



ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

**ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΣΤΗΝ ΕΠΙΣΤΗΜΗ ΤΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**

**Διπλωματική Εργασία
Μεταπτυχιακού Διπλώματος Ειδικευσης**

**«Ανάπτυξη μηχανής παραγωγής φυσικής γλώσσας για
οντολογίες OWL»**

**Γαλάνης Δημήτριος
Επιβλέπων: Ίων Ανδρουτσόπουλος**

ΑΘΗΝΑ, ΙΟΥΝΙΟΣ 2006

ΠΕΡΙΕΧΟΜΕΝΑ

1.	Εισαγωγή.....	1
1.1	Αντικείμενο της εργασίας.....	1
1.2	Διάρθρωση της εργασίας.....	2
1.3	Ευχαριστίες.....	3
2.	Παραγωγή φυσικής γλώσσας.....	4
2.1	Εισαγωγή.....	4
2.2	Συνήθης αρχιτεκτονική των συστημάτων παραγωγής φυσικής γλώσσας	6
2.2.1	Σχεδιασμός εγγράφου.....	7
2.2.1.1	Καθορισμός περιεχομένου (content determination).....	7
2.2.1.2	Καθορισμός δομής (document structuring).....	8
2.2.2	Μικροσχεδιασμός εγγράφου.....	8
2.2.2.1	Παραγωγή περιγραφών προτάσεων (lexicalisation).....	8
2.2.2.2	Ομαδοποίηση (Aggregation).....	9
2.2.2.3	Παραγωγή αναφορικών εκφράσεων.....	10
2.2.3	Παραγωγή επιφανειακής μορφής.....	12
3.	Οντολογίες.....	14
3.1	Τι είναι ο σημασιολογικός ιστός.....	14
3.2	Τι είναι μια οντολογία και πώς χρησιμοποιείται στο σημασιολογικό ιστό.....	14
3.3	Η γλώσσα OWL και γιατί επιλέχθηκε.....	15
3.4	OWL Lite.....	17
3.4.1	Βασικοί μηχανισμοί της OWL Lite.....	17
3.4.2	Ισοδυναμίες, ισότητες και ανισότητες στην OWL Lite.....	20
3.4.3	OWL Lite και χαρακτηριστικά ιδιοτήτων.....	22
3.4.4	Περιορισμοί ιδιοτήτων της OWL Lite.....	24
3.4.5	Περιορισμοί πληθαρίσμου στην OWL Lite.....	26
3.4.7	Τύποι δεδομένων της OWL Lite.....	29
3.4.8	Εισαγωγή οντολογιών στην OWL Lite.....	30
3.4.9	OWL Lite και ιδιότητες σχολιασμού.....	31
3.5	Οι OWL DL και OWL Full.....	32
4.	Το σύστημα ΠΦΓ της εργασίας.....	37
4.1	Εισαγωγή.....	37
4.2	Επιλογή περιεχομένου.....	38

4.3	Καθορισμός δομής	42
4.4	Παραγωγή περιγραφών προτάσεων	44
4.4.1	Σχεδιάγραμμα.....	44
4.4.2	Κωδικοποίηση των σχεδίων με την βοήθεια της γλώσσας XSL.	47
4.4.3	Σχεδιάγραμμα για τη μετατροπή σε φυσική γλώσσα μηνυμάτων που περιέχουν περιορισμούς ιδιοτήτων.....	51
4.4.4	Λεξικό	54
4.4.5	Ομαδοποίηση	57
4.5.1	1ος Κανόνας ομαδοποίησης: «Περιγραφή τάξης και ιδιότητας».....	57
4.5.2	2ος Κανόνας ομαδοποίησης: "Απλή σύζευξη".	58
4.5.3	3ος Κανόνας ομαδοποίησης: "Κοινό υποκείμενο και ρήμα"	59
4.5.4	4ος Κανόνας ομαδοποίησης: "Ιδιότητα και περιορισμός πληθαιθμού"	60
4.5.5	5ος Κανόνας ομαδοποίησης: "Απαλοιφή πλεονασμού"	61
4.6	Παραγωγή αναφορικών εκφράσεων.....	61
4.7	Παραγωγή επιφανειακής μορφής.....	63
4.8	Εργαλείο συγγραφής	63
4.8.1	Ο ρόλος του εργαλείου συγγραφής στο σύστημα ΠΦΓ της εργασίας.....	63
4.8.2	LexiconBrowserTab	65
4.8.3	TemplatesAndOrderingTab	67
5	Συμπεράσματα και προτάσεις για μελλοντικές εργασίες.....	69
	Αναφορές.....	71

Περίληψη

Στη διάρκεια αυτής της εργασίας δημιουργήθηκε ένα σύστημα παραγωγής φυσικής γλώσσας, το οποίο παράγει κείμενα στα ελληνικά και στα αγγλικά από συμβολικές πληροφορίες αποθηκευμένες σε οντολογίες OWL. Το σύστημα υιοθετεί πολλές από τις ιδέες του αντίστοιχου λογισμικού του ερευνητικού έργου IST M-PIRO. Παράγει απλούστερα κείμενα, αλλά αντίθετα από το λογισμικό του M-PIRO υποστηρίζει πλήρως την OWL. Στη διάρκεια της εργασίας αναπτύχθηκε, επίσης, ένα «εργαλείο συγγραφής», αντίστοιχης λειτουργικότητας με εκείνης του M-PIRO, το οποίο βοηθά τους χρήστες που δεν είναι ειδικοί στην επεξεργασία φυσικής γλώσσας να προσαρμόζουν εύκολα το σύστημα σε νέες οντολογίες, δημιουργώντας και τους απαραίτητους γλωσσικούς πόρους, μέσω μιας γραφικής διεπαφής. Το νέο εργαλείο συγγραφής υλοποιήθηκε ως επέκταση του συστήματος Protégé, ενός δημοφιλούς εργαλείου που επιτρέπει την εύκολη δημιουργία και τροποποίηση οντολογιών. Το λογισμικό της εργασίας δίνει τη δυνατότητα στους χρήστες του Protégé να παράγουν περιγραφές φυσικής γλώσσας των οντοτήτων και των τύπων οντοτήτων των οντολογιών OWL, κάτι που είναι συχνά χρήσιμο κατά τον έλεγχο των οντολογιών. Μπορεί, επίσης, να χρησιμοποιηθεί σε εξυπηρετητές του σημασιολογικού ιστού (π.χ. καταστημάτων, μουσείων κλπ.), οι οποίοι θα δημοσιεύουν το περιεχόμενό τους τόσο σε μορφή οντολογιών OWL (μορφή κατάλληλη π.χ. για τεχνητούς πράκτορες) όσο και ως εξατομικευμένα κείμενα φυσικής γλώσσας που θα παράγονται αυτόματα σε πολλές γλώσσες από τις οντολογίες OWL.

1. Εισαγωγή

1.1 Αντικείμενο της εργασίας

Στα πλαίσια του ερευνητικού έργου M-PIRO [1], αναπτύχθηκε ένα σύστημα παραγωγής φυσικής γλώσσας, το οποίο παράγει αγγλικές, ιταλικές και ελληνικές περιγραφές των αντικειμένων μιας συλλογής από συμβολικές πληροφορίες αποθηκευμένες σε οντολογίες και βάσεις δεδομένων. Το σύστημα έχει δοκιμαστεί με συλλογές εκθεμάτων μουσείων, ηλεκτρονικών ειδών προς πώληση, φαρμάκων κλπ. Παρέχει μεταξύ άλλων ένα εργαλείο συγγραφής [22], το οποίο βοηθά τους χρήστες που δεν είναι ειδικοί στην επεξεργασία φυσικής γλώσσας να προσαρμόζουν εύκολα το σύστημα σε καινούργιες συλλογές αντικειμένων μέσω μιας γραφικής διεπαφής. Στη διάρκεια προηγούμενης διπλωματικής εργασίας το εργαλείο συγγραφής επεκτάθηκε, έτσι ώστε να υποστηρίζει οντολογίες διατυπωμένες στη γλώσσα OWL του σημασιολογικού ιστού [7, 23, 47]. Όμως, η μηχανή παραγωγής φυσικής γλώσσας του συστήματος, δηλαδή η μονάδα που αναλαμβάνει να μετατρέψει τις συμβολικές πληροφορίες της οντολογίας σε φυσική γλώσσα, δεν υποστηρίζει την OWL. Για να αντιμετωπιστεί αυτό το πρόβλημα, η μορφή του εργαλείου συγγραφής της προηγούμενης διπλωματικής εργασίας μεταφράζει τις οντολογίες OWL στη γλώσσα αναπαράστασης οντολογιών της μηχανής παραγωγής, η οποία βασίζεται στο οντολογικό μοντέλο του M-PIRO. Το τελευταίο, όμως, έχει πολλές ασυμβατότητες με το οντολογικό μοντέλο της OWL, με αποτέλεσμα η μετάφραση να είναι ατελής και να χάνονται πολλές πληροφορίες των οντολογιών OWL.

Στα πλαίσια της παρούσας εργασίας δημιουργήθηκε μια νέα, απλούστερη μηχανή παραγωγής φυσικής γλώσσας, που υποστηρίζει εγγενώς την OWL. Επίσης, δημιουργήθηκε ένα νέο εργαλείο συγγραφής, το οποίο προστέθηκε ως επέκταση στο λογισμικό του Protégé [53]. Το Protégé υποστηρίζει τη δημιουργία και επεξεργασία οντολογιών (σε OWL και άλλες γλώσσες

αναπαράστασης) και αποτελεί ένα από τα πιο δημοφιλή εργαλεία που σχετίζονται με το σημασιολογικό ιστό. Το λογισμικό της εργασίας δίνει τη δυνατότητα στους χρήστες του Protégé να παράγουν περιγραφές φυσικής γλώσσας των οντοτήτων και των τύπων οντοτήτων των οντολογιών OWL, κάτι που είναι συχνά χρήσιμο κατά τον έλεγχο των οντολογιών. Το λογισμικό της εργασίας μπορεί, επίσης, να χρησιμοποιηθεί σε εξυπηρετητές του σημασιολογικού ιστού (π.χ. καταστημάτων, μουσείων κλπ.), οι οποίοι θα δημοσιεύουν το περιεχόμενό τους τόσο σε μορφή οντολογιών OWL (μορφή κατάλληλη π.χ. για τεχνητούς πράκτορες) όσο και ως εξατομικευμένα κείμενα φυσικής γλώσσας που θα παράγονται αυτόματα σε πολλές γλώσσες από τις οντολογίες OWL.

1.2 Διάρθρωση της εργασίας

Στη συνέχεια της εργασίας θα δούμε τα εξής:

Στο κεφάλαιο 2 παρουσιάζεται η συνήθης αρχιτεκτονική των συστημάτων παραγωγής φυσικής γλώσσας. Συγκριμένα, περιγράφονται οι μονάδες από τις οποίες αποτελείται ένα τέτοιο σύστημα, η είσοδος του συστήματος και η διαδικασία που ακολουθείται μέχρι την παραγωγή του τελικού κειμένου.

Στο κεφάλαιο 3 ορίζεται η έννοια της οντολογίας και πώς αυτή χρησιμοποιείται στο σημασιολογικό ιστό. Στη συνέχεια του κεφαλαίου δίνεται μια συνοπτική περιγραφή της γλώσσας οντολογιών OWL.

Στο κεφάλαιο 4 περιγράφεται αναλυτικά η αρχιτεκτονική και ο τρόπος λειτουργίας της μηχανής παραγωγής φυσικής γλώσσας που αναπτύχθηκε, καθώς και ο ρόλος του εργαλείου συγγραφής που δημιουργήθηκε.

Τέλος, στο κεφάλαιο 5 γίνεται μια σύνοψη της εργασίας και παρουσιάζονται κάποιες προτάσεις για επεκτάσεις του συστήματος που αναπτύχθηκε.

1.3 Ευχαριστίες

Ευχαριστώ τον επιβλέποντα καθηγητή μου κ. Ίωνα Ανδρουτσόπουλο για τη βοήθεια και καθοδήγηση που μου προσέφερε κατά τη διάρκεια της εκπόνησης της εργασίας. Επίσης, θα ήθελα να ευχαριστήσω την αδερφή μου Μαρία Γαλάνη για τις χρήσιμες παρατηρήσεις της όσον αφορά τη μορφή του τελικού κειμένου και τη Χριστίνα Παπαδημητρίου, η οποία μου δάνεισε το φορητό της υπολογιστή για αρκετά μεγάλα διαστήματα. Τέλος, ευχαριστώ τον κ. Παναγιώτη Κωνσταντόπουλο, που αποδέχτηκε τον ρόλο του δεύτερου αξιολογητή.

2. Παραγωγή φυσικής γλώσσας

2.1 Εισαγωγή

Η παραγωγή φυσικής γλώσσας (Natural Language Generation) είναι ένας τομέας της Τεχνητής Νοημοσύνης και της Υπολογιστικής Γλωσσολογίας. Επικεντρώνεται στη δημιουργία υπολογιστικών συστημάτων που μπορούν να παράγουν κατανοητά κείμενα στην αγγλική ή άλλες φυσικές γλώσσες έχοντας ως είσοδο μη γλωσσικές αναπαραστάσεις γνώσεων [29].

Η παραγωγή φυσικής γλώσσας (ΠΦΓ) είναι ένας ερευνητικός τομέας με πολλές πρακτικές εφαρμογές. Τα συστήματα ΠΦΓ μπορούν να αυτοματοποιήσουν σε μεγάλο βαθμό τις διαδικασίες παραγωγής κειμένων ρουτίνας, όπως για παράδειγμα τα μετεωρολογικά δελτία και τα εγχειρίδια χρήσεως συσκευών ή λογισμικού [4, 8]. Η ΠΦΓ είναι επίσης πολύ ενδιαφέρουσα ως ερευνητικός τομέας, καθώς εγείρει πολλά ερωτήματα που αφορούν α) το πώς πρέπει να αναπαριστώνται οι γλωσσικές πληροφορίες (π.χ. ορολογία) και οι συμβολικές πληροφορίες (π.χ. τύποι και σχέσεις οντοτήτων) ενός γνωστικού πεδίου (domain) ώστε να είναι δυνατή η αυτόματη παραγωγή κειμένων από αυτές, β) το πότε ένα κείμενο είναι καλογραμμένο και μπορεί εύκολα να γίνει κατανοητό από τους ανθρώπους και γ) το πώς ένας υπολογιστής μπορεί να παραγάγει καλογραμμένα και κατανοητά κείμενα.

Η είσοδος την οποία δέχεται ένα σύστημα ΠΦΓ μπορεί να αναπαρασταθεί ως ένα διάνυσμα πέντε στοιχείων $\langle k, l, t, u, h \rangle$, όπου k είναι η πηγή γνώσης που θα χρησιμοποιηθεί, l είναι οι διαθέσιμοι γλωσσικοί πόροι, t είναι ο επικοινωνιακός στόχος που πρέπει να επιτευχθεί, u είναι το μοντέλο του χρήστη και h το ιστορικό αλληλεπίδρασης. Στη συνέχεια αυτά τα πέντε στοιχεία περιγράφονται αναλυτικότερα.

- Πηγή γνώσης: Είναι οι πληροφορίες για κάποιο γνωστικό πεδίο (domain) και βρίσκονται αποθηκευμένες σε βάσεις δεδομένων ή/και οντολογίες.

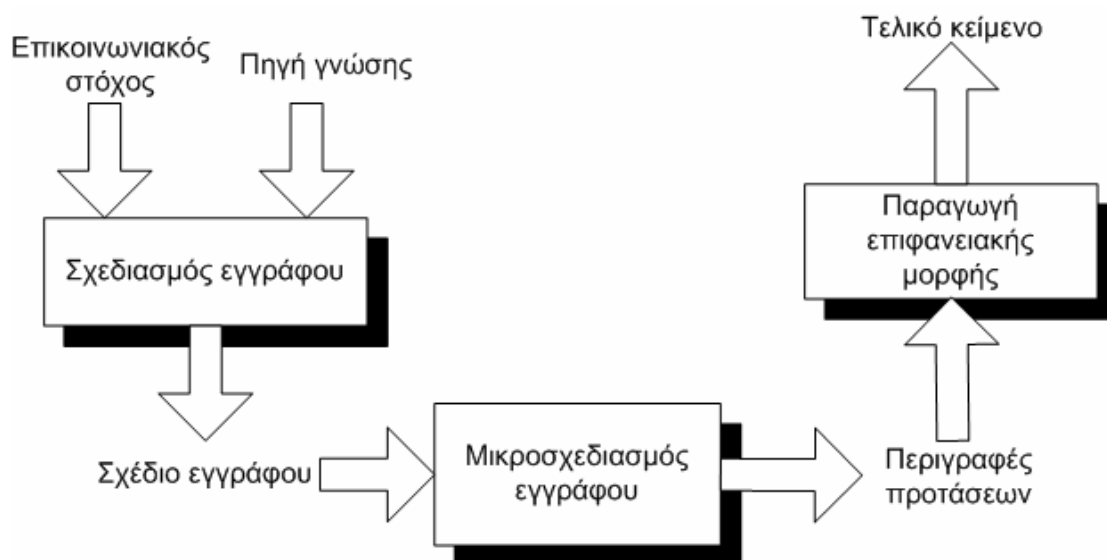
- Γλωσσικοί πόροι: Πρόκειται για πληροφορίες σχετικές με τη γλώσσα στην οποία θα παραχθούν τα κείμενα (π.χ. λεξικά, γραμματικές κλπ.).
- Επικοινωνιακός στόχος: Περιγράφει το σκοπό για τον οποίο πρέπει να παραχθεί το κείμενο σε φυσική γλώσσα. Για παράδειγμα, σε ένα σύστημα που παράγει μετεωρολογικά δελτία, ο επικοινωνιακός στόχος σε μια δεδομένη εκτέλεση του συστήματος μπορεί να είναι η παραγωγή ενός δελτίου για ένα συγκεκριμένο χρονικό διάστημα και τόπο.
- Μοντέλο του χρήστη: Το μοντέλο του χρήστη περιγράφει τον αναγνώστη στον οποίο απευθύνεται το κείμενο που πρόκειται να παραχθεί. Ανάλογα με το μοντέλο του χρήστη, το σύστημα μπορεί να διαμορφώσει τη μορφή ή το περιεχόμενο του παραγόμενου κειμένου, έτσι ώστε να είναι κατάλληλο για το συγκεκριμένο αναγνώστη. Για παράδειγμα, μπορεί να χρησιμοποιήσει διαφορετικό λεξιλόγιο ή να παραγάγει κείμενο για περισσότερο ή λιγότερο εξειδικευμένα θέματα, ανάλογα με το αν ο χρήστης είναι παιδί ή ενήλικος, αρχάριος ή ειδικός.
- Ιστορικό αλληλεπίδρασης: Περιγράφει το τι πληροφορίες έχουν ήδη μεταδοθεί στον αναγνώστη και με ποιον τρόπο (π.χ. ποιες γλωσσικές εκφράσεις χρησιμοποιήθηκαν), έτσι ώστε να είναι δυνατή α) η αποφυγή επαναλήψεων και β) η αναφορά με ποιο φυσικό τρόπο σε αντικείμενα που έχουν ήδη παρουσιαστεί. Επίσης, το ιστορικό αλληλεπίδρασης είναι χρήσιμο όταν θέλουμε ένα σύστημα ΠΦΓ να κάνει συγκρίσεις καινούργιων οντοτήτων με οντότητες που ήδη έχουν αναφερθεί. Ανάλογα με το είδος των μοντέλων χρηστών που υιοθετεί κάθε σύστημα (π.χ. προσωπικά μοντέλα ή μοντέλα που περιγράφουν κατηγορίες χρηστών), το ιστορικό αλληλεπίδρασης μπορεί να εμπεριέχεται ή όχι στα μοντέλα των χρηστών.

Στη συνέχεια περιγράφεται η συνήθης αρχιτεκτονική ενός συστήματος παραγωγής φυσικής γλώσσας.

2.2 Συνήθης αρχιτεκτονική των συστημάτων παραγωγής φυσικής γλώσσας

Η συνήθης αρχιτεκτονική ενός συστήματος παραγωγής φυσικής γλώσσας περιέχει χονδρικά τρεις μονάδες (modules).

- Σχεδιασμός εγγράφου (document planning): Επιλέγει τις πληροφορίες (π.χ. λογικά γεγονότα, facts) που θα μεταδοθούν στο χρήστη και καθορίζει το πώς θα δομηθούν (π.χ. με ποια σειρά θα παρουσιαστούν).
- Μικροσχεδιασμός εγγράφου (microplanning): Επιλέγει τις λέξεις και τις συντακτικές δομές που θα χρησιμοποιηθούν για να μεταδοθούν στο χρήστη οι πληροφορίες που επιλέχθηκαν στο προηγούμενο στάδιο. Σε γενικές γραμμές, παράγεται συνήθως αρχικά μια αφηρημένη περιγραφή πρότασης για κάθε στοιχειώδη πληροφορία (π.χ. λογικό γεγονός) που πρόκειται να μεταδοθεί. Η περιγραφή είναι αφηρημένη, με την έννοια ότι ενδέχεται να μην καθορίζει πλήρως την τελική μορφή των προτάσεων (π.χ. την ακριβή πτώση των ουσιαστικών, την ακριβή σειρά των συντακτικών όρων). Το στάδιο του μικρο-σχεδιασμού περιλαμβάνει, επίσης, την ομαδοποίηση των προτάσεων (aggregation), στη διάρκεια της οποίας συγχωνεύονται προτάσεις που προέρχονται από διαφορετικές στοιχειώδεις πληροφορίες, καθώς και την παραγωγή αναφορικών εκφράσεων, που περιγράφεται παρακάτω.
- Παραγωγή επιφανειακής μορφής (surface realization): Η μονάδα της παραγωγής επιφανειακής μορφής μετατρέπει τις αφηρημένες περιγραφές των προτάσεων σε κείμενο. Στα πιο περίπλοκα συστήματα χρησιμοποιείται συχνά μια γραμματική της γλώσσας-στόχου, η οποία καθορίζει, για παράδειγμα, τη σωστή σειρά των συντακτικών όρων, απαιτεί συμφωνία πτώσης και αριθμού μεταξύ επιθέτων και ουσιαστικών, κλπ., πληροφορίες που ενδέχεται να μην περιέχονται στις αφηρημένες περιγραφές των προτάσεων.



Εικόνα 1: Συνήθης αρχιτεκτονική συστημάτων ΠΦΓ.

Οι παραπάνω μονάδες εκτελούνται διαδοχικά με τον τρόπο που φαίνεται στην Εικόνα 1. Ο σχεδιασμός εγγράφου έχει ως είσοδο τον επικοινωνιακό στόχο και την πηγή γνώσης. Οι άλλες εισοδοι του συστήματος, δηλαδή οι γλωσσικοί πόροι, το μοντέλο χρήστη και το ιστορικό αλληλεπίδρασης επηρεάζουν εν γένει όλες τις μονάδες και παραλείπονται από το διάγραμμα χάριν απλότητας.

Κάθε μία από τις μονάδες της Εικόνας 1 διαχωρίζεται περαιτέρω σε μικρότερες υπομονάδες, οι οποίες περιγράφονται αναλυτικά παρακάτω.

2.2.1 Σχεδιασμός εγγράφου

2.2.1.1 Καθορισμός περιεχομένου (content determination)

Σε αυτό το στάδιο αποφασίζεται ποιες πληροφορίες της πηγής γνώσης θα μεταδοθούν στο χρήστη. Η επιλογή των πληροφοριών αυτών εξαρτάται από

- τον επικοινωνιακό στόχο,
- το μοντέλο του χρήστη,
- το ιστορικό αλληλεπίδρασης και
- άλλους περιορισμούς, όπως ότι το παραγόμενο

κείμενο πρέπει να έχει συγκεκριμένο μήκος. Οι στοιχειώδεις πληροφορίες (π.χ. λογικά γεγονότα) που τελικά επιλέγονται ονομάζονται «μηνύματα».

2.2.1.2 Καθορισμός δομής (document structuring)

Ένα κείμενο δεν αποτελεί μια τυχαία διάταξη από προτάσεις. Στο σημασιολογικό επίπεδο, πρέπει να έχει συνεκτικότητα (coherence), δηλαδή να δίνεται η αίσθηση ότι τα μηνύματα συνδέονται λογικά μεταξύ τους. Σε πιο περίπλοκα συστήματα, αυτό εξασφαλίζεται, μεταξύ άλλων, σχεδιάζοντας τις ρητορικές σχέσεις μεταξύ των μηνυμάτων (π.χ. επεξήγηση, παράθεση, αντίθεση κλπ.) και οργανώνοντας ανάλογα τα μηνύματα σε προτάσεις, παραγράφους, ενότητες κλπ., ώστε να είναι σαφείς οι σχέσεις αυτές. Στην περίπτωση του συστήματος της παρούσας εργασίας, όπου ο στόχος είναι μόνο η παραγωγή σχετικά σύντομων περιγραφών αντικειμένων (συνήθως μεγέθους μιας παραγράφου), η εμπειρία του M-PIRO δείχνει πως ικανοποιητικός βαθμός συνεκτικότητας είναι δυνατόν να επιτευχθεί σχεδιάζοντας απλώς τη σειρά με την οποία θα διατυπώνονται στο τελικό κείμενο τα μηνύματα (π.χ. πρώτα πληροφορίες σχετικές με το είδος του εκθέματος, κατόπιν πληροφορίες σχετικές τον τρόπο κατασκευής του, τη χρήση του κλπ.).

2.2.2 Μικροσχεδιασμός εγγράφου

2.2.2.1 Παραγωγή περιγραφών προτάσεων (lexicalisation)

Σε αυτό το στάδιο κάθε μήνυμα που έχει επιλεγεί από τη μονάδα καθορισμού περιεχομένου απεικονίζεται σε μια αφηρημένη περιγραφή πρότασης, κάτι που προϋποθέτει την επιλογή των λέξεων (ενδεχομένως μόνο των βασικών τύπων τους, χωρίς να φαίνεται ακόμα σε ποια πώση, αριθμό κλπ. θα χρησιμοποιηθούν) και των συντακτικών δομών που θα χρησιμοποιηθούν για

να εκφράσουν το μήνυμα. Η διαδικασία αυτή είναι εν γένει αρκετά σύνθετη, διότι υπάρχουν πολλοί διαφορετικοί τρόποι να εκφραστεί μια πληροφορία. Σε απλοϊκά συστήματα, όπως το σύστημα της παρούσας εργασίας, το στάδιο αυτό βασιζεται στη χρήση σχεδιοτύπων (templates). Τα σχεδιοτύπα είναι σε γενικές γραμμές προκαθορισμένα κείμενα που περιέχουν κενές θέσεις (slots), οι οποίες γεμίζουν με τιμές ορισμάτων του μηνύματος ή εκφράσεις φυσικής γλώσσας που αναφέρονται σε αυτές. Για παράδειγμα, ο τύπος μηνύματος `costs(X,Y)` μπορεί να έχει συνδεθεί στα αγγλικά με το σχεδιοτύπο «`ref(X) costs Y Euro`», όπου το `ref(X)` θα αντικατασταθεί κατά την παραγωγή αναφορικών εκφράσεων (βλ. παρακάτω) με μια κατάλληλη αναφορική έκφραση. Στην περίπτωση του μηνύματος `costs(presario350, 750)`, το προηγούμενο σχεδιοτύπο θα οδηγούσε τελικά σε μια πρόταση όπως «`The Presario 350 costs 750 Euro.`», «`This laptop costs 350 Euro`» ή «`It costs 750 Euro.`». Αντίστοιχα ο τύπος μηνύματος `creator(X,Y)` θα μπορούσε να έχει συνδεθεί με το σχεδιοτύπο «`ref(Y) was created by ref(X)`». Ένας τύπος μηνύματος είναι δυνατόν να είναι συνδεδεμένος με πολλά σχεδιοτύπα, επιτρέποντας μεγαλύτερη ποικιλία στα παραγόμενα κείμενα. Η επιλογή του κατάλληλου σχεδιοτύπου ενδέχεται να εξαρτάται, επίσης, από το μοντέλο του χρήστη (π.χ. διαφορετικές εκφράσεις για παιδιά ή ενήλικες) και το ιστορικό αλληλεπίδρασης (αποφυγή επαναλήψεων των ίδιων εκφράσεων). Τα σχεδιοτύπα ενδέχεται, επίσης, να φέρουν επισημειώσεις (annotations) που να δείχνουν τα μέρη του λόγου των λέξεων που περιέχουν ή τους συντακτικούς ρόλους (π.χ. υποκείμενο, αντικείμενο) των κενών θέσεων, ώστε να διευκολύνεται η ομαδοποίηση (βλ. παρακάτω).

2.2.2.2 Ομαδοποίηση (Aggregation)

Η κύρια λειτουργία της ομαδοποίησης είναι ο συνδυασμός των μεμονωμένων προτάσεων που δημιουργήθηκαν στο προηγούμενο στάδιο (μία πρόταση για κάθε μήνυμα), έτσι ώστε να δημιουργηθούν μεγαλύτερες αλλά συνολικά

περισσότερο ευανάγνωστες προτάσεις. Η χρήση της ομαδοποίησης μπορεί να βελτιώσει σε μεγάλο βαθμό την αναγνωσιμότητα και ποιότητα ενός κειμένου, μιας και τα κείμενα που έχουν μια ανεξάρτητη πρόταση ανά μήνυμα είναι αφύσικα και μονότονα. Παραδείγματα ομαδοποίησης που συνδυάζουν δύο προτάσεις σε μία πολυπλοκότερη είναι τα παρακάτω¹.

Jim has blue eyes. Jim has black hair.

=>

Jim has blue eyes and black hair.

September only had 30 mm of rain. It is usually our wettest month

=>

September, which is usually our wettest month, only had 30 mm of rain.

January was drier than average. January was colder than average

=>

January was drier and colder than average.

John gave a book to Mary. John gave a book to Fred.

=>

John gave books to both Mary and Fred.

2.2.2.3 Παραγωγή αναφορικών εκφράσεων

Σε οποιοδήποτε γνωστικό πεδίο υπάρχουν αρκετές οντότητες υπαρκτές ή αφηρημένες στις οποίες μπορούμε να αναφερθούμε. Η παραγωγή αναφορικών εκφράσεων (π.χ. «αυτό», «αυτό το έκθεμα», «αυτό το άγαλμα», «το προηγούμενο άγαλμα», «ο Δορυφόρος») είναι ένα σημαντικό πρόβλημα για την ΠΦΓ, διότι μπορούμε να αναφερθούμε στην ίδια οντότητα με πολλούς

¹ Τα περισσότερα από τα παραδείγματα ομαδοποίησης που παρουσιάζονται σε αυτό υποκεφάλαιο προέρχονται από το [10].

διαφορετικούς τρόπους. Αυτό ισχύει και στην περίπτωση που αναφερόμαστε σε μια οντότητα για πρώτη φορά, αλλά και όταν αναφερόμαστε σε αυτήν αφού έχει ήδη παρουσιαστεί για πρώτη φορά. Ένα άλλο παράδειγμα παραγωγής διαφορετικών αναφορικών εκφράσεων για μια οντότητα είναι το εξής. Έστω ότι θέλουμε να αναφερθούμε στον μήνα Αύγουστο του 2005, τότε μπορούμε να χρησιμοποιήσουμε τις παρακάτω αναφορικές εκφράσεις:

- Ο Αύγουστος του 2005*
- Ο Αύγουστος*
- Ο Αύγουστος της προηγούμενης χρονιάς*
- Ο προηγούμενος μήνας*

Το ποια αναφορική έκφραση θα επιλέξουμε για να αναφερθούμε στην οντότητα εξαρτάται από τις πληροφορίες που έχουν μεταδοθεί στον αναγνώστη έως εκείνη τη στιγμή, καθώς και από τα συμφραζόμενα (context). Για παράδειγμα, αν το κείμενο μιλάει για τη χρονιά 2005 και μόνο, τότε για να αναφερθούμε στον Αύγουστο του 2005 αρκεί να χρησιμοποιήσουμε την αναφορική έκφραση «ο Αύγουστος» (π.χ. «Ο Αύγουστος είχε πολλές βροχές.») ή αν το κείμενο ακριβώς πριν την αναφορική έκφραση μιλάει για την χρονιά 2006, τότε μπορεί να χρησιμοποιηθεί η αναφορική έκφραση «ο Αύγουστος της προηγούμενης χρονιάς» (π.χ. «Ο Αύγουστος της προηγούμενης χρονιάς ήταν ασυνήθιστα ζεστός.»). Στην περίπτωση που δεν καθορίζεται σε ποια χρονιά αναφέρεται το κείμενο, η καταλληλότερη αναφορική έκφραση είναι ο «ο Αύγουστος του 2005».

Η παραγωγή αναφορικών εκφράσεων αποτελείται συνήθως από δύο στάδια. Στο πρώτο στάδιο επιλέγονται οι ιδιότητες που θα χρησιμοποιηθούν για να αναφερθούμε στην οντότητα που θέλουμε και στο δεύτερο στάδιο γίνεται η επιλογή των λέξεων που θα χρησιμοποιηθούν για να παραχθεί η αναφορική έκφραση από τις ιδιότητες που επιλέχθηκαν. Οι ιδιότητες που θα επιλεγούν πρέπει να προσδιορίζουν μοναδικά την οντότητα στην οποία θέλουμε να αναφερθούμε, έτσι ώστε ο αναγνώστης του κειμένου να μπορεί να την

αναγνωρίσει εύκολα. Για παράδειγμα, έστω ότι έχουμε τρεις ανθρώπους A, B και C με τα παρακάτω χαρακτηριστικά.

	<i>Χρώμα μαλλιών</i>	<i>Χρώμα ματιών</i>	<i>Μήκος μαλλιών</i>
<i>A</i>	<i>Μαύρο</i>	<i>Μαύρο</i>	<i>Κοντό</i>
<i>B</i>	<i>Ξανθό</i>	<i>Μαύρο</i>	<i>Κοντό</i>
<i>C</i>	<i>Ξανθό</i>	<i>Μαύρο</i>	<i>Μακρύ</i>

Για να αναφερθούμε στον C μπορούμε να επιλέξουμε την ιδιότητα *Μήκος μαλλιών* και να παράγουμε την αναφορική έκφραση «ο άνθρωπος με τα μακριά μαλλιά», ενώ για να αναφερθούμε στον B μπορούμε να επιλέξουμε τις ιδιότητες *Χρώμα μαλλιών* και *Μήκος μαλλιών* και να παραγάγουμε την αναφορική έκφραση «ο άνθρωπος με τα ξανθά κοντά μαλλιά». Αν επιλέξουμε για να αναφερθούμε στον B μόνο την ιδιότητα *Μήκος μαλλιών* και παραγάγουμε την αναφορική έκφραση «ο άνθρωπος με τα κοντά μαλλιά», τότε ο αναγνώστης δεν θα μπορεί να διακρίνει αν το κείμενο αναφέρεται στον A ή στο B.

2.2.3 Παραγωγή επιφανειακής μορφής

Σε αυτό το στάδιο γίνεται η αντικατάσταση των αφηρημένων περιγραφών των προτάσεων από πλήρεις προτάσεις, προστίθενται σημεία στίξης και εισάγονται κατάλληλες ετικέτες μορφοποίησης (π.χ. ετικέτες HTML). Οι ετικέτες αυτές είναι χρήσιμες αν για παράδειγμα το τελικό κείμενο παρουσιαστεί σε μια ιστοσελίδα ή χρησιμοποιηθεί ως εισόδος σε ένα άλλο σύστημα που παράγει προφορικό λόγο. Στην τελευταία περίπτωση, οι ετικέτες μπορεί να παρέχουν πληροφορίες που βοηθούν στη βελτίωση της προσωδίας (π.χ. όρια συντακτικών δομών, πληροφορίες που θεωρούνται νέες ή γνωστές, κλπ.) Όπως αναφέρθηκε παραπάνω, κατά την παραγωγή της επιφανειακής δομής τα πιο περίπλοκα συστήματα ΠΦΓ χρησιμοποιούν συχνά μια γραμματική της γλώσσας-στόχου, η οποία καθορίζει, για παράδειγμα, τη

σωστή σειρά των συντακτικών όρων, απαιτεί συμφωνία πτώσης και αριθμού μεταξύ επιθέτων και ουσιαστικών, κλπ., πληροφορίες που ενδέχεται να μην περιέχονται στις αφηρημένες περιγραφές των προτάσεων. Σε απλούστερα συστήματα, όπως το σύστημα της παρούσας εργασίας, όπου η παραγωγή των περιγραφών των προτάσεων γίνεται με σχεδιάτυπα, συχνά δεν είναι απαραίτητα η χρήση τέτοιων γραμματικών.

3. Οντολογίες

3.1 Τι είναι ο σημασιολογικός ιστός

Ο σημασιολογικός ιστός (Semantic Web) [50] αποτελεί ένα όραμα για το μέλλον του διαδικτύου. Έχει σκοπό να επιτρέψει στα προγράμματα υπολογιστών να εντοπίζουν, να επεξεργάζονται και να ενοποιούν ευκολότερα τις πληροφορίες που διατίθενται στο διαδίκτυο. Οι πληροφορίες αυτές έχουν σήμερα κυρίως τη μορφή ιστοσελίδων αποτελούμενων σε μεγάλο βαθμό από κείμενα φυσικής γλώσσας, μια μορφή που κατανοούν και χειρίζονται εύκολα οι άνθρωποι αλλά όχι οι μηχανές. Ο σημασιολογικός ιστός οικοδομείται στηριζόμενος α) στη γλώσσα αναπαράστασης δεδομένων XML (eXtensible Markup Language) [11] και β) στη γλώσσα περιγραφής πόρων του παγκόσμιου ιστού RDF (Resource Description Framework) [45], μια μορφή της οποίας (RDF/XML) χρησιμοποιεί το συντακτικό της XML.

3.2 Τι είναι μια οντολογία και πώς χρησιμοποιείται στο σημασιολογικό ιστό

Ο όρος οντολογία προέρχεται από τη φιλοσοφία και αναφέρεται στην επιστήμη της περιγραφής των οντοτήτων στον κόσμο, αλλά και στο πώς αυτές συσχετίζονται. Στον τομέα της επιστήμης υπολογιστών και ειδικότερα στην τεχνητή νοημοσύνη μια οντολογία είναι ένα μοντέλο το οποίο αναπαριστά ένα γνωστικό πεδίο (domain) και το οποίο χρησιμοποιείται για την εξαγωγή συμπερασμάτων για τα αντικείμενα, τις ιδιότητες και τις σχέσεις των αντικειμένων του γνωστικού αυτού πεδίου [40].

Στο σημασιολογικό ιστό οι οντολογίες χρησιμοποιούνται κυρίως για να ορίσουν το σημασιολογικό λεξιλόγιο ενός γνωστικού πεδίου (είδη αντικειμένων, δυνατές σχέσεις, ιδιότητες κλπ). Το λεξιλόγιο αυτό χρησιμοποιείται κατόπιν για να αναπαρασταθεί-κωδικοποιηθεί η

σημασιολογία των εγγράφων και γενικότερα των πόρων που διατίθενται στο διαδίκτυο (π.χ. μέσω μετα-δεδομένων που ενσωματώνονται και περιγράφουν τα έγγραφα), ώστε να είναι ευκολότερη η επεξεργασία τους από προγράμματα υπολογιστών. Για να διευκολυνθεί η δημιουργία και η διανομή οντολογιών έχουν δημιουργηθεί ειδικές γλώσσες, οι γλώσσες οντολογιών (*Ontology Languages*), οι οποίες παρέχουν έναν τυπικό (formal) τρόπο για την κωδικοποίηση των οντολογιών. Η επικρατέστερη γλώσσα αυτού του είδους στο σημασιολογικό ιστό είναι σήμερα η OWL (*Web Ontology Language*), που περιγράφεται παρακάτω. Στην ίδια κατηγορία μπορεί να θεωρηθεί ότι ανήκουν, σε μικρότερο ή μεγαλύτερο βαθμό, και οι ακόλουθες γλώσσες [5, 6, 13, 27, 34, 41, 42]:

- KIF (*Knowledge Interchange Format*)
- F-Logic (*Frame Logic*)
- SHOE (*Simple HTML Ontology Extension*)
- OCML (*Operational Conceptual Modelling Language*)
- XOL (*XML-Based Ontology Language*)
- DAML+OIL (*DARPA Agent Markup Language + Ontology Inference Layer*)
- RDFS (*RDF Schema*)

3.3 Η γλώσσα OWL και γιατί επιλέχθηκε

Η γλώσσα οντολογιών ιστού OWL (*Web Ontology Language*) [1, 16, 26, 27] είναι μια σημασιολογική γλώσσα σήμανσης για τη δημιουργία και τη διανομή οντολογιών στο διαδίκτυο, η οποία αναπτύχθηκε από το *Web Ontology Working Group* της W3C [54].

Το *Web Ontology Working Group* της W3C αλλά και αρκετές ερευνητικές ομάδες στην Αμερική και στην Ευρώπη είχαν διαγνώσει την ανάγκη ύπαρξης μιας νέας ισχυρής γλώσσας οντολογιών, καθώς η RDF και το RDF Schema [5] δεν παρέχουν την απαραίτητη εκφραστικότητα. Με τη χρήση του λεξιλογίου της RDF και του RDF Schema είναι δυνατόν να οριστούν στιγμιότυπα

τάξεων, ιεραρχίες τάξεων και ιδιοτήτων και καθολικοί περιορισμοί που αφορούν το πεδίο ορισμού και το πεδίο τιμών ιδιοτήτων. Όμως το λεξιλόγιο τους δεν επιτρέπει τον ορισμό αρκετών άλλων πραγμάτων. Μερικά από αυτά είναι τα εξής:

- Περιορισμοί για το πεδίο ορισμού ιδιοτήτων, που να αφορούν τις οντότητες μόνο μερικών τάξεων. Για παράδειγμα, για την τάξη σαρκοφάγα ζώα δεν μπορεί να οριστεί ένας περιορισμός ο οποίος να δηλώνει ότι τα σαρκοφάγα ζώα τρέφονται μόνο με κρέας (σε αντίθεση με άλλα ζώα που τρέφονται και με άλλες τροφές).
- Δηλώσεις που να καθορίζουν ότι δύο τάξεις δεν έχουν κοινές οντότητες (π.χ. οι τάξεις Male και Female).
- Ορισμοί τάξεων χρησιμοποιώντας οποιοδήποτε συνδυασμό από τομές, ενώσεις και συμπληρώματα άλλων τάξεων.
- Ορισμοί περιορισμών που αφορούν το πόσες (μοναδικές) τιμές μια ιδιότητα μπορεί να έχει. Για παράδειγμα, δεν είναι δυνατό να οριστεί ότι ένας άνθρωπος έχει ακριβώς μία μητέρα.
- Ορισμοί ειδικών χαρακτηριστικών για τις ιδιότητες, όπως είναι για παράδειγμα η μεταβατικότητα ή ότι μία ιδιότητα είναι η αντίστροφη μιας άλλης.

Οι παραπάνω περιορισμοί της RDF και του RDF Schema οδήγησαν στον ορισμό της γλώσσας DAML+OIL [6], η οποία αποτέλεσε σημείο εκκίνησης για τον ορισμό της OWL. Η OWL αναπτύχθηκε ως επέκταση του λεξιλογίου της RDF και ενσωμάτωσε λύσεις για προβλήματα που εμφανίστηκαν κατά τον ορισμό και τη χρησιμοποίηση της γλώσσας DAML+OIL.

Παρότι ένα αρκετά μεγάλο μέρος των οντολογιών που βρίσκονται διαθέσιμες στο διαδίκτυο χρησιμοποιούν την γλώσσα DAML+OIL, το σύστημα παραγωγής φυσικής γλώσσας που αναπτύσσεται σε αυτή την εργασία χρησιμοποιεί ως είσοδο την OWL, διότι αυτή αποτελεί βελτίωση της DAML+OIL και διότι η W3C την προτείνει ως την πιο κατάλληλη γλώσσα οντολογιών για το σημασιολογικό ιστό.

Η OWL παρέχει τρεις υπογλώσσες (OWL Lite, OWL DL, OWL Full) με διαφορετική εκφραστική ισχύ, οι οποίες παρουσιάζονται παρακάτω ξεκινώντας από αυτή που έχει την μικρότερη (OWL Lite) και καταλήγοντας σε αυτή που έχει την μεγαλύτερη εκφραστική ισχύ (OWL Full)². Κάθε υπογλώσσα αποτελεί επέκταση της υπογλώσσας με την αμέσως μικρότερη εκφραστική ισχύ.

3.4 OWL Lite

Η OWL Lite σχεδιάστηκε για τους χρήστες που κυρίως ενδιαφέρονται για μια ταξινόμια με απλούς περιορισμούς. Για παράδειγμα, η OWL Lite επιτρέπει να ορίζονται περιορισμοί στον αριθμό των στοιχείων ενός συνόλου, αλλά επιτρέπει οι τιμές του αριθμού των στοιχείων να είναι μόνο 0 ή 1, ενώ οι OWL DL και OWL Full αίρουν αυτόν τον περιορισμό. Η OWL Lite είναι απλή και αυτό διευκολύνει την ανάπτυξη λογισμικού που την υποστηρίζει.

3.4.1 Βασικοί μηχανισμοί της OWL Lite

Η OWL Lite παρέχει τους ακόλουθους βασικούς μηχανισμούς ορισμού οντολογιών. (Η χρήση των προθεμάτων *rdf* ή *rdfs* υποδηλώνει ότι οι αντίστοιχοι μηχανισμοί έχουν κληρονομηθεί από τις RDF και RDF Schema αντίστοιχα.)

- Δήλωση *Class*: Ορίζει μια τάξη οντοτήτων. Για παράδειγμα, η παρακάτω δήλωση ορίζει την τάξη *Man*.

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

Οι τάξεις μπορούν να οργανωθούν σε μια ιεραρχία χρησιμοποιώντας τη δήλωση *subClassOf* (βλ. παρακάτω). Στην OWL υπάρχουν ενσωματωμένες

² Η περιγραφή της γλώσσας OWL έχει στηριχθεί σε μεγάλο βαθμό στα [7, 35, 38]

η τάξη *Thing*, που είναι υπερ-τάξη όλων των οντοτήτων, και η τάξη *Nothing*, η οποία δεν έχει υπό-τάξεις και δεν της ανήκει καμία οντότητα.

- Δήλωση *rdfs:subClassOf*: Οι ιεραρχίες των τάξεων μπορούν να οριστούν χρησιμοποιώντας μια ή περισσότερες δηλώσεις που ορίζουν ότι μια τάξη είναι υπό-τάξη μιας άλλης τάξης. Για παράδειγμα, η τάξη *Man* μπορεί να δηλωθεί ως υπο-τάξη της τάξης *Person*, και η *Person* ως υπο-τάξη της *Mammal*, όπως φαίνεται παρακάτω.

```
<owl:Class rdf:ID="Person">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Mammal" />
  </rdfs:subClassOf>
</owl:Class>

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

- Δήλωση *rdf:Property*: Χρησιμοποιείται για να δηλώσει μια ιδιότητα (property). Οι ιδιότητες χρησιμοποιούνται για να παραστήσουν ότι κάποιο χαρακτηριστικό των οντοτήτων έχει κάποια συγκεκριμένη τιμή, που είναι είτε μια άλλη οντότητα, είτε μια τιμή από συγκεκριμένους τύπους δεδομένων. Ακολουθούν παραδείγματα δηλώσεων των ιδιοτήτων *hasChild*, *hasRelative*, *hasSibling* και *hasAge*.

```
<owl:ObjectProperty rdf:ID="hasChild" />

<owl:ObjectProperty rdf:ID="hasRelative" />

<owl:ObjectProperty rdf:ID="hasSibling" />

<owl:DatatypeProperty rdf:ID="hasAge" />
```

Θεωρούμε ότι οι τρεις πρώτες χρησιμοποιούνται για να συσχετίσουν μια οντότητα της τάξης *Person* με μια άλλη οντότητα της ίδιας τάξης, ενώ η τέταρτη για να αποδώσει μια συγκεκριμένη τιμή του τύπου δεδομένων *Integer*. (Θα δούμε παρακάτω πώς δηλώνονται τα πεδία ορισμού και τιμών

των ιδιοτήτων.) Ιδιότητες σαν τις τρεις πρώτες ανήκουν στην κατηγορία ιδιοτήτων *ObjectProperty* (έχουν ως τιμές οντότητες), ενώ ιδιότητες σαν την τελευταία ανήκουν στη κατηγορία ιδιοτήτων *DatatypeProperty* (παίρνουν τιμές από ένα τύπο δεδομένων).

- Δήλωση *rdf:subPropertyOf*: Χρησιμοποιείται για να δηλώσει ότι μια ιδιότητα είναι υποπερίπτωση μιας άλλης ιδιότητας. Για παράδειγμα, η ιδιότητα *hasSibling* μπορεί να δηλωθεί ως υποπερίπτωση της ιδιότητας *hasRelative*, ως εξής:

```
<owl:ObjectProperty rdf:ID="hasSibling">
  <rdfs:subPropertyOf rdf:resource="#hasRelative"/>
</owl:ObjectProperty>
```

Χρησιμοποιώντας αυτή τη δήλωση μια μηχανή μπορεί να συμπεράνει ότι αν μια οντότητα *X* συνδέεται με μια οντότητα *Y* μέσω της ιδιότητας *hasSibling*, τότε η *X* συνδέεται και μέσω της ιδιότητας *hasRelative* με την *Y*.

- Δήλωση *rdfs:domain*: Δηλώνει το πεδίο ορισμού μιας ιδιότητας. Για παράδειγμα, η παρακάτω δήλωση δείχνει ότι η ιδιότητα *hasChild* έχει πεδίο ορισμού την τάξη *Mammal*.

```
<owl:ObjectProperty rdf:ID="hasChild">
  <rdfs:domain rdf:resource="#Mammal"/>
</owl:ObjectProperty>
```

Αν *George hasChild John*, τότε η οντότητα *George* πρέπει να ανήκει στην τάξη *Mammal*

- Δήλωση *rdfs:range*: Δηλώνει το πεδίο τιμών μιας ιδιότητας. Για παράδειγμα, η παρακάτω δήλωση δείχνει ότι η ιδιότητα *hasChild* έχει και πεδίο τιμών την τάξη *Mammal*.

```
<owl:ObjectProperty rdf:ID="hasChild">
  <rdfs:range rdf:resource="#Mammal"/>
</owl:ObjectProperty>
```

Αν *George hasChild John*, η οντότητα *John* πρέπει να ανήκει στην τάξη *Mammal*.

- Δηλώσεις οντοτήτων: Οι οντότητες είναι στιγμιότυπα (instances) τάξεων και οι ιδιότητες μπορούν να χρησιμοποιηθούν για συσχετίσουν μια

οντότητα με μια άλλη. Για παράδειγμα, οι παρακάτω δηλώσεις δείχνουν ότι οι οντότητες *George* και *John* είναι στιγμιότυπα της τάξης *Person*, άρα και της *Mammal*, πως ο *John* είναι παιδί του *George* και πως οι ηλικίες τους είναι 35 και 60 αντίστοιχα.

```
<Person rdf:ID="John">
  <hasParent rdf:resource="#George"/>
  <hasAge rdf:datatype="http://www.w3.org/2001/XMLSchema#integer"
  >35</hasAge>
</Person>

<Person rdf:ID="George">
  <hasAge rdf:datatype="http://www.w3.org/2001/XMLSchema#integer"
  >60</hasAge>
  <hasChild rdf:resource="#John"/>
</Person>
```

3.4.2 Ισοδυναμίες, ισότητες και ανισότητες στην OWL Lite

- Δήλωση *equivalentClass*: Δύο τάξεις μπορούν να δηλωθούν ως ισοδύναμες. Οι ισοδύναμες τάξεις έχουν τις ίδιες οντότητες. Για παράδειγμα, η τάξη *Car* μπορεί να δηλωθεί ως ισοδύναμη με τη τάξη *Automobile*, όπως φαίνεται παρακάτω.

```
<owl:Class rdf:ID="Car">
  <owl:equivalentClass rdf:resource="#Automobile"/>
</owl:Class>
```

Από μια τέτοια δήλωση μια μηχανή μπορεί να συμπεράνει ότι οποιαδήποτε οντότητα ανήκει στην τάξη *Car* ανήκει επίσης και στη τάξη *Automobile* και αντίστροφα. Οι δηλώσεις ισοδυναμίας τάξεων είναι χρήσιμες όταν, για παράδειγμα, επιθυμεί κανείς να συνδυάσει οντολογίες που ενδεχομένως χρησιμοποιούν διαφορετικά ονόματα τάξεων για ισοδύναμες έννοιες.

- Δήλωση *equivalentProperty*: Δύο ιδιότητες μπορούν να δηλωθούν ως ισοδύναμες. Για παράδειγμα, η ιδιότητα *hasLeader* μπορεί να δηλωθεί ως ισοδύναμη με την *hasHead* ως εξής:

```
<owl:ObjectProperty rdf:ID="hasHead">
  <owl:equivalentProperty>
    <owl:ObjectProperty rdf:ID="hasLeader"/>
  </owl:equivalentProperty>
</owl:ObjectProperty>
```

Από αυτή τη δήλωση μια μηχανή μπορεί να συμπεράνει ότι αν για δύο οντότητες X, Y έχουμε $X \text{ hasLeader } Y$ τότε και $X \text{ hasHead } Y$ και αντίστροφα.

- Δήλωση *sameAs*: Δύο οντότητες μπορούν να δηλωθούν ως ίδιες. Για παράδειγμα, η οντότητα *William_Jefferson_Clinton* μπορεί να δηλωθεί ίδια με την οντότητα *BillClinton* ως εξής:

```
<Person rdf:ID="William_Jefferson_Clinton">
  <owl:sameAs>
    <Person rdf:ID="BillClinton"/>
  </owl:sameAs>
</Person>
```

- Δήλωση *differentFrom*: Μια οντότητα μπορεί να δηλωθεί ως (αμοιβαία) διαφορετική από άλλες οντότητες. Για παράδειγμα η οντότητα *Jim* μπορεί να δηλωθεί διαφορετική από τις οντότητες *George* και *John* ως εξής.

```
<Man rdf:ID="George" />

<Man rdf:ID="John">
  <owl:differentFrom rdf:resource="#George"/>
</Man>

<Man rdf:ID="Jim">
  <owl:differentFrom rdf:resource="#George"/>
  <owl:differentFrom rdf:resource="#John"/>
</Man>
```

Η ρητή δήλωση ότι δύο οντότητες είναι διαφορετικές είναι πολύ σημαντική για μια γλώσσα σαν την OWL, καθώς αυτή δεν απαιτεί κάθε οντότητα να έχει μόνο ένα όνομα και επομένως μια μηχανή δεν θα μπορεί να συμπεράνει ότι οι οντότητες *Jim* και *George* είναι διαφορετικές.

- Δήλωση *AllDifferent*: Πολλές οντότητες μπορούν να δηλωθούν εύκολα ως αμοιβαία διαφορετικές ως εξής:

```
<owl:AllDifferent>
  <owl:distinctMembers rdf:parseType="Collection">
    <Man rdf:about="#Jim" />
    <Man rdf:about="#George" />
    <Man rdf:about="#John" />
  </owl:distinctMembers>
</owl:AllDifferent>
```

3.4.3 OWL Lite και χαρακτηριστικά ιδιοτήτων

- Δήλωση *inverseOf*: Μια ιδιότητα μπορεί να δηλωθεί ως αντίστροφη μιας άλλης. Αν η ιδιότητα *P1* δηλωθεί ως αντίστροφη της ιδιότητας *P2* και η οντότητα *X* σχετίζεται με μια άλλη οντότητα *Y* μέσω της ιδιότητας *P1*, τότε και η οντότητα *Y* σχετίζεται με τη οντότητα *X* μέσω της ιδιότητας *P2*. Για παράδειγμα αν η *hasChild* δηλωθεί ως αντίστροφη της *hasParent*, όπως φαίνεται παρακάτω, και *Jim hasParent John* τότε μια μηχανή μπορεί να συμπεράνει ότι *John hasChild Jim*.

```
<owl:ObjectProperty rdf:ID="hasChild">
  <owl:inverseOf>
    <owl:ObjectProperty rdf:ID="hasParent"/>
  </owl:inverseOf>
</owl:ObjectProperty>
```

- Δήλωση *TransitiveProperty*: Μια ιδιότητα μπορεί να δηλωθεί ως μεταβατική. Για παράδειγμα αν η ιδιότητα *hasAncestor* δηλωθεί μεταβατική, όπως φαίνεται παρακάτω και αν *Jim hasAncestor George* και

George has Ancestor John τότε μια μηχανή μπορεί να συμπεράνει ότι *Jim hasAncestor John*.

```
<owl:TransitiveProperty rdf:ID="hasAncestor">
  <rdfs:range rdf:resource="#Person"/>
  <rdfs:type
    rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:domain rdf:resource="#Person"/>
</owl:TransitiveProperty>
```

- Δήλωση *SymmetricProperty*: Μια ιδιότητα μπορεί να δηλωθεί ως συμμετρική. Για παράδειγμα αν η ιδιότητα *hasSibling* δηλωθεί συμμετρική, όπως φαίνεται παρακάτω, και αν *Jim hasSibling Mary* τότε μια μηχανή μπορεί να συμπεράνει ότι *Mary hasSibling Jim*.

```
<owl:SymmetricProperty rdf:ID="hasSibling">
  <rdfs:range rdf:resource="#Person"/>
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:type
    rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</owl:SymmetricProperty>
```

- Δήλωση *FunctionalProperty*: Αν μια ιδιότητα δηλωθεί ως συναρτησιακή, τότε έχει το πολύ μια τιμή για κάθε οντότητα του πεδίου ορισμού. Η δήλωση αυτή είναι μια ευκολία-συντόμευση που παρέχει η OWL, καθώς είναι ισοδύναμη με το να δηλώσουμε ταυτόχρονα τους περιορισμούς πληθαρίσθμου *minCardinality 0* και *maxCardinality 1* για αυτή την ιδιότητα (βλ. παρακάτω). Για παράδειγμα η ιδιότητα *hasPrimaryEmployer* μπορεί να δηλωθεί ως συναρτησιακή ως εξής:

```
<owl:ObjectProperty rdf:ID="hasPrimaryEmployer">
  <rdfs:type
    rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  >
</owl:ObjectProperty>
```

Από αυτή τη δήλωση μια μηχανή μπορεί να συμπεράνει ότι κάθε οντότητα δεν έχει παραπάνω από έναν εργοδότη.

- Δήλωση *inverseFunctionalProperty*: Μια ιδιότητα μπορεί να δηλωθεί ως αντιστρόφως συναρτησιακή. Μια τέτοια δήλωση σημαίνει ότι η αντίστροφη αυτής της ιδιότητας είναι συναρτησιακή και επομένως έχει το πολύ μια τιμή για κάθε οντότητα. Για παράδειγμα η ιδιότητα *hasUSSocialSecurityNumber* (το οποίο είναι μοναδικό χαρακτηριστικό για τους κατοίκους των Η.Π.Α) μπορεί να δηλωθεί ως αντιστρόφως συναρτησιακή ως εξής:

```
<owl:InverseFunctionalProperty
  rdf:ID="hasUSSocialSecurityNumber">
  <rdf:type
    rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</owl:InverseFunctionalProperty>
```

Από την παραπάνω δήλωση μια μηχανή μπορεί να συμπεράνει ότι δεν υπάρχουν δύο διαφορετικές οντότητες της τάξης *Person* (το πεδίο ορισμού της ιδιότητας) που να έχουν τον ίδιο αριθμό αμερικάνικης κοινωνικής ασφάλισης. Επίσης, μια μηχανή μπορεί να συμπεράνει ότι αν δύο οντότητες της τάξης *Person* έχουν τον ίδιο αριθμό κοινωνικής ασφάλισης, τότε αυτές οι οντότητες είναι ίδιες.

3.4.4 Περιορισμοί ιδιοτήτων της OWL Lite

Η OWL Lite επιτρέπει την επιβολή κάποιων περιορισμών στον τρόπο με τον οποίο οι ιδιότητες μπορούν να χρησιμοποιηθούν από τις οντότητες μιας τάξης. Τέτοιοι περιορισμοί καθώς και περιορισμοί πληθάριθμου (βλ. επόμενη ενότητα) δηλώνονται χρησιμοποιώντας την ετικέτα *owl:Restriction*. Η ετικέτα *owl:onProperty* δηλώνει την περιοριζόμενη ιδιότητα.

- Δήλωση *allValuesFrom*: Ο περιορισμός αυτός δηλώνεται για μια ιδιότητα, δείχνει σε μια τάξη και δηλώνεται μέσα σε μια άλλη τάξη. Έστω ότι ο περιορισμός δηλώνεται για μια ιδιότητα *P*, δείχνει σε μια τάξη *B* και δηλώνεται μέσα σε μια τάξη *A*. Αν τότε μια οντότητα που ανήκει στην *A*

σχετίζεται με μια δεύτερη οντότητα μέσω της ιδιότητας P , τότε η δεύτερη οντότητα πρέπει να ανήκει στη τάξη B . Ο συγκεκριμένος περιορισμός αφορά μόνο τις οντότητες που ανήκουν στη τάξη A και σχετίζονται μέσω της περιοριζόμενης ιδιότητας P με μια άλλη οντότητα. Επομένως η ετικέτα *allValuesFrom* εισάγει έναν τοπικό περιορισμό, σε αντίθεση με την ετικέτα *rdfs:range* που εισάγει έναν ολικό περιορισμό για το πεδίο τιμών μιας ιδιότητας και αφορά κάθε εμφάνιση της ιδιότητας. Για παράδειγμα μπορεί να οριστεί ένα περιορισμός που να δηλώνει ότι οι καμηλοπαρδάλεις τρέφονται μόνο με φύλλα ως εξής:

```
<owl:Class rdf:ID="giraffe">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:allValuesFrom>
        <owl:Class rdf:ID="leaf"/>
      </owl:allValuesFrom>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#eats"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

- Δήλωση *someValuesFrom*: Ο περιορισμός αυτός δηλώνεται για μια ιδιότητα, δείχνει σε μια τάξη και δηλώνεται μέσα σε μια άλλη τάξη. Ο περιορισμός ισχύει μόνο για τις οντότητες που ανήκουν στην τάξη μέσα στην οποία δηλώνεται, όταν σε αυτές εφαρμόζεται η περιοριζόμενη ιδιότητα και καθορίζει ότι τουλάχιστον μια από τις πιθανώς πολλές τιμές της ιδιότητας αυτής ανήκει στην τάξη που δείχνει ο περιορισμός. Στο παρακάτω παράδειγμα, ο περιορισμός *someValuesFrom* μέσα στην τάξη *SemanticWebPaper* για την ιδιότητα *hasKeyword* δείχνει στην τάξη *SemanticWebTopic*. Ο περιορισμός δηλώνει ότι σε κάθε οντότητα της τάξης *SemanticWebPaper* τουλάχιστον μία από τις τιμές της ιδιότητας *hasKeyword* (η ιδιότητα μπορεί να έχει πολλές τιμές για κάθε οντότητα *SemanticWebPaper*) είναι οντότητα της τάξης *SemanticWebTopic*.

```

<owl:Class rdf:ID="SemanticWebPaper">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasKeyword"/>
      </owl:onProperty>
      <owl:someValuesFrom>
        <owl:Class rdf:ID="SemanticWebTopic"/>
      </owl:someValuesFrom>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

```

3.4.5 Περιορισμοί πληθάριθμου στην OWL Lite

Η OWL Lite περιλαμβάνει μια απλή μορφή περιορισμών πληθάριθμου συνόλων. Οι περιορισμοί αυτοί δηλώνονται όπως οι περιορισμοί *someValuesFrom* και *allValuesFrom* μέσα σε μια τάξη και επομένως έχουν τοπική εμβέλεια, δηλαδή εφαρμόζονται μόνο όταν η ιδιότητα χρησιμοποιείται με οντότητα του πεδίου ορισμού της που ανήκει στην τάξη μέσα στην οποία δηλώθηκε ο περιορισμός..

- Περιορισμός *minCardinality*: Αν έχει οριστεί ένας περιορισμός ελάχιστου πληθάριθμου *minCardinality* n μέσα σε μια τάξη C για μια ιδιότητα P , τότε αυτό σημαίνει πως όλες οι οντότητες αυτής της τάξης θα σχετίζονται μέσω της ιδιότητας P με τουλάχιστον n οντότητες η κάθε μία. Για παράδειγμα μπορούμε να ορίσουμε για την τάξη *Parent* και την ιδιότητα *hasOffSpring* έναν περιορισμό *minCardinality* 1 (ένας γονιός έχει τουλάχιστον έναν απόγονο), ενώ για την τάξη *Person* και την ιδιότητα *hasOffSpring* έναν περιορισμό *minCardinality* 0 (ένα πρόσωπο δεν είναι απαραίτητο να έχει απόγονο) ως εξής:

```

<owl:Class rdf:ID="Parent">
  <rdfs:subClassOf>

```

```

<owl:Restriction>
  <owl:onProperty>
    <owl:ObjectProperty rdf:ID="hasOffSpring" />
  </owl:onProperty>
  <owl:minCardinality
    rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
  >1</owl:minCardinality>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>

```

Στη OWL Lite το n μπορεί να πάρει μόνο τις τιμές 0 και 1.

- Περιορισμός *maxCardinality*: Αν έχει οριστεί ένας περιορισμός μέγιστου πληθαιρίθμου *maxCardinality* n μέσα σε μια τάξη C για μια ιδιότητα P , τότε αυτό σημαίνει πως όλες οι οντότητες αυτής της τάξης θα σχετίζονται μέσω της ιδιότητας P με το πολύ n οντότητες η κάθε μία. Για παράδειγμα μπορούμε να ορίσουμε για την τάξη *USCitizens* και την ιδιότητα *hasRegisteredVotingState* έναν περιορισμό *maxCardinality* 1 (κάθε πολίτης των Η.Π.Α. έχει άδεια να ψηφίσει μόνο σε μια πολιτεία) ως εξής:

```

<owl:Class rdf:ID="USCitizens">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasRegisteredVotingState" />
      </owl:onProperty>
      <owl:maxCardinality
        rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >1</owl:maxCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

```

Όπως και στους περιορισμούς *minCardinality*, στην OWL Lite το n μπορεί να πάρει μόνο τις τιμές 0 και 1.

- Περιορισμός *Cardinality*: Αν έχει οριστεί ένας περιορισμός *Cardinality* n μέσα σε μια τάξη C για μια ιδιότητα P , τότε αυτό σημαίνει πως όλες οι

οντότητες αυτής της τάξης θα σχετίζονται μέσω της ιδιότητας P με ακριβώς n οντότητες η κάθε μία. Αυτός ο τύπος περιορισμού δημιουργήθηκε για ευκολία, καθώς αν δεν υπήρχε θα έπρεπε στην παραπάνω περίπτωση να δηλώσουμε 2 περιορισμούς για την ιδιότητα P που να αφορούν την τάξη C : τους $maxCardinality\ n$ και $minCardinality\ n$. Για παράδειγμα, με τη χρήση του περιορισμού $Cardinality$ μπορεί να οριστεί ότι κάθε άνθρωπος έχει ακριβώς μια φυσική μητέρα.

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

Όπως και στους περιορισμούς $minCardinality$ και $maxCardinality$, στην OWL Lite το n μπορεί να πάρει μόνο τις τιμές 0 και 1

3.4.6 OWL Lite και τομές τάξεων

Η OWL Lite επιτρέπει μόνο τομές επώνυμων τάξεων και περιορισμών. Για παράδειγμα, η τάξη *EmployedPerson* μπορεί να δηλωθεί ως τομή των τάξεων *EmployedThings* (η τάξη που περιέχει τα πράγματα που έχουν τουλάχιστον 1 εργοδότη) και *Person* ως εξής:

```
<owl:Class rdf:ID="EmployedPerson">
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Person"/>
        <owl:Class rdf:about="#EmployedThings"/>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
```



```
</owl:Class>
```

Από αυτή τη δήλωση μια μηχανή θα μπορούσε να συμπεράνει ότι κάθε οντότητα που ανήκει στην τάξη *EmployedPerson* έχει τουλάχιστον έναν εργοδότη. Επίσης, ένα παράδειγμα τάξης η οποία ορίζεται με την βοήθεια της τομής είναι αυτό της τάξης *PinotNoir*. Η τάξη *PinotNoir* ορίζεται ως η τάξη που περιέχει τα κρασιά τα οποία παρασκευάζονται από το πολύ ένα σταφύλι, το σταφύλι Pinot Noir. Η διαφορά του ορισμού αυτής της τάξης με την προηγούμενη είναι ότι στην τομή συμμετέχουν και περιορισμοί εκτός από επώνυμες τάξεις.

```
<owl:Class rdf:ID="PinotNoir">
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Wine" />
        <owl:Restriction>
          <owl:onProperty rdf:resource="#madeFromGrape" />
          <owl:hasValue rdf:resource="#PinotNoirGrape" />
        </owl:Restriction>
        <owl:Restriction>
          <owl:onProperty rdf:resource="#madeFromGrape" />
          <owl:maxCardinality
            rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
        </owl:Restriction>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>
```

3.4.7 Τύποι δεδομένων της OWL Lite

Οι τύποι δεδομένων (data types) που χρησιμοποιεί η OWL Lite είναι κατά μεγάλο μέρος οι ενσωματωμένοι τύποι δεδομένων (built-in) του XML Schema [55]. Οι παρακάτω ενσωματωμένοι τύποι του XML Schema συνιστώνται για χρήση στην γλώσσα OWL Lite.

xsd:string	xsd:normalizedString	xsd:boolean
xsd:decimal	xsd:float	xsd:double
xsd:integer	xsd:nonNegativeInteger	xsd:positiveInteger
xsd:nonPositiveInteger	xsd:negativeInteger	xsd:unsignedByte
xsd:long	xsd:int	xsd:short
xsd:unsignedLong	xsd:unsignedInt	xsd:unsignedShort
xsd:hexBinary	xsd:base64Binary	xsd:gYearMonth
xsd:dateTime	xsd:time	xsd:date
xsd:gYear	xsd:gMonthDay	xsd:gDay
xsd:anyURI	xsd:token	xsd:language
xsd:NMTOKEN	xsd:Name	xsd:NCName
xsd:byte	xsd:gMonth	

Οι παραπάνω τύποι, με την προσθήκη του τύπου δεδομένων του RDF Schema *rdfs:Literal*, αποτελούν τους ενσωματωμένους τύπους δεδομένων της OWL Lite. Άλλοι ενσωματωμένοι τύποι του XML Schema μπορούν να χρησιμοποιηθούν μόνο στην OWL Full, λαμβάνοντας όμως υπόψη τις προειδοποιήσεις που δίνονται στο [43].

Αν και το XML Schema παρέχει μηχανισμούς για την κατασκευή νέων τύπων δεδομένων από τους χρήστες, αυτοί οι τύποι συνιστάται να μην χρησιμοποιούνται στην OWL, καθώς δεν υποστηρίζονται από εργαλεία που υποστηρίζουν την OWL [14, 43, 35]. Την παρούσα χρονική στιγμή υπάρχει μια εξελισσόμενη προσπάθεια έτσι ώστε να βρεθεί μια λύση στο πώς οι υπόλοιποι τύποι του XML Schema (οι ενσωματωμένοι που έως τώρα δεν χρησιμοποιούνται και οι οριζόμενοι από το χρήστη) να μπορούν να χρησιμοποιούνται στην OWL. Διάφορες λύσεις έχουν προταθεί, οι οποίες όμως δεν έχουν επίσημα ενσωματωθεί στη γλώσσα OWL [19, 28]

3.4.8 Εισαγωγή οντολογιών στην OWL Lite

Η OWL Lite υποστηρίζει την εισαγωγή (*import*) οντολογιών. Με τη χρήση της δήλωσης *owl:imports*, μια οντολογία *A* μπορεί να εισαγάγει μια άλλη οντολογία *B* η οποία περιέχει ορισμούς που χρησιμοποιούνται από την *A*. Η

εισαγωγή οντολογιών είναι μεταβατική, δηλαδή αν η A εισάγει την B και η B εισάγει την C , τότε και η A εισάγει την C .

3.4.9 OWL Lite και ιδιότητες σχολιασμού

Η OWL Lite επιτρέπει το σχολιασμό των τάξεων, των ιδιοτήτων και των οντοτήτων με τις εξής ιδιότητες σχολιασμού (annotation properties):

- *versionInfo*: Μπορεί να χρησιμοποιηθεί για να δώσουμε πληροφορίες για την έκδοση της οντολογίας (με τη μορφή σχολίων φυσικής γλώσσας), όπως στο παρακάτω παράδειγμα.

```
<owl:Ontology rdf:about="">
  <owl:versionInfo>v 1.18 2006/06/13 9:56:51
  galanisd</owl:versionInfo>
  <rdfs:comment>An example ontology</rdfs:comment>
  <owl:imports rdf:resource="http://www.example.org/foo"/>
</owl:Ontology>
```

Υπάρχουν επίσης ιδιότητες σχολιασμού όπως *priorVersion*, *deprecatedClass*, *deprecatedProperty*, *backwardCompatibleWith*, *incompatibleWith* κλπ., που μπορούν να χρησιμοποιηθούν για να παραστήσουν τις σχέσεις μεταξύ διαφόρων εκδόσεων (versions) των οντολογιών (βλ. [35]).

- *rdfs:label*: Μπορεί να χρησιμοποιηθεί για να δώσουμε ένα όνομα φυσικής γλώσσας ή γενικότερα μια περιγραφή φυσικής γλώσσας σε τάξεις οντότητες και ιδιότητες. Ένα παράδειγμα είναι το παρακάτω.

```
<owl:Class rdf:about="#MusicalWork">
  <rdfs:label>Musical work</rdfs:label>
</owl:Class>
```

- *rdfs:comment*: Μπορεί να χρησιμοποιηθεί για να κάνουμε οποιοδήποτε σχολιασμό φυσικής γλώσσας, για παράδειγμα:

```
<owl:Class rdf:ID="tree">
  <rdfs:comment>Trees are a type of plants</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#plant"/>
```

```
</owl:Class>
```

- *rdfs:isDefinedBy*: Χρησιμοποιείται για τον ορισμό διευθύνσεων που παρέχουν πληροφορίες για την οντολογία, για παράδειγμα:

```
<Ontology rdf:about="">  
<rdfs:isDefinedBy rdf:resource="http://www.w3.org/TR/2004/REC-  
owl-semantic-20040210/" />  
</Ontology>
```

3.5 Οι OWL DL και OWL Full

Η OWL DL σχεδιάστηκε για τους χρήστες που επιθυμούν τη μέγιστη δυνατή εκφραστικότητα διατηρώντας τις α) την υπολογιστική πληρότητα (όλα τα συμπεράσματα να είναι εγγυημένα υπολογίσιμα) και β) την αποφασισιμότητα (όλοι οι υπολογισμοί να απαιτούν πεπερασμένο χρόνο). Έχει ονομαστεί έτσι, διότι αντιστοιχεί στις Περιγραφικές Λογικές (Description Logics [12]). Οι Περιγραφικές Λογικές είναι μια οικογένεια λογικών γλωσσών που επιτρέπουν την εύκολη περιγραφή τάξεων αντικειμένων, καθώς και την αποδοτική διενέργεια ελέγχων υπαγωγής (subsumption) και ταξινόμησης (classification), δηλαδή ελέγχων του κατά πόσον μια τάξη αποτελεί υπο-τάξη μιας άλλης και του κατά πόσον μια οντότητα ανήκει σε μια τάξη. Σήμερα υπάρχουν πολλές μηχανές εξαγωγής συμπερασμάτων για την οικογένεια αυτών των γλωσσών, οι οποίες μπορούν να χρησιμοποιηθούν και στην περίπτωση της OWL DL (και της OWL Lite, που είναι υποσύνολό της).

Η OWL Full σχεδιάστηκε για εκείνους τις χρήστες που επιθυμούν να έχουν τη μέγιστη εκφραστικότητα καθώς και τη συντακτική ελευθερία της RDF, αλλά χωρίς να υπάρχουν οι υπολογιστικές εγγυήσεις της OWL DL. Η OWL Full παραβιάζει τις περιορισμούς των μηχανών εξαγωγής συμπερασμάτων των Περιγραφικών Λογικών και επομένως οι μηχανές αυτές δεν μπορούν εν γένει να χρησιμοποιηθούν για εξαγωγή συμπερασμάτων από οντολογίες OWL Full. Οι OWL DL και OWL Full χρησιμοποιούν το ίδιο λεξιλόγιο, αλλά η OWL DL θέτει κάποιους περιορισμούς στη χρήση αυτού του λεξιλογίου. Γενικά η

διαφορά μεταξύ OWL DL και OWL Full είναι ότι η OWL DL απαιτεί τον αυστηρό διαχωρισμό των τύπων (π.χ. μια τάξη δεν μπορεί να είναι ταυτόχρονα οντότητα ή ιδιότητα). Επίσης η OWL DL απαιτεί κάθε ιδιότητα να είναι είτε *ObjectProperty* είτε *DatatypeProperty*, δηλαδή να συσχετίζει οντότητες του πεδίου ορισμού τις με τις οντότητες ή τιμές συγκεκριμένων τύπων δεδομένων αντίστοιχα (βλ. ενότητα 3.4.1). Παρακάτω περιγράφονται οι επεκτάσεις του λεξιλογίου τις OWL Lite που χρησιμοποιούνται τις OWL DL και OWL Full.

- *oneOf*: Επιτρέπει τον ορισμό μιας τάξης μέσω απαρίθμησης (enumeration) των οντοτήτων που την αποτελούν. Για παράδειγμα η τάξη *DaysOfTheWeek* μπορεί να περιγραφεί απαριθμώντας απλά τις οντότητες *Sunday*, *Monday*, *Tuesday*, *Wednesday*, *Thursday*, *Friday*, *Saturday*, τις φαίνεται στο παρακάτω παράδειγμα.

```
<owl:Class rdf:ID="DaysOfTheWeek">
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:about="#Monday"/>
    <owl:Thing rdf:about="#Tuesday"/>
    <owl:Thing rdf:about="#Wednesday"/>
    <owl:Thing rdf:about="#Thursday"/>
    <owl:Thing rdf:about="#Friday"/>
    <owl:Thing rdf:about="#Saturday"/>
    <owl:Thing rdf:about="#Sunday"/>
  </owl:oneOf>
</owl:Class>
```

- *hasValue*: Οι οντότητες μιας τάξης μπορεί να απαιτείται να έχουν μια συγκεκριμένη τιμή ιδιότητας. Για παράδειγμα οι οντότητες τις τάξης *Zinfandel* μπορούν να οριστούν ως τα κρασιά που έχουν κόκκινο χρώμα και επομένως η ιδιότητα *hasColor* πρέπει να έχει τιμή *Red*, κάτι που μπορεί να διατυπωθεί ως εξής:

```
<owl:Class rdf:ID="Zinfandel">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasColor" />
      <owl:hasValue rdf:resource="#Red" />
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

```

    </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>

```

- *disjointWith*: Δύο τάξεις μπορούν να δηλωθούν ως ξένες. Για παράδειγμα οι τάξεις *Man* και *Woman* μπορούν να δηλωθούν ως ξένες, ως εξής:

```

<owl:Class rdf:about="#Man">
  <owl:disjointWith rdf:resource="#Woman"/>
</owl:Class>

```

Από αυτή τη δήλωση μια μηχανή μπορεί να εντοπίσει μια ασυνέπεια όταν μια οντότητα έχει δηλωθεί ότι ανήκει και τις δύο τάξεις.

- *unionOf*, *intersectionOf*, *complementOf*: Αντιστοιχούν στην ένωση, την τομή και το συμπλήρωμα τάξεων. Για παράδειγμα με την χρήση της ένωσης μπορούμε να δηλώσουμε ότι μια τάξη περιέχει τις οντότητες που είναι είτε *USCitizens* είτε *DutchCitizens*, ως εξής:

```

<owl:Class rdf:ID=" USCitizensORDutchCitizens">
  <owl:equivalentClass>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#USCitizens"/>
        <owl:Class rdf:about="#DutchCitizens"/>
      </owl:unionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

```

Επίσης, με το συμπλήρωμα μπορούμε για παράδειγμα να ορίσουμε μια τάξη *NotAnimal* η οποία περιέχει τις οντότητες που δεν είναι ζώα (*complementOf Animal*):

```

<owl:Class rdf:ID="NotAnimal">
  <owl:equivalentClass>
    <owl:Class>
      <owl:complementOf rdf:resource="#Animal"/>
    </owl:Class>

```

```

</owl:equivalentClass>
</owl:Class>

```

Οι περιορισμοί που ισχύουν στην OWL Lite για την τομή (βλ. ενότητα 3.4.6) δεν ισχύουν για τις OWL DL και OWL Full.

- *minCardinality*, *maxCardinality*, *Cardinality*: Ενώ στην OWL Lite οι περιορισμοί πληθαρικών μπορούν να είναι μόνο 0 ή 1, στις OWL DL και OWL Full επιτρέπεται οποιοσδήποτε μη αρνητικός ακέραιος.
- *Σύνθετες τάξεις*: Σε πολλές δηλώσεις του λεξιλογίου της, η OWL Lite επιτρέπει μόνο τη χρήση επώνυμων τάξεων (για παράδειγμα στις δηλώσεις *subclassOf* και *equivalentClass*). Η OWL Full και η OWL DL αίρουν αυτούς τους περιορισμούς, επιτρέποντας παντού τη χρήση ανώνυμων τάξεων που προκύπτουν ως συνδυασμοί τομών, ενώσεων, συμπληρωμάτων, απαριθμήσεων τάξεων και περιορισμών. Για παράδειγμα, ο παρακάτω ορισμός τάξης δεν επιτρέπεται στην OWL Lite, γιατί περιέχει μια ανώνυμη τάξη η οποία προκύπτει από το συνδυασμό μιας ένωσης και μιας τομής, Στην OWL Full και OWL DL ένας τέτοιος ορισμός επιτρέπεται.

```

<owl:Class rdf:ID="ManWithRedOrBlackHair">
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Man"/>
        <owl:Class>
          <owl:unionOf rdf:parseType="Collection">
            <owl:Restriction>
              <owl:onProperty>
                <owl:ObjectProperty rdf:ID="hasHairColor"/>
              </owl:onProperty>
              <owl:hasValue>
                <Hair rdf:ID="Black"/>
              </owl:hasValue>
            </owl:Restriction>
            <owl:Restriction>
              <owl:onProperty rdf:resource="#hasHairColor"/>
              <owl:hasValue>
                <Hair rdf:ID="Red"/>
              </owl:hasValue>
            </owl:Restriction>
          </owl:unionOf>
        </owl:Class>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

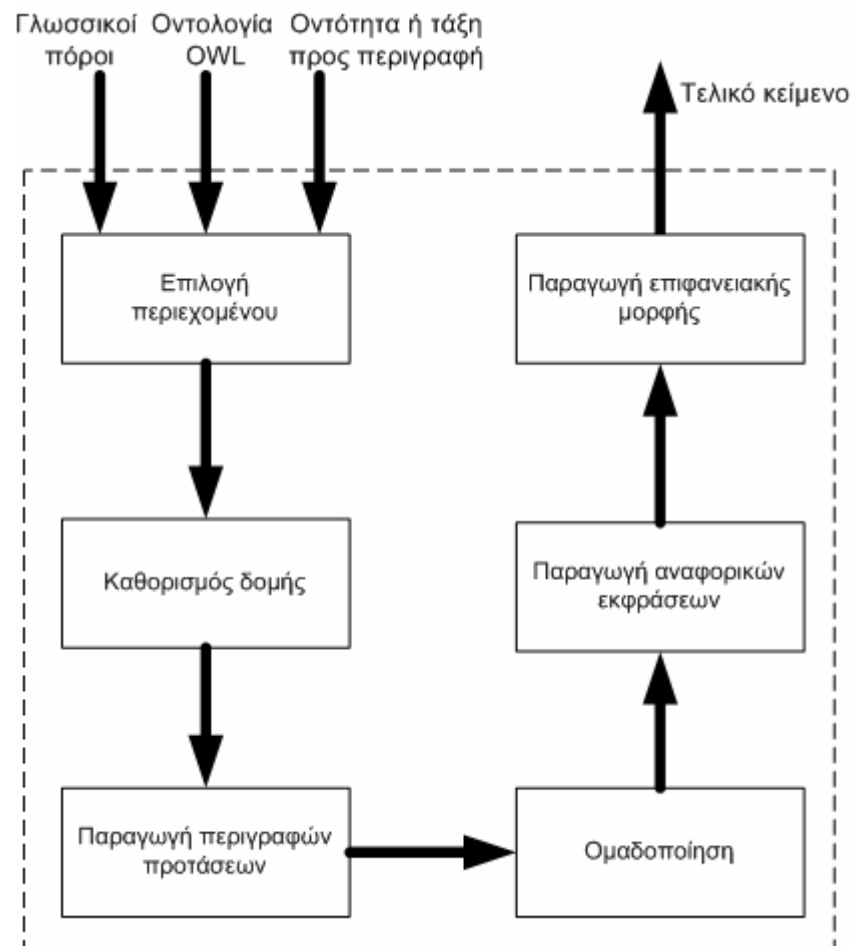
```

```
        </owl:hasValue>
      </owl:Restriction>
    </owl:unionOf>
  </owl:Class>
</owl:intersectionOf>
</owl:Class>
</owl:equivalentClass>
</owl:Class>
```


4. Το σύστημα ΠΦΓ της εργασίας

4.1 Εισαγωγή

Σε αυτό το κεφάλαιο περιγράφεται το σύστημα ΠΦΓ που υλοποιήθηκε και το οποίο ακολουθεί τη συνήθη αρχιτεκτονική των συστημάτων ΠΦΓ που παρουσιάστηκε στο κεφάλαιο 2, με τη διαφορά ότι δεν περιλαμβάνει μοντέλα χρηστών ούτε ιστορικό αλληλεπίδρασης (αυτά θα προστεθούν σε μελλοντικές εκδόσεις του συστήματος). Η αρχιτεκτονική του συστήματος που δημιουργήθηκε παρουσιάζεται αναλυτικότερα στην Εικόνα 2.



Εικόνα 2: Αρχιτεκτονική του συστήματος

Το σύστημα παράγει κείμενα σε δύο γλώσσες, τα ελληνικά και τα αγγλικά, έχοντας ως είσοδο κάθε φορά μια οντολογία σε OWL DL (ή OWL Lite),

γλωσσικούς πόρους και τον κωδικό της οντότητας ή της τάξης της οντολογίας που πρέπει να περιγραφεί. Τα κείμενα που παράγονται αποτελούν περιγραφές αντικειμένων και εννοιών που κωδικοποιούνται-περιγράφονται από οντότητες (Instances) και τάξεις (Classes) της γλώσσας OWL.

4.2 Επιλογή περιεχομένου

Σε αυτό το στάδιο επιλέγονται οι πληροφορίες της οντολογίας που θα μεταδοθούν στον αναγνώστη του κειμένου. Όπως έχουμε ήδη αναφέρει, οι στοιχειώδεις πληροφορίες που επιλέγονται από τη μονάδα καθορισμού περιεχομένου ονομάζονται συνήθως «μηνύματα».

Στην περίπτωση που έχει επιλεγεί προς περιγραφή μια οντότητα X , τα μηνύματα που επιλέγονται από το σύστημα αυτής της εργασίας είναι μια σειρά από δηλώσεις της μορφής $X P Y$, όπου P είναι ιδιότητα της X και Y είναι οντότητα (για ιδιότητες ObjectProperty) ή μέλος τύπου δεδομένων (για ιδιότητες DatatypeProperty) που αποτελεί τιμή της P . Επιλέγεται επίσης ως μήνυμα μια ακόμα δήλωση που καθορίζει σε ποια τάξη ανήκει η X .

Παρακάτω δίνεται η αναπαράσταση της οντότητας *BancroftChardonnay* (της οντολογίας <http://www.w3.org/TR/2003/CR-owl-guide-20030818/wine>) σε OWL. Η πρώτη γραμμή δηλώνει ότι η οντότητα ανήκει στην τάξη Chardonnay και ότι ο κωδικός της είναι *BancroftChardonnay*. Οι επόμενες πέντε γραμμές ορίζουν τις τιμές των ιδιοτήτων της οντότητας (π.χ. η τιμή της ιδιότητας *locatedIn* είναι η οντότητα με κωδικό *NapaRegion*).

```
<Chardonnay rdf:ID="BancroftChardonnay">
  <locatedIn rdf:resource="#NapaRegion" />
  <hasMaker rdf:resource="#Bancroft" />
  <hasSugar rdf:resource="#Dry" />
  <hasFlavor rdf:resource="#Moderate" />
  <hasBody rdf:resource="#Medium" />
</Chardonnay>
```

Το αποτέλεσμα της επιλογής περιεχομένου, δηλαδή τα μηνύματα για την οντότητα *BancroftChardonnay* παρουσιάζονται στο αρχείο XML της εικόνας που ακολουθεί. Το αρχείο αυτό περιέχει επίσης α) πληροφορίες που αφορούν τον τρόπο με τον οποίο θα μετατραπούν τα μηνύματα σε φυσική γλώσσα (ποιο σχεδιάτυπο θα χρησιμοποιηθεί σε κάθε μήνυμα) και β) πληροφορίες που αφορούν τη σειρά με την οποία θα εμφανιστούν τα μηνύματα στο τελικό κείμενο. Οι πληροφορίες (α) και (β) προστίθενται κατά τη διάρκεια του καθορισμού δομής και της παραγωγής περιγραφών προτάσεων, που περιγράφονται στις επόμενες υπο-ενότητες.

```
---Content Selection---
<?xml version="1.0" encoding="ISO-8859-7"?>
<owlMsgs INST-Name="http://www.w3.org/TR/2003/CR-owl-guide-
20030818/wine#BancroftChardonnay">
<hasBody Order="3" Templ="templ1"
Val="http://www.w3.org/TR/2003/CR-owl-guide-
20030818/wine#Medium"/>
<hasFlavor Order="3" Templ="templ1"
Val="http://www.w3.org/TR/2003/CR-owl-guide-
20030818/wine#Moderate"/>
<hasSugar Order="2" Templ="templ1"
Val="http://www.w3.org/TR/2003/CR-owl-guide-
20030818/wine#Dry"/>
<hasMaker Order="1" Templ="templ1"
Val="http://www.w3.org/TR/2003/CR-owl-guide-
20030818/wine#Bancroft"/>
<type Order="0" Templ="" Val="http://www.w3.org/TR/2003/CR-owl-
guide-20030818/wine#Chardonnay"/>
<locatedIn Order="4" Templ="templ2"
Val="http://www.w3.org/TR/2003/CR-owl-guide-
20030818/wine#NapaRegion"/>
</owlMsgs>
```

Η επιλογή περιεχομένου είναι κάπως τετριμμένη στην παρούσα έκδοση του συστήματος, αφού επιλέγονται όλα τα μηνύματα που σχετίζονται με την

οντότητα προς περιγραφή. Μελλοντικές επεκτάσεις του συστήματος θα μπορούσαν να χρησιμοποιούν μοντέλα χρηστών, ώστε να επιλέγονται μόνο μηνύματα που είναι πιθανόν ο χρήστης να βρει ενδιαφέροντα, που δεν έχουν ήδη μεταδοθεί σε αυτόν μέσω προηγούμενων περιγραφών κλπ. (βλ. κεφάλαιο 5).

Στην περίπτωση που έχει επιλεγεί προς περιγραφή μια τάξη X , τα μηνύματα είναι δηλώσεις της OWL που καθορίζουν με ποιες τάξεις είναι ισοδύναμη η X και ποιες τάξεις αποτελούν υπερτάξεις της (ανώνυμες ή μη). Πιο συγκεκριμένα, κάθε μήνυμα είναι μια δήλωση *subClassOf*, *equivalentClass*. Παρακάτω δίνεται ο ορισμός της τάξης *PinotNoir* (από την ίδια οντολογία) σε OWL, ο οποίος περιέχει δύο δηλώσεις. Η πρώτη καθορίζει ότι η τάξη *PinotNoir* είναι υποτάξη μιας ανώνυμης τάξης που περιέχει τα αντικείμενα που έχουν κόκκινο χρώμα και η δεύτερη καθορίζει ότι η τάξη *PinotNoir* είναι ισοδύναμη με μια τάξη που περιέχει τα κρασιά τα οποία φτιάχνονται από σταφύλια *Pinot Noir* και φτιάχνονται από το πολύ 1 είδος σταφυλιού.

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

<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="#hasColor" />
<owl:hasValue rdf:resource="#Red" />
</owl:Restriction>
</rdfs:subClassOf>

<owl:equivalentClass>
<owl:Class>
<owl:intersectionOf rdf:parseType="Collection">
<owl:Class rdf:about="#Wine" />
<owl:Restriction>
<owl:onProperty rdf:resource="#madeFromGrape" />
<owl:hasValue rdf:resource="#PinotNoirGrape" />
</owl:Restriction>
<owl:Restriction>
```

```
<owl:onProperty rdf:resource="#madeFromGrape" />
<owl:maxCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
</owl:Restriction>
</owl:intersectionOf>
</owl:Class>
</owl:equivalentClass>
</owl:Class>
```

Παρακάτω δίνεται το αποτέλεσμα της επιλογής περιεχομένου για την τάξη *PinotNoir*. Όπως και προηγουμένως, το παραγόμενο αρχείο XML περιέχει επίσης πληροφορίες για τα σχεδιάσματα που θα πρέπει να χρησιμοποιηθούν και τη σειρά των μηνυμάτων, πληροφορίες που παράγονται σε επόμενα στάδια. Κάθε μήνυμα σημειώνεται με την ετικέτα owlMsg .

```
---Content Selection---
<?xml version="1.0" encoding="ISO-8859-7"?>
<owlMsgs CLS-Name="http://www.w3.org/TR/2003/CR-owl-guide-
20030818/wine#PinotNoir">
<owlMsg>
<subclassOf>
<hasValueRestriction OnProperty="hasColor" Order="3"
Temp1="templ1" Val="http://www.w3.org/TR/2003/CR-owl-guide-
20030818/wine#Red"/>
</subclassOf>
</owlMsg>
<owlMsg>
<equivalentClass>
<intersectionOf>
<Cls>http://www.w3.org/TR/2003/CR-owl-guide-
20030818/wine#Wine</Cls>
<hasValueRestriction OnProperty="madeFromGrape"
Order="1" Temp1="templ1" Val="http://www.w3.org/TR/2003/CR-owl-
guide-20030818/wine#PinotNoirGrape"/>
<MaxCardRestriction OnProperty="madeFromGrape" Order="1"
```

```
Templ="templ3" Val="1"/>
</intersectionOf>
</equivalentClass>
</owlMsg>
</owlMsgs>
```

Για την υλοποίηση αυτού του σταδίου χρησιμοποιήθηκε το Jena [17], το οποίο είναι ένα εργαλείο ανοιχτού κώδικα (open source) που παρέχει βιβλιοθήκες για τη δημιουργία εφαρμογών σημασιολογικού ιστού. Το Jena παρέχει, μεταξύ άλλων, μια σειρά από τάξεις που βοηθούν στη δημιουργία, ανάγνωση και επεξεργασία οντολογιών OWL.

4.3 Καθορισμός δομής

Αυτό το στάδιο έχει ως είσοδο το αποτέλεσμα του προηγούμενου σταδίου, δηλαδή μια συλλογή από μηνύματα. Στην περίπτωση που έχει επιλεγεί προς περιγραφή μια οντότητα X , τότε αυτό που γίνεται σε αυτό το στάδιο είναι η ταξινόμηση των μηνυμάτων με βάση τη σειρά που έχει οριστεί για κάθε ιδιότητα P . Η σειρά αυτή δηλώνεται σε ένα αρχείο XML του οποίου η δομή φαίνεται στο παράδειγμα της επόμενης εικόνας. Ο ορισμός της δομής και των επιτρεπόμενων περιεχομένων αυτών των αρχείων δίνεται σε XML Schema στο Παράρτημα 1.

```
<?xml version="1.0" encoding="UTF-8"?>
<Ordering>
  <Order property="hasMaker">1</Order>
  <Order property="madeFromGrape">1</Order>
  <Order property="hasSugar">2</Order>
  <Order property="hasColor">3</Order>
  <Order property="hasBody">3</Order>
  <Order property="hasFlavor">3</Order>
  <Order property="locatedIn">4</Order>
  <Order property="madeIntoWine">4</Order>
```

```
<Order property="producesWine">4</Order>
<Order property="yearValue">4</Order>
<Order property="hasVintageYear">4</Order>
<Order property="hasFood">4</Order>
<Order property="hasDrink">4</Order>
<Order property="madeFromFruit">4</Order>
<Order property="course">4</Order>
<Order property="adjacentRegion">4</Order>
<Order property="hasWineDescriptor">4</Order>
</Ordering>
```

Για κάθε ιδιότητα έχει οριστεί μια σειρά (order), η οποία είναι ένας θετικός ακέραιος. Αν ένα μήνυμα A έχει μικρότερη σειρά από ένα άλλο B, τότε το A θα εμφανιστεί στο τελικό κείμενο πριν από το B. Αν έχουν την ίδια σειρά, το σύστημα επιλέγει τυχαία κάθε φορά ποιο μήνυμα θα εμφανιστεί πρώτο. Το μήνυμα που περιέχει τη δήλωση που καθορίζει ότι η οντότητα X ανήκει σε μια τάξη Y θεωρείται ότι έχει πάντα ως σειρά το 0 και επομένως προηγείται όλων των άλλων μηνυμάτων στο τελικό κείμενο. Αυτό γίνεται διότι συνήθως όταν περιγράφουμε ένα αντικείμενο, πρώτα αναφέρουμε την κατηγορία του και ύστερα τα υπόλοιπα χαρακτηριστικά του. Για παράδειγμα, το κείμενο «*This is a Chardonnay. It is made in the Bancroft winery and it is located in the Napa region*» είναι συνηθέστερο και φυσικότερο του «*This is made in the Bancroft winery. It is a Chardonnay and it is located in the Napa region*».

Στην περίπτωση που έχει επιλεγεί προς περιγραφή μια τάξη, η ταξινόμηση των μηνυμάτων δεν είναι το ίδιο εύκολο να γίνει όπως στην προηγούμενη περίπτωση, καθώς τα μηνύματα είναι αρκετά πολύπλοκα και περιέχουν το κάθε ένα σύνθετες δομές της γλώσσας OWL. Για παράδειγμα, ένα μήνυμα μπορεί να είναι μια δήλωση *owl:equivalentClass* που να περιέχει διάφορους περιορισμούς και επώνυμες τάξεις (όπως η τάξη *PinotNoir* παραπάνω). Οπότε κάθε μήνυμα αποτελείται από υπο-μηνύματα τα οποία θα μετατραπούν σε φυσική γλώσσα και για να γίνει η ταξινόμηση των υπο-μηνυμάτων πρέπει πρώτα να εξαχθούν από τα σύνθετα μηνύματα.

4.4 Παραγωγή περιγραφών προτάσεων

Σε αυτό το στάδιο κάθε μήνυμα μετατρέπεται σε μια περιγραφή προτάσεως. Όπως αναφέρθηκε στην ενότητα 2.2.2, στο απλοϊκό σύστημα αυτής της εργασίας η μετατροπή γίνεται χρησιμοποιώντας σχεδιάσματα (templates).

4.4.1 Σχεδιάσματα

Κάθε σχεδιάσμα του συστήματος της εργασίας είναι μια σειρά από θέσεις (slots) που οι τιμές τους απλά συνδέονται για να σχηματίσουν μια πρόταση. Τα σχεδιάσματα ορίζονται κατά την προσαρμογή του συστήματος στην οντολογία μιας νέας θεματικής περιοχής.

Αρχικά, θα παρουσιαστεί η δομή των σχεδισμάτων που χρησιμοποιούνται για να μετατραπούν σε φυσική γλώσσα τα μηνύματα που προκύπτουν όταν επιλέγεται προς περιγραφή μια οντότητα X . Τα μηνύματα αυτά έχουν τη μορφή $X P Y$ (βλ. ενότητα 4.2). (Για το μήνυμα που εκφράζει την τάξη της X χρησιμοποιείται ένα ειδικό, προκαθορισμένο σχεδιάσμα του συστήματος, που περιγράφεται αργότερα.) Ύστερα θα δούμε πώς μπορούν να οριστούν σχεδιάσματα και για τα πολυπλοκότερα μηνύματα που παράγονται όταν επιλέγονται προς περιγραφή τάξεις, αντί για οντότητες.

Για κάθε ιδιότητα P της οντολογίας πρέπει αναγκαστικά να οριστεί τουλάχιστον ένα σχεδιάσμα ανά γλώσσα, το οποίο καθορίζει πώς μπορεί να εκφραστεί στη συγκεκριμένη γλώσσα ένα μήνυμα $X P Y$. Στην περίπτωση που οριστούν περισσότερα από ένα σχεδιάσματα για την ίδια ιδιότητα και γλώσσα, τότε το σύστημα επιλέγει τυχαία κάθε φορά ποιο θα χρησιμοποιήσει, επιτυγχάνοντας μεγαλύτερη ποικιλία εκφράσεων στα παραγόμενα κείμενα. Μελλοντικές εκδόσεις του συστήματος θα μπορούσαν να συνδέσουν την επιλογή σχεδισμου με τα μοντέλα χρηστών, ώστε να αποφεύγεται η επιλογή σχεδισμων που έχουν ξαναχρησιμοποιηθεί πρόσφατα, να προτιμώνται σχεδιάσματα που αντιστοιχούν σε εκφράσεις που ταιριάζουν περισσότερο στον

τύπο του αναγνώστη (π.χ. απλούστερες εκφράσεις όταν παράγονται κείμενα για παιδιά) κλπ., όπως αναφέρθηκε στην ενότητα 2.2.2.

Κατά τον ορισμό ενός σχεδίου, καθορίζεται με ποιον τρόπο θα πρέπει να συμπληρώνεται κάθε κενή θέση του. Μια κενή θέση μπορεί να συμπληρωθεί από ένα από τα εξής:

- Μια αναφορική έκφραση που αναφέρεται στην οντότητα X , δηλαδή την οντότητα που περιγράφεται από το κείμενο. Η αναφορική έκφραση παράγεται αυτόματα από το σύστημα, αλλά πρέπει κατά τον ορισμό του σχεδίου να προσδιοριστεί η πτώση της.
- Την τιμή Y που έχει η ιδιότητα P στο μήνυμα, αν η ιδιότητα είναι τύπου *DatatypeProperty*.
- Μια αναφορική έκφραση που αναφέρεται στην οντότητα Y , αν πρόκειται για ιδιότητα *ObjectProperty*. Η αναφορική έκφραση παράγεται αυτόματα, αλλά πρέπει κατά τον ορισμό του σχεδίου να προσδιοριστεί η πτώση της.
- Ένας ρηματικός τύπος της φυσικής γλώσσας του σχεδίου, συνοδευόμενος από πληροφορίες για τη φωνή και το χρόνο του. Οι πληροφορίες αυτές χρησιμοποιούνται κατά την ομαδοποίηση.
- Μια συμβολοσειρά που μπορεί να περιέχει οποιαδήποτε έκφραση της φυσικής γλώσσας του σχεδίου. Αν η έκφραση είναι απλά ρηματικός τύπος ή πρόθεση, είναι προτιμότερο να χρησιμοποιηθούν οι προηγούμενες δύο επιλογές αντίστοιχα, που παρέχουν περισσότερες πληροφορίες για την έκφραση στη μονάδα ομαδοποίησης. Θα χρησιμοποιήσουμε ως παράδειγμα την ιδιότητα *hasFlavor*, που είναι τύπου *ObjectProperty*. Ένα αγγλικό σχέδιο που μπορεί να οριστεί για τα μηνύματα της μορφής $X \text{ hasFlavor } Y$ είναι το εξής:

Θέση 1: Αναφορική έκφραση που αναφέρεται στην οντότητα X , στην ονομαστική.

Θέση 2: Ο ρηματικός τύπος «has», ο οποίος είναι σε ενεργητική φωνή και χρόνο ενεστώτα.

Θέση 3: Αναφορική έκφραση που αναφέρεται στην οντότητα Υ, στην αιτιατική.

Θέση 4: Η συμβολοσειρά «flavor».

Με το σχεδιάτυπο αυτό, το μήνυμα *BancroftChardonnay hasFlavor Moderate* μετατρέπεται στην περιγραφή προτάσεως [*ref(nominative, BancroftChardonnay), verb(active, present, «has»), ref(accusative, Moderate), «flavor»*], όπου *BancroftChardonnay* και *Moderate* είναι ονόματα οντοτήτων της OWL και τα *ref(...)* υποδηλώνουν απαιτήσεις για αναφορικές εκφράσεις. Τα μηνύματα που εκφράζουν την τάξη μιας οντότητας X (π.χ. ότι το *BancroftChardonnay* ανήκει στην τάξη *Chardonnay*) μετατρέπονται σε φυσική γλώσσα χρησιμοποιώντας ένα προκαθορισμένο, κοινό σχεδιάτυπο ανά φυσική γλώσσα, το οποίο για τα αγγλικά είναι το εξής :

Θέση 1: Αναφορική έκφραση που αναφέρεται στην οντότητα X, στην ονομαστική.

Θέση 2: Ο ρηματικός τύπος «is», ο οποίος είναι σε ενεργητική φωνή και χρόνο ενεστώτα.

Θέση 3: Η συμβολοσειρά «a» ή «an». Η τελική επιλογή γίνεται στο στάδιο της παραγωγής της επιφανειακής μορφής.

Θέση 4: Αναφορική έκφραση που αναφέρεται στο όνομα της τάξης, στην αιτιατική.

Χρησιμοποιώντας το παραπάνω σχεδιάτυπο, στην περίπτωση της οντότητας *BancroftChardonnay* θα παραχθεί τελικά το κείμενο *Bancroft Chardonnay is a Chardonnay*. Η πληροφορία που παρέχει το συγκεκριμένο κείμενο είναι φυσικά περιττή, γιατί στην περίπτωση αυτή η τάξη της οντότητας είναι προφανής, αφού το όνομα της τάξης («Chardonnay») περιλαμβάνεται στο όνομα της οντότητας («Bancroft Chardonnay»). Αντίστοιχα προβλήματα εμφανίζονταν στο M-PIRO σε προτάσεις της μορφής «Το Αρχαιολογικό Μουσείο της Αθήνας είναι ένα μουσείο.». Σε μελλοντικές εκδόσεις του

συστήματος της εργασίας, που θα περιλαμβάνουν μοντέλα χρηστών, προτάσεις αυτού του είδους θα είναι δυνατόν να παραλείπονται δηλώνοντας ότι τα αντίστοιχα μηνύματα έχουν μηδενικό ενδιαφέρον για όλους τους χρήστες, όπως στο M-PIRO [11].

4.4.2 Κωδικοποίηση των σχεδίουτυπων με την βοήθεια της γλώσσας XSL.

Τα σχεδίουτυπα για τις ιδιότητες της οντολογίας ορίζονται με τη βοήθεια της γλώσσας XSL (Extensible Stylesheet Language) [33], η οποία είναι μια γλώσσα που παρέχει μηχανισμούς α) για τον ορισμό μετασχηματισμών που εφαρμόζονται σε έγγραφα XML και β) για τον ορισμό πληροφοριών για την παρουσίαση εγγράφων XML.

Τα τελευταία χρόνια, αρκετά συστήματα παραγωγής φυσικής γλώσσας [4, 5, 6, 7, 24] έχουν χρησιμοποιήσει τη γλώσσα XSL, διότι ταιριάζει απόλυτα στην σωληνωτή αρχιτεκτονική (pipeline) στην οποία βασίζονται. Όπως είδαμε και στο κεφάλαιο 2, κάθε φάση ενός συστήματος παραγωγής φυσικής γλώσσας έχει ως είσοδο το αποτέλεσμα της προηγούμενης φάσης. Επομένως, αν η είσοδος και η έξοδος των φάσεων ενός συστήματος ΠΦΓ κωδικοποιείται με την βοήθεια XML εγγράφων, τότε κάθε φάση του μπορεί να υλοποιηθεί ορίζοντας κατάλληλους μετασχηματισμούς του XML εγγράφου της εισόδου, σε ένα άλλο XML έγγραφο. Στο σύστημα που αναπτύχθηκε σε αυτή την εργασία η XSL και συγκεκριμένα η υπογλώσσά της XSLT (XSL Transformations), με την οποία είναι δυνατό να οριστούν μετασχηματισμοί (transformations), χρησιμοποιείται μόνο στο στάδιο της παραγωγής περιγραφών προτάσεων. Σε άλλα συστήματα η XSLT χρησιμοποιείται και στις φάσεις της ομαδοποίησης, της παραγωγής αναφορικών εκφράσεων και στην παραγωγή επιφανειακής μορφής.

Για να γίνει κατανοητό το πώς κωδικοποιούνται σε XSLT τα σχεδίουτυπα του συστήματος, θα χρησιμοποιήσουμε ως παράδειγμα το σχεδίουτυπο για τα μηνύματα $X \text{ hasFlavor } Y$ της προηγούμενης ενότητας. Αυτό το σχεδίουτυπο

κωδικοποιείται σε XSLT όπως φαίνεται παρακάτω, χρησιμοποιώντας ένα `template (xsl:template)` της XSL.

```
<xsl:template
match="hasFlavor[@Templ='templ1']|hasValueRestriction[@Templ='
templ1']"
<Msg Order="{@Order}" RestrType="{name()}"
property="hasFlavor">
<Owner case="nominative"/>
<verb tense="present" voice="active">has</verb>
<Filler case="accusative"><xsl:value-of
select="@Val"/></Filler>
<TEXT>flavor</TEXT>
</Msg>
</xsl:template>
```

Ο παραπάνω κώδικας καθορίζει τα εξής:

Αν το αρχείο XML που παρήγαγε η επιλογή περιεχομένου (ενότητα 4.2) περιέχει ένα μήνυμα με ιδιότητα *hasFlavor* και το σύστημα έχει επιλέξει το σχεδιάτυπο με κωδικό *templ1* για να μετατρέψει το μήνυμα σε φυσική γλώσσα (δηλαδή αν το αρχείο XML περιέχει ένα στοιχείο `<hasFlavor ... Templ='templ1'/>`), τότε αντικατάστησε το μήνυμα στο XML έγγραφο με ένα στοιχείο *Msg* που περιέχει τις θέσεις (slots) από τις οποίες αποτελείται το σχεδιάτυπο. Η ετικέτα *Owner* υποδηλώνει μια αναφορική έκφραση που θα αναφέρεται στην οντότητα *X* (την οντότητα που θα περιγράφει το κείμενο). Η ετικέτα *verb* δείχνει ότι η αντίστοιχη θέση περιέχει ρηματικό τύπο και παρέχει πληροφορίες για τη φωνή και το χρόνο του τύπου. Η ετικέτα *Filler* υποδηλώνει ότι στην αντίστοιχη θέση θα πρέπει να μπει η τιμή *Y* της ιδιότητας (αν η πρόκειται για *DatatypeProperty*) ή μια αναφορική έκφραση που να αναφέρεται στην οντότητα *Y* (αν πρόκειται για *ObjectProperty*). Τέλος η ετικέτα *TEXT* δείχνει ότι στην αντίστοιχη θέση θα πρέπει να μπει η συμβολοσειρά «flavor». Το τμήμα του κώδικα `<xsl:value-of select="@Val"/>` χρησιμοποιείται για την αντιγραφή της τιμής της ιδιότητας *hasFlavor* από το XML έγγραφο στο οποίο εφαρμόζεται ο μετασχηματισμός.

Με τη χρήση του σχεδίου για την ιδιότητα *hasFlavor* καθώς και με τη χρήση ανάλογων σχεδίων για τις υπόλοιπες ιδιότητες, το XML έγγραφο το οποίο δέχεται ως είσοδο η φάση παραγωγής περιγραφών προτάσεων μετασχηματίζεται στο παρακάτω XML έγγραφο.

```
---Lexicalisation---
<?xml version="1.0" encoding="ISO-8859-7"?>
<owlMsgs INST-Name="http://www.w3.org/TR/2003/CR-owl-guide-
20030818/wine#BancroftChardonnay">
<Msg Order="0">
<Owner/>
<IS_A/>
<Filler case="accusative">http://www.w3.org/TR/2003/CR-owl-guide-
20030818/wine#Chardonnay</Filler>
</Msg>
<Msg Order="2" property="hasSugar">
<Owner case="nominative"/>
<verb tense="present" voice="active">is</verb>
<Filler case="accusative">http://www.w3.org/TR/2003/CR-owl-guide-
20030818/wine#Dry</Filler>
</Msg>
<Msg Order="3" property="hasFlavor">
<Owner case="nominative"/>
<verb tense="present" voice="active">has</verb>
<Filler case="accusative">http://www.w3.org/TR/2003/CR-owl-guide-
20030818/wine#Moderate</Filler>
<TEXT>flavor</TEXT>
</Msg>
<Msg Order="4" property="locatedIn">
<Owner case="nominative"/>
<verb tense="present" voice="passive">is located</verb>
<TEXT>in</TEXT>
<Filler case="accusative">http://www.w3.org/TR/2003/CR-owl-guide-
20030818/wine#NapaRegion</Filler>
</Msg>
<Msg Order="5" property="hasBody">
<Owner case="nominative"/>
<verb tense="present" voice="active">has</verb>
<Filler case="accusative">http://www.w3.org/TR/2003/CR-owl-guide-
20030818/wine#Medium</Filler>
```

4. Το σύστημα ΠΦΓ της εργασίας

```
<TEXT>body</TEXT>
</Msg>
<Msg Order="5" property="hasMaker">
  <Owner case="nominative"/>
  <verb tense="present" voice="passive">is made</verb>
  <TEXT>in</TEXT>
  <Filler case="accusative">http://www.w3.org/TR/2003/CR-owl-guide-
  20030818/wine#Bancroft</Filler>
  <TEXT>winery</TEXT>
</Msg>
</owlMsgs>
```

Τα ονόματα των σχεδίουτυπων για όλες τις ιδιότητες αποθηκεύονται σε ένα αρχείο XML του οποίου η δομή φαίνεται παρακάτω.

```
<?xml version="1.0" encoding="UTF-8"?>
<AllTemplates language="en">
  <templates property="hasColor">
    <templ TemplName="templ1" used="T"/>
    <templ TemplName="templ2" used="T"/>
    <templ TemplName="templ3" used="T"/>
    <templ TemplName="templ4" used="T"/>
    <templ TemplName="templ5" used="T"/>
  </templates>
  <templates property="hasBody">
    <templ TemplName="templ1" used="T"/>
    <templ TemplName="templ2" used="T"/>
    <templ TemplName="templ3" used="T"/>
    <templ TemplName="templ4" used="T"/>
    <templ TemplName="templ5" used="T"/>
  </templates>
  <templates property="hasFlavor">
    <templ TemplName="templ1" used="T"/>
    <templ TemplName="templ2" used="F"/>
    <templ TemplName="templ3" used="T"/>
    <templ TemplName="templ4" used="T"/>
    <templ TemplName="templ5" used="T"/>
  </templates>
</AllTemplates>
```

Για κάθε σχεδιάτυπο αποθηκεύεται:

- Ο κωδικός του (*TemplName*).
- Μια ένδειξη (*used*) η οποία δηλώνει αν το σχεδιάτυπο είναι ενεργοποιημένο και επομένως μπορεί να χρησιμοποιηθεί από το σύστημα ΠΦΓ.

4.4.3 Σχεδιάτυπα για τη μετατροπή σε φυσική γλώσσα μηνυμάτων που περιέχουν περιορισμούς ιδιοτήτων.

Όπως είδαμε στο κεφάλαιο 2, η OWL επιτρέπει τον ορισμό κάποιων περιορισμών στον τρόπο με τον οποίο οι ιδιότητες μπορούν να χρησιμοποιηθούν από τις οντότητες μιας τάξης. Συγκεκριμένα, η OWL περιέχει περιορισμούς για τον πληθάρημο των τιμών μιας ιδιότητας (*owl:maxCardinality*, *owl:minCardinality*, *owl:cardinality*), αλλά και περιορισμούς για την ίδιες τις τιμές των ιδιοτήτων (*owl:allValuesFrom*, *owl:someValuesFrom*, *owl:hasValue*). Όταν επιλέγεται προς περιγραφή μια τάξη της οντολογίας, αντί για μια οντότητα, αυτοί οι περιορισμοί αποτελούν κομμάτια σύνθετων μηνυμάτων (βλ. ενότητα 4.2) που πρέπει να μετατραπούν σε φυσική γλώσσα από το σύστημα. Αυτό επιτυγχάνεται χρησιμοποιώντας τα ίδια σχεδιάτυπα με εκείνα που χρησιμοποιούνται στην περίπτωση των απλών μηνυμάτων, δηλαδή των μηνυμάτων που προκύπτουν όταν επιλέγονται προς περιγραφή οντότητες. Προκειμένου να ξεπεραστεί ένα πρόβλημα που προκύπτει από τη χρήση των ίδιων σχεδιάτυπων και το οποίο περιγράφεται παρακάτω, μια θέση σχεδιάτυπου είναι δυνατόν να συμπληρωθεί και με μια αναφορά σε μια τάξη της οντολογίας. Όπως θα δούμε σε επόμενες ενότητες, κάθε τάξη συσχετίζεται με ένα ουσιαστικό (ή γενικότερα ονοματικό σύνολο χωρίς άρθρο) του λεξικού. Η αναφορά σε μια τάξη είναι τελικά το ουσιαστικό (ή ονοματικό σύνολο) που έχει συσχετιστεί με την τάξη, στην κατάλληλη μορφή (ενικός ή πληθυντικός).

Θα χρησιμοποιήσουμε την ιδιότητα *madeFromGrape* ως παράδειγμα. Στην εικόνα που ακολουθεί δίνεται ένα σχεδιάγραμμα για την ιδιότητα *madeFromGrape*, το οποίο χρησιμοποιείται για τη μετατροπή απλών μηνυμάτων της μορφής *X P Y* σε προτάσεις.

Θέση 1: Αναφορική έκφραση που αναφέρεται στην οντότητα *X*, στην ονομαστική.

Θέση 2: Ο παθητικός ενεστωτικός ρηματικός τύπος «is made».

Θέση 3: Η συμβολοσειρά «from».

Θέση 4: Αναφορική έκφραση που αναφέρεται στην οντότητα *Y*, στην αιτιατική.

Θέση 5: Αναφορά στην τάξη *Grape*.

Για παράδειγμα από ένα τέτοιο σχεδιάγραμμα και το μήνυμα *ChateauDYchemSauterne madeFromGrape SemillonGrape* θα παραχθεί τελικά (μετά και την παραγωγή των αναφορικών εκφράσεων) μια πρόταση σαν την «Chateau DYchem Sauterne is made from Semillon grape» (το *ChateauDYchemSauterne* είναι μια οντότητα). Το ίδιο ακριβώς σχεδιάγραμμα όμως μπορεί να χρησιμοποιηθεί και για την μετατροπή ενός *owl:hasValue* περιορισμού της ιδιότητας *madeFromGrape* σε μια πρόταση. Για παράδειγμα στην τάξη *PinotNoir* (τμήμα του ορισμού της οποίας επαναλαμβάνεται παρακάτω) ο περιορισμός *Restriction* θα μετατραπεί σε μια πρόταση της μορφής «Pinot Noir is made from Pinot Noir grape».

```
<owl:Class rdf:ID="PinotNoir">
...
<owl:Restriction>
<owl:onProperty rdf:resource="#madeFromGrape" />
<owl:hasValue rdf:resource="#PinotNoirGrape" />
</owl:Restriction>
...
</owl:Class>
```


Στην περίπτωση αυτή το υπο-μήνυμα που πρέπει να εκφραστεί σε φυσική γλώσσα είναι το *PinotNoir madeFromGrape PinotNoirGrape*, δηλαδή έχει πάλι τη μορφή $X P Y$, αλλά τώρα το X είναι τάξη και όχι οντότητα. Αυτό δεν δημιουργεί πρόβλημα, γιατί το σύστημα γνωρίζει ότι η πρώτη θέση του σχεδίου αντιστοιχεί στο X (είτε αυτό είναι οντότητα είτε είναι τάξη). Γενικά για τη μετατροπή περιορισμών *hasValue* σε φυσική γλώσσα δεν χρειάζεται να οριστούν επιπλέον σχεδίου. Το σύστημα χρησιμοποιεί τα ίδια σχεδίου με εκείνα που χρησιμοποιεί στα αντίστοιχα μηνύματα $X P Y$ που προκύπτουν όταν επιλέγεται προς περιγραφή μια οντότητα. Τα ίδια σχεδίου χρησιμοποιούνται για να εκφραστούν σε φυσική γλώσσα και περιορισμοί άλλων ειδών. Για παράδειγμα με το παραπάνω σχεδίου ο παρακάτω περιορισμός της τάξης *PinotNoir*

```
<owl:Restriction>
<owl:onProperty rdf:resource="#madeFromGrape" />
<owl:maxCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
</owl:Restriction>
```

θα μετατραπεί στην πρόταση «Pinot Noir is made from at most 1 grape». Το σύστημα για να παραγάγει την παραπάνω πρόταση απλά προσθέτει την έκφραση «at most» πριν την τιμή του περιορισμού. Αν η τιμή του περιορισμού ήταν 2, θα χρησιμοποιούσε τον πληθυντικό του ουσιαστικού («at most 2 grapes») και αν επρόκειτο για περιορισμό *minCardinality* ή *maxCardinality*, θα χρησιμοποιούσε τις εκφράσεις «at least» ή «exactly». Αντίστοιχες εκφράσεις παράγονται στα ελληνικά.

Ομοίως τα ίδια σχεδίου είναι δυνατό να χρησιμοποιηθούν για την παραγωγή προτάσεων από περιορισμούς *owl:someValuesFrom* και *owl:allValuesFrom*. Για παράδειγμα ο περιορισμός της τάξης *bus_driver*:

```
<owl:Restriction>
<owl:onProperty>
```

```
<owl:ObjectProperty rdf:about="#drives" />
</owl:onProperty>
<owl:someValuesFrom rdf:resource="#bus" />
</owl:Restriction>
```

μπορεί να μετατραπεί με το προηγούμενο σχεδιάγραμμα στη πρόταση “*A bus driver drives at least 1 bus*”. Στην περίπτωση αυτή, η τάξη *bus* έχει συσχετιστεί στο λεξικό με το αγγλικό ουσιαστικό «bus» και αυτό επιτρέπει στο σύστημα να παραγάγει αυτόματα την έκφραση «at least 1 bus», δεδομένου ότι πρόκειται για περιορισμό *someValuesFrom*.

Στην περίπτωση των ετικετών *owl:oneOf* και *owl:unionOf* χρησιμοποιούνται οι λέξεις “or” και “ή”. Για παράδειγμα για τον περιορισμό:

```
<owl:Restriction>
  <owl:onProperty rdf:resource="#hasFlavor" />
  <owl:allValuesFrom>
    <owl:Class>
      <owl:oneOf rdf:parseType="Collection">
        <owl:Thing rdf:about="#Moderate" />
        <owl:Thing rdf:about="#Strong" />
      </owl:oneOf>
    </owl:Class>
  </owl:allValuesFrom>
</owl:Restriction>
```

παράγεται στα ελληνικά το κείμενο “Έχει μέτριο άρωμα ή ισχυρό άρωμα” και αντίστοιχα στα αγγλικά “It has moderate flavor or strong flavor”.

4.4.4 Λεξικό

Τα λεξικά του συστήματος για την ελληνική και αγγλική γλώσσα περιλαμβάνουν πληροφορίες μόνο για ουσιαστικά ή γενικότερα ονοματικές φράσεις (noun phrases) χωρίς άρθρα. Αποθηκεύονται σε XML αρχεία των

οποίων η μορφή φαίνεται στην παρακάτω εικόνα. Το Παράρτημα 1 ορίζει τη δομή και τα επιτρεπτά περιεχόμενα των αρχείων σε XML Schema.

```
<?xml version="1.0" encoding="UTF-8"?>
<Lexicon language="en">
<OntClass name="http://www.w3.org/TR/2003/CR-owl-guide-
20030818/wine#Wine">
<NP gender="nonpersonal" num="singular" countable="T" >
<singular>wine</singular>
<plural>wines</plural>
</NP>
</OntClass>
<OntInd name="http://www.w3.org/TR/2003/CR-owl-guide-
20030818/wine#Region">
<NP gender="nonpersonal" num="singular" countable="T" >
<singular>region</singular>
<plural>regions</plural>
</NP>
</OntInd>
</Lexicon>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<Lexicon language="gr">
<OntClass name="http://www.w3.org/TR/2003/CR-owl-guide-
20030818/wine#Wine">
<NP gender="neuter" num="singular" countable="T" >
<singular>
<nominative>κρασί</nominative>
<genitive>κρασιού</genitive>
<accusative>κρασί</accusative>
</singular>
<plural>
<nominative>κρασιά</nominative>
<genitive>κρασιών</genitive>
<accusative>κρασιά</accusative>
```

```
</plural>
</NP>
</OntClass>
<OntClass name="http://www.w3.org/TR/2003/CR-owl-guide-
20030818/wine#Region">
<NP gender="feminine" num="singular" countable="T" >
<singular>
<nominative>περιοχή</nominative>
<genitive>περιοχής</genitive>
<accusative>περιοχή</accusative>
</singular>
<plural>
<nominative>περιοχές</nominative>
<genitive>περιοχών</genitive>
<accusative>περιοχές</accusative>
</plural>
</NP>
</OntClass>
```

Κάθε τάξη ή οντότητα της οντολογίας συσχετίζεται με ένα ουσιαστικό ή ονοματική φράση που μπορεί να χρησιμοποιηθεί για να αναφερθούμε σε αυτή. Για κάθε ουσιαστικό ή ονοματική φράση αποθηκεύονται για τα αγγλικά οι εξής πληροφορίες:

- Η μορφή του ουσιαστικού ή της ονοματικής φράσης στον ενικό(singular) και τον πληθυντικό (plural).
- Το φυσικό γένος (ετικέτα gender).
- Ο αριθμός (ενικός ή πληθυντικός) που θα πρέπει να χρησιμοποιηθεί για να αναφερθούμε σε αυτή (ετικέτα num). Για παράδειγμα, το «trousers» χρησιμοποιείται πάντα στον πληθυντικό.
- Μια μεταβλητή που δηλώνει αν το ουσιαστικό είναι μετρήσιμο ή όχι (ετικέτα countable). Για παράδειγμα, δεν μπορούμε να πούμε «a water».

Ανάλογα, για κάθε ουσιαστικό ή ονοματική φράση αποθηκεύονται για τα ελληνικά οι εξής πληροφορίες:

- Η ονομαστική, γενική και αιτιατική (nominative, genitive, accusative) του ενικού και πληθυντικού
- Το γραμματικό γένος (ετικέτα gender).
- Ο αριθμός (ενικός ή πληθυντικός) του ουσιαστικού που θα πρέπει να χρησιμοποιηθεί σε αυτή. (ετικέτα num).
- Μια μεταβλητή που δηλώνει αν το ουσιαστικό είναι μετρήσιμο (ετικέτα countable).

4.4.5 Ομαδοποίηση

Όπως ήδη έχει αναφερθεί η λειτουργία της ομαδοποίησης (aggregation) είναι από τις πιο σημαντικές σε ένα σύστημα παραγωγής φυσικής γλώσσας, καθώς βελτιώνει την αναγνωσιμότητα του τελικού κειμένου. Η λειτουργία αυτή στο σύστημα που αναπτύχθηκε πραγματοποιείται με τη χρήση κανόνων. Πολλοί από αυτούς τους κανόνες χρησιμοποιούνται και στο σύστημα ΠΦΓ M-PIRO [2].

4.5.1 1^{ος} Κανόνας ομαδοποίησης: «Περιγραφή τάξης και ιδιότητας»

Αυτός ο κανόνας χρησιμοποιείται για να ομαδοποιήσει δύο προτάσεις όπου η μια ορίζει την τάξη μιας οντότητας και η άλλη παρέχει την τιμή μιας ιδιότητας της οντότητας. Ο κανόνας εφαρμόζεται μόνο όταν η φωνή του ρηματικού τύπου της πρότασης που παρέχει την τιμή της ιδιότητας είναι η παθητική, ανεξάρτητα από το χρόνο του ρηματικού τύπου. Το υποκείμενο και το βοηθητικό ρήμα της πρότασης που παρέχει την τιμή της ιδιότητας παραλείπονται. Ένα παράδειγμα εφαρμογής του κανόνα είναι το εξής:

Bancroft Chardonnay is a Chardonnay. It is made in Bancroft .

=>

Bancroft Chardonnay is a Chardonnay made in Bancroft .

Μια παραλλαγή του παραπάνω κανόνα χρησιμοποιεί την αγγλική άνω τελεία και εφαρμόζεται όταν η πρόταση η οποία ορίζει την τάξη μιας οντότητας ενώνεται με δύο ή περισσότερες άλλες προτάσεις που παρέχουν τιμές ιδιοτήτων της ίδιας οντότητας και μπορούν να ομαδοποιηθούν με κάποιον από τους επόμενους. Ένα παράδειγμα, στο οποίο μετά την άνω τελεία χρησιμοποιείται ο κανόνας απλής σύνδεσης της επόμενης ενότητας, είναι το παρακάτω:

Bancroft Chardonnay is a Chardonnay. It is made in Bancroft . It has moderate flavor.

=>

Bancroft Chardonnay is a Chardonnay; it is made in Bancroft and has moderate flavor.

Στα ελληνικά κείμενα ο κανόνας εφαρμόζεται χρησιμοποιώντας μια αναφορική αντωνυμία στο κατάλληλο γένος (“ο οποίος”, “η οποία”, “το οποίο”) ή εναλλακτικά την αντωνυμία “που”. Ένα παράδειγμα εφαρμογής του κανόνα φαίνεται παρακάτω.

Το Bancroft Chardonnay είναι ένα Chardonnay. Φτιάχνεται στο Bancroft.

=>

Το Bancroft Chardonnay είναι ένα Chardonnay που φτιάχνεται στο Bancroft.

ή

Το Bancroft Chardonnay είναι ένα Chardonnay το οποίο φτιάχνεται στο Bancroft.

4.5.2 2^{ος} Κανόνας ομαδοποίησης: “Απλή σύζευξη”.

Ο συνηθέστερος κανόνας ομαδοποίησης που χρησιμοποιείται στα συστήματα ΠΦΓ είναι η απλή σύζευξη δύο ή περισσότερων προτάσεων, όπου η κάθε μια

παρέχει την τιμή κάποιας ιδιότητας της ίδιας οντότητας ή τάξης (στη δεύτερη περίπτωση μέσω περιορισμών owl:hasValue).

Στα αγγλικά, ο κανόνας αυτός ενώνει δύο ή περισσότερες προτάσεις που έχουν το ίδιο υποκείμενο.

It is dry. It has medium body.

=>

It is dry and has medium body.

Στα ελληνικά, ο κανόνας χρησιμοποιείται με τον ίδιο τρόπο χρησιμοποιώντας τη λέξη “και”.

Έχει μέτριο άρωμα. Προέρχεται από την περιοχή Napa.

=>

Έχει μέτριο άρωμα και προέρχεται από την περιοχή Napa.

Όταν συνδέονται περισσότερες από δύο προτάσεις, τότε η λέξη “and” ή η λέξη “και” χρησιμοποιούνται μόνο για την τελευταία σύνδεση. Οι υπόλοιπες συνδέσεις γίνονται με τη χρήση κόμματος. Για παράδειγμα:

Bancroft Chardonnay is a Chardonnay, it is dry, it has moderate flavor, and it is located in the Napa region.

Το Bancroft Chardonnay είναι ένα Chardonnay, είναι ξηρό, η υφή του είναι μέτρια και έχει μέτριο άρωμα.

4.5.3 3^{ος} Κανόνας ομαδοποίησης: “Κοινό υποκείμενο και ρήμα”

Ο κανόνας αυτός εφαρμόζεται στην περίπτωση που δύο ή περισσότερες γειτονικές προτάσεις έχουν κοινό υποκείμενο και ρήμα. Το κοινό υποκείμενο και ρήμα εμφανίζεται μόνο μια φορά στην πρόταση που προκύπτει από την ένωση των αρχικών προτάσεων. Για παράδειγμα στα αγγλικά:

It has medium body. It has moderate flavor.

=>

It has medium body and moderate flavor.

Ανάλογα στα ελληνικά μπορεί να εφαρμοστεί ο ίδιος κανόνας, αν και το υποκείμενο μπορεί να εννοείται. Για παράδειγμα:

Έχει μέτρια υφή. Έχει ευχάριστο άρωμα

=>

Έχει μέτρια υφή και ευχάριστο άρωμα

Στις περιπτώσεις όπου μπορεί να εφαρμοστεί ο 3ος κανόνας, μπορεί να εφαρμοστεί και ο 2ος («Έχει μέτρια υφή και έχει ευχάριστο άρωμα.»). Προτιμάται όμως ο 3ος, γιατί οδηγεί σε συντομότερα κείμενα.

4.5.4 4^{ος} Κανόνας ομαδοποίησης: “Ιδιότητα και περιορισμός πληθαιριθμού”

Ο κανόνας εφαρμόζεται όταν έχουμε δύο προτάσεις για την ίδια ιδιότητα μιας τάξης και η μία προέρχεται από περιορισμό πληθαιριθμού, ενώ η άλλη από περιορισμό *owl:hasValue*. Οι δύο προτάσεις μπορούν να ενωθούν όπως φαίνεται στο παρακάτω παράδειγμα. Η πρόταση που έχει προέλθει από τον περιορισμό *owl:hasValue* πρέπει να είναι δεύτερη σε σειρά για να προκύψει αποδεκτό κείμενο. Για παράδειγμα:

It is made from at most 1 grape. It is made from Pinot Noir grape.

=>

It is made from at most 1 grape, Pinot Noir grape.

Ανάλογα στα ελληνικά έχουμε:

Παρασκευάζεται από το πολύ 1 σταφύλι. Παρασκευάζεται από το σταφύλι Cabernet Sauvignon.

=>

Παρασκευάζεται από το πολύ 1 σταφύλι, το σταφύλι Cabernet Sauvignon.

4.5.5 5^{ος} Κανόνας ομαδοποίησης: "Απαλοιφή πλεονασμού"

Ο κανόνας αυτός εφαρμόζεται σε περιπτώσεις όπου υπάρχει πλεονασμός σε εκφράσεις που περιγράφουν τις πιθανές τιμές μιας ιδιότητας. Τέτοιοι πλεονασμοί μπορούν να προκύψουν από την παραγωγή φυσικής γλώσσας από τις ετικέτες *owl:oneOf* και *owl:unionOf*. Για παράδειγμα:

It has moderate flavor or strong flavor.

=>

It has moderate or strong flavor.

Έχει μέτριο άρωμα ή ισχυρό άρωμα.

=>

Έχει μέτριο ή ισχυρό άρωμα

Ο κανόνας αφαιρεί τις πλεονάζουσες λέξεις.

4.6 Παραγωγή αναφορικών εκφράσεων

Μετά την ομαδοποίηση, οι περιγραφές των προτάσεων εισέρχονται στη φάση της παραγωγής αναφορικών εκφράσεων. Όπως έχουμε δει στο κεφάλαιο 2, αυτή η φάση μπορεί να είναι αρκετά περίπλοκη, καθώς πρέπει πρώτα να επιλεχθούν τα χαρακτηριστικά που διαχωρίζουν την οντότητα στην οποία θέλουμε να αναφερθούμε από τις υπόλοιπες και ύστερα να επιλεχθούν οι κατάλληλες λέξεις με τις οποίες θα εκφραστούν αυτά τα χαρακτηριστικά. Όμως στο σύστημα το οποίο αναπτύχθηκε, κάθε φορά παράγεται ένα κείμενο το οποίο περιγράφει μόνο μια οντότητα ή τάξη και όλες οι αναφορικές εκφράσεις αναφέρονται σε αυτήν. Οπότε, δεν απαιτείται η επιλογή των

χαρακτηριστικών που προσδιορίζουν μοναδικά αυτή την οντότητα ή τάξη, αλλά απλά η επιλογή κατάλληλων λέξεων. Πολλά συστήματα παραγωγής φυσικής γλώσσας [15, 16, 17, 36] ακολουθούν τον εξής απλό αλγόριθμο για την παραγωγή αναφορικών εκφράσεων σε τέτοιες περιπτώσεις. Αν αναφερόμαστε σε μια οντότητα για πρώτη φορά και η οντότητα έχει όνομα (π.χ. ένα άγαλμα με κωδικό «exhibit23» μπορεί να έχει το όνομα «Δορυφόρος» στα ελληνικά και «Doryphorus» στα αγγλικά) χρησιμοποιούμε το όνομά της, διαφορετικά μια δεικτική αναφορική έκφραση (π.χ. «this»). Τις επόμενες φορές που αναφερόμαστε στην οντότητα, χρησιμοποιούμε μια κατάλληλη αντωνυμία (π.χ. «it») στα αγγλικά ή κενό υποκείμενο στα ελληνικά. Για παράδειγμα, παρακάτω φαίνεται ένα κείμενο που παράγεται για την οντότητα *BancroftChardonnay*, της οποίας το αγγλικό όνομα είναι «Bancroft Chardonnay».

Bancroft Chardonnay is a Chardonnay, made in Bancroft . It is dry and has medium body. It has moderate flavor and is produced in the Napa region.

Αν η οντότητα δεν είχε όνομα (μάλλον απίθανο για κρασί, αλλά πιο πιθανό στην περίπτωση εκθεμάτων μουσείων), το κείμενο θα ήταν:

This is a Chardonnay made in Bancroft . It is dry and has medium body. It has moderate flavor and is produced in the Napa region.

Στα ελληνικά το παραπάνω κείμενο θα ήταν:

Αυτό είναι ένα Chardonnay που παρασκευάζεται στο Bancroft. Είναι ξηρό και έχει μέτρια υφή. Έχει μέτριο άρωμα και παρασκευάζεται στην περιοχή Napa.

Ομοίως, όταν αναφερόμαστε σε μια τάξη για πρώτη φορά, χρησιμοποιούμε το ονοματικό σύνολο με το οποίο συσχετίζεται η τάξη στο λεξικό, ενώ τις

επόμενες φορές χρησιμοποιούμε αντωνυμίες (ή κενό υποκείμενο στα ελληνικά). Για παράδειγμα:

Cabernet Sauvignon is a wine. It is made from at most 1 grape, Cabernet Sauvignon grape. It is dry and its color is red. It has moderate or strong flavor and medium or full body.

Το Cabernet Sauvignon είναι ένα κρασί. Παράγεται από το πολύ 1 σταφύλι, το σταφύλι Cabernet Sauvignon. Είναι ξηρό και έχει κόκκινο χρώμα. Έχει μέτριο ή ισχυρό άρωμα και μέτρια η πλήρη υφή.

4.7 Παραγωγή επιφανειακής μορφής

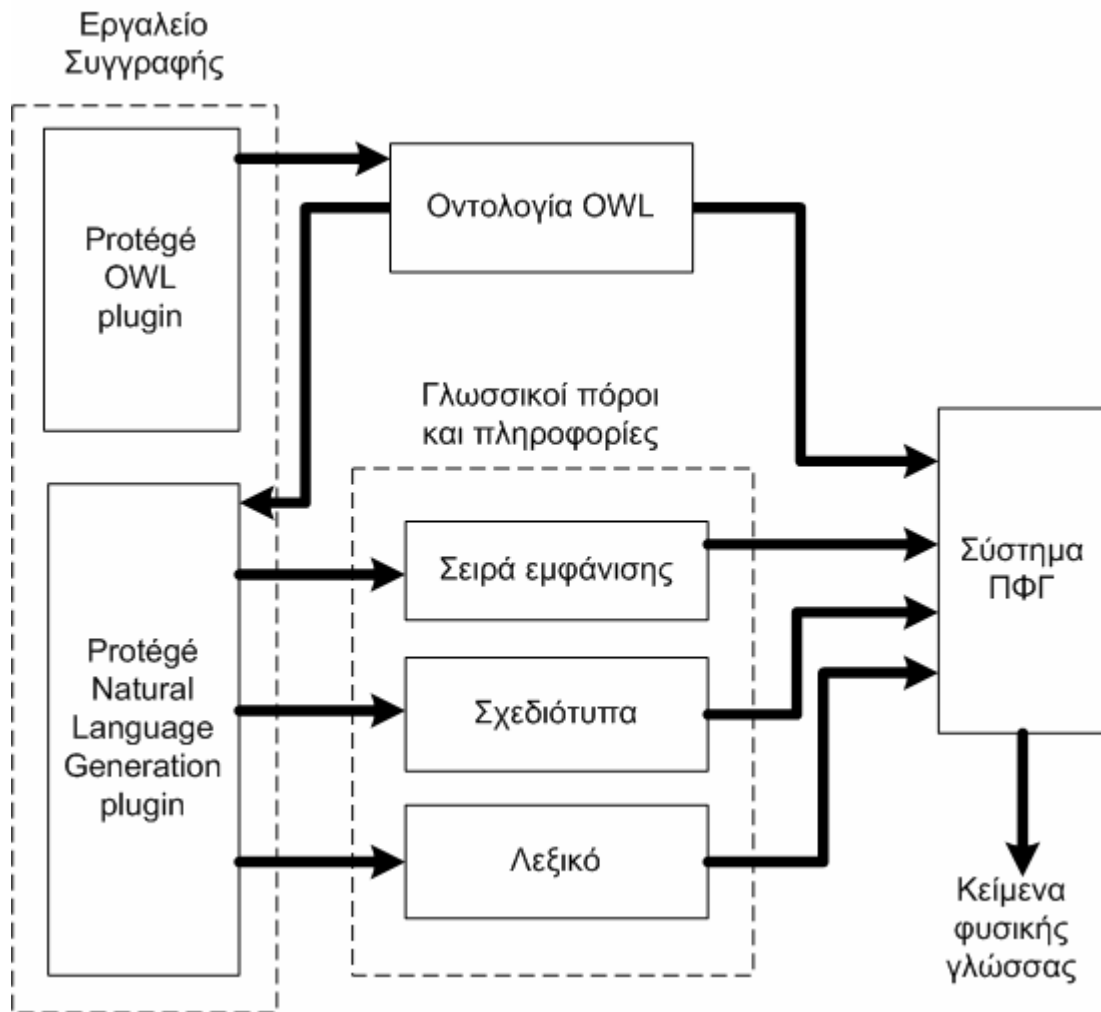
Όπως αναφέρθηκε στην ενότητα 2.2.3, σε συστήματα που βασίζονται σε σχεδιάσματα, όπως το σύστημα αυτής της εργασίας, το στάδιο παραγωγής επιφανειακής μορφής είναι εξαιρετικά απλοϊκό. Στο σύστημα της εργασίας, στη διάρκεια αυτού του σταδίου απλά προστίθενται τα κατάλληλα σημεία στίξης (π.χ. τελείες) και γίνεται η μετατροπή σε κεφαλαίο του πρώτου γράμματος κάθε πρότασης. Σε επόμενη έκδοση του συστήματος, θα προστίθενται επίσης ετικέτες HTML, συμπεριλαμβανομένων ετικετών που θα μετατρέπουν όλα τα ονόματα οντοτήτων και τάξεων σε υπερ-συνδέσμους, που θα επιτρέπουν στον αναγνώστη να ζητήσει περιγραφές των οντοτήτων ή τάξεων. Για παράδειγμα, στο κείμενο «This is a Chardonnay made in Bancroft.» τα «Chardonnay» και «Bancroft» θα είναι υπερ-σύνδεσμοι. Ακολουθώντας το δεύτερο σύνδεσμο, θα ζητείται από τη μηχανή παραγωγής να δημιουργήσει μια περιγραφή για το Bancroft.

4.8 Εργαλείο συγγραφής

4.8.1 Ο ρόλος του εργαλείου συγγραφής στο σύστημα ΠΦΓ της εργασίας

Όπως είναι φανερό το σύστημα ΠΦΓ που αναπτύχθηκε στα πλαίσια της εργασίας στηρίζεται σε κάποιους γλωσσικούς πόρους (π.χ. λεξικό, σχεδιάγραμμα) για την μετατροπή των συμβολικών πληροφοριών μιας OWL οντολογίας σε φυσική γλώσσα. Οι περισσότεροι από αυτούς τους πόρους, όπως και η ίδια η οντολογία, πρέπει να τροποποιούνται όποτε το σύστημα προσαρμόζεται σε μια άλλη συλλογή αντικειμένων. Για να είναι εύκολη η τροποποίηση της οντολογίας και των γλωσσικών πόρων, αναπτύχθηκε στη διάρκεια της εργασίας και ένα «εργαλείο συγγραφής», αντίστοιχο εκείνου του M-PIRO [22], που βοηθά τους «συγγραφείς», δηλαδή εκείνους που είναι υπεύθυνοι για την προσαρμογή του συστήματος σε μια νέα συλλογή αντικειμένων. Το εργαλείο συγγραφής υλοποιήθηκε ως επέκταση του Protégé [53]. Το Protégé είναι ένα πρόγραμμα ανοιχτού κώδικα (open source) για τη δημιουργία και επεξεργασία οντολογιών, του οποίου η αρχιτεκτονική επιτρέπει την εύκολη δημιουργία νέων μονάδων λογισμικού που μπορούν να προστεθούν στο ήδη υπάρχον σύστημα ως επεκτάσεις (plug-ins). Το Protégé ήδη διαθέτει ένα μεγάλο αριθμό επεκτάσεων και πολλούς χρήστες.

Μία από αυτές τις επεκτάσεις (Protégé OWL plug-in) βοηθά τους χρήστες του Protégé να δημιουργούν οντολογίες σε OWL μέσω μιας γραφικής διεπαφής. Στη διάρκεια της εργασίας αναπτύχθηκε μια ακόμα επέκταση (Protégé Natural Language Generation plug-in), η οποία επιτρέπει στους συγγραφείς να δημιουργούν ή να τροποποιούν όλους τους γλωσσικούς πόρους και τις επιπλέον πληροφορίες που απαιτούνται από το σύστημα ΠΦΓ. Στην εικόνα 3 παρουσιάζεται ο ρόλος του Protégé στο σύστημα ΠΦΓ της εργασίας. Πιο συγκεκριμένα, προστέθηκαν στο Protégé δύο καρτέλες (tabs), οι LexiconBrowserTab και TemplatesAndOrderingTab, που περιγράφονται στη συνέχεια.

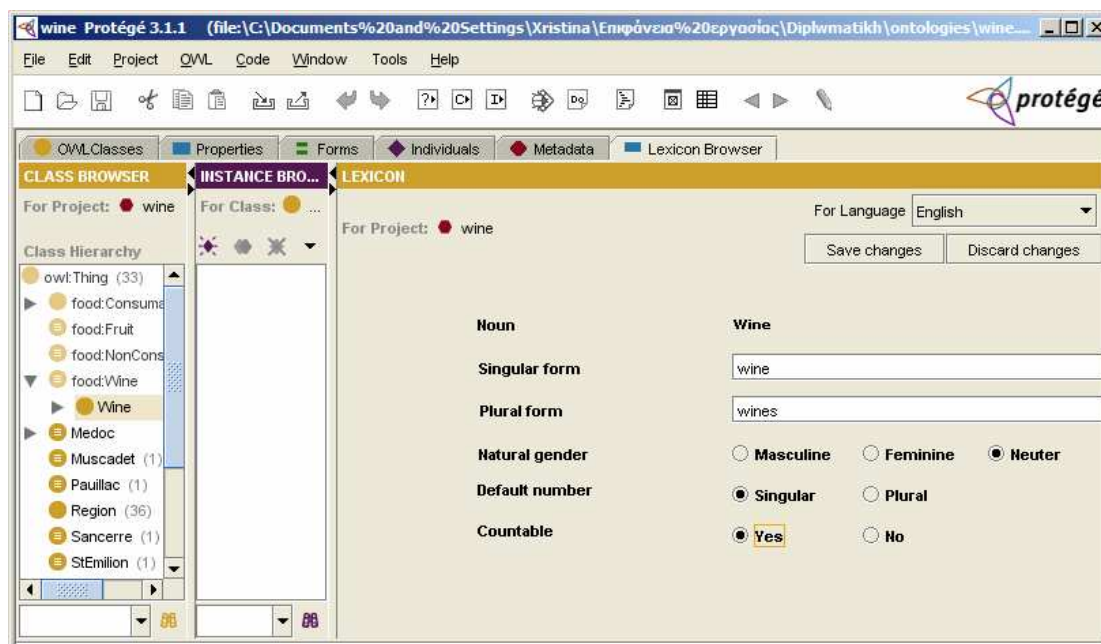


Εικόνα 3: Ο ρόλος του εργαλείου συγγραφής στο σύστημα ΠΦΓ της εργασίας.

4.8.2 LexiconBrowserTab

Σε αυτή την καρτέλα, ο συγγραφέας μπορεί να εισάγει τις πληροφορίες που αποθηκεύονται στο αγγλικό και ελληνικό λεξικό (Εικόνα 4 και 5). Οι πληροφορίες αυτές έχουν παρουσιαστεί στην ενότητα 4.4.5.

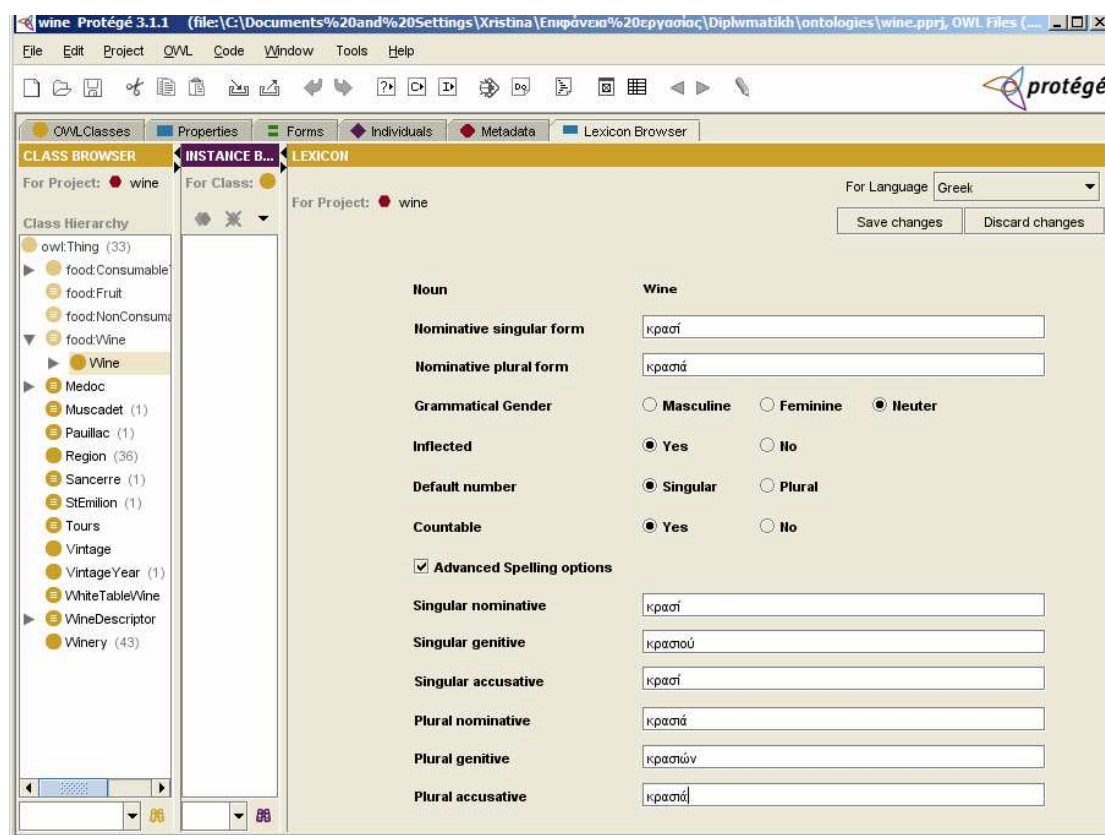
4. Το σύστημα ΠΦΓ της εργασίας



Εικόνα 4: Φόρμα εισαγωγής πληροφοριών στο αγγλικό λεξικό

Στη φόρμα που έχει δημιουργηθεί για το ελληνικό λεξικό (Εικόνα 5), ο συγγραφέας πρέπει να εισαγάγει για κάθε ουσιαστικό την μορφή του στην ονομαστική, γενική και αιτιατική του ενικού και πληθυντικού αριθμού. Για να συντομευθεί αυτή η διαδικασία, σε μια μελλοντική επέκταση του συστήματος θα πρέπει να αναπτυχθεί ένας αλγόριθμος αυτόματης δημιουργίας της μορφής του ουσιαστικού στη γενική και αιτιατική του ενικού και πληθυντικού από την ονομαστική του ενικού και πληθυντικού (ένας ανάλογος αλγόριθμος χρησιμοποιείται στο εργαλείο συγγραφής του M-PIRO). Προσωρινά, ο συγγραφέας πρέπει να επιλέγει το *Advanced Spelling Options* και να συμπληρώνει τις μορφές του ουσιαστικού σε όλες τις πτώσεις του ενικού και πληθυντικού, ενώ σε μελλοντική επέκταση θα συμπληρώνει μόνο τα πεδία *nominative singular/plural form* και θα χρησιμοποιεί το κουμπί *Advanced Spelling Options* για τη διόρθωση των παραγόμενων μορφών του ουσιαστικού μόνο όταν αυτό απαιτείται.

4. Το σύστημα ΠΦΓ της εργασίας



Εικόνα 5: Φόρμα εισαγωγής πληροφοριών στο ελληνικό λεξικό

4.8.3 TemplatesAndOrderingTab

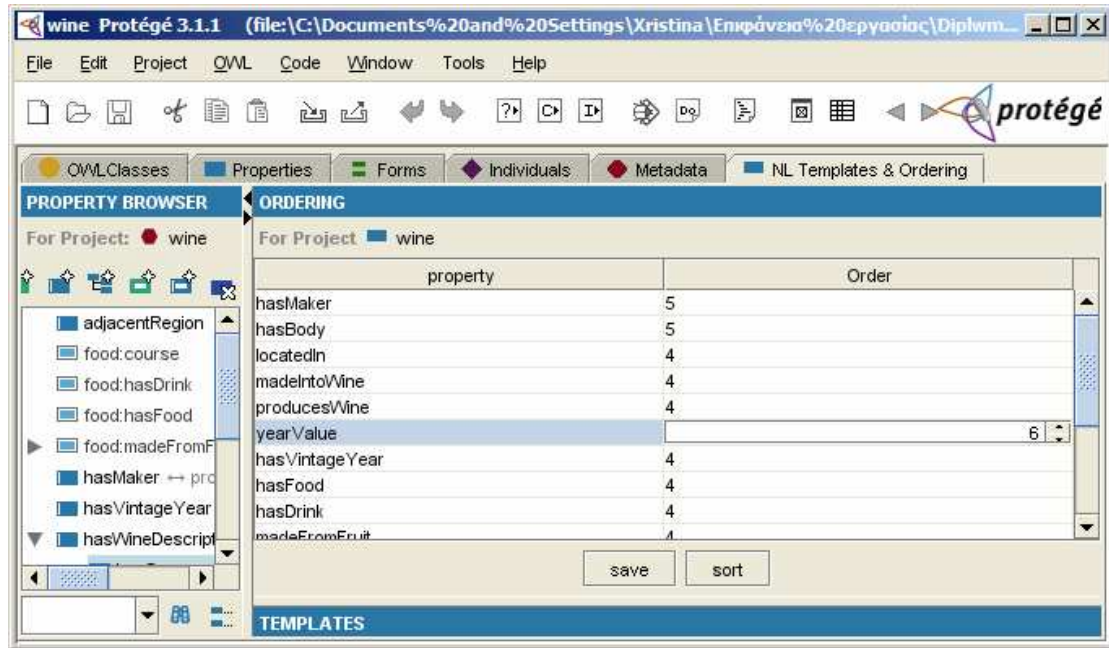
Σε αυτό το tab, ο χρήστης μπορεί να ορίσει τη σειρά των ιδιοτήτων (Εικόνα 7, βλ. και ενότητα 4.3) και να δημιουργήσει τα σχεδιάτυπα που χρησιμοποιεί το σύστημα ΠΦΓ (Εικόνα 8, βλ. και ενότητα 4.4.1).

Για να ορίσει τη σειρά, ο συγγραφέας εισάγει αριθμούς στα αντίστοιχα κελιά.

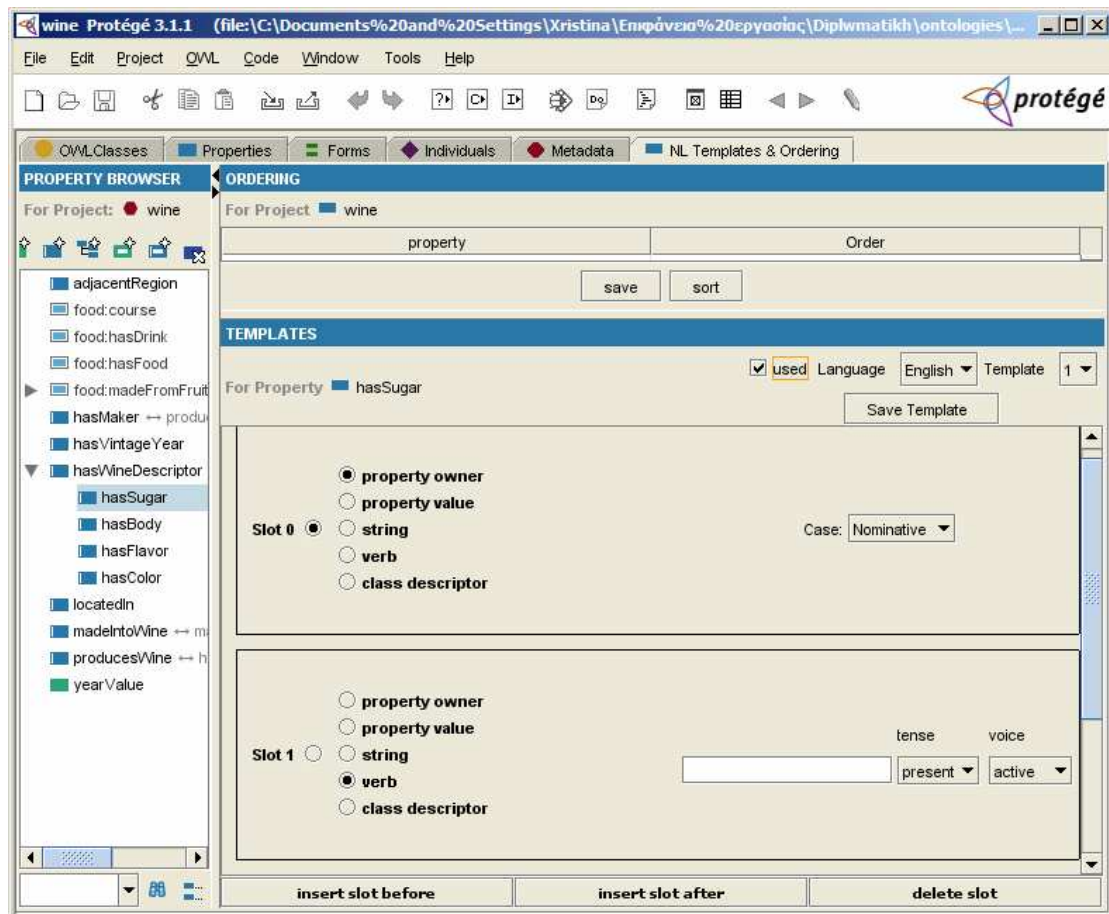
Το κουμπί sort ταξινομεί τις ιδιότητες κατά τους αριθμούς τους.

Κατά τον ορισμό των σχεδιάτυπων, ο συγγραφέας ορίζει τι περιέχει η κάθε θέση του σχεδιάτυπου (βλ. ενότητα 4.4.1).

4. Το σύστημα ΠΦΓ της εργασίας



Εικόνα 6: Φόρμα ορισμού της σειράς των ιδιοτήτων της οντολογίας



Εικόνα 7: Φόρμα ορισμού των σχεδίουτυπων.

5 Συμπεράσματα και προτάσεις για μελλοντικές εργασίες.

Στα πλαίσια της εργασίας δημιουργήθηκε ένα σύστημα το οποίο παράγει κείμενα στα ελληνικά και αγγλικά από συμβολικές πληροφορίες μιας οντολογίας OWL . Επίσης, επεκτάθηκε το σύστημα Protégé, ώστε χρήστες που δεν είναι ειδικοί στον τομέα της επεξεργασίας φυσικής γλώσσας να μπορούν να δημιουργούν τόσο την οντολογία OWL όσο και τις γλωσσικές πληροφορίες που απαιτεί το σύστημα παραγωγής φυσικής γλώσσας για τη λειτουργία του.

Το σύστημα παραγωγής φυσικής γλώσσας είναι δυνατόν να βελτιωθεί αν προστεθούν μοντέλα για τους χρήστες και το ιστορικό αλληλεπίδρασης, ώστε να περιλαμβάνονται στα παραγόμενα κείμενα μόνο πληροφορίες που ενδιαφέρουν τους χρήστες, να αποφεύγονται επαναλήψεις πληροφοριών, να επιλέγονται κατάλληλες εκφράσεις κλπ. Ακόμα, με τη χρήση ιστορικού αλληλεπίδρασης θα είναι δυνατή η παραγωγή κειμένων που περιέχουν συγκρίσεις (π.χ. “Ο φορητός υπολογιστής που βλέπετε έχει μεγαλύτερη επεξεργαστική ισχύ και μνήμη από όλους τους άλλους υπολογιστές που έχουν παρουσιαστεί έως τώρα”) [39]. Χρειάζονται, επίσης, επεκτάσεις που θα παράγουν αυτόματα τους διάφορους τύπους των κλιτών λέξεων του λεξικού.

Επιπλέον, είναι δυνατό το σύστημα να επεκταθεί, έτσι ώστε να μην παράγει κείμενα μόνο από OWL DL και OWL Lite, αλλά και από την γλώσσα SWRL (Semantic Web Rule Language) [51], που αποτελεί ένα συνδυασμό των γλωσσών OWL DL και OWL Lite με τις Unary/Binary Datalog RuleML υπογλώσσες της RuleML (Rule Markup Language) [46]. Η SWRL επιτρέπει την επέκταση μιας οντολογίας OWL με προτάσεις Horn. Για παράδειγμα σε μια OWL οντολογία που έχουν οριστεί οι ιδιότητες *hasParent* και *hasBrother* μπορεί να οριστεί ο παρακάτω κανόνας.

$$\text{hasParent}(?x1, ?x2) \wedge \text{hasBrother}(?x2, ?x3) \Rightarrow \text{hasUncle}(?x1, ?x3)$$

Το σύστημα ΠΦΓ της εργασίας μπορεί να επεκταθεί, έτσι ώστε να περιλαμβάνει στα παραγόμενα κείμενα και πληροφορίες που προκύπτουν ως συμπεράσματα από τη χρήση τέτοιων κανόνων ή να αποδίδει σε φυσική γλώσσα αυτούς τους κανόνες.

Αξιζει να σημειωθεί, ακόμα, ότι το ΕΚΕΦΕ «Δημόκριτος» πρόκειται να τροποποιήσει το εργαλείο συγγραφής του έργου M-PIRO, ώστε να υποστηρίζει τη μηχανή παραγωγής αυτής της εργασίας. Τέλος, η μηχανή ΠΦΓ αυτής της εργασίας πρόκειται να χρησιμοποιηθεί στο έργο «Ξένιος» για την παραγωγή περιγραφών εκθεμάτων μουσείων. Οι παραγόμενες γραπτές περιγραφές θα μετατρέπονται σε προφορική μορφή, μέσω σύνθεσης φωνής, και θα εκφωνούνται από κινητά ρομπότ-ξεναγούς σε επισκέπτες των μουσείων.

Αναφορές

- [1] A. Isard , J. Oberlander, I. Androutsopoulos, and C. Matheson (2003).
Speaking the users' languages. *IEEE Intelligent Systems* 18(1):40–45.
- [2] A. Melengoglou (2002). Multilingual aggregation in the M-PIRO system.
MSc thesis, School of Informatics, University of Edinburgh.
- [3] C. Mellish and X. Sun (2005). Natural Language Directed Inference in the
Presentation of Ontologies. *Proceedings of 10th European Workshop on
Natural Language Generation*, Aberdeen Britain.
- [4] C. Paris, and K. V. Linden (1996). DRAFTER: an interactive support tool
for writing multilingual instructions. *IEEE Computer* 29(7):49–56.
- [5] D. Brickley, R. V. Guha, B. McBride. *RDF Vocabulary Description Language
1.0: RDF Schema*
(<http://www.w3.org/TR/rdf-schema/>)
- [6] D. Connolly, F. van Harmelen, I. Horrocks, D. L. McGuinness, P. F. Patel-
Schneider, and L. A. Stein. *DAML+OIL (March 2001) Reference
Description*. (www.w3.org/TR/daml+oil-reference)
- [7] D. L. McGuinness and F. van Harmelen. *OWL Web Ontology Language
Overview*. W3C Recommendation 10 February 2004
(<http://www.w3.org/TR/2004/REC-owl-features-20040210/>)
- [8] E. Goldberg, N. Driedger, R. Kittredge (1994). Using natural nanguage
processing to produce weather forecasts. *IEEE Expert* 9(2):45-53.
- [9] E. Krahmer, S. van Erk and A. Verleg (2003). Graph-based Generation of
Referring expressions, *Computational Linguistics*, 29(1): 53-72, 2003.
- [10] E. Reiter, and R. Dale (2000). *Building Natural Language Generation
Systems*. Cambridge University Press.
- [11] *Extensible Markup Language (XML)* (www.w3.org/XML/)
- [12] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi and P. Patel-
Schneider (2003). *The Description Logic Handbook: Theory,
Implementation and Applications*. Cambridge University Press.
- [13] *Frame Logic* (<http://flora.sourceforge.net/aboutFlogic.php>)

- [14] G. Antoniou and F. van Harmelen. Web Ontology Language: OWL. (<http://www.cs.vu.nl/~frankh/postscript/OntoHandbook03OWL.pdf>)
- [15] G. Wilcock (2001). Pipelines, templates and transformations: XML for natural language generation. In Proceeding of NLPXML-2001.
- [16] G. Wilcock (2003). Integrating natural language generation with XML web technology. In Proceedings of EACL-2003 Demo Sessions.
- [17] G. Wilcock and K. Jokinen (2003). Generating Responses and Explanations from RDF/XML and DAML+OIL. In: Knowledge and Reasoning in Practical Dialogue Systems, IJCAI-2003, Acapulco 58-63.
- [18] G. Wilcock. (2003). Talking OWLs: Towards an Ontology Verbalizer. In: Human Language Technology for the Semantic Web and Web Services, ISWC'03, Sanibel Island, Florida 109-112.
- [19] H. Knublauch. User-Defined Datatypes in Protege-OWL. (<http://protege.stanford.edu/plugins/owl/xsp.html>).
- [20] H. Horacek (2003). A best-first search algorithm for generating referring expressions. Proceedings of EACL-2003.
- [21] H. Horacek (2004). On referring to Sets of Objects Naturally. Proceedings of the Third International Conference on Natural Language Generation (INLG-04)
- [22] I. Androutsopoulos, J. Oberlander and V. Karkaletsis (2005). Source Authoring for Multilingual Generation of Personalised Object Descriptions (preprint). Natural Language Engineering, Cambridge University Press (accepted for publication)
- [23] I. Androutsopoulos, S. Kallonis and V. Karkaletsis (2005). Exploiting OWL in the multilingual generation of object descriptions. Proceedings of 10th European Workshop on Natural Language Generation, Aberdeen Britain 2005.
- [24] I. Horrocks and P. F. Patel-Schneider (2003). Reducing OWL Entailment to Description Logic Satisfiability. In Proc. Of the 2nd International Semantic Web Conference (ISWC).
- [25] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen (2003). From

- SHIQ and RDF to OWL: The Making of a Web Ontology Language.
Journal of Web Semantics.
- [26] J. Heflin. OWL Web Ontology Language Use Cases and Requirements.
W3C Recommendation 10 February 2004
(<http://www.w3.org/TR/2004/REC-webont-req-20040210/>).
- [27] J. Hein, J. Hendler, and S. Luke (1999). SHOE: A Knowledge
Representation Language for Internet Applications. Technical Report CS-
TR-4078, University of Maryland, Department of Computer Science.
- [28] J. J. Carroll, J. Z. Pan XML Schema Datatypes in RDF and OWL. W3C
Working Group Note 14 March 2006.
(www.w3.org/TR/swbp-xsch-datatypes/)
- [29] Jena – A Semantic Web Framework for Java
(<http://jena.sourceforge.net/>)
- [30] K. Boncheva (2004). Open-source Tools for Creation, Maintenance, and
Storage of Lexical Resources for Language Generation from Ontologies. In:
Proceedings of 4th Language Resources and Evaluation Conference
(LREC'04).
- [31] K. Boncheva (2005). Generating Tailored Textual Summaries From
Ontologies. Proceedings of the Second European Semantic Web
Conference (ESWC'05), Heraklion.
- [32] K. Boncheva and Y. Wilks (2004). Automatic Report Generation from
Ontologies: the MIAKT approach. In Ninth International Conference on
Applications of Natural Language to Information Systems (NLDB'2004),
Manchester, UK.
- [33] K. van Deemter, M. Theune and E. Kraemer (2005). Real vs. Template-
based natural language generation: a false opposition? Computational
Linguistics 2005.
- [34] Knowledge Interchange Format (KIF)
(<http://logic.stanford.edu/kif/kif.html>)
- [35] M. Dean and G. Schreiber. OWL Web Ontology Language Reference.
W3C Recommendation 10 February 2004

- (<http://www.w3.org/TR/2004/REC-owl-ref-20040210/>)
- [36] M. E. Foster and M. White (2004). Techniques for Text Planning with XSLT. Proceedings of NLPXML-2004.
- [37] M. Horridge, H. Knublauch, A. Rector, R. Stevens and C. Wroe. A Practical Guide to Building OWL Ontologies Using the Protege-OWL Plugin and CO-ODE Tools.
(<http://www.co-ode.org/resources/tutorials/ProtegeOWLTutorial.pdf>)
- [38] M. K. Smith, C. Welty, and D. L. McGuinness. OWL Web Ontology Language Guide. W3C Recommendation 10 February 2004
(<http://www.w3.org/TR/2004/REC-owl-guide-20040210/>)
- [39] M. Milosavljevic (1999). The automatic generation of comparison in descriptions of entities. PhD thesis, Macquarie University.
- [40] N. Guarino (1998). Formal Ontology and Information Systems. Formal Ontology in Information Systems, Proc. Of the 1st International Conference, Trento, Italy, IOS Press pp. 3-15.
- [41] OCML: Operational Conceptual Modelling Language
(<http://kmi.open.ac.uk/projects/ocml/>).
- [42] P. D. Karp, V. K. Chaudhri, and J. Thomere (1999). XOL: An XML-based ontology exchange language. Version 0.3.
- [43] P. F. Patel-Schneider, P. Hayes I. Horrocks OWL Web Ontology Language Semantics and Abstract Syntax W3C Recommendation 10 February 2004.
- [44] P. Hayes, B. McBride. "RDF Semantics". W3C Recommendation 10 February 2004. (<http://www.w3.org/TR/rdf-mt/>)
- [45] Resource Description Framework (RDF) (<http://www.w3.org/RDF/>)
- [46] RuleML (www.ruleml.org/)
- [47] S. Kallonis (2005). Using existing ontologies and databases in the natural language generation system of the M-PIRO project. MSc dissertation, Department of Informatics, Athens University of Economics and Business.
- [48] S. Varges (2004). Overgenerating referring expressions involving

- relations. Proceedings of the Third International Conference on Natural Language Generation (INLG-04).
- [49] S. Varges and K. van Deemter (2005). Generating referring expressions containing quantifiers. Proceedings of IWCS 2005.
- [50] Semantic Web (<http://www.w3.org/2001/sw/>)
- [51] SWRL A Semantic Web Rule Language Combining OWL and RuleML. (<http://www.w3.org/Submission/SWRL/>)
- [52] The Extensible Stylesheet Language (XSL) (<http://www.w3.org/Style/XSL/>)
- [53] The Protégé Ontology Editor and Knowledge Acquisition System (<http://protege.stanford.edu/>)
- [54] W3C World Wide Consortium (<http://www.w3.org/>, <http://www.w3.org/Consortium/>)
- [55] XML Schema (<http://www.w3.org/XML/Schema>)

Παράρτημα 1

Η δομή του αρχείου στο οποίο καθορίζεται η σειρά των ιδιοτήτων (Ordering_schm.xsd).

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="Ordering" type="OrderType"/>

<xs:complexType name="OrderType">
<xs:sequence>
<xs:element ref="Order" minOccurs="1" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>

<xs:element name="Order">
<xs:complexType>
<xs:simpleContent>
<xs:extension base="xs:positiveInteger">
<xs:attribute name="property" type="xs:string" />
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>

</xs:schema>
```

Η δομή του αρχείου στο οποίο καθορίζονται οι πληροφορίες του αγγλικού λεξικού(Lexicon_en_schm.xsd).

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="Lexicon" type="EntryType"/>

<xs:complexType name="EntryType">
<xs:sequence>
```



```
<xs:choice minOccurs="1" maxOccurs="unbounded">
  <xs:element name="OntClass" type="NPType"/>
  <xs:element name="OntInd" type="NPType"/>
</xs:choice>
</xs:sequence>
</xs:complexType>

<xs:complexType name="NPType">
  <xs:sequence>
    <xs:element ref="NP" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
  <xs:attribute name="name" use="required" type="xs:anyURI"/>
</xs:complexType>

<xs:element name="NP">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="singular" minOccurs="1" maxOccurs="1"
        type="xs:string"/>
      <xs:element name="plural" minOccurs="1" maxOccurs="1"
        type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="gender" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="masculine"/>
          <xs:enumeration value="feminine"/>
          <xs:enumeration value="nonpersonal"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="num" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="singular"/>
          <xs:enumeration value="plural"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="countable" use="required">
      <xs:simpleType>
```

```
<xs:restriction base="xs:string">
<xs:enumeration value="T"/>
<xs:enumeration value="F"/>
</xs:restriction>
</xs:simpleType>
</xs:attribute>
</xs:complexType>
</xs:element>

</xs:schema>
```

Η δομή του αρχείου στο οποίο καθορίζονται οι πληροφορίες του ελληνικού λεξικού(Lexicon_gr_schm.xsd).

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="Lexicon" type="EntryType"/>

<xs:complexType name="EntryType">
<xs:sequence>
<xs:choice minOccurs="1" maxOccurs="unbounded">
<xs:element name="OntClass" type="NPType"/>
<xs:element name="OntInd" type="NPType"/>
</xs:choice>
</xs:sequence>
</xs:complexType>

<xs:complexType name="NPType">
<xs:sequence>
<xs:element ref="NP" minOccurs="1" maxOccurs="1"/>
</xs:sequence>
<xs:attribute name="name" use="required" type="xs:anyURI"/>
</xs:complexType>

<xs:element name="NP">
<xs:complexType>
<xs:sequence>
<xs:element name="singular" minOccurs="1" maxOccurs="1"
type="casesType"/>
```

```
<xs:element name="plural" minOccurs="1" maxOccurs="1"
type="casesType"/>
</xs:sequence>
<xs:attribute name="gender" use="required">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:enumeration value="masculine"/>
<xs:enumeration value="feminine"/>
<xs:enumeration value="neuter"/>
</xs:restriction>
</xs:simpleType>
</xs:attribute>
<xs:attribute name="num" use="required">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:enumeration value="singular"/>
<xs:enumeration value="plural"/>
</xs:restriction>
</xs:simpleType>
</xs:attribute>
<xs:attribute name="countable" use="required">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:enumeration value="T"/>
<xs:enumeration value="F"/>
</xs:restriction>
</xs:simpleType>
</xs:attribute>
</xs:complexType>
</xs:element>

<xs:complexType name="casesType">
<xs:sequence>
<xs:element name="nominative" minOccurs="1" maxOccurs="1"
type="xs:string"/>
<xs:element name="genitive" minOccurs="1" maxOccurs="1"
type="xs:string"/>
<xs:element name="accusative" minOccurs="1" maxOccurs="1"
type="xs:string"/>
</xs:sequence>
</xs:complexType>
```

```
</xs:schema>
```

Η δομή του αρχείου στο οποίο αποθηκεύονται τα ονόματα των σχεδιάσιμων (PropertiesTemplates_schm.xsd).

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="AllTemplates" type="AllTemplatesType"/>

<xs:complexType name="AllTemplatesType">
<xs:sequence>
<xs:element name="templates" minOccurs="1" maxOccurs="unbounded"
type="templatesType"/>
</xs:sequence>
</xs:complexType>

<xs:complexType name="templatesType">
<xs:sequence>
<xs:element ref="templ" minOccurs="5" maxOccurs="5" />
</xs:sequence>
<xs:attribute name="property" use="required" type="xs:string"/>
</xs:complexType>

<xs:element name="templ">
<xs:complexType>
<xs:attribute name="TemplName" use="required">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:enumeration value="templ1"/>
<xs:enumeration value="templ2"/>
<xs:enumeration value="templ3"/>
<xs:enumeration value="templ4"/>
<xs:enumeration value="templ5"/>
</xs:restriction>
</xs:simpleType>
</xs:attribute>
<xs:attribute name="used" use="required">
```

```
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:enumeration value="T"/>
<xs:enumeration value="F"/>
</xs:restriction>
</xs:simpleType>
</xs:attribute>
<xs:attribute name="used_for_restr" use="required">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:enumeration value="T"/>
<xs:enumeration value="F"/>
</xs:restriction>
</xs:simpleType>
</xs:attribute>
</xs:complexType>
</xs:element>

</xs:schema>
```

Η δομή του αρχείου στο οποίο αποθηκεύονται τα σχεδιάσματα.

(xsd.xsd)

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns="http://www.w3.org/1999/XSL/Transform"
xmlns:tmpl=""
targetNamespace="http://www.w3.org/1999/XSL/Transform"
elementFormDefault="qualified">

<xs:import namespace="" schemaLocation="template_schm.xsd"/>

<xs:element name="stylesheet">
<xs:complexType>
<xs:sequence>
<xs:element name="template" minOccurs="1" maxOccurs="unbounded"
type="templateType"/>
</xs:sequence>
<xs:attribute name="version" type="xs:string"/>
```

```
</xs:complexType>
</xs:element>

<xs:complexType name="templateType">
  <xs:sequence>
    <xs:choice minOccurs="1" maxOccurs="unbounded">
      <xs:element ref="tmpl:Msg"/>
      <xs:element name="choose" type="chooseType"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="match" type="xs:string"/>
  <xs:attribute name="name" type="xs:string"/>
</xs:complexType>

<xs:element name="call-template">
  <xs:complexType>
    <xs:attribute name="name" type="xs:string"/>
  </xs:complexType>
</xs:element>

<xs:element name="value-of">
  <xs:complexType>
    <xs:attribute name="select" type="xs:string"/>
  </xs:complexType>
</xs:element>

<xs:complexType name="chooseType">
  <xs:sequence>
    <xs:element name="when" minOccurs="3" maxOccurs="3" type="textType"/>
    <xs:element name="otherwise" minOccurs="1" maxOccurs="1"
      type="xs:string"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="textType">
  <xs:sequence>
    <xs:element ref="tmpl:Text" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
  <xs:attribute name="test" type="xs:string"/>
</xs:complexType>
```

```
</xs:schema>
```

(template_schm.xsd)

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
targetNamespace=""
xmlns=""
xmlns:tmpl=""
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
elementFormDefault="qualified">

<xs:import namespace="http://www.w3.org/1999/XSL/Transform"
schemaLocation="xsl.xsd"/>

<xs:element name="Filler">
<xs:complexType>
<xs:sequence>
<xs:element ref="xsl:value-of" minOccurs="1" maxOccurs="1"/>
</xs:sequence>
<xs:attribute name="case">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:enumeration value="nominative"/>
<xs:enumeration value="genitive"/>
<xs:enumeration value="accusative"/>
</xs:restriction>
</xs:simpleType>
</xs:attribute>
</xs:complexType>
</xs:element>

<xs:element name="Num">
<xs:complexType>
<xs:sequence>
<xs:element ref="xsl:value-of" minOccurs="1" maxOccurs="1"/>
</xs:sequence>
</xs:complexType>
```

```
</xs:element>

<xs:element name="Text" type="xs:NMTOKENS" />

<xs:element name="Msg">
  <xs:complexType>
    <xs:sequence>
      <xs:choice minOccurs="1" maxOccurs="unbounded">
        <xs:element ref="Owner" />
        <xs:element ref="Filler" />
        <xs:element ref="Num" />
        <xs:element ref="Text" />
        <xs:element ref="Verb" />
        <xs:element name="ClsDesc" type="xs:anyURI" />
        <xs:element ref="xsl:call-template"/>
      </xs:choice>
    </xs:sequence>
    <xs:attribute name="Order" type="xs:string"/>
    <xs:attribute name="RestrType" type="xs:string"/>
    <xs:attribute name="property" type="xs:string"/>
  </xs:complexType>
</xs:element>

<xs:element name="Owner">
  <xs:complexType>
    <xs:attribute name="case">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="nominative"/>
          <xs:enumeration value="genitive"/>
          <xs:enumeration value="accusative"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
</xs:element>

<xs:element name="Verb">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
```



```
<xs:attribute name="voice">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="passive"/>
      <xs:enumeration value="active"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
<xs:attribute name="tense">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="past"/>
      <xs:enumeration value="present"/>
      <xs:enumeration value="future"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>

</xs:schema>
```