



Πολυτεχνείο Κρήτης

**Τμήμα Ηλεκτρονικών Μηχανικών
& Μηχανικών Υπολογιστών**

**ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΓΡΑΦΙΚΟΥ
ΣΥΣΤΗΜΑΤΟΣ ΔΙΑΧΕΙΡΙΣΗΣ OWL ΟΝΤΟΛΟΓΙΩΝ
ΚΑΙ ΧΡΗΣΗ ΤΟΥ ΩΣ ΕΡΓΑΛΕΙΟ ΣΗΜΑΣΙΟΛΟΓΙΚΗΣ
ΠΕΡΙΓΡΑΦΗΣ ΠΕΡΙΕΧΟΜΕΝΟΥ ΠΟΛΥΜΕΣΩΝ
ΒΑΣΙΣΜΕΝΗ ΣΤΟ ΠΡΟΤΥΠΟ MPEG-7**

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΑΤΡΙΒΗ ΠΟΥ ΕΚΠΟΝΗΘΗΚΕ ΣΤΟ
ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ ΓΙΑ ΤΗΝ ΕΚΠΛΗΡΩΣΗ ΤΩΝ
ΥΠΟΧΡΕΩΣΕΩΝ ΓΙΑ ΤΗ ΛΗΨΗ ΤΟΥ ΜΕΤΑΠΤΥΧΙΑΚΟΥ
ΔΙΠΛΩΜΑΤΟΣ ΕΙΔΙΚΕΥΣΗΣ ΤΟΥ ΤΜΗΜΑΤΟΣ ΗΛΕΚΤΡΟΝΙΚΩΝ
ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΟΛΥΔΩΡΟΣ ΠΑΝΑΓΙΩΤΗΣ
ΙΟΥΛΙΟΣ 2007

ΠΕΡΙΛΗΨΗ

Στην παρούσα μεταπτυχιακή διατριβή παρουσιάζεται το GraphOnto το οποίο απαρτίζεται από ένα πλήρες σύστημα μαζί με ένα JAVA API που επιτρέπουν τον ορισμό και την επεξεργασία OWL οντολογιών και τη χρησιμοποίησή τους σε συστήματα περιγραφής και διαχείρισης πολυμέσων. Οι οντολογίες διακρίνονται σε ανώτερες (Upper), που βασίζονται σε διεθνή πρότυπα (standards), και σε περιοχής (domain-specific), που περιγράφουν σημασιολογικά συγκεκριμένα πεδία της πραγματικότητας. Το σύνθετο περιβάλλον εργασίας του συστήματος αποτελείται από μια ανώτερη OWL οντολογία που αναπαριστά το πρότυπο σημασιολογικής περιγραφής πολυμέσων MPEG-7 MDS και από συγκεκριμένου πεδίου OWL οντολογίες που επεκτείνουν την ανώτερη οντολογία με γνώση (knowledge) που αφορά συγκεκριμένα πεδία. Οι οντολογίες που εισάγονται στο σύστημα διαβάζονται και στη συνέχεια παράγονται με αυτόματο τρόπο διεπαφές (interfaces) για τη γραφική αναπαράσταση και διαχείριση των οντολογιών. Οι οντολογίες χρησιμοποιούνται για να καθοδηγήσουν τη σημασιολογική περιγραφή του περιεχομένου των πολυμέσων με έναν καθορισμένο τρόπο. Υποστηρίζονται οι αντιστοιχίσεις (mappings) μεταξύ OWL οντολογιών με τη χρήση των κατάλληλων OWL δομών και καθίσταται δυνατός ο ορισμός εξατομικευμένων οντολογιών. Υποστηρίζεται η εξατομίκευση του γραφικού περιβάλλοντος σε επίπεδο εφαρμογής, σε επίπεδο έργου και σε επίπεδο χρήστη. Επιπλέον, το σύστημα αλληλεπικοινωνεί με μια MPEG-7 αποθήκη μεταδεδομένων (Repository) για την αποθήκευση σημασιολογικών περιγραφών βασισμένων στο MPEG-7 MDS και για την επαναχρησιμοποίηση σημασιολογικών οντοτήτων κατά τη διάρκεια ορισμού νέων περιγραφών του οπτικό-ακουστικού υλικού. Το GraphOnto αξιολογήθηκε σε σχέση με την πιο διαδεδομένη εφαρμογή διαχείρισης OWL οντολογιών, το Protégé, και υπερέρχει τόσο σε λειτουργικότητα όσο και σε απόδοση.

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω τον καθηγητή κ. Σταύρο Χριστοδουλάκη για την επίβλεψη του στην ολοκλήρωση αυτής της διατριβής και για τις εμπειρίες που μου πρόσφερε στα πλαίσια της εργασίας μου στο Εργαστήριο Διανεμημένων Πληροφοριακών Συστημάτων και Εφαρμογών του Πολυτεχνείου Κρήτης.

Θα ήθελα να ευχαριστήσω προκαταβολικά τους αναγνώστες αυτής της διατριβής, τον αναπληρωτή καθηγητή κ. Ευριπίδη Πετράκη και τον επίκουρο καθηγητή κ. Μιχαήλ Λαγουδάκη για το χρόνο που θα διαθέσουν για την ανάγνωση της διατριβής και τις τυχόν παρατηρήσεις τους πάνω σε αυτήν.

Θα ήθελα ακόμα να ευχαριστήσω θερμά τη Χρύσα Τσιναράκη για την πολύτιμη βοήθεια και καθοδήγηση καθ'όλη τη διάρκεια εκπόνησης της παρούσας διατριβής, καθώς και όσους συνεργάτες από το Εργαστήριο Διανεμημένων Πληροφοριακών Συστημάτων και Εφαρμογών του Πολυτεχνείου Κρήτης βοήθησαν στην περάτωση αυτής της διατριβής.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου, την Αθηνά, τους γονείς της και όσους ακόμα μου συμπαράσταθηκαν σε αυτά τα τέσσερα χρόνια που διήρκησαν οι μεταπτυχιακές μου σπουδές.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1. ΕΙΣΑΓΩΓΗ.....	11
1.1. ΓΕΝΙΚΑ.....	11
1.2. ΣΥΝΕΙΣΦΟΡΑ.....	11
1.3. ΔΟΜΗ ΤΗΣ ΕΡΓΑΣΙΑΣ	12
2. ΣΧΕΤΙΚΕΣ ΤΕΧΝΟΛΟΓΙΕΣ	14
2.1. ΕΙΣΑΓΩΓΗ	14
2.2. OWL (WEB ONTOLOGY LANGUAGE).....	14
2.2.1. Εισαγωγή.....	14
2.2.2. Εισαγωγή στις γλώσσες του Semantic Web.....	15
2.2.3. RDF / RDFS.....	15
2.2.4. DAML+OIL.....	18
2.2.5. Εισαγωγή στην OWL και τα είδη της OWL.....	18
2.2.6. Η δομή μιας OWL οντολογίας.....	19
2.2.7. OWL κλάσεις (Classes).....	21
2.2.8. OWL Άτομα (Individuals).....	22
2.2.9. OWL Ιδιότητες (Properties)	22
2.2.10. Γνωρίσματα των OWL ιδιοτήτων.....	25
2.2.11. Περιορισμοί Ιδιοτήτων (Property Restrictions).....	26
2.2.12. Αντιστοιχήσεις OWL στοιχείων (OWL Ontology Mapping)	29
2.2.13. Σύνθετες κλάσεις (Complex Classes).....	32
2.3. MPEG-7	35
2.3.1. Εισαγωγή.....	35
2.3.2. Τα βασικά στοιχεία του MPEG-7.....	35
2.3.3. Τα τμήματα του MPEG-7.....	37
2.3.4. MPEG-7 MDS.....	38
2.4. ΑΝΑΚΕΦΑΛΑΙΩΣΗ	43
3. ΣΧΕΤΙΚΕΣ ΕΡΓΑΣΙΕΣ ΚΑΙ ΑΝΑΓΚΑΙΟΤΗΤΑ ΥΛΟΠΟΙΗΣΗΣ ΤΟΥ GRAPHONTO ..	45
3.1. ΕΙΣΑΓΩΓΗ	45
3.2. ΥΛΟΠΟΙΗΣΕΙΣ ΣΤΟΝ ΤΟΜΕΑ ΤΩΝ OWL ΟΝΤΟΛΟΓΙΩΝ	45
3.2.1. Jena.....	45
3.2.2. Protégé	46
3.2.3. Altova SemanticWorks	49
3.2.4. Smore	49
3.2.5. Swoop.....	51
3.3. ΑΞΙΟΛΟΓΗΣΗ	52
3.4. ΑΝΑΓΚΑΙΟΤΗΤΑ ΚΑΙ ΚΙΝΗΤΡΟ ΥΛΟΠΟΙΗΣΗΣ ΤΟΥ GRAPHONTO.....	57
3.5. ΑΝΑΚΕΦΑΛΑΙΩΣΗ	58
4. ΤΟ ΠΛΑΙΣΙΟ ΔΙΑΧΕΙΡΙΣΗΣ ΜΕΤΑΔΕΔΟΜΕΝΩΝ ΓΙΑ ΟΠΤΙΚΟΑΚΟΥΣΤΙΚΟ ΥΛΙΚΟ (DS-MIRF FRAMEWORK).....	60
4.1. ΕΙΣΑΓΩΓΗ	60
4.2. ΠΡΟΣΕΓΓΙΣΗ ΥΠΟΣΤΗΡΙΞΗΣ ΣΗΜΑΣΙΟΛΟΓΙΚΗΣ ΔΙΑΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑΣ	60
4.3. ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ ΠΛΑΙΣΙΟΥ DS-MIRF.....	61
4.4. ΣΥΣΤΑΤΙΚΑ ΤΟΥ ΠΛΑΙΣΙΟΥ DS-MIRF	63
4.4.1. Εισαγωγή.....	63
4.4.2. Οντολογική Υποδομή του Πλαισίου DS-MIRF.....	63
4.4.3. Το Συστατικό Λογισμικού GraphOnto.....	64
4.4.4. Η DS-MIRF Αποθήκη MPEG-7/21 Μεταδεδομένων.....	65
4.5. ΑΝΑΚΕΦΑΛΑΙΩΣΗ	66
5. ΑΡΧΙΤΕΚΤΟΝΙΚΗ.....	67
5.1. ΕΙΣΑΓΩΓΗ	67

5.2.	GRAPHONTO USER INTERFACE.....	68
5.2.1.	Εισαγωγή.....	68
5.2.2.	Βασικές Αρχές Σχεδιασμού του GraphOnto User Interface	68
5.2.3.	Περιγραφή του GraphOnto User Interface	70
5.3.	USER INTERFACE MANAGER	72
5.4.	ΔΙΑΧΕΙΡΙΣΤΗΣ ΜΟΝΤΕΛΩΝ ΟΝΤΟΛΟΓΙΩΝ	73
5.5.	WONDERWEB OWL API.....	73
5.6.	GRAPHONTO API.....	74
5.7.	ΜΕΤΑΤΡΟΠΕΑΣ OWL ΣΕ MPEG-7	74
5.8.	MPEG-7 ΑΠΟΘΗΚΗ ΜΕΤΑΔΕΔΟΜΕΝΩΝ	75
5.9.	ΕΡΓΑΛΕΙΑ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ ΚΑΤΑ ΤΗΝ ΥΛΟΠΟΙΗΣΗ	76
5.10.	ΑΝΑΚΕΦΑΛΑΙΩΣΗ	77
6.	ΤΟ GRAPHONTO ΩΣ ΕΝΑ ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ OWL ΟΝΤΟΛΟΓΙΩΝ	78
6.1.	ΕΙΣΑΓΩΓΗ	78
6.2.	ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ ΣΕ ΕΠΙΠΕΔΟ OWL ΟΝΤΟΛΟΓΙΑΣ	78
6.3.	ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ ΣΕ ΕΠΙΠΕΔΟ OWL ΚΛΑΣΕΩΝ.....	80
6.3.1.	Δένδρο ιεραρχίας κλάσεων	80
6.3.2.	Όψη ιδιοτήτων και σχολιασμών επιλεγμένης κλάσης	83
6.3.3.	Δένδρο ιδιοτήτων επιλεγμένης κλάσης.....	84
6.3.4.	Ορισμός επιλεγμένης κλάσης ως σύνθετης	88
6.3.5.	Όψη σχέσεων της επιλεγμένης κλάσης με τις υπόλοιπες κλάσεις	90
6.4.	ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ ΣΕ ΕΠΙΠΕΔΟ OWL ΙΔΙΟΤΗΤΩΝ	92
6.4.1.	Δένδρο ιεραρχίας ιδιοτήτων.....	92
6.4.2.	Όψη χαρακτηριστικών και σχολιασμών της επιλεγμένης ιδιότητας.....	94
6.4.3.	Όψη σχέσεων επιλεγμένης ιδιότητας με τις υπόλοιπες ιδιότητες.....	95
6.5.	ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ ΣΕ ΕΠΙΠΕΔΟ OWL ΑΤΟΜΩΝ	97
6.5.1.	Δένδρο ατόμων	97
6.5.2.	Όψη γενικών χαρακτηριστικών του επιλεγμένου ατόμου	100
6.5.3.	Όψη ιδιοτήτων τύπου δεδομένων του επιλεγμένου ατόμου	101
6.5.4.	Όψη ιδιοτήτων αντικειμένων του επιλεγμένου ατόμου	101
6.6.	ΔΙΑΚΡΙΣΗ ΣΕ ΑΝΩΤΕΡΕΣ ΟΝΤΟΛΟΓΙΕΣ ΚΑΙ ΣΕ ΟΝΤΟΛΟΓΙΕΣ ΠΕΡΙΟΧΗΣ	102
6.7.	ΈΛΕΓΧΟΣ ΟΡΘΟΤΗΤΑΣ ΠΛΗΡΟΦΟΡΙΑΣ	103
6.7.1.	Εισαγωγή.....	103
6.7.2.	Σημασιολογικοί έλεγχοι σε OWL κλάσεις.....	104
6.7.3.	Σημασιολογικοί έλεγχοι σε OWL περιορισμούς ιδιοτήτων	104
6.7.4.	Σημασιολογικοί έλεγχοι σε OWL ιδιότητες	105
6.7.5.	Σημασιολογικοί έλεγχοι σε OWL άτομα	106
6.8.	ΑΥΤΟΜΑΤΕΣ ΕΝΗΜΕΡΩΣΕΙΣ ΚΑΤΑ ΤΗΝ ΕΠΕΞΕΡΓΑΣΙΑ ΟΝΤΟΛΟΓΙΩΝ	107
6.8.1.	Εισαγωγή.....	107
6.8.2.	Αυτόματες ενημερώσεις σε OWL κλάσεις	108
6.8.3.	Αυτόματες ενημερώσεις σε OWL ιδιότητες	109
6.8.4.	Αυτόματες ενημερώσεις σε OWL άτομα	109
6.9.	ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ ΑΝΑΖΗΤΗΣΗΣ OWL ΔΟΜΩΝ	110
6.9.1.	Εισαγωγή.....	110
6.9.2.	Αναζήτηση OWL κλάσεων.....	110
6.9.3.	Αναζήτηση OWL ιδιοτήτων	111
6.9.4.	Αναζήτηση OWL ατόμων	111
6.9.5.	Αναζήτηση κειμένου στην OWL μορφή	112
6.10.	ΑΝΑΚΕΦΑΛΑΙΩΣΗ	112
7.	ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ ΕΞΑΤΟΜΙΚΕΥΣΗΣ ΧΡΗΣΗΣ ΚΑΙ ΑΠΟΚΡΥΨΗΣ ΠΛΗΡΟΦΟΡΙΑΣ	113
7.1.	ΕΙΣΑΓΩΓΗ	113
7.2.	ΑΝΕΞΑΡΤΗΤΗ ΣΗΜΑΣΙΟΛΟΓΙΚΗ ΠΕΡΙΓΡΑΦΗ ΕΚΤΟΣ ΠΛΑΙΣΙΟΥ ΟΝΤΟΛΟΓΙΑΣ	113
7.2.1.	Εισαγωγή.....	113
7.2.2.	Οδηγός δημιουργίας παράθυρου μεταδεδομένων.....	113

7.2.3.	Κατασκευή παράθυρου μεταδεδομένων.....	115
7.2.4.	Όψη γενικών χαρακτηριστικών επιλεγμένου ατόμου	116
7.2.5.	Όψη ιδιοτήτων τύπου δεδομένων επιλεγμένου ατόμου	117
7.2.6.	Όψη ιδιοτήτων αντικειμένων επιλεγμένου ατόμου	118
7.2.7.	Τελική μορφή δένδρου ατόμων.....	120
7.2.8.	Αποθήκευση σημασιολογικών περιγραφών.....	122
7.2.9.	Δημιουργία και αποθήκευση αρχείων προτύπων	122
7.3.	GRAPHONTO XML ΑΡΧΕΙΑ ΠΡΟΣΑΡΜΟΓΗΣ (CONFIGURATION FILES).....	123
7.3.1.	Εισαγωγή.....	123
7.3.2.	Πληροφορία αρχείων προσαρμογής.....	123
7.3.3.	Πλεονεκτήματα χρήσης αρχείων προσαρμογής.....	124
7.4.	ΑΝΑΚΕΦΑΛΑΙΩΣΗ	125
8.	ΥΠΟΣΤΗΡΙΞΗ ΣΗΜΑΣΙΟΛΟΓΙΚΗΣ ΠΕΡΙΓΡΑΦΗΣ ΠΕΡΙΕΧΟΜΕΝΟΥ ΠΟΛΥΜΕΣΩΝ	126
8.1.	ΕΙΣΑΓΩΓΗ	126
8.2.	ΚΑΝΟΝΕΣ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΥ ΑΠΟ OWL ΣΕ MPEG-7	127
8.2.1.	Εισαγωγή.....	127
8.2.2.	Κανόνες Μετασχηματισμού OWL Οντολογιών.....	127
8.2.3.	Κανόνες Μετασχηματισμού OWL Κλάσεων	129
8.2.4.	Κανόνες Μετασχηματισμού OWL Ιδιοτήτων τύπου δεδομένων	130
8.2.5.	Κανόνες Μετασχηματισμού OWL Ιδιοτήτων αντικειμένων	133
8.2.6.	Κανόνες Μετασχηματισμού OWL Ατόμων.....	136
8.2.7.	Κανόνες Μετασχηματισμού OWL/RDF Μεταδεδομένων.....	141
8.3.	ΧΡΗΣΗ ΟΝΤΟΛΟΓΙΩΝ ΚΑΝΟΝΩΝ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΥ	142
8.4.	ΥΛΟΠΟΙΗΣΗ ΜΕΤΑΣΧΗΜΑΤΙΣΜΩΝ ΚΑΙ ΑΠΟΘΗΚΕΥΣΗ ΣΕ MPEG-7 ΜΟΡΦΗ	143
8.4.1.	Εισαγωγή.....	143
8.4.2.	Μετασχηματισμός οντολογίας περιοχής και αποθήκευση MPEG-7 περιγραφής	143
8.4.3.	Μετασχηματισμός OWL μεταδεδομένων και αποθήκευση MPEG-7 περιγραφής	145
8.5.	ΕΠΑΝΑΧΡΗΣΙΜΟΠΟΙΗΣΗ ΑΠΟΘΗΚΕΥΜΕΝΩΝ MPEG-7 ΠΕΡΙΓΡΑΦΩΝ	147
8.6.	ΑΝΑΚΕΦΑΛΑΙΩΣΗ	149
9.	ΠΕΙΡΑΜΑΤΙΚΗ ΑΞΙΟΛΟΓΗΣΗ	150
10.	ΑΝΑΚΕΦΑΛΑΙΩΣΗ	153
10.1.	ΣΥΜΠΕΡΑΣΜΑΤΑ.....	153
10.2.	ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ	154
ΔΗΜΟΣΙΕΥΣΕΙΣ.....		156
ΑΝΑΦΟΡΕΣ-ΒΙΒΛΙΟΓΡΑΦΙΑ		157
ΠΑΡΑΡΤΗΜΑ 1: ΠΙΝΑΚΑΣ ΠΕΡΙΓΡΑΦΗΣ ΧΡΗΣΙΜΟΠΟΙΟΥΜΕΝΩΝ ΕΙΚΟΝΙΔΙΩΝ .		159
ΠΑΡΑΡΤΗΜΑ 2: XML SCHEMA ΣΤΟ ΟΠΟΙΟ ΥΠΑΚΟΥΝ ΤΑ ΑΡΧΕΙΑ ΠΡΟΣΑΡΜΟΓΗΣ (CONFIGURATION FILES)		161
ΠΑΡΑΡΤΗΜΑ 3: ΟΙ ΚΛΑΣΕΙΣ ΤΟΥ GRAPHONTO ΑΡΙ ΜΑΖΙ ΜΕ ΤΙΣ ΜΕΘΟΔΟΥΣ ΤΟΥΣ		163

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

Εικόνα 1: RDFS ορισμός των κλάσεων "Animal" και "Person".....	16
Εικόνα 2: RDFS ορισμός της ιδιότητας "hasSurname"	16
Εικόνα 3: RDF Περιγραφές για τους ανθρώπους "Jim" και "Joe".....	17
Εικόνα 4: Το μοντέλο γράφου που αναπαριστά τις RDF περιγραφές του σχήματος 3	17
Εικόνα 5: Η ρίζα (πρώτο tag) μιας OWL οντολογίας.....	19
Εικόνα 6: Ορισμοί οντοτήτων στην αρχή της οντολογίας.....	20
Εικόνα 7: Οι αλλαγές στο tag rdf:RDF που προκύπτουν μετά τον ορισμό οντοτήτων	21
Εικόνα 8: Ορισμοί οντοτήτων στην αρχή της οντολογίας.....	21
Εικόνα 9: OWL ορισμός των κλάσεων "Animal" και "Person".....	22
Εικόνα 10: OWL ορισμός του ατόμου "Jim".....	22
Εικόνα 11: OWL ορισμός της ιδιότητας τύπου δεδομένων "hasSurname" που έχει εύρος τιμών τον "string" τύπο της XML Schema και πεδίο ορισμού τα στιγμιότυπα της κλάσης "Person"..	23
Εικόνα 12: OWL ορισμός της ιδιότητας αντικειμένων "hasAnimal" που έχει εύρος τιμών τα στιγμιότυπα της κλάσης "Animal" και πεδίο ορισμού τα στιγμιότυπα της κλάσης "Person". ...	24
Εικόνα 13: OWL ορισμός της ιδιότητας τύπου δεδομένων "tennisGameScore" που έχει εύρος τιμών τις ακέραιες τιμές 0, 15, 30, 40.....	24
Εικόνα 14: OWL ορισμός της ιδιότητας αντικειμένων "physicalCondition" που έχει εύρος τιμών τα άτομα "InShape", "Injured", "Tired" και πεδίο ορισμού τα στιγμιότυπα της κλάσης "Player".	24
Εικόνα 15: OWL ορισμός του ατόμου "Jim" έχοντας τιμές στις ιδιότητες "hasAnimal" και "hasSurname".....	25
Εικόνα 16: OWL ορισμός του περιορισμού ότι κάθε άτομο της κλάσης "Person" θα έχει στην ιδιότητα "hasAnimal" τιμές που θα είναι όλες άτομα της κλάσης "Dog"......	27
Εικόνα 17: OWL ορισμός του περιορισμού ότι κάθε άτομο της κλάσης "Person" θα έχει στην ιδιότητα "hasAnimal" τουλάχιστον μία τιμή που να είναι άτομο της κλάσης "Dog".	27
Εικόνα 18: OWL ορισμός του περιορισμού ότι κάθε άτομο της κλάσης "Person" θα έχει στην ιδιότητα "hasAnimal" τιμή το άτομο "dogJack"......	28
Εικόνα 19: OWL ορισμός του περιορισμού ότι κάθε άτομο της κλάσης "Person" θα έχει ακριβώς μία τιμή στην ιδιότητα "hasAnimal".	28
Εικόνα 20: OWL ορισμός του περιορισμού ότι κάθε άτομο της κλάσης "Person" θα έχει τουλάχιστον μία τιμή στην ιδιότητα "hasAnimal".	29
Εικόνα 21: OWL ορισμός του περιορισμού ότι κάθε άτομο της κλάσης "Person" θα έχει το πολύ μία τιμή στην ιδιότητα "hasAnimal"......	29
Εικόνα 22: OWL ορισμός της κλάσης "Proswpo" που ορίζεται ισοδύναμη της κλάσης "Person" (αν η κλάση "Person" είχε οριστεί σε διαφορετική οντολογία θα έπρεπε να χρησιμοποιηθεί το πλήρες URI της)......	30
Εικόνα 23: OWL ορισμός της ιδιότητας αντικειμένων "exeiZwo" που ορίζεται ισοδύναμη της ιδιότητας "hasAnimal" (αν η ιδιότητα "hasAnimal" είχε οριστεί σε διαφορετική οντολογία θα έπρεπε να χρησιμοποιηθεί το πλήρες URI της).	30
Εικόνα 24: OWL ορισμός του ατόμου "Panagiotis" της κλάσης "Person" που ορίζεται ακριβώς όμοιο του ατόμου "Panos" της κλάσης "Person" (αν το άτομο "Panos" είχε οριστεί σε διαφορετική οντολογία θα έπρεπε να χρησιμοποιηθεί το πλήρες URI του).	31
Εικόνα 25: OWL ορισμός του ατόμου "Panagiotis" της κλάσης "Person" που ορίζεται διαφορετικό των ατόμων "Panos" και "Kwstas" της κλάσης "Person" (αν τα άτομα "Panos" και "Kwstas" είχαν οριστεί σε διαφορετικές οντολογίες θα έπρεπε να χρησιμοποιηθούν τα πλήρη URI τους).	31
Εικόνα 26: OWL ορισμός του ισχυρισμού ότι τα άτομα "Panagiotis", "Panos" και "Kwstas" της κλάσης "Person" είναι αμοιβαία διαφορετικά μεταξύ τους.	31
Εικόνα 27: OWL ορισμός της σύνθετης κλάσης "PersonWithAtLeastOneAnimal" της οποίας η επέκταση κλάσης προκύπτει από την τομή των ατόμων της κλάσης "Person" και των ατόμων που έχουν μία τουλάχιστον τιμή στην ιδιότητα αντικειμένων "has Animal".	32
Εικόνα 28: OWL ορισμός της σύνθετης κλάσης "PersonWithNoAnimal" της οποίας η επέκταση κλάσης προκύπτει από την τομή των ατόμων της κλάσης "Person" και των ατόμων που δεν έχουν καμία τιμή στην ιδιότητα αντικειμένων "hasAnimal" και της σύνθετης κλάσης	

"AllPerson" της οποίας η επέκταση κλάσης προκύπτει από την ένωση των επεκτάσεων κλάσης των κλάσεων "PersonWithNoAnimal" και "PersonWithAtLeastOneAnimal".	33
Εικόνα 29: OWL ορισμός της σύνθετης κλάσης "NoPerson" της οποίας η επέκταση κλάσης περιλαμβάνει όλα τα άτομα που δεν ανήκουν στην επέκταση κλάσης της κλάσης "Person"...	33
Εικόνα 30: OWL ορισμός της σύνθετης κλάσης "JimFriends" της οποίας η επέκταση κλάσης περιλαμβάνει μόνο τα άτομα "Joe", "Sam" και "Nick" της κλάσης "Person".	34
Εικόνα 31: OWL ορισμός της κλάσης "Bird" η οποία δηλώνεται ξένη ως προς τις κλάσεις "Dog" και "Cat".	34
Εικόνα 32: Ορισμός του MPEG-7 Τύπου Δεδομένων "zeroToOneType", που αναπαριστά πραγματικούς αριθμούς μεταξύ 0 και 1.	35
Εικόνα 33: Γενικός ορισμός περιγραφέα.	36
Εικόνα 34: Γενικός ορισμός Σχήματος Περιγραφής.	36
Εικόνα 35: Γενικός ορισμός Σχήματος Κατηγοριοποίησης.	37
Εικόνα 36: Επισκόπηση του MPEG-7 MDS.	39
Εικόνα 37: Στιγμιότυπο από το Protégé-Frames στη σελίδα όπου ορίζονται οι κλάσεις της οντολογίας και τα slots τους.	47
Εικόνα 38: Στιγμιότυπο από το Protégé-OWL στη σελίδα όπου ορίζονται οι κλάσεις της οντολογίας και τα χαρακτηριστικά τους.	47
Εικόνα 39: Στιγμιότυπο από το Protégé-OWL στη σελίδα όπου ορίζονται οι ιδιότητες της οντολογίας και τα χαρακτηριστικά τους.	48
Εικόνα 40: Στιγμιότυπο από το Protégé-OWL στη σελίδα όπου ορίζονται τα άτομα κλάσεων της οντολογίας και τα χαρακτηριστικά τους.	48
Εικόνα 41: Στιγμιότυπο από το SemanticWorks στη σελίδα όπου ορίζονται γραφικά οι κλάσεις μιας οντολογίας και τα χαρακτηριστικά τους.	49
Εικόνα 42: Στιγμιότυπο από το Smore όπου φαίνεται ο τρόπος υπομνημάτισης ιστοσελίδας και γραφικού ορισμού των κλάσεων, των ιδιοτήτων και των ατόμων κλάσεων.	51
Εικόνα 43: Στιγμιότυπο από το Swoor όπου φαίνεται το δένδρο ιεραρχίας κλάσεων και η πληροφορία της επιλεγμένης κλάσης.	52
Πίνακας 1: Πίνακας αξιολόγησης εργαλείων διαχείρισης OWL οντολογιών.	57
Εικόνα 44: Η Αρχιτεκτονική του Πλαισίου DS-MIRF	62
Εικόνα 45: Η Οντολογική υποδομή του Πλαισίου DS-MIRF	64
Εικόνα 46: Η Αρχιτεκτονική του GraphOnto συστήματος.	67
Εικόνα 47: Χαρακτηριστικό στιγμιότυπο της γραφικής διεπαφής του GraphOnto συστήματος όταν τρέχει σε περιβάλλον Windows.	70
Εικόνα 48: Η "OWL Source" καρτέλα στην οποία φαίνονται η OWL μορφή της τρέχουσας οντολογίας σε HTML μορφή.	80
Εικόνα 49: Η "Class Hierarchy" καρτέλα στην οποία φαίνονται αριστερά το δέντρο ιεραρχίας των κλάσεων της οντολογίας και δεξιά οι ιδιότητες και οι σχολιασμοί της επιλεγμένης κλάσης.	81
Εικόνα 50: Η "Class Hierarchy" καρτέλα στην οποία φαίνονται αριστερά το δέντρο ιεραρχίας των κλάσεων της οντολογίας και δεξιά οι σχέσεις της επιλεγμένης κλάσης με τις υπόλοιπες κλάσεις.	81
Εικόνα 51: Οι διαθέσιμες επιλογές του αναδυόμενου μενού του δένδρου ιεραρχίας κλάσεων.	83
Εικόνα 52: Το δένδρο ιδιοτήτων κλάσης και οι διαθέσιμες επιλογές του αναδυόμενου μενού του.	84
Εικόνα 53: Όψη βασικών χαρακτηριστικών ιδιότητας αντικειμένων στο παράθυρο επεξεργασίας ιδιότητας μέσα από κλάση.	85
Εικόνα 54: Όψη περιορισμών ιδιότητας αντικειμένων στο παράθυρο επεξεργασίας ιδιότητας μέσα από κλάση.	86
Εικόνα 55: Όψη σχέσεων ιδιότητας αντικειμένων με τις υπόλοιπες ιδιότητες στο παράθυρο επεξεργασίας ιδιότητας μέσα από κλάση.	87
Εικόνα 56: Όψη υπομνηματισμών ιδιότητας στο παράθυρο επεξεργασίας ιδιότητας μέσα από κλάση.	87
Εικόνα 57: Αρχική όψη παράθυρου δημιουργίας έκφρασης σύνθετης κλάσης.	89
Εικόνα 58: Πρώτο βήμα για προσθήκη περιορισμού στην έκφραση κλάσης στο παράθυρο δημιουργίας έκφρασης σύνθετης κλάσης.	89

Εικόνα 59: Ολοκληρωμένη έκφραση σύνθετης κλάσης, η οποία αποτελείται από την ένωση της κλάσης "Coach" και του περιορισμού στην ιδιότητα αντικειμένων "hasFriend" να έχει όλες οι τιμές από την κλάση "Coach", στο παράθυρο δημιουργίας έκφρασης σύνθετης κλάσης. ...	90
Εικόνα 60: Η "Property Hierarchy" καρτέλα στην οποία φαίνονται αριστερά το δέντρο ιεραρχίας των ιδιοτήτων της οντολογίας και δεξιά τα χαρακτηριστικά και οι σχολιασμοί της επιλεγμένης ιδιότητας.	92
Εικόνα 61: Η "Property Hierarchy" καρτέλα στην οποία φαίνονται αριστερά το δέντρο ιεραρχίας των ιδιοτήτων της οντολογίας και δεξιά οι σχέσεις της ιδιότητας με τις υπόλοιπες ιδιότητες.	93
Εικόνα 62: Οι διαθέσιμες επιλογές του αναδυόμενου μενού του δένδρου ιεραρχίας ιδιοτήτων.	94
Εικόνα 63: Η καρτέλα "Individuals" στην οποία φαίνονται αριστερά το δέντρο των ατόμων της οντολογίας και δεξιά τα γενικά χαρακτηριστικά του επιλεγμένου ατόμου.	97
Εικόνα 64: Η καρτέλα "Individuals" στην οποία φαίνονται αριστερά το δέντρο των ατόμων της οντολογίας και δεξιά οι τιμές του επιλεγμένου ατόμου στις ιδιότητες τύπου δεδομένων που του αντιστοιχούν.	98
Εικόνα 65: Η καρτέλα "Individuals" στην οποία φαίνονται αριστερά το δέντρο των ατόμων της οντολογίας και δεξιά οι τιμές του επιλεγμένου ατόμου στις ιδιότητες αντικειμένων που του αντιστοιχούν.	98
Εικόνα 66: Οι διαθέσιμες επιλογές του αναδυόμενου μενού του δένδρου ατόμων.	99
Εικόνα 67: Η όψη κλάσεων μιας ανώτερης οντολογίας.	103
Εικόνα 68: Η φόρμα αναζήτησης στην όψη ιδιοτήτων στην οποία φαίνεται ότι αναζήτηση ιδιότητας έφθασε στο τέλος του δένδρου ιεραρχίας ιδιοτήτων.	111
Εικόνα 69: Η φόρμα αναζήτησης στην όψη ιδιοτήτων στην οποία φαίνεται ότι αναζήτηση ιδιότητας δεν είχε αποτελέσματα.	111
Εικόνα 70: Το 1 ^ο βήμα του οδηγού δημιουργίας παραθύρου μεταδεδομένων.	114
Εικόνα 71: Το 2 ^ο βήμα του οδηγού δημιουργίας παραθύρου μεταδεδομένων.	115
Εικόνα 72: Η αρχική όψη του παραθύρου μεταδεδομένων που προέκυψε ύστερα από τα δύο βήματα του οδηγού.	116
Εικόνα 73: Η όψη του παραθύρου επιλογής ιδιοτήτων για τύπο που δεν είχε επιλεγεί ύστερα από τα δύο βήματα του οδηγού.	119
Εικόνα 74: Η εξαπλωση του δένδρου ατόμων του παραθύρου μεταδεδομένων ύστερα από την προσθήκη τιμών στα πρότυπα άτομα.	121
Εικόνα 75: Η XML μορφή ενός αρχείου προσαρμογής.	124
Εικόνα 76: Παράδειγμα OWL οντολογίας ποδοσφαίρου.	128
Εικόνα 77: MPEG-7 XML Έγγραφο που παράγεται από την Οντολογία της προηγούμενης εικόνας.	129
Εικόνα 78: OWL κλάσεις που αναπαριστούν αγωνιστικές δραστηριότητες στο ποδόσφαιρο (GameAction) και δραστηριότητες παικτών (PlayerAction).	130
Εικόνα 79: Τμήμα του MPEG-7 XML Εγγράφου που παράγεται από της OWL κλάσεις που φαίνονται στην προηγούμενη εικόνα.	130
Εικόνα 80: Ορισμός προκαθορισμένης τιμής σε OWL για τις ιδιότητες τύπου δεδομένων "lang" και "Email" της κλάσης "AEKPlayer" οι οποίες κληρονομούνται από πρόγονους της κλάσης στην ανώτερη οντολογία.	131
Εικόνα 81: Τμήμα του MPEG-7 XML Εγγράφου που παράγεται από την OWL κλάση και τις προκαθορισμένες τιμές της σε ιδιότητες τύπου δεδομένων που φαίνονται στην προηγούμενη εικόνα.	132
Εικόνα 82: Ορισμός προκαθορισμένης τιμής σε OWL για την ιδιότητα τύπου δεδομένων "ShirtNumber" για την κλάση "Goalkeeper".	132
Εικόνα 83: Τμήμα του MPEG-7 XML Εγγράφου που παράγεται από την OWL κλάση και την προκαθορισμένη τιμή της στην ιδιότητα τύπου δεδομένων που φαίνονται στην προηγούμενη εικόνα.	132
Εικόνα 84: OWL Άτομο τύπου "PostalAddressType", που αναπαριστά τη διεύθυνση των ποδοσφαιριστών της AEK και αποτελεί την υποχρεωτική τιμή της ιδιότητας "PostalAddress" για τα άτομα που ανήκουν στην κλάση "AEKPlayer".	134
Εικόνα 85: Η τιμή της ιδιότητας "PostalAddress" για τα άτομα που ανήκουν στην κλάση "AEKPlayer" σε MPEG-7 σύνταξη.	134

Εικόνα 86: MPEG-7 σύνταξη για τον ορισμό της προκαθορισμένης τιμής του ατόμου που αναπαριστά την εταιρεία "X" στην ιδιότητα "Sponsor" τιμή για τα άτομα της κλάσης "TournamentSoccerTeam"	135
Εικόνα 87: OWL Άτομο τύπου "TermUseType", που αναπαριστά την υποχρεωτική τιμή της ιδιότητας "role" για τα άτομα που ανήκουν στην κλάση "Midfielder".....	136
Εικόνα 88: MPEG-7 σύνταξη για τον ορισμό της προκαθορισμένης τιμής της ιδιότητας "role" τιμή για τα άτομα της κλάσης "Midfielder"	136
Εικόνα 89 : Άτομο που αναπαριστά τον ποδοσφαιριστή Βαγγέλη Μόρα.	139
Εικόνα 90: Τμήμα του MPEG-7 XML Εγγράφου που παράγεται από το άτομο που φαίνεται στην εικόνα 87.....	140
Εικόνα 91: Στιγμιότυπο της εφαρμογής με ανοιχτές όλες τις οντολογίες που περιλαμβάνει η οντολογική υποδομή του πλαισίου DS-MIRF και επιλογή μετασχηματισμού της τρέχουσας οντολογίας σε MPEG-7 μορφή.	144
Εικόνα 92: Στιγμιότυπο του παράθυρου μεταδεδομένων και εμφάνιση της επιλογής αναζήτησης ατόμων στις οντολογίες ή στην αποθήκη μεταδεδομένων.....	148
Πίνακας 2: Χαρακτηριστικά των οντολογιών των πειραμάτων	150
Πίνακας 3: Αποτελέσματα αξιολόγησης απόδοσης μεταξύ GraphOnto και Protégé στο διάβασμα οντολογίας.....	151
Πίνακας 4: Χαρακτηριστικά των οντολογιών που περιλαμβάνονται στην οντολογική υποδομή του DS-MIRF πλαισίου	152

Κεφάλαιο 1^ο

1. Εισαγωγή

1.1. Γενικά

Η μεγάλη ανάπτυξη του διαδικτύου που σημειώνεται τα τελευταία χρόνια και η αυξανόμενη ζήτηση για παροχή υπηρεσιών ψηφιακών πολυμέσων μέσω αυτού κατέδειξε την σπουδαιότητα των προτύπων (standards), όπως το MPEG-7, που περιγράφουν τα χαρακτηριστικά και τη σημασιολογία του περιεχομένου του οπτικό-ακουστικού υλικού. Η διαλειτουργικότητα (interoperability) ανάμεσα στα κοινώς παραδεκτά πρότυπα κρίνεται αναγκαία σε ανοιχτά περιβάλλοντα, έτσι ώστε να καταστεί δυνατή η επικοινωνία, η ανταλλαγή πληροφορίας και η συντονισμένη λειτουργία ανάμεσα σε υπηρεσίες διαχείρισης του περιεχομένου των πολυμέσων.

Η υποστήριξη διαλειτουργικότητας βασίζεται συνήθως σε οντολογίες όπου η σημασιολογία των προτύπων εμπεριέχεται σε ανώτερες οντολογίες. Παράλληλα με τα πρότυπα περιγραφής πολυμέσων, η γνώση για ένα συγκεκριμένο πεδίο θεωρείται εξαιρετικά χρήσιμη για τον εμπλουτισμό της λειτουργικότητας (δεικτοδότηση, προσδιορισμός ερωτήσεων, ανάκτηση κτλ.) των repositories αποθήκευσης πληροφορίας πολυμέσων. Όμως οι παρεχόμενες δομές από τα γενικά πρότυπα, όπως το MPEG-7, είναι γενικού σκοπού και κρίνονται ανεπαρκείς για να αναπαραστήσουν τη γνώση για ένα συγκεκριμένο πεδίο.

Για το λόγο αυτό, έχει προταθεί το DS-MIRF (Domain Specific Multimedia Indexing, Retrieval and Filtering) πλαίσιο (framework) στο οποίο έχουν αναπτυχθεί μεθοδολογίες για την επέκταση των ανώτερων οντολογιών που βασίζονται στα πρότυπα με γνώση για συγκεκριμένο πεδία που βρίσκεται σε οντολογίες που αφορούν συγκεκριμένα πεδία. Όμως, το μέγεθος και η πολυπλοκότητα των οντολογιών που περιγράφουν τα πρότυπα και τα διάφορα πεδία τις καθιστά δύσκολες στη χρήση και χρονοβόρες.

Όλα τα παραπάνω αποτέλεσαν το κίνητρο για την υλοποίηση του GraphOnto, ενός συστήματος το οποίο διευκολύνει τον ορισμό και τη δημιουργία ανώτερων οντολογιών και οντολογιών συγκεκριμένου πεδίου καθώς και τη χρήση αυτών των οντολογιών σε συστήματα περιγραφής πολυμέσων. Το GraphOnto, το οποίο βασίζεται στην OWL, (το κυρίαρχο πρότυπο στις γλώσσες ορισμού οντολογιών), παρέχει κατάλληλους μηχανισμούς για να αποκρύπτει τις πολύπλοκες έννοιες που ενδέχεται να περιέχουν οι οντολογίες και επιτρέπει στο χρήστη που επιθυμεί μόνο τη σημασιολογική περιγραφή του περιεχομένου πολυμέσων να επικεντρωθεί μόνο στις συγκεκριμένες έννοιες που τον ενδιαφέρουν.

1.2. Συνεισφορά

Το GraphOnto είναι ένα πλήρες σύστημα το οποίο μπορεί να παρουσιαστεί μέσα από δύο διαφορετικές όψεις: Αφενός ως ένα γραφικό σύστημα διαχείρισης OWL οντολογιών και

αφετέρου ως ένα εργαλείο που διευκολύνει τη σημασιολογική περιγραφή του οπτικό-ακουστικού υλικού με τη χρήση οντολογιών. Το GraphOnto, με το μεγάλο φάσμα παρεχόμενης λειτουργικότητας, καλύπτει τις ανάγκες τόσο του χρήστη που ασχολείται με τη δημιουργία οντολογιών όσο και του χρήστη που χρησιμοποιεί τις οντολογίες ως ένα χρήσιμο βοήθημα για τη σημασιολογική περιγραφή του περιεχομένου πολυμέσων.

Ως πλήρης εφαρμογή επεξεργασίας OWL οντολογιών, το GraphOnto παρέχει την πλήρη λειτουργικότητα που ορίζουν οι δομές της OWL (για την ακρίβεια της OWL-DL, μιας από τις τρεις υπογλώσσες της OWL) και δεν υστερεί σε σχέση με τα υπάρχοντα εργαλεία επεξεργασίας OWL οντολογιών (ανάμεσα στα οποία πιο γνωστό και ολοκληρωμένο είναι το Protégé). Απεναντίας, περιλαμβάνει πολλά επιπρόσθετα χαρακτηριστικά όπως η ταυτόχρονη διαχείριση πολλαπλών οντολογιών, η διάκριση ανάμεσα σε ανώτερες οντολογίες και οντολογίες περιοχής, η δυνατότητα ορισμού OWL ατόμων (individuals) που βασίζονται σε συγκεκριμένες οντολογίες εκτός του πλαισίου αυτών, η υποστήριξη εξατομίκευσης σε επίπεδο εφαρμογής και χρήστη και η χρήση αντιστοιχήσεων μεταξύ OWL οντολογιών, με βάση τις αντίστοιχες OWL δομές, που επιτρέπουν τον ορισμό εξατομικευμένων οντολογιών. Παράλληλα, συγκριτικές δοκιμές αξιολόγησης κατέδειξαν την υπεροχή του GraphOnto σε σχέση με το Protégé από πλευράς απόδοσης.

Ως εργαλείο για τη σημασιολογική περιγραφή το περιεχομένου οπτικό-ακουστικού υλικού, το GraphOnto χρησιμοποιείται στο πλαίσιο του DS-MIRF και επιτρέπει την περιγραφή αντικειμένων πολυμέσων κάνοντας χρήση της ανώτερης οντολογίας του DS-MIRF Framework η οποία αναπαριστά πλήρως σε OWL τη σημασιολογία του MPEG-7 MDS προτύπου. Παράλληλα επιτρέπει τη χρήση και τον ορισμό OWL οντολογιών που περιγράφουν διάφορα πεδία, οι οποίες ενσωματώνονται στο πλαίσιο DS-MIRF και επεκτείνουν την ανώτερη οντολογία με γνώση για συγκεκριμένα πεδία. Αυτές οι OWL οντολογίες μαζί με τις OWL σημασιολογικές περιγραφές που ορίζονται χρησιμοποιώντας τες, μετασχηματίζονται με βάση τους κανόνες μετασχηματισμού του πλαισίου DS-MIRF σε MPEG-7 σημασιολογικές περιγραφές. Αυτές οι MPEG-7 περιγραφές αποθηκεύονται είτε σε συμβατά με το MPEG-7 XML αρχεία είτε στην αποθήκη (repository) μεταδεδομένων του πλαισίου DS-MIRF που υποστηρίζει την αποθήκευση και ανάκτηση MPEG-7 περιγραφών.

Τέλος, το GraphOnto παρέχει τη λειτουργικότητα διαχείρισης OWL οντολογιών μέσω ενός JAVA API το οποίο μπορεί να χρησιμοποιηθεί από μια οποιαδήποτε εξωτερική εφαρμογή. Ήδη ορισμένα εργαλεία, όπως το OntoNL, που υλοποιούνται στο πλαίσιο DS-MIRF και βασίζονται σε μια οντολογική προσέγγιση, κάνουν χρήση του GraphOnto API.

1.3. Δομή της εργασίας

Το υπόλοιπο τμήμα αυτής της εργασίας είναι οργανωμένο ως εξής: Στο 2^ο κεφάλαιο γίνεται αναλυτική παρουσίαση των προτύπων OWL και MPEG-7 στα οποία βασίστηκαν ο σχεδιασμός και η υλοποίηση του συστήματος GraphOnto. Στο 3^ο κεφάλαιο γίνεται

ανασκόπηση σε σχετικές εργασίες που έχουν γίνει στον τομέα των OWL οντολογιών. Στο 4^ο κεφάλαιο παρουσιάζεται το πλαίσιο DS-MIRF μέσα στο οποίο εντάσσεται το GraphOnto. Στη συνέχεια, η αρχιτεκτονική του συστήματος περιγράφεται με λεπτομέρεια στο 5^ο κεφάλαιο. Στο 6^ο κεφάλαιο παρουσιάζεται η πλήρης παρεχόμενη λειτουργικότητα από το GraphOnto ως αμιγές σύστημα διαχείρισης OWL οντολογιών. Στο 7^ο κεφάλαιο γίνεται αναφορά στα επιπρόσθετα χαρακτηριστικά του GraphOnto που επιτρέπουν εξατομίκευση χρήσης και απόκρυψη πληροφορίας. Στο 8^ο κεφάλαιο παρουσιάζεται η πλήρης παρεχόμενη λειτουργικότητα από το GraphOnto ως ένα εργαλείο για τη σημασιολογική περιγραφή περιεχομένου πολυμέσων. Το 9^ο κεφάλαιο καταγράφει τα αποτελέσματα των συγκριτικών δοκιμασιών αξιολόγησης στον τομέα της απόδοσης μεταξύ του GraphOnto και του Protégé. Τέλος το 10^ο κεφάλαιο περιλαμβάνει τα συμπεράσματα που προέκυψαν από αυτήν την εργασία καθώς και τις κατευθύνσεις για μελλοντική έρευνα.

Κεφάλαιο 2^ο

2. Σχετικές τεχνολογίες

2.1. Εισαγωγή

Αυτό το κεφάλαιο περιγράφει τα δύο βασικά πρότυπα (standards) που μελετήθηκαν και επηρέασαν το σχεδιασμό και την υλοποίηση του GraphOnto συστήματος, τα οποία είναι:

1. Η OWL (Web Ontology Language) που αποτελεί το κυρίαρχο πρότυπο στις γλώσσες περιγραφής οντολογιών και της οποίας οι δομές υποστηρίζονται απόλυτα από το GraphOnto (ενότητα 2.2).
2. Το MPEG-7 και πιο συγκεκριμένα το MPEG-7 Multimedia Description Schemes (MDS), όπου περιγράφονται δομές για τον ορισμό μεταδεδομένων για οπτικό-ακουστικό υλικό (παράγραφος 2.3). Το MPEG-7 MDS αποτελεί τη βάση για τη σημασιολογική περιγραφή οπτικό-ακουστικού υλικού που υλοποιείται μέσα στο GraphOnto.

2.2. OWL (Web Ontology Language)

2.2.1. Εισαγωγή

Ο Παγκόσμιος Ιστός (World Wide Web) με τη σημερινή μορφή του είναι εξαιρετικά δύσκολο να χαρτογραφηθεί καθώς ο μόνος τρόπος περιγραφής της σημασιολογίας των εγγράφων του είναι οι λέξεις-κλειδιά. Έτσι κρίνεται απαραίτητο να χρησιμοποιηθούν κατάλληλες περιγραφές των πηγών (resources) του παγκόσμιου ιστού οι οποίες θα είναι κατανοητές από εφαρμογές στο διαδίκτυο.

Η OWL (Web Ontology Language) σχεδιάστηκε ως μια γλώσσα για τον ορισμό web οντολογιών που θα περιγράφουν το περιεχόμενο των εγγράφων του παγκόσμιου ιστού. Ο όρος οντολογία είναι δανεισμένος από τη φιλοσοφία και αναφέρεται στην επιστήμη που περιγράφει τα είδη των οντοτήτων στον πραγματικό κόσμο, τις ιδιότητες αυτών των οντοτήτων και στον τρόπο με τον οποίο σχετίζονται αυτές μεταξύ τους.

Πριν περιγράψουμε όμως την OWL θα κάνουμε μια μικρή αναδρομή για το πώς προέκυψε η ανάγκη να δημιουργηθούν οι γλώσσες περιγραφής οντολογιών (παράγραφος 2.2.2) και θα αναφερθούμε στην RDF / RDFS (εκτενής αναφορά στην παράγραφο 2.2.3 καθώς η OWL βασίζεται πάνω σε αυτήν και την επεκτείνει) και στην DAML+OIL (συνοπτικά στην παράγραφο 2.2.4) οι οποίες αποτελούν τις προγενέστερες της OWL γλώσσες οντολογιών.

2.2.2. Εισαγωγή στις γλώσσες του Semantic Web

Η XML [1] και η XML-Schema [2] επιτρέπουν την δομημένη και ιεραρχική περιγραφή των δεδομένων, αλλά στερούνται της δυνατότητας να περιγράψουν το νόημα και τη σημασία των δεδομένων. Για παράδειγμα υποστηρίζουν μια κοινή σύνταξη για την κωδικοποίηση των δεδομένων σε γλώσσα κατανοητή από υπολογιστή, δε διαθέτουν όμως δομές ικανές να περιγράψουν τη σημασιολογική πληροφορία των δεδομένων ώστε να κατανοηθούν και κατόπιν να χρησιμοποιηθούν από υπολογιστή. Έτσι, παρότι παρέχουν συντακτική διαλειτουργικότητα (interoperability), αδυνατούν να παράσχουν σημασιολογική διαλειτουργικότητα ανάμεσα σε εφαρμογές που ανταλλάσσουν πληροφορία στον παγκόσμιο σε γλώσσα κατανοητή από υπολογιστή.

Για το λόγο αυτό, το W3C [3] (World Wide Web Consortium) άρχισε να εργάζεται στο πεδίο αυτό από το τέλος της δεκαετίας του 90 προκειμένου να ορίσει σημασιολογικές προεκτάσεις στην XML. Το πρώτο βήμα προς αυτήν την κατεύθυνση ήταν η έκδοση της RDF το 1999.

2.2.3. RDF / RDFS

Η RDF (Resource Description Framework) [4], είναι μια ανεξάρτητη περιοχής και γενικού σκοπού γλώσσα που εκφράζει πληροφορία που αφορά πηγές (resources) του παγκόσμιου ιστού.

Στην RDF, οι πηγές είναι αντικείμενα που αναγνωρίζονται μοναδικά από URIs (δηλαδή αλφαριθμητικά που αναγνωρίζουν μοναδικά τις πηγές στον παγκόσμιο ιστό) [5] και χαρακτηρίζονται από ένα σύνολο από ιδιότητες οι οποίες ορίζουν δυαδικές σχέσεις μεταξύ των πηγών και των αντιστοιχισμένων τιμών. Αυτό εκφράζεται μέσω τριάδων (triplets), επίσημα αναφερόμενες ως προτάσεις (statements), της μορφής [υποκείμενο (subject), κατηγορημα (predicate), αντικείμενο (object)]. Πιο συγκεκριμένα, το υποκείμενο είναι η πηγή που περιγράφεται, το κατηγορημα είναι μια ιδιότητα της περιγραφόμενης πηγής και το αντικείμενο είναι η τιμή της ιδιότητας. Η τιμή μιας ιδιότητας μπορεί με τη σειρά της να είναι μια κυριολεκτική τιμή (literal) (π.χ. ένας ακέραιος, ένα αλφαριθμητικό ή μια ημερομηνία) ή μια άλλη πηγή. Τέλος, ένα σύνολο από προτάσεις που αφορούν την ίδια πηγή καλείται περιγραφή (description) της πηγής.

Η RDF είναι μια συντακτική επέκταση της XML (για παράδειγμα βελτιώνει την εκφραστικότητα της) και συνεπώς βασίζεται σε tags όπως και η XML. Για την ακρίβεια, ορίζει πως μπορεί να προσδιοριστεί μια περιγραφή πηγής, αλλά όχι και το νόημα μιας τέτοιας περιγραφής, η οποία πρέπει να βασίζεται σε κοινά λεξιλόγια, δηλαδή περιγραφές των ιδιοτήτων των πηγών. Για το λόγο αυτό, το W3C πρότεινε μια σημασιολογική επέκταση της RDF, η οποία ονομάστηκε RDFS [6] (RDF Schema) και η οποία επιτρέπει να οριστούν όροι που χρησιμοποιούνται σε RDF προτάσεις και να προσδιοριστεί το νόημα τους. Να σημειωθεί

ότι η RDFS δεν ορίζει ένα σύνολο από ιδιότητες, αλλά αντίθετα παρέχει δομές για να ορίζονται οι ιδιότητες. Αυτό επιτυγχάνεται με τον ορισμό ιδιοτήτων με βάση την κλάση (class) των πηγών στις οποίες εφαρμόζουν οι ιδιότητες.

Ως ένα παράδειγμα για το πώς λειτουργεί η RDFS, ας θεωρήσουμε τον RDFS ορισμό της κλάσης "Person" στο σχήμα 1, η οποία επιτρέπει την αναπαράσταση περιγραφών από ανθρώπους. Η κλάση "Person" είναι μια υποκλάση της κλάσης "Animal" που αναπαριστά περιγραφές ζώων. Επιπλέον, θεωρούμε ότι οι άνθρωποι έχουν ονόματα (που είναι αλφαριθμητικά), κι έτσι ορίζουμε μια ιδιότητα με το όνομα "hasSurname" (όπως φαίνεται στο σχήμα 2) η οποία έχει την κλάση "Person" ως πεδίο ορισμού (domain) και τον αλφαριθμητικό (string) τύπο ως εύρος τιμών (range) και η οποία μας επιτρέπει να προσδίδουμε ονόματα στους ανθρώπους. Η RDFS συνεπώς επιτρέπει στον ορισμό του εύρους τιμών να χρησιμοποιούνται οι έτοιμοι τύποι δεδομένων της XML Schema και αναφέρεται σε αυτούς μέσω του URI <http://www.w3.org/2001/XMLSchema>. Οι σημαντικότεροι από αυτούς τους τύπους δεδομένων είναι: xsd:string, xsd:normalizedString, xsd:boolean, xsd:decimal, xsd:float, xsd:double, xsd:integer, xsd:long, xsd:int, xsd:short, xsd:byte, xsd:nonNegativeInteger, xsd:positiveInteger, xsd:nonPositiveInteger, xsd:negativeInteger.

```
<rdfs:Class rdf:ID="Animal"/>
<rdfs:Class rdf:ID="Person">
  <rdfs:subClassOf rdf:resource="#Animal"/>
</rdfs:Class>
```

Εικόνα 1: RDFS ορισμός των κλάσεων "Animal" και "Person"

```
<rdfs:Property rdf:ID="hasSurname">
  <rdfs:domain rdf:resource="#Person" />
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</rdfs:Property>
```

Εικόνα 2: RDFS ορισμός της ιδιότητας "hasSurname"

Έτσι σε αναλογία με την XML Schema που επιτρέπει να οριστούν γλώσσες για την περιγραφή της δομής κλάσεων συγκεκριμένων εγγράφων, η RDFS επιτρέπει τον ορισμό γλωσσών για την περιγραφή της σημασιολογίας συγκεκριμένων κλάσεων. Οι γλώσσες που βασίζονται στην RDFS είναι στην ουσία οντολογίες που επιτρέπουν αλληλεπίδραση περιγραφών πληροφορίας που σχετίζονται σημασιολογικά με ένα συγκεκριμένο πεδίο, παρά το γεγονός ότι αυτές οι περιγραφές είναι δυνατόν να έχουν δομηθεί χρησιμοποιώντας διαφορετικά XML σχήματα.

Τα RDF / RDFS μοντέλα αναπαρίστανται σαν γράφοι (graphs). Ως παράδειγμα, ας θεωρήσουμε τις RDF περιγραφές που φαίνονται στο σχήμα 3 σχετικά με τις πηγές <http://example.org/PersonInfo#Jim> και <http://example.org/PersonInfo#Joe> που περιγράφουν αντίστοιχα τους ανθρώπους "Jim" και "Joe" και δομούνται με βάση τα RDFS σχήματα των σχημάτων 1 και 2.

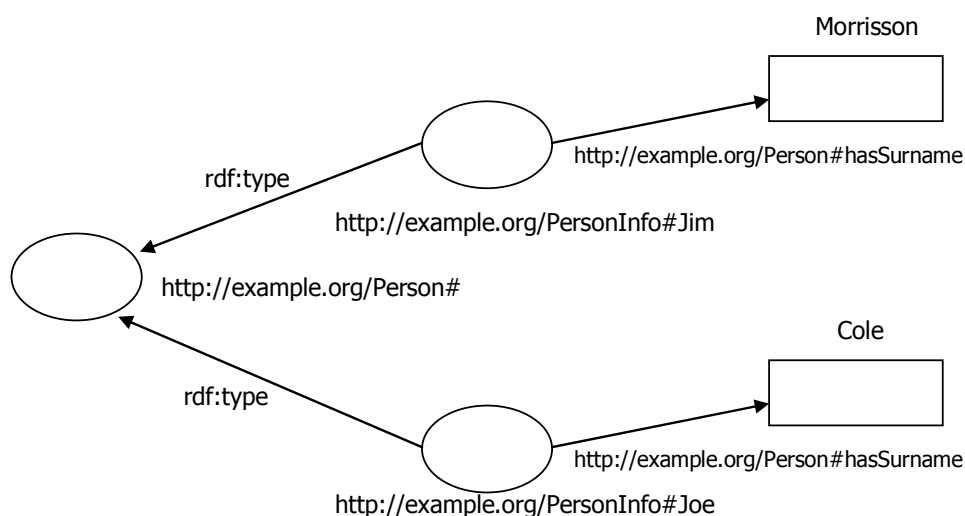

```

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" xmlns:schema="http://example.org/Person#">
  <schema:Person rdf:ID="http://example.org/PersonInfo#Jim">
    <hasSurname>Morrisson</hasSurname>
  </schema:Person>
  <schema:Person rdf:ID="http://example.org/PersonInfo#Joe">
    <hasSurname>Cole</hasSurname>
  </schema:Person>
</rdf:RDF>

```

Εικόνα 3: RDF Περιγραφές για τους ανθρώπους "Jim" και "Joe"

Το μοντέλο γράφου που αναπαριστά τις παραπάνω RDF περιγραφές και το οποίο δομείται με βάση τις προδιαγραφές της RDF φαίνεται στο σχήμα 4. Παρατηρούμε ότι οι πηγές αναπαρίστανται από ελλείψεις, ενώ οι κυριολεκτικές τιμές (literals) των ιδιοτήτων από ορθογώνια παραλληλόγραμμα. Οι κατευθυνόμενες ακμές αναπαριστούν τις ιδιότητες και ενώνουν τις πηγές με άλλες πηγές ή με τις τιμές των ιδιοτήτων τους. Το βέλος της ακμής φανερώνει την τιμή της ιδιότητας, ενώ η αρχή της ακμής φανερώνει το υποκείμενο της ιδιότητας. Παρατηρούμε επίσης ότι μια RDF περιγραφή που ανήκει στην κλάση "Person" είναι ουσιαστικά μια περιγραφή του τύπου (rdf:type) <http://example.org/Person#>. Συνεπώς μέσω του rdf:type, που θεωρείται και αυτό ιδιότητα, δηλώνεται ότι μια πηγή είναι στιγμιότυπο μιας κλάσης.



Εικόνα 4: Το μοντέλο γράφου που αναπαριστά τις RDF περιγραφές του σχήματος 3

Παρά το γεγονός ότι η RDF / RDFS οδήγησε σε μεγάλη πρόοδο όσον αφορά την υπερπήδηση των περιορισμών της XML, δεν είναι τόσο εκφραστική από σημασιολογική άποψη. Παραδείγματος χάρη, για μια ιδιότητα δεν μπορούν να οριστούν χαρακτηριστικά (όπως συμμετρικότητα ή μεταβατικότητα) ή περιορισμοί στον αριθμό εμφάνισης της ή στις τιμές που μπορεί να πάρει. Σαν αποτέλεσμα, το W3C αποφάσισε να επεκτείνει το XML/RDF πλαίσιο υποστηρίζοντας τυπικά γνωρίσματα γλωσσών αναπαράστασης γνώσης (knowledge

representation). Η πρώτη προσπάθεια προς αυτήν την κατεύθυνση έφερε την έκδοση της DAML+OIL.

2.2.4. DAML+OIL

Η DAML+OIL [7] είναι μια γλώσσα για αναπαράσταση οντολογιών, η οποία προέκυψε από τη συγχώνευση των αποτελεσμάτων δύο ερευνητικών προγραμμάτων που χρηματοδοτήθηκαν από τις Η.Π.Α. και την Ευρωπαϊκή Ένωση, τα DAML-ONT (DARPA Agent Markup Language for Ontologies) και OIL (Ontology Inference Layer) αντίστοιχα.

Η DAML+OIL είναι μια γλώσσα περιγραφικής λογικής (Description Logic) [8], που βασίζεται στην RDF / RDFS την οποία επεκτείνει με περισσότερους τύπους. Επιπλέον παρέχει μηχανισμούς εξαγωγής συμπερασμάτων (inference mechanisms) υποστηρίζοντας:

1. Υπαγωγή (subsumption): Αποφασίζει αν η κλάση A είναι υποκλάση της κλάσης B δοθέντων των ορισμών τους.
2. Αναγνώριση (recognition): Αποφασίζει αν το στιγμιότυπο (instance) A1 ανήκει στην κλάση A.
3. Ταξινόμηση (classification): Αναδιοργανώνει αυτόματα την ιεραρχία των κλάσεων με βάση τους ορισμούς τους.

Έχοντας ως βάση την DAML+OIL, το W3C πρότεινε την δημιουργία της OWL το 2002.

2.2.5. Εισαγωγή στην OWL και τα είδη της OWL

Η OWL [9] αποτελεί μια πρότυπη γλώσσα για την αναπαράσταση οντολογιών που χρησιμοποιούνται στο RDF / XML πλαίσιο και στοχεύει στην παροχή πλήρους υποστήριξης των γνωρισμάτων των γλωσσών αναπαράστασης γνώσης, επεκτείνοντας την εκφραστικότητα της DAML+OIL. Επειδή όμως η εκφραστικότητα και η πολυπλοκότητα της εξαγωγής συμπερασμάτων (reasoning) είναι μεγέθη αντιστρόφως ανάλογα, η στρατηγική που υιοθετήθηκε ήταν να παρέχονται τρία διαφορετικά είδη της OWL, τα οποία θα χρησιμοποιούνται ανάλογα με τις ανάγκες και τις απαιτήσεις σε εκφραστικότητα και πολυπλοκότητα του κάθε πεδίου που θα περιγράφεται μέσω της OWL. Τα τρία είδη της OWL είναι της εξής:

1. *OWL Lite*: Είναι το πιο μικρό υποσύνολο της OWL και εγγυάται υπολογιστική περατότητα (computational completeness) καθώς η εξαγωγή συμπερασμάτων μπορεί να υπολογιστεί στη χειρότερη περίπτωση (worst case) σε εκθετικό χρόνο. Όμως είναι σχετικά φτωχή σε εκφραστικότητα καθώς δεν υποστηρίζει έννοιες όπως τομή και ένωση συνόλων, ούτε επιτρέπει στο πλήθος των τιμών (cardinality constraints) μιας ιδιότητας να υπερβαίνει το 1.

2. *OWL-DL (OWL Description Logics)*: Η OWL-DL επεκτείνει την εκφραστικότητα της OWL Lite με λειτουργικότητα που υποστηρίζει περιγραφική λογική (Description Logics), και η οποία εμπλουτίζεται με τη δυνατότητα ορισμών νέων τύπων δεδομένων (datatypes). Η OWL-DL εγγυάται και αυτή με τη σειρά της υπολογιστική περατότητα στην εξαγωγή συμπερασμάτων.
3. *OWL Full*: Η OWL Full υποστηρίζει όλα τα γνωρίσματα της περιγραφικής λογικής και της RDF παρέχοντας την πιο πλούσια εκφραστικότητα από τα υπόλοιπα είδη της OWL. Αυτή όμως η μεγάλη εκφραστικότητα έχει ως αποτέλεσμα οι οντολογίες που είναι γραμμένες σε OWL Full να είναι undecidable, δηλαδή η OWL Full δεν εγγυάται υπολογιστική περατότητα στην εξαγωγή συμπερασμάτων.

Κάθε ένα από τα τρία είδη αποτελεί επέκταση του απλούστερου προγόνου του, και στο τι μπορεί έγκυρα να εκφραστεί και στο τι μπορεί αξιόπιστα να συμπεραθεί. Έτσι το παρακάτω σύνολο από σχέσεις ισχύει:

1. Κάθε έγκυρη OWL Lite οντολογία είναι και έγκυρη OWL-DL οντολογία.
2. Κάθε έγκυρη OWL-DL οντολογία είναι και έγκυρη OWL Full οντολογία.
3. Κάθε αξιόπιστο OWL Lite συμπέρασμα είναι και αξιόπιστο OWL-DL συμπέρασμα.
4. Κάθε αξιόπιστο OWL-DL συμπέρασμα είναι και αξιόπιστο OWL Full συμπέρασμα.

Το GraphOnto επιλέχθηκε να υποστηρίξει την OWL-DL η οποία παρέχει σαφώς μεγαλύτερη εκφραστικότητα σε αντίθεση με την OWL Lite και ταυτόχρονα εγγυάται την υπολογιστική περατότητα των συστημάτων εξαγωγής συμπερασμάτων σε αντίθεση με την OWL Full.

2.2.6. Η δομή μιας OWL οντολογίας

Το λεξιλόγιο της OWL είναι μια επέκταση του λεξιλογίου της RDF. Μια τυπική OWL οντολογία είναι ένα well-formed XML έγγραφο που ξεκινάει με το tag `rdf:RDF` το οποίο περιλαμβάνει μια σειρά από δηλώσεις των namespaces [10] που θα χρησιμοποιηθούν.

```
<rdf:RDF
  xmlns    ="http://www.w3.org/TR/2004/REC-owl-guide-20040210/wine#"
  xmlns:vin ="http://www.w3.org/TR/2004/REC-owl-guide-20040210/wine#"
  xml:base ="http://www.w3.org/TR/2004/REC-owl-guide-20040210/wine#"
  xmlns:food="http://www.w3.org/TR/2004/REC-owl-guide-20040210/food#"
  xmlns:owl ="http://www.w3.org/2002/07/owl#"
  xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xsd ="http://www.w3.org/2001/XMLSchema#">
```

Εικόνα 5: Η ρίζα (πρώτο tag) μιας OWL οντολογίας

Η πρώτη δήλωση προσδιορίζει το εξ'ορισμού (default) namespace το οποίο είναι το namespace της τρέχουσας οντολογίας και υποδηλώνει ότι κάθε όνομα που θα βρίσκεται μέσα στο έγγραφο και δεν έχει πρόθεμα αναφέρεται στην τρέχουσα οντολογία. Η δεύτερη δήλωση ταυτίζει το πρόθεμα vin: με το namespace της τρέχουσας οντολογίας, και παρομοίως η τέταρτη δήλωση το πρόθεμα food: με το namespace μιας άλλης οντολογίας που πρόκειται να χρησιμοποιηθεί. Η τρίτη δήλωση προσδιορίζει το βασικό (base) URI που συνήθως είναι το ίδιο το URI του εγγράφου που περιέχει την οντολογία. Η πέμπτη δήλωση είναι απαραίτητη σε κάθε οντολογία και αποτελεί ένα συμβατικό τρόπο για να δηλωθεί ότι το τρέχον έγγραφο είναι μια OWL οντολογία και ότι μπορεί να χρησιμοποιήσει τις δομές της OWL (οι οποίες ορίζονται στο <http://www.w3.org/2002/07/owl#> και έχουν το πρόθεμα owl:). Οι τρεις τελευταίες δηλώσεις είναι αναγκαίες επίσης καθώς η OWL κάνει χρήση των δομών και των τύπων δεδομένων των RDF, RDF-Schema και XML Schema (οι οποίοι ορίζονται στα <http://www.w3.org/1999/02/22-rdf-syntax-ns#>, <http://www.w3.org/2000/01/rdf-schema#> και <http://www.w3.org/2001/XMLSchema#> και έχουν τα αντίστοιχα προθέματα rdf:, rdfs: και xsd:).

Τα προθέματα που ορίζονται από τις δηλώσεις των namespaces έχουν σημασία μόνο ως τμήμα ενός tag και δεν μπορούν να υποκαταστήσουν το namespace. Συνεπώς η αναφορά μέσα στην τρέχουσα οντολογία στο στοιχείο PotableLiquid που ανήκει στην οντολογία που βρίσκεται στο <http://www.w3.org/TR/2004/REC-owl-guide-20040210/food#> μπορεί να γίνει μόνο μέσω του URI του στοιχείου, δηλαδή <http://www.w3.org/TR/2004/REC-owl-guide-20040210/food#PotableLiquid>. Ως μια βοήθεια για την αποφυγή συνεχούς γραψίματος μακροσκελών URIs, η OWL επιτρέπει τη χρήση ενός συνόλου από ορισμούς οντοτήτων (ENTITY) μέσα σε μια δήλωση τύπων εγγράφου (DOCTYPE) που προηγείται του αρχικού tag rdf:RDF. Οι συντμήσεις που προκύπτουν από τους ορισμούς οντοτήτων υποκαθιστούν τα συνήθως μακροσκελή URIs. Έτσι αν δηλωθούν τα παρακάτω, τότε το στοιχείο <http://www.w3.org/TR/2004/REC-owl-guide-20040210/food#PotableLiquid> μπορεί να γραφεί και ως &food;PotableLiquid.

```
<!DOCTYPE rdf:RDF [
  <!ENTITY vin "http://www.w3.org/TR/2004/REC-owl-guide-20040210/wine#" >
  <!ENTITY food "http://www.w3.org/TR/2004/REC-owl-guide-20040210/food#" > ]>
```

Εικόνα 6: Ορισμοί οντοτήτων στην αρχή της οντολογίας

Με αυτόν τον τρόπο απλοποιούνται και οι δηλώσεις των namespaces στο rdf:RDF tag και έτσι τυχόν αλλαγές στους ορισμούς οντοτήτων μπορούν να διαδοθούν με συνέπεια στην υπόλοιπη οντολογία.

```
<rdf:RDF
```

```
  xmlns    = "&vin;"
```

```
  xmlns:vin = "&vin;"
  xml:base  = "&vin;"
  xmlns:food = "&food;"
  xmlns:owl = "http://www.w3.org/2002/07/owl#"
  xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs = "http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xsd  = "http://www.w3.org/2001/XMLSchema#">
```

Εικόνα 7: Οι αλλαγές στο tag `rdf:RDF` που προκύπτουν μετά τον ορισμό οντοτήτων

Μετά τη δήλωση των namespaces ακολουθεί ένα σύνολο από ισχυρισμούς που αφορούν την οντολογία και βρίσκονται μέσα στο `owl:Ontology` tag.

```
<owl:Ontology rdf:about="">
  <rdfs:comment>An example OWL ontology</rdfs:comment>
  <owl:imports rdf:resource="http://www.w3.org/TR/2004/REC-owl-guide-20040210/food"/>
  <rdfs:label>Wine Ontology</rdfs:label>
</owl:Ontology>
```

Εικόνα 8: Ορισμοί οντοτήτων στην αρχή της οντολογίας

Το attribute `rdf:about` παρέχει ένα όνομα ή μια αναφορά για την οντολογία. Όταν ισούται με το κενό αλφαριθμητικό, που είναι η συνηθέστερη περίπτωση, τότε το όνομα της οντολογίας είναι το βασικό URI που έχει δηλωθεί στο παραπάνω tag που περιέχει τα namespaces. Με τη χρήση των `rdfs:label` και `rdfs:comment` είναι δυνατόν να παρέχονται προαιρετικές ετικέτες και σχόλια για την οντολογία σε γλώσσα κατανοητή από άνθρωπο. Τέλος δίνεται η δυνατότητα μέσω του `owl:imports` tag να εισαχθούν στην τρέχουσα οντολογία οι ισχυρισμοί μιας άλλης οντολογίας. Η ενέργεια αυτή δεν καθίσταται πάντα δυνατή καθώς η πρόσβαση σε διασπαρμένες πηγές στον παγκόσμιο ιστό δεν είναι πάντα εφικτή.

Μετά το κλείσιμο του `owl:Ontology` tag, ακολουθούν οι ορισμοί των βασικών στοιχείων της οντολογίας (τα οποία είναι οι κλάσεις, τα άτομα τους και οι ιδιότητες που αναπαριστούν τις σχέσεις μεταξύ αυτών των ατόμων και τα οποία θα περιγραφούν λεπτομερώς στις επόμενες παραγράφους) και η οντολογία τελειώνει οριστικά με το κλείσιμο του `rdf:RDF` tag (`</rdf:RDF>`).

2.2.7. OWL κλάσεις (Classes)

Οι κλάσεις αποτελούν τις ρίζες διαφόρων δέντρων ταξινόμησης και ορίζουν σύνολα από άτομα που μοιράζονται κάποιες κοινές ιδιότητες και χαρακτηριστικά. Κάθε άτομο στον κόσμο της OWL είναι μέλος της κλάσης `owl:Thing` και συνεπώς κάθε καινούρια ορισμένη

κλάση είναι υποκλάση (subclass) της owl:Thing. Κάθε κλάση αναγνωρίζεται μοναδικά από το rdf:ID της. Οι κλάσεις μπορούν να ιεραρχηθούν με τη χρήση του rdfs:subClassOf το οποίο συσχετίζει μια πιο συγκεκριμένη κλάση με μια πιο γενική κλάση. Μια κλάση μπορεί να σχετίζεται με περισσότερες από μια κλάσεις με τη σχέση rdfs:subClassOf. Με τη χρήση των rdfs:label και rdfs:comment είναι δυνατόν να παρέχονται προαιρετικές ετικέτες και σχόλια για τις κλάσεις σε γλώσσα κατανοητή από άνθρωπο τα οποία δεν επηρεάζουν τη λογική ερμηνεία των κλάσεων. Ένα παράδειγμα ορισμού από OWL κλάσεις φαίνεται στο παρακάτω σχήμα:

```
<owl:Class rdf:ID="Animal"/>
<owl:Class rdf:ID="Person">
  <rdfs:subClassOf rdf:resource="#Animal"/>
  <rdfs:label>Person of real world</rdfs:label>
</owl:Class>
```

Εικόνα 9: OWL ορισμός των κλάσεων "Animal" και "Person"

2.2.8. OWL Άτομα (Individuals)

Τα άτομα αποτελούν τα μέλη των κλάσεων. Το σύνολο των ατόμων που ανήκουν σε μια συγκεκριμένη κλάση ονομάζεται επέκταση κλάσης (class extension). Ένα άτομο μπορεί να μην ανήκει σε καμία κλάση ή να ανήκει σε περισσότερες από μία κλάσεις. Με τη χρήση του rdf:type συνδέεται το άτομο με την κλάση στην οποία ανήκει. Κάθε άτομο αναγνωρίζεται μοναδικά από το rdf:ID του. Τα rdfs:label και rdfs:comment μπορούν να χρησιμοποιηθούν για να ορίσουν ετικέτες και σχόλια για το άτομο. Ένα παράδειγμα από OWL ατόμων φαίνεται στο παρακάτω σχήμα:

```
<owl:Thing rdf:ID="Jim"/>
  <rdf:type rdf:resource="#Person"/>
</owl:Thing>
```

Είναι ισοδύναμο με:
<Person rdf:ID="Jim" />

Εικόνα 10: OWL ορισμός του ατόμου "Jim"

2.2.9. OWL Ιδιότητες (Properties)

Ο κόσμος των κλάσεων και των ατόμων που ανήκουν σε αυτές δεν θα είχε ιδιαίτερο ενδιαφέρον αν οριζόντουσαν μόνο ιεραρχίες μεταξύ των κλάσεων. Οι ιδιότητες (properties) μας επιτρέπουν να ισχυριζόμαστε γενικευμένα γεγονότα (facts) για τις κλάσεις και συγκεκριμένα γεγονότα για τα άτομα των κλάσεων. Οι ιδιότητες είναι δυαδικές σχέσεις και διακρίνονται σε 2 κατηγορίες:

- *Ιδιότητες Τύπων Δεδομένων (Datatype Properties)*: σχέσεις μεταξύ ατόμων κλάσεων και κυριολεκτικών τιμών της RDF ή τύπους δεδομένων της XML Schema.

- *Ιδιότητες Αντικειμένων (Object Properties)*: σχέσεις μεταξύ 2 ατόμων κλάσεων.

Κάθε ιδιότητα αναγνωρίζεται μοναδικά από το `rdf:ID` της. Οι ιδιότητες μπορούν να ιεραρχηθούν με τη χρήση του `rdfs:subPropertyOf` το οποίο συσχετίζει μια πιο συγκεκριμένη ιδιότητα με μια πιο γενική. Μια ιδιότητα μπορεί να σχετίζεται με περισσότερες από μια ιδιότητες με τη σχέση `rdfs:subPropertyOf`. Τα `rdfs:label` και `rdfs:comment` μπορούν να χρησιμοποιηθούν για να ορίσουν ετικέτες και σχόλια για την ιδιότητα. Επιπλέον οι ιδιότητες έχουν τα παρακάτω χαρακτηριστικά:

- *Πεδίο ορισμού (domain)*: Με τη χρήση του `rdfs:domain` ορίζονται οι κλάσεις στον οποίων τα άτομα εφαρμόζεται η κάθε ιδιότητα.
- *Εύρος τιμών (range)*: Με τη χρήση του `rdfs:range` ορίζεται το εύρος τιμών που μπορεί να πάρει κάθε ιδιότητα. Στην περίπτωση των ιδιοτήτων τύπων δεδομένων το εύρος τιμών μπορεί να είναι κάποιος XML Schema τύπος ή συγκεκριμένες κυριολεκτικές τιμές (literals) της RDF. Η OWL, όπως και η RDFS, χρησιμοποιεί τους περισσότερους έτοιμους τύπους δεδομένων της XML Schema και αναφέρεται σε αυτούς μέσω του URI <http://www.w3.org/2001/XMLSchema>. Οι σημαντικότεροι από αυτούς τους τύπους δεδομένων είναι: `xsd:string`, `xsd:normalizedString`, `xsd:boolean`, `xsd:decimal`, `xsd:float`, `xsd:double`, `xsd:integer`, `xsd:long`, `xsd:int`, `xsd:short`, `xsd:byte`, `xsd:nonNegativeInteger`, `xsd:positiveInteger`, `xsd:nonPositiveInteger`, `xsd:negativeInteger`. Ο ορισμός συγκεκριμένων κυριολεκτικών τιμών γίνεται με τη χρήση των `owl:oneOf`, `rdf:List` και `rdf:rest`. Αντίθετα, στην περίπτωση των ιδιοτήτων αντικειμένων το εύρος τιμών δύναται να είναι μία ή περισσότερες (με τη χρήση του `owl:unionOf`) κλάσεις ή συγκεκριμένα άτομα κλάσεων (με τη χρήση του `owl:oneOf`).

Ακολουθούν παραδείγματα ορισμού ιδιοτήτων τύπων δεδομένων και ιδιοτήτων αντικειμένων μαζί με τον ορισμό του πεδίου ορισμού τους και του εύρους τιμών τους.

```
<owl:DatatypeProperty rdf:ID="hasSurname">
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
```

Εικόνα 11: OWL ορισμός της ιδιότητας τύπου δεδομένων "hasSurname" που έχει εύρος τιμών τον "string" τύπο της XML Schema και πεδίο ορισμού τα στιγμιότυπα της κλάσης "Person".

```
<owl:ObjectProperty rdf:ID="hasAnimal">
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:range rdf:resource="#Animal"/>
</owl:ObjectProperty>
```

Εικόνα 12: OWL ορισμός της ιδιότητας αντικειμένων "hasAnimal" που έχει εύρος τιμών τα στιγμιότυπα της κλάσης "Animal" και πεδίο ορισμού τα στιγμιότυπα της κλάσης "Person".

```
<owl:DatatypeProperty rdf:ID="tennisGameScore">
  <rdfs:range>
    <owl:DataRange>
      <owl:oneOf>
        <rdf:List>
          <rdf:first rdf:datatype="&xsd;integer">0</rdf:first>
          <rdf:rest>
            <rdf:List>
              <rdf:first rdf:datatype="&xsd;integer">15</rdf:first>
              <rdf:rest>
                <rdf:List>
                  <rdf:first rdf:datatype="&xsd;integer">30</rdf:first>
                  <rdf:rest>
                    <rdf:List>
                      <rdf:first rdf:datatype="&xsd;integer">40</rdf:first>
                      <rdf:rest rdf:resource="&rdf:nil" />
                    </rdf:List>
                  </rdf:rest>
                </rdf:List>
              </rdf:rest>
            </rdf:List>
          </rdf:rest>
        </rdf:List>
      </owl:oneOf>
    </owl:DataRange>
  </rdfs:range>
</owl:DatatypeProperty>
```

Εικόνα 13: OWL ορισμός της ιδιότητας τύπου δεδομένων "tennisGameScore" που έχει εύρος τιμών τις ακέραιες τιμές 0, 15, 30, 40.

```
<owl:ObjectProperty rdf:ID="physicalCondition">
  <rdfs:domain rdf:resource="#Player"/>
  <rdfs:range>
    <owl:Class>
      <owl:oneOf rdf:parseType="Collection">
        <owl:Thing rdf:about="#InShape"/>
        <owl:Thing rdf:about="#Injured"/>
        <owl:Thing rdf:about="#Tired"/>
      </owl:oneOf>
    </owl:Class>
  </rdfs:range>
</owl:ObjectProperty>
```

Εικόνα 14: OWL ορισμός της ιδιότητας αντικειμένων "physicalCondition" που έχει εύρος τιμών τα άτομα "InShape", "Injured", "Tired" και πεδίο ορισμού τα στιγμιότυπα της κλάσης "Player".

Αφού οριστεί μια ιδιότητα, τότε τα άτομα των κλάσεων οι οποίες περιλαμβάνονται στο πεδίο ορισμού της μπορούν να την περιλαμβάνουν και να έχουν συγκεκριμένη τιμή για αυτήν την ιδιότητα.

```
<Animal rdf:ID="dogJack" />
```

```
<Person rdf:ID="Jim" >
```

```
<hasAnimal rdf:resource="#dogJack" />
<hasSurname rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Jimmy</hasSurname>
</Person>
```

Εικόνα 15: OWL ορισμός του ατόμου "Jim" έχοντας τιμές στις ιδιότητες "hasAnimal" και "hasSurname"

2.2.10. Γνωρίσματα των OWL ιδιοτήτων

Οι ιδιότητες της OWL δύνανται να έχουν και κάποια επιπρόσθετα γνωρίσματα παρέχοντας έναν ισχυρό μηχανισμό για αποτελεσματική εξαγωγή συμπερασμάτων. Τα περισσότερα από αυτά τα γνωρίσματα έχουν εφαρμογή μόνο στις ιδιότητες αντικειμένων. Με τον όρο $P(x,y)$ που θα χρησιμοποιηθεί παρακάτω εννοούμε ότι το άτομο x έχει την τιμή y στην ιδιότητα P .

Μεταβατική Ιδιότητα (Transitive Property)

Όταν μια ιδιότητα P προσδιοριστεί ως μεταβατική, τότε για κάθε άτομο x, y, z ισχύει ότι: αν $P(x,y)$ και $P(y,z)$, τότε υποδηλώνεται ότι $P(x,z)$. Η δήλωση μιας ιδιότητας ως μεταβατικής εφαρμόζεται μόνο σε ιδιότητες αντικειμένων και γίνεται με τη χρήση του `<rdf:type rdf:resource="&owl;TransitiveProperty" />` μέσα στο σώμα ορισμού της ιδιότητας.

Συμμετρική Ιδιότητα (Symmetric Property)

Όταν μια ιδιότητα P προσδιοριστεί ως συμμετρική, τότε για κάθε άτομο x, y ισχύει ότι: $P(x,y)$ αν και μόνο αν $P(y,x)$. Η δήλωση μιας ιδιότητας ως συμμετρικής εφαρμόζεται μόνο σε ιδιότητες αντικειμένων και γίνεται με τη χρήση του `<rdf:type rdf:resource="&owl;SymmetricProperty" />` μέσα στο σώμα ορισμού της ιδιότητας.

Συναρτησιακή Ιδιότητα (Functional Property)

Όταν μια ιδιότητα P προσδιοριστεί ως συναρτησιακή, τότε για κάθε άτομο x, y, z ισχύει ότι: αν $P(x,y)$ και $P(x,z)$ τότε υποδηλώνεται ότι $y=z$. Με άλλα λόγια το άτομο x επιτρέπεται να έχει μία το πολύ τιμή στην ιδιότητα P . Η τιμή αυτή θα είναι είτε ένα άτομο αν η P είναι ιδιότητα αντικειμένων είτε μία κυριολεκτική τιμή αν η P είναι ιδιότητα τύπου δεδομένων. Η δήλωση μιας ιδιότητας ως συναρτησιακής εφαρμόζεται και στα δύο είδη ιδιοτήτων και γίνεται με τη

χρήση του `<rdf:type rdf:resource="&owl;FunctionalProperty" />` μέσα στο σώμα ορισμού της ιδιότητας.

Αντίστροφα Συναρτησιακή Ιδιότητα (InverseFunctional Property)

Όταν μια ιδιότητα P προσδιορίζεται ως αντίστροφα συναρτησιακή, τότε για κάθε άτομο x, y, z ισχύει ότι: αν $P(y,x)$ και $P(z,x)$ τότε υποδηλώνεται ότι $y=z$. Με άλλα λόγια κάθε τιμή της ιδιότητας P σε οποιοδήποτε άτομο είναι μοναδική και κάθε υποκείμενο x έχει διαφορετική τιμή από τα υπόλοιπα άτομα στην ιδιότητα P . Συνεπώς η τιμή της ιδιότητας P μπορεί να λειτουργήσει και ως κλειδί για την ανεύρεση ενός ατόμου. Η δήλωση μιας ιδιότητας στην OWL-DL ως αντίστροφα συναρτησιακής εφαρμόζεται μόνο στις ιδιότητες αντικειμένων (η OWL-Full επιτρέπει τον ορισμό μιας ιδιότητας τύπου δεδομένων ως αντίστροφα συναρτησιακή) και γίνεται με τη χρήση του `<rdf:type rdf:resource="&owl;InverseFunctionalProperty" />` μέσα στο σώμα ορισμού της ιδιότητας.

Αντίστροφη Ιδιότητα (InverseOf Property)

Όταν μια ιδιότητα $P1$ προσδιορίζεται ως αντίστροφη της ιδιότητας $P2$, τότε για κάθε άτομο x, y ισχύει ότι: $P1(x,y)$ αν και μόνο αν $P2(y,x)$. Ο ορισμός σε μια ιδιότητα $P1$ της αντίστροφης της $P2$ και αντίστροφα εφαρμόζεται προφανώς μόνο στις ιδιότητες αντικειμένων και γίνεται με τη χρήση του `<owl:inverseOf rdf:resource="#propertyName" />` μέσα στο σώμα ορισμού της ιδιότητας (αρκεί ο ορισμός στο σώμα της μίας ιδιότητας).

2.2.11. Περιορισμοί Ιδιοτήτων (Property Restrictions)

Πέρα από τον ορισμό του εύρους τιμών μιας ιδιότητας, η OWL παρέχει τη δυνατότητα μέσα σε κάθε κλάση που ανήκει στο πεδίο ορισμού της να οριστούν επιπλέον περιορισμοί με τη χρήση του `<owl:Restriction>` όσον αφορά το εύρος τιμών της ιδιότητας ή το πλήθος των τιμών της που θα έχουν τα υποκείμενα αυτής της κλάσης. Συνεπώς οι περιορισμοί αυτοί δεν ορίζονται μέσα στο σώμα της ιδιότητας αλλά μέσα στο σώμα της κλάσης και γι'αυτό το λόγο απαιτείται η χρήση του `<owl:onProperty rdf:resource="#propertyURI" />` προκειμένου να υποδειχθεί η ιδιότητα στην οποία θα ισχύει ο περιορισμός. Πρέπει να σημειωθεί ότι οι περιορισμοί δηλώνονται μέσα σε μια δομή `<rdfs:subClassOf>` και έτσι θεωρούνται ανώνυμες κλάσεις (anonymous classes).

Περιορισμοί στο εύρος τιμών

Οι περιορισμοί που μπορούν να επιβληθούν μέσα σε μια κλάση όσον αφορά το εύρος τιμών μιας συγκεκριμένης ιδιότητας της είναι οι παρακάτω τρεις:

- *Όλες οι τιμές από (allValuesFrom):* Κάθε άτομο της κλάσης πρέπει στην ιδιότητα αυτή να έχει τιμές που να είναι αποκλειστικά άτομα της κλάσης ή των κλάσεων

(συμπεριλαμβανομένων και των τυχόν υποκλάσεων τους) που ορίζονται στον περιορισμό αυτό. Ο ορισμός του περιορισμού αυτού γίνεται με τη χρήση του `<owl:allValuesFrom rdf:resource="#className" />` μέσα στο σώμα της κλάσης και εφαρμόζεται προφανώς μόνο σε ιδιότητες αντικειμένων.

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

```
<owl:Class rdf:ID="Dog">
```

```
  <rdfs:subClassOf rdf:resource="#Animal"/>
</owl:Class>
<owl:Class rdf:ID="Person">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasAnimal" />
      <owl:allValuesFrom rdf:resource="#Dog" />
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Εικόνα 16: OWL ορισμός του περιορισμού ότι κάθε άτομο της κλάσης "Person" θα έχει στην ιδιότητα "hasAnimal" τιμές που θα είναι όλες άτομα της κλάσης "Dog".

- *Κάποιες τιμές από (someValuesFrom):* Κάθε άτομο της κλάσης πρέπει στην ιδιότητα αυτή να έχει τουλάχιστον μία τιμή που να είναι άτομο της κλάσης ή των κλάσεων (συμπεριλαμβανομένων και των τυχόν υποκλάσεων τους) που ορίζονται στον περιορισμό αυτό. Ο ορισμός του περιορισμού αυτού γίνεται με τη χρήση του `<owl:someValuesFrom rdf:resource="#className" />` μέσα στο σώμα της κλάσης και εφαρμόζεται προφανώς μόνο σε ιδιότητες αντικειμένων.

```
<owl:Class rdf:ID="Animal"/>
<owl:Class rdf:ID="Dog">
  <rdfs:subClassOf rdf:resource="#Animal"/>
</owl:Class>
<owl:Class rdf:ID="Person">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasAnimal" />
      <owl:someValuesFrom rdf:resource="#Dog" />
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Εικόνα 17: OWL ορισμός του περιορισμού ότι κάθε άτομο της κλάσης "Person" θα έχει στην ιδιότητα "hasAnimal" τουλάχιστον μία τιμή που να είναι άτομο της κλάσης "Dog".

- *Έχει τιμή (hasValue):* Κάθε άτομο της κλάσης πρέπει στην ιδιότητα αυτή να έχει μία τιμή που να ισούται με το άτομο που ορίζεται στον περιορισμό αυτό αν πρόκειται για ιδιότητα αντικειμένων ή να ισούται με την κυριολεκτική τιμή που ορίζεται στον περιορισμό αν πρόκειται για ιδιότητα τύπου δεδομένων. Ο ορισμός του περιορισμού αυτού γίνεται με τη

χρήση του `<owl:hasValue rdf:resource="#individualName" />` μέσα στο σώμα της κλάσης στην περίπτωση ιδιότητας αντικειμένων ή του `<owl:hasValue> literalValue` `</owl:hasValue>` στην περίπτωση ιδιότητας τύπου δεδομένων.

```
<owl:Class rdf:ID="Animal"/>
<Animal rdf:ID="dogJack" />
<owl:Class rdf:ID="Person">
```

```
<rdfs:subClassOf>
```

```
<owl:Restriction>
  <owl:onProperty rdf:resource="#hasAnimal" />
  <owl:hasValue rdf:resource="#dogJack" />
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
```

Εικόνα 18: OWL ορισμός του περιορισμού ότι κάθε άτομο της κλάσης "Person" θα έχει στην ιδιότητα "hasAnimal" τιμή το άτομο "dogJack".

Περιορισμοί πολλαπλότητας (Cardinality restrictions)

Οι περιορισμοί που μπορούν να επιβληθούν μέσα σε μια κλάση όσον αφορά την πολλαπλότητα της, δηλαδή το πλήθος των τιμών, μιας συγκεκριμένης ιδιότητας της είναι οι παρακάτω τρεις και εφαρμόζονται και στα δύο είδη ιδιοτήτων:

- *Ακριβές πλήθος (cardinality)*: Κάθε άτομο της κλάσης πρέπει να έχει στην ιδιότητα αυτή τόσες τιμές όσες και το ακριβές πλήθος που ορίζεται από τον περιορισμό. Ο ορισμός του περιορισμού αυτού γίνεται με τη χρήση του `<owl:cardinality>` μέσα στο σώμα της κλάσης.

```
<owl:Class rdf:ID="Person">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasAnimal" />
      <owl:cardinality rdf:datatype="xsd:nonNegativeInteger">1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Εικόνα 19: OWL ορισμός του περιορισμού ότι κάθε άτομο της κλάσης "Person" θα έχει ακριβώς μία τιμή στην ιδιότητα "hasAnimal".

- *Ελάχιστο πλήθος (minCardinality)*: Κάθε άτομο της κλάσης πρέπει να έχει στην ιδιότητα αυτή τουλάχιστον τόσες τιμές όσες και το ελάχιστο πλήθος που ορίζεται από τον περιορισμό. Ο ορισμός του περιορισμού αυτού γίνεται με τη χρήση του `<owl:minCardinality>` μέσα στο σώμα της κλάσης.

```

<owl:Class rdf:ID="Person">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasAnimal" />
      <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

```

Εικόνα 20: OWL ορισμός του περιορισμού ότι κάθε άτομο της κλάσης "Person" θα έχει τουλάχιστον μία τιμή στην ιδιότητα "hasAnimal".

- *Μέγιστο πλήθος (maxCardinality):* Κάθε άτομο της κλάσης πρέπει να έχει στην ιδιότητα αυτή το πολύ τόσες τιμές όσες και το μέγιστο πλήθος που ορίζεται από τον περιορισμό. Ο ορισμός του περιορισμού αυτού γίνεται με τη χρήση του <owl:maxCardinality> μέσα στο σώμα της κλάσης.

```

<owl:Class rdf:ID="Person">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasAnimal" />
      <owl:maxCardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

```

Εικόνα 21: OWL ορισμός του περιορισμού ότι κάθε άτομο της κλάσης "Person" θα έχει το πολύ μία τιμή στην ιδιότητα "hasAnimal".

2.2.12. Αντιστοιχίες OWL στοιχείων (OWL Ontology Mapping)

Για να έχουν μεγαλύτερη απήχηση οι οντολογίες θα πρέπει να είναι ευρέως διαμοιραζόμενες. Επιπλέον, για να μειωθεί η πνευματική προσπάθεια που πρέπει να καταβληθεί για να συγγραφεί μια οντολογία από την αρχή, θα πρέπει οι ήδη γραμμένες οντολογίες να μπορούν να επαναχρησιμοποιηθούν. Προκειμένου λοιπόν ορισμένα OWL στοιχεία που ανήκουν σε παλαιότερες οντολογίες μαζί με τα ήδη υπάρχοντα γνωρίσματα τους να χρησιμοποιηθούν σε μια νέα οντολογία χωρίς την ανάγκη να επανασυγγραφούν από την αρχή, η OWL μέσω κάποιων βασικών της δομών υποστηρίζει την αντιστοίχιση (mapping) αυτών των OWL στοιχείων με ορισμένα νέα στοιχεία που βρίσκονται στην νέα οντολογία. Η αντιστοίχιση αυτή μπορεί να αφορά βέβαια και στοιχεία που συνυπάρχουν στην ίδια οντολογία. Η OWL παρέχει τη δυνατότητα αντιστοιχίσεων και για τα τρία βασικά της στοιχεία (κλάσεις, ιδιότητες και άτομα κλάσεων).

Αντιστοιχήσεις μεταξύ κλάσεων

Για να αντιστοιχηθούν δύο OWL κλάσεις, η OWL χρησιμοποιεί τον όρο ισοδύναμη κλάση (equivalent class) με τη χρήση του `<owl:equivalentClass rdf:resource="#classURI"/>` μέσα στο σώμα ορισμού της κλάσης. Όταν δύο κλάσεις δηλωθούν ισοδύναμες, τότε αυτό υποδεικνύει ότι έχουν ακριβώς τα ίδια άτομα κλάσεων. Με λογικά συμπεράσματα προκύπτει επίσης ότι οι κλάσεις αυτές θα έχουν επιπλέον τις ίδιες ιδότητες και τους ίδιους περιορισμούς στις ιδιότητες αυτές.

```
<owl:Class rdf:ID="Proswpo">
  <owl:equivalentClass rdf:resource="#Person"/>
</owl:Class>
```

Εικόνα 22: OWL ορισμός της κλάσης "Proswpo" που ορίζεται ισοδύναμη της κλάσης "Person" (αν η κλάση "Person" είχε οριστεί σε διαφορετική οντολογία θα έπρεπε να χρησιμοποιηθεί το πλήρες URI της).

Αντιστοιχήσεις μεταξύ ιδιοτήτων

Για να αντιστοιχηθούν δύο OWL ιδιότητες, η OWL χρησιμοποιεί τον όρο ισοδύναμη ιδιότητα (equivalent property) με τη χρήση του `<owl:equivalentProperty rdf:resource="#propertyURI"/>` μέσα στο σώμα ορισμού της ιδιότητας. Όταν δύο ιδιότητες δηλωθούν ισοδύναμες, τότε αυτό υποδεικνύει ότι εφαρμόζονται στα ίδια ακριβώς άτομα κλάσεων. Με λογικά συμπεράσματα προκύπτει επίσης ότι οι ιδιότητες αυτές θα έχουν επιπλέον ίδιο πεδίο ορισμού, ίδιο εύρος τιμών και ίδια γνωρίσματα. Αντιστοιχήσεις προφανώς επιτρέπονται μόνο μεταξύ δύο ιδιοτήτων τύπου δεδομένων ή μεταξύ δύο ιδιοτήτων αντικειμένων και όχι μεταξύ μιας ιδιότητας τύπου δεδομένων και μιας ιδιότητας αντικειμένων.

```
<owl:ObjectProperty rdf:ID="exeiZwo">
  <owl:equivalentProperty rdf:resource="#hasAnimal"/>
</owl:ObjectProperty>
```

Εικόνα 23: OWL ορισμός της ιδιότητας αντικειμένων "exeiZwo" που ορίζεται ισοδύναμη της ιδιότητας "hasAnimal" (αν η ιδιότητα "hasAnimal" είχε οριστεί σε διαφορετική οντολογία θα έπρεπε να χρησιμοποιηθεί το πλήρες URI της).

Αντιστοιχήσεις μεταξύ ατόμων κλάσεων

Για την αντιστοίχιση των ατόμων κλάσεων, η OWL χρησιμοποιεί δύο όρους:

- *Την ακριβή ομοιότητα μεταξύ ατόμων (identity between individuals):* Δηλώνεται με τη χρήση του `<owl:sameAs rdf:resource="#individualURI"/>` μέσα στο σώμα ορισμού του ατόμου. Όταν δύο ή περισσότερα υποκείμενα κλάσεων δηλωθούν ακριβώς όμοια, τότε αυτό υποδεικνύει ότι ουσιαστικά πρόκειται για το ίδιο άτομο με διαφορετικά ονόματα. Με λογικά συμπεράσματα προκύπτει επίσης ότι τα άτομα αυτά θα έχουν ίδιο τύπο και ίδιες τιμές στις ιδιότητες τους.

```
<Person rdf:ID="Panos" />
<Person rdf:ID="Panagiotis" />
  <owl:sameAs rdf:resource="#Panos"/>
</Person>
```

Εικόνα 24: OWL ορισμός του ατόμου "Panagiotis" της κλάσης "Person" που ορίζεται ακριβώς όμοιο του ατόμου "Panos" της κλάσης "Person" (αν το άτομο "Panos" είχε οριστεί σε διαφορετική οντολογία θα έπρεπε να χρησιμοποιηθεί το πλήρες URI του).

- Την *διαφορετικότητα μεταξύ ατόμων* (*different individuals*): Αυτός ο μηχανισμός είναι ο αντίθετος της ακριβής ομοιότητας και δηλώνεται με τη χρήση του `<owl:differentFrom rdf:resource="#individualURI"/>` μέσα στο σώμα ορισμού του ατόμου ή με τη χρήση των δομών `<owl:AllDifferent>` και `<owl:distinctMembers rdf:parseType="Collection">` στο σώμα της οντολογίας. Όταν δύο ή περισσότερα υποκείμενα κλάσεων δηλωθούν διαφορετικά, τότε αυτό υποδεικνύει ότι ουσιαστικά πρόκειται για αποκλειστικά διαφορετικά άτομα τα οποία με κανένα λογικό συμπέρασμα δεν μπορούν να θεωρηθούν ίδια.

```
<Person rdf:ID="Panos" />
<Person rdf:ID="Kwstas" />
<Person rdf:ID="Panagiotis" />
  <owl:differentFrom rdf:resource="#Panos"/>
  <owl:differentFrom rdf:resource="#Kwstas"/>
</Person>
```

Εικόνα 25: OWL ορισμός του ατόμου "Panagiotis" της κλάσης "Person" που ορίζεται διαφορετικό των ατόμων "Panos" και "Kwstas" της κλάσης "Person" (αν τα άτομα "Panos" και "Kwstas" είχαν οριστεί σε διαφορετικές οντολογίες θα έπρεπε να χρησιμοποιηθούν τα πλήρη URI τους).

```
<Person rdf:ID="Panos" />
<Person rdf:ID="Kwstas" />
<Person rdf:ID="Panagiotis" />
<owl:AllDifferent>
  <owl:distinctMembers rdf:parseType="Collection">
    <Person rdf:about="#Panos" />
    <Person rdf:about="#Kwstas" />
    <Person rdf:about="#Panagiotis" />
  </owl:distinctMembers>
</owl:AllDifferent>
```

Εικόνα 26: OWL ορισμός του ισχυρισμού ότι τα άτομα "Panagiotis", "Panos" και "Kwstas" της κλάσης "Person" είναι αμοιβαία διαφορετικά μεταξύ τους.

Η διαφορά στα δύο παραπάνω σχήματα είναι ότι στο πρώτο σχήμα το άτομο "Panagiotis" δηλώνεται διαφορετικό από τα άτομα "Panos" και "Kwstas", χωρίς να γνωρίζουμε αν τα δύο τελευταία άτομα είναι διαφορετικά μεταξύ τους. Αντίθετα στο δεύτερο σχήμα δηλώνεται ρητά ότι και τα άτομα "Panos" και "Kwstas" είναι διαφορετικά μεταξύ τους.

2.2.13. Σύνθετες κλάσεις (Complex Classes)

Η OWL παρέχει επιπρόσθετους κατασκευαστές (constructors) με τους οποίους μπορεί να σχηματίσει κλάσεις. Αυτοί οι κατασκευαστές μπορούν να χρησιμοποιηθούν για να δημιουργήσουν τις λεγόμενες εκφράσεις κλάσεων (class expressions). Η OWL υποστηρίζει τις βασικές λειτουργίες συνόλων όπως είναι η ένωση (union), η τομή (intersection) και το συμπλήρωμα (complement). Επιπρόσθετα, κλάσεις μπορούν να οριστούν κατηγορηματικά (explicitly) απαριθμώντας τα άτομα τους. Τέλος, παρέχει τη δυνατότητα ισχυρισμού ότι δύο επεκτάσεις κλάσεων είναι ξένες (disjoint) μεταξύ τους.

Πρέπει να σημειωθεί ότι οι εκφράσεις κλάσεων μπορεί να είναι εμφωλευμένες (nested) χωρίς την απαίτηση για ύπαρξη ονομάτων για κάθε ενδιάμεση κλάση. Αυτό επιτρέπει τη δημιουργία μέσω των λειτουργιών συνόλου σύνθετων κλάσεων που αποτελούνται από επώνυμες (named) ή/και ανώνυμες (anonymous) κλάσεις που ουσιαστικά είναι περιορισμοί ιδιοτήτων.

Τελεστές συνόλων (Set operators)

Οι τελεστές συνόλων που υποστηρίζει η OWL (τομή, ένωση και συμπλήρωμα) χρησιμοποιούνται για το χειρισμό των επεκτάσεων κλάσεων.

Η δήλωση μιας σύνθετης κλάσης ως τομή δύο ή περισσότερων επεκτάσεων κλάσης γίνεται με τη χρήση του `<owl:intersectionOf rdf:parseType="Collection">` μέσα στο σώμα ορισμού της σύνθετης κλάσης. Όταν μια κλάση δηλωθεί ως σύνθετη και προκύπτει από τομή άλλων κλάσεων, τότε η επέκταση κλάσης της θα περιλαμβάνει ακριβώς και μόνο την τομή των επεκτάσεων κλάσης των περιλαμβανομένων στην τομή κλάσεων.

```
<owl:Class rdf:ID="PersonWithAtLeastOneAnimal">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#Person" />
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasAnimal" />
      <owl:minCardinality rdf:datatype="xsd:nonNegativeInteger">1</owl:cardinality>
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>
```

Εικόνα 27: OWL ορισμός της σύνθετης κλάσης "PersonWithAtLeastOneAnimal" της οποίας η επέκταση κλάσης προκύπτει από την τομή των ατόμων της κλάσης "Person" και των ατόμων που έχουν μία τουλάχιστον τιμή στην ιδιότητα αντικειμένων "has Animal".

Η δήλωση μιας σύνθετης κλάσης ως ένωση δύο ή περισσότερων επεκτάσεων κλάσης γίνεται με τη χρήση του `<owl:unionOf rdf:parseType="Collection">` μέσα στο σώμα ορισμού της σύνθετης κλάσης. Όταν μια κλάση δηλωθεί ως σύνθετη και προκύπτει από ένωση άλλων

κλάσεων, τότε η επέκταση κλάσης της θα περιλαμβάνει ακριβώς και μόνο την ένωση των επεκτάσεων κλάσης των περιλαμβανομένων στην ένωση κλάσεων.

```
<owl:Class rdf:ID="PersonWithNoAnimal">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#Person" />
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasAnimal" />
      <owl:cardinality rdf:datatype="xsd:nonNegativeInteger">0</owl:cardinality>
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>
<owl:Class rdf:ID="AllPerson">
  <owl:unionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#PersonWithNoAnimal" />
    <owl:Class rdf:about="#PersonWithAtLeastOneAnimal" />
  </owl:unionOf>
</owl:Class>
```

Εικόνα 28: OWL ορισμός της σύνθετης κλάσης "PersonWithNoAnimal" της οποίας η επέκταση κλάσης προκύπτει από την τομή των ατόμων της κλάσης "Person" και των ατόμων που δεν έχουν καμία τιμή στην ιδιότητα αντικειμένων "hasAnimal" και της σύνθετης κλάσης "AllPerson" της οποίας η επέκταση κλάσης προκύπτει από την ένωση των επεκτάσεων κλάσης των κλάσεων "PersonWithNoAnimal" και "PersonWithAtLeastOneAnimal".

Η δήλωση μιας σύνθετης κλάσης ως συμπλήρωμα μιας επέκτασης κλάσης γίνεται με τη χρήση του `<owl:complementOf rdf:resource="#classURI">` μέσα στο σώμα ορισμού της σύνθετης κλάσης. Όταν μια κλάση A δηλωθεί ως σύνθετη και αποτελεί το συμπλήρωμα μιας άλλης B, τότε η επέκταση κλάσης της θα περιλαμβάνει ακριβώς και μόνο όλα τα άτομα που δεν ανήκουν στην επέκταση κλάσης της B. Με ακριβείς όρους είναι η διαφορά συνόλων μεταξύ της επέκτασης κλάσης του `<owl:Thing>` και της επέκτασης κλάσης της κλάσης B.

```
<owl:Class rdf:ID="NoPerson">
  <owl:complementOf rdf:resource="#Person">
</owl:Class>
```

Εικόνα 29: OWL ορισμός της σύνθετης κλάσης "NoPerson" της οποίας η επέκταση κλάσης περιλαμβάνει όλα τα άτομα που δεν ανήκουν στην επέκταση κλάσης της κλάσης "Person".

Απαριθμούμενες κλάσεις (Enumerated Classes)

Η OWL παρέχει τα μέσα για τον ορισμό μιας κλάσης μέσα από την απευθείας απαρίθμηση (enumeration) των ατόμων της με τη χρήση του `<owl:oneOf rdf:parseType="Collection">` μέσα στο σώμα ορισμού της σύνθετης κλάσης. Αυτός ο ορισμός προσδιορίζει αποκλειστικά την επέκταση κλάσης της σύνθετης κλάσης και δεν επιτρέπει σε άλλα άτομα να δηλωθούν μέλη της κλάσης.

```
<Person rdf:ID="Joe" />
<Person rdf:ID="Sam" />
<Person rdf:ID="Nick" />
```

```
<owl:Class rdf:ID="JimFriends">
```

```
<owl:oneOf rdf:parseType="Collection">
  <Person rdf:about="#Joe" />
  <Person rdf:about="#Sam" />
  <Person rdf:about="#Nick" />
</owl:oneOf>
</owl:Class>
```

Εικόνα 30: OWL ορισμός της σύνθετης κλάσης "JimFriends" της οποίας η επέκταση κλάσης περιλαμβάνει μόνο τα άτομα "Joe", "Sam" και "Nick" της κλάσης "Person".

Ξένες κλάσεις (Disjoint Classes)

Η OWL παρέχει τα μέσα για τον ορισμό μιας κλάσης ως ξένη προς μία ή περισσότερες άλλες κλάσεις με τη χρήση του `<owl:disjointWith rdf:resource="#classURI"/>` μέσα στο σώμα ορισμού της κλάσης. Όταν μια κλάση A δηλωθεί ξένη ως προς μια κλάση B, τότε η επέκταση κλάσης της A δεν έχει κανένα κοινό μέλος με την επέκταση κλάσης της B και συνεπώς ένα άτομο που είναι στιγμιότυπο της κλάσης A δεν μπορεί να είναι ταυτόχρονα και στιγμιότυπο της κλάσης B.

```
<owl:Class rdf:ID="Dog">
  <rdfs:subClassOf rdf:resource="#Animal"/>
</owl:Class>
<owl:Class rdf:ID="Cat">
  <rdfs:subClassOf rdf:resource="#Animal"/>
</owl:Class>
<owl:Class rdf:ID="Bird">
  <rdfs:subClassOf rdf:resource="#Animal"/>
  <owl:disjointWith rdf:resource="#Cat"/>
  <owl:disjointWith rdf:resource="#Dog"/>
</owl:Class>
```

Εικόνα 31: OWL ορισμός της κλάσης "Bird" η οποία δηλώνεται ξένη ως προς τις κλάσεις "Dog" και "Cat".

Πρέπει να σημειωθεί ότι στο παραπάνω παράδειγμα δηλώνεται μόνο ότι η κλάση "Bird" είναι ξένη ως προς τις κλάσεις "Dog" και "Cat" και δεν ισχύει ο ισχυρισμός ότι οι κλάσεις "Dog" και "Cat" είναι ξένες μεταξύ τους. Για να ισχύσει αυτός ο ισχυρισμός θα πρέπει να δηλωθεί ρητά στο σώμα μιας τουλάχιστον από τις δύο ότι είναι ξένες και μεταξύ τους.

2.3. MPEG-7

2.3.1. Εισαγωγή

Το διεθνές ISO/IEC πρότυπο MPEG-7 (ISO/IEC 15938), που παρουσιάζεται σ' αυτή την υποενότητα, έχει τον επίσημο τίτλο "Διεπαφή Περιγραφής Περιεχομένου Πολυμέσων" ("Multimedia Content Description Interface") [11] [12] [13], έχει αναπτυχθεί από την Ομάδα Ειδημόνων σε Κινούμενες Εικόνες (Moving Picture Experts Group – MPEG) και έχει εκφραστεί στη γλώσσα της XML-Schema. Το MPEG-7 παρέχει ένα εκτεταμένο σύνολο Εργαλείων Περιγραφής (Description Tools) για την καθοδήγηση της δημιουργίας περιγραφών (descriptions) οπτικοακουστικού υλικού, που αποτελούν τη βάση για εφαρμογές που παρέχουν την απαραίτητη αποτελεσματική και αποδοτική πρόσβαση σε οπτικοακουστικό υλικό. Οι MPEG-7 περιγραφές οπτικοακουστικού υλικού μπορεί να δημιουργούνται είτε χειροκίνητα είτε αυτόματα και, μετά την αποθήκευσή τους, να προσπελαύνονται από διαφορετικές εφαρμογές όπως ερωτήσεις (queries), πλοήγηση (browsing) ή φιλτράρισμα (filtering).

2.3.2. Τα βασικά στοιχεία του MPEG-7

Τα κυριότερα στοιχεία του προτύπου MPEG-7 είναι τα Εργαλεία Περιγραφής, η Γλώσσα Ορισμού Περιγραφών (Description Definition Language – DDL) και τα Εργαλεία Συστήματος (System Tools).

Τα Εργαλεία Περιγραφής καθοδηγούν τη διαδικασία περιγραφής οπτικοακουστικού υλικού. Τα MPEG-7 εργαλεία περιγραφής περιλαμβάνουν τους Τύπους Δεδομένων (Datatypes), τους Περιγραφείς (Descriptors – D), τα Σχήματα Περιγραφής (Description Schemes – DS) και τα Σχήματα Κατηγοριοποίησης (Classification Schemes – CS).

Οι Τύποι Δεδομένων δεν αναφέρονται αποκλειστικά στην περιοχή των πολυμέσων και είναι επί της ουσίας επαναχρησιμοποιήσιμοι βασικοί τύποι ή δομές που χρησιμοποιούνται από περιγραφείς και περιγραφές σχημάτων. Ένα παράδειγμα MPEG-7 τύπου δεδομένων φαίνεται στην παρακάτω εικόνα, όπου ορίζεται ο τύπος "zeroToOneType", που αναπαριστά πραγματικούς αριθμούς μεταξύ 0 και 1.

```
<simpleType name="zeroToOneType">
  <restriction base="float">
    <minInclusive value="0.0"/>
    <maxInclusive value="1.0"/>
  </restriction>
</simpleType>
```

Εικόνα 32: Ορισμός του MPEG-7 Τύπου Δεδομένων "zeroToOneType", που αναπαριστά πραγματικούς αριθμούς μεταξύ 0 και 1.

Οι Περιγραφείς αναπαριστούν χαρακτηριστικά πολυμέσων και ορίζουν τη σύνταξη και τη σημασία (semantics) της αναπαράστασης κάθε χαρακτηριστικού. Τα χαρακτηριστικά είναι εμφανείς ιδιότητες που χαρακτηρίζουν τα δεδομένα και σημαίνουν κάτι συγκεκριμένο για κάποιον. Έτσι, είναι δυνατό να χρησιμοποιηθούν πολλοί περιγραφείς για το ίδιο χαρακτηριστικό. Ο γενικός ορισμός ενός περιγραφέα, που αποτελεί τη βάση για όλους τους περιγραφείς, φαίνεται στην επόμενη εικόνα.

```
<complexType name="DType" abstract="true">
  <complexContent>
    <extension base="mpeg7:Mpeg7BaseType"/>
  </complexContent>
</complexType>
```

Εικόνα 33: Γενικός ορισμός περιγραφέα.

Τα Σχήματα Περιγραφής παρέχουν περιγραφική πληροφορία και καθορίζουν τη δομή και τη σημασία των σχέσεων που υπάρχουν μεταξύ των συστατικών τους, που μπορεί να είναι τόσο περιγραφείς όσο και σχήματα περιγραφής. Ο γενικός ορισμός ενός σχήματος περιγραφής, που αποτελεί τη βάση για όλα τα σχήματα περιγραφής, φαίνεται παρακάτω:

```
<complexType name="DSType" abstract="true">
  <complexContent>
    <extension base="mpeg7:Mpeg7BaseType">
      <sequence>
        <element name="Header" type="mpeg7:HeaderType" minOccurs="0" maxOccurs
="unbounded"/>
      </sequence>
      <attribute name="id" type="ID" use="optional"/>
      <attributeGroup ref="mpeg7:timePropertyGrp"/>
      <attributeGroup ref="mpeg7:mediaTimePropertyGrp"/>
    </extension>
  </complexContent>
</complexType>
```

Εικόνα 34: Γενικός ορισμός Σχήματος Περιγραφής.

Τα Σχήματα Κατηγοριοποίησης επιτρέπουν την κατηγοριοποίηση (classification) μιας θεματικής περιοχής με ένα σύνολο ιεραρχικά οργανωμένων όρων (terms). Ο γενικός ορισμός ενός σχήματος κατηγοριοποίησης, που αποτελεί τη βάση για όλα τα σχήματα κατηγοριοποίησης, φαίνεται στην εικόνα που ακολουθεί.

```
<complexType name="ClassificationSchemeType">
  <complexContent>
    <extension base="mpeg7:ClassificationSchemeBaseType">
      <sequence>
        <element name="Term" type="mpeg7:TermDefinitionType" maxOccurs=
"unbounded"/>
      </sequence>
    </extension>
  </complexContent>
```

</complexContent>
</complexType>

Εικόνα 35: Γενικός ορισμός Σχήματος Κατηγοριοποίησης.

Η Γλώσσα Ορισμού Περιγραφών ορίζει τη σύνταξη των MPEG-7 εργαλείων περιγραφής και επιτρέπει τον ορισμό νέων περιγραφέων και σχημάτων περιγραφής. Επιπλέον, η γλώσσα ορισμού περιγραφών επιτρέπει την επέκταση και την τροποποίηση υπαρχόντων σχημάτων περιγραφής.

Τα Εργαλεία Συστήματος υποστηρίζουν μια δυαδική κωδικοποίηση για την αποδοτική αποθήκευση και μετάδοση των MPEG-7 περιγραφών, παρέχουν μηχανισμούς μετάδοσης περιγραφών (σε δυαδική μορφή ή σε μορφή κειμένου), πολύπλεξη (multiplexing) περιγραφών, συγχρονισμό (synchronization) και περιεχομένου, διαχείριση και προστασία πνευματικών δικαιωμάτων MPEG-7 περιγραφών κ.α.

2.3.3. Τα τμήματα του MPEG-7

Το πρότυπο MPEG-7 αποτελείται από τα εξής 8 τμήματα:

1. MPEG-7 Systems [14], που αποτελείται από τη δυαδική μορφή MPEG-7 κωδικοποίησης και την τελική αρχιτεκτονική (terminal architecture).
2. MPEG-7 DDL (Description Definition Language) [15], που είναι η γλώσσα ορισμού περιγραφών και χρησιμοποιείται για τον ορισμό και την πιθανή επέκταση των MPEG-7 εργαλείων περιγραφής. Η MPEG-7 DDL βασίζεται στη γλώσσα XML Schema (βλέπε υποενότητα 2.1.1). Καθώς η XML Schema δεν έχει σχεδιαστεί αποκλειστικά για την περιγραφή οπτικοακουστικού υλικού, έχουν γίνει συγκεκριμένες επεκτάσεις με τη μορφή τύπων. Έτσι, εκτός από τα δομικά στοιχεία και τους τύπους δεδομένων της XML Schema, η MPEG-7 DDL διαθέτει ένα σύνολο επεκτάσεων που καλύπτουν τις απαιτήσεις της περιοχής των πολυμέσων.
3. MPEG-7 Visual [16], που περιλαμβάνει (βασικά και σύνθετα) εργαλεία περιγραφής οπτικής (visual) πληροφορίας, επιτρέποντας έτσι τη χρήση του ευρέως αποδεκτού Μοντέλου Ανυσματικού Χώρου (Vector Space Model – VSM) για ανάκτηση (retrieval) βάσει οπτικού περιεχομένου. Τα MPEG-7 εργαλεία οπτικής περιγραφής καλύπτουν τα βασικά οπτικά χαρακτηριστικά: Χρώμα (Color), Υφή (Texture), Σχήμα (Shape), Κίνηση (Motion), Εντοπισμός (Localization) και Αναγνώριση Προσώπων (Face Recognition).
4. MPEG-7 Audio [17], που παρέχει εργαλεία περιγραφής ηχητικής (audio) πληροφορίας. Το τμήμα MPEG-7 Audio, μαζί με το τμήμα MPEG-7 MDS, παρέχουν δομές για την περιγραφή ηχητικού περιεχομένου. Με τη χρήση αυτών των δομών μπορούν να οριστούν τόσο χαμηλού επιπέδου (low-level) περιγραφείς για ηχητικά χαρακτηριστικά, όπως είναι τα

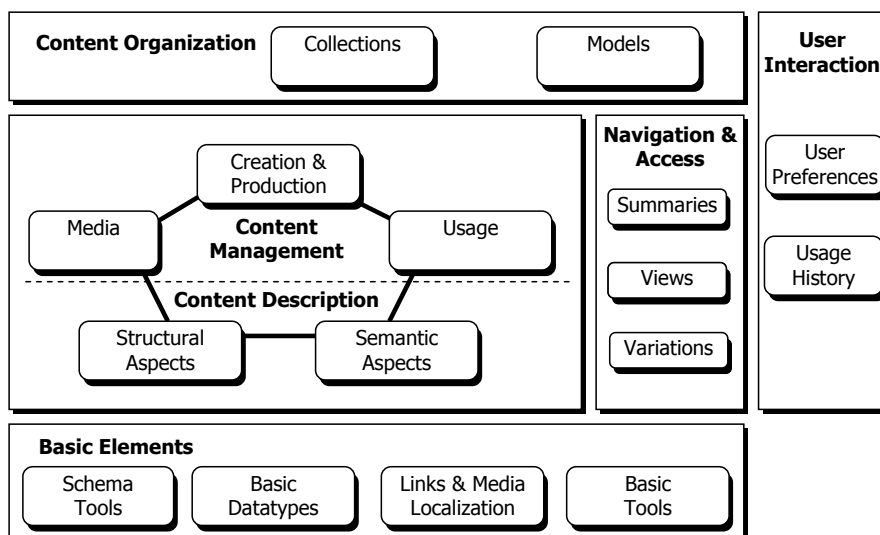
φασματικά (spectral), παραμετρικά (parametric), και χρονικά (temporal) χαρακτηριστικά του σήματος, που μπορούν να χρησιμοποιηθούν σε πολλές εφαρμογές, όσο και υψηλού επιπέδου (high-level) εργαλεία περιγραφής που απευθύνονται σε συγκεκριμένες εφαρμογές. Τα υψηλού επιπέδου εργαλεία περιγραφής περιλαμβάνουν γενικά εργαλεία περιγραφής για αναγνώριση (recognition) και δεικτοδότηση (indexing), εργαλεία περιγραφής χροιάς οργάνων (instrumental timbre), εργαλεία περιγραφής ομιλίας (spoken content), ένα σχήμα περιγραφής για ηχητικές υπογραφές (audio signatures) και εργαλεία περιγραφής μελωδιών (melodic description tools) για την υποστήριξη ερωτήσεων μέσω σφυρίγματος (query-by-humming).

5. MPEG-7 MDS (Multimedia Description Schemes) [18], που περιλαμβάνει το σύνολο των εργαλείων περιγραφής για γενικά χαρακτηριστικά και περιγραφές πολυμέσων. Γενικά χαρακτηριστικά, όπως το άνωσμα (vector), ο χρόνος (time), εργαλεία περιγραφής με χρήση κειμένου (textual description tools), ελεγχόμενα λεξικά (controlled vocabularies) κ.α., χρησιμοποιούνται τόσο για περιγραφές ηχητικού περιεχομένου όσο και για περιγραφές οπτικού περιεχομένου, και είναι έτσι γενικά για όλα τα μέσα. Σύνθετα εργαλεία περιγραφής χρησιμοποιούνται όταν πρέπει να περιγραφούν πάνω από ένα μέσα.
6. MPEG-7 Reference Software, που αποτελείται από μια υλοποίηση (implementation) των τμημάτων του MPEG-7 που είναι σε πρότυπη κατάσταση (normative status), το Μοντέλο Πειραματισμού (eXperimentation Model – XM). Το λογισμικό (software) είναι η πλατφόρμα προσομοίωσης (simulation) για MPEG-7 περιγραφείς, σχήματα περιγραφής, σχήματα κατηγοριοποίησης και τη γλώσσα ορισμού περιγραφών.
7. MPEG-7 Conformance Testing, που παρέχει κατευθυντήριες γραμμές (guidelines) και διαδικασίες (procedures) για τον έλεγχο συμβατότητας (conformance) των MPEG-7 υλοποιήσεων.
8. MPEG-7 Extraction and Use of Descriptions, που αποτελείται από πληροφοριακό υλικό, σε μορφή τεχνικής αναφοράς (technical report), για την εξαγωγή και χρήση κάποιων εργαλείων περιγραφής, που παρέχουν επιπλέον ενόραση (insight) στην υλοποίηση του MPEG-7 Reference Software και σε εναλλακτικές προσεγγίσεις.

Στη συνέχεια θα εστιάσουμε στο τμήμα MPEG-7 MDS που αποτελεί τη βάση για τη σημασιολογική περιγραφή οπτικό-ακουστικού υλικού που υλοποιείται μέσα στο GraphOnto.

2.3.4. MPEG-7 MDS

Το τμήμα MPEG-7 Multimedia Description Schemes (MDS) [18] παρέχει όλες τις δομές (constructs) που απαιτούνται για τον ορισμό μεταδεδομένων περιγραφής οπτικοακουστικού υλικού και υπηρεσιών (services) που παρέχονται πάνω στο υλικό αυτό. Μια επισκόπηση (overview) των βασικών δομικών στοιχείων, τα οποία θα περιγραφούν στη συνέχεια αναλυτικά, και της δομής του MPEG-7 MDS φαίνεται στην παρακάτω εικόνα.



Εικόνα 36: Επισκόπηση του MPEG-7 MDS.

Το τμήμα MPEG-7 MDS απαρτίζεται από τα εξής βασικά συστατικά (components):

Βασικά στοιχεία (Basic Elements)

Τα Βασικά Στοιχεία χρησιμοποιούνται παντού στο τμήμα MPEG-7 MDS ως θεμελιώδεις έννοιες (fundamental concepts). Τα Βασικά Στοιχεία του MPEG-7 MDS περιλαμβάνουν τα Εργαλεία Σχήματος (Schema Tools), τους Βασικούς Τύπους Δεδομένων (Basic Datatypes), τα Εργαλεία Διασύνδεσης και Εντοπισμού Μέσων (Linking and Media Localization Tools) και τα Βασικά Εργαλεία (Basic Tools).

Τα Εργαλεία Σχήματος χρησιμεύουν κατά το σχηματισμό (formation), το πακετάρισμα (packaging), και τον υπομνηματισμό (annotation) των MPEG-7 περιγραφών. Περιλαμβάνουν το Στοιχείο Ρίζα (Root Element), το Κορυφαίο Στοιχείο (Top-Level Element) και τα Πακέτα (Packages). Το Στοιχείο Ρίζα ορίζει τα όρια μιας MPEG-7 περιγραφής και δηλώνει αν αυτή είναι ή όχι πλήρης. Αν η MPEG-7 περιγραφή είναι πλήρης, το στοιχείο-ρίζα ακολουθείται από ένα κορυφαίο στοιχείο που προσανατολίζει την περιγραφή σε μια συγκεκριμένη άποψη (aspect) περιγραφής. Τα πακέτα ορίζονται βάσει του σχήματος περιγραφής Package DS, που περιγράφει την οργάνωση, από το χρήστη, MPEG-7 περιγραφών και σχημάτων περιγραφής σε ένα πακέτο που επιτρέπει την οργανωμένη επικοινωνία αυτών των MPEG-7 εργαλείων με εφαρμογές και χρήστες. Επιπλέον, το σχήμα περιγραφής DescriptionMetadata DS παρέχει μεταδεδομένα για τις περιγραφές, όπως ο χρόνος και το εργαλείο δημιουργίας ή η έκδοση.

Οι Βασικοί Τύποι Δεδομένων παρέχουν ένα εκτεταμένο σύνολο τύπων δεδομένων όπως πεδία (ranges) ακεραίων και πραγματικών αριθμών και μαθηματικές δομές όπως ανύσματα και πίνακες, που απαιτούνται για την περιγραφή οπτικοακουστικού υλικού.

Τα Εργαλεία Διασύνδεσης και Εντοπισμού Μέσων περιλαμβάνουν δομές για τη διασύνδεση αρχείων μέσων και τον εντοπισμό τμημάτων υλικού, ενώ τα Βασικά Εργαλεία παρέχουν δομές για την περιγραφή χρονικής πληροφορίας και πληροφορίας σε μορφή

κειμένου. Τα σχήματα περιγραφής χρονικής πληροφορίας βασίζονται στο πρότυπο ISO 8601 [ISO/IEC98], το οποίο έχει υιοθετηθεί και από την XML Schema. Τα σχήματα περιγραφής Time DS και MediaTime DS περιγράφουν χρονική πληροφορία για τον πραγματικό κόσμο και για τα μέσα αντίστοιχα.

Το MPEG-7 παρέχει ένα σύνολο βασικών δομών για τη δημιουργία υπομνημάτων (annotations) σε μορφή κειμένου (Textual Annotations), στο σχήμα περιγραφής TextAnnotation DS: (α) Το σχήμα περιγραφής FreeTextAnnotation DS αναπαριστά υπομνήματα σε μορφή ελεύθερου κειμένου (free text), που είναι τυχαίες συμβολοσειρές που μπορεί να διαθέτουν πληροφορία για τη χρησιμοποιούμενη γλώσσα; (β) Το σχήμα περιγραφής StructuredTextAnnotation DS αναπαριστά δομημένα υπομνήματα, που διαθέτουν πεδία που αντιστοιχούν στις ερωτήσεις «Ποιος;» ("Who?"), «Ποιο αντικείμενο;» ("What object?"), «Ποια ενέργεια;» ("What action?"), «Που;» ("Where?"), «Πότε;» ("When?"), «Γιατί;» ("Why?") και «Πως;» ("How?"); (γ) Το σχήμα περιγραφής DependencyStructure DS αναπαριστά υπομνήματα που περιγράφουν ρητά τη συντακτική εξάρτηση ανάμεσα σε γραμματικά στοιχεία που σχηματίζουν προτάσεις; και (δ) Το σχήμα περιγραφής KeywordAnnotation DS αναπαριστά υπομνήματα που περιέχουν λέξεις-κλειδιά (keywords) και ελεγχόμενους όρους (controlled terms) που περιέχονται σε σχήματα κατηγοριοποίησης.

Στοιχεία Περιγραφής και Διαχείρισης Περιεχομένου (Content Description and Management Elements)

Τα Στοιχεία Περιγραφής και Διαχείρισης Περιεχομένου χρησιμοποιούνται για την περιγραφή περιεχομένου πολυμέσων από διάφορες απόψεις (viewpoints) όπως ο τρόπος χρήσης, η δημιουργία και η παραγωγή, τα χαρακτηριστικά των μέσων (media), η δομική και η σημασιολογική άποψη.

Τα Στοιχεία Περιγραφής Περιεχομένου του MPEG-7 MDS είναι ο μηχανισμός περιγραφής της δομής (structure) και της σημασίας του οπτικοακουστικού υλικού. Τα Δομικά Εργαλεία (Structural Tools) περιγράφουν τη δομή του οπτικοακουστικού υλικού με τη χρήση τμημάτων (segments) video και ήχου, πλαισίων (frames), ακίνητων (still) και κινούμενων (moving) περιοχών (regions). Τα Σημασιολογικά Εργαλεία (Semantic Tools) περιγράφουν τις σημασιολογικές οντότητες (semantic entities) του πραγματικού κόσμου που περιέχονται στο οπτικοακουστικό υλικό.

Τα Δομικά Εργαλεία είναι οργανωμένα γύρω από το σχήμα περιγραφής Segment DS, που αναπαριστά τη χωρική (spatial), χρονική και χωροχρονική (spatiotemporal) δομή του οπτικοακουστικού υλικού. Το σχήμα περιγραφής Segment DS μπορεί να οργανωθεί τόσο με ιεραρχική (hierarchical) δομή όσο και με δομή γράφου (graph), επιτρέποντας έτσι την αντίστοιχη δόμηση του οπτικοακουστικού υλικού. Τα τμήματα μπορούν να περιγραφούν

περεταίρω με τη χρήση υπομνημάτων κειμένου, σημασιολογικών περιγραφών και αισθητών (perceptual) χαρακτηριστικών (όπως χρώμα, σχήμα, υφή κ.α.).

Τα Σημασιολογικά Εργαλεία επιτρέπουν την περιγραφή της σημασίας του οπτικοακουστικού περιεχομένου. Έτσι, υποστηρίζεται ο ορισμός αφηρημένων (abstract) και απτών (concrete) σημασιολογικών οντοτήτων που μπορεί να είναι Γεγονότα (Events), Αντικείμενα, Έννοιες, Τόποι (Places), Χρονικές Έννοιες και Παράγοντες (Agents) που περιλαμβάνουν Πρόσωπα (Persons), Ομάδες Προσώπων και Οργανισμούς (Organizations). Τα σχήματα περιγραφής για την αναπαράσταση των εννοιών αυτών προκύπτουν από το σχήμα περιγραφής SemanticBase DS. Το επίπεδο αφάισης (abstraction level) μιας σημασιολογικής οντότητας καθορίζεται από το στοιχείο AbstractionLevel, που ορίζεται στο σχήμα περιγραφής SemanticBase DS και διαθέτει ένα γνώρισμα, με Dimension, με τύπο μη αρνητικό ακέραιο. Όταν το AbstractionLevel απουσιάζει από μια σημασιολογική περιγραφή, η περιγραφή αναφέρεται σε συγκεκριμένο οπτικοακουστικό υλικό. Όταν AbstractionLevel.Dimension=0, υπάρχει η περιγραφή μιας συγκεκριμένης σημασιολογικής οντότητας (π.χ. ο ποδοσφαιριστής Zidane) που σχετίζεται με κάθε οπτικοακουστικό τμήμα όπου εμφανίζεται η οντότητα. Όταν το AbstractionLevel.Dimension έχει μη μηδενική τιμή, η σημασιολογική οντότητα είναι αφηρημένη και αναπαριστά μια κλάση (π.χ. η κλάση "PlayerObject" που αναπαριστά τους ποδοσφαιριστές). Οι σημασιολογικές οντότητες μπορεί να συσχετίζονται μέσω σχέσεων. Το MPEG-7 MDS διαθέτει πάνω από 100 πρότυπους τύπους σχέσεων που οργανώνονται σε 5 σχήματα κατηγοριοποίησης: (α) Το σχήμα κατηγοριοποίησης BaseRelation CS, που περιέχει βασικούς τύπους σχέσεων (equals, inside, refines κ.α.); (β) Το σχήμα κατηγοριοποίησης GraphRelation CS που περιέχει τύπους σχέσεων που υπάρχουν μεταξύ κόμβων γράφων (identity, equivalent κ.α.); (γ) Το σχήμα κατηγοριοποίησης SpatialRelation CS, που περιέχει τύπους χωρικών σχέσεων (over, below, north κ.α.); (δ) Το σχήμα κατηγοριοποίησης TemporalRelation CS, που περιέχει τύπους χρονικών σχέσεων (precedes, overlaps, contains κ.α.); (ε) Το σχήμα κατηγοριοποίησης SemanticRelation CS, που περιέχει τύπους σημασιολογικών σχέσεων (shows, agent, causer κ.α.).

Τα Στοιχεία Διαχείρισης Περιεχομένου επιτρέπουν την περιγραφή του κύκλου ζωής του περιεχομένου, από τη δημιουργία μέχρι την κατανάλωση (consumption). Παρέχουν πληροφορία για τη δημιουργία, τον τρόπο χρήσης και τα χαρακτηριστικά των μέσων που απαρτίζουν το περιεχόμενο.

Το σχήμα περιγραφής CreationInformation DS παρέχει πληροφορία σχετικά με τη δημιουργία του περιγραφόμενου οπτικοακουστικού υλικού, που απαρτίζεται από ένα τίτλο (σε μορφή κειμένου ή οπτικοακουστικού υλικού), υπομνήματα σε μορφή κειμένου και πληροφορία για δημιουργούς (creators), τόπους και ημερομηνίες δημιουργίας. Το σχήμα περιγραφής Classification DS περιγράφει πως το οπτικοακουστικό υλικό κατηγοριοποιείται με βάση τη θεματολογία (genre), το θέμα (subject), το σκοπό (purpose), τη γλώσσα κ.α. ενώ παρέχει πληροφορίες για κριτικές (reviews) και καθοδήγηση (guidance). Τέλος, το σχήμα

περιγραφής RelatedMaterial DS αναφέρεται σε υλικό που σχετίζεται με το περιγραφόμενο υλικό.

Το σχήμα περιγραφής MediaInformation DS ταυτοποιεί (identifies) την αρχική πηγή του περιεχομένου, από την οποία παράγονται διαφορετικά στιγμιότυπα του οπτικοακουστικού υλικού. Τα στιγμιότυπα του οπτικοακουστικού υλικού αναπαρίστανται ως Προφίλ Μέσων (Media Profiles), που είναι εκδόσεις του αρχικού υλικού που προκύπτουν με τη χρήση διαφορετικών κωδικοποιήσεων (encodings) και μορφών αποθήκευσης (storage) και διανομής (delivery).

Το σχήμα περιγραφής UsageInformation DS περιγράφει πληροφορίες για τον τρόπο χρήσης του οπτικοακουστικού υλικού και περιλαμβάνει πληροφορίες σχετικά με τη διαθεσιμότητα (availability), τα δικαιώματα χρήσης (usage rights), την καταγραφή χρήσης (usage record) καθώς και οικονομικές (financial) πληροφορίες.

Στοιχεία Πλοήγησης και Πρόσβασης (Navigation and Access Elements)

Τα Στοιχεία Πλοήγησης και Πρόσβασης χρησιμοποιούνται για να περιγράψουν την πλοήγηση μέσω συνόψεων (summaries) οπτικοακουστικού υλικού και διαφοροποιήσεις (variations) του υλικού. Τα Στοιχεία Πλοήγησης και Πρόσβασης του MPEG-7 MDS διευκολύνουν την πλοήγηση και την ανάκτηση οπτικοακουστικού υλικού με τον ορισμό συνόψεων, διαμερίσεων (partitions), αποσυνθέσεων (decompositions) και διαφοροποιήσεων.

Έτσι, παρέχεται η δυνατότητα ορισμού συνόψεων του οπτικοακουστικού υλικού, ώστε να υποστηριχτεί η πλοήγηση, η οπτικοποίηση (visualization) και η αναπαραγωγή του ήχου του οπτικοακουστικού υλικού. Το σχήμα περιγραφής Summary DS επιτρέπει δυο τρόπους δόμησης των συνόψεων: (α) Ιεραρχική δόμηση, οπότε η πληροφορία οργανώνεται σε επίπεδα που περιγράφουν το οπτικοακουστικό υλικό με περισσότερη ή λιγότερη λεπτομέρεια, ανάλογα με την απόστασή τους από τη ρίζα της σύνοψης; και (β) Ακολουθιακή δόμηση, όπου οι συνόψεις είναι ακολουθίες εικόνων ή τμημάτων video, πιθανώς συγχρονισμένες με ήχο.

Μια άλλη δυνατότητα είναι η περιγραφή διαμερίσεων και αποσυνθέσεων του οπτικοακουστικού υλικού στα πεδία του χρόνου, του τόπου και της συχνότητας (frequency). Οι διαμερίσεις και οι αποσυνθέσεις μπορούν να χρησιμοποιηθούν για την περιγραφή διαφορετικών απόψεων του οπτικοακουστικού υλικού.

Τέλος, παρέχεται, μέσω του σχήματος περιγραφής Variation DS, η δυνατότητα ορισμού περιγραφών διαφοροποιήσεων του οπτικοακουστικού υλικού (όπως συνόψεις, εκδόσεις σε διαφορετικές γλώσσες, εκδόσεις με τη χρήση διαφορετικών μέσων κ.α.). Έτσι, θα είναι δυνατή η επιλογή της καταλληλότερης διαφοροποίησης ενός οπτικοακουστικού προγράμματος που μπορεί, αν είναι απαραίτητο, να αντικαταστήσει το αρχικό, ώστε να

προσαρμοστεί (adapt) στις δυνατότητες των συσκευών (devices) και των προτιμήσεων των χρηστών (user preferences).

Στοιχεία Οργάνωσης Περιεχομένου (Content Organization Elements)

Τα Στοιχεία Οργάνωσης Περιεχομένου χρησιμοποιούνται για την περιγραφή του περιεχομένου πολυμέσων χρησιμοποιώντας κατηγοριοποίηση, μοντελοποίηση (modeling) και ορισμό συλλογών (collections) πολυμέσων. Τα Στοιχεία Οργάνωσης Περιεχομένου του MPEG-7 MDS επιτρέπουν την οργάνωση και τη μοντελοποίηση συλλογών οπτικοακουστικού περιεχομένου και περιγραφών. Το σχήμα περιγραφής Collection DS οργανώνει συλλογές (τμημάτων) οπτικοακουστικού περιεχομένου, γεγονότων και αντικειμένων, επιτρέποντας έτσι την περιγραφή των συλλογών με βάση τις κοινές ιδιότητες (properties) των συστατικών τους.

Στοιχεία Αλληλεπίδρασης με το Χρήστη (User Interaction Elements)

Τα Στοιχεία Αλληλεπίδρασης με το Χρήστη χρησιμοποιούνται για την περιγραφή των προτιμήσεων των χρηστών σχετικά με την κατανάλωση περιεχομένου πολυμέσων. Τα Στοιχεία Αλληλεπίδρασης με το Χρήστη περιγράφουν τις προτιμήσεις του χρήστη και το ιστορικό χρήσης (usage history) του για οπτικοακουστικό υλικό. Έτσι, είναι δυνατό το ταίριασμα (matching) μεταξύ των προτιμήσεων των χρηστών και των MPEG-7 περιγραφών, ώστε να υποστηριχθεί η εξατομίκευση (personalization) στην πρόσβαση, την παρουσίαση (presentation) και την κατανάλωση οπτικοακουστικού περιεχομένου. Οι προτιμήσεις των χρηστών αναπαρίστανται από το σχήμα περιγραφής UserPreference DS, που επιτρέπει τον καθορισμό του βαθμού προτίμησης κάθε στοιχείου προτίμησης, των χαρακτηριστικών μόνωσης (privacy) και της αυτόματης ή μη ενημέρωσης των προτιμήσεων. Οι προτιμήσεις χρηστών περιλαμβάνουν:

- Ένα σύνολο FilteringAndSearchPreferences (FASP) στοιχείων, που περιγράφουν τις προτιμήσεις των χρηστών σε ότι αφορά φιλτράρισμα και αναζήτηση (searching) σε οπτικοακουστικό περιεχόμενο.
- Ένα σύνολο BrowsingPreferences στοιχείων, που περιγράφουν τις προτιμήσεις των χρηστών σχετικά με την πλοήγηση σε οπτικοακουστικό υλικό.

2.4. Ανακεφαλαίωση

Στο κεφάλαιο αυτό, περιγράφηκαν τα δύο βασικά πρότυπα (standards) που χρησιμοποιεί το GraphOnto σύστημα, τα οποία είναι:

- Η OWL (Web Ontology Language), το κυρίαρχο πρότυπο στις γλώσσες περιγραφής οντολογιών, η οποία μέσω των πολυάριθμων δομών που διαθέτει χρησιμοποιείται για αναπαράσταση γνώσης και για σημασιολογική περιγραφή πραγμάτων και γεγονότων.
- Το MPEG-7 και πιο συγκεκριμένα ένα τμήμα του, το MPEG-7 Multimedia Description Schemes (MDS), το οποίο περιέχει δομές για τον ορισμό μεταδεδομένων για οπτικό-ακουστικό υλικό και αποτελεί τη βάση για τη σημασιολογική περιγραφή οπτικό-ακουστικού υλικού που υλοποιείται μέσα στο GraphOnto.

Κεφάλαιο 3^ο

3. Σχετικές εργασίες και αναγκαιότητα υλοποίησης του GraphOnto

3.1. Εισαγωγή

Στο κεφάλαιο αυτό, περιγράφονται περιληπτικά οι κυριότερες υλοποιήσεις που έχουν γίνει στον τομέα περιγραφής, δημιουργίας και επεξεργασίας των OWL οντολογιών. Οι υλοποιήσεις αυτές κατηγοριοποιούνται ως εξής:

- *Πλαίσια που παρέχουν προγραμματιστική πρόσβαση* για δημιουργία και διάβασμα OWL οντολογιών όπως το Jena API και το WonderWeb OWL API (το τελευταίο χρησιμοποιήθηκε κατά την υλοποίηση του GraphOnto και θα παρουσιαστεί στο 5^ο κεφάλαιο μαζί με τα υπόλοιπα συστατικά της αρχιτεκτονικής του GraphOnto).
- *Πλήρη γραφικά εργαλεία* για επεξεργασία OWL οντολογιών όπως τα Protégé, SemanticWorks, Smore και Swoop.

Ακολουθεί μια σύντομη αναφορά στις παραπάνω υλοποιήσεις, μια παρουσίαση των χαρακτηριστικών τους και στη συνέχεια πραγματοποιείται μια αξιολόγηση των δυνατοτήτων τους. Στο τέλος, φανερώνονται οι αδυναμίες τους, παρουσιάζονται οι ανάγκες που έπρεπε να καλυφθούν και τεκμηριώνεται η αναγκαιότητα υλοποίησης του GraphOnto.

3.2. Υλοποιήσεις στον τομέα των OWL οντολογιών

3.2.1. Jena

Το Jena [19] αποτελεί ένα Java πλαίσιο για τη δημιουργία εφαρμογών που να καλύπτουν τις απαιτήσεις του Semantic Web. Πιο συγκεκριμένα, παρέχει προγραμματιστικό περιβάλλον για τις RDF, RDFS, OWL και SPARQL (αποτελεί μια γλώσσα ερωτήσεων για την RDF) και ταυτόχρονα περιλαμβάνει μια βασισμένη σε κανόνες μηχανή εξαγωγής συμπερασμάτων (rule-based inference engine). Είναι ένα ελεύθερο, ανοιχτού κώδικα JAVA API που αναπτύχθηκε από την κοινότητα του SourceForge.net σε συνεργασία με το τμήμα Έρευνας σε Semantic Web των εργαστηρίων της Hewlet Packard (HP Labs Semantic Web Programme).

Τα βασικά μέρη που περιλαμβάνει το Jena πλαίσιο είναι:

- Ένα RDF API που παρέχει προγραμματιστική πρόσβαση για το διάβασμα και τη δημιουργία RDF οντολογιών οι οποίες μπορούν να αποθηκευθούν σε μορφή είτε RDF/XML είτε σε N-Triples.

- Ένα OWL API που παρέχει προγραμματιστική πρόσβαση για το διάβασμα και τη δημιουργία OWL οντολογιών οι οποίες μπορούν να αποθηκευθούν είτε στην κύρια μνήμη είτε μόνιμα σε ένα OWL/RDF αρχείο.
- Μια μηχανή ερωτήσεων που υποστηρίζει τη γλώσσα SPARQL [20].

3.2.2. Protégé

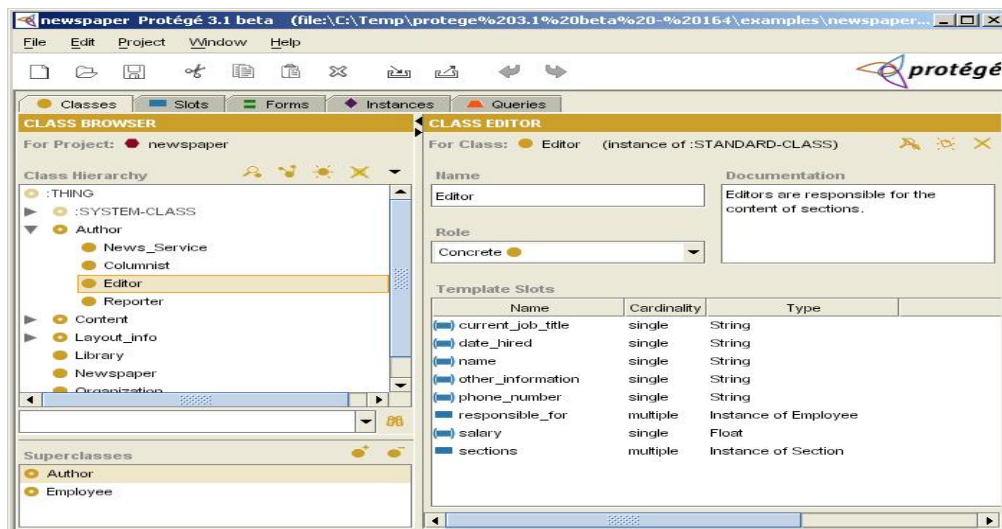
Το Protégé [21] είναι μια ελεύθερη, ανοιχτού κώδικα (open-source) και υλοποιημένη σε Java πλατφόρμα που παρέχει στην ολοένα αυξανόμενη κοινότητα χρηστών της μια σειρά από εργαλεία για την κατασκευή μοντέλων για συγκεκριμένα πεδία γνώσης με τη χρήση οντολογιών. Αποτελεί αυτή τη στιγμή την πιο δημοφιλή εφαρμογή στον κόσμο στον τομέα διαχείρισης οντολογιών. Η ανάπτυξη του έγινε από μια μεγάλη ομάδα ερευνητών στο τμήμα Ιατρικής Πληροφορικής (Stanford Medical Informatics) της Ιατρικής Σχολής του Πανεπιστημίου του Stanford (Stanford University School of Medicine) και συνεχίζεται ως σήμερα μέσω της μεγάλης κοινότητας που έχει δημιουργηθεί γύρω του. Το Protégé έχει χρηματοδοτηθεί από πλειάδα οργανισμών και ινστιτούτων [22] και διεθνή συνέδρια λαμβάνουν χώρα κάθε χρόνο με στόχο τη συνάντηση των ερευνητών που ασχολούνται με την ανάπτυξη και την επέκταση της πλατφόρμας.

Το Protégé υλοποιεί ένα σύνολο από δομές και ενέργειες για τη δημιουργία και διαχείριση οντολογιών δύο μορφών: οντολογιών βασισμένες σε Frames και OWL οντολογιών. Ο εσωτερικός του μηχανισμός διαβάσματος και αποθήκευσης των οντολογιών βασίζεται στα frames, ενώ ουσιαστικά η προέκταση του σε OWL οφείλεται σε ένα εξωτερικό plug-in. Είναι μια εφαρμογή που παρέχει δυνατότητες προσαρμογής στο χρήστη κατά τη διάρκεια της δημιουργίας μοντέλων γνώσης και της εισαγωγής δεδομένων. Επίσης, είναι επεκτάσιμο μέσω της plug-in αρχιτεκτονικής στην οποία είναι βασισμένο και τέλος παρέχει ένα Java API για τη δημιουργία εργαλείων γνώσης με τη χρήση οντολογιών.

Η πλατφόρμα του Protégé παρέχει δύο τρόπους για τη μοντελοποίηση των οντολογιών:

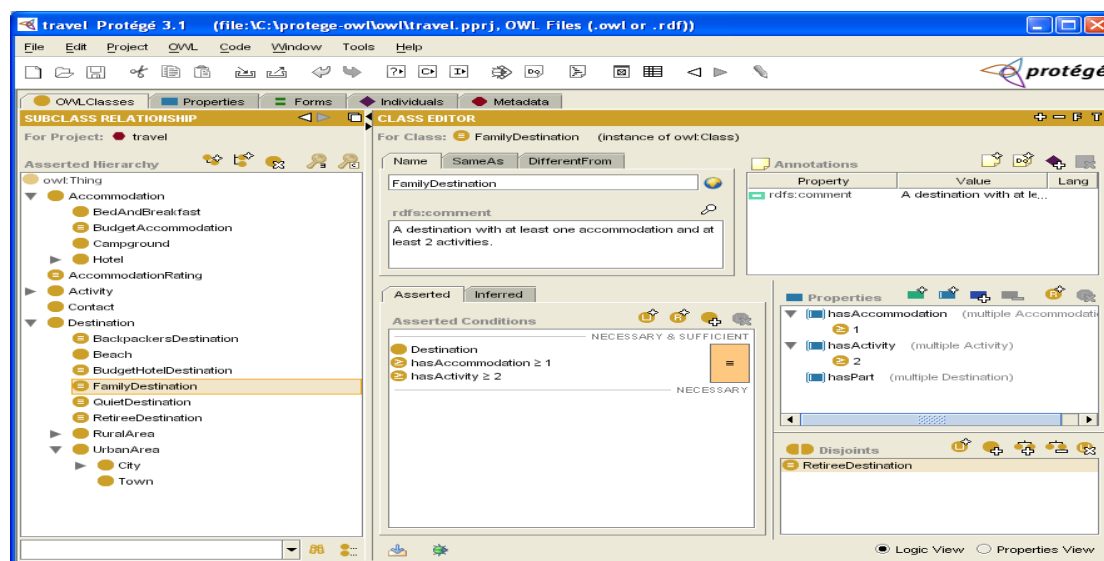
1. Μέσω του Protégé-Frames συντάκτη (editor) [23] που είναι ένα γραφικό βασισμένο σε φόρμες εργαλείο το οποίο υποστηρίζει την κατασκευή και αποθήκευση οντολογιών περιοχής που βασίζονται σε frames και την εισαγωγή δεδομένων βασισμένων σε αυτές τις οντολογίες. Ο Protégé-Frames συντάκτης αποτέλεσε την πρώτη έκδοση του Protégé και υλοποιεί ένα μοντέλο γνώσης που είναι συμβατό με το πρωτόκολλο Open Knowledge Base Connectivity (OKBC) [24]. Σε αυτό το μοντέλο, η οντολογία αποτελείται από ένα σύνολο από κλάσεις οι οποίες είναι οργανωμένες ιεραρχικά με τις πιο γενικές κλάσεις να βρίσκονται ψηλότερα από τις πιο συγκεκριμένες, και οι οποίες αναπαριστούν τις βασικές έννοιες μιας περιοχής, από ένα σύνολο από slots που σχετίζονται με τις κλάσεις για να περιγράψουν τις ιδιότητες και τις μεταξύ τους σχέσεις και από ένα σύνολο από

στιγμιότυπα αυτών των κλάσεων τα οποία εξειδικεύουν τις έννοιες των κλάσεων και έχουν συγκεκριμένες τιμές στις ιδιότητες τους.

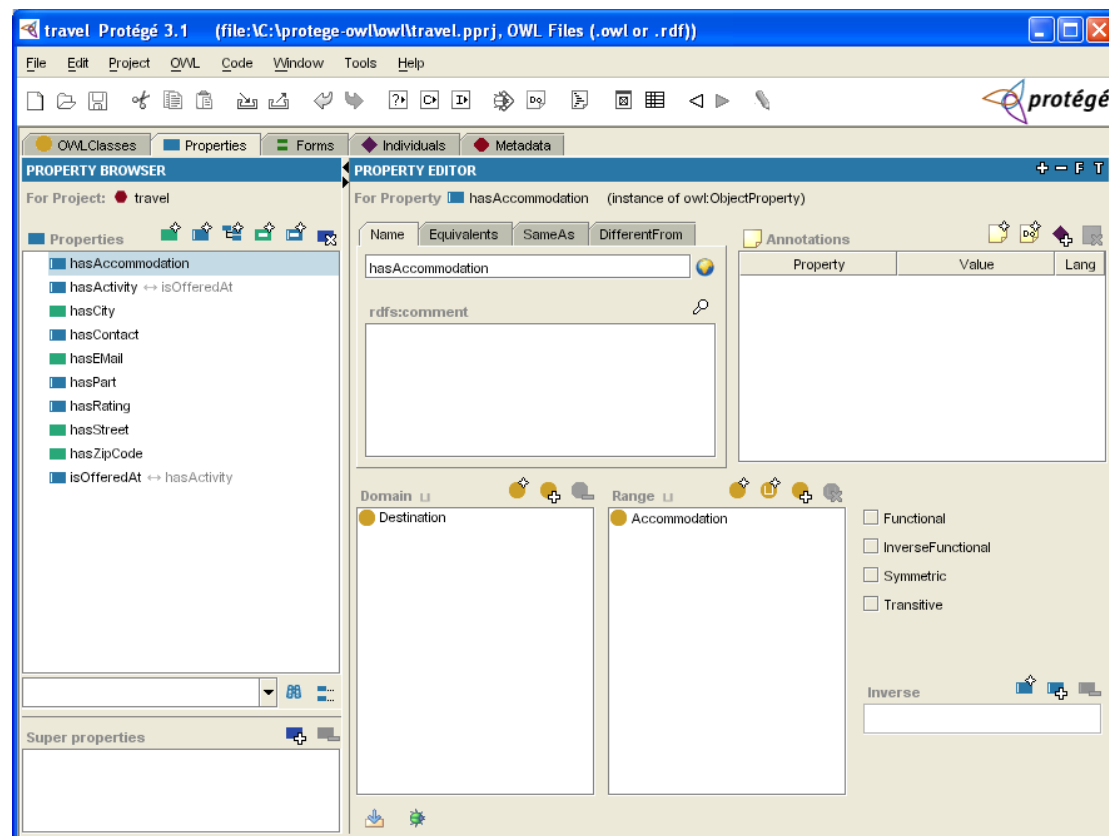


Εικόνα 37: Στιγμιότυπο από το Protégé-Frames στη σελίδα όπου ορίζονται οι κλάσεις της οντολογίας και τα slots τους.

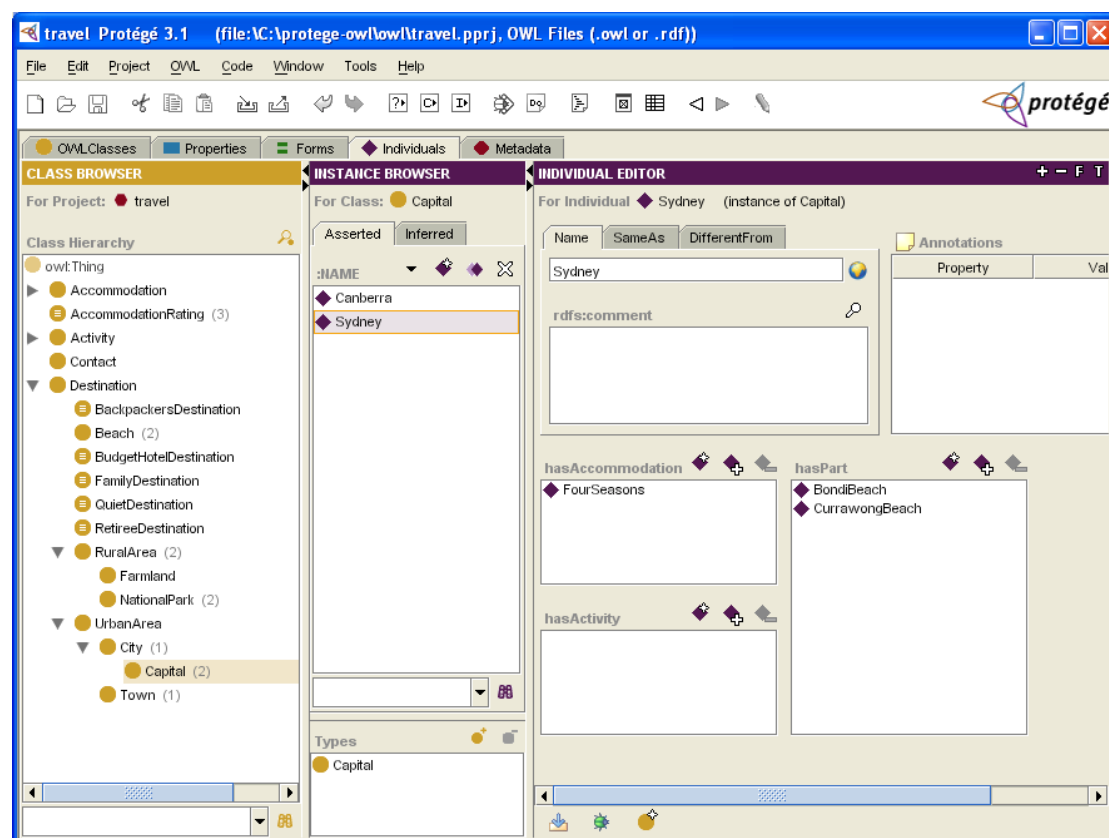
- Μέσω του Protégé-OWL συντάκτη (editor) [25] που είναι επίσης ένα γραφικό βασισμένο σε φόρμες εργαλείο το οποίο υποστηρίζει την κατασκευή και αποθήκευση οντολογιών περιοχής που βασίζονται στην OWL και την εισαγωγή δεδομένων βασισμένων σε αυτές τις οντολογίες. Ο Protégé-OWL συντάκτης αποτελεί μια επέκταση του Protégé ώστε να υποστηριχθεί η κυριάρχη γλώσσα οντολογιών OWL και παρουσιάζει πολλές ομοιότητες ως προς τη γραφική διεπαφή με τον Protégé-Frames συντάκτη. Παρέχει αρκετές δυνατότητες στους χρήστες όπως διάβασμα και αποθήκευση OWL και RDF οντολογιών, επεξεργασία OWL κλάσεων, ιδιοτήτων και ατόμων κλάσεων και ορισμό OWL εκφράσεων. Η αρχιτεκτονική του είναι ευέλικτη ώστε να επιτρέπει επεκτάσεις και είναι στενά συνδεδεμένη με το Jena [26].



Εικόνα 38: Στιγμιότυπο από το Protégé-OWL στη σελίδα όπου ορίζονται οι κλάσεις της οντολογίας και τα χαρακτηριστικά τους.



Εικόνα 39: Στιγμιότυπο από το Protégé-OWL στη σελίδα όπου ορίζονται οι ιδιότητες της οντολογίας και τα χαρακτηριστικά τους.

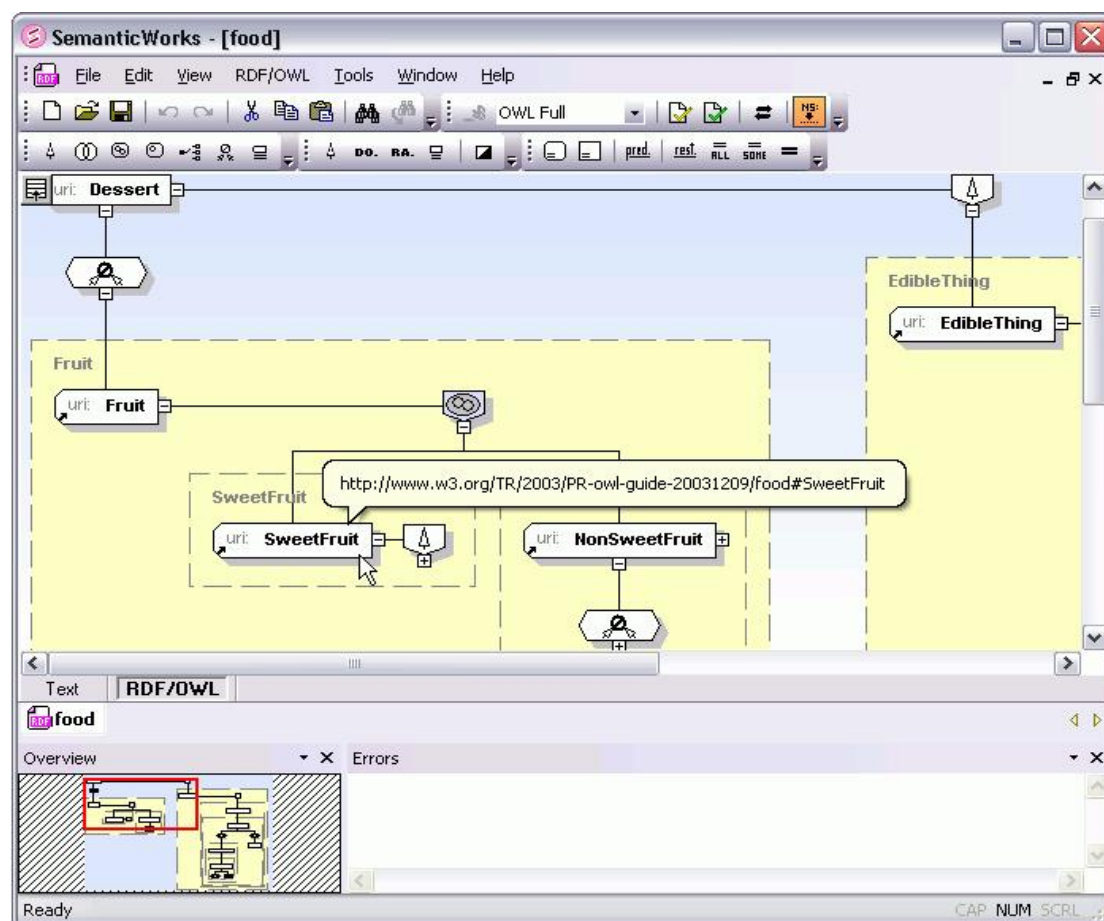


Εικόνα 40: Στιγμιότυπο από το Protégé-OWL στη σελίδα όπου ορίζονται τα άτομα κλάσεων της οντολογίας και τα χαρακτηριστικά τους.

3.2.3. Altova SemanticWorks

Το Altova SemanticWorks™ [27] είναι ένας γραφικός συντάκτης για RDF/OWL που δημιουργήθηκε από την Altova, μια εταιρία με παράδοση στην δημιουργία εφαρμογών βασισμένες σε XML με κυριότερη το XML Spy. Είναι εμπορική εφαρμογή (η τιμή του ξεκινάει από 99€) και διατίθεται δωρεάν για μια δοκιμαστική περίοδο 30 ημερών. Η τωρινή του έκδοση είναι η Altova SemanticWorks™ 2007.

Το Altova SemanticWorks™ 2007 επιτρέπει στο χρήστη να δημιουργεί και να επεξεργάζεται με γραφικό τρόπο έγγραφα με RDF στιγμιότυπα και OWL οντολογίες παρέχοντας πλήρη συντακτικό έλεγχο. Παρέχει παράλληλα βοήθεια στους συντάκτες οντολογιών με τη χρήση λιστών με τις επιτρεπόμενες επιλογές για τη δημιουργία έγκυρων εγγράφων εύκολα και γρήγορα. Προσφέρει επίσης την αυτόματη παραγωγή των δομών που σχεδιάστηκαν γραφικά σε RDF/OWL ή σε N-triples μορφή.



Εικόνα 41: Στιγμιότυπο από το SemanticWorks στη σελίδα όπου ορίζονται γραφικά οι κλάσεις μιας οντολογίας και τα χαρακτηριστικά τους.

3.2.4. Smore

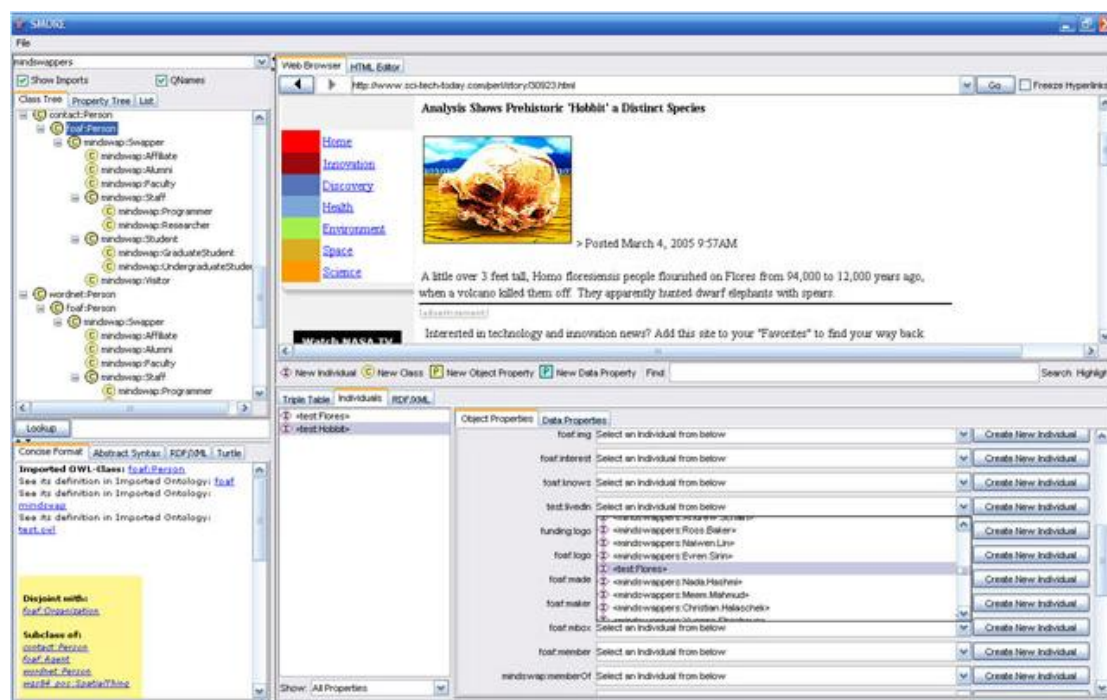
Το Smore [28] είναι ένα γραφικό εργαλείο που επιτρέπει στους χρήστες του να υπομνηματίζουν (markup) HTML έγγραφα με τη χρήση της OWL χρησιμοποιώντας οντολογίες

που βρίσκονται στον Παγκόσμιο Ιστό. Έχει αναπτυχθεί στα πλαίσια του SWAP Project (Semantic Web Agents Project) από ένα γκρουπ ερευνητών που ασχολείται με εφαρμογές του SemanticWeb και που ανήκει στο εργαστήριο MIND LAB του Πανεπιστημίου του Maryland (University of Maryland Institute for Advanced Computer Studies). Οι βασικοί του στόχοι είναι:

- Να επιτρέπει στους χρήστες με περιορισμένη γνώση των όρων και της σύνταξης της OWL να υπομνηματίζουν HTML έγγραφα.
- Να παρέχει λειτουργικότητα για τη χρήση των κλάσεων, ιδιοτήτων και ατόμων από υπάρχουσες οντολογίες, για περιορισμένη επεξεργασία των οντολογιών και για δημιουργία οντολογίας από την αρχή χρησιμοποιώντας όρους από έγγραφα του Παγκόσμιου Ιστού.
- Να παρέχει ένα ευέλικτο περιβάλλον στο χρήστη για τη δημιουργία απλών σελίδων του Παγκόσμιου Ιστού ταυτόχρονα με την υπομνημάτισή τους.

Το SMORE συγκεντρώνει μια σειρά από αξιόλογα χαρακτηριστικά που είναι τα παρακάτω:

- Είναι βασισμένο σε OWL καθώς η νέα έκδοσή του έχει σχεδιαστεί για εύκολη δημιουργία OWL οντολογιών από την πληροφορία των ιστοσελίδων, ενώ οι παλαιότερες εκδόσεις του ήταν βασισμένες στην RDF.
- Το SMORE επιτρέπει την εισαγωγή πολλαπλών υπάρχουσων οντολογιών στην υπό δημιουργία οντολογία.
- Επιτρέπει την εύκολη δημιουργία OWL οντοτήτων (κλάσεων, ιδιοτήτων και ατόμων) επιλέγοντας κείμενο από ιστοσελίδες.
- Υποστηρίζει επεξεργασία οντολογιών βασισμένη σε triples μορφή.
- Παρέχει έξυπνο συντάκτη για τη δημιουργία ατόμων κλάσεων. Χρησιμοποιώντας τα πεδία ορισμού και τα εύρη τιμών των ιδιοτήτων παρουσιάζει λίστες με τις επιτρεπόμενες τιμές των ιδιοτήτων με συνέπεια την απλοποίηση της εισαγωγής εγκύρων δεδομένων.
- Ενσωματώνει το γνωστό SWOOP Browser παρέχοντας έναν ξεκάθαρο και συνεπή τρόπο για την πλοήγηση στις κλάσεις και τις ιδιότητες, μαζί με μια πλήρη λειτουργικότητα αναζήτησης.
- Παρέχει μηχανισμό τονισμού λαθών και προσπάθειας αποφυγής τους, καθώς αναγνωρίζει μη έγκυρα triples, αποτρέποντας την οντολογία να γίνει μη συνεπής και βοηθώντας το χρήστη να διορθώσει τα τυχόν λάθη του.
- Περιέχει ενσωματωμένο συντάκτη για έγγραφα HTML επιτρέποντας στο χρήστη να επεξεργάζεται τις ιστοσελίδες παράλληλα με τη διαδικασία υπομνημάτισής τους σε OWL.



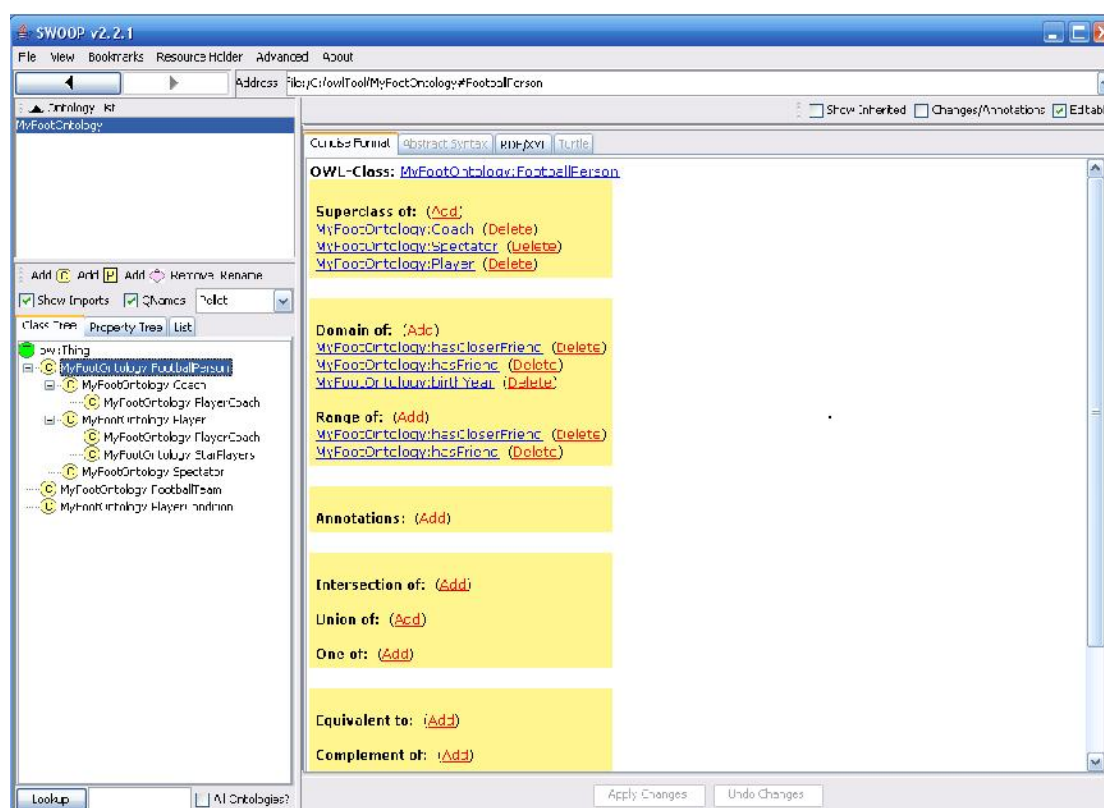
Εικόνα 42: Στιγμιότυπο από το Smore όπου φαίνεται ο τρόπος υπομνημάτισης ιστοσελίδας και γραφικού ορισμού των κλάσεων, των ιδιοτήτων και των ατόμων κλάσεων.

3.2.5. Swoop

Το Swoop [29] είναι ένας γραφικός συντάκτης OWL οντολογιών ο οποίος μέσω του Web-like γραφικού του περιβάλλοντος του παρέχει γρήγορη και εύκολη περιήγηση και επεξεργασία των OWL οντολογιών. Έχει αναπτυχθεί, όπως και το Smore, στα πλαίσια του SWAP Project (Semantic Web Agents Project) από ένα γκρουπ ερευνητών που ασχολείται με εφαρμογές του SemanticWeb και που ανήκει στο εργαστήριο MIND LAB του Πανεπιστημίου του Maryland (University of Maryland Institute for Advanced Computer Studies). Τα βασικά του χαρακτηριστικά είναι:

- Ο σχεδιασμός του προσομοιάζει έναν περιηγητή διαδικτύου (Web Browser). Πιο συγκεκριμένα περιέχει μπάρα διευθύνσεων (address bar) όπου μπορεί να εισαχθεί απευθείας το URI της οντολογίας (ή κλάσης ή ιδιότητας ή ατόμου) που επιθυμεί ο χρήστης να φορτωθεί. Η πλοήγηση ανάμεσα στις οντότητες της οντολογίας γίνεται μέσω υπερσυνδέσμων (hyperlink). Περιέχει επίσης κουμπιά ιστορικού (history buttons) όπως τα Προηγούμενο/Επόμενο (Back/Next) και σώσιμο σελιδοδείκτων (bookmarks) για μετέπειτα αναφορές.
- Όλη η επεξεργασία των οντολογιών είναι ενσωματωμένη σε ένα HTML γραφικό περιβάλλον. Χρησιμοποιούνται διάφοροι χρωματικοί συνδυασμοί και γραμματοσειρές για να δίνεται έμφαση στις πιθανές διαφοροποιήσεις της οντολογίας ή στις διαφορετικές αναπαραστάσεις των OWL δομών.

- Υποστηρίζει ταυτόχρονη επεξεργασία πολλαπλών οντολογιών για περιήγηση, σύγκριση και αντιστοιχήσεις.
- Βασίζεται στο WonderWeb OWL API για το διάβασμα των οντολογιών.
- Υποστηρίζει τη χρήση μιας μηχανής εξαγωγής συμπερασμάτων (για την ακρίβεια τον Pellet Reasoner [30]) και παρέχει βοήθεια στον χρήστη να εντοπίσει και να απομακρύνει τις πιθανές ασυνέπειες (inconsistencies) της οντολογίας.



Εικόνα 43: Στιγμιότυπο από το Swoop όπου φαίνεται το δένδρο ιεραρχίας κλάσεων και η πληροφορία της επιλεγμένης κλάσης.

3.3. Αξιολόγηση

Σε αυτήν την ενότητα θα παρουσιαστούν ορισμένες παρατηρήσεις, κριτικές και συμπεράσματα που προέκυψαν ύστερα από τη χρήση των παραπάνω εφαρμογών.

Οι παρατηρήσεις που προέκυψαν ύστερα από τη χρήση του Altova SemanticWorks είναι οι παρακάτω:

- Το απόλυτα γραφικό περιβάλλον της εφαρμογής είναι εύκολο στη χρήση, αλλά στερείται πληρότητας καθώς δεν είναι δυνατή η πλήρης παρουσίαση των δεντρών ιεραρχίας των κλάσεων, ιδιοτήτων και ατόμων. Επίσης, κατά τη διάρκεια της πλοήγησης στην οντολογία επιτρέπει τη δημιουργία κύκλων και έχει ως αποτέλεσμα την ανάπτυξη τεράστιων σχημάτων.

- Δε παρέχει μηχανισμό πρόληψης λαθών και ελέγχει στο τέλος αν η οντολογία είναι σωστά δομημένη και έγκυρη. Επιπλέον δεν προσφέρει αναλυτικές εξηγήσεις για το είδος των λαθών, παρά αρκείται σε αναφορές ότι η OWL σύνταξη είναι μη έγκυρη.
- Παρότι επιτρέπει ταυτόχρονη διαχείριση πολλαπλών οντολογιών, αυτές δεν επικοινωνούν μεταξύ τους ώστε να οριστούν αντιστοιχίσεις μεταξύ των δομών τους.
- Η μορφή που αποθηκεύει τις οντολογίες είναι βασισμένη σε RDF και όχι σε OWL.
- Επιτρέπει την απευθείας επεξεργασία από το χρήστη της OWL μορφής της οντολογίας, πράγμα που μπορεί να οδηγήσει εύκολα σε λάθη και δημιουργία μη σωστά δομημένης οντολογίας.
- Δεν υποστηρίζει να οριστεί το εύρος τιμών μιας ιδιότητας τύπου δεδομένων ως μια απαρίθμηση κυριολεκτικών τιμών.
- Δεν αναγνωρίζει πεδία ορισμών ιδιοτήτων αντικειμένων με παραπάνω από μία κλάσεις και τονίζει ως μη έγκυρα τα άτομα κλάσεων που έχουν στιγμιότυπα αυτών των ιδιοτήτων.
- Επιτρέπει σε άτομα κλάσεων να έχουν τιμές σε ιδιότητες των οποίων το πεδίο ορισμού δεν περιλαμβάνει την κλάση στην οποία ανήκουν αυτά τα άτομα.
- Επιτρέπει σε άτομα κλάσεων να αποτελούν τιμές σε ιδιότητες αντικειμένων των οποίων το πεδίο τιμών δεν περιλαμβάνει την κλάση στην οποία ανήκουν αυτά τα άτομα.
- Δεν υποστηρίζει αυτόματες ενημερώσεις των ατόμων (συμμετρικές ιδιότητες, αντίστροφες ιδιότητες).

Οι παρατηρήσεις που προέκυψαν ύστερα από τη χρήση του Smore είναι οι παρακάτω:

- Ουσιαστικά παρέχει μόνο δυνατότητες πλοήγησης σε υπάρχουσες οντολογίες και δεν επιτρέπει την επεξεργασία τους.
- Παρέχει πολύ περιορισμένες δυνατότητες δημιουργίας νέας οντολογίας καθώς επιτρέπει μόνο τη δημιουργία απλών κλάσεων, ιδιοτήτων και ατόμων χωρίς να αξιοποιεί τις υπόλοιπες μεγάλες σημασιολογικές δυνατότητες που προσφέρει η OWL.
- Είναι σχετικά δύσχρηστο και δυσκολεύει ακόμα και έναν έμπειρο χρήστη στον τομέα των οντολογιών.

Οι παρατηρήσεις που προέκυψαν ύστερα από τη χρήση του Swoop είναι οι παρακάτω:

- Δεν υποστηρίζει να οριστεί το εύρος τιμών μιας ιδιότητας τύπου δεδομένων ως μια απαρίθμηση κυριολεκτικών τιμών. Αν ανοιχθεί οντολογία που έχει τέτοια ιδιότητα, τότε το πρόγραμμα σκάει όταν ζητηθεί η OWL μορφή της οντολογίας.
- Τα άτομα κλάσεων αποθηκεύονται σε RDF μορφή και όχι σε OWL μορφή.
- Δε παρέχει μηχανισμό πρόληψης λαθών και ελέγχει στο τέλος αν η οντολογία είναι σωστά δομημένη και έγκυρη μόνο με τη χρήση συγκεκριμένης μηχανής συμπερασμάτων. Σε περίπτωση που δεν χρησιμοποιηθεί τέτοια μηχανή, επιτρέπει τον ορισμό οποιασδήποτε

δομής χωρίς να λαμβάνει υπόψιν κανέναν από τους σημασιολογικούς περιορισμούς που ορίζει η OWL. Οι παρατηρήσεις που ακολουθούν ενισχύουν το παραπάνω συμπέρασμα.

- Επιτρέπει σε μία ιδιότητα τύπου δεδομένων και μία ιδιότητα αντικειμένων να συνδέονται με σχέσεις ιεραρχίας ή ισοδυναμίας.
- Επιτρέπει τον ορισμό μιας ιδιότητας αντικειμένων ως ταυτόχρονα συναρτησιακή και μεταβατική ή ως ταυτόχρονα αντίστροφα συναρτησιακή και μεταβατική.
- Δε λαμβάνει υπόψιν στη δημιουργία ατόμων τυχόν περιορισμούς που ισχύουν για τις κλάσεις αυτών των ατόμων σε κάποιες ιδιότητες.
- Δεν ελέγχει τα πεδία ορισμού και τα εύρη τιμών των ιδιοτήτων αντικειμένων στη δημιουργία στιγμιοτύπων ιδιοτήτων σε άτομα κλάσεων.
- Επιτρέπει σε άτομα να έχουν ταυτόχρονα τύπο δύο κλάσεις που είναι ξένες μεταξύ τους.
- Επιτρέπει τη δημιουργία νέων ατόμων που να ανήκουν σε κλάση η οποία ορίζεται με απευθείας απαρίθμηση των μελών της.
- Δεν υποστηρίζει αυτόματες ενημερώσεις των ατόμων (συμμετρικές ιδιότητες, αντίστροφες ιδιότητες).

Οι παρατηρήσεις που προέκυψαν ύστερα από τη χρήση του Protégé (έκδοση 3.1.1) είναι οι παρακάτω:

- Επιτρέπει το άνοιγμα μόνο μιας οντολογίας κάθε φορά και έτσι δεν υπάρχουν ταυτόχρονα πολλαπλές ανοιχτές οντολογίες.
- Ως αρχεία οντολογιών αναγνωρίζει μόνο αυτά με κατάληξη .owl. Αν επιλεγεί άλλου τύπου αρχείο, αναζητά αρχείο στον επιλεγμένο φάκελο με το όνομα του επιλεγμένου αρχείου και την κατάληξη .owl. Όμως, αυτό καταλήγει είτε σε μη άνοιγμα του αρχείου της οντολογίας είτε σε άνοιγμα λανθασμένου αρχείου (σε περίπτωση που στον επιλεγμένο φάκελο υπάρχουν τα αρχεία MyOntology και MyOntology.owl που περιέχουν διαφορετικές οντολογίες και επιλεγεί το άνοιγμα του MyOntology, το αρχείο που θα ανοίγει πάντα θα είναι το MyOntology.owl).
- Παρότι όταν διαβάσει μια οντολογία εμφανίζει τις τυχόν ισοδύναμες κλάσεις και ιδιότητες, απουσιάζει η δυνατότητα ορισμού ισοδύναμων κλάσεων ή ιδιοτήτων και γενικότερα δεν παρέχει κάποια συγκεκριμένη όψη για ορισμό αντιστοιχίσεων μεταξύ των OWL δομών.
- Παρότι ταξινομεί αλφαβητικά τα δένδρα ιεραρχίας κλάσεων και ιδιοτήτων της οντολογίας που ανοίγει, στη συνέχεια για κάθε κλάση ή ιδιότητα που προστίθεται από το χρήστη δημιουργείται ένας κόμβος στο τέλος του αντίστοιχου δένδρου και όχι στη θέση που θα του αντιστοιχούσε με βάση το όνομα της κλάσης ή της ιδιότητας.
- Δε παρέχει πλήρη μηχανισμό πρόληψης λαθών και ελέγχει στο τέλος αν η οντολογία είναι έγκυρη μόνο με τη χρήση συγκεκριμένης μηχανής συμπερασμάτων. Σε περίπτωση που δεν χρησιμοποιηθεί τέτοια μηχανή, επιτρέπει τον ορισμό δομών αγνοώντας αρκετούς

σημασιολογικούς ορισμούς που ορίζει η OWL. Οι παρατηρήσεις που ακολουθούν ενισχύουν το παραπάνω συμπέρασμα.

- Επιτρέπει σε μια κλάση να έχει για μια συγκεκριμένη ιδιότητα περισσότερους του ενός περιορισμούς στο εύρος τιμών της (παραδείγματος χάρη μπορεί να υπάρξει περιορισμός όλες οι τιμές από μια κλάση A και ταυτόχρονα περιορισμός κάποιες τιμές από κλάση B, τη στιγμή που οι επεκτάσεις κλάσεων των A και B δεν έχουν κοινή στοιχεία).
- Επιτρέπει να οριστεί σε μια ιδιότητα περιορισμός εύρους τιμών με εύρος τιμών διαφορετικό (και όχι υποσύνολο) από το εύρος τιμών της ιδιότητας.
- Δεν κάνει σωστό έλεγχο στους περιορισμούς πολλαπλότητας (παραδείγματος χάρη μπορεί να υπάρξει σε μια κλάση για μια συγκεκριμένη ιδιότητα περιορισμός ελάχιστου πλήθους 2 και περιορισμός μέγιστου πλήθους 1).
- Επιτρέπει σε μια κλάση A και μια κλάση B να συνδέονται με σχέση ιεραχίας και ταυτόχρονα να είναι ξένες μεταξύ τους. Όμως αν κλάση A είναι υπερκλάση της B, τότε σημαίνει ότι η επέκταση κλάσης της A περιλαμβάνει την επέκταση κλάσης της B, πράγμα άτοπο αν είναι ταυτόχρονα και ξένες μεταξύ τους αφού τότε η επέκταση κλάσης της A και η αντίστοιχη της B δεν έχουν κοινά στοιχεία.
- Επιτρέπει τον ορισμό μιας ιδιότητας αντικειμένων ως ταυτόχρονα συναρτησιακή και μεταβατική ή ως ταυτόχρονα αντίστροφα συναρτησιακή και μεταβατική.
- Επιτρέπει σε ένα άτομο A και σε ένα άτομο B να είναι ταυτόχρονα ακριβώς όμοια και διαφορετικά.
- Επιτρέπει σε άτομα να έχουν ταυτόχρονα τύπο δύο κλάσεις που είναι ξένες μεταξύ τους.
- Επιτρέπει τη δημιουργία νέων ατόμων που να ανήκουν σε κλάση η οποία ορίζεται με απευθείας απαρίθμηση των μελών της.
- Επιτρέπει σε άτομα να έχουν ίδιες τιμές σε μια ιδιότητα αντικειμένων που έχει οριστεί αντίστροφα συναρτησιακή.
- Κατά την διαδικασία πρόσθεσης τιμών σε ιδιότητες ατόμων δε λαμβάνει υπόψιν τυχόν περιορισμούς πολλαπλότητας ή περιορισμός εύρους τιμών του τύπου όλες οι τιμές από.

Με βάση τις δυνατότητες της κάθε εφαρμογής που αναφέρθηκαν στις προηγούμενες παραγράφους και με βάση τις παραπάνω παρατηρήσεις, κατασκευάστηκε ο παρακάτω συγκεντρωτικός πίνακας αξιολόγησης των γραφικών εργαλείων που επιτρέπουν τη διαχείριση OWL οντολογιών.

Χαρακτηριστικά	SemanticWorks	Smore	Swoop	Protégé
Είδος γραφικού περιβάλλοντος	Πλήρως γραφικό περιβάλλον	HTML	HTML	Βασισμένο σε φόρμες
Μορφή αποθήκευσης της οντολογίας	RDF	OWL	OWL (αλλά τα άτομα σε RDF)	OWL

Χαρακτηριστικά	SemanticWorks	Smore	Swoop	Protégé
Ευκολία χρήσης	ΝΑΙ (αλλά επιτρέπει τη δημιουργία κύκλων και μεγάλων σχημάτων και την επεξεργασία της OWL μορφής της οντολογίας από το χρήστη)	ΟΧΙ	ΝΑΙ	ΝΑΙ
Εμφάνιση bugs	ΟΧΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ
Ελεύθερη πλατφόρμα	ΟΧΙ (εμπορική εφαρμογή)	ΝΑΙ (βασίζεται στο OWL API)	ΝΑΙ (βασίζεται στο OWL API)	ΝΑΙ (βασίζεται στο JENA API)
Επεκτασιμότητα	ΟΧΙ	ΟΧΙ	ΟΧΙ	ΝΑΙ
Σχεδιασμένο εξαρχής για RDF/OWL	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΟΧΙ (η OWL υποστηρίζεται μέσω plug-in)
Άνοιγμα όλων των αρχείων οντολογιών ανεξαρτήτου κατάληξης	ΟΧΙ (μόνο .owl και .rdf)	ΝΑΙ	ΝΑΙ	ΟΧΙ (μόνο .owl και .rdf)
Υποστήριξη της λειτουργικότητας της OWL-DL	ΝΑΙ (αλλά δεν αναγνωρίζει πεδία ορισμών με παραπάνω από μία κλάσεις και δεν επιτρέπει σε ιδιότητες τύπου δεδομένων να έχουν εύρος τιμών μια απαρίθμηση κυριολεκτικών τιμών)	Πολύ περιορισμένη (μόνο απλές κλάσεις, ιδιότητες και άτομα)	ΝΑΙ (αλλά δεν επιτρέπει σε ιδιότητες τύπου δεδομένων να έχουν εύρος τιμών μια απαρίθμηση κυριολεκτικών τιμών)	ΝΑΙ (αλλά απουσιάζει η δυνατότητα ορισμού ισοδύναμων κλάσεων και ιδιοτήτων)
Υποστήριξη πολλαπλών ανοιχτών οντολογιών	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΟΧΙ (υποστηρίζει μόνο το άνοιγμα οντολογιών που εισάγονται στην τρέχουσα οντολογία)
Υποστήριξη αντιστοιχήσεων μεταξύ οντολογιών	ΟΧΙ	ΟΧΙ	ΝΑΙ	ΟΧΙ

Χαρακτηριστικά	SemanticWorks	Smore	Swoop	Protégé
Δυνατότητα δημιουργίας νέας οντολογίας	NAI	OXI	NAI	NAI
Ενσωματωμένος μηχανισμός πρόληψης λαθών	OXI	NAI	OXI	OXI
Σύνδεση με εξωτερική μηχανή εξαγωγής συμπερασμάτων	OXI	NAI	NAI	NAI
Σημαιολογικοί έλεγχοι κατά τη διάρκεια επεξεργασίας της οντολογίας	Όχι ικανοποιητικός αριθμός	OXI	OXI	Όχι ικανοποιητικός αριθμός
Αυτόματες ενημερώσεις κατά τη διάρκεια επεξεργασίας της οντολογίας	OXI	OXI	OXI	NAI
Υποστήριξη εξατομικευμένων λειτουργιών	OXI	OXI	OXI	Παρέχονται προτιμήσεις μόνο για τα γραφικά του περιβάλλοντος εργασίας

Πίνακας 1: Πίνακας αξιολόγησης εργαλείων διαχείρισης OWL οντολογιών.

3.4. Αναγκαιότητα και κίνητρο υλοποίησης του GraphOnto

Όπως φάνηκε από την παραπάνω αξιολόγηση, παρουσιάζονται αισθητές αδυναμίες σε όλες τις υλοποιήσεις που έχουν γίνει μέχρι σήμερα στον τομέα των OWL οντολογιών, ακόμα και στην ευρέως γνωστή και πλούσια χρηματοδοτούμενη πλατφόρμα του Protégé. Οι αδυναμίες που καταγράφηκαν ενυπάρχουν στις τωρινές εκδόσεις αυτών των εργαλείων (η αξιολόγηση πραγματοποιήθηκε το Μάιο του 2007) και είναι σαφώς λιγότερες από αυτές που υπήρχαν την εποχή που αποφασίστηκε και ξεκίνησε η υλοποίηση του GraphOnto (Νοέμβριος 2003). Την εποχή εκείνη, δεν υφίστανταν κανένα από τα SemanticWorks, Smore και Swoop,

ενώ το Protégé έκανε τα πρώτα του βήματα στον τομέα των OWL οντολογιών παρέχοντας πολύ περιορισμένη λειτουργικότητα και έχοντας χαμηλή απόδοση.

Την ίδια εποχή, είχε αρχίσει η ανάπτυξη του DS-MIRF πλαισίου (περιγράφεται αναλυτικά στο επόμενο κεφάλαιο) το οποίο απαιτούσε την ύπαρξη μιας οντολογικής υποδομής που περιλαμβάνει πλήθος από οντολογίες, οι οποίες επιπρόσθετα διαχωρίζονται σε ανώτερες και σε οντολογίες περιοχής. Η λειτουργικότητα ταυτόχρονης διαχείρισης οντολογιών και η δυνατότητα διάκρισης των οντολογιών δεν παρέχονται ούτε από την τρέχουσα έκδοση του Protégé, πόσο μάλλον εκείνη την εποχή.

Ταυτόχρονα, το DS-MIRF πλαίσιο απαιτούσε την υποστήριξη πολυμεσικών εφαρμογών και συνεπώς προέκυψε η ανάγκη για ύπαρξη διαλειτουργικότητας μεταξύ του MPEG-7, επικρατέστερου προτύπου περιγραφής πολυμέσων, και της OWL, επικρατέστερης γλώσσας οντολογιών για περιγραφή γνώσης. Το Protégé είναι ένα αμιγές εργαλείο διαχείρισης οντολογιών και συνεπώς δεν μπορούσε να καλύψει ούτε αυτή την απαίτηση.

Τέλος, το DS-MIRF πλαίσιο έχει και ως στόχο το σημασιολογικό υπομνηματισμό πολυμέσων με τη βοήθεια οντολογιών. Ο υπομνηματισμός αυτός συνήθως γίνεται από διαφορετικούς και μη εξοικωμένους με την έννοια της οντολογίας χρήστες, με αποτέλεσμα να προκύπτει η αναγκαιότητα για παροχή εξατομικευμένης λειτουργικότητας, απόκρυψη των δύσκολων εννοιών της OWL και επικέντρωση στη δημιουργία σχετικά μικρών σημασιολογικών περιγραφών με τη χρήση επιλεγμένων κλάσεων και ιδιοτήτων. Τέτοια λειτουργία για χρησιμοποίηση με βάση κάποιες προτιμήσεις χρήστη ενός μέρους της οντολογίας και παραγωγή ανεξάρτητης σημασιολογικής περιγραφής δεν παρέχονταν από το Protégé και ούτε παρέχεται και στην τωρινή έκδοση.

Με βάση λοιπόν τις παραπάνω απαιτήσεις του DS-MIRF πλαισίου, κρίθηκε αναγκαία η υλοποίηση του GraphOnto, ενός συστήματος επεξεργασίας OWL οντολογιών το οποίο θα εντασσόταν μέσα στο DS-MIRF πλαίσιο. Το GraphOnto συνεπώς σχεδιάστηκε και υλοποιήθηκε με σκοπό να καλύψει τις παρακάτω ανάγκες (τις οποίες κανένα, ακόμα και σήμερα, εργαλείο διαχείρισης OWL οντολογιών δεν καλύπτει):

- Δυνατότητα δημιουργίας και επεξεργασίας OWL οντολογιών καλύπτοντας πλήρως τις δυνατότητες της OWL-DL.
- Ύπαρξη μηχανισμού που θα αποτρέπει τις λανθασμένες ενέργειες χρήστη και θα εξασφαλίζει τη συνέπεια των υπό επεξεργασία οντολογιών.
- Υποστήριξη ταυτόχρονης διαχείρισης πολλαπλών οντολογιών.
- Παροχή δυνατότητας εξατομίκευσης στο χρήστη και απόκρυψης πληροφορίας.
- Υποστήριξη σημασιολογικής περιγραφής πολυμέσων με τη χρήση του προτύπου MPEG-7.

3.5. Ανακεφαλαίωση

Στο κεφάλαιο αυτό παρουσιάστηκαν οι κυριότερες υλοποιήσεις που έχουν γίνει μέχρι σήμερα στον τομέα των OWL οντολογιών. Συγκεκριμένα, παρουσιάστηκε αρχικά το Jena API

που μαζί με το OWL API αποτελούν τα γνωστότερα πλαίσια που παρέχουν προγραμματιστική πρόσβαση για δημιουργία και διάβασμα OWL οντολογιών. Κατόπιν, περιγράφηκαν τα εργαλεία Protégé, Altova SemanticWorks, Smore και Swoor τα οποία αποτελούν γραφικές εφαρμογές για την επεξεργασία OWL οντολογιών. Ακολούθως, πραγματοποιήθηκαν δοκιμές με στόχο την αξιολόγηση των παραπάνω εργαλείων οι οποίες ανέδειξαν αρκετές αδυναμίες τους οι οποίες καταγράφηκαν σε ένα συγκεντρωτικό πίνακα.

Οι αδυναμίες αυτές σε συνδυασμό με τις ανάγκες που προέκυψαν από το DS-MIRF πλαίσιο, κατέστησαν απαραίτητο το σχεδιασμό και την υλοποίηση του GraphOnto. Η πλήρης OWL λειτουργικότητα που παρέχει το GraphOnto (περιγράφεται στο κεφάλαιο 6), τα επιπρόσθετα χαρακτηριστικά του που επιτρέπουν την εξατομίκευση του περιβάλλοντος εργασίας του χρήστη (περιγράφονται στο κεφάλαιο 7), μαζί με τη δυνατότητα χρήσης του GraphOnto σε πολυμεσικές εφαρμογές (περιγράφεται στο κεφάλαιο 8) καταδειχνουν την ανωτερότητα του GraphOnto σε σχέση με τις υπόλοιπες εφαρμογές.

Κεφάλαιο 4^ο

4. Το Πλαίσιο Διαχείρισης Μεταδεδομένων για Οπτικοακουστικό Υλικό (DS-MIRF Framework)

4.1. Εισαγωγή

Στο κεφάλαιο αυτό περιγράφεται το πλαίσιο διαχείρισης μεταδεδομένων για οπτικοακουστικό υλικό DS-MIRF [31] (Domain-Specific Multimedia Information and Filtering Framework), το οποίο αναπτύχθηκε στα πλαίσια του ερευνητικού προγράμματος DELOS II [32] [33]. Το πλαίσιο DS-MIRF παρέχει ένα σύνολο υπηρεσιών που υποστηρίζουν την ανάπτυξη εφαρμογών πολυμέσων, οι οποίες βασίζονται σε γνώση περιοχής και χρησιμοποιούν και επεκτείνουν τα πρότυπα MPEG-7 και MPEG-21. Το πλαίσιο DS-MIRF υλοποιεί, στον τομέα των πολυμέσων, μια προσέγγιση για την υποστήριξη σημασιολογικής διαλειτουργικότητας μεταξύ OWL και προτύπων που βασίζονται σε XML Schema, με σκοπό την εκμετάλλευση γνώσης περιοχής (domain knowledge) από εφαρμογές που βασίζονται στα πρότυπα.

Το υπόλοιπο του κεφαλαίου δομείται ως εξής: Στην ενότητα 4.2 παρουσιάζεται η προσέγγιση που προτείνεται μέσα στο πλαίσιο DS-MIRF για την υποστήριξη σημασιολογικής διαλειτουργικότητας μεταξύ προτύπων που βασίζονται σε XML Schema και OWL. Στην ενότητα 4.3 περιγράφεται η αρχιτεκτονική του πλαισίου DS-MIRF και στην ενότητα 4.4 γίνεται μια σύντομη παρουσίαση των συστατικών του, ανάμεσα στα οποία βρίσκεται και το GraphOnto.

4.2. Προσέγγιση Υποστήριξης Σημασιολογικής

Διαλειτουργικότητας

Στην ενότητα αυτή παρουσιάζεται συνοπτικά η προσέγγιση που υιοθετείται στο DS-MIRF πλαίσιο για την υποστήριξη σημασιολογικής διαλειτουργικότητας μεταξύ OWL και προτύπων που βασίζονται σε XML Schema. Συγκεκριμένα, παρουσιάζονται ένα μοντέλο αντιστοίχισης (mapping model) που επιτρέπει τη χρήση μεθοδολογιών (methodologies) και εργαλείων (tools) του Σημασιολογικού Ιστού από εφαρμογές που χρησιμοποιούν πρότυπα που βασίζονται σε XML Schema, και το λογισμικό (software) που το υλοποιεί. Τόσο το μοντέλο όσο και το λογισμικό επιτρέπουν τη διατήρηση της συμβατότητας με εφαρμογές που χρησιμοποιούν τις XML Schema εκδόσεις των προτύπων.

Το μοντέλο περιγράφει την αυτόματη μετατροπή των προτύπων που είναι εκφρασμένα σε XML Schema σε OWL (συγκεκριμένα, σε OWL-DL, που είναι η πιο εκφραστική έκδοση της OWL που εγγυάται περατότητα και αποτελεσματικότητα κατά την εξαγωγή συμπερασμάτων). Έτσι, η XML Schema σύνταξη των προτύπων εκφράζεται με τη μορφή

Κύριων Οντολογιών (Main Ontologies) σε OWL-DL και οι δομές των προτύπων γίνονται πρώτης τάξεως αντικείμενα του Σημασιολογικού Ιστού. Οι κύριες οντολογίες επιτρέπουν την ενσωμάτωση, στις δομές των προτύπων, γνώσης (knowledge) σχετικής με εφαρμογές και γνώσης περιοχής. Η γνώση αυτή είναι εκφρασμένη με τη μορφή OWL οντολογιών περιοχής (domain ontologies) και οντολογιών εφαρμογών (application ontologies). Επιπλέον, επιτρέπουν τη χρήση εργαλείων του Σημασιολογικού Ιστού, συμπεριλαμβανομένων των εργαλείων εξαγωγής συμπερασμάτων (reasoners), για τις περιγραφές που υπακούν στα πρότυπα. Το αποτέλεσμα της ενσωμάτωσης γνώσης περιοχής και εφαρμογών και της χρήσης εργαλείων του Σημασιολογικού Ιστού είναι ο εμπλουτισμός των περιγραφών με επιπλέον γνώση, που εξάγεται αυτόματα. Το μοντέλο και το λογισμικό υποστηρίζουν τη μετατροπή των OWL δομών σε XML Schema μορφή, ώστε να μπορούν να χρησιμοποιηθούν από εφαρμογές που είναι συμβατές με την XML Schema μορφή των προτύπων. Αυτό επιτυγχάνεται με την αυτόματη παραγωγή των Οντολογιών Ανακατασκευής (Reconstruction Ontologies), οι οποίες είναι OWL-DL οντολογίες που αναπαριστούν τις δομές της XML Schema που δε μπορούν να αναπαρασταθούν σε OWL.

Εφαρμογή της προσέγγισης αυτής αποτελεί η ανώτερη οντολογία για πολυμέσα και υπηρεσίες πολυμέσων που χρησιμοποιείται στο πλαίσιο DS-MIRF και αποτυπώνει πλήρως τη σημασιολογία του MPEG-7 MDS και της MPEG-21 DIA Architecture.

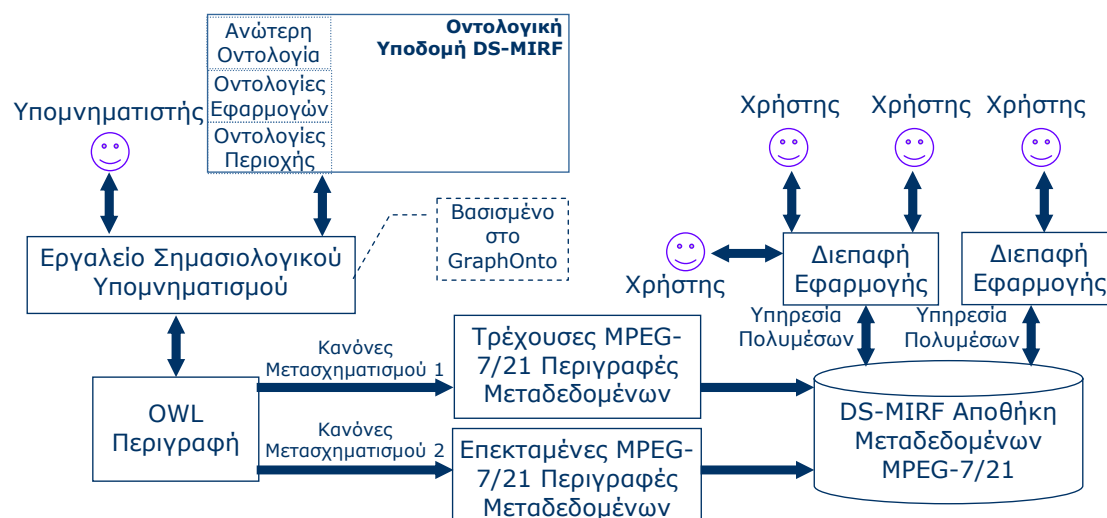
4.3. Αρχιτεκτονική του Πλαισίου DS-MIRF

Στην ενότητα αυτή παρουσιάζεται η αρχιτεκτονική του πλαισίου DS-MIRF το οποίο υλοποιεί, στον τομέα των πολυμέσων, την προσέγγιση που περιγράφηκε στην προηγούμενη ενότητα για την υποστήριξη σημασιολογικής διαλειτουργικότητας μεταξύ προτύπων που βασίζονται σε XML Schema και OWL. Το πλαίσιο DS-MIRF παρέχει ένα σύνολο υπηρεσιών που υποστηρίζουν την ανάπτυξη εφαρμογών πολυμέσων, οι οποίες βασίζονται σε γνώση περιοχής και χρησιμοποιούν και επεκτείνουν τα πρότυπα MPEG-7 και MPEG-21 [34]. Οι χρήστες του πλαισίου DS-MIRF διακρίνονται σε δυο κατηγορίες:

Τους υπομνηματιστές (annotators), οι οποίοι είναι υπεύθυνοι για τη σημασιολογική περιγραφή του περιεχομένου του διαθέσιμου οπτικοακουστικού υλικού, χρησιμοποιώντας γνώση περιοχής (π.χ. είναι υπεύθυνοι να περιγράψουν σημασιολογικά ένα τμήμα video που περιέχει ένα γκολ της ΑΕΚ σε βάρος του Ολυμπιακού). Έτσι, οι υπομνηματιστές είναι στην ουσία υπεύθυνοι για τον ορισμό σημασιολογικών μεταδεδομένων (semantic metadata) που περιγράφουν οπτικοακουστικό υλικό.

Τους τελικούς χρήστες (end users), οι οποίοι χρησιμοποιούν της εφαρμογές αναζήτησης, ανάκτησης και φιλτραρίσματος περιεχομένου πολυμέσων που υποστηρίζονται από της υπηρεσίες που παρέχει το πλαίσιο DS-MIRF. Στο υπόλοιπο του παρόντος κεφαλαίου οι τελικοί χρήστες θα αναφέρονται απλά ως χρήστες.

Η αρχιτεκτονική του πλαισίου DS-MIRF και η ροή πληροφορίας μεταξύ των συστατικών του απεικονίζονται στο παρακάτω σχήμα:



Εικόνα 44: Η Αρχιτεκτονική του Πλαισίου DS-MIRF

Όπως φαίνεται στο παραπάνω σχήμα, οι υπομνηματιστές χρησιμοποιούν κάποιο εργαλείο σημασιολογικού υπομνηματισμού (semantic annotation tool), που βασίζεται στο συστατικό λογισμικού (software component) GraphOnto. Το GraphOnto επιτρέπει τη διαχείριση οντολογιών και μεταδεδομένων που ορίζονται με βάση οντολογίες και επιτρέπει την εκμετάλλευση και επέκταση της οντολογικής υποδομής (ontological infrastructure) του πλαισίου DS-MIRF.

Η οντολογική υποδομή του πλαισίου DS-MIRF απαρτίζεται από:

- Μια Ανώτερη Οντολογία (Upper Ontology) που αποτυπώνει πλήρως τη σημασιολογία του MPEG-7 MDS και της MPEG-21 DIA Architecture [35].
- Ένα σύνολο από Οντολογίες Εφαρμογών (Application Ontologies), που επεκτείνουν τη σημασιολογία της Ανώτερης Οντολογίας με πληροφορία που σχετίζεται με εφαρμογές.
- Ένα σύνολο από Οντολογίες Περιοχής (Domain Ontologies), που επεκτείνουν τη σημασιολογία της Ανώτερης Οντολογίας και των Οντολογιών Εφαρμογών με γνώση περιοχής.

Οι οντολογίες που απαρτίζουν την οντολογική υποδομή του πλαισίου DS-MIRF είναι εκφρασμένες με τη χρήση των δομών της OWL, που είναι η επικρατούσα γλώσσα ορισμού οντολογιών. Έτσι, το αποτέλεσμα του σημασιολογικού υπομνηματισμού είναι μια OWL περιγραφή του περιεχομένου του οπτικοακουστικού υλικού. Η χρήση των οντολογιών επιτρέπει τον αυστηρό έλεγχο της σημασιολογικής ορθότητας των παραγόμενων σημασιολογικών περιγραφών με τη χρήση γνώσης περιοχής (π.χ. δεν επιτρέπεται μια ομάδα να έχει ταυτόχρονα ενεργούς, κατά τη διάρκεια της αγώνα ποδοσφαίρου, περισσότερους από

11 παίκτες). Η λειτουργικότητα αυτή παρέχεται από το GraphOnto, το οποίο υποστηρίζει, κατά τη διάρκεια του υπομνηματισμού, την επικύρωση της ορθότητας (validation) τόσο των οντολογιών όσο και των μεταδεδομένων που ορίζονται με βάση αυτές.

Στη συνέχεια, οι OWL/RDF περιγραφές μετατρέπονται, χρησιμοποιώντας το κατάλληλο σύνολο κανόνων μετασχηματισμού (transformation rules) που υλοποιούνται στο GraphOnto, σε MPEG-7/21 περιγραφές. Οι περιγραφές αυτές υπακούουν, κατά περίπτωση, στην τρέχουσα μορφή των προτύπων ή σε προτεινόμενες επεκτάσεις της που απεικονίζονται σε κάποιες από της οντολογίες εφαρμογών, καλύπτουν της απαιτήσεις συγκεκριμένων εφαρμογών και βασίζονται σε άλλα πρότυπα ή σε ευρέως αποδεκτά μοντέλα.

Οι MPEG-7/21 περιγραφές αποθηκεύονται στη DS-MIRF Αποθήκη Μεταδεδομένων MPEG-7/21 (DS-MIRF MPEG-7/21 Metadata Repository), η οποία παρέχει ένα σύνολο εξατομικευμένων, βασισμένων σε γνώση περιοχής υπηρεσιών για αναζήτηση, ανάκτηση, διανομή, παρουσίαση και φιλτράρισμα περιεχομένου πολυμέσων. Οι υπηρεσίες ανάκτησης βασίζονται στη γλώσσα MP7QL, που αναπτύχθηκε στα πλαίσια της παρούσας διατριβής και επιτρέπει ερωτήσεις σε όλες τις απόψεις των MPEG-7 περιγραφών, ενώ η εξατομίκευση των υπηρεσιών και οι υπηρεσίες φίλτρων βασίζονται στο μοντέλο προτιμήσεων χρηστών που αναπτύχθηκε στα πλαίσια της διατριβής. Οι υπηρεσίες που παρέχονται από τη DS-MIRF Αποθήκη Μεταδεδομένων MPEG-7/21 χρησιμοποιούνται στη συνέχεια από εφαρμογές που επικοινωνούν με της χρήστες μέσω διεπαφών εφαρμογής (application interfaces), όπως για παράδειγμα η βασισμένη σε οντολογίες γεννήτρια διεπαφών φυσικής γλώσσας για αποθήκες οπτικοακουστικού υλικού OntoNL [36].

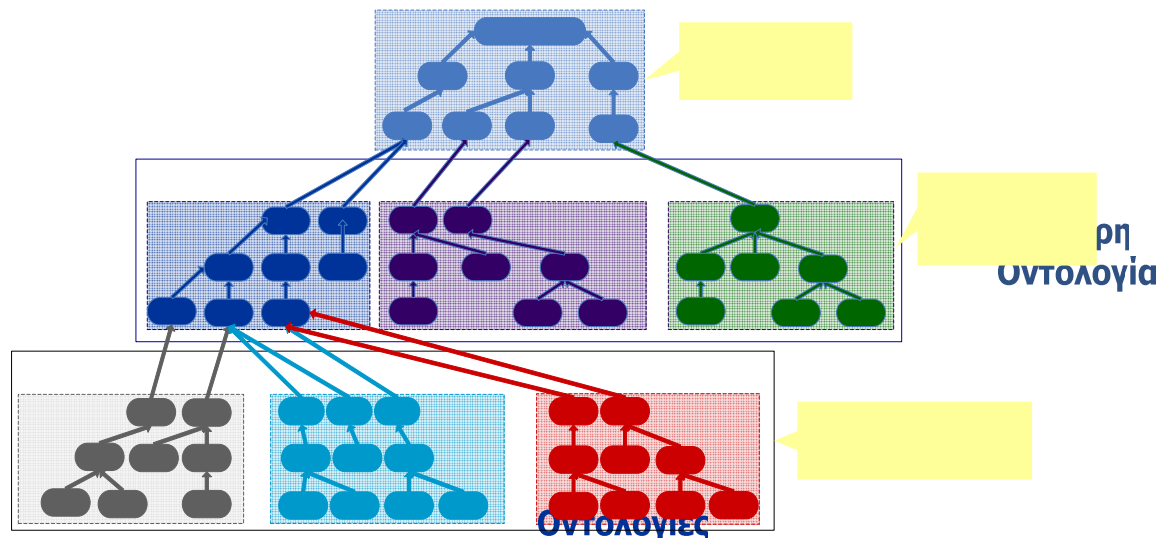
4.4. Συστατικά του Πλαισίου DS-MIRF

4.4.1. Εισαγωγή

Στην ενότητα αυτή γίνεται μια σύντομη αναφορά στα συστατικά του πλαισίου DS-MIRF. Πιο συγκεκριμένα, στην υποενότητα 4.4.2 περιγράφεται η οντολογική υποδομή του πλαισίου DS-MIRF, στην υποενότητα 4.4.3 παρουσιάζεται ο ρόλος του GraphOnto μέσα στο πλαίσιο και στην υποενότητα 4.4.4 παρουσιάζεται η DS-MIRF αποθήκη MPEG-7/21 μεταδεδομένων.

4.4.2. Οντολογική Υποδομή του Πλαισίου DS-MIRF

Σ' αυτή την υποενότητα περιγράφεται σύντομα η οντολογική υποδομή του πλαισίου DS-MIRF, η οποία απεικονίζεται στην επόμενη εικόνα:



Εικόνα 45: Η Οντολογική υποδομή του Πλαισίου DS-MIRF

Εφαρμογές
Τυποποιημένες
Σχέσεις

Η οντολογική υποδομή του πλαισίου DS-MIRF περιλαμβάνει:

- Μια OWL Ανώτερη Οντολογία που αποτυπώνει πλήρως τον MPEG-7 MDS και την MPEG-21 DIA και αποτελεί τον ακρογωνιαίο λίθο της οντολογικής υποδομής του πλαισίου DS-MIRF και τη βάση υποστήριξης διαλειτουργικότητας μεταξύ OWL και MPEG-7/21.
- Ένα σύνολο OWL Οντολογιών Εφαρμογών, που παρέχουν επιπλέον λειτουργικότητα σε OWL. Η λειτουργικότητα αυτή είτε διευκολύνει τη χρήση των προτύπων MPEG-7/21 (συνήθως περιλαμβάνει δομές που υπονοούνται στο κείμενο περιγραφής των προτύπων, όπως είναι οι τυποποιημένες σχέσεις – typed relationships) είτε υποστηρίζει προηγμένες (advanced) υπηρεσίες περιεχομένου πολυμέσων (όπως, για παράδειγμα, σημασιολογικές προτιμήσεις χρηστών – semantic user preferences). Οι οντολογίες εφαρμογών παρέχουν γενικού σκοπού δομές που είτε υπονοούνται στο κείμενο των προτύπων MPEG-7/21 είτε απουσιάζουν από αυτό.
- Τις Οντολογίες Περιοχής, που επεκτείνουν την ανώτερη οντολογία και τις οντολογίες εφαρμογών με γνώση περιοχής. Για παράδειγμα, οντολογίες σχετικές με αθλητισμό επεκτείνουν τις γενικευμένες δυνατότητες περιγραφής του MPEG-7 MDS στο πεδίο του αθλητισμού. Στα πλαίσια του DS-MIRF έχει αναπτυχθεί μια μεθοδολογία ενσωμάτωσης οντολογιών περιοχής στην οντολογική υποδομή του πλαισίου και έχουν οριστεί, για τον έλεγχο της ορθότητας της μεθοδολογίας, οντολογίες για ποδόσφαιρο και Formula 1.

4.4.3. Το Συστατικό Λογισμικού GraphOnto

Σ' αυτή την υποενότητα παρουσιάζεται ο ρόλος του GraphOnto συστήματος που επιτρέπει τον ορισμό και τη διαχείριση οντολογιών και μεταδεδομένων που ορίζονται με βάση

οντολογίες και επιτρέπει την εκμετάλλευση και επέκταση της οντολογικής υποδομής του πλαισίου DS-MIRF.

Καθώς οι οντολογίες μπορεί να περιγράφουν ιδιαίτερα περίπλοκες έννοιες, το GraphOnto διαθέτει μηχανισμούς απόκρυψης πληροφορίας (information hiding), ώστε οι υπομνηματιστές να μπορούν να εστιάσουν μόνο στις κλάσεις και τις ιδιότητες των κλάσεων με τις οποίες σκοπεύουν να εργαστούν. Το GraphOnto βασίζεται στην OWL, που είναι το επικρατέστερο πρότυπο αναπαράστασης οντολογιών και μπορεί να χρησιμοποιηθεί τόσο ως ένας γενικού σκοπού συντάκτης (editor) OWL οντολογιών όσο και ως συστατικό σημασιολογικού υπομνηματισμού για υλικό πολυμέσων.

Ως συστατικό σημασιολογικού υπομνηματισμού για υλικό πολυμέσων, το GraphOnto χρησιμοποιείται στο πλαίσιο DS-MIRF. Συγκεκριμένα, το GraphOnto επιτρέπει τόσο το σημασιολογικό υπομνηματισμό αντικειμένων πολυμέσων με τη χρήση της οντολογικής υποδομής του πλαισίου DS-MIRF (ανώτερη οντολογία, οντολογίες εφαρμογών και οντολογίες περιοχής) όσο και τον εμπλουτισμό της με νέες οντολογίες περιοχής και εφαρμογών. Τόσο οι οντολογίες περιοχής όσο και οι σημασιολογικές περιγραφές που ορίζονται με τη χρήση τους μπορούν να μετατραπούν σε MPEG-7 περιγραφές από το GraphOnto, με τη χρήση των κανόνων μετασχηματισμού του πλαισίου DS-MIRF. Οι περιγραφές αυτές μπορούν να αποθηκευτούν είτε σε XML αρχεία είτε στην DS-MIRF αποθήκη MPEG-7 μεταδεδομένων.

Τέλος, το GraphOnto API χρησιμοποιείται από διεπαφές εφαρμογών που βασίζονται σε οντολογίες και επικοινωνούν με την DS-MIRF αποθήκη MPEG-7 μεταδεδομένων.

4.4.4. Η DS-MIRF Αποθήκη MPEG-7/21 Μεταδεδομένων

Στην υποενότητα αυτή παρουσιάζεται συνοπτικά η DS-MIRF αποθήκη MPEG-7/21 μεταδεδομένων (MPEG-7/21 Repository [37]), η οποία επιτρέπει τη διαχείριση MPEG-7/21 περιγραφών οπτικοακουστικού υλικού μέσα από ένα σύνολο υπηρεσιών (services) εξατομικευμένων και βασισμένων σε γνώση περιοχής. Συγκεκριμένα, παρέχονται οι παρακάτω υπηρεσίες που επιτρέπουν:

- Την εισαγωγή MPEG-7 περιγραφών οπτικοακουστικού υλικού στην αποθήκη.
- Την εισαγωγή στην αποθήκη οντολογιών οι οποίες εκφράζονται σε MPEG-7 σύνταξη και στις οποίες μπορεί να βασίζονται οι MPEG-7 περιγραφές.
- Την ανάκτηση των MPEG-7 περιγραφών οπτικοακουστικού υλικού μέσω MP7QL ερωτήσεων. Η γλώσσα MP7QL είναι γλώσσα ερωτήσεων για MPEG-7 περιγραφών που επιτρέπει την έκφραση ερωτήσεων για όλα τα στοιχεία (elements) και τα χαρακτηριστικά (attributes) των MPEG-7 περιγραφών, καθώς και σημασιολογικές ερωτήσεις που βασίζονται σε οντολογίες.
- Την ενημέρωση και διαγραφή MPEG-7 περιγραφών οπτικοακουστικού υλικού οι οποίες βρίσκονται αποθηκευμένες στην αποθήκη δεδομένων.

Οι παραπάνω υπηρεσίες παρέχονται με τη μορφή υπηρεσιών του Παγκόσμιου Ιστού (Web Services) και η ανάπτυξη της αποθήκης δεδομένων βασίζεται στην XML βάση δεδομένων Oracle Berkeley DB XML [38].

4.5. Ανακεφαλαίωση

Στο κεφάλαιο αυτό περιγράφηκε αναλυτικά το πλαίσιο DS-MIRF (Domain-Specific Multimedia Information and Filtering Framework) που υλοποιήθηκε με σκοπό τη διαχείριση μεταδεδομένων για οπτικοακουστικό υλικό μέσω ενός παρεχόμενου συνόλου υπηρεσιών που υποστηρίζουν την ανάπτυξη εφαρμογών πολυμέσων, βασίζονται σε γνώση περιοχής και χρησιμοποιούν και επεκτείνουν τα πρότυπα MPEG-7 και MPEG-21. Αναφέρθηκε η προσέγγιση που υλοποιεί το πλαίσιο DS-MIRF για την υποστήριξη σημασιολογικής διαλειτουργικότητας μεταξύ OWL και προτύπων που βασίζονται σε XML Schema, με σκοπό την εκμετάλλευση γνώσης περιοχής (domain knowledge) από εφαρμογές που βασίζονται στα πρότυπα. Ακολούθως, αναλύθηκε η αρχιτεκτονική του πλαισίου και τέλος παρουσιάστηκαν τα συστατικά του, ανάμεσα στα οποία βρίσκεται και το GraphOnto.

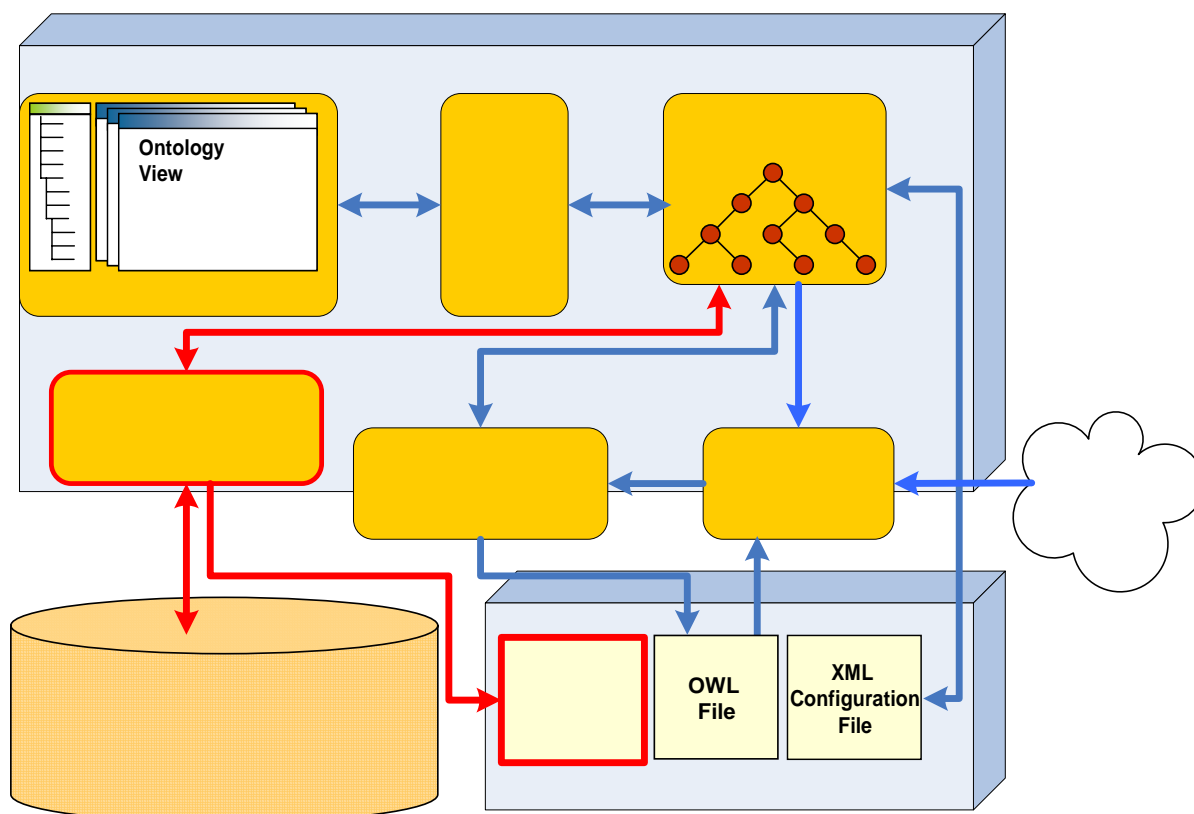
Κεφάλαιο 5^ο

5. Αρχιτεκτονική

5.1. Εισαγωγή

Όπως έχει ήδη αναφερθεί, οι ανάγκες που θα έπρεπε να ικανοποιήσει το GraphOnto είναι: α) η υλοποίηση ενός πλήρως λειτουργικού συστήματος διαχείρισης πολλαπλών ταυτόχρονα OWL οντολογιών που θα παρέχει παράλληλα εξατομίκευση στον χρήστη, β) η ενσωμάτωση αυτού του συστήματος μέσα σε ένα περιβάλλον διαχείρισης οπτικό-ακουστικού υλικού που θα υποστηρίζει το MPEG-7 μοντέλο και γ) η δυνατότητα επέκτασης του συστήματος σε άλλα περιβάλλοντα και η παροχή λειτουργικότητας επεξεργασίας οντολογιών σε εξωτερικές εφαρμογές.

Όλα τα παραπάνω, οδήγησαν στη δημιουργία μιας σύνθετης αρχιτεκτονικής αποτελούμενης από αρκετά ανεξάρτητα και αλληλοσυνδεδεμένα στοιχεία. Στο κεφάλαιο λοιπόν αυτό, παρουσιάζεται η αρχιτεκτονική του συστήματος της οποίας μια γενική άποψη φαίνεται στο παρακάτω σχήμα:



Εικόνα 46: Η Αρχιτεκτονική του GraphOnto συστήματος

Στις επόμενες ενότητες του κεφαλαίου, θα παρουσιαστούν αναλυτικά και ξεχωριστά τα στοιχεία της αρχιτεκτονικής ώστε να γίνει κατανοητός ο ρόλος τους και ο τρόπος σύνδεσής

τους με τα υπόλοιπα μέρη του συστήματος, ενώ στο τέλος του γίνεται μια αναφορά στα εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση του.

5.2. GraphOnto User Interface

5.2.1. Εισαγωγή

Το GraphOnto User Interface είναι το μέρος του συστήματος που αλληλεπιδρά με τους χρήστες της εφαρμογής, δηλαδή τους δημιουργούς των οντολογιών και των μεταδεδομένων. Πρόκειται για ένα απλό και φιλικό γραφικό περιβάλλον αλληλεπίδρασης με τους χρήστες το οποίο παρέχει πλήρη λειτουργικότητα για διαχείριση OWL-DL οντολογιών, δημιουργία μεταδεδομένων βασισμένων σε οντολογίες, μετασχηματισμό των OWL περιγραφών σε MPEG-7 περιγραφές και αποθήκευση τους σε μια MPEG-7 αποθήκη μεταδεδομένων.

5.2.2. Βασικές Αρχές Σχεδιασμού του GraphOnto User Interface

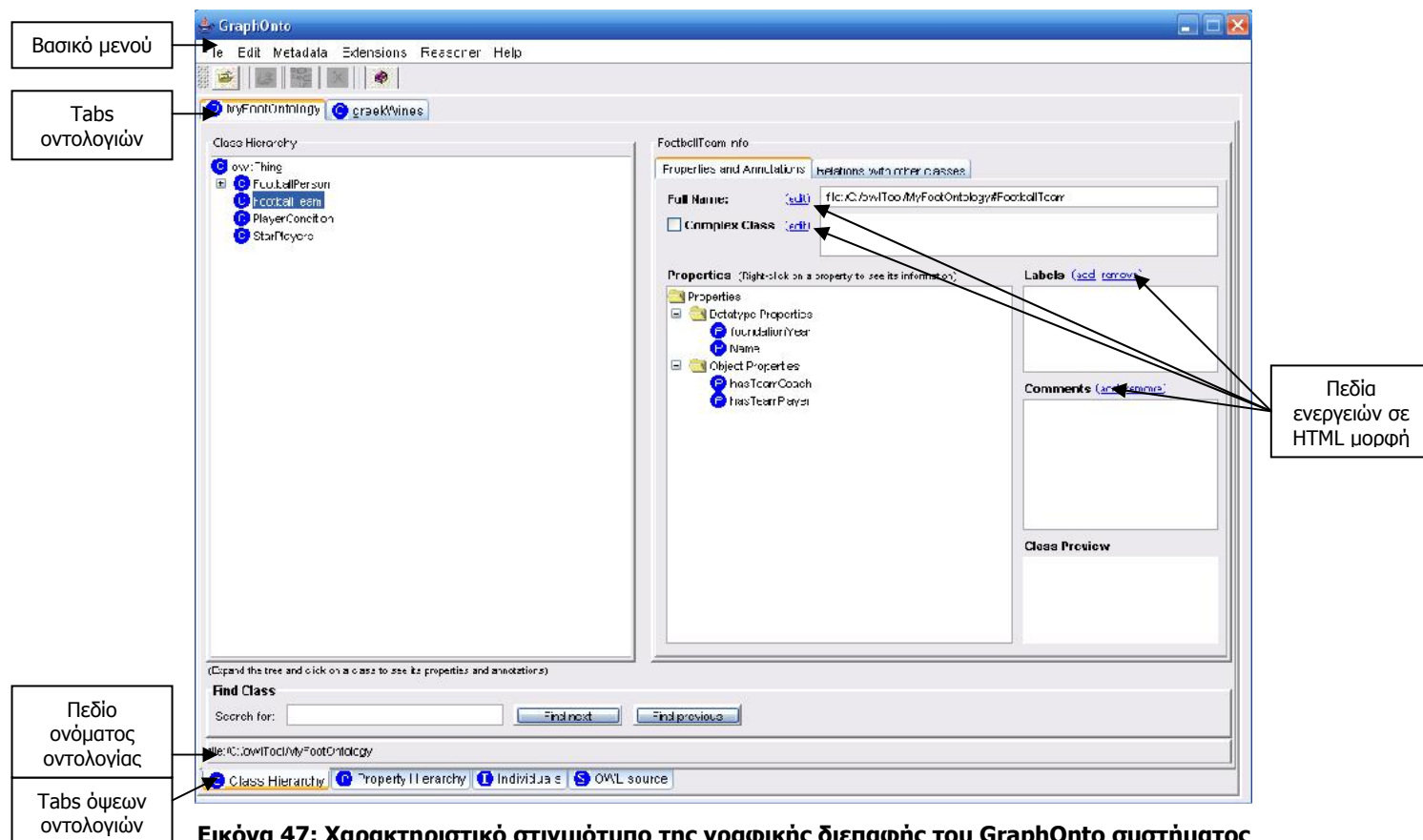
Κατά το σχεδιασμό του GraphOnto User Interface έγινε προσπάθεια να εφαρμοστούν οι παρακάτω βασικές αρχές σχεδιασμού γραφικών διεπαφών με το χρήστη:

1. *Ορατότητα της τρέχουσας κατάστασης του συστήματος (Visibility of System Status)*: Το γραφικό περιβάλλον που υλοποιήθηκε ενημερώνει διαρκώς τον χρήστη σχετικά με το που βρίσκεται αφού σε κάθε παράθυρο της εφαρμογής που είναι ανοιχτό, υπάρχει η κατάλληλη επικεφαλίδα. Επιπλέον, υπάρχουν τίτλοι για κάθε ανοιχτή οντολογία για την εύκολη αναγνώριση της τρέχουσας οντολογίας.
2. *Ομοιότητα μεταξύ του συστήματος και του πραγματικού κόσμου (Match Between System and the Real World)*: Κατά το σχεδιασμό του γραφικού περιβάλλοντος λήφθηκε υπόψιν το κοινό στο οποίο αυτό απευθύνεται η εφαρμογή και έτσι χρησιμοποιήθηκαν όροι οικείοι και γνωστοί από την OWL και το MPEG-7.
3. *Έλεγχος στα χέρια του χρήστη και ελευθερία χρήσης (User Control and Freedom)*: Ο χρήστης έχει συνεχώς τον έλεγχο της εφαρμογής και στις περισσότερες περιπτώσεις έχει την ελευθερία να ενεργήσει με τον τρόπο και τη σειρά που θέλει ο ίδιος, χωρίς να είναι δεσμευμένος να ακολουθήσει μια ακριβή και συγκεκριμένη σειρά ενεργειών. Παραδείγματος χάρη, στη δημιουργία μιας οντολογίας δεν είναι απαραίτητος ο ορισμός των OWL δομών με τη σειρά: κλάσεις, ιδιότητες, άτομα καθώς ο χρήστης μπορεί ανά πάσα στιγμή να συμπληρώσει πληροφορία στις OWL δομές. Από την άλλη πλευρά, η ελευθερία που παρέχεται δεν είναι ασύδοτη λόγω του κινδύνου απώλειας δεδομένων και ενεργειών (σε αυτή την περίπτωση ο χρήστης ενημερώνεται από ανάλογα απαγορευτικά μηνύματα).

4. *Συνέπεια και πρότυπα (Consistency and Standards)*: Σε όλη την εφαρμογή έγινε προσπάθεια να ακολουθηθεί η ίδια λογική σχεδιασμού (π.χ. ίδια χρώματα , κοινή ονομασία κουμπιών που εκτελούν την ίδια ενέργεια). Επίσης έγινε προσπάθεια ώστε να ακολουθεί τα πρότυπα άλλων γνωστών εφαρμογών προκειμένου να μη φανεί εντελώς ξένα στους χρήστες.
5. *Αναγνώριση παρά ανάκληση από τη μνήμη (Recognition rather than Recall)*: Ο χρήστης δεν είναι υποχρεωμένος να θυμάται πληροφορία μεταξύ των ενεργειών που κάνει και έτσι το γραφικό περιβάλλον έχει, όπως αναφέρθηκε και προηγουμένως, τίτλο σε κάθε παράθυρο που ανοίγει. Παράλληλα, υπάρχουν μικρές ετικέτες δίπλα σε κάποια πεδία οι οποίες παρέχουν κατάλληλη βοήθεια σχετικά με την επόμενη ενέργεια του χρήστη. Με αυτόν τον τρόπο μειώνονται οι πιθανότητες όπου ο χρήστης θα αναγκαστεί να επιστρέψει (recall) για να κατανοήσει πως πρέπει να δράσει στη συνέχεια.
6. *Πρόληψη λαθών (Error Prevention)*: Για να αποφευχθούν έγκαιρα και χωρίς ιδιαίτερα προβλήματα οι τυχόν λανθασμένες εισαγωγές ή ενέργειες από τον χρήστη, υπάρχουν κατάλληλες επεξηγήσεις για κάθε κουμπί και κάθε πεδίο κειμένου. Επίσης, οι λίστες επιλογής στοιχείων περιέχουν μόνο τις σωστά πιθανές επιλογές ώστε να προλαμβάνονται οι μη έγκυρες εισαγωγές και η εκ των υστέρων διόρθωσή τους.
7. *Ευελιξία και αποτελεσματικότητα χρήσης (Flexibility and Efficiency of Use)*: Το σύστημα απευθύνεται τόσο σε έμπειρους χρήστες όσο και σε λιγότερο έμπειρους ή ακόμα και αρχάριους επιτρέποντας τη ταυτόχρονη εξυπηρέτηση και των δύο ομάδων χρηστών, δίχως η ύπαρξη της μιας να συνεπάγεται καθυστέρηση στις ενέργειες της δεύτερης. Για λόγους απόδοσης έχει αποφευχθεί η χρήση μεγάλων σε μέγεθος παράθυρων (frames) καθώς και γραμμών κύλισης (scrollbars) σε πεδία που συνήθως περιέχουν μικρή ποσότητα πληροφορίας καθώς υπάρχει κίνδυνος αποπροσανατολισμού του χρήστη. Βέβαια αν η ποσότητα πληροφορίας σε ένα πεδίο υπερβαίνει τη συνηθισμένη, τότε οι γραμμές κύλισης χρησιμοποιούνται αναγκαστικά καθώς δεν είναι ούτε σχεδιαστικά σωστή ούτε πρακτικά δυνατή η αλλαγή μεγέθους των πεδίων.
8. *Αισθητικός και μινιμαλιστικός σχεδιασμός (Aesthetic and Minimalistic Design)*: Στο χρήστη δεν παρέχονται περιττές πληροφορίες και μακροσκελείς επεξηγήσεις διότι αυτό θα τον εμποδίσει να δει την πληροφορία που ουσιαστικά πρέπει να του προσφερθεί. Ακόμη και στην περίπτωση των λαθών, η επισήμανση αυτών γίνεται με τρόπο σύντομο ώστε να φαίνεται ξεκάθαρα το μήνυμα που πρέπει να δοθεί στο χρήστη. Επιπλέον, έχει επιλεγεί τα πεδία να έχουν σχετικά μικρό μέγεθος και οι ετικέτες σχετικά μικρή γραμματοσειρά για αισθητικούς λόγους.
9. *Βοήθεια στο χρήστη για αναγνώριση, διάγνωση και ανάνηψη από λάθη (Help Users Recognize, Diagnose and Recover from Errors)*: Τα μηνύματα λάθους είναι λακωνικά, εύκολα κατανοητά και απόλυτα ακριβή έτσι ώστε να αντιλαμβάνεται αμέσως ο χρήστης τα λάθη του και να μπορεί να τα διορθώσει εύκολα και γρήγορα.

5.2.3. Περιγραφή του GraphOnto User Interface

Στην παράγραφο αυτή θα γίνει μια σύντομη περιγραφή όσον αφορά τη φιλοσοφία που ακολουθήθηκε για το γραφικό σχεδιασμό της εφαρμογής της οποίας ένα χαρακτηριστικό στιγμιότυπο φαίνεται στην επόμενη εικόνα.



Εικόνα 47: Χαρακτηριστικό στιγμιότυπο της γραφικής διεπαφής του GraphOnto συστήματος όταν τρέχει σε περιβάλλον Windows.

- Για το σχεδιασμό της διεπαφής της εφαρμογής χρησιμοποιήθηκαν κλάσεις από τις βιβλιοθήκες (packages) Swing και AWT (Abstract Windowing Toolkit) της Java.
- Η εμφάνιση της εφαρμογής εξαρτάται από την εμφάνιση του λειτουργικού συστήματος στο οποίο τρέχει. Έτσι, όταν η εφαρμογή τρέχει σε περιβάλλον Windows προσομοιάζει εμφάνιση Windows εφαρμογής, ενώ όταν τρέχει σε περιβάλλον Linux προσομοιάζει εμφάνιση Linux εφαρμογής. Επακόλουθο αυτής της δυνατότητας προσαρμογής στο τρέχον περιβάλλον είναι ο μη ορισμός καθορισμένου χρώματος φόντου (background color), αφού αυτό διαφοροποιείται ανάλογα με τις ρυθμίσεις φόντου του λειτουργικού συστήματος.
- Προτιμήθηκε η χρήση των καρτέλων (tabs) τόσο για την γραφική απεικόνιση κάθε ανοιχτής οντολογίας (καρτέλες οντολογιών) όσο και κάθε όψης οντολογίας (καρτέλες όψεων οντολογίας) ώστε να είναι εύκολη και γρήγορη η εναλλαγή ανάμεσα στις οντολογίες και τις όψεις οντολογίας.

- Προτιμήθηκε η χρήση πλαισίων (panels) μέσα σε κάθε καρτέλα έτσι ώστε να είναι ομαδοποιημένα τα σχετικά πεδία και ταυτόχρονα να ξεχωρίζουν από τα υπόλοιπα μη σχετικά.
- Προτιμήθηκε η αριστερή στοίχιση όσον αφορά την τοποθέτηση των ετικετών των πεδίων. Αντίθετα, προτιμήθηκε η στοίχιση στο κέντρο για το κείμενο που περιέχεται στα κουμπιά.
- Χρησιμοποιήθηκε ένα βασικό μενού επιλογών το οποίο περιέχει τις περισσότερες βασικές λειτουργίες.
- Για τη γραφική αναπαράσταση των κλάσεων, ιδιοτήτων και ατόμων προτιμήθηκε η χρήση δένδρων (trees) ώστε να καταδεικνύονται και οι ιεραρχικές σχέσεις μεταξύ των παραπάνω δομών.
- Σε κάθε δένδρο περιλαμβάνεται και ένα αναδυόμενο με τη χρήση του δεξί κλικ μενού που περιλαμβάνει τις λειτουργίες διαχείρισης των δομών που αναπαρίστανται στο δένδρο.
- Κάθε στοιχείο σε δένδρο ή λίστα που αντιστοιχεί σε OWL δομή, έχει πριν από το όνομα της δομής ένα εικονίδιο που καταδεικνύει το είδος της δομής. Ο πίνακας στο παράρτημα 1 καταγράφει το σύνολο των εικονιδίων που χρησιμοποιούνται από την εφαρμογή.
- Κάθε λίστα που περιέχει OWL δομές (κλάσεις, ιδιότητες, υποκείμενα) περιέχει και ένα αναδυόμενο με τη χρήση του δεξί κλικ μενού με το οποίο ο χρήστης μπορεί να μεταβεί άμεσα στην επιλεγμένη δομή της λίστας, ακόμα και αν χρειαστεί αλλαγή της τρέχουσας όψης της οντολογίας. Για παράδειγμα όταν η λίστα αναφέρεται σε πεδίο ορισμού ιδιότητας, τότε περιέχει κλάσεις και βρίσκεται στην όψη ιδιοτήτων. Συνεπώς αν ζητηθεί να εμφανιστεί η πληροφορία μιας από τις κλάσεις της λίστας, θα επιλεγεί η όψη κλάσεων ως τρέχουσα προκειμένου να εμφανιστεί η πληροφορία της επιλεγμένης κλάσης.
- Οι ετικέτες και οι τίτλοι των πλαισίων και των καρτέλων έχουν γραμματοσειρά "Dialog" με μέγεθος 11 και χρώμα μαύρο. Ανάλογα με τη σπουδαιότητα του κάθε στοιχείου ενδέχεται η γραμματοσειρά του να έχει έντονη (bold) μορφή. Από τα παραπάνω εξαιρούνται:
 - ο Οι ετικέτες που έχουν το ρόλο βοήθειας και των οποίων το μέγεθος γραμματοσειράς είναι 9.
 - ο Οι ετικέτες που αντιστοιχούν σε ιδιότητες ατόμων στις οποίες έχουν οριστεί λιγότερες τιμές από αυτές που ορίζουν οι περιορισμοί πολλαπλότητας τους, έχουν χρώμα γραμματοσειράς κόκκινο.
- Στα πεδία κειμένου το χρώμα φόντου είναι άσπρο και η γραμματοσειρά του κειμένου που πληκτρολογεί ο χρήστης, είναι "Dialog" με χρώμα μαύρο και μέγεθος 10.
- Αποφεύχθηκε η χρήση των παραδοσιακών κουμπιών στη διαχείριση (προσθήκη, διαγραφή, επεξεργασία) των χαρακτηριστικών των OWL δομών για μινιμαλιστικούς λόγους. Αντίθετα, δημιουργήθηκαν κατάλληλα και μικρότερα σε μέγεθος πεδία ενεργειών σε HTML μορφή στα οποία η κάθε ενέργεια διαχείρισης αναπαρίσταται ως HTML σύνδεσμος (link). Συγκεκριμένα, δημιουργήθηκαν δύο τύποι τέτοιων πεδίων:

- Πεδίο επεξεργασίας τιμής που περιέχει ως σύνδεσμο τη λέξη edit και χρησιμοποιήθηκε δίπλα σε πεδία κειμένου που δέχονται μία και μοναδική τιμή.
- Πεδίο πρόσθεσης/αφαίρεσης τιμής που περιέχει ως συνδέσμους τις λέξεις add και remove και χρησιμοποιήθηκε δίπλα σε λίστες που δέχονται περισσότερες από μία τιμές.
- Η εισαγωγή νέων τιμών σε χαρακτηριστικά των OWL δομών γίνεται με την εμφάνιση μικρών εμβόλιμων παράθυρων διαλόγου στα οποία ο χρήστης εισάγει την τιμή ή διαλέγει ανάμεσα στις πιθανές τιμές που προτείνονται από το σύστημα.
- Για την εμφάνιση της OWL μορφής κειμένου της κάθε οντολογίας προτιμήθηκε HTML μορφή και όχι απλή μορφή κειμένου. Έτσι, τα rdf:ID των δομών της οντολογίας αναπαρίστανται ως HTML σύνδεσμοι διευκολύνοντας την εύρεση της πληροφορίας που αντιστοιχεί σε κάθε δομή.
- Η διαχείριση των λαθών του χρήστη γίνεται με την τεχνική των alerts, που συνίσταται σε μικρά εμβόλιμα παράθυρα διαλόγου με ένα απαγορευτικό στρογγυλό σήμα και κείμενο που εξηγεί το λάθος και τον τρόπο διόρθωσης. Χρησιμοποιείται απλή, κατανοητή και με ευγενικό ύφος γλώσσα έτσι ώστε ο χρήστης να αντιλαμβάνεται άμεσα το λάθος του και ταυτόχρονα να μην προσβάλλεται.

5.3. User Interface Manager

Αποτελεί το μέρος της αρχιτεκτονικής του συστήματος το οποίο χειρίζεται το γραφικό σύστημα διεπαφής και επικοινωνεί με το εσωτερικό μοντέλο αναπαράστασης των οντολογιών. Έχει συνεπώς ένα διττό ρόλο:

1. Είναι υπεύθυνο για την ανανέωση και ενημέρωση του εσωτερικού μοντέλου αναπαράστασης των οντολογιών με βάση την είσοδο (input) του χρήστη (για παράδειγμα όταν ο χρήστης επεξεργάζεται μια OWL δομή ή προσθέτει μια καινούρια σε μια οντολογία).
2. Είναι παράλληλα υπεύθυνο για την προσαρμογή του γραφικού συστήματος διεπαφής με βάση το μοντέλο των ανοιχτών για επεξεργασία οντολογιών. Πιο συγκεκριμένα, αναλαμβάνει να τροποποιήσει τις δεντρικές μορφές και τις φόρμες του γραφικού συστήματος διεπαφής έτσι ώστε να συμφωνούν με τη σημασιολογία των ορισμένων OWL δομών (για παράδειγμα ανακατασκευάζει την ιεραρχία των κλάσεων όταν σε μια κλάση προστεθεί μια υποκλάση ή παρέχει για τον ορισμό Individual που ανήκει σε συγκεκριμένη κλάση μια σωστά δομημένη φόρμα η οποία επιτρέπει τον ορισμό μόνο εκείνων των ιδιοτήτων που προκύπτουν από τη σημασιολογία της συγκεκριμένης κλάσης).

5.4. Διαχειριστής Μοντέλων Οντολογιών

Ο Διαχειριστής Μοντέλων Οντολογιών (Ontology Model Manager) είναι το τμήμα της αρχιτεκτονικής που είναι υπεύθυνο για την εσωτερική αναπαράσταση των ανοιχτών και υπό επεξεργασία οντολογιών οι οποίες αναπαρίστανται με τη χρήση κατάλληλων δομών δεδομένων που περιέχονται μέσα στο GraphOnto API που θα περιγραφτεί παρακάτω.

Παράλληλα υλοποιεί έλεγχο για τη συνέπεια (consistency) των οντολογιών χωρίς τη χρήση κάποιας μηχανής εξαγωγής συμπερασμάτων (OWL Reasoner) και επιτρέπει στο χρήστη να τροποποιήσει τις δομές μιας οντολογίας μόνο στην περίπτωση που οι κανόνες και η σημασιολογία της OWL δεν παραβιάζονται. Ένα παράδειγμα αυτού του ελέγχου είναι ότι αν μια ιδιότητα P έχει πολλαπλότητα 1 όταν ανήκει σε μια κλάση X , τότε ο χρήστης δεν μπορεί να ορίσει δεύτερη τιμή για την ιδιότητα P όταν ορίζει άτομα που έχουν ως τύπο την κλάση X (όλοι οι έλεγχοι που γίνονται όσον αφορά τη συνέπεια της οντολογίας αναφέρονται με λεπτομέρεια στο επόμενο κεφάλαιο).

Τέλος, ο Διαχειριστής Μοντέλων Οντολογιών (ΔΜΟ) έχει τη δυνατότητα να διακρίνει τη διαφορά ανάμεσα σε ανώτερες οντολογίες και σε οντολογίες περιοχής. Ο διαχωρισμός των ανώτερων οντολογιών και των οντολογιών περιοχής μπορεί να γίνει είτε απευθείας από το χρήστη μέσω του γραφικού συστήματος διεπαφής είτε μέσω των GraphOnto XML αρχείων προσαρμογής (configuration files). Τα αρχεία αυτά είναι ουσιαστικά περιγραφές του περιβάλλοντος εργασίας του χρήστη καθώς καθορίζουν από πριν τις ανώτερες οντολογίες και τις οντολογίες περιοχής που θα χρησιμοποιηθούν από τον κάθε χρήστη (Αναλυτική περιγραφή των αρχείων αυτών υπάρχει στην παράγραφο 7.3). Ο ΔΜΟ είναι υπεύθυνος συνεπώς για το άνοιγμα και την ανάγνωση αυτού του είδους αρχείων και στη συνέχεια καλεί το WonderWeb OWL API για να γίνει το διάβασμα των οντολογιών που περιέχονται σε αυτά τα αρχεία. Επιπρόσθετα, έχει τη δυνατότητα να αποθηκεύσει την τρέχουσα επιλογή του χρήστη από ανώτερες οντολογίες και οντολογίες περιοχής σε ένα νέο GraphOnto XML αρχείο προσαρμογής.

5.5. WonderWeb OWL API

Το WonderWeb OWL API [39] [40] είναι ένα ελεύθερο εξωτερικό Java API (Application Programming Interface) που παρέχει προγραμματιστική πρόσβαση (μέσω Java κλάσεων με κατάλληλα χαρακτηριστικά και μεθόδους) σε δομές δεδομένων που αναπαριστούν όλες τις OWL δομές. Το API αυτό είναι προσανατολισμένο προς τις OWL Lite και OWL-DL και παρέχει τη δυνατότητα διαβάσματος οντολογιών που βρίσκονται στον παγκόσμιο ιστό ή είναι αποθηκευμένες σε τοπικά συστήματα αρχείων. Το συγκεκριμένο API χρησιμοποιήθηκε από το σύστημα μας για τη συντακτική ανάλυση (parsing) των OWL οντολογιών.

Παρουσιάζει αρκετά πλεονεκτήματα, με βασικότερο την κάλυψη της πλήρους εκφραστικότητας που παρέχει η OWL-DL. Ταυτόχρονα, χαρακτηρίζεται από αξιοπιστία καθώς κάνει σε γενικές γραμμές επιτυχημένα και πολύ γρήγορη συντακτική ανάλυση των OWL

οντολογιών (ελάχιστες ατέλειες παρουσιάστηκαν κατά τη χρήση του, όπως επίσης ελάχιστες ήταν κι αυτές που έχουν αναφερθεί από τους υπόλοιπους του χρήστες στον παγκόσμιο ιστό). Τέλος, είναι εφοδιασμένο με ένα αξιοπρεπές documentation που παρέχει μια ικανοποιητική βοήθεια σε κάθε χρήστη του.

Από την άλλη πλευρά, η αρχική έκδοση που επιλέχθηκε στερείται υποστήριξης για επεξεργασία και αποθήκευση οντολογιών (παρέχει μόνο `get()` μεθόδους), οπότε είναι κατάλληλο μόνο για ανάγνωση οντολογιών και για το λόγο αυτό επεκτείνεται από το GraphOnto API. Επιπλέον είναι κάπως δύσχρηστο από προγραμματιστικής άποψης καθώς όλη η πληροφορία που διαβάζεται από την εκάστοτε οντολογία αποθηκεύεται σε αταξινόμητα σύνολα από αγνώστου τύπου αντικείμενα και έτσι το συνεχές casting των αντικειμένων κρίνεται απαραίτητο.

5.6. GraphOnto API

Πρόκειται για ένα Java API που υλοποιήθηκε στα πλαίσια του GraphOnto και το οποίο παρέχει Java κλάσεις με τα κατάλληλα χαρακτηριστικά και μεθόδους για την υποστήριξη της πλήρους επεξεργασίας OWL οντολογιών. Αποτελείται από 13 Java κλάσεις που αντιπροσωπεύουν το σύνολο των δομών της OWL-DL (περισσότερες πληροφορίες βρίσκονται στο παράρτημα 3). Βασίζεται στο WonderWeb OWL API του οποίου οι κλάσεις μετασχηματίζονται στις αντίστοιχες κλάσεις του GraphOnto API για να υποστηριχθεί η επεξεργασία των OWL οντολογιών.

Το GraphOnto API αλληλεπιδρά με τον Ontology Model Manager προκειμένου από τη μία πλευρά να δημιουργηθεί το εσωτερικό μοντέλο αναπαράστασης μιας οντολογίας όταν ολοκληρωθεί το διάβασμα της με βάση τα java αντικείμενα που δημιουργήθηκαν και από την άλλη, να ενημερώνει τα java αντικείμενα που αναπαριστούν τα στοιχεία της οντολογίας όταν πραγματοποιούνται αλλαγές στο μοντέλο της οντολογίας από τον Ontology Model Manager. Το GraphOnto API έχει τη δυνατότητα να αποθηκεύει τις οντολογίες που διαχειρίζεται σε συμβατά OWL αρχεία στο τοπικό σύστημα αρχείων.

Τέλος, πρέπει να αναφερθεί ότι το GraphOnto API μπορεί να χρησιμοποιηθεί και εκτός του πλαισίου του GraphOnto καθώς αποτελεί και μια ανεξάρτητη βιβλιοθήκη που υποστηρίζει την προγραμματιστική πρόσβαση σε δομές διαχείρισης OWL οντολογιών. Κατά συνέπεια, μπορεί να ενσωματωθεί πολύ εύκολα και σε διάφορες άλλες εφαρμογές που χρειάζονται έτοιμες δομές για τη διαχείριση των οντολογιών.

5.7. Μετατροπές OWL σε MPEG-7

Ο Μετατροπές OWL σε MPEG-7 (OWL2MPEG-7 Converter) είναι το στοιχείο της αρχιτεκτονικής το οποίο επιτρέπει στο GraphOnto να μην χρησιμοποιείται μόνο σαν σύστημα διαχείρισης οντολογιών, αλλά να μπορεί να αποτελέσει και μια εφαρμογή που διευκολύνει τη σημασιολογική περιγραφή (annotation) οπτικό-ακουστικού υλικού στο πλαίσιο του DS-MIRF.

Η σημασιολογική περιγραφή οπτικό-ακουστικού υλικού μπορεί να επιτευχθεί με τον ορισμό OWL ατόμων τα οποία περιγράφουν το περιεχόμενο του οπτικό-ακουστικού υλικού. Όμως, αυτή η εξαγόμενη περιγραφή παρότι είναι πλήρης από σημασιολογική και οντολογική άποψη, δεν μπορεί να καλύψει τις ανάγκες ενός περιβάλλοντος περιγραφής οπτικό-ακουστικού υλικού. Αντίθετα, είναι αναγκαίο η εξαγόμενη περιγραφή να είναι συμβατή με κάποιο πρότυπο σημασιολογικής περιγραφής του περιεχομένου του οπτικό-ακουστικού υλικού, και επιλέχθηκε για το σκοπό αυτό το κυρίαρχο πρότυπο, MPEG-7.

Ο ρόλος συνεπώς του Μετατροπέα OWL σε MPEG-7 είναι να μετασχηματίσει τις σημασιολογικές περιγραφές που έχουν βασιστεί σε OWL σε σημασιολογικές περιγραφές βασισμένες σε MPEG-7. Με βάση αρκετούς κανόνες μετασχηματισμού που αναπτύχθηκαν στο πλαίσιο του DS-MIRF (που περιγράφονται στο επόμενο κεφάλαιο), το τελικό αποτέλεσμα είναι η εξαγωγή σημασιολογικών περιγραφών σε XML μορφή που υπακούν στο σχήμα του MPEG-7 MDS. Οι περιγραφές αυτές μπορούν να αποθηκευτούν είτε σε MPEG-7 τοπικά αρχεία είτε σε μια αποθήκη μεταδεδομένων συμβατή με το MPEG-7.

Ο Μετατροπέας OWL σε MPEG-7 αλληλεπικοινωνεί με τον ΔΜΟ αφού αφενός μετασχηματίζει τις OWL οντολογίες και τις υπόλοιπες OWL περιγραφές σε MPEG-7 περιγραφές και αφετέρου επιτρέπει στον ΔΜΟ να επαναχρησιμοποιήσει MPEG-7 περιγραφές που είναι αποθηκευμένες στην συμβατή με το MPEG-7 αποθήκη μεταδεδομένων μετατρέποντας τις σε απλά OWL άτομα.

5.8. MPEG-7 Αποθήκη Μεταδεδομένων

Το GraphOnto λειτουργεί διαδραστικά με την MPEG-7/21 αποθήκη μεταδεδομένων του DS-MIRF πλαισίου που υποστηρίζει την αποθήκευση και την ανάκτηση XML περιγραφών συμβατές με το MPEG-7 σχήμα (MPEG-7 Compliant XML Repository). Αυτή η αποθήκη μεταδεδομένων αποτελεί μια εφαρμογή η οποία αναπτύχθηκε έξω από τα πλαίσια της παρούσας μεταπτυχιακής διατριβής έχοντας ως βάση το Berkeley DB XML και η οποία, εκτός από την αποθήκευση MPEG-7 περιγραφών επιτρέπει την ανάκτηση αυτών των περιγραφών μέσω κατάλληλα εκφρασμένων σε γλώσσα XQuery σημασιολογικών ερωτήσεων. Αυτή η αποθήκη μεταδεδομένων προτιμήθηκε καθώς παρέχει γρηγορότερη και σταθερότερη πρόσβαση σε MPEG-7 περιγραφές που είναι αποθηκευμένες σε αυτή σε σχέση με περιγραφές που βρίσκονται αποθηκευμένες σε απλά αρχεία στο τοπικό σύστημα αρχείων.

Η αποθήκη μεταδεδομένων δεν αποτελεί μια εξωτερική εφαρμογή με την οποία θα έπρεπε να συνδέεται το GraphOnto, αλλά μια ενσωματωμένη (embedded) XML βάση δεδομένων μέσα στο GraphOnto σύστημα. Δεν απαιτεί τη χρήση ενός εξυπηρετητή βάσεων δεδομένων (database server) ο οποίος θα έπρεπε να τρέχει παράλληλα με το GraphOnto. Αντίθετα, οι μόνες προϋποθέσεις λειτουργίας της είναι:

- Η επιλογή ενός φακέλου (folder) στο τοπικό σύστημα αρχείων στον οποίο θα βρίσκονται τα απαραίτητα αρχεία που δημιουργεί αυτόματα η αποθήκη.

- Η ύπαρξη ορισμένων dll αρχείων του Berkeley DB XML σε κατάλληλο ανάλογο με το τρέχον λειτουργικό σύστημα φάκελο στο τοπικό σύστημα αρχείων (σε περιβάλλον Windows ο κατάλληλος φάκελος είναι ο "Windows/System 32").
- Η ενσωμάτωση των Java κλάσεων και μεθόδων που παρέχει το API της αποθήκης μεταδεδομένων στον κώδικα του GraphOnto.

Η αποθήκη μεταδεδομένων αποθηκεύει τις MPEG-7 περιγραφές σε κατάλληλες δομές που ονομάζονται Containers. Εξ'αρχής, υπάρχει ένας προκαθορισμένος (default container), ενώ παρέχεται η δυνατότητα δημιουργίας νέων. Κάθε container μπορεί να περιέχει πολυάριθμα έγγραφα. Η εισαγωγή ενός XML εγγράφου που περιέχει MPEG-7 περιγραφές σε ένα ορισμένο container της αποθήκης γίνεται μόνο εφόσον το έγγραφο είναι σωστά δομημένο (well-formed) και έγκυρο (valid) με βάση το MPEG-7 MDS Schema.

Η επικοινωνία του GraphOnto και συγκεκριμένα του Μετατροπέα OWL σε MPEG7 με την παραπάνω αποθήκη μεταδεδομένων είναι διπλής κατεύθυνσης προκειμένου να είναι δυνατή η αποθήκευση αφενός των παραγόμενων από το GraphOnto σημασιολογικών MPEG-7 περιγραφών στην αποθήκη μεταδεδομένων και η επαναχρησιμοποίηση αφετέρου των σημασιολογικών οντοτήτων που βρίσκονται στην αποθήκη μεταδεδομένων από το GraphOnto κατά τη διάρκεια της διαδικασίας δημιουργίας σημασιολογικής περιγραφής οπτικοακουστικού υλικού.

5.9. Εργαλεία που χρησιμοποιήθηκαν κατά την υλοποίηση

Τα εργαλεία που χρησιμοποιήθηκαν κατά την υλοποίηση του GraphOnto συστήματος είναι τα παρακάτω:

1. Η γλώσσα προγραμματισμού Java [41] με τη βοήθεια του προγραμματιστικού περιβάλλοντος JBuilder 2005 [42] και τη χρήση του Java Development Kit 1.4.2 [41], χρησιμοποιήθηκε για την πλήρη υλοποίηση τόσο του γραφικού περιβάλλοντος της εφαρμογής όσο και του GraphOnto API.
2. Το OWL API, ως γνωστόν, χρησιμοποιήθηκε για το διάβασμα των OWL οντολογιών και την αναπαράστασή τους σε Java κλάσεις.
3. Η εφαρμογή XMLBeans [43] για την παραγωγή Java κλάσεων που αντιστοιχούν στα στοιχεία του XML Schema των αρχείων προσαρμογής.
4. Χρησιμοποιήθηκαν ορισμένα .dll αρχεία της Oracle Berkeley DB XML προκειμένου να είναι δυνατή η επικοινωνία του GraphOnto με την MPEG-7 αποθήκη μεταδεδομένων.

5.10.Ανακεφαλαίωση

Στο κεφάλαιο αυτό περιγράφηκε η σύνθετη αρχιτεκτονική του GraphOnto συστήματος η οποία αποτελείται από αρκετά ανεξάρτητα και αλληλοσυνδεδεόμενα μεταξύ τους στοιχεία. Για κάθε στοιχείο της αρχιτεκτονικής παρουσιάστηκαν τα χαρακτηριστικά του και ο ρόλος του μέσα στο σύστημα, ενώ δόθηκε ιδιαίτερη βαρύτητα στις βασικές αρχές σχεδιασμού που ακολουθεί το γραφικό περιβάλλον του συστήματος και στα γνωρίσματά του.

Κεφάλαιο 6°

6. Το GraphOnto ως ένα σύστημα διαχείρισης OWL οντολογιών

6.1. Εισαγωγή

Σε αυτό το κεφάλαιο θα παρουσιάσουμε το GraphOnto ως ένα αμιγές σύστημα διαχείρισης OWL οντολογιών. Σε αυτό το πλαίσιο, το GraphOnto αποτελεί ένα γραφικό, βασισμένο σε φόρμες συντάκτη OWL οντολογιών ο οποίος μπορεί να χρησιμοποιηθεί για να δημιουργήσει ή να τροποποιήσει αυτόνομες (standalone) OWL-DL οντολογίες, παρέχοντας ταυτόχρονα αρκετά επιπλέον χαρακτηριστικά τα οποία διευκολύνουν και εξατομικεύουν το έργο του χρήστη και τα οποία διαφοροποιούν σημαντικά το GraphOnto από τις υπόλοιπες εφαρμογές διαχείρισης OWL οντολογιών. Όπως αναφέρθηκε και σε προηγούμενο κεφάλαιο, αποφασίστηκε η υποστήριξη της OWL-DL καθώς παρέχει πολύ μεγαλύτερη εκφραστικότητα σε σχέση με την OWL-Lite και εγγυάται την υπολογιστική περατότητα των μηχανών εξαγωγής συμπερασμάτων σε σχέση με την OWL-Full (από εδώ και στο εξής όταν αναφερόμαστε σε OWL, θα εννοούμε OWL-DL).

6.2. Λειτουργικότητα σε επίπεδο OWL οντολογίας

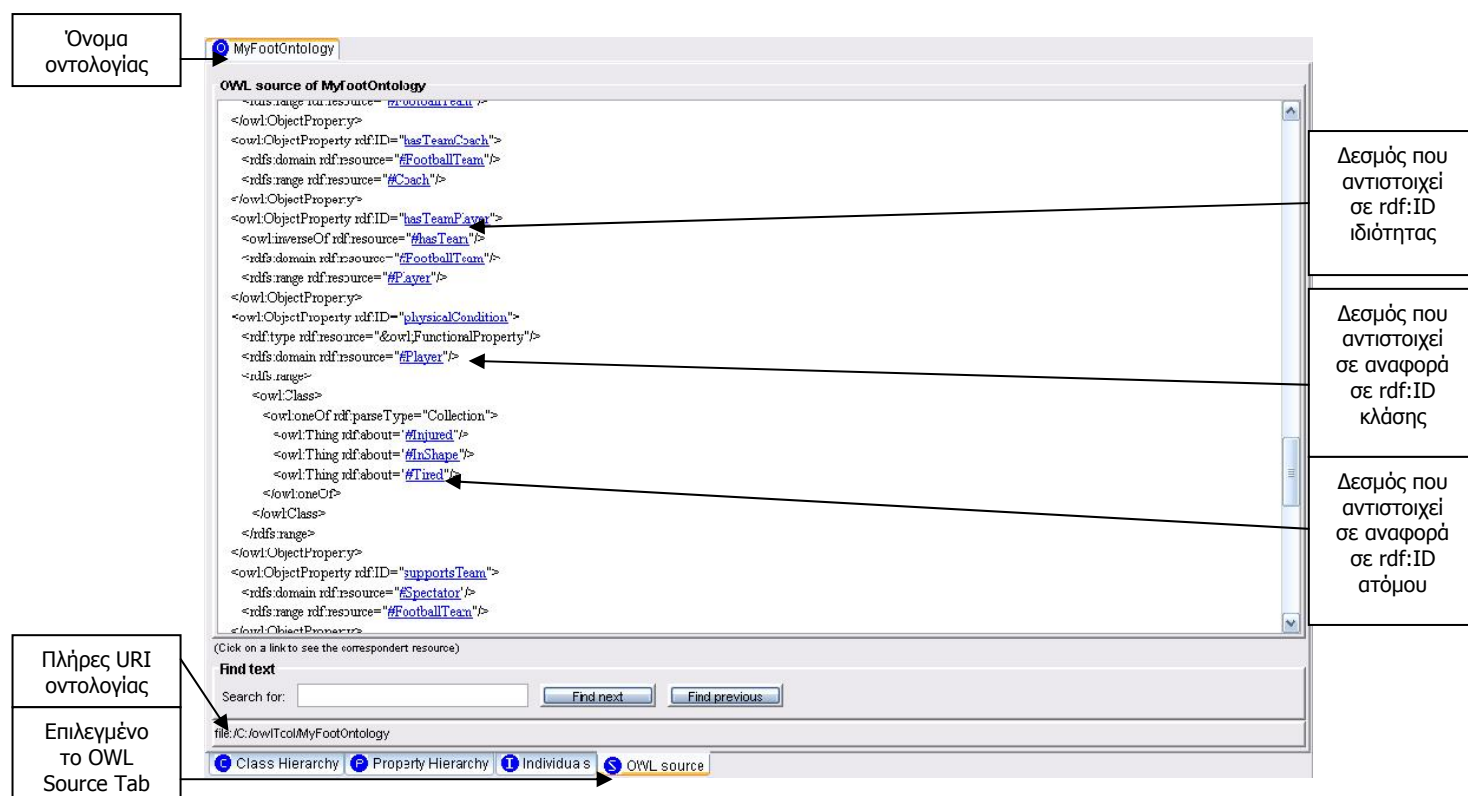
Το GraphOnto υποστηρίζει πλήρως τη γραφική επεξεργασία OWL οντολογιών. Το άνοιγμα των οντολογιών γίνεται με έναν από τους παρακάτω τρόπους:

- Δημιουργία νέας οντολογίας: Ο χρήστης δίνει το όνομα της οντολογίας και την τοποθεσία στο τοπικό σύστημα αρχείων όπου επιθυμεί να αποθηκευθεί η οντολογία ώστε να προκύψει το URI της.
- Διάβασμα και γραφική αναπαράσταση υπάρχουσας οντολογίας που είναι αποθηκευμένη στο τοπικό σύστημα αρχείων (local filesystem): Ο χρήστης αναζητάει και επιλέγει το αρχείο της οντολογίας μέσα στο τοπικό σύστημα αρχείων μέσω ενός περιηγητή αρχείων (file browser).
- Διάβασμα και γραφική αναπαράσταση υπάρχουσας οντολογίας που βρίσκεται σε κάποια τοποθεσία στον Παγκόσμιο Ιστό. Ο χρήστης σε αυτήν την περίπτωση εισάγει τη διεύθυνση (URL) της τοποθεσίας όπου βρίσκεται η οντολογία και εφόσον υπάρχει σύνδεση με το διαδίκτυο και όντως η διεύθυνση αυτή αντιστοιχεί σε OWL οντολογία επιτυγχάνεται άνοιγμα και διάβασμα της οντολογίας.

Πολλαπλές οντολογίες μπορούν να ανοιχθούν και να επεξεργάζονται ταυτόχρονα ώστε να διευκολύνεται ο ορισμός αντιστοιχίσεων μεταξύ των δομών τους. Για κάθε ανοιχτή οντολογία αντιστοιχεί μία καρτέλα (tab) η οποία έχει ως τίτλο το όνομα της και το εικονίδιο 1

(βλέπε παράρτημα 1) και το οποίο περιέχει στο κάτω μέρος του το πλήρες URI της οντολογίας. Η καρτέλα αυτή περιέχει τέσσερις εμφωλευμένες καρτέλες που η καθεμία περιλαμβάνει μία από τις παρακάτω τέσσερις διαφορετικές όψεις (views) που επιτρέπουν την περιήγηση (browsing) στην τρέχουσα οντολογία:

- Όψη κλάσεων (Class View) όπου η ιεραρχία των κλάσεων της τρέχουσας οντολογίας οπτικοποιείται μέσω ενός ιεραρχικού δένδρου στο οποίο οι υποκλάσεις κάθε κλάσης βρίσκονται σε μεγαλύτερο βάθος από αυτή.
- Όψη ιδιοτήτων (Property View) όπου η ιεραρχία των ιδιοτήτων της τρέχουσας οντολογίας οπτικοποιείται μέσω ενός ιεραρχικού δένδρου στο οποίο οι υποιδιότητες κάθε ιδιότητας βρίσκονται σε μεγαλύτερο βάθος από αυτή.
- Όψη ατόμων (Individuals View) όπου τα άτομα των κλάσεων της τρέχουσας οντολογίας οπτικοποιούνται μέσω ενός δένδρου στο οποίο τα άτομα ταξινομούνται ανάλογα με την κλάση στην οποία ανήκουν.
- Όψη OWL μορφής (OWL Source View) όπου εμφανίζεται η τρέχουσα OWL μορφή της τρέχουσας οντολογίας και η οποία οπτικοποιείται σε HTML μορφή και όχι σε απλή μορφή κειμένου. Μέσω της HTML μορφής όλα τα `rdf:ID` των δομών της τρέχουσας οντολογίας και οι αναφορές σε αυτά (η αναφορά σε `rdf:ID` γίνεται με τη χρήση του συμβόλου `#` μπροστά από το `rdf:ID`) αναπαριστούνται ως δεσμοί (links) οι οποίοι επιτρέπουν την εύκολη μετάβαση στα στοιχεία που αντιστοιχούν σε αυτούς. Οι ενέργειες που παρέχονται μέσω αυτών των δεσμών είναι:
 1. Επιλογή δεσμού που αντιστοιχεί σε `rdf:ID` ή σε αναφορά σε `rdf:ID` συγκεκριμένης κλάσης οδηγεί σε μετάβαση στην όψη κλάσεων όπου στο δένδρο ιεραρχίας κλάσεων είναι επιλεγμένη η συγκεκριμένη κλάση ώστε να παρουσιάζονται γραφικά τα χαρακτηριστικά της.
 2. Επιλογή δεσμού που αντιστοιχεί σε `rdf:ID` ή σε αναφορά σε `rdf:ID` συγκεκριμένης ιδιότητας οδηγεί σε μετάβαση στην όψη ιδιοτήτων όπου στο δένδρο ιεραρχίας ιδιοτήτων είναι επιλεγμένη η συγκεκριμένη ιδιότητα ώστε να παρουσιάζονται γραφικά τα χαρακτηριστικά της.
 3. Επιλογή δεσμού που αντιστοιχεί σε `rdf:ID` ή σε αναφορά σε `rdf:ID` συγκεκριμένου ατόμου οδηγεί σε μετάβαση στην όψη ατόμων όπου στο δένδρο ατόμων είναι επιλεγμένο το συγκεκριμένο άτομο ώστε να παρουσιάζονται γραφικά τα χαρακτηριστικά του.



Εικόνα 48: Η "OWL Source" καρτέλα στην οποία φαίνονται η OWL μορφή της τρέχουσας οντολογίας σε HTML μορφή.

Η επεξεργασία της οντολογίας περιλαμβάνει φυσικά τη δημιουργία, επεξεργασία και διαγραφή των OWL δομών που την απαρτίζουν. Το GraphOnto υποστηρίζει την πλήρη λειτουργικότητα που παρέχουν οι δομές της OWL-DL. Στις επόμενες ενότητες θα περιγραφεί αναλυτικά η προσφερόμενη λειτουργικότητα σε επίπεδο κλάσεων, ιδιοτήτων και ατόμων κλάσεων που παρέχεται μέσω των τριών πρώτων όψεων.

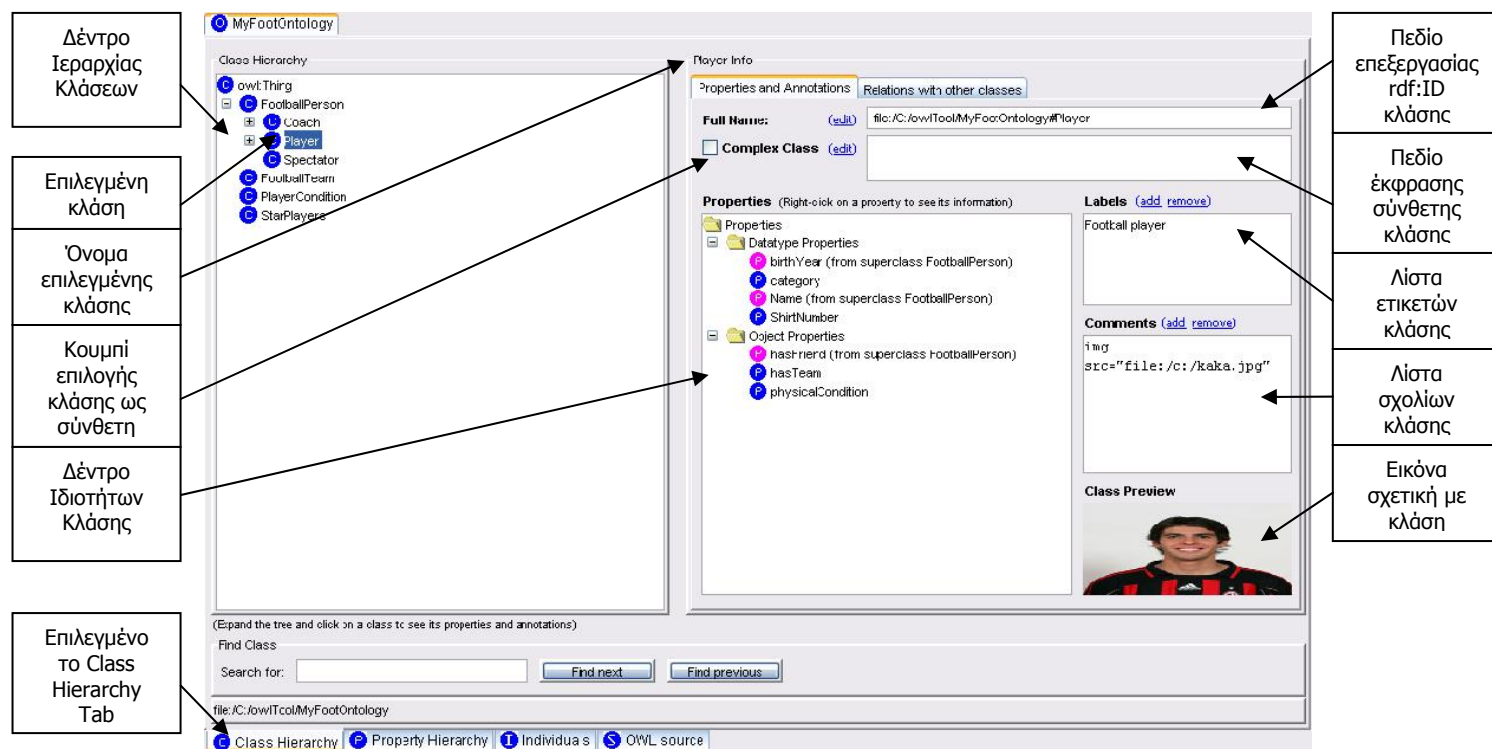
Οι επεξεργασμένες οντολογίες μπορούν να αποθηκευθούν στο τοπικό σύστημα αρχείων με έναν από τους δύο παρακάτω τρόπους:

- Αποθήκευση (Save Ontology) με το υπάρχον όνομα της οντολογίας στην υπάρχουσα τοποθεσία του τοπικού συστήματος αρχείων όπου βρίσκεται. Εφαρμόζεται μόνο για τις νεοδημιουργηθείσες οντολογίες ή για οντολογίες που ανοίχθηκαν από το τοπικό σύστημα αρχείων.
- Αποθήκευση (Save Ontology As) με τυχαίο όνομα και αποθήκευση σε τοποθεσία του τοπικού συστήματος αρχείων της επιλογής του χρήστη. Εφαρμόζεται σε όλες τις οντολογίες ανεξαρτήτως του τρόπου ανοίγματος τους.

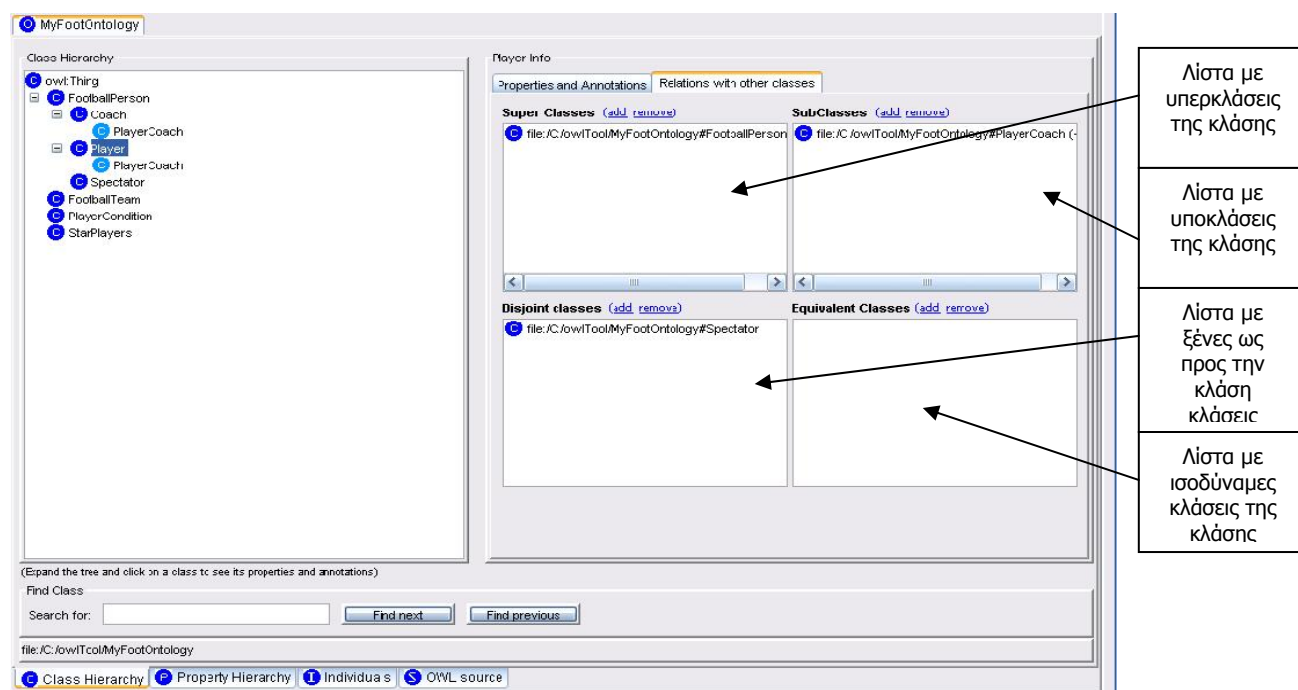
6.3. Λειτουργικότητα σε επίπεδο OWL κλάσεων

6.3.1. Δένδρο ιεραρχίας κλάσεων

Για τη δημιουργία σε μια οντολογία νέων κλάσεων και την επεξεργασία ή τη διαγραφή υπάρχοντων κλάσεων, το GraphOnto χρησιμοποιεί την όψη κλάσεων η οποία βρίσκεται μέσα στην καρτέλα "Class Hierarchy" η οποία φαίνεται στις δύο παρακάτω εικόνες:



Εικόνα 49: Η "Class Hierarchy" καρτέλα στην οποία φαίνονται αριστερά το δέντρο ιεραρχίας των κλάσεων της οντολογίας και δεξιά οι ιδιότητες και οι σχολιασμοί της επιλεγμένης κλάσης.



Εικόνα 50: Η "Class Hierarchy" καρτέλα στην οποία φαίνονται αριστερά το δέντρο ιεραρχίας των κλάσεων της οντολογίας και δεξιά οι σχέσεις της επιλεγμένης κλάσης με τις υπόλοιπες κλάσεις.

Όπως φαίνεται από τις παραπάνω εικόνες, για όλες τις κλάσεις της οντολογίας δημιουργούνται αντίστοιχοι κόμβοι που περιλαμβάνουν το `rdf:ID` της κλάσης και το εικονίδιο 3 (βλέπε παράρτημα 1) και οι οποίοι ταξινομούνται στο δένδρο ιεραρχίας κλάσεων. Η τοποθέτηση των κόμβων στο δένδρο γίνεται με βάση τους επόμενους κανόνες:

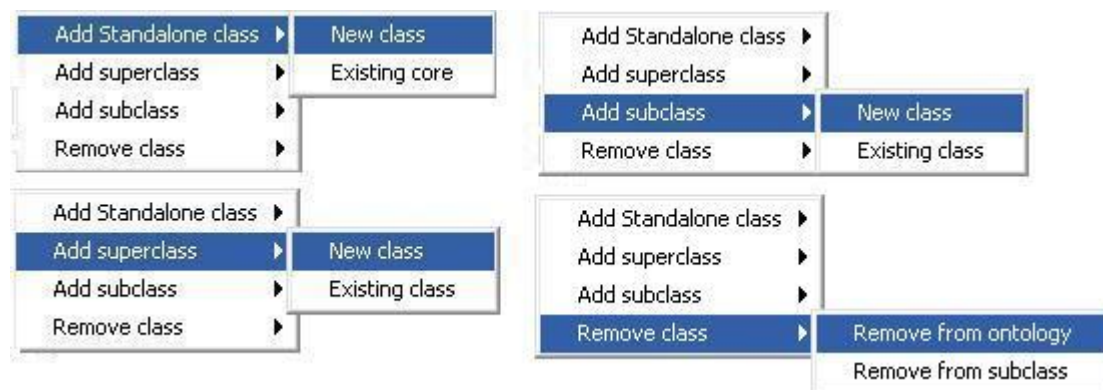
- Όλοι οι κόμβοι των κλάσεων τοποθετούνται κάτω από τη ρίζα του δένδρου που αναπαριστά το `owl:Thing` το οποίο θεωρείται η υπερκλάση όλων των OWL κλάσεων.
- Κάθε κόμβος κλάσης τοποθετείται σε μικρότερο βάθος σε σχέση με τους κόμβους των υποκλάσεων της και σε μεγαλύτερο βάθος σε σχέση με τους κόμβους των υπερκλάσεων της. Πρακτικά αυτό σημαίνει ότι ως παιδιά της ρίζας του δένδρου τοποθετούνται κόμβοι κλάσεων οι οποίες δεν έχουν καμία υπερκλάση.
- Για κάθε κλάση που έχει περισσότερες από μία υπερκλάσεις δημιουργούνται ισάριθμοι κόμβοι με τον αριθμό των υπερκλάσεων της. Καθένας από αυτούς τους κόμβους που περιλαμβάνουν το εικονίδιο 4 τοποθετείται κάτω από κάθε κόμβο που αντιστοιχεί σε υπερκλάση της κλάσης.
- Τα παιδιά κάθε κόμβου (συμπεριλαμβανομένης και της ρίζας του δένδρου) ταξινομούνται αλφαβητικά με βάση το `rdf:ID` της κλάσης στην οποία αντιστοιχούν.

Το δένδρο ιεραρχίας κλάσεων περιέχει ένα αναδυόμενο μενού (με τη χρήση του δεξιού κλικ του ποντικιού) παρέχοντας αρκετές και σημαντικές λειτουργίες όπως:

1. Εισαγωγή νέας αυτόνομης (που δεν έχει αρχικώς καμία υπερκλάση ή υποκλάση) κλάσης στην τρέχουσα οντολογία.
2. Εισαγωγή ως αυτόνομης μιας υπάρχουσας κλάσης που ανήκει όμως σε ανώτερη οντολογία (περισσότερες λεπτομέρειες για τις ανώτερες οντολογίες στην παράγραφο 6.2.10) και η οποία θα μπορεί να χρησιμοποιηθεί μόνο ως υπερκλάση κλάσεων της τρέχουσας οντολογίας.
3. Εισαγωγή νέας κλάσης στην τρέχουσα οντολογία η οποία θα αποτελέσει υπερκλάση της κλάσης που αντιστοιχεί στον επιλεγμένο κόμβο.
4. Εισαγωγή υπάρχουσας κλάσης της οντολογίας ως υπερκλάση της κλάσης που αντιστοιχεί στον επιλεγμένο κόμβο.
5. Εισαγωγή νέας κλάσης στην τρέχουσα οντολογία η οποία θα αποτελέσει υποκλάση της κλάσης που αντιστοιχεί στον επιλεγμένο κόμβο.
6. Εισαγωγή υπάρχουσας κλάσης της οντολογίας ως υποκλάση της κλάσης που αντιστοιχεί στον επιλεγμένο κόμβο.
7. Διαγραφή υπάρχουσας κλάσης από την τρέχουσα οντολογία.
8. Διαγραφή σε υπάρχουσα κλάση από το να είναι υποκλάση μιας υπερκλάσης της.

Οι λειτουργίες 3-8 απαιτούν από το χρήστη να έχει επιλέξει προηγουμένως έναν κόμβο του δένδρου και ο οποίος θα πρέπει να είναι διαφορετικός από τη ρίζα του δένδρου. Όλες οι παραπάνω ενέργειες μπορούν να επιλεγθούν και από το μενού "Edit" του βασικού

μενού της εφαρμογής. Οι τροποποιήσεις στην οντολογία που επιφέρουν οι ενέργειες αυτές έχουν ως αποτέλεσμα τη συνεχή αναδιάταξη του δένδρου ιεραρχίας κλάσεων, πάντα σύμφωνα με τους κανόνες που παρουσιάστηκαν προηγουμένως.



Εικόνα 51: Οι διαθέσιμες επιλογές του αναδυόμενου μενού του δένδρου ιεραρχίας κλάσεων.

Με την επιλογή ενός κόμβου, πλην της ρίζας φυσικά, εμφανίζονται οι πληροφορίες της αντίστοιχης κλάσης. Η εμφάνιση αυτών των πληροφοριών γίνεται μέσω δύο όψεων που περιγράφονται στις επόμενες παραγράφους:

- Όψη ιδιοτήτων και σχολιασμών επιλεγμένης κλάσης.
- Όψη σχέσεων της επιλεγμένης κλάσης με τις υπόλοιπες κλάσεις.

6.3.2. Όψη ιδιοτήτων και σχολιασμών επιλεγμένης κλάσης

Η *Όψη ιδιοτήτων και σχολιασμών* της επιλεγμένης κλάσης περιλαμβάνει τα παρακάτω στοιχεία:

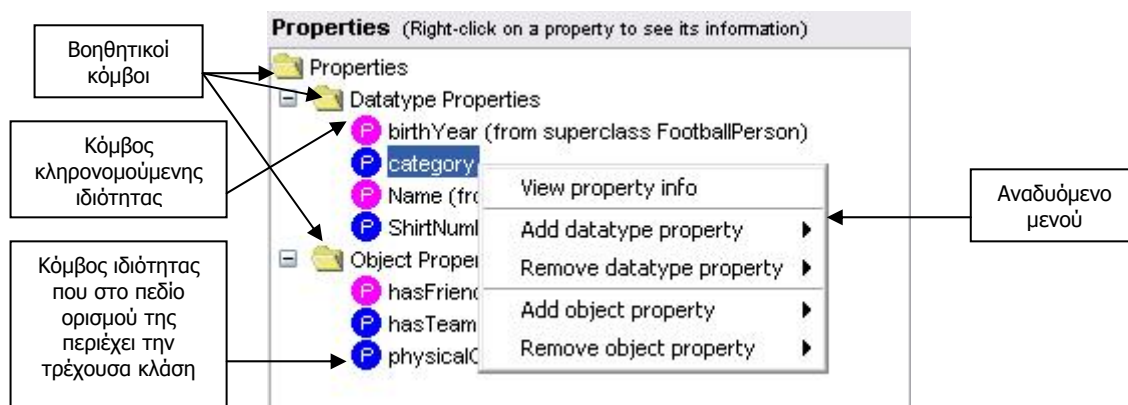
1. *Πεδίο επεξεργασίας rdf:ID κλάσης*: Το πεδίο αυτό περιέχει το πλήρες URI της επιλεγμένης κλάσης. Επιτρέπει μόνο την αλλαγή του rdf:ID και όχι του πλήρους URI της κλάσης. Κάθε πιθανή αλλαγή του rdf:ID μπορεί να επιφέρει αναδιάταξη κάποιων κόμβων του δένδρου ιεραρχίας κλάσεων, αφού αυτοί ταξινομούνται αλφαβητικά.
2. *Λίστα ετικετών κλάσης*: Η λίστα αυτή περιέχει τις ετικέτες που έχουν επισυναφθεί με την επιλεγμένη κλάση. Υποστηρίζεται η πρόσθεση νέας και η διαγραφή υπάρχουσας ετικέτας.
3. *Λίστα σχολίων κλάσης*: Η λίστα αυτή περιέχει τα σχόλια που έχουν επισυναφθεί με την επιλεγμένη κλάση. Υποστηρίζεται η πρόσθεση νέου και η διαγραφή υπάρχοντος σχολίου.
4. *Εικόνα σχετική με κλάση*: Πρόκειται για μια εικόνα που μπορεί να σχετίζεται με την επιλεγμένη κλάση. Η εικόνα αυτή εμφανίζεται μόνο σε περίπτωση ύπαρξης επισυνημμένου με την κλάση σχολίου το οποίο έχει τη μορφή `img src="imageLocation"` (όπου `imageLocation` είναι η διεύθυνση στο τοπικό σύστημα αρχείων ή στον παγκόσμιο ιστό στην οποία βρίσκεται η εικόνα).
5. *Δένδρο ιδιοτήτων κλάσης*: Το δένδρο αυτό περιέχει κόμβους που αντιστοιχούν στις ιδιότητες οι οποίες στο πεδίο ορισμού τους περιλαμβάνουν την τρέχουσα κλάση ή

υπερκλάσεις της τρέχουσας κλάσης (είναι προφανές ότι οι ιδιότητες μιας κλάσης εφαρμόζονται και στις υποκλάσεις της και συνεπώς μια κλάση κληρονομεί τις ιδιότητες κάθε υπερκλάσης της). Η πλούσια παρεχόμενη λειτουργικότητα μέσω αυτού του δένδρου περιγράφεται αναλυτικά στην παράγραφο 6.3.3.

6. *Κουμπί ορισμού κλάσης ως σύνθετη*: Το κουμπί αυτό παρέχει τη δυνατότητα ορισμού της επιλεγμένης κλάσης ως σύνθετη (ένωση ή τομή υπάρχοντων κλάσεων, συμπλήρωμα υπάρχουσας κλάσης ή απαρίθμηση υπάρχοντων ατόμων). Η λειτουργικότητα αυτή περιγράφεται στην παράγραφο 6.3.4.

6.3.3. Δένδρο ιδιοτήτων επιλεγμένης κλάσης

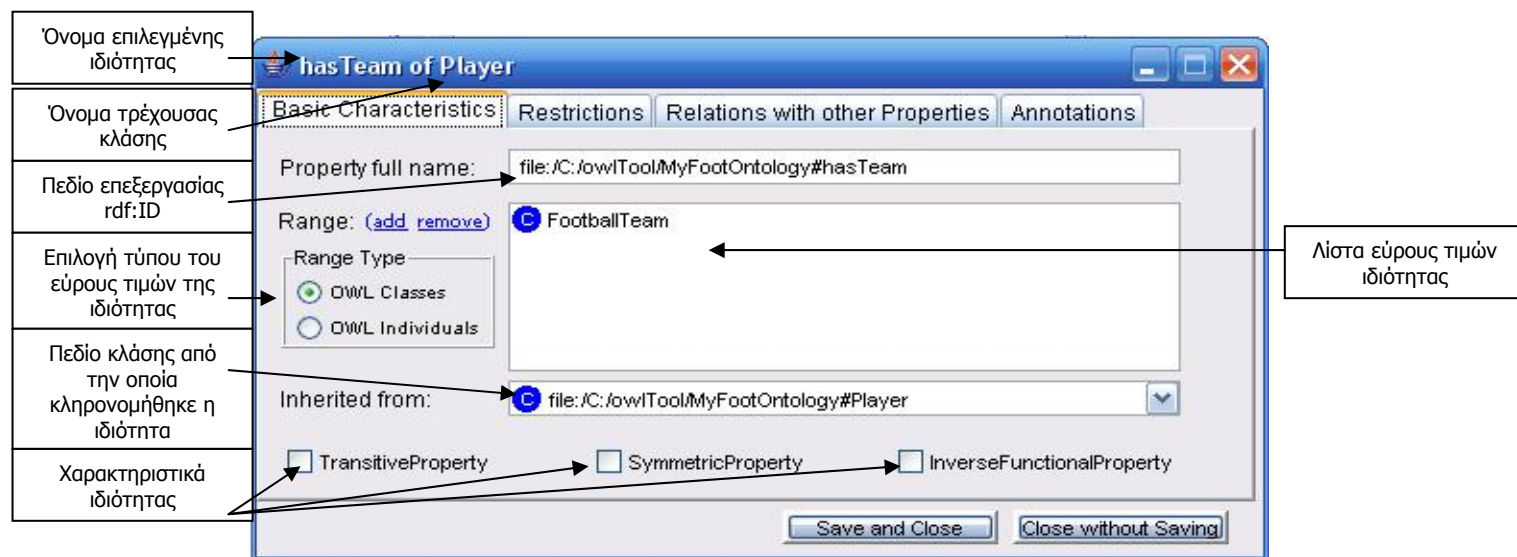
Στο *δένδρο ιδιοτήτων επιλεγμένης κλάσης*, οι ιδιότητες ομαδοποιούνται σε ιδιότητες τύπου δεδομένων και ιδιότητες αντικειμένων και ταξινομούνται αλφαβητικά σύμφωνα με το `rdf:ID` τους, αλλά βρίσκονται όλες στο ίδιο επίπεδο καθώς σε αυτό το δένδρο δεν είναι αναγκαία η ιεράρχηση τους. Κάθε κόμβος που αντιστοιχεί σε ιδιότητα που το πεδίο ορισμού της περιλαμβάνει την τρέχουσα κλάση περιέχει το εικονίδιο 6 και το `rdf:ID` της ιδιότητας, ενώ κάθε κόμβος που αντιστοιχεί σε ιδιότητα που έχει κληρονομηθεί από υπερκλάση της τρέχουσας κλάσης περιλαμβάνει το εικονίδιο 7, το `rdf:ID` της ιδιότητας και το `rdf:ID` της υπερκλάσης από την οποία κληρονομήθηκε. Το δένδρο ιδιοτήτων κλάσης περιέχει ένα αναδυόμενο μενού (με τη χρήση του δεξιού κλικ του ποντικιού) παρέχοντας αρκετές και σημαντικές λειτουργίες που είναι οι παρακάτω:



Εικόνα 52: Το δένδρο ιδιοτήτων κλάσης και οι διαθέσιμες επιλογές του αναδυόμενου μενού του.

1. *Εμφάνιση πληροφορίας της ιδιότητας*: Με την επιλογή αυτή εμφανίζεται το παράθυρο επεξεργασίας ιδιότητας μέσα από κλάση του οποίου το περιεχόμενο διαφοροποιείται ελαφρά ανάλογα με το αν η επιλεγμένη ιδιότητα είναι τύπου δεδομένων ή αντικειμένων και στο οποίο παρέχονται αρκετές επιλογές στο χρήστη μέσω των παρακάτω τεσσάρων διαφορετικών όψεων:

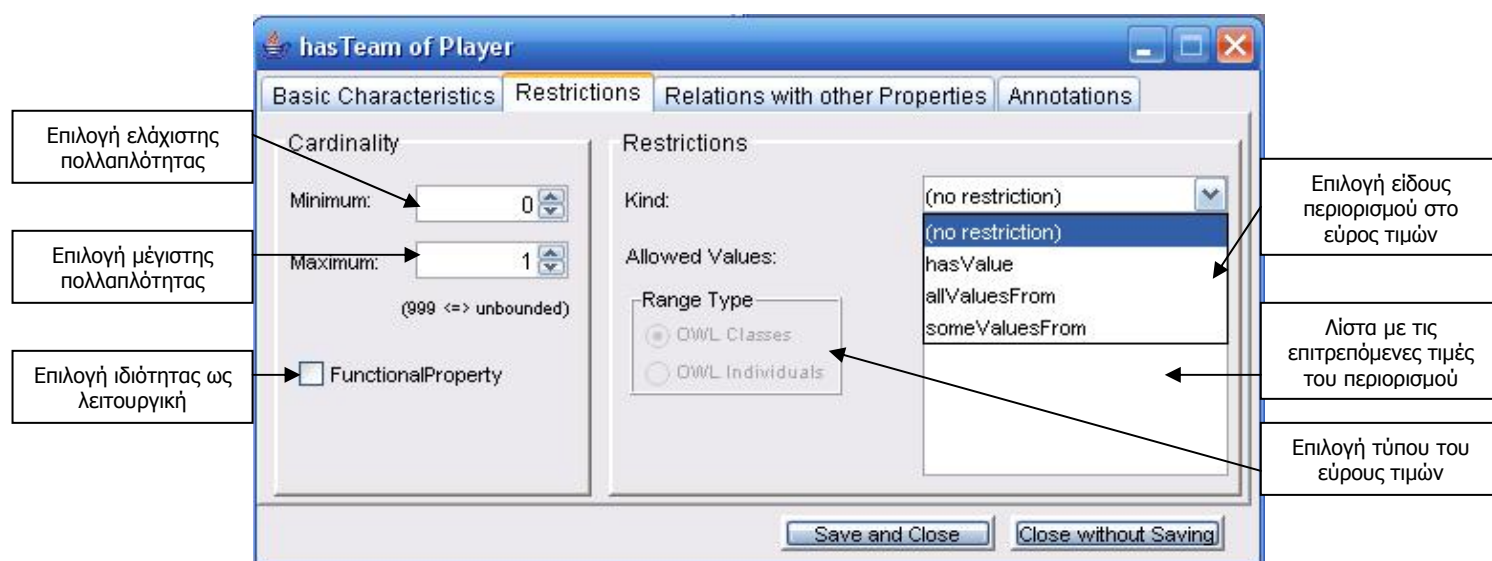
- Η *όψη βασικών χαρακτηριστικών ιδιότητας* επιτρέπει στο χρήστη να τροποποιήσει το `rdf:ID` της επιλεγμένης ιδιότητας, να ορίσει τον τύπο του εύρους τιμών της (XML τύπους δεδομένων ή κυριολεκτικές τιμές αν πρόκειται για ιδιότητα τύπου δεδομένων και OWL κλάσεις ή άτομα αν πρόκειται για ιδιότητα αντικειμένων) και το ίδιο το εύρος τιμών της και τέλος να δηλώσει τα τυχόν χαρακτηριστικά της σε περίπτωση ιδιότητας αντικειμένων. Σε περίπτωση που η ιδιότητα είναι κληρονομούμενη από υπερκλάση της τρέχουσας κλάσης δεν επιτρέπεται η επεξεργασία των παραπάνω στοιχείων παρά μόνο η εμφάνισή τους.



Εικόνα 53: Όψη βασικών χαρακτηριστικών ιδιότητας αντικειμένων στο παράθυρο επεξεργασίας ιδιότητας μέσα από κλάση.

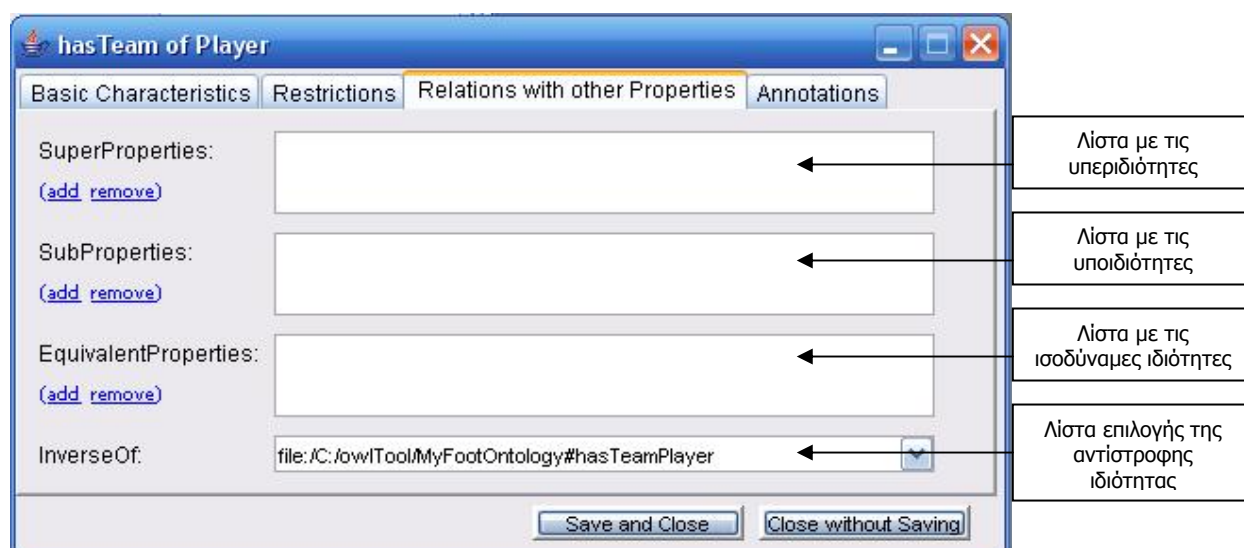
- Η *όψη περιορισμών* επιτρέπει τον ορισμό ή μη της επιλεγμένης ιδιότητας ως συναρτησιακή. Αν δεν επιλεγεί ως συναρτησιακή, τότε μπορεί να οριστεί η πολλαπλότητα της επιλεγμένης ιδιότητας για την τρέχουσα κλάση και με βάση αυτήν την επιλογή να προκύψουν οι τυχόν περιορισμοί πολλαπλότητας σύμφωνα με τους παρακάτω κανόνες:
 - Αν η ελάχιστη πολλαπλότητα είναι 0 και η μέγιστη απεριόριστη, τότε δεν προκύπτει κανένας περιορισμός πολλαπλότητας.
 - Αν η ελάχιστη πολλαπλότητα είναι ίση με τη μέγιστη, τότε προκύπτει περιορισμός ακριβούς πλήθους ίσος με την ελάχιστη πολλαπλότητα.
 - Αν η ελάχιστη πολλαπλότητα είναι μεγαλύτερη του 0 και διαφορετική από τη μέγιστη, τότε προκύπτει περιορισμός ελάχιστου πλήθους ίσος με την ελάχιστη πολλαπλότητα.
 - Αν η μέγιστη πολλαπλότητα δεν είναι απεριόριστη και είναι διαφορετική από την ελάχιστη, τότε προκύπτει περιορισμός μέγιστου πλήθους ίσος με την μέγιστη πολλαπλότητα.

Παράλληλα, η όψη αυτή επιτρέπει τον ορισμό περιορισμού στο εύρος τιμών της επιλεγμένης ιδιότητας για την τρέχουσα κλάση. Αν πρόκειται για ιδιότητα τύπου δεδομένων, τότε μπορεί να οριστεί περιορισμός του τύπου 'έχει τιμή' και να επιλεγεί μια συγκεκριμένη κυριολεκτική τιμή για αυτόν τον περιορισμό. Αν πρόκειται όμως για ιδιότητα αντικειμένων, τότε μπορεί να οριστεί περιορισμός επιλέγοντας έναν από τους τρεις τύπους: 'όλες οι τιμές από', 'κάποιες τιμές από' ή 'έχει τιμή'. Για τους δύο πρώτους τύπους, το εύρος τιμών του περιορισμού μπορεί να είναι είτε κλάσεις είτε άτομα, ενώ για τον τρίτο επιλέγεται υποχρεωτικά ένα άτομο.



Εικόνα 54: Όψη περιορισμών ιδιότητας αντικειμένων στο παράθυρο επεξεργασίας ιδιότητας μέσα από κλάση.

- Η *όψη σχέσεων με τις υπόλοιπες ιδιότητες* επιτρέπει τον ορισμό των σχέσεων της επιλεγμένης ιδιότητας με τις υπόλοιπες ιδιότητες. Πιο συγκεκριμένα, υποστηρίζεται η προσθαφαίρεση των υπεριδιοτήτων της επιλεγμένης ιδιότητας, των υποιδιοτήτων της και των ισοδύναμων με αυτή ιδιοτήτων. Αν πρόκειται για ιδιότητα αντικειμένων, είναι δυνατός ο ορισμός της αντίστροφης ιδιότητας της. Σε περίπτωση που η ιδιότητα είναι κληρονομούμενη από υπερκλάση της τρέχουσας κλάσης δεν επιτρέπεται η επεξεργασία των παραπάνω στοιχείων παρά μόνο η εμφάνισή τους.



Εικόνα 55: Όψη σχέσεων ιδιότητας αντικειμένων με τις υπόλοιπες ιδιότητες στο παράθυρο επεξεργασίας ιδιότητας μέσα από κλάση.

- Στην *όψη υπομνηματισμών* ο χρήστης έχει τη δυνατότητα να επισυνάψει ετικέτες και σχόλια για την επιλεγμένη ιδιότητα. Και σε αυτήν την όψη, στην περίπτωση που η ιδιότητα είναι κληρονομούμενη από υπερκλάση της τρέχουσας κλάσης δεν επιτρέπεται η επεξεργασία των παραπάνω στοιχείων παρά μόνο η εμφάνισή τους.



Εικόνα 56: Όψη υπομνηματισμών ιδιότητας στο παράθυρο επεξεργασίας ιδιότητας μέσα από κλάση.

2. *Προσθήκη νέας ιδιότητας τύπου δεδομένων στην κλάση:* Ουσιαστικά πρόκειται για εισαγωγή νέας ιδιότητας τύπου δεδομένων στην οντολογία η οποία θα έχει ως πεδίο ορισμού της την τρέχουσα κλάση. Ο χρήστης καλείται να δώσει το όνομα (rdf:ID) της νέας ιδιότητας και να ορίσει το εύρος τιμών της επιλέγοντας ανάμεσα στους επιτρεπόμενους XML τύπους δεδομένων.

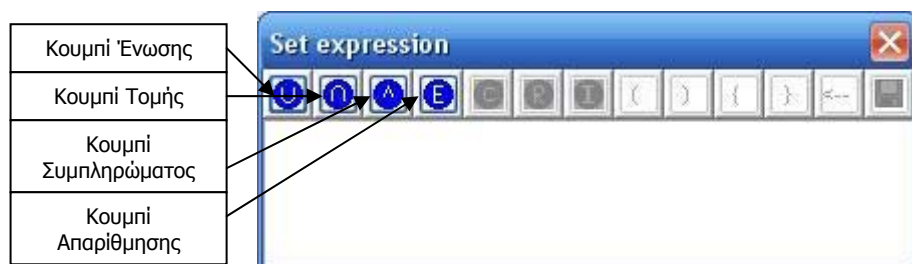
3. *Προσθήκη υπάρχουσας ιδιότητας τύπου δεδομένων στην κλάση*: Ουσιαστικά πρόκειται για προσθήκη της τρέχουσας κλάσης στο πεδίο ορισμού της υπάρχουσας στην οντολογία ιδιότητας τύπου δεδομένων.
4. *Διαγραφή υπάρχουσας ιδιότητας τύπου δεδομένων από την οντολογία*.
5. *Διαγραφή υπάρχουσας ιδιότητας τύπου δεδομένων από την τρέχουσα κλάση*: Εφαρμόζεται μόνο όταν η τρέχουσα κλάση περιέχεται στο πεδίο ορισμού της επιλεγμένης ιδιότητας και ουσιαστικά αφαιρεί την τρέχουσα κλάση από το πεδίο ορισμού της ιδιότητας.
6. *Προσθήκη νέας ιδιότητας αντικειμένων στην κλάση*: Ουσιαστικά πρόκειται για εισαγωγή νέας ιδιότητας αντικειμένων στην οντολογία η οποία θα έχει ως πεδίο ορισμού της την τρέχουσα κλάση. Ο χρήστης καλείται να δώσει το όνομα (rdf:ID) της νέας ιδιότητας και να ορίσει το εύρος τιμών της επιλέγοντας ανάμεσα από όλες τις κλάσεις των ανοιχτών οντολογιών.
7. *Προσθήκη υπάρχουσας ιδιότητας αντικειμένων στην κλάση*: Ουσιαστικά πρόκειται για προσθήκη της τρέχουσας κλάσης στο πεδίο ορισμού της υπάρχουσας στην οντολογία ιδιότητας αντικειμένων.
8. *Διαγραφή υπάρχουσας ιδιότητας αντικειμένων από την οντολογία*.
9. *Διαγραφή υπάρχουσας ιδιότητας αντικειμένων από την τρέχουσα κλάση*: Εφαρμόζεται μόνο όταν η τρέχουσα κλάση περιέχεται στο πεδίο ορισμού της επιλεγμένης ιδιότητας και ουσιαστικά αφαιρεί την τρέχουσα κλάση από το πεδίο ορισμού της ιδιότητας.

Για να εκτελεστούν όλες οι παραπάνω ενέργειες απαιτείται η επιλογή από το δένδρο ιδιοτήτων κλάσης ενός κόμβου που να αντιστοιχεί σε ιδιότητα τύπου δεδομένων ή σε ιδιότητα αντικειμένων και όχι επιλογή των βοηθητικών κόμβων που χρησιμοποιούνται για την ομαδοποίηση των ιδιοτήτων.

6.3.4. Ορισμός επιλεγμένης κλάσης ως σύνθετης

Μέσω του *κουμπιού ορισμού κλάσης ως σύνθετη* παρέχεται η δυνατότητα ορισμού της επιλεγμένης κλάσης ως σύνθετη (ένωση ή τομή υπάρχοντων κλάσεων, συμπλήρωμα υπάρχουσας κλάσης ή απαρίθμηση υπάρχοντων ατόμων). Όταν το κουμπί δεν είναι επιλεγμένο σημαίνει ότι η κλάση δεν είναι σύνθετη. Όταν επιλεγεί, ανοίγει το παράθυρο δημιουργίας έκφρασης σύνθετης κλάσης το οποίο περιέχει τα κατάλληλα κουμπιά για τη διαμόρφωση της έκφρασης της σύνθετης κλάσης. Τα κουμπιά αυτά ενεργοποιούνται και απενεργοποιούνται κατάλληλα όσο ο χρήστης σχηματίζει την έκφραση έτσι ώστε να μην του επιτρέψουν να σχηματίσει συντακτικά λανθασμένη έκφραση. Παράλληλα καθόλη τη διάρκεια της δημιουργίας της έκφρασης σύνθετης κλάσης, ο χρήστης έχει τη δυνατότητα διόρθωσης

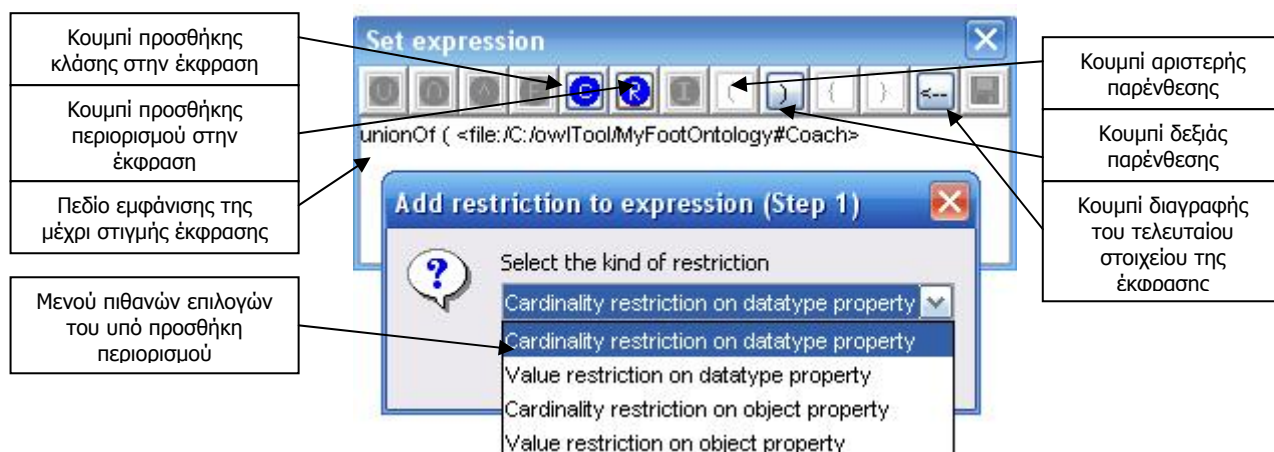
μέσω του κουμπιού διαγραφής το οποίο σβήνει το τελευταίο κάθε φορά στοιχείο της έκφρασης.



Εικόνα 57: Αρχική όψη παράθυρου δημιουργίας έκφρασης σύνθετης κλάσης.

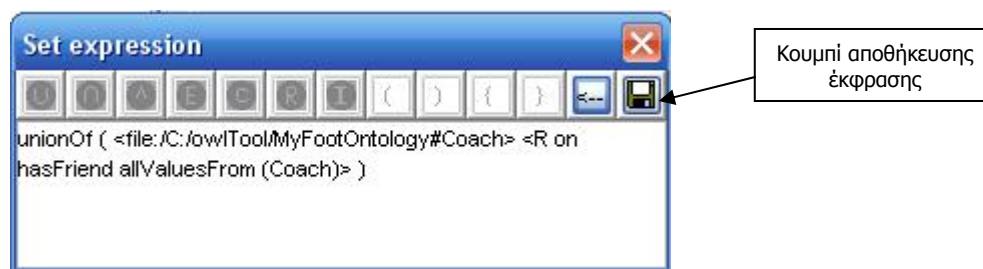
Τα πιθανά σενάρια σχηματισμού της έκφρασης είναι τα εξής:

1. Αρχική επιλογή ένωσης ή τομής: Η επόμενη επιλογή είναι υποχρεωτική και είναι το άνοιγμα της αριστερής παρένθεσης. Μέσα στην παρένθεση θα περιέχονται οι κλάσεις των οποίων η ένωση ή η τομή θα αποτελέσει τη σύνθετη κλάση. Ακολούθως, ο χρήστης μπορεί να προσθέσει διαδοχικά στην έκφραση επώνυμες κλάσεις που βρίσκονται σε κάποια από τις ανοιχτές οντολογίες ή ανώνυμες κλάσεις, δηλαδή περιορισμούς (στο εύρος τιμών ή πολλαπλότητας) σε ιδιότητες (τύπου δεδομένων ή αντικειμένων) που βρίσκονται επίσης σε κάποια από τις ανοιχτές οντολογίες.



Εικόνα 58: Πρώτο βήμα για προσθήκη περιορισμού στην έκφραση κλάσης στο παράθυρο δημιουργίας έκφρασης σύνθετης κλάσης.

Όταν ο χρήστης ολοκληρώσει την προσθήκη κλάσεων και περιορισμών στην έκφραση, επιλέγει τη δεξιά παρένθεση και αυτομάτως το κουμπί αποθήκευσης ενεργοποιείται.



Εικόνα 59: Ολοκληρωμένη έκφραση σύνθετης κλάσης, η οποία αποτελείται από την ένωση της κλάσης "Coach" και του περιορισμού στην ιδιότητα αντικειμένων "hasFriend" να έχει όλες οι τιμές από την κλάση "Coach", στο παράθυρο δημιουργίας έκφρασης σύνθετης κλάσης.

2. Αρχική επιλογή συμπληρώματος: Η επόμενη επιλογή είναι υποχρεωτική και είναι το άνοιγμα της αριστερής παρένθεσης. Μέσα στην παρένθεση θα περιέχεται η κλάση της οποίας το συμπλήρωμα θα αποτελέσει τη σύνθετη κλάση. Ακολούθως, ο χρήστης μπορεί να προσθέσει στην έκφραση μια μόνο επώνυμη κλάση που θα βρίσκεται σε κάποια από τις ανοιχτές οντολογίες. Ακολούθως επιλέγει τη δεξιά παρένθεση και αυτομάτως το κουμπί αποθήκευσης ενεργοποιείται.
3. Αρχική επιλογή απαρίθμησης: Η επόμενη επιλογή είναι υποχρεωτική και είναι το άνοιγμα του αριστερού άγκιστρου. Μέσα στα άγκιστρα θα περιέχονται τα άτομα κλάσεων των οποίων η απαρίθμηση θα αποτελέσει τη σύνθετη κλάση. Ακολούθως, ο χρήστης μπορεί να προσθέσει στην έκφραση άτομα που βρίσκονται σε κάποια από τις ανοιχτές οντολογίες. Όταν ολοκληρώσει την ατόμων στην έκφραση, επιλέγει το δεξιό άγκιστρο και αυτομάτως το κουμπί αποθήκευσης ενεργοποιείται.

Από τη στιγμή που το κουμπί αποθήκευσης είναι ενεργοποιημένο, η έκφραση σύνθετης κλάσης είναι συντακτικά σωστή. Συνεπώς, με το πάτημά του, το παράθυρο δημιουργίας έκφρασης σύνθετης κλάσης κλείνει, η επιλεγμένη κλάση δηλώνεται ως σύνθετη και η έκφραση σύνθετης κλάσης αποθηκεύεται και εμφανίζεται στο πεδίο έκφρασης σύνθετης κλάσης (βλέπε εικόνα 49). Η έκφραση αυτή μπορεί να επεξεργαστεί αργότερα από το χρήστη και να αποθηκευθεί ξανά με τη νέα της μορφή. Τέλος, σε περίπτωση που η επιλεγμένη κλάση είναι ήδη σύνθετη και αποεπιλεγεί το κουμπί ορισμού κλάσης ως σύνθετη, η κλάση παύει να είναι σύνθετη και η έκφραση σύνθετης κλάσης της διαγράφεται.

6.3.5. Όψη σχέσεων της επιλεγμένης κλάσης με τις υπόλοιπες κλάσεις

Η όψη σχέσεων της επιλεγμένης κλάσης με τις υπόλοιπες ουσιαστικά επιτρέπει την αντιστοίχιση κλάσεων από διαφορετικές οντολογίες και περιλαμβάνει τα παρακάτω στοιχεία:

1. *Λίστα με υπερκλάσεις της κλάσης*: Η λίστα αυτή περιλαμβάνει τις υπερκλάσεις της επιλεγμένης κλάσης οι οποίες μπορούν να βρίσκονται και σε διαφορετικές από την τρέχουσα οντολογίες. Υποστηρίζεται η πρόσθεση νέας υπερκλάσης η οποία μπορεί να επιλεγεί μέσα από όλες τις ανοιχτές οντολογίες και η διαγραφή υπάρχουσας υπερκλάσης. Προφανώς κάθε υπερκλάση που προστίθεται από το χρήστη στη λίστα είναι απευθείας υπερκλάση της επιλεγμένης κλάσης. Εφόσον η υπερκλάση βρίσκεται στην ίδια οντολογία με την επιλεγμένη κλάση, τότε στη λίστα εμφανίζεται δίπλα στο πλήρες URI της υπερκλάσης και ένας αριθμός που δείχνει τη διαφορά επιπέδων στο δένδρο ιεραρχίας κλάσεων μεταξύ της επιλεγμένης κλάσης και της συγκεκριμένης υπερκλάσης (π.χ. ο αριθμός -1 φανερώνει ότι η υπερκλάση βρίσκεται ένα επίπεδο παραπάνω από την τρέχουσα κλάση). Παράλληλα, η πρόσθεση νέας ή η διαγραφή υπάρχουσας υπερκλάσης που ανήκει στην ίδια οντολογία με την επιλεγμένη κλάση επιφέρει αλλαγές στη διάταξη του δένδρου ιεραρχίας κλάσεων.
2. *Λίστα με υποκλάσεις της κλάσης*: Η λίστα αυτή περιλαμβάνει τις υποκλάσεις της επιλεγμένης κλάσης οι οποίες μπορούν να βρίσκονται και σε διαφορετικές από την τρέχουσα οντολογίες. Υποστηρίζεται η πρόσθεση νέας υποκλάσης η οποία μπορεί να επιλεγεί μέσα από όλες τις ανοιχτές οντολογίες και η διαγραφή υπάρχουσας υποκλάσης. Προφανώς κάθε υποκλάση που προστίθεται από το χρήστη στη λίστα είναι απευθείας υποκλάση της επιλεγμένης κλάσης. Εφόσον η υποκλάση βρίσκεται στην ίδια οντολογία με την επιλεγμένη κλάση, τότε στη λίστα εμφανίζεται δίπλα στο πλήρες URI της υποκλάσης και ένας αριθμός που δείχνει τη διαφορά επιπέδων στο δένδρο ιεραρχίας κλάσεων μεταξύ της επιλεγμένης κλάσης και της συγκεκριμένης υποκλάσης (π.χ. ο αριθμός +2 φανερώνει ότι η υποκλάση βρίσκεται δύο επίπεδα παρακάτω από την τρέχουσα κλάση). Παράλληλα, η πρόσθεση νέας ή η διαγραφή υπάρχουσας υποκλάσης που ανήκει στην ίδια οντολογία με την επιλεγμένη κλάση επιφέρει αλλαγές στη διάταξη του δένδρου ιεραρχίας κλάσεων.
3. *Λίστα με ξένες ως προς την κλάση κλάσεις*: Η λίστα αυτή περιλαμβάνει τις ξένες ως προς την επιλεγμένη κλάση κλάσεις οι οποίες μπορούν να βρίσκονται και σε διαφορετικές από την τρέχουσα οντολογίες. Υποστηρίζεται η πρόσθεση νέας ξένης κλάσης η οποία μπορεί να επιλεγεί μέσα από όλες τις ανοιχτές οντολογίες και η διαγραφή υπάρχουσας ξένης κλάσης.
4. *Λίστα με ισοδύναμες κλάσεις της κλάσης*: Η λίστα αυτή περιλαμβάνει τις ισοδύναμες κλάσεις της επιλεγμένης κλάσης οι οποίες μπορούν να βρίσκονται και σε διαφορετικές από την τρέχουσα οντολογίες. Υποστηρίζεται η πρόσθεση νέας ισοδύναμης κλάσης η οποία μπορεί να επιλεγεί μέσα από όλες τις ανοιχτές οντολογίες και η διαγραφή υπάρχουσας ισοδύναμης κλάσης.

Η όψη αυτή ουσιαστικά παρέχει δυνατότητες εξατομίκευσης στο χρήστη καθώς μπορεί να δημιουργήσει μια δικιά του οντολογία με ονόματα κλάσεων της επιλογής του και να

αντιστοιχίσει τις κλάσεις αυτές με κλάσεις υπάρχοντων και γνωστών οντολογιών κληρονομώντας τα χαρακτηριστικά τους.

6.4. Λειτουργικότητα σε επίπεδο OWL ιδιοτήτων

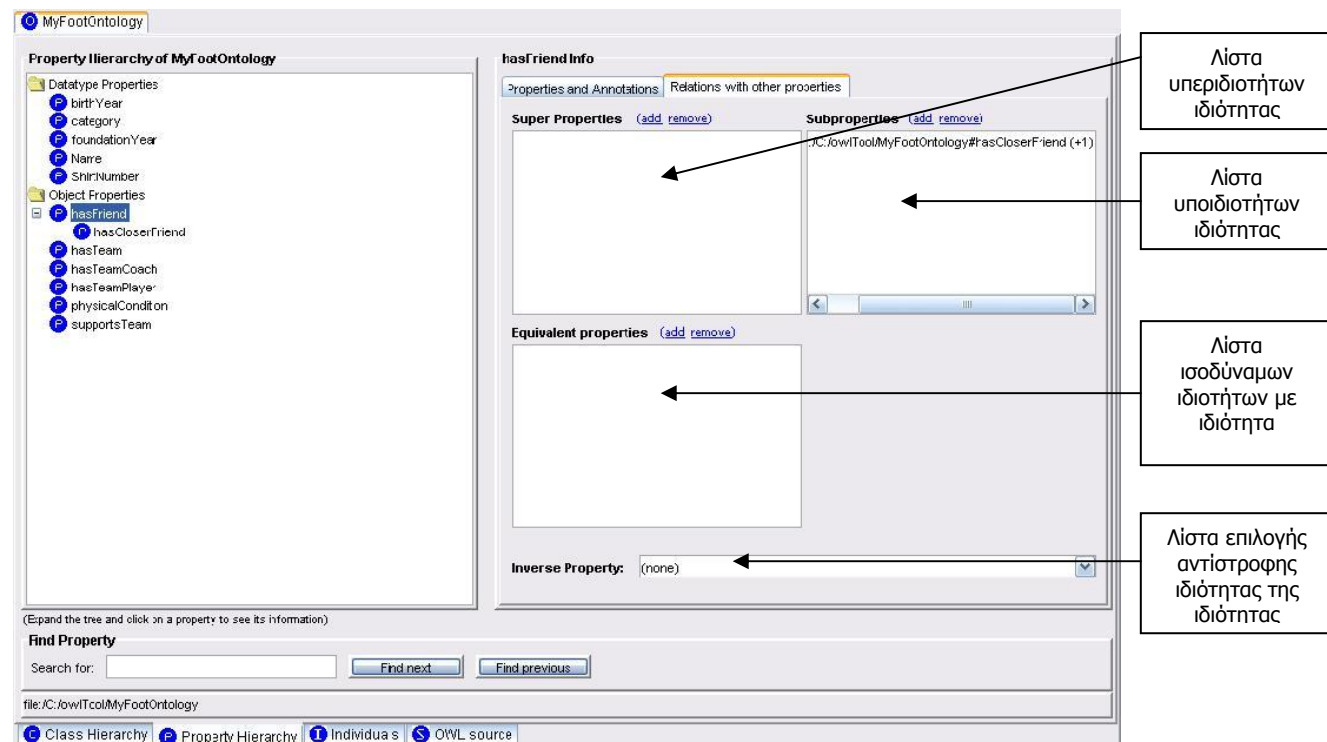
6.4.1. Δένδρο ιεραρχίας ιδιοτήτων

Για τη δημιουργία σε μια οντολογία νέων ιδιοτήτων και την επεξεργασία ή τη διαγραφή υπάρχοντων ιδιοτήτων, το GraphOnto χρησιμοποιεί και την όψη ιδιοτήτων η οποία βρίσκεται μέσα στην καρτέλα "Property Hierarchy" η οποία φαίνεται στις δύο παρακάτω εικόνες:

The screenshot shows the 'Property Hierarchy of MyFootOntology' window. On the left is a tree view of properties, and on the right is the 'hasFriendInfo' property configuration panel. Annotations with arrows point to various UI elements:

- Δένδρο ιεραρχίας ιδιοτήτων**: Points to the left-hand tree view.
- Βοηθητικοί κόμβοι**: Points to the 'Delete Properties' and 'Object Properties' sub-headers in the tree.
- Επιλεγμένη ιδιότητα**: Points to the 'hasFriend' property selected in the tree.
- Όνομα επιλεγμένης κλάσης**: Points to the 'FootballPerson' class selected in the 'Domain' field.
- Κουμπιά επιλογής τύπου εύρους τιμών ιδιότητας**: Points to the 'Range' section, specifically the 'OWL Class' radio button.
- Επιλεγμένο το Property Hierarchy Tab**: Points to the 'Property Hierarchy' tab in the bottom navigation bar.
- Πεδίο επεξεργασίας rdf:ID ιδιότητας**: Points to the 'Full Name' field.
- Λίστα πεδίου ορισμού ιδιότητας**: Points to the 'Domain' field.
- Λίστα εύρους τιμών ιδιότητας**: Points to the 'Range' field.
- Λίστα ετικετών ιδιότητας**: Points to the 'Labels' field.
- Λίστα σχολίων ιδιότητας**: Points to the 'Comments' field.
- Κουμπιά επιλογής χαρακτηριστικών ιδιότητας**: Points to the 'Characteristics' section, specifically the 'Symmetric' checkbox.

Εικόνα 60: Η "Property Hierarchy" καρτέλα στην οποία φαίνονται αριστερά το δένδρο ιεραρχίας των ιδιοτήτων της οντολογίας και δεξιά τα χαρακτηριστικά και οι σχολιασμοί της επιλεγμένης ιδιότητας.



Εικόνα 61: Η "Property Hierarchy" καρτέλα στην οποία φαίνονται αριστερά το δέντρο ιεραρχίας των ιδιοτήτων της οντολογίας και δεξιά οι σχέσεις της ιδιότητας με τις υπόλοιπες ιδιότητες.

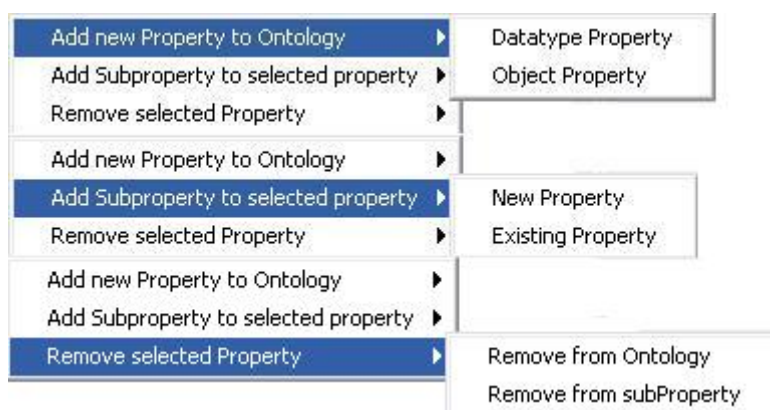
Όπως φαίνεται από τις παραπάνω εικόνες, για όλες τις ιδιότητες της οντολογίας δημιουργούνται αντίστοιχοι κόμβοι που περιλαμβάνουν το `rdf:ID` της ιδιότητας και το εικονίδιο 6 και οι οποίοι ταξινομούνται στο δένδρο ιεραρχίας ιδιοτήτων. Η τοποθέτηση των κόμβων στο δένδρο γίνεται με βάση τους επόμενους κανόνες:

- Όλοι οι κόμβοι των ιδιοτήτων τύπου δεδομένων ομαδοποιούνται και τοποθετούνται κάτω από το βοηθητικό κόμβο "Datatype Properties" και αντίστοιχα όλοι οι κόμβοι των ιδιοτήτων αντικειμένων κάτω από το βοηθητικό κόμβο "Object Properties".
- Κάθε κόμβος ιδιότητας τοποθετείται σε μικρότερο βάθος σε σχέση με τους κόμβους των υποιδιοτήτων της και σε μεγαλύτερο βάθος σε σχέση με τους κόμβους των υπερδιδοτήτων της. Πρακτικά αυτό σημαίνει ότι ως παιδιά των δύο βοηθητικών κόμβων τοποθετούνται κόμβοι ιδιοτήτων οι οποίες δεν έχουν καμία υπερδιδοτητα.
- Για κάθε ιδιότητα που έχει περισσότερες από μία υπερδιδοτητες δημιουργούνται ισάριθμοι κόμβοι με τον αριθμό των υπερδιδοτήτων της. Καθένας από αυτούς τους κόμβους, οι οποίοι περιλαμβάνουν το εικονίδιο 7, τοποθετείται κάτω από κάθε κόμβο που αντιστοιχεί σε υπερδιδοτητα της ιδιότητας.
- Τα παιδιά κάθε κόμβου (συμπεριλαμβανομένης και των δύο βοηθητικών κόμβων) ταξινομούνται αλφαβητικά με βάση το `rdf:ID` της ιδιότητας στην οποία αντιστοιχούν.

Το δένδρο ιεραρχίας ιδιοτήτων περιέχει ένα αναδυόμενο μενού (με τη χρήση του δεξιού κλικ του ποντικιού) παρέχοντας αρκετές και σημαντικές λειτουργίες όπως:

1. Εισαγωγή νέας αυτόνομης (που δεν έχει αρχικώς καμία υπεριδιότητα ή υποϊδιότητα) ιδιότητας τύπου δεδομένων στην τρέχουσα οντολογία.
2. Εισαγωγή νέας αυτόνομης (που δεν έχει αρχικώς καμία υπεριδιότητα ή υποϊδιότητα) ιδιότητας αντικειμένων στην τρέχουσα οντολογία.
3. Εισαγωγή νέας ιδιότητας (τύπου δεδομένων ή αντικειμένων) στην τρέχουσα οντολογία και ορισμός της ως υποϊδιότητας της επιλεγμένης ιδιότητας.
4. Ορισμός υπάρχουσας στην τρέχουσα οντολογία ιδιότητας (τύπου δεδομένων ή αντικειμένων) ως υποϊδιότητα της επιλεγμένης ιδιότητας.
5. Διαγραφή από την οντολογία της επιλεγμένης ιδιότητας.
6. Διαγραφή σε υπάρχουσα ιδιότητα από το να είναι υποϊδιότητα μιας υπεριδιότητάς της.

Οι λειτουργίες 3-6 απαιτούν από το χρήστη να έχει επιλέξει προηγουμένως έναν κόμβο του δένδρου και ο οποίος θα πρέπει να είναι διαφορετικός από τους δύο βοηθητικούς κόμβους του δένδρου. Οι τροποποιήσεις στην οντολογία που επιφέρουν οι ενέργειες αυτές έχουν ως αποτέλεσμα τη συνεχή αναδιάταξη του δένδρου ιεραρχίας ιδιοτήτων, πάντα σύμφωνα με τους κανόνες που παρουσιάστηκαν προηγουμένως.



Εικόνα 62: Οι διαθέσιμες επιλογές του αναδυόμενου μενού του δένδρου ιεραρχίας ιδιοτήτων.

Με την επιλογή ενός κόμβου, πλην των βοηθητικών κόμβων φυσικά, εμφανίζονται οι πληροφορίες της αντίστοιχης ιδιότητας. Η εμφάνιση αυτών των πληροφοριών γίνεται μέσω δύο όψεων που περιγράφονται στις επόμενες παραγράφους:

- Όψη χαρακτηριστικών και σχολιασμών της επιλεγμένης ιδιότητας.
- Όψη σχέσεων επιλεγμένης ιδιότητας με τις υπόλοιπες ιδιότητες

6.4.2. Όψη χαρακτηριστικών και σχολιασμών της επιλεγμένης ιδιότητας

Η *όψη χαρακτηριστικών και σχολιασμών* της επιλεγμένης ιδιότητας περιλαμβάνει τα παρακάτω στοιχεία:

1. *Πεδίο επεξεργασίας rdf:ID ιδιότητας*: Το πεδίο αυτό περιέχει το πλήρες URI της επιλεγμένης ιδιότητας. Επιτρέπει μόνο την αλλαγή του rdf:ID και όχι του πλήρους URI της ιδιότητας. Κάθε πιθανή αλλαγή του rdf:ID μπορεί να επιφέρει αναδιάταξη κάποιων κόμβων του δένδρου ιεραρχίας ιδιοτήτων, αφού αυτοί ταξινομούνται αλφαβητικά.
2. *Λίστα πεδίου ορισμού ιδιότητας*: Η λίστα αυτή περιέχει τις κλάσεις οι οποίες ανήκουν στο πεδίο ορισμού της επιλεγμένης ιδιότητας. Υποστηρίζεται η διεύρυνση του πεδίου ορισμού της ιδιότητας με πρόσθεση κλάσης σε αυτό και ο περιορισμός του με απομάκρυνση κλάσης από αυτό.
3. *Λίστα εύρους τιμών ιδιότητας*: Η λίστα αυτή περιέχει το εύρος τιμών της ιδιότητας το οποίο ποικίλει ανάλογα με τον τύπο της ιδιότητας και καθορίζεται μέσω των κουμπιών επιλογής τύπου εύρους τιμών. Για ιδιότητες τύπου δεδομένων υποστηρίζεται η χρήση έτοιμων XML τύπων δεδομένων ή απλών κυριολεκτικών τιμών, ενώ για ιδιότητες αντικειμένων υποστηρίζεται η χρήση OWL κλάσεων ή OWL ατόμων.
4. *Λίστα ετικετών ιδιότητας*: Η λίστα αυτή περιέχει τις ετικέτες που έχουν επισυναφθεί με την επιλεγμένη ιδιότητα. Επιτρέπεται η πρόσθεση νέας και η διαγραφή υπάρχουσας ετικέτας.
5. *Λίστα σχολίων ιδιότητας*: Η λίστα αυτή περιέχει τα σχόλια που έχουν επισυναφθεί με την επιλεγμένη ιδιότητα. Επιτρέπεται η πρόσθεση νέου και η διαγραφή υπάρχοντος σχολίου.
6. *Κουμπιά επιλογής χαρακτηριστικών ιδιότητας*: Τα κουμπιά αυτά χρησιμοποιούνται για την επιλογή των χαρακτηριστικών της επιλεγμένης ιδιότητας. Σε περίπτωση που η επιλεγμένη ιδιότητα είναι τύπου δεδομένων, τότε υπάρχει μόνο ένα κουμπί επιλογής που επιτρέπει το χαρακτηρισμό της ιδιότητας ως συναρτησιακής. Στην περίπτωση επιλεγμένης ιδιότητας αντικειμένων υπάρχουν τέσσερα διαφορετικά κουμπιά επιλογής που επιτρέπουν το χαρακτηρισμό της ιδιότητας ως μεταβατικής, συμμετρικής, συναρτησιακής και αντίστροφα συναρτησιακής.

6.4.3. Όψη σχέσεων επιλεγμένης ιδιότητας με τις υπόλοιπες ιδιότητες

Η *όψη σχέσεων της επιλεγμένης ιδιότητας* με τις υπόλοιπες ουσιαστικά επιτρέπει την αντιστοίχιση ιδιοτήτων από διαφορετικές οντολογίες και περιλαμβάνει τα παρακάτω στοιχεία:

1. *Λίστα με υπεριδιότητες της ιδιότητας*: Η λίστα αυτή περιλαμβάνει τις υπεριδιότητες της επιλεγμένης ιδιότητας οι οποίες μπορούν να βρίσκονται και σε διαφορετικές από την τρέχουσα οντολογίες. Υποστηρίζεται η πρόσθεση νέας υπεριδιότητας η οποία μπορεί να επιλεγεί μέσα από όλες τις ανοιχτές οντολογίες και η διαγραφή υπάρχουσας υπεριδιότητας. Προφανώς κάθε υπεριδιότητα που προστίθεται από το χρήστη στη λίστα

είναι απευθείας υπεριδιότητα της επιλεγμένης ιδιότητας. Εφόσον η υπεριδιότητα βρίσκεται στην ίδια οντολογία με την επιλεγμένη ιδιότητα, τότε στη λίστα εμφανίζεται δίπλα στο πλήρες URI της υπεριδιότητας και ένας αριθμός που δείχνει τη διαφορά επιπέδων στο δένδρο ιεραρχίας κλάσεων μεταξύ της επιλεγμένης ιδιότητας και της συγκεκριμένης υπεριδιότητας (π.χ. ο αριθμός -1 φανερώνει ότι η υπεριδιότητα βρίσκεται ένα επίπεδο παραπάνω από την τρέχουσα ιδιότητα). Παράλληλα, η πρόσθεση νέας ή η διαγραφή υπάρχουσας υπεριδιότητας που ανήκει στην ίδια οντολογία με την επιλεγμένη ιδιότητα επιφέρει αλλαγές στη διάταξη του δένδρου ιεραρχίας ιδιοτήτων.

2. *Λίστα με υποιδιότητες της ιδιότητας*: Η λίστα αυτή περιλαμβάνει τις υποιδιότητες της επιλεγμένης ιδιότητας οι οποίες μπορούν να βρίσκονται και σε διαφορετικές από την τρέχουσα οντολογίες. Υποστηρίζεται η πρόσθεση νέας υποιδιότητας η οποία μπορεί να επιλεγεί μέσα από όλες τις ανοιχτές οντολογίες και η διαγραφή υπάρχουσας υποιδιότητας. Προφανώς κάθε υποιδιότητα που προστίθεται από το χρήστη στη λίστα είναι απευθείας υποιδιότητα της επιλεγμένης ιδιότητας. Εφόσον η υποιδιότητα βρίσκεται στην ίδια οντολογία με την επιλεγμένη ιδιότητα, τότε στη λίστα εμφανίζεται δίπλα στο πλήρες URI της υποιδιότητας και ένας αριθμός που δείχνει τη διαφορά επιπέδων στο δένδρο ιεραρχίας κλάσεων μεταξύ της επιλεγμένης ιδιότητας και της συγκεκριμένης υποιδιότητας (π.χ. ο αριθμός +2 φανερώνει ότι η υποιδιότητα βρίσκεται δύο επίπεδα παρακάτω από την τρέχουσα ιδιότητα). Παράλληλα, η πρόσθεση νέας ή η διαγραφή υπάρχουσας υποιδιότητας που ανήκει στην ίδια οντολογία με την επιλεγμένη ιδιότητα επιφέρει αλλαγές στη διάταξη του δένδρου ιεραρχίας ιδιοτήτων.
3. *Λίστα με ισοδύναμες ιδιότητες της ιδιότητας*: Η λίστα αυτή περιλαμβάνει τις ισοδύναμες ιδιότητες της επιλεγμένης ιδιότητας οι οποίες μπορούν να βρίσκονται και σε διαφορετικές από την τρέχουσα οντολογίες. Υποστηρίζεται η πρόσθεση νέας ισοδύναμης ιδιότητας η οποία μπορεί να επιλεγεί μέσα από όλες τις ανοιχτές οντολογίες και η διαγραφή υπάρχουσας ισοδύναμης ιδιότητας.
4. *Λίστα επιλογής αντίστροφης ιδιότητας*: Η λίστα αυτή εμφανίζεται μόνο όταν η επιλεγμένη ιδιότητα είναι ιδιότητα αντικειμένων και περιέχει τις πιθανές ιδιότητες αντικειμένων από τις οποίες μία μπορεί να οριστεί ως αντίστροφη της επιλεγμένης ιδιότητας.

Η όψη αυτή ουσιαστικά παρέχει δυνατότητες εξατομίκευσης στο χρήστη καθώς μπορεί να δημιουργήσει μια δικιά του οντολογία με ονόματα ιδιοτήτων της επιλογής του και να αντιστοιχήσει τις ιδιότητες αυτές με ιδιότητες υπάρχοντων και γνωστών οντολογιών κληρονομώντας τα χαρακτηριστικά τους.

6.5. Λειτουργικότητα σε επίπεδο OWL ατόμων

6.5.1. Δένδρο ατόμων

Για τη δημιουργία σε μια οντολογία νέων ατόμων κλάσεων και την επεξεργασία ή τη διαγραφή υπάρχοντων ατόμων, το GraphOnto χρησιμοποιεί την όψη ατόμων η οποία βρίσκεται μέσα στην καρτέλα "Individuals" η οποία φαίνεται στις τρεις παρακάτω εικόνες:

The screenshot shows the GraphOnto interface with the 'Individuals' tab selected. On the left, a class hierarchy tree is visible, with 'Player' selected. On the right, the 'Diskan Info' panel displays details for the selected individual 'Player rdf:ID="Biskan"'. The interface includes a search bar at the bottom and a tabbed view at the bottom with 'Individuals' selected.

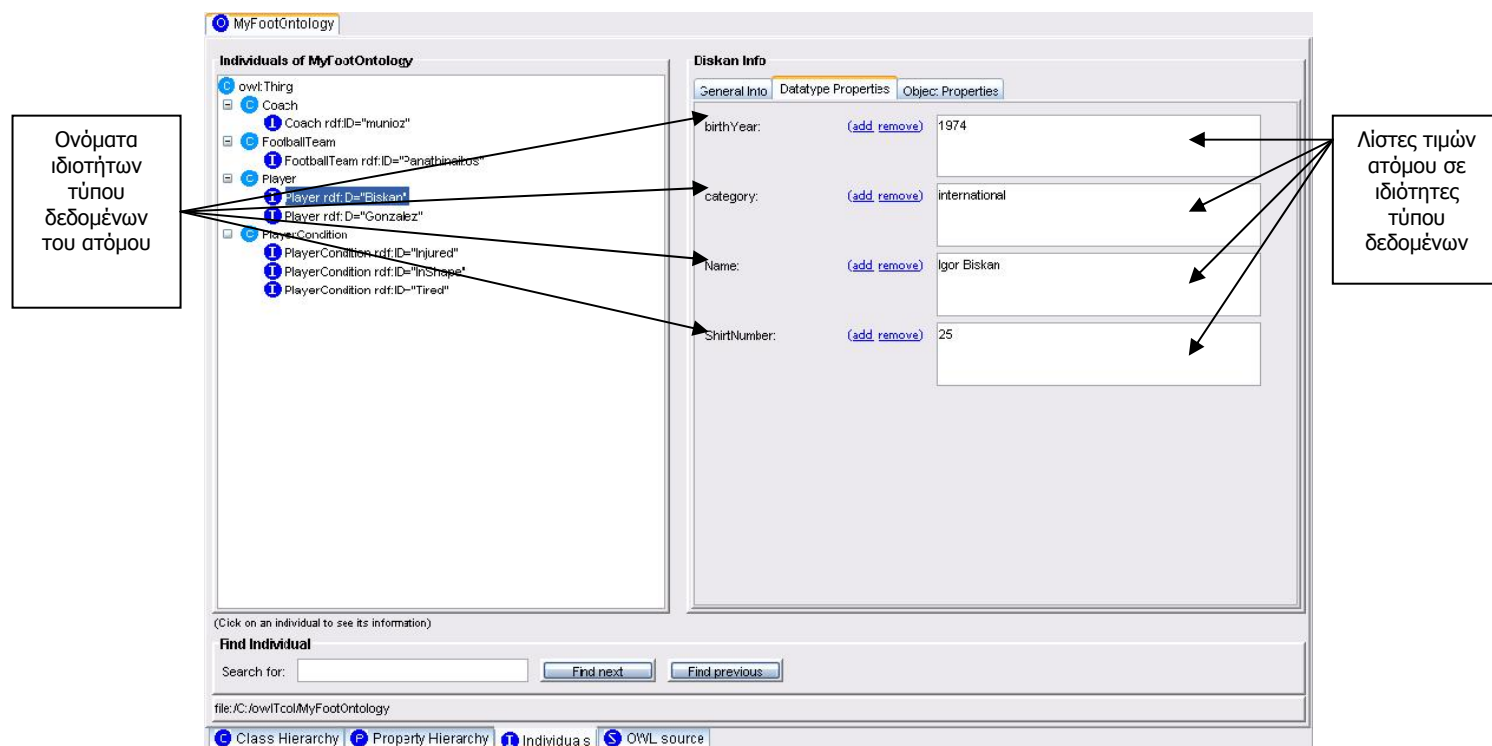
Annotations on the left side of the image:

- Δένδρο ατόμων (Points to the class hierarchy tree)
- Επιλεγμένο άτομο (Points to the selected individual 'Player rdf:ID="Biskan"')
- Όνομα επιλεγμένου ατόμου (Points to the name 'Biskan' in the individual list)
- Επιλεγμένο το "Individuals Tab" (Points to the 'Individuals' tab at the bottom)

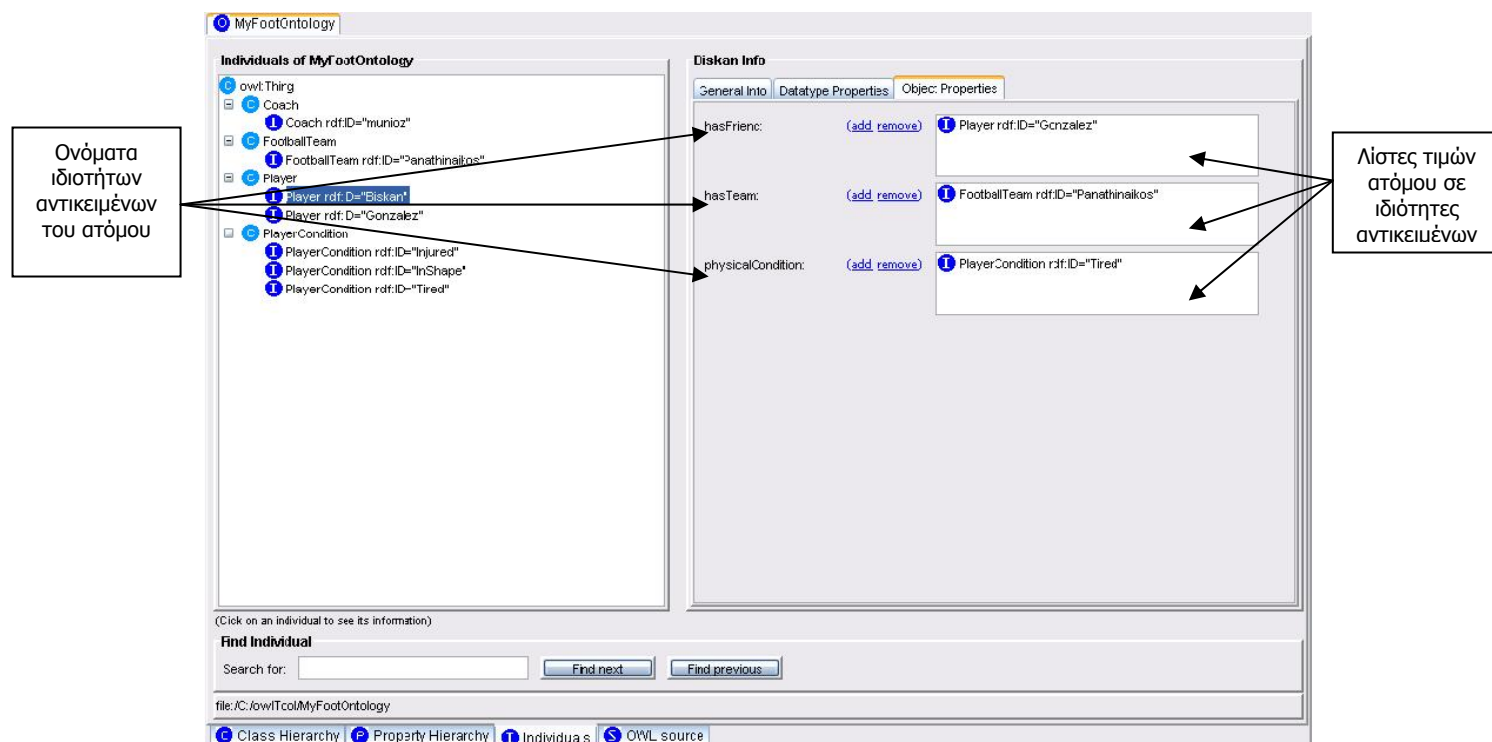
Annotations on the right side of the image:

- Πεδίο επεξεργασίας rdf:ID ατόμου (Points to the 'Full name' field)
- Λίστα τύπων ατόμου (Points to the 'Type' dropdown menu)
- Λίστα ετικετών ατόμου (Points to the 'Labels' text area)
- Λίστα σχολίων ατόμου (Points to the 'Comments' text area)
- Λίστα ακριβώς όμοιων ατόμων με άτομο (Points to the 'sameAs' list)
- Λίστα διαφορετικών ατόμων με άτομο (Points to the 'differentFrom' list)

Εικόνα 63: Η καρτέλα "Individuals" στην οποία φαίνονται αριστερά το δέντρο των ατόμων της οντολογίας και δεξιά τα γενικά χαρακτηριστικά του επιλεγμένου ατόμου.



Εικόνα 64: Η καρτέλα "Individuals" στην οποία φαίνονται αριστερά το δέντρο των ατόμων της οντολογίας και δεξιά οι τιμές του επιλεγμένου ατόμου στις ιδιότητες τύπου δεδομένων που του αντιστοιχούν.



Εικόνα 65: Η καρτέλα "Individuals" στην οποία φαίνονται αριστερά το δέντρο των ατόμων της οντολογίας και δεξιά οι τιμές του επιλεγμένου ατόμου στις ιδιότητες αντικειμένων που του αντιστοιχούν.

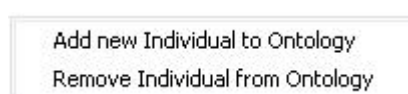
Όπως φαίνεται από τις παραπάνω εικόνες, για όλες τις κλάσεις των οποίων άτομα υπάρχουν στην οντολογία δημιουργούνται αντίστοιχοι βοηθητικοί κόμβοι που περιλαμβάνουν

το rdf:ID της κλάσης και το εικονίδιο 4, ενώ για όλα τα άτομα της οντολογίας δημιουργούνται αντίστοιχοι κόμβοι που περιλαμβάνουν το όνομα της κλάσης στην οποία ανήκουν, το rdf:ID του ατόμου και το εικονίδιο 10. Η τοποθέτηση όλων των παραπάνω κόμβων στο δένδρο ατόμων γίνεται με βάση τους επόμενους κανόνες:

- Όλοι οι βοηθητικοί κόμβοι των κλάσεων τοποθετούνται στο ίδιο επίπεδο ακριβώς κάτω από τη ρίζα του δένδρου που αναπαριστά το owl:Thing χωρίς να λαμβάνονται υπόψη οι τυχόν σχέσεις ιεραρχίας μεταξύ των κλάσεων.
- Όλοι οι κόμβοι των ατόμων ομαδοποιούνται, ταξινομούνται αλφαβητικά με βάση το rdf:ID των ατόμων και τοποθετούνται ακριβώς κάτω από τον βοηθητικό κόμβο που αντιστοιχεί στην κλάση στην οποία ανήκουν (τύπος).
- Κόμβοι ατόμων που δεν έχουν κανένα τύπο, τοποθετούνται ακριβώς κάτω από τη ρίζα του δένδρου που αναπαριστά το owl:Thing και περιλαμβάνουν τη λέξη Thing ως όνομα κλάσης.
- Για κάθε άτομο που έχει περισσότερους από έναν τύπους δημιουργούνται ισάριθμοι κόμβοι με το πλήθος των τύπων του. Καθένας από αυτούς τους κόμβους, οι οποίοι περιλαμβάνουν τα ονόματα όλων των τύπων ως όνομα κλάσης και το εικονίδιο 11, τοποθετείται ακριβώς κάτω από κάθε βοηθητικό κόμβο που αντιστοιχεί σε τύπο του ατόμου.

Το δένδρο ατόμων περιέχει ένα αναδυόμενο μενού (με τη χρήση του δεξί κλικ του ποντικιού) παρέχοντας τις παρακάτω λειτουργίες οι οποίες μπορεί να προκαλέσουν μια μερική αναδιάρθρωση του δένδρου και οι οποίες μπορούν να επιλεγθούν και από το μενού "Edit" του βασικού μενού της εφαρμογής:

1. Εισαγωγή νέου ατόμου στην τρέχουσα οντολογία για την οποία απαιτείται αρχικά ο ορισμός του τύπου του και στη συνέχεια η εισαγωγή του rdf:ID του. Σε περίπτωση που στον τύπο (κλάση) του νέου ατόμου δεν ανήκει κανένα άλλο άτομο της οντολογίας, τότε, πέρα από τον κόμβο του νέου ατόμου, δημιουργείται και ο βοηθητικός κόμβος που θα αντιστοιχεί στον τύπο αυτό.
2. Διαγραφή υπάρχοντος ατόμου από την τρέχουσα οντολογία (για τη λειτουργία αυτή απαιτείται η επιλογή ενός κόμβου του δένδρου που να αντιστοιχεί σε άτομο). Σε περίπτωση που στον τύπο (κλάση) του διαγραφέντος ατόμου δεν ανήκει κανένα άλλο άτομο της οντολογίας, τότε, πέρα από τον κόμβο του διαγραφέντος ατόμου, διαγράφεται και ο βοηθητικός κόμβος που αντιστοιχεί στον τύπο αυτό.



Εικόνα 66: Οι διαθέσιμες επιλογές του αναδυόμενου μενού του δένδρου ατόμων.

Με την επιλογή ενός κόμβου από το δένδρο, πλην φυσικά της ρίζας και των βοηθητικών κόμβων που αντιστοιχούν σε κλάσεις εμφανίζονται οι πληροφορίες του αντίστοιχου ατόμου. Η εμφάνιση αυτών των πληροφοριών γίνεται μέσω των τριών όψεων που περιγράφονται στις επόμενες παραγράφους:

- Όψη γενικών χαρακτηριστικών του επιλεγμένου ατόμου.
- Όψη ιδιοτήτων τύπου δεδομένων του επιλεγμένου ατόμου
- Όψη ιδιοτήτων αντικειμένων του επιλεγμένου ατόμου

6.5.2. Όψη γενικών χαρακτηριστικών του επιλεγμένου ατόμου

Η *όψη γενικών χαρακτηριστικών* του επιλεγμένου ατόμου η οποία περιλαμβάνει τα παρακάτω στοιχεία:

1. *Πεδίο επεξεργασίας rdf:ID ατόμου*: Το πεδίο αυτό περιέχει το πλήρες URI του επιλεγμένου ατόμου. Επιτρέπει μόνο την αλλαγή του rdf:ID και όχι του πλήρους URI του ατόμου. Κάθε πιθανή αλλαγή του rdf:ID μπορεί να επιφέρει αναδιάταξη κάποιων κόμβων του δένδρου ατόμων, αφού αυτοί ταξινομούνται αλφαβητικά.
2. *Λίστα τύπων ατόμου*: Η λίστα αυτή περιέχει τους τύπους (κλάσεις) στους οποίους ανήκει το επιλεγμένο άτομο. Υποστηρίζεται η πρόσθεση νέου τύπου στο επιλεγμένο άτομο και η αφαίρεση υπάρχοντος τύπου. Κάθε τέτοια προσθαφαίρεση επιφέρει φυσικά τροποποιήσεις στο πλήθος και στη τοποθέτηση των κόμβων στο δένδρο ατόμων.
3. *Λίστα ετικετών ατόμου*: Η λίστα αυτή περιέχει τις ετικέτες που έχουν επισυναφθεί με το επιλεγμένο άτομο. Επιτρέπεται η πρόσθεση νέας και η διαγραφή υπάρχουσας ετικέτας.
4. *Λίστα σχολίων ατόμου*: Η λίστα αυτή περιέχει τα σχόλια που έχουν επισυναφθεί με το επιλεγμένο άτομο. Επιτρέπεται η πρόσθεση νέου και η διαγραφή υπάρχοντος σχολίου.
5. *Λίστα με ακριβώς όμοια άτομα του ατόμου*: Η λίστα αυτή περιλαμβάνει τα ακριβώς όμοια άτομα με το επιλεγμένο άτομο τα οποία μπορούν να βρίσκονται και σε διαφορετικές από την τρέχουσα οντολογίες. Υποστηρίζεται η πρόσθεση νέου ακριβώς όμοιου ατόμου το οποίο μπορεί να επιλεγεί μέσα από όλες τις ανοιχτές οντολογίες και η διαγραφή υπάρχοντος ακριβώς όμοιου ατόμου.
6. *Λίστα με διαφορετικά άτομα του ατόμου*: Η λίστα αυτή περιλαμβάνει τα διαφορετικά άτομα με το επιλεγμένο άτομο τα οποία μπορούν να βρίσκονται και σε διαφορετικές από την τρέχουσα οντολογίες. Υποστηρίζεται η πρόσθεση νέου διαφορετικού ατόμου το οποίο μπορεί να επιλεγεί μέσα από όλες τις ανοιχτές οντολογίες και η διαγραφή υπάρχοντος διαφορετικού ατόμου. Η λίστα αυτή μαζί με τη λίστα ακριβώς ομοίων ατόμων ουσιαστικά παρέχουν δυνατότητες εξατομίκευσης στο χρήστη καθώς μπορεί να δημιουργήσει μια δικιά του οντολογία με ονόματα ατόμων της επιλογής του και να αντιστοιχήσει τα άτομα

αυτά με άτομα υπάρχοντων και γνωστών οντολογιών κληρονομώντας τα χαρακτηριστικά τους.

6.5.3. Όψη ιδιοτήτων τύπου δεδομένων του επιλεγμένου ατόμου

Η *όψη ιδιοτήτων τύπου δεδομένων* του επιλεγμένου ατόμου δεν περιέχει στατικά στοιχεία αλλά λίστες οι οποίες δημιουργούνται δυναμικά για κάθε επιλεγμένο άτομο και επιτρέπουν την εισαγωγή τιμών σε ιδιότητες τύπου δεδομένων. Συγκεκριμένα, για κάθε ιδιότητα τύπου δεδομένων που εφαρμόζεται στην κλάση (ή κλάσεις) που ανήκει το επιλεγμένο άτομο, δημιουργείται δυναμικά μια αντίστοιχη λίστα που επιτρέπει την:

1. Προσθήκη τιμών (κυριολεκτικές τιμές) του ατόμου για τη συγκεκριμένη ιδιότητα τύπου δεδομένων οι οποίες πρέπει να είναι σύμφωνες με το εύρος τιμών και τους τυχόν περιορισμούς της ιδιότητας (βλέπε αναλυτικά επόμενη παράγραφο).
2. Αφαίρεση υπάρχοντων τιμών (κυριολεκτικές τιμές) του ατόμου για τη συγκεκριμένη ιδιότητα τύπου δεδομένων.

6.5.4. Όψη ιδιοτήτων αντικειμένων του επιλεγμένου ατόμου

Η *όψη ιδιοτήτων αντικειμένων* του επιλεγμένου ατόμου δεν περιέχει στατικά στοιχεία αλλά λίστες οι οποίες δημιουργούνται δυναμικά για κάθε επιλεγμένο άτομο και επιτρέπουν την εισαγωγή τιμών σε ιδιότητες αντικειμένων. Συγκεκριμένα, για κάθε ιδιότητα αντικειμένων που εφαρμόζεται στην κλάση (ή κλάσεις) που ανήκει το επιλεγμένο άτομο, δημιουργείται δυναμικά μια αντίστοιχη λίστα που επιτρέπει την:

1. Πρόσθεση τιμών (OWL άτομα) του ατόμου για τη συγκεκριμένη ιδιότητα αντικειμένων οι οποίες πρέπει να είναι σύμφωνες με το εύρος τιμών και τους τυχόν περιορισμούς της ιδιότητας (βλέπε αναλυτικά επόμενη παράγραφο). Η πρόσθεση νέας τιμής μπορεί να γίνει με επιλογή κάποιου από τα υπάρχοντα άτομα της οντολογίας που πληρεί τις παραπάνω προϋποθέσεις ή με τη δημιουργία νέου ατόμου του οποίου ο τύπος θα πληρεί αυτές τις προϋποθέσεις. Στην τελευταία περίπτωση, το νέο άτομο που δημιουργείται προστίθεται στην οντολογία και συνεπώς και στο δένδρο ατόμων.
2. Αφαίρεση υπάρχοντων τιμών (OWL άτομα) του ατόμου για τη συγκεκριμένη ιδιότητα αντικειμένων.

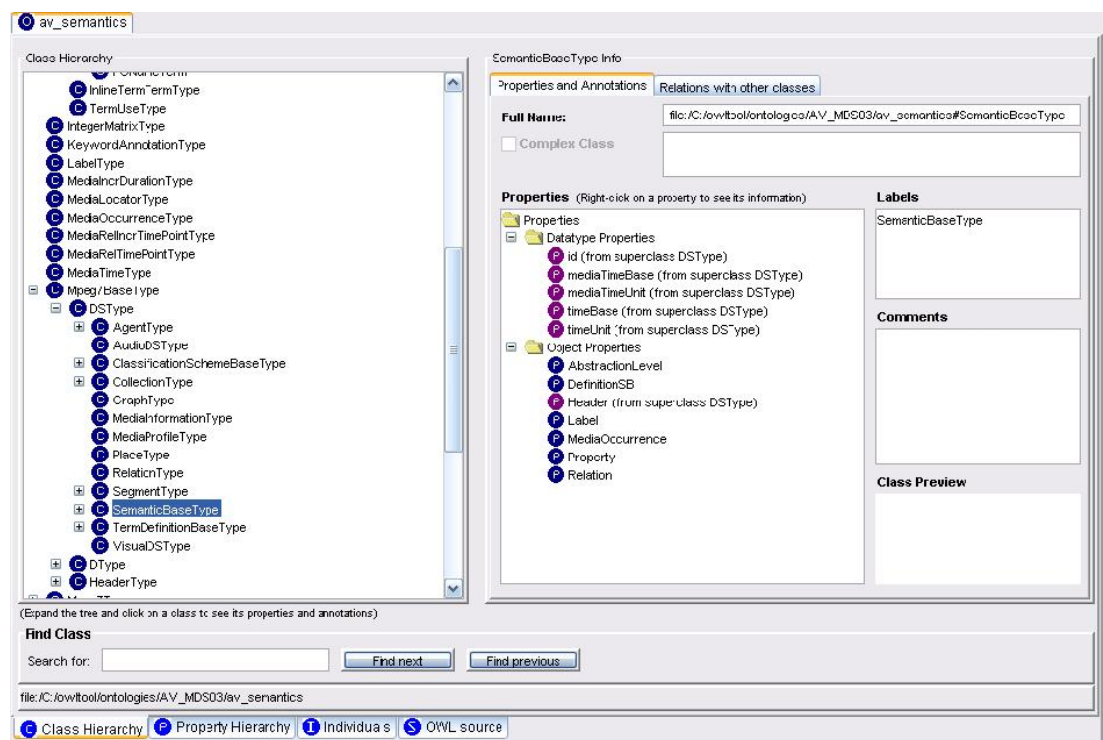
6.6. Διάκριση σε ανώτερες οντολογίες και σε οντολογίες περιοχής

Το GraphOnto παρέχει τη δυνατότητα διάκρισης των οντολογιών σε δύο είδη: α) Ανώτερες (Upper) και β) Περιοχής (Domain), τις οποίες διαχειρίζεται με διαφορετικό τρόπο. Οι ανώτερες οντολογίες είναι συνήθως οντολογίες που αναπαριστούν κάποια εγκεκριμένα πρότυπα (standards) ή κάποια ευρέως γνωστά και αποδεκτά μοντέλα. Για το λόγο αυτό και προκειμένου να μην απωλεσθεί πολύτιμη πληροφορία από τις ανώτερες οντολογίες, το σύστημα επιτρέπει μόνο το άνοιγμα και τη γραφική αναπαράσταση τους ώστε να είναι δυνατή μόνο η περιήγηση σε αυτές και όχι η επεξεργασία και αποθήκευσή τους. Η πλήρης επεξεργασία οντολογιών που περιγράφηκε στις προηγούμενες παραγράφους επιτρέπεται μόνο στις οντολογίες περιοχής, οι οποίες βέβαια μπορούν να χρησιμοποιήσουν κλάσεις των ανώτερων οντολογιών ως υπερκλάσεις των δικών τους κλάσεων.

Για το άνοιγμα και την εμφάνιση μιας ανώτερης οντολογίας απαιτείται η εισαγωγή της στο περιβάλλον εργασίας η οποία γίνεται επιλέγοντας από το μενού File την επιλογή "Import Core Ontologies" και κατόπιν εισάγοντας το URI της ανώτερης οντολογίας ή τη θέση της στο τοπικό σύστημα αρχείων. Η εισαγωγή μιας ανώτερης οντολογίας στο περιβάλλον εργασίας πρέπει να προηγείται του ανοίγματος οντολογίας περιοχής ή δημιουργίας νέας οντολογίας περιοχής (Να σημειωθεί εδώ ότι οι τρόποι ανοίγματος οντολογίας που περιγράφηκαν στην παράγραφο 6.2 αφορούν οντολογίες περιοχής). Για κάθε ανοιχτή ανώτερη οντολογία, αντιστοιχεί μια καρτέλα ακριβώς όμοια με εκείνη μιας οντολογίας περιοχής η οποία έχει ως τίτλο το όνομα της και το εικονίδιο 2 και το οποίο περιέχει στο κάτω μέρος του το πλήρες URI της οντολογίας. Η καρτέλα αυτή ως γνωστόν, περιέχει τέσσερις εμφωλευμένες καρτέλες που η κάθε μια περιλαμβάνει τις τέσσερις διαφορετικές όψεις που επιτρέπουν την περιήγηση (browsing) στην τρέχουσα οντολογία. Οι όψεις αυτές είναι παρόμοιες με τις όψεις μιας οντολογίας περιοχής έχοντας μόνο τις εξής διαφορές:

- Δίπλα σε κάθε πεδίο των όψεων δεν περιλαμβάνονται τα κουμπιά προσθαφαίρεσης Add / Remove που επιτρέπουν την επεξεργασία του αντίστοιχου πεδίου, αφού μόνο η περιήγηση επιτρέπεται σε ανώτερη οντολογία.
- Τα εικονίδια που χρησιμοποιούνται στο δένδρο ιεραρχίας κλάσεων, στο δένδρο ιδιοτήτων κλάσης, στο δένδρο ιεραρχίας ιδιοτήτων και στο δένδρο ατόμων έχουν διαφορετικό χρώμα από αυτά των οντολογιών περιοχής (περισσότερες λεπτομέρειες για τα χρησιμοποιούμενα εικονίδια υπάρχουν στο παράρτημα).

Η εμφάνιση της όψης κλάσεων μιας ανώτερης οντολογίας που είναι αντιπροσωπευτική και των άλλων όψεων φαίνεται στην παρακάτω εικόνα:



Εικόνα 67: Η όψη κλάσεων μιας ανώτερης οντολογίας.

6.7. Έλεγχος ορθότητας πληροφορίας

6.7.1. Εισαγωγή

Κατά τη διάρκεια της επεξεργασίας των οντολογιών, αρκετοί σημασιολογικοί έλεγχοι λαμβάνουν χώρα από τον Διαχειριστή Μοντέλων Οντολογιών (όπως αναφέρεται στην παράγραφο 5.2.3) έτσι ώστε να μην παραβιάζεται η συνέπεια των οντολογιών. Όλοι αυτοί οι περιορισμοί επιτρέπουν στο GraphOnto να δημιουργεί γενικά σωστές και συνεπείς οντολογίες χωρίς να είναι απαραίτητη η χρήση μιας OWL μηχανής εξαγωγής συμπερασμάτων (Reasoner). Ο χρήστης καθοδηγείται σωστά από το γραφικό σύστημα διεπαφής, το οποίο σε κάποιες περιπτώσεις ειδοποιεί το χρήστη με μικρά μηνύματα λάθους και σε κάποιες εμφανίζει μόνο τις σωστές δυνατές επιλογές.

Στο επίπεδο επεξεργασίας όλων των OWL δομών μιας οντολογίας διενεργείται ο παρακάτω έλεγχος από το σύστημα: Κάθε rdf:ID (όνομα) OWL δομής ανεξαρτήτου τύπου είναι μοναδικό μέσα στην οντολογία αφού αυτό το όνομα αντιστοιχεί σε ένα URI και συνεπώς δεν επιτρέπεται σε μια OWL δομή (κλάση, ιδιότητα ή άτομο) να έχει ίδιο όνομα με μια άλλη OWL δομή.

Οι σημασιολογικοί έλεγχοι για τις δομές της OWL περιγράφονται στις επόμενες παραγράφους.

6.7.2. Σημασιολογικοί έλεγχοι σε OWL κλάσεις

Σε επίπεδο επεξεργασίας OWL κλάσεων, οι παρακάτω σημασιολογικοί έλεγχοι διενεργούνται από το σύστημα:

- Έστω κλάσεις A και B και η A είναι υποκλάση της B. Στην περίπτωση αυτή δεν επιτρέπεται η A να είναι υπερκλάση, ισοδύναμη ή ξένη της B.
- Έστω κλάσεις A και B και η A είναι υπερκλάση της B. Στην περίπτωση αυτή δεν επιτρέπεται η A να είναι υποκλάση, ισοδύναμη ή ξένη της B.
- Έστω κλάσεις A και B και η A είναι ισοδύναμη της B. Στην περίπτωση αυτή δεν επιτρέπεται η A να είναι υπερκλάση, υποκλάση ή ξένη της B.
- Έστω κλάσεις A και B και η A είναι ξένη της B. Στην περίπτωση αυτή δεν επιτρέπεται η A να είναι υπερκλάση, υποκλάση ή ισοδύναμη της B.

6.7.3. Σημασιολογικοί έλεγχοι σε OWL περιορισμούς ιδιοτήτων

Στο επίπεδο ορισμού των περιορισμών στις ιδιότητες των κλάσεων, πραγματοποιούνται οι παρακάτω έλεγχοι:

- Το πεδίο τιμών των περιορισμών `allValuesFrom` ή `someValuesFrom` για κάποια ιδιότητα αντικειμένων δεν επιτρέπεται να ξεπερνά το εύρος που ορίζει το πεδίο τιμών της ιδιότητας. Πιο συγκεκριμένα:
 - Αν το πεδίο τιμών της ιδιότητας αποτελείται από κλάσεις και επιλεχθεί το πεδίο τιμών του περιορισμού να αποτελείται από κλάσεις, τότε επιτρέπονται μόνο οι κλάσεις του πεδίου τιμών της ιδιότητας και οι τυχόν υποκλάσεις τους.
 - Αν το πεδίο τιμών της ιδιότητας αποτελείται από κλάσεις και επιλεχθεί το πεδίο τιμών του περιορισμού να αποτελείται από άτομα κλάσεων, τότε επιτρέπονται μόνο εκείνα τα άτομα που ανήκουν στις κλάσεις του πεδίου τιμών της ιδιότητας και τις τυχόν υποκλάσεις τους.
 - Αν το πεδίο τιμών της ιδιότητας αποτελείται από άτομα κλάσεων, τότε το πεδίο τιμών του περιορισμού επιτρέπεται να περιέχει μόνο κάποια από τα άτομα κλάσεων του πεδίου τιμών της ιδιότητας.
- Η τιμή (OWL άτομο) του περιορισμού `hasValue` για κάποια ιδιότητα αντικειμένων πρέπει να ανήκει στο εύρος που ορίζει το πεδίο τιμών της ιδιότητας. Πιο συγκεκριμένα:
 - Αν το πεδίο τιμών της ιδιότητας αποτελείται από κλάσεις, τότε η τιμή (OWL άτομο) πρέπει να ανήκει σε κάποια από τις κλάσεις του πεδίου τιμών της ιδιότητας ή στις τυχόν υποκλάσεις τους.
 - Αν το πεδίο τιμών της ιδιότητας αποτελείται από OWL άτομα, τότε η τιμή (OWL άτομο) πρέπει να είναι κάποιο από τα OWL άτομα του πεδίου τιμών της ιδιότητας.

- Η τιμή (value) του περιορισμού `hasValue` για κάποια ιδιότητα τύπου δεδομένων που έχει ως πεδίο τιμών απλές τιμές πρέπει να είναι κάποια από αυτές τις τιμές.
- Αν σε μια κλάση *A* ορίζεται ένας περιορισμός `allValuesFrom` ή `someValuesFrom` για κάποια ιδιότητα αντικειμένων, τότε οι υποκλάσεις της δεν επιτρέπεται να έχουν δικό τους περιορισμό για την ίδια ιδιότητα που το πεδίο τιμών του να ξεπερνά το εύρος που ορίζει το πεδίο τιμών του περιορισμού για την ιδιότητα που ορίζεται στην κλάση *A*.
- Αν σε μια κλάση *A* ορίζεται ένας περιορισμός πολλαπλότητας για κάποια ιδιότητα, τότε οι υποκλάσεις της δεν επιτρέπεται να έχουν δικό τους περιορισμό πολλαπλότητας για την ίδια ιδιότητα που το εύρος του να ξεπερνά το εύρος του περιορισμού για την ιδιότητα που ορίζεται στην κλάση *A*. Για παράδειγμα, αν η κλάση *A* έχει στην ιδιότητα *P* περιορισμό μέγιστης πολλαπλότητας ίσο με 2, οι υποκλάσεις της δεν μπορούν να έχουν περιορισμό πολλαπλότητας μεγαλύτερο του 2 στην ιδιότητα *P*.
- Δεν επιτρέπεται να οριστεί περιορισμός πολλαπλότητας σε μια ιδιότητα που έχει οριστεί ως συναρτησιακή.

6.7.4. Σημασιολογικοί έλεγχοι σε OWL ιδιότητες

Στο επίπεδο επεξεργασίας OWL ιδιοτήτων, οι παρακάτω σημασιολογικοί έλεγχοι διενεργούνται από το σύστημα:

- Έστω ιδιότητες *A* και *B* και η *A* είναι υποιδιότητα της *B*. Στην περίπτωση αυτή δεν επιτρέπεται η *A* να είναι υπεριδιότητα, ισοδύναμη ή αντίστροφη (μόνο για ιδιότητες αντικειμένων) της *B*.
- Έστω ιδιότητες *A* και *B* και η *A* είναι υπεριδιότητα της *B*. Στην περίπτωση αυτή δεν επιτρέπεται η *A* να είναι υποιδιότητα, ισοδύναμη ή αντίστροφη (μόνο για ιδιότητες αντικειμένων) της *B*.
- Έστω ιδιότητες *A* και *B* και η *A* είναι ισοδύναμη της *B*. Στην περίπτωση αυτή δεν επιτρέπεται η *A* να είναι υπεριδιότητα, υποιδιότητα ή αντίστροφη (μόνο για ιδιότητες αντικειμένων) της *B*.
- Έστω ιδιότητες αντικειμένων *A* και *B* και η *A* είναι αντίστροφη της *B*. Στην περίπτωση αυτή δεν επιτρέπεται η *A* να είναι υπεριδιότητα, υποιδιότητα ή ισοδύναμη της *B*.
- Έστω ιδιότητα αντικειμένων που έχει δηλωθεί ως συναρτησιακή. Στην περίπτωση αυτή δεν μπορεί να δηλωθεί ταυτόχρονα ως μεταβατική.
- Έστω ιδιότητα αντικειμένων που έχει δηλωθεί ως αντίστροφα συναρτησιακή. Στην περίπτωση αυτή δεν μπορεί να δηλωθεί ταυτόχρονα ως μεταβατική.
- Έστω ιδιότητα αντικειμένων που έχει δηλωθεί ως μεταβατική. Στην περίπτωση αυτή δεν μπορεί να δηλωθεί ταυτόχρονα ως συναρτησιακή ή αντίστροφα συναρτησιακή.

6.7.5. Σημασιολογικοί έλεγχοι σε OWL άτομα

Στο επίπεδο επεξεργασίας OWL ατόμων, οι παρακάτω σημασιολογικοί έλεγχοι διενεργούνται από το σύστημα:

- Έστω OWL άτομα A και B και το A είναι ακριβώς όμοιο με το B. Στην περίπτωση αυτή δεν επιτρέπεται το A να είναι διαφορετικό από το B.
- Έστω OWL άτομα A και B και το A είναι διαφορετικό από το B. Στην περίπτωση αυτή δεν επιτρέπεται το A να είναι ακριβώς όμοιο με το B.
- Έστω κλάσεις A και B και η A είναι ξένη με την B. Στην περίπτωση αυτή δεν επιτρέπεται σε κανένα OWL άτομο να ανήκει και στις δύο κλάσεις.
- Δεν είναι δυνατός ο ορισμός νέου OWL ατόμου που να ανήκει σε μια κλάση η οποία είναι σύνθετη και αποτελείται από απαρίθμηση (enumeration) OWL ατόμων.
- Όταν σε μια κλάση C υπάρχει περιορισμός εύρους τιμών του τύπου “έχει τιμή” για μια συγκεκριμένη ιδιότητα τύπου δεδομένων P, τότε σε κάθε άτομο της κλάσης C δεν επιτρέπεται να υπάρχουν διαφορετικές τιμές στην ιδιότητα P από την κυριολεκτική τιμή που ορίζει ο περιορισμός.
- Αν δεν ισχύει περιορισμός εύρους τιμών σε μια ιδιότητα τύπου δεδομένων P που εφαρμόζεται σε ένα άτομο, τότε ισχύει ένα από τα παρακάτω:
 - Αν η ιδιότητα P έχει ως εύρος τιμών έναν XML τύπο δεδομένων, τότε οι τιμές του ατόμου στην ιδιότητα P δεν επιτρέπεται να έχουν διαφορετικό τύπο από το συγκεκριμένο XML τύπο δεδομένων.
 - Αν η ιδιότητα P έχει ως εύρος τιμών μια απαρίθμηση κυριολεκτικών τιμών, τότε οι τιμές του ατόμου στην ιδιότητα P δεν επιτρέπεται να μην περιέχονται μέσα σε αυτήν την απαρίθμηση.
- Όταν σε μια κλάση C υπάρχει περιορισμός εύρους τιμών του τύπου “όλες οι τιμές από” για μια συγκεκριμένη ιδιότητα αντικειμένων P, τότε σε κάθε άτομο της κλάσης C:
 - Σε περίπτωση που ο περιορισμός περιλαμβάνει κλάσεις, δεν επιτρέπεται να υπάρχουν τιμές (άτομα) στην ιδιότητα P που να μην ανήκουν στις κλάσεις (και τις τυχόν υποκλάσεις τους) που ορίζει ο περιορισμός.
 - Σε περίπτωση που ο περιορισμός περιλαμβάνει άτομα, δεν επιτρέπεται να υπάρχουν τιμές (άτομα) στην ιδιότητα P που να μην ανήκουν στην απαρίθμηση ατόμων που ορίζει ο περιορισμός.
- Όταν σε μια κλάση C υπάρχει περιορισμός εύρους τιμών του τύπου “έχει τιμή” για μια συγκεκριμένη ιδιότητα αντικειμένων P, τότε σε κάθε άτομο της κλάσης C δεν επιτρέπεται να υπάρχουν διαφορετικές τιμές στην ιδιότητα P από την τιμή (άτομο) που ορίζει ο περιορισμός.
- Αν δεν ισχύει περιορισμός εύρους τιμών σε μια ιδιότητα αντικειμένων P που εφαρμόζεται σε ένα άτομο A, τότε ισχύει ένα από τα παρακάτω:

- Αν η ιδιότητα P έχει ως εύρος τιμών κλάσεις, τότε οι τιμές (υποκείμενα) του ατόμου A στην ιδιότητα P δεν επιτρέπεται να μην ανήκουν στις κλάσεις (ή τις τυχόν υποκλάσεις) του εύρους τιμών.
- Αν η ιδιότητα P έχει ως εύρος τιμών μια απαρίθμηση ατόμων, τότε οι τιμές του ατόμου A στην ιδιότητα P δεν επιτρέπεται να μην περιέχονται μέσα σε αυτήν την απαρίθμηση.
- Όταν σε μια κλάση C υπάρχει περιορισμός εύρους τιμών του τύπου “έχει τιμή” για μια συγκεκριμένη ιδιότητα τύπου δεδομένων P , τότε σε κάθε άτομο της κλάσης C δεν επιτρέπεται να υπάρχουν διαφορετικές τιμές στην ιδιότητα P από την κυριολεκτική τιμή που ορίζει ο περιορισμός.
- Όταν σε μια κλάση C υπάρχει περιορισμός πολλαπλότητας ελάχιστου πλήθους για μια συγκεκριμένη ιδιότητα P , τότε σε όλα τα άτομα της κλάσης C δεν επιτρέπεται να υπάρχουν λιγότερες τιμές στην ιδιότητα P από το ελάχιστο πλήθος που ορίζει ο περιορισμός.
- Όταν σε μια κλάση C υπάρχει περιορισμός πολλαπλότητας ακριβούς πλήθους για μια συγκεκριμένη ιδιότητα P , τότε σε όλα τα άτομα της κλάσης C δεν επιτρέπεται να υπάρχουν λιγότερες ή περισσότερες τιμές στην ιδιότητα P από το ακριβές πλήθος που ορίζει ο περιορισμός.
- Όταν σε μια κλάση C υπάρχει περιορισμός πολλαπλότητας μέγιστου πλήθους για μια συγκεκριμένη ιδιότητα P , τότε σε όλα τα άτομα της κλάσης C δεν επιτρέπεται να υπάρχουν περισσότερες τιμές στην ιδιότητα P από το μέγιστο πλήθος που ορίζει ο περιορισμός.
- Όταν σε ένα άτομο A εφαρμόζεται μια συναρτησιακή ιδιότητα P , τότε στο άτομο A δεν επιτρέπεται να οριστούν παραπάνω από μία τιμές στην ιδιότητα P .
- Όταν σε ένα σύνολο ατόμων εφαρμόζεται μια αντίστροφα συναρτησιακή ιδιότητα P , τότε κάθε άτομο του συνόλου πρέπει να έχει αναγκαστικά μοναδική τιμή στην ιδιότητα P .

6.8. Αυτόματες ενημερώσεις κατά την επεξεργασία οντολογιών

6.8.1. Εισαγωγή

Κατά τη διάρκεια της επεξεργασίας των οντολογιών, πέρα από τους αρκετούς σημασιολογικούς ελέγχους που περιγράφηκαν παραπάνω, ο Διαχειριστής Μοντέλων Οντολογιών εξάγει κάποια λογικά συμπεράσματα με βάση την πληροφορία των οντολογιών και ενημερώνει αυτόματα τις δομές της οντολογίας. Αυτές οι αυτόματες ενημερώσεις μαζι φυσικά με τους παραπάνω περιορισμούς ισχυροποιούν ακόμα περισσότερο το GraphOnto στη

δημιουργία γενικά σωστών και συνεπών οντολογιών χωρίς να είναι απαραίτητη η χρήση μιας OWL μηχανής εξαγωγής συμπερασμάτων (Reasoner).

Οι αυτόματες ενημερώσεις για τις δομές της OWL περιγράφονται στις επόμενες παραγράφους.

6.8.2. Αυτόματες ενημερώσεις σε OWL κλάσεις

Σε επίπεδο επεξεργασίας OWL κλάσεων, οι παρακάτω αυτόματες ενημερώσεις διενεργούνται από το σύστημα:

- Όταν σε μια κλάση A προστεθεί μια υπερκλάση B, τότε η κλάση A και όλες οι υποκλάσεις της κληρονομούν αυτόματα τις ιδιότητες και τους περιορισμούς ιδιοτήτων της κλάσης B.
- Όταν σε μια κλάση A προστεθεί μια υποκλάση B, τότε η κλάση B και όλες οι υποκλάσεις της κληρονομούν αυτόματα τις ιδιότητες και τους περιορισμούς ιδιοτήτων της κλάσης A.
- Όταν σε μια κλάση A αφαιρεθεί μια υπερκλάση B, τότε από την κλάση A και τις υποκλάσεις της αφαιρούνται αυτόματα οι ιδιότητες και οι περιορισμοί ιδιοτήτων που είχαν κληρονομηθεί από την κλάση B. Παράλληλα, αφαιρούνται όλες οι τιμές αυτών των κληρονομηθέντων ιδιοτήτων και από όλα τα άτομα της κλάσης A και των υποκλάσεών της.
- Όταν σε μια κλάση A αφαιρεθεί μια υποκλάση B, τότε από την κλάση B και τις υποκλάσεις της αφαιρούνται αυτόματα οι ιδιότητες και οι περιορισμοί ιδιοτήτων που είχαν κληρονομηθεί από την κλάση A. Παράλληλα, αφαιρούνται όλες οι τιμές αυτών των κληρονομηθέντων ιδιοτήτων και από όλα τα άτομα της κλάσης B και των υποκλάσεών της.
- Όταν μια κλάση A είναι ισοδύναμη με μια κλάση B και οριστεί A ισοδύναμη με μια κλάση C, τότε αυτόματα ορίζεται ότι και η κλάση B είναι ισοδύναμη με τη C.
- Όταν διαγραφεί μια κλάση A από την οντολογία, τότε αυτόματα:
 - Διαγράφονται όλες οι τυχόν σχέσεις (υπερκλάση, υποκλάση, ισοδύναμη κλάση ή ξένη κλάση) των υπόλοιπων κλάσεων με την κλάση A.
 - Διαγράφεται η κλάση A από όλα τα πεδία ορισμού ιδιοτήτων (τύπου δεδομένων και αντικειμένων) που τυχόν την περιέχουν. Σε περίπτωση που το πεδίο ορισμού κάποιας ιδιότητας περιλαμβάνει μόνο την κλάση A, τότε το πεδίο ορισμού αυτής της ιδιότητας παραμένει κενό.
 - Διαγράφεται η κλάση A από όλα τα εύρη τιμών ιδιοτήτων αντικειμένων που τυχόν την περιέχουν. Σε περίπτωση που το εύρος τιμών κάποιας ιδιότητας περιλαμβάνει μόνο την κλάση A, τότε το εύρος τιμών αυτής της ιδιότητας παραμένει κενό.
 - Διαγράφεται η κλάση A από όλα τα εύρη τιμών των περιορισμών σε ιδιότητες αντικειμένων που τυχόν την περιέχουν. Σε περίπτωση που το εύρος τιμών

κάποιου περιορισμού περιλαμβάνει μόνο την κλάση A, τότε διαγράφεται και ο ίδιος ο περιορισμός από την κλάση που τον περιέχει.

- ο Διαγράφεται η κλάση A από όλους τους τύπους ατόμων που τυχόν την περιέχουν. Σε περίπτωση που ο τύπος κάποιου ατόμου περιλαμβάνει μόνο την κλάση A, τότε τύπος αυτού του ατόμου γίνεται το owl:Thing.

6.8.3. Αυτόματες ενημερώσεις σε OWL ιδιότητες

Σε επίπεδο επεξεργασίας OWL ιδιοτήτων, οι παρακάτω αυτόματες ενημερώσεις διενεργούνται από το σύστημα:

- Όταν μια ιδιότητα A είναι ισοδύναμη με μια ιδιότητα B και οριστεί A ισοδύναμη με μια ιδιότητα C, τότε αυτόματα ορίζεται ότι και η ιδιότητα B είναι ισοδύναμη με τη C.
- Όταν διαγραφεί μια ιδιότητα A από την οντολογία, τότε αυτόματα:
 - ο Διαγράφονται όλες οι τυχόν σχέσεις (υπεριδιότητα, υποιδιότητα, ισοδύναμη ιδιότητα ή αντίστροφη ιδιότητα) των υπόλοιπων ιδιοτήτων με την ιδιότητα A.
 - ο Διαγράφεται η ιδιότητα A από όλες τις κλάσεις του πεδίου ορισμού της και τις τυχόν υποκλάσεις τους. Ταυτόχρονα σε όλα τα άτομα που ανήκουν στις παραπάνω κλάσεις διαγράφονται όλες οι τιμές που έχουν στην ιδιότητα A.
 - ο Διαγράφονται από τις κλάσεις όλοι οι τυχόν περιορισμοί (πολλαπλότητας και εύρους τιμών) που μπορεί να έχουν στην ιδιότητα A.

6.8.4. Αυτόματες ενημερώσεις σε OWL άτομα

Σε επίπεδο επεξεργασίας OWL ατόμων, οι παρακάτω αυτόματες ενημερώσεις διενεργούνται από το σύστημα:

- Όταν ένα άτομο A είναι ακριβώς όμοιο με ένα άτομο B και οριστεί A ακριβώς όμοιο με ένα άτομο C, τότε αυτόματα ορίζεται ότι και το άτομο B είναι ακριβώς όμοιο με το C.
- Όταν διαγραφεί ένα άτομο A από την οντολογία, τότε αυτόματα:
 - ο Διαγράφονται όλες οι τυχόν σχέσεις (ακριβώς όμοιο ή διαφορετικό) των υπόλοιπων ατόμων με το άτομο A.
 - ο Διαγράφεται το άτομο A από όλα τα εύρη τιμών ιδιοτήτων αντικειμένων που τυχόν το περιέχουν. Σε περίπτωση που το εύρος τιμών κάποιας ιδιότητας περιλαμβάνει μόνο το άτομο A, τότε το εύρος τιμών αυτής της ιδιότητας παραμένει κενό.
 - ο Διαγράφεται το άτομο A από όλα τα εύρη τιμών των περιορισμών σε ιδιότητες αντικειμένων που τυχόν το περιέχουν. Σε περίπτωση που το εύρος τιμών κάποιου περιορισμού περιλαμβάνει μόνο το άτομο A, τότε διαγράφεται και ο ίδιος ο περιορισμός από την κλάση που τον περιέχει.

- Διαγράφεται το άτομο A από όλα τα άτομα τα οποία το περιέχουν ως τιμές σε ιδιότητες αντικειμένων τους.
- Όταν σε ένα άτομο A προστεθεί σε μια συμμετρική ιδιότητα P ως τιμή το άτομο B, τότε αυτόματα και στο άτομο B προστίθεται στην ιδιότητα P ως τιμή το άτομο A.
- Όταν σε ένα άτομο A σε μια συμμετρική ιδιότητα αντικειμένων P αφαιρεθεί η τιμή άτομο B, τότε αυτόματα και στο άτομο B στην ιδιότητα P αφαιρείται αν υπάρχει η τιμή άτομο A.
- Όταν σε ένα άτομο A προστεθεί σε μια ιδιότητα αντικειμένων P1, που έχει ως αντίστροφη την ιδιότητα P2, ως τιμή το άτομο B, τότε αυτόματα και στο άτομο B προστίθεται στην ιδιότητα P2 ως τιμή το άτομο A.
- Όταν σε ένα άτομο A σε μια ιδιότητα αντικειμένων P1, που έχει ως αντίστροφη την ιδιότητα P2, αφαιρεθεί η τιμή άτομο B, τότε αυτόματα και στο άτομο B στην ιδιότητα P2 αφαιρείται η τιμή άτομο A.

6.9. Λειτουργικότητα αναζήτησης OWL δομών

6.9.1. Εισαγωγή

Οι OWL οντολογίες συνήθως περιέχουν μεγάλη ποσότητα πληροφορίας και όταν ανοιχθούν από το GraphOnto είναι πολύ πιθανόν να δημιουργηθούν δένδρα ιεραρχίας κλάσεων, ιεραρχίας ιδιοτήτων και ατόμων με μεγάλο μήκος και βάθος και παράλληλα η OWL μορφή κειμένου της οντολογίας να εκτείνεται σε πολυάριθμες γραμμές κειμένου. Τα παραπάνω καθιστούν δύσκολη και χρονοβόρα την αναζήτηση από το χρήστη μιας συγκεκριμένης OWL δομής. Για το λόγο αυτό, το GraphOnto παρέχει μέσα και στις τέσσερις διαφορετικές όψεις της οντολογίας μια λειτουργικότητα αναζήτησης που απλοποιεί τον τρόπο και εξαλείφει το χρόνο αναζήτησης.

Πιο συγκεκριμένα, μέσα σε κάθε όψη υπάρχει ένα πεδίο αναζήτησης στο οποίο συμπληρώνεται η επιθυμητή λέξη προς αναζήτηση και δύο κουμπιά πλοήγησης για μετακίνηση προς τα κάτω ή προς τα πάνω. Για την έναρξη της αναζήτησης αρκεί μόνο η πληκτρολόγηση λέξης τουλάχιστον δύο χαρακτήρων και δεν απαιτείται κάποια επιπλέον ενέργεια. Σε περίπτωση αποτυχημένης αναζήτησης κατάλληλο μήνυμα εμφανίζεται. Στις επόμενες παραγράφους αναλύονται οι αναζητήσεις σε κλάσεις, ιδιότητες, άτομα και OWL μορφή.

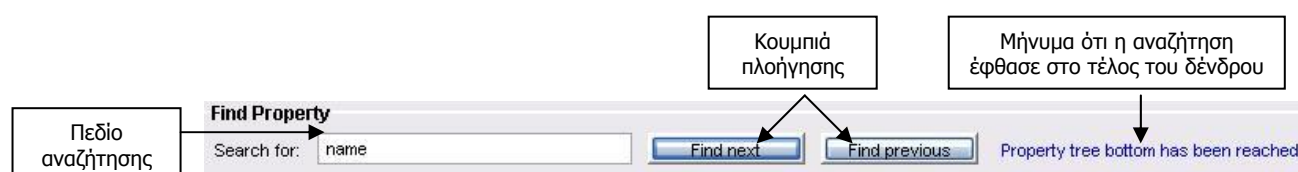
6.9.2. Αναζήτηση OWL κλάσεων

Στην όψη κλάσεων, η λέξη αναζήτησης αναζητείται διασχίζοντας pre-order το δένδρο ιεραρχίας κλάσεων και ψάχνοντας για κόμβο κλάσης της οποίας το rdf:ID να περιέχει (όχι απαραίτητα στην αρχή) τη ζητούμενη λέξη. Στην αναζήτηση δε λαμβάνεται υπόψη η ρίζα του δένδρου (ο κόμβος που αντιστοιχεί στο owl:Thing). Αν βρεθεί τέτοιος κόμβος, η

αναζήτηση σταματά και ο συγκεκριμένος κόμβος επιλέγεται και συνεπώς εμφανίζονται τα χαρακτηριστικά της κλάσης στην οποία αντιστοιχεί. Με τη βοήθεια των κουμπιών πλοήγησης μπορεί να συνεχιστεί η αναζήτηση της λέξης προχωρώντας σύμφωνα ή αντίθετα με την pre-order διάσχιση του δένδρου ιεραρχίας κλάσεων μέχρι να φθάσει στο τέλος ή στην αρχή του οπότε και το σύστημα ειδοποιεί με κατάλληλο μήνυμα.

6.9.3. Αναζήτηση OWL ιδιοτήτων

Στην όψη ιδιοτήτων, η λέξη αναζήτησης αναζητείται διασχίζοντας pre-order το δένδρο ιεραρχίας ιδιοτήτων (συνεπώς λόγω της μορφολογίας του δένδρου η αναζήτηση γίνεται πρώτα στις ιδιότητες τύπου δεδομένων και στη συνέχεια στις ιδιότητες αντικειμένων) και ψάχνοντας για κόμβο ιδιότητας της οποίας το rdf:ID να περιέχει (όχι απαραίτητα στην αρχή) τη ζητούμενη λέξη. Στην αναζήτηση δε λαμβάνονται υπόψη οι βοηθητικοί κόμβοι. Αν βρεθεί τέτοιος κόμβος, η αναζήτηση σταματά και ο συγκεκριμένος κόμβος επιλέγεται και συνεπώς εμφανίζονται τα χαρακτηριστικά της ιδιότητας στην οποία αντιστοιχεί. Με τη βοήθεια των κουμπιών πλοήγησης μπορεί να συνεχιστεί η αναζήτηση της λέξης προχωρώντας σύμφωνα ή αντίθετα με την pre-order διάσχιση του δένδρου ιεραρχίας ιδιοτήτων μέχρι να φθάσει στο τέλος ή στην αρχή του οπότε και το σύστημα ειδοποιεί με κατάλληλο μήνυμα.



Εικόνα 68: Η φόρμα αναζήτησης στην όψη ιδιοτήτων στην οποία φαίνεται ότι αναζήτηση ιδιότητας έφθασε στο τέλος του δένδρου ιεραρχίας ιδιοτήτων.



Εικόνα 69: Η φόρμα αναζήτησης στην όψη ιδιοτήτων στην οποία φαίνεται ότι αναζήτηση ιδιότητας δεν είχε αποτελέσματα.

6.9.4. Αναζήτηση OWL ατόμων

Στην όψη ατόμων, η λέξη αναζήτησης αναζητείται διασχίζοντας pre-order το δένδρο ατόμων και ψάχνοντας για κόμβο ατόμου του οποίου το rdf:ID να περιέχει (όχι απαραίτητα στην αρχή) τη ζητούμενη λέξη. Στην αναζήτηση δε λαμβάνονται υπόψη η ρίζα του δένδρου και οι βοηθητικοί κόμβοι. Αν βρεθεί τέτοιος κόμβος, η αναζήτηση σταματά και ο συγκεκριμένος κόμβος επιλέγεται και συνεπώς εμφανίζονται τα χαρακτηριστικά της ιδιότητας στην οποία αντιστοιχεί. Με τη βοήθεια των κουμπιών πλοήγησης μπορεί να συνεχιστεί η αναζήτηση της λέξης προχωρώντας σύμφωνα ή αντίθετα με την pre-order διάσχιση του

δένδρου ατόμων μέχρι να φθάσει στο τέλος ή στην αρχή του οπότε και το σύστημα ειδοποιεί με κατάλληλο μήνυμα.

6.9.5. Αναζήτηση κειμένου στην OWL μορφή

Στην όψη OWL μορφής, η λέξη αναζήτησης αναζητείται διασχίζοντας το κείμενο της OWL μορφής και ψάχνοντας τη ζητούμενη λέξη μέσα στο κείμενο. Αν βρεθεί η λέξη, η αναζήτηση σταματά και η υπογραμμίζεται η πρώτη εμφάνιση της λέξης στο κείμενο. Με τη βοήθεια των κουμπιών πλοήγησης μπορεί να συνεχιστεί η αναζήτηση της λέξης προχωρώντας μέσα στο κείμενο προς τα κάτω ή προς τα πάνω μέχρι να φθάσει στο τέλος ή στην αρχή του κειμένου οπότε και το σύστημα ειδοποιεί με κατάλληλο μήνυμα.

6.10.Ανακεφαλαίωση

Στο κεφάλαιο αυτό παρουσιάστηκε η πλήρης και πλούσια παρεχομένη λειτουργικότητα του GraphOnto ως ένα αμιγές και πλήρες σύστημα διαχείρισης OWL-DL οντολογιών παρέχοντας πλήρη λειτουργικότητα για διαχείριση όλων των OWL δομών (κλάσεων, ιδιοτήτων τύπου δεδομένων, ιδιοτήτων αντικειμένων και ατόμων). Ταυτόχρονα δείχθηκε ότι χάρη στους σημασιολογικούς ελέγχους και τις αυτόματες ενημερώσεις που λαμβάνουν χώρα κατά τη διάρκεια επεξεργασίας μιας οντολογίας, παράγονται γενικά σωστές και έγκυρες οντολογίες και μειώνεται η ανάγκη χρήσης μιας μηχανής παραγωγής συμπερασμάτων.

Κεφάλαιο 7^ο

7. Λειτουργικότητα εξατομίκευσης χρήσης και απόκρυψης πληροφορίας

7.1. Εισαγωγή

Όπως αναφέρθηκε και στο 1^ο κεφάλαιο το GraphOnto σχεδιάστηκε για να παρέχει πέρα από την πλήρη λειτουργικότητα της OWL-DL, και κατάλληλους μηχανισμούς για να αποκρύπτει τις πολύπλοκες έννοιες που ενδέχεται να περιέχουν οι οντολογίες και να επιτρέπει τη χρήση του και από λιγότερο έμπειρους χρήστες που επιθυμούν μόνο τη δημιουργία σημασιολογικών περιγραφών βασισμένες στο περιεχόμενο οντολογιών. Ταυτόχρονα, στόχος της σχεδίασης ήταν και η παροχή δυνατότητας εξατομίκευσης του περιβάλλοντος εργασίας του χρήστη.

Οι δύο αυτές απαιτήσεις καλύφθηκαν μέσω των δύο παρακάτω επιπρόσθετων χαρακτηριστικών του GraphOnto τα οποία το διαφοροποιούν σημαντικά από τα υπόλοιπα γραφικά συστήματα διαχείρισης OWL οντολογιών και τα οποία είναι:

1. Η ανεξάρτητη σημασιολογική περιγραφή εκτός πλαισίου οντολογίας (που περιγράφεται στην ενότητα 7.2).
2. Τα GraphOnto XML αρχεία προσαρμογής (που περιγράφονται στην ενότητα 7.3).

7.2. Ανεξάρτητη σημασιολογική περιγραφή εκτός πλαισίου οντολογίας

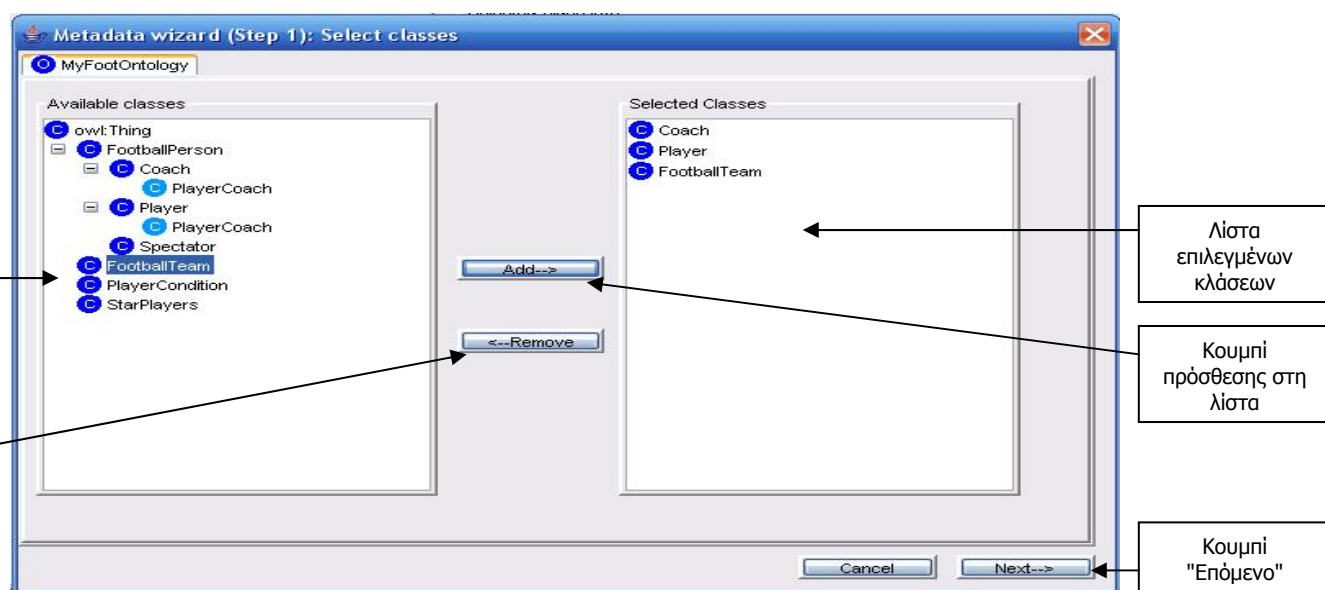
7.2.1. Εισαγωγή

Το GraphOnto παρέχει τη δυνατότητα δημιουργίας νέων αυτόνομων και εξατομικευμένων σημασιολογικών περιγραφών μέσω του ορισμού νέων OWL ατόμων έξω από το πλαίσιο και την εμβέλεια μιας οντολογίας. Τα άτομα αυτά έχουν μοναδικό συγκεκριμένο τύπο και επιλεγμένες ιδιότητες. Αυτή η λειτουργία υποστηρίζεται μέσω του *παραθύρου μεταδεδομένων*.

7.2.2. Οδηγός δημιουργίας παράθυρου μεταδεδομένων

Για τη δημιουργία του παραθύρου μεταδεδομένων, απαιτείται το άνοιγμα ενός οδηγού (wizard) μέσω του μενού "Metadata" του βασικού μενού της εφαρμογής. Ο οδηγός ακολουθεί τα δύο παρακάτω βήματα:

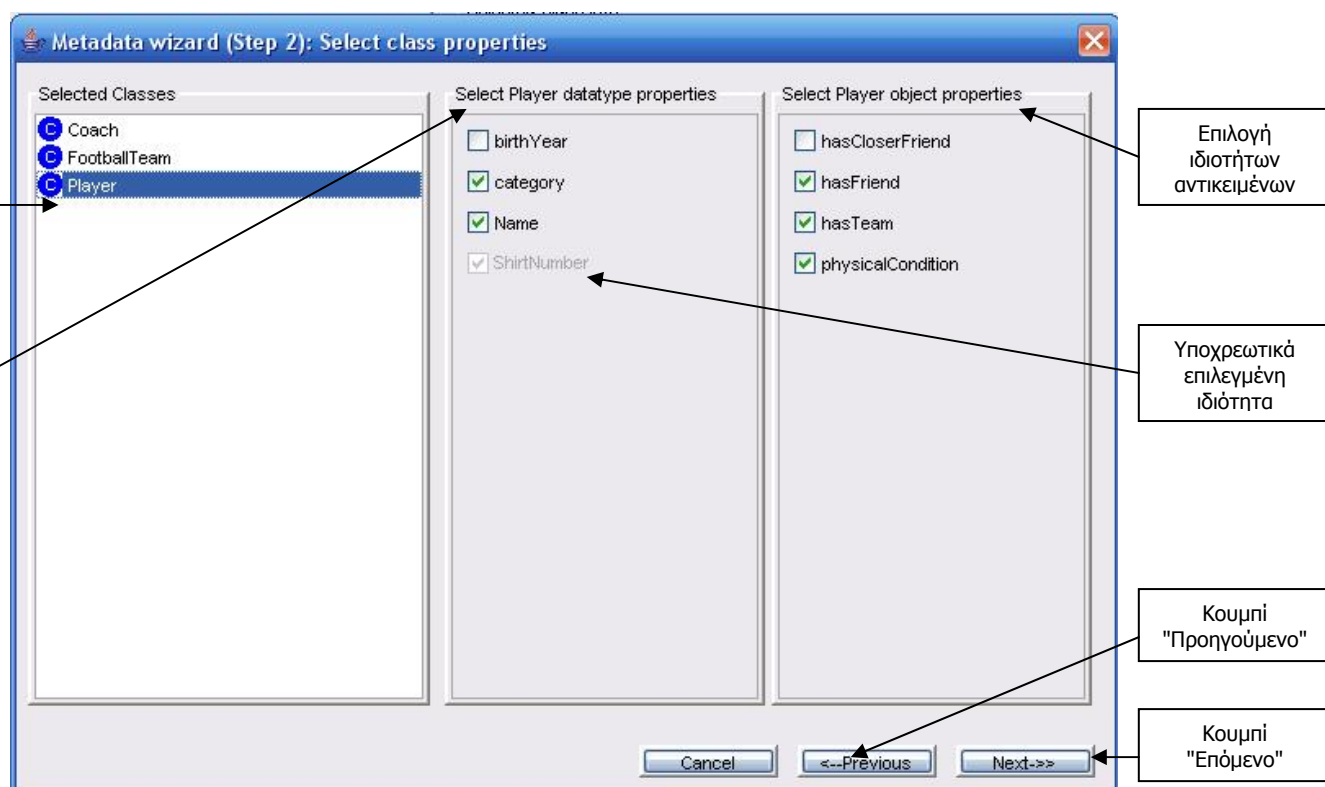
1. Επιλογή κλάσεων: Στο πρώτο παράθυρο του οδηγού επιλέγονται ανάμεσα από όλες τις κλάσεις των ανοιχτών οντολογιών οι επιθυμητές κλάσεις που θα αποτελέσουν τους τύπους των νέων ατόμων που θα δημιουργηθούν. Για κάθε ανοιχτή οντολογία εμφανίζεται το δένδρο ιεραρχίας κλάσεων της και μια λίστα η οποία θα περιέχει τις επιλεγμένες κλάσεις. Οι σημασιολογικοί έλεγχοι που αναφέρθηκαν σε προηγούμενη παράγραφο εφαρμόζονται σε κάθε περίπτωση και συνεπώς ανάμεσα στο σύνολο των επιλεγμένων κλάσεων δεν επιτρέπεται να περιέχονται σύνθετες κλάσεις που ορίζονται με απευθείας απαρίθμηση ατόμων. Τα στοιχεία της λίστας αυτής προσθαφαιρούνται με τη χρήση δύο κατάλληλων κουμπιών. Τέλος, η μεταπήδηση στο επόμενο βήμα γίνεται με τη χρήση του κουμπιού "Επόμενο".



Εικόνα 70: Το 1^ο βήμα του οδηγού δημιουργίας παραθύρου μεταδεδομένων.

2. Επιλογή ιδιοτήτων: Στο δεύτερο παράθυρο του οδηγού για κάθε επιλεγμένη κλάση του 1^{ου} βήματος επιλέγονται ανάμεσα από όλες τις ιδιότητες της οι επιθυμητές ιδιότητες στις οποίες θα περιέχουν τιμές τα νέα άτομα που θα δημιουργηθούν. Δίδεται βέβαια η δυνατότητα επιστροφής στο 1^ο βήμα του οδηγού για τυχόν τροποποίηση των επιλεγμένων κλάσεων με τη χρήση του κουμπιού "Προηγούμενο". Όλες αυτές οι επιλεγμένες κλάσεις τοποθετούνται σε μια λίστα και για κάθε τέτοια κλάση εμφανίζονται όλες οι ιδιότητες τύπου δεδομένων και αντικειμένων που της αντιστοιχούν. Δίπλα σε κάθε όνομα ιδιότητας υπάρχει ένα κουμπι επιλογής που επιτρέπει την επιλογή ή αποεπιλογή της συγκεκριμένης ιδιότητας. Κουμπι που αντιστοιχεί σε ιδιότητα για την οποία ισχύει περιορισμός πολλαπλότητας ελάχιστου ή ακριβούς πλήθους μεγαλύτερο του μηδενός (0), είναι εκ των προτέρων επιλεγμένο και απενεργοποιημένο ώστε να καθίσταται υποχρεωτική η επιλογή αυτής της ιδιότητας για λόγους συνέπειας των νέων ατόμων που θα δημιουργηθούν. Όταν ολοκληρωθεί η επιλογή του χρήστη, η μεταπήδηση στο επόμενο

βήμα γίνεται με τη χρήση του κουμπιού "Επόμενο", ο οδηγός τερματίζεται και εμφανίζεται το αυτόματα παραγόμενο παράθυρο μεταδεδομένων.



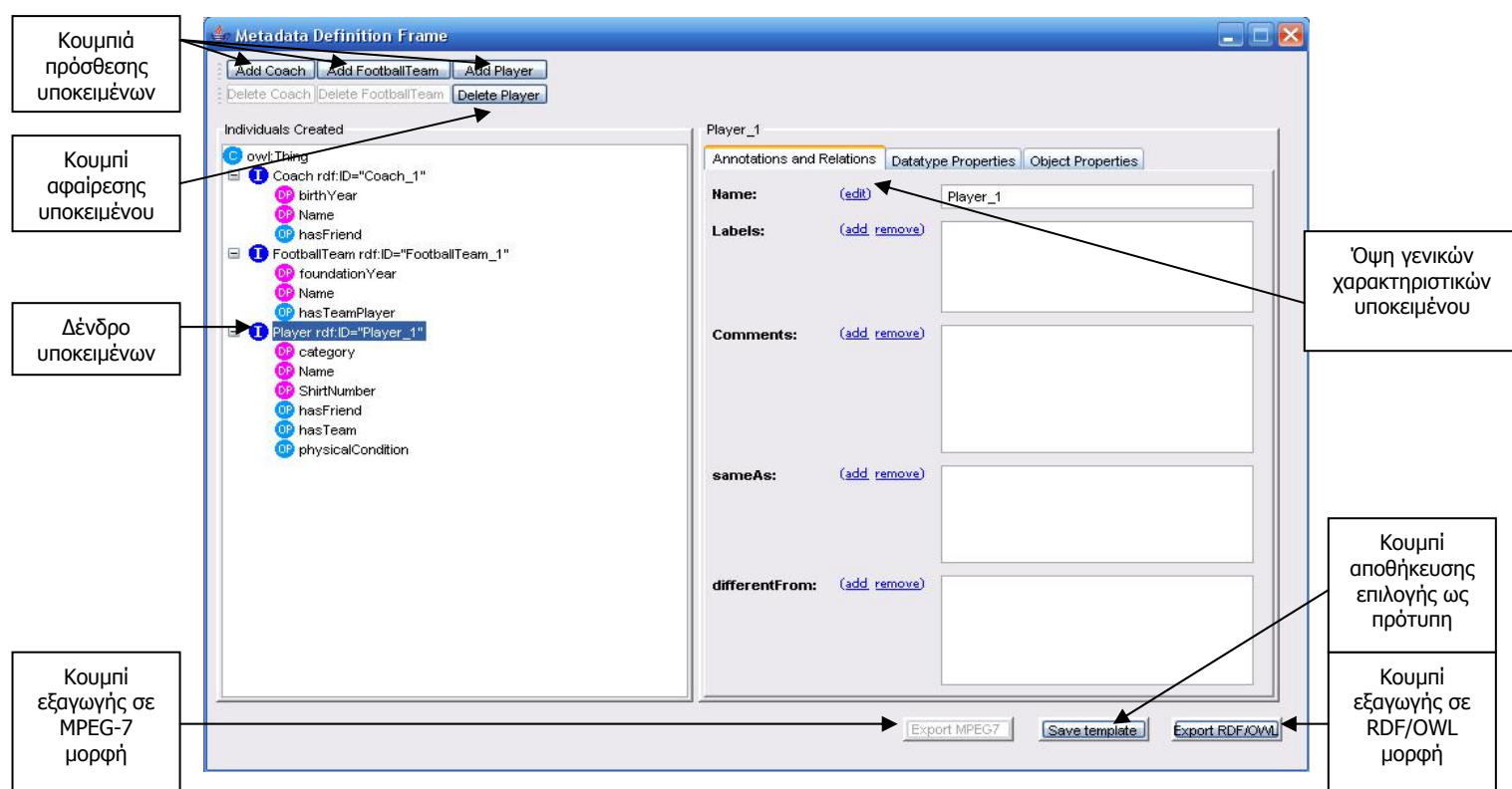
Εικόνα 71: Το 2^ο βήμα του οδηγού δημιουργίας παραθύρου μεταδεδομένων.

7.2.3. Κατασκευή παράθυρου μεταδεδομένων

Το παράθυρο μεταδεδομένων που εμφανίζεται είναι προσαρμοσμένο ακριβώς στις επιλογές κλάσεων και ιδιοτήτων που προέκυψαν μέσω του οδηγού και μοιάζει πάρα πολύ στην όψη ατόμων οντολογίας. Πιο συγκεκριμένα, η κατασκευή του βασίζεται στους εξής κανόνες:

- Δημιουργείται ένα δένδρο ατόμων με ρίζα το owl:Thing.
- Δημιουργείται για κάθε επιλεγμένη κλάση ένα πρότυπο άτομο με rdf:ID του στυλ όνομακλάσης_I (όπου I είναι ένας αύξων αριθμός που αντιστοιχεί σε κάθε επιλεγμένη κλάση και ο οποίος αρχικά έχει την τιμή 1 και αυξάνεται κάθε φορά που προστίθεται ένα άτομο της επιλεγμένης κλάσης στη σημασιολογική περιγραφή). Για το πρότυπο αυτό άτομο προστίθεται ως παιδί της ρίζας στο δένδρο ατόμων ένας νέος κόμβος που αντιστοιχεί σε αυτό το άτομο και που αποτελείται από το rdf:ID της κλάσης, το rdf:ID του πρότυπου ατόμου και το εικονίδιο 10.
- Για κάθε επιλεγμένη ιδιότητα τύπου δεδομένων μιας επιλεγμένης κλάσης, δημιουργείται αντίστοιχος κόμβος που αποτελείται από το rdf:ID της ιδιότητας και το εικονίδιο 15 και ο οποίος προστίθεται ως παιδί του κόμβου πρότυπου ατόμου που έχει τύπο την επιλεγμένη κλάση.

- Για κάθε επιλεγμένη ιδιότητα αντικειμένων μιας επιλεγμένης κλάσης, δημιουργείται αντίστοιχος κόμβος που αποτελείται από το `rdf:ID` της ιδιότητας και το εικονίδιο 16 και ο οποίος προστίθεται ως παιδί του κόμβου πρότυπου ατόμου που έχει τύπο την επιλεγμένη κλάση.
- Για κάθε επιλεγμένη κλάση, δημιουργείται ένα κουμπί που επιτρέπει την πρόσθεση στην τρέχουσα σημασιολογική περιγραφή νέου ατόμου με `rdf:ID` του στυλ όνομακλάσης_I που θα έχει τύπο την επιλεγμένη κλάση και ένα κουμπί που επιτρέπει τη διαγραφή ενός συγκεκριμένου ατόμου που έχει τύπο την επιλεγμένη κλάση.



Εικόνα 72: Η αρχική όψη του παραθύρου μεταδεδομένων που προέκυψε ύστερα από τα δύο βήματα του οδηγού.

Με την επιλογή ενός κόμβου από το δένδρο που αντιστοιχεί σε πρότυπο άτομο, εμφανίζονται οι πληροφορίες του αντίστοιχου ατόμου. Η εμφάνιση αυτών των πληροφοριών γίνεται μέσω τριών όψεων (που περιγράφονται στις επόμενες παραγράφους) όπως ακριβώς και στην όψη ατόμων οντολογίας.

7.2.4. Όψη γενικών χαρακτηριστικών επιλεγμένου ατόμου

Η *όψη γενικών χαρακτηριστικών* του επιλεγμένου ατόμου περιλαμβάνει ορισμένα στοιχεία επεξεργασίας των χαρακτηριστικών και των σχέσεων του ατόμου (τα στοιχεία αυτά έχουν την ίδια ακριβώς λειτουργία όπως και στην όψη ατόμων οντολογίας και για το λόγο αυτό θα περιγραφούν συνοπτικά). Αξίζει να παρατηρηθεί ότι στο παράθυρο μεταδεδομένων απουσιάζει η λίστα τύπων ατόμου, καθώς οι τύποι των ατόμων έχουν προκαθοριστεί από την

επιλογή κλάσεων του οδηγού και δεν μπορούν να διαφοροποιηθούν. Τα στοιχεία λοιπόν που περιλαμβάνονται είναι:

1. *Πεδίο επεξεργασίας rdf:ID ατόμου*: Το πεδίο αυτό περιέχει το rdf:ID του επιλεγμένου ατόμου και επιτρέπει την αλλαγή του.
2. *Λίστα ετικετών ατόμου*: Η λίστα αυτή περιέχει τις ετικέτες που έχουν επισυναφθεί με το επιλεγμένο άτομο. Επιτρέπεται η πρόσθεση νέας και η διαγραφή υπάρχουσας ετικέτας.
3. *Λίστα σχολίων ατόμου*: Η λίστα αυτή περιέχει τα σχόλια που έχουν επισυναφθεί με το επιλεγμένο άτομο. Επιτρέπεται η πρόσθεση νέου και η διαγραφή υπάρχοντος σχολίου.
4. *Λίστα με ακριβώς όμοια άτομα του ατόμου*: Η λίστα αυτή περιλαμβάνει τα ακριβώς όμοια άτομα με το επιλεγμένο άτομο. Υποστηρίζεται η πρόσθεση νέου ακριβώς όμοιου ατόμου το οποίο μπορεί να επιλεγεί ανάμεσα στην τρέχουσα σημασιολογική περιγραφή ή στις ανοιχτές οντολογίες και η διαγραφή υπάρχοντος ακριβώς όμοιου ατόμου.
5. *Λίστα με διαφορετικά άτομα του ατόμου*: Η λίστα αυτή περιλαμβάνει τα διαφορετικά άτομα με το επιλεγμένο άτομο. Υποστηρίζεται η πρόσθεση νέου διαφορετικού ατόμου το οποίο μπορεί να επιλεγεί ανάμεσα στην τρέχουσα σημασιολογική περιγραφή ή στις ανοιχτές οντολογίες και η διαγραφή υπάρχοντος διαφορετικού ατόμου. Η λίστα αυτή μαζί με τη λίστα ακριβώς ομοίων ατόμων ουσιαστικά παρέχουν δυνατότητες εξατομίκευσης στο χρήστη καθώς μπορεί να αντιστοιχήσει τα άτομα της τρέχουσας σημασιολογικής περιγραφής με άτομα υπάρχοντων και γνωστών οντολογιών κληρονομώντας τα χαρακτηριστικά τους.

7.2.5. Όψη ιδιοτήτων τύπου δεδομένων επιλεγμένου ατόμου

Η *όψη ιδιοτήτων τύπου δεδομένων* του επιλεγμένου ατόμου δεν περιέχει στατικά στοιχεία αλλά λίστες οι οποίες δημιουργούνται δυναμικά για κάθε άτομο και επιτρέπουν την εισαγωγή τιμών στις επιλεγμένες ιδιότητες τύπου δεδομένων του ατόμου που προέκυψαν από τον οδηγό δημιουργίας παράθυρου μεταδεδομένων. Συγκεκριμένα, για κάθε επιλεγμένη ιδιότητα τύπου δεδομένων του επιλεγμένου ατόμου, δημιουργείται δυναμικά μια αντίστοιχη λίστα που επιτρέπει την:

1. Προσθεση τιμών (κυριολεκτικές τιμές) του ατόμου για τη συγκεκριμένη ιδιότητα τύπου δεδομένων οι οποίες πρέπει να είναι σύμφωνες με το εύρος τιμών και τους τυχόν περιορισμούς της ιδιότητας. Για κάθε νέα τιμή που προστίθεται στην ιδιότητα, δημιουργείται ένας αντίστοιχος κόμβος που περιέχει τη νέα τιμή και το εικονίδιο 17 και ο οποίος προστίθεται στο δένδρο ατόμων ως παιδί του κόμβου της συγκεκριμένης ιδιότητας τύπου δεδομένων στο επιλεγμένο άτομο.

2. Αφαίρεση υπάρχοντων τιμών (κυριολεκτικές τιμές) του ατόμου για τη συγκεκριμένη ιδιότητα τύπου δεδομένων. Η αφαίρεση μιας ορισμένης τιμής συνεπάγεται και την απομάκρυνση του αντίστοιχου κόμβου της από το δένδρο ατόμων.

7.2.6. Όψη ιδιοτήτων αντικειμένων επιλεγμένου ατόμου

Η *όψη ιδιοτήτων αντικειμένων* του επιλεγμένου ατόμου δεν περιέχει στατικά στοιχεία αλλά λίστες οι οποίες δημιουργούνται δυναμικά για κάθε άτομο και επιτρέπουν την εισαγωγή τιμών στις επιλεγμένες ιδιότητες αντικειμένων του ατόμου που προέκυψαν από τον οδηγό δημιουργίας παράθυρου μεταδεδομένων. Συγκεκριμένα, για κάθε επιλεγμένη ιδιότητα αντικειμένων του επιλεγμένου ατόμου, δημιουργείται δυναμικά μια αντίστοιχη λίστα που επιτρέπει την:

1. Πρόσθεση τιμών (OWL άτομα) του ατόμου για τη συγκεκριμένη ιδιότητα αντικειμένων οι οποίες πρέπει να είναι σύμφωνες με το εύρος τιμών και τους τυχόν περιορισμούς της ιδιότητας. Η πρόσθεση νέας τιμής μπορεί να γίνει με επιλογή κάποιου από τα υπάρχοντα άτομα της τρέχουσας σημασιολογικής περιγραφής ή των ανοιχτών οντολογιών που πληρεί τις προϋποθέσεις του εύρους τιμών και των περιορισμών ή με τη δημιουργία νέου ατόμου του οποίου ο τύπος θα πληρεί επίσης αυτές τις προϋποθέσεις. Στην τελευταία περίπτωση, το νέο άτομο που δημιουργείται προστίθεται στην τρέχουσα σημασιολογική περιγραφή. Για κάθε νέα τιμή (άτομο) που προστίθεται στην ιδιότητα, δημιουργείται ένας αντίστοιχος κόμβος που περιέχει το `rdf:ID` του τύπου του ατόμου, το `rdf:ID` του ατόμου και το εικονίδιο 10 και ο οποίος προστίθεται στο δένδρο ατόμων ως παιδί του κόμβου της συγκεκριμένης ιδιότητας αντικειμένων στο επιλεγμένο άτομο. Για τον κόμβο αυτό δημιουργείται και το υποδένδρο του σύμφωνα με τους παρακάτω κανόνες:

- Αν ο κόμβος αυτός αντιστοιχεί σε υπάρχον άτομο της σημασιολογικής περιγραφής, τότε το υποδένδρο του είναι κενό για λόγους οικονομίας μνήμης καθώς το υποδένδρο του θα είναι αναπτυγμένο σε άλλο σημείο του δένδρου ατόμων.
- Αν ο κόμβος αυτός αντιστοιχεί σε υπάρχον άτομο οντολογίας και υπάρχει ήδη κόμβος στο δένδρο που να αντιστοιχεί σε αυτό το άτομο, τότε το υποδένδρο του είναι κενό για λόγους οικονομίας μνήμης καθώς το υποδένδρο του θα είναι αναπτυγμένο σε άλλο σημείο του δένδρου ατόμων.
- Αν ο κόμβος αυτός αντιστοιχεί σε υπάρχον άτομο οντολογίας και δεν υπάρχει άλλος κόμβος στο δένδρο που να αντιστοιχεί σε αυτό το άτομο, τότε το υποδένδρο αυτού του ατόμου αναπτύσσεται δημιουργώντας κόμβους για τις ιδιότητες τύπου δεδομένων και αντικειμένων στις οποίες περιέχει τιμές και κόμβους για τις ίδιες τις τιμές. Οι τιμές αυτές εμφανίζονται για λόγους πληρότητας της πληροφορίας του ατόμου, όμως δεν επιτρέπεται να τροποποιηθούν ή να προσθαφαιρεθούν τιμές σε αυτό το άτομο. Για τις τιμές του (άτομα) στις

ιδιότητες αντικειμένων δημιουργούνται με τον ίδιο τρόπο αντίστοιχοι κόμβοι και τα υποδένδρα αναπτύσσονται τους σύμφωνα με τους τρέχοντες κανόνες.

- Αν ο κόμβος αυτός αντιστοιχεί σε νέο άτομο της σημασιολογικής περιγραφής και ο τύπος στον οποίο ανήκει βρίσκεται ανάμεσα στις επιλεγμένες κλάσεις, τότε το υποδένδρο του αναπτύσσεται δημιουργώντας κόμβους για τις επιλεγμένες ιδιότητες τύπου δεδομένων και αντικειμένων για το συγκεκριμένο τύπο.
- Αν ο κόμβος αυτός αντιστοιχεί σε νέο άτομο της σημασιολογικής περιγραφής και ο τύπος στον οποίο ανήκει δε βρίσκεται ανάμεσα στις επιλεγμένες κλάσεις, τότε για να αναπτυχθεί το υποδένδρο του απαιτείται επιλογή για τις ιδιότητες του συγκεκριμένου τύπου. Έτσι εμφανίζεται ένα νέο παράθυρο που περιέχει τις ιδιότητες τύπου δεδομένων και αντικειμένων που εφαρμόζονται σε αυτόν τον τύπο ώστε να γίνει η επιλογή των επιθυμητών ιδιοτήτων.



Εικόνα 73: Η όψη του παραθύρου επιλογής ιδιοτήτων για τύπο που δεν είχε επιλεγεί ύστερα από τα δύο βήματα του οδηγού.

Αφού ολοκληρωθεί η επιλογή τότε το υποδένδρο του ατόμου αναπτύσσεται δημιουργώντας κόμβους για τις επιλεγμένες ιδιότητες τύπου δεδομένων και αντικειμένων για το συγκεκριμένο τύπο.

2. Αφαίρεση υπάρχοντων τιμών (άτομα) του ατόμου για τη συγκεκριμένη ιδιότητα αντικειμένων. Η αφαίρεση μιας ορισμένης τιμής συνεπάγεται και την απομάκρυνση του αντίστοιχου κόμβου της από παιδί του κόμβου που αντιστοιχεί στη συγκεκριμένη ιδιότητα του ατόμου, όχι όμως απαραίτητα και από το δένδρο ατόμων. Η επιλογή σχετικά με το αν θα παραμείνει ο κόμβος αυτός στο δένδρο ατόμων γίνεται με βάση τους παρακάτω κανόνες:

- a) Αν ο κόμβος προς διαγραφή δεν έχει ανεπτυγμένο υποδένδρο, τότε ο κόμβος απομακρύνεται από το δένδρο ατόμων.

- b) Αν ο κόμβος προς διαγραφή έχει ανεπτυγμένο υποδένδρο και το άτομο που αντιστοιχεί σε αυτόν ανήκει στην τρέχουσα σημασιολογική περιγραφή τότε ο κόμβος προς διαγραφή παραμένει στο δένδρο και:
- Τοποθετείται ως παιδί της ρίζας (owl:Thing) αν δεν υπάρχει άλλος κόμβος στο δένδρο που να αντιστοιχεί στο άτομο.
 - Ενώ αν υπάρχει και άλλος κόμβος N στο δένδρο που να αντιστοιχεί στο άτομο (ο οποίος προφανώς δεν θα έχει ανεπτυγμένο υποδένδρο σύμφωνα με τους κανόνες προσθήκης τιμών σε ιδιότητες αντικειμένων), τότε ο κόμβος προς διαγραφή μαζί με το υποδένδρο του αντικαθιστά τον κόμβο N στο δένδρο.
- c) Αν ο κόμβος προς διαγραφή έχει ανεπτυγμένο υποδένδρο και το άτομο που αντιστοιχεί σε αυτόν ανήκει σε οντολογία τότε:
- Ο κόμβος προς διαγραφή απομακρύνεται από το δένδρο αν δεν υπάρχει άλλος κόμβος στο δένδρο που να αντιστοιχεί στο άτομο.
 - Ενώ αν υπάρχει και άλλος κόμβος N στο δένδρο που να αντιστοιχεί στο άτομο (ο οποίος προφανώς δεν θα έχει ανεπτυγμένο υποδένδρο σύμφωνα με τους κανόνες προσθήκης τιμών σε ιδιότητες αντικειμένων), τότε ο κόμβος προς διαγραφή μαζί με το υποδένδρο του αντικαθιστά τον κόμβο N στο δένδρο.

7.2.7. Τελική μορφή δένδρου ατόμων

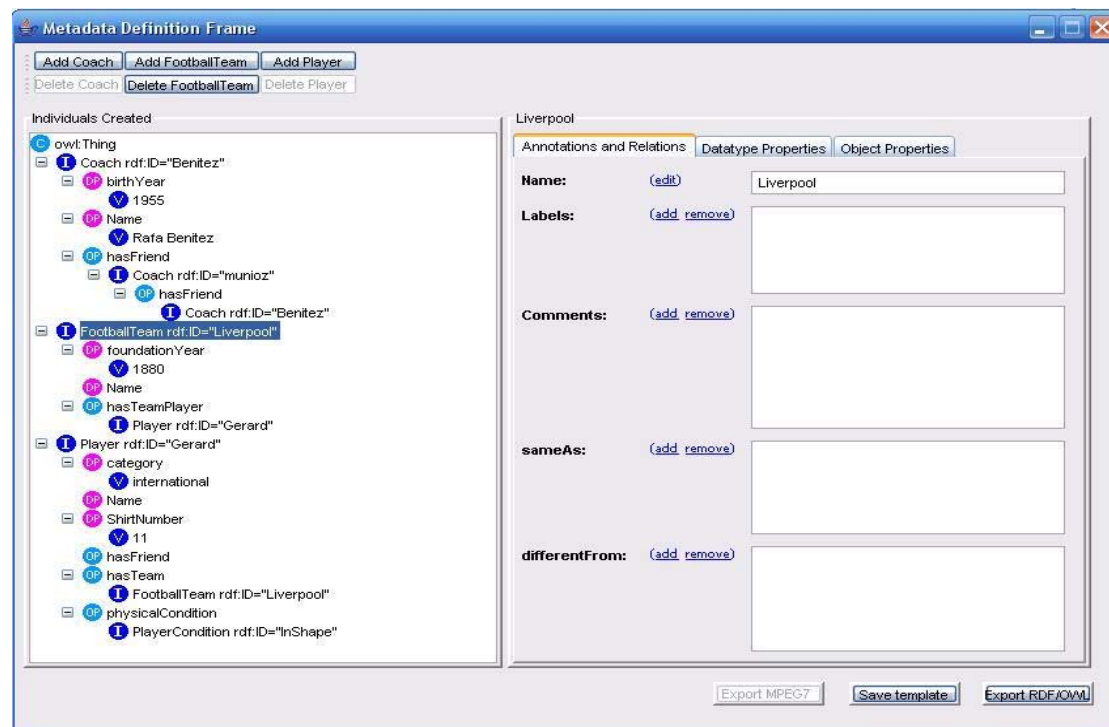
Με βάση όλα τα παραπάνω και ύστερα από κάποιες προσθήκες τιμών στις επιλεγμένες ιδιότητες για τα πρότυπα άτομα, το δένδρο ατόμων μπορεί να εξαπλωθεί σε μεγάλο βάθος και η εμφάνιση του θα είναι της παρακάτω μορφής (όπως φαίνεται και στην εικόνα που ακολουθεί):

- Κομβος ατόμου
 - Κόμβος ιδιότητας τύπου δεδομένων
 - Κόμβος τιμής ιδιότητας τύπου δεδομένων
 - Κόμβος τιμής ιδιότητας τύπου δεδομένων
 - ...
 - Κόμβος ιδιότητας τύπου δεδομένων
 - Κόμβος τιμής ιδιότητας τύπου δεδομένων
 - Κόμβος τιμής ιδιότητας τύπου δεδομένων
 - ...
 - ...
 - Κόμβος ιδιότητας αντικειμένων
 - Κόμβος τιμής (ατόμου) ιδιότητας αντικειμένων
 - Αντίστοιχο υποδένδρο για την τιμή (άτομο)
 - Κόμβος τιμής (ατόμου) ιδιότητας αντικειμένων
 - Αντίστοιχο υποδένδρο για την τιμή (άτομο)
 - ...
 - Κόμβος ιδιότητας αντικειμένων
 - Κόμβος τιμής (ατόμου) ιδιότητας αντικειμένων
 - Αντίστοιχο υποδένδρο για την τιμή (άτομο)

- Κόμβος τιμής (ατόμου) ιδιότητας αντικειμένων
- Αντίστοιχο υποδένδρο για την τιμή (άτομο)

-...

-...



Εικόνα 74: Η εξαπλώση του δένδρου ατόμων του παράθυρου μεταδεδομένων ύστερα από την προσθήκη τιμών στα πρότυπα άτομα.

Πέρα λοιπόν από την επιλογή ενός κόμβου από το δένδρο που αντιστοιχεί σε άτομο που περιγράφηκε παραπάνω, μπορεί να υπάρξει και επιλογή άλλου τύπου κόμβου. Πριν την περιγραφή των κόμβων άλλου τύπου, αξίζει να σημειωθεί ότι σε περίπτωση επιλογής ενός κόμβου που αντιστοιχεί σε άτομο και δεν έχει ανεπτυγμένο υποδένδρο, το σύστημα αυτόματα επιλέγει άλλον κόμβο του ατόμου που έχει ανεπτυγμένο υποδένδρο ώστε να εμφανιστεί η πλήρης πληροφορία του ατόμου. Επίσης, πρέπει να αναφερθεί ότι κάθε φορά τα κουμπιά διαγραφής ατόμων είναι απενεργοποιημένα πλην του κουμπιού που αντιστοιχεί στην κλάση του επιλεγμένου ατόμου.

Με την επιλογή ενός κόμβου από το δένδρο που αντιστοιχεί σε ιδιότητα τύπου δεδομένων ή σε τιμή ιδιότητας τύπου δεδομένων επιτρέπεται η εμφάνιση μόνο της όψης ιδιοτήτων τύπου δεδομένων του ατόμου στο οποίο ανήκει η ιδιότητα ή η τιμή της ιδιότητας και συνεπώς η επεξεργασία μόνο των ιδιοτήτων τύπου δεδομένων του ατόμου. Αντίστοιχα, με την επιλογή ενός κόμβου από το δένδρο που αντιστοιχεί σε ιδιότητα αντικειμένων επιτρέπεται η εμφάνιση μόνο της όψης ιδιοτήτων αντικειμένων του ατόμου στο οποίο ανήκει η ιδιότητα και συνεπώς η επεξεργασία μόνο των ιδιοτήτων αντικειμένων του ατόμου.

Συνεπώς με βάση τις λειτουργίες του, το παράθυρο μεταδεδομένων δίνει τη δυνατότητα δημιουργίας ολοκληρωμένης και εξατομικευμένης σημασιολογικής περιγραφής με

τη δημιουργία πολυάριθμων ατόμων επιλεγμένων κλάσεων τα οποία συνδέονται μεταξύ τους μέσω ιδιοτήτων αντικειμένων. Πρέπει να τονιστεί στο σημείο αυτό ότι στη δημιουργία των ατόμων τηρούνται σε επίπεδο ατόμων όλοι οι σημασιολογικοί έλεγχοι και οι αυτόματες ενημερώσεις που διενεργούνται και στις οντολογίες και έτσι τα άτομα αυτά καθίστανται συνεπή ως προς τον τύπο τους.

7.2.8. Αποθήκευση σημασιολογικών περιγραφών

Όλα τα άτομα και οι τιμές τους στις ιδιότητες μπορούν να αποθηκεύθουν σε RDF/OWL μορφή σε ξεχωριστό από τις οντολογίες αρχείο το οποίο ουσιαστικά αποτελεί μια νέα οντολογία που περιέχει μόνο άτομα. Η διαδικασία αρχίζει μόνο και εφόσον έχουν εισαχθεί στις ιδιότητες των ατόμων τιμές των οποίων ο αριθμός ικανοποιεί τους τυχόν περιορισμούς πολλαπλότητας ελάχιστου ή ακριβούς πλήθους. Με τη χρήση κατάλληλου εσωτερικού μηχανισμού, όλη η πληροφορία που βρίσκεται στο δένδρο ατόμων μετατρέπεται σε OWL μορφή και σχηματίζονται τα πλήρη URIs των ατόμων της τρέχουσας σημασιολογικής περιγραφής με βάση το όνομα του αρχείου το οποίο θα επιλεγεί για την αποθήκευση της περιγραφής.

7.2.9. Δημιουργία και αποθήκευση αρχείων προτύπων

Πέρα από την αποθήκευση της δημιουργηθείσας σημασιολογικής περιγραφής, παρέχεται η δυνατότητα αποθήκευσης της τρέχουσας επιλογής κλάσεων και ιδιοτήτων ώστε να μπορεί να επαναχρησιμοποιηθεί σε επόμενες σημασιολογικές περιγραφές χωρίς να απαιτηθεί ξανά η επιλογή κλάσεων και ιδιοτήτων μέσω των δύο βημάτων του οδηγού δημιουργίας παράθυρου μεταδεδομένων. Η τρέχουσα επιλογή κλάσεων και ιδιοτήτων μετατρέπεται μέσω κατάλληλου μηχανισμού σε OWL μορφή και αποθηκεύεται σε αρχεία προτύπων (templates). Η επιλογή της OWL ως μορφής των αρχείων προτύπων και όχι κάποιας άλλης μορφής, έγινε καθαρά για λόγους διαλειτουργικότητας (interoperability) της εφαρμογής αφού ένα αρχείο προτύπων μπορεί να δημιουργηθεί και εκτός της εφαρμογής του GraphOnto (για παράδειγμα μέσω ενός απλού συντάκτη κειμένου ή ενός OWL συντάκτη από ένα χρήστη με εμπειρία στη γραφή οντολογιών) και να χρησιμοποιηθεί χωρίς κανένα πρόβλημα, αρκεί να είναι σωστά δομημένο, από την εφαρμογή.

Κάθε αποθηκευμένο αρχείο προτύπων μπορεί να ανοιχθεί από το χρήστη μέσω του μενού Metadata, της επιλογής Load Metadata Template και της αναζήτησής του στο τοπικό σύστημα αρχείων. Παράλληλα όλα τα αρχεία προτύπων που ανοίχθηκαν ή δημιουργήθηκαν κατά την τρέχουσα χρήση της εφαρμογής είναι άμεσα προσπελάσιμα από το χρήστη (χωρίς αναζήτηση στο τοπικό σύστημα αρχείων) μέσω του μενού Metadata και της επιλογής Open Templates. Συμπερασματικά, αυτή η δυνατότητα επαναχρησιμοποίησης της επιλογής κλάσεων και ιδιοτήτων επιτρέπει την εξατομίκευση του γραφικού περιβάλλοντος του GraphOnto τόσο σε επίπεδο χρήστη, καθώς κάθε χρήστης εφαρμογής μπορεί να δημιουργήσει και να

αποθηκεύσει την επιλογή του για μελλοντικές χρήσεις, όσο και σε επίπεδο εφαρμογής, αφού τα αποθηκευμένα πρότυπα αρχεία μπορούν να χρησιμοποιηθούν από οποιοδήποτε χρήστη.

7.3. GraphOnto XML αρχεία προσαρμογής (configuration files)

7.3.1. Εισαγωγή

Το περιβάλλον εργασίας ενός χρήστη μπορεί να περιλαμβάνει αρκετές οντολογίες, ανώτερες και περιοχές, καθώς και κάποια αρχεία προτύπων για δημιουργία σημασιολογικής περιγραφής εκτός πλαισίου οντολογιών. Το γεγονός αυτό καθιστά χρονοβόρα τη διαδικασία διαδοχικού ανοίγματος όλων των οντολογιών και των αρχείων προτύπων καθώς απαιτείται από το χρήστη να επιλέγει κάθε μία οντολογία μέσα από το τοπικό σύστημα αρχείων ή να εισάγει τη διεύθυνση της στον Παγκόσμιο Ιστό. Ταυτόχρονα, το περιβάλλον εργασίας ενδέχεται να είναι σταθερό, δηλαδή ο χρήστης σε κάθε χρήση να χρησιμοποιεί το ίδιο σύνολο οντολογιών και αρχείων προτύπων. Το γεγονός αυτό καθιστά σε κάθε νέα χρήση της εφαρμογής μάλλον πληκτική τη διαδικασία διαδοχικού ανοίγματος όλων των οντολογιών και των προτύπων αρχείων.

Για λόγους λοιπόν απλοποίησης και μείωσης του απαιτούμενου φόρτου του χρήστη, αποφυγής χρήσης πολλών διαφορετικών επιλογών από το μενού της εφαρμογής (διαφορετικές επιλογές για άνοιγμα ανώτερης οντολογίας, για άνοιγμα οντολογίας περιοχής και για άνοιγμα αρχείου προτύπου), αλλά και για λόγους εξατομίκευσης των προτιμήσεων του, το GraphOnto παρέχει τη δυνατότητα χρήσης των αρχείων προσαρμογής (configuration files). Τα αρχεία προσαρμογής είναι XML δομημένα αρχεία που υπακούν σε ένα απλό XML Schema (περιγράφεται στο παράρτημα 2) και επιτρέπουν τον προσδιορισμό των χαρακτηριστικών του περιβάλλοντος εργασίας του χρήστη. Η επιλογή της XML ως μορφής των αρχείων προσαρμογής και όχι κάποιας άλλης μορφής, έγινε και αυτή καθαρά για λόγους διαλειτουργικότητας (interoperability) της εφαρμογής αφού ένα αρχείο προσαρμογής μπορεί να δημιουργηθεί πολύ εύκολα και εκτός της εφαρμογής του GraphOnto (για παράδειγμα μέσω ενός απλού συντάκτη κειμένου ή ενός XML συντάκτη από ένα χρήστη με ελάχιστη εμπειρία στην XML) και να χρησιμοποιηθεί χωρίς κανένα πρόβλημα, αρκεί να είναι σωστά δομημένο και έγκυρο με βάση το σχήμα, από την εφαρμογή.

7.3.2. Πληροφορία αρχείων προσαρμογής

Τα αρχεία προσαρμογής περιέχουν την παρακάτω πληροφορία:

- Το όνομα και το URI (τοπικό ή στο διαδίκτυο) των οντολογιών που θα ανοιχθούν, θα αναπαρασταθούν γραφικά και θα χρησιμοποιηθούν ως ανώτερες οντολογίες.

- Το όνομα και το URI (τοπικό ή στο διαδίκτυο) των οντολογιών που θα ανοιχθούν, θα αναπαρασταθούν γραφικά και θα χρησιμοποιηθούν ως οντολογίες περιοχής.
- Το όνομα και το URI (τοπικό ή στο διαδίκτυο) των αρχείων προτύπων για δημιουργία σημασιολογικής περιγραφής εκτός πλαισίου οντολογιών που θα ανοιχθούν και θα γίνουν διαθέσιμα στο χρήστη.
- Τις πιθανές εξαγόμενες μορφές της τρέχουσας χρήσης της εφαρμογής όπως παραδείγματος χάρη το MPEG-7. Η πληροφορία αυτή προς το παρόν συνδέεται άμεσα μόνο με τη λειτουργία του GraphOnto ως εργαλείο για τη σημασιολογική περιγραφή πολυμέσων και το μετασχηματισμό OWL περιγραφών σε MPEG-7 περιγραφών και θα παρουσιαστεί στην επόμενη ενότητα.

Ένα παράδειγμα ενός αρχείου προσαρμογής το οποίο περιέχει πληροφορία για δύο ανώτερες οντολογίες, έξι οντολογίες περιοχής και ένα αρχείο προτύπων που θα ανοιχθούν από την εφαρμογή φαίνεται στην παρακάτω εικόνα:

```
<?xml version="1.0" encoding="UTF-8"?>
<OntologyConfiguration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="C:\owlTool\GraphOntoConfig.xsd" name="">
  <CoreOntologies>
    <File filename="file:/C:/owltool/ontologies/AV_MDS03/av_semantics" name="AV Semantics"/>
    <File filename="file:/C:/owltool/ontologies/AV_MDS03/TypedRelationships" name="Middle
Ontology"/>
  </CoreOntologies>
  <DomainOntologies>
    <File filename="file:/C:/owltool/ontologies/AV_MDS03/soccer/socceragents" name="Soccer
Agents"/>
    <File filename="file:/C:/owltool/ontologies/AV_MDS03/soccer/soccertimes" name="Soccer Times"/>
    <File filename="file:/C:/owltool/ontologies/AV_MDS03/soccer/soccerobjects" name="Soccer
Objects"/>
    <File filename="file:/C:/owltool/ontologies/AV_MDS03/soccer/soccerplaces" name="Soccer
Places"/>
    <File filename="file:/C:/owltool/ontologies/AV_MDS03/soccer/soccerstates" name="Soccer States"/>
    <File filename="file:/C:/owltool/ontologies/AV_MDS03/soccer/soccerevents" name="Soccer
Events"/>
  </DomainOntologies>
  <Templates>
    <File filename="file:/C:/owltool/templates/game" name="Game"/>
  </Templates>
  <ExportedFormats/>
</OntologyConfiguration>
```

Εικόνα 75: Η XML μορφή ενός αρχείου προσαρμογής.

7.3.3. Πλεονεκτήματα χρήσης αρχείων προσαρμογής

Το βασικό πλεονέκτημα των αρχείων προσαρμογής είναι ότι επιτρέπουν στο χρήστη να ανοίξει ταυτόχρονα πολλαπλές οντολογίες και αρχεία προτύπων με τη χρήση ενός μόνου κλικ μέσω μιας απλής διαδικασίας αποκρύπτοντας του ουσιαστικά τις διάφορες επιλογές του

μενού της εφαρμογής που χρησιμοποιούνται για την εισαγωγή ανώτερων οντολογιών, οντολογιών περιοχής και αρχείων προτύπων.

Αναλυτικότερα, μέσω του μενού File, της επιλογής Open Configuration File και της αναζήτησης του κατάλληλου αρχείου προσαρμογής στο τοπικό σύστημα αρχείων, οι επιθυμητές από το χρήστη οντολογίες και αρχεία προτύπων ανοίγουν χωρίς να απαιτηθεί ξανά παρέμβαση του χρήστη. Για το διάβασμα του αρχείου προσαρμογής χρησιμοποιούνται ορισμένες Java κλάσεις οι οποίες προέκυψαν με βάση τα δομημένα στοιχεία του XML Schema που υπακούν τα αρχεία προσαρμογής.

Ταυτόχρονα, δίνεται η δυνατότητα στο χρήστη κάθε στιγμή να αποθηκεύσει σε ένα αρχείο προσαρμογής την τρέχουσα επιλογή ανοιχτών ανώτερων οντολογιών, οντολογιών περιοχής και αρχείων προτύπων. Το τελευταίο επιτυγχάνεται μέσω του μενού File, της επιλογής Save Ontology Selection to Configuration File και της εισαγωγής του ονόματος και της διεύθυνσης στο τοπικό σύστημα αρχείων του αρχείου προσαρμογής.

Με βάση τα παραπάνω συνεπώς, τα αρχεία προσαρμογής αποτελούν ένα μεγάλο σύμμαχο του χρήστη στην ελαχιστοποίηση του φόρτου εργασίας του και παράλληλα παρέχουν εξατομικευμένη υποστήριξη σε δύο επίπεδα:

- Σε επίπεδο χρήστη (user level), αφού μπορεί να υπάρξουν διαφορετικά αρχεία προσαρμογής για κάθε διαφορετικό χρήστη.
- Σε επίπεδο έργου (task level), αφού μπορεί να υπάρξουν διαφορετικά αρχεία προσαρμογής για τον ίδιο χρήστη για διαφορετικά έργα.

7.4. Ανακεφαλαίωση

Στο κεφάλαιο αυτό περιγράφηκαν δύο βασικά χαρακτηριστικά του GraphOnto που δεν υπάρχουν στα υπόλοιπα συστήματα διαχείρισης OWL οντολογιών και που έχουν ως στόχο την παροχή δυνατότητας εξατομίκευσης του περιβάλλοντος εργασίας (τόσο σε επίπεδο χρήστη όσο και σε επίπεδο έργου) και δυνατότητας απόκρυψης πληροφορίας. Αφενός η ανεξάρτητη σημασιολογική περιγραφή εκτός πλαισίου οντολογίας επιτρέπει στον αρχάριο χρήστη να δημιουργήσει με εύκολο τρόπο άτομα κλάσεων με βάση τις εξατομικευμένες ανάγκες του. Αφετέρου, τα εξατομικευμένα GraphOnto XML αρχεία προσαρμογής επιτρέπουν στο χρήστη να πραγματοποιήσει μια μακριά σειρά από ενέργειες με τη χρήση μιας μόνο επιλογής.

Κεφάλαιο 8^ο

8. Υποστήριξη σημασιολογικής περιγραφής περιεχομένου πολυμέσων

8.1. Εισαγωγή

Παράλληλα με την πλήρη λειτουργικότητα του ως αμιγές σύστημα διαχείρισης OWL οντολογιών, το GraphOnto χρησιμοποιείται και ως ένα εργαλείο σημασιολογικής περιγραφής πολυμέσων (βίντεο και εικόνες) μέσα στο DS-MIRF πλαίσιο. Η δυνατότητα του GraphOnto να διακρίνει τις οντολογίες σε ανώτερες και περιοχής βρίσκει εφαρμογή μέσα στην οντολογική υποδομή του DS-MIRF πλαισίου η οποία ως γνωστόν περιέχει:

- Μια ανώτερη οντολογία που αποτυπώνει πλήρως το MPEG-7 MDS πρότυπο.
- Μια οντολογία εφαρμογής που αντιμετωπίζεται και αυτή ως ανώτερη από το σύστημα και που αποτυπώνει πλήρως τις Σχέσεις Τύπων του MPEG-7 (Typed Relations of MPEG-7).
- Σημαντικό αριθμό από οντολογίες περιοχής που καλύπτουν τα πεδία του ποδοσφαίρου και της F1.

Όλες αυτές οι οντολογίες μπορούν να ανοιχθούν γρήγορα και εύκολα με ένα μόνο κλικ με τη χρήση ενός αρχείου προσαρμογής το οποίο θα περιλαμβάνει τις παραπάνω οντολογίες και το οποίο θα καθορίζει το περιβάλλον εργασίας με βάση τις προδιαγραφές του DS-MIRF πλαισίου. Επιπλέον, το GraphOnto μπορεί να χρησιμοποιηθεί για την περαιτέρω επέκταση της οντολογικής υποδομής του DS-MIRF πλαισίου με τη δημιουργία νέων οντολογιών περιοχής και προσθήκη τους στο περιβάλλον εργασίας του DS-MIRF.

Με βάση τις δύο ανώτερες οντολογίες και με τη χρήση του Μετατροπέα OWL σε MPEG-7, το GraphOnto έχει τη δυνατότητα να μετασχηματίζει:

- Τις OWL οντολογίες περιοχής σε ιεραρχίες MPEG-7 αφηρημένων σημασιολογικών οντοτήτων.
- Τις βασισμένες σε OWL σημασιολογικές περιγραφές σε σημασιολογικές περιγραφές βασισμένες σε MPEG-7.

Οι μετασχηματισμοί αυτοί πραγματοποιούνται με βάση τους κανόνες μετασχηματισμού (περιγράφονται αναλυτικά στην ενότητα 8.2) που αναπτύχθηκαν στα πλαίσια του DS-MIRF και με τη βοήθεια δύο OWL οντολογιών κανόνων μετασχηματισμού (περιγράφονται αναλυτικά στην ενότητα 8.3) που υλοποιήθηκαν. Το τελικό αποτέλεσμα είναι η εξαγωγή σημασιολογικών περιγραφών σε XML μορφή που υπακούν στο σχήμα του MPEG-7 MDS και η αποθήκευσή τους (περιγράφεται αναλυτικά στην ενότητα 8.4) σε XML έγγραφα ή στην αποθήκη μεταδεδομένων του DS-MIRF. Τέλος, παρέχεται η δυνατότητα

επαναχρησιμοποίησης αποθηκευμένων MPEG-7 σημασιολογικών περιγραφών (περιγράφεται αναλυτικά στην ενότητα 8.5) κατά τη διάρκεια δημιουργίας νέων σημασιολογικών περιγραφών.

8.2. Κανόνες μετασχηματισμού από OWL σε MPEG-7

8.2.1. Εισαγωγή

Όπως αναφέρθηκε και στην προηγούμενη ενότητα, οι μετασχηματισμοί από OWL σε MPEG-7 είναι δύο ειδών και συνεπώς δύο ειδών είναι και οι κανόνες μετασχηματισμού οι οποίοι παρουσιάζονται αμέσως μετά:

- Κανόνες Μετασχηματισμού OWL Οντολογιών Περιοχής: Οι κανόνες μετασχηματισμού OWL οντολογιών περιοχής σε MPEG-7 έγγραφα παράγουν MPEG-7 έγγραφα, όπου οι οντολογίες περιοχής αναπαρίστανται ως συλλογές αφηρημένων MPEG-7 οντοτήτων. Οι κανόνες μετασχηματισμού OWL οντολογιών περιοχής σε MPEG-7 έγγραφα είναι ένα σύνολο από κανόνες για κάθε OWL δομή (οντολογίες, κλάσεις, ιδιότητες τύπου δεδομένων, ιδιότητες αντικειμένων, άτομα) που περιγράφονται αναλυτικά στις επόμενες παραγράφους (8.2.2, 8.2.3, 8.2.4, 8.2.5, 8.2.6).
- Κανόνες Μετασχηματισμού OWL/RDF Μεταδεδομένων.

8.2.2. Κανόνες Μετασχηματισμού OWL Οντολογιών

Μια OWL οντολογία μετατρέπεται στο παραγόμενο XML έγγραφο σε ένα στοιχείο (element) περιγραφής ("Description"), τύπου (xsi:type) "SemanticDescriptionType". Το στοιχείο αυτό περιέχει ένα "Semantics" στοιχείο που έχει ως ταυτότητα (id) το όνομα της οντότητας (!ENTITY) που αντιστοιχεί στην οντολογία. Μέσα στο "Semantics" στοιχείο της περιγραφής που αναπαριστά την OWL οντολογία περιέχονται όλα τα συστατικά της οντολογίας (κλάσεις, ιδιότητες και άτομα). Επιπλέον, ισχύουν τα παρακάτω:

1. Για να δηλωθεί ότι η συγκεκριμένη σημασιολογική περιγραφή αποτελείται μόνο από αφηρημένες σημασιολογικές οντότητες, το παραγόμενο "Semantics" στοιχείο περιέχει ένα στοιχείο "AbstractionLevel" με διάσταση (dimension) που έχει την τιμή 1.
2. Αν υπάρχει ετικέτα (rdfs:label) της οντολογίας, αυτή μετασχηματίζεται σε ένα "Label" στοιχείο που έχει το περιεχόμενο της ετικέτας στο "Name" στοιχείο του.
3. Αν υπάρχει σχόλιο (rdfs:comment) της οντολογίας, αυτό μετασχηματίζεται σε ένα "Definition" στοιχείο, που έχει το περιεχόμενο του σχολίου στο "FreeTextAnnotation" στοιχείο του.
4. Για κάθε οντολογία που εισάγεται (owl:imports) στην τρέχουσα οντολογία, δημιουργείται ένα "Property" στοιχείο, που:

- ο Έχει στο "Name" στοιχείο του την τιμή "Ontology".
 - ο Αν η ταυτότητα (id) της οντολογίας είναι γνωστή, την έχει ως τιμή στο "Definition" στοιχείο του.
 - ο Έχει ένα "Term" στοιχείο, που έχει στο "Name" στοιχείο του την τιμή "href" και στο "Definition" στοιχείο του το URI της οντολογίας.
5. Για την τρέχουσα οντολογία δημιουργείται ένα "Property" στοιχείο, που:
- ο Έχει στο "Name" στοιχείο του την τιμή "Ontology Self".
 - ο Έχει ως τιμή στο "Definition" στοιχείο του την ταυτότητα (id) της τρέχουσας οντολογίας.
 - ο Έχει ένα "Term" στοιχείο, που έχει στο "Name" στοιχείο του την τιμή "href" και στο "Definition" στοιχείο του το URI της τρέχουσας οντολογίας.

Ένα παράδειγμα υλοποίησης των παραπάνω μετασχηματισμών φαίνονται στις δύο επόμενες εικόνες:

```
<!ENTITY soccer "http://127.0.0.1:8080/ontologies/AV_MDS03/soccer/soccer#">
<owl:Ontology rdf:about="">
  <rdfs:label>Soccer Ontology</rdfs:label>
  <rdfs:comment>OWL Ontology for Soccer</rdfs:comment>
</owl:Ontology>
...
```

Εικόνα 76: Παράδειγμα OWL οντολογίας ποδοσφαίρου.

```
<Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:mpeg:mpeg7:schema:2001
C:\dbxml\mds-visual-2003sa.xsd">
  <Description xsi:type="SemanticDescriptionType">
    <Semantics id="soccer">
      <AbstractionLevel dimension="1"/>
      <Label>
        <Name>Soccer Ontology</Name>
      </Label>
      <Definition>
        <FreeTextAnnotation>OWL Ontology for Soccer</FreeTextAnnotation>
      </Definition>
      <Property>
        <Name>Ontology Self</Name>
        <Definition>socceragents</Definition>
        <Term>
          <Name>href</Name>
          <Definition>
            "http://127.0.0.1:8080/ontologies/AV_MDS03/soccer/socceragents#"
          </Definition>
        </Term>
      </Property>
      ...
    </Semantics>
  </Description>
```


</Mpeg7>

Εικόνα 77: MPEG-7 XML Έγγραφο που παράγεται από την Οντολογία της προηγούμενης εικόνας.

8.2.3. Κανόνες Μετασχηματισμού OWL Κλάσεων

Οι κλάσεις της κάθε οντολογίας περιοχής που είναι υποκλάσεις (άμεσες ή έμμεσες) της κλάσης "SemanticBaseType" της ανώτερης οντολογίας είναι αυτές που παίρνουν μέρος στη διαδικασία μετασχηματισμού και μετατρέπονται σε στοιχεία "SemanticBase" σύμφωνα με τα παρακάτω:

1. Ο τύπος (xsi:type) κάθε παραγόμενου στοιχείου είναι το όνομα του πιο κοντινού προγόνου του στην ανώτερη οντολογία. Το όνομα αυτό καθορίζεται στην ετικέτα του προγόνου.
2. Τα παραγόμενα στοιχεία είναι αφηρημένα και αυτό επιτυγχάνεται με την εισαγωγή σε αυτά του στοιχείου "AbstractionLevel" που έχει στο χαρακτηριστικό (attribute) "Dimension" την τιμή 1.
3. Αν υπάρχει ετικέτα της κλάσης, αυτή μετασχηματίζεται σε ένα "Label" στοιχείο που έχει το περιεχόμενο της ετικέτας στο "Name" στοιχείο του.
4. Αν υπάρχει σχόλιο (rdfs:comment) της κλάσης, αυτό μετασχηματίζεται σε ένα "Definition" στοιχείο, που έχει το περιεχόμενο του σχολίου στο "FreeTextAnnotation" στοιχείο του.
5. Αν ο άμεσος πρόγονος της τρέχουσας κλάσης είναι κάποια κλάση που ανήκει σε οντολογία περιοχής (την τρέχουσα ή κάποια άλλη), για την αναπαράσταση της ιεραρχίας κλάσεων δημιουργούνται, μέσα στο τρέχον "SemanticBase" στοιχείο, δύο στοιχεία "Relation", που αναπαριστούν της σχέσεις γενίκευσης και εξειδίκευσης:
 - a. Ένα στοιχείο "Relation" τύπου (type) "specializes", με εκκίνηση (source) το στοιχείο που αναπαριστά την τρέχουσα κλάση και στόχο (target) το στοιχείο που αναπαριστά την άμεση υπερκλάση της.
 - b. Ένα στοιχείο "Relation" τύπου "generalizes", με εκκίνηση το στοιχείο που αναπαριστά την άμεση υπερκλάση της τρέχουσας κλάσης και στόχο το στοιχείο που αναπαριστά την τρέχουσα κλάση.
 - c. Αν η τρέχουσα κλάση διαθέτει περισσότερους από ένα άμεσους προγόνους προερχόμενους από οντολογία περιοχής, ένα ζευγάρι σχέσεων τύπου specializes/generalizes δημιουργείται για καθένα από της άμεσους προγόνους της.

Ένα παράδειγμα υλοποίησης των παραπάνω μετασχηματισμών φαίνονται στις δύο επόμενες εικόνες:

```
<owl:Class rdf:ID="GameAction">
  <rdfs:subClassOf rdf:resource="&BaseURL;&MDS;EventType"/>
```

```
<rdfs:label>Game Action</rdfs:label>
<rdfs:comment>Class Representing Soccer Game Actions</rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="PlayerAction">
  <rdfs:subClassOf rdf:resource="#GameAction"/>
  <rdfs:label>Player Action</rdfs:label>
</owl:Class>
```

Εικόνα 78: OWL κλάσεις που αναπαριστούν αγωνιστικές δραστηριότητες στο ποδόσφαιρο (GameAction) και δραστηριότητες παικτών (PlayerAction).

```
<SemanticBase xsi:type="EventType" id="GameAction">
  <AbstractionLevel dimension="1"/>
  <Label>
    <Name>Game Action</Name>
  </Label>
  <Definition>
    <FreeTextAnnotation>
      Class Representing Soccer Game Actions
    </FreeTextAnnotation>
  </Definition>
</SemanticBase>
<SemanticBase xsi:type="EventType" id="PlayerAction">
  <AbstractionLevel dimension="1"/>
  <Label>
    <Name>Player Action</Name>
  </Label>
  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2003:specializes"
    source="#PlayerAction" target="#GameAction"/>
  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2003:generalizes"
    source="#GameAction" target="#PlayerAction"/>
</SemanticBase>
```

Εικόνα 79: Τμήμα του MPEG-7 XML Εγγράφου που παράγεται από της OWL κλάσεις που φαίνονται στην προηγούμενη εικόνα.

8.2.4. Κανόνες Μετασχηματισμού OWL Ιδιοτήτων τύπου δεδομένων

Οι ιδιότητες τύπου δεδομένων των OWL οντολογιών που έχουν προκαθορισμένη τιμή (που καθορίζεται μέσω του περιορισμού owl:hasValue) σε μια συγκεκριμένη κλάση είναι αυτές που παίρνουν μέρος στη διαδικασία μετασχηματισμού και συνεπώς οι μετασχηματισμένες μορφές των ιδιοτήτων τύπου δεδομένων περικλείονται μέσα στο στοιχείο "SemanticBase". Οι ιδιότητες τύπου δεδομένων διακρίνονται σε δύο κατηγορίες όσον αφορά το μετασχηματισμό τους:

- Η ιδιότητα κληρονομείται (άμεσα ή έμμεσα) από της προγόνους της τρέχουσας κλάσης στην ανώτερη οντολογία ή αποτελεί υποϊδιότητα (owl:subPropertyOf) κάποιας ιδιότητας που κληρονομείται (άμεσα ή έμμεσα) από της προγόνους της τρέχουσας κλάσης στην ανώτερη οντολογία.

- Αν η ιδιότητα αναπαριστά κάποιο MPEG-7 χαρακτηριστικό (η πληροφορία αυτή περιέχεται στις οντολογίες κανόνων μετασχηματισμού που θα περιγραφούν στην επόμενη παράγραφο), τότε δημιουργείται το αντίστοιχο της χαρακτηριστικό με το όνομα που περιέχεται στις οντολογίες κανόνων μετασχηματισμού για τη συγκεκριμένη ιδιότητα και τιμή την τιμή του περιορισμού. Για παράδειγμα, η ιδιότητα "lang" που φαίνεται στην Εικόνα 80 μετασχηματίζεται στο χαρακτηριστικό "xml:lang" στην Εικόνα 81.
- Αν η ιδιότητα αναπαριστά κάποιο MPEG-7 στοιχείο (η πληροφορία αυτή περιέχεται στις οντολογίες κανόνων μετασχηματισμού), τότε δημιουργείται το αντίστοιχο της στοιχείο με το όνομα που περιέχεται στις οντολογίες κανόνων μετασχηματισμού για τη συγκεκριμένη ιδιότητα και περιεχόμενο την τιμή του περιορισμού. Για παράδειγμα, η ιδιότητα "Email" που φαίνεται στην Εικόνα 80 μετασχηματίζεται στο στοιχείο "Email" στην Εικόνα 81. (το οποίο περιέχεται μέσα στο "SemanticBase" στοιχείο που αναπαριστά την κλάση "AEKPlayer").
- Αν η ιδιότητα δεν αναπαριστά κάποιο MPEG-7 στοιχείο ή χαρακτηριστικό (η πληροφορία αυτή περιέχεται στις οντολογίες κανόνων μετασχηματισμού), αλλά το περιεχόμενο του τρέχοντος στοιχείου τότε η τιμή του περιορισμού της γίνεται περιεχόμενο του στοιχείου.

```
<owl:Class rdf:ID="AEKPlayer">
  <rdfs:subClassOf rdf:resource="&BaseURL;&MDS;AgentObjectType"/>
  ...
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#lang"/>
      <owl:hasValue>GR</owl:hasValue>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#Email"/>
      <owl:hasValue>players@aeafc.gr</owl:hasValue>
    </owl:Restriction>
  </rdfs:subClassOf>
  ...
</owl:Class>
```

Εικόνα 80: Ορισμός προκαθορισμένης τιμής σε OWL για τις ιδιότητες τύπου δεδομένων "lang" και "Email" της κλάσης "AEKPlayer" οι οποίες κληρονομούνται από πρόγονους της κλάσης στην ανώτερη οντολογία

```
<SemanticBase xsi:type="AgentObjectType" id="AEKPlayer" xml:lang="GR">
  <AbstractionLevel dimension="1"/>
  ...
  <Email>players@aeafc.gr</Email>
  ...
</SemanticBase>
```

</SemanticBase>

Εικόνα 81: Τμήμα του MPEG-7 XML Εγγράφου που παράγεται από την OWL κλάση και τις προκαθορισμένες τιμές της σε ιδιότητες τύπου δεδομένων που φαίνονται στην προηγούμενη εικόνα.

- b) Η ιδιότητα ορίζεται στην τρέχουσα κλάση ή σε κάποιο πρόγονό της σε οντολογία περιοχής και δεν αποτελεί υποϊδιότητα κάποιας ιδιότητας που κληρονομείται (άμεσα ή έμμεσα) από της προγόνους της τρέχουσας κλάσης στην ανώτερη οντολογία. Τότε δημιουργείται ένα "Property" στοιχείο, που έχει το όνομα της ιδιότητας (όπως αυτό καθορίζεται στο rdf:ID της) ως τιμή στο "Name" στοιχείο του και την τιμή του στο "Definition" στοιχείο του.

```
<owl:Class rdf:ID="Goalkeeper">
  <rdfs:subClassOf rdf:resource="&BaseURL;&MDS;AgentObjectType"/>
  ...
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#ShirtNumber"/>
      <owl:hasValue>1</owl:hasValue>
    </owl:Restriction>
  </rdfs:subClassOf>
  ...
</owl:Class>
```

Εικόνα 82: Ορισμός προκαθορισμένης τιμής σε OWL για την ιδιότητα τύπου δεδομένων "ShirtNumber" για την κλάση "Goalkeeper".

```
<SemanticBase xsi:type="AgentObjectType" id="AEKPlayer" xml:lang="GR">
  <AbstractionLevel dimension="1"/>
  ...
  <Property>
    <Name>ShirtNumber</Name>
    <Definition>1</Definition>
  </Property>
  ...
</SemanticBase>
```

Εικόνα 83: Τμήμα του MPEG-7 XML Εγγράφου που παράγεται από την OWL κλάση και την προκαθορισμένη τιμή της στην ιδιότητα τύπου δεδομένων που φαίνονται στην προηγούμενη εικόνα.

8.2.5. Κανόνες Μετασχηματισμού OWL Ιδιοτήτων αντικειμένων

Οι ιδιότητες αντικειμένων των OWL οντολογιών που έχουν προκαθορισμένη τιμή (που καθορίζεται μέσω του περιορισμού owl:hasValue) σε μια συγκεκριμένη κλάση είναι αυτές που παίρνουν μέρος στη διαδικασία μετασχηματισμού και συνεπώς οι μετασχηματισμένες μορφές των ιδιοτήτων αντικειμένων περιλαμβάνονται μέσα στο στοιχείο "SemanticBase". Μια OWL

ιδιότητα αντικειμένων, σε ότι αφορά τη σύνταξη, αποτελείται από ένα στοιχείο με το όνομα της ιδιότητας, όπως αυτό καθορίζεται στο `rdf:ID` της ιδιότητας, (για παράδειγμα, το στοιχείο `"AddressLine"` στην Εικόνα 84), το οποίο περιέχει ένα στοιχείο με το άτομο-τιμή της ιδιότητας (για παράδειγμα, το στοιχείο `"TextualType"` στην Εικόνα 84) ή αναφέρεται (με τη χρήση του `rdf:about`) σε αυτό. Το άτομο-τιμή της ιδιότητας διαθέτει ταυτότητα και περιέχει της τυχόν προκαθορισμένες τιμές των ιδιοτήτων του. Για της ιδιότητες αντικειμένων που έχουν προκαθορισμένη τιμή διακρίνονται οι εξής περιπτώσεις:

1. Η ιδιότητα κληρονομείται (άμεσα ή έμμεσα) από της προγόνους της τρέχουσας κλάσης στην ανώτερη οντολογία ή αποτελεί υποϊδιότητα κάποιας ιδιότητας που κληρονομείται (άμεσα ή έμμεσα) από της προγόνους της τρέχουσας κλάσης στην ανώτερη οντολογία. Στην περίπτωση αυτή δημιουργείται ένα στοιχείο που έχει το όνομα της ιδιότητας (ή της κοντινότερης υπερ-ιδιότητάς της στην ανώτερη οντολογία) όπως αυτό καθορίζεται στην ετικέτα της ιδιότητας στην ανώτερη οντολογία. Το στοιχείο αυτό:

- Αν έχει τύπο που διαθέτει ταυτότητα (`id`), έχει ως ταυτότητα την ταυτότητα (`rdf:ID`) του ατόμου-τιμής της ιδιότητας.
- Περιέχει της ιδιότητές του που έχουν καθορισμένη τιμή. Αν κάποιες από αυτές είναι ιδιότητες αντικειμένων που η τιμή τους αναπαρίσταται από κάποιο άτομο που ανήκει σε κάποια υποκλάση του τύπου της ιδιότητας, τότε ο τύπος (`xml:type`) του παραγόμενου στοιχείου είναι το όνομα του πιο κοντινού προγόνου της κλάσης όπου ανήκει το άτομο στην ανώτερη οντολογία.

Για παράδειγμα, έστω ότι στην κλάση `"AEKPlayer"`, που αναπαριστά της ποδοσφαιριστές της ΑΕΚ, η υποχρεωτική τιμή της ιδιότητας `"PostalAddress"` είναι η τιμή που φαίνεται στην εικόνα 84. Η ιδιότητα `"PostalAddress"` μαζί με την υποχρεωτική τιμή της μετασχηματίζεται στην MPEG-7 σύνταξη που φαίνεται στην εικόνα 85.

```
<PostalAddressType rdf:ID="AEK_PostalAddress">
  <lang>EN</lang>
  <AddressLine>
    <TextualType rdf:ID="AEK_AL">
      <contentString>
        Γράμμου 69-71, Μαρούσι
      </contentString>
    </TextualType>
  </AddressLine>
```

```
<PostingIdentifier>
```

```
  <TextualType rdf:ID="AEK_PI">
    <contentString>15124</contentString>
  </TextualType>
</PostingIdentifier>
</PostalAddressType>
```

Εικόνα 84: OWL Άτομο τύπου "PostalAddressType", που αναπαριστά τη διεύθυνση των ποδοσφαιριστών της ΑΕΚ και αποτελεί την υποχρεωτική τιμή της ιδιότητας "PostalAddress" για τα άτομα που ανήκουν στην κλάση "AEKPlayer"

```
<PostalAddress id="AEK_PostalAddress" xml:lang="EN">
  <AddressLine id="AEK_AL">
    Γράμμου 69-71, Μαρούσι
  </AddressLine>
  <PostingIdentifier id="AEK_PI">15124</PostingIdentifier>
</PostalAddress>
```

Εικόνα 85: Η τιμή της ιδιότητας "PostalAddress" για τα άτομα που ανήκουν στην κλάση "AEKPlayer" σε MPEG-7 σύνταξη

2. Η ιδιότητα ορίζεται στην τρέχουσα κλάση ή σε κάποιο πρόγονό της σε οντολογία περιοχής και δεν αποτελεί υποϊδιότητα κάποιας ιδιότητας που κληρονομείται (άμεσα ή έμμεσα) από της προγόνους της τρέχουσας κλάσης στην ανώτερη οντολογία. Στην περίπτωση αυτή παίρνουν μέρος στη διαδικασία μετασχηματισμού ιδιότητες αντικειμένων για της οποίες ισχύει κάτι από τα ακόλουθα:

- a) Η τρέχουσα κλάση είναι υποκλάση (άμεση ή έμμεση) της κλάσης "SemanticBaseType" και το πεδίο τιμών της ιδιότητας είναι η κλάση "SemanticBaseType" ή κάποια (άμεση ή έμμεση) υποκλάση της. Τότε δημιουργούνται δύο στοιχεία "Relation", που συνδέουν την κλάση με την ιδιότητα:
 - Ένα στοιχείο "Relation" τύπου (type) "property", με εκκίνηση το στοιχείο που αναπαριστά την τρέχουσα κλάση και στόχο το στοιχείο που αναπαριστά την τιμή της ιδιότητας.
 - Ένα στοιχείο "Relation" τύπου (type) "propertyOf", με εκκίνηση το στοιχείο που αναπαριστά την τιμή της ιδιότητας και στόχο το στοιχείο που αναπαριστά την τρέχουσα κλάση.

Για παράδειγμα, έστω ότι οι ομάδες που παίρνουν μέρος σε κάποιο τουρνουά που οργανώνει η εταιρεία "X" πρέπει να έχουν την εταιρεία αυτή ως χορηγό. Έστω τώρα ότι οι ομάδες ποδοσφαίρου εν γένει αναπαρίστανται από την κλάση "SoccerTeam", η ιδιότητα "Sponsor" συνδέει μια ομάδα με το χορηγό της και οι ομάδες που παίρνουν μέρος στο συγκεκριμένο τουρνουά αναπαρίστανται από την κλάση "TournamentSoccerTeam", οπότε η ιδιότητα "Sponsor" έχει γι' αυτές την προκαθορισμένη τιμή του ατόμου που αναπαριστά την εταιρεία "X". Η προκαθορισμένη αυτή τιμή για τα άτομα της "TournamentSoccerTeam" μετατρέπεται σε MPEG-7 σύνταξη στα στοιχεία "Relation" που φαίνονται στην εικόνα 86.

```
<Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2003:property"
source="#TournamentSoccerTeam" target="#X"/>
<Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2003:propertyOf" source="#X"
target="#TournamentSoccerTeam"/>
```

Εικόνα 86: MPEG-7 σύνταξη για τον ορισμό της προκαθορισμένης τιμής του ατόμου που αναπαριστά την εταιρεία "X" στην ιδιότητα "Sponsor" τιμή για τα άτομα της κλάσης "TournamentSoccerTeam"

- b) Η τρέχουσα κλάση είναι υποκλάση (άμεση ή έμμεση) της κλάσης "SemanticBaseType" και το πεδίο τιμών της ιδιότητας είναι "TermUseType" (ή κάποια άμεση ή έμμεση υποκλάση της). Τότε, δημιουργείται (μέσα στο στοιχείο "SemanticBase" που αναπαριστά την τρέχουσα κλάση) ένα "Property" στοιχείο, που έχει το όνομα της ιδιότητας, όπως αυτό καθορίζεται στο rdf:ID της ιδιότητας, ως τιμή της στο "Name" στοιχείο του και την τιμή της ιδιότητας σε ένα "Term" στοιχείο του.
- Αν το πεδίο τιμών της ιδιότητας είναι κάποια (άμεση ή έμμεση) υποκλάση της "TermUseType" που ορίζεται στην ανώτερη οντολογία, ο τύπος (xsi:type) του στοιχείου "Term" έχει ως τιμή το όνομα του τύπου δεδομένων του.
 - Αν το πεδίο τιμών της ιδιότητας ορίζεται σε οντολογία περιοχής, ο τύπος του στοιχείου "Term" έχει ως τιμή το όνομα του κοντινότερου προγόνου του τύπου δεδομένων του στην ανώτερη οντολογία.

Για παράδειγμα, έστω ότι κάθε ποδοσφαιριστής έχει ρόλο, ο οποίος αναπαρίσταται από την ιδιότητα αντικειμένων "role", που έχει πεδίο τιμών "TermUseType". Έστω τώρα ότι οι μέσοι παίκτες αναπαρίστανται από την κλάση "Midfielder", που έχει υποχρεωτικά την τιμή "Midfielder", που φαίνεται στην εικόνα 87, για την ιδιότητα "role". Η προκαθορισμένη αυτή τιμή για τα άτομα της "Midfielder" μετατρέπεται σε MPEG-7 σύνταξη στο στοιχείο "Property" που φαίνεται στην εικόνα 88 (το οποίο περιέχεται μέσα στο "SemanticBase" στοιχείο που αναπαριστά την κλάση "Midfielder").

```
<TermUseType rdf:ID="MidfielderTerm">
  <TermName>
    <TermNameType rdf:ID="MidfielderNT">
      <lang>en</lang>
      <contentString>Midfielder</contentString>
    </TermNameType>
  </TermName>
  <Definition>
    <TextualType rdf:ID="MidfielderDef">
      <lang>en</lang>
      <contentString>
        Player who links together the offensive and defensive functions of a team, plays behind
        their forwards.
      </contentString>
    </TextualType>
  </Definition>
</TermUseType>
```

Εικόνα 87: OWL Άτομο τύπου "TermUseType", που αναπαριστά την υποχρεωτική τιμή της ιδιότητας "role" για τα άτομα που ανήκουν στην κλάση "Midfielder"

```
<Property>
  <Name>role</Name>
  <Term id="MidfielderTerm">
    <Name lang="en">Midfielder</Name>
    <Definition id="MidfielderDef" lang="en">
      Player who links together the offensive and defensive functions of a team, plays behind their
forwards.
    </Definition>
  </Term>
</Property>
```

Εικόνα 88: MPEG-7 σύνταξη για τον ορισμό της προκαθορισμένης τιμής της ιδιότητας "role" τιμή για τα άτομα της κλάσης "Midfielder"

8.2.6. Κανόνες Μετασχηματισμού OWL Ατόμων

Τα άτομα που ανήκουν στην κλάση "SemanticBaseType" και της υποκλάσεις της (άμεσες ή έμμεσες), στην κλάση "TermUseType" και της υποκλάσεις της (άμεσες ή έμμεσες) και στην κλάση "RelationType" και της υποκλάσεις της (άμεσες ή έμμεσες) είναι αυτά που παίρνουν μέρος στη διαδικασία μετασχηματισμού.

Άτομα που ανήκουν στην επέκταση κλάσης της κλάσης "SemanticBaseType"

Αν κάποιο άτομο ανήκει στην κλάση "SemanticBaseType" ή σε κάποια (άμεση ή έμμεση) υποκλάση της, τότε αυτό μετασχηματίζεται σε ένα "SemanticBase" στοιχείο του οποίου τα χαρακτηριστικά και τα στοιχεία διαμορφώνονται σύμφωνα με τα παρακάτω:

1. Ο τύπος (xsi:type) του παραγόμενου στοιχείου είναι το όνομα του πιο κοντινού προγόνου στην ανώτερη οντολογία της κλάσης στην οποία ανήκει το άτομο. Το όνομα αυτό καθορίζεται στην ετικέτα του προγόνου.
2. Το παραγόμενο στοιχείο δεν είναι αφηρημένο και αυτό επιτυγχάνεται με την εισαγωγή σε αυτά του στοιχείου "AbstractionLevel" που έχει στο χαρακτηριστικό "Dimension" την τιμή 0.
3. Αν υπάρχει ετικέτα του ατόμου, αυτή μετασχηματίζεται σε ένα "Label" στοιχείο που έχει το περιεχόμενο της ετικέτας στο "Name" στοιχείο του.
4. Αν υπάρχει σχόλιο του ατόμου, αυτό μετασχηματίζεται σε ένα "Definition" στοιχείο, που έχει το περιεχόμενο του σχολίου στο "FreeTextAnnotation" στοιχείο του.
5. Αν η κλάση στην οποία ανήκει το άτομο είναι κάποια κλάση που ανήκει σε οντολογία περιοχής (την τρέχουσα ή κάποια άλλη) δημιουργούνται μέσα στο τρέχον "SemanticBase" στοιχείο, δύο στοιχεία "Relation", που δηλώνουν ότι το παραγόμενο στοιχείο αποτελεί παράδειγμα του αφηρημένου στοιχείου που αντιστοιχεί στην κλάση του ατόμου:

- a. Ένα στοιχείο "Relation" τύπου "exemplifies", με εκκίνηση το στοιχείο που αναπαριστά το άτομο και στόχο το στοιχείο που αναπαριστά την κλάση του.
 - b. Ένα στοιχείο "Relation" τύπου "exemplifiedBy", με εκκίνηση το στοιχείο που αναπαριστά την κλάση του ατόμου και στόχο το στοιχείο που αναπαριστά το άτομο
 - c. Αν το άτομο ανήκει σε περισσότερες από μία υποκλάσεις της "SemanticBaseType" προερχόμενες από οντολογία περιοχής, ένα ζευγάρι σχέσεων exemplifies/exemplifiedBy δημιουργείται για κάθε τέτοια κλάση.
6. Οι κανόνες που εφαρμόζονται για τις προκαθορισμένης τιμής ιδιότητες τύπου δεδομένων των κλάσεων των οντολογιών περιοχής εφαρμόζονται και για τις τιμές των ιδιοτήτων τύπου δεδομένων των ατόμων που ανήκουν στην κλάση "SemanticBaseType" και της υποκλάσεις της. Πιο συγκεκριμένα διακρίνονται οι εξής περιπτώσεις:
 - a. Η ιδιότητα ορίζεται στην ανώτερη οντολογία ή αποτελεί υποϊδιότητα κάποιας ιδιότητας που ορίζεται στην ανώτερη οντολογία:
 - i. Αν η ιδιότητα αναπαριστά κάποιο MPEG-7 χαρακτηριστικό (η πληροφορία αυτή περιέχεται στις οντολογίες κανόνων μετασχηματισμού), τότε δημιουργείται το αντίστοιχο της στοιχείο με το όνομα που περιέχεται στις οντολογίες κανόνων μετασχηματισμού για τη συγκεκριμένη ιδιότητα και τιμή την τιμή της ιδιότητας.
 - ii. Αν η ιδιότητα αναπαριστά κάποιο MPEG-7 στοιχείο (η πληροφορία αυτή περιέχεται στις οντολογίες κανόνων μετασχηματισμού), τότε δημιουργείται το αντίστοιχο της στοιχείο με το όνομα που περιέχεται στις οντολογίες κανόνων μετασχηματισμού για τη συγκεκριμένη ιδιότητα και με την τιμή της ιδιότητας ως περιεχόμενο.
 - iii. Αν η ιδιότητα δεν αναπαριστά κάποιο MPEG-7 στοιχείο ή χαρακτηριστικό (η πληροφορία αυτή περιέχεται στις οντολογίες κανόνων μετασχηματισμού), αλλά το περιεχόμενο του τρέχοντος στοιχείου τότε η τιμή της γίνεται περιεχόμενο του στοιχείου.
 - b. Η ιδιότητα ορίζεται σε οντολογία περιοχής και δεν αποτελεί υποϊδιότητα κάποιας ιδιότητας που ορίζεται στην ανώτερη οντολογία. Τότε δημιουργείται ένα "Property" στοιχείο, που έχει το όνομα της ιδιότητας ως τιμή στο "Name" στοιχείο του και την τιμή της ιδιότητας στο "Definition" στοιχείο του.
7. Οι κανόνες που εφαρμόζονται για τις προκαθορισμένης τιμής ιδιότητες αντικειμένων των κλάσεων των οντολογιών περιοχής εφαρμόζονται και για τις τιμές των ιδιοτήτων αντικειμένων των ατόμων που ανήκουν στην κλάση "SemanticBaseType" και της υποκλάσεις της. Πιο συγκεκριμένα διακρίνονται οι εξής περιπτώσεις:
 - a. Η ιδιότητα ορίζεται στην ανώτερη οντολογία ή αποτελεί υποϊδιότητα κάποιας ιδιότητας που ορίζεται στην ανώτερη οντολογία. Στην περίπτωση αυτή

δημιουργείται ένα στοιχείο που έχει το όνομα της ιδιότητας (ή της κοντινότερης υπερ-ιδιότητάς της στην ανώτερη οντολογία). Το στοιχείο αυτό έχει ως ταυτότητα το `rdf:ID` του ατόμου-τιμής της ιδιότητας και περιέχει της ιδιότητες του στοιχείου αυτού που έχουν καθορισμένη τιμή.

- b. Η ιδιότητα ορίζεται σε οντολογία περιοχής και δεν αποτελεί υποιδιότητα κάποιας ιδιότητας που ορίζεται στην ανώτερη οντολογία. Στην περίπτωση αυτή παίρνουν μέρος στη διαδικασία μετασχηματισμού ιδιότητες αντικειμένων για της οποίες ισχύει κάτι από τα ακόλουθα:
 - i. Η κλάση του ατόμου είναι υποκλάση (άμεση ή έμμεση) της κλάσης `"SemanticBaseType"` και το πεδίο τιμών της ιδιότητας είναι η κλάση `"SemanticBaseType"` ή κάποια (άμεση ή έμμεση) υποκλάση της. Τότε δημιουργούνται, μέσα στο τρέχον `"SemanticBase"` στοιχείο, δύο στοιχεία `"Relation"`, που συνδέουν την κλάση με την ιδιότητα:
 - 1) Ένα στοιχείο `"Relation"` τύπου (type) `"property"`, με εκκίνηση το στοιχείο που αναπαριστά την τρέχουσα κλάση και στόχο το στοιχείο που αναπαριστά την τιμή της ιδιότητας.
 - 2) Ένα στοιχείο `"Relation"` τύπου (type) `"propertyOf"`, με εκκίνηση το στοιχείο που αναπαριστά την τιμή της ιδιότητας και στόχο το στοιχείο που αναπαριστά την τρέχουσα κλάση.
 - ii. Η κλάση του ατόμου είναι υποκλάση (άμεση ή έμμεση) της κλάσης `"SemanticBaseType"` και το πεδίο τιμών της ιδιότητας είναι `"TermUseType"` (ή κάποια άμεση ή έμμεση υποκλάση της). Τότε, δημιουργείται (μέσα στο στοιχείο `"SemanticBase"` που αναπαριστά το τρέχον άτομο) ένα `"Property"` στοιχείο, που έχει το όνομα της ιδιότητας ως τιμή στο `"Name"` στοιχείο του και την τιμή της ιδιότητας σε ένα `"Term"` στοιχείο του.
 - 1) Αν το πεδίο τιμών της ιδιότητας είναι κάποια (άμεση ή έμμεση) υποκλάση της `"TermUseType"` που ορίζεται στην ανώτερη οντολογία, ο τύπος (`xsi:type`) του στοιχείου `"Term"` έχει ως τιμή το όνομα του τύπου δεδομένων του.
 - 2) Αν το πεδίο τιμών της ιδιότητας ορίζεται σε οντολογία περιοχής, ο τύπος του στοιχείου `"Term"` έχει ως τιμή το όνομα του κοντινότερου προγόνου του τύπου δεδομένων του στην ανώτερη οντολογία.

Για παράδειγμα, έστω το άτομο `"Moras"`, που αναπαριστά τον ποδοσφαιριστή Βαγγέλη Μόρα και φαίνεται στην εικόνα 89. Το άτομο αυτό μετασχηματίζεται στο τμήμα XML εγγράφου που φαίνεται στην εικόνα 90. Να σημειωθεί ότι η ιδιότητα `Email` είναι ιδιότητα

τύπου δεδομένων που αντιστοιχεί σε στοιχείο, η ιδιότητα `FamilyName` είναι ιδιότητα αντικειμένων, η ιδιότητα `contentString` είναι ιδιότητα τύπου δεδομένων που αναπαριστά την τιμή του στοιχείου `FamilyName` και η ιδιότητα `initial` είναι ιδιότητα τύπου δεδομένων που αντιστοιχεί σε χαρακτηριστικό.

```
<PlayerObject rdf:ID="Moras">
  <rdfs:label>Μόρας</rdfs:label>
  <rdfs:comment>Βαγγέλης Μόρας</rdfs:comment>
  <Agent>
    <Player rdf:ID="MorasV">
      <ElectronicAddress>
        <ElectronicAddressType rdf:ID="MorasV_ElectronicAddress1">
          <Email>players@aekfc.gr</Email>
        </ElectronicAddressType>
      </ElectronicAddress>
      <PersonName>
        <PersonNameType rdf:ID="MorasV_PersonName1">
          <FamilyName>
            <NameComponentType
rdf:ID="MorasV_PersonName1_FamilyName1">
              <contentString>Μόρας</contentString>
              <initial>M</initial>
            </NameComponentType>
          </FamilyName>
          <GivenName>
            <NameComponentType rdf:ID="MorasV_PersonName1_GivenName1">
              <contentString>Βαγγέλης</contentString>
            </NameComponentType>
          </GivenName>
        </PersonNameType>
      </PersonName>
    </Player>
  </Agent>
</PlayerObject>
```

Εικόνα 89 : Άτομο που αναπαριστά τον ποδοσφαιριστή Βαγγέλη Μόρα.

```
<SemanticBase xsi:type="AgentObjectType" id="Moras">
  <AbstractionLevel dimension="0"/>
  <Label>
    <Name>Μόρας</Name>
  </Label>
  <Definition>
    <FreeTextAnnotation>
      Βαγγέλης Μόρας
```

```
</FreeTextAnnotation>
```

```
</Definition>
<Agent xsi:type="PersonType" id="MorasV">
  <ElectronicAddress id="MorasV_ElectronicAddress1">
    <Email>players@aekfc.gr</Email>
  </ElectronicAddress>
  <Name id="MorasV_PersonName1">
    <FamilyName id="MorasV_PersonName1_FamilyName1" initial="M">
```

```

        Μόρας
      </FamilyName>
      <GivenName id="MorasV_PersonName1_GivenName1">
        Βαγγέλης
      </GivenName>
    </Name>
  </Agent>
  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2003:exemplifies"
    source="#Moras" target="#PlayerObject"/>
  <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2003:exemplifiedBy"
    source="#PlayerObject" target="#Moras"/>
</SemanticBase>

```

Εικόνα 90: Τμήμα του MPEG-7 XML Εγγράφου που παράγεται από το άτομο που φαίνεται στην εικόνα 87.

Άτομα που ανήκουν στην επέκταση κλάσης της κλάσης "TermUseType"

Αν το άτομο ανήκει στην κλάση "TermUseType" ή σε κάποια (άμεση ή έμμεση) υποκλάση της, δημιουργείται (μέσα στο στοιχείο "Semantics" που αναπαριστά την οντολογία) ένα "Property" στοιχείο, που αναπαριστά το άτομο.

1. Αν το άτομο ανήκει σε κάποια (άμεση ή έμμεση) υποκλάση της "TermUseType" που ορίζεται στην ανώτερη οντολογία, ο τύπος του στοιχείου "Property" έχει ως τιμή το όνομα της κλάσης του.
2. Αν το άτομο ανήκει σε κάποια κλάση που ο κοντινότερος πρόγονός της στην ανώτερη οντολογία δεν είναι η κλάση "TermUseType", ο τύπος του στοιχείου "Property" έχει ως τιμή το όνομα του κοντινότερου προγόνου στην ανώτερη οντολογία της κλάσης του ατόμου.

Άτομα που ανήκουν στην επέκταση κλάσης της κλάσης "RelationType"

Αν το άτομο ανήκει στην κλάση "RelationType" ή σε κάποια (άμεση ή έμμεση) υποκλάση της, δημιουργείται ένα "Relation" στοιχείο μέσα στο στοιχείο "Semantics" που αναπαριστά την οντολογία, για το οποίο ισχύουν τα εξής:

1. Ως τιμή του χαρακτηριστικού "type" την τιμή της ιδιότητας "typeRT".
2. Ως τιμή του χαρακτηριστικού "source" την τιμή της ιδιότητας "source".
3. Ως τιμή του χαρακτηριστικού "target" την τιμή της ιδιότητας "target".
4. Ως τιμή του χαρακτηριστικού "directed" την τιμή της ιδιότητας "directed".
5. Ως τιμή του χαρακτηριστικού "strength" την τιμή της ιδιότητας "strength".
6. Αν διαθέτει το χαρακτηριστικό "inverse", δημιουργείται ένα "Relation" στοιχείο μέσα στο στοιχείο "Semantics" που αναπαριστά την οντολογία, για το οποίο ισχύουν τα εξής:

- a. Ως τιμή του χαρακτηριστικού "type" την τιμή της ιδιότητας "inverse" του ατόμου.
- b. Ως τιμή του χαρακτηριστικού "source" την τιμή της ιδιότητας "target" του ατόμου.
- c. Ως τιμή του χαρακτηριστικού "source" την τιμή της ιδιότητας "target" του ατόμου.
- d. Ως τιμή του χαρακτηριστικού "directed" την τιμή της ιδιότητας "directed" του ατόμου.
- e. Ως τιμή του χαρακτηριστικού "strength" την τιμή της ιδιότητας "strength" του ατόμου.

8.2.7. Κανόνες Μετασχηματισμού OWL/RDF Μεταδεδομένων

Τα OWL/RDF Μεταδεδομένα που παράγονται εκτός πλαισίου οντολογιών μπορούν και αυτά να μετασχηματιστούν σε MPEG-7 εφόσον τα μεταδεδομένα (OWL άτομα) αυτά βασίζονται σε οντολογίες περιοχής οι οποίες έχουν μετασχηματιστεί σε MPEG-7 περιγραφές σύμφωνα με τους κανόνες που αναφέρθηκαν στην προηγούμενη παράγραφο. Για κάθε OWL/RDF έγγραφο παράγεται ένα XML έγγραφο που περιέχει ένα στοιχείο περιγραφής ("Description"), τύπου (xsi:type) "SemanticDescriptionType". Το στοιχείο αυτό περιέχει ένα "Semantics" στοιχείο, μέσα στο οποίο περιέχονται όλα τα στοιχεία που αναπαριστούν τα OWL/RDF άτομα που αναπαριστούν τα μεταδεδομένα. Πιο συγκεκριμένα έχουμε:

1. Καθώς η συγκεκριμένη σημασιολογική περιγραφή αποτελείται από μη αφηρημένες σημασιολογικές οντότητες, το παραγόμενο "Semantics" στοιχείο περιέχει ένα στοιχείο "AbstractionLevel" με διάσταση που έχει την τιμή 0.
2. Για κάθε οντολογία που εισάγεται (owl:imports) στο OWL/RDF έγγραφο, δημιουργείται ένα "Property" στοιχείο, που:
 - a. Έχει στο "Name" στοιχείο του την τιμή "Ontology".
 - b. Αν η ταυτότητα της οντολογίας είναι γνωστή, την έχει ως τιμή στο "Definition" στοιχείο του.
 - c. Έχει ένα "Term" στοιχείο, που έχει στο "Name" στοιχείο του την τιμή "href" και στο "Definition" στοιχείο του το URI της οντολογίας.
3. Για το μετασχηματισμό των ατόμων του OWL/RDF εγγράφου χρησιμοποιούνται οι κανόνες που περιγράφηκαν στην προηγούμενη παράγραφο.

8.3. Χρήση οντολογιών κανόνων μετασχηματισμού

Οι παραπάνω κανόνες μετασχηματισμού παρέχουν ένα πολύ σημαντικό εφόδιο στο σύστημα για να μετατρέψει τις OWL οντολογίες και σημασιολογικές περιγραφές σε MPEG-7

σημασιολογικές περιγραφές πολυμέσων. Όμως από μόνοι τους δεν αρκούν για την παραγωγή έγκυρων MPEG-7 αρχείων λόγω των παρακάτω ιδιομορφιών:

- Κάθε ιδιότητα τύπου δεδομένων μπορεί να μετατραπεί είτε σε χαρακτηριστικό, είτε σε στοιχείο είτε σε περιεχόμενο. Συνεπώς χρειάζεται επιπρόσθετη πληροφορία για κάθε ιδιότητα τύπου δεδομένων προκειμένου να αποφασισθεί ο τύπος στον οποίο θα μετατραπεί.
- Σε μία OWL οντολογία κάθε ιδιότητα τύπου δεδομένων έχει μοναδικό όνομα, ενώ στο MPEG-7 μπορούν να οριστούν χαρακτηριστικά ή στοιχεία με ίδιο όνομα αλλά να χρησιμοποιούνται σε διαφορετικά στοιχεία. Συνεπώς χρειάζεται επιπρόσθετη πληροφορία για κάθε ιδιότητα τύπου δεδομένων προκειμένου να αποφασισθεί το όνομα του χαρακτηριστικού ή του στοιχείου στο οποίο αντιστοιχεί.
- Σε ένα OWL άτομο η σειρά με την οποία θα εμφανιστούν οι ιδιότητες του και οι τιμές σε αυτές δεν έχει καμία σημασία. Αντίθετα, η τοποθέτηση στη σωστή σειρά των στοιχείων που περιέχονται σε ένα MPEG-7 στοιχείο είναι κρίσιμης σημασίας, καθώς κάθε παραβίαση της σωστής σειράς των στοιχείων οδηγεί σε μη έγκυρη (invalid) MPEG-7 περιγραφή.

Εξαιτίας όλων των παραπάνω λόγων, κρίθηκε αναγκαία η δημιουργία δύο οντολογιών κανόνων μετασχηματισμού:

1. Της OWL2XMLRules η οποία περιέχει κλάσεις και ιδιότητες που περιγράφουν τη γενική συμπεριφορά ενός σύνθετου XML στοιχείου που πρόκειται να δημιουργηθεί από τη μετατροπή μιας OWL δομής.
2. Της OWL2MPEG7Rules η οποία εισάγει την OWL2XMLRules και δημιουργεί άτομα των κλάσεων της. Τα άτομα αυτά περιγράφουν τη συμπεριφορά κάθε MPEG-7 σύνθετου στοιχείου που πρόκειται να δημιουργηθεί από τη μετατροπή μιας OWL δομής. Για παράδειγμα, υπάρχει πληροφορία για τη σειρά των περιλαμβανόμενων στοιχείων κάθε MPEG-7 στοιχείου που θα προκύψει από μετασχηματισμό OWL κλάσης ή ατόμου, όπως επίσης πληροφορία όσον αφορά τον τύπο μετασχηματισμού και το όνομα μιας OWL ιδιότητας τύπου δεδομένων.

Η χρησιμοποίηση των παραπάνω οντολογιών από το σύστημα κατά τη διεργασία μετασχηματισμού των OWL δομών σε MPEG-7 στοιχεία απαιτεί την ενσωμάτωσή τους στην τρέχουσα χρήση της εφαρμογής. Αυτή η ενσωμάτωση επιτυγχάνεται με την εισαγωγή σε ένα αρχείο προσαρμογής, το οποίο θα περιλαμβάνει και την οντολογική υποδομή του DS-MIRF πλαισίου, ενός στοιχείου εξαγόμενης μορφής (ExportedFormats) που θα περιέχει ένα στοιχείο αρχείου (File) με τα χαρακτηριστικά και τιμές:

- Όνομα της εξαγόμενης μορφής (formatname) με τιμή Mpeg7.
- Τύπος της εξαγόμενης μορφής (formattype) με τιμή Mpeg7.
- Όνομα αρχείου κανόνων (rulesfilename) με τιμή το URI της οντολογίας OWL2MPEG7Rules.

- URI της ανώτερης οντολογίας (coreontologyURI) με τιμή το URI της ανώτερης οντολογίας που περιγράφει πλήρως το MPEG-7 MDS.

Με το άνοιγμα αυτού του αρχείου προσαρμογής, πέρα από το διάβασμα των ανώτερων οντολογιών και των οντολογιών περιοχής που περιλαμβάνει, διαβάζονται και οι οντολογίες κανόνων μετασχηματισμού OWL2XMLRules και OWL2MPEG7Rules προκειμένου να χρησιμοποιηθεί η πληροφορία που περιέχουν και να καθοδηγήσουν το Μετατροπέα OWL σε MPEG-7 ώστε να παράγει έγκυρες MPEG-7 περιγραφές.

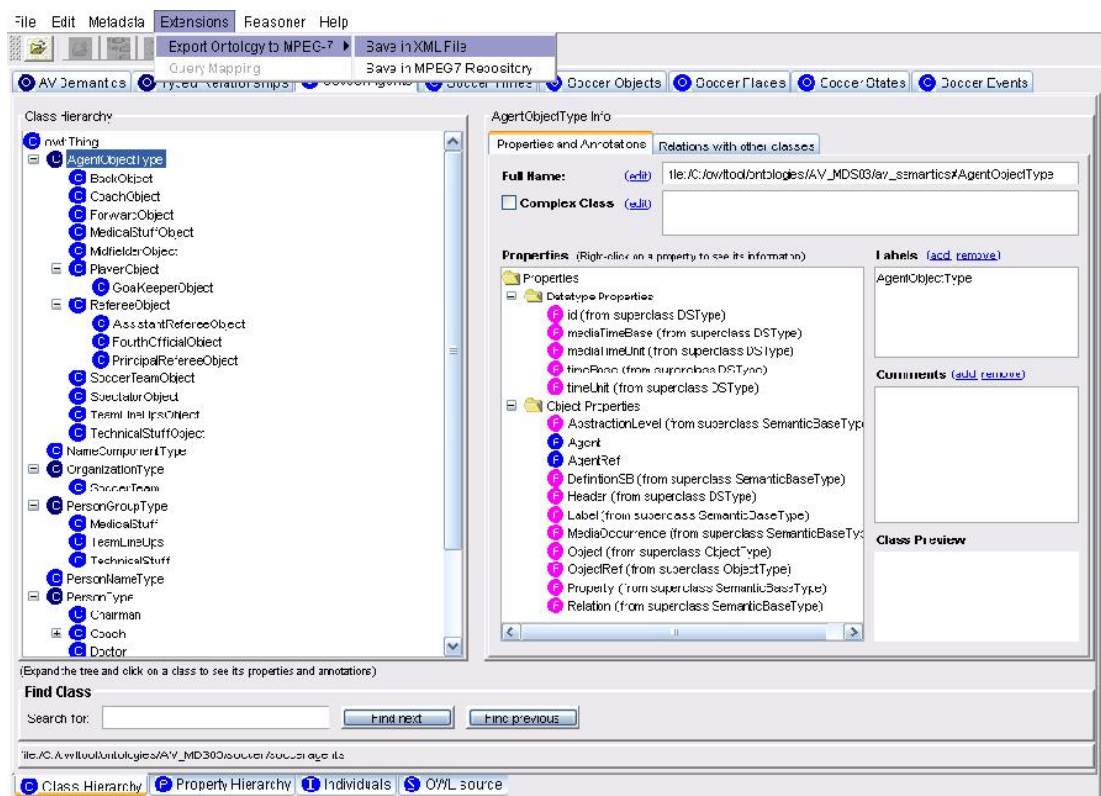
8.4. Υλοποίηση μετασχηματισμών και αποθήκευση σε MPEG-7 μορφή

8.4.1. Εισαγωγή

Το GraphOnto, όπως αναφέρθηκε, έχει τη δυνατότητα να μετασχηματίσει σε MPEG-7 έγγραφα τόσο τις οντολογίες περιοχής που επεκτείνουν την ανώτερη οντολογία του MPEG-7 MDS όσο και τις διάφορες σημασιολογικές περιγραφές που βασίζονται σε αυτές τις οντολογίες περιοχής, αλλά που έχουν οριστεί εκτός των πλαισίων τους. Για να επιτευχθούν αυτοί οι μετασχηματισμοί απαιτείται το άνοιγμα ενός αρχείου προσαρμογής που θα περιέχει την οντολογική υποδομή του DS-MIRF πλαισίου και θα ορίζει τις οντολογίες κανόνων μετασχηματισμού που περιγράφηκαν στην προηγούμενη παράγραφο.

8.4.2. Μετασχηματισμός οντολογίας περιοχής και αποθήκευση MPEG-7 περιγραφής

Ύστερα από το άνοιγμα του κατάλληλου αρχείου προσαρμογής και του διαβάσματος όλων των περιλαμβανομένων σε αυτό οντολογιών, ο μετασχηματισμός μιας OWL οντολογίας περιοχής σε μια MPEG-7 περιγραφή αφηρημένων οντοτήτων αποτελεί εύκολη υπόθεση για το χρήστη, αφού αρκεί η επιλογή "Export Ontology to MPEG-7" από το μενού "Extensions" του βασικού μενού της εφαρμογής όπως φαίνεται και στην επόμενη εικόνα.



Εικόνα 91: Στιγμιότυπο της εφαρμογής με ανοιχτές όλες τις οντολογίες που περιλαμβάνει η οντολογική υποδομή του πλαισίου DS-MIRF και επιλογή μετασχηματισμού της τρέχουσας οντολογίας σε MPEG-7 μορφή.

Στη συνέχεια, οι δύο παρακάτω δυνατότητες αποθήκευσης εμφανίζονται:

1. *Αποθήκευση σε XML αρχείο:* Η MPEG-7 περιγραφή που θα προκύψει από το μετασχηματισμό της OWL οντολογίας θα αποθηκευθεί σε ένα σωστά δομημένο και έγκυρο MPEG-7 έγγραφο του οποίου το όνομα και την τοποθεσία στο τοπικό σύστημα αρχείων θα επιλέξει ο χρήστης.
2. *Αποθήκευση στην MPEG-7 αποθήκη μεταδεδομένων:* Για την MPEG-7 περιγραφή που θα προκύψει από το μετασχηματισμό της OWL οντολογίας δημιουργείται ένα σωστά δομημένο και έγκυρο MPEG-7 έγγραφο του οποίου το όνομα ταυτίζεται με το όνομα της οντολογίας. Το έγγραφο αυτό θα τοποθετηθεί σε επιλεγμένο από το χρήστη container της αποθήκης. Πιο συγκεκριμένα, η επιλογή του container γίνεται με τον εξής τρόπο:
 - a. Αν δεν οριστεί όνομα container, τότε το έγγραφο τοποθετείται στον προκαθορισμένο (default) container.
 - b. Αν οριστεί όνομα container και υπάρχει container με αυτό το όνομα στην αποθήκη, τότε το έγγραφο τοποθετείται σε αυτόν τον container.
 - c. Αν οριστεί όνομα container και δεν υπάρχει container με αυτό το όνομα στην αποθήκη, τότε δημιουργείται ένας νέος container με όνομα την επιλογή του χρήστη και το έγγραφο τοποθετείται σε αυτόν το νέο container.

Αφού επιλεγθεί η τοποθεσία αποθήκευσης του εγγράφου, η οντολογία μετασχηματίζεται στην MPEG-7 περιγραφή αφηρημένων οντοτήτων μέσω του μετατροπέα OWL σε MPEG-7 και της χρήσης των κανόνων μετασχηματισμού και της πληροφορίας που περιέχεται στις οντολογίες κανόνων μετασχηματισμού. Αναλυτικότερα, τα βήματα που ακολουθούνται για το μετασχηματισμό της οντολογίας και την εξαγωγή έγκυρης MPEG-7 πληροφορίας είναι τα ακόλουθα:

1. Μετασχηματισμός των χαρακτηριστικών της οντολογίας και των οντολογιών που εισάγονται σύμφωνα με τους κανόνες μετασχηματισμού OWL οντολογιών.
2. Μετασχηματισμός των ατόμων της οντολογίας των οποίων ο τύπος είναι η κλάση "TermUseType" ή κάποια υποκλάση της σύμφωνα με τους κανόνες μετασχηματισμού OWL ατόμων.
3. Μετασχηματισμός των ατόμων της οντολογίας των οποίων ο τύπος είναι η κλάση "RelationType" ή κάποια υποκλάση της σύμφωνα με τους κανόνες μετασχηματισμού OWL ατόμων.
4. Μετασχηματισμός των κλάσεων της οντολογίας οι οποίες αποτελούν υποκλάσεις της κλάσης "SemanticBaseType" σύμφωνα με τους κανόνες μετασχηματισμού OWL ατόμων και μετασχηματισμός των ιδιοτήτων τύπου δεδομένων και αντικειμένων στις οποίες οι κλάσεις έχουν περιορισμό εύρους τιμών "έχει τιμή" σύμφωνα με τους κανόνες μετασχηματισμού OWL ιδιοτήτων τύπου δεδομένων και αντικειμένων.
5. Μετασχηματισμός των ατόμων της οντολογίας των οποίων ο τύπος είναι η κλάση "SemanticBaseType" ή κάποια υποκλάση της σύμφωνα με τους κανόνες μετασχηματισμού OWL ατόμων και μετασχηματισμός των ιδιοτήτων τύπου δεδομένων και αντικειμένων στις οποίες τα άτομα έχουν τιμή σύμφωνα με τους κανόνες μετασχηματισμού OWL ιδιοτήτων τύπου δεδομένων και αντικειμένων.

Με την ολοκλήρωση των παραπάνω βημάτων, η OWL οντολογία έχει μετασχηματιστεί σε μια MPEG-7 περιγραφή η οποία έχει αποθηκευθεί σε ένα XML έγγραφο, που υπακούει στο MPEG-7 MDS. Το αρχείο αυτό θα βρίσκεται είτε στο τοπικό σύστημα αρχείων οπότε θα μπορεί να ανοιχθεί από έναν XML συντάκτη είτε θα έχει εισαχθεί στην MPEG-7 αποθήκη μεταδεδομένων οπότε ένα μήνυμα πληροφορίας από την εφαρμογή ειδοποιεί το χρήστη για την επιτυχή εισαγωγή του στην αποθήκη.

8.4.3. Μετασχηματισμός OWL μεταδεδομένων και αποθήκευση MPEG-7 περιγραφής

Ύστερα από το άνοιγμα του κατάλληλου αρχείου προσαρμογής και του διαβάσματος όλων των περιλαμβανομένων σε αυτό οντολογιών, υπάρχει ως γνωστόν και η δυνατότητα δημιουργίας OWL σημασιολογικών περιγραφών με βάση αυτές τις οντολογίες αλλά εκτός των

πλαισίων τους μέσω του παράθυρου μεταδεδομένων. Έτσι, σύμφωνα με τη γνωστή διαδικασία μπορούν να επιλεγούν κλάσεις και ιδιότητες προκειμένου να δημιουργηθούν νέα άτομα στο παράθυρο μεταδεδομένων και κατόπιν να συμπληρωθούν οι τιμές των ατόμων στις επιλεγμένες ιδιότητες. Η διαδικασία εισαγωγής τιμών στις ιδιότητες των ατόμων είναι ακριβώς ίδια με αυτήν που περιγράφηκε στην παράγραφο 6.2.9. Μόνη διαφορά αυτή τη φορά, είναι ότι το κουμπί εξαγωγής σε MPEG-7 είναι ενεργοποιημένο.

Με το πάτημα αυτού του κουμπιού αρχίζει η διαδικασία εξαγωγής της πληροφορίας των OWL ατόμων σε MPEG-7 περιγραφές. Η διαδικασία βέβαια, όπως συμβαίνει και στην περίπτωση της εξαγωγής σε OWL/RDF μορφή, αρχίζει μόνο και εφόσον έχουν εισαχθεί στις ιδιότητες των ατόμων τιμές των οποίων ο αριθμός ικανοποιεί τους τυχόν περιορισμούς πολλαπλότητας ελάχιστου ή ακριβούς πλήθους. Στη συνέχεια, εμφανίζεται μια λίστα με δύο επιλογές που αναφέρονται στους δύο διαφορετικούς τύπους που μπορεί να έχει το MPEG-7 έγγραφο και οι οποίοι είναι:

1. *Ανεξάρτητη MPEG-7 σημασιολογική περιγραφή:* Επιλέγεται τα μετασχηματισμένα άτομα να αποτελέσουν μια αυτόνομη σημασιολογική περιγραφή. Ο μετασχηματισμός τους γίνεται σύμφωνα με τους κανόνες μετασχηματισμού OWL/RDF μεταδεδομένων. Όπως και στο μετασχηματισμό των οντολογιών, μετασχηματίζονται πρώτα τα άτομα των οποίων ο τύπος είναι η κλάση "TermUseType" ή κάποια υποκλάση της, ακολουθούν τα άτομα των οποίων ο τύπος είναι η κλάση "RelationType" ή κάποια υποκλάση της και τέλος μετασχηματίζονται τα άτομα των οποίων ο τύπος είναι η κλάση "SemanticBaseType" ή κάποια υποκλάση της και μαζί με αυτά και οι ιδιότητες τύπου δεδομένων και αντικειμένων στις οποίες έχουν τιμή.
2. *Υπομνηματισμός (annotation) περιεχομένου πολυμέσων:* Επιλέγεται τα μετασχηματισμένα άτομα να χρησιμοποιηθούν για τον υπομνηματισμό συγκεκριμένου αρχείου πολυμέσων. Ακολούθως, ζητείται η επιλογή του τύπου (εικόνα ή βίντεο) του πολυμέσου και στη συνέχεια έπεται η αναζήτηση του αρχείου πολυμέσων στο τοπικό σύστημα αρχείων. Αν επιλεγεί εικόνα ως τύπος του πολυμέσου, αναζητώνται αρχεία με μία από τις ακόλουθες καταλήξεις: .jpg, .bmp, .gif, .png, .tiff, .ico, ενώ αν αντίθετα επιλεγεί βίντεο ως τύπος του πολυμέσου, αναζητώνται αρχεία με μία από τις ακόλουθες καταλήξεις: .avi, .mpg, .mpeg, .wmv, .mkv, .mp4, .mov, .rm. Ο μετασχηματισμός των ατόμων γίνεται σύμφωνα με τους κανόνες μετασχηματισμού OWL/RDF μεταδεδομένων με μερικές ελαφρές τροποποιήσεις όπως:
 - a. Το στοιχείο περιγραφής ("Description") είναι πλέον του τύπου (xsi:type) "ContentEntityType".
 - b. Το στοιχείο περιγραφής περιέχει ένα στοιχείο "MultimediaContent" του τύπου "ImageType" ή "VideoType".
 - c. Το στοιχείο "MultimediaContent" περιλαμβάνει ένα στοιχείο "Image" ή "Video".

- d. Το στοιχείο "Image" ή "Video" περιλαμβάνει ένα στοιχείο "MediaLocator" που έχει ένα στοιχείο "MediaUri" το οποίο περιλαμβάνει την τοποθεσία του αρχείου πολυμέσων στο τοπικό σύστημα αρχείων.
- e. Το στοιχείο "Semantics" παύει να υπάρχει μέσα στο στοιχείο περιγραφής, αλλά αντίθετα δημιουργείται ένα στοιχείο "Semantic" ακριβώς με τους ίδιους κανόνες και το οποίο περιλαμβάνεται πλέον μέσα στο στοιχείο "Image" ή "Video". Η σειρά μετασχηματισμού των ατόμων είναι ακριβώς όμοια με αυτή που ακολουθείται στην ανεξάρτητη σημασιολογική περιγραφή.

Αφού οριστεί ο τύπος του εγγράφου και συνεπώς η δομή του, η οποία όπως φάνηκε από τα παραπάνω διαφοροποιείται ανάλογα με τον τύπο, οι δύο γνωστές δυνατότητες αποθήκευσης εμφανίζονται:

1. *Αποθήκευση σε XML αρχείο:* Η MPEG-7 περιγραφή που θα προκύψει από το μετασχηματισμό των OWL μεταδεδομένων θα αποθηκευθεί σε ένα σωστά δομημένο και έγκυρο MPEG-7 έγγραφο του οποίου το όνομα και την τοποθεσία στο τοπικό σύστημα αρχείων θα επιλέξει ο χρήστης.
2. *Αποθήκευση στην MPEG-7 αποθήκη μεταδεδομένων:* Για την MPEG-7 περιγραφή που θα προκύψει από το μετασχηματισμό των OWL μεταδεδομένων δημιουργείται ένα σωστά δομημένο και έγκυρο MPEG-7 έγγραφο του οποίου το όνομα ζητείται από το χρήστη. Το έγγραφο αυτό θα τοποθετηθεί σε κατάλληλα επιλεγμένο από την ίδια την αποθήκη container της αποθήκης. Πιο συγκεκριμένα, η επιλογή του container γίνεται με τον εξής τρόπο:
 - a. Αν η οντολογία στην οποία ανήκουν τα άτομα έχει μετασχηματισθεί σε MPEG-7 περιγραφή και έχει εισαχθεί στην αποθήκη, τότε το έγγραφο εισάγεται στον container στον οποίο βρίσκεται και η MPEG-7 περιγραφή της οντολογίας.
 - b. Αν η οντολογία στην οποία ανήκουν τα άτομα δεν έχει εισαχθεί στην αποθήκη ως MPEG-7 περιγραφή, τότε το έγγραφο εισάγεται στον προκαθορισμένο container.

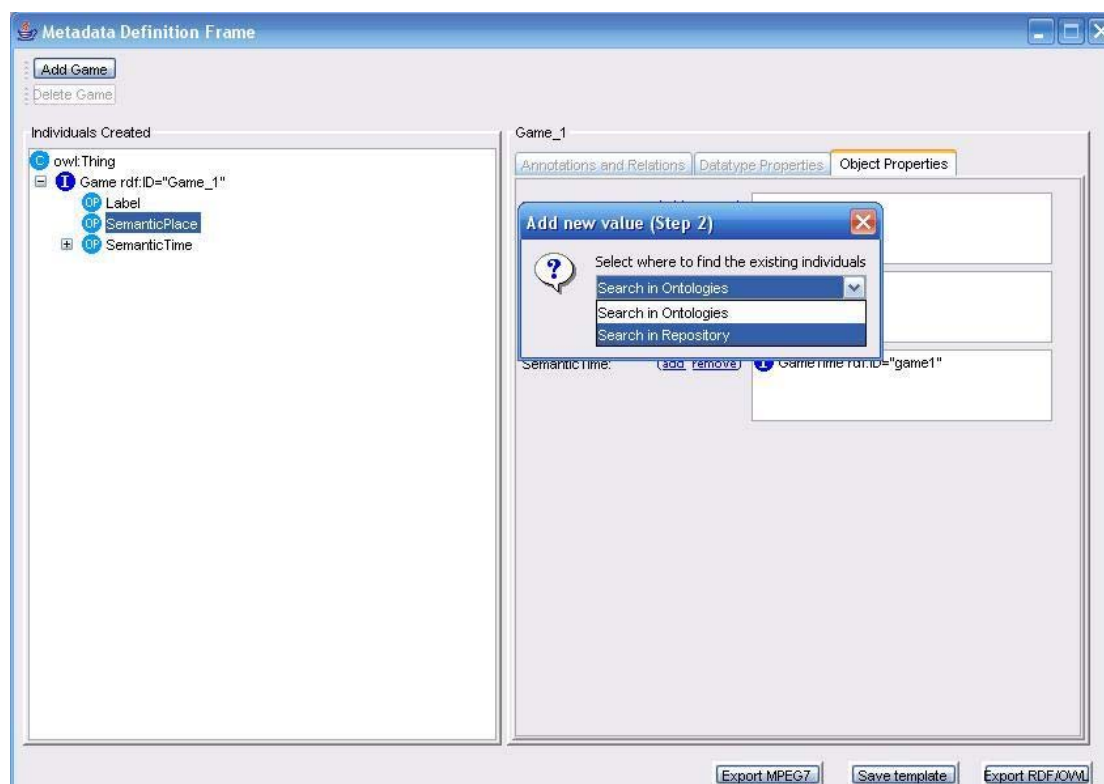
Με την επιλογή του τρόπου αποθήκευσης, τα OWL άτομα έχουν μετασχηματιστεί σε μια MPEG-7 περιγραφή (είτε ανεξάρτητη είτε αποτελεί υπομνηματισμό περιεχομένου πολυμέσων) η οποία έχει αποθηκευθεί σε ένα XML έγγραφο, που υπακούει στο MPEG-7 MDS, είτε στο τοπικό σύστημα αρχείων είτε στην MPEG-7 αποθήκη μεταδεδομένων.

8.5. Επαναχρησιμοποίηση αποθηκευμένων MPEG-7 περιγραφών

Όπως έχει ήδη αναφερθεί στην περιγραφή της αρχιτεκτονικής του συστήματος (στο κεφάλαιο 5 και συγκεκριμένα στην ενότητα 5.8), η επικοινωνία του συστήματος με την

αποθήκη μεταδεδομένων είναι αμφίδρομη. Με άλλα λόγια, πέρα από την αποθήκευση στην αποθήκη των OWL ατόμων σε MPEG-7 μορφή, επιτρέπεται η επαναχρησιμοποίηση τους κατά τη διάρκεια δημιουργίας νέων σημασιολογικών περιγραφών. Με αυτόν τον τρόπο, αποφεύγεται σε κάθε νέα χρήση της εφαρμογής ο επανορισμός ατόμων που περιέχουν την ίδια πληροφορία.

Αναλυτικότερα, στο παράθυρο μεταδεδομένων κατά τη διάρκεια πρόσθεσης νέας τιμής σε μια ιδιότητα αντικειμένων ενός ατόμου, υπάρχει πλέον η δυνατότητα αναζήτησης υπάρχοντων ατόμων όχι μόνο μέσα στις οντολογίες και την τρέχουσα σημασιολογική περιγραφή, αλλά και μέσα στην αποθήκη μεταδεδομένων. Έτσι, όταν επιλεγεί να προστεθεί ένα υπάρχον άτομο ως τιμή σε μια ιδιότητα αντικειμένων, εμφανίζονται δύο επιλογές (όπως φαίνεται και στην εικόνα που ακολουθεί): η γνωστή επιλογή αναζήτησης στις ανοιχτές οντολογίες και η επιλογή αναζήτησης στην αποθήκη μεταδεδομένων.



Εικόνα 92: Στιγμιότυπο του παράθυρου μεταδεδομένων και εμφάνιση της επιλογής αναζήτησης ατόμων στις οντολογίες ή στην αποθήκη μεταδεδομένων.

Επιλέγοντας την αναζήτηση ατόμων μέσα στην αποθήκη μεταδεδομένων, το σύστημα αναζητά, μέσω μιας μεθόδου του API της αποθήκης, σημασιολογικές περιγραφές οι οποίες προήλθαν από μετασχηματισμό OWL ατόμων τα οποία ανήκουν στις κλάσεις (και τις τυχόν υποκλάσεις τους) που ορίζει το εύρος τιμών και οι τυχόν περιορισμοί της επιλεγμένης ιδιότητας αντικειμένων. Ακολουθώς εμφανίζεται μια λίστα με τις επιτρεπόμενες σημασιολογικές περιγραφές που ανακτήθηκαν από την αποθήκη. Η επιλογή κάποιας περιγραφής από τη λίστα έχει ως συνέπεια τη δημιουργία ενός νέου OWL ατόμου με βάση την

πληροφορία αυτής της περιγραφής και την προσθήκη του ως τιμή στην επιλεγμένη ιδιότητα του τρέχοντος ατόμου.

8.6. Ανακεφαλαίωση

Στο κεφάλαιο αυτό παρουσιάστηκε η πλήρης και πλούσια παρεχόμενη λειτουργικότητα του GraphOnto ως ένα εργαλείο που επιτρέπει τη σημασιολογική περιγραφή οπτικοακουστικού υλικού η οποία βασίζεται στο πρότυπο MPEG-7. Χρησιμοποιώντας τους κανόνες και τις οντολογίες μετασχηματισμού από OWL σε MPEG-7, μετασχηματίζει τις οντολογίες περιοχής και τις ανεξάρτητες περιγραφές OWL μεταδεδομένων σε MPEG-7 περιγραφές οπτικοακουστικού υλικού. Αυτές οι MPEG-7 περιγραφές αποθηκεύονται σε XML έγγραφα ή στην MPEG-7 αποθήκη μεταδεδομένων προκειμένου να είναι δυνατή η επαναχρησιμοποίησή τους.

Κεφάλαιο 9°

9. Πειραματική Αξιολόγηση

Με βάση όσα αναφέρθηκαν προηγουμένως, το GraphOnto είναι ένα πλήρες γραφικό εργαλείο για τη διαχείριση OWL οντολογιών παρέχοντας στον τομέα αυτό την ίδια λειτουργικότητα με υπάρχοντα εργαλεία ανάπτυξης οντολογιών, ανάμεσα στα οποία το πιο γνωστό είναι το Protégé (το οποίο περιγράφηκε στο κεφάλαιο 3). Παράλληλα βέβαια το GraphOnto με τις επεκτάσεις του παρέχει σημαντική επιπρόσθετη λειτουργικότητα η οποία το κάνει να ξεχωρίζει από τα υπόλοιπα εργαλεία. Όμως ακόμα και στον κοινό τομέα της διαχείρισης οντολογιών, το GraphOnto διακρίνεται για την καλύτερη απόδοσή του σε σχέση με το Protégé.

Για τη σύγκριση λοιπόν της απόδοσης του GraphOnto και του Protégé, πραγματοποιήθηκε μια σειρά από πειράματα στην πιο χρονοβόρα λειτουργία, το διάβασμα υπάρχουσων οντολογιών. Οι μετρικές αξιολόγησης που χρησιμοποιήθηκαν είναι:

1. Ο χρόνος που απαιτείται για το διάβασμα μιας υπάρχουσας οντολογίας η οποία βρίσκεται στο τοπικό σύστημα αρχείων και για τη γραφική της απεικόνιση από τις δύο εφαρμογές.
2. Το ποσό της κυρίας μνήμης που δεσμεύουν οι δύο εφαρμογές κατά το διάβασμα των οντολογιών.

Οι τέσσερις OWL οντολογίες που επιλέχθηκαν για να χρησιμοποιηθούν στα πειράματα μας είναι οι εξής:

1. MiniFoot Ontology: Μία μικρή οντολογία που αναφέρεται στον τομέα ποδοσφαίρου.
2. Travel Ontology: Μια οντολογία που περιγράφει ταξιδιωτικές υπηρεσίες.
3. AVSemantics Ontology: Μια οντολογία που αντιπροσωπεύει πλήρως το τμήμα του MPEG-7 MDS που αναφέρεται στην αναπαράσταση σημασιολογικών περιγραφών.
4. Food Ontology: Μια σύνθετη οντολογία που περιγράφει τρόφιμα και χρησιμοποιείται από το w3c.org.

Το πλήθος των OWL δομών που περιέχονται στις παραπάνω οντολογίες φαίνονται στον παρακάτω πίνακα:

Όνομα οντολογίας	Πλήθος από classes	Πλήθος από Datatype properties	Πλήθος από Object properties	Πλήθος από Individuals
MiniFoot	8	5	7	9
Travel	6	13	8	11
AVSemantics	97	79	126	0
Food	63	0	4	45

Πίνακας 2: Χαρακτηριστικά των οντολογιών των πειραμάτων

Η αξιολόγηση έλαβε χώρα σε υπολογιστικό σύστημα με τα εξής χαρακτηριστικά:

- Επεξεργαστής: Intel Core Duo Processor T2400 1,83GHz.
- Κύρια μνήμη: 1 GByte of RAM.
- Σκληρός δίσκος: τύπος SATA, χωρητικότητα 160GB και ταχύτητα περιστροφής 4200 rpm.
- Λειτουργικό σύστημα: Microsoft Windows XP Media Center Edition.

Τα αποτελέσματα της αξιολόγησης που φαίνονται στον πίνακα 2 δείχνουν ότι το GraphOnto υπερέρχει το Protégé και στις 4 οντολογίες και στις 2 μετρικές. Πιο συγκεκριμένα, το GraphOnto χρησιμοποιεί περίπου τα 4/7 της μνήμης που χρησιμοποιεί το Protégé για την ίδια οντολογία, ενώ όσον αφορά το χρόνο που απαιτείται για το διάβασμα οντολογίας, στο GraphOnto απαιτείται κάτι λιγότερο από το μισό του χρόνου που απαιτείται από το Protégé (ειδικά για την Travel οντολογία ο χρόνος που απαιτείται είναι λιγότερος από το 1/3 του χρόνου που απαιτείται από το Protégé). Η βασική εξήγηση που μπορεί να δοθεί στα παραπάνω αποτελέσματα είναι ότι το Protégé έχει αναπτυχθεί γύρω από μια frame-based προσέγγιση για την αναπαράσταση των οντολογιών και ότι η διαχείριση των OWL οντολογιών γίνεται από το OWL plug-in του Protégé, τη στιγμή που η υλοποίηση του GraphOnto βασίστηκε εξ αρχής αποκλειστικά στην OWL.

Όνομα οντολογίας	Χρόνος διαβάσματος οντολογίας από GraphOnto	Χρόνος διαβάσματος οντολογίας από Protégé	Χρήση κυρίας μνήμης από GraphOnto	Χρήση κυρίας μνήμης από Protégé
MiniFoot	1,44 seconds	5,80 seconds	42724 Kbytes	73692 Kbytes
Travel	1,47 seconds	6,80 seconds	40240 Kbytes	71540 Kbytes
AVSemantics	3,29 seconds	7,88 seconds	45384 Kbytes	72984 Kbytes
Food	3,33 seconds	8,35 seconds	44388 Kbytes	76104 Kbytes

Πίνακας 3: Αποτελέσματα αξιολόγησης απόδοσης μεταξύ GraphOnto και Protégé στο διάβασμα οντολογίας

Παράλληλα, θα πρέπει να τονιστεί το μεγάλο συγκριτικό πλεονέκτημα του GraphOnto σε σχέση με το Protégé που έγκειται στη χρήση των αρχείων προσαρμογής με τη βοήθεια των οποίων πολλαπλές οντολογίες ανοίγουν με ένα μόνο κλικ. Ενδεικτικά, θα αναφερθεί η περίπτωση του αρχείου προσαρμογής που περιέχει την οντολογική υποδομή του DS-MIRF πλαισίου η οποία αποτελείται ως γνωστόν από:

- Μια ανώτερη οντολογία που αποτυπώνει πλήρως το MPEG-7 MDS πρότυπο.
- Μια οντολογία εφαρμογής που αντιμετωπίζεται και αυτή ως ανώτερη από το σύστημα και που αποτυπώνει πλήρως τις Σχέσεις Τύπων του MPEG-7 (Typed Relations of MPEG-7).
- Έξι οντολογίες περιοχής που καλύπτουν το πεδίο του ποδοσφαίρου.

Ταυτόχρονα, θα πρέπει να τονιστεί ότι διαβάζονται, χωρίς όμως να εμφανίζονται στο γραφικό περιβάλλον, και οι οντολογίες κανόνων μετασχηματισμού από OWL σε MPEG-7, δηλαδή η οντολογία OWL2XMLRULES και η οντολογία OWL2MPEG7RULES. Το πλήθος των OWL δομών που περιέχονται σε όλες τις παραπάνω οντολογίες φαίνονται στον παρακάτω πίνακα:

Όνομα οντολογίας	Πλήθος από classes	Πλήθος από Datatype properties	Πλήθος από Object properties	Πλήθος από Individuals
AVSemantics	97	79	126	0
TypedRelationships	145	3	0	0
SoccerAgents	39	3	27	94
SoccerTimes	9	1	6	8
SoccerObjects	5	0	2	0
SoccerPlaces	19	0	18	12
SoccerStates	1	4	1	0
SoccerEvents	117	1	25	10
OWL2XMLRULES	8	11	3	0
OWL2MPEG7RULES	0	4	1	91
Σύνολο	440	106	209	215

Πίνακας 4: Χαρακτηριστικά των οντολογιών που περιλαμβάνονται στην οντολογική υποδομή του DS-MIRF πλαισίου

Οι μετρήσεις που έγιναν έδειξαν ότι όλες οι παραπάνω οντολογίες ανοίχθηκαν με μία μόνο ενέργεια χρήστη και διαβάστηκαν:

- Σε χρόνο μόνο 13,68 δευτερολέπτων, δηλαδή μόνο 73% περισσότερος χρόνος από αυτόν που χρειάζεται το Protégé για να διαβάσει μόνο την πρώτη οντολογία (AVSemantics).
- Και καταλαμβάνοντας 104720 Kbytes της κύριας μνήμης, δηλαδή μόνο 43% περισσότερη μνήμη από αυτήν που χρειάζεται το Protégé για να διαβάσει μόνο την πρώτη οντολογία (AVSemantics).

Κεφάλαιο 10^ο

10. Ανακεφαλαίωση

10.1. Συμπεράσματα

Στην παρούσα μεταπτυχιακή διατριβή, παρουσιάστηκε το GraphOnto, ένα γραφικό σύστημα που απλοποιεί τον ορισμό και την επεξεργασία τόσο οντολογιών που βασίζονται σε διεθνή πρότυπα (όπως το MPEG-7 για σημασιολογική περιγραφή περιεχομένου πολυμέσων) όσο και οντολογιών περιοχής (domain specific ontologies). Η γλώσσα οντολογιών που υποστηρίζεται είναι η OWL η οποία αποτελεί στις μέρες μας την κυρίαρχη γλώσσα ορισμού οντολογιών. Η λειτουργικότητα που παρέχει το GraphOnto μπορεί να δειχθεί μέσω δύο διαφορετικών πρισμάτων.

Το GraphOnto αποτελεί έναν πλήρη και γενικό γραφικό συντάκτη OWL (OWL-DL για την ακρίβεια) οντολογιών ο οποίος καλύπτει την πλήρη σημασιολογία της OWL. Παρέχει τη λειτουργικότητα που υποστηρίζουν και οι υπόλοιποι συντάκτες οντολογιών, ανάμεσα στους οποίους ξεχωρίζει το πολύ γνωστό Protégé. Σε σχέση με το τελευταίο παρουσιάζει πολλά επιπλέον σημαντικά χαρακτηριστικά όπως η ταυτόχρονη διαχείριση πολλαπλών οντολογιών, ο σημασιολογικός έλεγχος κατά τη διάρκεια επεξεργασίας των οντολογιών ώστε να μην παραβιάζονται οι σημασιολογικοί περιορισμοί της OWL, η υποστήριξη αντιστοιχίσεων μεταξύ OWL οντολογιών μέσω των OWL εκφράσεων αντιστοίχισης που επιτρέπουν τη δημιουργία προσωποποιημένων οντολογιών και τέλος η εξατομίκευση του γραφικού περιβάλλοντος σε επίπεδο εφαρμογής, έργου και χρήστη. Επιπλέον, πειραματικές μετρήσεις που διενεργήθηκαν απέδειξαν ότι το GraphOnto υπερέχει σημαντικά σε απόδοση (σε όρους χρόνου και χρήσης κύριας μνήμης) του Protégé στο διάβασμα οντολογιών, την πιο χρονοβόρα λειτουργία ενός OWL συντάκτη.

Το GraphOnto είναι ταυτόχρονα ένα γραφικό εργαλείο που επιτρέπει το σημασιολογικό υπομνηματισμό περιεχομένου πολυμέσων κάνοντας χρήση και επεκτείνοντας την οντολογική υποδομή του. Ειδικότερα, υποστηρίζει το σημασιολογικό υπομνηματισμό οπτικοακουστικών αντικειμένων με τη βοήθεια της ανώτερης οντολογίας του DS-MIRF που περιγράφει πλήρως το MPEG-7 MDS. Επιπρόσθετα, επιτρέπει τη χρήση και τον ορισμό οντολογιών περιοχής οι οποίες ενσωματώνονται στο πλαίσιο και επεκτείνουν την ανώτερη οντολογία με γνώση περιοχής. Αυτές οι οντολογίες περιοχής μαζί με τις σημασιολογικές περιγραφές πολυμέσων οι οποίες ορίζονται με βάση αυτές τις οντολογίες, μπορούν να μετασχηματιστούν μέσω των κανόνων μετασχηματισμού του DS-MIRF σε MPEG-7 σημασιολογικές περιγραφές περιεχομένου πολυμέσων. Αυτές οι MPEG-7 περιγραφές δύνανται να αποθηκευθούν είτε σε έγκυρα MPEG-7 αρχεία είτε στην MPEG-7 αποθήκη μεταδεδομένων του DS-MIRF.

Παράλληλα, το GraphOnto σύστημα περιλαμβάνει το GraphOnto API που είναι ένα JAVA API που παρέχει προγραμματιστική πρόσβαση για πλήρη διαχείριση (πρόσθεση, επεξεργασία, διαγραφή) όλων των δομών μιας OWL οντολογίας. Το API αυτό χρησιμοποιείται μέσα στο DS-MIRF πλαίσιο για την ανάπτυξη εφαρμογών ανάκτησης και φιλτραρίσματος οι οποίες βασίζονται σε οντολογίες.

Τέλος, θα πρέπει να τονιστεί ότι το GraphOnto χρησιμοποιήθηκε και στα πλαίσια του DELOS II Excellence Network in Digital Libraries της Ευρωπαϊκής Κοινότητας [32] [33] και συγκεκριμένα στα έργα (tasks):

- Στο έργο 3.6 (Vapeon) το GraphOnto χρησιμοποιείται για την δημιουργία οντολογιών εμπλουτισμένων με εικόνες (pictorially enriched ontologies).
- Στο έργο 3.9 (Automatic, Context-Of-Capture Based, Categorization, Structure Detection and Segmentation of News Telecasts) το GraphOnto χρησιμοποιείται για τη διαχείριση των οντολογιών που αποτελούν μέρος μιας εφαρμογής αυτόματης εξαγωγής τμημάτων πολυμέσων.
- Στο έργο 3.10 (CoCoMA) το GraphOnto χρησιμοποιείται για τη δημιουργία, αποθήκευση και διαχείριση σημασιολογικών μεταδεδομένων που αφορούν περιεχόμενο πολυμέσων.
- Στο έργο 3.11 (Natural Language and Speech Interfaces to Knowledge Repositories) το GraphOnto χρησιμοποιείται ως η βάση για τη δημιουργία οντολογιών και για την υποστήριξη MPEG-7 σημασιολογικών περιγραφών.
- Στο έργο 5.4 (Interoperability of e-Learning Applications with digital libraries) το GraphOnto χρησιμοποιείται για τη διαχείριση 3D οντολογιών και 3D MPEG-7 περιγραφών που αποθηκεύονται στην MPEG-7 αποθήκη μεταδεδομένων.
- Στο έργο 5.5 (Ontology Driven Interoperability) το GraphOnto χρησιμοποιείται για να διαχειρίζεται αντιστοιχίσεις μεταξύ οντολογιών που αφορούν τον τομέα της τέχνης.

Το GraphOnto είναι διαθέσιμο προς κατέβασμα (downloading) και χρήση στη διεύθυνση <http://astral.ced.tuc.gr/delos/content/demostrators/graphonto.zip>, ενώ στη διεύθυνση http://astral.ced.tuc.gr/delos/cls_resource_description.jsp?id=10410 βρίσκεται ένα αρχείο με οδηγίες χρήσης του.

10.2.Μελλοντικές επεκτάσεις

Μέσα στο πλαίσιο στο οποίο το GraphOnto αποτελεί ένα αμιγές σύστημα διαχείρισης OWL οντολογιών, θα μπορούσε να προταθεί ως μια μελλοντική επέκταση του συστήματος η ενσωμάτωση σε αυτό μιας OWL μηχανής εξαγωγής συμπερασμάτων (OWL Reasoner). Η χρήση αυτής της μηχανής μαζί με τους πολυάριθμους σημασιολογικούς ελέγχους που ήδη πραγματοποιούνται, θα ισχυροποιούσε τη δυνατότητά του να παράγει σωστές και συνεπείς οντολογίες. Επίσης, ένα τολμηρό βήμα που θα μπορούσε να γίνει αφού μελετηθούν ακριβώς

οι πιθανές επιπτώσεις του, θα ήταν η υποστήριξη της OWL Full, ώστε να καταστεί δυνατή η παροχή ακόμα μεγαλύτερης εκφραστικότητας στις δημιουργούμενες οντολογίες.

Ταυτόχρονα, μια σίγουρα πολύ χρήσιμη και άμεσα εφαρμόσιμη επέκταση θα ήταν η παροχή μέσω υπηρεσιών διαδικτύου (web services) αρκετών λειτουργιών του GraphOnto όπως η ανάκτηση και το διάβασμα πληροφορίας οντολογιών για παροχή δυνατότητας περιήγηση μέσα στις οντολογίες, ο έλεγχος οντολογιών σχετικά με τη συνέπεια τους, ο ορισμός μεταδεδομένων με βάση οντολογίες, ο ορισμός αντιστοιχήσεων μεταξύ δομών οντολογιών ή τέλος η προγραμματιστική πρόσβαση στην MPEG-7 αποθήκη μεταδεδομένων.

Επιπρόσθετα, αξίζει να τονισθεί ότι η πλήρης λειτουργικότητα του GraphOnto API συμβάλλει ήδη στην ανάπτυξη μέσα στο DS-MIRF πλαίσιο μιας εφαρμογής σχηματισμού ερωτήσεων βασισμένες σε οντολογίες (Ontology-based Query Formulation Tool). Η εφαρμογή αυτή βρίσκεται σε αρχικό στάδιο και στόχος της είναι να απευθύνει ερωτήσεις σε όλες τις οντολογίες. Οι ερωτήσεις αυτές, ανάλογα με την οντολογία στην οποία απευθύνονται, θα μετασχηματίζονται με τη βοήθεια των OWL αντιστοιχήσεων σε νέες ισοδύναμες ερωτήσεις ώστε να καλύπτουν τις δομές της κάθε οντολογίας.

Μια επιπλέον χρήση του GraphOnto API, θα ήταν η χρησιμοποίησή του σε μια εφαρμογή που θα παράγει με αυτοματοποιημένο τρόπο Java κλάσεις που θα βασίζονται στις δομές της κάθε οντολογίας (OWL Ontology Data Binding). Η εφαρμογή αυτή θα ναι παρόμοια με τα διάφορα εργαλεία XML Data Binding που παράγουν αυτόματα Java κλάσεις με βάση ένα συγκεκριμένο XML Schema. Μια απλή προσέγγιση αυτής της ιδέας θα ήταν η δημιουργία Java κλάσεων για κάθε κλάση της οντολογίας και η προσθήκη σε κάθε Java κλάση χαρακτηριστικών και μεθόδων που θα αντιστοιχούν στις ιδιότητες τύπου δεδομένων και αντικειμένων οι οποίες εφαρμόζονται στην αντίστοιχη OWL κλάση.

Παράλληλα, η λειτουργικότητα που παρέχει το παράθυρο μεταδεδομένων της εφαρμογής για δημιουργία σημασιολογικής περιγραφής με βάση OWL οντολογίες και για μετέπειτα μετασχηματισμό της σε MPEG-7 περιγραφή, παρέχει τη δυνατότητα ενσωμάτωσης αυτού του παράθυρου σε διάφορες εφαρμογές σημασιολογικού υπομνηματισμού περιεχομένου πολυμέσων, όπως για παράδειγμα ένα εργαλείο τμηματοποίησης (segmentation) οπτικοακουστικού υλικού.

Τέλος, η ευέλικτη σχεδίαση του συστήματος όσον αφορά το μετασχηματισμό των OWL δομών σε MPEG-7 οντότητες, με τη χρήση του στοιχείου εξαγόμενης μορφής των αρχείων προσαρμογής και των οντολογιών κανόνων μετασχηματισμού, μπορεί να επιτρέψει στο μέλλον το μετασχηματισμό των OWL δομών σε οντότητες κάποιου άλλου προτύπου. Μόνη προϋπόθεση αποτελεί η μελέτη και η δημιουργία κανόνων μετασχηματισμού της OWL μορφής στη μορφή του νέου προτύπου.

Δημοσιεύσεις














1. Tsinaraki C., Polydoros P., Christodoulakis S.: *"Interoperability support between MPEG-7/21 and OWL in DS-MIRF"*, In Transactions on Knowledge and Data Engineering (IEEE-TKDE), Special Issue on the Semantic Web Era, 2007.
2. Tsinaraki C., Polydoros P., Syntzanaki A., Martin A., Perego A., Christodoulakis S.: *"Semantic, Constraint & Preference Based Authoring of Multi-topic Multimedia Presentations"*, In the Journal of Digital Information Management (JDIM), December 2006.
3. Polydoros P., Tsinaraki C., Christodoulakis S.: *"GraphOnto: OWL-Based Ontology Management and Multimedia Annotation in the DS-MIRF Framework"*, In the Journal of Digital Information Management (JDIM), December 2006.
4. Tsinaraki C., Polydoros P., Syntzanaki A., Martin A., Perego A., Christodoulakis S.: *"Semantic, Constraint & Preference Based Authoring of Multi-topic Multimedia Presentations"*, In the proceedings of the Workshop on Multimedia Semantics 2006 (WMS 2006), pp. 5-13, 19-21 June 2006, Chania, Crete.
5. Polydoros P., Tsinaraki C., Christodoulakis S.: *"GraphOnto: OWL-Based Ontology Management and Multimedia Annotation in the DS-MIRF Framework"*, In the proceedings of the Workshop on Multimedia Semantics 2006 (WMS 2006), pp. 14-24, 19-21 June 2006, Chania, Crete.
6. C. Tsinaraki, P. Polydoros, F. Kazasis, S. Christodoulakis: *"Ontology-based Semantic Indexing for MPEG-7 and TV-Anytime Audiovisual Content"*, Multimedia Tools and Applications Journal, special issue on Video Segmentation for Semantic Annotation and Transcoding, 26, 299-325, Aug. 2005.
7. Tsinaraki C., Polydoros P., Christodoulakis S.: *"GraphOnto: A Component and a User Interface for the Definition and Use of Ontologies in Multimedia Information Systems"*, In the Proceedings of AVIVDiLib 2005, Cortona, Italy, April 2005.
8. Tsinaraki C., Polydoros P., Christodoulakis S.: *"Interoperability support for Ontology-based Video Retrieval Applications"*, In the Proceedings of Conference on Image and Video Retrieval (CIVR) 2004, Dublin/Ireland, July 2004.
9. Tsinaraki C., Polydoros P., Christodoulakis S.: *"Integration of OWL ontologies in MPEG-7 and TVAnytime compliant Semantic Indexing"*, In the proceedings of the 16th International Conference on Advanced Information Systems Engineering (CAISE), Riga, Latvia, June 2004.
10. Tsinaraki C., Polydoros P., Christodoulakis S.: *"Integration of OWL ontologies in MPEG-7 and TVAnytime compliant Semantic Indexing"*, In the proceedings of the 3rd HDMS, Athens, Greece, June 2004.
11. Tsinaraki C., Polydoros P., Moumoutzis N., Christodoulakis S.: *"Coupling OWL with MPEG-7 and TV-Anytime for Domain-specific Multimedia Information Integration and Retrieval"*, In the Proceedings of RIAO 2004, April 2004, Avignon/France

Αναφορές-Βιβλιογραφία

1. <http://www.w3.org/XML/>
2. <http://www.w3.org/XML/Schema>
3. <http://www.w3.org/>
4. <http://www.w3.org/RDF/>
5. <http://www.w3.org/Addressing/>
6. <http://www.w3.org/TR/rdf-schema/>
7. <http://www.w3.org/TR/daml+oil-reference>
8. <http://dl.kr.org/>
9. <http://www.w3.org/TR/owl-guide/>
10. <http://www.w3.org/TR/REC-xml-names/>
11. Salembier P.: "MPEG-7 Multimedia Description Schemes". IEEE TA on Circuits and Systems for Video Technology, 11, 6, 2001, 748-759.
12. Martines J. (Ed.): "ISO/IEC JTC1/SC29/WG11N5525 – MPEG-7 Overview (version 9)". International Organisation for Standardisation (ISO), <http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm>, Pattaya, March 2003.
13. Chang S.F., Sikora T., Puri A.: "Overview of the MPEG-7 standard". IEEE Transactions on Circuits and Systems for Video Technology 11, pp. 688–695.
14. International Standardization Organization (ISO): "15938-1:2001: Information Technology – Multimedia content description interface – Part 1: Systems". (2001) Version 1.
15. International Standardization Organization (ISO): "15938-2:2001: Information Technology – Multimedia content description interface – Part 2: Description Definition Language". (2001) Version 1.
16. International Standardization Organization (ISO): "15938-3:2001: Information Technology – Multimedia content description interface – Part 3: Visual". (2001) Version 1.
17. International Standardization Organization (ISO): "15938-4:2001: Information Technology – Multimedia content description interface – Part 4: Audio". (2001) Version 1.
18. International Standardization Organization (ISO): "15938-5:2003: Information Technology – Multimedia content description interface – Part 5: Multimedia description schemes". (2003) First Edition.
19. <http://jena.sourceforge.net/index.html>
20. <http://www.w3.org/TR/rdf-sparql-query/>
21. <http://protege.stanford.edu/>

22. <http://protege.stanford.edu/aboutus/aboutus.html>
23. <http://protege.stanford.edu/overview/protege-frames.html>
24. <http://www.ai.sri.com/~okbc/>
25. <http://protege.stanford.edu/overview/protege-owl.html>
26. <http://protege.stanford.edu/plugins/owl/jena-integration.html>
27. http://www.altova.com/products/semanticworks/semantic_web_rdf_owl_editor.html
28. <http://www.mindswap.org/2004/SMORE/>
29. <http://www.mindswap.org/2004/SWOOP/>
30. <http://www.mindswap.org/2003/pellet/>
31. Tsinaraki C., Polydoros P., Christodoulakis S.: "Interoperability support between MPEG-7/21 and OWL in DS-MIRF", In Transactions on Knowledge and Data Engineering (IEEE-TKDE), Special Issue on the Semantic Web Era, 2007.
32. <http://www.delos.info/>
33. <http://astral.ced.tuc.gr/delos/>
34. International Standardization Organization (ISO): "N5231:2002: MPEG-21 Overview v.5" (2002), <http://www.chiariglione.org/mpeg/standards/mpeg-21/mpeg-21.htm>.
35. International Standardization Organization (ISO): "N7204:2005: Introducing MPEG-21 DIA – an Overview" (2005), <http://www.chiariglione.org/mpeg/tutorials/technologies/mp21-dia/index.htm>.
36. Karanastasi A., Christodoulakis S.: "OntoNL: An Ontology-based Natural Language Interface Generator for Multimedia Repositories". AvivDiLib 2005, pp. 205-216.
37. Tsinaraki C., Polydoros P., Syntzanaki A., Martin A., Perego A., Christodoulakis S.: "Semantic, Constraint & Preference Based Authoring of Multi-topic Multimedia Presentations", In the Journal of Digital Information Management (JDIM), December 2006.
38. <http://www.oracle.com/database/berkeley-db/index.html>
39. Sean Bechhofer, Phillip Lord, Raphael Volz. Cooking the Semantic Web with the OWL API. 2nd International Semantic Web Conference, ISWC, Sanibel Island, Florida, October 2003.
40. <http://owl.man.ac.uk/api.shtml>
41. Java™ 2 Platform - *Standard Edition*, v 1.4.1 API Specification, <http://java.sun.com/j2se/1.4.2/docs/api/>
42. <http://info.borland.com/techpubs/jbuilder/index.html>
43. <http://xmlbeans.apache.org/>

Παράρτημα 1: Πίνακας περιγραφής χρησιμοποιούμενων εικονιδίων

Αριθμός εικονιδίου	Απεικόνιση εικονιδίου	Περιγραφή
1.		Χρησιμοποιείται για την απεικόνιση οντολογίας περιοχής στον τίτλο της καρτέλας που αντιστοιχεί στην οντολογία.
2.		Χρησιμοποιείται για την απεικόνιση ανώτερης οντολογίας στον τίτλο της καρτέλας που αντιστοιχεί στην οντολογία.
3.		Το βασικό εικονίδιο για απεικόνιση κλάσης.
4.		Χρησιμοποιείται στο δένδρο ιεραρχίας κλάσεων για την απεικόνιση κλάσης με παραπάνω από μία υπερκλάσεις και στο δένδρο ατόμων για την απεικόνιση του βοηθητικού κόμβου που αντιστοιχεί σε κλάση.
5.		Χρησιμοποιείται στο δένδρο ιεραρχίας κλάσεων για την απεικόνιση κλάσης που ανήκει σε ανώτερη οντολογία.
6.		Το βασικό εικονίδιο για απεικόνιση ιδιότητας.
7.		Χρησιμοποιείται στο δένδρο ιεραρχίας ιδιοτήτων για την απεικόνιση ιδιότητας με παραπάνω από μία υπεριδιότητες και στο δένδρο ιδιοτήτων κλάσης για απεικόνιση ιδιότητας κλάσης που έχει κληρονομηθεί από υπερκλάση.
8.		Χρησιμοποιείται στο δένδρο ιεραρχίας ιδιοτήτων για την απεικόνιση ιδιότητας που ανήκει σε ανώτερη οντολογία.
9.		Χρησιμοποιείται στο δένδρο ιδιοτήτων κλάσης για απεικόνιση ιδιότητας κλάσης που έχει κληρονομηθεί από υπερκλάση και που ανήκει σε ανώτερη οντολογία.
10.		Το βασικό εικονίδιο για απεικόνιση ατόμου.
11.		Χρησιμοποιείται στο δένδρο ατόμων για την απεικόνιση ατόμου με παραπάνω από ένα τύπο.
12.		Χρησιμοποιείται στο δένδρο ατόμων για την απεικόνιση ατόμου που ανήκει σε ανώτερη οντολογία.
13.		Χρησιμοποιείται για την απεικόνιση XML τύπου δεδομένων σε λίστες εύρους τιμών ιδιοτήτων τύπου δεδομένων.

14.		Χρησιμοποιείται για την απεικόνιση κυριολεκτικής τιμής που αποτελεί τμήμα απαρίθμησης σε λίστες εύρους τιμών ιδιοτήτων τύπου δεδομένων και στο παράθυρο δημιουργίας έκφρασης σύνθετης κλάσης για να αναπαραστήσει την απαρίθμηση ατόμων.
15.		Χρησιμοποιείται για την απεικόνιση ιδιότητας τύπου δεδομένων στο δένδρο ατόμων του παραθύρου μεταδεδομένων.
16.		Χρησιμοποιείται για την απεικόνιση ιδιότητας αντικειμένων στο δένδρο ατόμων του παραθύρου μεταδεδομένων.
17.		Χρησιμοποιείται για την απεικόνιση κυριολεκτικής τιμής στο δένδρο ατόμων του παραθύρου μεταδεδομένων.
18.		Χρησιμοποιείται στο παράθυρο δημιουργίας έκφρασης σύνθετης κλάσης για να αναπαραστήσει το συμπλήρωμα κλάσης.
19.		Χρησιμοποιείται στο παράθυρο δημιουργίας έκφρασης σύνθετης κλάσης για να αναπαραστήσει την τομή κλάσεων.
20.		Χρησιμοποιείται στο παράθυρο δημιουργίας έκφρασης σύνθετης κλάσης για να αναπαραστήσει την ένωση κλάσεων.
21.		Χρησιμοποιείται στο παράθυρο δημιουργίας έκφρασης σύνθετης κλάσης για να αναπαραστήσει περιορισμό (ανώνυμη κλάση).
22.		Χρησιμοποιείται ως εικονίδιο της καρτέλας που περιλαμβάνει την όψη OWL μορφής της κάθε οντολογίας.

Παράρτημα 2: XML Schema στο οποίο υπακούν τα αρχεία προσαρμογής (configuration files)

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:element name="OntologyConfiguration">
    <xs:annotation>
      <xs:documentation>Ontology configuration</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="CoreOntologies"/>
        <xs:element ref="DomainOntologies"/>
        <xs:element ref="Templates"/>
        <xs:element ref="ExportedFormats"/>
      </xs:sequence>
      <xs:attribute name="application" type="xs:string" use="optional" default="MPEG-7/21
Description"/>
      <xs:attribute name="name" type="xs:string" use="required" form="unqualified"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="CoreOntologies">
    <xs:annotation>
      <xs:documentation>Files containing Core Ontologies</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="File" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="DomainOntologies">
    <xs:annotation>
      <xs:documentation>Files containing Domain Ontologies</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="File" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Templates">
    <xs:annotation>
      <xs:documentation>Files containing Templates</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="File" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="ExportedFormats">
```

```
<xs:annotation>
  <xs:documentation>Exported Formats Supported</xs:documentation>
</xs:annotation>
<xs:complexType>
  <xs:sequence>
    <xs:element name="File" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:attribute name="rulesfilename" type="xs:anyURI"/>
        <xs:attribute name="coreontologyURI" type="xs:anyURI"/>
        <xs:attribute name="formatname" type="xs:string"/>
        <xs:attribute name="formattype" type="xs:string" default="xsd"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="File">
  <xs:annotation>
    <xs:documentation>File describing the structure of a Metadata Category</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:attribute name="filename" type="xs:anyURI"/>
    <xs:attribute name="name" type="xs:string"/>
  </xs:complexType>
</xs:element>
</xs:schema>
```

Παράρτημα 3: Οι κλάσεις του GraphOnto API μαζί με τις μεθόδους τους

graphOnto

Class myOWLontology

java.lang.Object

└ graphOnto.myOWLontology

All Implemented Interfaces:

java.io.Serializable

public class **myOWLontology**

extends java.lang.Object

implements java.io.Serializable

See Also:

[Serialized Form](#)

Constructor Summary

[myOWLontology](#)(java.net.URI onturi)

Method Summary

void	addClass (myOWLclass my)
void	addComment (java.lang.Object an)
void	addDataProperty (myOWLdp mydp)
void	addImportedOntology (java.net.URI owluri)
void	addIndividual (myOWLindividual myind)
void	addLabel (java.lang.Object an)
void	addObjectProperty (myOWLop myop)
myOWLclass	getClassAt (int index)
java.util.Vector	getClasses ()
java.util.TreeSet	getClassesSorted ()
myOWLclass	getClassWithName (java.lang.String name)
java.lang.String	getCommentAt (int index)
java.util.Vector	getComments ()
java.util.Vector	getDataProperties ()
java.util.TreeSet	getDataPropertiesSorted ()
myOWLdp	getDataPropertyAt (int index)
myOWLdp	getDataPropertyWithName (java.lang.String name)

java.util.Vector	<u>getImportedOntologies</u> ()
java.net.URI	<u>getImportedOntologyAt</u> (int index)
<u>myOWLindividual</u>	<u>getIndividualAt</u> (int index)
java.util.Vector	<u>getIndividuals</u> ()
<u>myOWLindividual</u>	<u>getIndividualWithName</u> (java.lang.String name)
java.lang.String	<u>getLabelAt</u> (int index)
java.util.Vector	<u>getLabels</u> ()
java.lang.String	<u>getMPEG7</u> ()
java.util.Vector	<u>getObjectProperties</u> ()
java.util.TreeSet	<u>getObjectPropertiesSorted</u> ()
<u>myOWLop</u>	<u>getObjectPropertyAt</u> (int index)
<u>myOWLop</u>	<u>getObjectPropertyWithName</u> (java.lang.String name)
java.lang.String	<u>getOntologyName</u> ()
java.net.URI	<u>getOntologyURI</u> ()
java.util.Vector	<u>getRelationIndividuals</u> ()
java.util.Vector	<u>getSemanticBaseIndividuals</u> ()
java.util.Vector	<u>getTermUseIndividuals</u> ()
boolean	<u>isCore</u> ()
java.lang.String	<u>isDataOrObjectProperty</u> (java.lang.String name)
void	<u>removeClass</u> (<u>myOWLclass</u> my)
void	<u>removeComment</u> (java.lang.Object c)
void	<u>removeLabel</u> (java.lang.Object l)
void	<u>setCore</u> (boolean b)
void	<u>toMPEG7</u> (boolean toRepository)
java.lang.String	<u>toString</u> ()

graphOnto

Class myOWLclass

java.lang.Object

└ graphOnto.myOWLclass

All Implemented Interfaces:

java.io.Serializable

```
public class myOWLclass
```

```
extends java.lang.Object
```

```
implements java.io.Serializable
```

See Also:

[Serialized Form](#)

Constructor Summary

[myOWLclass](#)(java.net.URI u)

Method Summary

void	addComment (java.lang.Object an)
void	addConsistingClass (myOWLclass my)
void	addConsistingIndividual (myOWLindividual myind)
void	addConsistingRestriction (Restriction res)
void	addDataCardinalityRestriction (cardinalityRestriction cr)
void	addDataProperty (myOWLdp dp, myOWLclass sc)
void	addDataRestriction (dataRestriction res)
void	addDataRestrictionsOfSuperclass (myOWLclass supermy)
void	addDirectSubclass (myOWLclass submy)
void	addDisjointClass (myOWLclass dismy)
void	addEquivalentClass (myOWLclass equmy)
void	addIncludedClass (myOWLclass incmy)
void	addLabel (java.lang.Object an)
void	addObjectCardinalityRestriction (cardinalityRestriction cr)
void	addObjectProperty (myOWLop op, myOWLclass sc)
void	addObjectRestriction (objectRestriction res)
void	addObjectRestrictionsOfSuperclass (myOWLclass supermy)
void	addOwnedIndividual (myOWLindividual myind)
void	addPropertiesOfSuperclass (myOWLclass supermy)
void	addSuperClass (myOWLclass supermy)
boolean	containsDataProperty (myOWLdp mydp)
boolean	containsObjectProperty (myOWLop myop)
java.lang.String[]	dummygetObjectRestrictions ()
boolean	extendsRelationType ()
boolean	extendsSemanticBase ()
boolean	extendsTermUseType ()
java.util.Vector	getAllSubClasses ()
java.util.Vector	getAllSuperClasses ()

Παράρτημα 3: Οι κλάσεις του GraphOnto API μαζί με τις μεθόδους τους

java.lang.String	getClassType ()
myOWLclass	getClosestMPEG7Ancestor ()
java.lang.String[]	getComments ()
java.lang.String	getComplexType ()
myOWLclass	getConsistingClassAt (int index)
java.util.Vector	getConsistingClasses ()
myOWLindividual	getConsistingIndividualAt (int index)
java.util.Vector	getConsistingIndividuals ()
Restriction	getConsistingRestriction (int index)
java.util.Vector	getConsistingRestrictions ()
cardinalityRestriction	getDataCardinalityRestrictionOf (myOWLdp prop)
java.util.Vector	getDataCardinalityRestrictions ()
java.util.Vector	getDataProperties ()
java.util.TreeSet	getDataPropertiesSorted ()
myOWLdp	getDataPropertyAt (int index)
dataRestriction	getDataRestrictionOf (myOWLdp prop)
java.util.Vector	getDataRestrictions ()
myOWLclass	getDirectSubclassAt (int index)
java.util.Vector	getDirectSubclasses ()
myOWLclass	getDisjointClassAt (int index)
java.util.Vector	getDisjointClasses ()
myOWLclass	getEquivalentClassAt (int index)
java.util.Vector	getEquivalentClasses ()
myOWLclass	getEquivalentMPEG7Class ()
myOWLclass []	getFromWhomInherited (myOWLdp dp)
myOWLclass []	getFromWhomInherited (myOWLop op)
javax.swing.ImageIcon	getImage ()
java.util.Vector	getIncludedClasses ()
myOWLindividual	getIndividualWithName (java.lang.String name)
myOWLclass	getIntermediateToSuperclass (myOWLclass supermy)
java.lang.String[]	getLabels ()
int	getMaximumDataCardinality (myOWLdp dp)
int	getMaximumObjectCardinality (myOWLop op)
int	getMinimumDataCardinality (myOWLdp dp)

Παράρτημα 3: Οι κλάσεις του GraphOnto API μαζί με τις μεθόδους τους

int	<u>getMinimumObjectCardinality</u> (<u>myOWLOp</u> op)
java.lang.String	<u>getMPEG7source</u> ()
<u>myOWLontology</u>	<u>getMyOntology</u> ()
java.net.URI	<u>getName</u> ()
<u>cardinalityRestriction</u>	<u>getObjectCardinalityRestrictionOf</u> (<u>myOWLOp</u> prop)
java.util.Vector	<u>getObjectCardinalityRestrictions</u> ()
java.util.Vector	<u>getObjectProperties</u> ()
java.util.TreeSet	<u>getObjectPropertiesSorted</u> ()
<u>myOWLOp</u>	<u>getObjectPropertyAt</u> (int index)
<u>objectRestriction</u>	<u>getObjectRestrictionOf</u> (<u>myOWLOp</u> prop)
java.util.Vector	<u>getObjectRestrictions</u> ()
java.lang.String	<u>getOntology</u> ()
java.lang.String	<u>getOWLSource</u> ()
<u>myOWLindividual</u>	<u>getOwnedIndividualAt</u> (int index)
java.util.Vector	<u>getOwnedIndividuals</u> ()
java.util.Vector	<u>getReserve</u> ()
java.lang.String	<u>getSmallName</u> ()
int	<u>getSubClassDepth</u> (<u>myOWLclass</u> supermy)
java.util.Vector	<u>getSubclassesDepth</u> ()
<u>myOWLclass</u>	<u>getSuperClassAt</u> (int index)
int	<u>getSuperClassDepth</u> (<u>myOWLclass</u> supermy)
java.util.Vector	<u>getSuperClasses</u> ()
java.util.Vector	<u>getSuperclassesDepth</u> ()
boolean	<u>hasExactyOneSuperClass</u> ()
boolean	<u>hasSuperClass</u> ()
boolean	<u>isComplex</u> ()
boolean	<u>isMyDataProperty</u> (<u>myOWLdp</u> dp)
boolean	<u>isMyObjectProperty</u> (<u>myOWLOp</u> op)
boolean	<u>isNotParsed</u> ()
boolean	<u>isRootClass</u> ()
java.lang.String[][]	<u>joinRules</u> ()
java.lang.String[][]	<u>joinRulesExcluded</u> (<u>myOWLclass</u> superclass)
void	<u>removeAllConsistingClasses</u> ()
void	<u>removeAllConsistingIndividuals</u> ()

void	<u>removeAllConstistingClasses</u> ()
void	<u>removeAllConstistingIndividuals</u> ()
void	<u>removeAllConstistingRestrictions</u> ()
void	<u>removeComment</u> (java.lang.Object c)
void	<u>removeConsistingClass</u> (myOWLclass my)
void	<u>removeConsistingIndividual</u> (myOWLindividual myind)
void	<u>removeDataCardinalityRestrictionOf</u> (myOWLdp prop)
void	<u>removeDataPropertiesofOldSuperclasses</u> ()
void	<u>removeDataProperty</u> (myOWLdp mydp)
void	<u>removeDataRestrictionOf</u> (myOWLdp prop)
void	<u>removeDirectSubclass</u> (myOWLclass submy)
void	<u>removeDisjointClass</u> (myOWLclass my)
void	<u>removeEquivalentClass</u> (myOWLclass equmy)
void	<u>removeImage</u> ()
void	<u>removeLabel</u> (java.lang.Object l)
void	<u>removeObjectCardinalityRestrictionOf</u> (myOWLop prop)
void	<u>removeObjectPropertiesofOldSuperclasses</u> ()
void	<u>removeObjectProperty</u> (myOWLop myop)
void	<u>removeObjectRestriction</u> (objectRestriction res)
void	<u>removeObjectRestrictionOf</u> (myOWLop prop)
void	<u>removeOwnedIndividual</u> (myOWLindividual myind)
void	<u>removeSuperClass</u> (myOWLclass supermy)
void	<u>setClassType</u> (java.lang.String t)
void	<u>setComplex</u> (boolean com)
void	<u>setComplexType</u> (java.lang.String t)
void	<u>setName</u> (java.net.URI u)
void	<u>setNotParsed</u> (boolean b)
void	<u>setOntology</u> (java.lang.String s)
java.lang.String	<u>toMPEG7</u> ()
void	<u>toOWL</u> (java.util.Hashtable URItName)
java.lang.String	<u>toString</u> ()
java.lang.String	<u>writeAllOfMPEG7Property</u> ()

graphOnto

Class myOWLdp

java.lang.Object

└ graphOnto.myOWLdp

All Implemented Interfaces:

java.io.Serializable

public class **myOWLdp**

extends java.lang.Object

implements java.io.Serializable

See Also:

[Serialized Form](#)

Constructor Summary

[myOWLdp](#)(java.net.URI arg)

Method Summary

void	addComment (java.lang.String an)
void	addDomain (myOWLclass my)
void	addEquivalentProperty (myOWLdp ep)
void	addLabel (java.lang.String an)
void	addRange (java.net.URI s)
void	addRangeEnumeration (java.lang.String de)
void	addRangeEnumeration (java.lang.String[] array)
void	addSubProperty (myOWLdp sp)
void	addSuperDomain (myOWLclass my)
void	addSuperProperty (myOWLdp superp)
boolean	containsDomain (myOWLclass my)
boolean	containsRange (java.net.URI u)
boolean	containsRangeEnumeration (java.lang.String[] arg)
java.util.Vector	getAllSubProperties ()
java.util.Vector	getAllSuperProperties ()
java.lang.String[]	getComments ()
myOWLclass	getDomainAt (int index)
java.util.Vector	getDomains ()
java.util.Vector	getEquivalentProperties ()
myOWLdp	getEquivalentPropertyAt (int index)
java.lang.String[]	getLabels ()

myOWLdp	getMPEG7SuperProperty ()
myOWLontology	getMyOntology ()
java.net.URI	getName ()
java.lang.String	getOntology ()
java.lang.String	getOWLSource ()
java.lang.String[][]	getRangeEnumeration ()
java.net.URI[]	getRanges ()
java.lang.String	getSmallName ()
java.util.Vector	getSubProperties ()
myOWLdp	getSubPropertyAt (int index)
java.util.Vector	getSubpropertyDepth ()
myOWLclass	getSuperDomainAt (int index)
java.util.Vector	getSuperDomains ()
java.util.Vector	getSuperProperties ()
myOWLdp	getSuperPropertyAt (int index)
java.util.Vector	getSuperpropertyDepth ()
boolean	hasEquivalentProperty (myOWLdp dp)
boolean	hasRangeEnumeration ()
boolean	hasSubProperty (myOWLdp sp)
boolean	hasSuperProperty (myOWLdp sp)
boolean	isFunctiona ()
boolean	isNotParsed ()
boolean	isSubPropertyOfMPEG7 ()
void	removeAllComments ()
void	removeAllEquivalentProperties ()
void	removeAllLabels ()
void	removeAllRangeEnumeration ()
void	removeAllRanges ()
void	removeAllSubProperties ()
void	removeAllSuperDomains ()
void	removeAllSuperProperties ()
void	removeComment (java.lang.String l)
void	removeDomain (myOWLclass my)
void	removeEquivalentProperty (myOWLdp ep)

void	removeLabel (java.lang.String l)
void	removePropertyRelations ()
void	removeRange (java.net.URI r)
void	removeRangeEnumeration (java.lang.String[] array)
void	removeSubProperty (myOWLdp sp)
void	removeSuperDomain (myOWLclass my)
void	removeSuperProperty (myOWLdp superp)
void	setFunctional (boolean b)
void	setName (java.net.URI r)
void	setNotParsed (boolean b)
void	setOntology (java.lang.String o)
void	setRangeEnumeration (boolean b)
void	toOWL (java.util.Hashtable URIToName)
java.lang.String	toString ()

graphOnto

Class myOWLop

java.lang.Object

└ graphOnto.myOWLop

All Implemented Interfaces:

java.io.Serializable

public class **myOWLop**

extends java.lang.Object

implements java.io.Serializable

See Also:

[Serialized Form](#)

Constructor Summary

[myOWLop](#)(java.net.URI arg)

Method Summary

void	addComment (java.lang.String an)
void	addDomain (myOWLclass my)
void	addEquivalentProperty (myOWLop ep)
void	addIndividual (myOWLindividual myind)
void	addLabel (java.lang.String an)
void	addRange (myOWLclass my)

Παράρτημα 3: Οι κλάσεις του GraphOnto API μαζί με τις μεθόδους τους

void	<u>addSubProperty</u> (<u>myOWLop</u> sp)
void	<u>addSuperDomain</u> (<u>myOWLclass</u> my)
void	<u>addSuperProperty</u> (<u>myOWLop</u> superp)
boolean	<u>containsDomain</u> (<u>myOWLclass</u> my)
boolean	<u>containsIndividual</u> (<u>myOWLindividual</u> myind)
boolean	<u>containsRange</u> (<u>myOWLclass</u> my)
java.util.Vector	<u>getAllSubProperties</u> ()
java.util.Vector	<u>getAllSuperProperties</u> ()
java.lang.String[]	<u>getComments</u> ()
<u>myOWLclass</u>	<u>getDomainAt</u> (int index)
java.util.Vector	<u>getDomains</u> ()
java.util.Vector	<u>getEquivalentProperties</u> ()
<u>myOWLop</u>	<u>getEquivalentPropertyAt</u> (int index)
<u>myOWLindividual</u>	<u>getIndividualAt</u> (int index)
java.util.Vector	<u>getIndividuals</u> ()
boolean	<u>getIndividualsMode</u> ()
<u>myOWLop</u>	<u>getInverseProperty</u> ()
java.lang.String[]	<u>getLabels</u> ()
<u>myOWLop</u>	<u>getMPEG7SuperProperty</u> ()
<u>myOWLontology</u>	<u>getMyOntology</u> ()
java.net.URI	<u>getName</u> ()
java.lang.String	<u>getOntology</u> ()
java.lang.String	<u>getOWLSource</u> ()
java.util.Vector	<u>getPossibleClassesForRestriction</u> ()
java.util.Vector	<u>getPossibleIndividualsForRestriction</u> ()
<u>myOWLclass</u>	<u>getRangeAt</u> (int index)
java.util.Vector	<u>getRanges</u> ()
java.lang.String	<u>getSmallName</u> ()
java.util.Vector	<u>getSubProperties</u> ()
<u>myOWLop</u>	<u>getSubPropertyAt</u> (int index)
java.util.Vector	<u>getSubpropertyDepth</u> ()
<u>myOWLclass</u>	<u>getSuperDomainAt</u> (int index)
java.util.Vector	<u>getSuperDomains</u> ()
java.util.Vector	<u>getSuperProperties</u> ()

myOWLop	getSuperPropertyAt (int index)
java.util.Vector	getSuperpropertyDepth ()
boolean	hasEquivalentProperty (myOWLop sp)
boolean	hasSubProperty (myOWLop sp)
boolean	hasSuperProperty (myOWLop sp)
boolean	isFunctiona ()
boolean	isInverseFunctiona ()
boolean	isNotParsed ()
boolean	isSubPropertyOfMPEG7 ()
boolean	isSymmetric ()
boolean	isTransitive ()
void	removeAllComments ()
void	removeAllEquivalentProperties ()
void	removeAllIndividuals ()
void	removeAllLabels ()
void	removeAllRanges ()
void	removeAllSubProperties ()
void	removeAllSuperDomains ()
void	removeAllSuperProperties ()
void	removeComment (java.lang.String l)
void	removeDomain (myOWLclass my)
void	removeEquivalentProperty (myOWLop u)
void	removeIndividual (myOWLindividual myind)
void	removeLabel (java.lang.String l)
void	removePropertyRelations ()
void	removeRange (myOWLclass my)
void	removeSubProperty (myOWLop sp)
void	removeSuperDomain (myOWLclass my)
void	removeSuperProperty (myOWLop superp)
void	setFunctiona (boolean b)
void	setIndividualsMode (boolean b)
void	setInverseFunctiona (boolean b)
void	setInverseProperty (myOWLop op)
void	setName (java.net.URI r)

void	setNotParsed (boolean b)
void	setOntology (java.lang.String o)
void	setSymmetric (boolean b)
void	setTransitive (boolean b)
void	toOWL (java.util.Hashtable URIToName)
java.lang.String	toString ()

graphOnto

Class myOWLIndividual

java.lang.Object

└ graphOnto.myOWLIndividual

All Implemented Interfaces:

java.io.Serializable

public class **myOWLIndividual**

extends java.lang.Object

implements java.io.Serializable

See Also:

[Serialized Form](#)

Constructor Summary

[myOWLIndividual](#)(java.net.URI u)

Method Summary

void	addComment (java.lang.Object an)
void	addContainedDataProperty (myOWLdp property, java.lang.String range)
void	addContainedObjectProperty (myOWLop property, myOWLIndividual range)
void	addDifferentFromIndividual (myOWLIndividual myind)
void	addLabel (java.lang.Object an)
void	addSameAsIndividual (myOWLIndividual myind)
void	addType (myOWLclass my)
java.lang.String	checkDisjointConsistency (myOWLclass my)
java.lang.String[]	getComments ()
java.util.Hashtable	getContainedDataProperties ()
java.util.Hashtable	getContainedObjectProperties ()
java.lang.String[]	getContainedValuesOfDataProperty (myOWLdp property)
java.util.Vector	getContainedValuesOfObjectProperty (myOWLop property)

myOWLindividual	getDifferentFromIndividualAt (int index)
java.util.Vector	getDifferentFromIndividuals ()
java.lang.String[]	getLabels ()
myOWLontology	getMyOntology ()
java.net.URI	getName ()
java.lang.String	getOntology ()
java.lang.String	getOWLSource ()
myOWLindividual	getSameAsIndividualAt (int index)
java.util.Vector	getSameAsIndividuals ()
java.lang.String	getSmallName ()
myOWLclass	getTypeAt (int index)
java.util.Vector	getTypes ()
boolean	hasOneType ()
boolean	hasTypeSameWithRangeOf (myOWLop myop)
boolean	isNotParsed ()
boolean	isRoot ()
boolean	isThing ()
void	removeAllValuesOfDataProperty (myOWLdp mydp)
void	removeAllValuesOfObjectProperty (myOWLop myop)
void	removeComment (java.lang.String l)
void	removeContainedDataProperty (myOWLdp property, java.lang.String range)
void	removeContainedObjectProperty (myOWLop property, myOWLindividual range)
void	removeDifferentFromIndividual (myOWLindividual myind)
void	removeLabel (java.lang.String l)
void	removeSameAsIndividual (myOWLindividual myind)
void	removeType (myOWLclass my)
void	setName (java.net.URI u)
void	setNotParsed (boolean b)
void	setRoot (boolean b)
void	setThing (boolean b)
java.lang.String	toMPEG7AsIncluded ()
java.lang.String	toMPEG7AsRelation ()
java.lang.String	toMPEG7AsSemanticBase (boolean isMetadata)

java.lang.String	toMPEG7AsTermUse()
void	toOWL (java.util.Hashtable URIToName)
java.lang.String	toString()
boolean	typeExtendsRelation()
boolean	typeExtendsSemanticBase()
boolean	typeExtendsTermUse()
boolean	typeIsRelation()
boolean	typeIsSemanticBase()
boolean	typeIsTermUse()
void	writeAnnotations()
void	writeContainedDataProperties (java.util.Hashtable URIToName)
void	writeContainedObjectProperties (java.util.Hashtable URIToName)
void	writeRelations (java.util.Hashtable URIToName)

graphOnto

Class Restriction

java.lang.Object

└ [graphOnto.Restriction](#)

Direct Known Subclasses:

[cardinalityRestriction](#), [dataRestriction](#), [objectRestriction](#)

public abstract class **Restriction**

extends java.lang.Object

Constructor Summary

[Restriction](#)()

Method Summary

abstract
java.lang.String [getType](#)()

graphOnto

Class cardinalityRestriction

java.lang.Object

└ [graphOnto.Restriction](#)

└ [graphOnto.cardinalityRestriction](#)

All Implemented Interfaces:

java.io.Serializable

public class **cardinalityRestriction**

extends [Restriction](#)

implements java.io.Serializable

See Also:

[Serialized Form](#)

Constructor Summary

<u>cardinalityRestriction</u> ()
<u>cardinalityRestriction</u> (<u>myOWLclass</u> my, <u>myOWLdp</u> mydp)
<u>cardinalityRestriction</u> (<u>myOWLclass</u> my, <u>myOWLop</u> myop)

Method Summary

<u>myOWLdp</u>	<u>getDataProperty</u> ()
<u>myOWLclass</u>	<u>getDomainClass</u> ()
int	<u>getMaximumConstraint</u> ()
int	<u>getMinimumConstraint</u> ()
<u>myOWLop</u>	<u>getObjectProperty</u> ()
java.lang.String	<u>getType</u> ()
boolean	<u>refersToData</u> ()
void	<u>setData</u> (boolean d)
void	<u>setDataProperty</u> (<u>myOWLdp</u> mydp)
void	<u>setDomainClass</u> (<u>myOWLclass</u> my)
void	<u>setMaximumConstraint</u> (int max)
void	<u>setMinimumConstraint</u> (int min)
void	<u>setObjectProperty</u> (<u>myOWLop</u> myop)
java.lang.String[]	<u>toOWL</u> ()
java.lang.String	<u>toString</u> ()

graphOnto

Class dataRestriction

java.lang.Object

└ [graphOnto.Restriction](#)

└ **graphOnto.dataRestriction**

All Implemented Interfaces:

java.io.Serializable

public class **dataRestriction**

extends [Restriction](#)

implements java.io.Serializable

See Also:

[Serialized Form](#)

Constructor Summary

[dataRestriction](#)()

[dataRestriction](#)([myOWLclass](#) my, [myOWLdp](#) p)

Method Summary

myOWLclass	getDomainClass ()
java.lang.String	getKind ()
java.lang.Object[]	getPermittedValues ()
myOWLdp	getProperty ()
java.lang.String	getType ()
void	setDomainClass (myOWLclass my)
void	setPermittedValues (Object[] pv)
void	setKind (java.lang.String k)
void	setProperty (myOWLdp p)
java.lang.String	toString ()

graphOnto

Class **objectRestriction**

java.lang.Object

└ [graphOnto.Restriction](#)

└ **graphOnto.objectRestriction**

All Implemented Interfaces:

java.io.Serializable

public class **objectRestriction**

extends [Restriction](#)

implements java.io.Serializable

See Also:

[Serialized Form](#)

Constructor Summary

[objectRestriction](#)()

[objectRestriction](#)([myOWLclass](#) my, [myOWLop](#) myop, java.lang.String k)

Method Summary

void	addEquivalentRestrictions (objectRestriction r)
void	addPermittedClassValue (myOWLclass my)

void	addPermittedIndividualValue (myOWLindividual myind)
boolean	equals (objectRestriction r)
myOWLclass	getDomainClass ()
java.util.Vector	getEquivalentRestrictions ()
java.lang.String	getKind ()
myOWLclass	getPermittedClassValueAt (int index)
java.util.Vector	getPermittedClassValues ()
myOWLindividual	getPermittedIndividualValueAt (int index)
java.util.Vector	getPermittedIndividualValues ()
myOWLop	getProperty ()
java.lang.String	getType ()
boolean	hasIndividualValues ()
void	removePermittedClassValue (myOWLclass my)
void	removePermittedIndividualValue (myOWLindividual myind)
void	setDomainClass (myOWLclass my)
void	setIndividualValues (boolean ind)
void	setKind (java.lang.String k)
void	setProperty (myOWLop myop)
java.lang.String	toString ()

graphOnto

Class equivalentClassesSet

java.lang.Object

└ graphOnto.equivalentClassesSet

public class **equivalentClassesSet**

extends java.lang.Object

Constructor Summary

[equivalentClassesSet](#) ([myOWLclass](#) my)

Method Summary

void	addMember (myOWLclass my)
myOWLclass	getMemberAt (int index)
java.util.Vector	getMembers ()
void	mergeSets (equivalentClassesSet otherSet)

void	<u>removeMember</u> (<u>myOWLclass</u> my)
------	---

graphOnto

Class equivalentDataPropertiesSet

java.lang.Object

└ graphOnto.equivalentDataPropertiesSet

public class **equivalentDataPropertiesSet**

extends java.lang.Object

Constructor Summary

[equivalentDataPropertiesSet](#)([myOWLdp](#) mydp)

Method Summary

void	<u>addMember</u> (<u>myOWLdp</u> mydp)
<u>myOWLdp</u>	<u>getMemberAt</u> (int index)
java.util.Vector	<u>getMembers</u> ()
void	<u>mergeSets</u> (<u>equivalentDataPropertiesSet</u> otherSet)
void	<u>removeMember</u> (<u>myOWLdp</u> mydp)

graphOnto

Class equivalentObjectPropertiesSet

java.lang.Object

└ graphOnto.equivalentObjectPropertiesSet

public class **equivalentObjectPropertiesSet**

extends java.lang.Object

Constructor Summary

[equivalentObjectPropertiesSet](#)([myOWLop](#) myop)

Method Summary

void	<u>addMember</u> (<u>myOWLop</u> myop)
<u>myOWLop</u>	<u>getMemberAt</u> (int index)
java.util.Vector	<u>getMembers</u> ()
void	<u>mergeSets</u> (<u>equivalentObjectPropertiesSet</u> otherSet)
void	<u>removeMember</u> (<u>myOWLop</u> myop)

graphOnto

Class sameAsIndividualsSet

java.lang.Object

└ graphOnto.sameAsIndividualsSet

public class **sameAsIndividualsSet**

extends java.lang.Object

Constructor Summary

<u>sameAsIndividualsSet</u> (<u>myOWLindividual</u> myind)

Method Summary

void	<u>addMember</u> (<u>myOWLindividual</u> myind)
<u>myOWLindividual</u>	<u>getMemberAt</u> (int index)
java.util.Vector	<u>getMembers</u> ()
void	<u>mergeSets</u> (<u>sameAsIndividualsSet</u> otherSet)
void	<u>removeMember</u> (<u>myOWLindividual</u> myind)