

Α.Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΙΩΑΝΝΗΣ Δ. ΒΑΛΑΤΣΟΣ
ΕΥΑΓΓΕΛΟΣ Π. ΒΑΡΝΑΛΗΣ

ΚΑΘΗΓΗΤΗΣ: ΜΙΧΑΛΗΣ ΣΑΛΑΜΠΙΑΣΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ ΓΙΑ ΤΗΝ ΔΗΜΙΟΥΡΓΙΑ
ΕΠΙΣΗΜΕΙΩΣΕΩΝ (ANNOTATION TOOL)**



ΘΕΣΣΑΛΟΝΙΚΗ 2009

Περιεχόμενα

<u>ΕΙΣΑΓΩΓΗ</u>	5
1 <u>ΣΗΜΑΣΙΟΛΟΓΙΚΟΣ ΙΣΤΟΣ (SEMANTIC WEB)</u>	7
1.1 Στόχοι της Πτυχιακής	7
1.2 Το Διαδίκτυο σήμερα	8
1.3 Τεχνολογίες Semantic Web	10
1.3.1 Μεταδεδομένα (Metadata)	10
1.3.2 Οντολογίες	13
1.3.3 Λογική	16
1.3.4 Πράκτορες	17
1.3.5 Semantic Web vs. Τεχνητή Νοημοσύνη	18
1.4 Προσέγγιση με βάση τα επίπεδα	20
2 <u>ΑΡΧΕΙΑ ΠΑΓΚΟΣΜΙΟΥ ΙΣΤΟΥ (WEB FILES) : XML</u>	24
2.1 Εισαγωγή	24
2.2 Η γλώσσα XML	27
2.2.1 Πρόλογος (Prolog)	27
2.2.2 Στοιχεία(elements)	28
2.2.3 Ιδιότητες(attributes)	28
2.2.4 Σχόλια	29
2.2.5 Οδηγίες Επεξεργασίας(processing instructions)	30
2.2.6 Η κατάλληλη δομή των XML εγγράφων	30
2.2.7 Το δένδρο των XML εγγράφων	30
2.3 Δομή (structure)	32
2.3.1 DTDs	33
2.3.2 XML schema	39
3 <u>ΠΕΡΙΓΡΑΦΩΝΤΑΣ ΤΗΣ ΠΗΓΗΣ ΤΟΥ Web: RDF</u>	47

3.1	Εισαγωγή	47
3.2	RDF : Βασικές Ιδέες	50
3.2.1	Πηγές	50
3.2.2	Ιδιότητες	51
3.2.3	Προτάσεις	51
3.2.4	Τρεις απόψεις για τις Προτάσεις (statements)	51
3.2.5	Η πρόσβαση του προγραμματιστή(Reification)	55
3.2.6	Τύποι Δεδομένων	56
3.2.7	Μια κρίσιμη άποψη για το RDF	57
3.3	RDF : σύνταξη βασισμένη στην XML	59
3.3.1	Η ιδιότητα rdf:resource	63
3.3.2	Φωλιασμένες Περιγραφές	64
3.3.3	Το στοιχείο rdf: type	65
3.3.4	Ο μηχανισμός Reification	66
3.4	RDF Schema : Βασικές Ιδέες	67
3.4.1	Κλάσεις και Ιδιότητες	67
3.4.2	Ιεραρχίες κλάσεων και κληρονομικότητα	68
4	<u>ONTOΛΟΓΙΕΣ OWL (WEB ONTOLOGY LANGUAGE)</u>	71
4.1	Εισαγωγή	71
4.2	OWL και RDF/RDFs	71
4.3	Συμβατότητα OWL και RDF/RDFs	74
4.4	Περιγραφή της γλώσσας OWL	77
4.4.1	Σύνταξη	77
4.4.2	Επικεφαλίδα(header)	78
4.4.3	Στοιχεία Κλάσεων(class elements)	79
4.4.4	Ιδιότητες στοιχείων(properties elements)	80
4.4.5	Περιορισμοί Ιδιοτήτων	81
4.4.6	Ειδικές Ιδιότητες (special properties)	85
4.4.7	Συνδυασμοί (Boolean Combinations)	86
4.4.8	Απαριθμήσεις (Enumerations)	88
4.4.9	Αναφορές (Instances)	88
5	<u>ΕΦΑΡΜΟΓΕΣ ΣΗΜΑΣΙΟΓΙΚΟΥ ΙΣΤΟΥ (SEMANTIC WEB)</u>	91
5.1	Ηλεκτρονική Μάθηση (E – Learning)	91
5.2	Υπηρεσίες Παγκόσμιου Ιστού (WEB Services)	95

6	<u>WEB ANNOTATION</u>	99
6.1	Η διαδικασία των επισημειώσεων στον Παγκόσμιο Ιστό	99
6.2	Annotation Tools – Παραδείγματα	99
6.3	Διαδικτυακή υλοποίηση της εφαρμογής Annotation Tool (Web Notes Tool)	104
7	<u>ΕΦΑΡΜΟΓΕΣ ΣΗΜΑΣΙΟΓΙΚΟΥ ΙΣΤΟΥ (SEMANTIC WEB)</u>	120
7.1	Γενικά	120
7.2	Πιθανές επεκτάσεις	122
	<u>BIBΛΙΟΓΡΑΦΙΑ</u>	123

Εισαγωγή

Η ανάπτυξη του παγκόσμιου ιστού ή World Wide Web έγινε με στόχο να δημιουργηθεί ένα πλέγμα πληροφοριών για τους ανθρώπους. Πράγματι οι σελίδες του Web απευθύνονται αποκλειστικά και μόνο στους ανθρώπους, με την έννοια ότι η πληροφορία είναι αναγνώσιμη και κατανοητή μόνο από αυτούς.

Το W3C ο οργανισμός που εξελίσσει τα πρότυπα του Web στο πλαίσιο μιας συγκεκριμένης δράσης, αναπτύσσει τη λογική του Σημασιολογικού ιστού ή Semantic Web. Ο Σημασιολογικός Ιστός θα επιτρέψει στην πληροφορία να είναι κατανοητή και από τους υπολογιστές, καθιστώντας τους ικανούς να αυτοματοποιήσουν σε μεγάλο βαθμό την επεξεργασία της πληροφορίας που υπάρχει στον Web. Έτσι, από μία σελίδα Web, οι άνθρωποι θα μπορούν να διαβάσουν την πληροφορία που περιέχει αλλά και οι υπολογιστές να κατανοήσουν σημαντικό τμήμα του περιεχομένου της και να προβούν σε συγκεκριμένες ενέργειες που έχει ορίσει ο χρήστης.

Παρακάτω θα δούμε, τι είναι ο Σημασιολογικός Ιστός, τις τεχνολογίες που υπάρχουν σε αυτόν καθώς και τον τρόπο που χρησιμοποιήθηκαν στην εφαρμογή μας. Αναλυτικότερα, στο Κεφάλαιο 1 διαβάζουμε και γνωρίζουμε την γενικότερη έννοια του Semantic Web, στο Κεφάλαιο 2 μαθαίνουμε τα βασικά της γλώσσας XML καθώς και τις δομές της, στο Κεφάλαιο 3 ασχολούμαστε με τις βασικές ιδέες και το τι είναι το RDF και στο Κεφάλαιο 4 κατανοούμε τις οντολογίες δίνοντας τον ορισμό τους και τα είδη τα οποία υπάρχουν σε συνδυασμό με τις ιδιότητες τους. Στην συνέχεια παρουσιάζουμε δύο ενδεικτικά παραδείγματα για το πού βρίσκεται εφαρμογή ο Σημασιολογικός Ιστός (Κεφάλαιο 5), αναλύουμε ακριβώς τον τρόπο λειτουργίας της διαδικτυακής εφαρμογής που υλοποιήσαμε στο Κεφάλαιο 6 και τέλος στο Κεφάλαιο 7 ολοκληρώνουμε την πτυχιακή μας με κάποια

συμπεράσματα τα οποία βγάλαμε μέσα από την ενασχόλησή μας με την ανάπτυξη αυτού του εργαλείου.

L *Σημαιολογικός Ιστός(Semantic Web)*

1.1 Στόχοι της Πτυχιακής

Η ανάπτυξη του Παγκόσμιου Ιστού (World Wide Web) κατέστησε το διαδίκτυο προσβάσιμο σε εκατομμύρια χρήστες, επιτρέποντας την δημοσιοποίηση και πρόσβαση σε έγγραφα στο διαδίκτυο. Η εκρηκτική ανάπτυξη του Παγκόσμιου Ιστού δημιούργησε προβλήματα "πληροφοριακής υπερφόρτισης".

Η παγκόσμια ερευνητική κοινότητα έχει στραφεί εδώ και λίγα χρόνια σε μία νέα κατεύθυνση εξέλιξης του ιστού, η οποία ονομάζεται "Σημαιολογικός Ιστός" (Semantic Web) και περιλαμβάνει τη σαφή αναπαράσταση του νοήματος των πληροφοριών και των εγγράφων, επιτρέποντας την αυτόματη επεξεργασία και ενοποίηση διαδικτυακών πόρων από "έξυπνα" προγράμματα-πράκτορες.

Με απλά λόγια στόχος του Semantic Web είναι το Διαδίκτυο να γίνει μια παγκόσμια πλατφόρμα ανταλλαγής και επεξεργασίας δεδομένων από ετερογενείς πηγές πληροφορίας.

Στόχος της πτυχιακής μας ήταν να βοηθήσουμε με την εφαρμογή που δημιουργήσαμε στον γρήγορο και ακριβή εντοπισμό πληροφοριών στον παγκόσμιο ιστό καθώς και να προσπαθήσουμε να παρέχουμε σε διαδικτυακούς πράκτορες τις απαραίτητες πληροφορίες ώστε να διευκολύνουμε την επικοινωνία μεταξύ πληθώρας ετερογενών ηλεκτρονικών συσκευών με πρόσβαση στο διαδίκτυο.

Υποκινούμενη από αυτήν την τεχνολογική τάση του Web στο πλαίσιο της πτυχιακής μας υλοποιήσαμε ένα διαδικτυακό εργαλείο επισημειώσεων το οποίο δίνει την δυνατότητα στον χρήστη να επισημειώνει σε μια ιστοσελίδα αντικείμενα που του είναι χρήσιμα και θα ήθελε να κάνει χρήση μελλοντικά.

Κάνοντας χρήση σύγχρονων τεχνολογιών δώσαμε στον χρήστη την ευκαιρία να κάνει διαδικτυακά και όχι τοπικά, τις σημειώσεις του σε μια σελίδα και το αρχείο το οποίο δημιουργείται να το αποθηκεύει κάπου τοπικά είτε να το ανεβάζει στο διαδίκτυο έτσι ώστε να μπορεί να κάνει χρήση του από οπουδήποτε έχει πρόσβαση σε αυτό και να συνεχίζει από το σημείο που σταμάτησε.

Χάρη όμως και στις τεχνολογίες τις οποίες χρησιμοποιήσαμε οι επισημειώσεις αυτές μπορούν, με τη σωστή χρήση τους, να γίνονται κατανοητές και από τις μηχανές. Στην ουσία ο χρήστης της εφαρμογής προσθέτει μεταδεδομένα σε μία ιστοσελίδα με σκοπό αυτά να χρησιμοποιηθούν για να γίνει πιο εύκολη η αναζήτηση πληροφοριών.

Έτσι πετυχαίνεται ο στόχος της εφαρμογής μας ο οποίος ήταν η ανάπτυξη μιας εφαρμογής για την δημιουργία επισημειώσεων και τον χειρισμό μεταδεδομένων από πλευράς χρήστη για την κατανόηση αυτών και από τους υπολογιστές.

1.2 Το Διαδίκτυο σήμερα

Ο παγκόσμιος ιστός (web) έχει αλλάξει τον τρόπο με τον οποίον επικοινωνούν οι άνθρωποι μεταξύ τους και τον τρόπο που οι επιχειρήσεις διευθύνονται. Αυτή η ανάπτυξη έχει επίσης αλλάξει τον τρόπο που εμείς σκεφτόμαστε για τους υπολογιστές. Αρχικά τους χρησιμοποιούσαμε για μαθηματικές πράξεις. Τώρα η κυρίως χρήση τους είναι για επεξεργασία πληροφοριών, εφαρμογές που προέρχονται από συστήματα βάσεων δεδομένων, επεξεργασία κειμένων και παιχνίδια.

Το περισσότερο περιεχόμενο του web είναι κατάλληλο για ανθρώπινη επεξεργασία. Ακόμα και το περιεχόμενο που παράγεται αυτόματα από τις βάσεις

δεδομένων παρουσιάζεται συνήθως χωρίς την αρχική του δομή. Τυπικές χρήσεις του web περιλαμβάνουν έρευνα και χρήση πληροφοριών από ανθρώπους, επικοινωνία με άλλα άτομα, έρευνα προϊόντων on-line καταστημάτων και παραγγελία αυτών συμπληρώνοντας φόρμες.

Αυτές οι δραστηριότητες δεν υποστηρίζονται σωστά από τα εργαλεία λογισμικού. Τα πιο χρήσιμα, αναγκαία εργαλεία είναι οι μηχανές αναζήτησης και ειδικότερα αυτές που έχουν σαν κλειδί αναζήτησης λέξεις όπως το Yahoo και Google. Είναι προφανές ότι το διαδίκτυο δε θα ήταν τόσο δημοφιλές σήμερα εάν δεν υπήρχαν οι μηχανές αναζήτησης.

Παρόλο τις βελτιώσεις στην τεχνολογία των μηχανών αναζήτησης, οι δυσκολίες ουσιαστικά παραμένουν οι ίδιες. Φαίνεται ότι η ποσότητα των περιεχομένων του web ξεπερνά την τεχνολογική πρόοδο.

Αλλά ακόμη και αν μια αναζήτηση είναι επιτυχημένη, είναι το άτομο το οποίο θα περιηγηθεί στα επιλεγμένα έγγραφα για να εξάγει τις πληροφορίες που ψάχνει. Αυτό είναι -μια που δεν υπάρχει υποστήριξη για συλλογή πληροφοριών- πολύ χρονοβόρα διαδικασία. Επίσης, αποτελέσματα από αναζήτηση στο web δεν είναι εύκολα προσβάσιμα από άλλα εργαλεία λογισμικού.

Το μεγαλύτερο εμπόδιο ώστε στους χρήστες να παρέχεται περισσότερη υποστήριξη, είναι ότι το περιεχόμενο του Παγκόσμιου Ιστού(web) δεν είναι προσπελάσιμο από μηχανές. Φυσικά υπάρχουν εργαλεία όπου ανακτούν ολόκληρα κείμενα, τα διασπούν στα επιθυμητά μέρη, ελέγχουν την ορθογραφία και μετρούν τις λέξεις. Αλλά όταν πρέπει να ερμηνεύσουν προτάσεις και να εξάγουν χρήσιμες πληροφορίες, οι ικανότητες που έχουν τα προγράμματα αυτά είναι πολύ περιορισμένες.

Είναι δύσκολη η διάκριση μεταξύ των προτάσεων

1. I am a professor of computer science. Και
2. I am a professor of computer science, you may think. Well, . . .

Μια διαφορετική προσέγγιση είναι να ξαναπαρουσιάσουμε το περιεχόμενο του web σε μια φόρμα που να είναι περισσότερο προσπελάσιμη από μηχανές και να χρησιμοποιήσουμε ευφυείς τεχνικές ώστε να εκμεταλλευτούμε αυτές τις παρουσιάσεις. Αναφερόμαστε σε αυτήν την τεχνολογική επανάσταση του web ως *Semantic Web*. Είναι σημαντικό να καταλάβουμε ότι το *Semantic Web* δε θα λειτουργήσει παράλληλα με το υπάρχον παγκόσμιο ιστό αλλά θα αναπτύσσεται βαθμιαία από αυτό.

1.3 Τεχνολογίες Semantic Web

Παρακάτω θα παρουσιάσουμε κάποια πρότυπα που είναι απαραίτητα για να πετύχουμε τη λειτουργικότητα του Semantic Web.

1.3.1 Μεταδεδομένα(Metadata)

Την περίοδο αυτή, το περιεχόμενο του web έχει καθοριστεί για τους ανθρώπους παρά από τις μηχανές. Η HTML είναι η κυρίαρχη γλώσσα με την οποία γράφονται οι ιστοσελίδες. Ένα μέρος μιας τυπικής ιστοσελίδας μοιάζει με το παρακάτω κείμενο:

```
<h1>Agilitas Physiotherapy Centre</h1>
```

Welcome to the Agilitas Physiotherapy Centre home page.

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 τους ανθρώπους η παραπάνω πληροφορία παρουσιάζεται με ικανοποιητικό τρόπο, αλλά για τις μηχανές θα υπάρξουν κάποιες δυσκολίες. Η αναζήτηση με βάση τις λέξεις “Physiotherapy” και “ Consultation hours” θα είναι επιτυχημένη. Όμως θα υπάρξει πρόβλημα στη διάκριση των λέξεων “therapists” και “secretary”, και ακόμη περισσότερο βρίσκοντας τη φράση “consultation hours” για την οποία πρέπει να ακολουθήσει το σύνδεσμο “ State Of Origin”.

Η προσέγγιση του Semantic Web προτείνει να λύσει το πρόβλημα από την πλευρά της ιστοσελίδας. Εάν η HTML αντικατασταθεί από γλώσσες που είναι πιο κατάλληλες τότε οι ιστοσελίδες μπορούν να μεταφέρουν και να παρέχουν δυνατότητες επεξεργασίας για το περιεχόμενο τους περισσότερο ικανοποιητικές. Επιπρόσθετα περιλαμβάνοντας επεξεργασμένες πληροφορίες, έχει σαν σκοπό να παραχθεί ένα έγγραφο για τους ανθρώπους, το οποίο θα μπορεί να περιέχει πληροφορίες για το περιεχόμενο του.

Στο παραπάνω παράδειγμα οι πληροφορίες μπορούν να απεικονίζονται με τον ακόλουθο τρόπο:

```

<company>
  <treatmentOffered>Physiotherapy</treatmentOffered>
  <companyName>Agilintas Physiotherapy Centre</companyName>
  <staff>
    <therapist>Lisa Davenport</therapist>
    <therapist>Steve Matthews</therapist>
    <secretary>Kelly Townsend</secretary>
  </staff>
</company>

```

Αυτή η παρουσίαση είναι πιο εύκολα επεξεργάσιμη από τις μηχανές. Ο όρος *μεταδεδομένα* αναφέρεται σε τέτοιες πληροφορίες: δεδομένα για τα δεδομένα (data about data). Με την τρέχουσα ανάπτυξη των ιστοσελίδων, οι χρήστες δεν θα είναι απαραίτητο να είναι επαγγελματίες της πληροφορικής ώστε να αναπτύσσουν ιστοσελίδες, θα είναι σε θέση να χρησιμοποιούν εργαλεία για αυτόν το λόγο. Ακόμα, η ερώτηση παραμένει, γιατί οι χρήστες πρέπει να εγκαταλείψουν την HTML για τις γλώσσες SemanticWeb.

Ίσως μπορούμε να δώσουμε μια αισιόδοξη απάντηση εάν συγκρίνουμε την κατάσταση σήμερα με την αρχική κατάσταση του Ιστοτόπου. Οι πρώτοι χρήστες αποφάσισαν να υιοθετήσουν την HTML επειδή ήταν ήδη υιοθετημένη ως πρότυπο και ανέμεναν τα οφέλη.

Άλλοι ακολούθησαν όταν περισσότερα και καλύτερα web εργαλεία έγιναν διαθέσιμα. Και σύντομα η HTML ήταν ένα παγκοσμίως αποδεκτό πρότυπο. Ομοίως, αυτό το καιρό παρατηρούμε την υιοθέτηση της XML γλώσσας. Ενώ δεν είναι επαρκής από μόνη της για την πραγματοποίηση του Semantic Web, ωστόσο η XML είναι ένα σημαντικό πρώτο βήμα.

Αυτό θα είναι ένα αποφασιστικό βήμα στο εγχείρημα του Semantic Web, αλλά είναι επίσης και μια πρόκληση. Άλλωστε η μέγιστη πρόκληση προς το

παρών δεν έχει σχέση με την επιστήμη αλλά μάλλον με την υιοθέτηση τεχνολογίας.

1.3.2 Οντολογίες

Ο όρος οντολογία (ontology) προέρχεται από τη φιλοσοφία. Σήμερα όμως, και σύμφωνα με τον ορισμό του T. R. Gruber's που αργότερα τροποποίησε ο R. Studer: *“Οι οντολογίες ορίζονται σαν μια επίσημη, ρητή προδιαγραφή μιας κοινής σύλληψης. Η σύλληψη αναφέρεται σε ένα αφηρημένο μοντέλο από φαινόμενα του κόσμου, έχοντας αναγνωρίσει τις σχετικές έννοιες από αυτά τα φαινόμενα. Ρητή, σημαίνει ότι τα είδη των εννοιών που χρησιμοποιούνται και οι περιορισμοί τους είναι σαφώς ορισμένα . Η επίσημη αναφέρεται στο γεγονός ότι η οντολογία πρέπει να είναι αναγνώσιμη από μηχανές. Η κοινή αντικατοπτρίζει ότι η οντολογία πρέπει να έχει συναινετική γνώση που να είναι αποδεκτή από τις κοινωνίες”*.

Γενικά μια οντολογία περιγράφει επίσημα μια περιοχή μιας ομιλίας. Χαρακτηριστικά, μια οντολογία αποτελείται από έναν πεπερασμένο κατάλογο όρων και των σχέσεων μεταξύ αυτών των όρων. Οι όροι υποδηλώνουν σημαντικές έννοιες (κλάσεις των αντικειμένων) της περιοχής αυτής. Για παράδειγμα, σε ένα πανεπιστήμιο, τα μέλη προσωπικού, οι σπουδαστές, τα μαθήματα, τα θέατρα διάλεξης, είναι μερικές σημαντικές έννοιες. Οι σχέσεις περιλαμβάνουν τις ιεραρχίες των κλάσεων.

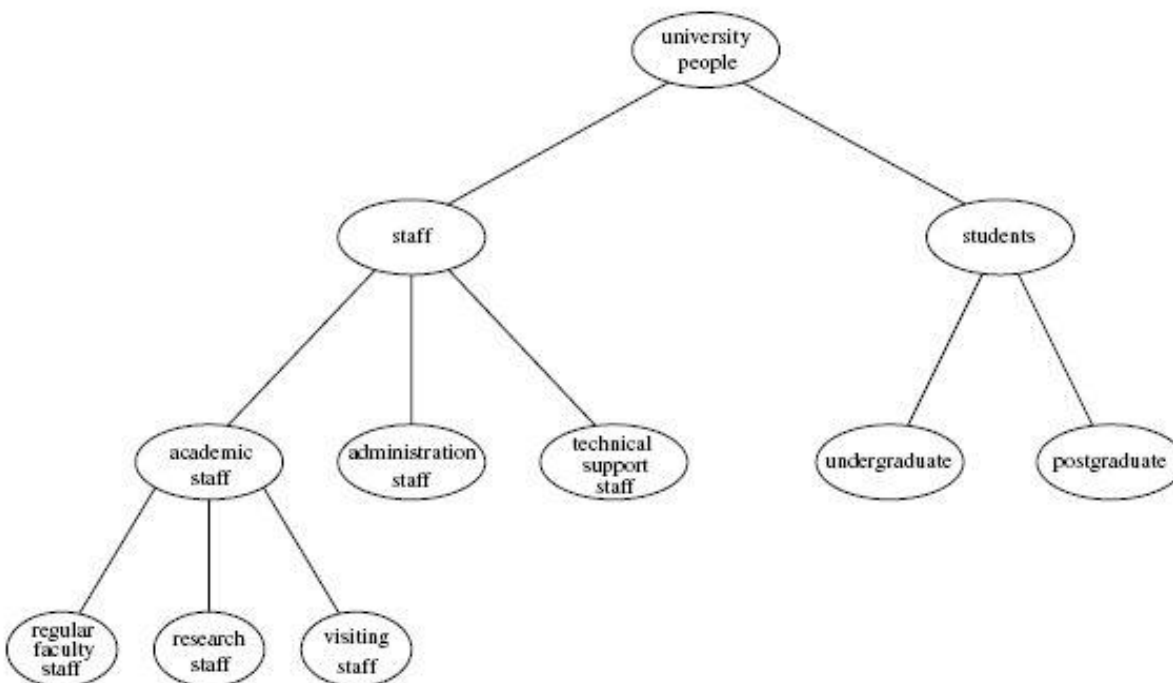


Figure 1.1 A hierarchy

Μια ιεραρχία καθορίζει μια κλάση C για να είναι μια υποκλάση μιας άλλης κλάσης C' εάν κάθε αντικείμενο στη C συμπεριλαμβάνεται επίσης στη C' . Για παράδειγμα, όλη η σχολή είναι μέλη προσωπικού. Το σχήμα 1.1 παρουσιάζει μια ιεραρχία για το εν λόγω πανεπιστήμιο.

Οι οντολογίες είναι χρήσιμες για την οργάνωση και τη πλοήγηση των ιστοχώρων. Πολλοί ιστοχώροι εκθέτουν σήμερα στην αριστερή πλευρά της σελίδας τα κορυφαία επίπεδα μιας ιεραρχίας όρων. Ο χρήστης μπορεί να κάνει κλικ σε έναν από τους όρους για να επεκτείνει τις υποκατηγορίες.

Επίσης, οι οντολογίες είναι χρήσιμες για την αναζήτηση των ιστοχώρων με περισσότερη ακρίβεια. Οι μηχανές αναζήτησης μπορούν να ψάξουν σελίδες που αναφέρονται σε μια έννοια ή σε μια οντολογία αντί της συλλογής όλων των σελίδων στις οποίες εμφανίζονται οι λέξεις κλειδιά.

Επιπλέον, οι αναζητήσεις στον Ιστοχώρο μπορούν να εκμεταλλευτούν τη γενίκευση/ειδίκευση (generalization/specialization) των πληροφοριών. Εάν ένα ερώτημα αποτύχει να βρει οποιαδήποτε σχετικά κείμενα, η μηχανή αναζήτησης μπορεί να προτείνει στο χρήστη ένα γενικότερο ερώτημα. Είναι ακόμα κατανοητό για μια μηχανή να τρέξει τέτοια ερωτήματα για να μειώσει το χρόνο αντίδρασης σε περίπτωση που ο χρήστης υιοθετεί αυτό το ερώτημα. Ή εάν πάρα πολλές απαντήσεις ανακτώνται, η μηχανή αναζήτησης μπορεί να προτείνει στο χρήστη μερικές ειδικεύσεις.

Στην τεχνητή νοημοσύνη (AI) υπάρχει μια μακροχρόνια παράδοση στην ανάπτυξη και χρησιμοποίηση των γλωσσών οντολογίας. Και εδώ μπορεί να στηριχθεί το Semantic Web για να αρχίσει να χτίζει τις αρχές του. Αυτή τη στιγμή, οι σημαντικότερες γλώσσες οντολογίας για το Web είναι οι ακόλουθες:

- Το RDF είναι ένα πρότυπο δεδομένων για αντικείμενα (“resources”) και τις σχέσεις μεταξύ τους, παρέχει μια απλή σημασιολογία για αυτό το πρότυπο δεδομένων και αυτά τα πρότυπα δεδομένων μπορούν να αντιπροσωπευθούν σε μια σύνταξη XML.
- Το RDF Schema είναι μια γλώσσα περιγραφής λεξιλογίου για την περιγραφή των ιδιοτήτων και των κλάσεων μιας RDF πηγής, με μια σημασιολογία για τις ιεραρχίες γενίκευσης τέτοιων ιδιοτήτων και κλάσεων.
- Η OWL είναι μια γλώσσα περιγραφής λεξιλογίου που περιγράφει καλύτερα τις ιδιότητες και τις κατηγορίες όπως οι σχέσεις μεταξύ των κλάσεων και άλλες πιο πολύπλοκες σχέσεις όπως ισότητες, χαρακτηριστικά ιδιοτήτων κ.α. Θεωρείται μια από τις θεμελιώδεις τεχνολογίες που υποστηρίζουν το Semantic Web.

1.3.3 Λογική

Η Λογική είναι η επιστήμη που μελετά τις αρχές της αιτίας και πηγαίνει πίσω στα χρόνια του Αριστοτέλη. Γενικά, η Λογική προσφέρει, πρώτα, έναν επίσημο τρόπο έκφρασης της γνώσης. Δεύτερον, η Λογική μας παρέχει μια κατανοητή επίσημη σημασιολογία: στις περισσότερες Λογικές, η έννοια των προτάσεων καθορίζεται χωρίς την ανάγκη να κατασταθεί λειτουργική η γνώση. Συχνά μιλάμε με δηλωτική γνώση δηλαδή εμείς περιγράφουμε το “τι” χωρίς να φροντίζουμε για το “πώς” μπορεί να εξαχθεί το συμπέρασμα. Και τρίτον αυτοματοποιημένα “reasoners” μπορούν να εξαγάγουν συμπεράσματα από τη γνώση που μας έχει δοθεί κάνοντάς την έτσι περισσότερο σαφή. Τέτοια “reasoners” έχουν μελετηθεί εκτενώς στην τεχνητή νοημοσύνη (AI).

Η Λογική μπορεί να χρησιμοποιηθεί για να αποκαλύψει τη γνώση μιας οντολογίας. Με αυτόν τον τρόπο, μπορεί επίσης να βοηθήσει ώστε να αποκαλύψει τις κρυφές και τις αστάθειες μιας σχέσης.

Αλλά η Λογική είναι πιο γενική από τις οντολογίες. Χρησιμοποιείται από ευφυείς πράκτορες (agents) για τη λήψη αποφάσεων και την επιλογή των σχεδίων δράσης.

Ένα σημαντικό πλεονέκτημα της λογικής είναι ότι μπορεί να παρέχει εξηγήσεις (*explanations*) για τα συμπεράσματα: η σειρά των βημάτων με την οποία εξάγεται ένα συμπέρασμα μπορεί να ανιχνευθεί εκ νέου.

Για τη Λογική για να είναι χρήσιμη στο Web πρέπει να χρησιμοποιείται από κοινού με άλλα δεδομένα, και πρέπει να είναι επεξεργάσιμη από τις μηχανές. Οι αρχικές προσεγγίσεις αποδίδουν σε επίπεδο XML, αλλά στο μέλλον οι κανόνες και οι αποδείξεις θα πρέπει να αντιπροσωπευθούν στο επίπεδο γλωσσών RDF και οντολογίας, όπως DAML+OIL και η OWL.

1.3.4 Πράκτορες

Οι πράκτορες είναι κομμάτια λογισμικού που λειτουργούν αυτόνομα και εκ των προτέρων. Εννοιολογικά εξελίχθηκαν από τις έννοιες του αντικειμενοστραφούς προγραμματισμού και της ανάπτυξης λογισμικού βασισμένης στα στοιχεία (components).

Ένας προσωπικός πράκτορας στο Semantic Web (σχήμα 1.2) θα λάβει μερικούς στόχους και προτιμήσεις από τον χρήστη, θα αναζητήσει πληροφορίες από το Web, θα επικοινωνήσει με άλλους πράκτορες, θα συγκρίνει τις πληροφορίες για τις απαιτήσεις και τις προτιμήσεις των χρηστών, θα επιλέξει ορισμένες επιλογές, και θα δώσει τις απαντήσεις στο χρήστη.

Πρέπει να σημειωθεί ότι οι πράκτορες δεν θα αντικαταστήσουν τους ανθρώπινους χρήστες στο Semantic Web, ούτε αυτοί θα λάβουν απαραίτητως τις αποφάσεις για αυτούς. Σε πολλές, εάν όχι στις περισσότερες περιπτώσεις ο ρόλος τους θα είναι να συλλέξουν και να οργανώσουν τις πληροφορίες, και να παρουσιάσουν τις επιλογές στους χρήστες ώστε να μπορέσουν να επιλέξουν όπως ένας ταξιδιωτικός πράκτορας όπου ψάχνει προσφορές ταξιδιού που να ταιριάζουν με τις προτιμήσεις του εκάστοτε πελάτη.

Οι πράκτορες στο Semantic Web θα χρησιμοποιήσουν όλες τις τεχνολογίες που έχουμε περιγράψει:

- Τα Μεταδεδομένα(Metadata) θα χρησιμοποιηθούν για να αναγνωρίσουν και να εξαγάγουν πληροφορίες από το Web.
- Οι οντολογίες θα χρησιμοποιηθούν για να βοηθήσουν στις αναζητήσεις του Ιστού, για να ερμηνεύσουν τις πληροφορίες που ανακτώνται, και για να επικοινωνήσουν με άλλους πράκτορες.

- Η Λογική θα χρησιμοποιηθεί για την επεξεργασία των ανακτημένων πληροφοριών και για την εξαγωγή συμπερασμάτων.

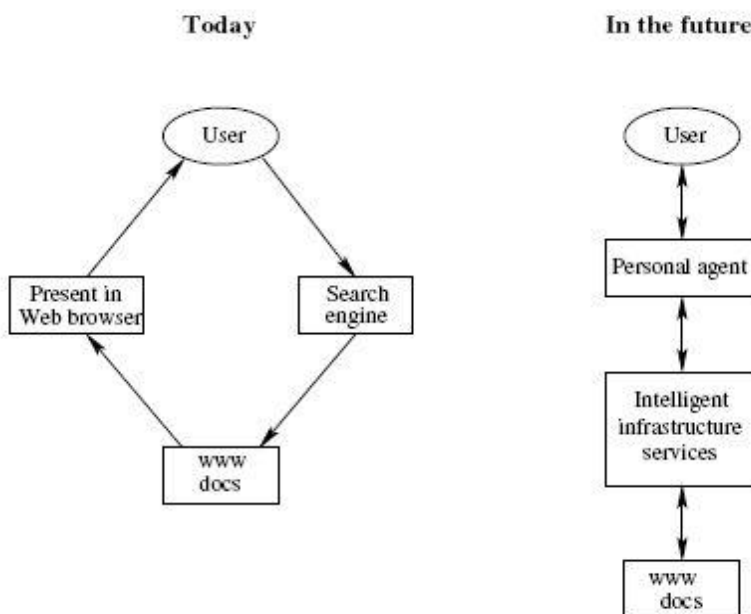


Figure 1.2 Intelligent personal agents

Περαιτέρω τεχνολογίες επίσης θα απαιτηθούν, όπως γλώσσες επικοινωνίας για τους πράκτορες. Για εφαρμογές με αυξημένες τεχνολογικές απαιτήσεις θα είναι χρήσιμο να χρησιμοποιηθούν πληροφορίες και στοιχεία που έχουν συγκεντρωθεί από πράκτορες και να δημιουργηθούν πρότυπα εύκολα στην χρήση τους από τους χρήστες.

1.3.5 Semantic Web Vs Τεχνητή Νοημοσύνη

Όπως έχουμε πει, οι περισσότερες από τις τεχνολογίες που απαιτούνται για την πραγματοποίηση του Semantic Web χτίζονται επάνω στον τομέα της τεχνητής νοημοσύνης. Λαμβάνοντας υπόψη ότι η Τεχνητή Νοημοσύνη έχει μεγάλη ιστορία, κάποιος μπορεί να πει ότι, στη χειρότερη περίπτωση, το Semantic Web θα επαναλάβει τα λάθη της Τεχνητής Νοημοσύνης.

Αυτή η ανησυχία είναι αδικαιολόγητη. Η πραγματοποίηση του οράματος του Semantic Web δε βασίζεται στην ανθρώπινη νοημοσύνη, στην πραγματικότητα, οι προκλήσεις προσεγγίζονται με έναν διαφορετικό τρόπο. Το πρόβλημα της Τεχνητής Νοημοσύνης είναι περισσότερο επιστημονικό, ίσως συγκρίσιμο με τα κύρια προβλήματα της φυσικής ή της βιολογίας.

Αλλά στο Semantic Web, λύσεις που δεν ήταν επιτυχημένες στην Τεχνητή Νοημοσύνη είναι πιθανόν να λειτουργήσουν. Ακόμα κι αν ένας ευφυής πράκτορας δεν είναι ικανός να καταλήξει σε όλα τα συμπεράσματα από ότι ένας ανθρώπινος χρήστης, ο πράκτορας θα συμβάλει καλύτερα και αποτελεσματικότερα στο Web. Βάση αυτού εντοπίζουμε μια άλλη διαφορά. Εάν ο τελικός σκοπός της Τεχνητής Νοημοσύνης είναι να δημιουργήσει έναν ευφυή πράκτορα που εκθέτει την ανθρώπινη νοημοσύνη, ο στόχος του Semantic Web είναι να βοηθήσει τους ανθρώπινους χρήστες στις καθημερινές τους δραστηριότητές.

Είναι σαφές ότι το Semantic Web κάνει εκτενή χρήση της τρέχουσας τεχνολογίας της Τεχνητής Νοημοσύνης και αυτό μας οδηγεί σε μια καλύτερη τεχνολογία για ένα καλύτερο Semantic Web. Δεν χρειάζεται όμως να περιμένουμε έως ότου φθάσει η Τεχνητή Νοημοσύνη σε ένα υψηλότερο επίπεδο, η τρέχουσα τεχνολογία είναι ήδη επαρκής για να συμβαδίσουν ταυτόχρονα προς την πραγματοποίηση του οράματος του Semantic Web.

1.4 Προσέγγιση με βάση τα επίπεδα

Η ανάπτυξη του Semantic Web προχωρά με βήματα, κάθε βήμα δημιουργεί ένα επίπεδο πάνω από ένα άλλο. Η πραγματική αιτιολόγηση για αυτήν την προσέγγιση είναι ότι είναι ευκολότερο η ανάπτυξη με μικρά βήματα, ενώ είναι δυσκολότερο η απευθείας προσέγγιση του τελικού στόχου. Συνήθως υπάρχουν αρκετές ερευνητικές ομάδες που κινούνται σε διαφορετικές κατευθύνσεις και με αυτόν τον τρόπο υπάρχει ανταγωνισμός ιδεών που είναι μια σημαντική κινητήρια δύναμη για την επιστημονική πρόοδο.

Εντούτοις, από την πλευρά της εφαρμοσμένης μηχανικής υπάρχει μια ανάγκη για τυποποίηση. Έτσι, εάν οι περισσότεροι ερευνητές συμφωνούν σε ορισμένα ζητήματα και διαφωνούν σε άλλα, έχει νόημα για να καθοριστούν τα σημεία της συμφωνίας. Με αυτόν τον τρόπο, ακόμα κι αν αποτύχουν οι πιο φιλόδοξες ερευνητικές προσπάθειες, θα υπάρξουν τουλάχιστον μερικές θετικές εκβάσεις.

Μόλις καθιερωθούν τα πρώτα πρότυπα, οι περισσότερες ομάδες και οι επιχειρήσεις θα τα υιοθετήσουν, αντί να περιμένουν για να δουν εάν οι εναλλακτικές ερευνητικές προσπάθειες θα είναι επιτυχής στο τέλος.

Η φύση του Semantic Web είναι τέτοια που οι επιχειρήσεις και οι χρήστες πρέπει να δημιουργήσουν εργαλεία, να προσθέσουν περιεχόμενο, και να χρησιμοποιήσουν αυτό το περιεχόμενο. Δεν μπορούμε να περιμένουμε έως ότου υλοποιηθεί το πλήρες όραμα για το Semantic Web, μπορεί να χρειαστούν άλλα δέκα χρόνια για να εξελιχθεί στην πλήρη μορφή του.

Στην δημιουργία ενός επιπέδου Semantic Web πάνω από ένα άλλο, δύο αρχές πρέπει να ακολουθηθούν:

- Προς τα κάτω συμβατότητα. Οι πράκτορες πλήρως ενήμεροι για ένα επίπεδο πρέπει επίσης να είναι ικανοί να ερμηνεύσουν και να χρησιμοποιήσουν τις

πληροφορίες που γράφονται σε χαμηλότερα επίπεδα. Για παράδειγμα, οι πράκτορες που είναι ενημερωμένοι για τη σημασιολογία του OWL μπορούν να εκμεταλλευθούν πλήρως τις πληροφορίες που γράφονται στο RDF και RDF Schema.

- Προς τα πάνω κατανόηση. Το σχέδιο πρέπει να είναι τέτοιο που πράκτορες ενήμεροι για ένα επίπεδο πρέπει να είναι σε θέση να αξιοποιήσουν τουλάχιστον μερικές από τις πληροφορίες στα πιο υψηλά επίπεδα. Για παράδειγμα, ένας πράκτορας ενημερωμένος μόνο για τη σημασιολογία των RDF και RDF Schema μπορεί να ερμηνεύσει τη γνώση που γράφτηκε σε OWL, χωρίς να δίνει σημασία σε εκείνα τα στοιχεία πέρα από το RDF και RDF Schema. Φυσικά, δεν υπάρχει καμία απαίτηση για όλα τα εργαλεία να παρέχουν αυτή τη δυνατότητα, το νόημα είναι ότι η επιλογή αυτή πρέπει να είναι ανοιχτή.

Ενώ αυτές οι ιδέες είναι θεωρητικά ελκυστικές και έχουν χρησιμοποιηθεί σαν κατευθυντήριες αρχές για την ανάπτυξη του Semantic Web, η πραγματοποίησή τους στην πράξη αποδείχθηκε δύσκολη, και μερικοί συμβιβασμοί έπρεπε να γίνουν.

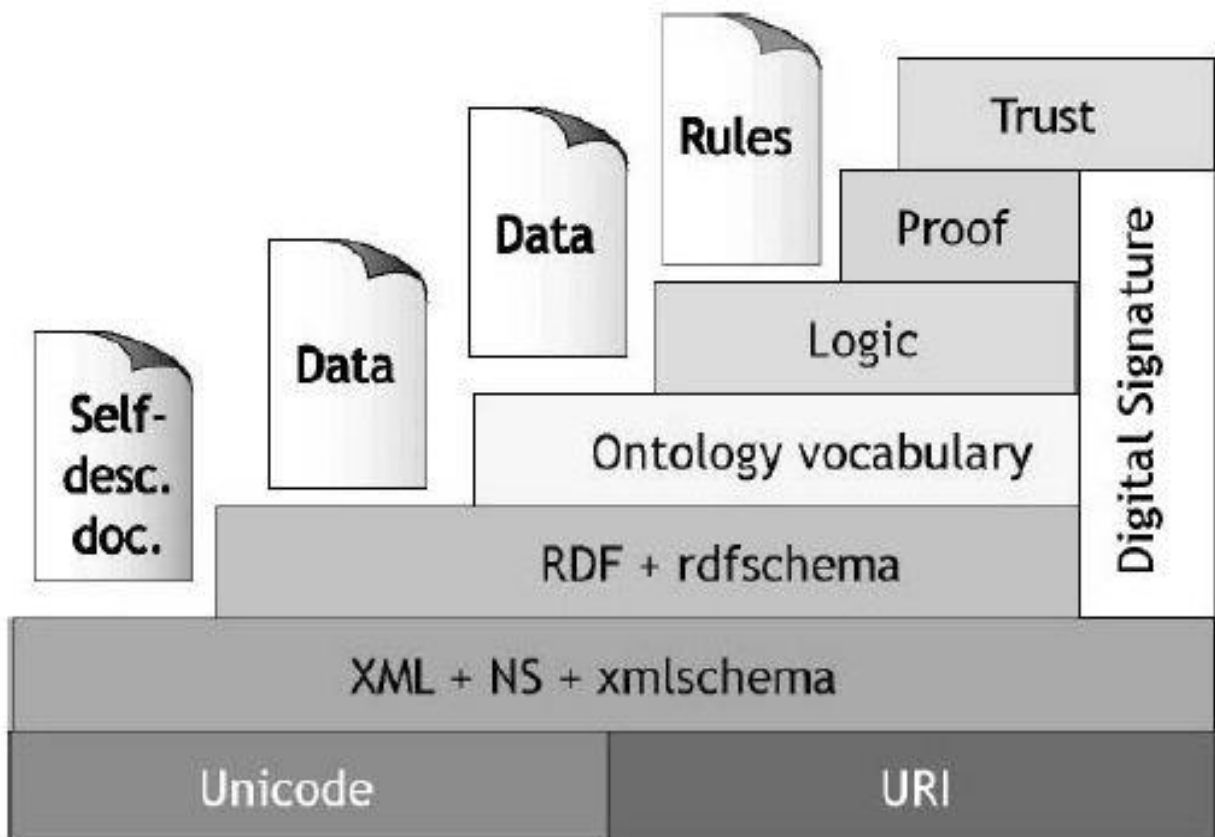
Το παρακάτω σχήμα δείχνει τα κύρια επίπεδα του Semantic Web. Στο τελευταίο επίπεδο βρίσκουμε την XML, μια γλώσσα που αφήνει έναν χρήστη να γράψει δομημένα έγγραφα όπου το λεξιλόγιο το καθορίζει ο ίδιος. Η XML είναι κατάλληλη για την αποστολή εγγράφων στο Web.

Το RDF είναι ένα βασικό πρότυπο δεδομένων, όπως το μοντέλο οντότητας - σχέσης, για το γράψιμο απλών δηλώσεων για τα Web αντικείμενα. Το πρότυπο RDF δε στηρίζεται στην XML, αλλά έχει μια σύνταξη βασισμένη στην XML. Επομένως, στο σχήμα, βρίσκεται πάνω από το επίπεδο XML.

Το RDF Schema παρέχει πρότυπα για την οργάνωση των Web αντικειμένων στις ιεραρχίες. Τα βασικά στοιχεία είναι κλάσεις και ιδιότητες, σχέσεις των υποκλάσεων και των ιδιοτήτων, και περιορισμοί περιοχών και εύρους. Το RDF Schema είναι βασισμένο στο RDF.

Το RDF Schema μπορεί να αντιμετωπισθεί σαν πρώτη γλώσσα για το γράψιμο των οντολογιών. Αλλά υπάρχει η ανάγκη για πιο ισχυρές γλώσσες οντολογίας που επεκτείνουν το RDF Schema και επιτρέπουν τις παρουσιάσεις πιο σύνθετων σχέσεων μεταξύ των Web αντικειμένων.

Το επίπεδο Logic χρησιμοποιείται για την περαιτέρω ενίσχυση της οντολογίας και για να επιτρέψει το γράψιμο δηλωτικής γνώσης.



Το επίπεδο Proof περιλαμβάνει την πραγματική παραγωγική διαδικασία καθώς επίσης και την παρουσίαση των αποδείξεων στις Web γλώσσες (από τα χαμηλότερα επίπεδα) και την επικύρωση των αποδείξεων.

Τέλος, το επίπεδο Trust θα προκύψει μέσω της χρήσης ψηφιακών υπογραφών και άλλων ειδών γνώσης, με βάση τις συστάσεις από τους πράκτορες που εμπιστευόμαστε ή από πράκτορες που έχουν βαθμολογηθεί και τους καταναλωτικούς οργανισμούς. Βρίσκοντας στην κορυφή της πυραμίδας το επίπεδο Trust μας υπενθυμίζει ότι είναι μια υψηλού επιπέδου και κρίσιμη έννοια: ο Παγκόσμιος Ιστός θα επιτύχει το μέγιστο των δυνατοτήτων του μόνο όταν οι χρήστες έχουν εμπιστοσύνη στις λειτουργίες του (ασφάλεια) και στην ποιότητα των πληροφοριών που τους παρέχεται.

Η αρχιτεκτονική Semantic Web συζητείται αυτήν την περίοδο και μπορεί στο μέλλον να υπάρχουν σημαντικές τροποποιήσεις.

2 *Αρχαία Παγκόσμιου Ιστού(Web) : XML*

2.1 Εισαγωγή

Στην σημερινή εποχή η HTML (hypertext markup language) είναι η γλώσσα στην οποία είναι γραμμένες οι ιστοσελίδες. Προήλθε από την SGML (standardized generalized markup language) η οποία αναπαριστούσε πληροφορίες οι οποίες ήταν κατανοητές τόσο από τις μηχανές όσο και από τους ανθρώπους. Η παρουσία της ήταν σημαντική καθώς βοήθησε στην αποτελεσματική επικοινωνία και στην τεχνολογική πρόοδο. Στον παγκόσμιο ιστό (WWW) τα πρότυπα καθορίζονται από τον οργανισμό W3C (World Wide Web Consortium) που σκοπό έχει την εξέλιξη του Web.

Οι γλώσσες που προσαρμόζονται στη SGML καλούνται εφαρμογές SGML. Η HTML δημιουργήθηκε επειδή η SGML δεν ήταν αρκετή για τις απαιτήσεις του διαδικτύου και είναι και αυτή μία SGML εφαρμογή. Η XML ανήκει στην κατηγορία αυτή.

Παράδειγμα xml δομής :

```
<book>
  <title>
    Nonmonotonic Reasoning: Context-Dependent Reasoning
  </title>
  <author>V. Marek</author>
  <author>M. Truszczyński</author>
  <publisher>Springer</publisher>
  <year>1993</year>
  <ISBN>0387976892</ISBN>
```


</book>

Πριν παρουσιάσουμε τις διαφορές μεταξύ της HTML και της XML ας δούμε κάποιες ομοιότητες. Όπως και η HTML έτσι και η XML βασίζεται στις ετικέτες (tags). Οι ετικέτες μπορεί να είναι εμφωλευμένα αλλά πρέπει σε κάθε περίπτωση να κλείνουν αν και στην HTML μερικές ετικέτες όπως το
 μπορεί να μείνει ανοιχτό. Το περιεχόμενο το οποίο υπάρχει μέσα στις ετικέτες καθώς και οι ετικέτες ονομάζονται στοιχεία (elements) ενός XML αρχείου.

Γενικά και οι δύο γλώσσες (HTML και XML) μπορούν εύκολα να διαβαστούν από τους ανθρώπους καθώς έχουν σχεδιαστεί γι αυτόν τον σκοπό. Μπορούν όμως να γίνουν το ίδιο εύκολα κατανοητές από τις μηχανές; Φανταστείτε έναν έξυπνο πράκτορα να προσπαθεί να ανακτήσει τα ονόματα από τους συγγραφείς των βιβλίων όπως στο προηγούμενο παράδειγμα. Υποθέστε ότι η ιστοσελίδα σε HTML μπορεί να εντοπιστεί με μια αναζήτηση στον παγκοσμιο ιστό. Δεν υπάρχει κάποια συγκεκριμένη πληροφορία για το ποιοί είναι αυτοί οι συγγραφείς. Μια λογική εικασία θα ήταν ότι τα ονόματα των συγγραφέων θα εμφανιζόταν μετά τον τίτλο ή μετά την λέξη by όμως κάτι τέτοιο δεν είναι σίγουρο.

Τα προβλήματα ξεκινάν από το γεγονός ότι η HTML δεν περιέχει δομημένη πληροφορία εν αντιθέση με την XML η οποία είναι εύκολη στην προσβασή της από τις μηχανές επειδή κάθε κομμάτι της πληροφορίας περιγράφεται λεπτομερώς. Για παράδειγμα η ετικέτα <author> εμπεριέχεται στην ετικέτα <book> και περιγράφει τις ιδιότητες ενός βιβλίου. Μια μηχανή επομένως έχει την δυνατότητα να επεξεργαστεί το XML αρχείο και να αντιληφθεί ότι το στοιχείο(element) author αναφέρεται στο στοιχείο book κάτι το οποίο δεν μπορεί να γίνει στην HTML. Ένα επιπλέον πλεονέκτημα είναι ότι η XML επιτρέπει τον καθορισμό των περιορισμών στις τιμές (για παράδειγμα καθορίζεται ότι το έτος πρέπει να είναι

ένας τετραψήφιος αριθμός και ότι πρέπει να είναι μικρότερο από το 3000). Η XML επιτρέπει την αναπαράσταση της πληροφορίας έτσι ώστε οι μηχανές να έχουν πρόσβαση σε αυτήν.

Η HTML είναι καταλληλότερη στο να απεικονίζει πληροφορία αλλά η XML διαχωρίζει το περιεχόμενο από τον τρόπο που απεικονίζεται αυτό. Η ίδια πληροφορία μπορεί να παρουσιαστεί με πολλούς διαφορετικούς τρόπους χωρίς να χρειάζονται αντίγραφα από το ίδιο περιεχόμενο.

Ας δούμε ένα παράδειγμα το οποίο περιγράφει έναν νόμο της φυσικής.

HTML

```
<h2>Relationship force-mass</h2>
<i>F = M × a</i>
```

XML

```
<equation>
  <meaning>Relationship force-mass</meaning>
  <leftside>F</leftside>
  <rightside>M × a</rightside>
</equation>
```

Εάν συγκρίνουμε την HTML δομή με αυτή που είχαμε δει στο προηγούμενο παράδειγμα παρατηρούμε ότι και στα δύο χρησιμοποιούνται οι ίδιες ετικέτες. Αυτό δεν προκαλεί έκπληξη μιας και η HTML είναι προκαθορισμένη (predefined) γλώσσα. Εν αντίθεση, το δεύτερο παράδειγμα της XML δομής χρησιμοποιεί εντελώς διαφορετικές ετικέτες από το πρώτο. Η HTML δίνει βαρύτητα στο να περιγράφει την πληροφορία έχοντας καθορισμένες ετικέτες. Στην XML μπορούμε να χρησιμοποιήσουμε την πληροφορία με πολλούς τρόπους και εξαρτάται από τον χρήστη να καθορίσει ένα "λεξιλόγιο" κατάλληλο για την εφαρμογή. Γι αυτό τον λόγο η XML δεν έχει καθορισμένες ετικέτες (tags) και επιτρέπει στους χρήστες να καθορίζουν δικές τους ετικέτες (tags).

Επειδή οι άνθρωποι δεν μπορούν να επικοινωνήσουν αποτελεσματικά οι εφαρμογές του παγκόσμιου ιστού πρέπει να συμφωνήσουν σε κοινά "λεξιλόγια". Αυτό είναι κάτι στο οποίο δραστηροποιείται το w3c και το οποίο έχει προτείνει την XML, SVG, SMIL και RDF. Πρέπει να σημειωθεί ότι η XML μπορεί να επικρατήσει ως πρότυπο για την ανταλλαγή των δεδομένων μεταξύ των εφαρμογών.

2.2 Η γλώσσα XML

Ένα xml αρχείο περιέχει κατά βάση τον πρόλογο (prolog) και στοιχεία(elements).

2.2.1 Πρόλογος (Prolog)

Η prolog είναι η δήλωση ενός XML αρχείου και περιέχει μια προαιρετική αναφορά σε εξωτερικά αρχεία. Για παράδειγμα :

```
<?xml version="1.0" encoding="UTF-16"?>
```

Καθορίζει ότι το τρέχον έγγραφο είναι ένα xml αρχείο και καθορίζει την έκδοση και την κωδικοποίηση των χαρακτήρων που χρησιμοποιείται από το σύστημα. Ο καθορισμός της κωδικοποίησης δεν είναι υποχρεωτικός αλλά η δηλωσή του είναι μια καλή πρακτική. Μερικές φορές καθορίζουμε εάν το αρχείο αναφέρεται ή όχι σε κάποια εξωτερική αναφορά.

```
<? xml version="1.0" encoding="UTF-16" standalone="no"?>
```

Μια αναφορά σε ένα εξωτερικό έγγραφο είναι ως εξής:

```
<! DOCTYPE book SYSTEM "book.dtd">
```

2.2.2 Στοιχεία (elements)

Τα XML στοιχεία(elements) αντιπροσωπεύουν τα αντικείμενα ενός xml εγγράφου όπως για παράδειγμα τα βιβλία, τους συγγραφείς και τους εκδότες. Τα στοιχεία συνθέτουν τα xml έγγραφα και αποτελούνται από την ετικέτα(tag) η οποία ανοίγει, το περιεχόμενο και την ετικέτα(tag) η οποία κλείνει.

Παράδειγμα :

```
<lecturer>David Billington</lecturer>
```

Τα όνοματα τα οποία δίνουμε στις ετικέτες πλην εξαιρέσεων μπορούμε να τα επιλέξουμε χωρίς περιορισμούς. Ένας περιορισμός είναι ότι η ετικέτα πρέπει να ξεκινάει με γράμμα, με κάτω παύλα ή με άνω και κάτω τελεία και σε καμία περίπτωση με την λέξη xml σε οποιαδήποτε μορφή (κεφαλαία - μικρά).

Το περιεχόμενο ενός στοιχείου (element) μπορεί να είναι κείμενο, άλλα στοιχεία ή να είναι κενό. Για παράδειγμα:

```
<lecturer>
  <name>David Billington</name>
  <phone>+61-7-3875 507</phone>
</lecturer>
```

Εάν δεν υπάρχει περιεχόμενο το στοιχείο χαρακτηρίζεται ως κενό(empty).

```
<lecturer></lecturer>
```

2.2.3 Ιδιότητες (attributes)

Ενα άδειο στοιχείο δεν σημαίνει ότι δεν περιέχει κάποια πληροφορία μιας και αυτή μπορεί να εμπεριέχεται στις ιδιότητες (attributes) του στοιχείου. Ιδιότητα είναι μια τιμή με όνομα μέσα σε μια ετικέτα η οποία ανοίγει :

```
<lecturer name="David Billington" phone="+61-7-3875 507"/>
```

Παρακάτω βλέπουμε ένα παράδειγμα με τις ιδιότητες ενός στοιχείου :

```
<order orderNo="23456" customer="John Smith"
      date="October 15, 2002">
  <item itemNo="a528" quantity="1"/>
  <item itemNo="c817" quantity="3"/>
</order>
```

Η ίδια πληροφορία θα μπορούσε να δοθεί αντικαθιστώντας τις ιδιότητες με εμφωλευμένα στοιχεία :

```
<order>
  <orderNo>23456</orderNo>
  <customer>John Smith</customer>
  <date>October 15, 2002</date>
  <item>
    <itemNo>a528</itemNo>
    <quantity>1</quantity>
  </item>
  <item>
    <itemNo>c817</itemNo>
    <quantity>3</quantity>
  </item>
</order>
```

Εδώ να σημειώσουμε ότι οι ιδιότητες σε ένα xml έγγραφο δεν μπορεί να είναι εμφωλευμένες.

2.2.4 Σχόλια

Τα σχόλια είναι ένα κομμάτι του αρχείου το οποίο αγνοείται όταν διαβάζεται το xml αρχείο.

```
<!-- This is a comment -->
```

2.2.5 Οδηγίες Επεξεργασίας(Processing Instructions)

Οι οδηγίες αυτές παρέχουν έναν μηχανισμό στο πως θα περαστούν πληροφορίες σε μια εφαρμογή και στο πως αυτή θα χειριστεί τα στοιχεία.

Παράδειγμα :

```
<?stylesheet type="text/css" href="mystyle.css"?>
```

2.2.6 Η κατάλληλη δομή των XML εγγράφων

Ένα XML έγγραφο είναι σωστά δομημένο εάν συντακτικά είναι σωστό. Παρακάτω βλέπουμε κάποιους συντακτικούς κανόνες :

- Υπάρχει μόνο ένα στοιχείο στο έγγραφο το οποίο αποκαλείται ρίζα.
- Κάθε στοιχείο αποτελείται από μια ετικέτα η οποία ανοίγει και πρέπει οπωσδήποτε να κλείσει.
- Εάν ανοίξουν δύο ετικέτες διαδοχικά θα πρέπει να κλείσει η πρώτη και στην συνέχεια η δεύτερη

```
<author><name>Lee Hong</author></name>.
```

- Οι ιδιότητες ενός στοιχείου έχουν μοναδικά ονόματα.
- Τα στοιχεία και τα ονόματα των ετικετών πρέπει να είναι επιτρεπτά.

2.2.7 Το δένδρο των XML εγγράφων

Υπάρχει η δυνατότητα να αναπαριστούμε XML έγγραφα σε δενδροειδή δομές. Για παράδειγμα :

```
<? xml version="1.0" encoding="UTF-16"?>
<!DOCTYPE email SYSTEM "email.dtd">
<email>
```

```

<head>
  <from name="Michael Maher"
        address="michaelmaher@cs.gu.edu.au"/>
  <to name="Grigoris Antoniou"
        address="grigoris@cs.unibremen.de"/>
  <subject>Where is your draft? </subject>
</head>
<body>
  Grigoris, where is the draft of the paper
  you promised me last week?
</body>
</email>

```

Παρακάτω βλέπουμε την δενδροειδή δομή ενός XML εγγράφου η οποία υπακούει στους συντακτικούς κανόνες.

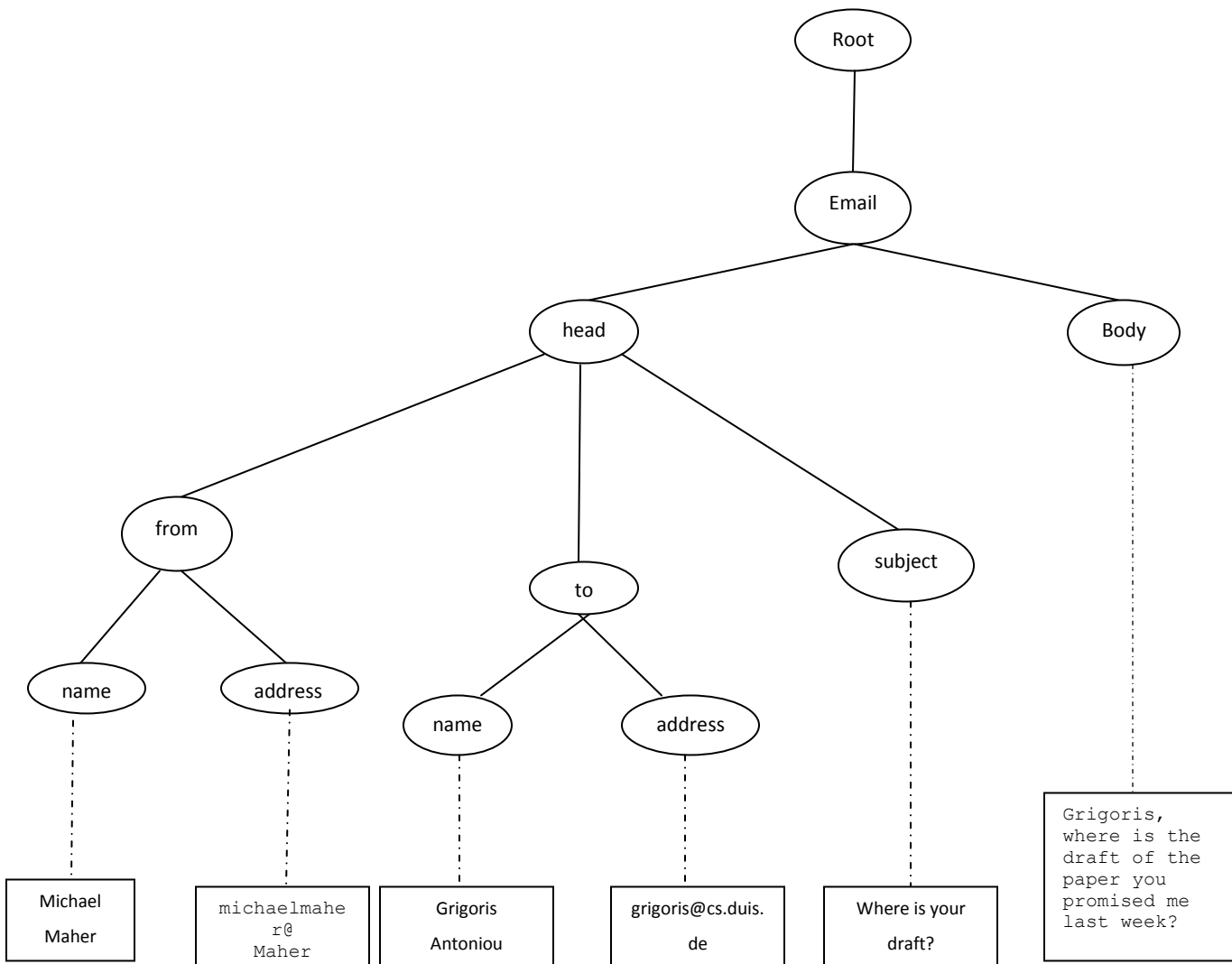
- Υπάρχει μία ρίζα
- Δεν υπάρχουν κύκλοι
- Κάθε κόμβος εκτός της ρίζας έχει έναν γονέα
- Κάθε κόμβος έχει μια ετικέτα
- Η σειρά των στοιχείων είναι σημαντική.

Παρόλο που η σειρά των στοιχείων είναι σημαντική κάτι τέτοιο δεν ισχύει για τις ιδιότητες. Τα παρακάτω δύο στοιχεία θεωρούνται ότι είναι το ίδιο :

```

<person lastname="Woo" firstname="Jason"/>
<person firstname="Jason" lastname="Woo"/>

```



2.3 Δομή(Structure)

Ένα σωστό δομημένο XML έγγραφο πρέπει να εναρμονίζεται στους συντακτικούς κανόνες που όμως δεν μπορούμε να πούμε πως είναι συγκεκριμένοι.

Ας υποθέσουμε ότι τώρα έχουμε δύο εφαρμογές οι οποίες προσπαθούν να επικοινωνήσουν και επιθυμούν να χρησιμοποιήσουν το ίδιο "λεξιλόγιο". Γι αυτόν τον σκοπό είναι απαραίτητο να καθορίζουμε όλα τα στοιχεία και τις ιδιότητες τις οποίες πρόκειται να χρησιμοποιήσουμε. Επιπλέον πρέπει να καθοριστεί, τι τιμές

μπορούν να πάρουν οι ιδιότητες, ποια στοιχεία μπορούν ή πρέπει να είναι εμφωλευμένα σε άλλα στοιχεία κ.τ.λ.

Υπάρχουν δύο τρόποι για να καθορίσουμε την δομή XML εγγράφων :DTDs ο οποίος είναι ο παλαιότερος και πιο αυστηρός και XML schema ο οποίος είναι πιο ελαστικός και με περισσότερες δυνατότητες επέκτασης.

2.3.1 DTDs

Εξωτερικά και εσωτερικά DTDs

Τα περιεχόμενα ενός DTD μπορεί να είναι σε ξεχωριστό φάκελο(εξωτερικό DTD) ή μέσα σε ένα XML αρχείο(εσωτερικό DTD). Συνήθως είναι καλύτερα να χρησιμοποιούμε εξωτερικά DTDs επειδή οι αναφορές τους μπορούν να χρησιμοποιηθούν σε πολλά έγγραφα με αποτέλεσμα να είναι πιο εύκολη η συντήρησή τους.

Ας θεωρήσουμε το στοιχείο ενός XML εγγράφου

```
<lecturer>
  <name>David Billington</name>
  <phone>+61-7-3875 507</phone>
</lecturer>
```

όπως είχαμε δει σε προηγούμενο παράδειγμα. Ένα DTD γι' αυτό το στοιχείο είναι:

```
<! ELEMENT lecturer (name, phone)>
<! ELEMENT name (#PCDATA)>
<!ELEMENT phone (#PCDATA)>
```

Η σημασία αυτού του DTD είναι :

- Ότι οι τύποι του στοιχείου lecturer, name και phone μπορούν να χρησιμοποιηθούν στο έγγραφο.

- Ένα στοιχείο lecturer περιέχει ένα τύπο name και ένα phone με αυτή την σειρά.
- Το στοιχείο name και το στοιχείο phone μπορούν να έχουν οποιοδήποτε περιεχόμενο.

Μπορούμε να εκφράσουμε ότι το στοιχείο lecture μπορεί να περιέχει το name στοιχείο ή το phone στοιχείο με τον παρακάτω τρόπο :

```
<!ELEMENT lecturer (name|phone)>
```

Είναι πιο δύσκολο όταν θέλουμε να καθορίσουμε ότι το στοιχείο lecturer περιέχει το στοιχείο name και το στοιχείο phone, σε οποιαδήποτε σειρά.

```
<! ELEMENT lecturer ((name, phone)| (phone, name))>
```

Ιδιότητες(attributes)

Έστω το στοιχείο :

```
<order orderNo="23456" customer="John Smith"
      date="October 15, 2002">
  <item itemNo="a528" quantity="1"/>
  <item itemNo="c817" quantity="3"/>
</order>
```

Ένα DTD για το στοιχείο αυτό είναι :

```
<! ELEMENT order (item+)>
<! ATTLIST order
  orderNo ID #REQUIRED
  customer CDATA #REQUIRED
  date CDATA #REQUIRED>
<! ELEMENT item EMPTY>
<! ATTLIST item
  itemNo ID #REQUIRED
  quantity CDATA #REQUIRED
  comments CDATA #IMPLIED>
```

Εάν το συγκρίνουμε με το προηγούμενο παράδειγμα βλέπουμε ότι το στοιχείο `item` καθορίζεται να είναι άδειο. Παρατηρούμε ότι μπροστά από `item` υπάρχει ο τελεστής `+`. Υπάρχουν τρεις τελεστές :

? : εμφανίζεται μηδέν ή μια φορά

* : εμφανίζεται μηδέν ή περισσότερες φορές

+ : εμφανίζεται μια ή περισσότερες φορές

Κανένας από τους τελεστές αυτούς δεν σημαίνει ότι εμφανίζεται ακριβώς μια. Επιπλέον όταν καθορίζουμε στοιχεία θα πρέπει να καθορίζουμε και ιδιότητες. Αυτό γίνεται στην λίστα των ιδιοτήτων (`attribute list`). Το πρώτο συστατικό είναι το όνομα του στοιχείου ακολουθούμενο από το όνομα της ιδιότητας, τον τύπο και τον τύπο των τιμών. Το όνομα μιας ιδιότητας είναι ένα όνομα το οποίο χρησιμοποιείται σε ένα XML έγγραφο χρησιμοποιώντας το DTD.

Τύποι Ιδιοτήτων (Attribute types)

Οι τύποι ιδιοτήτων είναι παρόμοιοι με τους προκαθορισμένους τύπους δεδομένων. Οι πιο σημαντικοί είναι :

- `CDATA`, μια λέξη (ακολουθία από χαρακτήρες),
- `ID`, ένα όνομα το οποίο είναι μοναδικό σε ολόκληρο το XML έγγραφο
- `IDREF`, μια αναφορά σε ένα άλλο στοιχείο το οποίο έχει μια `ID` τιμή ίδια με την `IDREF`,
- `IDREFS`, μια σειρά από `IDREFs`,
- `(u1|...|un)`, μια δομή από όλες τις πιθανές τιμές.

Η επιλογές δεν είναι ικανοποιητικές. Για παράδειγμα οι ημερομηνίες και τα νούμερα δεν μπορούν να καθοριστούν και πρέπει να μεταγλωττιστούν ως συμβολοσειρές.

Τύποι τιμών

- #REQUIRED. Η ιδιότητα πρέπει να εμφανίζεται μέσα στο XML έγγραφο.
- #IMPLIED. Η εμφάνιση της ιδιότητας είναι προαιρετική. Για παράδειγμα τα σχόλια είναι προαιρετικά.
- #FIXED “value”. Κάθε στοιχείο πρέπει να έχει αυτή την ιδιότητα της οποίας η τιμή δίνεται μετά από #FIXED στο DTD. Όταν δίνουμε μια τιμή σε ένα XML έγγραφο δεν σημαίνει τίποτα επειδή υπερβαίνεται η τιμή #FIXED.
- “value”. Καθορίζει την εξ ορισμού τιμή για την ιδιότητα. Εάν μια συγκεκριμένη τιμή εμφανίζεται σε ένα XML έγγραφο υπερβαίνει την εξ ορισμού.

Παρακάτω βλέπουμε ένα παράδειγμα από το IDREF και IDREFs. Πρώτα ας δούμε το DTD:

```
<! ELEMENT family (person*)>
<! ELEMENT person (name)>
<! ELEMENT name (#PCDATA)>
<! ATTLIST person
    id ID #REQUIRED
    mother IDREF #IMPLIED
    father IDREF #IMPLIED
    children IDREFS #IMPLIED>
```

Ένα XML στοιχείο το οποίο εναρμονίζεται στο παραπάνω DTD :

```
<family>
    <person id="bob" mother="mary" father="peter">
```

```

        <name>Bob Marley</name>
    </person>
    <person id="bridget" mother="mary">
        <name>Bridget Jones</name>
    </person>
    <person id="mary" children="bob bridget">
        <name>Mary Poppins</name>
    </person>
    <person id="peter" children="bob">
        <name>Peter Marley</name>
    </person>
</family>

```

XML οντότητες (entities)

Μια XML οντότητα μπορεί να παίζει πολλούς ρόλους, ως μέρος τοποθέτησης χαρακτήρων που επαναλαμβάνονται, μέρος εξωτερικών δεδομένων ή ως μέρος καθορισμού στοιχείων. Μια τυπική χρησιμότητα των εσωτερικών οντοτήτων είναι παρόμοια με τις σταθερές στις γλώσσες προγραμματισμού. Για παράδειγμα, υποθέστε ότι ένα έγγραφο έχει αναφορές σχετικά με το πότε εκδόθηκε η ανάλογη οντότητα, θα είναι :

```
<!ENTITYthisyear"2007">
```

Μετά από αυτή την δήλωση όπου χρειαστεί να χρησιμοποιηθεί το τρέχων έτος μπορούμε να χρησιμοποιήσουμε αυτή την οντότητα &thisyear. Ο μόνος τρόπος για να ενημερώσουμε το έτος σε 2008 είναι να αλλάξουμε την δηλωση της οντότητας.

Ας δούμε ένα συνοπτικό παράδειγμα :

```

<! ELEMENT email (head, body)>
<! ELEMENT head (from, to+, cc*, subject)>

```

```

<! ELEMENT from EMPTY>
<! ATTLIST from
    name CDATA #IMPLIED
    address CDATA #REQUIRED>
<! ELEMENT to EMPTY>
<! ATTLIST to
    name CDATA #IMPLIED
    address CDATA #REQUIRED>
<! ELEMENT cc EMPTY>
<! ATTLIST cc
    name CDATA #IMPLIED
    address CDATA #REQUIRED>
<! ELEMENT subject (#PCDATA)>
<! ELEMENT body (text, attachment*)>
<! ELEMENT text (#PCDATA)>
<! ELEMENT attachment EMPTY>
<! ATTLIST attachment
    encoding (mime|binhex) "mime"
    file CDATA #REQUIRED>

```

Εδώ θα παρατηρήσουμε κάποια ενδιαφέροντα κομμάτια αυτού του DTD :

- Το στοιχείο head περιέχει ένα στοιχείο from, τουλάχιστον ένα το στοιχείο, μηδέν ή περισσότερα cc στοιχεία και ένα subject στοιχείο.
- Στο from, to και το cc στοιχείο η ιδιότητα name δεν απαιτείται ενώ η ιδιότητα address απαιτείται.
- Το στοιχείο body περιέχει ένα στοιχείο text πιθανότατα ακολουθούμενο από έναν αριθμό attachment στοιχείων.
- Η ιδιότητα encoding σε ένα attachment στοιχείο πρέπει να έχει την τιμή “mime” ή “binhex”.

2.3.2 XML Schema

Το XML schema προσφέρει περισσότερες δυνατότητες στην δημιουργία XML εγγράφων. Ένα από τα χαρακτηριστικά του είναι ότι η σύνταξη βασίζεται στην XML εξ ολοκλήρου. Με το XML schema δεν χρειάζεται να υπάρχει ξεχωριστή σύνταξη όπως στο DTD. Μια ακόμα σημαντική βελτίωση είναι ότι μπορούν να καθοριστούν καινούριοι τύποι επεκτείνοντας ή περιορίζοντας ήδη υπάρχοντες. Τέλος το XML schema παρέχει τύπους δεδομένων οι οποίοι μπορούν να χρησιμοποιηθούν σε XML έγγραφα. (Στο DTD περιοριζόμαστε μόνο στις ακολουθίες χαρακτήρων).

Παράδειγμα στοιχείου XML schema :

```
<xsd:schema
  xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
  version="1.0">
```

Στοιχεία(elements)

Η σύνταξη ενός στοιχείου στο XML schema :

```
<element name=". . ."/>
```

μπορεί να έχει μεγάλο αριθμό προαιρετικών ιδιοτήτων όπως η type

```
type=". . ."
```

ή περιορισμούς

- `minOccurs="x"`, όπου το x μπορεί να είναι οποιοσδήποτε φυσικός αριθμός (συμπεριλαμβανομένου του μηδεν).
- `maxOccurs="x"`, όπου το x μπορεί να είναι οποιοσδήποτε φυσικός αριθμός (συμπεριλαμβανομένου του μηδέν) ή απεριόριστος.

MinOccurs και maxOccurs αποτελούν γενικεύσεις των τελεστών ?,*,+ οι οποίοι ισχύουν στο DTD. Όταν αυτοί οι περιορισμοί δεν διευκρινίζονται, το minOccurs και maxOccurs, έχουν την τιμή 1 εξ ορισμού.

Ας δούμε κάποια παραδείγματα.

```
<element name="email"/>
<element name="head" minOccurs="1" maxOccurs="1"/>
<element name="to" minOccurs="1"/>
```

Τύποι Ιδιοτήτων(attribute types)

Παράδειγμα σύνταξης των ιδιοτήτων

```
<attribute name="..." />
```

Μπορεί να έχει πολλές προαιρετικές ιδιότητες όπως

```
type="..."
```

ή

use="x" όπου το x μπορεί να είναι προαιρετικό, απαιτούμενο ή απαγορευμένο.

ή

εξ ορισμού τιμές

Ας δούμε κάποια παραδείγματα :

```
<attribute name="id" type="ID" use="required"/>
<attribute name="speaks" type="Language" use="optional"
default="en"/>
```

Τύποι δεδομένων

Έχουμε ήδη αναγνωρίσει ως μειονέκτημα του DTD την ύπαρξη αυστηρών τύπων δεδομένων. Το XML schema παρέχει έναν δυναμικό τρόπο στον καθορισμό τύπων δεδομένων. Καταρχήν υπάρχει μια ποικιλία από κατασκευαστικούς τύπους δεδομένων(built in data types). Ας δούμε μερικά :

- Αριθμητικούς τύπους δεδομένων (integer,short,Byte,long,float,decimal)
- Συμβολοσειρές(strings) περιλαμβάνει string, ID, IDREF, CDATA, language
- Τύπους δεδομένων αναφορικά με μέρες και χρόνο, time, date, gMonth, gYear

Επίσης υπάρχουν προκαθορισμένοι τύποι δεδομένων εκ μέρους του χρήστη(user-defined data types) οι οποίοι χρησιμοποιούν απλούς τύπους δεδομένων(*simple data types*). Αυτοί δεν μπορούν να χρησιμοποιήσουν στοιχεία ή ιδιότητες και περίπλοκους τύπους δεδομένων(*complex data types*), οι οποίοι όμως χρησιμοποιούν στοιχεία και ιδιότητες. Θα συζητήσουμε αρχικά τους περίπλοκους τύπους δεδομένων αναβάλλοντας αυτή των απλών μέχρι να συζητήσουμε σχετικά με την απαγόρευση. Οι περίπλοκοι τύποι καθορίζονται απο τύπους δεδομένων που προϋπήρχαν καθορίζοντας κάποιες ιδιότητες και χρησιμοποιούν :

- Μια ακολουθία από τύπους δεδομένων που προϋπήρχαν, η σειρά των οποίων είναι σημαντική.
- Συλλογή στοιχείων που πρέπει να εμφανίζονται αλλά η σειρά τους δεν έχει σημασία.
- Συλλογή αντικειμένων από τα οποία θα επιλεγεί ένα.

Παράδειγμα :

```
<complexType name="lecturerType">
  <sequence>
    <element name="firstname" type="string"
      minOccurs="0" maxOccurs="unbounded"/>
    <element name="lastname" type="string"/>
  </sequence>
  <attribute name="title" type="string" use="optional"/>
</complexType>
```

Η σημασία του παραδείγματος είναι ότι ένα στοιχείο σε ένα XML έγγραφο το οποίο δηλώνεται να είναι τύπου `lecturerType` μπορεί να έχει ως ιδιότητα το `title`, να περιλαμβάνει στοιχεία `firstname` και πρέπει να περιέχει ακριβώς ένα στοιχείο `lastname`.

Επέκταση των τύπων δεδομένων

Οι τύποι δεδομένων που υπάρχουν μπορούν να επεκταθούν με νέα στοιχεία και ιδιότητες. Ας δούμε ένα παράδειγμα όπου επεκτείνουμε τον τύπο `lecturer`.

```
<complexType name="extendedLecturerType">
  <extension base="lecturerType">
    <sequence>
      <element name="email" type="string"
        minOccurs="0" maxOccurs="1"/>
    </sequence>
    <attribute name="rank" type="string" use="required"/>
  </extension>
</complexType>
```

Στο παράδειγμα αυτό ο τύπος `lecturer` επεκτείνεται με ένα στοιχείο `email` και με μια ιδιότητα `rank`. Το αποτέλεσμα που προκύπτει είναι :

```
<complexType name="extendedLecturerType">
  <sequence>
    <element name="firstname" type="string"
      minOccurs="0" maxOccurs="unbounded"/>
    <element name="lastname" type="string"/>
    <element name="email" type="string"
      minOccurs="0" maxOccurs="1"/>
  </sequence>
  <attribute name="title" type="string" use="optional"/>
  <attribute name="rank" type="string" use="required"/>
</complexType>
```

```
</complexType>
```

Υπάρχει μια σχέση μεταξύ του αρχικού τύπου δεδομένων και της επέκτασής του. Οι αναφορές των τύπων δεδομένων που έχουν επεκταθεί αποτελούν και αναφορές του αρχικού τύπου δεδομένων. Μπορεί να περιλαμβάνουν επιπλέον πληροφορίες αλλά σε καμία περίπτωση δεν μπορούν να περιλαμβάνουν λιγότερες πληροφορίες.

Περιορισμοί τύποι δεδομένων

Ένας υπάρχων τύπος δεδομένων μπορεί να περιοριστεί προσθέτοντας περιορισμούς στις τιμές τις οποίες μπορεί να πάρει. Είναι σημαντικό να καταλάβουμε ότι ο περιορισμός ενός τύπου δεδομένων δεν σημαίνει ότι δεν μπορεί να υπάρξει επέκταση αυτού του ίδιου τύπου δεδομένων. Ο περιορισμός δεν επιτυγχάνεται διαγράφοντας στοιχεία και ιδιότητες. Γι αυτό ισχύει ότι και όταν επεκτείνουμε ένα τύπο δεδομένων : αναφορές των τύπων δεδομένων που έχουν περιοριστεί αποτελούν και αναφορές του αρχικού τύπου δεδομένων. Ικανοποιούν τουλάχιστον τους περιορισμούς του αρχικού τύπου δεδομένων και προσθέτει κι άλλους.

Ας δούμε ένα παράδειγμα όπου περιορίζεται ο τύπος lecturer :

```
<complexType name="restrictedLecturerType">
  <restriction base="lecturerType">
    <sequence>
      <element name="firstname" type="string"
        minOccurs="1" maxOccurs="2"/>
    </sequence>
    <attribute name="title" type="string" use="required"/>
  </restriction>
</complexType>
```

Οι περιορισμοί απεικονίζονται με έντονα γράμματα.

Απλοί τύποι δεδομένων μπορούν να προσδιοριστούν απλά περιορίζοντας ήδη υπάρχοντες τύπους. Για παράδειγμα μπορούμε να προσδιορίσουμε τον τύπο `dayOfMonth` ο οποίος δέχεται τιμές από 1 έως και 31 :

```
<simpleType name="dayOfMonth">
  <restriction base="integer">
    <minInclusive value="1"/>
    <maxInclusive value="31"/>
  </restriction>
</simpleType>
```

Είναι επίσης δυνατό να καθορίζουμε ένα τύπο δεδομένων παραθέτοντας όλες τις πιθανές τιμές τις οποίες μπορεί να πάρει. Για παράδειγμα μπορούμε να καθορίσουμε έναν τύπο `dayOfWeek` :

```
<simpleType name="dayOfWeek">
  <restriction base="string">
    <enumeration value="Mon"/>
    <enumeration value="Tue"/>
    <enumeration value="Wed"/>
    <enumeration value="Thu"/>
    <enumeration value="Fri"/>
    <enumeration value="Sat"/>
    <enumeration value="Sun"/>
  </restriction>
</simpleType>
```

Παρακάτω βλέπουμε ένα συγκεντρωτικό παράδειγμα για ότι έχουμε αναφέρει προσδιορίζοντας ένα XML schema για το email :

```
<element name="email" type="emailType"/>
<complexType name="emailType">
  <sequence>
    <element name="head" type="headType"/>
    <element name="body" type="bodyType"/>
```

```

    </sequence>
</complexType>

<complexType name="headType">
    <sequence>
        <element name="from" type="nameAddress"/>
        <element name="to" type="nameAddress"
            minOccurs="1" maxOccurs="unbounded"/>
        <element name="cc" type="nameAddress"
            minOccurs="0" maxOccurs="unbounded"/>
        <element name="subject" type="string"/>
    </sequence>
</complexType>

<complexType name="nameAddress">
    <attribute name="name" type="string" use="optional"/>
    <attribute name="address" type="string" use="required"/>
</complexType>

<complexType name="bodyType">
    <sequence>
        <element name="text" type="string"/>
        <element name="attachment" minOccurs="0"
            maxOccurs="unbounded">
    </sequence>
</complexType>

<complexType>
    <attribute name="encoding" use="optional"
        default="mime">
        <simpleType>
            <restriction base="string">
                <enumeration value="mime"/>
                <enumeration value="binhex"/>
            </restriction>
        </simpleType>
    </attribute>

```

```
<attribute name="file" type="string"  
          use="required"/>  
</complexType>  
</element>  
</sequence>  
</complexType>
```

3 Περιγράφοντας τις πηγές του Web: RDF

3.1 Εισαγωγή

Η XML είναι μια καθολική μεταγλώσσα για τον καθορισμό της σήμανσης. Παρέχει ένα πλαίσιο, και ένα σύνολο εργαλείων, για την ανταλλαγή των δεδομένων και των μεταδεδομένων μεταξύ των εφαρμογών. Όμως, η XML δεν παρέχει μέσα για τη σημασιολογία των δεδομένων. Για παράδειγμα, δεν υπάρχει σκόπιμη έννοια που να σχετίζεται με τις φωλιασμένες ετικέτες, έγκειται στην ευχέρεια της κάθε εφαρμογής να ερμηνεύσει τα φωλιασμένα στοιχεία. Σε αυτό το σημείο θα χρησιμοποιήσουμε ένα παράδειγμα. Υποθέστε ότι θέλουμε να εκφράσουμε το ακόλουθο γεγονός:

David Billington is a lecturer of Discrete Mathematics.

Υπάρχουν διάφοροι τρόποι να παρουσιάσουμε αυτήν την πρόταση σε XML. Τρεις πιθανοί είναι:

```
<course name="Discrete Mathematics">
  <lecturer>David Billington</lecturer>
</course>
```

```
<lecturer name="David Billington">
  <teaches>Discrete Mathematics</teaches>
</lecturer>
```

```
<teachingOffering>
  <lecturer>David Billington</lecturer>
```

```
<course>Discrete Mathematics</course>
</teachingOffering>
```

Προσέξτε ότι οι πρώτοι δύο τρόποι αν και αντιπροσωπεύουν τις ίδιες πληροφορίες έχουν τοποθετηθεί αντίθετα μεταξύ τους. Έτσι δεν υπάρχει κανένας τυποποιημένος τρόπος ανάθεσης των εννοιών των φωλιασμένων ετικετών. Αν και την ονομάζουμε συχνά «γλώσσα» την RDF (Resource Description Framework) είναι ουσιαστικά ένα πρότυπο δεδομένων.

Η βασική δομική μονάδα της αποτελείται από τρία μέρη το αντικείμενο, την ιδιότητα, και την τιμή, αποκαλούμενη ως πρόταση. Το προηγούμενο παράδειγμα είναι μια τέτοια πρόταση. Φυσικά, ένα αφηρημένο πρότυπο δεδομένων χρειάζεται μια συγκεκριμένη σύνταξη προκειμένου να παρουσιαστεί και να διαβιβαστεί, και στην RDF έχει δοθεί μια σύνταξη σε XML. Κατά συνέπεια, κληρονομεί τα οφέλη που σχετίζονται με την XML. Όμως, είναι σημαντικό να καταλάβουμε ότι άλλες συντακτικές για παρουσίαση του RDF, που δεν είναι βασισμένες σε XML, είναι επίσης δυνατές. Η XML σύνταξη δεν είναι ένα απαραίτητο συστατικό του προτύπου RDF.

Η RDF δεν υπόκειται σε καμία περιοχή (domain) δεδομένου, ότι δηλαδή δεν γίνονται υποθέσεις για μια ιδιαίτερη περιοχή. Είναι στην ευχέρεια των χρηστών να καθορίσουν την ορολογία τους σε μια γλώσσα αποκαλούμενη RDF Schema (RDFS). Το όνομα RDF Schema είναι τώρα ευρέως διαδεδομένο ως μια ανεπιτυχής επιλογή. Προτείνει ότι το RDF Schema έχει μια παρόμοια σχέση με το RDF όπως το XMLSchema με την XML, αλλά στην πραγματικότητα δεν είναι αυτό που θέλουμε. Το XML Schema περιορίζει τη δομή των εγγράφων XML, ενώ το RDF Schema καθορίζει το λεξιλόγιο που χρησιμοποιείται σε ένα πρότυπο δεδομένων RDF.

Σε ένα RDFS μπορούμε να καθορίσουμε το λεξιλόγιο, να διευκρινίσουμε

ποιες ιδιότητες ισχύουν για ποια είδη αντικειμένων και ποιες τιμές μπορούν να πάρουν, και να περιγράψουμε τις σχέσεις μεταξύ των αντικειμένων. Για παράδειγμα, μπορούμε να γράψουμε

Lecturer is a subclass of academic staff member.

Αυτή η πρόταση σημαίνει ότι όλοι οι Lecturers είναι επίσης academic staff members. Είναι σημαντικό να καταλάβουμε ότι υπάρχει μια σκόπιμη έννοια που συνδέεται με το “is a subclass of”. Δεν εξαρτάται από την εφαρμογή για να ερμηνευθεί αυτός ο όρος, είναι μια σκόπιμη έννοια και πρέπει να σεβαστεί από όλα τα λογισμικά RDF. Μέσω του καθορισμού της σημασιολογίας ορισμένων συστατικών, το RDF/RDFS επιτρέπει σε μας να διαμορφώσουμε πρότυπα για συγκεκριμένες περιοχές. Παρουσιάζουμε τη σημαντικότητα του RDF Schema με ένα παράδειγμα. Θεωρείστε τα ακόλουθα στοιχεία XML:

```
<academicStaffMember>Grigoris Antoniou</academicStaffMember>
<professor>Michael Maher</professor>
<course name="Discrete Mathematics">
  <isTaughtBy>David Billington</isTaughtBy>
</course>
```

Υποθέστε ότι θέλουμε να συλλέξουμε όλα τα ακαδημαϊκά μέλη προσωπικού (academicStaffMember). Μια έκφραση για ένα Xpath μπορεί να είναι:

```
//academicStaffMember
```

Το αποτέλεσμα θα είναι μόνο “Grigoris Antoniou”. Ενώ είναι σωστή από την άποψη της XML, αυτή η απάντηση δεν είναι σημασιολογικά ικανοποιητική

επειδή οι αναγνώστες θα είχαν επίσης συμπεριλάβει στην απάντηση τον “Michael Maher” και τον “David Billington” .

Αυτό το είδος πληροφοριών κάνει χρήση του σημασιολογικού προτύπου μιας συγκεκριμένης περιοχής και δεν μπορεί να παρουσιαστεί σε XML ή σε RDF αλλά είναι χαρακτηριστικό ενός RDF Schema. Κατά συνέπεια τα RDFS καθιστούν τις σημασιολογικές πληροφορίες επεξεργάσιμες από μηχανές, σύμφωνα με το όραμα του σημασιολογικού Ιστού.

3.2 RDF: Βασικές Ιδέες

Οι θεμελιώδεις αρχές του RDF είναι οι πηγές, οι ιδιότητες, και οι προτάσεις.

3.2.1 Πηγές

Μπορούμε να σκεφτούμε μια πηγή σαν ένα αντικείμενο, για «κάτι» που θέλουμε να μιλήσουμε. Οι πηγές μπορούν να είναι συντάκτες, βιβλία, εκδότες, θέσεις, άνθρωποι, ξενοδοχεία, δωμάτια, αναζήτησης ερωτημάτων, και ούτω καθεξής. Κάθε πηγή έχει ένα URI (Uniform Resource Identifier). Ένα URI μπορεί να είναι ένα URL (Uniform Resource Locator, ή μια διεύθυνση Web) ή κάποιο άλλο είδος μοναδικού προσδιοριστικού, σημειώστε ότι ένα προσδιοριστικό δεν επιτρέπει απαραίτητα την “πρόσβαση” σε μια πηγή.

Τα URI schemes έχουν καθοριστεί όχι μόνο για τις θέσεις στο Web αλλά και για αντικείμενα που διαφέρουν όπως τους αριθμούς τηλεφώνου, τους αριθμούς ISBN, και τις γεωγραφικές θέσεις. Γενικά, υποθέτουμε ότι ένα URI είναι το προσδιοριστικό μιας Web πηγής.

3.2.2 Ιδιότητες

Οι ιδιότητες είναι ένα ειδικό κομμάτι των πηγών, περιγράφουν τις σχέσεις μεταξύ των πηγών, παραδείγματος χάριν «γραπτός από», «ηλικία», «τίτλος», και ούτω καθεξής. Οι ιδιότητες στα RDF αναγνωρίζονται επίσης από τα URIs (και στην πράξη από τα URLs). Αυτή η ιδέα της χρησιμοποίησης των URIs για να προσδιορίσουμε «τα πράγματα» και τις σχέσεις μεταξύ τους είναι αρκετά σημαντικό. Αυτή η επιλογή μας δίνει ένα σφαιρικό, παγκόσμιο, μοναδικό scheme. Η χρήση ενός τέτοιου scheme περιορίζει πολύ το ομώνυμο πρόβλημα παρουσίασης των δεδομένων.

3.2.3 Προτάσεις

Οι προτάσεις επιβεβαιώνουν τις ιδιότητες των πηγών. Μια πρόταση είναι ένα αντικείμενο-ιδιότητα με τριπλή αξία, που αποτελείται από μια πηγή, μια ιδιότητα, και μια τιμή. Οι τιμές μπορεί είτε να είναι πηγές είτε literals. Literals είναι οι ατομικές τιμές (strings), με την δομή των οποίων δεν θα ασχοληθούμε.

3.2.4 Τρεις απόψεις για τις Προτάσεις (Statements)

Ένα παράδειγμα μιας πρότασης είναι:

David Billington is the owner of the Web page

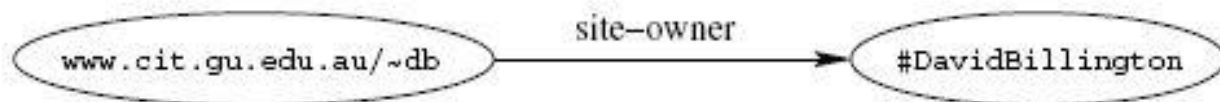
<http://www.cit.gu.edu.au/~db>.

Ο απλούστερος τρόπος για να ερμηνεύσουμε αυτήν την πρόταση είναι να χρησιμοποιήσουμε τον ορισμό

(<http://www.cit.gu.edu.au/~db>,
<http://www.mydomain.org/site-owner>, #DavidBillington).

Μπορούμε να το σκεφτούμε αυτό σαν μιας τριπλής αξίας πρότασης (x, P, y) , σαν μια λογική φόρμουλα $P(x, y)$, όπου το δυαδικό κατηγορημα P σχετίζεται με τα αντικείμενα x, y . Στην πραγματικότητα, το RDF προσφέρει μόνο δυαδικά κατηγορήματα (ιδιότητες). Σημειώστε ότι το “site-owner” και τα δύο αντικείμενα αναγνωρίζονται από τα URLs.

Η δεύτερη άποψη είναι βασισμένη στη γραφική παράσταση. Το παρακάτω σχήμα παρουσιάζει τη γραφική παράσταση που αντιστοιχεί στην παραπάνω πρόταση.



Είναι μια κατευθυνόμενη γραφική παράσταση με κόμβους και τόξα, τα τόξα κατευθύνονται από την πηγή (το υποκείμενο της πρότασης) προς την τιμή (το αντικείμενο της πρότασης). Αυτό το είδος της γραφικής παράστασης είναι γνωστό στην κοινωνία της Τεχνητής Νοημοσύνης ως σημασιολογικό δίκτυο.

Όπως έχουμε πει, η τιμή μιας πρότασης μπορεί να είναι μια πηγή. Επομένως, μπορεί να συνδεθεί με άλλες πηγές. Δείτε τα ακόλουθα παραδείγματα:

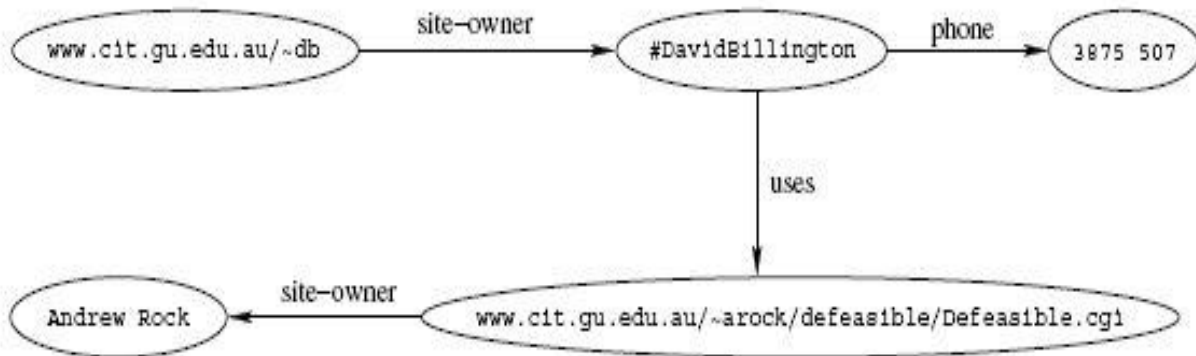
(<http://www.cit.gu.edu.au/~db>, <http://www.mydomain.org/site-owner>,
#DavidBillington)

(#DavidBillington, <http://www.mydomain.org/phone>, "3875507")

(#DavidBillington, <http://www.mydomain.org/uses>,
<http://www.cit.gu.edu.au/~arock/defeasible/Defeasible.cgi>)

(<http://www.cit.gu.edu.au/~arock/defeasible/Defeasible.cgi>,
<http://www.mydomain.org/site-owner>, "Andrew Rock")

Η γραφική παράσταση βρίσκεται στο παρακάτω σχήμα.



Οι γραφικές παραστάσεις είναι ένα ισχυρό εργαλείο ως προς την κατανόηση των χρηστών. Αλλά το όραμα του Semantic Web απαιτεί τις παρουσιάσεις να είναι προσιτές και επεξεργάσιμες από τις μηχανές.

Επομένως, υπάρχει μια τρίτη δυνατότητα παρουσίασης βασισμένη στην

XML. Σύμφωνα με αυτήν την δυνατότητα, ένα έγγραφο RDF παρουσιάζεται από ένα στοιχείο XML με την ετικέτα `rdf: RDF`. Το περιεχόμενο αυτού του στοιχείου είναι ένας αριθμός από περιγραφές, οι οποίες χρησιμοποιούν ετικέτες `rdf:Description`.

Κάθε περιγραφή κάνει μια πρόταση για μια πηγή, η οποία αναγνωρίζεται με έναν από τους τρεις τρόπους:

- Με μια ιδιότητα που αναφέρεται σε μια υπάρχον πηγή.
- Με μια ιδιότητα ταυτότητας, δημιουργώντας μια καινούρια πηγή.
- Χωρίς όνομα, δημιουργώντας έναν ανώνυμο πόρο.

Στη συνέχεια φαίνεται η παρουσίαση της πρότασής 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 rdf:resource="#DavidBillington"/>
</rdf:Description>
</rdf:RDF>
```

Η πρώτη γραμμή διευκρινίζει ότι χρησιμοποιούμε XML. Το στοιχείο `rdf:Description` κάνει μια πρόταση για την πηγή `http://www.cit.gu.edu.au/~db`. Μέσα στην περιγραφή η ιδιότητα χρησιμοποιείται σαν ετικέτα, και το περιεχόμενο είναι η τιμή της ιδιότητας.

Οι περιγραφές δίνονται με μια ορισμένη σειρά, με άλλα λόγια η σύνταξη XML επιβάλλει μια αλληλουχία. Η σειρά των περιγραφών (ή των πηγών) δεν είναι

σημαντική σύμφωνα με το αφηρημένο πρότυπο RDF. Αυτό πάλι δείχνει ότι το πρότυπο γραφικών παραστάσεων είναι το πραγματικό πρότυπο δεδομένων RDF και εκείνο για το XML είναι μια πιθανή τμηματική αντιπροσώπευση της γραφικής παράστασης.

3.2.5 Η πρόσβαση του προγραμματιστή (Reification)

Στα RDF είναι δυνατό να δηλώσουμε προτάσεις για τις ίδιες τις προτάσεις, όπως:

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

Αυτό το είδος των προτάσεων μπορεί να χρησιμοποιηθεί για να περιγράψει πίστη ή εμπιστοσύνη σε άλλες προτάσεις, οι οποίες είναι σημαντικές σε μερικά είδη εφαρμογών. Η λύση είναι να ορίσουμε ένα μοναδικό προσδιοριστικό σε κάθε πρόταση, το οποίο μπορεί να χρησιμοποιηθεί για να αναφερθεί στη πρόταση. Το RDF επιτρέπει αυτό χρησιμοποιώντας έναν μηχανισμό reification.

Η βασική ιδέα είναι να εισάγουμε ένα βοηθητικό αντικείμενο, για παράδειγμα, `belief1`, και να το συσχετίσουμε σε κάθε ένα από τα τρία μέρη της αρχικής δήλωσης μέσω των ιδιοτήτων του υποκειμένου, του κατηγορήματος και του αντικειμένου.

Στο προηγούμενο παράδειγμα το υποκείμενο του `belief1` θα ήταν ο David Billington, το κατηγορήμα θα ήταν ο `creator`, και το αντικείμενο το <http://www.cit.gu.edu.au/~db>. Σημειώστε ότι αυτή η προσέγγιση είναι απαραίτητη επειδή υπάρχουν μόνο τριπλές τιμές στα RDF, επομένως δεν μπορούμε να προσθέσουμε άμεσα ένα προσδιοριστικό γιατί έπειτα θα είχε τέσσερις τιμές.

3.2.6 Τύποι Δεδομένων

Θεωρείστε τον αριθμό τηλεφώνου «3875507». Ένα πρόγραμμα που διαβάζει αυτό το πρότυπο δεδομένων RDF δεν μπορεί να ξέρει εάν τα κυριολεκτικά «3875507» πρόκειται να ερμηνευθούν ως ακέραιοι αριθμοί ή ως ένα string, ή πράγματι εάν είναι ένας ακέραιος αριθμός, εάν είναι στη δεκαδική ή δεκαεξαδική αντιπροσώπευση. Ένα πρόγραμμα μπορεί μόνο να ξέρει πώς να ερμηνεύει αυτήν τη πηγή εάν στην εφαρμογή δίνονται ρητά οι πληροφορίες ότι στην κυριολεξία του, προορίζεται να αντιπροσωπεύσει έναν αριθμό, και ποιον αριθμό υποτίθεται πρέπει να αντιπροσωπεύσει.

Η κοινή πρακτική στις γλώσσες προγραμματισμού ή τα συστήματα βάσεων δεδομένων είναι να παρέχουν αυτό το είδος πληροφοριών με την ένωση ενός τύπου δεδομένων με την πραγματική σημασία τους, σε αυτήν την περίπτωση, ένας τύπος δεδομένων όπως το δεκαδικό ή ο ακέραιος αριθμός.

Χρησιμοποιώντας τα προηγούμενα, θα μπορούσαμε να περιγράψουμε την ηλικία του David Billington's ως ένα ακέραιο αριθμό 27 χρησιμοποιώντας τρεις τιμές.

```
(#DavidBillington, http://www.mydomain.org/age,  
"27"^^http://www.w3.org/2001/XMLSchema#integer )
```

Αυτό το παράδειγμα παρουσιάζει δύο πράγματα: τη χρήση της σημείωσης ^^ για να δείξει τον τύπο μιας πραγματικής σημασίας, και τη χρήση των τύπων δεδομένων που προκαθορίζονται από το XML Schema.

Αν θέλαμε να ήμασταν αυστηροί, η χρήση οποιουδήποτε εξωτερικά ορισμένου τύπου δεδομένων scheme, επιτρέπεται στα έγγραφα RDF, αλλά στην πράξη το ευρύτερο χρησιμοποιημένο scheme είναι αυτό από XML schema. Το XML Schema προκαθορίζει μια μεγάλη ποικιλία τύπων δεδομένων, συμπεριλαμβανομένων Booleans, integers, floating-point

numbers, times, and dates.

3.2.7 Μια κρίσιμη άποψη για το RDF

Ήδη έχουμε επισημάνει ότι τα RDF χρησιμοποιούν μόνο δυαδικές ιδιότητες. Αυτός ο περιορισμός φαίνεται να είναι αρκετά σοβαρός επειδή συχνά χρησιμοποιούμε τα κατηγορήματα με περισσότερο από δύο στοιχεία. Ευτυχώς, τέτοια κατηγορήματα μπορούν να εξομοιωθούν από έναν αριθμό δυαδικών κατηγορημάτων. Παρουσιάζουμε αυτήν την τεχνική για έναν referee κατηγορήματος με τρία στοιχεία. Η έννοια του διαιτητή (X, Y, Z) είναι

X is the referee in a chess game between players Y and Z.

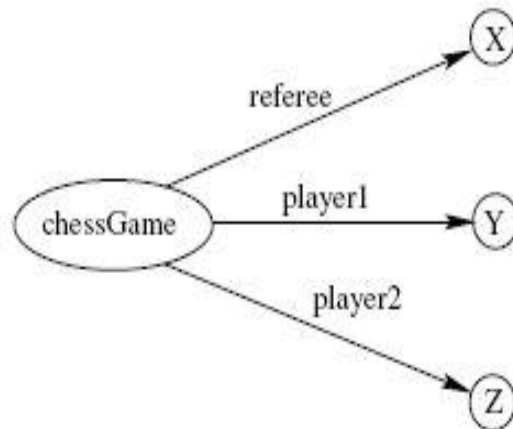
Εισάγουμε τώρα μια νέα βοηθητική πηγή chessGame και το δυαδικό κατηγορήμα το REF, player1, και player2. Κατόπιν μπορούμε να αντιπροσωπεύσουμε το referee (X, Y, Z) ως εξής:

ref (chessGame, X)

player1 (chessGame, Y)

player2(chessGame, Z)

Η γραφική αντιπροσώπευση παρουσιάζεται στο παρακάτω σχήμα. Το πρόβλημα παραμένει ότι το αρχικό κατηγορήμα με τα τρία στοιχεία ήταν απλούστερο και πιο φυσικό.



Ένα άλλο πρόβλημα με τα RDF έχει να κάνει με το χειρισμό των ιδιοτήτων. Όπως προηγούμενα αναφέραμε, οι ιδιότητες είναι πρόσθετα είδη των πηγών. Επομένως, οι ιδιότητες μπορούν να χρησιμοποιηθούν ως αντικείμενο στο πλαίσιο αντικείμενο-ιδιότητα-τιμή. Ενώ αυτή η δυνατότητα προσφέρει ευελιξία, είναι μάλλον ασυνήθιστη για τη διαμόρφωση των γλωσσών, και μπορεί να μπερδεύει αυτούς που γράφουν τις γλώσσες.

Επίσης, ο μηχανισμός reification είναι αρκετά ισχυρός και εμφανίζεται να είναι τοποθετημένος σε λάθος μέρος σε μια απλή γλώσσα όπως η RDF. Η παραγωγή προτάσεων που αφορά τις προτάσεις εισάγει ένα επίπεδο πολυπλοκότητας που δεν είναι απαραίτητο για το βασικό επίπεδο του Semantic Web. Αντ' αυτού, θα ήταν πιο φυσικό να περιληφθεί στα ισχυρότερα επίπεδα, τα οποία παρέχουν πλουσιότερες αντιπροσωπευτικές ικανότητες. Τέλος, η σύνταξη των RDF που είναι βασισμένη στην XML είναι κατάλληλη για την επεξεργασία από τις μηχανές αλλά δεν είναι φιλική προς τον άνθρωπο.

Είναι αλήθεια ότι η RDF έχει ικανοποιητική εκφραστική δύναμη και μάλλον τελικά το Semantic Web δεν θα προγραμματιστεί σε RDF, αλλά μάλλον με τα φιλικά προς το χρήστη εργαλεία που θα μεταφράσουν αυτόματα τις υψηλότερες

αντιπροσωπεύσεις σε RDF. Και δεδομένου ότι είναι πιθανό ότι η RDF θα γίνει πρότυπο, τα οφέλη των δεδομένων σε RDF μπορούν να συγκριθούν με τη σύνταξη των πληροφοριών στην HTML τις πρώτες μέρες του Web.

3.3 RDF: σύνταξη βασισμένη στην XML

Ένα έγγραφο RDF αποτελείται από ένα rdf:RDF element, το περιεχόμενο του οποίου είναι ένας αριθμός από περιγραφές. Παραδείγματος χάριν, εξετάστε την περιοχή των μαθημάτων του πανεπιστημίου και τους ομιλητές στο πανεπιστήμιο του Griffith το έτος 2001.

```
<! DOCTYPE rdf: RDF [
<! 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>Grigoris Antoniou</uni:name>
<uni:title>Professor</uni:title>
</rdf:Description>

<rdf:Description rdf:about="949318">
<uni:name>David Billington</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>David Billington</uni:isTaughtBy>
</rdf:Description>

<rdf:Description rdf:about="CIT1112">
<uni:courseName>Concrete Mathematics</uni:courseName>
<uni:isTaughtBy>Grigoris Antoniou</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>David Billington</uni:isTaughtBy>
</rdf:Description>

<rdf:Description rdf:about="CIT3116">
<uni:courseName>Knowledge Representation</uni:courseName>
<uni:isTaughtBy>Grigoris Antoniou</uni:isTaughtBy>
</rdf:Description>

</rdf:RDF>
```

Κατ' αρχάς, ο μηχανισμός χώρων ονομάτων της XML χρησιμοποιείται εδώ,

αλλά με έναν εκτεταμένο τρόπο. Οι χώροι ονομάτων στην XML χρησιμοποιούνται μόνο για λόγους αποσαφήνισης. Στα RDF οι εξωτερικοί χώροι ονομάτων αναμένονται για να είναι RDF έγγραφα που καθορίζουν τις πηγές οι οποίοι μετά χρησιμοποιούνται έπειτα για να εισαχθούν στο RDF έγγραφο. Αυτός ο μηχανισμός επιτρέπει την επαναχρησιμοποίηση των πηγών από άλλους ανθρώπους που μπορούν να αποφασίσουν να εισάγουν πρόσθετα χαρακτηριστικά γνωρίσματα σε αυτές τις πηγές. Το αποτέλεσμα είναι η εμφάνιση της μεγάλης, διανεμημένης γνώσης.

Δεύτερον, η ιδιότητα `rdf:about` του στοιχείου `rdf:Description` είναι, για να κυριολεκτήσουμε, ισοδύναμο κατά την έννοια αυτή, μιας ιδιότητας ταυτότητας, αλλά συχνά χρησιμοποιείται για να προτείνει ότι το αντικείμενο για το οποίο μια πρόταση δημιουργείται «έχει ήδη δηλωθεί» αλλού. Τυπικά μιλώντας, ένα σύνολο προτάσεων RDF απλά διαμορφώνει μια μεγάλη γραφική παράσταση, συσχετίζοντας πράγματα με άλλα πράγματα μέσω των ιδιοτήτων, και δεν υπάρχει κανένα σημείο όπου «καθορίζουμε» ένα αντικείμενο σε μια θέση και αναφερόμαστε σε αυτό κάπου αλλού.

Εντούτοις, στη σύνταξη XML, είναι μερικές φορές χρήσιμο να προτείνουμε ότι μια θέση στην αλληλουχία της XML να είναι ο «καθορισμός» της θέσης, ενώ άλλες θέσεις να δηλώνουν «πρόσθετες» ιδιότητες για ένα αντικείμενο που «έχει καθοριστεί» αλλού. Στην πραγματικότητα το προηγούμενο παράδειγμα είναι ελαφρώς παραπλανητικό.

Εάν θελήσουμε να είμαστε απολύτως σωστοί, πρέπει να αντικαταστήσουμε όλα τα γεγονότα και τα Ids του προσωπικού, όπως το 949352 και το CIT3112, με αναφορές σε εξωτερικούς χώρους ονομάτων, παραδείγματος χάριν

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

Εμείς το κάναμε αυτό για να βελτιώσουμε έτσι την αναγνωσιμότητα του αρχικού παραδείγματός μας επειδή ενδιαφερόμαστε πρώτιστα για τις βασικές ιδέες του RDF. Όμως, οι αναγνώστες πρέπει να γνωρίζουν ότι αυτό θα ήταν ο ακριβής τρόπος για ένα σωστό έγγραφο RDF.

Το περιεχόμενο των στοιχείων του `rdf:Description` καλούνται στοιχεία ιδιοτήτων. Παραδείγματος χάριν, στην περιγραφή

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

τα δύο στοιχεία `uni:courseName` και `uni:isTaughtBy` καθορίζουν ζευγάρια ιδιοκτησία-τιμών για το CIT3116. Η προηγούμενη περιγραφή αντιστοιχεί σε δύο προτάσεις RDF.

Τρίτον, η ιδιότητα `rdf:datatype="&xsd;integer"` χρησιμοποιείται για να δείξει το τύπο δεδομένων της τιμής της ιδιότητας που αναφέρεται στην ηλικία. Ακόμα κι αν η ιδιότητα της ηλικίας έχει καθοριστεί για να έχει "`&xsd;integer`" σαν εύρος, ακόμα υπάρχει η απαίτηση για να δείξει τον τύπο της τιμής από την ιδιότητα κάθε φορά που χρησιμοποιείται.

Αυτό πρόκειται να εξασφαλίσει ότι ένας επεξεργαστής RDF μπορεί να ορίσει το σωστό τύπο της τιμής μιας ιδιότητας ακόμα κι αν δεν έχει δει τον αντίστοιχο ορισμό του RDF Schema.

Τέλος, τα στοιχεία ιδιοτήτων μιας περιγραφής πρέπει να διαβαστούν συνδεδεμένα μεταξύ τους. Στο προηγούμενο παράδειγμα, το υποκείμενο ονομάζεται «Knowledge Representation» και διδάσκεται από τον Grigoris Antoniou.

3.3.1 Η Ιδιότητα `rdf:resource`

Το προηγούμενο παράδειγμα δεν ήταν ικανοποιητικό από την άποψη ότι: οι σχέσεις μεταξύ των μαθημάτων (`courses`) και των ομιλητών (`lecturers`) δεν ορίστηκε τυπικά αλλά υπήρξε σαν αυτονόητο μέσω της χρήσης του ίδιου ονόματος. Σε μια μηχανή, η χρήση του ίδιου ονόματος μπορεί να είναι μια σύμπτωση, παραδείγματος χάριν, ο David Billington που διδάσκει CIT3112 μπορεί να μην είναι το ίδιο πρόσωπο, με το πρόσωπο με την ταυτότητα 949318 που συμβαίνει να ονομάζεται David Billington.

Αντ' αυτού χρειαζόμαστε μια επίσημη προδιαγραφή του γεγονότος αυτού, παραδείγματος χάριν, ο δάσκαλος CIT1111 είναι μέλος προσωπικού με τον αριθμό 949318, του οποίου το όνομα είναι David Billington. Μπορούμε να επιτύχουμε αυτο 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>David Billington</uni:name>
  <uni:title>Associate Professor</uni:title>
</rdf:Description>
```

Σημειώνουμε ότι σε περίπτωση που είχαμε καθορίσει την πηγή του μέλους προσωπικού με τον αριθμό ταυτότητας 939318 στο έγγραφο RDF, που χρησιμοποιώντας τις ιδιότητες της ταυτότητας αντί της ιδιότητας *about*, θα έπρεπε να χρησιμοποιήσουμε 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>David Billington</uni:name>
  <uni:title>Associate Professor</uni:title>
</rdf:Description>
```

Το ίδιο πράγμα ισχύει για τις εξωτερικά ορισμένες πηγές: για παράδειγμα, αναφερόμαστε στις εξωτερικά ορισμένες πηγές CIT1111 χρησιμοποιώντας

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

σαν τιμή του rdf:about, όπου το www.mydomain.org/uni-ns/ είναι το URI, όπου εκεί βρίσκεται το CIT1111. Με άλλα λόγια, μια περιγραφή με μια ταυτότητα (ID) καθορίζει ένα κομμάτι URI, το οποίο μπορεί να χρησιμοποιηθεί για να παραπέμψει την καθορισμένη περιγραφή.

3.3.2 Φωλιασμένες Περιγραφές

Οι περιγραφές μπορούν να οριστούν μέσα σε άλλες περιγραφές. Για παράδειγμα, μπορούμε να αντικαταστήσουμε τις περιγραφές του προηγούμενου παραδείγματος με τις ακόλουθες, φωλιασμένες περιγραφές:


```

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

```

Άλλες σειρές μαθημάτων, όπως το CIT3112, μπορούν ακόμα να αναφερθούν στη νέα πηγή 949318. Με άλλα λόγια, αν και μια περιγραφή μπορεί να καθοριστεί μέσα σε μια άλλη περιγραφή, το πεδίο του είναι γενικότερο.

3.3.3 Το στοιχείο `rdf:type`

Στα παραδείγματά μας μέχρι τώρα, οι περιγραφές εμπίπτουν σε δύο κατηγορίες: `courses` και `lecturers`. Αυτό το γεγονός είναι σαφές στους ανθρώπους, αλλά δεν έχει δηλωθεί πουθενά, έτσι δεν είναι προσιτό για τις μηχανές. Στην RDF είναι δυνατό να γίνουν τέτοιες προτάσεις χρησιμοποιώντας το στοιχείο `rdf:type`. Παρακάτω υπάρχουν 2 περιγραφές που περιλαμβάνουν τέτοιες πληροφορίες:

```

<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>David Billington</uni:name>
<uni:title>Associate Professor</uni:title>
</rdf:Description>

```

Σημειώστε ότι το `rdf:type` μας επιτρέπει να εισάγουμε κάποια δομή στο έγγραφο RDF.

3.3.4 Ο μηχανισμός Reification

Μερικές φορές επιθυμούμε να κάνουμε προτάσεις για κάποιες άλλες προτάσεις. Για να το κάνουμε αυτό πρέπει να είμαστε σε θέση να αναφερθούμε σε μια πρόταση χρησιμοποιώντας ένα προσδιοριστικό. Η RDF μας επιτρέπει τέτοια αναφορά μέσω ενός μηχανισμού reification, ο οποίος μετατρέπει την πρόταση σε μια πηγή. Για παράδειγμα η περιγραφή:

```

<rdf:Description rdf:about="949352">
  <uni:name>Grigoris Antoniou</uni:name>
</rdf:Description>

```

μετατρέπεται σε:

```

<rdf:Statement rdf:about="StatementAbout949352">
  <rdf:subject rdf:resource="949352"/>
  <rdf:predicate rdf:resource="&uni;name"/>
  <rdf:object>Grigoris Antoniou</rdf:object>
</rdf:Statement>

```

Σημειώστε ότι τα `rdf:subject`, `rdf:predicate`, και `rdf:object` μας επιτρέπουν να έχουμε πρόσβαση στα μέρη μιας πρότασης. Μπορούμε να αναφερθούμε στην

πρόταση χρησιμοποιώντας την ταυτότητά της (ID), όπως με κάθε άλλη περιγραφή. Εάν περισσότερες από μια ιδιότητες περιλαμβάνονται σε ένα στοιχείο περιγραφής (Description Element), τα στοιχεία αντιστοιχούν σε περισσότερες από μια προτάσεις.

3.4 RDF Schema: Βασικές ιδέες

Η RDF είναι μια παγκόσμια γλώσσα που αφήνει τους χρήστες να περιγράψουν πηγές χρησιμοποιώντας το δικό τους λεξιλόγιο. Η RDF δεν κάνει υποθέσεις για οποιαδήποτε περιοχή (domain) μιας εφαρμογής, ούτε προσδιορίζει τη σημασιολογία κάθε περιοχής. Είναι επάνω στην ευχέρεια του χρήστη για να το κάνει χρησιμοποιώντας το RDF Schema (RDFS).

3.4.1 Κλάσεις και Ιδιότητες

Θα αναφερθούμε ξανά στο πανεπιστήμιο του Griffith. Πρώτα πρέπει να διευκρινίσουμε τα «πράγματα» για τα οποία θέλουμε να μιλήσουμε, όπως συγκεκριμένοι ομιλητές π.χ. David Billington, και συγκεκριμένα μαθήματα, όπως τα διακριτά μαθηματικά. Αλλά, θέλουμε επίσης να μιλήσουμε για τα μαθήματα, μαθήματα πρώτου έτους, ομιλητές, καθηγητές, και ούτω καθεξής. Στην πρώτη περίπτωση μιλάμε για τα μεμονωμένα αντικείμενα (πηγές), στη δεύτερη μιλάμε για τις κλάσεις που καθορίζουν τους τύπους των αντικειμένων.

Μια κλάση μπορεί να θεωρηθεί ως ένα σύνολο στοιχείων. Μεμονωμένα αντικείμενα που ανήκουν σε μια κλάση αναφέρονται σαν αναφορά εκείνης της κλάσης. Έχουμε ήδη καθορίσει τη σχέση μεταξύ των αναφορών και των κλάσεων στην RDF χρησιμοποιώντας το στοιχείο `rdf:type`.

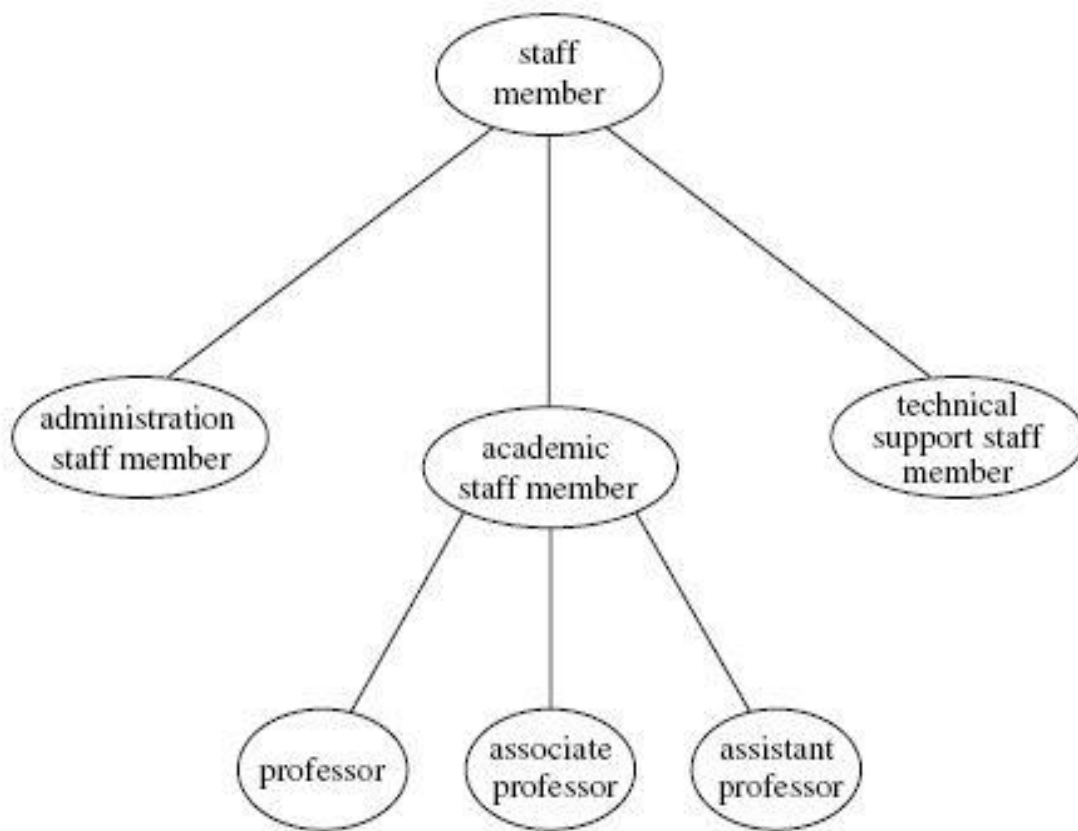
Μια σημαντική χρήση των κλάσεων είναι να επιβάλουν τους περιορισμούς στις προτάσεις σε ένα έγγραφο RDF χρησιμοποιώντας το schema.

3.4.2 Ιεραρχίες κλάσεων και Κληρονομικότητα.

Εφόσον έχουμε κλάσεις θα επιθυμούσαμε επίσης να ορίσουμε και σχέσεις μεταξύ τους. Παραδείγματος χάριν, υποθέστε ότι έχουμε τις παρακάτω κλάσεις:

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

Αυτές οι κλάσεις δεν είναι ανεξάρτητες η μια από την άλλη. Για παράδειγμα, κάθε καθηγητής είναι ακαδημαϊκό μέλος προσωπικού (every professor is an academic staff member). Λέμε ότι ο “professor” είναι υποκλάση του “academic staff member”, ή ισοδύναμα ότι το “academic staff member” είναι υπερκλάση του “professor”. Η σχέση των υποκλάσεων καθορίζει μια ιεραρχία από κλάσεις, όπως φαίνεται στο παρακάτω σχήμα. Γενικά, το A είναι μια υποκλάση του B εάν κάθε αναφορά στο A είναι επίσης μια αναφορά στο B.



Μια ιεραρχική οργάνωση των κλάσεων έχει μια πολύ σημαντική πρακτική σημασία. Προσέξτε τον παρακάτω περιορισμό:

Courses must be taught by academic staff members only.

Υποθέστε ότι ο Michael Maher ορίστηκε ως ένας καθηγητής. Έτσι, σύμφωνα με τον προηγούμενο περιορισμό, δεν έχει την άδεια για να διδάξει μαθήματα. Ο λόγος είναι ότι δεν υπάρχει καμία πρόταση που να διευκρινίζει ότι ο Michael Maher είναι επίσης academic staff member. Δε θα ήταν σωστό να ξεπεράσουμε αυτή τη δυσκολία προσθέτοντας αυτήν την πρόταση στην περιγραφή μας. Αντ' αυτού θα θέλαμε ο Michael Maher να κληρονομεί τη δυνατότητα να διδάξει από την κλάση academic staff members. Αυτό ακριβώς γίνεται στο RDF

Schema.

Οι κλάσεις, η κληρονομικότητα, και οι ιδιότητες είναι, φυσικά, γνωστές σε άλλους τομείς των υπολογιστών, όπως στον αντικειμενοστραφή προγραμματισμό. Αλλά ενώ οι ομοιότητες είναι πολλές, υπάρχουν και πολλές διαφορές. Στον αντικειμενοστραφή προγραμματισμό, ένα αντικείμενο μιας κλάσης καθορίζει τις ιδιότητες που ισχύουν σε αυτό. Για να προσθέσουμε νέες ιδιότητες σε μια κλάση σημαίνει ότι πρέπει να τροποποιήσουμε την κλάση.

Όμως, στα RDFS, οι ιδιότητες καθορίζονται συνολικά, που σημαίνει ότι δεν είναι ενθυλακωμένες όπως οι ιδιότητες στις κλάσεις. Είναι δυνατό να καθορίσουμε νέες ιδιότητες που ισχύουν για μια υπάρχουσα κλάση χωρίς να κάνουμε αλλαγές για αυτή τη κλάση.

Από τη μια πλευρά αυτό είναι ένας ισχυρός μηχανισμός με τις εξής συνέπειες: μπορούμε να χρησιμοποιήσουμε τις κλάσεις που έχουν οριστεί από άλλους και να τις προσαρμόσουμε στις απαιτήσεις μας μέσω νέων ιδιοτήτων. Από την άλλη πλευρά αυτός ο χειρισμός των ιδιοτήτων παρεκκλίνει από την τυποποιημένη προσέγγιση που έχει προκύψει στον τομέα του αντικειμενοστραφούς προγραμματισμού.

4 *Οντολογίες (Web Ontology Language)*

4.1 *Εισαγωγή*

Η εκφραστικότητα του RDF και του RDF schema τα οποία περιγράψαμε στο προηγούμενο κεφάλαιο ήταν σκόπιμα περιορισμένη. Το RDF περιορίζεται σε δυαδικά κατηγορήματα ενώ το RDF schema περιορίζεται σε μία ιεραρχία ιδιοτήτων.

Παρολ' αυτά η ομάδα του W3C που ασχολείται με τις οντολογίες στον παγκόσμιο ιστό αναγνωρίζει μια σειρά από χαρακτηριστικά τα οποία μπορούν να χρησιμοποιηθούν στον Σημασιολογικό Ιστό (Semantic Web) και θα παρέχουν περισσότερες πληροφορίες από το RDF και RDFschema. Επιστημονικές ομάδες τόσο στις Ηνωμένες Πολιτείες όσο και στην Ευρώπη έχουν ήδη αναγνωρίσει την αναγκαιότητα μίας νέας γλώσσας οντολογίας. Αυτό έχει ως αποτέλεσμα τον καθορισμό μίας καινούριας γλώσσας η οποία ονομάζεται DAML+OIL.

Η γλώσσα αυτή αποτέλεσε την αφετηρία στην δημιουργία από τον οργανισμό W3C της OWL, μιας γλώσσας η οποία έγινε καθολικά αποδεκτή ως η γλώσσα που χρησιμοποιείται στον Σημασιολογικό Ιστό.

Σε αυτό το κεφάλαιο θα περιγράψουμε αρχικά το κίνητρο για την δημιουργία της OWL, τις απαιτήσεις της, καθώς και την σχέση της με το RDF schema. Έπειτα περιγράφουμε τα στοιχεία (elements) της γλώσσας OWL .

4.2 *OWL και RDF/RDFS*

Απαιτήσεις για τις γλώσσες οντολογίας

Οι γλώσσες οντολογίας επιτρέπουν στους χρήστες να γράφουν ρητά και επίσημα πρότυπα. Οι κύριες απαιτήσεις της OWL είναι η καλά καθορισμένη σύνταξη, αποδοτική υποστήριξη συλλογισμού, επίσημη σημασιολογία, ικανοποιητική εκφραστική δύναμη και ευκολία στην έκφραση.

Η σημασία της σωστής σύνταξης μιας γλώσσας είναι γνωστή από διάφορες γλώσσες προγραμματισμού και είναι απαραίτητη συνθήκη για την επεξεργασία της από τις μηχανές. Όλες οι γλώσσες που έχουμε παρουσιάσει μέχρι τώρα έχουν παρόμοια σύνταξη. DAML+OIL και η OWL είναι χτισμένες πάνω στη RDF και στα RDFS.

Φυσικά είναι αμφισβητήσιμο εάν η RDF που είναι βασισμένη στην XML είναι φιλική προς τον χρήστη και γι αυτό υπάρχουν εναλλακτικές λύσεις που ταιριάζουν περισσότερο στους χρήστες(παραδείγματος χάριν η σύνταξη της OWL). Εντούτοις αυτό το μειονέκτημα δεν είναι πολύ σημαντικό μιας και οι χρήστες θα αναπτύσσουν τις οντολογίες τους χρησιμοποιώντας εργαλεία δημιουργίας ή γενικότερα εργαλεία ανάπτυξης οντολογίας παρά θα γράφουν κατευθείαν μια οντολογία σε DAML+OIL ή σε OWL.

Μια επίσημη σημασιολογία(formal semantics) περιγράφει την έννοια της γνώσης ακριβώς. Με τον όρο ακριβώς εννοούμε ότι η σημασιολογία δεν αναφέρεται στις υποκειμενικές διαισθήσεις ούτε είναι ανοικτή σε διαφορετικές ερμηνείες από διαφορετικούς ανθρώπους.

Μια χρησιμότητα της επίσημης σημασιολογίας(formal semantics) είναι στο να επιτρέπει τους ανθρώπους να έχουν λόγο στην γνώση. Για την γνώση όσον αναφορά την οντολογία μπορούμε να έχουμε λόγο στα παρακάτω :

- Μέλος κλάσης(Class Membership). Εάν το x είναι μια αναφορά της κλάσης C και η C είναι υποκλάση της D τότε μπορούμε να αναφερόμαστε στο x ως μια αναφορά του D .

- Ισότητα των κλάσεων (Equivalence of classes). Εάν η κλάση A είναι ίση με μία κλάση B και η B είναι ίση με την C τότε και η A είναι ίση με την C .
- Συνέπεια. Ας υποθέσουμε ότι έχουμε δηλώσει μια αναφορά x της κλάσης A και ότι η A είναι υποκλάση του αποτελέσματος $B \cap C$, η A είναι υποκλάση της D . Μετά έχουμε ασυνέπεια μιας και η A θα έπρεπε να είναι άδεια αλλά περιέχει μια αναφορά x .
- Ταξινόμηση. Εάν έχουμε δηλώσει ορισμένα ζευγάρια ιδιοτήτων ως συνθήκη για ένα μέλος σε μια κλάση A , ύστερα εάν υπάρχει x που ικανοποιεί αυτές τις συνθήκες, το x πρέπει να είναι αναφορά της A .

Η σημασιολογία είναι μια προϋπόθεση για την υποστήριξη του συλλογισμού. Ο συλλογισμός είναι σημαντικός επειδή επιτρέπει :

- τον έλεγχο της συνέπειας της οντολογίας και της γνώσης
- τον έλεγχο για αναπάντεχες σχέσεις μεταξύ των κλάσεων
- αυτόματη ταξινόμηση στις αναφορές των κλάσεων.

Όρια της εκφραστικής δύναμης του RDF schema

Η RDF και το RDFS επιτρέπουν την παρουσίαση κάποιων στοιχείων οντολογίας.

Παρακάτω παραθέτουμε ποιιά στοιχεία δεν μπορούν να αναπαρασταθούν :

- Η εμβέλεια των ιδιοτήτων. `rdfs:range` καθορίζει το εύρος μιας ιδιότητας για όλες τις κλάσεις. Αντίθετα στο RDF schema δεν μπορούμε να καθορίσουμε περιορισμούς ως προς το εύρος για ορισμένες κλάσεις. Για παράδειγμα δεν μπορούμε να πούμε ότι οι αγελάδες τρώνε μόνο φυτά ενώ άλλα ζώα μπορούν να φάνε κρέας.
- Η μη ένωση των κλάσεων. Μερικές φορές ευχόμαστε να πούμε ότι οι κλάσεις δεν μπορούν να ενωθούν. Για παράδειγμα, η κλάση `male` και `female`

δεν μπορούν να ενωθούν. Αντίθετα στο RDF schema μπορούμε να δούμε σχέσεις υποκλάσεων.

- Συνδυασμός των κλάσεων. Μερικές φορές θέλουμε να δημιουργήσουμε καινούριες κλάσεις οι οποίες θα προκύπτουν από τον συνδυασμό άλλων κλάσεων χρησιμοποιώντας την ένωση, την τομή και το συμπλήρωμα. Για παράδειγμα μπορούμε να καθορίσουμε ότι η κλάση person θα είναι η ένωση των κλάσεων male και female. Στο RDF schema δεν μπορούμε να προβούμε σε τέτοιους καθορισμούς.
- Περιορισμοί. Μερικές φορές θέλουμε να βάλουμε περιορισμούς σχετικά με τις τιμές που μπορεί να πάρει μια ιδιότητα. Για παράδειγμα όταν θέλουμε να πούμε πως ένα άτομο(person) έχει μόνο δύο γονείς. Στο RDF schema κάτι τέτοιο δεν μπορεί να δηλωθεί.
- Χαρακτηριστικά ιδιοτήτων. Πολλές φορές λέμε ότι μία ιδιότητα είναι μεταβατική(transitive), μοναδική(unique), ή αντίστροφη(inverse) ως προς μία άλλη ιδιότητα.

Επομένως χρειαζόμαστε μια γλώσσα οντολογίας πιο πλούσια από το RDF schema, μια γλώσσα η οποία θα προσφέρει περισσότερα στοιχεία. Δηλαδή μια γλώσσα η οποία θα υποστηρίζει μεγάλες κλάσεις οντολογίας και γνώσης.

4.3 Συμβατότητα OWL και RDF/RDFS

Η OWL θα είναι μια επέκταση του RDF schema χρησιμοποιώντας την σημασία των κλάσεων και των ιδιοτήτων της RDF αλλά θα προστεθούν νέα χαρακτηριστικά που θα εμπλουτίζουν την γλωσσά με μεγαλύτερη εκφραστικότητα.

Τρεις υποκατηγορίες της OWL

- OWL Full

Ολόκληρη η γλώσσα OWL ονομάζεται OWL full και χρησιμοποιεί όλες τις δυνατότητες της. Επίσης επιτρέπει τον συνδυασμό των χαρακτηριστικών της με αυτού της RDF και του RDF schema. Αυτό περιλαμβάνει την δυνατότητα της αλλαγής των προκαθορισμένων χαρακτηριστικών.

Το πλεονέκτημα της OWL full είναι ότι είναι πλήρως συμβατή με την RDF τόσο σε επίπεδο σύνταξης όσο και σε επίπεδο σημασιολογίας. Κάθε έγγραφο RDF/RDF schema είναι πλήρες συμβατό με την OWL full. Το μειονέκτημα της OWL Full είναι ότι η γλώσσα έχει γίνει τόσο ισχυρή με αποτέλεσμα να είναι δύσκολη η υποστήριξη τους πλήρους συλλογισμού.

- OWL DL

Προκειμένου να επεκταθεί η υπολογιστική αποδοτικότητα, η OWL DL (Description Logic) είναι μια υπογλώσσα της OWL Full που περιορίζει πως οι δομητές(constructors) από την OWL και την RDF μπορούν να χρησιμοποιηθούν. Ουσιαστικά οι δομητές της OWL απαγορεύουν ο ένας τον άλλον εξασφαλίζοντας ότι η γλώσσα είναι μέρος μιας λογικής (description logic).

Το πλεονέκτημα αυτού είναι ότι επιτρέπει την αποδοτική υποστήριξη του συλλογισμού. Το μειονέκτημα είναι ότι χάνουμε την πλήρη συμβατότητα με την RDF. Ένα RDF έγγραφο θα πρέπει γενικά να επεκτείνεται με κάποιους τρόπους και με κάποιους άλλους να περιορίζεται πριν γίνει ένα OWL DL έγγραφο. Κάθε OWL DL έγγραφο είναι και RDF έγγραφο.

- OWL Lite

Ένας επιπλέον περιορισμός της γλώσσας OWL. Το πλεονέκτημα αυτής της γλώσσας είναι ότι μπορεί πιο εύκολα να χρησιμοποιηθεί από τους χρήστες και να εφαρμοστεί από τους μηχανικούς λογισμικού. Το μειονέκτημά της είναι ότι δεν παρέχει μεγάλη εκφραστικότητα.

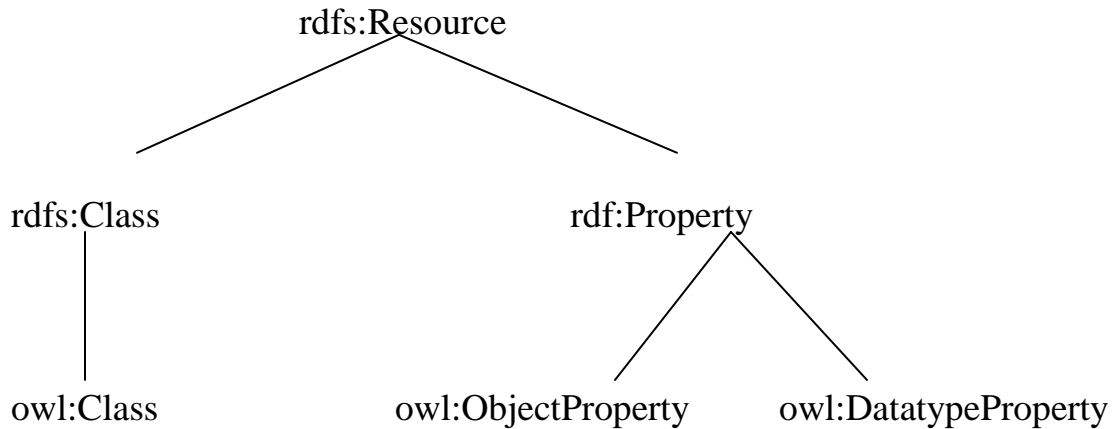
Οι δημιουργοί οντολογιών που υιοθετούν την OWL πρέπει να εξετάσουν ποιό υποσύνολο της γλώσσας ανταποκρίνεται στις ανάγκες τους. Η επιλογή μεταξύ της OWL DL και της OWL Full κυρίως εξαρτάται από τον βαθμό στον οποίο οι χρήστες απαιτούν μεταγενέστερες εγκαταστάσεις του RDF schema. Όταν χρησιμοποιούν την OWL Full σε σύγκριση με την OWL DL η υποστήριξη συλλογισμού είναι λιγότερο προβλέψιμη επειδή οι πλήρεις εφαρμογές της OWL Full θα είναι αδύνατες.

Υπάρχουν ακριβείς έννοιες για την συμβατότητα μεταξύ των τριών γλωσσών στις οποίες χωρίζεται η OWL :

- Κάθε OWL Lite οντολογία είναι και OWL DL οντολογία.
- Κάθε OWL DL οντολογία είναι και OWL Full οντολογία.
- Κάθε έγκυρο OWL Lite συμπέρασμα είναι και ένα έγκυρο OWL DL .
- Κάθε έγκυρο OWL DL συμπέρασμα είναι και ένα έγκυρο OWL Full .

Η OWL ακόμα χρησιμοποιεί την RDF και το RDF schema σε πολλές περιπτώσεις:

- Όλες οι υποκατηγορίες της OWL χρησιμοποιούν την RDF για την σύνταξη τους.
- Οι αναφορές δηλώνονται όπως στην RDF χρησιμοποιώντας περιγραφές και τρόπους αναπαραστάσεις των πληροφοριών της RDF.



Σχήμα : Οι σχέσεις των υποκλάσεων μεταξύ της OWL και RDF/RDFS

- Οι δομητές της OWL όπως `owl:Class`, `owl:DatatypeProperty` και `owl:ObjectProperty` είναι ειδικεύσεις των αντίστοιχων RDF τους.

4.4 Περιγραφή της γλώσσας OWL

4.4.1 Σύνταξη

Η OWL στηρίζεται στο RDF schema και η RDF χρησιμοποιεί την XML. Όμως το RDF/XML δεν παρέχει μια ευανάγνωστη σύνταξη και γι αυτό το λόγο στην OWL έχουν καθοριστεί επιπλέον συντακτικές φόρμες :

- Μια σύνταξη βασισμένη στην XML η οποία δεν ακολουθεί τις RDF συμβάσεις με αποτέλεσμα να διαβάζεται ευκολότερα από τους χρήστες
- Όπου η σύνταξη είναι συμπαγέστερη και ευανάγνωστη.
- Μια γραφική σύνταξη βασισμένη στις συντάξεις UML το οποίο χρησιμοποιείται ευρέως και γι αυτό είναι ένας εύκολος τρόπος για τους χρήστες να εξοικειωθούν με την OWL.

4.4.2 Επικεφαλίδα(header)

Τα έγγραφα OWL συνήθως αποκαλούνται OWL οντολογίες και είναι RDF έγγραφα. Το στοιχείο ρίζα μια OWL οντολογίας είναι ένα rdf:RDF στοιχείο :

```
<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#">
```

Μια OWL οντολογία μπορεί να ξεκινά με μια συλλογή ισχυρισμών. Αυτοί οι ισχυρισμοί ομαδοποιούνται κάτω από ένα στοιχείο owl:Ontology το οποίο περιέχει σχόλια, την έκδοση και συνυπολογίζει 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 ένας από αυτούς τους ισχυρισμούς έχει συνέπειες για τη λογική έννοια της οντολογίας owl:imports το οποίο περιέχει άλλες οντολογίες που το περιεχόμενο τους ήταν μέρος αυτής της οντολογίας. Σημειώστε ότι τα namespaces(χώρος ονομάτων) χρησιμοποιούνται για την αποσαφήνιση, οι εισαγόμενες οντολογίες παρέχουν ορισμούς που μπορούν να χρησιμοποιηθούν. Συνήθως υπάρχει ένα εισαγόμενο στοιχείο για κάθε namespace(χώρο ονομάτων)

που χρησιμοποιείται αλλά είναι πιθανό να εισαχθούν επιπρόσθετες οντολογίες, για παράδειγμα οντολογίες οι οποίες παρέχουν ορισμούς χωρίς να εισάγονται νέα ονόματα.

Επίσης το `owl:imports` είναι μια μεταβατική ιδιότητα : εάν μια οντολογία A εισάγει μια οντολογία B και μια οντολογία B εισάγει μια οντολογία C τότε και η οντολογία A εισάγει την οντολογία C.

4.4.3 Στοιχεία κλάσεων(*Class elements*)

Οι κλάσεις καθορίζονται χρησιμοποιώντας το στοιχείο `owl:Class`. Για παράδειγμα μπορούμε να καθορίσουμε μια κλάση `associateProfessor` :

```
<owl:Class rdf:ID="associateProfessor">
  <rdfs:subClassOf rdf:resource="#academicStaffMember"/>
</owl:Class>
```

Μπορούμε επίσης να πούμε ότι αυτή η κλάση χωρίζει τις κλάσεις `assistantProfessor` και `Professor` που χρησιμοποιούν τα στοιχεία `owl:disjointWith`. Αυτά τα στοιχεία μπορούν να περιληφθούν στο προηγούμενο καθορισμό ή να προστεθούν 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 ισότητα των κλάσεων μπορεί να οριστεί χρησιμοποιώντας το στοιχείο `owl:equivalentClass`:

```
<owl:Class rdf:ID="faculty">
  <owl:equivalentClass rdf:resource="#academicStaffMember"/>
</owl:Class>
```

Τελικά υπάρχουν δύο προκαθορισμένες κλάσεις, owl:Thing και owl:Nothing. Η πρώτη είναι μια γενική κλάση η οποία περιέχει τα πάντα ενώ η δεύτερη είναι μια άδεια κλάση. Θα πρέπει όμως να γνωρίζουμε ότι κάθε κλάση είναι μια υποκλάση της owl:Thing και υπερκλάση της owl:Nothing.

4.4.4 Ιδιότητες στοιχείων(property elements)

Στην OWL υπάρχουν δύο είδη ιδιοτήτων :

- Ιδιότητες αντικειμένου, οι οποίες είναι σχετικές με άλλα αντικείμενα. Χαρακτηριστικά παραδείγματα είναι το isTaughtBy και supervises.
- Ιδιότητες τύπων δεδομένων, οι οποίες συσχετίζουν αντικείμενα με τιμές τύπων δεδομένων. Για παράδειγμα οι phone, title και age. Η OWL δεν έχει κάποιους προκαθορισμένους τύπους δεδομένων ούτε παρέχει ειδικές δυνατότητες. Αντιθέτως επιτρέπει να χρησιμοποιούνται τύποι δεδομένων του XML schema και γι αυτό γίνονται μέρος του Σημασιολογικού Ιστού(Semantic Web).

Παρακάτω βλέπουμε ένα παράδειγμα ενός τύπου μια ιδιότητας :

```
<owl:DatatypeProperty rdf:ID="age">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema
  #nonNegativeInteger"/>
</owl:DatatypeProperty>
```


Οι τύποι δεδομένων που καθορίζονται από τους χρήστες συχνά συλλέγονται στο XML schema και μετά χρησιμοποιούνται σε μια OWL οντολογία.

Ας δούμε ένα παράδειγμα από μια ιδιότητα ενός αντικειμένου :

```
<owl:ObjectProperty rdf:ID="isTaughtBy">
  <rdfs:domain rdf:resource="#course"/>
  <rdfs:range rdf:resource="#academicStaffMember"/>
  <rdfs:subPropertyOf rdf:resource="#involves"/>
</owl:ObjectProperty>
```

Η OWL επιτρέπει σε μας να συσχετίσουμε ιδιότητες. Ένα τυπικό παράδειγμα είναι το isTaughtBy και το teaches:

```
<owl:ObjectProperty rdf:ID="teaches">
  <rdfs:range rdf:resource="#course"/>
  <rdfs:domain rdf:resource="#academicStaffMember"/>
  <owl:inverseOf rdf:resource="#isTaughtBy"/>
</owl:ObjectProperty>
```

4.4.5 Περιορισμοί ιδιοτήτων

Με την ιδιότητα rdfs:subClassOf μπορούμε να καθορίσουμε μια κλάση C να είναι υποκλάση μια άλλης κλάσης έστω C1. Μετά από αυτόν τον καθορισμό κάθε αναφορά στην C αποτελεί και μια αναφορά στην C1.

Ας υποθέσουμε ότι θέλουμε να δηλώσουμε ότι η κλάση C ικανοποιεί κάποιες συνθήκες οι οποίες ικανοποιούνται και από όλες τις αναφορές της C. Αυτό είναι ισοδύναμο με το να πούμε ότι η C είναι υποκλάση της C1 όπου η C1 συλλέγει όλα τα αντικείμενα τα οποία ικανοποιούν αυτές τις συνθήκες. Αυτό είναι

ακριβώς που γίνεται στην OWL. Σημειώστε εδώ ότι η C1 μπορεί να μείνει ανώνυμη.

Το ακόλουθο στοιχείο απαιτεί τα μαθήματα των πρώτων χρόνων να γίνονται από πανεπιστημιακούς 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>
```

Owl:allValuesFrom χρησιμοποιείται για να καθορίσει την κλάση των πιθανών τιμών της ιδιότητας που ορίζεται από την ιδιότητα owl:onProperty και τις τιμές τις οποίες μπορεί να πάρει (με άλλα λόγια όλες οι τιμές της ιδιότητας πρέπει να προέρχονται από αυτήν την κλάση). Στο παράδειγμά μας μόνο οι πανεπιστημιακοί καθηγητές επιτρέπεται ως τιμή για την ιδιότητα isTaughtBy.

Μπορούμε να δηλώσουμε ότι τα μαθηματικά διδάσκονται από τον David Billington :

```
<owl:Class rdf:about="#mathCourse">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#isTaughtBy"/>
      <owl:hasValue rdf:resource="#949318"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Owl:hasValue δηλώνει μία συγκεκριμένη τιμή την οποία πρέπει να έχει η ιδιότητα και καθορίζεται από το owl:onProperty.

Επίσης μπορούμε να δηλώσουμε ότι όλα τα ακαδημαϊκά μέλη πρέπει να είναι καθηγητές τουλάχιστον σε ένα προπτυχιακό μάθημα :

```
<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 την owl:allValuesFrom και την owl:someValuesFrom. Το παράδειγμα που χρησιμοποιείται απαιτεί κάθε πρόσωπο για να διδάσκει, να είναι καθηγητής. Από λογική άποψη έχουμε έναν καθολικό προσδιορισμό της ποσότητας (universal quantification).

Το τελευταίο παράδειγμα απαιτεί να υπάρχει τουλάχιστον ένα προπτυχιακό μάθημα το οποίο διδάσκεται από ένα ακαδημαϊκό μέλος του προσωπικού. Είναι ακόμα δυνατό ο ίδιος ακαδημαϊκός να διδάσκει μεταπτυχιακά μαθήματα. Από λογική άποψη έχουμε έναν υπαρξιακό προσδιορισμό της ποσότητας.

Γενικά, ένα στοιχείο owl:Restriction περιέχει ένα στοιχείο owl:onProperty και έναν ή περισσότερους περιορισμούς. Ένας τύπος περιορισμού κατά την δήλωση καθορίζει περιορισμούς στις τιμές που μπορεί να πάρει μια ιδιότητα:

Owl:allValuesFrom, owl:hasValue και owl:someValuesFrom.

Ένας άλλος τύπος καθορίζει επιπλέον περιορισμούς. Για παράδειγμα μπορούμε να απαιτήσουμε κάθε μάθημα να γίνεται από κάποιον :

```

<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>

```

Εδώ είναι σκόπιμο να σημειώσουμε ότι το «1» θα μεταγλωττιστεί ως nonNegativeInteger και γι αυτό χρησιμοποιούμε το xsd namespace για την δήλωση στην επικεφαλίδα έτσι ώστε να το συσχετίσουμε με το XML schema.

Ένα διαμέρισμα πρέπει να έχει τουλάχιστον δέκα και λιγότερα από τριάντα άτομα :

```

<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 πιθανό να καθορίσουμε έναν αριθμό, για παράδειγμα ένας μαθητής διδακτορικού πρέπει να έχει ακριβώς δύο επιβλέποντες. Αυτό μπορεί να επιτευχθεί χρησιμοποιώντας τον ίδιο αριθμό σε `owl:minCardinality` και `owl:maxCardinality`. Για μεγαλύτερη ευκολία η OWL προσφέρει και το `owl:cardinality`.

Κλείνοντας σημειώνουμε ότι το `owl:Restriction` καθορίζει μια ανώνυμη κλάση η οποία δεν έχει ID, δεν καθορίζεται με το `owl:Class` και έχει μόνο τοπική εμβέλεια δηλαδή μπορεί μόνο να χρησιμοποιηθεί εκεί όπου εμφανίζεται ο περιορισμός. Όταν μιλάμε για κλάσεις κρατάμε δύο πράγματα στο μυαλό μας, ένα ότι οι κλάσεις καθορίζονται με το `owl:Class` το οποίο περιέχει ένα ID και δεύτερο τις τοπικές ανώνυμες κλάσεις ως μια συλλογή από αντικείμενα τα οποία ικανοποιούν περιοριστικές συνθήκες.

4.4.6 Ειδικές ιδιότητες(Special Properties)

Κάποιες ιδιότητες των στοιχείων μπορούν να καθοριστούν κατευθείαν :

- `owl:TransitiveProperty` καθορίζει μια μεταβατική ιδιότητα όπως «έχει καλύτερο βαθμό από», «είναι ψηλότερος από», «είναι απόγονος των» κ.τ.λ.
- `owl:SymmetricProperty` καθορίζει μια συμμετρική ιδιότητα όπως «έχει τον ίδιο βαθμό με» κ.τ.λ.
- `owl:FunctionalProperty` καθορίζει μια ιδιότητα έχει το πολύ μια τιμή ανά αντικείμενο όπως «ηλικία», «ύψος» κ.τ.λ.

- `owl:InverseFunctionalProperty` καθορίζει μια ιδιότητα για την οποία δύο διαφορετικά αντικείμενα δεν μπορούν να έχουν την ίδια τιμή. Για παράδειγμα η ιδιότητα “`isTheSocialSecurityNumberfor`”.

Ένα παράδειγμα της σύνταξη αυτών βλέπουμε παρακάτω :

```
<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>
```

4.4.7 Συνδυασμοί(Boolean Combinations)

Είναι πιθανό να μιλάμε για συνδυασμούς των κλάσεων(ένωση,τομή). Για παράδειγμα μπορούμε να πούμε ότι τα μαθήματα και τα ακαδημαϊκά μέλη δεν ενώνονται :

```
<owl:Class rdf:about="#course">
  <rdfs:subClassOf>
    <owl:Class>
      <owl:complementOf rdf:resource="#staffMember"/>
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
```

Αυτό μας λέει ότι κάθε μάθημα είναι μια αναφορά στο ακαδημαϊκό προσωπικό και κανένα μάθημα δεν αποτελεί μέλος του ακαδημαϊκού προσωπικού. Σημειώστε ότι η δήλωση θα μπορούσε να εκφραστεί χρησιμοποιώντας το `owl:disjointWith`.

Η ένωση των κλάσεων επιτυγχάνετε χρησιμοποιώντας 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>

```

Το παραπάνω παράδειγμα δεν λέει ότι η καινούρια κλάση είναι υποκλάση της ένωσης αλλά ότι η καινούρια κλάση είναι ίση με την ένωση. Με άλλα λόγια δηλώσαμε μια ισότητα κλάσεων. Επίσης δεν καθορίζεται ότι οι δύο κλάσεις πρέπει να μην έχουν την δυνατότητα να ενωθούν. Είναι πιθανό για ένα μέλος του προσωπικού να είναι και μαθητής.

Η τομή δηλώνεται με 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>

```

Έχουμε κατασκευάσει την τομή δύο κλάσεων, μια από αυτή καθορίζεται ανώνυμη. Η κλάση των αντικειμένων ανήκει στο CS διαμέρισμα. Οι συνδυασμοί μπορεί να είναι εμφωλευμένοι. Το παρακάτω παράδειγμα καθορίζει ότι το διοικητικό προσωπικό δεν είναι τα μέλη του προσωπικού ούτε το προσωπικό τεχνητής υποστήριξης.

```

<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>

```

4.4.8 Απαριθμήσεις(Enumerations)

Μια απαρίθμηση(enumeration) είναι ένα owl:one of στοιχείο το οποίο καθορίζει μια κλάση η οποία περιέχει όλα τα στοιχεία της :

```

<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>

```

4.4.9 Αναφορές (Instances)

Οι αναφορές στις κλάσεις καθορίζονται όπως στην RDF :

```

<rdf:Description rdf:ID="949352">
  <rdf:type rdf:resource="#academicStaffMember"/>

```



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

```
ή
```

```
<academicStaffMember rdf:ID="949352"/>
```

Επίσης μπορούμε να παρέχουμε περαιτέρω λεπτομέρειες όπως :

```
<academicStaffMember rdf:ID="949352">
```

```
  <uni:age rdf:datatype="xsd:integer">39</uni:age>
```

```
</academicStaffMember>
```

Δυστυχώς τα τυπικά συστήματα βάσεων επειδή δύο αναφορές έχουν διαφορετικό όνομα ή ID δεν υποδηλώνει ότι είναι και διαφορετικές. Για παράδειγμα, εάν πούμε ότι κάθε μάθημα διδάσκεται από περισσότερα από ένα μέλος του προσωπικού:

```
<owl:ObjectProperty rdf:ID="isTaughtBy">
```

```
  <rdf:type rdf:resource="owl:FunctionalProperty" />
```

```
</owl:ObjectProperty>
```

Στην συνέχεια δηλώνουμε ότι μία σειρά μαθημάτων διδάσκεται από δυο μέλη του προσωπικού :

```
<course rdf:ID="CIT1111">
```

```
  <isTaughtBy rdf:resource="#949318"/>
```

```
  <isTaughtBy rdf:resource="#949352"/>
```

```
</course>
```

αυτό δεν προκαλεί την OWL να μας δείξει κάποιο σφάλμα. Σε τελευταία ανάλυση, το σύστημα θα μπορούσε να συμπεράνει ότι οι πόροι «949318» και «949352» είναι προφανώς ίσοι. Για να εξασφαλίσουμε ότι τα διαφορετικά άτομα αναγνωρίζονται πρέπει ρητά να βεβαιώσουμε την ανισότητα τους:

```
<lecturer rdf:ID="949318">
```

```
  <owl:differentFrom rdf:resource="#949352"/>
```

```
</lecturer>
```

Επειδή τέτοιες δηλώσεις ανισότητας εμφανίζονται συχνά, και ο απαραίτητος αριθμός τέτοιων δηλώσεων θα μεγάλωνε εάν θέλαμε να δηλώσουμε την ανισότητα ενός μεγάλου αριθμού ατόμων, η OWL παρέχει μια σημείωση στενογραφίας για να βεβαιώσει την ανισότητα όλων των ατόμων σε έναν δεδομένο κατάλογο:

```
<owl:AllDifferent>
  <owl:distinctMembers rdf:parseType="Collection">
    <lecturer rdf:about="#949318"/>
    <lecturer rdf:about="#949352"/>
    <lecturer rdf:about="#949111"/>
  </owl:distinctMembers>
</owl:AllDifferent>
```

Σημειώνουμε εδώ ότι το `owl:distinctMembers` μπορεί μόνο να συνδυαστεί με το `owl:allDifferent`.

5 Εφαρμογές – Παραδείγματα Σημασιολογικού Ιστού(SW)

5.1 Ηλεκτρονική Μάθηση (E-Learning)

Ο παγκόσμιος ιστός αλλάζει πολλούς τομείς της ανθρώπινης δραστηριότητας, μεταξύ αυτών και την εκπαίδευση. Αν και αποδοτικές και σε πολλές περιπτώσεις αποτελεσματικές, οι παραδοσιακές διαδικασίες εκμάθησης δεν είναι κατάλληλες για κάθε άνθρωπο. Η εμφάνιση του Διαδικτύου έχει προετοιμάσει το έδαφος για νέες εκπαιδευτικές διαδικασίες.

Οι αλλαγές είναι ήδη ορατές στις πιο υψηλές βαθμίδες εκπαίδευσης. Όλο και περισσότερα πανεπιστήμια επικεντρώνουν εκ νέου τις δραστηριότητές τους για να παρέχουν περισσότερη ευελιξία στους φοιτητές τους. Τα εικονικά πανεπιστήμια και τα online μαθήματα είναι μόνο ένα μικρό μέρος αυτών των δραστηριοτήτων.

Μπορούμε επίσης να αναμείνουμε ότι η ηλεκτρονική μάθηση θα έχει ακόμα μεγαλύτερο αντίκτυπο στις δραστηριότητες δια βίου μάθησης. Ένας από τους πιο κρίσιμους μηχανισμούς για την ανταγωνιστικότητα μιας εταιρείας είναι η βελτίωση των δεξιοτήτων των υπαλλήλων της. Οι εταιρείες απαιτούν εκπαιδευτικές διαδικασίες που να είναι προσαρμοσμένες στις συγκεκριμένες ανάγκες τους, και να μπορούν να ενσωματωθούν ιδανικά στις καθημερινές εργασίες τους. Αυτές οι απαιτήσεις δεν είναι συμβατές με την παραδοσιακή εκμάθηση, αλλά με την ηλεκτρονική μάθηση μπορούν αυτές οι υποσχέσεις να γίνουν πραγματικότητα.

Προβλήματα

Σε σύγκριση με τις παραδοσιακές μεθόδους εκμάθησης, η ηλεκτρονική μάθηση δεν καθοδηγείται από κάποιον εκπαιδευτικό. Συγκεκριμένα, οι ενδιαφερόμενοι μπορούν να έχουν πρόσβαση στο υλικό με μια σειρά που προκαθορίζεται μόνο από τους ίδιους, και μπορούν να συνθέσουν τα μαθήματα με βάση τις δικές τους ανάγκες. Για αυτή τη προσέγγιση, το εκπαιδευτικό υλικό πρέπει να εξοπλιστεί με πρόσθετες πληροφορίες για να υποστηρίξει αποτελεσματικά την ανάκτηση συγκεκριμένων πληροφοριών.

Η χρήση των μεταδεδομένων (metadata) είναι μια αποτελεσματική λύση που έχει ακολουθηθεί, με έναν πολύ περιορισμένο τρόπο, για πολύ καιρό. Στην κοινότητα της ηλεκτρονικής μάθησης, έχουν προκύψει πρότυπα όπως το IEEE LOM. Συσχετίζουν πληροφορίες υλικών εκμάθησης, όπως εκπαιδευτικές και παιδαγωγικές ιδιότητες, δικαιώματα πρόσβασης, όροι χρησιμοποίησης, και σχέσεις με άλλες εκπαιδευτικές πηγές.

Αν και αυτά τα πρότυπα είναι χρήσιμα, έχουν ένα μειονέκτημα κοινό για όλες τις λύσεις που είναι βασισμένες αποκλειστικά στα μεταδεδομένα: έχουν έλλειψη της σημασιολογίας. Κατά συνέπεια ο συνδυασμός υλικών από διαφορετικούς συντάκτες μπορεί να είναι δύσκολος, η ανάκτηση δεν μπορεί να υποστηριχθεί βέλτιστα και η ανάκτηση με την οργάνωση των πηγών εκμάθησης πρέπει να γίνει χειροκίνητα. Αυτού του είδους των προβλημάτων μπορούν να αποφευχθούν εάν υιοθετηθεί η προσέγγιση Semantic Web.

Η συμβολή της τεχνολογίας Semantic Web

Οι βασικές ιδέες του Semantic Web, δηλαδή, η οντολογία, και τα μεταδεδομένα που είναι επεξεργάσιμα από τις μηχανές, καθιερώνουν μια ελπιδοφόρο προσέγγιση για να ικανοποιήσουν τις απαιτήσεις της ηλεκτρονικής μάθησης. Μπορεί να υποστηρίξει και τη σημασιολογική αναζήτηση και την

εννοιολογική πλοήγηση των εκπαιδευτικών υλικών:

- Τα υλικά εκμάθησης, ενδεχομένως από διαφορετικούς συντάκτες, μπορούν να συνδεθούν με μια κοινή οντολογία. Οι εξατομικευμένες σειρές μαθημάτων μπορούν να σχεδιαστούν μέσω της σημασιολογικής αναζήτησης, και τα υλικά εκμάθησης μπορούν να ανακτηθούν στα πλαίσια των πραγματικών προβλημάτων, όπως επιθυμούν οι εκπαιδευόμενοι.
- Η γνώση μπορεί να προσεγγιστεί με οποιαδήποτε σειρά επιθυμεί ο εκπαιδευόμενος, σύμφωνα με τα ενδιαφέροντα και τις ανάγκες του. Φυσικά, η κατάλληλη σημασιολογική επισημείωση θα θέσει περιορισμούς σε περιπτώσεις όπου οι αναγκαίες προϋποθέσεις είναι απαραίτητες.
- Το Semantic Web μπορεί να παρέχει μια ομοιόμορφη πλατφόρμα για τις επιχειρησιακές διαδικασίες των εταιρειών, και οι δραστηριότητες εκμάθησης μπορούν να ενσωματωθούν σε αυτές τις διαδικασίες. Αυτή η λύση μπορεί να είναι ιδιαίτερα χρήσιμη για τις εμπορικές επιχειρήσεις.

Οι οντολογίες στην ηλεκτρονική μάθηση(Ontologies for e-Learning)

Σε ένα περιβάλλον ηλεκτρονικής μάθησης η κατάσταση στην οποία διαφορετικοί συντάκτες χρησιμοποιούν διαφορετικές ορολογίες, μπορεί εύκολα να προκύψει με συνέπεια ο συνδυασμός υλικών εκμάθησης να γίνεται ολοένα και πιο δύσκολος. Η πολυπλοκότητα αυξάνει το πρόβλημα της ανάκτησης πληροφοριών από το γεγονός ότι χαρακτηριστικά οι εκπαιδευτικοί και οι εκπαιδευόμενοι έχουν πολύ διαφορετικά υπόβαθρα και επίπεδα γνώσης.

Επομένως, απαιτείται κάποιος μηχανισμός για την κοινή κατανόηση. Οι οντολογίες είναι ένας ισχυρός μηχανισμός για τον σκοπό αυτό. Σε ένα περιβάλλον e-learning διακρίνουμε τρεις τύπους γνώσεων και κατά συνέπεια τρεις τύπους οντολογιών: περιεχομένου, παιδαγωγική, και δομής.

- Μια οντολογία περιεχομένου περιγράφει τις βασικές έννοιες της περιοχής στην οποία η εκμάθηση πραγματοποιείται (π.χ., ιστορία ή πληροφορική). Περιλαμβάνει επίσης τις σχέσεις μεταξύ αυτών των εννοιών, και μερικές βασικές ιδιότητες. Παραδείγματος χάριν, η μελέτη της Αθήνας στα κλασσικά χρόνια είναι μέρος της ιστορίας της αρχαίας Ελλάδας, η οποία είναι συνέχεια της αρχαίας ιστορίας. Η οντολογία πρέπει να περιλαμβάνει τη σχέση «είναι μέρος» και το γεγονός ότι είναι μεταβατική. Με αυτόν τον τρόπο, ένας αυτοματοποιημένος πράκτορας(agent) μπορεί να συμπεράνει ότι η γνώση σχετικά με την κλασσική Αθήνα μπορεί να βρεθεί κάτω από την αρχαία ιστορία. Μια οντολογία περιεχομένου μπορεί επίσης να κάνει χρήση των σχέσεων για να καταλάβει τα συνώνυμα, τις συντμήσεις, και ούτω καθεξής.
- Τα παιδαγωγικά ζητήματα μπορούν να αντιμετωπιστούν με μια παιδαγωγική οντολογία. Για παράδειγμα, το υλικό μπορεί να ταξινομηθεί ως διάλεξη, σεμινάριο, παράδειγμα, άσκηση, λύση, και ούτω καθεξής.
- Τέλος, μια οντολογία δομής χρησιμοποιείται για να προσδιορίσει τη λογική δομή των υλικών εκμάθησης. Χαρακτηριστική γνώση αυτού του είδους περιλαμβάνει ιεραρχικές και σχέσεις πλοήγησης όπως previous, next, hasPart, isPartOf, requires, and isBasedOn. Οι σχέσεις μεταξύ αυτών των σχέσεων μπορούν επίσης να καθοριστούν, παραδείγματος χάριν, hasPart και isPartOf είναι αντίστροφες σχέσεις.

Είναι φυσικό να αναπτυχθούν τα συστήματα ηλεκτρονικής μάθησης στο Web και κατά συνέπεια μια γλώσσα οντολογίας Web πρέπει να χρησιμοποιηθεί.

5.2 Υπηρεσίες Παγκόσμιου Ιστού (Web Services)

Γενικά

Με τον όρο υπηρεσίες παγκόσμιου Ιστού εννοούμε τους ιστοχώρους που όχι μόνο παρέχουν στατικές πληροφορίες, αλλά περιλαμβάνουν την αλληλεπίδραση με τους χρήστες και συχνά επιτρέπουν στους χρήστες κάποιες ενέργειες. Συνήθως μια διάκριση γίνεται μεταξύ απλών και σύνθετων Web services.

Απλές υπηρεσίες παγκόσμιου Ιστού περιλαμβάνουν ένα ενιαίο πρόγραμμα προσβασιμότητας στον ιστό, έναν αισθητήρα, ή μια συσκευή που δεν στηρίζονται επάνω, σε άλλες υπηρεσίες παγκόσμιου Ιστού δεν απαιτούν την περαιτέρω αλληλεπίδραση με το χρήστη πέρα από μια απλή απάντηση. Χαρακτηριστικά παραδείγματα είναι πληροφορίες παροχής υπηρεσιών, όπως ένας ανιχνευτής πτήσης και μια υπηρεσία που επιστρέφει τον ταχυδρομικό κώδικα μιας δεδομένης διεύθυνσης.

Σύνθετες υπηρεσίες παγκόσμιου Ιστού αποτελούνται από απλούστερες υπηρεσίες και απαιτούν συχνά την αλληλεπίδραση με το χρήστη, ο οποίος μπορεί να κάνει επιλογές ή να παρέχει πληροφορίες υπό όρους. Για παράδειγμα η αλληλεπίδραση με το χρήστη με ένα online κατάστημα μουσικής περιλαμβάνει αναζήτηση για CD και τίτλους με διάφορα κριτήρια, ανάγνωση κριτικών, προσθήκη CD στο καλάθι αγορών, παροχή πληροφοριών πιστωτικών καρτών και πληροφορίες διεύθυνσης.

Προβλήματα

Τα SOAP, WSDL, UDDI, και BPEL4WS είναι οι στάνταρ τεχνολογίες για να δημιουργήσουμε μια εφαρμογή Web service. Όμως, αποτυγχάνουν στους στόχους του αυτοματισμού και της διαλειτουργικότητας επειδή απαιτούν χρήστες

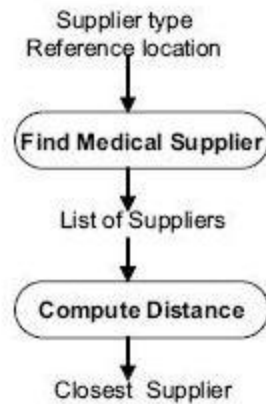
στο επίπεδο των βρόχων. Για παράδειγμα, πρέπει να αναζητηθεί πληροφορία και να συμπληρωθούν φόρμες. Επίσης, η WSDL καθορίζει τη λειτουργικότητα μιας υπηρεσίας μόνο σε συντακτικό επίπεδο έτσι άλλα προγράμματα δεν μπορούν να κάνουν χρήση των πληροφοριών αυτής της λειτουργικότητας.

Η συμβολή της τεχνολογίας Semantic Web

Το Semantic Web διευθετεί τους περιορισμούς της τρέχουσας τεχνολογίας των Web υπηρεσιών με την αύξηση της περιγραφής των υπηρεσιών με ένα σημασιολογικό επίπεδο με στόχο να πετύχει αυτοματισμούς στην αναζήτηση, σύνθεση, επιτήρηση και στην εκτέλεση. Η αυτοματοποίηση αυτών των εργασιών είναι από πολλούς τομείς επιθυμητή.

Παράδειγμα

Το παράδειγμα αναφέρεται στην εξειδίκευση μιας γενικότερης εργασίας βρίσκοντας το πλησιέστερο ιατρικό προμηθευτή (medical provider). Μια στρατηγική για να εκτελέσουμε αυτήν την εργασία είναι: (1) ανάκτηση των λεπτομερειών από όλους τους ιατρικούς προμηθευτές (ενός συγκεκριμένου τύπου) και (2) επιλογή του πλησιέστερου υπολογίζοντας την απόσταση ανάμεσα στην τοποθεσία του προμηθευτή και μιας αναφοράς τοποθεσίας. Αυτή η ροή μπορεί να αναπαρασταθεί με το παρακάτω σχήμα.



Work flow for finding the closest medical supplier

Για το παράδειγμα αρκεί να αναπαράγουμε αυτή τη ροή με μια υπηρεσία MedicareSupplier και με μια Web υπηρεσία που υπολογίζει την απόσταση μεταξύ των ταχυδρομικών κωδικών. Η υπηρεσία της τεχνολογίας Semantic Web έχει σαν στόχο την αυτοματοποίηση της εκτέλεσης τέτοιων εργασιών βασισμένες στη σημασιολογική περιγραφή των υπηρεσιών παγκόσμιου Ιστού. Χρησιμοποιώντας τέτοιες περιγραφές μπορούν να επιλεγούν οι σωστές υπηρεσίες και να συνδυαστούν με τέτοιο τρόπο που θα λύσουν την εργασία πολύ πιο γρήγορα.

Ένα κοινό χαρακτηριστικό όλων των εν ενεργεία πλαισίων για τις περιγραφές των υπηρεσιών του Semantic Web (ανάμεσα σε αυτές είναι OWL-S και WSMO25) είναι ότι συνδυάζουν δύο ειδών οντολογίες για να αποκτήσουν μια περιγραφή υπηρεσίας.

Πρώτα, μια γενική οντολογία Web υπηρεσιών (generic Web service ontology), όπως η OWL-S, καθορίζει γενικές έννοιες Web υπηρεσιών (π.χ. inputs, outputs) και συνιστούν τη ραχοκοκαλιά της περιγραφής υπηρεσίας του semantic Web.

Δεύτερον, μια οντολογία domain καθορίζει τη γνώση στη περιοχή των Web υπηρεσιών, όπως τύπους παραμέτρων υπηρεσιών (π.χ. Πόλη) και λειτουργικότητες (π.χ. FindMedicalSupplier), που συμπληρώνει αυτό το γενικό πλαίσιο.

6 *Web Annotation*

6.1 Η διαδικασία των επισημειώσεων στον Παγκόσμιο Ιστό

Στην σημερινή εποχή πολλές ομάδες ανθρώπων ασχολούνται με την δημιουργία ιστοσελίδων οι οποίες θα δίνουν την δυνατότητα στους χρήστες να κάνουν επισημειώσεις στο διαδίκτυο(annotate service).

Η υπηρεσία βρίσκεται σε αρχικό στάδιο αλλά πολλοί χρήστες την αναζητούν. Η βασική ιδέα είναι ότι ένας χρήστης μπορεί να κρατήσει σημειώσεις σε μια ιστοσελίδα και άλλοι χρήστες μπορούν να δουν, να αλλάξουν τις ήδη υπάρχοντες σημειώσεις και να προσθέσουν δικές τους οποιαδήποτε στιγμή.

Για παράδειγμα μια μελλοντική χρήση θα ήταν ένας χρήστης να σημειώσει κάποιες ερωτήσεις σε μια ιστοσελίδα και μέσω ηλεκτρονικού ταχυδρομείου να αποστείλει αυτές τις επισημειωμένες ιστοσελίδες σε μία εταιρεία. Η εταιρεία θα μπορούσε να απαντήσει κατευθείαν πάνω στην ιστοσελίδα χωρίς να χρειάζεται να κατεβάσει κάτι ή να δημιουργήσει έναν λογαριασμό. Με σκοπό την υλοποίηση αυτής της υπηρεσίας έχουν δημιουργηθεί πολλά διαδικτυακά εργαλεία.

6.2 Annotations Tools - Παραδείγματα

Παρακάτω αναφέρουμε ενδεικτικά κάποια από τα εργαλεία που έχουν δημιουργηθεί με σκοπό την δυνατότητα των επισημειώσεων στο διαδίκτυο:

1. Diigo

Παρέχει την δυνατότητα επισημειώσεων σε μια σελίδα καθώς και την αποθήκευση, την κατηγοριοποίηση και την από κοινού χρήση αυτής μαζί με τις σημειώσεις. Δημιουργεί κατηγορίες ομάδων (για παράδειγμα ομάδα διαχειριστών) και δίνει την δυνατότητα με δεξί κλικ για την αναζήτηση επισημειωμένων στοιχείων. Τέλος είναι υποστηριζόμενο από τον Internet Explorer και τον Mozilla Firefox.

2. Stickis

Προσθέτει σημειώσεις σε μία ιστοσελίδα, αποθηκεύει, κοινοποιεί και κατηγοριοποιεί τις σημειώσεις. Δυνατότητα προβολής, αποθήκευσης και σχολιασμού στις σημειώσεις από άλλους χρήστες από οποιαδήποτε ιστοσελίδα. Δημιουργεί ομάδες και υποστηρίζεται από τον Internet Explorer.

Toolbar
Create notes.
Turn on and off fetching notes.
Go to your Stickis home page.

Invite
..someone to connect on Stickis.

Navigation
How to get around the Stickis website.

The screenshot shows the Stickis website interface. At the top, there is a navigation bar with links for Home, Your Network, Your Notes, Your Alerts, and Search. Below this is a section for 'Latest notes' and 'Latest alerts', each displaying a grid of notes from various users. At the bottom, there are three main sections: 'Write a note' with a 'Learn how' button, 'Invite a friend' with an 'Invite' button, and 'Add a blog' with an 'Add' button. The callouts point to the toolbar (Stick On, New note, Home), the 'Invite' button, the 'Latest notes' section, the 'Latest alerts' section, the 'Invite a friend' section, and the 'Add a blog' section.

Latest notes
A ticker of notes from sources in Your Network.

Latest alerts
Annotations on webpages, tags, or sources which you've set up to watch in Your Alerts.

Invite a friend
...to see what they're annotating, and let them see your notes too.

Subscribe to blogs
...from Stickis to see what your favorite bloggers are writing about wherever you browse.

3. Sharedcopy

Στο εργαλείο αυτό δίνεται μια δυνατότητα που δεν υπάρχει στα προηγούμενα δύο. Πέρα από τις σημειώσεις και τον αντίστοιχο χρωματισμό των ανάλογων στοιχείων της σελίδας δίνεται η δυνατότητα στον χρήστη να σχεδιάζει γραμμές, κύκλους και τετράγωνα σε αυτήν. Επίσης αποθηκεύει τις ιστοσελίδες που έχουν σημειωθεί και μεταφέρονται μέσω URLs και αποτελεί παράδειγμα διαλειτουργικότητας καθώς υποστηρίζεται από όλους τους browsers.

4. Fleck

Προσθέτει σημειώσεις, δυνατότητα κοινοποίησης των επισημειωμένων σελίδων στο blog του χρήστη. Επιλογή για τον διαμοιρασμό των επισημειωμένων ιστοσελίδων με φίλους μέσω URLs. Υποστηρίζεται από τον Internet Explorer και τον Mozilla Firefox.

How this works
...then you can share this Flecked page with other people...

Bookmarklet
Drag this link up to your Bookmarks Toolbar to install our Bookmarklet:
→ [Fleck!](#)

Extension
Add Fleck to your favorite browser:
[Add to Firefox](#)
 (20kb, Free download, no Spyware)
[Add to Explorer](#) (Soon!)
 (200kb, Free download, no Spyware)

Fleck Examples
Check these out for inspiration and amusement.

Domain	Nr. of Flecks
wired.com	38
techcrunch.com	11
gogelnogel.typepad.com	9
bomega.com	7
palore.com	5
israelwebtour.com	5
google.nl	5
fleck.com	5
blueace.nl	5
attendr.com	4

5. MyStickies

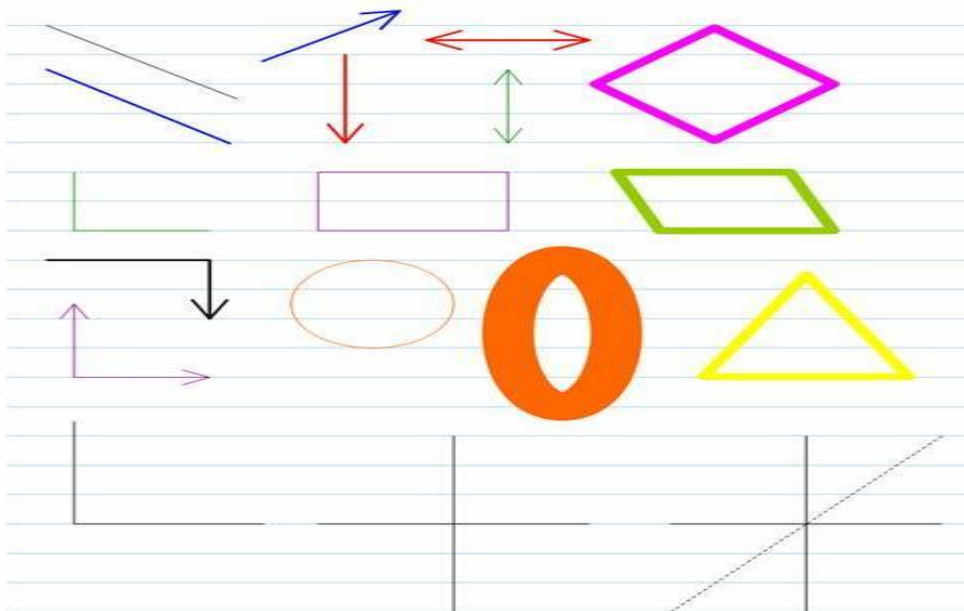
Ένα εργαλείο με όχι και τόσες πολλές δυνατότητες. Προσθέτει σημειώσεις, αποθηκεύει, κατηγοριοποιεί και αναζητεί επισημειώσεις και υποστηρίζεται από τον Mozilla Firefox.

6. JKN

Καινοτομία αυτού του εργαλείου είναι ότι μπορεί να προσθέσει σημειώσεις σε μία ή περισσότερες ιστοσελίδες. Επίσης προσθέτει νούμερα μεταξύ των λέξεων, δυνατότητα σχολίων και εκτύπωση των επισημειωμένων σελίδων. Κύριο πλεονέκτημα του έναντι των ανταγωνιστικών εργαλείων είναι ότι μπορούν οι σελίδες να διαμοιραστούν μέσω URL, email ή blog. Υποστηρίζεται από τον Internet Explorer και τον Mozilla Firefox.

7. Draw Here

Προσθέτει σημειώσεις, χρωματίζει τα αντίστοιχα στοιχεία της ιστοσελίδας και όπως και στο Sharedcopy δίνεται η δυνατότητα στους χρήστες να σημειώνουν χρησιμοποιώντας σχήματα. Διαμοιράζονται οι επισημειωμένες σελίδες μέσω email και blog και υποστηρίζεται από όλους τους browsers.



Στο σχήμα παρατηρούμε τα σχήματα τα οποία μας παρέχει το Draw Here.

Υπάρχουν και άλλα τέτοια διαδικτυακά εργαλεία όπως το trailfire, το ShiftSpace.org κ.α.

6.3 Διαδικτυακή υλοποίηση της εφαρμογής Annotation

Tool

Η εφαρμογή μας αποτελεί μια διαδικτυακή υλοποίηση του Annotation Tool που δημιουργήθηκε με σκοπό της παροχής της δυνατότητας στον χρήστη να επισημαίνει τα στοιχεία μιας οποιαδήποτε ιστοσελίδας.

Οι επισημειώσεις (Annotation) είναι μια από τις πιο κοινές μορφές μεταδεδομένων στα πλαίσια του Semantic Web. Οι επισημειώσεις μπορούν να είναι σχόλια, σημειώσεις, εξηγήσεις, ερωτήσεις, αναφορές, παραδείγματα, συμβουλές, διορθώσεις ή οποιοσδήποτε άλλος τύπος εξωτερικής παρατήρησης που μπορεί να συνδεθεί με μια ιστοσελίδα ή ένα επιλεγμένο μέρος ενός εγγράφου.

Δεδομένου ότι είναι εξωτερικές παρατηρήσεις, είναι δυνατό να σχολιαστούν οποιαδήποτε έγγραφα του Ιστού μεμονωμένα, χωρίς να πρέπει να επεξεργαστούν τα συγκεκριμένα έγγραφα. Από τεχνική άποψη, οι επισημειώσεις μπορούν να θεωρηθούν ως μεταδεδομένα, εφόσον δίνουν πρόσθετες πληροφορίες για ένα υπάρχον κομμάτι των δεδομένων.

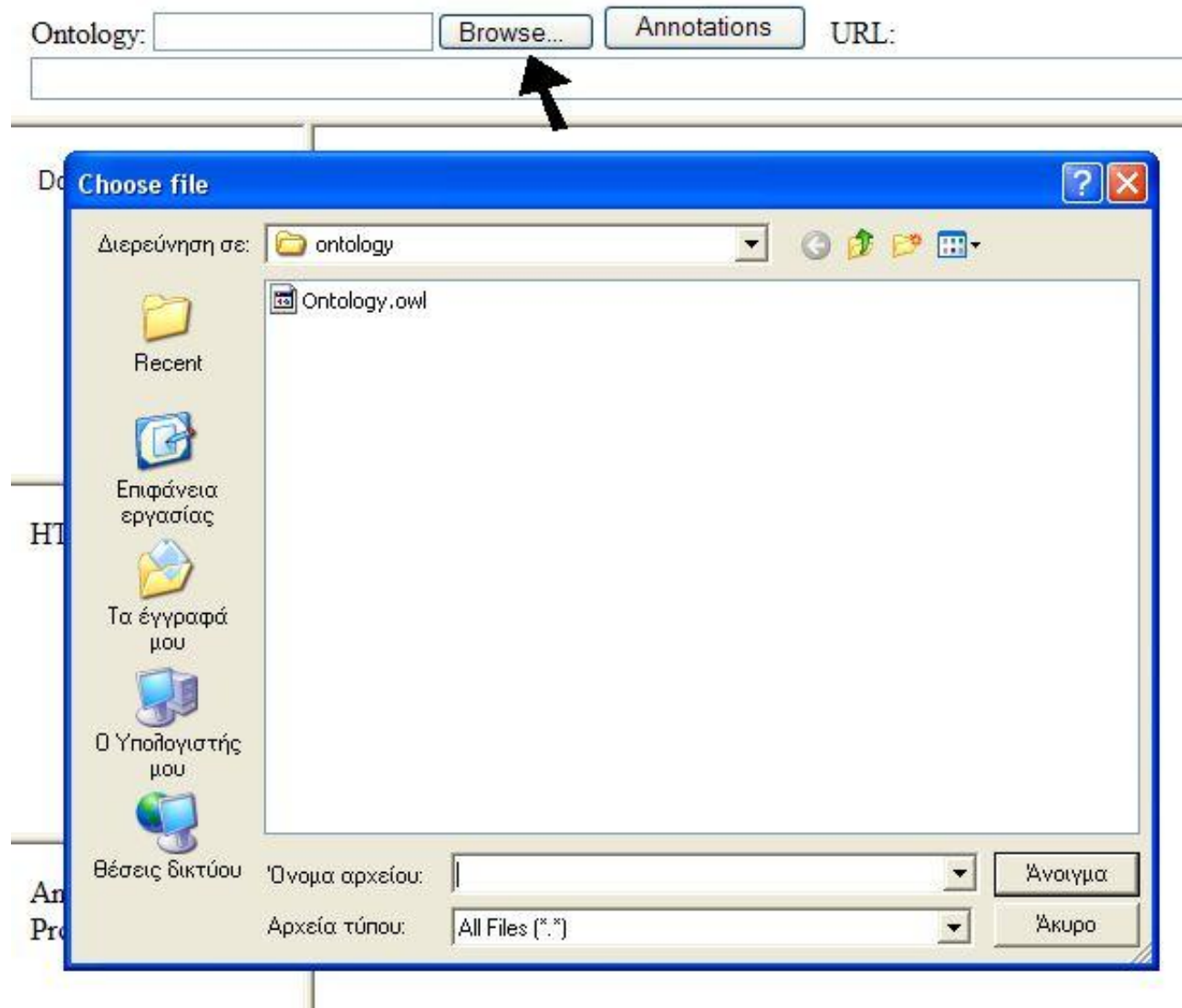
Μπορούν να αποθηκευτούν είτε στους υπολογιστές των χρηστών για ιδιωτική χρήση είτε σε servers. Οι επισημειώσεις που αποθηκεύονται σε servers μπορούν να τεθούν στην διάθεση οποιουδήποτε ώστε να μπορούν να μοιράζονται τις απόψεις και τα σχόλιά για έναν ιστοχώρο σχετικά με το περιεχόμενο, τη χρησιμότητά του, κ.λπ.

Όπως αναφέραμε και στα προηγούμενα κεφάλαια το Semantic Web στοχεύει στους πράκτορες μηχανών (machine agents) που αναπτύσσονται και με

τις κατάλληλες τεχνικές μπορούν να ψάξουν, να φιλτράρουν, να συμπυκνώσουν, ή να διαπραγματευτούν γνώση για τους ανθρώπινους χρήστες. Σε αυτό το σημείο εισέρχονται οι επισημειώσεις για να βοηθήσουν τους πράκτορες μηχανών ώστε να διευκολυνθεί το έργο του Semantic Web.

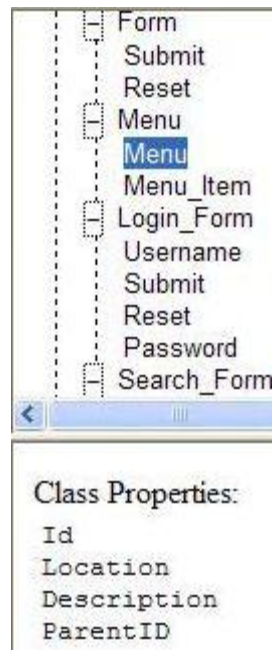
Παρακάτω θα δείξουμε την διαδικασία της σύναψης επισημειώσεων με το εργαλείο που δημιουργήσαμε.

Αρχικά πρέπει να φορτωθεί μια οντολογία. Από το κουμπί “Browse...” ο χρήστης αναζητεί την οντολογία όπως φαίνεται στο παρακάτω σχήμα:

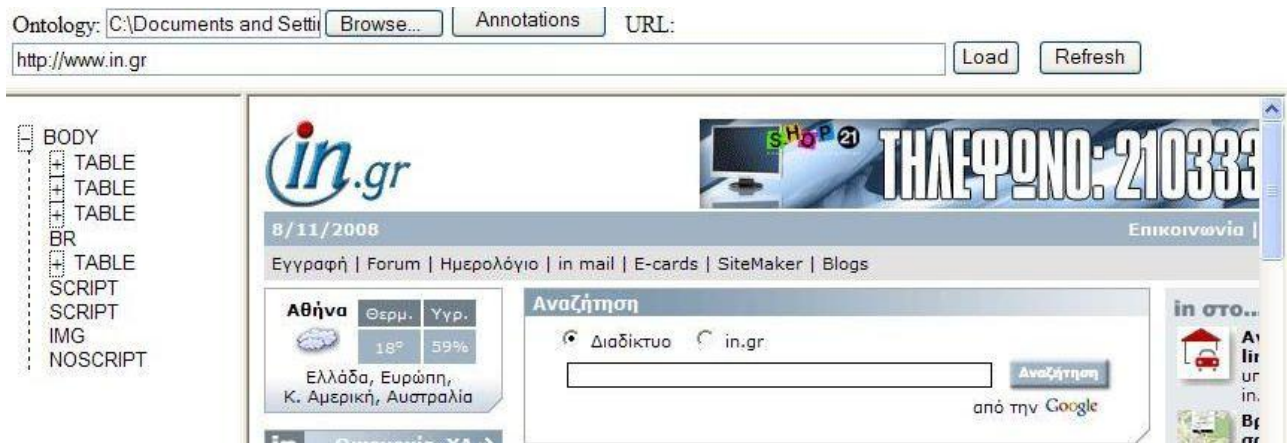


Η οντολογία θα φορτωθεί στα δεξιά του προγράμματος, στο πάνω πλαίσιο(frame) εμφανίζεται η οντολογία, ενώ στο κάτω πλαίσιο οι ιδιότητές της όπως φαίνεται στο επόμενο σχήμα.

Για καλύτερη ευχρηστία ο χρήστης μπορεί να αυξάνει και να μειώνει το μέγεθος όλων των πλαίσιο του προγράμματος ανάλογα με τις ανάγκες του.



Αφού φορτωθεί η οντολογία, ο χρήστης θα πρέπει να πληκτρολογήσει την διεύθυνση που επιθυμεί να επισημείωση και να πατήσει το κουμπί “Load”. Η επιθυμητή ιστοσελίδα πρέπει να εμφανιστεί στο κυρίως πλαίσιο του προγράμματος.



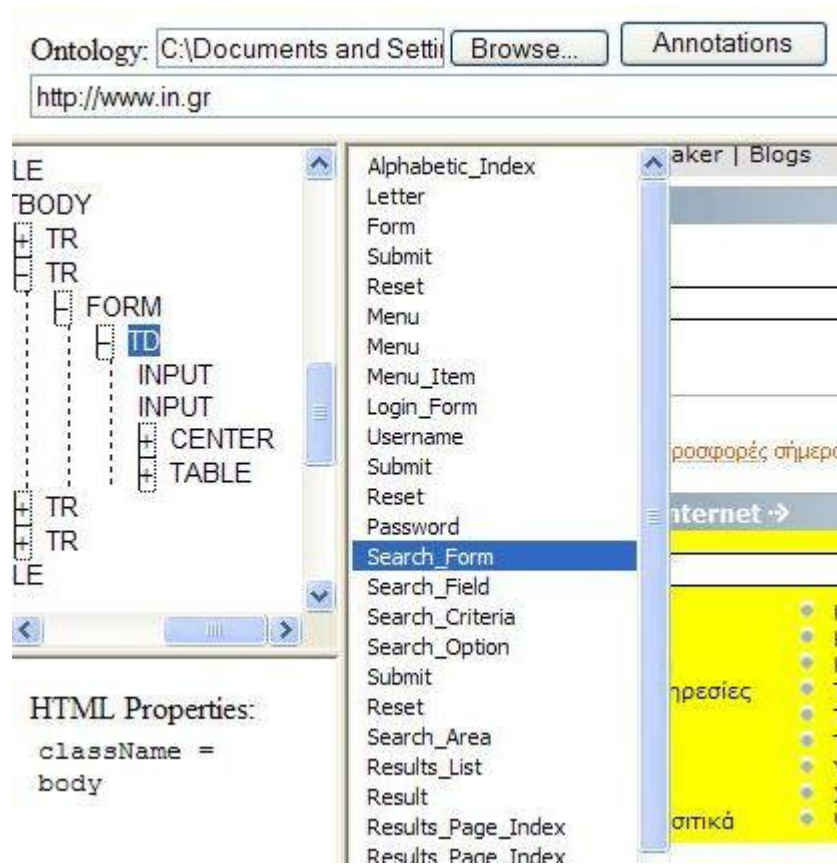
Στη συνέχεια αυτόματα εμφανίζεται στα αριστερά το Dom Tree της σελίδας που έχει ηλεκτρολογηθεί. Εάν η σελίδα αποτελείται από κομμάτια AJAX τα οποία φορτώνονται αργότερα τότε θα πρέπει να χρησιμοποιήσει και το κουμπί “Refresh” ώστε το Dom Tree να είναι πλήρες.

Ο χρήστης μπορεί να επιλέξει όποιο στοιχείο της σελίδας θέλει για επισημείωση απλά αναζητώντας το μέσω του Dom Tree. Η αναζήτηση αυτή γίνεται εύκολα επειδή κάθε φορά που επιλέγεται ένα στοιχείο του Dom Tree το ανάλογο μέρος στη σελίδα εμφανίζεται με κίτρινο χρώμα. Αυτό γίνεται ώστε να μην μπορεί ο χρήστης να κάνει λάθος ακόμα και όταν επιλέξει άλλο στοιχείο από αυτό που επιθυμεί.

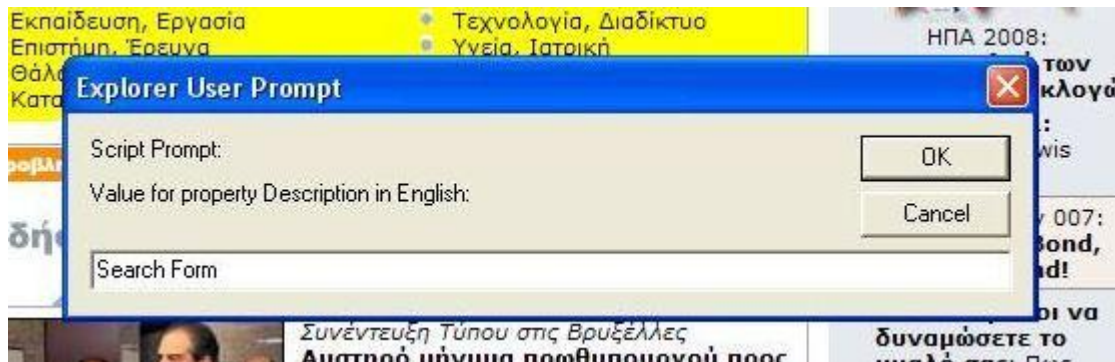
The screenshot shows a web browser interface. On the left, the DOM tree is visible, showing a hierarchy of elements: LE, BODY, TR, TR, FORM, TD, INPUT, INPUT, CENTER, TABLE, TR, TR, LE. The 'td' element is highlighted in blue. Below the DOM tree, the 'HTML Properties' panel shows 'className = body'. The main content area displays a search bar with 'Αναζήτηση' and 'από την Google' buttons. Below this is a section titled 'Κατάλογος Ελληνικού Internet' with a search bar and a list of categories including: Αθλητισμός, Απόδημος Ελληνισμός, Αυτοκίνητο, Μοτοσικλέτα, Βιομηχανία, Εμπόριο, Υψηρείες, Ειδήσεις, ΜΜΕ, Εκπαίδευση, Εργασία, Επιστήμη, Έρευνα, Θάλασσα, Σκάφος, Κατασκευές, Δόμηση, Μεσικά, Καταστήματα On-line, Κοινωνία, Εθελοντισμός, Κράτος, Πολιτική, Ταξίδια, Τουρισμός, Τέχνες, Πολιτισμός, Τεχνολογία, Διαδίκτυο, Υγεία, Ιατρική, Χρήμα, Ασφάλειες, Νομικά, Ψυχαγωγία, Χόμπι. The browser's address bar shows 'http://www.in.gr'.

Επίσης οι ιδιότητες του εκάστοτε στοιχείου εμφανίζονται εάν υπάρχουν στο πλαίσιο(frame) κάτω από το Dom Tree. Για παράδειγμα ο χρήστης θέλει να επισημειώσει το κομμάτι της αναζήτησης που εμφανίζεται παραπάνω. Αφού το επιλέξει από το Dom Tree θα εμφανιστεί με κίτρινο χρώμα στη σελίδα και θα μπορεί να δει τις ιδιότητες του στοιχείου (“className = body”).

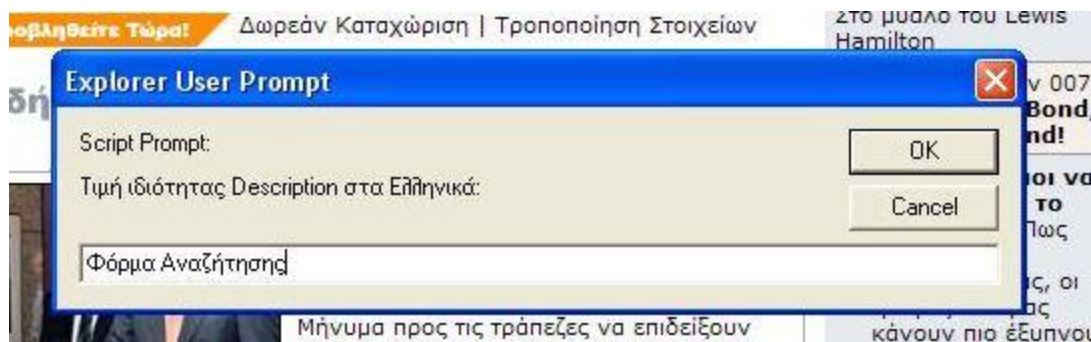
Ακολούθως, με δεξί κλικ στο επιθυμητό στοιχείο, που στην περίπτωση αυτή είναι το “td”, εμφανίζεται η οντολογία.



Ανάλογα με το στοιχείο κάνουμε την κατάλληλη επιλογή. Στο παράδειγμα, ο χρήστης θα επιλέξει το “Search_Form”. Εφόσον γίνει αυτή η επιλογή γίνεται ένας διάλογος με το χρήστη με μια σειρά από παράθυρα διαλόγου(message boxes), το πρώτο από τα οποία τον παροτρύνει να συμπληρώσει μια τιμή για την περιγραφή της ιδιότητας του στοιχείου στα αγγλικά.

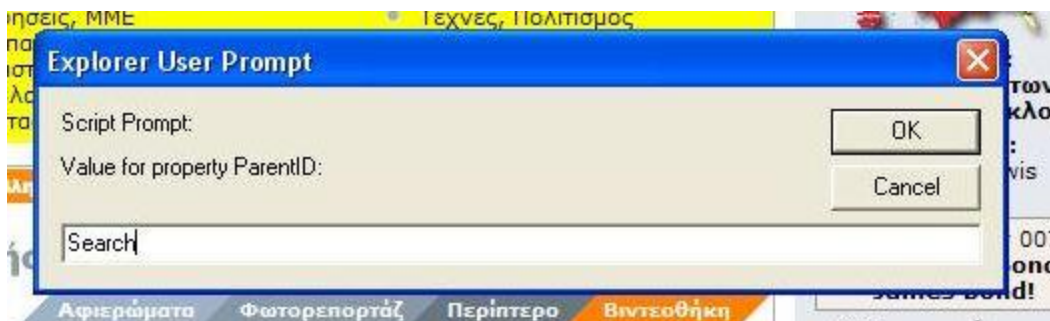


Το επόμενο παράθυρο διαλόγου (message box) αφορά την τιμή για την περιγραφή

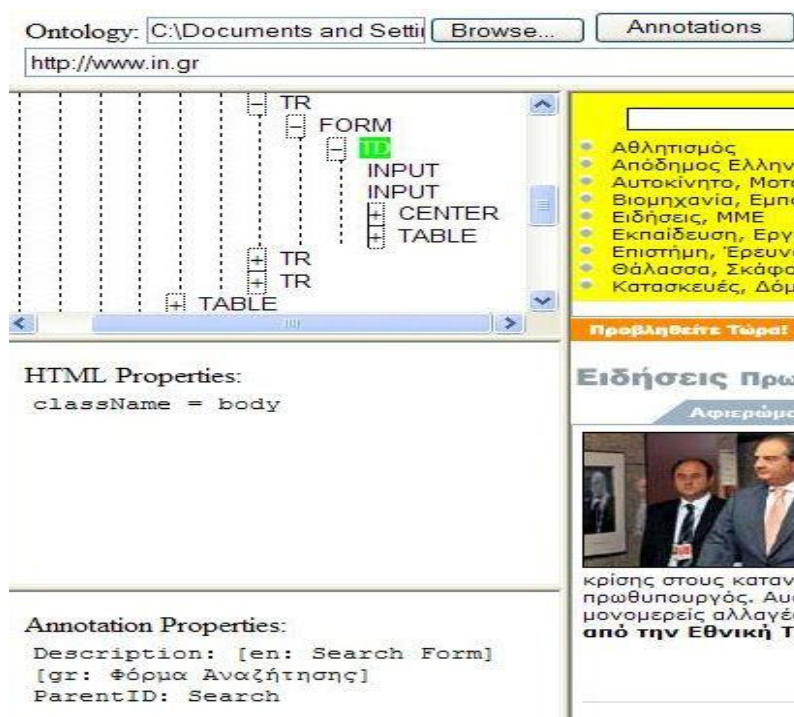


της ιδιότητας του στοιχείου στα ελληνικά.

Το τελευταίο παράθυρο διαλόγου αφορά την τιμή για την ιδιότητα του γονικού στοιχείου στα αγγλικά.



Αφού ολοκληρωθεί αυτός ο διάλογος, το στοιχείο που επισημειώθηκε έχει αλλάξει χρώμα για να ξεχωρίζει, και οι ιδιότητες του εμφανίζονται στο τελευταίο πλαίσιο(frame) της αριστερής πλευράς όπως φαίνεται στην παρακάτω εικόνα.



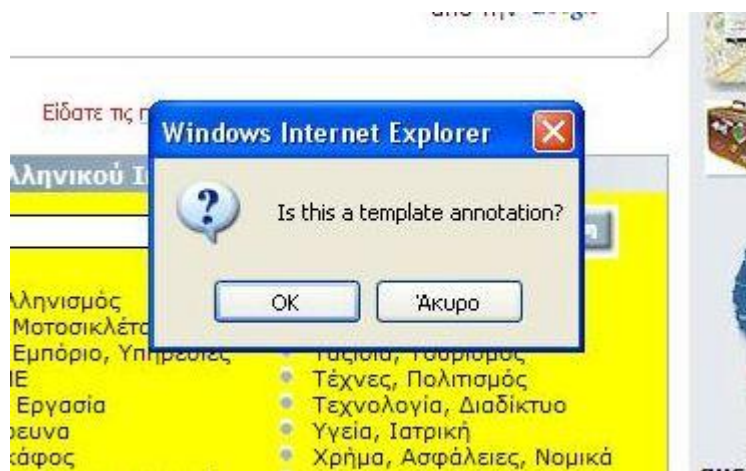
Με τον παραπάνω τρόπο γίνεται η επισημείωση ενός στοιχείου. Η

προηγούμενη διαδικασία, μπορεί να επαναληφθεί για όσες φορές επιθυμεί ο χρήστης, ακόμα και για όλα τα στοιχεία της ιστοσελίδας.

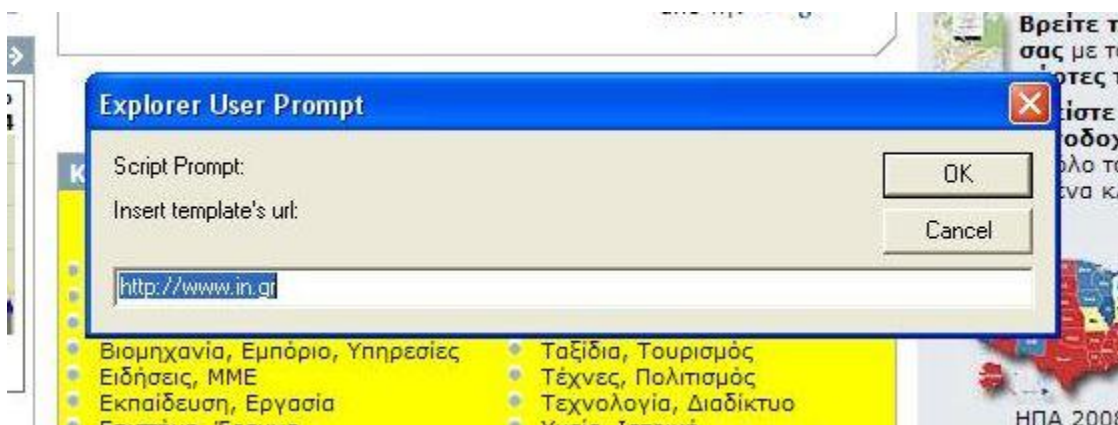
Το επόμενο βήμα είναι να εξάγουμε το αποτέλεσμα σε ένα rdf αρχείο. Αυτό θα επιτευχθεί με το κουμπί “Annotations”, στην αρχή κάνοντας ένα διάλογο με το χρήστη για κάποιες επιπλέον πληροφορίες και τέλος αποθήκευση των παραπάνω στο rdf αρχείο.



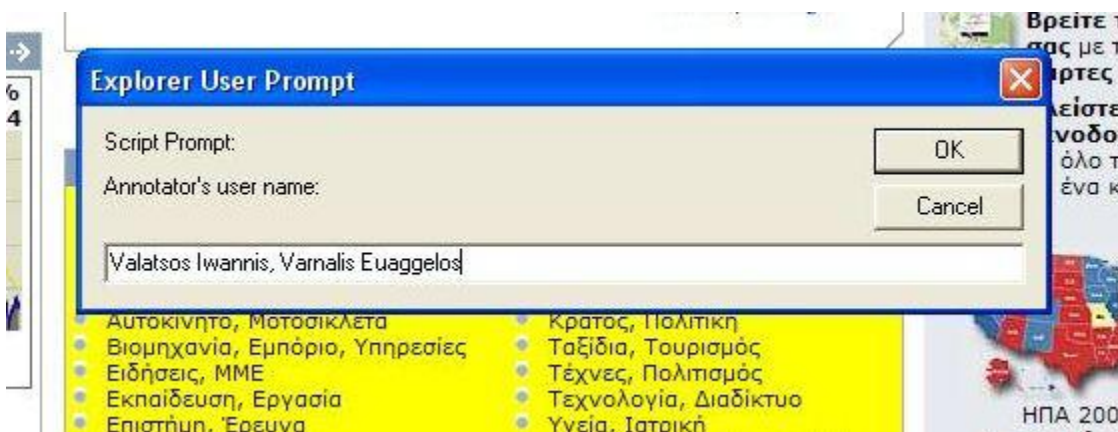
Οι διάλογοι καθώς και η αποθήκευση απεικονίζονται στις παρακάτω εικόνες.



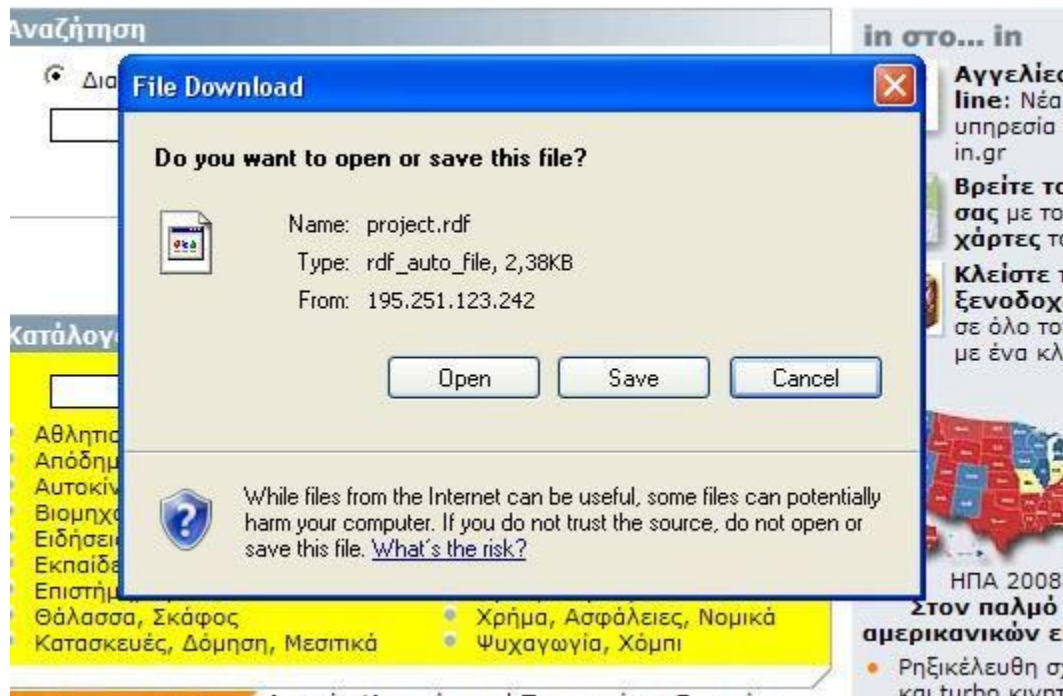
Η δεύτερη ερώτηση έχει να κάνει με την εισαγωγή της διεύθυνση της ιστοσελίδας.



Έπειτα ο χρήστης εισάγει το όνομά του.



Στη συνέχεια εμφανίζεται η επιλογή για την αποθήκευση σε ένα rdf αρχείο.



Τέλος παρακάτω εμφανίζεται το κομμάτι κώδικα μέσα στο rdf αρχείο.

```
<SeESyntax:Annotation rdf:about="http://www.in.gr">
```

```
<SeESyntax:Template      rdf:ID="_template_Valatsos      Iwannis,      Varnalis
Euaggelos">True</SeESyntax:Template>
```

```
<SeESyntax:Contains
rdf:ID="_document_html_body_table_tbody_tr_td_table_tbody_tr_form_td__Vala
tsos Iwannis, Varnalis Euaggelos">
```

```
<SeEBrowser:Search_Form
rdf:about="http://www.in.gr#document/html/body/table/tbody/tr/td/table/tbody/tr/f
orm/td/">
```

```
<SeEBrowser:Id
rdf:ID="_document_html_body_table_tbody_tr_td_table_tbody_tr_form_td__Id_Valatsos Iwannis, Varnalis Euaggelos">Search_Form1</SeEBrowser:Id>
```

```
<SeEBrowser:Description
rdf:ID="_document_html_body_table_tbody_tr_td_table_tbody_tr_form_td__Description_Valatsos Iwannis, Varnalis Euaggelos">[en: Search Form][gr: Φόρμα Αναζήτησης]</SeEBrowser:Description>
```

```
<SeEBrowser:ParentID
rdf:ID="_document_html_body_table_tbody_tr_td_table_tbody_tr_form_td__ParentID_Valatsos Iwannis, Varnalis Euaggelos">Search</SeEBrowser:ParentID></SeEBrowser:Search_Form>
```

```
</SeESyntax:Contains>
```

```
</SeESyntax:Annotation>
```

```
<rdf:Description rdf:about="#_template_Valatsos Iwannis, Varnalis Euaggelos">
  <SeESyntax:Annotator>Valatsos Iwannis, Varnalis
  Euaggelos</SeESyntax:Annotator>
</rdf:Description>
```

```
<rdf:Description
rdf:about="#_document_html_body_table_tbody_tr_td_table_tbody_tr_form_td__Valatsos Iwannis, Varnalis Euaggelos">
  <SeESyntax:Annotator>Valatsos Iwannis, Varnalis
```

```
Euaggelos</SeESyntax:Annotator>
</rdf:Description>
```

```
<rdf:Description
rdf:about="#_document_html_body_table_tbody_tr_td_table_tbody_tr_form_td__I
d_Valatsos Iwannis, Varnalis Euaggelos">
    <SeESyntax:Annotator>Valatsos          Iwannis,          Varnalis
Euaggelos</SeESyntax:Annotator>
</rdf:Description>
```

```
<rdf:Description
rdf:about="#_document_html_body_table_tbody_tr_td_table_tbody_tr_form_td__
Description_Valatsos Iwannis, Varnalis Euaggelos">
    <SeESyntax:Annotator>Valatsos          Iwannis,          Varnalis
Euaggelos</SeESyntax:Annotator>
</rdf:Description>
```

```
<rdf:Description
rdf:about="#_document_html_body_table_tbody_tr_td_table_tbody_tr_form_td__
ParentID_Valatsos Iwannis, Varnalis Euaggelos">
    <SeESyntax:Annotator>Valatsos          Iwannis,          Varnalis
Euaggelos</SeESyntax:Annotator>
</rdf:Description>
```

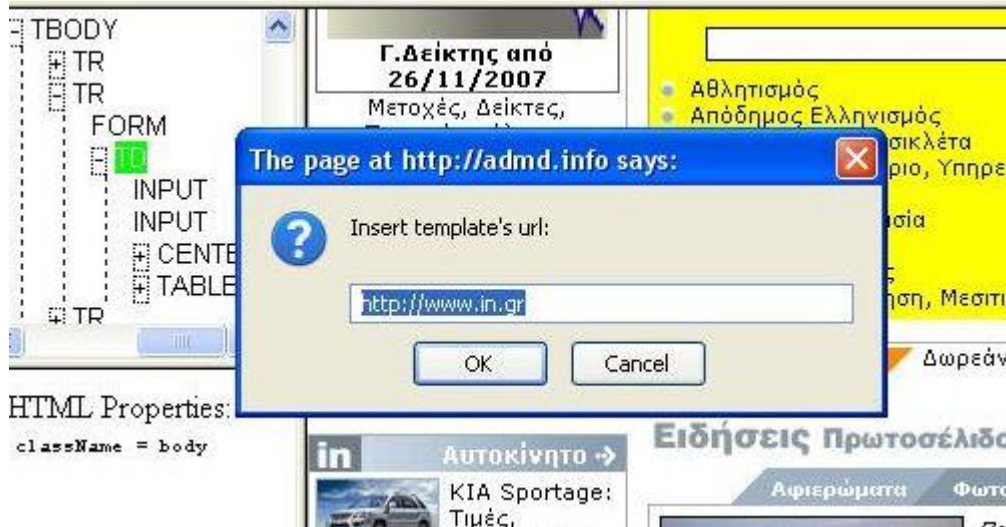
Τέλος ο χρήστης έχει μια ακόμη εναλλακτική λύση. Να μην αποθηκεύσει το αρχείο αλλά να το ανεβάσει απευθείας στο server αρχείο αλλά να το <http://erodios.it.teithe.gr:8088/AnnotationStorageServer/>

Η διαδικασία έχει ως εξής:

Ο χρήστης πρέπει να πατήσει το κουμπί “Upload” ώστε να ξεκινήσει ένας διάλογος όπου η πρώτη ερώτηση φαίνεται παρακάτω:

The screenshot shows a web browser window with the address bar containing 'http://www.in.gr'. The browser's toolbar includes buttons for 'Ontology: D:\Documents and Settings\...', 'Browse...', 'Annotations', 'Upload', and 'URL:'. Below the toolbar, there is a search bar and a 'Load' button. The main content area displays a search results page with a yellow background. A dialog box is overlaid on the page, titled 'The page at http://admd.info says:' and containing the question 'Is this a template annotation?'. The dialog has 'OK' and 'Cancel' buttons. The background page shows a search results page with various categories like 'Αθλητισμός', 'Απόδημος Ελληνισμός', 'Αυτοκίνητο, Μοτοσικλέτα', 'Βιομηχανία, Εμπόριο, Υπηρεσίες', 'Ειδήσεις, ΜΜΕ', 'Καταστήματα On-line', 'Κοινωνία, Εθελοντισμός', 'Κράτος, Πολιτική', 'Ταξίδια, Τουρισμός', and 'Τέχνες, Πολιτισμός'. There is also a section for 'Αυτοκίνητο' with a KIA Sportage advertisement.

Με την επόμενη ερώτηση καταχωρούμε το url της ιστοσελίδας.



Στη συνέχεια εισάγουμε το username και το password του χρήστη.



Τέλος, εάν εμφανιστεί άσπρη οθόνη μετά την εισαγωγή του password αυτό είναι δείγμα ότι το ανέβασμα στο server έγινε με επιτυχία. Σε διαφορετική περίπτωση εμφανίζεται μήνυμα λάθους.

7 Συζήτηση – Συμπεράσματα

7.1 Γενικά

Πολλοί λένε ότι το Semantic Web θα είναι το κύριο χαρακτηριστικό του Web 3.0. Εύκολο να το λέμε αλλά στην πράξη δεν είναι τόσο απλό. Αυτή τη στιγμή υπάρχουν κενά τεχνολογίας και τεχνογνωσίας που πρέπει να καλυφθούν ώστε να γίνει κάτι τέτοιο μαζική πραγματικότητα.

Εδώ είναι που έρχονται τα πολλά ερωτήματα. Καθώς πέρα από τα τεχνικά προβλήματα, υπάρχουν τα θέματα με τα πνευματικά δικαιώματα, και με τα πολλαπλά πρότυπα(format). Καθώς οι ιστοσελίδες θα πρέπει να είναι πλήρως κατανοητές όχι μόνο στους ανθρώπους αλλά και στα μηχανήματα.

Το μεγαλύτερο όμως πρόβλημα από όλα ως τώρα όσο και αν φαίνεται παράλογο είναι η συνειδητοποίηση. Στην Ελλάδα το Web 2.0 προχωράει αλλά ο μέσος Έλληνας θα παραξενευτεί αν του αναφέρεις αυτόν τον όρο. Πόσο μάλλον αν αρχίσεις να αναφέρεσαι στο Semantic Web. Η μετάβαση λοιπόν προς το παρόν έχει ανάγκη από πολύ έρευνα και δεν αναμένετε στα επόμενα χρόνια να δούμε ένα μαζικό Semantic Web.

Σενάριο που είναι ακόμη σε μεγάλο βαθμό "επιστημονική φαντασία", έχει όμως οδηγήσει την επιστημονική και ερευνητική κοινότητα σε έναν αγώνα δρόμου για την ανάπτυξη συστημάτων και εφαρμογών που θα εκμεταλλεύονται και θα επεξεργάζονται τις έννοιες (concepts) και το νόημα (semantics) που υπάρχουν ενσωματωμένα σε κάθε τεχνολογικό σύστημα.

Στο εγγύς μέλλον με τις κατάλληλες ενέργειες θα επιτρέπονται στις μηχανές να “κατανοούν” και να επεξεργάζονται καλύτερα τα δεδομένα, τα οποία σήμερα απλά μας παρουσιάζουν.

Επιτυγχάνοντας αυτό, παρέχεται η δυνατότητα επίλυσης προβλημάτων που θεωρούμε βαρετά ή χρονοβόρα, βοηθώντας στη γρήγορη εύρεση πληροφοριών, όπως ιατρικές πληροφορίες, κριτική για μια κινηματογραφική ταινία, εντολή παραγγελίας ενός βιβλίου κ.λπ.

Ας δούμε ένα μελλοντικό παράδειγμα λειτουργίας των μηχανών πάνω στο Semantic Web. Ζητάμε από μια μηχανή να μας δώσει πληροφορίες για το ιατρικό ιστορικό μας. Η μηχανή αναζητά τη βασική πληροφορία από το δικό μας ιστορικό που βρίσκεται αποθηκευμένη στον ηλεκτρονικό φάκελο ενός νοσοκομείου. Ταυτόχρονα, αναλύει την πληροφορία αυτή και μπορεί να μας εμφανίσει πληροφορίες που βρίσκονται σε άλλες σελίδες και αφορούν αυτά που αναφέρονται στο ιστορικό μας.

Επίσης, μπορεί να εντοπίσει στοιχεία γιατρών και νοσοκομείων που μπορούν να μας εξυπηρετήσουν, καταχωρώντας τα στοιχεία τους στην ηλεκτρονική διεύθυνσή μας. Και βέβαια μπορεί να μας υποδείξει την πλέον ενδεδειγμένη διαδρομή για κάθε ένα από αυτά τα κέντρα, λαμβάνοντας υπόψη και στοιχεία από την κίνηση στους δρόμους.

Όσο και αν αυτή η αυτοματοποίηση δείχνει ότι ανήκει στο μέλλον, εντούτοις είναι εφικτή, εάν περαστεί μέσα σε κάθε πληροφορία που υπάρχει στον παγκόσμιο ιστό, πληροφορία που μπορούν να καταλάβουν οι υπολογιστές.

Αυτό προσπαθήσαμε να κάνουμε και εμείς με το εργαλείο που δημιουργήσαμε. Κύριος σκοπός είναι να εξάγουμε ένα αρχείο που θα περιέχει πληροφορίες, για τις πληροφορίες που ήδη υπάρχουν και μας παρουσιάζονται μέσα από τις ιστοσελίδες.

Δε χρειάζεται να υλοποιηθεί το Semantic Web στη πλήρη μορφή του για να δούμε τα πρώτα αποτελέσματα.

Συμπερασματικά μπορούμε να πούμε ότι ενδιαφερόμαστε για τη δημιουργία εργαλείων που επιτρέπουν τους τελικούς χρήστες κυριολεκτικά να χτίζουν το Semantic Web, δηλαδή να δίνουν τα semantics του.

7.2 Πιθανές επεκτάσεις

Με την έρευνα πάνω στο Semantic Web να προχωρά γρήγορα, το επόμενο βήμα στη δημιουργία αυτών των εφαρμογών δεν θα είναι στην ανάπτυξη της εσωτερικής τους δομή, αλλά στο σχεδιασμού του περιβάλλοντος χρήστη.

Για την καλύτερη προώθηση της εφαρμογής αλλά και τη πλήρη κατανόηση της ακόμα και από τον πιο απλό χρηστή καλό θα ήταν να υπάρξει μια συντονισμένη προσπάθεια για τη δημιουργία των κατάλληλων εγχειριδίων χρήσης αλλά και βοηθειών μέσα στην ίδια την εφαρμογή.

Μια μελλοντική επέκταση του εργαλείου είναι να δημιουργηθούν οι κατάλληλες μηχανές ώστε να μπορούν να λαμβάνουν το αρχείο με την περιγραφή των πληροφοριών που εξάγουμε και μέσα από κατάλληλη επεξεργασία να μας παρέχει πρόσθετες πληροφορίες.

Βιβλιογραφία

- 1.** Marshall.C, Shipman.F.M. “Which Semantic Web?”
- 2.** A Semantic Web Primer Grigoris Antoniou and Frank van Harmelen (Second Edition) March 2008
- 3.** Updegrove.A. “THE SEMANTIC WEB: AN INTERVIEW WITH TIM BERNERS-LEE” June 2005, Consortiuminfo.org Consortium Standards Bulletin- June 2005
- 4.** Kong.H, Kahan.J, Koivunen.M-R, Prud'Hommeaux.E, Swick.R.R. Annotea: “An Open RDF Infrastructure for Shared Web Annotations” May 1-5, 2001
- 5.** Berners-Lee.T. “Semantic Web Road map” September 1998
- 6.** Berners-Lee.T, Hendler.J , Lassila.O. “The Semantic Web A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities” Scientific American, May 17, 2001
- 7.** Dwight,Pletzer, “XML” ,Pearson Education(US)
- 8.** Johan, Hjelm, “Creating the Semantic Web with RDF”, Jonh Wiley and Soms Ltd, 2001

- 9.** Dean, Allemng, James A., Hendler, Elsevier Science & Technology, “Semantic Web for the Working Ontologist”, 2008
- 10.** D.Fensel, Springer-Verlag Berlin and Heidelberg GmbH & Co. KG, “Ontologism”, 2003
- 11.** IGI Global, “Semantic Web Services”, 2007
- 12.** Shelley, Powers, O Reilly Media, Inc USA, “Practical RDF”, 2003
- 13.** Gabrick.K.A, Weiss.D.B J2EE and XML Development (2002)