



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Διαχείριση Χωροχρονικής Πληροφορίας σε Εφαρμογές Σημασιολογικού Ιστού

Εκπονήθηκε από:
ΓΕΩΡΓΙΟΣ
ΧΑΤΖΗΓΕΩΡΓΑΚΙΔΗΣ

Επιτροπή:
ΑΝ. ΚΑΘ. ΕΥΡΗΣΙΔΗΣ Γ.Μ. ΠΕΤΡΑΚΗΣ
(Επιβλέπων)
ΕΠΚ. ΚΑΘ. ΑΙΚΑΤΕΡΙΝΗ ΜΑΝΙΑ
ΚΑΘ. ΜΙΧΑΗΛΗΣ ΖΕΡΒΑΚΗΣ

7 Ιουλίου 2011

Περιεχόμενα

1	Εισαγωγή	4
1.1	Κίνητρο	4
1.2	Ορισμός Προβλήματος	5
1.3	Προτεινόμενη Λύση	5
1.4	Δομή της Εργασίας	7
2	Σχετικές Εργασίες/Διατριβές	8
2.1	Σημσιολογικός Ιστός, Οντολογίες, Χωρο-Χρονική Αναπαράσταση	8
2.2	Οντολογία SOWL	11
2.3	Γλώσσα Ερωτήσεων SOWL	17
2.4	Ανάλυση Κίνησης σε Video (Motion Analysis)	20
3	Υλοποίηση Συστήματος	22
4	Πειραματικές Μετρήσεις	40
4.1	Εκτίμηση Απόδοσης SOWL (SPARQL)	40
4.2	Εκτίμηση Απόδοσης SOWL (Αντικείμενα/Πλαίσιο)	42
4.3	Εκτίμηση Απόδοσης SOWL (Πλαίσια/Video)	43
4.4	Εκτίμηση Απόδοσης SOWL (Στιγμιότυπα/Οντολογία)	44
4.5	Ερωτήσεις (Queries), Παραδείγματα	45
5	Επίλογος (Συμπεράσματα και Επεκτάσεις)	49
	Α' Δομή των κλάσεων	52

Περίληψη

Η αναπαράσταση γνώσης σε εφαρμογές και στο διαδίκτυο είναι, τα τελευταία χρόνια, ένα ολοένα αυξανόμενο ενδιαφέροντος πεδίο έρευνας. Η παρούσα εργασία είναι μία εφαρμογή Σημασιολογικού Ιστού, η οποία χρησιμοποιεί την SOWL οντολογία [1] (επιτρέπει την αναπαράσταση χωροχρονικών δεδομένων σε σημασιολογικό επίπεδο), καθώς και την γλώσσα ερωτήσεων για την οντολογία αυτή. Και τα δύο έχουν αναπτυχθεί στο Εργαστήριο Προγραμματισμού και Τεχνολογίας Ευφυών Υπολογιστικών Συστημάτων (Intelligent Systems Laboratory). Σε συνδιασμό με ένα μηχανισμό ανίχνευσης κίνησης αντικειμένων σε video [2] (ο οποίος αναπτύχθηκε στο εργαστήριο ψηφιακής επεξεργασίας σήματος και εικόνας του Πολυτεχνείου Κρήτης), η εφαρμογή διαβάζει αρχεία τα οποία περιέχουν περιβάλλοντα παραλληλόγραμμα (Minimum Bounding Rectangles -εν συντομία MBRs) αντικειμένων που έχουν εντοπιστεί σε συγκεκριμένο video ενδιαφέροντος μέσω της εφαρμογής ανίχνευσης που αναφέρθηκε, υπολογίζει χωρικές σχέσεις μεταξύ των αντικειμένων κάθε στιγμή και τα εισάγει στην οντολογία SOWL. Στη συνέχεια παρέχει την δυνατότητα ερωτήσεων πάνω στα δεδομένα που εισήχθησαν, ώστε ο χρήστης να μπορεί να λάβει πληροφορίες της θέσης του κάθε αντικειμένου σχετικά με τα υπόλοιπα αντικείμενα ανά πάσα στιγμή. Η εφαρμογή, εκτός από MBR's υπολογίζει τις σχέσεις και μεταξύ πολυγώνων, πράγμα το οποίο επιδεικνύεται σε μια δευτερεύουσα λειτουργικότητα (mode), η οποία παρέχει τη δυνατότητα δημιουργίας τυχαίων κινούμενων σχημάτων. Ο υπολογισμός των σχέσεων γίνεται, όπως θα δούμε σε πολυωνυμικό χρόνο, ενώ ο χρόνος απάντησης των ερωτήσεων(queries) θα αναλυθεί πειραματικά. Η εφαρμογή που αναπτύχθηκε υποστηρίζει την επικοινωνία με τον χρήστη και επιτρέπει στον χρήστη να επιλέξει video το οποίο έχει υποστεί εντοπισμό κίνησης, να εισάγει στην οντολογία SOWL την χωροχρονική πληροφορία που έχει εξαχθεί και να απευθύνει ερωτήσεις που σχετικά με αυτήν στην γλώσσα ερωτήσεων.

Κεφάλαιο 1

Εισαγωγή

1.1 Κίνητρο

Σκοπός της εργασίας είναι η επίδειξη των δυνατοτήτων του Σημασιολογικού Ιστού (Semantic Web). Η ραγδαία ανάπτυξη και χρήση του διαδικτύου (World Wide Web) στις μέρες μας έχει δημιουργήσει πλεόνασμα πληροφοριών των οποίων η διαχείριση και οργάνωση είναι δύσκολο έργο. Τα τελευταία χρόνια, η ερευνητική κοινότητα παγκοσμίως έχει στραφεί σε μία νέα επέκταση του ιστού, τον Σημασιολογικό Ιστό. Ο Σημασιολογικός Ιστός επιτρέπει σαφέστερη νοηματοδότηση των πληροφοριών, επιτρέποντας την αυτοματοποίηση της επεξεργασίας τους από μηχανές. Με το Σημασιολογικό Ιστό η πληροφορία επεκτείνεται με αναγνώσιμα από μηχανές δεδομένα (metadata). Δίνεται έτσι η δυνατότητα δημιουργίας εφαρμογών οι οποίες θα επιτρέπουν την αυτοματοποίηση πολλών εργασιών εκ μέρους των χρηστών του διαδικτύου. Μία ομάδα η οποία στο μέλλον θα αποκτήσει ιδιαίτερο ενδιαφέρον από την ερευνητική κοινότητα είναι αυτή των εφαρμογών παρακολούθησης και διαχείρισης της εξέλιξης στο χώρο και στο χρόνο αντικειμένων, που εμφανίζονται σε κάποιο video. Σε έναν αγώνα ποδοσφαίρου, για παράδειγμα, είναι θεμιτή η ανάπτυξη μιας εφαρμογής, η οποία κατά τη διεξαγωγή του αγώνα, αυτόματα θα εντοπίζει τους παίκτες, θα τους αναγνωρίζει, και θα διαχειρίζεται σημαντικές πληροφορίες για τον καθέναν από αυτούς. Στη συγκεκριμένη εργασία επεξεργάζονται πληροφορίες που αφορούν τα αντικείμενα που εμφανίζονται σε ένα video, τις θέσεις τους, καθώς και τις σχετικές τους θέσεις με άλλα αντικείμενα στο χώρο και στο χρόνο. Ο χρήστης έτσι αποκτά τη δυνατότητα υποβολής ερωτήσεων πάνω στην επεξεργασμένη πληροφορία σχετικά με τα αντικείμενα που παρατηρεί στο video. Με αυτόν τον τρόπο του παρέχεται η δυνατότητα λήψης σημαντικών πληροφοριών, χωρίς την παρέμβαση ανθρώπινου παράγοντα.

Ένα άλλο παράδειγμα χρήσης του Σημασιολογικού Ιστού σε εφαρμογή αναπαραγωγής video είναι η παρακολούθηση και ανάλυση της εξέλιξης ενός αγώνα Formula 1. Με τα εργαλεία που χρησιμοποιούμε στην παρούσα εργασία, είναι δυνατό, με αυτόματο τρόπο να γίνεται αποθήκευση των πληροφοριών σχετικά με τη θέση του κάθε μονοθεσίου, ανά πάσα στιγμή, σχετικά με τα υπόλοιπα. Ο χρήστης, μέσω μιας εφαρμογής που θα είναι τοποθετημένη στον επίσημο ιστοτόπο της Formula 1, θα μπορεί να ρωτάει με εύκολο τρόπο το σύστημα για την θέση του αγαπημένου του οδηγού σχετικά με τους υπολοίπους, και να λαμβάνει απάντηση αυτόματα. Η ανάγκη για τη μελλοντική ανάπτυξη και χρήση τέτοιων εφαρμογών πάνω στο Σημασιολογικό Ιστό ήταν το κίνητρο για την εκπόνηση της διπλωμα-

τικής αυτής εργασίας.

1.2 Ορισμός Προβλήματος

Αρχικά, για να γίνει δυνατή η εξαγωγή γνώσης σχετικά με τα αντικείμενα ενός video, είναι απαραίτητη η αποδοτική ανάλυσή του, ο εντοπισμός των αντικειμένων του και η εξαγωγή των ακριβών θέσεων τους κάθε στιγμή. Λόγω της φύσης ενός video, παρουσιάζονται διάφορα εμπόδια στην επίτευξη του σκοπού αυτού. Η κίνηση του φακού της κάμερας, για παράδειγμα είναι ένα πρόβλημα που χρίζει αντιμετώπισης, καθώς εισάγει μεγάλη διαφορετικότητα μεταξύ των πλαισίων (frames) του video, βάσει της οποίας γίνεται η αναγνώριση των αντικειμένων. Για τον ίδιο λόγο, η απότομη αλλαγή φωτισμού, επηρεάζει αρνητικά τη διαδικασία της ανάλυσης. Κρίνεται λοιπόν αναγκαίο να χρησιμοποιηθεί ένας αποδοτικός αλγόριθμος, ο οποίος θα μας παρέχει την κατάλληλη πληροφορία αφού πρώτα έχει χειριστεί τα ζητήματα που αναφέρθηκαν.

Στην καθημερινότητά μας, χρησιμοποιούμε κάποιες συγκεκριμένες εκφράσεις στην προσπάθειά μας να περιγράψουμε τη θέση ενός αντικειμένου. Στην πλειονότητά τους, αυτές οι εκφράσεις χρησιμοποιούν κάποιο επιπλέον αντικείμενο ως αντικείμενο αναφοράς. Για παράδειγμα περιγράφουμε την θέση της πόλης Θεσσαλονίκη ως 'Βόρεια της Αθήνας'. Κρίνεται απαραίτητο λοιπόν σε εφαρμογές περιγραφής γνώσης θέσεων αντικειμένων, να χρησιμοποιηθούν παρόμοιοι τρόποι περιγραφής τοποθεσιών, ώστε να διευκολύνεται ο χρήστης στις ερωτήσεις που κάνει στο σύστημα. Εφ' όσον σε ένα video υπάρχει κίνηση των αντικειμένων, η πλήρης περιγραφή της γνώσης απαιτεί να ορίσουμε και τη χρονική διάρκεια κατά την οποία ισχύει μια σχέση μεταξύ δύο αντικειμένων στο χώρο. Διευκολύνεται έτσι η διατύπωση ερωτήσεων όπως 'Ποιο αντικείμενο βρίσκεται βόρεια του αντικειμένου A πριν τη χρονική στιγμή X'. Κρίνεται απαραίτητη λοιπόν η χρήση χρονικών τελεστών που επιτρέπουν το σχηματισμό τέτοιων ερωτήσεων. Τέλος, για λόγους διευκόλυνσης προς τον χρήστη, είναι αναγκαία η δημιουργία μιας γραφικής διεπαφής, φιλικής και εύχρηστης, η οποία θα παρέχει γρήγορη πρόσβαση σε όλες τις λειτουργίες της εφαρμογής.

1.3 Προτεινόμενη Λύση

Η πληροφορία προς αποθήκευση του Σημασιολογικού Ιστού, κωδικοποιείται σε λεξιλόγια αναπαράστασης γνώσεων, γνωστά ως *Οντολογίες*. Οι οντολογίες αποθηκεύουν γνώση σε μορφή αναγνώσιμη από υπολογιστές για την αυτοματοποίηση της επεξεργασίας τους. Οι οντολογίες θα βρίσκονται σε συγκεκριμένες τοποθεσίες στο διαδίκτυο, ώστε οι χρήστες να έχουν εύκολη πρόσβαση στην πληροφορία την οποία κρατούν. Τα δεδομένα που εξάγονται στην παρούσα εργασία θα αποθηκεύονται στην οντολογία SOWL [1], ενώ οι ερωτήσεις πάνω στα δεδομένα αυτά θα γίνονται με τη χρήση της Γλώσσας ερωτήσεων SOWL (SOWL Query Language) [1]. Η ανάπτυξη και των δύο έγινε στο Εργαστήριο Προγραμματισμού και Τεχνολογίας Ευφυών Υπολογιστικών Συστημάτων του Πολυτεχνείου Κρήτης.

Για την αναπαράσταση των σχέσεων μεταξύ αντικειμένων στο χώρο, θα χρησιμοποιηθούν οι εξής δύο κατηγορίες σχέσεων:

- **Σχέσεις Κατεύθυνσης (Directional Relations) [3]** : Οι σχέσεις αυτές περιγράφουν την θέση ενός αντικειμένου στο χώρο, σχετικά με ένα σημείο αναφοράς (δεύτερο αντικείμενο), χρησιμοποιώντας τα σημεία του ορίζοντα (Βόρεια, Νότια, Ανατολικά, Δυτικά, Βόρειο-Ανατολικά, Βόρειο-Δυτικά, Νότιο-Ανατολικά, Νότιο-Δυτικά)
- **Τοπολογικές Σχέσεις [4]**. Οι σχέσεις αυτές περιγράφουν τη θέση ενός αντικειμένου σχετικά με ένα άλλο, όσον αφορά τη γειτνίαση τους (Μέσα/Έξω, Εφαπτόμενα, Όμοια κλπ).

Η οντολογία SOWL [1] αποθηκεύει χρονικές στιγμές και χρονικά διαστήματα με τη μορφή ημερομηνίας και ώρας. Για την περιγραφή σχέσεων μεταξύ χρονικών διαστημάτων και χρονικών στιγμών, χρησιμοποιείται μία ομάδα τελεστών, οι οποίοι περιγράφουν ποιοτικά και ποσοτικά τον τρόπο με τον οποίο τα χρονικά διαστήματα σχετίζονται μεταξύ τους. Οι τελεστές αυτοί ονομάζονται Τελεστές Allen (Allen Operators) [5] και θα αναλυθούν σε παρακάτω κεφάλαιο.

Ο υπολογισμός των σχέσεων μεταξύ των εξαγομένων αντικειμένων γίνεται χρησιμοποιώντας τις συντεταγμένες των σημείων τους στην οθόνη. Υπολογίζοντας το κέντρο μάζας για κάθε αντικείμενο, διευκολύνεται ο προσδιορισμός των μεταξύ τους σχέσεων κατεύθυνσης. Αυτό επιτυγχάνεται υπολογίζοντας τη γωνία που σχηματίζεται μεταξύ δύο κεντρών μάζας. Έτσι, εάν για παράδειγμα υπολογιστεί γωνία 90° μεταξύ των κεντρών δύο αντικειμένων A και B, συμπαιρένουμε το εξής: *Το A αντικείμενο τοποθετείται Βόρεια του B ΚΑΙ το B Νότια του A*. Χρησιμοποιώντας τις συντεταγμένες των υπολοίπων σημείων, υπολογίζουμε και τις Τοπολογικές Σχέσεις [4].

Για την ανάλυση κίνησης και την εξαγωγή αντικειμένων από ένα video, γίνεται χρήση ενός αλγορίθμου που αναπτύχθηκε στο εργαστήριο Ψηφιακής Επεξεργασίας Σήματος και Εικόνας του Πολυτεχνείου Κρήτης [3]. Ο αλγόριθμος, εντοπίζει τα κινούμενα αντικείμενα και σχεδιάζει τα περιβάλλοντα παραλληλόγραμμά τους, τα οποία αποθηκεύει με τη μορφή αρχείου κειμένου στην έξοδό του. Το αρχείο περιέχει τις συντεταγμένες της διαγωνίου του κάθε παραλληλογράμμου σε κάθε πλαίσιο (frame) του video.

Αφού διαβάζονται τα περιβάλλοντα παραλληλόγραμμο από το αρχείο, δημιουργούνται τα κατάλληλα στιγμιότυπα. Πριν την εισαγωγή τους στην οντολογία SOWL, υπολογίζονται πρώτα οι μεταξύ τους σχέσεις κατεύθυνσης (Directional) [3], και έπειτα οι τοπολογικές (Topologic) [4] σχέσεις. Η εισαγωγή γίνεται για κάθε καρέ του video. Η ανάγνωση του αρχείου με τις συντεταγμένες των περιγραμμάτων, ο υπολογισμός των σχέσεων και η εισαγωγή στην οντολογία γίνονται παράλληλα με την αναπαραγωγή του video, αφού ο χρήστης το επιλέξει. Μετά την επιτυχή εισαγωγή των δεδομένων στην οντολογία, ο χρήστης μέσω της γραφικής διεπαφής της εφαρμογής μπορεί να κάνει χρήση της Γλώσσας Ερωτήσεων SOWL [1], για να ρωτήσει σχετικά με τις τοποθεσίες των αντικειμένων που τον ενδιαφέρουν.

1.4 Δομή της Εργασίας

Η εφαρμογή αναπτύχθηκε σε γλώσσα Java στο γραφικό περιβάλλον ανάπτυξης λογισμικού Eclipse. Για τη γραφική διεπαφή και το χειρισμό της αναπαραγωγής video χρησιμοποιήθηκε το Java API QtJambi¹ το οποίο είναι Java Bindings της C++ βιβλιοθήκης Qt² η οποία χρησιμοποιείται για την ανάπτυξη πολλών cross-platform εφαρμογών. Τα αρχεία των Java κλάσεων, οργανώθηκαν σε πακέτα, ανάλογα με τη λειτουργία τους.

Στα παρακάτω κεφάλαια της διατριβής αυτής, θα γίνει μια αναλυτική περιγραφή της διαδικασίας που ακολουθήθηκε προκειμένου να υλοποιηθεί η εφαρμογή. Στο Κεφάλαιο 2, θα γίνει αναφορά και ανάλυση όλων των σχετικών με την παρούσα, εργασιών και υλοποιήσεων λογισμικού, οι οποίες χρησιμοποιήθηκαν για την εκπόνησή της. Στο Κεφάλαιο 3 θα γίνει ανάλυση της αρχιτεκτονικής του συστήματος που υλοποιήθηκε, ακολουθούμενη από κάποιες πειραματικές μετρήσεις σχετικά με το χρόνο τον οποίο η εφαρμογή χρειάζεται για να επιστρέψει απαντήσεις μέσω της γλώσσας ερωτήσεων. Επίσης θα αναλυθεί και ο τρόπος λειτουργίας της τελικής εφαρμογής μαζί με κάποια παραδείγματα χρήσης της. Τέλος, στο Κεφάλαιο 4 θα γίνει μια σύντομη αναφορά για μελλοντικές χρήσεις/επεκτάσεις της παρούσας εργασίας παράλληλα με τον επίλογο.

¹Reference : <http://qt-jambi.org/>

²Reference : <http://qt.nokia.com/products/>

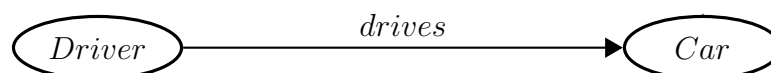
Κεφάλαιο 2

Σχετικές Εργασίες/Διατριβές

2.1 Σημασιολογικός Ιστός, Οντολογίες, Χωρο-Χρονική Αναπαράσταση

Ο όρος Σημασιολογικός Ιστός (Semantic Web) με λίγα λόγια θα μπορούσε να αποτυπωθεί ως μια σαφής κωδικοποίηση και ιεράρχηση των πληροφοριών που περιέχουν οι ιστοσελίδες για τη διευκόλυνση της επεξεργασίας τους από υπολογιστές. Μηχανές, όπως αυτόνομοι πράκτορες, θα μπορούν έτσι να αυτοματοποιήσουν εργασίες οι οποίες γίνονται σήμερα από ανθρώπους. Σκοπός της εργασίας αυτής είναι να επιδείξει την δυνατότητα ύπαρξης μιας τέτοιας μηχανής. Ιδρυτής του όρου Σημασιολογικός Ιστός είναι ο Tim Berners-Lee ο οποίος υπήρξε ιδρυτής και του Παγκοσμίου Ιστού (World Wide Web). Ο Tim Berners-Lee ορίζει την έννοια "Σημασιολογικός Ιστός" ως *"Ένα δίκτυο δεδομένων, το οποίο μπορεί να υποστεί επεξεργασία έμμεσα και άμεσα από μηχανές"*¹.

Όπως αναφέρθηκε, η πληροφορία στον Σημασιολογικό Ιστό, κωδικοποιείται σε οντολογίες. Μία οντολογία παρέχει τη δυνατότητα κωδικοποίησης και αποθήκευσης εννοιών και οντοτήτων καθώς και σχέσεων μεταξύ τους. Ένα παράδειγμα μιας μικρής οντολογίας θα μπορούσε να είναι ένα αυτοκίνητο και ο οδηγός του, τα οποία ως οντότητες συνδέονται μεταξύ τους μέσω σχέσεων, όπως για παράδειγμα η σχέση *οδηγεί*. Κάποια μοντέλα αναπαράστασης σημασιολογικών εννοιών, τα οποία είναι υπεύθυνα για τη σύνταξη οντολογιών, επιτρέπουν τη ύπαρξη ιεραρχίας μεταξύ των οντοτήτων που περιγράφουν. Έτσι, σε μια οντολογία μπορούν να εισαχθούν δεδομένα με συγκεκριμένη ιεραρχία (ύπαρξη κλάσεων και υποκλάσεων). Παρακάτω βλέπουμε ένα σχεδιάγραμμα μιας σχέσης μεταξύ δύο οντοτήτων η οποία μπορεί να περιγραφεί από μία οντολογία.



Σχήμα 2.1: Μια 'τριπλέτα' (triple) στο Semantic Web

¹Πηγή: http://en.wikipedia.org/wiki/Semantic_Web

Όπως βλέπουμε στην εικόνα, μεταξύ των δύο οντοτήτων παρεμβάλλεται η σχέση που τα συνδέει. Το σύνολο οντότητα(subject)-ιδιότητα(predicate)-οντότητα(object), στο Σημασιολογικό Ιστό ονομάζεται τριπλέτα (triplet) και είναι ο κύριος μηχανισμός περιγραφής σχέσεων οντοτήτων. Στο παράδειγμά μας έχουμε δύο οντότητες που ανήκουν στις κλάσεις Driver και Car και συνδέονται με την ιδιότητα drives.

Ένα ευρέως χρησιμοποιούμενο μοντέλο σημασιολογικών εννοιών για την ανάπτυξη οντολογιών είναι το RDF (Resource Description Framework)², το οποίο χρησιμοποιεί τη σύνταξη της XML (Extensible Markup Language)³ για να περιγράψει την ζητούμενη πληροφορία σε μορφή επεξεργάσιμη από υπολογιστές. Κάθε οντότητα και ιδιότητα χαρακτηρίζεται μοναδικά από το URI (Uniform Resource Identifier)⁴ της, το οποίο μπορεί να είναι κάποια τοποθεσία στο δίκτυο που χαρακτηρίζει το εν λόγω αντικείμενο/ιδιότητα. Το URI είναι ένας καθολικός τρόπος ταυτοποίησης των πόρων που περιγράφει μία οντολογία. Ένα παράδειγμα μιας τριπλέτας subject - predicate - object γραμμένο σε RDF είναι το εξής:

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:action="http://example.tuc.gr/action#"
  xmlns:person="http://example.tuc.gr/person#"
  xmlns:cars="http://example.tuc.gr/cars#">

  <rdf:Description rdf:about="person:Nick">
    <action:drives>cars:BMW</action:drives>
  </rdf:Description>
</rdf:RDF>
```

Στο παράδειγμα παρουσιάζεται το υποκείμενο με URI <http://example.tuc.gr/person#Nick> να έχει την ιδιότητα <http://example.tuc.gr/action#drives> πάνω στο αντικείμενο <http://example.tuc.gr/cars#BMW>. Η εντολή "xmlns:rdf=" ορίζει ένα χώρο ονομάτων (namespace) για την κάθε οντότητα/ιδιότητα ώστε να χρησιμοποιείται για συντομία. Στο πεδίο του αντικειμένου της σχέσης μπορεί να βρισκείται κάποια τιμή ενός τύπου δεδομένων, όπως String, integer, float, dateType κ.α.

Μια άλλη γλώσσα περιγραφής που χρησιμοποιείται στο Σημασιολογικό Ιστό είναι το RDFS (Resource Description Framework Schema)⁵, το οποίο ουσιαστικά επεκτείνει το RDF, δίνοντάς μας τη δυνατότητα να αναπαριστούμε ιεραρχικά τις οντότητες με τη μορφή κλάσεων και υποκλάσεων. Αυτό επιτυγχάνεται δηλώνοντας σαν ιδιότητα ενός υποκειμένου την rdf:Type και σαν αντικείμενο την rdfs:Class. Η RDFS, επίσης παρέχει τη δυνατότητα ιεραρχίας των ιδιοτήτων (properties), με την κλάση rdf:Property. Οι κυριολεκτικές τιμές (αριθμοί, ημερομηνίες, κτλ) που μπορούν να δοθούν σε κάποιο υποκείμενο μιας τριπλέτας, περιγράφονται από την κλάση rdfs:Literal. Οι ιδιότητες που δηλώνουμε για κάποια κλάση ισχύουν για όλες τις υποκλάσεις της και για όλα τα στιγμιότυπά (individuals) της. Ένα παράδειγμα δήλωσης ιεραρχίας σε RDFS είναι το εξής:

²<http://www.w3.org/RDF/>

³<http://www.w3.org/XML/>

⁴http://en.wikipedia.org/wiki/Uniform_Resource_Identifier

⁵<http://www.w3.org/TR/rdf-schema/>

```
objects:cars rdf:type rdfs:Class
cars:BMW rdf:type objects:cars6
```

Το RDFS επίσης δίνει τη δυνατότητα ύπαρξης πεδίου ορισμού και πεδίου τιμών (domain, range) για μία ιδιότητα μίας τριπλέτας, με τα `rdfs:domain` και `rdfs:range` να τα δηλώνουν αντίστοιχα. Υποστηρίζεται και ιεραρχία μεταξύ των ιδιοτήτων με τη χρήση του `rdfs:subPropertyOf`, ενώ η δήλωση υποκλάσεων γίνεται με τη χρήση του `rdfs:subClassOf`.

Στην εργασία αυτή για την αναπαράσταση/δημιουργία/επεξεργασία των οντολογιών, χρησιμοποιούμε την γλώσσα OWL (Ontology Web Language)⁷, η οποία είναι μια γλώσσα αναπαράστασης γνώσης που επεκτείνει τις RDF/RDFS, ώστε να δοθεί μεγαλύτερη περιγραφικότητα στις οντότητες των οντολογιών και τις σχέσεις μεταξύ τους. Σκοπός είναι η μεγένθυση της δυνατότητας των μηχανών στο να επεξεργάζονται δεδομένα. Οι προσθήκες της OWL είναι οι εξής:

- Προσθήκη δυνατότητας λογικών συνδιασμών μεταξύ των κλάσεων (Ένωση, Τομή, Συμπληρώματα, κτλ.)
- Προσθήκη σχέσεων μεταξύ των κλάσεων, όπως ισοδύναμες (equivalent) ή ξεχωριστές (disjoint)
- Προσθήκη επιπλέον ιδιοτήτων στις ιδιότητες, όπως συμμετρική (symmetric), μεταβατική (transitive) κτλ.
- Προσθήκη πληθικότητας (cardinality) μεταξύ των κλάσεων

Μετά τη δημιουργία μιας οντολογίας σε OWL, οφείλει να γίνει έλεγχος ορθότητάς της (Reasoning). Για το σκοπό αυτόν, έχουν δημιουργηθεί ειδικές εφαρμογές οι οποίες ονομάζονται αιτιολογητές (reasoners). Αιτιολογητής είναι ένα πρόγραμμα το οποίο τρέχει πάνω στην επιλεγμένη οντολογία και είναι υπεύθυνο για τον έλεγχο ορθότητας (consistency) της οντολογίας κοιτάζοντας κάθε φορά εάν καταπατώνται κανόνες, πληθικότητες, ή άλλα χαρακτηριστικά ιδιοτήτων. Ο αιτιολογητής, επίσης, έχει τη δυνατότητα να συμπεραίνει νέες τριπλέτες μεταξύ αντικειμένων, αναλύοντας τις ήδη υπάρχουσες. Στην παρούσα εργασία χρησιμοποιήθηκε ο αιτιολογητής Pellet Reasoner⁸.

Η αναπαράσταση χωρο-χρονικών δεδομένων, λοιπόν στην περίπτωσή μας, παρατηρούμε ότι διευκολύνεται ιδιαίτερα με τη χρήση της OWL. Οι Τοπολογικές Σχέσεις [4], οι Σχέσεις Κατεύθυνσης [3] και οι κλάσεις τις οποίες αυτές αφορούν, όπως αναλύθηκαν παραπάνω, δηλώνονται ως ιδιότητες (properties), ενώ για την αναπαράσταση χρονικών δεδομένων, επιλέχθηκε, ανάμεσα από άλλα μοντέλα, το 4-D Perdurantist (Fluents) [6] μοντέλο, για λόγους που θα δούμε σε επόμενο εδάφιο. Όλα μαζί αυτά συντελούν το χωρο-χρονικό κομμάτι της οντολογίας SOWL, πάνω στην οποία δομήθηκε η συγκεκριμένη εργασία.

⁶Χρησιμοποιήθηκαν χώροι ονομάτων (namespaces) για συντομία

⁷<http://www.w3.org/2004/OWL/>

⁸<http://clarkparsia.com/pellet/>

2.2 Οντολογία SOWL

Όπως αναφέρθηκε, για την εκπόνηση της συγκεκριμένης εργασίας, χρησιμοποιήθηκε η οντολογία SOWL [1], στην οποία αποθηκεύονται όλα τα δεδομένα και οι σχέσεις μεταξύ τους, τα οποία επεξεργάζεται η εφαρμογή. Η SOWL έχει τη δυνατότητα να περιγράψει δυναμικά και στατικά δεδομένα στο χώρο και στο χρόνο, καθώς και να χειριστεί ποιοτικές (qualitative) και ποσοτικές (quantitative) σχέσεις μεταξύ των δεδομένων αυτών. Με την παράλληλη χρήση ενός αιτιολογητή (reasoner), η οντολογία αποκτά συνοχή και νέες χρήσιμες σχέσεις μεταξύ αντικειμένων προκύπτουν. Αυτό διευκολύνεται με τη χρήση μιας γλώσσας κανόνων οι οποίοι εισάγονται στην οντολογία, της SWRL (Semantic Web Rule Language)⁹, που επιτρέπουν τον αιτιολογητή (reasoner) που αναλύει την οντολογία να βγάζει σημαντικά συμπεράσματα για ποιοτικές και ποσοτικές σχέσεις μεταξύ των οντοτήτων. Παρακάτω δίνεται ένα παράδειγμα ενός SWRL κανόνα¹⁰:

```
hasParent(?x1,?x2) ^ hasBrother(?x2,?x3) => hasUncle(?x1,?x3)
```

Ο παραπάνω κανόνας δηλώνει ότι εάν στην οντολογία εντοπιστεί η τριπλέτα $x1$ *hasParent* $x2$ και η τριπλέτα $x2$ *hasBrother* $x3$, ο αιτιολογητής που θα αναλύσει την οντολογία θα πρέπει να συμπεράνει και να εισάγει και την τριπλέτα $x2$ *hasUncle* $x3$.

Για την περιγραφή της γνώσης της εξέλιξης στο χρόνο σε οντολογίες υπάρχουν αρκετά μοντέλα, καθ' ένα εκ των οποίων έχει συγκεκριμένα πλεονεκτήματα και μειονεκτήματα. Κάποια χαρακτηριστικά μοντέλα είναι τα εξής:

- Temporal Description Logics [7, 8]
- Reification [1]
- N-ary Relations [9]
- Temporal RDF [10]
- Versioning [11]
- Named Graphs [12]
- 4D-Fluents (Perdurantist) [6]

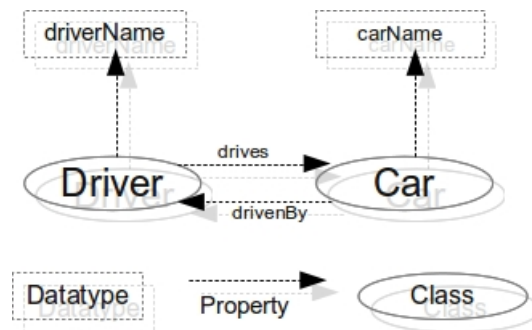
Χαρακτηριστικά είναι τα προβλήματα που εισάγουν οι μέθοδοι αυτές σε μία οντολογία. Για παράδειγμα, συχνό πρόβλημα είναι η περιττότητα δεδομένων (Data Redundancy) που εμφανίζεται στην οντολογία, στην προσπάθεια περιγραφής της εξέλιξης γεγονότων στο χρόνο. Άλλο σημαντικό πρόβλημα με το οποίο ερχόμαστε αντιμέτωποι με κάποιες από τις μεθόδους, είναι η ανάγκη επέκτασης του λεξιλογίου της OWL, στην προσπάθεια επέκτασης της εκφραστικότητας, κάτι το οποίο δεν είναι θεμιτό, καθώς δημιουργεί προβλήματα συμβατότητας.

Μία αρκετά αποδοτική μέθοδος που χρησιμοποιείται από την οντολογία SOWL [1] και συνεπώς χρησιμοποιείται και στην παρούσα εργασία είναι η 4D-Fluents [6] μέθοδος. Τα

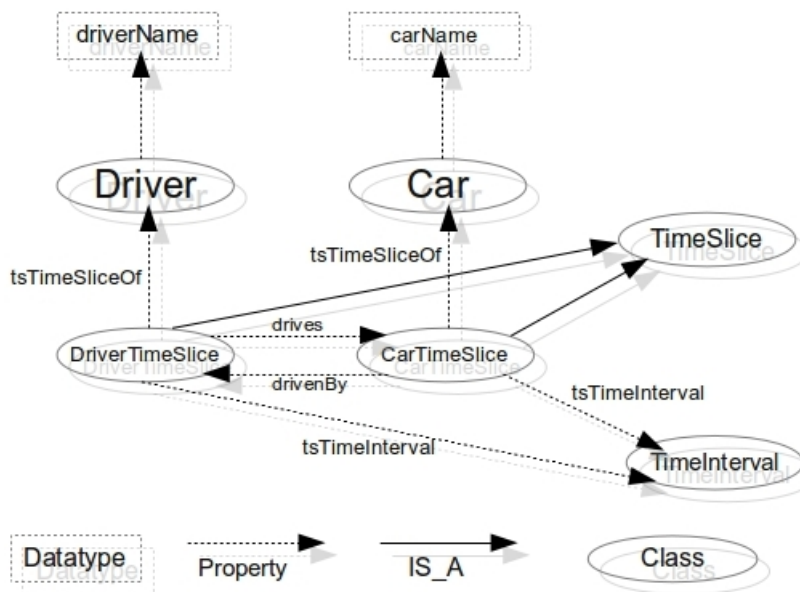
⁹<http://www.w3.org/Submission/SWRL/>

¹⁰http://en.wikipedia.org/wiki/Semantic_Web_Rule_Language

γεγονότα που συμβαίνουν στο χρόνο παρουσιάζονται ως αντικείμενα τεσσάρων διαστάσεων (4η διάσταση είναι ο χρόνος). Για τη μετατροπή μιας οντολογίας σε 4D-Fluents, προστίθενται τρεις νέες κλάσεις, η TimeSlice, η Interval και η Instant. Η κλάση TimeSlice δέχεται στιγμιότυπα (individuals) που έχουν υπόσταση στο χρόνο μέσω της ιδιότητας tsTimeInterval που τα συνδέει με ένα συγκεκριμένο χρονικό διάστημα. Δηλώνει ένα χρονικό κομμάτι ενός αντικειμένου μιας κλάσης. Το κάθε χρονικό κομμάτι, συνδέεται μέσω της ιδιότητας tsTimeSliceOf, με το στιγμιότυπο (individual) το οποίο περιγράφει χρονικά. Μεταξύ των χρονικών κομματιών εισάγονται ιδιότητες, οι οποίες αυτόματα ισχύουν για όσο χρονικό διάστημα υφίσταται το κάθε ένα από αυτά. Τα στιγμιότυπα τα οποία αποκτούν κάποιο χρονικό κομμάτι ονομάζονται δυναμικά, ενώ όσα δεν έχουν χρονική υπόσταση, ονομάζονται στατικά. Εάν υπάρχουν δυναμικά δεδομένα, η οντολογία χαρακτηρίζεται ως δυναμική. Στα Σχήματα 2.2 και 2.3 παρουσιάζονται μία στατική και μία δυναμική οντολογία η οποία χρησιμοποιεί 4D-Fluents.



Σχήμα 2.2: Παράδειγμα στατικής οντολογίας



Σχήμα 2.3: Παράδειγμα οντολογίας 4D-Fluents [6]

Το μοντέλο 4D-Fluents αφήνει ανέπαφες τις κλάσεις της οντολογίας, εισάγει κάποια περιττότητα δεδομένων, αλλά διατηρεί όλη την εκφραστικότητα της OWL, χωρίς την επέκτασή της, διευκολύνοντας, έτσι, την υποστήριξη από διάφορους Reasoners.

Η οντολογία SOWL [1] υποστηρίζει κάποιους χρονικούς τελεστές ώστε να μπορεί ο χρήστης αποδοτικά να περιγράφει ποιοτικές και ποσοτικές σχέσεις μεταξύ των διαφόρων χρονικών στιγμών και χρονικών διαστημάτων που βρίσκονται στην οντολογία. Παρακάτω απαριθμούνται οι εν λόγω τελεστές (τελεστές Allen [5]).

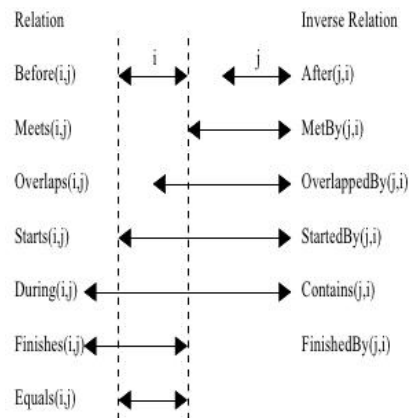
- **Τελεστές Χρονικών Στιγμών**

- **At(timePoint)**: Είναι αληθής εάν υπάρχει χρονικό διάστημα που περιέχει την χρονική στιγμή.
- **StartsAt(timepoint)**: Είναι αληθής εάν υπάρχει χρονικό διάστημα που ξεκινάει την χρονική στιγμή.
- **EndsAt(timepoint)**: Είναι αληθής εάν υπάρχει χρονικό διάστημα που τελειώνει την χρονική στιγμή.
- **Before(timepoint)**: αληθής αληθής εάν υπάρχει χρονικό διάστημα που έχει τελειώσει πριν την χρονική στιγμή.
- **After(timepoint)**: Είναι αληθής εάν υπάρχει χρονικό διάστημα που τελειώνει μετά την χρονική στιγμή.

- **Τελεστές Χρονικών Διαστημάτων**

- **Meets(intervalStarts,intervalEnds)**: Είναι αληθής εάν το πρώτο χρονικό διάστημα συναντά το δεύτερο.
- **MetBy(intervalStarts,intervalEnds)**: Είναι αληθής εάν το δεύτερο χρονικό διάστημα συναντά το πρώτο.
- **Overlaps(intervalStarts,intervalEnds)**: Είναι αληθής εάν το πρώτο χρονικό διάστημα επικαλύπτει το δεύτερο.
- **OverlappedBy(intervalStarts,intervalEnds)**: Είναι αληθής εάν το δεύτερο χρονικό διάστημα επικαλύπτει το πρώτο.
- **During(intervalStarts,intervalEnds)**: Είναι αληθής εάν το πρώτο χρονικό διάστημα συμβαίνει κατά τη διάρκεια του δευτέρου.
- **Contains(intervalStarts,intervalEnds)**: Είναι αληθής εάν το πρώτο χρονικό διάστημα περιέχει το δεύτερο.
- **Starts(intervalStarts,intervalEnds)**: Είναι αληθής εάν τα δύο χρονικά διαστήματα ξεκινούν μαζί.
- **StartedBy(intervalStarts,intervalEnds)**: Είναι αληθής εάν τα δύο χρονικά διαστήματα ξεκινούν μαζί.
- **Ends(intervalStarts,intervalEnds)**: Είναι αληθής εάν τα δύο χρονικά διαστήματα τελειώνουν μαζί.
- **EndedBy(intervalStarts,intervalEnds)**: Είναι αληθής εάν τα δύο χρονικά διαστήματα τελειώνουν μαζί.

- **Equals(intervalStarts,intervalEnds)**: Είναι αληθής εάν τα δύο χρονικά διαστήματα είναι όμοια.
- **AlwaysAt(intervalStarts,intervalEnds)**: Συνδιασμός τεσσάρων τελεστών Allen (Contains, Equals, Started_by, Ended_by). Αν οποιοσδήποτε από αυτούς είναι αληθής τότε και αυτός είναι αληθής.
- **SometimeAt(intervalStarts,intervalEnds)**: Συνδιασμός εννέα τελεστών Allen (Equals, Overlaps, Overlapped_By, Starts, Started_by, Ends, Ended_by, Contains, During). Αν οποιοσδήποτε από αυτούς είναι αληθής τότε και αυτός είναι αληθής.



Σχήμα 2.4: Οι τελεστές Allen [5]για Χρονικά Διαστήματα

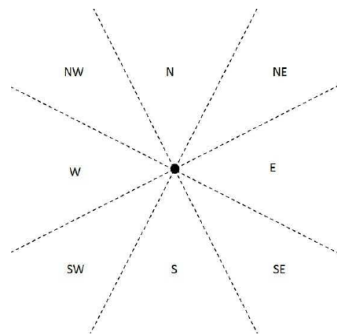
Για την περιγραφή των αντικειμένων (individuals) στο χώρο, η SOWL [1] χρησιμοποιεί τις Σχέσεις Κατεύθυνσης (Directional Relations) [3] και τις Τοπολογικές Σχέσεις (Topologic Relations) [4], που αναφέρθηκαν παραπάνω. Είναι προφανές, πως για την εξαγωγή των απαραίτητων πληροφοριών, χρειαζόμαστε ανά πάσα στιγμή τις συντεταγμένες στην οθόνη των αντικειμένων, ώστε να λάβει χώρα η γεωμετρική τους ανάλυση και να εξαχθούν οι εν λόγω σχέσεις, οι οποίες απαριθμούνται παρακάτω:

- **Σχέσεις Κατεύθυνσης (Directional Relations)**

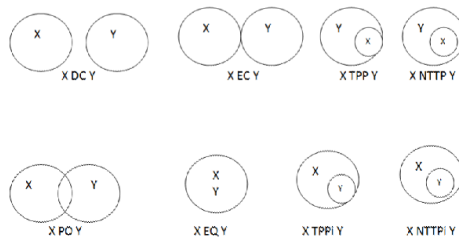
- **A Nof B**: Το A αντικείμενο βρίσκεται Βόρεια του B
- **A NEof B**: Το A αντικείμενο βρίσκεται Βορειοανατολικά του B
- **A Eof B**: Το A αντικείμενο βρίσκεται Ανατολικά του B
- **A SEof B**: Το A αντικείμενο βρίσκεται Νοτιοανατολικά του B
- **A Sof B**: Το A αντικείμενο βρίσκεται Νότια του B
- **A SWof B**: Το A αντικείμενο βρίσκεται Νοτιοδυτικά του B
- **A Wof B**: Το A αντικείμενο βρίσκεται Δυτικά του B
- **A NWof B**: Το A αντικείμενο βρίσκεται Βορειοδυτικά του B
- **A sameXY B**: Το A αντικείμενο έχει το ίδιο κέντρο βάρους με το B

• Τοπολογικές Σχέσεις (Topologic Relations)

- **A DC B (DisConnected)**: Το A αντικείμενο βρίσκεται εκτός του B (Δεν τέμνονται)
- **A EC B (Externally EConnected)**: Το A αντικείμενο εφάπτεται του B και είναι εκτός του B (το B είναι εκτός του A)
- **A TPP B (Tangential Proper Part)**: Το A αντικείμενο εφάπτεται στο B και είναι εντός του B
- **A NTPP B (Non-Tangential Proper Part)**: Το A αντικείμενο δεν εφάπτεται στο B και είναι εντός του B
- **A PO B (Partially Overlapping)**: Το A αντικείμενο τέμνει το B
- **A EQ B (EQual)**: Το A αντικείμενο είναι ακριβώς το ίδιο με το B
- **A TPPi B (Tangential Proper Part inverse)**: Το A αντικείμενο εφάπτεται του B και είναι εκτός του B (το B είναι εντός του A)
- **A NTTPI B (Non-Tangential Proper Part inverse)**: Το A αντικείμενο δεν εφάπτεται του B και είναι εκτός του B (το B είναι εντός του A)



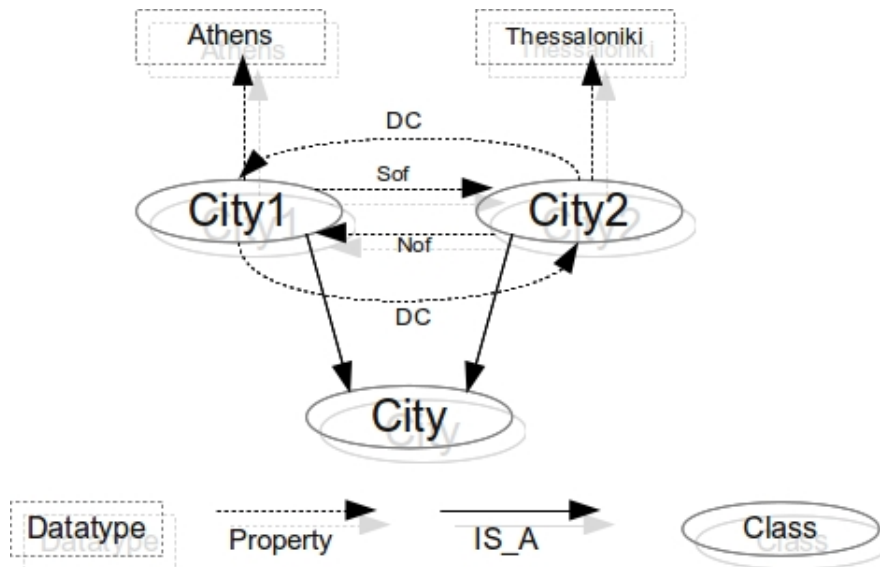
Σχήμα 2.5: Οι Σχέσεις Κατεύθυνσης με κωνική διατόμη επιπέδου [3]



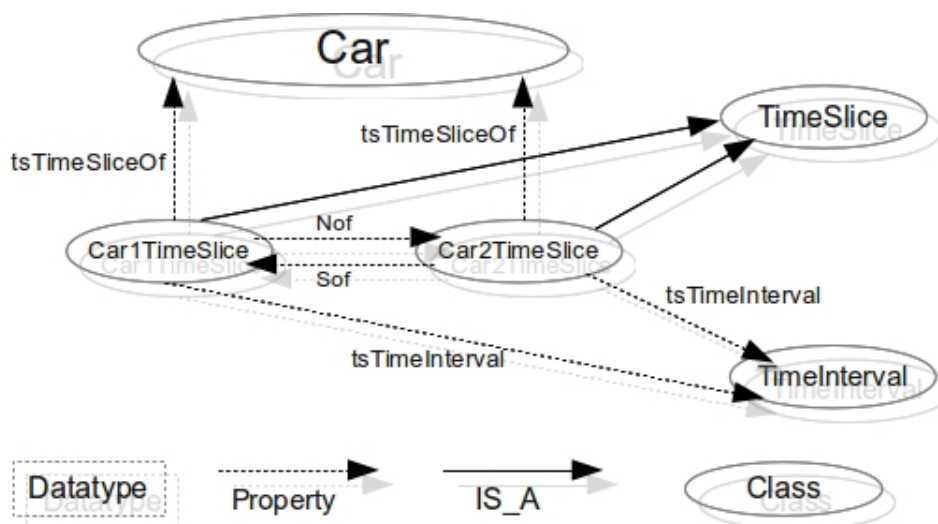
Σχήμα 2.6: RCC-8 Τοπολογικές σχέσεις [4]

Οι Σχέσεις Κατεύθυνσης που εισάγονται στην οντολογία, υπολογίζονται χρησιμοποιώντας τα κέντρα μάζας των χωρικών αντικειμένων τα οποία περιγράφει. Ο υπολογισμός

γίνεται χρησιμοποιώντας ένα μοντέλο κωνικής διατομής του επιπέδου που ορίζει κάθε κέντρο ενός χωρικού αντικειμένου. Θεωρώντας το κέντρο μάζας ενός αντικειμένου ως κέντρο αναφοράς, εξάγουμε τη σχέση κατεύθυνσης που ισχύει μεταξύ αυτού και ενός άλλου αντικειμένου χρησιμοποιώντας τη γωνία την οποία σχηματίζουν μεταξύ τους. Στα σχήματα 2.4 και 2.5 παρουσιάζονται οι RCC-8 Τοπολογικές Σχέσεις και οι Σχέσεις Κατεύθυνσης υπολογισμένες με Κωνική Διατομή Επιπέδου (Cone Shaped Directional Relations). Τα Σχήματα 2.6 και 2.7 απεικονίζουν ένα παράδειγμα οντολογίας που χρησιμοποιεί χωρικές σχέσεις και ένα παράδειγμα οντολογίας με χωρο-χρονικές σχέσεις.



Σχήμα 2.7: Παράδειγμα οντολογίας με χωρικές σχέσεις



Σχήμα 2.8: Παράδειγμα οντολογίας με χωρο-χρονικές σχέσεις

2.3 Γλώσσα Ερωτήσεων SOWL

Εφόσον έχουμε στα χέρια μας μια εμπλουτισμένη οντολογία, είναι χρήσιμο να μπορούμε να κάνουμε ερωτήσεις σχετικά με τα στιγμιότυπα (individuals) τα οποία περιέχει και τις ιδιότητες (properties) που ισχύουν μεταξύ τους. Για το σκοπό αυτό δημιουργήθηκε η SPARQL (SPARQL Protocol and RDF Query Language)¹¹, η οποία είναι μία γλώσσα ερωτήσεων για χρήση σε οντολογίες. Η SPARQL έχει παρόμοια σύνταξη με την SQL¹² και στις ερωτήσεις (queries) της διατυπώνουμε τριπλέτες για των οποίων τα υποείμενα και αντικείμενα θέλουμε να λάβουμε πληροφορίες. Η επέκταση της SPARQL για να επιστρέφει απαντήσεις σε χωροχρονικές ερωτήσεις είναι η Γλώσσα Ερωτήσεων SOWL (SOWL Query Language) [1], η οποία προσθέτει τους χωροχρονικούς τελεστές της οντολογίας SOWL σε μία SPARQL ερώτηση (στην παρούσα περίπτωση, οι χρονικοί τελεστές είναι οι τελεστές Allen [5] και αυτοί του 4D-Fluents μοντέλου [6]). Η Γλώσσα Ερωτήσεων SOWL μεταφράζει τις ερωτήσεις σε SPARQL ερωτήσεις, αφού τις περάσει από στάδια συντακτικής ανάλυσης (parsing) και διερμηνείας (interpreting). Χρησιμοποιώντας το Jena API¹³ σε γλώσσα JAVA, διαβάζει τις οντολογίες, υποβάλλει τις ερωτήσεις που γίνονται και επιστρέφει τα αποτελέσματα σε Result Sets.

Στη γλώσσα SPARQL (και στην SOWL, καθώς έχουνε όμοιο συντακτικό), οι μεταβλητές δηλώνονται τοποθετώντας το αγγλικό ερωτηματικό ("?") πριν το όνομα της μεταβλητής. Οι ερωτήσεις γίνονται πάνω σε τριπλέτες. Οι τριπλέτες που αποτελούν την ερώτηση, χωρίζονται με την τελεία ("."). Οι ιδιότητες, οι κλάσεις και τα στιγμιότυπα δηλώνονται με το URI τους. Για λόγους συντομίας, πριν από την ερώτηση μπορούμε να τοποθετήσουμε μια λίστα από προθέματα ώστε να τα χρησιμοποιήσουμε στην ερώτηση, πριν από κάποιο όνομα κλάσης, ιδιότητας, ή στιγμιότυπου. Παράδειγμα:

```
PREFIX owl:<http://www.w3.org/2002/07/owl#>
PREFIX xsd:<http://www.w3.org/2001/XMLSchema>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX time:<http://www.w3.org/2006/time#>
```

Ερώτηση χωρίς πρόθεμα:

```
select distinct ?x ?y
where{
  ?x http://example.org/example#hasEmployee ?y.
  ?x http://example.org/example#companyName "C1"
}
```

Ερώτηση με πρόθεμα:

```
PREFIX ex1: <http://example.org/example#>

select distinct ?x ?y
where{
  ?x ex1:hasEmployee ?y.
  ?x ex1:companyName "C1" }
```

¹¹<http://www.w3.org/TR/rdf-sparql-query/>

¹²<http://en.wikipedia.org/wiki/SQL>

¹³<http://jena.sourceforge.net/>

Με τη λέξη-κλειδί (keyword) "SELECT", ο χρήστης επιλέγει το υποσύνολο δεδομένων που επιθυμεί να λάβει. Το `distinct` χρησιμοποιείται για να αποφευχθεί η εμφάνιση του ίδιου αποτελέσματος περισσότερες από μία φορές κατά την εμφάνιση των αποτελεσμάτων. Με τη λέξη-κλειδί "FROM", επιλέγεται το σύνολο δεδομένων (dataSet) πάνω στο οποίο γίνεται η ερώτηση, ενώ όταν δε δηλώνεται σύνολο δεδομένων, η ερώτηση γίνεται πάνω στο προεπιλεγμένο σύνολο δεδομένων, το οποίο περιλαμβάνει όλες τις κλάσεις της οντολογίας. Η λέξη κλειδί "WHERE" ακολουθείται από το κύριο σώμα της ερώτησης, το οποίο μπορεί να εμπεριέχει τα εξής:

- Τριπλέτες (triplets)
- Φίλτρα (filters)
- Προαιρετικές εκφράσεις μονοπατιών (optional path expressions)
- Ενώσεις (unions)

Η παραπάνω ερώτηση, στην SOWL μορφή της, αποτελείται από δύο τριπλέτες. Την `?x ex:hasEmployee ?y` και την `?x ex:companyName "C1"`. Σύμφωνα με όσα αναφέρθηκαν, η ερώτηση ζητά εργαζομένους (employees), οι οποίοι να δουλεύουν σε κάποια εταιρία (company) ΚΑΙ η εταιρία αυτή να χει όνομα "C1". Τα φίλτρα, (filters) χρησιμοποιούνται για να εξαλείψουμε, χρησιμοποιώντας λογικές παραστάσεις, ανεπιθύμητα αποτελέσματα, για παράδειγμα:

```
select ?x , ?y
where{
  ?x ex:takesSalary ?y
  filter( ?y > 1000 )
}
```

Η παραπάνω έκφραση θα επιστρέψει αποτελέσματα για τριπλέτες στις οποίες η μεταβλητή `?y` είναι μεγαλύτερη του 1000. Οι προαιρετικές εκφράσεις μονοπατιών (optional path expressions), χρησιμοποιούνται για την προαιρετική αναζήτηση δεδομένων και δηλώνονται με τη λέξη κλειδί "OPTIONAL". Για παράδειγμα, έστω ότι ζητούνται τα δεδομένα όλων των εργαζομένων (employees) που δουλεύουν σε μια εταιρία (company1):

```
select distinct ?name , ?phone ?email
where{
  ?employee ex:worksfor ex:company1;
  ex:employeeName ?name.
  optional { ?employee ex:employeePhone ?phone }
  optional { ?employee ex:employeeEmail ?email }
}
```

Έτσι, εάν κάποιος από τους υπαλλήλους που ζητούνται δεν έχει κάποιο από αυτά τα στοιχεία, θα επιστραφεί ως αποτέλεσμα, κάτι που δεν θα συνέβαινε εάν η ερώτηση είχε συνταχθεί χωρίς τις λέξεις κλειδιά "OPTIONAL". Τέλος, η λέξη κλειδί "UNION", χρησιμοποιείται για να ληφθεί η ένωση από κάποια σύνολα αποτελεσμάτων της ερώτησης. Για παράδειγμα:

```
PREFIX ex1: <http://example.org/company1/employees>
PREFIX ex2: <http://example.org/company2/employees>
select ?name
where {
  { ?employee ex1:employeeName ?name }
  union { ?employee ex2:employeeName ?name }
}
```

Έτσι, ο χρήστης θα λάβει πληροφορίες για υπαλλήλους που δουλεύουν είτε στην πρώτη, είτε στη δεύτερη εταιρία. Όπως αναφέρθηκε, η γλώσσα ερωτήσεων SOWL [1], προσθέτει τους χωροχρονικούς τελεστές της οντολογίας SOWL σε μια SPARQL ερώτηση. Στην ουσία επεκτείνει την εκφραστικότητα του συντακτικού της SPARQL, μεταφράζοντας χωρο-χρονικές ερωτήσεις (οι οποίες περιέχουν χωρο-χρονικούς τελεστές) σε απλές ερωτήσεις SPARQL κάτι που βέβαια αποκρύπτεται από το χρήστη. Παρακάτω παρουσιάζεται μία ερώτηση σε Γλώσσα Ερωτήσεων SOWL και η μετάφρασή της σε SPARQL ερώτηση, με χρονική αναπαράσταση 4D-Fluents [6].

SOWL Query:

```
select distinct ?x ?y
where{
  ?x ex1:hasEmployee ?y.
  ?x ex1:companyName "C1"
}
```

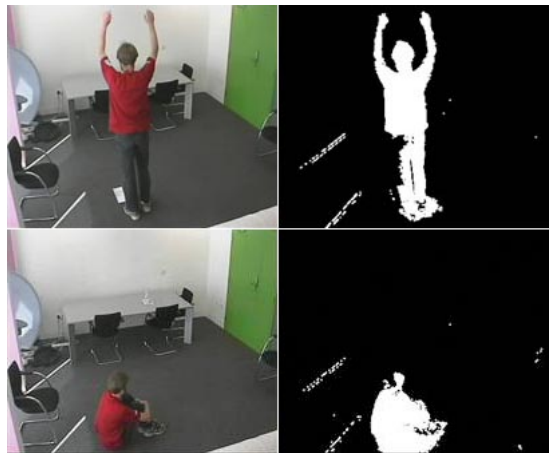
SPARQL Translation (4dfluents) :

```
select distinct ?x ?y where {
{
  ?_timeSlice_0 ex1:tsTimeSliceOf ?x.
  ?_timeSlice_0 ex1:tsTimeInterval ?_interval_0.
  ?_timeSlice_0 ex1:hasEmployee ?_timeSlice_1 .
  ?_timeSlice_1 ex1:tsTimeSliceOf ?y.
  ?_timeSlice_1 ex1:tsTimeInterval ?_interval_0
} union {
  ?x ex1:hasEmployee ?y.
  optional { ?_temporalVar rdf:type ex1:TimeSlice. ?x rdf:type ex1:TimeSlice }
  filter( !bound(?_temporalVar) )}. ?x ex1:companyName "C1" }
```

2.4 Ανάλυση Κίνησης σε Video (Motion Analysis)

Για την υλοποίηση της εργασίας χρησιμοποιήθηκαν video στα οποία είχε γίνει ανάλυση και εντοπισμός των κινούμενων αντικειμένων τους. Ο αλγόριθμος που χρησιμοποιήθηκε [2] είναι προορισμένος για επεξεργαστή Ψηφιακής Επεξεργασίας Σήματος (DSP). Γι' αυτό το λόγο χρησιμοποιήθηκε ένα κλειδί USB (hasp) που προσομοιώνει ένα τέτοιο επεξεργαστή. Ο κώδικας είναι γραμμένος σε C++, με τη χρήση της βιβλιοθήκης MIL (Matrox Imaging Library)¹⁴. Το λογισμικό αυτό, χρησιμοποιεί δυο τεχνικές οι οποίες επιτρέπουν την αναγνώριση κίνησης μέσα σε ένα video. Η πρώτη πραγματοποιεί αφαίρεση του τρέχοντος πλαισίου (frame) από το προηγούμενο, ώστε να εντοπιστούν διαφορές (Background Subtraction). Η δεύτερη τεχνική, αφορά τον εντοπισμό ακμών των αντικειμένων μέσα στο τρέχον πλαίσιο (Edge Detection).

- **Αφαίρεση Υπόβαθρου (Background Subtraction)** : Επιλέγεται αρχικά ένα κενό πλαίσιο φόντου, το οποίο θα είναι το πλαίσιο αναφοράς. Έπειτα υλοποιείται αριθμητική αφαίρεση μεταξύ των εικονοστοιχείων (pixels) του τρέχοντος πλαισίου και του πλαισίου αναφοράς. Ανάλογα με ένα κατώφλι (threshold), το οποίο ορίζεται στην αρχή του κώδικα, η διαφορά μεταξύ των αντίστοιχων εικονοστοιχείων κρίνεται ως σημαντική ή όχι. Σημαντική διαφορά, έχει ως αποτέλεσμα την κατηγοριοποίηση του σημείου ως σημείο ενδιαφέροντος, καθώς η διαφορά ίσως οφείλεται στην ύπαρξη κίνησης. Το πλαίσιο αναφοράς αλλάζει περιοδικά (background update), καθώς η πάροδος του χρόνου στο video μπορεί να εισάγει θόρυβο, όπως διαφορά φωτισμού ή κίνηση κάμερας. Ο αλγόριθμος παρέχει και έναν μηχανισμό αυτόματου επαναπροσδιορισμού του κατωφλίου ανάλογα με την αλλαγή φωτισμού ή την προσθήκη επιπλέον θορύβου. Το Σχήμα 2.8 απεικονίζει το πλαίσιο ενός video που έχει υποστεί Αφαίρεση Υπόβαθρου (Background Subtraction) [2]. Τα εικονοστοιχεία με λευκό χρώμα δηλώνουν το κινούμενο αντικείμενο του οποίου η κίνηση έχει εντοπιστεί¹⁵.

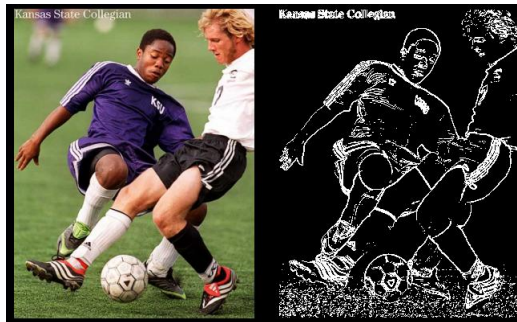


Σχήμα 2.9: Παράδειγμα πλαισίου video που έχει υποστεί Αφαίρεση Υπόβαθρου (Background Subtraction)

¹⁴<http://www.matrox.com/imaging/en/products/software/>

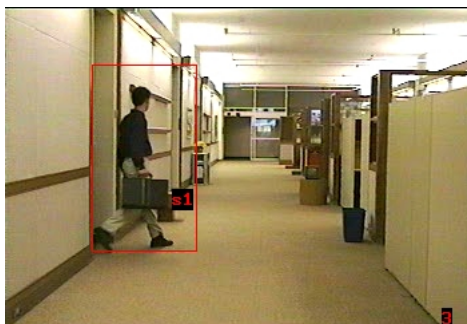
¹⁵Πηγή: http://www.ubicomp.nileu.edu.eg/Projects/Pervasive_Healthcare_Delivery.html

- **Ανίχνευση Ακμών (Edge Detection)** : Η ανίχνευση ακμών, ελέγχει γειτονικά εικονοστοιχεία σε κάθε πλαίσιο για έντονες διαφορές μεταξύ τους, και ανάλογα δημιουργεί περιγράμματα αντικειμένων μέσα στο video. Κάθε εικονοστοιχείο υποδιαίρεται στα τρία βασικά χρώματα, (κόκκινο, πράσινο και μπλε) και μέσω ενός Sobel¹⁶ φίλτρου εντοπίζονται οι ακμές. Το Sobel φίλτρο εντοπίζει τις ακμές στα σημεία στα οποία η μετάβαση από το ένα χρώμα στο άλλο γίνεται πολύ έντονα. Το Σχήμα 2.9 απεικονίζει το πλαίσιο ενός video που έχει υποστεί Ανίχνευση Ακμών (Edge Detection) [2]. Τα εικονοστοιχεία με λευκό χρώμα δηλώνουν τις ακμές του αντικειμένου του οποίου η κίνηση έχει εντοπιστεί¹⁷.



Σχήμα 2.10: Παράδειγμα πλαισίου video που έχει υποστεί Ανίχνευση Ακμών (Edge Detection)

Ο αλγόριθμος δημιουργεί περιβάλλοντα παραλληλόγραμμα (MBRs) γύρω από τα κινούμενα αντικείμενα (blobs) τα οποία εντοπίζει κάθε στιγμή και τα διατηρεί, μέχρι το αντικείμενο να πάψει να υπάρχει στο οπτικό πεδίο της κάμερας. Δημιουργεί ένα αρχείο εξόδου στο οποίο τυπώνει τον αριθμό του κάθε αντικειμένου που έχει εντοπίσει, τον αριθμό του πλαισίου στο οποίο το εντόπισε και τις συντεταγμένες της διαγωνίου του. Το αρχείο αυτό χρησιμοποιείται από την εφαρμογή η οποία αναπτύχθηκε στα πλαίσια της εργασίας. Το Σχήμα 2.10 απεικονίζει το πλαίσιο ενός video στο οποίο έχει εφαρμοστεί Αφαίρεση Υπόβαθρου (Background Subtraction) και Ανίχνευση Ακμών (Edge Detection) και έχουν εντοπιστεί τα κινούμενα αντικείμενα (blobs). Διακρίνονται τα περιβάλλοντα παραλληλόγραμμα (MBRs) γύρω από κάθε κινούμενο αντικείμενο (blob) [2].



Σχήμα 2.11: Παράδειγμα πλαισίου video στο οποίο έχει εντοπιστεί ένα κινούμενο αντικείμενο

¹⁶http://en.wikipedia.org/wiki/Sobel_operator

¹⁷Πηγή: <http://archive.gamedev.net/reference/programming/features/imageproc/>

Κεφάλαιο 3

Υλοποίηση Συστήματος

Στην παρούσα ενότητα θα γίνει αναλυτική περιγραφή της αρχιτεκτονικής του συστήματος που αναπτύχθηκε. Η εφαρμογή ονομάζεται *SVI (Semantic Video Interface)* και τρέχει πάνω σε ένα γραφικό περιβάλλον το οποίο αναπτύχθηκε ώστε να έχει τη δυνατότητα ο χρήστης να επιλέγει τις διάφορες λειτουργίες με εύκολο και αποδοτικό τρόπο. Στο Σχήμα 3.1 φαίνεται το γενικό σχηματικό διάγραμμα (Block Diagram) λειτουργίας της εφαρμογής. Στο διάγραμμα απεικονίζονται οι κύριες κλάσεις που συνεργάζονται για να μας δώσουν το επιθυμητό αποτέλεσμα. Θα γίνει μια αναλυτική περιγραφή των κλάσεων, καθώς και του τρόπου με τον οποίον αυτές επικοινωνούν μεταξύ τους.

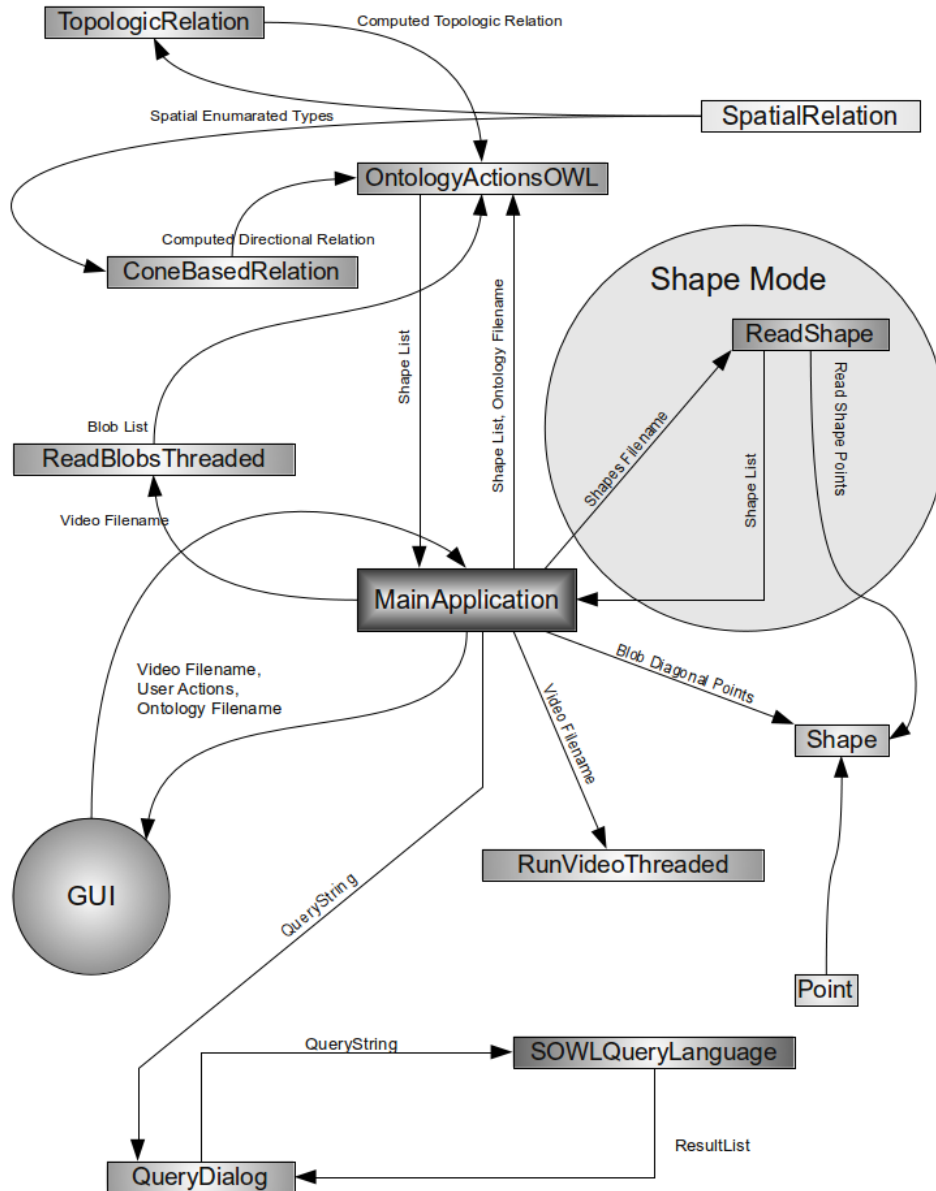
Κλάση Shape

Η κλάση αυτή περιγράφει τον τύπο δεδομένων για τα Χωρικά Αντικείμενα (Spatial Objects) που λαμβάνουμε από το κάθε video που αναλύεται. Αφού αυτά αντικειμενοποιηθούν στον τύπο δεδομένων Shape, αποθηκεύονται σε διάγραμμα για να γίνει σε επόμενο στάδιο η εξαγωγή των μεταξύ τους σχέσεων στο χώρο. Η κλάση Shape δημιουργεί στιγμιότυπα πολυγώνων. Περιέχει ένα σύνολο μεταβλητών, οι οποίες χαρακτηρίζουν ένα πολύγωνο. Οι μεταβλητές αυτές είναι οι εξής:

- **shapeNumber**: Ο αριθμός του πολυγώνου
- **noOfPoints**: Ο αριθμός σημείων του πολυγώνου
- **shapePoints**: Πίνακας που κρατάει τα σημεία του κάθε πολυγώνου
- **massCenter**: Το κέντρο μάζας του πολυγώνου

Η κλάση παρέχει μία μέθοδο για τον υπολογισμό του περιβάλλοντος παραλληλογράμμου (Minimum Bounding Rectangle) του κάθε πολυγώνου, υπολογίζοντας τη διαγωνιά του. Η διαγωνιά του περιβάλλοντος παραλληλογράμμου εντοπίζεται συγκρίνοντας όλα τα σημεία του πολυγώνου μεταξύ τους και επιλέγοντας τα εξής:

- **Μέγιστη Τετμημένη, Ελάχιστη Τεταγμένη**: Το επάνω δεξιά σημείο της διαγωνιάς



Σχήμα 3.1: Γενικό Σχηματικό Διάγραμμα (Block Diagram) της εφαρμογής

- **Ελάχιστη Τετμημένη, Μέγιστη Τεταγμένη:** Το κάτω αριστερά σημείο της διαγωνίου

Μία ακόμη χρήσιμη συνάρτηση που παρέχει η κλάση, είναι αυτή του υπολογισμού κέντρου μάζας του πολυγώνου [14]. Το κέντρο μάζας υπολογίζεται γεωμετρικά, χρησιμοποιώντας τις συντεταγμένες του κάθε σημείου του πολυγώνου, σύμφωνα με τους εξής τύπους (για x και y συντεταγμένες του κέντρου βάρους):

$$C_x = \sum \frac{C_{i,x} A_i}{A_i} \quad (3.1)$$

$$C_y = \sum \frac{C_{i,y}A_i}{A_i} \quad (3.2)$$

Όπου A_i είναι η απόσταση μεταξύ δύο συνεχόμενων σημείων και $C_{i,x}$ η x συντεταγμένη του μέσου της κάθε πλευράς του πολυγώνου.¹

Κλάση Point

Η κλάση Point παρέχει τον τύπο δεδομένων για τα σημεία στο χώρο και κρατάει τις συντεταγμένες τους (x,y). Επιστρέφει με δύο απλές μεθόδους την τετμημένη και την τεταγμένη ξεχωριστά.

Κλάση SpatialRelation

Η κλάση αυτή περιέχει απαριθμήσεις (enumerations) των χωρικών σχέσεων τις οποίες θα χρησιμοποιήσουμε. Διευκολύνει την πρόσβαση στις σχέσεις αυτές στα διάφορα σημεία του κώδικα της εφαρμογής. Οι απαριθμήσεις περιέχουν τις Σχέσεις Κατεύθυνσης [3] και τις Τοπολογικές Σχέσεις [4]. Δηλώνονται αντίστοιχα ως DirectionalRelations και TopologicRelations.

Κλάση ConeBasedRelation

Στο αρχείο αυτό, γίνεται υπολογισμός των Σχέσεων Κατεύθυνσης με Κωνική Διατομή (Cone Shaped Directional Relations) [3], όπως αναφέρθηκε στο Κεφάλαιο 2. Η κλάση αποτελείται από μία μέθοδο, η οποία δέχεται σαν ορίσματα δύο στιγμιότυπα της κλάσης Shape, τα οποία αντιπροσωπεύουν δύο χωρικά αντικείμενα (spatial objects). Τα χωρικά αυτά αντικείμενα μπορεί να προέρχονται από την ανάλυση κίνησης η οποία εφαρμόστηκε σε ένα video. Αφού λάβουμε τα κέντρα μάζας των αντικειμένων, υπολογίζουμε τη γωνία μεταξύ τους ως εξής: Αρχικά υπολογίζουμε την κλίση της ευθείας που σχηματίζουν τα δύο κέντρα μάζας. Έπειτα λαμβάνουμε τη γωνία σε ακτίνια (rad) μέσω της συνάρτησης *Math.atan()*. Τέλος μετατρέπουμε τη γωνία του τόξου σε μοίρες πολλαπλασιάζοντας με 180 και διαιρώντας με τον αριθμό π. Παρακάτω δίνεται το κομμάτι κώδικα που εφαρμόζει τους εν λόγω υπολογισμούς.

```
double tanTheta =
((double)center2.getY()-(double)center1.getY())/((double)center2.getX()-(double)center1.getX());
double arctan = Math.atan((double)tanTheta);
double angle = arctan*180/Math.PI;
```

Στο Σχήμα 3.2 απεικονίζονται οι γωνίες που προκύπτουν μεταξύ των κεντρών μάζας δύο σχημάτων και η Σχέση Κατεύθυνσης με Κωνική Διατομή (Cone Shaped Directional

¹<http://en.wikipedia.org/wiki/Centroid>

Relations) [3] που εξάγεται σύμφωνα με τα παρακάτω:

- **Θετική Γωνία**

- **Γωνία $\leq 22.5^\circ$**

- * Εάν το κέντρο του σχήματος 2 είναι δεξιά του κέντρου του σχήματος 1:
Σχήμα2 Eof Σχήμα1
- * Εάν το κέντρο του σχήματος 2 είναι αριστερά του κέντρου του σχήματος 1:
Σχήμα2 Wof Σχήμα1

- **Γωνία $> 22.5^\circ$ και $\leq 67.5^\circ$**

- * Εάν το κέντρο του σχήματος 2 είναι δεξιά του κέντρου του σχήματος 1:
Σχήμα2 NEof Σχήμα1
- * Εάν το κέντρο του σχήματος 2 είναι αριστερά του κέντρου του σχήματος 1:
Σχήμα2 SWof Σχήμα1

- **Γωνία $> 67.5^\circ$**

- * Εάν το κέντρο του σχήματος 2 είναι δεξιά του κέντρου του σχήματος 1:
Σχήμα2 Nof Σχήμα1
- * Εάν το κέντρο του σχήματος 2 είναι αριστερά του κέντρου του σχήματος 1:
Σχήμα2 Sof Σχήμα1

- **Αρνητική Γωνία**

- **Γωνία $\geq -22.5^\circ$**

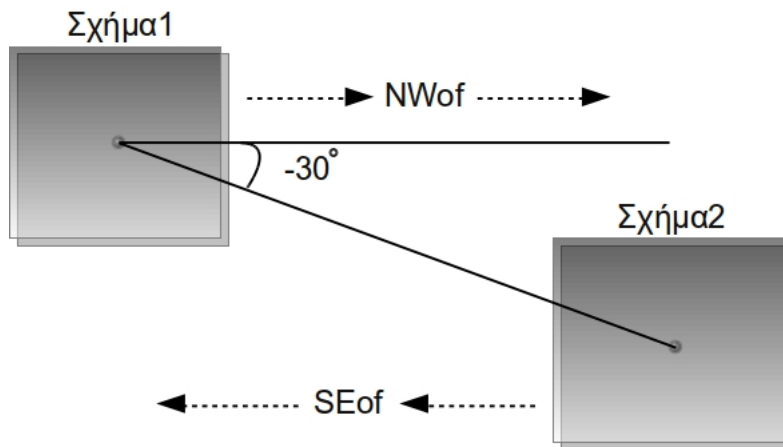
- * Εάν το κέντρο του σχήματος 2 είναι δεξιά του κέντρου του σχήματος 1:
Σχήμα2 Eof Σχήμα1
- * Εάν το κέντρο του σχήματος 2 είναι αριστερά του κέντρου του σχήματος 1:
Σχήμα2 Wof Σχήμα1

- **Γωνία $< -22.5^\circ$ και $\geq -67.5^\circ$**

- * Εάν το κέντρο του σχήματος 2 είναι δεξιά του κέντρου του σχήματος 1:
Σχήμα2 SEof Σχήμα1
- * Εάν το κέντρο του σχήματος 2 είναι αριστερά του κέντρου του σχήματος 1:
Σχήμα2 NWof Σχήμα1

- **Γωνία $< -67.5^\circ$**

- * Εάν το κέντρο του σχήματος 2 είναι δεξιά του κέντρου του σχήματος 1:
Σχήμα2 Sof Σχήμα1
- * Εάν το κέντρο του σχήματος 2 είναι αριστερά του κέντρου του σχήματος 1:
Σχήμα2 Nof Σχήμα1



Σχήμα 3.2: Παράδειγμα εξαγωγής Σχέσης Κατεύθυνσης με Κωνική Διατομή

Στο παραπάνω Σχήμα η γωνία μεταξύ των δύο κεντρών μάζας είναι -30° . Επομένως, σύμφωνα με τα παραπάνω, το Σχήμα2 είναι Νοτιο-Ανατολικά του Σχήμα1, καθώς το κέντρο μάζας του βρίσκεται δεξιά του κέντρου μάζας του Σχήμα1. Αντίστοιχα, το Σχήμα1 είναι Βόρειο-Δυτικά του Σχήμα2. Στις περιπτώσεις οι οποίες δεν ανήκουν στις παραπάνω, θεωρούμε ότι τα κέντρα μάζας είναι όμοια και επιτρέφουμε την τιμή *sameXY*. Η μέθοδος επιστρέφει απαρίθμηση (enumeration) των σχέσεων όπως περιγράφηκαν στην κλάση SpatialRelation.

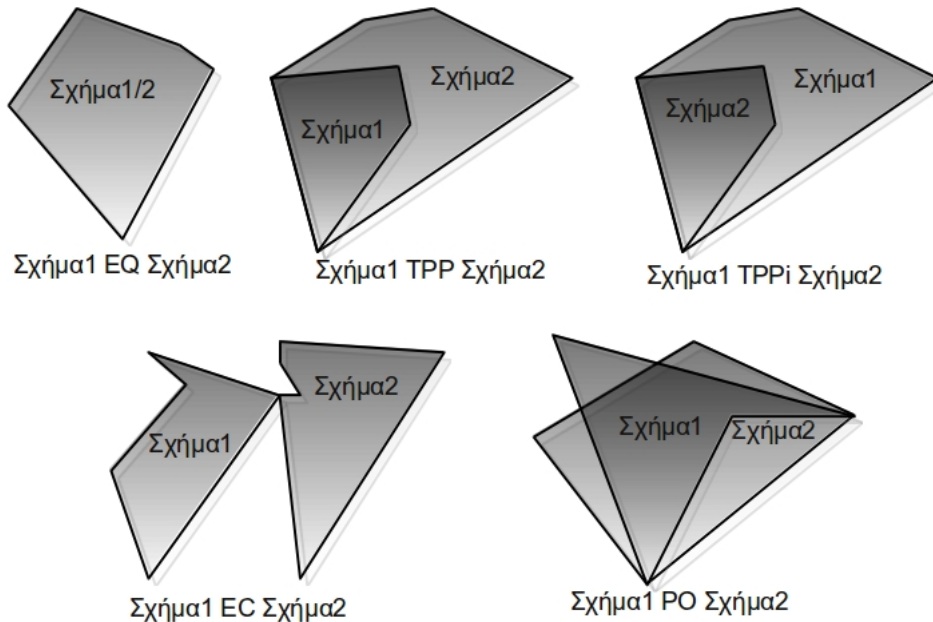
Κλάση TopologicRelation

Η κλάση αυτή, αποτελείται από μία βασική μέθοδο η οποία δέχεται δύο στιγμιότυπα της κλάσης Shape, τα οποία είναι χωρικά αντικείμενα που έχουν προκύψει από την ανάλυση του video. Περιέχει τρεις επιπλέον εσωτερικές μεθόδους οι οποίες χρησιμεύουν στους υπολογισμούς που λαμβάνουν χώρα. Η βασική μέθοδος, εφαρμόζει γεωμετρικούς υπολογισμούς στις συντεταγμένες των σημείων των σχημάτων, ώστε να εξαγάγει τις κατάλληλες RCC-8 Τοπολογικές Σχέσεις [4] που ισχύουν μεταξύ των δύο χωρικών αντικειμένων.

Διακρίνουμε δύο περιπτώσεις σχέσεων μεταξύ των χωρικών αντικειμένων που ελέγχονται, όσον αφορά τα σημεία των πολυγώνων τους:

- **Τα πολύγωνα έχουν κοινά σημεία :**
 - Εάν τα σημεία είναι όλα κοινά, τα σχήματα είναι όμοια και έχουμε RCC-8 ιδιότητα $\Sigma\chi\eta\mu a1 EQ \Sigma\chi\eta\mu a2$.
 - Εάν όλα τα σημεία του σχήματος 1 που δεν είναι κοινά με αυτά του σχήματος 2 βρίσκονται στο εσωτερικό του σχήματος 2, έχουμε RCC-8 ιδιότητα $\Sigma\chi\eta\mu a1 TPP \Sigma\chi\eta\mu a2$.
 - Εάν όλα τα σημεία του σχήματος 2 που δεν είναι κοινά με αυτά του σχήματος 1 βρίσκονται στο εσωτερικό του σχήματος 1, έχουμε RCC-8 ιδιότητα $\Sigma\chi\eta\mu a1 TPPi \Sigma\chi\eta\mu a2$.

- Εάν όλα τα σημεία του σχήματος 1 που δεν είναι κοινά με αυτά του σχήματος 2 βρίσκονται στο εξωτερικό του σχήματος 2, και αυτά του σχήματος 2 στο εξωτερικό του σχήματος 1, έχουμε RCC-8 ιδιότητα *Σχήμα1 EC Σχήμα2*.
- Σε όλες τις άλλες περιπτώσεις, τα σχήματα τέμνονται και έχουμε *Σχήμα1 PO Σχήμα2*.

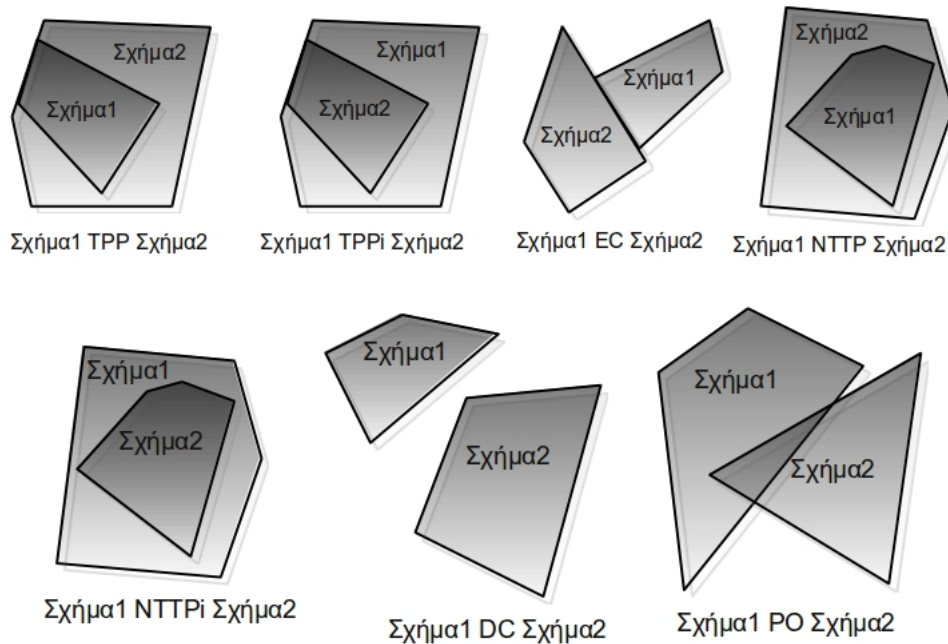


Σχήμα 3.3: RCC-8 Τοπολογικές Σχέσεις με τα πολύγωνα να έχουν κοινά σημεία μεταξύ τους

• Τα πολύγωνα δεν έχουν κοινά σημεία :

- Εάν κάποια σημεία του σχήματος 1 βρίσκονται πάνω σε κάποια γραμμή του σχήματος 2 χωρίς να υπάρχει διατομή γραμμών και τα υπόλοιπα σημεία είναι στο εσωτερικό του, έχουμε RCC-8 ιδιότητα *Σχήμα1 TPP Σχήμα2*.
- Εάν κάποια σημεία του σχήματος 2 βρίσκονται πάνω σε κάποια γραμμή του σχήματος 1 χωρίς να υπάρχει διατομή γραμμών και τα υπόλοιπα σημεία είναι στο εσωτερικό του, έχουμε RCC-8 ιδιότητα *Σχήμα1 TPPi Σχήμα2*.
- Εάν κάποια σημεία του σχήματος 1 βρίσκονται πάνω σε κάποια γραμμή του σχήματος 2 χωρίς να υπάρχει διατομή γραμμών και τα υπόλοιπα σημεία είναι στο εξωτερικό του, ενώ τα ίδια ισχύουν και για το σχήμα 2, έχουμε RCC-8 ιδιότητα *Σχήμα1 EC Σχήμα2*.
- Εάν όλα τα σημεία του σχήματος 1 βρίσκονται στο εσωτερικό του σχήματος 2, έχουμε RCC-8 ιδιότητα *Σχήμα1 NTTP Σχήμα2*.
- Εάν όλα τα σημεία του σχήματος 2 βρίσκονται στο εσωτερικό του σχήματος 1, έχουμε RCC-8 ιδιότητα *Σχήμα1 NTTPi Σχήμα2*.
- Εάν όλα τα σημεία του σχήματος 1 βρίσκονται στο εξωτερικό του σχήματος 2, ενώ το ίδιο ισχύει και για τα σημεία του σχήματος 2, έχουμε RCC-8 ιδιότητα *Σχήμα1 DC Σχήμα2*.

- Σε όλες τις άλλες περιπτώσεις, τα σχήματα τέμνονται και έχουμε $RCC-8$ ιδιότητα $Σχ\acute{\eta}\mu\alpha 1 PO Σχ\acute{\eta}\mu\alpha 2$.

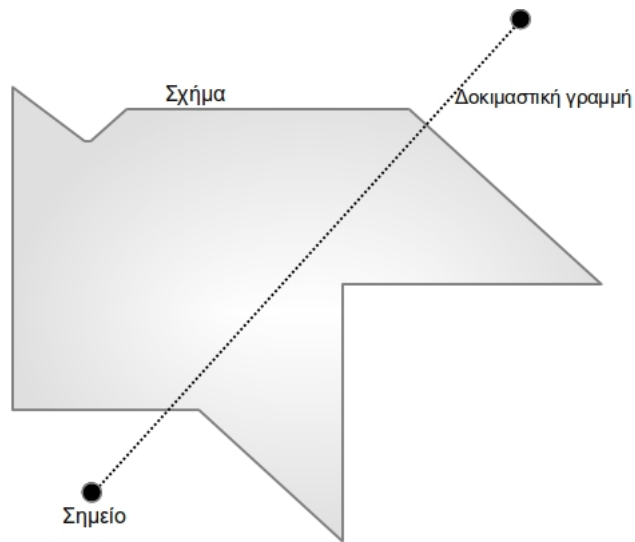


Σχ\acute{\eta}\mu\alpha 3.4: $RCC-8$ Τοπολογικές Σχ\acute{\eta}\σεις με τα πολ\acute{\upsilon}\gammaωνα να μην \acute{\epsilon}\χουν κοιν\acute{\alpha} σημεία μεταξύ τους

Στην κλάση αυτή υλοποιήθηκαν τρεις ακόμα εσωτερικές μέθοδοι, οι οποίες χρησιμοποιούνται από την κύρια μέθοδο για τον υπολογισμό των $RCC-8$ σχέσεων μεταξύ των πολυγώνων των δύο αντικειμένων. Η πρώτη μέθοδος υπολογίζει εάν ένα σημείο βρίσκεται μέσα σε ένα πολ\acute{\upsilon}\γωνο, ενώ η δεύτερη υπολογίζει εάν δύο γραμμές τέμνονται. Η τρίτη μέθοδος, υπολογίζει, δοθέντων τριών σημείων, εάν πηγαίνοντας από το πρώτο στο δεύτερο και από το δεύτερο στο τρίτο, κάνουμε στροφή σύμφωνα με την κίνηση των δεικτών του ρολογιού, ή αντίθετη από αυτήν (Reference: "Algorithms in C", R. Sedgwick) [13]. Οι μέθοδοι περιγράφονται παρακάτω:

- **pointInPolygon()** [13] : Η μέθοδος αυτή εντοπίζει εάν ένα δοσμένο σαν όρισμα σημείο (τύπος δεδομένων Point) βρίσκεται στο εσωτερικό ενός δοσμένου σαν όρισμα σχήματος (τύπος δεδομένων Shape). Η μέθοδος θεωρεί μία τυχαία ευθεία γραμμή (τύπος δεδομένων Line) μεγαλύτερη από όλες τις πιθανές γραμμές που μπορούν να δημιουργηθούν. Αυτό επιτυγχάνεται δημιουργώντας ένα σημείο, η τεταγμένη, ή η τετμημένη του οποίου είναι μεγάλος ακέραιος αριθμός. Η ευθεία γραμμή δοκιμής (testline), όπως ονομάζεται, έχει ως αρχή το σημείο που εξετάζεται και τέλος το μακρυνό σημείο το οποίο δημιουργήθηκε. Έπειτα, καλείται η μέθοδος lineIntersection() για κάθε γραμμή του πολυγώνου, ώστε να διαπιστώσει εάν καθεμία από αυτές τέμνει την ευθεία γραμμή δοκιμής. Ο αριθμός των τομών, εάν είναι άρτιος, τότε το σημείο βρίσκεται εκτός του πολυγώνου, ενώ αν είναι περιττός, βρίσκεται εντός. Στο Σχ\acute{\eta}\mu\alpha 3.5 απεικονίζονται ένα πολ\acute{\upsilon}\γωνο και ένα σημείο το οποίο βρίσκεται εκτός του πολυγώνου.

Ο αριθμός των τομών της δοκιμαστικής γραμμής με τις γραμμές του πολυγώνου είναι άρτιος (=2).



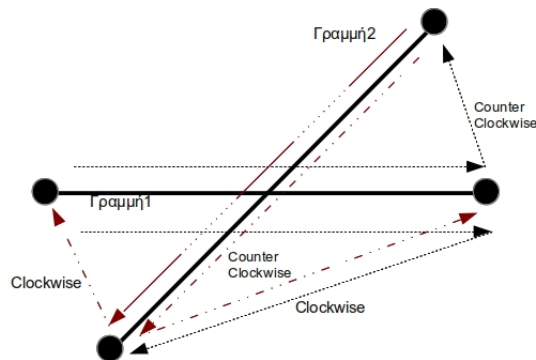
Σχήμα 3.5: Ο τρόπος υπολογισμού εάν ένα σημείο βρίσκεται στο εσωτερικό κάποιου πολυγώνου

- **clockORcounterClock() [13]** : Η μέθοδος αυτή δέχεται σαν ορίσματα τρία σημεία (τύπος δεδομένων Point), και επιστρέφει τιμή +1, εάν ταξιδεύοντας από το πρώτο στο δεύτερο και από το δεύτερο σημείο κάνουμε στροφή με φορά ίδια με αυτήν των δεικτών του ρολογιού και -1 εάν κάνουμε στροφή με αντίθετη φορά. Εάν το τρίτο σημείο βρίσκεται μεταξύ του πρώτου και του δεύτερου επιστρέφει 0. Ο υπολογισμός γίνεται συγκρίνοντας τις κλίσεις που σχηματίζουν οι ευθείες που ορίζουν αυτά τα τρία σημεία.
- **lineIntersection() [13]** : Η μέθοδος αυτή δέχεται δύο γραμμές (τύπος δεδομένων Line) σαν ορίσματα και επιστρέφει true εάν αυτές οι δύο γραμμές τέμνονται. Χρησιμοποιεί την *clockORcounterClock()*. Δοσμένων δύο γραμμών, εάν τα σημεία της δεύτερης, βρίσκονται σε διαφορετικό επίπεδο εκ των δύο επιπέδων που ορίζει η πρώτη γραμμή και το ίδιο ισχύει και για τη δεύτερη, τότε οι γραμμές τέμνονται. Στο Σχήμα 3.6 απεικονίζονται δύο γραμμές οι οποίες τέμνονται, και ο τρόπος μετάβασης από το ένα σημείο στο άλλο.

Κλάση `OntologyActionsOWL`

Η κλάση αυτή είναι υπεύθυνη για όλες τις λειτουργίες που απαιτούν επεξεργασία οντολογιών. Χρησιμοποιεί την OWL API² βιβλιοθήκη γι' αυτόν τον σκοπό. Παρέχει μία μέθοδο για τον υπολογισμό σχέσεων μεταξύ των Χωρικών Αντικειμένων (spatial objects)

²<http://owlapi.sourceforge.net/>



Σχήμα 3.6: Ο τρόπος υπολογισμού της τομής δύο γραμμών

που έχουν εξαχθεί από κάποιο video, καθώς και μια μέθοδο φόρτωσης και καθαρισμού των αντικειμένων (individuals) της οντολογίας. Ο τρόπος λειτουργίας των μεθόδων είναι ο εξής:

- Υπολογισμός Σχέσεων και Εισαγωγή στην Οντολογία (calculateRelationsAndInsert()):** Η μέθοδος αυτή δέχεται τρία ορίσματα: Ένα διάνυσμα που περιέχει στιγμιότυπα της κλάσης Shape (κινούμενα αντικείμενα του video), μία συμβολοσειρά με το όνομα του αρχείου της οντολογίας προς επεξεργασία και μία συμβολοσειρά με την ακριβή χρονική στιγμή την οποία κλήθηκε η μέθοδος. Η τελευταία χρησιμεύει στην τοποθέτηση του χρονικού κομματιού (time slice) του κάθε αντικειμένου στο σωστό χρονικό διάστημα. Το χρονικό διάστημα αυτό έχει διάρκεια ίση με τη διαφορά χρόνου μεταξύ δύο διαδοχικών κλήσεων της μεθόδου, συνεπώς, η αρχή του βρίσκεται στη χρονική στιγμή που δίνεται ως όρισμα και το τέλος του στη χρονική στιγμή που δίνεται ως όρισμα κατά την επόμενη κλήση της μεθόδου. Έτσι, το χρονικό διάστημα μεταξύ δύο διαδοχικών κινήσεων ενός αντικειμένου σε ένα video, θα είναι το χρονικό διάστημα για το οποίο υφίσταται το συγκεκριμένο χρονικό κομμάτι του (time slice). Αρχικά δηλώνονται κάποια χρήσιμα προθέματα (prefixes) για διευκόλυνση και έπειτα λαμβάνουμε όλες τις χωρικές σχέσεις (Spatial Relations), από την οντολογία προς επεξεργασία, σύμφωνα με τον παρακάτω ψευδοκώδικα:

```

Ιδιότητα ανατολικά = λήψηΙδιότητας(URI_ιδιότητας);
Ισχυρισμός χωρικόςΙσχυρισμός = ισχυρισμός(ανατολικά -> ΣχέσηΚατεύθυνσης);
προσθήκηΣτηνΟντολογία(χωρικόςΙσχυρισμός);
  
```

Μέσω της κλάσης OWLDataFactory η οποία παρέχει πρόσβαση στα δεδομένα της οντολογίας, ανακτούμε την ιδιότητα αντικειμένων (Object Property) με URI το οποίο επιθυμούμε. Έπειτα κάνουμε τον ισχυρισμό (Assertion) ότι ανήκει στην υπερκλάση ιδιοτήτων των Σχέσεων Κατεύθυνσης (Directional Relations) [3] και εισάγουμε τον ισχυρισμό στην οντολογία. Με το σώσιμο της οντολογίας στο τέλος της μεθόδου, θα αποθηκευτεί και ο εν λόγω ισχυρισμός. Με τον ίδιο τρόπο ανακτούμε όλες τις σχέσεις και κλάσεις που χρειαζόμαστε, όπως αναφέρθηκε και στο τρίτο κεφάλαιο, σχετικά με την οντολογία SOWL [1]. Ο τρόπος εισαγωγής των στιγμιότυπων (individuals) στις κλάσεις της οντολογίας είναι ο εξής:

- **Polygon** : Εδώ εισάγονται τα στιγμιότυπων (individuals) των πολυγώνων τα οποία περιγράφουν τη γεωμετρία των Χωρικών Αντικειμένων (Spatial Objects). Είναι υποκλάση της κλάσης Geometry.
- **LinearRing** : Εδώ εισάγονται οι συντεταγμένες του κάθε πολυγώνου μέσω της ιδιότητας τύπου δεδομένων (Datatype Property) *posList*.
- **Location** : Εδώ εισάγεται η τοποθεσία για το κάθε Χωρικό Αντικείμενο. Η τοποθεσία συνδέεται με κάποιο στιγμιότυπο της κλάσης Geometry, το οποίο μπορεί να έχει συγκεκριμένη υπόσταση (να συνδέεται με κάποιο LinearRing) ή να είναι ποιοτικά ορισμένη η θέση του. Η σύνδεση γίνεται μέσω της ιδιότητας αντικειμένων (Object Property) *hasGeometry*.
- **SpatialObject** : Εδώ εισάγονται όλα τα Χωρικά Αντικείμενα (Spatial Objects). Καθένα από αυτά συνδέεται μέσω της ιδιότητας αντικειμένων (Object Property) *hasLocation*, με κάποιο στιγμιότυπο της κλάσης Location.
- **Instant (4D-Fluents [6])**: Εδώ εισάγονται όλες οι διαφορετικές Χρονικές Στιγμές (Time Instnants) που αποτελούν τα Χρονικά Διαστήματα (Time Intervals).
- **DateTimeInterval (4D-Fluents [6])**: Εδώ εισάγονται όλα τα διαφορετικά Χρονικά Διαστήματα (Intervals) των Χρονικών Κομματιών (Time Slices), καθ' ένα από τα οποία διαθέτει για αρχή και τέλος κάποια Χρονική Στιγμή (Instant).
- **TimeSlice (4D-Fluents [6])**: Στην κλάση αυτή γίνεται η εισαγωγή των Χρονικών Κομματιών (Time Slices) των αντικειμένων (individuals) καθεμιάς από τις κλάσεις Polygon, Location και SpatialObject. Κάθε ένα από τα Χρονικά Κομμάτια συνδέεται με κάποιο στιγμιότυπο των προαναφερθέντων κλάσεων μέσω της σχέσης *tsTimeSliceOf*. Η χρονική διάρκεια του κάθε κομματιού, χαρακτηρίζεται από τη σύνδεση αυτού με κάποιο από τα χρονικά διαστήματα (Intervals), μέσω της ιδιότητας αντικειμένων (Object Property) *tsTimeInterval*. Η οντολογία εφ' όσον υπάρχουν χρονικά κομμάτια αντικειμένων θεωρείται δυναμική. Στην αντίθετη περίπτωση, η οντολογία θεωρείται στατική.

Ο υπολογισμός των σχέσεων γίνεται σε ένα διπλό βρόχο, ο οποίος εκτελείται σε πολυωνυμικό χρόνο $O(\frac{n^2}{2})$. Αυτό επιτυγχάνεται υπολογίζοντας τις σχέσεις μεταξύ των αντικειμένων μονόπλευρα. Για παράδειγμα, μεταξύ δύο αντικειμένων A και B, υπολογίζεται μόνο η τριπλέτα A σχέση1 B. Εξοικονομούμε έτσι χρόνο, καθώς ο αιτιολογητής (Reasoner) που θα ενεργοποιηθεί έπειτα, θα συμπεράνει και τις αντίστροφες τριπλέτες (B σχέση2 A), χρησιμοποιώντας τους κανόνες SWRL. Κάθε χρονικό κομμάτι (Time Slice) εισάγεται στην οντολογία με το εξής όνομα:

(Όνομα Κλάσης) + (Αριθμός Αντικειμένου) + (TimeSlice) + Παρούσα Χρονική Στιγμή (π.χ.Polygon2Timeslice12)

Οι αναφορές των κομματιών και η εισαγωγή των αντίστοιχων ισχυρισμών (assertions) στην οντολογία, γίνονται μέσω του OWL API σύμφωνα με τον παρακάτω ψευδοκώδικα (παράδειγμα για χρονικά κομμάτια της κλάσης Location):

Στιγμιότυπο χρονικόΚομμάτιΤοποθεσίας = λήψηΑναφοράςΣτιγμιότυπου(URI_Στιγμιότυπου#"ΌνομαΣτιγμιότυπου");
 Ιδιότητα χρονικόΚομμάτιΤου = λήψηΙδιότητας(URI_ιδιότητας);


```
Ισχυρισμός χρονικόςΙσχυρισμός = ισχυρισμός(χρονικόΚομμάτιΤοποθεσίας -> χρονικόΚομμάτιΤου -> τοποθεσία);  
προσθήκηΣτηνΟντολογία(χρονικόςΙσχυρισμός);
```

Με αυτόν τον τρόπο δημιουργούνται όλες οι αναφορές (references) στην οντολογία για στιγμιότυπα (individuals), κλάσεις, υποκλάσεις, ιδιότητες αντικειμένων (Object Properties), ιδιότητες τύπων δεδομένων (Datatype Properties), κλάσεις και υποκλάσεις ιδιοτήτων. Η παρούσα χρονική στιγμή διατηρείται σε μία μεταβλητή με το όνομα *moves* (κινήσεις). Τα στιγμιότυπα της κλάσης *LinearRing* αποθηκεύονται ως εξής:

```
(LinearRing) + (αριθμός πολυγώνου) + _ + (Παρούσα Χρονική Στιγμή) (π.χ. LinearRing3_11 )
```

Κάθε φορά που καλείται η συνάρτηση η οποία αναλύεται, ορίζεται η χρονική στιγμή που λαμβάνεται σαν όρισμα, να είναι το τέλος των χρονικών διαστημάτων που δημιουργήθηκαν κατά την προηγούμενη κλήση της. Για τα χρονικά διαστήματα που προστίθενται κάθε φορά ορίζεται να έχουν άγνωστο τέλος (*unknownEnd*). Ως αρχή τους τίθεται η χρονική τιμή που λήφθηκε ως όρισμα. Ο παρακάτω ψευδοκώδικας περιγράφει τη διαδικασία:

```
Στιγμιότυπο προηγούμενοΧρονικόΔιάστημα = λήψηΑναφοράςΣτιγμιότυπου(URI_Στιγμιότυπου#"ΌνομαΣτιγμιότυπου");  
Ισχυρισμός χρονικόςΙσχυρισμός = ισχυρισμός(προηγούμενοΧρονικόΔιάστημα -> ΚλάσηΧρονικώνΔιαστημάτων);  
προσθήκηΣτηνΟντολογία(χρονικόςΙσχυρισμός);
```

```
ΓιαΚάθεΑξίωμαΤου προηγούμενοΧρονικόΔιάστημα{  
Αν(αξίωμαΠεριέχει -> "unknownEnd") τότε διάγραψεΑξίωμα;  
}
```

```
Ισχυρισμός χρονικόςΙσχυρισμός =  
ισχυρισμός(προηγούμενοΧρονικόΔιάστημα -> έχειΤέλος -> ΠαρούσαΧρονικήΣτιγμή);  
προσθήκηΣτηνΟντολογία(χρονικόςΙσχυρισμός);
```

Αφού δημιουργηθούν τα επιθυμητά χρονικά διαστήματα (Time Intervals), υπολογίζουμε τις σχέσεις μεταξύ των αντικειμένων. Για κάθε ένα ζευγάρι αντικειμένων που ελέγχουμε, καλούμε τις ανάλογες συναρτήσεις των κλάσεων *TopologicRelation* και *ConeBasedRelation* και ανάλογα με την επιστρεφόμενη τιμή, λαμβάνουμε την ιδιότητα που χρειαζόμαστε από την οντολογία, κάνουμε τον απαραίτητο ισχυρισμό (assertion), και εισάγουμε. Παράδειγμα:

```
.  
. .  
Αν(σχέση == ΤοπολογικέςΣχέσεις -> TPPi)  
Ισχυρισμός χωρικόςΙσχυρισμός = ισχυρισμός(ΧρονικόΚομμάτιΠολυγώνου1 -> TPPi -> ΧρονικόΚομμάτιΠολυγώνου2);  
. .  
προσθήκηΣτηνΟντολογία(χωρικόςΙσχυρισμός);
```

Έπειτα, σώζεται η οντολογία.

- **Καθαρισμός Οντολογίας (clearOntology())**: Η μέθοδος αυτή λαμβάνει αναφορές από την οντολογία για όλα τα στιγμιότυπα (individuals) των κλάσεων *Polygon*, *LinearRing*, *Location*, *SpatialObject*, *Instant*, *Interval* και *TimeSlice*. Έπειτα τα διαγράφει μέσω του εργαλείου `OWLEntityRemover()` της OWL API. Η μέθοδος δέχεται σαν όρισμα το όνομα αρχείου της οντολογίας της οποίας τα στιγμιότυπα θα διαγραφεί. Ένα παράδειγμα διαγραφής σε ψευδοκώδικα είναι το εξής:

```
ΓιαΚάθεΣτιγμιότυποΤου(ΚλάσηΤοποθεσιών) {  
    διαγραφή(Στιγμιότυπο)  
}
```

Εδώ λαμβάνουμε σε κάθε επανάληψη ένα στιγμιότυπο (individual) από την κλάση τοποθεσιών (Location) και το υποχρεώνουμε να δεχτεί τον αφαιρετή οντοτήτων (Entity Remover), ο οποίος το αφαιρεί από την οντολογία.

- **Φόρτωση Οντολογίας (loadOntologyShapes())**: Η μέθοδος πραγματοποιεί φόρτωση της οντολογίας η οποία της δίνεται σαν όρισμα. Επιστρέφει ένα διάνυσμα από στιγμιότυπα της κλάσης Shape, τα οποία είναι η τελευταία χρονικά έκδοση των εισαχθέντων στην οντολογία Χωρικών Αντικειμένων (SpatialObjects).

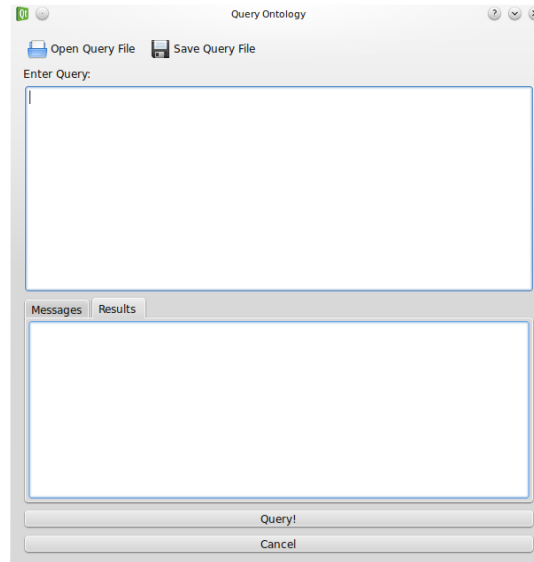
Κλάση QueryDialog

Η κλάση Query Dialog είναι υπεύθυνη για τη δημιουργία ενός παράθυρου διαλόγου με τα απαραίτητα συστατικά για επεξεργασία/υποβολή ερωτήσεων με τη Γλώσσα Ερωτήσεων SOWL [1]. Τα αποτελέσματα τυπώνονται στον κατάλληλο πίνακα στο κάτω μέρος του παραθύρου, στη μία εκ των δύο καρτελών (tabs) που διαθέτει. Στη δεύτερη καρτέλα τυπώνονται μηνύματα τα οποία επιστρέφει η Γλώσσα Ερωτήσεων SOWL. Παράλληλα διαθέτει και ένα απλό μηχανισμό αποθήκευσης και φόρτωσης αρχείων ερωτήσεων με την κατάληξη ".sowl" ώστε να μπορεί ο χρήστης να εκτελεί τις ερωτήσεις που τον ενδιαφέρουν χωρίς να πρέπει να τις πληκτρολογεί κάθε φορά. Το Σχήμα 3.7 απεικονίζει το παράθυρο το οποίο δημιουργεί η κλάση QueryDialog. Διακρίνονται και τα κουμπιά φόρτωσης και αποθήκευσης ερωτήσεων.

Κλάση ReadBlobsThreaded

Η κλάση αυτή (υλοποιείται στο αρχείο MainApplication) υλοποιεί ένα νήμα και είναι υπεύθυνη για την ανάγνωση ενός δοθέντος σαν όρισμα αρχείου το οποίο έχει δημιουργηθεί προηγουμένως από την ανάλυση ενός video. Το αρχείο αυτό διαβάζεται ταυτόχρονα με την αναπαραγωγή του video και περιέχει δεδομένα για το κάθε χωρικό αντικείμενο το οποίο έχει εντοπιστεί σε αυτό. Δημιουργείται κατά την εκτέλεση του αλγορίθμου εντοπισμού κίνησης σε video που αναλύθηκε στο Κεφάλαιο 3. Το αρχείο αυτό αποτελείται από γραμμές όπως η παρακάτω:

```
Frame #1005 - Blob #0: left=107.0, right=201.0, up= 42.0, bottom=195.0
```



Σχήμα 3.7: Το παράθυρο που δημιουργεί η κλάση QueryDialog

Ο πρώτος αριθμός περιγράφει τον αριθμό του πλαισίου (frame) του Video στο οποίο βρίσκεται το Χωρικό Αντικείμενο (Spatial Object) το οποίο περιγράφεται από το δεύτερο αριθμό (Blob #X). Οι επόμενοι τέσσερις αριθμοί περιγράφουν τη διαγώνιο του περιβάλλοντος παραλληλογράμμου (Minimum Bounding Rectangle) του αντικειμένου ως εξής: (left, bottom) = συντεταγμένες σε pixels του κάτω αριστερά σημείου της διαγωνίου και (right, up) = συντεταγμένες του πάνω δεξιά σημείου της διαγωνίου.

Αφού διαβαστούν οι συντεταγμένες της διαγωνίου, δημιουργείται ένα νέο στιγμιότυπο της κλάσης Shape (παρέχει κατασκευάστρια μέθοδο για δημιουργία παραλληλογράμμων, με την παροχή μόνο των σημείων της διαγωνίου τους), και προστίθεται σε ένα διάνυσμα τύπου δεδομένων Shape. Το διάνυσμα αυτό έπειτα παρέχεται στην μέθοδο calculateRelationsAndInsert() της κλάσης OntologyActionsOWL, για τον υπολογισμό των σχέσεων μεταξύ των σχημάτων και εισαγωγή στην οντολογία. Σύμφωνα με όσα προαναφέρθηκαν για την χρονική στιγμή που εισάγεται στην οντολογία, για λόγους διευκόλυνσης στις ερωτήσεις, η ημερομηνία του κάθε χρονικού σημείου (Instant) που δημιουργείται τίθεται να είναι η 01-01-2011, ενώ το ρολόι μηδενίζεται στην ώρα 00:00:00.

Κλάση RunVideoThreaded

Στην κλάση αυτή υλοποιείται ο κώδικας για την αναπαραγωγή του επιλεγμένου video με τη χρήση του συστήματος αναπαραγωγής πολυμέσων του Qt Jambi Framework. Η κλάση αυτή υλοποιείται στο αρχείο MainApplication.

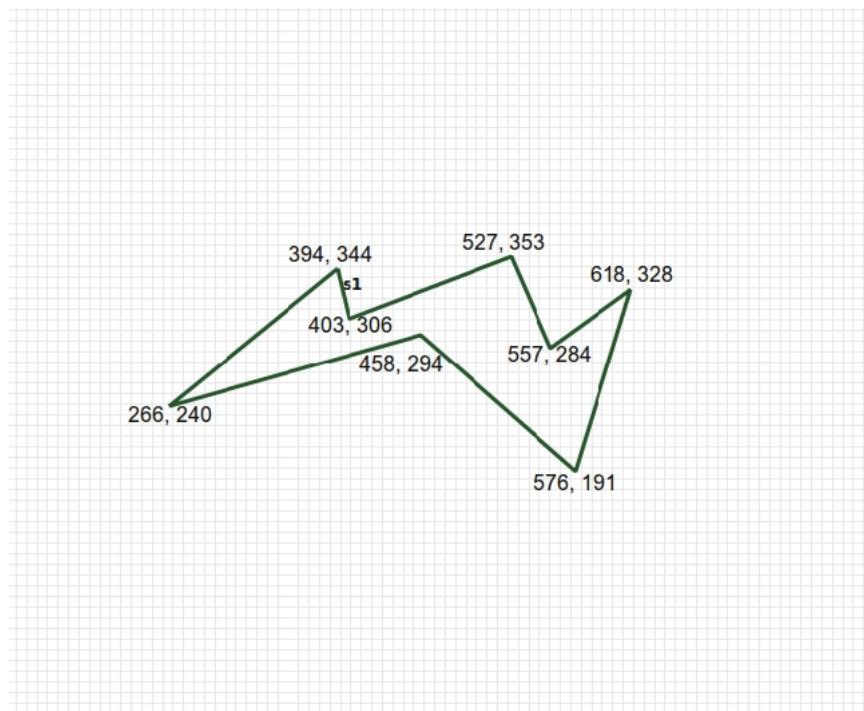
Κλάση ReadShape

Η κλάση ReadShape χρησιμοποιείται για τη δεύτερη λειτουργικότητα (mode) της εφαρμογής κατά την οποία προβάλλονται πολύγωνα στην οθόνη, τα οποία, ο χρήστης μπορεί

μέσω ενός κουμπιού στη γραφική διεπαφή της εφαρμογής να μετακινήσει με τυχαίο τρόπο. Η λειτουργικότητα αυτή είναι δευτερεύουσα για την εφαρμογή και χρησιμοποιείται κυρίως για πειραματικές μετρήσεις και δοκιμές. Η κλάση παρέχει μία μέθοδο η οποία δέχεται σαν όρισμα ένα αρχείο κειμένου, το οποίο περιέχει μέσα του συντεταγμένες πολυγώνων στην εξής μορφή:

```
394 344
266 240
458 294
576 191
618 328
557 284
527 353
403 306
394 344
"0"
```

Κάθε ζευγάρι αριθμών αναπαριστά και ένα σημείο του πολυγώνου. Η πάνω αριστερή γωνία της οθόνης θεωρείται πως είναι το σημείο με συντεταγμένες pixel (0,0). Ο τελευταίος αριθμός μέσα σε εισαγωγικά παρέχει τον αύξοντα αριθμό του πολυγώνου μέσα στο αρχείο. Στο Σχήμα 3.8 απεικονίζεται το σχήμα που δημιουργείται με τις παραπάνω συντεταγμένες.



Σχήμα 3.8: Σχήμα που δημιουργείται από ανάγνωση αρχείου με συντεταγμένες

Κλάση MainApplication

Η κλάση αυτή αποτελεί τον βασικό κορμό πάνω στον οποίο στηρίζεται η λειτουργία της εφαρμογής. Εδώ φορτώνεται το γραφικό περιβάλλον της, το οποίο χειρίζεται σημαντικές λειτουργίες, όπως η αναπαραγωγή του video, η σύνταξη ερωτήσεων και η τύπωση των αποτελεσμάτων τους σε πίνακα, η φόρτωση/καθαρισμός/σώσιμο οντολογιών και άλλες εσωτερικές λειτουργίες με τις οποίες ο χρήστης δεν έρχεται άμεσα σε επαφή.

Η εφαρμογή παρέχει δύο λειτουργικότητες (modes). Η κύρια λειτουργικότητά της, διαβάζει αρχεία video που έχουν υποστεί ανάλυση και των οποίων τα κινούμενα αντικείμενα προσθέτει σε μία οντολογία. Η δεύτερη λειτουργικότητα είναι χρήσιμη κυρίως για πειραματικές μετρήσεις και παρέχει στο χρήστη τη δυνατότητα να μετακινεί και να εισάγει στην οντολογία πολύγωνα τα οποία διαβάζονται από ένα αρχείο. Η MainApplication κληρονομεί την κλάση QMainWindow του Qt Jambi Framework, η οποία παρέχει μεθόδους για δημιουργία ενός κυρίως παραθύρου για εφαρμογές που διαθέτουν γραφική διεπαφή με το χρήστη. Στη συνέχεια προστίθενται πάνω στο κυρίως παράθυρο, όλα τα γραφικά στοιχεία που απαιτούνται. Το Qt Jambi Framework διαθέτει ένα σύστημα γραφικών στοιχείων (widgets), τα οποία μπορούν να προστεθούν πάνω σε ένα κυρίως παράθυρο. Τέτοια γραφικά στοιχεία χειρίζονται και την αναπαραγωγή των video καθώς και την προβολή των σχημάτων.

Η κλάση MainApplication, εκτός από τη δημιουργία του γραφικού περιβάλλοντος, επικοινωνεί με τα υπόλοιπα κομμάτια της εφαρμογής ως εξής:

- **Φόρτωση Δεδομένων Οντολογίας (loadOntologyShapes()):** Η μέθοδος αυτή καλεί την αντίστοιχη μέθοδο της κλάσης OntologyActionsOWL, η οποία διαβάζει μέσω του OWL API τα δεδομένα των χωρικών αντικειμένων (Spatial Objects) που βρίσκονται μέσα στην οντολογία. Εάν η λειτουργικότητα σχημάτων είναι ενεργοποιημένη, τυπώνει την τελευταία, χρονικά, έκδοση του κάθε ενός στην οθόνη. Τα χωρικά αντικείμενα που διαβάζονται αποθηκεύονται σε ένα διάνυσμα. Η μέθοδος αυτή δίνει τιμή στην καθολική (global) μεταβλητή ontologyFileName, αφού ο χρήστης επιλέξει, μέσω ενός παραθύρου διαλόγου (dialog window), την οντολογία την οποία θέλει να φορτώσει. Μέχρι να αλλάξει η τιμή της μεταβλητής, η επιλεγμένη οντολογία θα θεωρείται φορτωμένη και ό,τι λειτουργία ενεργοποιείται θα την αφορά.
- **Σώσιμο Οντολογίας (saveOntology()):** Η μέθοδος αυτή καλείται από το γραφικό περιβάλλον της εφαρμογής με σκοπό να σώσει στο δίσκο την παρούσα φορτωμένη οντολογία με κάποιο επιθυμητό όνομα αρχείου (filename). Επικοινωνεί με την κλάση OntologyActionsOWL και της παρέχει το όνομα αρχείου της προς σώσιμο οντολογίας. Οι λειτουργίες πάνω στις οντολογίες γίνεται με τη βιβλιοθήκη OWL API για τη γλώσσα προγραμματισμού Java. Η βιβλιοθήκη παρέχει όλες τις κατάλληλες κλάσεις και μεθόδους για αποδοτική επεξεργασία των οντολογιών.
- **Διαγραφή Δεδομένων Οντολογίας (deleteOntologyData()):** Η μέθοδος αυτή διαγράφει όλα τα στιγμιότυπα (individuals) που έχουν εισαχθεί σε μία SOWL οντολογία [1] στις κλάσεις Polygon, LinearRing, Location, SpatialObject, Instant, Interval και TimeSlice. Χρησιμεύει πριν την εισαγωγή των δεδομένων ενός νέου Video σε μία SOWL οντολογία ώστε να αποφύγουμε ασυνέπειες (inconsistencies) στην οντολογία. Λειτουργεί επίσης καλώντας την αντίστοιχη μέθοδο της κλάσης OntologyActionsOWL.

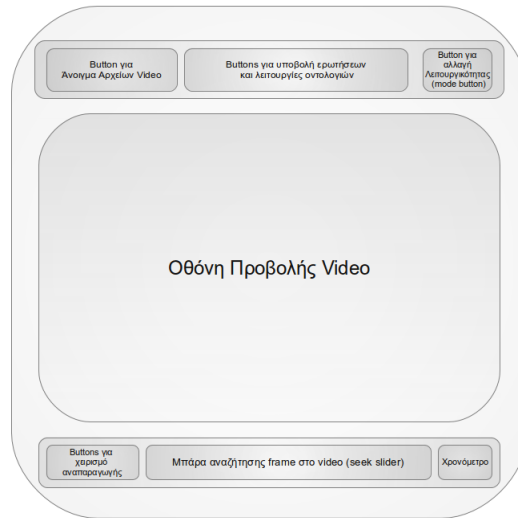
- **Προσθήκη στην Οντολογία (addToOntology()):** Η μέθοδος αυτή καλείται όταν η λειτουργικότητα (mode) τυχαία κινούμενων πολυγώνων είναι ενεργοποιημένη. Καλεί τη μέθοδο calculateRelationsAndInsert() της κλάσης OntologyActionsOWL(), η οποία αφού υπολογίσει τις Τοπολογικές Σχέσεις [4] και τις Σχέσεις Κατεύθυνσης [3] μεταξύ των σχημάτων που βρίσκονται σε ένα διάλυμα το οποίο λαμβάνει σαν όρισμα, τα εισάγει στην οντολογία μαζί με τις σχέσεις τους. Για λόγους διευκόλυνσης στις ερωτήσεις, η ημερομηνία του κάθε χρονικού σημείου (Instant) που δημιουργείται τίθεται να είναι η 01-01-2011, ενώ το ρολόι μηδενίζεται στην ώρα 00:00:00. Με κάθε νέα κίνηση των σχημάτων που εισάγονται αυξάνεται το ρολόι κατά ένα δευτερόλεπτο. Η ιδιότητα (Datatype Property) που έχει χρησιμοποιηθεί στην οντολογία SOWL για τον ορισμό χρονικών στιγμών δεν επιτρέπει την εισαγωγή δεκάτων δευτερολέπτου. Επίσης, οι κανόνες SWRL που περιέχει απαγορεύουν την εισαγωγή χρονικών διαστημάτων (Time Intervals) με αρχή και τέλος δύο χρονικές στιγμές που είναι ακριβώς ίδιες.
- **Άνοιγμα Αρχείου (open(), loadFile()):** Οι μέθοδοι αυτές είναι υπεύθυνες για το άνοιγμα και παρουσίαση αρχείων Video ή αρχείων πολυγώνων (ανάλογα με τη λειτουργικότητα). Στην πρώτη περίπτωση, αφού επιλεγεί η οντολογία στην οποία θα γίνει η εισαγωγή, αρχικά καθαρίζεται (deleteOntologyData()) και έπειτα υλοποιούνται δύο νήματα. Το ένα είναι υπεύθυνο για την αναπαραγωγή του Video, ενώ το δεύτερο δημιουργεί ένα στιγμιότυπο της κλάσης ReadBlobsThreaded, η οποία αναζητά ένα αρχείο κειμένου με όνομα αρχείου όμοιο με αυτό του Video που επιλέχθηκε. Το αρχείο αυτό περιέχει τα χαρακτηριστικά των αντικειμένων (Spatial Objects) τα οποία προήλθαν από την ανάλυση κίνησης του Video. Εφ' όσον το εντοπίσει διαβάζει τα σχήματα, τα αντικειμενοποιεί, τα προσθέτει σε ένα διάλυμα τύπου Shape και τα εισάγει στην επιλεγμένη οντολογία. Στην περίπτωση της λειτουργικότητας πολυγώνων (Shape Mode), γίνεται ανάγνωση του επιλεγμένου αρχείου με τις συντεταγμένες (pixels) του κάθε σχήματος, και με τον ίδιο τρόπο εισαγωγή στην οντολογία.
- **Υποβολή Ερώτησης (submitQuery()):** Η μέθοδος αυτή δημιουργεί στιγμιότυπο της κλάσης QueryDialog η οποία εμφανίζει ένα παράθυρο διαλόγου πάνω από το κυρίως παράθυρο της εφαρμογής. Στο παράθυρο διαλόγου αυτό σχηματίζεται το γραφικό περιβάλλον της φόρμας υποβολής ερωτήσεων. Αφού ο χρήστης επιλέξει να υποβάλλει ερώτηση, αρχικά του ζητείται να επιλέξει την οντολογία πάνω στην πληροφορία της οποίας θέλει να ενημερωθεί. Έπειτα γίνεται φόρτωση της οντολογίας μέσω της loadOntologyShapes(). Αφού ο χρήστης τυπώσει την ερώτηση και πατήσει το πλήκτρο "Query!", πραγματοποιείται σύνδεση με τη Γλώσσα Ερωτήσεων SOWL [1] και γίνεται η ερώτηση. Η Γλώσσα Ερωτήσεων SOWL επιστρέφει ένα σετ αποτελεσμάτων με τη μορφή λίστας τύπου QuerySolution, η οποία περιέχει αριθμημένα τα αποτελέσματα της ερώτησης. Τέλος, αυτά τυπώνονται στον πίνακα του παραθύρου ερωτήσεων, μαζί με κάποια μηνύματα μετάφρασης ερωτήσεων που επιστρέφει η γλώσσα. Παρακάτω φαίνεται η γραμμή κώδικα που υποβάλλει την ερώτηση στη Γλώσσα Ερωτήσεων SOWL:

```
List<QuerySolution> results = queryFrame.submitQuery(queryTextEdit.toPlainText());
```

Γραφικό Περιβάλλον (GUI), Δυνατότητες/Επικοινωνία Χρήστη

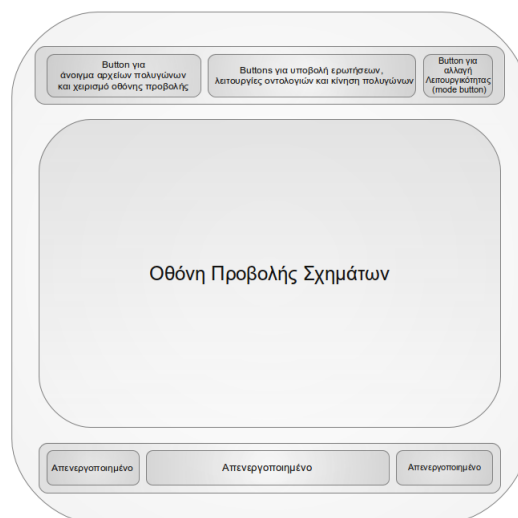
Το γραφικό περιβάλλον της εφαρμογής, σχεδιάστηκε με τρόπο ώστε ο χρήστης να

έχει εύκολη πρόσβαση στις λειτουργίες της. Η δημιουργία του γραφικού περιβάλλοντος έγινε χρησιμοποιώντας τη βιβλιοθήκη Qt Jambi, της εταιρίας Nokia η οποία προσφέρει μία σειρά από σύγχρονα εργαλεία για σχεδιασμό γραφικών. Παρακάτω παρουσιάζονται δύο σχήματα με τα περιεχόμενα του παραθύρου της εφαρμογής για κάθε λειτουργικότητά της.



Σχήμα 3.9: Η λειτουργικότητα Video Mode της εφαρμογής

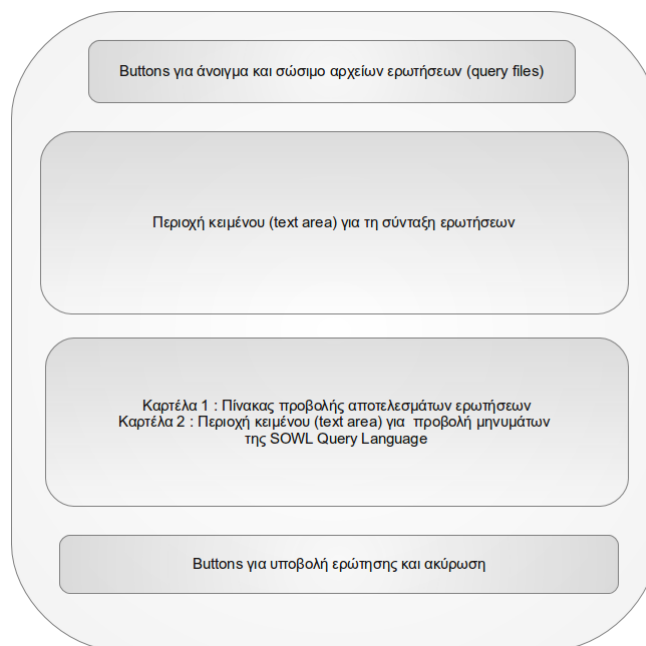
Στη βασική λειτουργικότητα της εφαρμογής, με όνομα Video Mode (Σχήμα 3.9), δίνονται τα απαραίτητα κουμπιά (buttons) τοποθετημένα πάνω σε μια γραμμή εργαλείων (toolbar) στην κορυφή του παραθύρου της εφαρμογής. Τα κουμπιά αυτά, χρησιμεύουν στην επιλογή αρχείων video, στην προβολή της φόρμας υποβολής ερωτήσεων, στον καθαρισμό της φορτωμένης οντολογίας και στην αλλαγή λειτουργικότητας. Στο κέντρο του κυρίως παραθύρου βρίσκεται η οθόνη αναπαραγωγής του επιλεγμένου video (παρέχεται από τη βιβλιοθήκη Qt Jambi). Στο κάτω μέρος του παραθύρου βρίσκονται κάποια εργαλεία διαχείρισης του αναπαραγόμενου video, όπως *Αναπαραγωγή (Play)*, *Παύση (Pause)* και *Σταμάτημα (Stop)*. Υπάρχει επίσης και μια μπάρα αναζήτησης frame στο video, καθώς και ένα χρονόμετρο (παρέχονται από τη βιβλιοθήκη Qt Jambi).



Σχήμα 3.10: Η λειτουργικότητα Shape Mode της εφαρμογής

Στο Σχήμα 3.10 βλέπουμε τη δευτερεύουσα λειτουργικότητα (mode) της εφαρμογής, η οποία δίνει τη δυνατότητα στο χρήστη να προβάλλει στην οθόνη, να μετακινήσει και να εισάγει στην οντολογία κάποια πολύγωνα τα οποία βρίσκονται σε αρχεία κειμένου. Τα επιπλέον κουμπιά στο κυρίως παράθυρο της εφαρμογής χρησιμεύουν στην προσθήκη επιπλέον πολυγώνων στην οθόνη, στον καθαρισμό της οθόνης και στη μετακίνηση και προσθήκη τους (και των μεταξύ τους χωρο-χρονικών σχέσεων) στην φορτωμένη SOWL οντολογία [1]. Μέσω ενός κουμπιού επαναφοράς, τα σχήματα επανέρχονται στις θέσεις που κατείχαν πριν κινηθούν. Στο κέντρο του παραθύρου, αυτήν τη φορά, βρίσκεται η οθόνη προβολής των σχημάτων (παρέχεται από τη βιβλιοθήκη Qt Jambi). Τα εργαλεία διαχείρισης του αναπαραγόμενου video είναι απενεργοποιημένα.

Στο σχήμα 3.11 παρουσιάζεται ο τρόπος με τον οποίο το παράθυρο υποβολής ερωτήσεων είναι στημένο. Στο πάνω μέρος έχουμε δύο κουμπιά για το άνοιγμα και την αποθήκευση αρχείων ερωτήσεων με την κατάληξη ".sowl", τα οποία ο χρήστης επιλέγει από κάποιο σημείο στο δίσκο. Πιο κάτω βρίσκεται η περιοχή σύνταξης των ερωτήσεων, όπου ο χρήστης μπορεί να εισάγει μια ερώτηση (query) που επιθυμεί να υποβάλλει σχετικά με την πληροφορία της φορτωμένης οντολογίας. Ακόμα πιο κάτω βρίσκονται δύο καρτέλες (tabs). Στην προεπιλεγμένη καρτέλα υπάρχει ένας πίνακας στον οποίο τυπώνονται τα αποτελέσματα που επιστρέφει η Γλώσσα Ερωτήσεων SOWL σχετικά με την ερώτηση που της υποβλήθηκε. Στη δεύτερη καρτέλα τυπώνονται μηνύματα που επιστρέφει η γλώσσα σχετικά με την ερώτηση. Τέλος, στο κάτω μέρος του παραθύρου, βρίσκονται δύο κουμπιά, ένα για την υποβολή της τυπωμένης ερώτησης και ένα για ακύρωση και κλεισιμο του παραθύρου.



Σχήμα 3.11: Η λειτουργικότητα Video Mode της εφαρμογής

Κεφάλαιο 4

Πειραματικές Μετρήσεις

Στο παρόν κεφάλαιο, πραγματοποιούνται συγκεκριμένες πειραματικές μετρήσεις σχετικά με την απόδοση του συστήματος. Όταν ο χρήστης πληκτρολογήσει κάποια ερώτηση και την υποβάλλει, καλείται η Γλώσσα Ερωτήσεων SOWL [1], η οποία επιστρέφει τις απαντήσεις. Σκοπός είναι να δείξουμε την εξάρτηση του χρόνου απόκρισης στις ερωτήσεις, από το περιεχόμενο ενός video. Η Γλώσσα ερωτήσεων SOWL, ουσιαστικά επεκτείνει τη Γλώσσα Ερωτήσεων SPARQL. Η επέκταση γίνεται όσον αφορά τους τελεστές της γλώσσας, οι οποίοι εμπλουτίζονται με χωρικούς και χρονικούς τελεστές, με σκοπό να γίνει δυνατή η σύνταξη χωρο-χρονικών ερωτήσεων. Έπειτα λαμβάνει χώρα μια μετάφραση του ερωτήματος σε σύνταξη γλώσσας SPARQL. Συνεπώς, το σύστημα της SPARQL είναι υπεύθυνο για τις απαντήσεις που θα λάβουμε. Οι ερωτήσεις γίνονται πάνω σε οντολογίες SOWL, οι οποίες εμπλουτίστηκαν σύμφωνα με όσα αναφέρθηκαν στο προηγούμενο κεφάλαιο. Αναφέρεται ότι η ανάπτυξη της εφαρμογής έγινε στο περιβάλλον Eclipse με μέγιστο heap size 512 MB, σε λειτουργικό σύστημα Arch Linux με παραθυρικό περιβάλλον KDE. Ο υπολογιστής που χρησιμοποιήθηκε διαθέτει διπύρνηο επεξεργαστή Intel Core Duo T9300 2.5GHz, 6MB cache με 4 GB DDR2 RAM.

4.1 Εκτίμηση Απόδοσης SOWL (SPARQL)

Με την εισαγωγή χωρικών αντικειμένων στην οντολογία, δημιουργούνται και εισάγονται παράλληλα πολλά στιγμιότυπα (individuals) διαφόρων κλάσεων. Η κλάση TimeSlice είναι αυτή που επιβαρύνεται περισσότερο, καθώς κάθε χρονικό κομμάτι ενός χωρικού αντικειμένου περιγράφεται από διάφορα άλλα χρονικά κομμάτια (TimeSlices). Η SPARQL προκειμένου να εξάγει τα αποτελέσματα, κάνει εξαντλητικό έλεγχο μεταξύ όλων των πιθανών τριπλετών. Συνεπώς, όσο περισσότερα στιγμιότυπα (individuals) εισάγονται στην οντολογία, τόσο αυξάνεται ο χρόνος που απαιτείται για να επιστραφεί απάντηση. Η προσθήκη των χωρο-χρονικών τελεστών που εφαρμόζει η Γλώσσα Ερωτήσεων SOWL [1], της δίνει μεγαλύτερη εκφραστικότητα σε ερωτήσεις που αφορούν εξέλιξη στο χώρο και στο χρόνο. Μία μικρή ερώτηση γραμμένη σε Γλώσσα Ερωτήσεων SOWL, η οποία περιέχει χωρο-χρονικούς τελεστές μεταφράζεται σε ερώτηση πολύ μεγαλύτερης έκτασης σε SPARQL. Αξίζει να σημειωθεί ότι για να συντάξει ο χρήστης μια τέτοια ερώτηση απευθείας σε SPARQL, πρέπει να έχει πλήρη γνώση του τρόπου με τον οποίο λειτουργεί το μοντέλο 4D-Fluents [6], καθώς και του τρόπου με τον οποίο δομείται η οντολογία. Μια απλή ερώτηση σε μορφή Γλώσσας Ερωτήσεων SOWL είναι η εξής:

```
select distinct ?x ?y where{      ?x spatial:Nof ?y AT("2011-01-01T00:00:19"). }
```

Η ερώτηση ζητά το όποιο χωρικό αντικείμενο είναι Βόρεια κάποιου άλλου κατά τη χρονική στιγμή 00:00:19. Η μετάφραση της παραπάνω ερώτησης σε ερώτηση SPARQL είναι η εξής:

```
TRANSLATED QUERY : select distinct ?x ?y where { {
?_timeSlice_0 ex1:tsTimeSliceOf ?x.
?_timeSlice_0 ex1:tsTimeInterval ?_interval_0.
?_timeSlice_0 spatial:locatedAt ?_timeSlice_1 .
?_timeSlice_1 ex1:tsTimeSliceOf ?_location_0.
?_timeSlice_1 ex1:tsTimeInterval ?_interval_0
} union {
?x spatial:locatedAt ?_location_0.
optional { ?_temporalVar rdf:type ex1:TimeSlice. ?x rdf:type ex1:TimeSlice }
filter( !bound(?_temporalVar) )} . {
?_timeSlice_2 ex1:tsTimeSliceOf ?_location_0.
?_timeSlice_2 ex1:tsTimeInterval ?_interval_1.
?_timeSlice_2 spatial:hasGeometry ?_timeSlice_3 .
?_timeSlice_3 ex1:tsTimeSliceOf ?_geometry_0.
?_timeSlice_3 ex1:tsTimeInterval ?_interval_1
} union {
?_location_0 spatial:hasGeometry ?_geometry_0.
optional { ?_temporalVar rdf:type ex1:TimeSlice. ?_location_0 rdf:type ex1:TimeSlice }
filter( !bound(?_temporalVar) )} . {
?_timeSlice_4 ex1:tsTimeSliceOf ?y.
?_timeSlice_4 ex1:tsTimeInterval ?_interval_2.
?_timeSlice_4 spatial:locatedAt ?_timeSlice_5 .
?_timeSlice_5 ex1:tsTimeSliceOf ?_location_1.
?_timeSlice_5 ex1:tsTimeInterval ?_interval_2
} union {
?y spatial:locatedAt ?_location_1.
optional { ?_temporalVar rdf:type ex1:TimeSlice. ?y rdf:type ex1:TimeSlice }
filter( !bound(?_temporalVar) )} . {
?_timeSlice_6 ex1:tsTimeSliceOf ?_location_1.
?_timeSlice_6 ex1:tsTimeInterval ?_interval_3.
?_timeSlice_6 spatial:hasGeometry ?_timeSlice_7 .
?_timeSlice_7 ex1:tsTimeSliceOf ?_geometry_1.
?_timeSlice_7 ex1:tsTimeInterval ?_interval_3
} union {
?_location_1 spatial:hasGeometry ?_geometry_1.
optional { ?_temporalVar rdf:type ex1:TimeSlice. ?_location_1 rdf:type ex1:TimeSlice }
filter( !bound(?_temporalVar) )} . {
?_timeSlice_8 ex1:tsTimeSliceOf ?_geometry_0.
?_timeSlice_8 ex1:tsTimeInterval ?_interval_4.
?_timeSlice_8 spatial:Nof ?_timeSlice_9 .
?_timeSlice_9 ex1:tsTimeSliceOf ?_geometry_1.
?_timeSlice_9 ex1:tsTimeInterval ?_interval_4.
?_atTimeInstant_0 time:inXSDDateTime "2011-01-01T00:00:19"sd:dateTime.
```

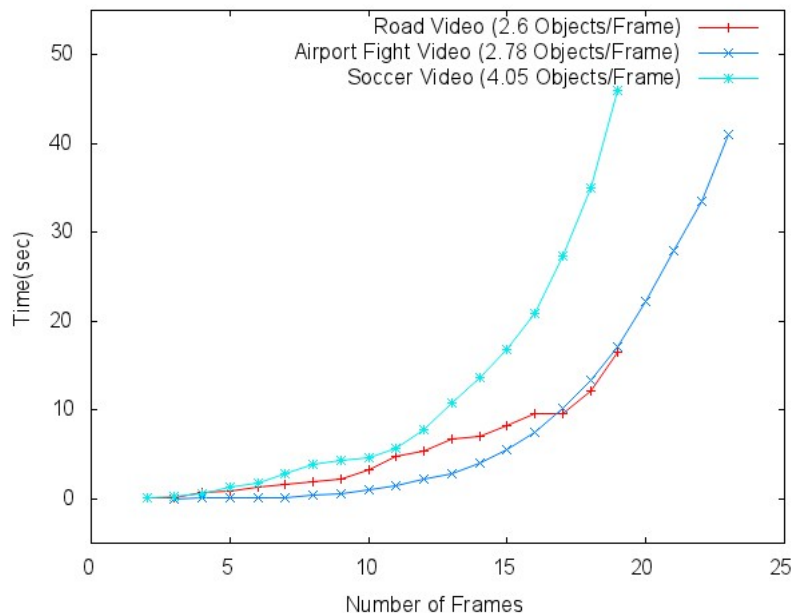
```

?_interval_4 time:hasBeginning ?_instant_1.
?_interval_4 time:hasEnd ?_instant_2.
{ ?_instant_1 time:before ?_atTimeInstant_0.
?_instant_2 time:after ?_atTimeInstant_0.}
UNION
{ ?_instant_1 ex2:equals ?_atTimeInstant_0.}
UNION
{?_instant_2 ex2:equals ?_atTimeInstant_0.}
} union {
?_geometry_0 spatial:Nof ?_geometry_1.
optional { ?_temporalVar rdf:type ex1:TimeSlice. ?_geometry_0 rdf:type ex1:TimeSlice }
filter( !bound(?_temporalVar) )
} . }

```

Είναι λοιπόν φανερά τα πλεονεκτήματα που εισάγει η Γλώσσα Ερωτήσεων SOWL [1] σχετικά με τη διευκόλυνση διατύπωσης ερωτήσεων. Ο μεγάλος αριθμός, όμως, των μεταβλητών που εισάγονται στην ερώτηση, σε συνδιασμό με το μέγεθος της οντολογίας που αυξάνεται με την πάροδο του χρόνου στο video, αναμένεται να έχει αρνητική επίδραση στο χρόνο απόκρισης στις ερωτήσεις.

4.2 Εκτίμηση Απόδοσης SOWL (Αντικείμενα/Πλαίσιο)



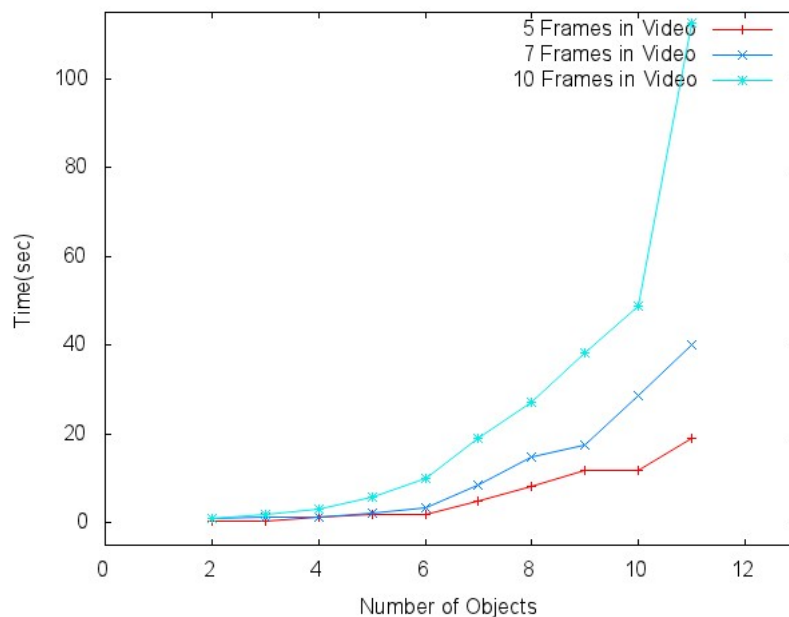
Σχήμα 4.1: Χρόνοι απάντησης για διάφορα video με διαφορετικό αριθμό αντικειμένων ανά πλαίσιο το καθένα

Ο αριθμός των στιγμιοτύπων και των σχέσεων που εισάγονται στην οντολογία, αυξάνεται συνεχώς με την πάροδο του χρόνου στο video. Το μέγεθος της αύξησης αυτής, όμως εξαρτάται άμεσα από το περιεχόμενο του video. Για παράδειγμα σε ένα video στο οποίο υπάρχει κίνηση 100 αντικειμένων, ο αριθμός των κινούμενων αντικειμένων και των σχέσεων που θα εντοπίζονται σε κάθε πλαίσιο θα είναι πολύ μεγάλος. Στο Σχήμα 4.1, παρουσιάζονται τα διαγράμματα χρόνου απάντησης για τρία videos με διαφορετικό μέσο αριθμό κινούμενων αντικειμένων ανά πλαίσιο, συναρτήσει του αριθμού πλαισίων του κάθε video. Η ερώτηση που υποβάλεται για τη λήψη των μετρήσεων είναι η ίδια με αυτήν της προηγούμενης ενότητας:

```
select distinct ?x ?y where{      ?x spatial:Nof ?y AT("2011-01-01T00:00:19"). }
```

Η αύξηση λοιπόν των κινούμενων αντικειμένων σε ένα video, αυξάνει δραστικά τον χρόνο απόκρισης στις ερωτήσεις. Οι χρόνοι που απαιτούνται για βίντεο άνω των 15 πλαισίων στο παραπάνω παράδειγμα, δεν είναι ανεκτά μικροί για καθημερινή χρήση.

4.3 Εκτίμηση Απόδοσης SOWL (Πλαίσια/Video)



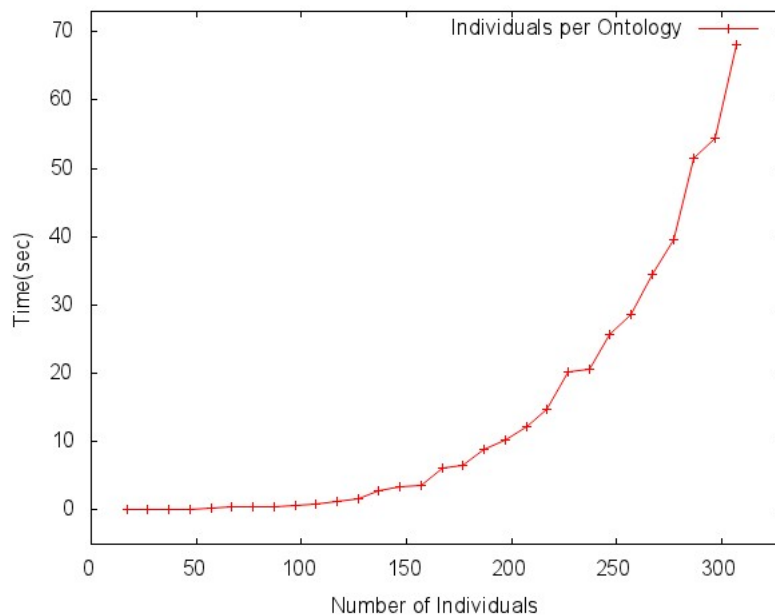
Σχήμα 4.2: Χρόνοι απάντησης για συγκεκριμένο αριθμό πλαισίων συναρτήσει του αριθμού των αντικειμένων

Όπως αναφέρθηκε, ο αριθμός των στιγμιοτύπων και των σχέσεων που εισάγονται στην οντολογία, αυξάνεται συνεχώς με την πάροδο του χρόνου στο video. Συνεπώς όσο μεγαλύτερο είναι το video, τόσο περισσότερη πληροφορία εισάγεται στην οντολογία. Αναμένεται, λοιπόν, μείωση της απόδοσης του συστήματος στις αποκρίσεις ερωτήσεων, όσο αυξάνεται το μέγεθος ενός video, ακόμα και εάν ο αριθμός κινούμενων αντικειμένων σε αυτό είναι σταθερός. Στο Σχήμα 4.2 παρουσιάζονται τα διαγράμματα χρόνου απάντησης για

τρία videos με διαφορετικό μέγεθος. Αυτήν τη φορά, για συγκεκριμένο αριθμό πλαισίων σε κάθε video, σχεδιάζεται ο απαιτούμενος χρόνος για λήψη αποτελεσμάτων συναρτήσει των χωρικών αντικειμένων τα οποία βρίσκονται σε κάθε πλαίσιο. Σημειώνεται ότι ο αριθμός των αντικειμένων σε κάθε πλαίσιο, κατά τη λήψη των μετρήσεων ήταν σταθερός. Για τον σκοπό αυτό χρησιμοποιήθηκε η δεύτερη λειτουργικότητα της εφαρμογής με τα τυχαίως κινούμενα πολύγωνα.

Η πολυπλοκότητα που εισάγει, στους υπολογισμούς της SPARQL, η προσθήκη ολοένα και περισσότερων στιγμιότυπων στην οντολογία είναι φανερή. Παρατηρούμε ότι οι περισσότερο ανεκτοί χρόνοι λήψης αποτελεσμάτων, είναι για video που περιέχουν το πολύ 6 κινούμενα αντικείμενα. Ο αριθμός των επιτρεπόμενων πλαισίων ανά video όμως παραμένει πολύ μικρός. Οι χρόνοι αναμονής είναι πολύ μεγάλοι για καθημερινή χρήση. Επισημαίνεται λοιπόν η ανάγκη μελλοντικής βελτίωσης του αλγορίθμου με τον οποίον η γλώσσα SPARQL αναζητά απαντήσεις.

4.4 Εκτίμηση Απόδοσης SOWL (Στιγμιότυπα/Οντολογία)



Σχήμα 4.3: Χρόνοι απάντησης σε συνάρτηση με τον αριθμό των στιγμιότυπων της οντολογίας

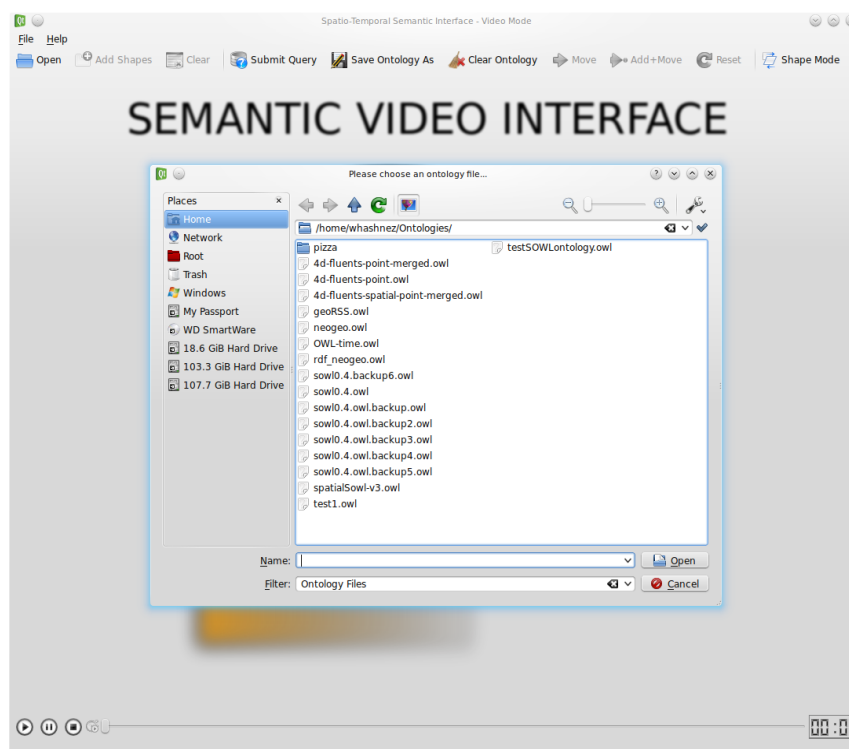
Προκειμένου γίνει πιο εμφανής η επιρροή του αριθμού των στιγμιότυπων (individuals) που εισάγονται στην οντολογία, παρατίθεται ακόμη ένα διάγραμμα (Σχήμα 4.3) χρόνου επιστροφής αποτελεσμάτων. Στο συγκεκριμένο διάγραμμα, για το ίδιο απλό ερώτημα που χρησιμοποιήθηκε και στα υπόλοιπα, αναγράφεται ο χρόνος που απαιτείται για τη λήψη αποτελεσμάτων συναρτήσει των στιγμιότυπων (individuals) που περιέχει η οντολογία. Είναι

εμφανής η επιρροή που ασκεί στην πολυπλοκότητα των υπολογισμών της SPARQL η προσθήκη ολοένα και περισσότερων στιγμιότυπων. Σημειώνεται ότι τα στιγμιότυπα αυτά μπορεί να ανήκουν σε οποιαδήποτε κλάση.

Τα διαγράμματα απόδοσης που παρουσιάστηκαν, φανερώνουν την επίπτωση της συνεχούς προσθήκης στιγμιότυπων στην οντολογία στο χρόνο που απαιτείται για να ληφθούν απαντήσεις. Από τη μορφή των γραφημάτων διαπιστώνεται πως η πολυπλοκότητα του αλγορίθμου της SPARQL προσεγγίζει την εκθετική ως προς τον αριθμό των στιγμιότυπων (individuals). Για πολύ μεγάλες οντολογίες είναι ανέφικτη η χρήση της SPARQL για χωροχρονικές ερωτήσεις, προς το παρόν.

4.5 Ερωτήσεις (Queries), Παραδείγματα

Στην παρούσα ενότητα θα γίνει μια επίδειξη του τρόπου λειτουργίας της εφαρμογής. Επιλέχθηκε ένα video το οποίο έχει ληφθεί από κάμερα εποπτείας σε αεροδρόμιο. Επιδεικνύεται, έτσι, η χρησιμότητα ύπαρξης τέτοιων εφαρμογών στο μέλλον, σε τομείς όπως τα συστήματα παρακολούθησης. Στο Σχήμα 4.4, φαίνεται το κυρίως παράθυρο (main window) της εφαρμογής. Είναι ενεργοποιημένο ένα παράθυρο διαλόγου από όπου γίνεται η επιλογή της οντολογίας στην οποία επιθυμεί ο χρήστης να εισάγει τα δεδομένα των αντικειμένων του video. Στο παράδειγμα επιλέγεται η οντολογία με όνομα *testSOWLontology.owl*.



Σχήμα 4.4: Επιλογή οντολογίας

Αφού γίνει επιλογή της οντολογίας και του video προς αναπαραγωγή, εκκινούν-

ται δύο νήματα. Το ένα είναι υπεύθυνο για την αναπαραγωγή του video, ενώ το δεύτερο για την ανάγνωση των χωρικών αντικειμένων από το αρχείο εισόδου. Παράλληλα με την αναπαραγωγή, υπολογίζονται και εισάγονται οι σχέσεις μεταξύ των χωρικών αντικειμένων (στην περίπτωση μας ανθρώπων) στην οντολογία που επιλέχθηκε. Στο Σχήμα 4.5 φαίνεται το κυρίως παράθυρο της εφαρμογής κατά τη διάρκεια αναπαραγωγής του video. Όταν ολοκληρωθεί η εισαγωγή των χωρικών δεδομένων που λήφθηκαν, στην οντολογία, ο χρήστης λαμβάνει μια ειδοποίηση στο κάτω αριστερό μέρος του παραθύρου της εφαρμογής. Αξίζει να σημειωθεί, πως ο τρόπος λειτουργίας του αλγορίθμου εντοπισμού κίνησης σε video που χρησιμοποιήθηκε, έχει ως αποτέλεσμα πολλές φορές τον λανθασμένο εντοπισμό αντικειμένων. Ο λόγος ύπαρξης αυτού του μη επιθυμητού αποτελέσματος είναι η κίνηση της κάμερας, ή η αλλαγή φωτισμού στο video (Μπορεί να προέρχονται και από το θόρυβο που υπάρχει στο video). Όπως αναφέρθηκε, ο αλγόριθμος εντοπισμού κίνησης σε video, λειτουργεί αφαιρώντας μία προς μία, τις τιμές των χρωμάτων των εικονοστοιχείων δύο συνεχόμενων πλαισίων (frames) του video. Παράλληλα αναζητώνται απότομες αλλαγές στην υφή της εικόνας ώστε να εντοπιστούν ακμές αντικειμένων. Η κίνηση της κάμερας ή η απότομη αλλαγή φωτισμού του video επηρεάζουν κυρίως την πρώτη διεργασία, δημιουργώντας τον εσφαλμένο εντοπισμό αντικειμένων. Με μια πιο προσεκτική ματιά στο Σχήμα 4.5 φαίνεται η ύπαρξη ενός τέτοιου εσφαλμένα εντοπισμένου αντικειμένου, με όνομα *s1* στο επάνω δεξιά σημείο της εικόνας (τα εσφαλμένα εντοπισμένα κινούμενα αντικείμενα ονομάζονται *artifacts*).

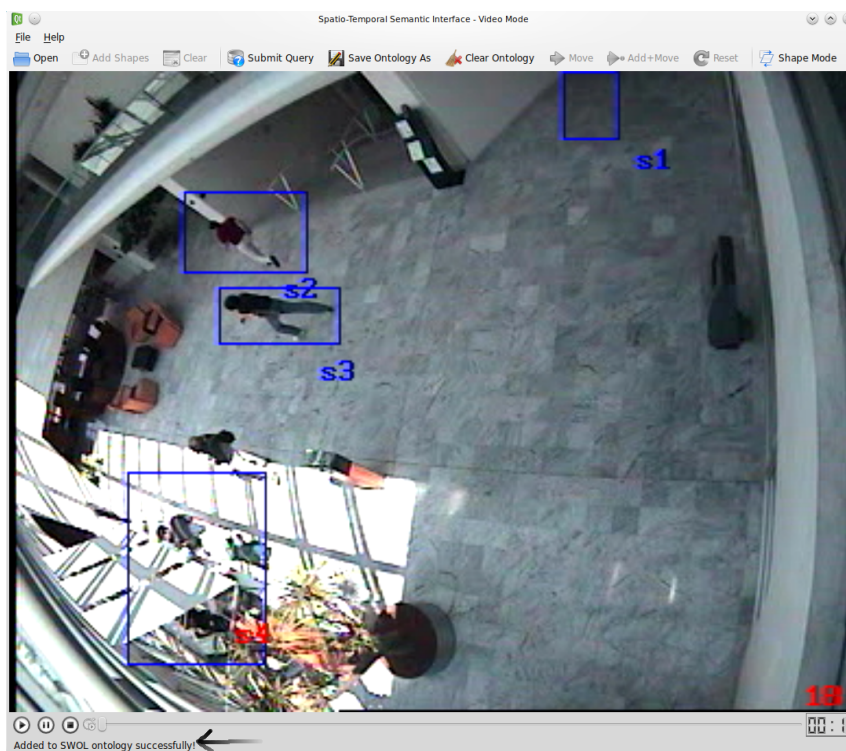


Figure 4.5: Αναπαραγωγή του video.

Αφού η εισαγωγή ολοκληρωθεί, ο χρήστης μπορεί να προχωρήσει στην υποβολή της ερωτήσεώς του. Με το πάτημα του κουμπιού *Submit Query* εμφανίζεται το παράθυρο διαλόγου επιλογής της οντολογίας στην οποία θα γίνει η ερώτηση. Εδώ επιλέχθηκε η οντολογία με όνομα *testSOWLontology.owl*. Αφού γίνει επιλογή της οντολογίας και πραγ-

ματοποιηθεί φόρτωσή της, εμφανίζεται η φόρμα υποβολής ερωτήσεων. Εκεί εισάγεται η ερώτηση σύμφωνα με τον τρόπο σύνταξης της Γλώσσας Ερωτήσεων SOWL [1]. Στο Σχήμα 4.6 φαίνεται η φόρμα υποβολής ερωτήσεων με μια ερώτηση τυπωμένη. Η ερώτηση που τίθεται στο συγκεκριμένο παράδειγμα είναι η εξής:

```
select distinct ?x ?y
where{
  ?x spatial:Nof ?y AT("2011-01-01T00:00:18").
}
```

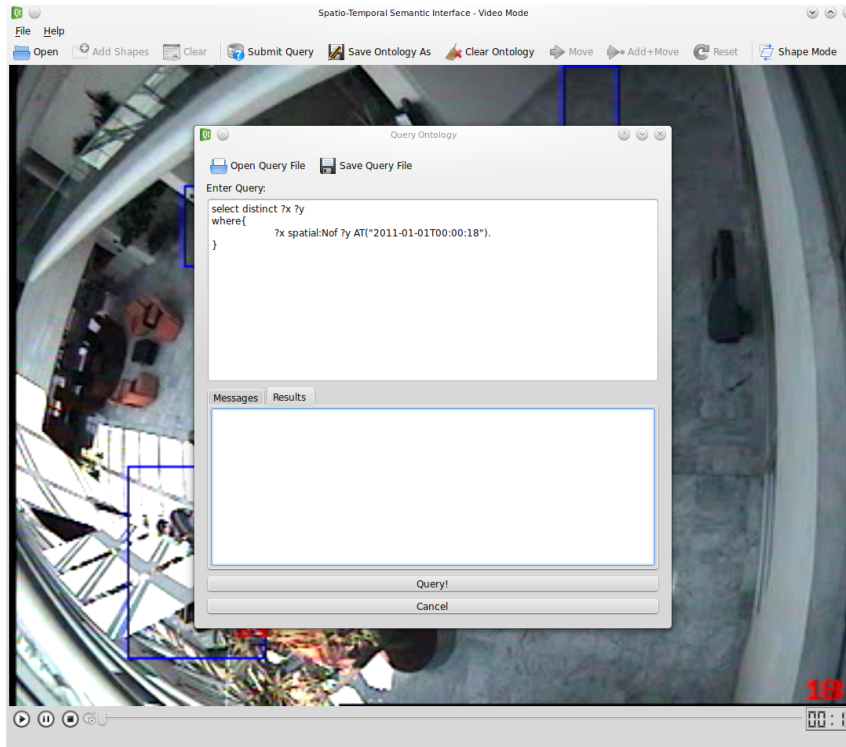
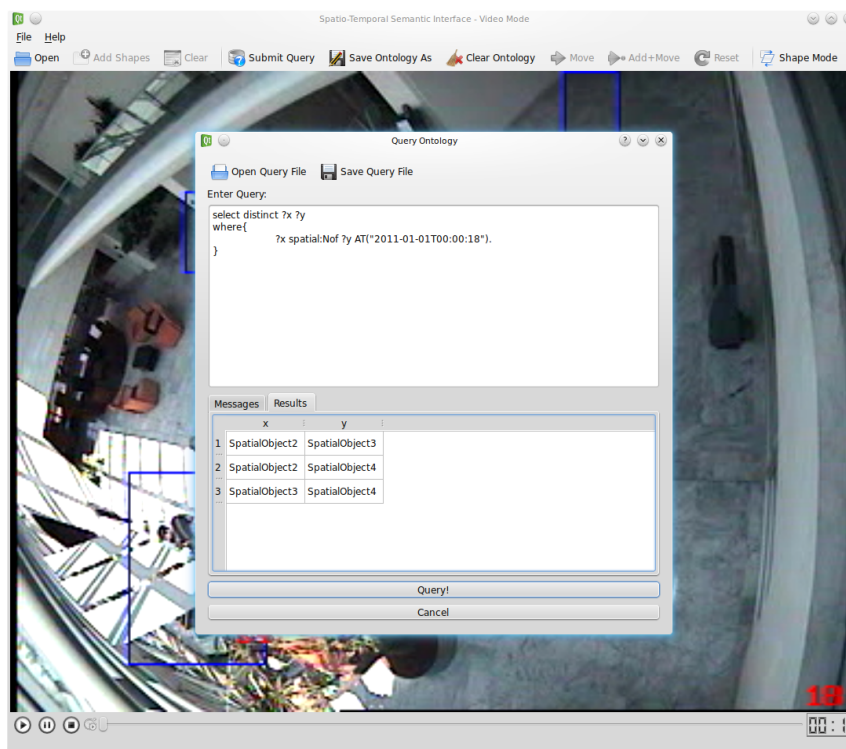


Figure 4.6: Υποβολή ερώτησης

Η παραπάνω ερώτηση αφορά όλα τα χωρικά αντικείμενα τα οποία στο πλαίσιο 18 του video (χρονική στιγμή 00:00:18), βρίσκονται Βόρεια (ισχύει η σχέση Nof) από κάποιο άλλο. Παρατηρώντας το Σχήμα 4.5, είναι φανερό ότι στο συγκεκριμένο πλαίσιο αυτό αληθεύει για τα εξής χωρικά αντικείμενα:

- ΧωρικόΑντικείμενο2 Nof ΧωρικόΑντικείμενο3
- ΧωρικόΑντικείμενο3 Nof ΧωρικόΑντικείμενο4
- ΧωρικόΑντικείμενο2 Nof ΧωρικόΑντικείμενο4

Αφού πατηθεί το κουμπί υποβολής της ερώτησης, ενεργοποιείται η Γλώσσα Ερωτήσεων SOWL [1], και μας επιστρέφει τα αποτελέσματα, τα οποία τυπώνονται στον κατάλληλο πίνακα. Στο Σχήμα 4.7 παρουσιάζεται η φόρμα υποβολής ερωτήσεων αφού ληφθεί η απάντηση. Σύμφωνα με τα παραπάνω, επαληθεύεται ότι η απάντηση που λήφθηκε είναι ορθή.



Σχήμα 4.7: Λήψη αποτελεσμάτων

Κεφάλαιο 5

Επίλογος (Συμπεράσματα και Επεκτάσεις)

Στη διπλωματική αυτή εργασία, έγινε προσπάθεια επίδειξης των δυνατοτήτων του Σημαιολογικού Ιστού σε εφαρμογές video στα οποία εφαρμόζεται ανάλυση κίνησης και εξαγωγή των κινούμενων αντικειμένων τους. Το επίπεδο όμως στο οποίο βρίσκονται σήμερα τα εργαλεία που χρησιμοποιήθηκαν είναι σε αρκετά πειραματικό στάδιο. Η δυσκολία της χρήσης τους από ανθρώπους μη εξειδικευμένους στο αντικείμενο, σε συνδυασμό με την χαμηλή τους απόδοση σε υπολογιστές καθημερινής χρήσης, είναι τα κύρια εμπόδια για την καθημερινή χρήση μιας τέτοιας εφαρμογής. Λόγω της στροφής του ενδιαφέροντος ολοένα και μεγαλύτερου ποσοστού της ερευνητικής κοινότητας παγκοσμίως, σε θέματα Σημαιολογικού Ιστού, αναμένεται ολοένα και μεγαλύτερη βελτίωση της απόδοσης και λειτουργικότητας των συγκεκριμένων εργαλείων. Έτσι, με τη βελτίωση της απόδοσης των τηλεπικοινωνιακών δικτύων, των αλγορίθμων ανάλυσης βίντεο και των γλωσσών ερωτήσεων σε οντολογίες, η εφαρμογή αυτή μπορεί να αποκτήσει πληθώρα χρήσιμων επεκτάσεων. Η εφαρμογή αναπτύχθηκε ώστε να λειτουργεί με οποιαδήποτε πολύγωνα, συνεπώς, θα μπορούσε να χρησιμοποιηθεί και παράλληλα με αλγορίθμους εντοπισμού κίνησης σε video που εξάγουν ολόκληρο το περίγραμμα των αντικειμένων που εντοπίζουν. Η βελτίωση της απόδοσης της Γλώσσας Ερωτήσεων SPARQL, θα επιτρέψει την χρήση της σε video με πραγματικό ρυθμό πλαισίων (frame rate).

Μια χρήσιμη, μελλοντική επέκταση της εφαρμογής δεδομένων των βελτιώσεων που αναφέρθηκαν, θα μπορούσε να χρησιμοποιηθεί σε συσκευές κινητών τηλεφώνων. Ο χρήστης, χρησιμοποιώντας την κάμερα της συσκευής του θα μπορεί να παρακολουθεί κινούμενα αντικείμενα τα οποία θα εντοπίζονται σε πραγματικό χρόνο. Με παράλληλη σύνδεση στο Σημαιολογικό Ιστό, θα μπορούν αυτόματα να αποθηκεύονται μαζί με το video, πολλές εξαγόμενες πληροφορίες σχετικά με τα εντοπισθέντα κινούμενα αντικείμενα. Δίνεται έτσι η δυνατότητα στους χρήστες του Σημαιολογικού Ιστού, να υποβάλουν ποικίλες ερωτήσεις σχετικά με τα αντικείμενα του video, λαμβάνοντας άμεσα χρήσιμες πληροφορίες σχετικά με τη θέση τους στο χώρο και στο χρόνο.

Βιβλιογραφία

- [1] Sotiris Batsakis, Euripides G.M. Petrakis, "*SOWL:Spatio-temporal Representation, Reasoning and Querying over the Semantic Web*". 6th International Conference on Semantic Systems (I-SEMANTICS' 2010), Graz, Austria, September 1-3, 2010 (presentation).
- [2] Γεωργιάκης Μιχάλης, Ζερβάκης Μιχάλης, "*Αναγνώριση κινούμενων αντικειμένων σε βίντεο και υλοποίηση σε (Digital Signal Processor) εξειδικευμένο επεξεργαστή*". Χανιά 2009.
- [3] D. Randell, Z. Cui, and A. Cohn, "*A Spatial Logic Based on Regions and Connection*". In Proc. 3rd Int. Conf. on Knowledge Representation and Reasoning, Morgan Kaufmann, San Mateo, pp. 165-176, 1992.
- [4] A. Frank, "*Qualitative Spatial Reasoning : Cardinal Directions as an Example*". In International Journal of Geographic Information Systems. 10(3), pp. 269-290, 1996.
- [5] J. F. Allen, "*Maintaining Knowledge About Temporal Intervals*". Communications of the ACM. 26:832-843, 1983.
- [6] C. Welty and R. Fikes, "*A Reusable Ontology for Fluents in OWL*". Frontiers in Artificial Intelligence and Applications, 150:226–236, 2006.
- [7] A. Artale, and E. Franconi, "*A Survey of Temporal Extensions of Description Logics*". Annals of Mathematics and Artificial Intelligence, 30(1-4), 2001.
- [8] C. Lutz, F. Wolter, and M. Zakharyashev, "*Temporal Description Logics: A Survey*". In Proc. TIME08, IEEE Press, 2008.
- [9] N. Noy and B. Rector, "*Defining N-ary Relations on the Semantic Web*". W3C Working Group Note 12, April 2006, <http://www.w3.org/TR/swbp-n-aryRelations/>
- [10] C. Gutierrez, C. Hurtado, and A. Vaisman, "*Introducing Time into RDF*". In IEEE Transactions on Knowledge and Data Engineering, 19(2), pp. 207-218, 2007.
- [11] M. Klein and D. Fensel, "*Ontology Versioning for the Semantic Web*". In International Semantic Web Working Symposium (SWWS'01), pages 75–92, California, USA, July–August 2001.
- [12] J. Tappolet, and A. Bernstein, "*Applied Temporal RDF: Efficient Temporal Querying of RDF Data with SPARQL*". In Proceedings of the European Semantic Web Conference, LNCS 5554, 308-322, 2009.
- [13] A. Wesley, "*Algorithms in C*". Princeton University.

- [14] Euripides G.M. Petrakis, *"Image Representation, Indexing and Retrieval based on Spatial Relationships and Properties of Objects"*. Dept. of Computer Science, University of Crete, 1993.

Παράρτημα Α΄

Δομή των κλάσεων

Στο παράρτημα αυτό παρουσιάζουμε τη δομή των κλάσεων του Eclipse Project της εφαρμογής:

- **determineDirectionalRelations**

- *ConeBasedRelation.java* : Υπολογίζει τις Σχέσεις Κατεύθυνσης (Directional Relations) [3] μεταξύ των κινούμενων αντικειμένων. Χρησιμοποιείται από την *OntologyActionsOWL*. Κληρονομεί την *SpatialRelation*.

- **determineTopologicRelations**

- *TopologicRelation.java* : Υπολογίζει τις RCC-8 Τοπολογικές Σχέσεις [4] μεταξύ των κινούμενων αντικειμένων. Χρησιμοποιείται από την *OntologyActionsOWL*. Κληρονομεί την *SpatialRelation*.

- **mainApplication**

- *MainApplication.java* : Δημιουργεί το κυρίως παράθυρο της εφαρμογής και χειρίζεται όλες τις λειτουργίες τις οποίες εκτελεί ο χρήστης μέσω του γραφικού περιβάλλοντος. Κληρονομεί την *QMainWindow* από το Qt Jambi Framework.
- *ReadShape.java* : Διαβάζει ένα αρχείο με συντεταγμένες πολυγώνων και τα παρέχει στην εφαρμογή σε μορφή διανύσματος. Χρησιμοποιείται από την *MainApplication*.

- **ontologyActions**

- *MergeOntologies.java* : Χρησιμοποιείται από τη *MainApplication* για να συγχωνεύσει (merge) δύο οντολογίες.
- *OntologyActionsOWL.java* : Εφαρμόζει όλες τις σχετικές με την οντολογία ενέργειες (Εισαγωγή, εκκαθάριση, αποθήκευση κτλ). Χρησιμοποιείται από τη *MainApplication*.

- **shapes**

- *Line.java* : Παρέχει τον τύπο δεδομένων για γραμμές.

- *Point.java* : Παρέχει τον τύπο δεδομένων για σημεία.
- *Rectangle.java* (*deprecated*): Παρέχει τον τύπο δεδομένων για παραλληλόγραμμα.
- *Shape.java* : Παρέχει τον τύπο δεδομένων για πολύγωνα.

- **spatialRelations**

- *SpatialRelation.java* : Απαριθμεί όλες τις χωρικές σχέσεις.

- [gr.tuc.intelligence.sowl.gui](#) ¹
- [gr.tuc.intelligence.sowl.interpreter](#)
- [gr.tuc.intelligence.sowl.ontologyParser](#)
- [gr.tuc.intelligence.sowl.queryParser](#)

¹Τα υπογραμμισμένα πακέτα είναι η υλοποίηση της SOWL Query Language