

Τμήμα
Μηχανικών
Πληροφορικής τ.ε.

Τεχνολογικό Εκπαιδευτικό Ίδρυμα
Δυτικής Ελλάδας

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Σημασιολογικός παγκόσμιος ιστός και εφαρμογές του

Επιμέλεια / Συγγραφή

Αγλαΐνη Γεωργία ΑΜ: 1310

Επιβλέπων καθηγητής

Δρ. Τζήμας Ιωάννης

Επίκουρος Καθηγητής του Τμήματος Μηχανικών Πληροφορικής Τ.Ε.
του ΑΤΕΙ Δυτικής Ελλάδας

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Αντίρριο, __ / __ / _____

1. [Όνοματεπώνυμο, Υπογραφή]
2. [Όνοματεπώνυμο, Υπογραφή]
3. [Όνοματεπώνυμο, Υπογραφή]

*Στην μνήμη του παππού μου,
Κωνσταντίνου Σταθελάκου*

Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω την οικογένειά μου για την πολύτιμη βοήθειά τους και τη στήριξή τους όλα αυτά τα χρόνια η οποία φρόντισε για την καλύτερη δυνατή μόρφωσή μου. Επιπλέον θα ήθελα να εκφράσω τις ευχαριστίες μου για τους φίλους μου, ιδιαίτερα τον Κωνσταντίνο Τσούνη που ήταν δίπλα μου στο διάστημα της εκπόνησης της πτυχιακής μου εργασίας κάνοντας το κλίμα της συγγραφής της πιο χαρούμενο.

Επίσης θέλω να ευχαριστήσω θερμά τον καθηγητή μου Ιωάννη Τζήμα του τμήματος Μηχανικών Πληροφορικής Τ.Ε. Δυτικής Ελλάδας κυρίως για την εμπιστοσύνη που μου έδειξε και την υπομονή του κατά τη διάρκεια υλοποίησης της πτυχιακής μου εργασίας, όπως επίσης και για την πολύτιμη βοήθειά και καθοδήγησή του για την επίλυση διάφορων θεμάτων.

Απόσπασμα

Ο Σημασιολογικός Ιστός είναι «ένα πλέγμα διασυνδεδεμένων δεδομένων». Τα δεδομένα είναι δομημένα με κοινή μορφή και συνδέονται, έτσι ώστε να μπορούν να υποβάλλονται σε επεξεργασία άμεσα και έμμεσα από μηχανές. Τα δεδομένα μπορούν να διαμοιραστούν και να επαναχρησιμοποιηθούν από διαφορετικές εφαρμογές, οργανισμούς και κοινότητες.

Οι άνθρωποι μπορούν να δημιουργήσουν αποθήκες δεδομένων στο Σημασιολογικό Ιστό, να φτιάξουν λεξιλόγια, και να γράψουν κανόνες για τη διαχείριση τους. Εφαρμογές της γενικής ιδέας του Σημασιολογικού Ιστού στην έρευνα της βιομηχανίας, της βιολογίας και των επιστημών του ανθρώπου έχουν ήδη αποδείξει την εγκυρότητα και τη χρησιμότητά της.

Το 2001 Scientific American article by Berners- Lee , Hendler , και Lassila περιγράφει μια αναμενόμενη εξέλιξη του υφιστάμενου Web σε ένα Σημασιολογικό Ιστό . Το 2006 , ο Berners - Lee και οι συνεργάτες του ανέφεραν ότι : « Αυτή η απλή ιδέα ... παραμένει σε μεγάλο βαθμό μη πραγματοποιηθέντα " . Το 2013 , πάνω από τέσσερα εκατομμύρια domains Web περιέχονται στον Σημασιολογικού Ιστού σήμανσης .

Στις τεχνολογίες χειρισμού των Συνδεδεμένων Δεδομένων περιλαμβάνονται οι RDF, SPARQL, OWL, SKOS, κλπ. Το κίνημα του Σημασιολογικού Ιστού καθοδηγείται από το διεθνή οργανισμό τυποποίησης World Wide Web Consortium (W3C).

Abstract

The Semantic Web is "an interconnected data grid." The data is structured in a common format and are connected so that they can be processed directly and indirectly by machines. The data was divided and can be reused by different applications, organizations and communities.

According to the W3C, "The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries". The term was coined by Tim Berners-Lee for a web of data that can be processed by machines. While its critics have questioned its feasibility, proponents argue that applications in industry, biology and human sciences research have already proven the validity of the original concept.

The 2001 Scientific American article by Berners-Lee, Hendler, and Lassila described an expected evolution of the existing Web to a Semantic Web. In 2006, Berners-Lee and colleagues stated that: "This simple idea...remains largely unrealized". In 2013, more than four million Web domains contained Semantic Web markup.

On handling Linked Data technologies including RDF, SPARQL, OWL, SKOS, etc. The movement of the Semantic Web is guided by the international standards body World Wide Web Consortium (W3C).

Πίνακας περιεχομένων

Κεφάλαιο 1	9
1.1 Εισαγωγή.....	9
1.2 Σκοπός	10
1.3 Ιστορία.....	11
1.4 Προκλήσεις	11
1.5 Βιβλιογραφία- αναφορά πηγής.....	12
Κεφάλαιο 2	13
2.1 Γενικά για τον Σημασιολογικό Ιστό.....	13
2.2 Στόχος του Σημασιολογικού Ιστού	14
2.3 Σημασιολογικά Εργαλεία.....	15
2.4 Οι μηχανές αναζήτησης στο σημασιολογικό ιστό.....	16
2.5 Παγκόσμιος ιστός	17
2.6 Προσέγγιση σημασιολογικού ιστού.....	19
2.6.1 Περιορισμοί.....	19
2.7 Αναζήτηση.....	19
2.8 Στόχοι	20
2.9 Τεχνολογίες Σημασιολογικού Ιστού.....	21
2.9.1 Μεταδεδομένα.....	21
2.9.2 Οντολογίες.....	21
2.9.2 Σκοπός	21
2.9.4 Λογική.....	22
2.9.5 Πράκτορες.....	22
2.10 Πρότυπα	22
2.11 Βιβλιογραφία- αναφορά πηγής.....	23
Κεφάλαιο 3	24
3.1. Εισαγωγή στην HTML και XML.....	24
3.2 HTML.....	24
3.2.1 HTML tags.....	26
3.3 XML.....	26
3.4 Διαφορές HTML- XML	27
3.5 Ομοιότητες HTML- XML	27
3.6 Εισαγωγή στα DTD	27

3.6.1 XML DTD (document type definition)	28
3.6.2 XML document with an internal DTD	29
3.6.3 External DTD declaration	29
3.6.4 XML document with a reference to an external DTD.....	29
3.7 DTD Elements.....	30
3.7.1 Ορισμός ενός στοιχείου	30
3.8 DTD Attributes (χαρακτηριστικά).....	30
3.9 Βιβλιογραφία- αναφορά πηγής.....	33
Κεφάλαιο 4.....	35
4.1 Η γλώσσα σήμανσης XML	35
4.2 Μοντελοποίηση εγγράφων	37
4.3 XML έγγραφα	39
4.4 Επεξεργασία XML Εγγράφων.....	39
4.5 Η σύνταξη της XML.....	40
4.5.1 Η ανατομία XML εγγράφων	40
4.6 Δενδρική αναπαράσταση εγγράφου	41
4.7 XML declaration (δήλωση)	43
4.8 XML elements (στοιχεία).....	44
<i>Here is a table displaying correct an incorrect examples of names of xml elements:</i>	<i>44</i>
4.9 XML attributes(χαρακτηριστικά).....	45
4.10 XML elements vs. attributes.....	46
4.11 Avoid XML attributes?	46
4.12 Σύγκριση μεταξύ των τεχνολογιών μοντελοποίησης DTD και XML Schema	46
4.13 Βιβλιογραφία- αναφορά πηγής.....	47
Κεφάλαιο 5.....	48
5.1 XML path.....	48
5.1.1 Τι είναι η Xpath?	48
5.1.2 XPath path expressions	48
5.1.3 XPath is used in XSLT	48
5.1.4 XPath example	48
5.2 XSLT.....	50
5.2.1 Displaying XML with XSLT.....	50
5.2.2 XSLT example	50
5.3 Βιβλιογραφία- αναφορά πηγής.....	51

Κεφάλαιο 6	52
6.1 Περιγραφή μεταδεδομένων με την RDF	52
6.2 Η RDF/XML μορφή	53
6.3 Απλή αναπαράσταση με τη γλώσσα RDF-S	60
6.4 Σημασιολογία των RDF και RDF-S	62
6.5 Βιβλιογραφία- αναφορά πηγής	63
Κεφάλαιο 7	64
7.1 Εκφραστική αναπαράσταση με την OWL	64
7.2 Περιγραφές και αξιώματα κλάσεων	66
7.3 Περιγραφές και αξιώματα ιδιοτήτων	68
7.4 Άτομα	69
7.5 OWL Lite, OWL DL και OWL Full	70
7.6 RDF/XML σύνταξη	71
7.7 Βιβλιογραφία- αναφορά πηγής	72
Κεφάλαιο 8	73
8.1 Ενδεικτικές εφαρμογές των τεχνολογιών του σημασιολογικού ιστού	73
8.2 Οίκος Elsevier	74
8.2.1 Το πρόβλημα του Οίκου Elsevier	74
8.3 Συμβολή του σημασιολογικού ιστού (χρήση οντολογιών και θησαυρών)	74
8.4 Οντολογία EMTREE	76
8.5 Συμβολή της τεχνολογίας του σημασιολογικού ιστού (χρήση οντολογιών)	76
8.6 Τηλεκπαίδευση	76
8.7 Η συμβολή του σημασιολογικού ιστού	77
8.8 Καθοδήγηση από τον εκπαιδευόμενο	77
8.9 Ευέλικτη (μη γραμμική) πρόσβαση στη γνώση	77
8.10 Οντολογίες για τηλεκπαίδευση	78
8.10.1 Οντολογίες περιεχομένου	78
8.10.2 Παιδαγωγικές οντολογίες	78
8.10.2 Οντολογίες Δομής	78
8.11 Βιβλιογραφία- αναφορά πηγής	78
Κεφάλαιο 9	79
9.1 Συμπεράσματα- Επίλογος	79
9.2 Βιβλιογραφία- αναφορά πηγής	79
Βιβλιογραφία	80

Κεφάλαιο 1

1.1 Εισαγωγή

Ο Σημασιολογικός Ιστός αποτελεί το επόμενο μεγάλο βήμα εξέλιξης στο διαδίκτυο. Η ανάπτυξη του Παγκόσμιου Ιστού (World Wide Web) κατέστησε το διαδίκτυο προσβάσιμο σε εκατομμύρια χρήστες, επιτρέποντας την απρόσκοπτη 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 αρκετά χρόνια σε μία νέα κατεύθυνση εξέλιξης του ιστού, η οποία ονομάζεται "Σημασιολογικός 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. Ο Σημασιολογικός Ιστός θα επιτρέψει τον γρήγορο και ακριβή εντοπισμό πληροφοριών στον παγκόσμιο ιστό καθώς και την ανάπτυξη ευφών διαδικτυακών πρακτόρων οι οποίοι θα διευκολύνουν την επικοινωνία μεταξύ πληθώρας ετερογενών ηλεκτρονικών συσκευών με πρόσβαση στο διαδίκτυο.

Η κεντρική ιδέα είναι η ύπαρξη δεδομένων στον ιστό, οριζόμενων και συνδεδεμένων με έναν τρόπο που να επιτρέπει την αποδοτική ανακάλυψη, αυτοματοποίηση, ενσωμάτωση και επαναχρησιμοποίηση τους μέσω διαφόρων εφαρμογών χωρίς να είναι απαραίτητη η ανθρώπινη παρέμβαση. Ο Σημασιολογικός Ιστός περιέχει πόρους (resources) που αντιστοιχούν σε αντικείμενα υπερμέσων και ιστοσελίδες, αλλά και αντικείμενα όπως φυσικά πρόσωπα, οργανισμοί ή γεγονότα. Εμπεριέχει πολλά και διαφορετικά ήδη σχέσεων ανάμεσα στους πόρους, ενσωματώνοντας έννοιες και όρους κατανόησης του περιεχομένου. Έτσι ο Σημασιολογικός Ιστός δεν αποτελεί έναν ιστό από έγγραφα, αλλά έναν ιστό σχέσεων μεταξύ των πόρων, αναδεικνύοντας τα αντικείμενα του πραγματικού κόσμου. Η δομή του ιστού όπου τα αντικείμενα αναγνωρίζονται με μοναδικό και κοινώς αποδεκτό τρόπο μπορεί να αποκαλύψει τον ορισμό των αντικειμένων, τις πιθανές σχέσεις τους με οποιαδήποτε άλλα αντικείμενα, να επιτρέψει την εφαρμογή κανόνων λογικής και την εξαγωγή συμπερασμάτων. [1]

Ο Σημασιολογικός Ιστός (Web 3.0) είναι μια επέκταση του σημερινού Ιστού που θα φέρει δομή στο ουσιαστικό περιεχόμενο των ιστοσελίδων. Η λογική πίσω από αυτό είναι ότι η δημοσιευμένη πληροφορία θα περιέχει μετα-δεδομένα, τα οποία θα είναι κοινά για όλους, θα μπορούν να «κατανοούνται» και από μηχανές, οι οποίες θα βοηθήσουν στην καλύτερη συλλογή και επεξεργασία τους.

Ο Σημασιολογικός Ιστός βασίζεται σε τεχνολογίες που ήδη υπάρχουν (URI και XML) αλλά και σε νέες τεχνολογίες (RDF, RDFS, OWL, κα.), οι οποίες αναπτύσσονται με την βοήθεια της κοινότητας. Δεδομένου ότι ο νέος Ιστός σκοπεύει να είναι μια μεγάλη

βάση όπου δεδομένα από διαφορετικά πεδία θα συνδέονται μεταξύ τους, αναμένεται να παίξει μεγάλο ρόλο στη ζωή μας.

Μερικά από τα πεδία στα οποία αναμένεται να έχει την μεγαλύτερη επίδραση είναι στην υγεία, στην παιδεία και στις επιχειρήσεις. Υπάρχουν ήδη πολλές προσπάθειες από εταιρίες, ερευνητές και μη κερδοσκοπικές οργανώσεις για να παραγάγουν πρότυπα οντολογιών, κυρίως για τα παραπάνω πεδία, για να υπάρχουν κοινές γλώσσες και περισσότερα δεδομένα τα οποία να μπορούν να συνδυαστούν για καλύτερα αποτελέσματα.

Στην υγεία, γίνεται προσπάθεια για τη δημιουργία ενοποιημένων γλωσσών ιατρικής ορολογίας και υπηρεσίες που θα βοηθάνε το ιατρικό προσωπικό και θα κατευθύνουν τους καταναλωτές σε αξιόπιστες πληροφορίες υγείας σχετικά με την κατάστασή τους.

Στην εκπαίδευση, ο Σημασιολογικός Ιστός θα συμβάλει σημαντικά στην μάθηση κυρίως στον τρόπο αναζήτησης πληροφοριών, στην οργάνωση των αποτελεσμάτων και στη δημιουργία ενός προγράμματος μάθησης ειδικό για το καθένα.

Στον επιχειρηματικό τομέα, θα υπάρχει καλύτερη οργάνωση των εταιριών, καλύτερες εμπειρίες για τους χρήστες στις διαδικτυακές αγορές και καλύτερος συντονισμός μεταξύ διαφορετικών εταιριών.

Στην καθημερινότητά θα υπάρχουν επιδράσεις του Web 3.0 στα κοινωνικά δίκτυα και εικονικές κοινότητες. Θα υπάρχουν εφαρμογές, οι οποίες θα δίνουν περισσότερες, πιο έμπιστες, πληροφορίες και θα διευκολύνουν σημαντικά τις διαδικτυακές δραστηριότητες. [2]

1.2 Σκοπός

Ο κύριος σκοπός του σημασιολογικού ιστού είναι να εξελίξει τον τωρινό ιστό. Αυτό πραγματοποιείται κάνοντας του χρήστες να βρίσκουν, να μοιράζονται και να συνδυάζουν πληροφορίες πιο εύκολα. Οι άνθρωποι είναι ικανοί να χρησιμοποιούν τις δυνατότητες του ιστού, όπως να βρίσκουν μεταφράσεις για άλλες γλώσσες, να δεσμεύουν ένα βιβλίο από την βιβλιοθήκη, να ψάχνουν την χαμηλότερη τιμή για ένα DVD, κ.α. Παρόλα αυτά, οι μηχανές δεν μπορούν να χρησιμοποιήσουν αυτές τις δυνατότητες χωρίς την καθοδήγηση από κάποιον άνθρωπο, επειδή οι ιστοσελίδες είναι σχεδιασμένες να διαβάζονται από ανθρώπους και όχι από μηχανές. Το σημασιολογικό διαδίκτυο είναι μία μορφή πληροφορίας η οποία μπορεί να χρησιμοποιηθεί και από μηχανές, έτσι ώστε να αναλαμβάνουν οι ίδιοι οι υπολογιστές τις κουραστικές δουλειές που περιλαμβάνουν εύρεση, ένωση και επεξεργασία πάνω στις πληροφορίες που βρίσκονται στο διαδίκτυο.

Το σημασιολογικό διαδίκτυο, όπως αρχικά οραματίστηκε, είναι ένα σύστημα το οποίο επιτρέπει στις μηχανές να "καταλαβαίνουν" τα πολύπλοκα ανθρώπινα αιτήματα βασισμένα στο νόημά τους. Τέτοιου είδους "κατανόηση" απαιτεί από τις συσχετιζόμενες πηγές πληροφοριών να έχουν μία σημασιολογική δομή.

Ο Tim Berners-Lee αρχικά οραματίστηκε το σημασιολογικό διαδίκτυο ως εξής:
I have a dream for the Web [in which computers] become capable of analyzing all the data on the Web – the content, links, and transactions between people and computers. A "Semantic Web", which makes this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy and our daily lives will be handled by

machines talking to machines. The "intelligent agents" people have touted for ages will finally materialize.

Το σημασιολογικό διαδίκτυο θεωρείται ότι ολοκληρώνει διαφορετικά περιεχόμενα μεταξύ τους, πληροφορίες εφαρμογών και συστημάτων. Επίσης έχει εφαρμογές και στην δημοσίευση, στα blog και σε πολλούς άλλους τομείς. [2]

1.3 Ιστορία

Η ιδέα για τον *Σημασιολογικό Ιστό* πρωτοεμφανίστηκε στις αρχές της δεκαετίας του '60 και προέρχεται από τον γνωστικό επιστήμονα Άλαν Κόλλινς (Allan M. Collins), τον γλωσσολόγο M. Ross Quillian και την ψυχολόγο Elizabeth F. Loftus μέσα από πολλές δημοσιεύσεις, ως ένας τρόπος παρουσίασης της σημασιολογικής γνώσης. Επεκτείνει το διαδίκτυο, το οποίο αποτελείται από απλές σελίδες που μπορούν να αναγνωστούν μόνο από ανθρώπους, σε σελίδες που περιέχουν πληροφορίες ανάγνωσης για τις μηχανές (μετα- δεδομένα) και στο πως συνδέονται μεταξύ τους οι σελίδες, δημιουργώντας έτσι αυτόματες υπηρεσίες που χρησιμοποιούν τον Ιστό πιο έξυπνα και πραγματοποιούν εργασίες για τους χρήστες. Ο όρος "Σημασιολογικός Ιστός" εφευρέθηκε από τον Τιμ Μπέρνερς Λι, δημιουργό του παγκόσμιου ιστού (World Wide Web) και διευθυντή του World Wide Web Consortium ("W3C"), το οποίο επιβλέπει την ανάπτυξη των προτάσεων για τα στάνταρ του Σημασιολογικού Ιστού. Ο ίδιος ορίζει τον σημασιολογικό ιστό ως: " ένας ιστός από πληροφορίες ο οποίος μπορεί να επεξεργαστεί άμεσα και έμμεσα από μηχανές." [2]

1.4 Προκλήσεις

Κάποιες από τις προκλήσεις που έχει να αντιμετωπίσει το Σημασιολογικό Διαδίκτυο είναι η απεραντοσύνη, η ασάφεια, η αβεβαιότητα, η ασυνέπεια και η εξαπάτηση. Αυτόματα συστήματα λογικής θα πρέπει να αντιμετωπίσουν όλα αυτά τα προβλήματα έτσι ώστε να μπορέσουν να ανταπεξέλθουν σε όλα αυτά που υπόσχεται το Σημασιολογικό Διαδίκτυο.

- Απεραντοσύνη: Το παγκόσμιο διαδίκτυο περιέχει πάρα πολλές δισεκατομμύρια σελίδες. Ο ιατρικός όρος οντολογία SNOMED CT περιέχει από μόνος του 370.000 ονόματα κλάσεων και η υπάρχουσα τεχνολογία δεν έχει καταφέρει ακόμα να μειώσει όλους τους σημασιολογικούς διπλούς όρους. Οποιοδήποτε λογικό σύστημα θα πρέπει να έρθει αντιμέτωπο με έναν πραγματικά τεράστιο αριθμό από δεδομένα εισαγωγής.
- Ασάφεια: Όροι όπως "νέος" ή "ψηλός" είναι ασαφής. Αυτό προέρχεται από την ασάφεια των ερωτημάτων του χρήστη, από έννοιες που εμφανίζονται από παρόχους περιεχομένων, από ταίριασμα των όρων των ερωτημάτων με τους όρους των παρόχων και από την προσπάθεια συνδυασμού διαφορετικών βάσεων γνώσης με επικαλυπτόμενες αλλά διακριτά διαφορετικές έννοιες. Η ασφαλής λογική είναι η πιο κοινή τεχνική με την οποία αντιμετωπίζουμε την ασάφεια.

- Αβεβαιότητα: Τέτοιοι όροι είναι ακριβής με αβέβαιες τιμές. Για παράδειγμα, ένας ασθενής μπορεί να εμφανίσει ένα σύνολο από συμπτώματα τα οποία ανταποκρίνονται σε έναν αριθμό από διαφορετικές διαγνώσεις όπου η κάθε μία έχει διαφορετική πιθανότητα να είναι αληθής. Οι πιθανοτικές λογικές τεχνικές εφαρμόζονται γενικότερα για να διευθετήσουν την αβεβαιότητα.
- Ασυνέπεια: Αυτές είναι λογικές αντιφάσεις οι οποίες εμφανίζονται εντελή κατά την διάρκεια ανάπτυξης μεγάλων οντολογιών ή και όταν οντολογίες από διαφορετικές πηγές ενώνονται. Ο παραγωγικός συλλογισμός _αποτυγχάνει_ καταστροφικά όταν αντιμετωπίζει την ασυνέπεια.
- Εξαπάτηση: Αυτό συμβαίνει όταν ο δημιουργός της πληροφορίας εσκεμμένα παραπληροφορεί τον καταναλωτή για την πληροφορία. Οι κρυπτογραφικές τεχνικές είναι αυτές που χρησιμοποιούνται για να μειώσουν αυτήν την απειλή.

Η λίστα με τις προκλήσεις είναι περισσότερο ενδεικτική παρά εξαντλητική και εστιάζει στις προκλήσεις για τα στάδια της "ενωτικής λογικής" και της "απόδειξης" του Σημασιολογικού διαδικτύου. Το γκρουπ του κονσόρτσιουμ του παγκόσμιου ιστού για την αβέβαιη λογική του παγκόσμιου ιστού μαζεύει αυτά τα προβλήματα κάτω από την ομπρέλα της "αβεβαιότητας". Πολλές από τις τεχνικές που αναφέρονται εδώ θα χρειαστούν επεκτάσεις στην γλώσσα οντολογίας του ιστού. Αυτή είναι μία περιοχή ενεργής έρευνας. [2]

1.5 Βιβλιογραφία- αναφορά πηγής

[1]: πηγή από διαδίκτυο

<http://lpis.csd.auth.gr/mtpx/sw/>

[2]: πηγή από το διαδίκτυο

https://el.wikipedia.org/wiki/%CE%A3%CE%B7%CE%BC%CE%B1%CF%83%CE%B9%CE%BF%CE%BB%CE%BF%CE%B3%CE%B9%CE%BA%CF%8C%CF%82_%CE%99%CF%83%CF%84%CF%8C%CF%82

Κεφάλαιο 2

2.1 Γενικά για τον Σημασιολογικό Ιστό

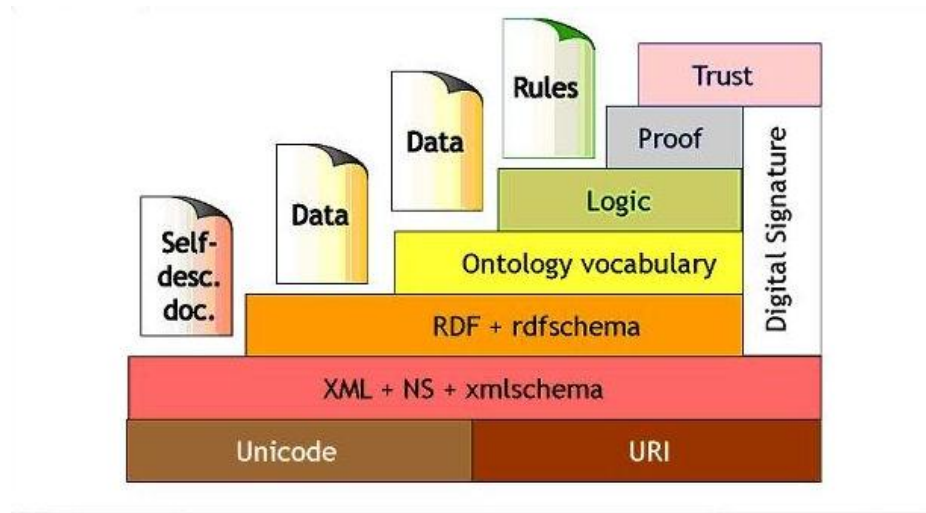
Ένα πολύ μεγάλο μέρος της ανθρώπινης επικοινωνίας μέσω υπολογιστή καλύπτεται σήμερα από το Διαδίκτυο και τη βασικότερη εφαρμογή του, τον παγκόσμιο ιστό (World Wide Web, WWW). Παρόλα αυτά το WWW έχει ένα βασικό μειονέκτημα: βασίζεται στην συντακτική περιγραφή του περιεχομένου (content). Έτσι οι ιστοσελίδες περιέχουν κείμενα ή άλλου είδους δημοσιευμένα περιεχόμενα τα οποία είναι κατανοητά μόνο από τους ανθρώπους. Το χαρακτηριστικό αυτό του WWW θεωρείται σαν ένα μειονέκτημα του, αφού επιτρέπει την αυτόνομη επεξεργασία περιεχομένου στο Διαδίκτυο. Τέτοια επεξεργασία προϋποθέτει μια καταλληλότερη περιγραφή του δημοσιευμένου περιεχομένου και αλγόριθμους που να προσδίδουν την επιθυμητή “ευφυΐα” στο υπολογιστικό περιβάλλον. Μια τέτοια περιγραφή θα πρέπει εκτός από το βασικό περιεχόμενο των ιστοσελίδων να περιλαμβάνει και δεδομένα που προορίζονται αποκλειστικά για τους υπολογιστές και που θα περιγράφουν τη σημασιολογία (semantic) του περιεχομένου. Όσον αφορά στους αλγόριθμους εφόσον υπάρχει η σωστή περιγραφή θα αναπτύσσονται ανάλογα με την περίπτωση και τη χρήση της πληροφορίας. Τον παραπάνω περιορισμό του WWW καθώς και τη λύση του οραματίστηκε ο πρώτος ιδρυτής του, Tim Berner Lee, ο οποίος ονόμασε αυτό το εξελικτικό στάδιο του WWW σαν “Σημασιολογικό Ιστό” (Semantic Web).

Ο Σημασιολογικός Ιστός έχει σαν στόχο να προσδώσει δομή στο νόημα του περιεχομένου των ιστοσελίδων, δημιουργώντας ένα περιβάλλον όπου οι πράκτορες λογισμικού (software agents) περιπλανώμενοι από σελίδα σε σελίδα θα μπορούν να εκτελούν προηγμένες εργασίες για τους χρήστες. Ο Σημασιολογικός Ιστός δεν είναι ένας διαφορετικός ιστός αλλά μια επέκταση του υπάρχοντος Παγκόσμιου Ιστού, στην οποία η πληροφορία αποκτά καλά ορισμένο νόημα, δίνοντας τη δυνατότητα για πιο αποτελεσματική συνεργασία ανάμεσα στον άνθρωπο και στον υπολογιστή, αφού πλέον θα υπάρχει μια κοινή γλώσσα ανάμεσα τους (η σημασιολογική περιγραφή). Θα μπορούσαμε να πούμε ότι ο Σημασιολογικός Ιστός αποτελεί τη βάση για μια πλήρως κατανοητή μορφή τεχνητής νοημοσύνης. Όπως είναι γνωστό, η τεχνητή νοημοσύνη ασχολείται κυρίως με δύο θέματα: την αναπαράσταση γνώσης (knowledge representation) και τις μεθόδους αναζήτησης και συμπερασμού (reasoning) (I.Vlahavas, P.Kefalas, N.Bassiliades, F.Kokkoras, I.Sakellariou. Artificial Intelligence – 2dr Edition 2005).

Έτσι για να λειτουργήσει ο Σημασιολογικός Ιστός θα πρέπει οι υπολογιστές να έχουν πρόσβαση σε δομημένες βάσεις γνώσης και σε κανόνες συμπερασμού τους οποίους να μπορούν να χρησιμοποιήσουν για να διενεργήσουν αυτόματη συλλογιστική. Οι υπάρχουσες τεχνολογίες αναπαράστασης γνώσης έχουν σημειώσει κάποια πρόοδο αλλά δεν είναι ακόμη ικανές να υλοποιήσουν τις προσδοκίες της τεχνητής νοημοσύνης. Οι κλασικές μέθοδοι αναπαράστασης γνώσης προϋποθέτουν μια κοινή, παγκόσμια βάση γνώσης και κανόνων έτσι ώστε όλοι να χρησιμοποιούν τους ίδιους κανόνες. Από την άλλη πλευρά ο Σημασιολογικός Ιστός δεν επιβάλλει την

απόλυτη συμφωνία μεταξύ των διαφόρων συστημάτων που θα διασυνδέει. Η μεγαλύτερη συμβολή της πρωτοβουλίας του Σημασιολογικού Ιστού μέχρι σήμερα είναι η προτυποποίηση που παρέχει σε γλώσσες και τεχνολογίες.

Κατά τον Tim Berner Lee της W3C ο σημασιολογικός ιστός θα έχει αυτή τη μορφή:



Σχήμα 1: Μορφή σημασιολογικού ιστού

Απ' ότι μπορούμε να δούμε από το παραπάνω σχήμα ο σημασιολογικός ιστός σύμφωνα με το όραμα του Tim Berner Lee έχει την παρακάτω μορφή. Στο κατώτερο επίπεδο βρίσκονται το Unicode που εξασφαλίζει την επικοινωνία ανάμεσα σε διαφορετικές γλώσσες και τα URI (Uniform Resource Identifier), οι "διευθύνσεις" δηλαδή των πόρων. Στο αμέσως επόμενο βρίσκεται η XML, η NS και το XML Schema ως κύρια γλώσσα έκφρασης στον ιστό. Ακολουθεί η RDF (Resource Description Framework) και το RDF Schema που είναι η κύρια γλώσσα μεταδεδομένων. Έπειτα είναι το λεξιλόγιο οντολογιών. Στη συνέχεια το λογικό επίπεδο ακολουθούμενο από το επίπεδο απόδειξης και τέλος το επίπεδο αξιοπιστίας. Η κύρια γλώσσα έκφρασης στον ιστό είναι η XML (eXtensible Markup Language). Όμως δεν καταφέρνει να αποτελέσει μια απόλυτη λύση για το σημασιολογικό ιστό, αυτό που κάνει είναι να παρουσιάσει κάποιες σημασιολογικές ιδιότητες μέσα από τη συντακτική δομή της.

Αυτό εξαρτάται όμως από τον ορισμό κειμένου που δίνει κάποιο DTD (Document Type Definition) για το συγκεκριμένο αρχείο. Επομένως, παρόλο που μπορεί κανείς να εξάγει κάποιες σημασιολογικές έννοιες από τις ετικέτες XML, η σημασιολογική έννοια κάθε ετικέτας δεν ορίζεται έξω από ένα DTD. Για την ανάπτυξη λοιπόν ενός σημασιολογικού ιστού είναι απαραίτητο οι πόροι στον ιστό να αναπαρίστανται ή να σχολιάζονται με δομημένες περιγραφές του περιεχομένου και των σχέσεων τους, κατανοητές από Η/Υ. Αυτό είναι εφικτό με τη χρήση λεξιλογίων και κατασκευασμάτων που δηλώνονται ρητά σε μια οντολογία περιοχής. [3]

2.2 Στόχος του Σημασιολογικού Ιστού

Το σημασιολογικό διαδίκτυο φιλοδοξεί να επιτρέψει τόσο στους χρήστες όσο και στα πληροφοριακά συστήματα να μπορούν να κατανοούν και να επεξεργάζονται λογικά τις ίδιες πληροφορίες και δεδομένα. Προφανώς αυτή η προοπτική θα ανοίξει νέους

συναρπαστικούς ορίζοντες, αφού τα πληροφοριακά συστήματα θα μπορούν πέρα από το να αντλούν μεγάλες ποσότητες πληροφοριών, όπως κάνουν σήμερα, επιπλέον θα τις επεξεργάζονται λογικά και θα εξάγουν λογικά συμπεράσματα. Τα αποτελέσματα μιας τέτοιας διεργασίας θα μπορούν να δημοσιεύονται και αυτά στο διαδίκτυο έτσι ώστε να μπορούν να χρησιμοποιηθούν από τρίτους, αυξάνοντας έτσι τη συνολική γνώση και αξία του σημασιολογικού ιστού.

Προχωρώντας ένα βήμα πιο πέρα, θα δημιουργηθούν και θα λειτουργούν έξυπνες υπηρεσίες οι οποίες θα μπορούν να εντοπίζουν και να χρησιμοποιούν η μια την άλλη, να συνεργάζονται μεταξύ τους χωρίς να έχουν εξαρχής προετοιμαστεί γι' αυτό, και όλα αυτά αυτόματα χωρίς την ανθρώπινη παρέμβαση. [3]

2.3 Σημασιολογικά Εργαλεία

Τα σημασιολογικά εργαλεία είναι κάποια εργαλεία που προσπαθούν να καλύψουν το κενό στην επικοινωνία ανθρώπου μηχανής. Ενώ οι περισσότερες από τις τρέχουσες μορφές περιεχομένου του ιστού σχεδιάζονται να είναι κατανοητές από τον άνθρωπο, δεν είναι κατανοητές από Η/Υ. Το περιεχόμενο του σημασιολογικού ιστού είναι σχεδιασμένο έτσι ώστε να το κατανοούν τόσο ο άνθρωπος όσο και οι Η/Υ. Για να επιτευχθεί αυτό γίνεται μια προσπάθεια ανάπτυξης των εννοιών που να μπορεί να τις “καταλάβει” ένας Η/Υ. Αυτό με λίγα λόγια γίνεται με την οργάνωση της επιστημονικής γνώσης και ορολογίας σε μια τέτοια μορφή που να μπορεί με κάποιες τεχνικές να την κατανοεί ένας Η/Υ. Τέτοια εργαλεία είναι:

- **Lists:** Λίστες όρων όπως authority files, γλωσσάρια και λεξικά
- **MetTermadata-like models:** όπως κατάλογοι και γεωγραφικά λεξικά
- **Classification and categorization schemes:** Σχέδια ταξινόμησης και κατηγοριοποίησης όπως θεματικές επικεφαλίδες, σχήματα ταξινόμησης και κατηγοριοποίησης
- **Relationship models:** Σχεσιακά Μοντέλα όπως θησαυροί, σημασιολογικά δίκτυα, χάρτες εννοιών και οντολογίες
- **Metadata content standards:** Πρότυπα για το περιεχόμενο των μεταδεδομένων και ειδικά αυτά που ασχολούνται με την αντιπροσώπευση της γνώσης
- **Domain-specific content mark-up languages:** όπως DTD (Document Type Definition) και τα σχήματα XML
- **General knowledge representation languages:** Γενικές γλώσσες αντιπροσώπευσης της γνώσης, όπως ο υπολογισμός του πρώτου

κατηγορήματος διαταγής, λογικές περιγραφής, εννοιολογικές γραφικές παραστάσεις, γενικά συστήματα πλαισίων και οι αντικειμενοστραφείς γλώσσες διαμόρφωσης.

Πολλά από τα σημασιολογικά εργαλεία που αναφέρονται παραπάνω θεωρούνται συμβατικά συστήματα οργάνωσης της γνώσης όπως οι Term lists, Classification and categorization schemes και κάποια από τα σχεσιακά μοντέλα. Η συμβατική αναπαράσταση εννοιών έχει αξία σε μαθησιακά περιβάλλοντα, αφού παρέχουν για παράδειγμα πρόσβαση σε κείμενα και εφημερίδες, για τα οποία υπάρχει αναπαράσταση εννοιών βασισμένη σε όρους. Πολλά από αυτά τα εργαλεία έχουν μια σημαντική αδυναμία σε περιβάλλον Ψηφιακών Βιβλιοθηκών λόγω των σχετικά αδύναμων δομών τους. Δηλαδή καταλήγουν τελικά να παίρνουν τη μορφή γλωσσικών όρων. Πολλά από τα σημαντικά χαρακτηριστικά των επιστημονικών εννοιών όμως, όπως η αναπαράστασή τους, οι έννοιες, οι ιδιότητες, οι σχέσεις με άλλες έννοιες δεν μπορούν να αναπαρασταθούν με τόσο απλούς γλωσσικούς όρους. Ο σημασιολογικός ιστός θα χρησιμοποιεί σημασιολογικά εργαλεία, όπως οντολογίες για να περιγράψει τους διάφορους πόρους στο δίκτυο. Έτσι, ενώ μέχρι σήμερα το δίκτυο ήταν ένας χώρος που περιέχει μεγάλη ποσότητα ανοργάνωτης, κατά κύριο λόγο, πληροφορίας, πλέον ο ιστός θα γίνει φορέας γνώσης. Αυτό θα επιτευχθεί με την λογική δόμηση της πληροφορίας, του περιεχομένου δηλαδή του ιστού με ένα σημασιολογικό τρόπο, γεγονός που θα επιφέρει πολλές αλλαγές σε σχέση με τον τρόπο που λειτουργεί τώρα ο ιστός. [3]

2.4 Οι μηχανές αναζήτησης στο σημασιολογικό ιστό

Στον τομέα των μηχανών αναζήτησης ο σημασιολογικός ιστός θα προσφέρει πολλές καινοτομίες. Με την καλύτερη οργάνωση της πληροφορίας μέσω θησαυρών και οντολογιών μπορούν να παρέχουν καλύτερα και ποιοτικότερα αποτελέσματα χωρίς να φορτώνουν τόσο πολύ το δίκτυο. Επίσης ο χρήστης θα είναι πλέον ικανός να διαμορφώσει πιο ελεύθερα ερωτήματα, όχι απαραίτητα λέξεις – κλειδιά ή με τελεστές Boolean και η μηχανή αναζήτησης να του φέρει ικανοποιητικά αποτελέσματα αφού θα μπορεί να τα κατανοήσει. Αυτό θα επιτευχθεί προφανώς με τη χρήση της XML και των οντολογιών, με τις οποίες θα μπορεί η μηχανή να κατανοήσει και να αξιολογήσει λογικά το περιεχόμενο μιας σελίδας. Ακόμη θα είναι σε θέση να κατανοεί τα ερωτήματα σημασιολογικά και όχι απλά σαν λέξεις – κλειδιά. Αυτό σημαίνει ότι αν για παράδειγμα έχουμε ένα ερώτημα “return all the reviewers for book «The Semantic Web: an introduction»” η σημασιολογική μηχανή θα μας επιστρέψει ένα σύνολο από reviewers για το συγκεκριμένο βιβλίο αντί να θεωρήσει κάθε όρο του ερωτήματος ως λέξη – κλειδί και να μας φέρει ως αποτέλεσμα κάθε σελίδα που περιέχει τους όρους αυτούς και τις ρίζες τους. Επομένως μιλάμε για μεγαλύτερη ακρίβεια στην ανάκτηση, δηλαδή για πιο ποιοτικά αποτελέσματα. Αυτή τη στιγμή οι μηχανές αναζήτησης στον ιστό έχουν δύο προσεγγίσεις, αυτή των μηχανών μεγάλης κλίμακας που στηρίζονται σε έξυπνους πράκτορες και αυτή των μηχανών μικρής κλίμακας που στηρίζονται στην ανασκόπηση σελίδων.

Στην πρώτη περίπτωση έχουμε μεγάλη ανάκτηση αλλά μικρή ακρίβεια. Αυτό οφείλεται στο γεγονός ότι στην ουσία έξυπνοι πράκτορες συλλέγουν ιστοσελίδες και τις αποθηκεύουν σε μια μεγάλη κεντρική βάση δεδομένων (οι πράκτορες μπορούν να συλλέξουν σχεδόν όλες τις σελίδες του web) αλλά ο μεγάλος αριθμός των ιστοσελίδων μειώνει τον αριθμό των σχετικών αποτελεσμάτων, επομένως η ακρίβεια είναι σε χαμηλά επίπεδα.

Στην δεύτερη περίπτωση, ακριβώς επειδή έχει γίνει ανασκόπηση της κάθε σελίδας, τα αποτελέσματα φανερώνουν μεγάλη σχετικότητα, άρα υψηλή ακρίβεια. Όμως, όπως είναι λογικό, αυτή η δουλειά γίνεται σε περιορισμένο όγκο ιστοσελίδων, επομένως έχουμε μικρό ποσοστό ανάκτησης.

Και τα δύο είδη μηχανών αναζήτησης είναι βασισμένα σε λέξεις – κλειδιά, πράγμα που σημαίνει ότι και στις δύο περιπτώσεις παρουσιάζεται πρόβλημα, όσον αφορά στο ερώτημα που διατυπώνεται, λόγω των πολλαπλών σημασιών που μπορεί να έχει μια λέξη. Έχουν αναπτυχθεί διάφορες τεχνικές για να ξεπεράσουν αυτό το πρόβλημα με αλγόριθμους που προσπαθούν να καθορίσουν τις συνώνυμες λέξεις για παράδειγμα, αλλά υπάρχει πάντα το πρόβλημα ότι πολλές λέξεις με άσχετη σημασία μπορούν να προέρχονται από κοινή ρίζα ή κοντινές στη σημασία λέξεις να έχουν εντελώς άσχετη ρίζα.

Η λύση σε αυτό το πρόβλημα είναι προφανώς να υπάρχουν semantics στις ιστοσελίδες ώστε να καθορίζεται το νόημά τους, αλλά δεν έχει επικρατήσει κάποια γλώσσα που να καθιερώνει κάποιες σταθερές HTML ετικέτες. Η XML φαίνεται πολλά υποσχόμενη αφού κρατάει το περιεχόμενο, την δομή και την παρουσίαση χωριστά και είναι πολύ πιο επαρκής για την αναπαράσταση γνώσης. Η συντακτική δομή της όμως την περιορίζει καθώς μπορεί να παρουσιάσει μόνο κάποιες σημασιολογικές ιδιότητες. Σε αυτόν τον τομέα θα εισχωρήσουν οι οντολογίες για την πιο κατανοητή περιγραφή των ιστοσελίδων.

Οι τυπικές χρήσεις του Ιστού βασίζονται στην αναζήτηση, επεξεργασία και χρησιμοποίηση των πληροφοριών:

- Αναζήτηση άλλων ατόμων με κοινά χαρακτηριστικά και επαφή μαζί τους
- Αναζήτηση προϊόντων σε καταλόγους e-shops
- Αναζήτηση πληροφοριών στον Παγκόσμιο Ιστό

Οι δραστηριότητες αυτές δεν υποστηρίζονται επαρκώς από τα σύγχρονα εργαλεία λογισμικού. Βασικό εμπόδιο είναι το νόημα του περιεχομένου του Ιστού που δεν είναι προς το παρόν προσπελάσιμο από υπολογιστές (machine accessible).

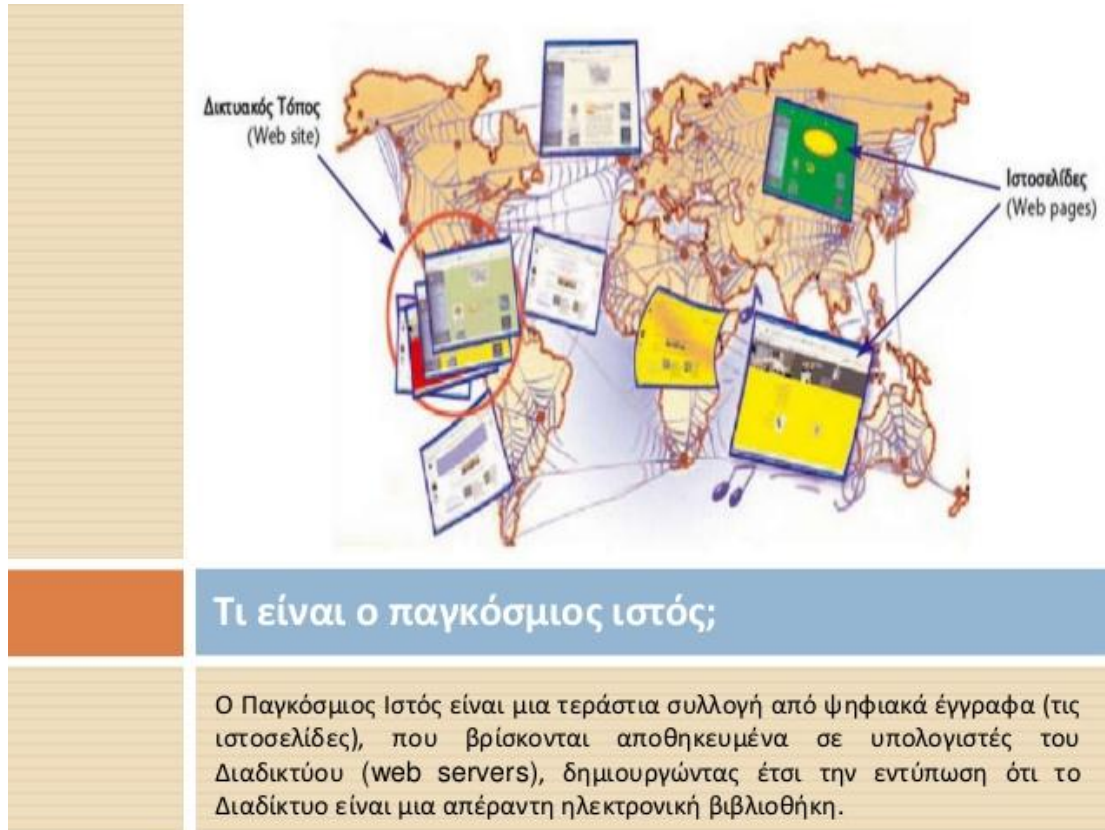
Το περιεχόμενο των HTML ιστοσελίδων είναι εύκολα κατανοητό από τον άνθρωπο, αλλά όχι από τα προγράμματα. [3]

2.5 Παγκόσμιος ιστός

Ο Παγκόσμιος Ιστός είναι μια τεράστια συλλογή από ψηφιακά έγγραφα (τις ιστοσελίδες), που βρίσκονται αποθηκευμένα σε υπολογιστές του Διαδικτύου.

Παγκόσμιος Ιστός είναι το δίκτυο των συνδεδεμένων υπολογιστών και δικτύων σε παγκόσμια κλίμακα, το οποίο χρησιμοποιεί συγκεκριμένη ομάδα πρωτοκόλλων επικοινωνίας, γνωστή ως "http". Κάθε δίκτυο-δομική μονάδα του διαδικτύου αποτελείται από συνδεδεμένους υπολογιστές σε τοπικό επίπεδο, για παράδειγμα το δίκτυο υπολογιστών των κεντρικών γραφείων μιας εταιρίας. Αυτά τα δίκτυα με τη σειρά τους συνδέονται σε ευρύτερα δίκτυα, όπως εθνικά και υπερεθνικά. Το ευρύτερο

δίκτυο στον κόσμο λέγεται παγκόσμιος Ιστός το οποίο είναι μοναδικό (δηλαδή δεν υπάρχουν παραπάνω από ένα δίκτυα υπολογιστών παγκόσμιας κλίμακας), και συμπεριλαμβάνεται τόσο τα γήινα δίκτυα, όσο και τα δίκτυα των δορυφόρων της και άλλων διαστημικών συσκευών που είναι συνδεδεμένα σε αυτό. [4]



Σχήμα 2: Ο Παγκόσμιος Ιστός αποτελείται από μια τεράστια συλλογή ηλεκτρονικών [5]

Μια ιστοσελίδα μπορεί να περιλαμβάνει κείμενο, εικόνες, ήχο, ή βίντεο και μεταφέρεται στο Διαδίκτυο ταξιδεύοντας από κάποιον κεντρικό υπολογιστή ,στον οποίο είναι αποθηκευμένη , στον υπολογιστή μας. Κάθε ιστοσελίδα έχει την δική της διεύθυνση στον Παγκόσμιο Ιστό και αν θέλουμε να επισκεφτούμε την ιστοσελίδα αυτή πρέπει να ξέρουμε την διεύθυνση της. Η διεύθυνση αυτή καλείται URL(Uniform Resource Locator- Ενιαίος Προσδιοριστής Πόρου) ή πιο απλά διεύθυνση ιστοσελίδας. Μια διεύθυνση ιστοσελίδας είναι μοναδική και έχει την μορφή:



- **http:** σύνολο κανόνων στους οποίους υπακούει το λογισμικό πλοήγησης
- **www: World Wide Web** δηλώνει ότι πρόκειται για διεύθυνση στον Παγκόσμιο Ιστό και είναι προαιρετικό.
- **internet όπως στο παράδειγμα ή κάποιο άλλο όνομα:** έχει σχέση με το όνομα του ιδιοκτήτη του δικτυακού τόπου.
- **com:** κατάληξη που δηλώνει την περιοχή στην οποία ανήκει η ιστοσελίδα :gr-> Ελλάδα, it->Ιταλία, de->Γερμανία ή τον τύπο της ιστοσελίδας όπως στο παράδειγμα, com->εμπορική, org-> οργανισμός. [6]



2.6 Προσέγγιση σημασιολογικού ιστού

Σκοπός είναι οι ιστοσελίδες να μπορούν να μεταφέρουν οι ίδιες πληροφορίες σχετικά με το περιεχόμενό τους. [7]

2.6.1 Περιορισμοί

Το διαδικτυακό περιεχόμενο είναι κατανοητό από τον άνθρωπο αλλά όχι από τους υπολογιστές (προγράμματα), άρα είναι δύσκολα επεξεργάσιμο για την εξαγωγή χρήσιμης γνώσης. Αδυναμία αυτοματοποίησης διαδικασιών αποδοτικής αναζήτησης, επεξεργασίας και ενοποίησης δεδομένων. Απαιτείται ανθρώπινη προσπάθεια και χρόνος για τον εντοπισμό των σχετικών πληροφοριών από τα ανακτηθέντα έγγραφα που επιστρέφει ένα ερώτημα (query) σε μια μηχανή αναζήτησης. Τεράστιος όγκος διαθέσιμης πληροφορίας, που συχνά προέρχεται από ετερογενείς πηγές. Μη-δομημένη πληροφορία: Ο Παγκόσμιος Ιστός περιορίζεται ουσιαστικά στην παρουσίαση της πληροφορίας χωρίς να μεταφέρει τη δομή της(ακόμα και αν το διαδικτυακό περιεχόμενο προέρχεται από κάποια Βάση Δεδομένων παρουσιάζεται συνήθως χωρίς την αρχική δομική πληροφορία που μπορεί να βρεθεί στη βάση).

2.7 Αναζήτηση

Βασικά εργαλεία χρήσης του Παγκόσμιου Ιστού: Μηχανές αναζήτησης (search engines) που βασίζονται σε λέξεις-κλειδιά π.χ. Google, Yahoo κ.λπ. Προβλήματα μηχανών αναζήτησης:

- Υψηλή ανάκληση, μικρή ακρίβεια: όταν ένα ερώτημα σε μια μηχανή αναζήτησης επιστρέφει ένα μεγάλος όγκο αποτελεσμάτων, καθίσταται δύσκολος ο εντοπισμός της περισσότερο σχετικής πληροφορίας.
- Χαμηλή ή καθόλου ανάκληση: όταν ένα ερώτημα σε μια μηχανή αναζήτησης επιστρέφει ελάχιστα ή και καθόλου αποτελέσματα.
- Ευαισθησία στο λεξιλόγιο: τα σημασιολογικά παρόμοια ερωτήματα δεν επιστρέφουν παρόμοια αποτελέσματα αναζήτησης (ασυνέπειες στην ορολογία, διφορούμενες λέξεις-κλειδιά).
- Δεν λαμβάνουν υπόψη το υπονοούμενο περιεχόμενο [τα αποτελέσματα είναι μεμονωμένες ιστοσελίδες, άρα αν χρειαζόμαστε πληροφορίες που έχουν διασκορπιστεί σε διάφορα έγγραφα, θα πρέπει να κάνουμε πολλά ερωτήματα (queries) προκειμένου να συλλέξουμε τις σχετικές πληροφορίες. Αυτό συνεπάγεται μη-αυτόματη εξαγωγή σχετικής γνώσης.]
- Είναι απομονωμένες εφαρμογές: τα αποτελέσματα δεν είναι άμεσα προσπελάσιμα από άλλα εργαλεία λογισμικού.

Όλα τα παραπάνω προβλήματα ξεκινούν από το γεγονός πως το νόημα του περιεχομένου του Ιστού δεν είναι προσπελάσιμο από υπολογιστές. Έτσι η ερμηνεία των προτάσεων και η εξαγωγή χρήσιμων πληροφοριών είναι πολύ περιορισμένη. Υπάρχουν εργαλεία που είναι σε θέση να ανακτήσουν κείμενα, να τα χωρίσουν σε τμήματα, να ελέγξουν τη σύνταξη και την ορθογραφία, αλλά δε μπορούν να αντιληφθούν την ερμηνεία των προτάσεων, έτσι ώστε να εξάγουν χρήσιμες πληροφορίες για τους χρήστες. [7]

2.8 Στόχοι

Αναπαράσταση του διαδικτυακού περιεχομένου σε μορφή που είναι ευκολότερα επεξεργάσιμη από υπολογιστές (machine- processable) και χρήση νοημόνων τεχνικών για την εκμετάλλευση αυτών των αναπαραστάσεων. Η γνώση θα είναι οργανωμένη σε εννοιολογικές περιοχές ανάλογα με τη σημασία της. Έξυπνη αναζήτηση και ανάκτηση περιεχομένου (η αναζήτηση με λέξεις-κλειδιά θα αντικατασταθεί από απαντήσεις ερωτημάτων(queries)). Έτσι η αιτηθείσα γνώση θα ανακτάται, θα εξάγεται και θα παρουσιάζεται με τρόπο φιλικό προς το χρήστη. Υποστήριξη ερωτημάτων σε περισσότερα από ένα έγγραφα. Η δημιουργία ενός Παγκόσμιου μέσου για την ανταλλαγή δεδομένων. Έτσι, οι διαφορές στην ορολογία θα επιλύονται με τη χρήση πρότυπων αφηρημένων μοντέλων πεδίου (abstract domain models). Αυτοματοποιημένα εργαλεία θα υποστηρίζουν τη συντήρηση της γνώσης, ελέγχοντας για ασυνέπειες και θα εξάγουν νέα γνώση. Πιο «ευφυείς» υπηρεσίες και εφαρμογές (προσωπικοί πράκτορες). Η ομαλή διασύνδεση της διαχείρισης των προσωπικών πληροφοριών, η ολοκλήρωση επιχειρηματικών εφαρμογών και η παγκόσμια ανταλλαγή εμπορικών, επιστημονικών και πολιτιστικών δεδομένων. Διαλειτουργικότητα μεταξύ ετερογενών συστημάτων και ενοποίηση δεδομένων. [7]

2.9 Τεχνολογίες Σημασιολογικού Ιστού

Βασικές τεχνολογίες για να επιτευχθούν οι προηγούμενοι στόχοι:

- Μεταδεδομένα (Metadata)
- Οντολογίες (Ontologies)
- Λογική (Logic)
- Πράκτορες (Agents) [7]

2.9.1 Μεταδεδομένα

Το περιεχόμενο του Ιστού είναι μορφοποιημένο για ανθρώπους παρά για προγράμματα. Δεν παρέχουν καμία πληροφορία σχετικά με τη δομή και τη σημασιολογία του περιεχομένου. Ακόμη και προχωρημένα συστήματα (π.χ. ευφυείς πράκτορες) δε μπορούν να έχουν επαρκή πρόσβαση. Τα μεταδεδομένα συλλαμβάνουν μέρος του νοήματος των δεδομένων, τη σημασιολογία τους. Έτσι περνάμε από την HTML στην XML, το πρώτο βήμα στην πραγματοποίηση του Σημασιολογικού Ιστού. Ο όρος μεταδεδομένα (Metadata) αναφέρεται σε τέτοιου είδους πληροφορίες: δεδομένα σχετικά με δεδομένα. Αυτή η αναπαράσταση είναι πολύ πιο εύκολα επεξεργάσιμη και κατανοητή από υπολογιστές.

Παράδειγμα: Έστω πως ψάχνω στο Διαδίκτυο να βρω πληροφορίες για ένα κέντρο φυσιοθεραπείας το οποίο είναι κοντά στο σπίτι μου και οι ώρες λειτουργίας του συμφωνούν με το πρόγραμμα μου. [7]

2.9.2 Οντολογίες

Μια οντολογία είναι μια ρητά και τυπικά εκφρασμένη περιγραφή κάποιων εννοιών (κλάσεις αντικειμένων) που ανήκουν σε ένα πεδίο ενδιαφέροντος, καθώς και των σχέσεων που υφίστανται μεταξύ τους. [7]

2.9.2 Σκοπός

Αποδοτική ανταλλαγή πληροφοριών μεταξύ ανθρώπων, υπολογιστών, ανθρώπων και υπολογιστών. Καθιστά την γνώση επεξεργάσιμη από τους υπολογιστές. Επικοινωνία μεταξύ πρακτόρων λογισμικού (software agents). Διαλειτουργικότητα. Επαναχρησιμοποίηση γνώσης πεδίου. Διευκόλυνση της ανάλυσης της γνώσης πεδίου. Οι σχέσεις μιας οντολογίας συνήθως περιλαμβάνουν ιεραρχίες κλάσεων.

Στον παγκόσμιο ιστό οι οντολογίες ορίζουν την τυπική σημασιολογία της πληροφορίας, διευκολύνουν την επεξεργασία της πληροφορίας από τον Η/Υ και παρέχουν μια κοινή αντίληψη ενός πεδίου, η οποία είναι απαραίτητη για την εξάλειψη των διαφορών στην ορολογία => σημασιολογική λειτουργικότητα.

Είναι ιδιαίτερα χρήσιμες οι οντολογίες για τη βελτίωση της ακρίβειας των μηχανών αναζήτησης στον Παγκόσμιο Ιστό:

Αναζήτηση σελίδων που αναφέρονται σε μια ακριβή έννοια κάποιας οντολογίας, αντί να συλλέγουν όλες τις σελίδες στις οποίες εμφανίζονται ορισμένες, γενικά διφορούμενες, λέξεις-κλειδιά και αξιοποίηση πληροφοριών γενίκευσης-εξειδίκευσης.

Η τεχνητή νοημοσύνη διαθέτει μεγάλη παράδοση στην ανάπτυξη και στη χρήση γλωσσών οντολογιών. Προς το παρόν οι σημαντικότερες γλώσσες οντολογιών για τον Σημασιολογικό Παγκόσμιο Ιστό είναι:

- RDF
- RDF Schema
- OWL [7]

2.9.4 Λογική

Είναι χρήσιμη σε πολλές εφαρμογές:

Έξυπνη αναζήτηση και ερωτήσεις. Έλεγχος συνέπειας (consistency checking). Κατηγοριοποίηση των στιγμιότυπων σε κλάσεις μη ρητά δηλωμένες (classification). Ανακάλυψη κρυφών συσχετίσεων. [7]

2.9.5 Πράκτορες

Είναι προγράμματα λογισμικού που λειτουργούν αυτόνομα και προνοητικά με στόχο τη συλλογή και την οργάνωση πληροφοριών και την παρουσίαση επιλογών, από τις οποίες θα διαλέγουν οι χρήστες. Θα χρησιμοποιούν τις προαναφερθείσες τεχνολογίες:

- Μεταδεδομένα: προσδιορισμός και εξαγωγή πληροφοριών από πηγές του Ιστού.
- Οντολογίες: υποστήριξη διαδικτυακών αναζητήσεων, ερμηνεία ανακτηθέντων πληροφοριών και επικοινωνία με άλλους πράκτορες.
- Λογική: επεξεργασία ανακτηθέντων πληροφοριών και εξαγωγή συμπερασμάτων.

Η ανάπτυξη του Σημασιολογικού Ιστού εξελίσσεται σε επίπεδα.

Ανάγκη προτυποποίησης. Όταν δημιουργούμε ένα επίπεδο (layer) του σημασιολογικού ιστού πάνω σε κάποιο άλλο, πρέπει να ακολουθούμε δύο αρχές:

- Συμβατότητα προς τα κάτω.
- Μερική κατανόηση προς τα πάνω. [7]

2.10 Πρότυπα

XML (eXtended Markup Language): είναι το πλέον διαδεδομένο πρότυπο δομημένης αναπαράστασης δεδομένων και η υποστήριξή του εξασφαλίζει τη δυνατότητα ανταλλαγής δεδομένων ανάμεσα σε διαφορετικές εφαρμογές και συστήματα.

RDF (Resource Description Framework): είναι ένα μοντέλο δεδομένων για τα αντικείμενα και τις μεταξύ τους σχέσεις, με αναπαράσταση και σε μορφή XML. Το RDF Schema με τη σειρά του είναι ένα λεξιλόγιο για την περιγραφή των ιδιοτήτων και των κλάσεων RDF αντικειμένων, με μια σημασιολογία για τις γενικευμένες ιεραρχίες τέτοιων ιδιοτήτων και κλάσεων.

OWL (Web Ontology Language): προσθέτει επιπλέον δυνατότητες για την περιγραφή των ιδιοτήτων και των κλάσεων.

RuleML (Rule Markup Language): είναι μια γλώσσα για τη σήμανση κανόνων. [7]

2.11 Βιβλιογραφία- αναφορά πηγής

[3]: πηγή από σύγγραμμα

- Αξιολόγηση σημασιολογικού ιστού και κοινωνικές τεχνολογίες για ψηφιακές βιβλιοθήκες της Έμελυ Χαντζηχρήστου Ιόνιο Πανεπιστήμιο Τμήμα Αρχαιονομίας – Βιβλιοθηκονομίας
Μεταπτυχιακό Πρόγραμμα Σπουδών: Επιστήμη της πληροφορίας – υπηρεσίες πληροφόρησης σε ψηφιακό περιβάλλον

[4]: πηγή από διαδίκτυο

https://el.wikipedia.org/wiki/%CE%A0%CE%B1%CE%B3%CE%BA%CF%8C%CF%83%CE%BC%CE%B9%CE%BF%CF%82_%CE%99%CF%83%CF%84%CF%8C%CF%82

[5]: πηγή από διαδίκτυο

http://ebooks.edu.gr/modules/ebook/show.php/DSB100/534/3531,14509/index4_2.html

[6]: πηγή από διαδίκτυο

http://hermes.di.uoa.gr/exe_activities/diadiktio/21_____ .html

[7]: πηγή από διαλέξεις

- Το υλικό των διαφανειών προήλθε από τη σημαντική συνεισφορά της Β. Γκαντούνα, MSc.

Κεφάλαιο 3

3.1. Εισαγωγή στην HTML και XML

Δήλωση XML, στοιχεία, χαρακτηριστικά, σχόλια, οδηγίες επεξεργασίας.
Δόμηση XML εγγράφων.

SGML (Standard Generalized Markup Language): Διεθνές πρότυπο για τον ορισμό μεθόδων αναπαράστασης πληροφοριών, οι οποίες είναι ανεξάρτητες συσκευής και συστήματος και είναι αναγνώσιμες από άνθρωπο και υπολογιστή.

Εφαρμογές SGML: Οι γλώσσες που είναι σύμφωνες με αυτό το πρότυπο .Δύο τέτοιες είναι: Η HTML και η XML: αναπτύχθηκε για να καλύψει σημαντικές ελλείψεις της HTML. [8]

3.2 HTML

Είναι μια γλώσσα σήμανσης (markup). Ένα HTML αρχείο είναι ένα αρχείο κειμένου που περιέχει ετικέτες σήμανσης (Markup tags). Οι ετικέτες περιγράφουν στο πρόγραμμα πλοήγησης πώς θα εμφανίζεται τη σελίδα. Τα HTML αρχεία έχουν κατάληξη htm ή html. Η **HTML** (ακρωνύμιο του αγγλικού **HyperText Markup Language**, ελληνικής Γλώσσα Σήμανσης Υπερκειμένου) είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες, και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων. [8]

Η HTML γράφεται υπό μορφή στοιχείων HTML τα οποία αποτελούνται από *ετικέτες* (tags), οι οποίες περικλείονται μέσα σε σύμβολα «μεγαλύτερο από» και «μικρότερο από» (για παράδειγμα <html>), μέσα στο περιεχόμενο της ιστοσελίδας. Οι ετικέτες HTML συνήθως λειτουργούν ανά ζεύγη (για παράδειγμα <h1> και </h1>), με την πρώτη να ονομάζεται *ετικέτα έναρξης* και τη δεύτερη *ετικέτα λήξης* (ή σε άλλες περιπτώσεις *ετικέτα ανοίγματος* και *ετικέτα κλεισίματος* αντίστοιχα). Ανάμεσα στις ετικέτες, οι σχεδιαστές ιστοσελίδων μπορούν να τοποθετήσουν κείμενο, πίνακες, εικόνες κλπ.

Ο σκοπός ενός web browser είναι να διαβάσει τα έγγραφα HTML και τα συνθέτει σε σελίδες που μπορεί κανείς να διαβάσει ή να ακούσει. Ο browser δεν εμφανίζει τις ετικέτες HTML, αλλά τις χρησιμοποιεί για να ερμηνεύσει το περιεχόμενο της σελίδας.

Τα στοιχεία της HTML χρησιμοποιούνται για να κτίσουν όλους του ιστότοπους. Η HTML επιτρέπει την ενσωμάτωση εικόνων και άλλων αντικειμένων μέσα στη σελίδα, και μπορεί να χρησιμοποιηθεί για να εμφανίσει διαδραστικές φόρμες. Παρέχει τις μεθόδους δημιουργίας δομημένων εγγράφων (δηλαδή εγγράφων που αποτελούνται

από το περιεχόμενο που μεταφέρουν και από τον κώδικα μορφοποίησης του περιεχομένου) καθορίζοντας δομικά σημαντικά στοιχεία για το κείμενο, όπως κεφαλίδες, παραγράφους, λίστες, συνδέσμους, παραθέσεις και άλλα. Μπορούν επίσης να ενσωματώνονται σενάρια εντολών σε γλώσσες όπως η JavaScript, τα οποία επηρεάζουν τη συμπεριφορά των ιστοσελίδων HTML.

Οι Web browsers μπορούν επίσης να αναφέρονται σε στυλ μορφοποίησης CSS για να ορίζουν την εμφάνιση και τη διάταξη του κειμένου και του υπόλοιπου υλικού. Ο οργανισμός W3C, ο οποίος δημιουργεί και συντηρεί τα πρότυπα για την HTML και τα CSS, ενθαρρύνει τη χρήση των CSS αντί διαφόρων στοιχείων της HTML για σκοπούς παρουσίασης του περιεχομένου.

Η Σημασιολογική HTML είναι ένας τρόπος γραφής της HTML με έμφαση στο νόημα της σημασμένης πληροφορίας παρά στην εμφάνισή της. Η HTML συμπεριλαμβάνει σημασιολογικά στοιχεία από την σύλληψή της, συμπεριλαμβάνει όμως και στοιχεία αποκλειστικά εμφανισιακά, όπως τις ετικέτες , <i> και <center>. Υπάρχουν επίσης και τα σημασιολογικά ουδέτερα span και div. Από τα τέλη της δεκαετίας του 1990, όταν τα CSS άρχισαν να δουλεύουν στους περισσότερους browser, οι σχεδιαστές ιστοσελίδων ενθαρρύνονταν να αποφεύγουν τη χρήση των εμφανισιακών ετικετών της HTML και να προτιμούν να διαχωρίζουν το περιεχόμενο από την παρουσίαση.

Σε μια συζήτηση για τον Σημασιολογικό ιστό, ο Τιμ Μπέρνερς Λι και άλλοι έδωσαν παραδείγματα τρόπων με τους οποίους έξυπνα λογισμικά πελάτη ιστού θα μπορούν μια μέρα να περιηγούνται στον Ιστό και να βρίσκουν, να φιλτράρουν και να συσχετίζουν τα προηγούμενα άσχετα και αδημοσίευτα δεδομένα, προς όφελος των ανθρώπων χρηστών. Δέκα χρόνια μετά, δεν είναι ακόμα κοινή η χρήση τέτοιων προγραμμάτων, αλλά μερικές από τις ιδέες του Web 2.0, των mashups και των ιστοσελίδων σύγκρισης τιμών προϊόντων μπορεί να πλησιάζουν στο σκεπτικό εκείνο. Η βασική διαφορά μεταξύ αυτών των εφαρμογών και των έξυπνων λογισμικών του Μπέρνερς Λι, είναι ότι η τρέχουσα μέθοδο συγκέντρωσης πληροφοριών συνήθως σχεδιάζεται από προγραμματιστές ιστού, οι οποίοι ήδη ξέρουν τις τοποθεσίες ιστού και τα API των δεδομένων που θέλουν να συνενώσουν, να συγκρίνουν και να συνδυάσουν.

Ένας σημαντικός τύπος λογισμικού περιηγείται αυτόματα το διαδίκτυο και διαβάζει ιστοσελίδες, χωρίς προηγούμενη γνώση του περιεχομένου τους, είναι και τα προγράμματα των μηχανών αναζήτησης. Αυτά τα λογισμικά εξαρτούνται από την σημασιολογική σαφήνεια των ιστοσελίδων που βρίσκουν, καθώς χρησιμοποιούν διάφορες τεχνικές και αλγόριθμους για να διαβάσουν και να ταξινομήσουν εκατομμύρια σελίδων καθημερινά, και να παρέχουν στους επισκέπτες τους τη δυνατότητα αναζήτησης, χωρίς την οποία ο Παγκόσμιος Ιστός θα είχε μόνο ένα μικρό κλάσμα της χρησιμότητάς του.

Οι σημασιολογικές δομές που ήδη υπάρχουν στην HTML, είναι σημαντικό να εφαρμόζονται καθολικά, για να βοηθούν το περιεχόμενο του δημοσιευμένου κειμένου να γίνεται καλύτερα αντιληπτό. Με αυτό τον τρόπο, τα λογισμικά των μηχανών αναζήτησης, αλλά και άλλα λογισμικά που φτιάχνουν mashup ή άλλα υβρίδια από το περιεχόμενο των ιστότοπων θα μπορούν καλύτερα να εκτιμούν τη σημασία του κειμένου που βρίσκουν στα έγγραφα HTML. Προς το σκοπό αυτό, οι ετικέτες της HTML που εξυπηρετούσαν μόνο εμφανισιακό σκοπό θεωρούνται ξεπερασμένες στην XHTML, και απαγορεύονται στην HTML5.

Τέλος, η σημασιολογικά γραμμένη HTML βελτιώνει την προσιτότητα των εγγράφων στον ιστό. Για παράδειγμα, στους browser για χρήστες με προβλήματα όρασης ή ακοής, είναι πιο εύκολη η σωστή απόδοση της δομής του εγγράφου με τρόπο άλλο από τον οπτικό, όταν αυτό είναι σωστά σημασιολογικά σημασμένο.

Ο Παγκόσμιος Ιστός αποτελείται κυρίως από αρχεία HTML τα οποία μεταφέρονται από εξυπηρετητές προς browsers χρησιμοποιώντας το πρωτόκολλο HTTP. Ωστόσο, μια που το ίδιο πρωτόκολλο μπορεί επιπλέον να μεταφέρει εικόνες, ήχο και άλλο περιεχόμενο, χρειάζεται ένας τρόπος αναγνώρισης του τύπου του περιεχομένου που μεταφέρεται. Έτσι, μαζί με το ίδιο το αρχείο, μεταφέρονται και μερικές επιπλέον πληροφορίες, ή μεταδεδομένα, μεταξύ αυτών και ο τύπος MIME (για παράδειγμα text/html ή application/xhtml+xml) καθώς και η κωδικοποίηση χαρακτήρων που χρησιμοποιείται.

Στους σύγχρονους browser, ο τύπος MIME που συνοδεύει το αρχείο HTML μπορεί να επηρεάζει τον τρόπο που αυτό εμφανίζεται. Για παράδειγμα, ένα αρχείο το οποίο συνοδεύεται από έναν τύπο XHTML MIME, αναμένεται να είναι γραμμένο σε γλώσσα σωστά διατυπωμένη, κατά τα πρότυπα της XML. Αν δεν είναι, τότε τα σφάλματα στον κώδικα μπορεί να αποτρέψουν την ορθή απεικόνιση του αρχείου. Επειδή η XHTML 1.0 και η XHTML 1.1 είναι πάντα συμβατές με την XML, το W3C υποστηρίζει ότι δεν παίζει ρόλο ποιος τύπος MIME θα χρησιμοποιηθεί. [9]

3.2.1 HTML tags

Χρησιμοποιούνται για να δηλώσουν τα στοιχεία (elements) της HTML. Βρίσκονται μεταξύ των <και > π.χ. <tag_name>,
. Συνήθως βρίσκονται σε ζεύγη ,π.χ. Semantic Web , όπου η πρώτη ονομάζεται ετικέτα αρχής και η δεύτερη ετικέτα τέλους. Το κείμενο που βρίσκεται ανάμεσα στις ετικέτες αποτελεί το περιεχόμενο του στοιχείου <tag_name> Content </tag_name>. Δεν παίζουν ρόλο τα κεφαλαία στα ονόματα των ετικετών (δεν είναι case sensitive). Μπορεί να περιέχουν χαρακτηριστικά (attributes) που προσδιορίζουν καλύτερα τα στοιχεία, π.χ. <body bgcolor="red">. [8]

3.3 XML

Η XML (από το EXtensible Markup Language) είναι μία γλώσσα σήμανσης, που περιέχει ένα σύνολο κανόνων για την ηλεκτρονική κωδικοποίηση κειμένων. Ορίζεται, κυρίως, στην προδιαγραφή XML 1.0 (XML 1.0 Specification), που δημιούργησε ο διεθνής οργανισμός προτύπων W3C (World Wide Web Consortium), αλλά και σε διάφορες άλλες σχετικές προδιαγραφές ανοιχτών προτύπων. Η XML σχεδιάστηκε δίνοντας έμφαση στην απλότητα, τη γενικότητα και τη χρησιμότητα στο Διαδίκτυο. Είναι μία μορφοποίηση δεδομένων κειμένου, με ισχυρή υποστήριξη Unicode για όλες τις γλώσσες του κόσμου. Αν και η σχεδίαση της XML εστιάζει στα κείμενα, χρησιμοποιείται ευρέως για την αναπαράσταση αυθαίρετων δομών δεδομένων, που προκύπτουν για παράδειγμα στις υπηρεσίες ιστού.

Υπάρχει μία ποικιλία διεπαφών προγραμματισμού εφαρμογών, που μπορούν να χρησιμοποιούν οι προγραμματιστές, για να προσπελούν δεδομένα XML, αλλά και

διάφορα συστήματα σχημάτων XML, τα οποία είναι σχεδιασμένα για να βοηθούν στον ορισμό γλωσσών, που προκύπτουν από την XML.

Έως το 2009, έχουν αναπτυχθεί εκατοντάδες γλώσσες που βασίζονται στην XML, συμπεριλαμβανομένων του RSS, του SOAP και της XHTML. Προεπιλεγμένες κωδικοποιήσεις βασισμένες στην XML, υπάρχουν για τις περισσότερες σουίτες εφαρμογών γραφείου, συμπεριλαμβανομένων του Microsoft Office (Office Open XML), του OpenOffice.org (OpenDocument) και του iWork της εταιρίας Apple. Η XML είναι μια γλώσσα σήμανσης όπως η HTML. [10]

Επεκτάσιμη: δεν έχει προκαθορισμένα tags, αλλά αφήνει το χρήστη να ορίσει τα δικά του tags. Είναι αυτό-περιγραφική γλώσσα. Σχεδιάστηκε για να περιγράφει τα δεδομένα, όχι την παρουσίαση τους. Συμπληρώνει την HTML και δεν την αντικαθιστά. [8]

3.4 Διαφορές HTML- XML

Η HTML καθορίζει τον τρόπο παρουσίασης του περιεχομένου, ενώ η XML καθορίζει τη δομή του περιεχομένου (DTD, XML Schema). Η HTML διαθέτει προκαθορισμένο σύνολο ετικετών, ενώ η XML δεν έχει προκαθορισμένες ετικέτες, αλλά αφήνει τον χρήστη να ορίσει τις δικές του. Στην HTML οι ετικέτες συνήθως είναι σε ζεύγη αρχής-τέλους, αλλά μπορεί και όχι
, ενώ στην XML όλες οι ετικέτες πρέπει να κλείνουν, δηλαδή να είναι πάντα σε ζεύγη. Στην HTML δεν περιέχονται δομικές πληροφορίες => δύσκολα κατανοητή από υπολογιστές, ενώ στην XML είναι εύκολα κατανοητή από υπολογιστές. Η HTML καθορίζει τη μορφοποίηση του εγγράφου, ενώ η XML ξεχωρίζει το περιεχόμενο από την μορφοποίηση. Στην HTML οι ετικέτες δεν είναι case-sensitive, ενώ στην XML οι ετικέτες είναι case-sensitive. Η XML επιτρέπει τον ορισμό περιορισμών σε τιμές οι οποίες πρέπει να είναι τετραψήφιος αριθμός. Η XML χρησιμοποιείται και ως μια ενιαία μορφή ανταλλαγής δεδομένων μεταξύ εφαρμογών. [8]

3.5 Ομοιότητες HTML- XML

Είναι γλώσσες σήμανσης (markup): δίνουν τη δυνατότητα στο χρήστη να δημιουργήσει περιεχόμενο και να παρέχει πληροφορίες σχετικά με το ρόλο του συγκεκριμένου περιεχομένου. Χρησιμοποιούν ετικέτες(tags). Εύκολα κατανοητές από τον άνθρωπο. [8]

3.6 Εισαγωγή στα DTD

Ένα DTD παρέχει στις εφαρμογές τη χρήσιμη πληροφορία για το ποια ονόματα και δομές μπορούν να χρησιμοποιηθούν σε ένα συγκεκριμένο τύπο εγγράφου. Η χρήση ενός DTD κατά τη σύνταξη εγγράφων που είναι γνωστό ότι μπορεί να βεβαιωθεί ότι όλα τα έγγραφα που ανήκουν σε αυτόν τον συγκεκριμένο τύπο θα δομηθούν και θα ονομαστούν σύμφωνα με ένα συνεπή και ομοιόμορφο τρόπο. Τα DTDs είναι λιγότερο σημαντικά για την επεξεργασία των εγγράφων που είναι γνωστό ότι είναι ήδη καλά ορισμένα, αλλά απαιτούνται ακόμα για να εκμεταλλευτούν τους ειδικούς τύπους των ιδιοτήτων που χρησιμοποιούνται στην XML, όπως είναι οι ενσωματωμένοι μηχανισμοί

για την αναφορά παραπομπής ID/IDREF ή τη χρήση των προεπιλεγμένων ιδιοτήτων. Η XML παρέχει έναν τρόπο διαμοιρασμού των δεδομένων ανεξάρτητο από κάθε εφαρμογή. Με ένα DTD ανεξάρτητες ομάδες ανθρώπων μπορούν να συμφωνήσουν να χρησιμοποιούν ένα κοινό DTD για την ανταλλαγή δεδομένων. Τα DTDs είναι απλά και περιορισμένα. Ο διάδοχος των DTDs, το XML Schema, θα επιτρέψει στους σχεδιαστές να προσδιορίσουν τους δικούς τους τύπους δεδομένων (data types) για τα στοιχεία και τις ιδιότητες.

Δεδομένου ότι το XML Schema είναι ακόμα μια προδιαγραφή (specification) δεν έχει καθοριστεί ακόμα καθορίζεται σε τελική μορφή, και οι κατασκευαστές λογισμικού είναι προτιμότερο να χρησιμοποιούν τα DTDs, τα οποία αποτελούν πρότυπο (standard) και υποστηρίζονται πρακτικά από όλα τα εργαλεία της XML.

Κάθε DTD αποτελείται από ένα σύνολο από κανόνες ή δηλώσεις. Κάθε δήλωση προσθέτει ένα στοιχείο, σύνολο από ιδιότητες, οντότητες ή σημειώσεις που πρόκειται να χρησιμοποιηθούν σε ένα έγγραφο XML. Τα DTD μπορούν να συνδυαστούν με τη χρήση παραμετρικών οντοτήτων, μια τεχνική που ονομάζεται παραμετροποίηση. Μπορούν επίσης να προστεθούν δηλώσεις μέσα στο εσωτερικό υποσύνολο του κειμένου. [12]

Η σειρά των δηλώσεων είναι σημαντική σε δύο περιπτώσεις. Στην πρώτη περίπτωση, εάν υπάρχουν πλεονάζουσες δηλώσεις (όταν υπάρχει, δηλαδή, η δήλωση του ίδιου ονόματος ενός στοιχείου πάνω από μια φορά), επικρατεί η πρώτη δήλωση που εμφανίζεται και οι υπόλοιπες αγνοούνται. Στη δεύτερη περίπτωση, εάν χρησιμοποιούνται παραμετρικές οντότητες, αυτές πρέπει να δηλώνονται πριν τη χρήση της αναφοράς τους. Η σύνταξη των δηλώσεων είναι ευανάγνωστη όταν χρησιμοποιούνται κενά διαστήματα. Η πρόσθεση επιπλέον διαστημάτων μπορεί να γίνει οπουδήποτε εκτός από την αλφαριθμητική σειρά των χαρακτήρων στην αρχή της δήλωσης, όπου προσδιορίζεται ο τύπος της δήλωσης. Όλες οι παρακάτω δηλώσεις είναι σωστές:

```
<! ELEMENT CUSTOMER ALL>
<! ELEMENT
CUSTOMER
ALL>
<! ELEMENT ΧΡΩΜΑ (ΠΡΑΣΙΝΟ |
ΚΟΚΚΙΝΟ |
ΜΠΛΕ ) * >
```

3.6.1 XML DTD (document type definition)

Με ένα DTD, ανεξάρτητες ομάδες ανθρώπων μπορούν να συμφωνήσουν να χρησιμοποιούν ένα τυποποιημένο DTD για την ανταλλαγή δεδομένων μέσω.

Η εφαρμογή σας μπορεί να χρησιμοποιήσει ένα τυποποιημένο DTD για να βεβαιωθείτε ότι τα δεδομένα που λαμβάνετε από τον έξω κόσμο είναι έγκυρη.

Μπορείτε επίσης να χρησιμοποιήσετε ένα DTD για να επαληθεύσει τα δικά σας δεδομένα.

3.6.2 XML document with an internal DTD

```
<? xml version="1.0"?>
<! DOCTYPE note [
<! ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend</body>
</note>
```

To DTD ερμηνεύεται ως εξής:

- **!DOCTYPE note** defines that the root element of this document is note
- **!ELEMENT note** defines that the note element must contain four elements: "to,from,heading,body"
- **!ELEMENT to** defines the to element to be of type "#PCDATA"
- **!ELEMENT from** defines the from element to be of type "#PCDATA"
- **!ELEMENT heading** defines the heading element to be of type "#PCDATA"
- **!ELEMENT body** defines the body element to be of type "#PCDATA"

3.6.3 External DTD declaration

If the DTD is declared in an external file, the <!DOCTYPE> definition must contain a reference to the DTD file.

3.6.4 XML document with a reference to an external DTD

```
<?xml version="1.0"?>
<!DOCTYPE note SYSTEM "note.dtd">
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

And here is the file "note.dtd", which contains the DTD:

```
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
```

```
<!ELEMENT from (#PCDATA)>  
<!ELEMENT heading (#PCDATA)>  
<!ELEMENT body (#PCDATA)>
```

3.7 DTD Elements

Βασικά δομικά στοιχεία ενός εγγράφου XML.

```
<!ELEMENT element-name (element-content)>  
<!ELEMENT example (#CDATA)>
```

Μπορούμε να ορίσουμε την ιεραρχία των στοιχείων ενός XML εγγράφου.

Παράδειγμα:

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

Τα στοιχεία lecturer, name και phone μπορούν να χρησιμοποιηθούν μέσα στο έγγραφο.

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

Ένα στοιχείο τύπου lecturer περιέχει ένα στοιχείο name και ένα στοιχείο phone, με αυτή τη σειρά.

```
<!ELEMENT name (#CDATA)>, <!ELEMENT phone (#CDATA)>
```

Τα στοιχεία τύπου name και phone μπορούν να έχουν οποιοδήποτε περιεχόμενο.

3.7.1 Ορισμός ενός στοιχείου

- Ορίζοντας μία μόνο εμφάνιση ενός element
- Ορίζοντας τουλάχιστον μία εμφάνιση ενός element Ορίζοντας μηδέν ή περισσότερες εμφανίσεις ενός element
- Ορίζοντας μία ή καμία εμφάνιση ενός element
- Ορίζοντας συνδυασμούς των ανωτέρω: <!ELEMENT note(to+, from, subject, message*)>

3.8 DTD Attributes (χαρακτηριστικά)

Σε ένα DTD, τα χαρακτηριστικά δηλώνονται με μια δήλωση ATTLIST. Μια δήλωση ενός attribute (χαρακτηριστικό) έχει την ακόλουθη σύνταξη:

```
<!ATTLIST element-name attribute-name attribute-type attribute-value>
```

DTD example:

```
<!ATTLIST payment type CDATA "check">
```

XML example:

```
<payment type="check" />
```

The attribute-type can be one of the following:

Type	Description
CDATA	The value is character data
(<i>en1 en2 ..</i>)	The value must be one from an enumerated list
ID	The value is a unique id
IDREF	The value is the id of another element
IDREFS	The value is a list of other ids
NMTOKEN	The value is a valid XML name
NMTOKENS	The value is a list of valid XML names
ENTITY	The value is an entity
ENTITIES	The value is a list of entities
NOTATION	The value is a name of a notation
xml:	The value is a predefined xml value

The **attribute-value** can be one of the following:

Value	Explanation
Value	The default value of the attribute
#REQUIRED	The attribute is required
#IMPLIED	The attribute is optional
#FIXED <i>value</i>	The attribute value is fixed

A Default Attribute Value

DTD:

```
<!ELEMENT square EMPTY>
```

```
<!ATTLIST square width CDATA "0">
```

Valid XML:

```
<square width="100" />
```

Στο παραπάνω παράδειγμα, το «τετράγωνο» στοιχείο ορίζεται να είναι ένα κενό στοιχείο με ένα "πλάτος" χαρακτηριστικό του τύπου CDATA. Εάν δεν έχει καθοριστεί το πλάτος, έχει μια προεπιλεγμένη τιμή 0.

REQUIRED

Syntax

<!ATTLIST element-name attribute-name attribute-type #REQUIRED>

Παράδειγμα:

DTD:

<!ATTLIST person number CDATA #REQUIRED>

Valid XML:

<person number="5677" />

Invalid XML:

<person />

Χρησιμοποίησε τη λέξη-κλειδί #REQUIRED αν δεν έχετε μια επιλογή για μια προκαθορισμένη τιμή, αλλά εξακολουθούν να θέλουν να αναγκάσουν την ιδιότητα να είναι παρόντες.

#IMPLIED

Syntax

<!ATTLIST element-name attribute-name attribute-type #IMPLIED>

Παράδειγμα:

DTD:

<!ATTLIST contact fax CDATA #IMPLIED>

Valid XML:

<contact fax="555-667788" />

Valid XML:

<contact />

Χρησιμοποίησε τη λέξη-κλειδί #IMPLIED, αν δεν θέλετε να αναγκάσει το συγγραφέα να συμπεριλάβει ένα χαρακτηριστικό, και δεν έχετε μια επιλογή για μια προκαθορισμένη τιμή.

#FIXED

Syntax

<!ATTLIST element-name attribute-name attribute-type #FIXED "value">

Παράδειγμα:

DTD:

`<!ATTLIST sender company CDATA #FIXED "Microsoft">`

Valid XML:

`<sender company="Microsoft" />`

Invalid XML:

`<sender company="W3Schools" />`

Χρησιμοποίησε τη λέξη-κλειδί #FIXED όταν θέλετε ένα χαρακτηριστικό να έχει μια σταθερή τιμή χωρίς να επιτρέπει στον συγγραφέα να το αλλάξει. Εάν ένας συγγραφέας περιλαμβάνει μια άλλη αξία, η μονάδα ανάλυσης XML θα επιστρέψει ένα σφάλμα.

Enumerated Attribute Values**Syntax**`<!ATTLIST element-name attribute-name (en1|en2|..) default-value>`**Παράδειγμα:**

DTD:

`<!ATTLIST payment type (check|cash) "cash">`

XML example:

`<payment type="check" />`

or

`<payment type="cash" />`

Χρησιμοποιήστε καταγραμμένων τιμών των παραμέτρων, όταν θέλετε η τιμή του χαρακτηριστικού να είναι ένα από ένα σταθερό σύνολο νομικών αξιών. [11]

3.9 Βιβλιογραφία- αναφορά πηγής

[8]: πηγή από διαλέξεις

- Το υλικό των διαφανειών προήλθε από τη σημαντική συνεισφορά της Β. Γκαντούνα, MSc.

[9]: πηγή από διαδίκτυο

http://www.w3schools.com/html/html_intro.asp

[10]: πηγή από διαδίκτυο

<https://en.wikipedia.org/wiki/HTML>

[11]: πηγή από διαδίκτυο

http://www.w3schools.com/xml/xml_dtd_intro.asp

[12]: πηγή από διαδίκτυο

https://en.wikipedia.org/wiki/Document_type_definition

Κεφάλαιο 4

4.1 Η γλώσσα σήμανσης XML

Ένας τύπος εγγράφου χρησιμοποιείται για όλους τους σκοπούς με αποτέλεσμα να είναι απαραίτητη η υπερβολική χρήση ετικετών αντί για τον καθορισμό νέων με συγκεκριμένο σκοπό. Επιπλέον οι περισσότερες από τις ετικέτες σκοπεύουν μόνο στη διαμόρφωση της εμφάνισης του εγγράφου. Για την επίλυση αυτών των προβλημάτων έγινε προσπάθεια υιοθέτησης της SGML για χρήση στο διαδίκτυο, αλλά κάτι τέτοιο αποδείχτηκε πολύπλοκο και πολύ δύσκολο. Επομένως ήταν αναγκαίος ο καθορισμός μιας νέας γλώσσας, η οποία να είναι υποσύνολο της SGML αλλά να διατηρεί τα βασικά χαρακτηριστικά της. Η γλώσσα αυτή είναι η XML.

Οι σχεδιαστικοί στόχοι της XML είναι οι εξής:

- να είναι εύχρηστη στο Internet
- να υποστηρίζει πλήθος εφαρμογών
- να είναι συμβατή με την SGML
- ευκολία ανάπτυξης προγραμμάτων που επεξεργάζονται XML έγγραφα
- ο αριθμός των προαιρετικών χαρακτηριστικών στην XML να είναι όσο το δυνατόν πιο μικρός, ιδανικό επίπεδο το μηδέν.
- τα XML έγγραφα θα πρέπει να είναι ευανάγνωστα
- ο σχεδιασμός XML θα πρέπει να προετοιμάζεται γρήγορα
- ο σχεδιασμός XML θα πρέπει να είναι τυπικός και περιεκτικός
- τα XML έγγραφα θα πρέπει να δημιουργούνται εύκολα
- η περιεκτικότητα στον XML συμβολισμό είναι μικρής σημασίας

Τα κύρια χαρακτηριστικά της XML είναι:

- παρέχει ακριβή καθορισμό του περιεχομένου, ώστε τα αποτελέσματα μιας αναζήτησης να είναι ικανοποιητικά.
- παρέχει δυνατότητα καθορισμού των στοιχείων του εγγράφου (π.χ. Τίτλος, Συγγραφέας, Εκδότης...)
- επιτρέπει τη μεταφορά μέσω διαδικτύου, δομημένης πληροφορίας ανεξάρτητα από πλατφόρμες και συστήματα

Η XML συμπληρώνει και δεν αντικαθιστά την HTML. Ενώ η HTML χρησιμοποιείται για διατύπωση και εμφάνιση των δεδομένων η XML αναπαριστά τη συναφή έννοια των δεδομένων. Στην HTML οι ετικέτες είναι προκαθορισμένες ενώ η XML παρέχει τη δυνατότητα να καθορίζουν οι χρήστες τις ετικέτες και τις δομημένες μεταξύ τους σχέσεις.

Επομένως, ο χρήστης με τη βοήθεια της XML μπορεί να δημιουργήσει μια νέα γλώσσα σήμανσης καθορίζοντας ένα νέο σύνολο από ετικέτες ή να χρησιμοποιήσει ένα σύνολο από ετικέτες το οποίο είναι καθορισμένο από κάποιον άλλο. Αυτό έχει ως αποτέλεσμα

να υπάρχει απεριόριστος αριθμός γλωσσών σήμανσης που προέρχονται από την XML, και για το λόγο αυτό η XML είναι μια μετα-γλώσσα.

Τα πλεονεκτήματα της γλώσσας σήμανσης XML είναι τα εξής:

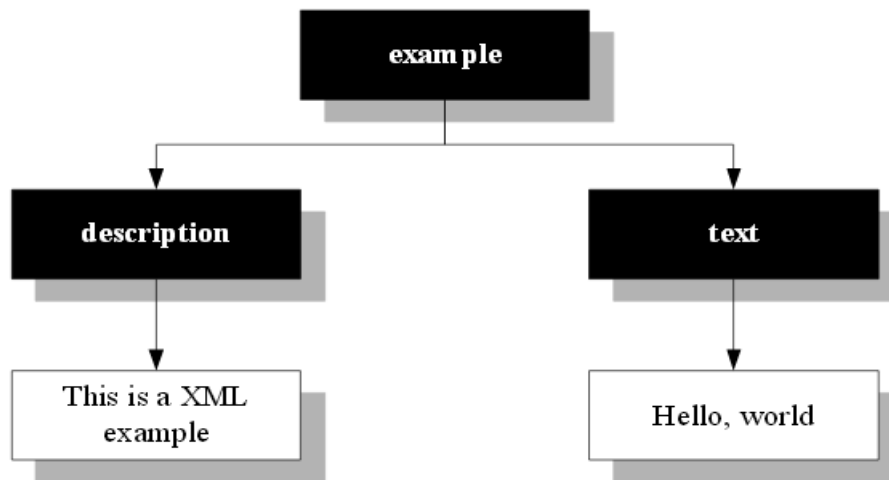
Πιο ικανοποιητικές αναζητήσεις
Ανάπτυξη ευέλικτων εφαρμογών
Ενσωμάτωση δεδομένων από ανόμοιες πηγές πληροφορίας
Δεδομένα από πολλαπλές εφαρμογές
Υπολογισμός και διαχείριση δεδομένων τοπικά
Πολλαπλοί τρόποι εμφάνισης των δεδομένων
Ενημέρωση πεδίων
Διαθέσιμα πρότυπα
Μορφοποίηση εμφάνισης για το διαδίκτυο.

Η γλώσσα προγραμματισμού XML περιγράφει μια κατηγορία πληροφοριών που καλούνται XML έγγραφα (documents) καθώς επίσης περιγράφει τμηματικά τη συμπεριφορά των προγραμμάτων που τα επεξεργάζονται. Τα XML έγγραφα είναι αυστηρά δομημένα και μπορούν να αναπαρασταθούν με δενδρική δομή. Το πρώτο στοιχείο του εγγράφου, το οποίο περικλείει και όλα τα άλλα, ονομάζεται «ρίζα» (root). Κάθε στοιχείο που περιέχει στοιχεία ή δεδομένα ονομάζεται «πατέρας» (parent), ενώ κάθε στοιχείο ή δεδομένο το οποίο περιέχεται σε κάποιο άλλο στοιχείο ονομάζεται «παιδί» (child). Κάθε στοιχείο ή δεδομένο πρέπει να ανήκει σε κάποιο άλλο ή να είναι παιδί της ρίζας.

Ένα παράδειγμα XML εγγράφο είναι το παρακάτω:

```
<?xml version="1.0"?>  
<example>  
<text> Hello, world!</text>  
<description>This is a XML example</description>  
</example>
```

Το παράδειγμα αυτό μπορεί να αναπαρασταθεί με την εξής δενδρική μορφή:



Εικόνα 1-1 Η δένδρική μορφή ενός XML εγγράφου

Για τη δημιουργία XML εγγράφων μπορεί να χρησιμοποιηθεί οποιαδήποτε εργαλείο συγγραφής ή επεξεργασίας κειμένου(π.χ. Microsoft Notepad, Unixvi, Apple Simple Text κλπ). Επιπλέον, υπάρχουν διαθέσιμα πολλά εργαλεία επεξεργασίας XML εγγράφων τα οποία παρέχουν πλεονεκτήματα όπως διαφορετικά χρώματα εμφάνισης των ετικετών, απόκρυψη των ετικετών και εμφάνιση μόνο του περιεχομένου τους κλπ. Επίσης πολλά από αυτά τα εργαλεία παρέχουν μηχανισμούς ελέγχου εγκυρότητας και ορθής μορφοποίησης των εγγράφων, δύο χαρακτηριστικά που αναφέρονται παρακάτω.

4.2 Μοντελοποίηση εγγράφων

Ένα από τα σημαντικότερα πλεονεκτήματα της XML είναι η δυνατότητα που παρέχει για καθορισμό νέων γλωσσών σήμανσης, καθορίζοντας στοιχεία και ιδιότητες τα οποία περιγράφουν με τον καλύτερο τρόπο συγκεκριμένο είδος πληροφορίας. Για τον πλήρη και τυπικό καθορισμό μιας νέας γλώσσας σήμανσης, για τον περιορισμό του συνόλου των στοιχείων και των ιδιοτήτων της και για τον έλεγχο της γραμματικής και της σύνταξης των ετικετών χρησιμοποιούνται τα μοντέλα εγγράφων (document models). Τα μοντέλα εγγράφων καθορίζουν ποια έγγραφα συμμορφώνονται με τη γλώσσα σήμανσης XML. Το μοντέλο είναι ένα έγγραφο γραμμένο με βάση ένα συντακτικό που έχει καθοριστεί για να περιγράφει XML γλώσσες, και ορίζει μονοσήμαντα τη γραμματική και το σύνολο ετικετών της γλώσσας σήμανσης. Η γλώσσα σήμανσης που καθορίζεται από το μοντέλο ονομάζεται τύπος εγγράφου ή εφαρμογή XML. Η χρήση ενός μοντέλου εγγράφου είναι προαιρετική καθώς η XML έχει σχεδιαστεί έτσι ώστε να μην είναι απαραίτητο ένα τέτοιο έγγραφο και η χρήση του εξαρτάται αποκλειστικά από την εφαρμογή και τον δημιουργό της.

Η χρήση όμως των μοντέλων εγγράφων έχει σημαντικά πλεονεκτήματα:

Οι προκαθορισμένοι κανόνες καθιστούν πιο εύκολη τη δημιουργία λογισμικού για τα έγγραφα και ελαχιστοποιούν την πιθανότητα λαθών. Όταν το έγγραφο απαιτεί συγκεκριμένα στοιχεία αυτά μπορούν να καθοριστούν με τη βοήθεια του μοντέλου. Το μοντέλο καθοδηγεί τον χρήστη σε ένα περιβάλλον συγγραφής XML εγγράφων, παρέχοντας στο χρήστη μια λίστα από τα απαραίτητα στοιχεία.

Τα μειονεκτήματα της χρήσης των μοντέλων εγγράφων είναι η δυσκολία διαχείρισης, η καθυστέρηση κατά την επεξεργασία και η παρεμπόδιση δημιουργίας νέων στοιχείων καθώς το μοντέλο καθορίζει ποια στοιχεία μπορούν ή όχι να χρησιμοποιηθούν.

Οι πιο δημοφιλείς τύποι μοντέλων εγγράφων είναι δυο:

Ορισμός τύπου εγγράφων (Document Type Definition- DTD) και XML Schema.

Ένα DTD καθορίζει ένα τύπο εγγράφου ως εξής:

Καθορίζει το σύνολο των επιτρεπτών στοιχείων. Δεν επιτρέπεται η χρήση άλλων ονομάτων για τα στοιχεία από αυτά που καθορίζονται από τα παραπάνω.

Καθορίζει το μοντέλο περιεχομένου κάθε εγγράφου. Το μοντέλο περιεχομένου είναι ένα υπόδειγμα το οποίο καθορίζει ποιο στοιχείο ή δεδομένο μπορεί να είναι μέρος ενός στοιχείου, με ποια σειρά, πόσες φορές και αν είναι υποχρεωτικό.

Καθορίζει ένα σύνολο επιτρεπτών ιδιοτήτων για κάθε στοιχείο. Κάθε ορισμός ιδιότητας καθορίζει το όνομα, τον τύπο δεδομένων, τις προκαθορισμένες τιμές και την συμπεριφορά ης ιδιότητας.

Παρέχει ένα πλήθος μηχανισμών για την εύκολη διαχείριση του μοντέλου, π.χ. χρήση οντοτήτων (entities), εισαγωγή τμημάτων από εξωτερικά αρχεία.

Εναλλακτικά μπορεί να χρησιμοποιηθεί το μοντέλο εγγράφου XML. Τα DTD παρουσιάζουν μειονεκτήματα όπως η περιορισμένη δυνατότητα καθορισμού τύπων δεδομένων και η διαφορετική σύνταξη του DTD από τα XML έγγραφα. Τα XML Schema δίνουν λύση στους παραπάνω περιορισμούς.

Τα πιο σημαντικά πλεονεκτήματα των XML Schema σε σχέση με τα DTD είναι:

Υποστηρίζουν περισσότερους από 44 τύπους δεδομένων και επιτρέπουν τη δημιουργία καινούριων. Έχουν την ίδια σύνταξη με ένα XML έγγραφο. Είναι αντικειμενοστραφείς. Επιτρέπουν την δημιουργία συνόλων. Επιτρέπουν τον καθορισμό περιεχομένου ενός στοιχείου ως μοναδικό key. Επιτρέπουν τον καθορισμό πολλαπλών στοιχείων με τι ίδιο όνομα και διαφορετικό περιεχόμενο χωρίς να δημιουργείται σύγχυση κατά την επεξεργασία.

Η επιλογή ανάμεσα στα δύο μοντέλα εγγράφων είναι θέμα χρήστη και εφαρμογής. Τα XML Schema είναι μεταγενέστερα και παρουσιάζουν σημαντικά πλεονεκτήματα έναντι των DTD αλλά δεν τα αντικαθιστούν. Τα DTD πλεονεκτούν σε μέγεθος, έχουν γνώριμη και απλή σύνταξη.

Μερικές εφαρμογές της XML, καθορισμένες με DTD ή XML Schema είναι οι εξής:

- CML (Chemical Markup Language): Γλώσσα σήμανσης για την περιγραφή πολύπλοκων μορίων
- MathML (Mathematical Markup Language): Γλώσσα σήμανσης για την περιγραφή εξισώσεων στο διαδίκτυο

- VML (Vector Markup Language): Γλώσσα σήμανσης για τον σχεδιασμό γραφικών που περιγράφονται με διάνυσμα
- RDF (Resource Description Framework): Γλώσσα σήμανσης για την αναπαράσταση μεταδεδομένων

4.3 XML έγγραφα

Υπάρχουν δύο τύποι XML εγγράφων. Τα μορφοποιημένα ή καλά ορισμένα (well-formed) και τα έγκυρα (valid). Ένα καλά ορισμένο XML έγγραφο ακολουθεί τους γενικούς κανόνες σύνταξης της XML, οι οποίοι είναι πιο αυστηροί από αυτούς της HTML και της SGML.

Μερικοί από τους συγκεκριμένους κανόνες είναι:

Οι χαρακτήρες δεδομένων της XML έχουν πάντα μία ετικέτα τέλους οποιουδήποτε είδους είτε π.χ. το ζεύγος <ετικέτα> </ετικέτα>, είτε μια ετικέτα κενού στοιχείου π.χ. <ετικέτα/>. Η σήμανση της XML ξεκινάει πάντα με το σύμβολο < ή με το &. Οι τύποι των στοιχείων και τα ονόματα των εισαγωγικών αναγνωρίζουν πεζά και κεφαλαία. Οι τιμές των ιδιοτήτων απαιτούν εισαγωγικά.

Τα έγκυρα XML έγγραφα ακολουθούν ένα μοντέλο εγγράφου, δηλαδή ένα σετ κανόνων που ένα έγγραφο ακολουθεί και τους οποίους το λογισμικό έχει την ικανότητα να διαβάσει πριν την επεξεργασία και την εμφάνιση του εγγράφου. Γενικά αυτοί οι κανόνες καθορίζουν το όνομα και το περιεχόμενο κάθε στοιχείου (element). Τα έγκυρα XML έγγραφα προσφέρουν πολύ περισσότερα στην επεξεργασία του εγγράφου από τα αντίστοιχα ορθά μορφοποιημένα. Η συγγραφή, επεξεργασία, αποθήκευση και εμφάνιση του εγγράφου έγιναν πιο εύκολα διότι το έγγραφο είναι σε ένα δομικό περιβάλλον. Η βασική διαφορά μεταξύ των ορθά μορφοποιημένων και των έγκυρων XML εγγράφων είναι η σχέση τους με το μοντέλο εγγράφου. Οι δημιουργοί των XML εγγράφων συντάσσουν έγκυρα έγγραφα, ενώ ένα ικανό λογισμικό πλοήγησης XML (XML browsers) χρειάζεται μόνο να ελέγχει το έγγραφο για ορθή μορφοποίηση πριν το εμφανίσει.

4.4 Επεξεργασία XML Εγγράφων

Ένα λογισμικό μοντέλο που ονομάζεται επεξεργαστής XML (XML processor) χρησιμοποιείται για να διαβάσει XML έγγραφα και παρέχει πρόσβαση στο περιεχόμενο και τη δομή τους. Ο επεξεργαστής XML λειτουργεί εκ μέρους ενός άλλου μοντέλου που ονομάζεται εφαρμογή (application). Αυτή η προδιαγραφή περιγράφει την απαιτούμενη συμπεριφορά του επεξεργαστή και συγκεκριμένα πως θα πρέπει να διαβάσει τα XML δεδομένα και ποιες πληροφορίες πρέπει να παρέχει στην εφαρμογή.

4.5 Η σύνταξη της XML

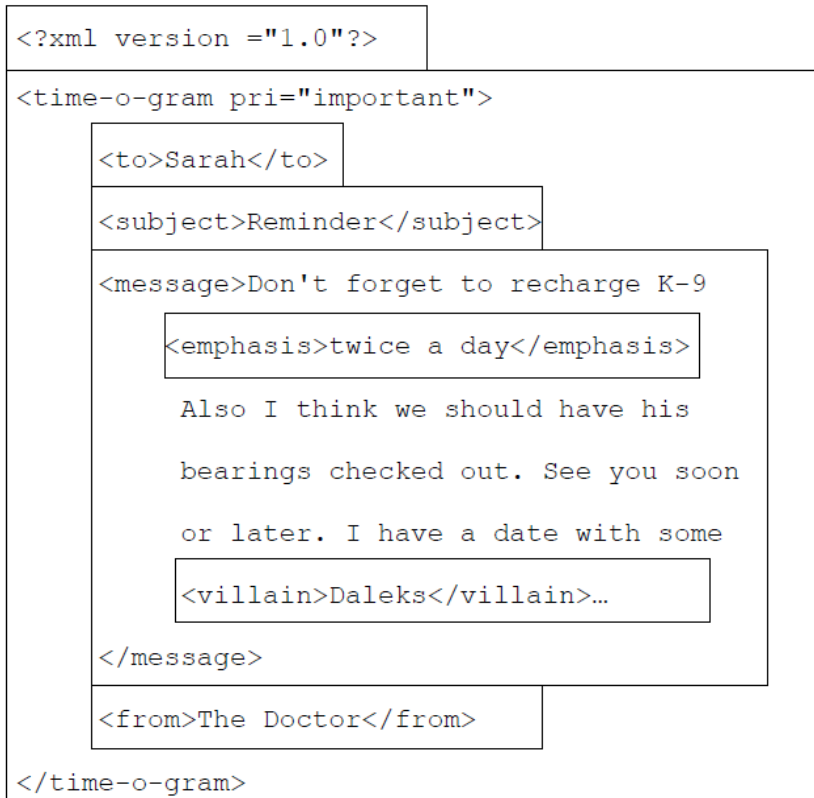
4.5.1 Η ανατομία XML εγγράφων

Η XML παρέχει ένα σύνολο κανόνων για τη δημιουργία σημασιολογικών ετικετών (tags) που χρησιμοποιούνται για να περιγράφουν τα δεδομένα. Ένα στοιχείο (element) της XML αποτελείται από μια ετικέτα έναρξης και μια ετικέτα τέλους με δεδομένα ανάμεσα. Οι ετικέτες περιγράφουν τα δεδομένα τα οποία αποτελούν την τιμή ενός στοιχείου. Το παρακάτω παράδειγμα απεικονίζει ένα απλό XML έγγραφο το οποίο αναπαριστά ένα υπόμνημα.

```
<?xml version ="1.0">
<time-o-gram pri="important" >
  <to>Sarah</to>
  <subject>Reminder</subject>
  <message>Don't forget to recharge K-9
    <emphasis>twice a day</emphasis>
    Also I think we should have his
    bearings checked out. See you soon
    or later. I have a date with some
    <villain>Daleks</villain>...
  </message>
  <from>The Doctor</from>
</time-o-gram>
```

Το παραπάνω παράδειγμα απεικονίζει ένα απλό XML έγγραφο που ακολουθεί τους συντακτικούς κανόνες της XML. Η XML επιτρέπει τον καθορισμό και τη χρήση οποιαδήποτε ετικέτας, σε αντίθεση με την HTML, η οποία θέτει περιορισμούς λόγω της χρήσης προκαθορισμένων ετικετών. Ωστόσο, ο τρόπος με τον οποίο οργανώνεται η πληροφορία αποτελεί ένα σημαντικό κομμάτι της XML. Η επιλογή των ονομάτων των στοιχείων θα πρέπει να γίνεται με βάση την ανάγνωση ενός XML εγγράφου ώστε να είναι κατανοητή είτε από ένα χρήστη είτε από ένα υπολογιστή.

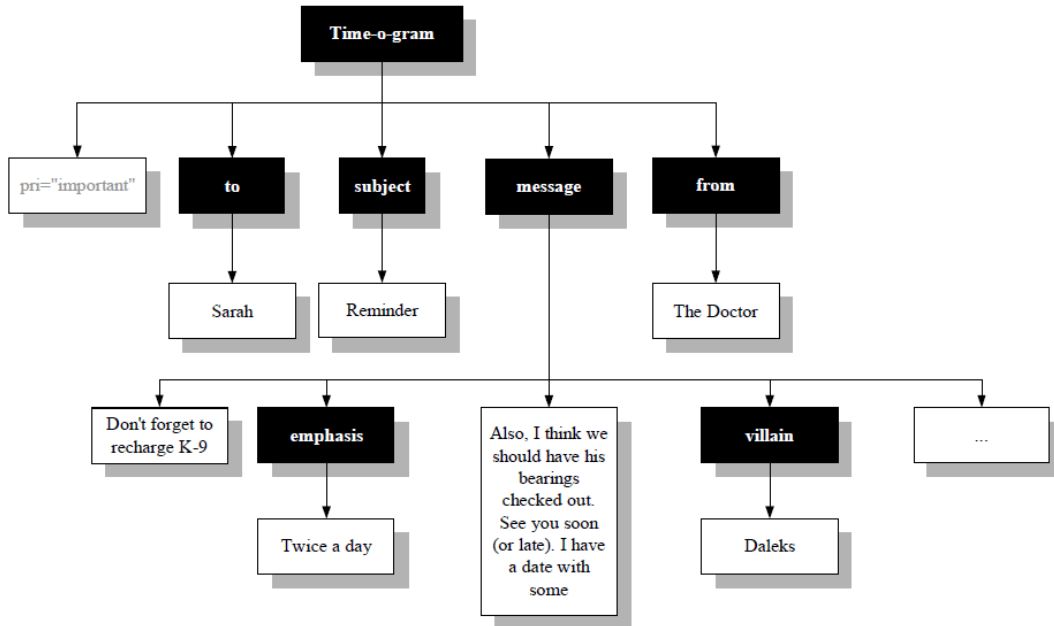
Στο σχήμα που ακολουθεί απεικονίζεται το παραπάνω XML έγγραφο το οποίο έχει χωριστεί σε περιοχές, κάθε μια από τις οποίες αναπαριστάται από ένα πλαίσιο και όπου κάθε ένα μπορεί να περιέχει άλλα πλαίσια. Το πρώτο πλαίσιο περιέχει τη δήλωση που δίνει πληροφορίες διαχείρισης για το έγγραφο. Τα υπόλοιπα πλαίσια ονομάζονται στοιχεία. Στο εσωτερικό του υπάρχουν τοποθετημένα τα στοιχεία που αντιπροσωπεύουν τα διακριτά μέρη του εγγράφου. Από το διάγραμμα φαίνεται ότι τα κυριότερα μέρη του στοιχείου <time-o-gram> είναι τα στοιχεία: <to> που περιγράφουν τον προορισμό, ο αποστολέας <from>, το αντικείμενο <subject> και το τμήμα που περιέχει το μήνυμα <message>. Το τελευταίο στοιχείο είναι το πιο περίπλοκο λόγω του ότι συνδυάζει στοιχεία και κείμενο μαζί. Το μεγαλύτερο στοιχείο με όνομα <time-o-gram> περικλείει τα άλλα στοιχεία.



Εικόνα 2-1 Τα στοιχεία ενός υπομνήματος

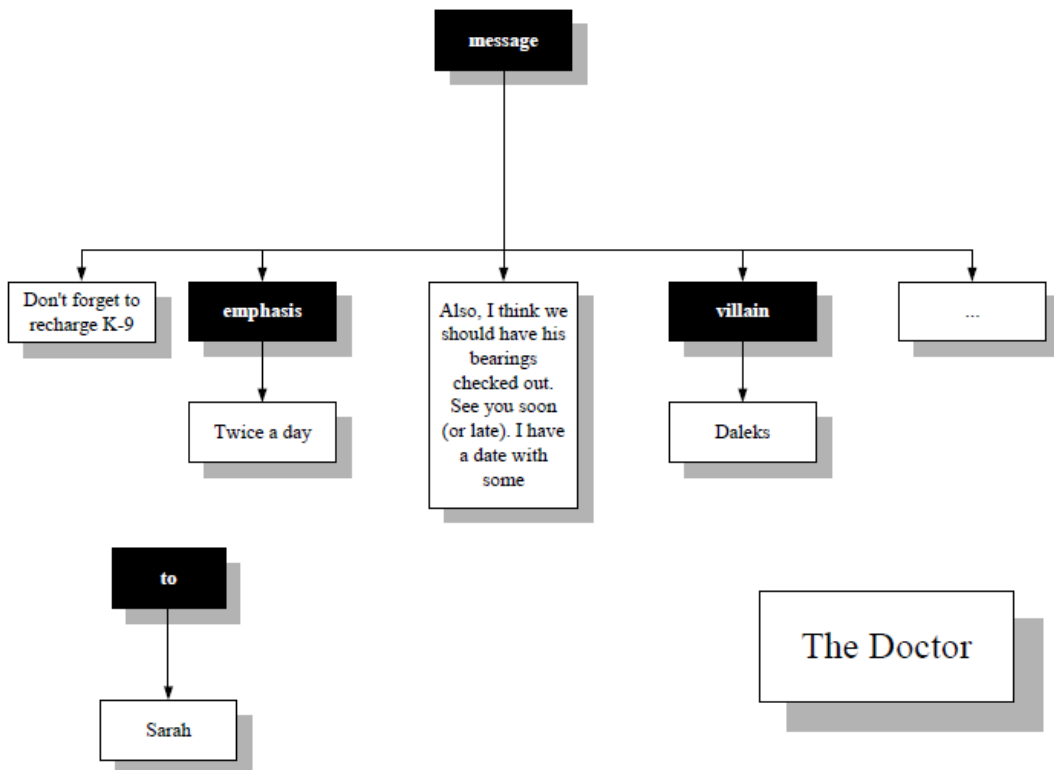
4.6 Δενδρική αναπαράσταση εγγράφου

Τα στοιχεία χωρίζουν το έγγραφο στα μέρη που το αποτελούν. Μπορούν να περιέχουν κείμενο, άλλα στοιχεία ή και τα δυο. Η εικόνα που ακολουθεί παρουσιάζει μια ιεραρχία στοιχείων που θυμίζει δέντρο. Είναι μια πολύ χρήσιμη παρουσίαση που δείχνει την σχέση μεταξύ τους. Το στοιχείο που βρίσκεται στην κορυφή του δένδρου ονομάζεται στοιχείο ρίζα (<time-o-gram>). Επίσης ονομάζεται και στοιχείο του εγγράφου διότι περιλαμβάνει όλα τα υπόλοιπα στοιχεία και επιπλέον καθορίζει τα όρια του εγγράφου. Τα αντικείμενα στο τέλος κάθε αλυσίδας ονομάζονται φύλλα και αναπαριστούν το περιεχόμενο του εγγράφου. Κάθε αντικείμενο του δένδρου ονομάζεται κόμβος. Ωστόσο, υπάρχει ένα σημείο στην εικόνα 2-2 που δεν έχει ακόμα αναφερθεί: το πλαίσιο στα αριστερά με την ετικέτα pri. Στο XML έγγραφο περιέχονταν μέσα στο στοιχείο <time-o-gram>, ενώ στη δενδρική αναπαράσταση βρίσκεται εκτός αυτού. Ο κόμβος αυτός αναπαριστά ένα ειδικό είδος περιεχομένου που ονομάζεται ιδιότητα (attribute) και παρέχει επιπλέον πληροφορίες σχετικά με το στοιχείο. Η ιδιότητα αποτελείται από ένα όνομα (pri) και από μία τιμή (important). Οι ιδιότητες χρησιμοποιούνται κυρίως για να τροποποιούν την συμπεριφορά ενός στοιχείου και όχι για να περιέχουν δεδομένα.



Εικόνα 2-2 Η δενδρική αναπαράσταση του απλού XML παραδείγματος

Το στοιχείο ρίζα ονομάζεται και πρόγονος (ancestor) όλων των άλλων στοιχείων. Τα στοιχεία που βρίσκονται ακριβώς κάτω από το στοιχείο ρίζας αποτελούν τα παιδιά του (child elements). Αυτά με την σειρά τους δεν έχουν παιδιά και η διαδικασία συνεχίζεται μέχρι να φτάσουμε στο κόμβο που δεν έχει παιδί. Τα στοιχεία που μοιράζονται τον ίδιο γονέα ονομάζονται αδέρφια (siblings).



Εικόνα 2-3 Μερικά τμήματα του δένδρου

Κάθε κόμβος σε ένα δέντρο μπορεί να θεωρηθεί ως η ρίζα σε ένα μικρότερο υπό-δένδρο. Τα υπό-δένδρα έχουν όλες τις ιδιότητες ενός κανονικού δένδρου και η κορυφή κάθε υπό-δένδρου είναι ο πρόγονος όλων των κόμβων που βρίσκονται από κάτω. Στην εικόνα 2-3 παρουσιάζονται μερικά παραδείγματα υπό-δένδρων. [13]

4.7 XML declaration (δήλωση)

```
<? xml version='1.0' ?>
<? xml version='1.0' encoding='US-ASCII' ?>
<? xml version='1.0' encoding='US-ASCII' standalone='yes' ?>
<? xml version='1.0' encoding='UTF-8' ?>
<? xml version='1.0' encoding='UTF-16' ?>
<? xml version='1.0' encoding='ISO-10646-UCS-2' ?>
<? xml version='1.0' encoding='ISO-8859-1' ?>
<? xml version='1.0' encoding='Shift-JIS' ?>
```

Η δήλωση XML συνήθως εμφανίζεται ως η πρώτη γραμμή σε ένα έγγραφο XML . Η δήλωση XML δεν απαιτείται , ωστόσο , αν χρησιμοποιηθεί θα πρέπει να είναι η πρώτη γραμμή στο έγγραφο και κανένα άλλο περιεχόμενο ή λευκό κενό μπορεί να προηγείται.

Η δήλωση XML στο χάρτη έγγραφο αποτελείται από τα ακόλουθα:

- The version number, `<? xml version="1.0"?>`.
Αυτό είναι υποχρεωτικό . Παρά το γεγονός ότι ο αριθμός μπορεί να αλλάξει για μελλοντικές εκδόσεις της XML , 1.0 είναι η τρέχουσα έκδοση .
- The encoding declaration, `<? xml version="1.0" encoding="UTF-8"?>`
Αυτό είναι προαιρετικό . Αν χρησιμοποιηθεί , η δήλωση κωδικοποίησης πρέπει να εμφανίζεται αμέσως μετά την έκδοση πληροφορίες στη δήλωση XML , και πρέπει να περιέχει μια τιμή που αντιπροσωπεύει μια υπάρχουσα κωδικοποίηση χαρακτήρων.

Η δήλωση XML μπορεί επίσης να περιέχουν ένα αυτόνομο δήλωση , για παράδειγμα , `<?xml version="1.0" encoding="UTF-8" standalone ="yes"?>`. Όπως και η δήλωση κωδικοποίησης , η αυτόνομη δήλωση είναι προαιρετική . Αν χρησιμοποιηθεί , η αυτόνομη δήλωση πρέπει να εμφανίζεται τελευταία στη δήλωση XML .

Η δήλωση κωδικοποίησης προσδιορίζει ποια κρυπτογράφηση χρησιμοποιείται για την αναπαράσταση των χαρακτήρων του εγγράφου . Αν και ανάλυσης XML μπορεί να καθορίσει αυτόματα αν ένα έγγραφο θα χρησιμοποιεί το UTF-8 ή UTF-16 Unicode encoding, η δήλωση αυτή πρέπει να χρησιμοποιείται σε έγγραφα που υποστηρίζουν άλλες κωδικοποιήσεις .

Παράδειγμα:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

4.8 XML elements (στοιχεία)

Τα στοιχεία είναι τα βασικά δομικά στοιχεία ενός εγγράφου XML.

Σύνταξη ενός XML Element:

```
<html>
<body>
  <head>
<title>w3resource-xml</title>
</head>
  <body>
<p><b>This is w3resource.com.</b>The largest tutorial on web
based development.</p>
<p>We are learning <i>xml</i></p>
</body>
</html>
```

Παράδειγμα ενός xml αρχείου:

```
<xml>
<tutorial>w3resource
<one>html</one>
  <two>xml
    <subtopic1>Learningxml</subtopic1>
    <subtopic2>LearningDTD</subtopic2>
    <subtopic3>LearningXSLT</subtopic3>
    <subtopic4>Learning xpath</subtopic4>
  </two>
  <three>css</three>
  <four>java script</four>
  <five>ajax</five>
  <six>php</six>
  <seven>mysql</seven>
  <eight>svg</eight>
</tutorial>
</xml>
```

Here is a table displaying correct an incorrect examples of names of xml elements:

<tutorial></tutorial>	Correct
<topic2></topic2>	Correct
<3topic3></3topic3>	Incorrect since elements are begin with invalid characters

<code><three,topic3></three,topic3></code>	Incorrect since elements contain invalid characters
<code><threetopic3></threetopic3></code>	Correct
<code><three;topic3></three;topic3></code>	Incorrect since elements contain invalid characters
<code><three_topic3></three_topic3></code>	Correct
<code><three_topic:topic3></three_topic:topic3></code>	Incorrect, reference to undeclared namespace prefix
<code><!topic></ !topic></code>	Will generate error since the declaration has an invalid name
<code><[topic]></[topic]></code>	Will generate an error that complains that an element began with an invalid character.

Το όνομά στοιχείο που αρχίζει με μια ακολουθία χαρακτήρων XML , προορίζονται για μελλοντική χρήση .

Περιεχόμενο των στοιχείων μπορεί να είναι textdata , δεν υπάρχουν δεδομένα (αναφέρεται ως κενό στοιχείο) και άλλα στοιχεία (όπως ονομάζεται στοιχείο-παιδί).

4.9 XML attributes(χαρακτηριστικά)

Τα στοιχεία XML μπορεί να έχει χαρακτηριστικά , όπως HTML . Τα χαρακτηριστικά σχεδιασμένα να περιέχουν δεδομένα που σχετίζονται με ένα συγκεκριμένο στοιχείο . Οι αξίες των χαρακτηριστικών θα πρέπει πάντα να αναφέρονται. Είτε μονά ή διπλά εισαγωγικά μπορεί να χρησιμοποιηθεί . Για το φύλο ενός ατόμου , το < άτομο > μπορεί να γραφτεί ως εξής:

```
<person gender="female">
ή <person gender='female'>
```

Αν η ίδια η τιμή του χαρακτηριστικού περιέχει διπλά εισαγωγικά που μπορείτε να χρησιμοποιήσετε μονά εισαγωγικά , όπως σε αυτό το παράδειγμα :

```
<gangster name='George "Shotgun" Ziegler'>
ή μπορείτε να χρησιμοποιήσετε οντότητες χαρακτήρων :
<gangster name="George &quot;Shotgun&quot; Ziegler">
```

4.10 XML elements vs. attributes

```
<person gender="female">
<firstname>Anna</firstname>
<lastname>Smith</lastname>
</person>
```

```
<person>
<gender>female</gender>
<firstname>Anna</firstname>
<lastname>Smith</lastname>
</person>
```

Στο πρώτο παράδειγμα το φύλο είναι ένα χαρακτηριστικό . Στην τελευταία , το φύλο είναι ένα στοιχείο . Και τα δύο παραδείγματα παρέχουν τις ίδιες πληροφορίες . Δεν υπάρχουν κανόνες για το πότε να χρησιμοποιείτε τα χαρακτηριστικά ή πότε να χρησιμοποιήσετε τα στοιχεία σε XML.

4.11 Avoid XML attributes?

Μερικά πράγματα που πρέπει να εξετάσει, όταν χρησιμοποιείτε τα χαρακτηριστικά είναι :

- attributes cannot contain multiple values (elements can)
- attributes cannot contain tree structures (elements can)
- attributes are not easily expandable (for future changes)

Μην καταλήξετε σαν αυτό :

```
<note day="10" month="01" year="2008"
to="Tove" from="Jani" heading="Reminder"
body="Don't forget me this weekend!">
</note>
```

4.12 Σύγκριση μεταξύ των τεχνολογιών μοντελοποίησης DTD και XML Schema

Τα XML Schema παρέχουν μια αντικειμενοστραφή προσέγγιση για τον καθορισμό της δομής ενός XML εγγράφου και με αυτά μπορούν επίσης να καθοριστούν τύποι δεδομένων με βάση ένα φάσμα προκαθορισμένων τύπων που υπάρχουν στα Schema. Αν και τα XML Schema υπερτερούν των DTD υπάρχουν στην πραγματικότητα αρκετοί λόγοι για τους οποίους προτιμάται η χρήση των DTD αντί των XML Schema για τη μοντελοποίηση του περιεχομένου των εγγράφων XML.

Ένας από τους πιο σημαντικούς λόγους είναι ότι το XML Schema είναι μια νέα τεχνολογία. Αυτό έχει ως αποτέλεσμα να υπάρχουν περισσότερα προγράμματα

επεξεργασίας των XML εγγράφων που υποστηρίζουν το DTD. Επίσης οι οργανισμοί που χρησιμοποιούν για μεγάλο χρονικό διάστημα το διαδίκτυο για τη μεταφορά αρχείων μέσα από αυτό μπορεί να έχουν αναπτύξει τη μοντελοποίηση των εγγράφων τους με βάση το DTD.

Το DTD είναι ένα καθιερωμένο πρότυπο μοντελοποίησης XML εγγράφων και υπάρχουν πολλά παραδείγματα στο διαδίκτυο που είναι ελεύθερα διαθέσιμα για επαναχρησιμοποίηση και η δομή τους περιγράφεται από ένα DTD. Αυτά τα αρχεία DTD μπορούν να χρησιμοποιηθούν από κάθε χρήστη για να δημιουργήσει με βάση αυτά τα δικά του αρχεία DTD γρηγορότερα από ότι εάν επιχειρούσε να αναπτύξει από την αρχή ένα ολόκληρο DTD.

Τέλος, είναι σημαντικό να αναφερθεί ότι το XML Schema είναι και αυτό ένα έγγραφο XML. Περιέχει ένα XML Name-space που αναφέρεται σε αυτό το έγγραφο και ένα Schema που καθορίζει τη δομή του. Αυτά τα στοιχεία είναι πρόσθετη πληροφορία (overhead) για το Όταν ένας αναλυτής εξετάζει XML έγγραφο. Όταν ένας αναλυτής (parser) εξετάζει ένα XML έγγραφο που η δομή του περιγράφεται με ένα XML Schema, τότε πρέπει να συνδέσει όλα τα παραπάνω στοιχεία σε ένα έγγραφο, να ερμηνεύσει το Schema, να φορτώσει το name-space και να επικυρώσει το Schema, και όλα αυτά προτού ακόμα αναλύσει το αληθινό XML έγγραφο. Εάν χρησιμοποιείται η XML ως πρωτόκολλο επικοινωνίας μεταξύ δύο συστημάτων, η χρήση των οποίων είναι συχνή και είναι απαραίτητη η γρήγορη επεξεργασία των δεδομένων, τότε όλη αυτή η επιπλέον πληροφορία μπορεί να υποβαθμίσει σημαντικά την απόδοση αυτών των συστημάτων.

Επιπλέον, με τα XML Schema μπορούν να αναπτυχθούν κοινά λεξιλόγια με αποτέλεσμα να επιτρέπεται στις μηχανές να πραγματοποιήσουν διάφορους κανόνες που καθορίζονται από τους χρήστες. Παρέχουν τρόπους για τον προσδιορισμό της δομής, του περιεχομένου και της σημασιολογίας των XML εγγράφων. Εν κατακλείδι, τα XML Schema είναι πιο πλούσια και πιο ισχυρά για την περιγραφή και τη μοντελοποίηση των πληροφοριών από ότι είναι τα DTD. [14]

4.13 Βιβλιογραφία- αναφορά πηγής

[13]: πηγή από σύγγραμμα

- Η γλώσσα σήμανσης XML του επίκουρου καθηγητή Δημήτριου Σαμψών, Πανεπιστημιακές σημειώσεις του Πανεπιστημίου Πειραιώς, Τμήμα Διδακτικής της Τεχνολογίας και Ψηφιακών Συστημάτων

[14]: πηγή από διαδίκτυο

http://www.w3schools.com/xml/xml_elements.asp

Κεφάλαιο 5

5.1 XML path

5.1.1 Τι είναι η Xpath?

XPath (η XML γλώσσα Path) είναι μια γλώσσα για την εύρεση πληροφοριών σε ένα έγγραφο XML . Είναι μια σύνταξη για τον καθορισμό τμήματα ενός εγγράφου XML. Χρησιμοποιεί εκφράσεις μονοπατιού για να πλοηγηθείτε σε έγγραφα XML. Περιέχει μια βιβλιοθήκη τυπικές λειτουργίες. Είναι ένα σημαντικό στοιχείο για την XSLT. Χρησιμοποιείται επίσης σε XQuery , XPointer και Xlink. Η xpath είναι σύσταση του W3C.

5.1.2 XPath path expressions

Η XPath χρησιμοποιεί εκφράσεις μονοπατιού για να επιλέξετε κόμβους ή κόμβο - συνόλων σε ένα έγγραφο XML . Αυτές οι εκφράσεις μονοπατιού μοιάζουν πολύ με τις εκφράσεις που βλέπετε όταν εργάζεστε με ένα παραδοσιακό σύστημα αρχείων του υπολογιστή . Εκφράσεις XPath Σήμερα μπορεί επίσης να χρησιμοποιηθεί σε JavaScript , Java , XML Schema , PHP , Python , C και C ++ , και πολλές άλλες γλώσσες .

5.1.3 XPath is used in XSLT

Η XPath είναι ένα σημαντικό στοιχείο στο πρότυπο XSLT . Χωρίς XPath γνώση δεν θα είναι σε θέση να δημιουργήσετε έγγραφα XSLT .

5.1.4 XPath example

We will use the following XML document:

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
<book category="cooking">
  <title lang="en">Everyday Italian</title>
  <author>Giada De Laurentiis</author>
  <year>2005</year>
  <price>30.00</price>
</book>
<book category="children">
  <title lang="en">Harry Potter</title>
  <author>J K. Rowling</author>
```

```
<year>2005</year>
<price>29.99</price>
</book>
```

```
<book category="web">
  <title lang="en">XQuery Kick Start</title>
  <author>James McGovern</author>
  <author>Per Bothner</author>
  <author>Kurt Cagle</author>
  <author>James Linn</author>
  <author>Vaidyanathan Nagarajan</author>
  <year>2003</year>
  <price>49.99</price>
</book>
```

```
<book category="web">
  <title lang="en">Learning XML</title>
  <author>Erik T. Ray</author>
  <year>2003</year>
  <price>39.95</price>
</book>
```

```
</bookstore>
```

In the table below we have listed some XPath expressions and the result of the expressions

XPath Expression	Result
<code>/bookstore/book[1]</code>	Selects the first book element that is the child of the bookstore element
<code>/bookstore/book[last()]</code>	Selects the last book element that is the child of the bookstore element
<code>/bookstore/book[last()-1]</code>	Selects the last but one book element that is the child of the bookstore element
<code>/bookstore/book[position()<3]</code>	Selects the first two book elements that are children of the bookstore element
<code>//title[@lang]</code>	Selects all the title elements that have an attribute named lang
<code>//title[@lang='en']</code>	Selects all the title elements that have a "lang" attribute with a value of "en"
<code>/bookstore/book[price>35.00]</code>	Selects all the book elements of the bookstore element that have a price element with a value greater than 35.00
<code>/bookstore/book[price>35.00]/title</code>	Selects all the title elements of the book elements of the bookstore element that have a price element with a value greater than 35.00

5.2 XSLT

Με XSLT π μπορεί να μετατρέψει ένα έγγραφο XML σε HTML .

5.2.1 Displaying XML with XSLT

XSLT (Extensible Stylesheet Language Transformations) είναι η συνιστώμενη γλώσσα φύλλο στυλ για την XML . XSLT είναι πολύ πιο πολύπλοκη από CSS . Με XSLT , μπορείτε να προσθέσετε / αφαιρέσετε στοιχεία και χαρακτηριστικά προς ή από το αρχείο εξόδου . Μπορείτε επίσης να αλλάξετε τη διάταξη και να ταξινομήσετε τα στοιχεία , να διενεργούν ελέγχους και να λαμβάνουν αποφάσεις σχετικά με το ποια στοιχεία για να κρύψει και οθόνη , και πολλά άλλα .XSLT χρησιμοποιεί XPath για να βρείτε πληροφορίες σε ένα έγγραφο XML . [15]

5.2.2 XSLT example

We will use the following XML document:

```
<?xml version="1.0" encoding="UTF-8"?>
<breakfast_menu>

<food>
<name>Belgian Waffles</name>
<price>$5.95</price>
<description>Two of our famous Belgian Waffles with plenty of real maple syrup</description>
<calories>650</calories>
</food>

<food>
<name>Strawberry Belgian Waffles</name>
<price>$7.95</price>
<description>Light Belgian waffles covered with strawberries and whipped cream</description>
<calories>900</calories>
</food>

<food>
<name>Berry-Berry Belgian Waffles</name>
<price>$8.95</price>
<description>Light Belgian waffles covered with an assortment of fresh berries and whipped cream</description>
<calories>900</calories>
</food>

<food>
<name>French Toast</name>
<price>$4.50</price>
<description>Thick slices made from our homemade sourdough bread</description>
<calories>600</calories>
</food>

<food>
<name>Homestyle Breakfast</name>
<price>$6.95</price>
<description>Two eggs, bacon or sausage, toast, and our ever-popular hash browns</description>
<calories>950</calories>
</food>

</breakfast_menu>
```

Use XSLT to transform XML into HTML, before it is displayed in a browser:

Example XSLT Stylesheet:

```
<?xml version="1.0" encoding="UTF-8"?>
<html xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<body style="font-family:Arial;font-size:12pt;background-color:#EEEEEE">
<xsl:for-each select="breakfast_menu/food">
  <div style="background-color:teal;color:white;padding:4px">
    <span style="font-weight:bold"><xsl:value-of select="name"/> - </span>
    <xsl:value-of select="price"/>
  </div>
  <div style="margin-left:20px;margin-bottom:1em;font-size:10pt">
    <p>
      <xsl:value-of select="description"/>
      <span style="font-style:italic"> (<xsl:value-of select="calories"/> calories per serving)</span>
    </p>
  </div>
</xsl:for-each>
</body>
</html>
```

5.3 Βιβλιογραφία- αναφορά πηγής

[15]: πηγή από διαδίκτυο

http://www.w3schools.com/xsl/xpath_intro.asp

Κεφάλαιο 6

6.1 Περιγραφή μεταδεδομένων με την RDF

Η RDF είναι μια γλώσσα η οποία χρησιμοποιείται για την απλή περιγραφή πόρων (resources) του διαδικτύου. Η περιγραφή αυτή εστιάζεται κυρίως στην απόδοση μετα-πληροφορίας για τις οντότητες αυτές, όπως είναι η περιγραφή του τίτλου, του ονόματος, της ημερομηνίας δημιουργίας και άλλων χαρακτηριστικών κάποιου πόρου του διαδικτύου. Με την έννοια πόρος αναφερόμαστε σε οποιαδήποτε οντότητα του Παγκόσμιου ιστού, όπως είναι μια ιστοσελίδα, ένα τμήμα ή ένα σύνολο από ιστοσελίδες, ηλεκτρονικά αρχεία ή ακόμα και αντικείμενα τα οποία δεν είναι άμεσα διαθέσιμα στο διαδίκτυο, όπως είναι για παράδειγμα ένα βιβλίο. Η RDF βασίζεται στην ιδέα ότι οι πόροι οι οποίοι πρέπει να περιγραφούν έχουν ιδιότητες (properties) οι οποίες έχουν συγκεκριμένη τιμή. Έτσι λοιπόν μια μετα-πληροφορία για ένα πόρο αποτελείται από μια ιδιότητα και την τιμή που έχει ο πόρος για την ιδιότητα αυτή. Μια έκφραση για έναν πόρο ονομάζεται RDF πρόταση (sentence).

Συνοψίζοντας λοιπόν, μια RDF πρόταση αποτελείται από μια τριάδα (triple) ενός υποκειμένου (subject), ιδιότητας (property) και αντικειμένου (object). Τη θέση του υποκειμένου καταλαμβάνει ο πόρος, τη θέση της ιδιότητας η ιδιότητα που του αποδίδουμε, ενώ τη θέση του αντικειμένου η τιμή που έχει ο πόρος αυτός για την ιδιότητα. Η τιμή αυτή μπορεί να είναι κάποιος άλλος πόρος ή κάποια τιμή δεδομένων. Συντακτικά οι προτάσεις αυτές δηλώνονται από μια διατεταγμένη τριάδα της μορφής, $s \ p \ o$. Όπου τα s , p και o αντιπροσωπεύουν το υποκείμενο, την ιδιότητα και το αντικείμενο, αντίστοιχα ενώ η τριάδα τερματίζεται με το σύμβολο της τελείας. Ένα σύνολο από τριάδες RDF μπορούμε να το αντιληφθούμε και ως έναν γράφο. Σε αυτόν τον γράφο τα αντικείμενα και τα υποκείμενα παίζουν το ρόλο των κόμβων ενώ οι ιδιότητες παίζουν το ρόλο των ακμών τους συνδέουν. Όπως αναφέραμε και στην εισαγωγή η RDF είναι μια γλώσσα αναπαράστασης γνώσης για το Σημασιολογικό Ιστό. Έτσι λοιπόν το πρότυπο της RDF καθορίζει και μια σύνταξη η οποία έχει σαν σκοπό οι RDF τριάδες να δομούνται με έναν τρόπο επεξεργάσιμο από υπολογιστικά συστήματα και εφαρμογές. Η σύνταξη αυτή δε θα μπορούσε να βασίζεται σε άλλο πρότυπο παρά μόνο στην XML. Η σύνταξη αυτή αναφέρεται ως RDF/XML σύνταξη (Beckett 2003).

Για λόγους γενικότητας η RDF χρησιμοποιεί URI references για να προσδιορίσει τις οντότητες οι οποίες βρίσκονται στη θέση του υποκειμένου, της ιδιότητας και του αντικειμένου. Ένα URI reference (URIRef) αποτελείται από ένα URI και από ένα προαιρετικό fragment identifier. Για παράδειγμα το URIRef <http://www.example.org/index.html#section2> αποτελείται από το URI <http://www.example.org/index.html> και από τον fragment identifier Section2 τον οποίο διακρίνουμε από το URI με τη χρήση του συμβόλου #. Ας δούμε τώρα πως μπορούμε να ορίσουμε μετα-πληροφορία σε διάφορους πόρους χρησιμοποιώντας τη γλώσσα RDF. Έστω ότι θέλουμε να περιγράψουμε τη γνώση ότι η ιστοσελίδα

`http://www.example.org/index.html` έχει δημιουργό (creator) τον John Smith. Στην πρόταση αυτή μπορούμε να διακρίνουμε ένα υποκείμενο, τη σελίδα, ένα αντικείμενο τον John Smith και μια ιδιότητα, αυτή του δημιουργού της ιστοσελίδας. Αυτή η μετα-πληροφορία μπορεί να περιγραφεί σε μια τριάδα RDF ως εξής:

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

Όπου 85730 είναι ένα αντικείμενο που αντιπροσωπεύει τον John Smith.

Παρατηρούμε ότι η RDF χρησιμοποιεί URIs όχι μόνο για να προσδιορίσει τον πότο τον οποίο και περιγράφουμε αλλά και την ιδιότητα και το αντικείμενο. Όπως γίνεται αντιληπτό ίσως είναι λίγο άβολο να γράφουμε συνέχεια ολόκληρα τα URI των οντοτήτων που χρησιμοποιούμε. Έτσι λοιπόν πολλές φορές χρησιμοποιούνται συντομεύσεις σε μορφή προθεμάτων (prefixes) τα οποία ονομάζονται QNames. Θα χρησιμοποιήσουμε τα QNames, rdf, rdfs, dc, ex, exterms και exdateid για να αναπαραστήσουμε τα URIs `http://www.w3.org/1999/02/22-rdf-syntax-ns`, `http://www.w3.org/2000/01/rdf-schema`, `http://purl.org/dc/elements/1.1/`, `http://www.example.org/`, `http://www.example.org/terms/`, και `http://www.example.org/date/`, αντίστοιχα. Έτσι λοιπόν η παραπάνω τριάδα μπορεί να γραφτεί ως, `ex:index.html dc:creator exstaff:85730`. Όπως είναι φυσικό αν επιθυμούμε μπορούμε να προσδώσουμε επιπλέον μεταπληροφορίες για στον αρχικό μας πόρο. Για παράδειγμα μπορούμε να πούμε ότι η ημερομηνία δημιουργίας (creation-date) της ιστοσελίδας ήταν η 16^η Αυγούστου του 1999 ή ότι η γλώσσα στην οποία είναι γραμμένη είναι τα Αγγλικά. Οι πληροφορίες αυτές μπορούν να περιγραφούν με τις ακόλουθες τριάδες:

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

```
ex:index.html exterms:language "English".
```

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

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

6.2 Η RDF/XML μορφή

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

Ας εξηγήσουμε το παραπάνω RDF/XML κείμενο. Αρχικά παρατηρούμε ότι όπως σε όλα τα XML κείμενα έτσι και σε αυτό εισάγουμε τη δήλωση `<?xml version="1.0"?>` πριν από κάθε άλλο ορισμό. Εν συνεχεία το στοιχείο `rdf:RDF` δηλώνει ότι αυτό που

ακολουθεί είναι ένα σύνολο από RDF προτάσεις. Μέσα στο στοιχείο αυτό δηλώνουμε επίσης και τις συντομεύσεις που θα χρησιμοποιήσουμε για τα URIs.

Πιο συγκεκριμένα η συντόμευση `rdf` δηλώνει ότι οποιαδήποτε στοιχείο έχει ως πρόθεμα το `rdf:` αναφέρεται στο χώρο ονομάτων (*namespace*) που καθορίζεται από το URI <http://www.w3.org/1999/02/22-rdf-syntax-ns#>. Αντίστοιχα έχουμε ορίσει τα προθέματα `dc` και `exterms`. Στη συνέχεια ακολουθούν οι RDF/XML προτάσεις που δηλώνουν τις μετα-πληροφορίες που αναπαριστώνται από το γράφο του Σχήματος.

```
<exterms:language>English</exterms:language> <dc:creator
rdf:resource="http://www.example.org/staff/ffd8ffe000104a4649460001020100c800c80000ffe20c584943435f50524f46494c45000
10100000c484c696e6f021000006d6e74725247422058595a2007ce000200090006003
ffd10000616373704d534654000000004945432073524742000000000000000000000
/8500000f6d600010000000d32d48502020000000000000000000000000000000000
73000000000000000000000000000000000000000000000000000000000000000000
0"/0740000204000000147258595a00000218000000146758595a0000022c0000001462
>58595a0000024000000014646d6e6400002540000070646d646400002c4000008
8767565640000034c00000867669657700003d400000246c756d69000003f80000
00146d6561730000040c00000247465636800004300000000c725452430000043c0
000080c675452430000043c0000080c625452430000043c0000080c7465787400000
00436f70797269676874202863292031393938204865776c6574742d5061636b61726
420436f6d70616e7900006465736300000000000000012735247422049454336313936
362d322e310000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000
</rdf:Description> </rdf:RDF>
```

Σχήμα 1 Γράφος RDF προτάσεων.

Στη συνέχεια περιγράφουμε τις ιδιότητες και τις τιμές τις οποίες ο πόρος αυτός έχει για τις ιδιότητες αυτές. Πιο συγκεκριμένα στην έκτη γραμμή χρησιμοποιήσαμε ως *ετικέτα* (*tag*) το στοιχείο `exterms:create-date` για να δηλώσουμε την ιδιότητα που κατέχει ο πόρος, ενώ ως περιεχόμενο της ετικέτας, δηλαδή αυτό το οποίο βρίσκεται ανάμεσα στην ετικέτα εκκίνησης (*start-tag*) `<exterms:creation-date>` και την ετικέτα τερματισμού (*end-tag*) `</exterms:creation-date>`, αποτελεί και την τιμή του πόρου για την ιδιότητα αυτή, δηλαδή το αντικείμενο της τριάδας.

Με τον ίδιο ακριβώς τρόπο εργαζόμαστε και για την ιδιότητα που περιγράφει τη γλώσσα της ιστοσελίδας, όμως όπως παρατηρούμε η σύνταξη της τρίτης ιδιότητας είναι διαφορετική. Αυτό συμβαίνει γιατί η τιμή που είχαν οι προηγούμενες δύο ιδιότητες ήταν λεκτικά ενώ στην τρίτη ιδιότητα η τιμή είναι ένας άλλος πόρος. Για να ξεχωρίσουμε τις δύο αυτές περιπτώσεις, όπως είναι φυσικό χρησιμοποιούμε διαφορετική σύνταξη. Έτσι λοιπόν, στην περίπτωση αυτή χρησιμοποιούμε το QName `dc:creator` για να δηλώσουμε το όνομα της ιδιότητας όμως επιπλέον χρησιμοποιούμε το στοιχείο `rdf:resource` που παρέχει η γλώσσα RDF για να μπορέσουμε να δηλώσουμε το URIref του πόρου με τον οποίο συνδέετε ο πόρος τον οποίο και περιγράφουμε. Είναι σημαντικό στο σημείο αυτό να σημειώσουμε ότι ενώ μπορούμε να χρησιμοποιήσουμε τα QNames για την καταγραφή του ονόματος μιας ετικέτας, η χρήση τους στις τιμές των στοιχείων απαγορεύεται από το πρότυπο της RDF/XML (Beckett 2003). Το

σύνολο των στοιχείων το οποίο παρέχει η γλώσσα RDF για την περιγραφή γνώσης αποτελεί το *λεξιλόγιο (vocabulary)* της γλώσσας. Από το προηγούμενο παράδειγμα παρατηρούμε ότι η χρήση πόρων στη θέση των αντικειμένων μας δίνει τη δυνατότητα να δημιουργήσουμε αλυσίδες από τριάδες RDF όπου το αντικείμενο της μιας τριάδας εμφανίζεται ως υποκείμενο της άλλης. Αυτό το χαρακτηριστικό είναι αρκετά χρήσιμο καθώς μας δίνεται η δυνατότητα οι τιμές που αποδίδουμε στην ιδιότητα ενός πόρου να έχουν και αυτές με τη σειρά τους κάποια περιγραφή και άρα να αναπαριστούν μια περίπλοκη δομή.

Για παράδειγμα στις προηγούμενες τριάδες αποδώσαμε ως τιμή της ιδιότητας ημερομηνία δημιουργίας το λεκτικό August 16, 1999. Ίσως μια καλύτερη προσέγγιση να ήταν η ημερομηνία αυτή να αναπαρασταθεί με κάποιο πόρο, ο οποίος θα ονομάζεται ημερομηνία και ο οποίος θα έχει τις ιδιότητες, μήνας, ημέρα και έτος οι οποίες στη συγκεκριμένη περίπτωση θα έχουν τις τιμές August, 16 και 1999, αντίστοιχα. Έτσι λοιπόν μπορούμε να κάνουμε τις παρακάτω δηλώσεις. Η μορφή του RDF γράφου για τις παραπάνω δηλώσεις φαίνεται στο Σχήμα 2. Αυτός ο τρόπος γραφής όμως συνεπάγεται ότι οποτεδήποτε χρειαζόμαστε να αναφερθούμε σε μια ενδιάμεση οντότητα, όπως είναι η ημερομηνία, πρέπει να εισάγουμε έναν καινούργιο πόρο για την οντότητα αυτή και να του αποδώσουμε κάποιο URIref. Στην περίπτωση που ο πόρος αυτός χρησιμοποιείται μόνο μια φορά χωρίς να κατέχει κάποια παραπάνω χρησιμότητα στη γνώση μας ίσως να μην επιθυμούμε να εισάγουμε ρητά τον πόρο αυτόν αποδίδοντάς του κάποιο URIref. Σε αυτήν την περίπτωση η RDF μας παρέχει την έννοια του *κενού κόμβου (blank node)* (Klyne, G., Carroll, J. 2003). Διαισθητικά η ύπαρξη ενός κενού κόμβου σε μια θέση υποδηλώνει την παρουσία κάποιας οντότητας στη θέση αυτή χωρίς όμως να δηλώνεται ρητά, μέσω κάποιου URIref, ποια είναι αυτή η οντότητα. Γραφικά ένας κενός κόμβος αναπαριστάται χρησιμοποιώντας το οβάλ σχήμα που χρησιμοποιούμε για τους πόρους, χωρίς όμως να αναγράφουμε κάποιο URIref.

Όπως είναι προφανές ένα πολύπλοκο RDF κείμενο μπορεί να περιέχει πολλούς κενούς κόμβους.

Σε αυτήν την περίπτωση είναι απαραίτητο να διακρίνουμε του κενούς κόμβους μεταξύ τους. Γι αυτό το λόγο και η RDF χρησιμοποιεί τους λεγόμενους προσδιοριστές κενών κόμβων (*blank node identifiers*). Οι προσδιοριστές κενών κόμβων έχουν συντακτικά τη μορφή `_:name`, η οποία υποδηλώνει την ύπαρξη ενός κενού κόμβου και ταυτόχρονα δίνει και ένα όνομα στον κόμβο αυτό. Για παράδειγμα, αν χρησιμοποιήσουμε κενό κόμβο για την αναπαράσταση της ημερομηνίας οι τριάδες μας θα έχουν την ακόλουθη μορφή,


```

ffd8ffe000104a4649460001020100c800c80000ffe20c584943435f50524f46
494c4500010100000c484c696e6f021000006d6e74725247422058595a200
7ce00020009000600310000616373704d534654000000004945432073524
7420000000000000000000000000000000f6d600100000000d32d4850202000
000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000116370727400001500000003364
65736300001840000006c7774707400001f000000014626b70740000020
4000000147258595a0000021800000146758595a0000022c00000014625
8595a0000024000000014646d6e6400002540000070646d6464000002c
400000088767565640000034c000008676696577000003d4000000246c7
56d69000003f800000146d656173000040c000000247465636800000430
000000c725452430000043c0000080c675452430000043c0000080c62545
2430000043c0000080c746578740000000436f7079726967687420286329
2031393938204865776c6574742d5061636b61726420436f6d70616e79000
06465736300000000000000012735247422049454336313936362d322e310
000000000000000000000000012735247422049454336313936362d322e31000
000000000000000000000000000000000000000000000000000000000000

```

Σχήμα 2 Δημιουργία της δομής της ημερομηνίας.

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

```

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

```

Η δήλωση `rdf:parseType="Resource"` χρησιμοποιείται για να υποδείξει ότι τα περιεχόμενα ενός στοιχείου πρέπει να ερμηνευτούν ως η περιγραφή ενός νέου κενού κόμβου. Η περιγραφή αυτού του κενού κόμβου γίνεται χωρίς να απαιτείται η δημιουργία μιας RDF περιγραφής με τη χρήση του στοιχείου `rdf:Description`, όπως είδαμε στο προηγούμενο παράδειγμα.

Έτσι λοιπόν η δήλωση `rdf:parseType="Resource"` μας λέει ότι η τιμή της ιδιότητας `externs:creation-date` είναι ένας νέος κενός κόμβος και ότι οτιδήποτε ακολουθεί είναι οι τιμές του κενού κόμβου στις ιδιότητες που περιγράφονται. Αξίζει να αναφέρουμε στο σημείο αυτό ότι υπάρχουν και άλλοι τρόποι να αναπαραστήσουμε κενούς κόμβους στη σύνταξη RDF/XML.

Πιο συγκεκριμένα μπορούμε να γράψουμε:

```

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

```

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

Στο προηγούμενο παράδειγμα είδαμε πως μπορούμε να δημιουργήσουμε τη δομή ημερομηνία από τα επιμέρους στοιχεία της ημέρας, του μήνα και του έτους. Όταν όμως δώσαμε τιμή στην ημέρα και στο μήνα χρησιμοποιήσαμε και για τις δύο αυτές ιδιότητες ένα λεκτικό το οποίο είναι ουσιαστικά μια συμβολοσειρά.

Έτσι λοιπόν η τιμή “16” δεν αναπαριστά τον αριθμό δεκαέξι αλλά μια συμβολοσειρά της οποίας ο πρώτος χαρακτήρας είναι ο “1” και ο δεύτερος ο “6”. Αυτού του είδους τα λεκτικά στην ορολογία της RDF ονομάζονται απλά *λεκτικά (plain literals)*.

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

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

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

Παρατηρούμε ότι χρησιμοποιούμε το στοιχείο `rdf:datatype` για να προσδιορίσουμε τον τύπο δεδομένων που πρέπει να χρησιμοποιηθεί για να ερμηνεύσουμε την τιμή που ακολουθεί. Όπως αναφέραμε προηγουμένως η χρήση συντομεύσεων στις τιμές ενός στοιχείου απαγορεύεται. Στην περίπτωση όμως των τύπων δεδομένων της XML μπορούμε να χρησιμοποιήσουμε τα στοιχεία *XML (XML entities)* για να βελτιώσουμε την αναγνωσιμότητα των κειμένων μας. Ο τρόπος για να το πετύχουμε αυτό είναι ο

ακόλουθος. Αρχικά ορίζουμε τη συντόμευση xsd ως, `<!DOCTYPE rdf:RDF [!ENTITY xsd "http://www.w3.org/2001/XMLSchema#"]>`.

Στη συνέχεια μπορούμε να χρησιμοποιήσουμε τη συντόμευση αυτή γράφοντας `&xsd:date` αντί για ολόκληρο το URIref.

Μέχρι στιγμής είδαμε πως να χρησιμοποιούμε τις τριάδες RDF για να αποδίδουμε τιμές σε συγκεκριμένες ιδιότητες που κατέχει κάποιος πόρος του διαδικτύου. Αν θέλαμε να κάνουμε μια αντιστοίχιση με τις Περιγραφικές Λογικές, θα λέγαμε ότι μέχρι στιγμής έχουμε περιγράψει μόνο πως μπορούμε να δημιουργήσουμε ισχυρισμούς ρόλων.

Πιο συγκεκριμένα μια τριάδα $s \ p \ o$ μπορεί να θεωρηθεί ως ένας ισχυρισμός της μορφής $p(s,o)$, ο οποίος μας λέει ότι το άτομο s έχει την τιμή o στην ιδιότητα (ρόλο) p . Το ερώτημα που τίθεται, λοιπόν, είναι αν η RDF προσφέρει κάποιο μηχανισμό για την περιγραφή σχέσεων στιγμιοτύπου ανάμεσα σε κάποιον πόρο και σε κάποια έννοια (κλάση). Η λειτουργικότητα αυτή παρέχεται στην RDF μέσω του στοιχείου `rdf:type`. Όταν χρησιμοποιούμε το στοιχείο `rdf:type` στη θέση της ιδιότητας μιας τριάδας RDF ουσιαστικά δηλώνουμε ότι αυτό το οποίο ακολουθεί στη θέση του αντικειμένου αποτελεί μια κλάση της οποίας μέλος αποτελεί ο πόρος που βρίσκεται στη θέση του υποκειμένου.

Έτσι λοιπόν μπορούμε να πούμε ότι ο John Smith είναι κάποιος άνθρωπος γράφοντας την τριάδα, `ex:85730 rdf:type exterm:Person`. Η RDF/XML σύνταξη της τριάδας αυτής είναι ίδια με όσες έχουμε δει μέχρι στιγμής με τη μόνη διαφορά ότι στη θέση της ιδιότητας πρέπει να αναγραφεί το στοιχείο `rdf:type`. Το στοιχείο `rdf:type` όμως έχει και έναν εναλλακτικό πιο συνοπτικό τρόπο σύνταξης στη μορφή RDF/XML. Αυτός δημιουργείται αφαιρώντας το στοιχείο `rdf:type` και αντικαθιστώντας το `rdf:Description` με το QName της κλάσης.

Χρησιμοποιώντας αυτόν τον τρόπο γραφής στο προηγούμενο παράδειγμα έχουμε την ακόλουθη σύνταξη:

```
<exterm:Person rdf:about="ex:85730"/>
```

Όπως αναφέραμε και στο παράδειγμα της δημιουργίας της δομής της ημερομηνίας ένας πόρος που βρίσκεται στη θέση του αντικειμένου μπορεί σε μια άλλη τριάδα να εμφανιστεί στη θέση του αντικειμένου και έτσι μπορούμε να δημιουργήσουμε αλυσίδες από τριάδες RDF. Έτσι λοιπόν είναι καθ' όλα νόμιμο κάποιος να γράψει τις τριάδες `ex:Harry rdf:type ex:Eagle` και `ex:Eagle rdf:type ex:EndangeredSpeciesList`.

Η πρώτη τριάδα μας δηλώνει ότι ο Harry είναι στιγμιότυπο της κλάσης των αετών, δηλαδή μέλος του συνόλου των αετών, ενώ η δεύτερη τριάδα μας δηλώνει ότι η κλάση των αετών είναι στιγμιότυπο της κλάσης που αναπαριστά τη λίστα των ειδών που απειλούνται προς εξαφάνιση.

Αυτό σημαίνει ότι ο πόρος που αναπαριστά τους αετούς επενεργεί ταυτόχρονα και ως κλάση (έννοια) αλλά και ως άτομο. Η δυνατότητα δημιουργίας τέτοιων εκφράσεων ονομάζεται *μετα-μοντελοποίηση (metamodeling)* διότι η κλάση `EndangeredSpeciesList` είναι ουσιαστικά μια κλάση που επενεργεί πάνω σε μια άλλη κλάση, δηλαδή μια μετα-κλάση. Η RDF όμως προχωράει ένα βήμα παραπάνω επιτρέποντας σε έναν πόρο να

χρησιμοποιηθεί σε οποιαδήποτε θέση μιας τριάδας. Προφανώς αυτή η λειτουργικότητα δεν προσφέρεται από τις ΠΛ, οι οποίες διαχωρίζουν αυστηρά το σύνολο των ατόμων, των κλάσεων (εννοιών) και το σύνολο των ιδιοτήτων (ρόλων). Σε μερικές εφαρμογές χρειαζόμαστε τη δυνατότητα να μπορούμε να δημιουργούμε προτάσεις οι οποίες αναφέρονται σε άλλες προτάσεις RDF.

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

Μπορούμε επίσης να περιγράψουμε πληροφορία όπως για παράδειγμα το ποιος δημιούργησε την πρόταση αυτή γράφοντας `ex:triple100` `dc:creator` `exstaff:85730`.

Η μορφή που έχουν οι προτάσεις αυτές σε RDF/XML σύνταξη είναι η ακόλουθη:

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

Ολοκληρώνοντας την παρουσίασή μας στην RDF αξίζει να σημειώσουμε ότι η RDF παρέχει επιπλέον στοιχεία για την περιγραφή συνόλων (*bags*), ακολουθιών (*sequences*), εναλλακτικών επιλογών (*alternatives*) και συλλογών (*collections*) πόρων. Το RDF λεξιλόγιο που παρέχεται για τον ορισμό αυτών των δομών είναι τα στοιχεία, `rdf:bag`, `rdf:seq` και `rdf:alt` για τη δημιουργία συνόλων, ακολουθιών και εναλλακτικών επιλογών, αντίστοιχα και τα στοιχεία, `rdf:first`, `rdf:rest` και `rdf:nil` για τη δημιουργία συλλογών. Για παράδειγμα μπορούμε να ορίσουμε ένα σύνολο από φοιτητές χρησιμοποιώντας τον παρακάτω τμήμα δήλωσης RDF/XML:

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

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

6.3 Απλή αναπαράσταση με τη γλώσσα RDF-S

Η RDF μας παρέχει τη δυνατότητα να δημιουργήσουμε απλές προτάσεις για τους πόρους τους οποίους θέλουμε να περιγράψουμε χρησιμοποιώντας ιδιότητες, τιμές και URIref για τον προσδιορισμό των συστατικών που συμμετέχουν σε μια πρόταση. Η RDF όμως δεν παρέχει δυνατότητα να ορίσουμε και να περιγράψουμε ένα επιπλέον λεξιλόγιο το οποίο πιθανόν να επιθυμούμε να χρησιμοποιήσουμε στις εφαρμογές μας. Πιο συγκεκριμένα δεν έχουμε τη δυνατότητα να ορίσουμε τις κλάσεις (έννοιες) οι οποίες εμφανίζονται σε μια εφαρμογή.

Επιπρόσθετα, είναι φυσικό να επιθυμούμε την περιγραφή των κλάσεων και των ιδιοτήτων μας δηλώνοντας σχέσεις υπαγωγής ανάμεσά τους. Η γλώσσα η οποία παρέχει τη λειτουργικότητα αυτή είναι η RDF-S (**Brickey, D., Guha, R.V., 2000**). Ουσιαστικά η RDF-S παρέχει ένα επιπλέον λεξιλόγιο πάνω σε αυτό της RDF το οποίο περιλαμβάνει στοιχεία τα οποία προορίζονται στο να προσδώσουν την επιπρόσθετη αυτή λειτουργικότητα.

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

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

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

```
ex:Van rdfs:subClassOf ex:MotorVehicle
ex:Truck rdfs:subClassOf ex:MotorVehicle
```

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

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

```

xml:base="http://example.org/schemas/vehicles">
<rdfs:Class rdf:ID="MotorVehicle"/>
<rdfs:Class rdf:ID="Truck">
  <rdfs:subClassOf rdf:resource="#MotorVehicle"/> </rdfs:Class>
<rdfs:Class rdf:ID="Van">
  <rdfs:subClassOf rdf:resource="#MotorVehicle"/> </rdfs:Class>

```

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

Εκτός από τη δυνατότητα περιγραφής κλάσεων, μέσω σχέσεων υπαγωγής, η RDF-S παρέχει επιπλέον τη δυνατότητα δημιουργίας περιγραφών για τις ιδιότητες που επιθυμούμε να χρησιμοποιήσουμε σε μια εφαρμογή μας.

Πιο συγκεκριμένα μπορούμε να περιγράψουμε σχέσεις υπαγωγής ανάμεσα σε δύο ιδιότητες, αλλά επιπρόσθετα μπορούμε να ορίσουμε το πεδίο ορισμού και το πεδίο τιμών μιας σχέσης. Το πεδίο ορισμού μας δηλώνει το τι τύπου επιτρέπεται να είναι οι πόροι οι οποίοι βρίσκονται στη θέση του υποκειμένου μιας τριάδας ενώ το πεδίο τιμών μας λέει τι τύπου είναι οι πόροι οι οποίοι βρίσκονται στη θέση του αντικειμένου. Τα στοιχεία που παρέχονται από την RDF-S για την υλοποίηση της λειτουργικότητας αυτής είναι τα στοιχεία `rdfs:subPropertyOf`, `rdfs:domain` και `rdfs:range`.

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

Τέλος όπως είναι αντιληπτό η ιδιότητα `ex:driver` συνδέει ένα όχημα με έναν άνθρωπο. Η γνώση αυτή μπορεί να περιγραφεί με τις τριάδες, `ex:driver rdfs:domain ex:MotorVehicle` και `ex:driver rdfs:range ex:Person`.

Η RDF/XML μορφή των παραπάνω προτάσεων είναι η ακόλουθη:

```

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

```

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

6.4 Σημασιολογία των RDF και RDF-S

Όπως κάθε γλώσσα αναπαράστασης γνώσης έτσι και για τις γλώσσες RDF και RDF-S ορίζεται τυπική σημασιολογία (formal semantics) η οποία αποσκοπεί στο να αποδώσει νόημα (σημασία) στο λεξιλόγιο των γλωσσών RDF και RDF-S αλλά και σημασία στις προτάσεις (τριάδες) RDF (Hayes 2003).

Στην περίπτωση της σημασιολογίας των Περιγραφικών Λογικών, η κατάσταση ήταν σχετικά απλή και ξεκάθαρη. Υπενθυμίζουμε ότι μία ερμηνεία σε μια ΠΛ αντιστοιχεί ένα άτομο σε ένα αντικείμενο, μια έννοια σε ένα σύνολο αντικειμένων και έναν ρόλο (ιδιότητα) σε ένα σύνολο από ζευγάρια αντικειμένων του χώρου ερμηνείας.

Στην περίπτωση όμως των RDF και RDF-S η κατάσταση είναι πολύ πιο περίπλοκη. Αυτό συμβαίνει λόγω των δυνατοτήτων της μετα-μοντελοποίησης που παρέχει η RDF. Υπενθυμίζουμε ότι στη μετα-μοντελοποίηση ένας πόρος μπορεί να παίζει τόσο το ρόλο μιας κλάσης, μιας ιδιότητας αλλά και ενός ατόμου. Αυτός ο πολυμορφικός χαρακτήρας των πόρων καταστεί δύσκολη την ερμηνεία τους καθώς σε μια τέτοια περίπτωση αφενός κάποιος πόρος θα πρέπει να ερμηνευτεί ως αντικείμενο του χώρου ερμηνείας αλλά επιπλέον θα πρέπει να ερμηνευτεί και ως σύνολο αντικειμένων ή ακόμα και σύνολο ζευγαριών αντικειμένων. Η θεωρία μοντέλων (model theory) της RDF είναι αρκετά περίπλοκη και δύσκολη.

Έτσι λοιπόν η παρουσίασή μας θα εστιαστεί στις βασικές έννοιες και τα συστατικά της θεωρίας αυτής. Ο αναγνώστης μπορεί στη συνέχεια να απευθυνθεί στο (Hayes 2003) για μια πλήρη παρουσίαση της θεωρίας μοντέλων της RDF.

Όπως αναφέραμε οι πόροι σε ένα RDF αρχείο μπορούν να κατέχουν διπλό ρόλο. Για να μπορέσουμε να ξεπεράσουμε τη δυσκολία αυτή οι ερμηνείες των προτάσεων και του λεξιλογίου των γλωσσών RDF και RDF-S βασίζονται σε μια απλή αλλά ταυτόχρονα περιεκτική ιδέα. Οι RDF ερμηνείες πραγματοποιούνται σε δύο στάδια.

Στο πρώτο στάδιο όλοι οι πόροι ανεξάρτητα με το αν αναπαριστούν άτομα, κλάσεις ή ιδιότητες ερμηνεύονται (αντιστοιχούνται) σε αντικείμενα ενός χώρου ερμηνείας. Στο δεύτερο στάδιο για κάθε ένα από τα αντικείμενα αυτά προσδίδουμε μια επέκταση (*extension*) η οποία ουσιαστικά τους αποδίδει το τελικό τους νόημα.

Έστω για παράδειγμα η τριάδα `c rdf:type c` η οποία δηλώνει ότι το `c` είναι στιγμιότυπο του εαυτού του. Αυτό σημαίνει ότι το `c` συμπεριφέρεται τόσο ως άτομο όσο και ως κλάση. Σε πρώτο βήμα χρησιμοποιώντας μια συνάρτηση ερμηνείας *IS*, αντιστοιχίζουμε τον πόρο `c` σε ένα άτομο του χώρου ερμηνείας *IR*, δηλαδή δημιουργούμε την

αντιστοίχιση $c \rightarrow IS(c)$ όπου $IS(c) \in IR$. Αυτό συνεπάγεται ότι όσον αφορά την εμφάνιση του c ως άτομο του έχουμε αποδώσει νόημα (ερμηνεία). Σε δεύτερο βήμα χρησιμοποιούμε μια δεύτερη συνάρτηση ερμηνείας, την ICEXT, η οποία αποδίδει μια επέκταση στο αντικείμενο $IS(c)$. Εφόσον το c συμπεριφέρεται και ως κλάση η επέκταση αυτή θα αντιστοιχίσει στο αντικείμενο $IS(c)$ ένα σύνολο αντικειμένων του χώρου ερμηνείας. Αναθέτουμε δηλαδή στο αντικείμενο $IS(c)$ το σύνολο ICEXT($IS(c)$).

Έτσι λοιπόν στο δεύτερο αυτό βήμα προσδίδουμε νόημα και στην έκφραση του πόρου c ως κλάση. Με τον ίδιο τρόπο το c θα μπορούσε σε κάποια άλλη πρόταση RDF να εμφανίζεται ως ιδιότητα. Σε αυτήν την περίπτωση θα χρησιμοποιούσαμε μια άλλη συνάρτηση ερμηνείας, την IEXT, η οποία και θα αντιστοιχίσει στο αντικείμενο $IS(c)$ ένα σύνολο από ζευγάρια αντικειμένων του χώρου ερμηνείας. Στο Σχήμα 3 φαίνεται σχηματικά ένα παράδειγμα μιας τέτοιας ερμηνείας για τις τριάδες c rdf:type c . και c d . οι οποίες δηλώνουν ότι το c είναι στιγμιότυπο του εαυτού του και ότι επίσης συμπεριφέρεται ως ιδιότητα που συνδέει τον εαυτό της με κάποιον άλλο πόρο d .

Όπως φαίνεται από το σχήμα ο πόρος c έχει ερμηνευτεί τόσο ως αντικείμενο ($IS(c)$), κλάση η οποία και περιέχει ως μέλος της την ερμηνεία του c ως άτομο, δηλαδή ισχύει $IS(c) \in ICEXT(IS(c))$, αλλά και ως ιδιότητα η οποία περιέχει μια σύνδεση από το αντικείμενο $IS(c)$ στο αντικείμενο $IS(d)$, δηλαδή ισχύει $(IS(c), IS(d)) \in IEXT(IS(c))$. [16]

```
ffd8ffe000104a4649460001020100c800c80000ffe20c5
84943435f50524f46494c450001010000c484c696e6f
021000006d6e74725247422058595a2007ce00020009
000600310000616373704d5346540000000049454320
7352474200000000000000000000000000f6d60001
00000000d32d485020200000000000000000000000
000000000000000000000000000000000000000000
0000000000000000000000000000116370727400000150
000000336465736300000184d000006c777470740000
01f00000014626b7074000020400000147258595a
0000021800000146758595a0000022c000000146258
595a0000024000000014646d6e640000025400000070
646d6464000002c40000088767565640000034c0000
008676696577000003d4000000246c756d69000003f8
000000146d6561730000040c00000024746563680000
04300000000c725452430000043c0000080c67545243
0000043c0000080c625452430000043c0000080c7465
78740000000436f7079726967687420286329203139
3938204865776c6574742d5061636b61726420436f6d
70616e79000064657363000000000000001273524742
2049454336313936362d322e31000000000000000000
000012735247422049454336313936362d322e310000
000000000000000000000000000000000000000000
000000
```

Σχήμα 3 Σύνθετη ερμηνεία της θεωρίας μοντέλων της RDF

6.5 Βιβλιογραφία- αναφορά πηγής

[16]: πηγή από διαλέξεις

<http://www.image.ece.ntua.gr/~gstoil/RDF-OWL.pdf>

Κεφάλαιο 7

7.1 Εκφραστική αναπαράσταση με την OWL

Στις προηγούμενες ενότητες είδαμε τις γλώσσες RDF και RDF-S. Είδαμε ότι το λεξιλόγιο των γλωσσών αυτών προσφέρει μια πολύ βασική εκφραστική δυνατότητα. Οι γλώσσες αυτές έχουν ουσιαστικά ως σκοπό να αποτελέσουν το θεμέλιο για πιο εκφραστικές γλώσσες ανώτερων επιπέδων, όπως είναι οι γλώσσες του λογικού επιπέδου. Μιλώντας με όρους Περιγραφικής Λογικής είδαμε ότι οι RDF και RDF-S παρέχουν τη δυνατότητα δημιουργίας σχέσεων υπαγωγής μεταξύ κλάσεων (δηλαδή εννοιών) και ιδιοτήτων (δηλαδή ρόλων) αλλά και τη δυνατότητα δημιουργίας ισχυρισμών.

Επιπρόσθετα, η RDF-S παρέχει τη δυνατότητα καθορισμού του πεδίου ορισμού και του πεδίου τιμών μιας ιδιότητας με τη μορφή αξιωμάτων. Όπως θα δούμε στη συνέχεια τα δύο αυτά αξιώματα μπορούν να περιγραφούν ως δύο αξιώματα υπαγωγής στις ΠΛ. Οι RDF και RDF-S, λοιπόν, δεν παρέχουν τη δυνατότητα να ορίσουμε κλάσεις από την ένωση, την τομή ή την άρνηση μιας ή και περισσότερων άλλων κλάσεων. Οι δυνατότητες αυτές και πολλές ακόμα παρέχονται από τις ΠΛ. Άρα οι ΠΛ είναι ένας καλός υποψήφιος για την δημιουργία μιας γλώσσας αναπαράστασης γνώσης για το λογικό επίπεδο. Όμως, όπως έχουμε αναφέρει ήδη μια τέτοια γλώσσα θα πρέπει να πληροί ορισμένες προϋποθέσεις για να μπορέσει να χρησιμοποιηθεί στο Σημασιολογικό Ιστό.

Πιο συγκεκριμένα θα πρέπει να διαθέτει μια μορφή σύνταξης η οποία να συμβαδίζει με το πρότυπο XML. Επιπρόσθετα η γλώσσα αυτή θα πρέπει να είναι συμβατή με τις γλώσσες των χαμηλότερων επιπέδων, δηλαδή τις RDF και RDF-S, αλλά ακόμα να είναι κατανοητή και εύκολη στη χρήση. Η νέα γλώσσα η οποία έρχεται να υλοποιήσει τη λειτουργικότητα του λογικού επιπέδου είναι η OWL (Web Ontology Language) (Bechhofer et. al., 2004). Όπως θα δούμε στη συνέχεια, η OWL έχει πολύ μεγάλη σχέση με εκφραστικές ΠΛ όπως οι γλώσσες *SHOIN(D)* και *SHIF(D)*. Το πρότυπο της OWL καθορίζει ουσιαστικά τρεις υπογλώσσες αυξανόμενης εκφραστικής δυνατότητας.

Οι γλώσσες αυτές είναι οι ακόλουθες:

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

- **OWL DL:** Η γλώσσα αυτή δίνει τη μέγιστη εκφραστική δυνατότητα που προσφέρεται από τη γλώσσα OWL χωρίς όμως να χάνονται οι καλές υπολογιστικές ιδιότητές της. Αυτό σημαίνει ότι η γλώσσα αυτή, σε αντίθεση με την τελευταία υπογλώσσα της OWL, είναι *αποφασίσιμη (decidable)*. Συγκριτικά με τις ΠΛ, η OWL DL παρέχει την ίδια εκφραστική δυνατότητα με τη γλώσσα *SHOIN(D)*.
- **OWL Full:** Η γλώσσα αυτή προσφέρει το ίδιο λεξιλόγιο με τη γλώσσα OWL DL. Επιπρόσθετα όμως παρέχει τη συντακτική ελευθερία και τα χαρακτηριστικά της γλώσσας RDF και πιο συγκεκριμένα τη δυνατότητα μεταμοντελοποίησης. Η γλώσσα αυτή είναι εμφανώς *μη-αποφασίσιμη (undecidable)* (**Boris 2005**).

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

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

Αντίστοιχα με τις ΠΛ η OWL περιλαμβάνει ένα αλφάβητο το οποίο αποτελείται από *κλάσεις (classes)*, *ιδιότητες (properties)* και *άτομα (individuals)*. Οι κλάσεις της OWL αποτελούν ένα ανάλογο των εννοιών των Περιγραφικών Λογικών. Έτσι λοιπόν διασθητικά μια κλάση αναπαρηστά ένα σύνολο από αντικείμενα τα οποία έχουν κοινά χαρακτηριστικά, όπως για παράδειγμα η κλάση των ανθρώπων ή η κλάση των αυτοκινήτων. Όπως οι ΠΛ έτσι και η OWL περιλαμβάνει τόσο ατομικές κλάσεις όσο και *περιγραφές κλάσεων (class descriptions)*. Επιπρόσθετα, παρέχονται και *αξιώματα κλάσεων (class axioms)* όπως αξιώματα υπαγωγής, ισοδυναμίας αλλά και πολλά ακόμα που θα δούμε στην συνέχεια. Οι ιδιότητες είναι το αντίστοιχο των ρόλων στις ΠΛ, έτσι λοιπόν αναπαριστούν δυαδικές σχέσεις, δηλαδή ζευγάρια αντικειμένων.

Επιπρόσθετα η OWL προσφέρει τη δυνατότητα ορισμού αξιωμάτων ιδιοτήτων (property axioms), όπως αξιώματα μεταβατικών ρόλων αλλά και άλλα αξιώματα τα οποία δεν εμφανίζονται άμεσα παρά μόνο έμμεσα στις ΠΛ. Τέλος η OWL προσφέρει τη δυνατότητα ορισμού ισχυρισμών, οι οποίοι στην περίπτωση της OWL ονομάζονται *γεγονότα (facts)*.

7.2 Περιγραφές και αξιώματα κλάσεων

Η πρώτη δυνατότητα που μας παρέχει η OWL είναι η περιγραφή κλάσεων ως η *τομή* (*intersection*) μιας λίστας άλλων περιγραφών κλάσεων. Σε αφηρημένη σύνταξη μια τέτοια δήλωση έχει τη μορφή `intersectionOf(C1, C2, ..., Cn)`, όπου τα C_i , $1 \leq i \leq n$ είναι περιγραφές κλάσεων.

Έτσι λοιπόν μια τέτοια έκφραση δηλώνει την κλάση η οποία είναι η τομή όλων των κλάσεων C_1, C_2, \dots, C_n . Η αντίστοιχη περιγραφή έννοιας στις ΠΛ θα ήταν η `C1ΠC2Π...ΠCn`. Όπως και στην περίπτωση των ΠΛ μια τέτοια έκφραση απλώς δηλώνει την ύπαρξη μιας κλάσης χωρίς να της δίνει κάποιο όνομα. Αν θέλουμε να ονομάσουμε την κλάση αυτή θα πρέπει να χρησιμοποιήσουμε τα αξιώματα της OWL τα οποία θα δούμε αργότερα. Με παρόμοιο τρόπο η OWL μας προσφέρει τη δυνατότητα δημιουργίας περιγραφών κλάσεων από την *ένωση* (*union*) μιας λίστας περιγραφών κλάσεων ή το *συμπλήρωμα* (*complement*) μιας περιγραφής κλάσης. Στην πρώτη περίπτωση η αφηρημένη σύνταξη είναι η εξής, `unionOf(C1, C2, ..., Cn)`, ενώ στη δεύτερη περίπτωση η σύνταξη είναι η εξής, `complementOf(C)`. Το συμπλήρωμα μιας κλάσης περιγραφής δηλώνει την κλάση η οποία περιλαμβάνει όλα τα αντικείμενα τα οποία δεν ανήκουν στην κλάση C .

Όπως οι ΠΛ έτσι και η OWL παρέχει περιγραφές κλάσεων μέσω περιορισμού του τύπου και του πλήθους των τιμών που μπορεί να πάρει μια ιδιότητα. Η μέθοδος αυτή ονομάζεται *περιορισμός ιδιότητας* (*property restriction*). Σε αφηρημένη σύνταξη ένας περιορισμός ιδιότητας δηλώνεται με τη λέξη κλειδί (*keyword*) `restriction`. Στη συνέχεια δηλώνουμε την ιδιότητα στην οποία εφαρμόζουμε τον περιορισμό και τέλος δηλώνουμε τον περιορισμό που επιθυμούμε.

Πιο συγκεκριμένα η OWL προσφέρει τον καθολικό και τον υπαρξιακό περιορισμό ιδιότητας. Οι περιορισμοί αυτοί δηλώνονται με τις λέξεις κλειδιά `allValuesFrom` και `someValuesFrom` ακολουθούμενες με από μια περιγραφή κλάσης. Για παράδειγμα αν θέλουμε να δηλώσουμε την κλάση των αντικειμένων τα οποία έχουν ένα παιδί το οποίο είναι Θηλυκό τότε σε αφηρημένη σύνταξη μπορούμε να γράψουμε, `restriction(hasChild someValuesFrom(Female))`.

Παρόμοια, η κλάση των αντικειμένων που όλα τους τα παιδιά, αν έχουν, είναι είτε ψηλά είτε έξυπνα δηλώνεται ως, `restriction(hasChild allValuesFrom(unionOf(Tall, Smart)))`. Η αντίστοιχη περιγραφή έννοιας στις ΠΛ είναι η `∇hasChild.(Tall∪Smart)`.

Επιπρόσθετα, η OWL παρέχει περιορισμούς πληθυκότητας. Όπως και οι ΠΛ η OWL προσφέρει τους τελεστές *το-πολύ* και *το-λιγότερο* όμως, επιπλέον, παρέχει και έναν άλλο τελεστή με τον οποίο μπορούμε να δηλώσουμε μια κλάση αντικειμένων περιγράφοντας την ακριβή πληθυκότητα των μελών της σε μια ιδιότητα. Οι κατασκευαστές αυτοί δηλώνονται με τις λέξεις κλειδιά, `minCardinality`, `maxCardinality` και `cardinality`, αντίστοιχα, ακολουθούμενες από έναν φυσικό αριθμό ο οποίος δηλώνει την πληθυκότητα. Για παράδειγμα μπορούμε να δηλώσουμε την κλάση των αντικειμένων που περιέχουν το λιγότερο 4 παιδιά ή την κλάση των αντικειμένων που έχουν ακριβώς δύο αυτοκίνητα ως, `restriction(hasChild minCardinality(4))` και `restriction(hasCar cardinality(2))`.

Τέλος μπορούμε να περιγράψουμε κλάσεις ανάλογα με το αν συμμετέχουν σε μια σχέση με κάποιο συγκεκριμένο αντικείμενο. Σε αφηρημένη σύνταξη ο περιορισμός αυτός δηλώνεται με τη λέξη κλειδί value ακολουθούμενη από το όνομα του αντικειμένου. Για παράδειγμα, μπορούμε να περιγράψουμε την κλάση των αντικειμένων που συμμετέχουν με το αντικείμενο Αγγλία στη σχέση citizen-of.

Διαισθητικά αυτό περιγράφει την κλάση που είναι κάτοικοι της Αγγλίας και η μορφή της σε αφηρημένη σύνταξη είναι η εξής, $restriction(citizen-of\ value(England))$. Παρατηρήστε ότι στην προκειμένη περίπτωση η λέξη England είναι ένα άτομο και όχι μια κλάση. Σε σύνταξη ΠΛ αυτό είναι ανάλογο με την ονοματική έννοια {England}, ενώ η παραπάνω κλάση αντιστοιχεί στην ΠΛ έννοια $\exists citizen-of.\{England\}$.

Τέλος μας παρέχεται η δυνατότητα ορισμού κλάσεων με *απαρίθμηση (enumeration)* των μελών τους. Ένα παράδειγμα είναι ο ορισμός της έννοιας (στην περίπτωση μας κλάσης) των ημερών της εβδομάδας, απαριθμώντας τα μέλη της, δηλαδή τη Δευτέρα, την Τρίτη κ.ο.κ. Στην OWL μια τέτοια κλάση μπορεί να οριστεί με τη εξής δήλωση σε αφηρημένη σύνταξη, $oneof(o1, o2, \dots, on)$, όπου τα $o1, o2, \dots, on$ είναι άτομα. Η δήλωση μιας τέτοιας έννοιας σε μια ΠΛ γλώσσα η οποία διαθέτει τον κατασκευαστή της ονοματικής έννοιας και τον κατασκευαστή της ένωσης έχει την εξής σύνταξη, $\{Δευτέρα\} \sqcup \{Τρίτη\} \sqcup \dots \sqcup \{Κυριακή\}$.

Ας δούμε τώρα τι μορφές αξιωμάτων κλάσεων μας προσφέρει η OWL. Όπως είναι αναμενόμενο η OWL μας δίνει τη δυνατότητα να ορίσουμε αξιώματα υπαγωγής και αξιώματα ισοδυναμίας. Η δυνατότητα περιγραφής σχέσεων υπαγωγής ήταν ήδη διαθέσιμη από την RDF-S μέσω του στοιχείου `rdfs:subClassOf`. Έτσι λοιπόν η OWL χρησιμοποιεί το στοιχείο αυτό για να ορίσει σχέσεις υπαγωγής ανάμεσα σε δύο κλάσεις. Η αφηρημένη σύνταξη των δηλώσεων αυτών είναι η $subClassOf(C1, C2)$.

Εκτός όμως από αυτήν την απλή μέθοδο δήλωσης μιας σχέσης υπαγωγής η OWL μας προσφέρει και μια επέκτασή της. Πιο συγκεκριμένα μπορούμε να δηλώσουμε μια σχέση υπαγωγής ανάμεσα σε μια κλάση και σε μια τομή από περιγραφές κλάσεων. Η δήλωση αυτή έχει τη μορφή $Class(A\ partial\ C1, C2, \dots, Cn)$ ενώ σε ΠΛ η μορφή της είναι η εξής, $A \sqsubseteq C1 \sqcap C2 \sqcap \dots \sqcap Cn$. Έτσι λοιπόν συνδυάζοντας περιγραφές κλάσεων και το παραπάνω αξίωμα μπορούμε να δημιουργήσουμε μια δήλωση της μορφής, $Class(A\ partial\ unionOf(C, D), restriction(R\ allValuesFrom(F)))$ η οποία έχει την ίδια σημασία με την δήλωση $A \sqsubseteq (C \sqcup D) \sqcap \forall R.F$, σε σύνταξη ΠΛ.

Αντίθετα από την RDF και RDF-S και παρόμοια με τις ΠΛ η OWL μας προσφέρει τη δυνατότητα να ορίσουμε σχέσεις ισοδυναμίας ανάμεσα σε κλάσεις. Πιο συγκεκριμένα μας δίνεται η δυνατότητα να ορίσουμε ισοδυναμίες ανά δύο σε μια λίστα από n κλάσεις. Οι δηλώσεις αυτές έχουν τη μορφή $EquivalentClasses(C1, C2, \dots, Cn)$ ενώ η σημασιολογία τους είναι η ίδια με τη σημασιολογία που προσφέρεται από τις δηλώσεις $C_i \equiv C_j, 1 \leq i < j \leq n$ σε μια ΠΛ. Όπως και στην περίπτωση των σχέσεων υπαγωγής έτσι και τώρα έχουμε τη δυνατότητα να ορίσουμε μια κλάση ως ισοδύναμη με την τομή μιας λίστας κλάσεων. Η σύνταξη των αξιωμάτων αυτών είναι η ακόλουθη, $Class(A\ complete\ C1, C2, \dots, Cn)$, ενώ η σημασιολογία του αξιώματος αυτού είναι ίδια με αυτή του αξιώματος $A \equiv C1 \sqcap C2 \sqcap \dots \sqcap Cn$ στις ΠΛ.

Επιπρόσθετα χρησιμοποιώντας την περιγραφή κλάσεων με απαρίθμηση μπορούμε να δηλώσουμε μια κλάση ως την απαρίθμηση κάποιων ατόμων. Η σύνταξη είναι η, $EnumeratedClass(A\ complete\ o1, o2, \dots, on)$. Τέλος η OWL μας προσφέρει τη

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

7.3 Περιγραφές και αξιώματα ιδιοτήτων

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

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

Όπως και στις ΠΛ, μας δίνεται η δυνατότητα να ορίσουμε ένα αξίωμα υπαγωγής ιδιοτήτων. Οι ορισμοί αυτοί δίνονται από την αφηρημένη σύνταξη, *SubPropertyOf(R S)*, όπου *R* και *S* είναι ιδιότητες αντικειμένων ή τύπων δεδομένων. Επιπλέον η OWL μας προσφέρει τη δυνατότητα να ορίσουμε μια ιδιότητα ως υποιδιότητα μιας λίστας ιδιοτήτων. Η δήλωση αυτή γίνεται γράφοντας, *ObjectProperty(R super(R1) ... super(Rn))*, όπου οι σχέσεις *R, R1 ... Rn* είναι ιδιότητες αντικειμένων ή τύπων δεδομένων.

Επιπρόσθετα μπορούμε να ορίσουμε μια ιδιότητα ως την αντίστροφη μιας άλλης γράφοντας, *ObjectProperty(R inverseOf(S))*, το οποίο δηλώνει ότι η ιδιότητα *R* είναι η αντίστροφη της *S*. Αυτό το αξίωμα σε ΠΛ αντιστοιχεί

σε δύο αξιώματα της μορφής $S \sqsubseteq R^{-}$ και $R^{-} \sqsubseteq S$ τα οποία σε σημασιολογία Περιγραφικών Λογικών ερμηνεύονται ως, $\forall a, b \in \Delta^I$ αν $S^I(a, b)$ τότε $R^I(b, a)$ και αντίστροφα. Αντίθετα με τις ΠΛ στην OWL μας δίνεται η δυνατότητα να ορίσουμε ισοδυναμία μεταξύ δύο ιδιοτήτων χρησιμοποιώντας τη σύνταξη *EquivalentProperties(R S)*, όπου *R* και *S* είναι όπως πριν. Το αξίωμα αυτό δε μας προσφέρει ουσιαστικά κάποια μεγαλύτερη εκφραστική δυνατότητα σε σχέση με τις ΠΛ αφού είναι ισοδύναμο με δύο αξιώματα υπαγωγής της μορφής, $S \sqsubseteq R$ και $R \sqsubseteq S$. Στη συνέχεια θα γράφουμε $S \equiv R$ ως συντόμευση των δύο αυτών υπαγωγών.

Επιπλέον, η OWL κληρονομεί τα αξιώματα ορισμού πεδίου ορισμού και συνόλου τιμών μιας ιδιότητας αντικειμένων ή τύπων δεδομένων. Η αφηρημένη σύνταξη για

αυτά τα αξιώματα είναι η $\text{ObjectProperty}(\text{R domain}(C))$ και $\text{ObjectProperty}(\text{R range}(C))$, αντίστοιχα, όπου R είναι μια ιδιότητα αντικειμένων και C είναι μια κλάση. Στην περίπτωση των ιδιοτήτων τύπων δεδομένων η λέξη κλειδί ObjectProperty αντικαθίσταται με τη λέξη κλειδί DatatypeProperty και η κλάση C στην περίπτωση του ορισμού του πεδίου τιμών αντικαθίσταται με έναν τύπο δεδομένων. Πιο συγκεκριμένα ο ορισμός γίνεται γράφοντας, $\text{DatatypeProperty}(\text{R range}(d))$.

Επίσης η OWL μας προσφέρει μια σειρά από αξιώματα ιδιοτήτων κάποια από τα οποία δεν εμφανίζονται άμεσα, παρά μόνο έμμεσα, στις Περιγραφικές Λογικές. Πιο συγκεκριμένα μπορούμε να δηλώσουμε ότι μια ιδιότητα αντικειμένων είναι *συμμετρική* (*symmetric*). Μια ιδιότητα R ονομάζεται συμμετρική αν $\forall a,b, R(a,b)$ συνεπάγεται ότι $R(b,a)$. Η δήλωση αυτή γίνεται γράφοντας σε αφηρημένη σύνταξη, $\text{ObjectProperty}(\text{R symmetric})$. Οι ΠΛ δεν παρέχουν άμεσα μια τέτοια δήλωση όμως μπορούμε να δηλώσουμε ένα αξίωμα της μορφής $R \equiv R^{-}$ το οποίο ουσιαστικά αναγκάζει την R να ερμηνευτεί ως συμμετρική σε όλες τις ερμηνείες που ικανοποιούν το αξίωμα. Μπορούμε επιπλέον να δηλώσουμε ότι μια ιδιότητα αντικειμένων R είναι μεταβατική, γράφοντας, $\text{ObjectProperty}(\text{R transitive})$.

Επιπρόσθετα, μπορούμε να δηλώσουμε μια ιδιότητα αντικειμένων ή τύπων δεδομένων ως *συναρτησιακή* (*functional*). Μια σχέση ονομάζεται συναρτησιακή αν, $\forall a,b,c R(a,b)$ και $R(a,c)$ συνεπάγεται ότι $b=c$. Για παράδειγμα η σχέση εχωΦυσικοΠατερα είναι συναρτησιακή καθώς είναι αδύνατο κάποιος να έχει παραπάνω από έναν φυσικό πατέρα. Η δήλωση αυτή γίνεται γράφοντας, $\text{ObjectProperty}(\text{R functional})$.

Αντίθετα από τα προηγούμενα, το αξίωμα αυτό μπορεί να γίνει και σε ιδιότητες τύπων δεδομένων. Σε αυτήν την περίπτωση το αξίωμα έχει τη μορφή $\text{DatatypeProperty}(\text{T functional})$, όπου T είναι μια ιδιότητα τύπων δεδομένων. Για παράδειγμα η ιδιότητα τύπων δεδομένων εχωΗλικια είναι συναρτησιακή. Εκ πρώτης όψης το αξίωμα αυτό φαίνεται να μας δίνει μεγαλύτερη εκφραστική δυνατότητα από αυτή των ΠΛ. Στην πραγματικότητα όμως ένα τέτοιο αξίωμα μπορεί σε μια ΠΛ να δηλωθεί από το αξίωμα υπαγωγής $T \sqsubseteq \leq 1R$, το οποίο διαισθητικά δηλώνει ότι κάθε αντικείμενο του Δ^I συμμετέχει στη σχέση R το πολύ με ένα άλλο αντικείμενο.

Τέλος μας δίνεται η δυνατότητα να ορίσουμε μια σχέση ως *αντίστροφη συναρτησιακή* (*inverse functional*). Διαισθητικά το αξίωμα αυτό δηλώνει ότι το πολύ ένα αντικείμενο μπορεί να συνδέεται με ένα άλλο αντικείμενο, δηλαδή αν ισχύουν τα $R(a,c)$ και $R(b,c)$ τότε πρέπει να ισχύει και $a=b$.

7.4 Άτομα

Όπως οι ΠΛ έτσι και η OWL μας παρέχει τη δυνατότητα δημιουργίας ισχυρισμών ανάμεσα σε ένα άτομο και μια κλάση, ανάμεσα σε ένα ζεύγος ατόμων και μια ιδιότητα αντικειμένων ή ανάμεσα σε ένα ζεύγος ατόμου και τιμής και μια ιδιότητα τύπων δεδομένων. Σε αφηρημένη σύνταξη οι δηλώσεις αυτές γίνονται γράφοντας $\text{Individual}(\text{o type}(C1) \dots \text{type}(Cn) \text{value}(R1, o1) \dots \text{value}(Rn, on))$, όπου τα $o, o1$ έως on είναι άτομα τα $C1 \dots Cn$ είναι κλάσεις και τα $R, R1$ έως και Rn είναι ιδιότητες. Στη περίπτωση όπου μια ιδιότητα R_i είναι ιδιότητα τύπων δεδομένων το άτομο o_i πρέπει

να αντικατασταθεί από μια τιμή, δηλαδή κάποιο plain ή typed literal. Η OWL όμως προσφέρει και επιπλέον δυνατότητες στη δημιουργία ισχυρισμών.

Πιο συγκεκριμένα μας δίνεται η δυνατότητα να ορίσουμε μια λίστα από άτομα ως *ταυτοδύναμα* γράφοντας `SameIndividual(o1 ... on)` ή ότι μια λίστα από άτομα είναι διαφορετικά μεταξύ τους γράφοντας `DifferentIndividuals(o1 ... on)`.

7.5 OWL Lite, OWL DL και OWL Full

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

Αντίθετα από τις OWL DL και OWL Full η OWL Lite προσφέρει ένα πιο περιορισμένο σύνολο περιγραφών και αξιωμάτων κλάσεων. Αυτό είχε ως σκοπό τη δημιουργία μιας λιγότερο εκφραστικής υπογλώσσας στην οποία οι αλγόριθμοι εξαγωγής συμπερασμάτων θα είναι πολύ γρήγοροι. Έτσι λοιπόν στην OWL Lite κάποια από τα δομικά στοιχεία της OWL που είδαμε σε προηγούμενες ενότητες απαγορεύεται να χρησιμοποιηθούν. Πιο συγκεκριμένα η OWL Lite απαγορεύει την χρήση των λέξεων κλειδιών, `owl:oneOf`, `owl:unionOf`, `owl:complementOf`, `owl:hasValue` και `owl:disjointWith`. Επιπρόσθετα, οι περιορισμοί πληθυκότητας, `minCardinality` και `maxCardinality` περιορίζονται σε συναρτησιακούς περιορισμούς.

Αυτό σημαίνει ότι οι βαθμοί πληθυκότητας μπορούν να είναι μόνο το 0 και το 1. Αν θέλαμε να μιλήσουμε με όρους Περιγραφικών Λογικών θα λέγαμε ότι η OWL Lite απαγορεύει τις ονοματικές έννοιες (*nominlas*), οι οποίες συμβολίζονται με το γράμμα O , ενώ αντί για τον πλήρη περιορισμό πληθυκότητας (N) επιτρέπει μόνο συναρτησιακούς περιορισμούς (F). Σε αυτό το σημείο ο αναγνώστης είναι απολύτως φυσικό να υποθέσει ότι η OWL Lite απαγορεύει επίσης την περιγραφή κλάσεων με τη χρήση του συμπληρώματος (\neg), το οποίο και συμβολίζεται με C , ή τη χρήση του κατασκευαστή της ένωσης (\sqcup), ο οποίος υπενθυμίζουμε ότι συμβολίζεται με το γράμμα (U). Στην πραγματικότητα όμως κάτι τέτοιο δεν είναι αληθές. Πιο συγκεκριμένα το σύνολο των στοιχείων (κατασκευαστών) που προσφέρει η OWL Lite είναι τέτοιο ώστε αυτοί οι κατασκευαστές να μπορούν να προσομοιωθούν. Για παράδειγμα έστω η έννοια A η οποία σε σύνταξη ΠΛ ορίζεται από το αξίωμα ισοδυναμίας $A \equiv C \sqcap D$. Αν ο χρήστης θέλει να ορίσει την άρνηση της έννοιας A μπορεί να εργαστεί ως εξής. Ορίζει έναν νέο ρόλο, έστω τον `forA`.

Στη συνέχεια ορίζει επιπλέον την έννοια A ως ισοδύναμη της έννοιας $\geq 1 \text{ for } A$. Τέλος, ορίζει την έννοια `negA` ως $\text{negA} \equiv \leq 0 \text{ for } A$. Η έννοια `negA` είναι το συμπλήρωμα της έννοιας, A . Αυτό συμβαίνει γιατί η έννοια $\geq 1 \text{ for } A$ είναι το συμπλήρωμα της έννοιας

$\leq \text{for } A$. Έτσι λοιπόν χρησιμοποιώντας κάθε φορά και έναν νέο ρόλο μπορούμε να ορίσουμε την άρνηση μιας έννοιας. Παρομοίως μπορούμε να εργαστούμε και για τον ορισμό της ένωσης δύο εννοιών.

Συνοψίζοντας, η OWL Lite αντιστοιχεί στην ΠΛ *SHIF(D)* η οποία είναι μια αρκετά εκφραστική γλώσσα και άρα ο αρχικός σκοπός για τη δημιουργία μιας μη εκφραστικής γλώσσας απέτυχε.

7.6 RDF/XML σύνταξη

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

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

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

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

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

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


```

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

```

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

```

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

```

Εφόσον θεωρητικά μια OWL οντολογία πρέπει να επεκτείνει μια RDF οντολογία, ένα OWL αρχείο εκκινεί με τις ίδιες ετικέτες με τις οποίες εκκινεί και ένα RDF αρχείο. Πιο συγκεκριμένα ένα OWL αρχείο ξεκινά με την ετικέτα <rdf:RDF>. Στη συνέχεια όμως προκειμένου να δηλώσουμε ότι αυτό που ακολουθεί είναι μια OWL οντολογία χρησιμοποιούμε την ετικέτα owl:Ontology. Η ετικέτα αυτή περιέχει το στοιχείο rdf:about το οποίο αποδίδει ένα όνομα για την οντολογία. Εκτός από τα στοιχεία τα οποία είδαμε και έχουν σχέση με περιγραφή και αξιώματα κλάσεων και ιδιοτήτων η OWL προσφέρει και μια σειρά από στοιχεία τα οποία χρησιμοποιούνται για την περιγραφή μεταπληροφορίας για την οντολογία. Πιο συγκεκριμένα το στοιχείο owl:priorVersion δηλώνει ένα URL στο οποίο βρίσκεται μια OWL οντολογία η οποία είναι προηγούμενη έκδοση της οντολογίας που δηλώνεται από το τρέχον αρχείο OWL.

Επίσης το στοιχείο owl:versionInfo χρησιμοποιείται για να δηλώσει πληροφορίες για το τρέχον αρχείο όπως για παράδειγμα ημερομηνία και ώρα δημιουργίας και άλλα. Τέλος ένα στοιχείο της OWL το οποίο χρησιμοποιείται αρκετά αλλά επίσης έχει αποτελέσει αμφιλεγόμενο σημείο για τους ερευνητές όσον αφορά τη σημασιολογία του είναι το στοιχείο owl:imports. Με το στοιχείο αυτό μπορούμε να εισάγουμε τους ορισμούς, κλάσεις, αξιώματα και ιδιότητες μιας άλλης οντολογίας στο τρέχον αρχείο. [16]

7.7 Βιβλιογραφία- αναφορά πηγής

[16]: πηγή από διαλέξεις

<http://www.image.ece.ntua.gr/~gstoil/RDF-OWL.pdf>

Κεφάλαιο 8

8.1 Ενδεικτικές εφαρμογές των τεχνολογιών του σημασιολογικού ιστού

Τη διαχείριση και την πρόσβαση σε πληροφορίες στην περίπτωση του γνωστού εκδοτικού οίκου Elsevier.

Την ενοποίηση δεδομένων στην περίπτωση μεγάλων εταιριών με πολλές βάσεις δεδομένων.

Την αξιοποίηση των τεχνολογιών του Σημασιολογικού Ιστού σε ένα περιβάλλον τηλεκπαίδευσης.

Διαχείριση γνώσης.

Υπηρεσίες τηλεκπαίδευσης.

Πλατφόρμα διαλειτουργικότητας μεταξύ συσκευών. [17]

8.2 Οίκος Elsevier

Ο οίκος Elsevier είναι ένας σημαντικός επιστημονικός εκδοτικός οίκος. Στη Διαδικτυακή παρουσία του οίκου Elsevier τα προϊόντα του, δηλαδή τα επιστημονικά άρθρα που δημοσιεύει, είναι ακόμη οργανωμένα με βάση τον παραδοσιακό τρόπο κατά τον οποίο ένα άρθρο είναι διαθέσιμο στο χρήστη αποκλειστικά με τη μορφή με την οποία εμφανίζεται σε ένα περιοδικό (δηλαδή ανά τεύχος έκδοσης) και όχι με βάση το συγκεκριμένο ερευνητικό πεδίο στο οποίο αναφέρεται. [17]

8.2.1 Το πρόβλημα του Οίκου Elsevier

Οι πελάτες του οίκου Elsevier μπορούν να αποκτήσουν συνδρομές στο Διαδικτυακό περιεχόμενο, το οποίο όμως είναι οργανωμένο ανά εκδόσεις περιοδικών. Αυτά τα "παραδοσιακά" περιοδικά μπορούν να περιγραφούν και ως κάθετα προϊόντα. Τα επιστημονικά περιοδικά διαχωρίζονται σε έναν αριθμό ξεχωριστών στηλών - πεδίων (π.χ. βιολογία, χημεία, ιατρική, πληροφορική) και κάθε προϊόν καλύπτει μία τέτοια στήλη (ή το πιθανότερο τμήμα μίας τέτοιας στήλης).

Ωστόσο, η παραδοσιακή κατηγοριοποίηση σε ξεχωριστά επιστημονικά πεδία που καλύπτονται από διαφορετικά περιοδικά δεν είναι πλέον ικανοποιητική εξαιτίας της ραγδαίας ανάπτυξης στις διάφορες επιστήμες. Οι πελάτες της Elsevier ενδιαφέρονται να καλύψουν συγκεκριμένες θεματικές περιοχές που εκτείνονται σε περισσότερα από ένα παραδοσιακά επιστημονικά πεδία. Η ζήτηση σχετίζεται πλέον με οριζόντια προϊόντα. Είναι δύσκολο για μεγάλους εκδοτικούς οίκους να προσφέρουν τέτοια οριζόντια προϊόντα. Οι πληροφορίες που δημοσιεύονται είναι εγκλωβισμένες μέσα στα διαφορετικά περιοδικά.

Κάθε περιοδικό έχει το δικό του σύστημα ευρετηρίου (indexing), το οποίο είναι οργανωμένο σύμφωνα με διαφορετικά φυσικά, συντακτικά και σημασιολογικά πρότυπα. Όμως οι περιορισμοί της φυσικής και συντακτικής ετερογένειας μπορούν να αρθούν. Ο οίκος Elsevier έχει ήδη μεταφράσει μεγάλο μέρος του περιεχομένου του σε μορφή XML, η οποία επιτρέπει την παράλληλη υποβολή ερωτημάτων σε περισσότερα από ένα περιοδικά.

Ωστόσο, το σημασιολογικό πρόβλημα παραμένει άλυτο σε μεγάλο βαθμό. Είναι εφικτή η αναζήτηση σε πολλά περιοδικά για άρθρα που περιέχουν τις ίδιες λέξεις-κλειδιά, αλλά αυτός ο τρόπος δεν επιστρέφει ικανοποιητικά αποτελέσματα σε περίπτωση που χρησιμοποιούνται συνώνυμοι όροι ανάμεσα στα διάφορα επιστημονικά πεδία. Αυτό που χρειάζεται είναι ένας τρόπος αναζήτησης στα διάφορα περιοδικά με χρήση ενός συναφούς συνόλου εννοιών με βάση το οποίο θα δημιουργείται το ευρετήριο όλων αυτών των περιοδικών. [17]

8.3 Συμβολή του σημασιολογικού ιστού (χρήση οντολογιών και θησαυρών)

Οι οντολογίες και οι θησαυροί είναι τρόποι κωδικοποίησης και οργάνωσης της γνώσης ενός συγκεκριμένου γνωστικού πεδίου. Οι θησαυροί είναι ένα είδος οντολογίας και αποτελούν τη νοηματική οργάνωση οριζοντίων και κάθετων ιεραρχικών συσχετισμών των εννοιών. Οι οντολογίες, ως βασικό μέσο αναπαράστασης γνώσης στο σημασιολογικό ιστό, επιτρέπουν το διαμοιρασμό της γνώσης και των πληροφοριών και

παρέχουν τους μηχανισμούς για την αποτελεσματική αξιοποίηση αυτών, τόσο από ανθρώπους όσο και από πράκτορες λογισμικού. Τα τελευταία χρόνια χρησιμοποιούνται σε εφαρμογές όπου απαιτείται συνδυασμός πληροφοριών από διάσπαρτες βάσεις δεδομένων.

Οι οντολογίες και οι θησαυροί όρων έχουν αποδειχθεί ιδιαίτερος χρήσιμα εργαλεία για την αποτελεσματική πρόσβαση σε πληροφορίες. Μας βοηθούν να ξεπεράσουμε τα προβλήματα της αναζήτησης ελεύθερου κειμένου (free-text search), συσχετίζοντας και ομαδοποιώντας συναφείς όρους σε ένα συγκεκριμένο πεδίο, αλλά και παρέχοντας ένα ελεγχόμενο λεξιλόγιο για το ευρετήριο των πληροφοριών.

Παραδείγματα:

Οι θησαυροί MeSH και EMTREE αναφέρονται στον τομέα των ιατρικών πληροφοριών. Οι θησαυροί αυτοί χρησιμοποιούνται ήδη από βάσεις δεδομένων όπως η Science Direct.

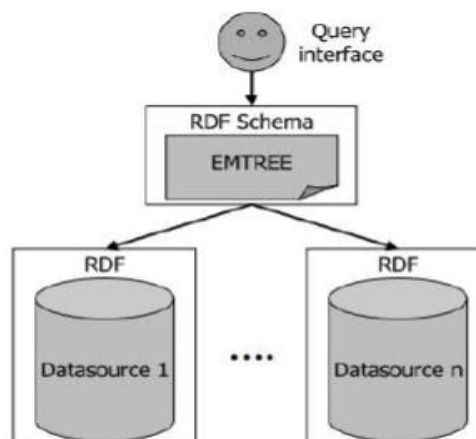
Η γλώσσα RDF χρησιμοποιείται ως μια μορφή διαλειτουργικότητας ανάμεσα στις ετερογενείς πηγές δεδομένων.

Παράδειγμα:

Ο οίκος Elsevier πειραματίζεται με τη δυνατότητα παροχής πρόσβασης σε πολλές πηγές πληροφοριών στην περιοχή των βιολογικών επιστημών μέσω μίας διασύνδεσης, χρησιμοποιώντας το θησαυρό EMTREE ως τη μοναδική υποκείμενη οντολογία με βάση την οποία θα ευρετηριάζονται όλες οι κάθετες πηγές πληροφοριών.

Η γλώσσα RDF χρησιμοποιείται ως μια μορφή διαλειτουργικότητας ανάμεσα στις ετερογενείς πηγές δεδομένων.

Η ίδια η οντολογία EMTREE αναπαρίσταται στη γλώσσα RDF. Κάθε μία από τις διαφορετικές πηγές δεδομένων αντιστοιχίζεται σε αυτή τη συνεκτική οντολογία, η οποία χρησιμοποιείται στη συνέχεια ως το μοναδικό σημείο εισόδου για όλες αυτές τις πηγές δεδομένων.



Κατά τα πειράματα του, ο οίκος Elsevier χρηματοδότησε το πρόγραμμα DOPE (Drug Ontology Project for Elsevier). Ο θησαυρός EMTREE χρησιμοποιήθηκε για τη δημιουργία ευρετηρίου εκατομμυρίων ιατρικών περιλήψεων (abstracts) και άρθρων. Κατασκευάστηκε ένα περιβάλλον διεπαφής για την αναζήτηση ιατρικών όρων. [17]

8.4 Οντολογία EMTREE

Η οντολογία EMTREE χρησιμοποιείται για:

Να αποσαφηνίσει το αρχικό ερώτημα του χρήστη που διατυπώνεται με κείμενο ελεύθερης μορφής. Π.χ. ο όρος "AIDS" αναφέρεται στην ασθένεια AIDS ή είναι ο πληθυντικός της λέξης "aid".

Να κατηγοριοποιήσει τα αποτελέσματα της αναζήτησης (αριστερή στήλη της προηγούμενης εικόνας): κάθε καταχώριση είναι ουσιαστικά μια κατηγορία της οντολογίας EMTREE στη σωστή ιεραρχική θέση της. Οι ίδιες αυτές κατηγορίες χρησιμοποιούνται για την παραγωγή μιας οπτικής ομαδοποίησης των αποτελεσμάτων αναζήτησης.

Να περιορίσει ή να διευρύνει το ερώτημα αναζήτησης με ουσιαστικό τρόπο σε περίπτωση που η αναζήτηση παράγει υπερβολικά πολλά ή λίγα αποτελέσματα. [17]

8.5 Συμβολή της τεχνολογίας του σημασιολογικού ιστού (χρήση οντολογιών)

Η χρήση οντολογιών (ως σημασιολογικά μοντέλα δεδομένων) μπορεί να οργανώσει ορθολογικά τις ανόμοιες πηγές δεδομένων σε ένα ενιαίο σώμα πληροφοριών.

Αν δημιουργήσουμε οντολογίες για πηγές δεδομένων και περιεχομένου και προσθέσουμε γενικές πληροφορίες πεδίου, η ενοποίηση των ανόμοιων πηγών στην εταιρεία μπορεί να επιτευχθεί χωρίς να διαταραχθούν οι υπάρχουσες εφαρμογές.

Η οντολογία αντιστοιχίζεται με τις πηγές δεδομένων (πεδία, έγγραφα, αρχεία, έγγραφα) παρέχοντας στις εφαρμογές άμεση πρόσβαση στα δεδομένα μέσω της οντολογίας. [17]

8.6 Τηλεκπαίδευση

Η εκπαίδευση χαρακτηρίζεται από τις ακόλουθες ιδιότητες:

Καθοδήγηση από τον εκπαιδευτικό: ο διδάσκων επιλέγει το περιεχόμενο του μαθήματος και καθορίζει το πρόγραμμα, τη σειρά των μαθημάτων και τον ρυθμό της εκμάθησης.

Γραμμική πρόσβαση στη γνώση: οι μαθητές δεν μπορούν να αλλάξουν τη σειρά των μαθημάτων.

Η πρόσβαση στη γνώση δε μπορεί να γίνει εξατομικευμένα αλλά είναι σχεδιασμένη για μαζική συμμετοχή.

Η ανάπτυξη του Παγκόσμιου Ιστού άνοιξε το δρόμο για την υλοποίηση νέων εκπαιδευτικών διαδικασιών, όπως η εξ αποστάσεως εκπαίδευση.

Στην ανώτατη εκπαίδευση, τα πανεπιστημιακά ιδρύματα εστιάζουν στο να παρέχουν μεγαλύτερη ευελιξία στους φοιτητές χρησιμοποιώντας πλατφόρμες e-learning δημιουργώντας ένα εικονικό περιβάλλον τάξης.

Οι φοιτητές μπορούν πλέον να επιλέγουν το υλικό των μαθημάτων, καθώς και το επίπεδο και το ρυθμό εκμάθησης.

Σε σύγκριση με την παραδοσιακή εκπαίδευση, η τηλεεκπαίδευση δεν είναι καθοδηγούμενη από τον εκπαιδευτικό. Οι εκπαιδευόμενοι μπορούν να έχουν πρόσβαση στην ύλη, χωρίς κάποια προκαθορισμένη σειρά και μπορούν να συνθέσουν μεμονωμένα μαθήματα επιλέγοντας την εκπαιδευτική ύλη.

Μια τέτοια προσέγγιση θα μπορούσε να λειτουργήσει μόνο αν η διδακτέα ύλη είναι εφοδιασμένη με πρόσθετες πληροφορίες για την υποστήριξη της αποτελεσματικής δεικτοδότησης και ανάκτησης. Μια λύση είναι η χρήση μετα-δεδομένων (metadata). Στην κοινότητα της τηλεεκπαίδευσης έχουν εμφανιστεί διάφορα πρότυπα όπως το LOM της IEEE. Συσχετίζει τη διδακτέα ύλη με πληροφορίες όπως εκπαιδευτικές και παιδαγωγικές ιδιότητες, δικαιώματα πρόσβασης και όρους χρήσης, καθώς και τη σχέση της με άλλες εκπαιδευτικές πηγές.

Ωστόσο, το βασικό μειονέκτημα αυτών των προτύπων που βασίζονται αποκλειστικά σε μεταδεδομένα είναι η απουσία της σημασιολογίας. Ο συνδυασμός της ύλης που προέρχεται από διαφορετικούς συγγραφείς, όπου ο κάθε συγγραφέας μπορεί να χρησιμοποιεί διαφορετική ορολογία, είναι δύσκολος.

Ομοίως, η ανάκτηση πληροφοριών μπορεί να μη γίνεται με το βέλτιστο τρόπο. Η ανάκτηση και η οργάνωση του εκπαιδευτικού υλικού πρέπει να γίνεται χειρωνακτικά από τους εκπαιδευτές, ενώ θα μπορούσε να γίνεται αυτόματα με τη χρήση ενός αυτοματοποιημένου πράκτορα. [17]

8.7 Η συμβολή του σημασιολογικού ιστού

Χρήση οντολογιών σε συνδυασμό με τα επεξεργάσιμα από υπολογιστές μετα-δεδομένα δημιουργούν μια πολλά υποσχόμενη προσέγγιση για τις απαιτήσεις της σύγχρονης τηλεεκπαίδευσης.

Ο Σημασιολογικός Ιστός μπορεί να υποστηρίξει τόσο τα σημασιολογικά ερωτήματα όσο και την εννοιολογική πλοήγηση στο εκπαιδευτικό υλικό.

8.8 Καθοδήγηση από τον εκπαιδευόμενο

Η διδακτέα ύλη, προερχόμενη πιθανώς από διαφορετικούς συγγραφείς, μπορεί να συνδεθεί με κοινά αποδεκτές οντολογίες.

Είναι δυνατός ο σχεδιασμός εξατομικευμένων μαθημάτων μέσω σημασιολογικών ερωτημάτων.

Το εκπαιδευτικό υλικό μπορεί να ανακτηθεί με βάση τις ανάγκες του κάθε εκπαιδευόμενου. [17]

8.9 Ευέλικτη (μη γραμμική) πρόσβαση στη γνώση

Το εκπαιδευτικό υλικό μπορεί να προσπελαστεί με οποιαδήποτε σειρά ανάλογα με τις ανάγκες και το επίπεδο του κάθε εκπαιδευόμενου.

8.10 Οντολογίες για τηλεκπαίδευση

Ουσιαστικά είναι ένας μηχανισμός εδραίωσης μια κοινής αντίληψης για τους εκπαιδευόμενους και τους εκπαιδευτές.

Στα πλαίσια της τηλεκπαίδευσης, υπάρχουν τρία είδη οντολογιών:

- Οντολογίες περιεχομένου.
- Παιδαγωγικές οντολογίες.
- Οντολογίες δομής. [17]

8.10.1 Οντολογίες περιεχομένου

Περιγράφουν τις βασικές έννοιες ενός πεδίου. Π.χ. του μαθήματος "Ιστορία". Περιέχουν τις σχέσεις μεταξύ των εννοιών και τις βασικές ιδιότητες τους. Η οντολογία πρέπει να περιλαμβάνει τη σχέση "is part of". Επίσης, μπορεί να περιέχει σχέσεις για να αποτυπώσει συνώνυμα, συντομεύσεις-συντμήσεις, κ.ο.κ. [17]

8.10.2 Παιδαγωγικές οντολογίες

Περιγράφουν-διευθετούν τα διάφορα παιδαγωγικά θέματα.

8.10.2 Οντολογίες Δομής

Περιγράφουν τη λογική δομή του εκπαιδευτικού υλικού. Περιλαμβάνουν ιεραρχικές σχέσεις, αλλά και σχέσεις πλοήγησης. Π.χ. Σχέσεις όπως "προηγούμενο", "επόμενο", "is part of", "has part", "requires", "is based on". Είναι δυνατόν να περιλαμβάνουν ορισμούς για τη σχέση μεταξύ συγκεκριμένων σχέσεων. Οι σχέσεις "has part" και "is part of" είναι αντίστροφες.[17]

8.11 Βιβλιογραφία- αναφορά πηγής

[17]: πηγή από διαλέξεις

- Το υλικό των διαφανειών προήλθε από τη σημαντική συνεισφορά της Β. Γκαντούνα, MSc

Κεφάλαιο 9

9.1 Συμπεράσματα- Επίλογος

Ήδη κυκλοφορούν κάποια συστήματα υπό τη μορφή πρωτοτύπων, που λειτουργούν κατακευκμένοι χώροι αποθήκευσης (repository) για τον Παγκόσμιο Ιστό (WWW). Τα προϊόντα αυτά, υποστηρίζουν την ομοσπονδιακή (federated) ανάπτυξη και αποθήκευση των μοντέλων αναφοράς και των οντολογιών (ontologies), καθώς και την αντιστοίχιση και προβολή ανάμεσα σε διαφορετικά συστήματα μοντέλα/σχήματα, ενώ υποστηρίζουν τα πρότυπα του W3C (όπως XML και RDF).

Ο σημασιολογικός Ιστός (Semantic Web) αποτελεί μια καινοτομία "εν τη γενέσει" της, η οποία υπόσχεται την ενσωμάτωση όψεων (views) βάσεων δεδομένων, "συναλλαγών" μεταξύ βάσεων δεδομένων (database transactions), λογικών αναπαραστάσεων, συνδέσμων Ιστού (Web links) και αντικειμενοστρεφών αναπαραστάσεων, σε μια σημασιολογική βάση.



Τα παραπάνω χαρακτηριστικά οδηγούν σε ένα νέο τρόπο στην επεξεργασία ερωτήσεων (query processing). Η τεχνολογία που παρουσιάζεται θα είναι υποχρεωτική, κατά τους δημιουργούς της, στις περισσότερες κατακευκμένες εφαρμογές-πελάτη, ενώ δεν υπάρχει περιορισμός ως προς τα πεδία (αγορές) εφαρμογής της.

Το όραμα της δημιουργίας του Semantic Web στηρίζεται στην επέκταση των υπάρχοντων πλαισίων περιγραφής μετα- δεδομένων και ειδικότερα στην ύπαρξη σημασιολογικού περιεχομένου που είναι δυνατόν να υπόκειται σε αυτόματη επεξεργασία από τον υπολογιστή χωρίς την επέμβαση του ανθρώπινου παράγοντα. Η ερευνητική προσπάθεια έγκειται στην δημιουργία γενικών πλαισίων όπως το UN/SPSC και γλωσσών όπως η OWL τα οποία θα υποστηρίζουν οντολογικά όσο το δυνατόν περισσότερα πεδία εφαρμογών γίνεται. Πρόσφατα, η ομάδα εργασίας οντολογίας του Παγκοσμίου Ιστού εξέδωσε ένα υπό διαμόρφωση προσχέδιο τελικού σχολιασμού των παραδειγμάτων της OWL το οποίο συνοδεύει τον ορισμό της γλώσσας. [18]

9.2 Βιβλιογραφία- αναφορά πηγής

[18]: πηγή από διαδίκτυο

<http://dsmc.eap.gr/semweb.php>

Βιβλιογραφία

Από συγγράμματα και διπλωματικές εργασίες ή διαλέξεις

- Η γλώσσα σήμανσης XML του επίκουρου καθηγητή Δημήτριου Σαμψών, Πανεπιστημιακές σημειώσεις του Πανεπιστημίου Πειραιώς, Τμήμα Διδακτικής της Τεχνολογίας και Ψηφιακών Συστημάτων
- Αξιολόγηση σημασιολογικού ιστού και κοινωνικές τεχνολογίες για ψηφιακές βιβλιοθήκες της Έμελυ Χαντζηχρήστου Ιόνιο Πανεπιστήμιο Τμήμα Αρχαιονομίας – Βιβλιοθηκονομίας
Μεταπτυχιακό Πρόγραμμα Σπουδών: Επιστήμη της πληροφορίας – υπηρεσίες πληροφόρησης σε ψηφιακό περιβάλλον
- Το υλικό των διαφανειών προήλθε από τη σημαντική συνεισφορά της Β. Γκαντούνα, MSc.

Από διαδίκτυο - ιστότοπους

- https://el.wikipedia.org/wiki/%CE%A3%CE%B7%CE%BC%CE%B1%CF%83%CE%B9%CE%BF%CE%BB%CE%BF%CE%B3%CE%B9%CE%BA%CF%8C%CF%82_%CE%99%CF%83%CF%84%CF%8C%CF%82
- <http://dsmc.eap.gr/semweb.php>
- http://glotta.ntua.gr/NLP/NLP-PostGrad/Assignments/Giannoulis_Semantic_Search.pdf
- <http://www.image.ece.ntua.gr/~gstoil/RDF-OWL.pdf>
- <http://lpis.csd.auth.gr/mtpx/sw/>
- https://el.wikipedia.org/wiki/%CE%A0%CE%B1%CE%B3%CE%BA%CF%8C%CF%83%CE%BC%CE%B9%CE%BF%CF%82_%CE%99%CF%83%CF%84%CF%8C%CF%82
- http://ebooks.edu.gr/modules/ebook/show.php/DSB100/534/3531,14509/index_4_2.html

- http://hermes.di.uoa.gr/exe_activities/diadiktio/21_____html
- http://www.w3schools.com/html/html_intro.asp
- <https://en.wikipedia.org/wiki/HTML>
- http://www.w3schools.com/xml/xml_dtd_intro.asp
- https://en.wikipedia.org/wiki/Document_type_definition
- http://www.w3schools.com/xml/xml_elements.asp
- http://www.w3schools.com/xsl/xpath_intro.asp
- <http://www.image.ece.ntua.gr/~gstoil/RDF-OWL.pdf>