες υπολογιστικές ιδιότητες εάν η λογική επεκτεινόταν.

6.3. Τρία είδη OWL

6.3.1. OWL Full

Πρόκειται για ολόκληρη τη γλώσσα OWL. Το πλεονέκτημα της OWL Full είναι ότι είναι πλήρως συμβατή με την RDF, τόσο συντακτικά όσο και σημασιολογικά: οποιοδήποτε έγγραφο RDF είναι επίσης ένα πλήρες έγγραφο OWL, και οποιοδήποτε έγκυρο συμπέρασμα RDF/RDFS είναι επίσης ένα έγκυρο και πλήρες συμπέρασμα OWL.

6.3.2. OWL DL

Προκειμένου να επανακτηθεί η υπολογιστική αποδοτικότητα, η OWL DL (Description Logic) είναι μια “υπογλώσσα” της OWL Full, που περιορίζει τη χρήση των constructors των OWL και RDF. Το πλεονέκτημά της είναι ότι επιτρέπει την αποδοτική υποστήριξη reasoning. Το μειονέκτημα είναι ότι χάνουμε την πλήρη συμβατότητα με RDF: ένα έγγραφο RDF γενικά θα πρέπει να επεκταθεί με κάποιους τρόπους και να περιοριστεί σε άλλα σημεία προτού να είσαι ένα έγκυρο έγγραφο OWL DL. Κάθε έγκυρο έγγραφο OWL DL είναι ένα έγκυρο έγγραφο RDF.

Σε μια οντολογία OWL DL πρέπει να υπακούσουμε στους εξής περιορισμούς:

- *Διαχωρισμός του λεξιλογίου*: Οποιοδήποτε resource επιτρέπεται να ανήκει μόνο σε μία κλάση, έναν τύπο δεδομένων, ένα object property, μία τιμή δεδομένων, ένα individual, ή τμήμα του ενσωματωμένου λεξιλογίου, και όχι σε παραπάνω από ένα από αυτά. Αυτό σημαίνει για παράδειγμα ότι μία κλάση δεν μπορεί να είναι ταυτόχρονα ένα individual, ή ότι μία ιδιότητα δεν μπορεί να παίρνει μερικές τιμές

από έναν τύπο δεδομένων και μερικές από μία κλάση (κάτι τέτοιο θα την καθιστούσε τόσο ιδιότητα data type όσο και ιδιότητα object).

- *Σαφής διάκριση.* Τα resources δεν αρκεί να είναι διαχωρισμένα όπως είπαμε παραπάνω, αλλά αυτό πρέπει να έχει δηλωθεί και ρητά. Για παράδειγμα, αν μια οντολογία περιλαμβάνει τα ακόλουθα:

```
<owl:Class rdf:ID="C1">  
  <rdfs:subClassOf rdf:about="#C2" />  
</owl:Class>
```

αυτό ήδη συνεπάγεται ότι η C2 είναι μία κλάση. Ωστόσο, μία οντολογία OWL DL πρέπει ρητά να δηλώσει αυτή την πληροφορία:

```
<owl:Class rdf:ID="C2"/>
```

- *Διαχωρισμός ιδιοτήτων.* Σύμφωνα με τον πρώτο περιορισμό, οι object properties και οι data type properties είναι disjoint. Κάτι τέτοιο υπονοεί ότι το παρακάτω δεν μπορεί ποτέ να οριστεί για data type properties:

```
owl:inverseOf,  
owl:FunctionalProperty,  
owl:InverseFunctionalProperty, και  
owl:SymmetricProperty.
```

- Δεν επιτρέπονται περιορισμοί πληθυκότητας σε μεταβατικές ιδιότητες (ή σε subproperties, οι οποίες είναι επίσης μεταβατικές φυσικά).

- Ανώνυμες κλάσεις επιτρέπονται μόνο ως domain και range είτε του owl:equivalentClass είτε του owl:disjointWith, και ως range (και όχι domain) του rdfs:subClassOf.

6.3.3. OWL Lite

Ένας επιπλέον περιορισμός της OWL DL σε ένα υποσύνολο των constructors της γλώσσας. Για παράδειγμα, η OWL Lite δεν περιλαμβάνει συγκεκριμένες κλάσεις, δηλώσεις disjointness και την αυθαίρετη πληθυκότητα. Το πλεονέκτημα αυτών είναι να προκύπτει έτσι μία γλώσσα απλούστερη στην υλοποίηση.

Μία οντολογία OWL Lite πρέπει να είναι οντολογία OWL DL και πρέπει επιπλέον να ικανοποιεί τους παρακάτω περιορισμούς:

- Οι constructors `owl:oneOf`, `owl:disjointWith`, `owl:unionOf`, `owl:complementOf` και `owl:hasValue` δεν επιτρέπονται.
- Δηλώσεις πληθυκότητας (*minimal*, *maximal*, και *exact cardinality*) μπορούν μόνο να πραγματοποιηθούν στις τιμές 0 ή 1 και όχι σε αυθαίρετους μη-αρνητικούς ακεραίους.
- Δηλώσεις `owl:equivalentClass` δεν μπορούν να γίνουν για ανώνυμες κλάσεις αλλά μόνο μεταξύ προσδιοριστικών κλάσεων.

Η επιλογή μεταξύ της OWL Lite και της OWL Full εξαρτάται από το σημείο μέχρι το οποίο οι χρήστες απαιτούν περισσότερη εκφραστικότητα.

Η επιλογή μεταξύ της OWL DL και της OWL Full βασικά εξαρτάται από το βαθμό στον οποίο οι χρήστες απαιτούν τη δυνατότητα “metamodeling”, όπως για παράδειγμα τον ορισμό κλάσεων για άλλες κλάσεις ή την απόδοση ιδιοτήτων σε κλάσεις. Όταν χρησιμοποιούμε την OWL Full έναντι της OWL DL, δεν μπορούμε να είμαστε σίγουροι όσον αφορά στο reasoning, καθώς δεν είναι δυνατό να έχουμε ολοκληρωμένες υλοποιήσεις σε OWL Full.

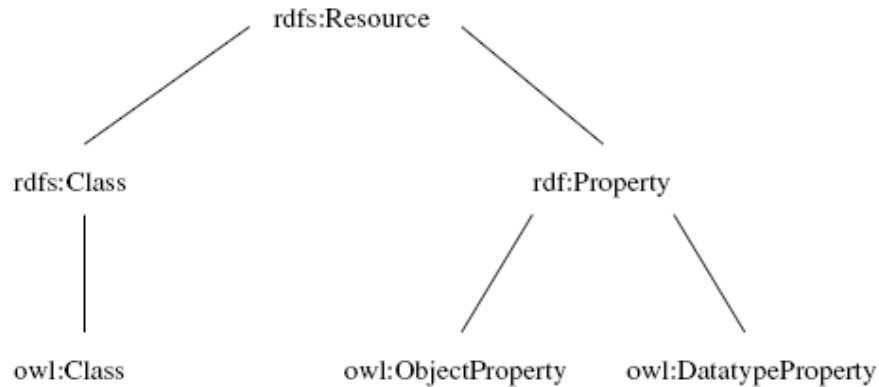
Για τα τρία είδη αυτής της γλώσσας ισχύουν τα εξής:

- Κάθε OWL Lite οντολογία είναι επίσης οντολογία OWL DL
- Κάθε OWL DL οντολογία είναι επίσης οντολογία OWL Full
- Κάθε OWL Lite συμπέρασμα είναι επίσης συμπέρασμα OWL DL
- Κάθε OWL DL συμπέρασμα είναι επίσης συμπέρασμα OWL Full

Η OWL χρησιμοποιεί τα RDF και RDF Schema σε μεγάλο βαθμό:

- Σε όλα τα είδη της OWL χρησιμοποιείται η RDF για τη σύνταξη
- Τα instances δηλώνονται όπως στην RDF, χρησιμοποιώντας RDF descriptions
- Οι constructors της OWL όπως `owl:Class`, and `owl:DatatypeProperty` και `owl:ObjectProperty` είναι ειδικές περιπτώσεις των αντιστοιχών της RDF.

Στο παρακάτω σχήμα παρατίθενται οι σχέσεις κλάσεων-υποκλάσεων των προτύπων OWL και RDF/RDFS.



Ένα από τα βασικά κίνητρα πίσω από τη διαστρωματική αρχιτεκτονική του σημασιολογικού Ιστού είναι η ελπίδα για την προς τα κάτω συμβατότητα με την αντίστοιχη επαναχρησιμοποίηση του λογισμικού στα διάφορα στρώματα. Εντούτοις, το πλεονέκτημα της πλήρους προς τα κάτω συμβατότητας για την OWL (ότι κάθε OWL επεξεργαστής θα παράσχει επίσης τις σωστές ερμηνείες οποιουδήποτε εγγράφου RDFS) επιτυγχάνεται μόνο για την OWL Full, με κόστος την υπολογιστική δυσκολία εντοπισμού.

6.4. Η γλώσσα OWL

6.4.1. Σύνταξη OWL

Η OWL στηρίζεται στα RDF και RDFS και χρησιμοποιεί την σύνταξη RDF με βάση την XML. Δεδομένου ότι αυτή είναι η αρχική σύνταξη για την OWL, την χρησιμοποιούμε, αλλά η RDF/XML δεν παρέχει μια πολύ αναγνώσιμη σύνταξη. Λόγω αυτού, άλλες συντακτικές μορφές για την OWL έχουν καθοριστεί επίσης:

- Μία σύνταξη βασισμένη σε XML που δεν ακολουθεί τις συμβάσεις RDF και έτσι είναι περισσότερο αναγνώσιμη από τους χρήστες.
- Μία πιο αόριστη σύνταξη, που είναι ακόμη πιο αναγνώσιμη από την σύνταξη XML ή την RDF/XML σύνταξη.
- Μία γραφικής μορφής σύνταξη, βασισμένη σε συμβάσεις της UML, η οποία χρησιμοποιείται σε μεγάλο βαθμό και συνεπώς με αυτό τον τρόπο είναι εύκολο να την οικειοποιηθεί ο χρήστης την OWL.

6.4.2. Header

Τα έγγραφα OWL συνήθως καλούνται OWL οντολογίες και είναι έγγραφα RDF. Το στοιχείο root της οντολογίας είναι ένα στοιχείο `rdf:RDF` το οποίο έχει τα εξής namespaces:

```
<rdf:RDF
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#">
```

Μία οντολογία OWL ενδεχομένως να ξεκινάει με μία συλλογή από assertions, οι οποίες είναι συγκεντρωμένες κάτω από ένα στοιχείο `owl:Ontology`, το οποίο περιλαμβάνει σχόλια, έλεγχο της έκδοσης και ίσως και άλλες οντολογίες.

Για παράδειγμα:

```
<owl:Ontology rdf:about="">
  <rdfs:comment>An example OWL ontology</rdfs:comment>
  <owl:priorVersion
    rdf:resource="http://www.mydomain.org/uni-ns-old"/>
  <owl:imports
    rdf:resource="http://www.mydomain.org/persons"/>
  <rdfs:label>University Ontology</rdfs:label>
</owl:Ontology>
```

Μόνο το `owl:imports` έχει συνέπειες στη λογική ερμηνεία της οντολογίας, καθώς αυτό αναφέρεται σε οντολογίες των οποίων το περιεχόμενο θεωρείται τμήμα της συγκεκριμένης οντολογίας. Σημειώνουμε εδώ ότι ενώ τα namespaces χρησιμοποιούνται για την αποσαφήνιση, οι οντολογίες που έχουν γίνει imported παρέχουν ορισμούς που μπορούν να χρησιμοποιηθούν. Συνήθως γίνεται ένα import για κάθε namespace που χρησιμοποιείται, αλλά είναι δυνατό να γίνουν import και άλλες οντολογίες που περιλαμβάνουν ορισμούς χωρίς να εισάγουν νέα ονόματα.

Επίσης αναφέρουμε εδώ ότι το import είναι μία μεταβατική ιδιότητα: αν η οντολογία A κάνει import μία οντολογία B και η B κάνει import μία οντολογία C, τότε η A κάνει import την οντολογία C επίσης.

6.4.3. Class Elements

Οι κλάσεις ορίζονται χρησιμοποιώντας το στοιχείο `owl:Class`. Για παράδειγμα, μπορούμε να ορίσουμε την κλάση `associateProfessor` ως εξής:

```
<owl:Class rdf:ID="associateProfessor">  
  <rdfs:subClassOf rdf:resource="#academicStaffMember"/>  
</owl:Class>
```

Μπορούμε επίσης να πούμε ότι αυτή η κλάση είναι `disjoint` από την `assistantProfessor` και την `professor` χρησιμοποιώντας το στοιχείο `owl:disjointWith`. Αυτά τα στοιχεία μπορούν να συμπεριληφθούν στον προηγούμενο ορισμό, ή μπορούν να προστεθούν με αναφορά στο ID χρησιμοποιώντας το `rdf:about`. Αυτός ο μηχανισμός κληρονομείται από την RDF.

```
<owl:Class rdf:about="#associateProfessor">  
  <owl:disjointWith rdf:resource="#professor"/>  
  <owl:disjointWith rdf:resource="#assistantProfessor"/>  
</owl:Class>
```

Η ισοδυναμία των κλάσεων μπορεί να οριστεί χρησιμοποιώντας το στοιχείο `owl:equivalentClass`:

```
<owl:Class rdf:ID="faculty">  
  <owl:equivalentClass rdf:resource="#academicStaffMember"/>  
</owl:Class>
```

Τέλος, υπάρχουν δύο ήδη προκαθορισμένες κλάσεις, οι `owl:Thing` και `owl:Nothing`. Η πρώτη είναι η πιο γενική κλάση, η οποία περιλαμβάνει τα πάντα, και η τελευταία είναι η κενή κλάση. Έτσι, κάθε κλάση είναι υποκλάση της `owl:Thing` και υπερκλάση της `owl:Nothing`.

6.4.4. Property Elements

Στην OWL υπάρχουν δύο είδη ιδιοτήτων:

- Ιδιότητες αντικειμένου (Object Properties), που συσχετίζουν αντικείμενα σε άλλα αντικείμενα, όπως για παράδειγμα τα `is-TaughtBy` και `supervises`.

- Ιδιότητες τύπου δεδομένων (Data type properties), που συσχετίζουν αντικείμενα σε συγκεκριμένες τιμές δεδομένων, όπως για παράδειγμα τα phone, title και age.

Η OWL δεν έχει προκαθορισμένους τύπους δεδομένων, ούτε παρέχει ιδιαίτερες διευκολύνσεις στους ορισμούς. Αντίθετα, μας επιτρέπει να χρησιμοποιούμε τύπους δεδομένων του XML Schema, κάνοντας έτσι χρήση της διαστρωματικής μορφής του Semantic Web.

Ένα παράδειγμα μίας ιδιότητας datatype είναι:

```
<owl:DatatypeProperty rdf:ID="age">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema
    #nonNegativeInteger"/>
</owl:DatatypeProperty>
```

Τύποι δεδομένων που δημιουργούνται από τον ίδιο το χρήστη συνήθως συλλέγονται σε ένα σχήμα XML και κατόπιν χρησιμοποιούνται σε μία OWL οντολογία. Παράδειγμα μίας ιδιότητας αντικείμενου (object type) είναι:

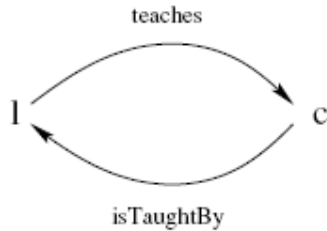
```
<owl:ObjectProperty rdf:ID="isTaughtBy">
  <rdfs:domain rdf:resource="#course"/>
  <rdfs:range rdf:resource="#academicStaffMember"/>
  <rdfs:subPropertyOf rdf:resource="#involves"/>
</owl:ObjectProperty>
```

Περισσότερα του ενός domain και range μπορούν να δηλωθούν, περίπτωση κατά την οποία λαμβάνουμε υπόψη τον συνδυασμό των domains ή ranges.

Η OWL μας επιτρέπει να συσχετίσουμε “αντίστροφες ιδιότητες”. Ένα τυπικό παράδειγμα είναι το ζευγάρι isTaughtBy και teaches:

```
<owl:ObjectProperty rdf:ID="teaches">
  <rdfs:range rdf:resource="#course"/>
  <rdfs:domain rdf:resource="#academicStaffMember"/>
  <owl:inverseOf rdf:resource="#isTaughtBy"/>
</owl:ObjectProperty>
```

Το ακόλουθο σχήμα παρουσιάζει τη σχέση μεταξύ μίας ιδιότητας και της αντίστροφής της. Τα domain και range μπορούν να κληρονομηθούν από την αντίστροφη ιδιότητα (ανταλλαγή του domain με το range).



6.4.5. Property Restrictions

Με το στοιχείο `rdfs:subClassOf` μπορούμε να προσδιορίσουμε μία κλάση C ως υποκλάση μιας άλλης κλάσης C' . Τότε, κάθε instance της C αποτελεί επίσης instance της C' . Ας υποθέσουμε ότι θέλουμε να δηλώσουμε πως η κλάση C ικανοποιεί συγκεκριμένες υποθέσεις, με άλλα λόγια όλα τα instances της C ικανοποιούν αυτές τις υποθέσεις. Αυτό είναι αντίστοιχο με το γεγονός ότι η κλάση C είναι υποκλάση της C' , όπου η C' συγκεντρώνει όλα τα αντικείμενα που ικανοποιούν αυτές τις υποθέσεις. Με τον ίδιο ακριβώς τρόπο γίνεται κάτι τέτοιο στην OWL. Σημειώνουμε εδώ ότι η κλάση C' μπορεί να παραμείνει ανώνυμη.

Το παρακάτω στοιχείο απαιτεί τα μαθήματα του πρώτου έτους να διδάσκονται αποκλειστικά από καθηγητές:

```

<owl:Class rdf:about="#firstYearCourse">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#isTaughtBy"/>
      <owl:allValuesFrom rdf:resource="#Professor"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
  
```

Το στοιχείο `owl:allValuesFrom` χρησιμοποιείται για να προσδιορίσουμε την κλάση με τις δυνατές τιμές που μπορεί να ανατεθούν στην ιδιότητα που ορίζεται από το `owl:onProperty` (δηλαδή όλες οι τιμές αυτής της ιδιότητας πρέπει να προέρχονται από αυτή την κλάση). Στο παράδειγμά μας μόνο οι `professors` επιτρέπεται να αποτελούν τιμές της ιδιότητας `isTaughtBy`.

Μπορούμε να δηλώσουμε ότι το μάθημα των μαθηματικών διδάσκεται από τον Kostas Kalogirou ως εξής:

```

<owl:Class rdf:about="#mathCourse">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#isTaughtBy"/>
      <owl:hasValue rdf:resource="#949352"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

```

Το owl:hasValue δηλώνει μία συγκεκριμένη τιμή που μπορεί να πάρει η ιδιότητα που ορίζεται από το owl:onProperty.

Μπορούμε να δηλώσουμε ότι όλα τα ακαδημαϊκά μέλη πρέπει να διδάσκουν τουλάχιστον ένα μάθημα προπτυχιακό ως εξής:

```

<owl:Class rdf:about="#academicStaffMember">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#teaches"/>
      <owl:someValuesFrom
        rdf:resource="#undergraduateCourse"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

```

Ας συγκρίνουμε το owl:allValuesFrom με το owl:someValuesFrom. Το παράδειγμα που χρησιμοποιεί το πρώτο από αυτά απαιτεί κάθε άτομο που διδάσκει ένα instance της κλάσης μάθημα πρώτου έτους, να είναι καθηγητής.

Το παράδειγμα που χρησιμοποιεί το τελευταίο απαιτεί να υπάρχει κάποιο προπτυχιακό μάθημα το οποίο να διδάσκεται από ένα instance της κλάσης, ένα ακαδημαϊκό μέλος. Είναι πιθανό ο ίδιος ακαδημαϊκός να διδάσκει και μεταπτυχιακά μαθήματα. Γενικά, ένα στοιχείο owl:Restriction περιλαμβάνει μία ή περισσότερες δηλώσεις περιορισμών. Ένας τύπος ορίζει περιορισμούς για το είδος των τιμών που μπορεί να πάρει μία ιδιότητα: owl:allValuesFrom, owl:hasValue και owl:someValuesFrom.

Ένας άλλος τύπος ορίζει περιορισμούς πληθυκότητας. Για παράδειγμα μπορούμε να απαιτήσουμε κάθε μάθημα να διδάσκεται τουλάχιστον από ένα άτομο:

```

<owl:Class rdf:about="#course">
  <rdfs:subClassOf>
    <owl:Restriction>

```



```

    <owl:onProperty rdf:resource="#isTaughtBy"/>
    <owl:minCardinality
      rdf:datatype="&xsd;nonNegativeInteger">
      1
    </owl:minCardinality>
  </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>

```

Παρατηρούμε ότι έπρεπε να ορίσουμε πως το "1" πρέπει να ερμηνευτεί ως non-NegativeInteger (και όχι ως string), και ότι χρησιμοποιήσαμε τη δήλωση xsd namespace στο header στοιχείο για να αναφερθούμε στο έγγραφο XML Schema. Επίσης θα μπορούσαμε να ορίσουμε ότι, για πρακτικούς λόγους, ένα τμήμα πρέπει να έχει τουλάχιστον δέκα ή το πολύ τριάντα μέλη:

```

<owl:Class rdf:about="#department">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasMember"/>
      <owl:minCardinality
        rdf:datatype="&xsd;nonNegativeInteger">
        10
      </owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasMember"/>
      <owl:maxCardinality
        rdf:datatype="&xsd;nonNegativeInteger">
        30
      </owl:maxCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

```

Είναι δυνατό να ορίσουμε έναν ακριβή αριθμό, για παράδειγμα ένας υποψήφιος διδάκτορας πρέπει να έχει ακριβώς δύο επιβλέποντες. Αυτό μπορεί να επιτευχθεί χρησιμοποιώντας τον ίδιο αριθμό στα owl:minCardinality και owl:maxCardinality. Για διευκόλυνση η OWL προσφέρει επίσης το owl:cardinality.

Σημειώνουμε τέλος ότι το owl:Restriction προσδιορίζει ανώνυμες κλάσεις που δεν έχουν ID, δεν μπορούν να οριστούν από την owl:Class και έχουν μόνο τοπικό χαρακτήρα: μπορούν να χρησιμοποιηθούν μόνο εκεί που εμφανίζεται ο περιορισμός.

Όταν μιλάμε για κλάσεις, πρέπει να έχουμε υπόψη τα εξής: υπάρχουν οι κλάσεις που ορίζονται από την `owl:Class` με ορισμένο ID, και οι τοπικές ανώνυμες κλάσεις, ως συλλογές αντικειμένων που ικανοποιούν συγκεκριμένους περιορισμούς ή αποτελούν συνδυασμούς άλλων κλάσεων. Οι τελευταίες συχνά καλούνται *class expressions*.

6.4.6. Special Properties

Μερικές ιδιότητες μπορούν να οριστούν απευθείας:

owl:TransitiveProperty ορίζει μια μεταβατική ιδιότητα, όπως “has better grade than”, “is taller than”, ή “is ancestor of”.

owl:SymmetricProperty ορίζει μια συμμετρική ιδιότητα, όπως “has same grade as” ή “is sibling of”.

owl:FunctionalProperty ορίζει μια ιδιότητα που έχει το πολύ έως μία τιμή για κάθε αντικείμενο, όπως “age”, “height”, ή “directSupervisor”.

owl:InverseFunctionalProperty ορίζει μια ιδιότητα για τη οποία δύο διαφορετικά αντικείμενα δεν μπορούν να έχουν την ίδια τιμή, όπως για παράδειγμα, η ιδιότητα “isTheSocialSecurityNumberfor” (ένας αριθμός κοινωνικής ασφάλισης αποδίδεται μόνο σε ένα άτομο).

Ένα παράδειγμα για τα παραπάνω είναι το εξής:

```
<owl:ObjectProperty rdf:ID="hasSameGradeAs">
  <rdf:type rdf:resource="&owl;TransitiveProperty" />
  <rdf:type rdf:resource="&owl;SymmetricProperty" />
  <rdfs:domain rdf:resource="#student" />
  <rdfs:range rdf:resource="#student" />
</owl:ObjectProperty>
```

6.4.7. Boolean Combinations

Είναι δυνατό να κάνουμε Boolean συνδυασμούς κλάσεων (union, intersection, complement). Για παράδειγμα μπορούμε να πούμε ότι τα μαθήματα και τα μέλη του προσωπικού είναι disjoint ως εξής:

```
<owl:Class rdf:about="#course">
  <rdfs:subClassOf>
```

```

<owl:Class>
  <owl:complementOf rdf:resource="#staffMember"/>
</owl:Class>
</rdfs:subClassOf>
</owl:Class>

```

Σύμφωνα με αυτό κάθε μάθημα είναι ένα instance του συμπληρώματος των staff members, δηλαδή κανένα μάθημα δεν είναι staff member. Αυτή η δήλωση θα μπορούσε να εκφραστεί επίσης χρησιμοποιώντας το owl:disjointWith.

Η ένωση των κλάσεων πραγματοποιείται με το **owl:unionOf**:

```

<owl:Class rdf:ID="peopleAtUni">
  <owl:unionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#staffMember"/>
    <owl:Class rdf:about="#student"/>
  </owl:unionOf>
</owl:Class>

```

Εδώ δε λέμε ότι η καινούρια κλάση είναι υποκλάση της ένωσης, αλλά ότι είναι ισοδύναμη με αυτήν. Έχουμε δηλώσει δηλαδή την ισοδυναμία των κλάσεων. Επίσης, δεν προσδιορίσαμε ότι οι δύο κλάσεις πρέπει να είναι disjoint: είναι δυνατό ένα μέλος προσωπικού να είναι επίσης και φοιτητής.

Η διατομή δηλώνεται χρησιμοποιώντας το **owl:intersectionOf**:

```

<owl:Class rdf:ID="facultyInCS">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#faculty"/>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#belongsTo"/>
      <owl:hasValue rdf:resource="#CSDepartment"/>
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>

```

Παρατηρούμε ότι ορίσαμε την τομή δύο κλάσεων, μία από τις οποίες ήταν ανώνυμη: η κλάση των αντικειμένων που ανήκουν στο τμήμα CS, η οποία συνδυάστηκε με την κλάση *faculty*. Το αποτέλεσμα είναι το προσωπικό που εργάζεται στο CS τμήμα.

Οι Boolean συνδυασμοί μπορούν να συγχωνευθούν αυθαίρετα. Για παράδειγμα, στη συνέχεια ορίζουμε το διοικητικό προσωπικό ως τους εργαζόμενους εκείνους που δεν είναι ούτε στο faculty ούτε στο technical support staff:

```
<owl:Class rdf:ID="adminStaff">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#staffMember"/>
    <owl:Class>
      <owl:complementOf>
        <owl:Class>
          <owl:unionOf rdf:parseType="Collection">
            <owl:Class rdf:about="#faculty"/>
            <owl:Class rdf:about="#techSupportStaff"/>
          </owl:unionOf>
        </owl:Class>
      </owl:complementOf>
    </owl:Class>
  </owl:intersectionOf>
</owl:Class>
```

6.4.8. Enumerations

Μία απαρίθμηση είναι ένα στοιχείο **owl:oneOf** που χρησιμοποιείται για να προσδιορίσουμε μία κλάση θέτοντας σε λίστα όλα της τα στοιχεία:

```
<owl:Class rdf:ID="weekdays">
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:about="#Monday"/>
    <owl:Thing rdf:about="#Tuesday"/>
    <owl:Thing rdf:about="#Wednesday"/>
    <owl:Thing rdf:about="#Thursday"/>
    <owl:Thing rdf:about="#Friday"/>
    <owl:Thing rdf:about="#Saturday"/>
    <owl:Thing rdf:about="#Sunday"/>
  </owl:oneOf>
</owl:Class>
```

6.4.9. Instances

Τα instances των κλάσεων δηλώνονται όπως στην RDF:

```
<rdf:Description rdf:ID="949352">
  <rdf:type rdf:resource="#academicStaffMember"/>
</rdf:Description>
```

ή αντίστοιχα:

```
<academicStaffMember rdf:ID="949352"/>
```

Μπορούμε επίσης να παρέχουμε περισσότερες πληροφορίες, όπως:

```
<academicStaffMember rdf:ID="949352">  
  <uni:age rdf:datatype="xsd:integer">39</uni:age>  
</academicStaffMember>
```

Αντίθετα με τις βάσεις δεδομένων, στην OWL δεν ισχύει η μοναδικότητα των ονομάτων. Το γεγονός ότι δύο instances έχουν διαφορετικό όνομα ή ID δεν υπονοεί ότι είναι διαφορετικά μεταξύ τους. Για παράδειγμα, αν δηλώσουμε ότι κάθε μάθημα διδάσκεται από το πολύ ένα άτομο:

```
<owl:ObjectProperty rdf:ID="isTaughtBy">  
  <rdf:type rdf:resource="owl:FunctionalProperty" />  
</owl:ObjectProperty>
```

και στη συνέχεια δηλώσουμε ότι ένα μάθημα διδάσκεται από δύο άτομα

```
<course rdf:ID="CIT1111">  
  <isTaughtBy rdf:resource="#949318"/>  
  <isTaughtBy rdf:resource="#949352"/>  
</course>
```

αυτό δεν οδηγεί τον OWL reasoner στην εμφάνιση ενός error. Άλλωστε, το σύστημα θα μπορούσε να συμπεράνει ότι τα resources "949318" και "949352" είναι προφανώς ίσα. Για να βεβαιωθούμε ότι διαφορετικά άτομα όντως αναγνωρίζονται ως διαφορετικά, πρέπει να το δηλώσουμε εμφανώς:

```
<lecturer rdf:ID="949318">  
  <owl:differentFrom rdf:resource="#949352"/>  
</lecturer>
```

Επειδή όμως τέτοιες δηλώσεις εμφανίζονται συχνά και ενδέχεται να προκύψει πολύ μεγάλος αριθμός, η OWL παρέχει τη δυνατότητα έκφρασης της ανισότητας των διαφόρων στοιχείων σε μια δεδομένη λίστα:

```
<owl:AllDifferent>
  <owl:distinctMembers rdf:parseType="Collection">
    <lecturer rdf:about="#949318"/>
    <lecturer rdf:about="#949352"/>
    <lecturer rdf:about="#949111"/>
  </owl:distinctMembers>
</owl:AllDifferent>
```

Σημειώνουμε ότι το **owl:distinctMembers** μπορεί να χρησιμοποιηθεί μόνο σε συνδυασμό με το **owl:allDifferent**.

6.4.10. Data Types

Παρόλο που το XML Schema παρέχει έναν μηχανισμό για την ανάπτυξη user-defined datatypes (όπως τον τύπο του `adultAge` που περιλαμβάνει τους ακεραίους από το 18, ή τον τύπο που περιλαμβάνει όλα τα strings που ξεκινάνε με έναν αριθμό), τέτοιοι τύποι δεδομένων δεν μπορούν να χρησιμοποιηθούν στην OWL. Οι τύποι που χρησιμοποιούνται περισσότερο είναι `string`, `integer`, `Boolean`, `time` και `date`.

6.4.11. Πληροφορίες για την έκδοση

Έχουμε ήδη συναντήσει τη δήλωση `owl:priorVersion` ως τμήμα του header για να αναφερθούμε σε παλαιότερες εκδόσεις της συγκεκριμένης οντολογίας. Αυτή η πληροφορία μπορεί να φανεί χρήσιμη στους χρήστες. Εκτός από το

- **owl:priorVersion**, η OWL έχει τρεις επιπλέον δηλώσεις για να υποδείξει την έκδοση:

- **owl:versionInfo**, που περιλαμβάνει ένα `string` το οποίο δίνει πληροφορίες σχετικά με την τωρινή έκδοση, όπως για παράδειγμα `RCS/CVS`.

- **owl:backwardCompatibleWith**, που περιλαμβάνει μία αναφορά σε κάποια άλλη οντολογία. Υποδεικνύει ότι η συγκεκριμένη οντολογία είναι η προηγούμενη έκδοση από κάποια άλλη, και είναι συμβατή με αυτήν. Συγκεκριμένα υποδεικνύει ότι όλα τα προσδιοριστικά από την προηγούμενη έκδοση ερμηνεύονται με τον ίδιο τρόπο και

στην καινούρια έκδοση. Αυτό είναι χρήσιμο για τους συγγραφείς που μπορούν να αλλάξουν τα έγγραφά τους και να προσαρμοστούν στη νέα έκδοση (απλά κάνοντας update στις δηλώσεις namespace και owl:imports ώστε να αναφέρονται στο URL της καινούριας έκδοσης).

- **owl:incompatibleWith**, υποδεικνύει ότι η οντολογία στην οποία περιλαμβάνεται το στοιχείο αυτό είναι η τελευταία έκδοση αυτής στην οποία αναφερόμαστε αλλά δεν είναι backwardcompatible με αυτήν..

6.5. Παράδειγμα

African Wildlife Οντολογία

Αυτό το παράδειγμα παρουσιάζει μία οντολογία που περιγράφει την άγρια ζωή της Αφρικής. Στο σχήμα που ακολουθεί τον κώδικα φαίνονται οι βασικές κλάσεις και οι σχέσεις υποκλάσεων, ενώ παρατηρούμε ότι οι πληροφορίες σχετικά με τις υποκλάσεις αποτελούν μέρος της πληροφορίας που περιλαμβάνεται στην οντολογία.

Στην οντολογία περιλαμβάνονται σχόλια κάνοντας χρήση του rdfs:comment.

```
<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#">

  <owl:Ontology rdf:about="xml:base"/>

  <owl:Class rdf:ID="animal">
    <rdfs:comment>Animals form a class.</rdfs:comment>
  </owl:Class>

  <owl:Class rdf:ID="plant">
    <rdfs:comment>
      Plants form a class disjoint from animals.
    </rdfs:comment>
    <owl:disjointWith rdf:resource="#animal"/>
  </owl:Class>

  <owl:Class rdf:ID="tree">
    <rdfs:comment>Trees are a type of plant.</rdfs:comment>
    <rdfs:subClassOf rdf:resource="#plant"/>
  </owl:Class>

  <owl:Class rdf:ID="branch">
    <rdfs:comment>Branches are parts of trees.</rdfs:comment>
```

```

<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#is_part_of"/>
    <owl:allValuesFrom rdf:resource="#tree"/>
  </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="leaf">
<rdfs:comment>Leaves are parts of branches.</rdfs:comment>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#is_part_of"/>
    <owl:allValuesFrom rdf:resource="#branch"/>
  </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="herbivore">
<rdfs:comment>
  Herbivores are exactly those animals that eat only plants
  or parts of plants.
</rdfs:comment>
<owl:intersectionOf rdf:parseType="Collection">
  <owl:Class rdf:about="#animal"/>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#eats"/>
    <owl:allValuesFrom>
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <owl:Class rdf:about="#plant"/>
          <owl:Restriction>
            <owl:onProperty rdf:resource="#is_part_of"/>
            <owl:allValuesFrom rdf:resource="#plant"/>
          </owl:Restriction>
        </owl:unionOf>
      </owl:Class>
    </owl:allValuesFrom>
  </owl:Restriction>
</owl:intersectionOf>
</owl:Class>

<owl:Class rdf:ID="carnivore">
<rdfs:comment>
  Carnivores are exactly those animals that eat animals.
</rdfs:comment>
<owl:intersectionOf rdf:parseType="Collection">
  <owl:Class rdf:about="#animal"/>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#eats"/>

```



```

    <owl:someValuesFrom rdf:resource="#animal"/>
  </owl:Restriction>
</owl:intersectionOf>
</owl:Class>

<owl:Class rdf:ID="giraffe">
  <rdfs:comment>
    Giraffes are herbivores, and they eat only leaves.
  </rdfs:comment>
  <rdfs:subClassOf rdf:resource="#herbivore"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#eats"/>
      <owl:allValuesFrom rdf:resource="#leaf"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="lion">
  <rdfs:comment>
    Lions are animals that eat only herbivores.
  </rdfs:comment>
  <rdfs:subClassOf rdf:resource="#carnivore"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#eats"/>
      <owl:allValuesFrom rdf:resource="#herbivore"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="tasty_plant">
  <rdfs:comment>
    Tasty plants are plants that are eaten
    both by herbivores and carnivores.
  </rdfs:comment>
  <rdfs:subClassOf rdf:resource="#plant"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#eaten_by"/>
      <owl:someValuesFrom>
        <owl:Class rdf:about="#herbivore"/>
      </owl:someValuesFrom>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#eaten_by"/>
      <owl:someValuesFrom>
        <owl:Class rdf:about="#carnivore"/>
      </owl:someValuesFrom>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

```

```

</owl:someValuesFrom>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>

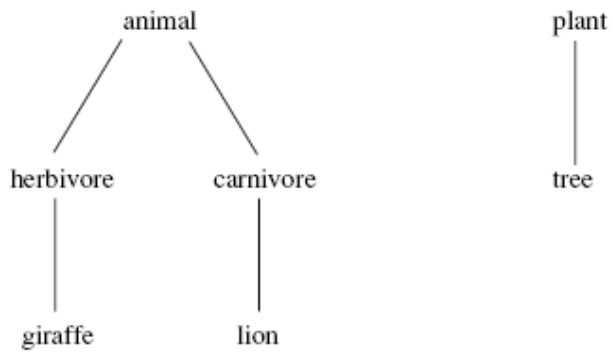
<owl:TransitiveProperty rdf:ID="is_part_of"/>

<owl:ObjectProperty rdf:ID="eats">
  <rdfs:domain rdf:resource="#animal"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="eaten_by">
  <owl:inverseOf rdf:resource="#eats"/>
</owl:ObjectProperty>

</rdf:RDF>

```



Κλάσεις και υποκλάσεις της οντολογίας

7. Agents

7.1. Ευφυείς πράκτορες: τι είναι

Ο όρος ευφυής αναφέρεται στην ευέλικτη και αυτόνομη συμπεριφορά, με την οποία ένας πράκτορας αντιδρά κατάλληλα στο περιβάλλον του και παίρνει πρωτοβουλίες για να πετύχει τους στόχους του, ενώ παράλληλα αναφέρεται και σε κάποιου είδους κοινωνική ικανότητα, καθώς ο πράκτορας μπορεί να αλληλεπιδράσει αποτελεσματικά με άλλους ομολόγους του, και γιατί όχι, και με τους ίδιους τους χρήστες.

Εμείς εδώ αναφερόμαστε σε αυτόνομα, έξυπνα πακέτα λογισμικού [14, 15], τα οποία λειτουργούν σε ένα περιβάλλον εκτός του ελέγχου τους, προσπαθούν να πραγματοποιήσουν τους στόχους τους, οι οποίοι καθορίζονται από τον εκάστοτε χρήστη που ελέγχει και τον αντίστοιχο πράκτορα, και προς αυτή την κατεύθυνση συνεργάζονται και με άλλους agents.

Εκτός από πακέτα λογισμικού, οι πράκτορες μπορεί να θεωρηθούν και ένας τρόπος δημιουργίας πολύπλοκων συστημάτων λογισμικού. Το *close coupling* αναφέρεται σε λογισμικό σχεδιασμένο έτσι ώστε να μπορεί να εργάζεται σε συνεργασία με άλλα modules, καλώντας functions κατανοητές και οικείες προς αυτά. Αντίθετα, ένας agent είναι ένα σύστημα *loosely coupled*. Με τους όρους *close* και *loose coupling* εννοούμε, στη μεν πρώτη περίπτωση, ότι υπάρχει πάντα κάποιο ειδικό σύστημα που μπορεί να αλληλεπιδρά με άλλα συστήματα, κάτω από συγκεκριμένες συνθήκες και σύμφωνα με ορισμένους κανόνες, ενώ στη δεύτερη περίπτωση, καθένας λειτουργεί σχετικά ευέλικτα, χρησιμοποιώντας τον ίδιο εξοπλισμό, αλλά προσαρμόζοντας τις διαδικασίες στις επιμέρους συνθήκες που επικρατούν.

Το Web, στη μορφή που είναι σήμερα, είναι *loosely coupled*, καθώς μία ποικιλία προγραμμάτων, που έχουν σχεδιαστεί ανεξάρτητα το ένα από το άλλο, μπορούν να λειτουργήσουν και να συνεργαστούν. Σε ένα τέτοιο περιβάλλον θα μπορούσε πολύ άνετα να ταιριάζει ένα σύστημα που κάνει χρήση των agents.

Οι agents χρησιμοποιούνται για εμπορικούς λόγους εδώ και καιρό. Κάποιοι ελεγκτές βιομηχανικών διαδικασιών έχουν πολλά από τα χαρακτηριστικά των agents, αλλά είναι άμεσα συσχετισμένοι με τη συγκεκριμένη διαδικασία για την οποία χρησιμοποιούνται και δεν είναι ανοιχτοί σε σχετικά ανεξέλεγκτα περιβάλλοντα και requests άλλων, άγνωστων agents.

Γενικά, οποιοδήποτε σύστημα μπορεί να μοντελοποιηθεί ως μία αλληλουχία διαδικασιών που αλληλεπιδρούν στο ίδιο περιβάλλον, μπορεί να προσεγγιστεί μέσω ενός agent, εφόσον οι διαδικασίες είναι δυνατό να προσδιοριστούν επαρκώς.

7.2. Βασικοί τύποι δεδομένων

Η τακτική που χρησιμοποιήθηκε στην τεχνητή νοημοσύνη, σύμφωνα με την οποία ένας λογικός reasoner λαμβάνει δεδομένα και με βάση κάποιους κανόνες προσπαθεί να καταλήξει στην κατάλληλη συμπεριφορά, δεν είχε ικανοποιητικά αποτελέσματα. Έτσι, καταλήξαμε στη χρήση των agents, οι οποίοι διακρίνονται στους εξής:

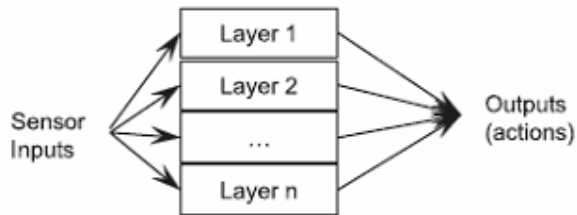
- Logic-based agents
- Reactive agents
- Belief-desire-intention agents
- Layered architectures

Από τις κατηγορίες αυτές θα μας χρησιμεύσει η πρώτη, όσον αφορά στο Semantic Web, καθώς οι συγκεκριμένοι agents έχουν ως βάση τους τη λογική, την οποία χρησιμοποιούν και οι οντολογίες.

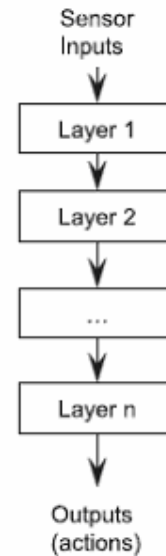
7.3. Αρχιτεκτονικές με στρώματα

Αν και οι αρχιτεκτονικές αυτές έχουν θεωρηθεί ως μια κατηγορία πρακτόρων, ως ένα ορισμένο βαθμό μια προσέγγιση που διακρίνεται σε στρώματα θα μπορούσε να εφαρμοστεί για οποιοδήποτε είδος πράκτορα. Υπάρχουν δύο προσεγγίσεις στη διάταξη σε στρώματα, αποκαλούμενες αντίστοιχα οριζόντια και κάθετη. Με την οριζόντια διάταξη σε στρώματα, κάθε στρώμα επεξεργάζεται ένα είδος input και

λαμβάνουμε το output. Τα αποτελέσματα κάθε στρώματος πρέπει να ισορροπηθούν για να παράγουν ένα συνεπές ολοκληρωμένο αποτέλεσμα.



Οριζόντια διάταξη



Κατακόρυφη διάταξη

Στην περίπτωση της κατακόρυφης διάταξης, τα διαφορετικά είδη επεξεργασίας πραγματοποιούνται στα διαφορετικά στρώματα. Αν και όλα τα στρώματα μπορεί να λαμβάνουν input, η επεξεργασία πραγματοποιείται από το ένα στρώμα στο άλλο.

7.4. Αλληλεπιδράσεις πρακτόρων

Ο πιθανός αριθμός αλληλεπιδράσεων μεταξύ των πρακτόρων είναι τεράστιος. Οι περισσότεροι πράκτορες θα πρέπει να επικοινωνήσουν για να συντονιστούν οι δραστηριότητές τους, οι από κοινού ενέργειες, να διαπραγματευτούν, να ζητήσουν ενέργειες άλλων πρακτόρων, και να αλλάξουν ίσως τα σχέδιά τους σύμφωνα με τις ενέργειες άλλων πρακτόρων. Τα συστήματα πρακτόρων που λειτουργούν στο σημασιολογικό Ιστό θα πρέπει επίσης να είναι σε θέση να αποκτήσουν την εμπιστοσύνη άλλων πρακτόρων, να συνεργαστούν με τους πράκτορες που δεν έχουν σχεδιαστεί ως τμήμα του ίδιου προγράμματος, και να λειτουργήσουν στο κατά ένα μεγάλο μέρος ανεξέλεγκτο περιβάλλον του Ιστού. Οι αλληλεπιδράσεις πρακτόρων θα πραγματοποιηθούν σε διάφορα επίπεδα:

- *Κοινή υποδομή*

Στο πιο βασικό επίπεδο, οι πράκτορες χρειάζονται μια κοινή υποδομή για να είναι σε θέση να επικοινωνήσουν. Οι πράκτορες που υπάρχουν στη μνήμη ενός ενιαίου υπολογιστή μπορούν να επικοινωνήσουν με τη χρησιμοποίηση μιας μορφής κοινής μνήμης ή άλλης τυποποιημένης interprocess μεθόδου επικοινωνίας αλλά για το σημασιολογικό Ιστό, οι πράκτορες θα χρησιμοποιήσουν πιθανώς το Διαδίκτυο (ή τα ιδιωτικά intranets, σε μερικές περιπτώσεις).

- *Μηνύματα*

Ένα επίπεδο υψηλότερα, ένας πράκτορας πρέπει να είναι σε θέση να αποκρυπτογραφήσει τα μηνύματα που λαμβάνει. Αυτό απαιτεί τις δομές πρότυπων μηνυμάτων και τους τύπους μηνυμάτων, μαζί με μια τυποποιημένη γλώσσα για να εκφράσει τα μηνύματα.

- *Οντολογία*

Ανεβαίνοντας ένα επίπεδο, ένας πράκτορας πρέπει να καταλάβει το λεξιλόγιο που χρησιμοποιείται στα μηνύματα που λαμβάνει. Στο σημείο αυτό αναφερόμαστε στην έννοια της οντολογίας. Εάν ένας πράκτορας δεν ξέρει την οντολογία που χρησιμοποιείται σε ένα μήνυμα, είτε θα πρέπει να την βρει και να εισαγάγει το νέο λεξιλόγιο σε αυτά που ξέρει ήδη, ή θα πρέπει να απαντήσει και να πει ότι δεν είναι σε θέση να επεξεργαστεί το μήνυμα.

- *Λογική*

Οι πράκτορες πρέπει επίσης να είναι σε θέση να εξηγήσουν το συλλογισμό τους σε άλλους πράκτορες. Επιπλέον, ίσως χρειαστεί να πραγματοποιηθεί ανταλλαγή δεδομένων μεταξύ πρακτόρων, οπότε θα είναι απαραίτητη μία γλώσσα επικοινωνίας.

- *Έγκριση και εμπιστοσύνη*

Οι πράκτορες θα πρέπει να είναι σε θέση να ανταλλάξουν τα πιστοποιητικά με τα οποία αναγνωρίζεται η προέλευση, η εμπιστοσύνη και η δικαιοδοσία τους να λάβουν ορισμένα μέτρα (όπως τη χρέωση μιας πιστωτικής κάρτας, παραδείγματος χάριν). Ένας πράκτορας θα πρέπει να είναι σε θέση να βεβαιώσει και να αποδείξει ότι έχει το δικαίωμα να ενεργήσει εκ μέρους του ιδιοκτήτη του για να δεσμεύσει πόρους (όπως η πληρωμή για τις υπηρεσίες).

7.5. Η εξέλιξη των πρακτόρων

Ο απλούστερος πιθανός πράκτορας θα αναλάμβανε έναν συγκεκριμένο στόχο: την συγκέντρωση πληροφοριών. Ένας πράκτορας αναζητά ιστοχώρους για να εντοπίσει τα σχετικά με το ζητούμενο sites, τα οποία στη συνέχεια εξετάζει επίσης. Μερικά πακέτα λογισμικού ανακτούν τον καιρό ή άλλες πληροφορίες από ποικίλες πηγές. Αυτό το είδος λογισμικού λαμβάνεται υπόψη με δυσκολία ως πράκτορας, δεδομένου ότι δεν συνεργάζεται με τα διαφορετικά είδη πρακτόρων και δεν εκθέτει την εύκαμπτη, ευφυή συμπεριφορά.

Τα πλέον απλά συστήματα πρακτόρων θα χρησιμοποιηθούν από παλιούς χρήστες ή εταιρίες - συστήματα που έχουν περιορίσει τις δυνατότητες και για τα οποία οι ερωτήσεις εμπιστοσύνης και ασφάλειας δεν θα έχουν βαρύτητα. Δεδομένου ότι τα πρότυπα εξελίσσονται, όλο και περισσότερες ικανότητες θα δοκιμαστούν. Τα λεξιλόγια θα αναπτυχθούν και θα επινοηθούν τρόποι για να εντοπίζονται αυτά όταν απαιτούνται. Δύο ή περισσότερες ομάδες θα μπορούν να συνεργαστούν για να κάνουν την εργασία πρακτόρων τους από κοινού. Σιγά-σιγά, περισσότερες δυνατότητες θα μπουν σε κοινή χρήση. Οι πράκτορες θα γίνουν εύκαμπτοι και αρκετά σύνθετοι έτσι ώστε νέες, χρήσιμες συμπεριφορές ενδεχομένως να προκύψουν.

7.6. Πλαίσια και πρότυπα

Το ίδρυμα για τους ευφυείς φυσικούς πράκτορες (FIPA) είναι μια κοινοπραξία μεταξύ επιχειρήσεων παγκοσμίως. Στόχος του είναι να παράγει "τα πρότυπα για τη λειτουργικότητα των ετερογενών πρακτόρων λογισμικού." Υποστηρίζει την εργασία για τις οντολογίες, τη σημασιολογία, τις υπηρεσίες, το σχεδιασμό, και την ασφάλεια, για να ονομάσει τους πιο προεξέχοντες στόχους.

Το FIPA έχει δημιουργήσει ένα αρκετά περιεκτικό πλαίσιο προδιαγραφών, που προσπαθεί να καλύψει ολόκληρη τη σειρά που απαιτείται για να διευκρινίσει την εντολή, τον έλεγχο, και τις αλληλεπιδράσεις πρακτόρων. Έχει προσπαθήσει σκληρά για να σιγουρευτεί ότι κάθε προδιαγραφή στρέφεται προσεκτικά προς μια περιορισμένη περιοχή. Οι προδιαγραφές FIPA είναι, ως επί το πλείστον, αρκετά αφηρημένες. Παραδειγματος χάριν, η προδιαγραφή δομών μηνυμάτων ACL καθορίζει τη δομή και το περιεχόμενο των μηνυμάτων, αλλά ανεξαρτήτως της

μορφής ανταλλαγής. Τα πραγματικά μηνύματα πρακτόρων θα μπορούσαν να κατασκευαστούν χρησιμοποιώντας XML, RDF, ή ένα ευρύ φάσμα άλλων προσεγγίσεων. Υπάρχει επίσης μια αντιπροσώπευση μηνυμάτων ACL στην προδιαγραφή XML που διευκρινίζει πώς να εκφράσει τα μηνύματα FIPA χρησιμοποιώντας XML. Αυτή η αφηρημένη προσέγγιση είναι ισχυρή και χωρίζει τα ουσιαστικά χαρακτηριστικά σχεδιασμού από τις εκτιμήσεις που απαιτούνται για την πρακτική εφαρμογή.

Ακολουθεί ένας μικρός κατάλογος τίτλων προδιαγραφών FIPA:

Abstract Architecture Specification
ACL Message Structure Specification
ACL Message Representation in XML Specification
Agent Software Integration Specification
Communicative Act Library Specification
Content Language Library Specification
CCL (Constraint Choice) Content Language Specification
KIF Content Language Specification
RDF Content Language Specification
SL (Syntax and Lexical) Content Language Specification
Nomadic Application Support Specification
Query Interaction Protocol Specification
Request Interaction Protocol Specification
Request When Interaction Protocol Specification
Device Ontology Specification
Ontology Service Specification
Personal Assistant Specification
Personal Travel Assistance Specification

Είναι προφανές ότι το FIPA έχει υιοθετήσει μια λεπτομερή μέθοδο στην κάλυψη του επιλεγμένου τομέα. Η χρήση RDF προσαρμόζεται (και επίσης η XML, αν και δεν συμπεριλαμβάνεται στον κατάλόγο μας). Επίσης παρατηρούμε τις προδιαγραφές σχετικά με τις οντολογίες: Η προδιαγραφή υπηρεσιών οντολογίας (Ontology Service Specification) παρέχει ένα πλαίσιο για τους πράκτορες για να μοιραστούν τις οντολογίες, οι οποίες θα είναι εξαιρετικά σημαντικές όταν πρέπει να λειτουργήσει από κοινού μια σειρά των χωριστά σχεδιασμένων πρακτόρων. Η Agent Software

Integration είναι μια άλλη προνοητική προδιαγραφή που προσπαθεί να παράσχει έναν τρόπο ώστε να ενσωματωθεί το λογισμικό μη- fipa, έτσι ώστε να μπορεί να συμμετέχει σε μια κοινωνία πρακτόρων συμβατή με το fipa. Η πρώτη προδιαγραφή στον κατάλογο διευκρινίζει μια αφηρημένη αρχιτεκτονική. Αυτό είναι μια προσπάθεια να σιγουρευτεί ότι ένα σύστημα πρακτόρων σχεδιάζεται με ορισμένα είδη χαρακτηριστικών γνωρισμάτων. Παραδειγματος χάρη, η αφηρημένη αρχιτεκτονική λέει αυτό, εν μέρει, για τους πράκτορες (www.fipa.org/specs/fipa00001/SC00001L.html#_Toc26668620):

An agent is a computational process that implements the autonomous, communicating functionality of an application. Typically, agents communicate using an Agent Communication Language. A concrete instantiation of agent is a mandatory element of every concrete instantiation of the FIPA Abstract Architecture.

Η λίστα συνεχίζει με τις σχέσεις μεταξύ ενός πράκτορα και άλλων στοιχείων της αφηρημένης αρχιτεκτονικής. Η αφηρημένη αρχιτεκτονική προστέθηκε σε FIPA λίγο αργότερα αφότου παράχθηκε η πρώτη έκδοση του πλαισίου. Κατά συνέπεια, μερικές εφαρμογές το ακολουθούν και μερικές όχι.

8. Επίλογος-Συμπεράσματα

8.1. Συμπεράσματα

Έως τώρα περιγράψαμε τις βασικές τεχνολογίες του Σημασιολογικού Ιστού. Τώρα λοιπόν καλούμαστε να συνδυάσουμε όλες αυτές τις τεχνολογίες μέσα από ένα υποθετικό σενάριο:

- ♦ Κάθε διαπραγματεμένο συμβαλλόμενο μέρος αντιπροσωπεύεται από έναν πράκτορα λογισμικού.
- ♦ Οι πράκτορες πρέπει να συμφωνήσουν σχετικά με την έννοια ορισμένων όρων, με τη χρήση μιας κοινής οντολογίας, η οποία είναι για παράδειγμα γραμμένη σε OWL.
- ♦ Τα γεγονότα, οι προσφορές, και οι αποφάσεις μπορούν να αντιπροσωπευθούν χρησιμοποιώντας τις δηλώσεις RDF, οι οποίες γίνονται πραγματικά χρήσιμες όταν συνδυάζονται με μια οντολογία.
- ♦ Οι πληροφορίες ανταλλάσσονται μεταξύ των πρακτόρων σε κάποια γλώσσα που βασίζεται σε XML.
- ♦ Οι στρατηγικές διαπραγμάτευσης των πρακτόρων περιγράφονται σε μια λογική γλώσσα.
- ♦ Ένας πράκτορας αποφασίζει για το επόμενο βήμα, λαμβάνοντας υπόψη τα συμπεράσματα που προέκυψαν μετά από τη διαπραγμάτευση, τα γεγονότα, τις προσφορές και τις αντιπροσφορές.

8.2. Το μέλλον

Θα υλοποιηθεί λοιπόν το όραμα του Σημασιολογικού Ιστού; Πολλοί πιστεύουν σε αυτό και μάλιστα επενδύουν σε αυτό. Μια μεγάλη ποικιλία εργαλείων είναι ήδη διαθέσιμη για κάθε πτυχή του κύκλου ζωής του Semantic Web (editors, εργαλεία αποθήκευσης, απεικόνισης, θέσεως queries και εξαγωγής συμπερασμάτων, versioning). Αυτά τα εργαλεία βέβαια είναι προς το παρόν σε εφαρμογή σε

ακαδημαϊκό επίπεδο, αλλά λαμβάνονται γρήγορα από τον εμπορικό τομέα, ειδικότερα, από ιδιαίτερα καινοτόμες εφαρμογές, και στις Ηνωμένες Πολιτείες και στην Ευρωπαϊκή Ένωση.

Το σημαντικότερο πρόβλημα, ωστόσο, αφορά στο χειρισμό ενός πλήθους διαφορετικών οντολογιών που αναμένεται να προκύψουν. Πολλές προσεγγίσεις ερευνώνται και ενδεχομένως τα πρώτα επιτυχή cases να προκύψουν μέσα στα intranets των μεγάλων οργανώσεων. Σε τέτοια περιβάλλοντα, ο κεντρικός έλεγχος μπορεί να επιβάλει τη χρήση των προτύπων και τις τεχνολογίες, και ενδεχομένως οι πρώτες πραγματικές επιτυχημένες περιπτώσεις θα προκύψουν.

Κατά συνέπεια πιστεύουμε ότι η διαχείριση γνώσης για τις μεγάλες επιχειρήσεις μπορεί να είναι ο πιο ελπιδοφόρος τομέας. Άλλοι τομείς που θα ακολουθήσουν σύντομα είναι η αποκαλούμενη e-science: η χρήση του Semantic Web από τους επιστήμονες (όπως ακριβώς η χρήση από τους επιστήμονες ήταν καταλυτικής σημασίας για το World Wide Web). Το ηλεκτρονικό εμπόριο, με όλα τα σχετικά προβλήματα όσον αφορά στην ασφάλεια και την εμπιστοσύνη είναι μία μόνο εφαρμογή του σημασιολογικού Ιστού.

8.3. Σύνοψη

Ολοκληρώνοντας, στα πλαίσια της συγκεκριμένης εργασίας πραγματοποιήθηκε η υλοποίηση ενός συστήματος εκμάθησης της έννοιας του Σημασιολογικού Ιστού και όσων αυτός περιλαμβάνει, τόσο σε επίπεδο περιγραφής της λειτουργίας του όσο και σε επίπεδο περιγραφής της αρχιτεκτονικής και των τεχνολογιών που χρησιμοποιεί.

Συγκεκριμένα, παρουσιάστηκαν αναλυτικά οι ανάγκες μέσα από τις οποίες έχει προκύψει το όραμα του Semantic Web, το οποίο θα μας επιτρέψει την επίτευξη της διασυνδεσιμότητας σε *τεχνικό* (ύπαρξη δικτύωσης, κοινών πρωτοκόλλων επικοινωνίας κτλ) και *αφαιρετικό επίπεδο* (κοινή αποδοχή τεχνολογιών), την ολοκλήρωση και διαμόρφωση δεδομένων βάσει κοινών μοντέλων πληροφορίας (πρότυπα), τη διαχείριση του περιεχομένου και των δεδομένων μέσω αποτελεσματικών αρχιτεκτονικών και την ευκολία στην πρόσβαση των πληροφοριών για την ανακάλυψη, ανάκτηση, επεξεργασία και παρουσίαση του ψηφιακού αποθέματος. Όλες αυτές οι συνιστώσες αποτελούν με έναν όρο τη *σημασιολογική* διαλειτουργικότητα.

Παρουσιάστηκαν οι γλώσσες εννοιολογικής αναπαράστασης που επιτρέπουν τη διατύπωση αυξημένης σημασιολογίας και δομής, παραθέτοντας τα σχετικά παραδείγματα μέσα από τα οποία γίνεται ακόμη πιο κατανοητό το περιεχόμενο και οι κανόνες της θεωρίας που έχο παρουσιαστεί. Πιο συγκεκριμένα, έγινε εκτενής περιγραφή της RDF, που απαρτίζει τη βάση για την κωδικοποίηση, ανταλλαγή, επεξεργασία και επαναχρησιμοποίηση μεταδεδομένων. Η βασική επιδίωξή της είναι να επιτρέψει τον ορισμό της σημασιολογίας που εγκλείεται σε πληροφοριακούς πόρους με τυπικό, διαλειτουργικό και ανθρωπίνως αναγνώσιμο τρόπο μέσω ενός μηχανισμού για την περιγραφή πληροφοριακών πόρων, ο οποίος δεν κάνει καμία υπόθεση για τη φύση του συγκεκριμένου πεδίου εφαρμογής ή τη δομή του εγγράφου που περιέχει την πληροφορία. Εκτός από τη διαλειτουργικότητα μεταξύ συστημάτων, η RDF στοχεύει στην επαναχρησιμοποίηση, διαμοιρασμό και επεκτασιμότητα των ορισμών μεταδεδομένων και κατά συνέπεια, στην αυτόματη επεξεργασία πόρων που ανταλλάσσονται μέσω του Διαδικτύου.

Επιπλέον, περιγράφηκε ο OWL, η οποία είναι σχεδιασμένη για χρήση από εφαρμογές που πρέπει να επεξεργαστούν το περιεχόμενο της πληροφορίας, αντί απλώς να το προβάλλουν στο χρήστη. Η OWL υποστηρίζει τη δυνατότητα διατύπωσης πλουσιότερης σημασιολογίας από τις γλώσσες XML, RDF, και RDF Schema, παρέχοντας επιπλέον λεξιλόγιο και τυπική σημασιολογία για την περιγραφή των ιδιοτήτων και των κλάσεων. Μεταξύ άλλων είναι σε θέση να εκφράσει σχέσεις μεταξύ κλάσεων (π.χ μη αλληλοεπικάλυψη), πληθυκότητα (π.χ «ακριβώς ένα»), ισοδυναμία, πλουσιότερους τύπους ιδιοτήτων, χαρακτηριστικά ιδιοτήτων (π.χ συμμετρία) και απαριθμητές κλάσεις.

Τέλος, τονίστηκε η σημασία των πρακτόρων μέσα στο περιβάλλον του Σημασιολογικού Ιστού, οι οποίοι, χρησιμοποιώντας διάφορες τεχνικές, μπορούν να «μάθουν» τα ενδιαφέροντα και τις συνήθειες κάθε χρήστη ξεχωριστά και έτσι να του προσφέρουν ουσιαστική βοήθεια. Είναι ακόμα πιθανό να χρειαστεί να προσαρμόσουν τη λειτουργία τους και τον τρόπο παροχής υπηρεσιών ανάλογα με το πώς οι ανάγκες του χρήστη αλλάζουν στο πέρασμα του χρόνου. Έτσι, όλες οι διαδικασίες αυτοματοποιούνται προς όφελος του χρήστη.

Συμπερασματικά, η παρούσα διπλωματική εισήγαγε τον εκπαιδευόμενο στις βασικές αρχές οργάνωσης, σχεδιασμού και υλοποίησης του Semantic Web, παρουσιάζοντας συγκεκριμένα εργαλεία των οποίων μπορούν να κάνουν χρήση προς την ανάπτυξη

ολοκληρωμένων εφαρμογών που θα ακολουθούν τις προδιαγραφές του Σημαιολογικού Ιστού.

8.4. Μελλοντικές επεκτάσεις

Η παρούσα διπλωματική εργασία αποτελεί μία καλή βάση για περαιτέρω ανάπτυξη πάνω σε θέματα σχετικά με την εκμάθηση του Σημαιολογικού Ιστού. Πιο συγκεκριμένα, μια επέκταση και αναβάθμισή της μπορεί να πραγματοποιηθεί μέσω:

- της ενσωμάτωσης των ήδη αναφερθέντων αλλά και άλλων εργαλείων ανάπτυξης οντολογιών και πρακτικής εφαρμογής αυτών μέσα στο ίδιο το περιβάλλον του Tutorial
- παροχής της δυνατότητας interactivity, μέσω ασκήσεων και εκπαιδευτικών δραστηριοτήτων που θα παρέχουν κάποιο feedback στον εκπαιδευόμενο και θα προσαρμόζονται στις ανάγκες και το επίπεδο γνώσεων που διαθέτει (κάνοντας χρήση server)
- προσθήκη επιλογής στο μενού για εισαγωγή σε βάση δεδομένων έτοιμων αρχείων RDF ή OWL και επεξεργασία τους

9. Βιβλιογραφία

- [1] Dr Davies, J., Pr Fensel, D., Pr Harmelen, F., (2003). "Towards the Semantic Web. Ontology-driven Knowledge Management".
- [2] Decker, S., Harmelen, F., Broekstra, J., et al. "The Semantic Web - on the respective Roles of XML and RDF".
- [3] Wilson, R., (2004). "The role of ontologies in Teaching and Learning".
- [4] Natalya F. Noy, Deborah L. McGuinness. "Ontology Development 101: "A guide to Creating your First Ontology".
- [5] Piotr Kaminski B. Math (1997). "Integrating Information on the Semantic Web Using Partially Ordered Multi Hypersets".
- [6] "Case Methods for Online Learning", available at <http://elearnmag.org/subpage.cfm?section=tutorials&article=11-1>
- [7] Antoniou, G., Harmelen, F., (2004). "A Semantic Web Primer".
- [8] Allen, J., (2001). "Making a Semantic Web".
- [9] Daconta, M., Obrst, L., Smith, K., (2003). "The Semantic Web: A Guide to the Future of XML, Web Services and Knowledge Management".
- [10] Rosario, B., Feldman, R., "Visualizing Large Medical Text Collections Using Semantic Metadata".
- [11] Malet,G., Munoz, F., Appleyard, R., et al. (1998). "A Model for Enhancing Internet Medical Document Retrieval with "Medical Core Metadata"
- [12] <http://www.cyc.com/cyc>
- [13] [http:// www.w3.org/2004/OWL/](http://www.w3.org/2004/OWL/)
- [14] "Agents and the Semantic Web" available at http://oopsla.snu.ac.kr/~jnkim/pdf/IS_2.pdf
- [15] Dave Beckett (2004). RDF/XML Syntax Specification (Revised). W3C Recommendation 10th February 2004. Available at: <http://www.w3.org/TR/rdf-syntax-grammar/>
- [16] T. Berners-Lee (1998). Semantic Web Road Map. W3C. Available at: <http://www.w3.org/DesignIssues/Semantic.html>
- [17] Broekstra, J., Klein, M., Decker, S. et al (2000). "Adding formal semantics to the web: building on top of rdf schema. In ECDL Workshop on the Semantic Web: Models, Architectures and Management."