ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ένα σύστημα για την σταδιακή ενημέρωση του γράφου γνώσης YAGO2geo με γεωχωρική πληροφορία από το OpenStreetMap

Μιχάλης Σ. Μήτσιος

Επιβλέπων: **Μανώλης Κουμπαράκης,** Καθηγητής

ΑΘΗΝΑ

ΟΚΤΩΒΡΙΟΣ 2020

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ενημέρωση του Yago2geo
χρησιμοποιώντας
OSM Πληροφορία

Μιχάλης Σ. Μήτσιος

Α.Μ.: 1115201500096

ΕΠΙΒΛΕΠΟΝΤΕΣ: **Μανώλης Κουμπαράκης,** Καθηγητής

ΠΕΡΙΛΗΨΗ

Το YAGO αποτελεί μια από τις μεγαλύτερες βάσεις γνώσης σήμερα, που διαθέτει τα δεδομένα της ως ανοικτή πληροφορία. Μια επέκταση του YAGO2 είναι το YAGO2geo. Το YAGO2geo αποτελεί έναν γνωσιακό γράφο με ακριβή γεωχωρική πληροφορία όπου ένα μεγάλο μέρος της προέρχεται από το OpenStreetMap. Ο βασικός σκοπός της πτυχιακής εργασίας είναι να διατηρήσει την ακρίβεια της γεωχωρικής αυτής πληροφορίας κρατώντας την ενημερωμένη στο σήμερα. Για να επιτευχθεί το προηγούμενο, η εργασία χωρίζεται σε τρία μέρη. Το πρώτο μέρος, αποτελεί την ανάκτηση των πιο πρόσφατων δεδομένων που χρειάζονται. Το δεύτερο και κυριότερο μέρος εμπεριέχει την διασύνδεση των δεδομένων αυτών με εκείνων του YAGO2geo. Αυτό επιτυγχάνεται με την χρήση της PostgreSQL , μιας σχεσιακής βάσης δεδομένων ανοιχτού κώδικα, μέσω της οποίας θα γίνει η ενημέρωση των δεδομένων. Το τρίτο σκέλος αποτελεί το ανέβασμα της νέας πληροφορίας με την χρήση ενός SPARQL endpoint όπου και βρίσκεται ο Γνωσιακός Γράφος του YAGO2geo.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Σημασιολογικός Ιστός

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: YAGO, Yago2geo, OpenStreetMap, Γεωχωρική Πληροφορία, RDF, PostgreSQL

ABSTRACT

Yago is one of the largest open source Knowledge Bases nowadays. An extension of YAGO2 is YAGO2geo. YAGO2geo represents a knowledge graph with precise geospatial information. A large part of this data comes from OpenStreetMap. The main purpose of this thesis is to preserve the precision of the geospatial information by keeping it updated. To achieve this, the process can be divided into three parts. The first describes the acquisition of the most recent data we need. The second and the main part of the process includes the link and the update of YAGO2geo information using the previous data. In order to accomplish this, we are using the PostgreSQL database, a free and open-source relational database. The thrird and last part is to upload all the updated data on YAGO2geo Knowledge Graph by using a SPARQL endpoint.

SUBJECT AREA: Semantic Web

KEYWORDS: YAGO, Yago2geo, OpenStreetMap, Geospatial information, RDF, PostgreSQL

Στην οικογένειά μου και σε όσους με στήριξαν.

ΕΥΧΑΡΙΣΤΙΕΣ

Αρχικά θα ήθελα να ευχαριστήσω τον καθηγητή Μανώλη Κουμπαράκη που μου έδωσε την ευκαιρία να δουλέψω πάνω στο συγκεκριμένο θέμα. Παράλληλα θα ήθελα να ευχαριστήσω όλη την ομάδα του project για τον χρόνο που αφιέρωσαν, για τις παρατηρήσεις και τα σχόλιά τους πάνω στην πτυχιακή μου εργασία. Τις θερμές μου ευχαριστίες έχει ο Γιώργος Μανδηλαράς για τις συμβουλές του και την καθοδήγησή του καθ' όλη την διάρκεια. Τέλος, θα ήθελα να ευχαριστήσω όλους αυτούς που στάθηκαν δίπλα μου και συνέβαλαν στην εκπόνηση της πτυχιακής μου εργασίας.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

1. ΕΙΣΑΓΩΓΗ

1.1 Σημασιολογικός Ιστός (Semantic Web)

Ο Σημασιολογικός Ιστός[1] αποτελεί μια επέκταση του παγκόσμιου ιστού, που έχει ως στόχο να κάνει τα δεδομένα του διαδικτύου κατανοητά από τους υπολογιστές. Για να μπορέσει να κωδικοποιηθεί η σημασιολογία και η λογική του διαδικτύου σε μια μορφή που να είναι αναγνωρίσιμη στους υπολογιστές, χρησιμοποιούνται τεχνολογίες όπως Resource Description Framework (RDF) και Web Ontology Language (OWL). Οι τεχνολογίες αυτές παράγουν τα αντίστοιχα μετά-δεδομένα (metadata) για κάθε δημοσιευμένη πληροφορία, και αυτά με την σειρά τους μπορούν να «κατανοηθούν» από τους υπολογιστές.

Ο τρόπος με τον οποίο γίνεται η παραπάνω κωδικοποίηση είναι με την αναπαράσταση της πληροφορίας ως συνδεδεμένα δεδομένα(linked data). Σαν αποτέλεσμα τα απλά δεδομένα αποκτούν μια λογική υπόσταση η οποία αναγνωρίζεται και από την μηχανή. Αυτό συνεπάγεται πως μέσω του Semantic Web τα δεδομένα θα μπορούν πλέον να διαμοιράζονται ανεξαρτήτως ορίων (είτε τα όρια αυτά είναι κοινωνικά, επιχειρησιακά, είτε τα πλαίσια μιας εφαρμογής), ενώ η περισυλλογή και επεξεργασία τους γίνεται πολύ ευκολότερη.

Εν ολίγοις ο κύριος σκοπός του Semantic Web είναι να εξελίξει το τωρινό διαδίκτυο. Αυτό θα γίνει μέσω της ύπαρξης ενός συστήματος το οποίο θα επιτρέπει σε μηχανές να κατανοούν τα πολύπλοκα ανθρώπινα αιτήματα. Τέτοιου είδους κατανόηση απαιτεί, οι αντίστοιχες πληροφορίες, να έχουν και την ανάλογη σημασιολογική δομή. Με παρόμοιο τρόπο είχε οραματιστεί αρχικά ο δημιουργός του, Tim Berners-Lee, τον ρόλο του Σημασιολογικού Ιστού, όπου οι υπολογιστές θα είναι ικανοί να αναλύσουν όλα τα δεδομένα του Web.

Το Semantic Web στηρίζεται σε κάποια standards set τα οποία παρέχονται από το World Wide Web Consortium (W3C)[1][19] και έχουν αναπτυχθεί από standards organization. Με απλά λόγια τα set αυτά ορίζονται από μορφές και τεχνολογίες που παρέχουν μια τυπική περιγραφή εννοιών, όρων και σχέσεων εντός ενός knowledge domain.

Μερικές από αυτές τις τεχνολογίες είναι:

- Resource Description Framework (RDF), μια γενική μέθοδο περιγραφής πληροφοριών
- SPARQL Protocol and RDF Query Language (SPARQL)[20], μια RDF γλώσσα ερωτημάτων
- N-triples , μια μορφή αποθήκευσης και μετάδοσης πληροφορίας
- Turtle, μια λιτή RDF τριπλή γλώσσα πληροφοριών

[1] https://www.w3.org/

1.2 Resource Description Framework (RDF)

Το RDF [9] αποτελεί μέλος της οικογένειας των World Wide Web Consortium (W3C) και σχεδιάστηκε ως ένα μοντέλο οργάνωσης των μετα-δεδομένων (metadata). Πλέον όμως μπορεί να χρησιμοποιηθεί και ως μια γενική μέθοδος για εννοιολογική περιγραφή ή για μοντελοποίηση πληροφορίας που εφαρμόζεται στο διαδίκτυο χρησιμοποιώντας μια ποικιλία από συντακτικά σύμβολα και σειριοποίηση δεδομένων.

Η λογική πίσω από το RDF μοντέλο είναι παρόμοια με την προσέγγιση των κλασσικών εννοιολογικών μοντέλων (όπως οι σχέσεις μεταξύ οντοτήτων και τα διαγράμματα κλάσεων). Πιο συγκεκριμένα περιγράφει την δημιουργία μιας δήλωσης(statement) για την εκάστοτε πληροφορία. Η δήλωση αυτή εκφράζεται μέσω ενός συγκεκριμένου σχήματος **subject-predicate-object** γνωστό και ως triples. Το subject δηλώνει την πηγή/πληροφορία, ενώ το predicate δηλώνει τα χαρακτηριστικά ή τις πτυχές του subject. Εκφράζει, ουσιαστικά, την σχέση μεταξύ του subject και του object. Για παράδειγμα εάν έχουμε την εξής δήλωση:

Lake has geometry POLYGON

Τότε subject είναι το “Lake”, predicate το “has geometry” και object το POLYGON. Εν κατακλείδι το RDF μοντέλο αποτελεί ένα σύστημα το οποίο συσχετίζει διαφόρων ειδών οντότητες. Γνωρίζοντας αυτό οποιαδήποτε πληροφορία, σχέση ή γεγονός μεταξύ οντοτήτων μπορεί να αποτυπωθεί με αυτόν τον τρόπο. Η αναπαράσταση της πληροφορίας σε ένα RDF statement περιγράφεται είτε από ένα URI(uniform resource identifier),το οποίο δε σημαίνει απαραίτητα ότι αναφέρεται σε μια Internet-based πληροφορία, είτε από ένα blank node,το οποίο είναι ένας κόμβος στο RDF Γράφο που δεν του αντιστοιχεί κάποιο URI ή τιμή(literal).

Αυτή η τεχνολογία με την οποία μπορεί να αναπαρασταθεί η πληροφορία αποτελεί και την καρδιά του Semantic Web.

Σύμφωνα με τα παραπάνω μπορούμε να καταλήξουμε σε κάποιες αλήθειες σχετικά με το τι μας προσφέρει το RDF Format.

Open and Expressive

Το RDF, εξαιτίας της απλότητάς του, έχει σειριοποιηθεί σε διάφορες σειριακές μορφές(Serialization formats) όπως είναι:

- Turtle: μια συμπαγής, φιλική προς τον άνθρωπο μορφή.
- N-Triples: πολύ απλό, easy-to-parse, line-based μορφή.
- N-Quads: superset από N-Triples για σειριοποίηση πολλαπλών RDF Γράφων.
- JSON-LD: σειριοποίηση βασισμένη σε JSON.
- N3: Παρόμοιο με το Turtle, με κάποια έξτρα χαρακτηριστικά.
- RDF/XML: XML-based syntax, the first standard format for serializing RDF.
- RDF/JSON, ένας εναλλακτικός τρόπος σύνταξης RDF triples.

Data Interoperability

Το RDF Format χαρακτηρίζεται από την διαλειτουργικότητά του, καθώς μπορεί να αντιληφθεί και να εκφράσει τα μετα-δεδομένα, σε μη-δομημένη(unstructured), ημι-δομημένη(semi-structured) και δομημένη(structured) πληροφορία. Αφού δημιουργηθεί μια RDF αναπαράσταση είναι εύκολο να ενσωματωθούν αργότερα καινούρια datasets και νέα χαρακτηριστικά σε αυτή. Επίσης είναι εύκολο να συγκεντρωθούν δεδομένα από διαφορετικές πηγές σαν να προέρχονται από μία. Αυτό μας επιτρέπει την εννοιολογική σύνθεση δεδομένων που προέρχονται από διαφορετικές εφαρμογές ανεξάρτητα από την

μορφή σειριοποίησης(turtle,n-triples). Τέλος οι απλές δομές RDF καθιστούν ικανή την ύπαρξη συνωνύμων μεταξύ διαφορετικών δομών, έτσι ώστε να ομαδοποιούνται οι ίδιοι τύποι και να αντιμετωπίζονται αντιστοίχως.

Schema Unbound

Ένα ακόμα πλεονέκτημα του RDF είναι πως δεν περιορίζεται από κάποιο συγκεκριμένο σχήμα-δομή(schema) όπως αυτήν που συναντάμε στις βάσεις δεδομένων, όπου όλα τα tables έχουν προκαθορισμένη μορφή και η πληροφορία εκφράζεται με συγκεκριμένο τρόπο(δύσκολη η αλλαγή σε διαφορετική δομή). Το RDF έχει την δυνατότητα να αναπαραστήσει στιγμιότυπα(instances) των δεδομένων μαζί με την δομή που θα έχουν, δηλαδή το σχήμα που τα χαρακτηρίζει καλύτερα -είτε αυτό είναι απλά records με τα χαρακτηριστικά τους είτε ολοκληρωμένα λεξιλόγια και οντολογίες-. Εξαιτίας αυτού η πληροφορία εκφράζεται με τον καλύτερο δυνατό τρόπο είτε αυτή αποτελείται από απλά δεδομένα, ολόκληρους τομείς ή και ολόκληρο τον κόσμο.

Increment, Evolve, Extend, Adapt

Γνωρίζοντας τα προηγούμενα μπορούμε να συμπεράνουμε πως το RDF διακρίνεται από μια ρευστότητα, αφού μπορεί να υποβληθεί σε επεξεργασία ακόμη και με την απουσία πληροφοριών. Με λίγα λόγια τα δεδομένα ή το σχήμα μπορούν να χρησιμοποιηθούν και έπειτα να επεκταθούν σταδιακά ή μερικώς. Η μη ολοκληρωμένη αναπαράσταση αντιμετωπίζεται ως μια ολόκληρη, και η όλη δομή μπορεί να επεκταθεί και να εξελιχθεί καθώς νέες δομές αναπτύσσονται και ενσωματώνονται. Αυτό που μας προσφέρει το RDF Format ουσιαστικά είναι πως μας παρέχει πληροφορία που αναπαρίσταται ως ένα σχήμα το οποίο μπορεί να εξελιχθεί και να προσαρμοστεί σύμφωνα το τι δεδομένα και τι δομές υπάρχουν.

Graph Representation, Open World Applications and Semantic Web

Όπως έγινε αναφορά παραπάνω τα RDF triples όταν συνδυάζονται δημιουργούν έναν Γράφο(Graph). Οι Γράφοι επειδή είναι αρθρωτοί μπορούν εύκολα να συνδυαστούν ο ένας με τον άλλον αλλά και να αντιμετωπιστεί ο καθένας ατομικά, ως μια ολότητα. Από προγραμματιστική οπτική αυτό μπορεί να σημαίνει την παράλληλη επεξεργασία πληροφοριών. Επίσης επειδή το μοντέλο αντιμετωπίζεται ως Γράφος αυτό έχει σαν αποτέλεσμα την χρήση ξεχωριστών τεχνικών για αναζήτηση, pattern matching αλλά και την οπτικοποίησή του. Λογικά από τα παραπάνω συμπεραίνουμε το δεύτερο σκέλος αυτού του πλεονεκτήματος, που είναι η δημιουργία ενός τεράστιου Γράφου που απαρτίζεται από μικρότερους. Κάθε εφαρμογή με την χρήση του RDF θα μπορεί να συνδέσει την πληροφορία της με αντίστοιχη πληροφορία που πηγάζει από αλλού. Δεδομένα, δομές και λεξιλόγια θα είναι διαθέσιμα για τον κόσμο και συνεργατικά θα αναπτύσσονται διαρκώς από διάφορες πηγές-εφαρμογές. Αυτό αποτελεί την κύρια ιδέα του Semantic Web, όπου το κλειδί για την υλοποίησή του είναι η χρήση του Resource Description Framework.

1.3 SPARQL

Η SPARQL (SPARQL Protocol and RDF Query Language) είναι μια RDF query Language (γλώσσα ερωτημάτων) για βάσεις δεδομένων ικανή να αναγνωρίσει και να επεξεργαστεί την RDF μορφή που συναντήσαμε προηγουμένως. Ουσιαστικά τα queries της SPARQL παρουσιάζουν μια αλληλουχία-συνδυασμό(graph patterns) από πρότυπα triples(subject-predicate-object) τον οποίο έπειτα συγκρίνουν με τα triples που βρίσκονται στην βάση και

επιλέγουν αυτά που ταιριάζουν με το συγκεκριμένο πρότυπο. Για την δημιουργία αυτού του προτύπου γίνεται η χρήση μεταβλητών αναλόγως την θέση των subject-predicate-object.

Για παράδειγμα εάν θέλουμε να βρούμε τα κατοικίδια του John από το παρακάτω Dataset[1]:

Πίνακας 1.1 Dataset Example

Subject	Predicate	Object
ttr:John	tto:pet	ttr:LunaCat
ttr:John	tto:pet	ttr:TomCat
ttr:William	tto:pet	ttr:RexDog

Το παρακάτω query θα κάνει την δουλεία.

```
PREFIX tto:<http://example.org/tuto/ontology#>
PREFIX ttr:<http://example.org/tuto/resource#>

select * where {
        ttr:John tto:pet ?pet.
}
```

1.1 Query John's Pets

Η SPARQL θα ελέγξει τα δεδομένα της που βρίσκονται αποθηκευμένα υπό την μορφή triples και θα δει ποιο ταιριάζει στο πρότυπο triple που της δόθηκε. Το query δηλώνει πως για τα subject και predicate θέλει τις συγκεκριμένες τιμές ttr:John και tto:pet αντίστοιχα ενώ με την χρήση της μεταβλητής **?pet** δίνει την ελευθερία για οποιαδήποτε τιμή στην θέση του object. Επομένως η τιμή που θα επιστραφεί είναι " ttr:LunaCat" και "ttr:TomCat". Η μορφή των queries μπορεί να γίνει πιο περίπλοκη πόσο μάλλον όταν έχουμε ένα πολύ σύνθετο Γράφο με πολλές διασυνδεδεμένες οντότητες.

1.4 Γνωσιακή Βάση (Knowledge Base)

Στον σημερινό κόσμο η ανάγκη για περισσότερη γνώση και πληροφορία όλο και αυξάνεται. Για την ικανοποίηση αυτής της ανάγκης έρχεται η εφαρμογή των Knowledge Bases, όπου είναι μια συλλογή από οντότητες που περιγράφουν γεγονότα και πληροφορίες για τον κόσμο, σε συνδυασμό με μια μηχανή διεπαφής (interface), που σκοπός της είναι η ερμηνεία αυτής. Αυτή η συλλογή γνώσης είναι προσπελάσιμη και επεξεργάσιμη από τους υπολογιστές. Τις περισσότερες φορές μπορούμε να μιλάμε για μια δομημένη συλλογή πληροφορίας στην οποία τα αντικείμενα έχουν δείκτες που δείχνουν σε άλλα αντικείμενα που και αυτά με την σειρά τους έχουν τους δικούς τους δείκτες.

Αυτός είναι και ο λόγος όπου η αναπαράσταση γεγονότων και γενικότερα γνώσης μπορεί να περιγραφή καλύτερα με την χρήση μιας γνωσιακής βάσης, παρά με την χρήση ενός

[1] http://sparql-playground.sib.swiss/

Database. Για παράδειγμα εάν έχουμε την γενικότερη γνώση πως «Όλοι οι άνθρωποι έχουν εγκέφαλο» τότε η αναπαράσταση της από μια βάση δεδομένων θα έπρεπε να συμπεριλαμβάνει για κάθε άνθρωπο μια στήλη όπου να αναφέρει το χαρακτηριστικό αυτό. Αντίθετα μια γνωσιακή βάση γνωρίζοντας πως κάθε άνθρωπος έχει το παραπάνω χαρακτηριστικό θα μπορούσε έπειτα, λογικά, να το συμπεράνει για κάθε άνθρωπο που της δίνεται, καθώς θα έχει συνδέσει τις οντολογίες «άνθρωπος» και «εγκέφαλο» με την συσχέτιση «έχει». Η ιδανική αναπαράσταση της αποτελείται από ένα object model (την οντολογία) με classes, subclasses και instances.

Όλη η παραπάνω διασυνδεδεμένη πληροφορία που εμπεριέχεται σε ένα KB έχει ως αποτέλεσμα την κατασκευή ενός Γράφου που συνδέει οντολογίες μεταξύ τους. Αυτός ο Γράφος είναι γνωστός και ως Γνωσιακός Γράφος (Knowledge Graph[2]). Ως επέκταση αυτού διάφορες Γνωσιακές Βάσης μπορούν να διασυνδέσουν τους Γνωσιακούς τους Γράφους μεταξύ τους, ώστε να διαμοιράζονται τις πληροφορίες τους, και έτσι να σχηματίσουν ένα τεράστιο δίκτυο γνωστό και ως Linked Open Data Cloud[3].

Μερικές από τις πιο γνωστές Γνωσιακές Βάσεις είναι:

Wikidata

Είναι ένα ελεύθερο συνεργατικό εγχείρημα που ανήκει στο ίδρυμα Wikimedia. Αποτελεί μια γνωσιακή βάση δομημένων δεδομένων η οποία έχει ως σκοπό την συλλογή πληροφοριών από όλων τον κόσμο, οι οποίες μπορούν έπειτα να χρησιμοποιηθούν από τα projects του Wikimedia (πχ Wikipedia) ή και από οποιονδήποτε άλλον. Ένας τρόπος για προσπέλαση της πληροφορίας είναι online μέσω SPARQL endpoints. Το Wikidata[4] δεν αποθηκεύει μόνο γεγονότα αλλά και τις αντίστοιχες πηγές τους ώστε να μπορεί να ελεγχθεί η ακεραιότητά τους αργότερα. Το data model της Wikidata στηρίζεται σε 2 έννοιες αυτήν του αντικειμένου (item) και αυτή της δήλωσης (statement). Κάθε item αναπαριστά μία οντότητα και χαρακτηρίζεται από ένα αναγνωριστικό (qid) και ενδεχομένως να έχει labels με πληροφορίες, ενώ τα statements είναι links που παραπέμπουν σε αντίστοιχες οντότητες άλλων Wikimedia projects (πχ Wikipedia). Επίσης ένα ακόμα χαρακτηριστικό είναι πως η γνώση αυτή διατίθεται σε πολλές γλώσσες, ενώ ο κάθε χρήστης μπορεί να αλλάξει ή και να προσθέσει νέα πληροφορία.

DBpedia

Αποτελεί και αυτή μια γνωσιακή βάση υπό την μορφή συνδεδεμένων δεδομένων. Σκοπός της είναι η εξαγωγή, διασύνδεση και επαναχρησιμοποίηση δομημένης πληροφορίας ,διαμέσου του web, από την Wikipedia. Τα άρθρα του Wikipedia μπορεί να αποτελούνται κυρίως από ελεύθερο κείμενο αλλά εμπεριέχουν και πρόσθετη πληροφορία όπως εικόνες, γεωγραφικές συντεταγμένες, links σε άλλες σελίδες ή και σε αλλόγλωσση έκδοση του ίδιου του άρθρου. Αυτήν την δομημένη πληροφορία εξάγει το DBpedia[5][17] από το Wikipedia . Για την αναπαράσταση των δεδομένων χρησιμοποιεί RDF τεχνολογία. Για την πρόσβαση σε αυτά τα δεδομένα χρησιμοποιείται μια γλώσσα για queries ειδική για RDF που ονομάζεται SPARQL. Συμπληρωματικά η DBpedia ανήκει στο Linked Open Data Cloud, και διασυνδέεται με μια ποικιλία από πηγές δεδομένων χρησιμοποιώντας RDF συνδέσμους.

YAGO (Yet Another Great Ontology) before YAGO4

Είναι μια ελεύθερη Γνωσιακή Βάση για τον πραγματικό κόσμο που εμπεριέχει οντότητες αλλά και τις συσχετίσεις μεταξύ αυτών. Τα δεδομένα του προέρχονται από Wikipedia, WordNet και Geonames. Πλέον περιέχει πάνω από 10 εκατομμύρια οντότητες (όπως ανθρώπους, οργανισμούς, πόλεις) και 120 εκατομμύρια γεγονότα που σχετίζονται με αυτές τις οντότητες. Το YAGO[6] οργανώνει τις οντότητες του σε κλάσεις (πχ η Αθήνα ανήκει στην κλάση των πόλεων) αλλά και ορίζει τις συσχετίσεις που υπάρχουν μεταξύ δύο entities (πχ «τρώω» είναι μια συσχέτιση μεταξύ ενός ανθρώπου και ενός φαγητού). Συμπληρωματικά κάτι που ξεχωρίζει την οντολογία του YAGO είναι πως σε πολλά από τα γεγονότα και τις οντότητές του αποδίδεται μια χρονική και μια χωρική διάσταση. Ταυτόχρονα η πληροφορία αυτή διατίθεται και σε διαφορετικές γλώσσες. Προφανώς τα παραπάνω αποτελούν αναβαθμίσεις του αρχικού project,YAGO, YAGO2[7], YAGO3. Τα γεγονότα του YAGO ακολουθούν την μορφή RDF, όπου αναπαρίστανται από triples (subject, predicate, object). Το μεγάλο πλεονέκτημα του είναι ότι δίνει πολύ μεγάλη σημασία στην ποιότητα των δεδομένων του, γιατί αυτά που λαμβάνει από τις πηγές του, φιλτράρονται και δεν καταλήγουν όλα στον αποθηκευτικό του χώρο. Στα δεδομένα αυτά μπορεί κάποιος να αποκτήσει πρόσβαση μέσω SPARQL endpoint.

YAGO4

Οι προηγούμενες εκδόσεις του YAGO είχαν ως κύρια ιδέα την περισυλλογή δεδομένων σχετικά με οντότητες, από τα πλαίσια πληροφορίας (infoboxes) και τις κατηγορίες του Wikepedia και τον συνδυασμό τους με έναν οντολογικό άξονα, προερχόμενο από τις κλάσεις του WordNet. Παρόλο που τα αποτελέσματα του είχαν πολύ υψηλή ακρίβεια, καθώς η θορυβώδης πληροφορία αποκόπτεται, αδυνατούσε να φτάσει το αντίστοιχο εύρος και μέγεθος του Freebase ή του Wikidata. Αυτό γινόταν εξαιτίας του ότι το YAGO εστίαζε στην πληροφορία που προερχόταν από τα Wikipedia infoboxes. Πρόσφατα λοιπόν εκδόθηκε το YAGO4[8] το οποίο συνδυάζει το αυστηρό τυπολόγιο του schema.org και τον τεράστιο όγκο δεδομένων του Wikidata. Το αποτέλεσμα είναι 2 δισεκατομμύρια τριπλέτες για 64 εκατομμύρια οντότητες. Το YAGO αποτελεί και αυτό έναν από του κύριους πυλώνες του Linked Open Data cloud και διαμοιράζεται τις πληροφορίες του.

1.5 OpenStreetMap (OSM)

Το OpenStreetMap[1] (OSM)[10] είναι ένας χάρτης με ελεύθερη άδεια, ο οποίος αναπτύσσεται από μια κοινότητα εθελοντών που σκοπό έχουν τον συνεχή εμπλουτισμό και την διατήρηση του με δεδομένα. Τα δεδομένα αυτά μπορεί να είναι δρόμοι, πόλεις, λίμνες, βουνά, χώρες και γενικά ότι μπορεί να καταγραφεί και έχει θέση πάνω σε έναν παγκόσμιο χάρτη. Τα δεδομένα που διατίθενται από το OSM έχουν πολύ σημαντικό ρόλο καθώς η εφαρμογή τους ποικίλει από ηλεκτρονικούς χάρτες μέχρι και την αποκωδικοποίηση κειμένου σε γεωγραφικό πλάτος και ύψος. Το έργο αυτό θα μπορούσε να παρομοιαστεί με την Wikipedia καθώς η επιτυχία αυτής ενέπνευσε τον δημιουργό του OSM, Steve Coast.

Η δομή της πληροφορίας στο OSM ακολουθεί μια τοπολογική δομή δεδομένων, που ως πυρήνας της παρουσιάζονται 4 στοιχεία:

Nodes: Είναι σημεία με μια γεωγραφική θέση, η οποία αποθηκεύεται ως συντεταγμένες (ζευγάρι γεωγραφικό πλάτος - γεωγραφικό μήκος) σύμφωνα με το World Geodetic System. Η χρήση τους πέρα από το να αποτελούν μέρη ενός way είναι να δείξουν χαρακτηριστικά του χάρτη τα οποία δεν έχουν μέγεθος, όπως για παράδειγμα μια κορυφή βουνού ή γενικά κάποιο point of interest(καφέ, μαγαζί).

Ways: Είναι διατεταγμένες λίστες από Nodes και σχηματίζουν γραμμές στον χάρτη. Τα ways μπορούν να σχηματίζουν ένα σύστημα από πολλαπλές γραμμές όπως δρόμους , ποτάμια, ή έχουν την δυνατότητα να σχηματίζουν πολύγονα εάν το σχήμα τους είναι ένας κλειστός βρόχος. Στην περίπτωση αυτήν έχουμε να κάνουμε με περιοχές όπως είναι ένα δάσος ή μια λίμνη.

Relations: Είναι ουσιαστικά συσχετίσεις οι οποίες μπορούν να εμπεριέχουν ως μέλη Nodes, Ways και άλλα Relations. Κάθε ένα από τα μέλη μπορεί να έχει προαιρετικά έναν ρόλο (όπως για παράδειγμα inner outer). Ουσιαστικά οι συσχετίσεις παρουσιάζουν μια σχέση μεταξύ ενός Node και ενός Way όπως για παράδειγμα είναι: πολύγονα ως περιοχές τα οποία έχουν τρύπες εσωτερικά, οι οποίες περιγράφονται από το δικό τους πολύγωνο.

Tags: Είναι ένα ζεύγος κλειδιού-τιμής που παρέχει πληροφορία για τα Nodes, Ways ή και τα Relations. Τέτοια πληροφορία είναι της μορφής name:Όλυμπος όπου στα αριστερά αναγράφεται το χαρακτηριστικό και στα δεξιά η τιμή του. Επιπλέον συνδυασμός διάφορων tags μπορούν να εκφράζουν μια συγκεκριμένη οντότητα. Όπως για παράδειγμα μία λιμνοθάλασσα μπορεί να περιγραφή με την χρήση των tags natural:water & water:lagoon[2].

Το OSM παρέχει τα δεδομένα του με ποικίλους τρόπους ένας από αυτούς είναι μέσω ενός server[3] όπου αυτόν έχω ακολουθήσει στην πτυχιακή μου. Μέσω της σελίδας αυτής μπορεί κάποιος να κατεβάσει τα δεδομένα για την γεωγραφική περιοχή που επιθυμεί. Τα δεδομένα ενημερώνονται καθημερινά και διατίθενται υπό μορφή file.osm.pbf.

[1] https://www.openstreetmap.org/
[2] https://wiki.openstreetmap.org/wiki/Item:Q5595
[3] https://download.geofabrik.de/

1.6 **Yago2geo**

Ο κύριος σκοπός αυτής της πτυχιακής είναι να κρατήσει ενημερωμένο στο σήμερα τον Γνωσιακό Γράφο YAGO2geo[1][11] χρησιμοποιώντας την πληθώρα πληροφοριών που παρέχει το OSM.

Το Yago2geo αποτελεί ένα έργο που αναπτύχθηκε από το πανεπιστήμιό μας από τους Ν. Καραλής και Γ. Μανδηλαράς υπό την επίβλεψη του κ.Μανώλη Κουμπαράκη. Αποτελεί μια επέκταση του YAGO2 που στοχεύει στην αύξηση της ποιότητας της γεωχωρικής πληροφορίας. Το YAGO2 περιέχει μόνο αποκομμένα ζευγάρια συντεταγμένων ως πληροφορία-τοποθεσίας τα οποία δεν περιγράφουν ολοκληρωτικά την γεωμετρία της οντότητας. Το Yago2geo δίνει σχήμα σε αυτά τα ζεύγη εισάγοντας τις έννοιες της γραμμής και του πολύγωνου. Ως αποτέλεσμα εξασφαλίζεται η καλύτερη εικόνα, θέση και ο χώρος που καταλαμβάνουν οι οντότητές. Το Yago2geo αντλεί την πληροφορία από 2 πηγές. Αρχικά χρησιμοποιήθηκαν επίσημα δεδομένα από 3 χώρες: Ελλάδα, Ηνωμένο Βασίλειο και Βόρεια Ιρλανδία. Δεύτερον προκειμένου να παρθούν δεδομένα από όλων τον κόσμο χρησιμοποιήθηκαν οι πηγές : GADM και OSM.

Έτσι λοιπόν μια από τις κύριες πηγές του Yago2geo είναι το OSM. Όμως, όπως είδαμε, το OSM αποτελεί το μεγαλύτερο εθελοντικό, crowdsourced και ανοιχτό για το κοινό σύνολο δεδομένων. Αυτό έχει ως αποτέλεσμα μια πλούσια αλλά και ταυτόχρονα συνεχώς μεταβαλλόμενη πηγή δεδομένων η οποία αλλάζει μέρα με την μέρα.

Προκειμένου λοιπόν οι γεωμετρίες του Yago2geo σε παγκόσμιο επίπεδο να παραμείνουν ενημερωμένες θα πρέπει να γίνεται ανά κάποια μικρά χρονικά διαστήματα ημερών, μια αναβάθμιση της πληροφορίας από την πηγή. Έτσι εξασφαλίζεται η ακεραιότητά και η ποιότητα της.

Ο τρόπος με τον οποίο εξασφαλίζεται η πρόσβαση στην πληροφορία θα είναι με την χρήση stSPARQL [12] και GeoSPARQL[13]query που γίνονται πάνω σε ένα συγκεκριμένο endpoint το οποίο περιέχει την RDF-δομημένη πληροφορία.

1.7 Goal and Structure

Ο κύριος σκοπός της πτυχιακής εργασίας είναι να συνεισφέρει στο project του YAGO2geo κρατώντας τις πληροφορίες προερχόμενες από το OpenStreetMap ενήμερες στα πιο πρόσφατα δεδομένα. Το OSM περιέχει μεγάλο όγκο πληροφοριών για τα γεωχωρικά δεδομένα, προερχόμενες βέβαια από το κοινό(που συνεπάγεται πως εμπεριέχει noisy δεδομένα). Η πληροφορία που εξάγει το YAGO2geo από το OSM δεν είναι ατόφια. Επιλέγεται μόνο όσα δεδομένα διασυνδέθηκαν με υπάρχουσες οντότητες του YAGO. Με αυτόν τον τρόπο αποβάλλεται ο θόρυβος, η επιπλέον και ανακριβής πληροφορία και γίνεται η αναπαράστασή τους ύπο RDF μορφή. Αυτό δηλώνει πως τα παραπάνω δεδομένα μπορούν να διατεθούν στο ευρύτερο πλαίσιο του Semantic Web. Τα δεδομένα αυτά θα μπορούσαν να χρησιμοποιηθούν από άλλες εφαρμογές ή και να διασυνδεθούν με άλλα. Για αυτόν τον λόγο θα πρέπει η πληροφορία αυτή να ενημερώνεται ανά κάποια χρονικά διαστήματα και όχι να παραμένει στάσιμη ή προκειμένου να ενημερωθεί να χρειαστεί να γίνει εκ νέου η διαδικασία. Στο δεύτερπ κεφάλαιο βλέπουμε projects(DBpedia Live, LinkedGeoData) τα οποία στοχεύουν στο να κρατήσουν την πληροφορία τους ενημερωμένη στο σήμερα σε σχέση με την πηγή τους. Επίσης εξετάζουμε και το εργαλείο TripleGeo που

[1] http://yago2geo.di.uoa.gr/

διαχειρίζεται γεωχωρικά δεδομένα. Τέλος στο τρίτο κεφάλαιο αναπτύσσουμε το δικό μας update του knowledge base, με την χρήση διάφορων osm εργαλείων για την επεξεργασία των OSM δεδομένων.

2 Related Work

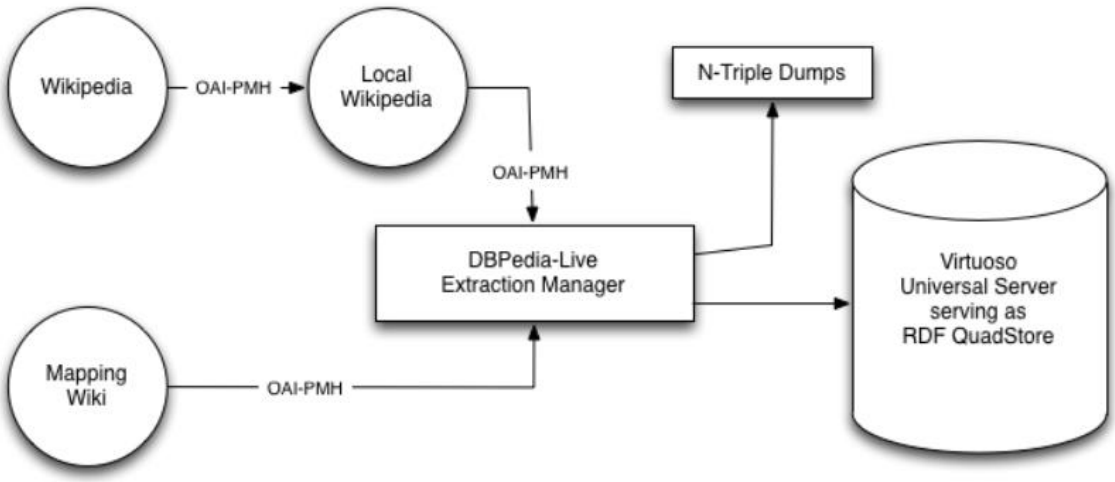
2.1 DBPedia Live

Το DBpedia Live[14] ανακτά όλες τις τροποποιήσεις που λαμβάνουν χώρα στο Wikipedia χωρίς καθυστέρηση και εξάγει όλη την πληροφορία και την μεταφέρει σε μια online SPARQL database. Σε αυτό έπειτα γίνονται τα όποια ερωτήματα (queries) που χρειάζονται. Το εργαλείο εξαγωγής μπορεί να χειριστεί μέχρι και 1 εκατομμύριο αλλαγές ανά μέρα σε έναν κοινό server, ενώ τα online query τα διαχειρίζεται η Virtuoso Database.

Το DBPedia αποθηκεύει περιεχόμενο από την Wikipedia το οποίο αναπαριστά ως ένα Σημασιολογικό Ιστό (Semantic Web) από Διασυνδεδεμένη Μνήμη. Η αρχική αναπαράσταση αυτή προήλθε από στατικά δεδομένα με πληροφορία από την Wikipedia και χρειάστηκε 6 μήνες μέχρι να δημοσιευθεί το DBPedia. Από τότε και έπειτα η αναβάθμιση του DBPedia γινόταν περιοδικά ανά 6-18 μήνες ακολουθώντας την προηγούμενη διαδικασία λαμβάνοντας όμως υπόψη τα καινούρια Wikipedia dumps που υπήρχαν. Για αυτό και τα δεδομένα του DBPedia ήταν πάντα 6-18 μήνες πίσω από τα αντίστοιχα της Wikipedia.

Επειδή λοιπόν η χρήση της DBPedia είχε ραγδαία αύξηση σε συνδυασμό με τα δυναμικά και συνεχώς μεταβαλλόμενα δεδομένα του Wikipedia αυτό έκανε έντονη την ανάγκη για συνεχή ενημέρωση των δεδομένων της DBPedia μέσω της επεξεργασία των Wikipedia changelogs. Αποτέλεσμα των παραπάνω αποτελεί το εργαλείο DBpedia Live.

2.1.1 The components of DBPedia Live



2.1 DBPedia Live System Architecture

Τα βασικά μέρη του DBpedia Live είναι τα εξής:

- Local Wikipedia: Έχει εγκατασταθεί ένα τοπικό αντίγραφο του Wikipedia το οποίο διατηρείται συγχρονισμένο με το Wikipedia. Με την χρήση του πρωτοκόλλου Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMN) δίνει την δυνατότητα στην εφαρμογή να έχει μια συνεχή ροή με ενημερώσεις που προέρχονται από το wiki. Το OAI-PMN επίσης χρησιμοποιείται για να φορτώσει τα updates στο DBPedia-Live Extraction Manager.

- Mapping Wiki: Ουσιαστικά είναι η χαρτογράφιση του DBpedia[1]. Χρησιμοποιώντας το πρωτόκολλο OAI-PMN παίρνουμε μια συνεχή ροή από ενημερώσεις στο DBPedia mappings. Αυτό γιατί μια αλλαγή στο mapping μπορεί να επηρεάσει αρκετές σελίδες της Wikipedia, και γι' αυτό θα πρέπει όλες να ξαναεπεξεργαστούν έπειτα.

- DBPedia Live Extraction Manager: Αυτό το component είναι ουσιαστικά το DBPedia-Live extraction framework. Όταν έρχεται μια σελίδα (που έχει γίνει προσφάτως updated),η δομή αυτή απομονώνει και αποσπά την καινούρια RDF πληροφορία. Αυτές οι νέες RDF δηλώσεις αποθηκεύονται στο backend data store, όπου και αντικαθιστούν την αντίστοιχη παλιότερη πληροφορία. Οι προσφάτως εξαγόμενες RDF δηλώσεις κρατούνται και σε έναν συμπιεσμένο N-Triples αρχείο. Αυτό γίνεται έτσι ώστε να δοθεί η δυνατότητα σε εφαρμογές παρόμοιας λογικής, δηλαδή που λαμβάνουν diff files ώστε να κάνουν ενημέρωση των δεδομένων τους, να μπορέσουν να τα εισάγουν στο δικό τους RDF data store.

2.1.2 DBPedia Extraction Framework

Το DBPedia Extraction Framework είναι ένα λογισμικό γραμμένο σε Scala που διακρίνεται από μια ποικιλία λειτουργιών γύρω από την εξαγωγή RDF γνώσης-πληροφορίας γύρω από τα Wikis. Ένα κομμάτι αυτού αποτελεί το DBPedia Live module, το οποίο έχει σκοπό να παρέχει συνεχόμενα μια ενημερωμένη έκδοση του DBPedia, με επεξεργασία σελίδων Wikipedia αμέσως μετά την ενημέρωσή τους από κάποιον χρήστη. Ο πυρήνας του εργαλείου αποτελεί μια ουρά(queue), που γεμίζει με τις πιο πρόσφατες αλλαγμένες σελίδες, σε συνδυασμό με μια σχεσιακή βάση δεδομένων, την Live Cache, που διαχειρίζεται τις διαφορές-αλλαγές μεταξύ μιας παλιάς και μιας ανανεωμένης σελίδας. Προκειμένου να γίνει αυτό συνδέεται με μια Wiki-instance στην οποία καταγράφονται οι αλλαγές των σελίδων αυτών.

Αφού τροφοδοτείται η ουρά η διαδικασία που ακολουθεί είναι η εξής:

- Εξάγεται μια σελίδα από την ουρά
- Triples Εξάγονται από την σελίδα σύμφωνα με κάποιους extractors
- Τα νέα Triples συγκρίνονται με τις αντίστοιχες παλιές από το Live Cache

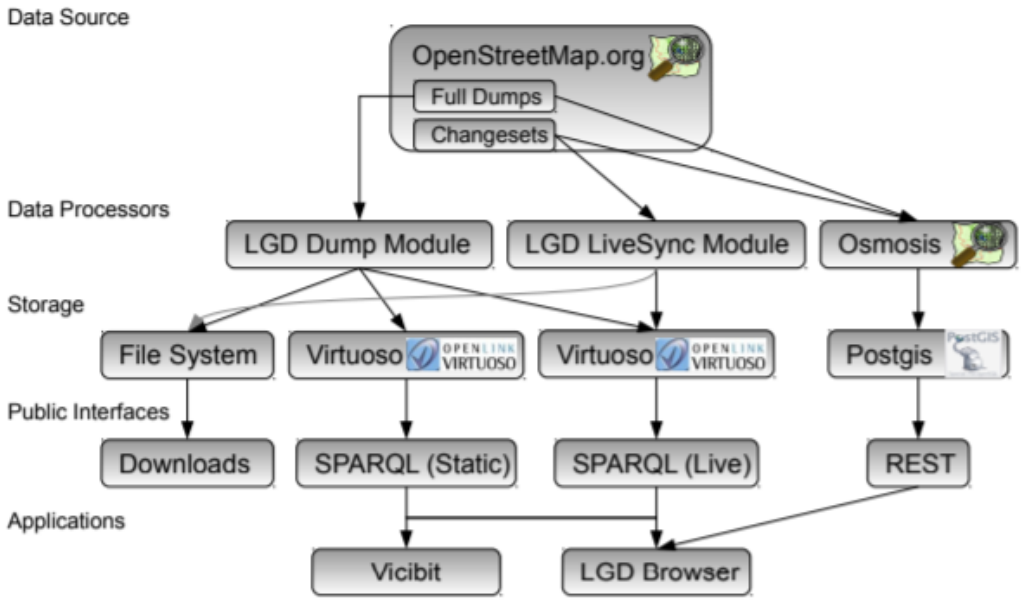
[1] http://mappings.dbpedia.org

- Τα Triples που έχουν διαγραφεί ή αυτά που έχουν προστεθεί κρατούνται σε έναν φάκελο και το Live Cache ενημερώνεται.

2.2 LinkedGeoData (LGD)

Ο στόχος του project LinkedGeoData[15] είναι να ανεβάσει τα δεδομένα του OpenStreetMap στο Semantic Web. Ένα τέτοιο εγχείρημα μπορεί να επιλύσει προβλήματα της πραγματικής ζωής, τα οποία χρειάζονται ένα υπόβαθρο περιεκτικής γνώσης σχετικά με χωρικά χαρακτηριστικά (π.χ. την ύπαρξη αξιοθέατων κατά μήκος ενός ποδηλατοδρόμου). Αρχικά θα πρέπει την πληροφορία που παίρνουν από το OSM να την μετατρέψουν σε RDF μορφή. Αφού υλοποιηθεί αυτό το βήμα, η όποια πληροφορία μπορεί να διασυνδεθεί-διαμοιραστεί αργότερα με άλλους Γνωσιακούς Γράφους όπως το DBPedia και GeoNames. Παράλληλα τα δεδομένα θα πρέπει να συνάδουν με το σήμερα, δηλαδή να ενημερώνονται τακτικά ή live.

2.2.1 Αρχιτεκτονική LGD



2.2 LinkedGeoData Architecture

Παρατηρώντας την αρχιτεκτονική του LGD φαίνεται πως η πληροφορία του OSM ακολουθεί 3 διαφορετικές διαδρομές. Ο LGD Dump Module μετατρέπει έναν OSM planet file σε RDF και φορτώνει τα παραγόμενα δεδομένα σε ένα triple store. Μια αντιγραφή του triple store αυτού διατίθεται ως βάση σε ένα live SPARQL endpoint. Το LGD Live Sync Module κατεβάζει changesets ανά λεπτό από το OSM και υπολογίζει τα αντίστοιχα changesets σε RDF επίπεδο ώστε να αναβαθμιστεί και αργότερα το triple store. Ομοίως με το DBPedia το LGD παράγει και αυτό δικά του RDF changesets ώστε να μπορούν οι χρήστες του να συγχρονίζουν τα δικά τους triples χρησιμοποιώντας τα LGD triples.

Για την πρόσβαση στα δεδομένα αυτά το LinkedGeoData προσφέρει: downloads, μια REST API διεπαφή, Linked Data και SPARQL endpoints.

2.2.2 Live Synchronization

Προκειμένου να γίνει συγχρονισμένη ενημέρωση του LGD[16] με το OSM θα πρέπει να ενημερώνονται τα δεδομένα σε πολύ σύντομο χρονικό διάστημα. Το OSM έχει προνοήσει για τέτοιες καταστάσεις και για αυτό ανεβάζει changesets ανά μέρα, ώρα αλλά και λεπτό με όλες τις αλλαγές που έχουν προκύψει κατά το διάστημα αυτό (έχοντας ως αφετηρία το αμέσως προηγούμενο changeset). Για να είναι live οι αλλαγές το LGD, επιλέγει να χρησιμοποιεί τα minutely changesets ως η πιο άμεση και συντομότερη επιλογή. Έτσι θα υπάρχει η ψευδαίσθηση ότι οι αλλαγές γίνονται live με την μικρή καθυστέρηση του ενός λεπτού.

Αυτό δημιουργεί όμως και 2 προβλήματα. Αρχικά η όλη διαδικασία επεξεργασίας και update θα πρέπει να γίνεται σε πολύ μικρό χρονικό διάστημα, λιγότερο του ενός λεπτού, ώστε να είναι έτοιμο για το επόμενο minutely changeset. Επιπλέον δεν θα πρέπει να αφήνει κατάλοιπα, δηλαδή όλες οι RDF δηλώσεις που σχετίζονται με την αλλαγμένη οντότητα θα πρέπει να αναφέρονται στην πιο πρόσφατη κατάστασή της και όχι σε κάποια προηγούμενη.

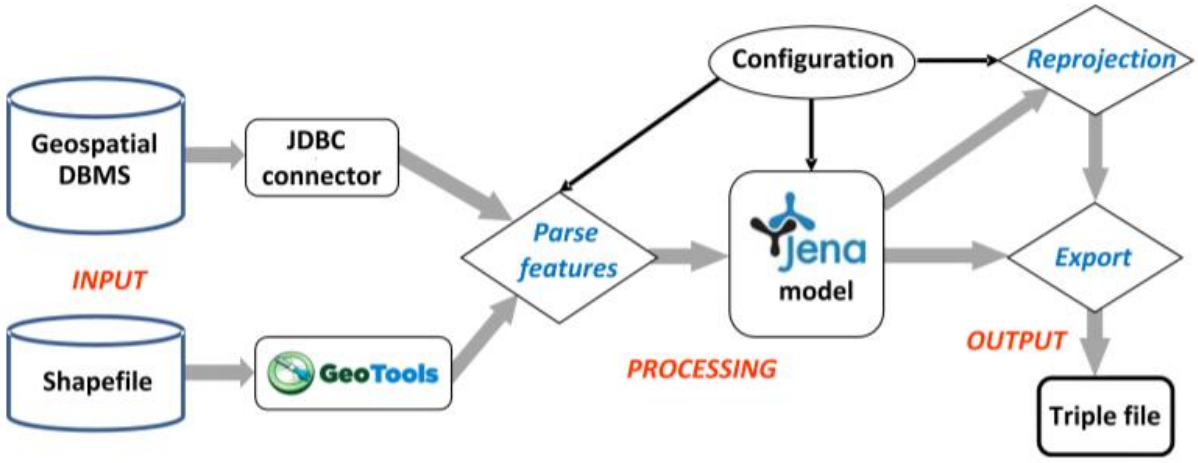
Το LGD όταν επεξεργάζεται ένα OSC αρχείο παράγει 2 αντίστοιχα αρχεία όπου το ένα έχει όλες τις τριπλέτες που θα προστεθούν και το άλλο όλες τις τριπλέτες που θα προστεθούν στο RDF store.

Τέλος αξίζει να σημειωθεί ότι πριν το προηγούμενο βήμα τα changeset υφίστανται ένα φιλτράρισμα το οποίο διατηρεί μόνο τα node-ways τα οποία έχουν τον επιθυμητό συνδυασμό key-value στα tags τους. Για παράδειγμα μπορεί να απορρίπτει τα tags με key='railway' εκτός και εάν έχουν value='station'. Αυτό γίνεται δημιουργώντας 2 lists μια με black-listed tags και μια με white-listed tags.

Σε γενικές γραμμές το εργαλείο αυτό έχει πάψει να εξελίσσεται και έχει εγκαταλειφθεί.

2.3 TripleGeo

Το TripleGeo[11] είναι ένα open-source εργαλείο που ανήκει στην ευρύτερη κατηγορία των Extract-Transform-Load (ETL) συστημάτων. Έχει την δυνατότητα να εξάγει γεωχωρικά χαρακτηριστικά από διάφορες πηγές και να τα μετατρέψει σε τριπλέτες, οι οποίες με την σειρά τους αποθηκεύονται σε ένα RDF store. Το Triplegeo μπορεί να εξάγει γεωμετρίες και χαρακτηριστικά είτε από αρχεία που ακολουθούν κάποια συγκεκριμένη γεωγραφική μορφή (OSM, PBF, CSV) είτε από ευρέως χρησιμοποιούμενα DBMSs. Μια επιπρόσθετη λειτουργία που έχει είναι ότι μπορεί να μεταγλωττίσει τις γεωμετρίες με βάση ένα διαφορετικό Coordinate Reference System, πριν τις εξάγει στην τελική μορφή τους. Τέλος τα triples μπορούν να εξάγονται υπό την μορφή του GeoSPARQL πρότυπου, αλλά και υπό την μορφή άλλων λεξιλογίων.



2.3 Αρχιτεκτονική TripleGeo

2.3.1 TripleGeo Components

- Input Data: Όπως περιεγράφηκε μπορεί να είναι είτε από φακέλους με γεωγραφική πληροφορία, είτε από DBMSs.
- Connectors: Προκειμένου να πάρουμε την γεωγραφική πληροφορία από το DBMS χρειαζόμαστε τον κατάλληλο JDBC driver, ενώ για να ανακτήσουμε την ίδια πληροφορία από ένα αρχείου χρησιμοποιείται η GeoTools βιβλιοθήκη.
- Configuration: Το κέντρο ελέγχου της εφαρμογής αποτελεί ένα configuration αρχείο το οποίο προσδιορίζεται από τον χρήστη και έχει μέσα πληροφορίες όπως: τον προσδιορισμό του input, το είδος των δεδομένων, τις γεωμετρικές αναπαραστάσεις κ.ά.
- Parser: προσπελαύνει κάθε εγγραφή εισόδου και μετατρέπει κάθε γεωμετρία στην κατάλληλη αναπαράσταση σύμφωνα με το Configuration.
- Jena Model: Είναι ένα main-memory data structure το οποίο χρησιμοποιείται για να διατηρήσει όλη την πληροφορία των triples, με την μορφή ενός RDF Γράφου.
- Reprojection: Είναι μια προαιρετική ενέργεια κατά την οποία οι γεωμετρίες αλλάζουν σύμφωνα με κάποιο CRS.
- Export: Τέλος τα triples εξάγονται σε ένα αρχείο με την βοήθεια του Jena API. Ο χρήστης έχει την δυνατότητα να επιλέξει μεταξύ διάφορων triples format, καθορίζοντας την επιλογή του στο configuration.

3 UPDATE YAGO2geo USING OSM DATA

3.1 Γενική δομή της διαδικασίας

Αρχικά η διαχείριση και ενημέρωση των δεδομένων γίνεται με την χρήση της βάσης δεδομένων PostgreSQL και κάποιων OSM εργαλείων(Git repository[1]).

Ολόκληρη η διαδικασία μπορεί να χωριστεί σε 5 απλούστερα βήματα:

1. Κατέβασμα των πιο πρόσφατων OpenStreetMap δεδομένων
2. Προσπέλαση και επεξεργασία των RDF δεδομένων του YAGO2geo.ttl αρχείου(μια μορφή αρχείου που περιέχει τα δεδομένα του YAGO2geo) και ανέβασμα αυτών στην βάση PostgreSQL.
3. Ανέβασμα όλων των OSM δεδομένων από το βήμα 1 στην ίδια βάση δεδομένων.
4. Ενημέρωση των YAGO2geo δεδομένων χρησιμοποιώντας τα OSM δεδομένα της βάσης.
5. Ενημέρωση του YAGO2geo KB με την χρήση SPARQL queries.

3.2 Set up-Προετοιμασία

Πριν αρχίσει η διαδικασία θα πρέπει να κατεβάσουμε έναν osm.pbf αρχείο από το Geofabrik server ο οποίος θα εμπεριέχει όλα τα δεδομένα από την πλευρά του OSM.

Επίσης απαιτείται να κατεβάσουμε τα δεδομένα του YAGO2geo τα οποία διατίθενται σε ένα .ttl αρχείο (Turtle). Terse RDF Triple Language (Turtle) είναι ένας συγκεκριμένος τρόπος σύνταξης ενός αρχείου για να εκφράζονται τα RDF data model.

Συμπληρωματικά θα χρειαστούμε την PostgreSQL βάση δεδομένων μαζί με την Postgis (προέκταση του PostgreSQL για την διαχείριση χωρικών δεδομένων).

Τέλος γίνεται η χρήση μερικών εργαλείων της οικογένειας των OSM tools σύμφωνα με τα οποία τα .osm.pbf αρχεία θα επεξεργαστούν (αναφέρονται λεπτομερώς παρακάτω).

3.2.1 Geofabrik server

Ο Geofabrik είναι ένας server από τον οποίο μπορεί κάποιος να εξάγει πληροφορία από το OpenStreetMap project. Τα δεδομένα που διατίθενται στον server ενημερώνονται καθημερινά. Επίσης ο συγκεκριμένος server διαθέτει τα δεδομένα του οργανωμένα σε περιοχές. Ειδικότερα στο root directory διαθέτει αρχεία τα οποία εμπεριέχουν ολόκληρη την πληροφορία για κάθε ήπειρο, και έπειτα στο subdirectory κάθε μιας από αυτές εμπεριέχει αρχεία τα οποία αντιστοιχούν στις πληροφορίες των χωρών της εκάστοτε ηπείρου. Η μορφή των φακέλων που εμπεριέχουν τα γεωχωρικά δεδομένα μπορεί να είναι σε μορφή PBF, XML ή και ESRI shapefile. Τα αρχεία με κατάληξη .osm.pbf περιέχουν τα δεδομένα του OSM στην μορφή PBF. Συμπληρωματικά ο server διαθέτει και καθημερινά diff files τα οποία περιέχουν όποιες αλλαγές γίνονται ανά μέρα. Τα diff files ακολουθούν την συμπιεσμένη μορφή OSM XML με κατάληξη .osc.gz.

[1] https://gitlab.com/mikee201097/yago2geo_update

3.2.2 PBF Format

Το PBF Format (Protocolbuffer Binary Format) προορίζεται ως μια εναλλακτική για το XML format. Συγκριτικά τα δεδομένα για έναν ολόκληρο πλανήτη συμπιεσμένη με PBF format έχει περίπου το μισό μέγεθος από δεδομένα συμπιεσμένα ως gzipped μορφή και περίπου 30% μικρότερο μέγεθος από μία bzipped μορφή. Επιπλέον η γραφή είναι 5 φορές ταχύτερη σε PBF απ' ότι σε έναν gzipped αρχείο, και η ανάγνωσή του είναι 6 φορές πιο γρήγορη από την αντίστοιχη ενός gzipped αρχείου. Ουσιαστικά αυτή η μορφή αρχείου σχεδιάστηκε έτσι ώστε να υποστηρίζει μελλοντική επεκτασιμότητα και ευελιξία.

Τα osm.pbf αρχεία έχουν στην αρχή τα nodes, δεύτερα αναφέρουν τα ways και τελευταία την πληροφορία για τα relations. Τα nodes χαρακτηρίζονται από γεωγραφικές συντεταγμένες (latitude, longitude), ενώ τα ways και τα relations απλά χαρακτηρίζονται από αναφορές στα id των nodes και στα nodes, ways και relations αντίστοιχα.

3.3 Αρχική Προσέγγιση

Πρόκειται για την αρχική ιδέα η οποία αναπτύχθηκε αλλά εν συνεχεία δεν αποδείχθηκε αποτελεσματική και απορρίφθηκε. Αρχικά έγινε προσπάθεια να προσεγγίσουμε το 2ο βήμα, κατά το οποίο θα πρέπει να περάσουν τα δεδομένα από το .osm.pbf στην βάση υπό το επιθυμητό schema.

Η στρατηγική που ακολουθήθηκε ήταν η εξής:

- Να διαβάζεται το osm.pbf αρχείο με την χρήση κάποιου Reader
- Καθώς συλλέγουμε την πληροφορία, αυτή φορτώνεται στην PostgreSQL database
- Όμως επειδή δεν χρειάζονται όλες οι οντότητες για να ενημερωθεί η πληροφορία του YAGO2geo, μιας και το YAGO2geo περιέχει μόνο συγκεκριμένες κατηγορίες (forest, lakes..), δεν χρειάζεται να φορτωθεί όλη η πληροφορία από το osm.pbf file στην βάση δεδομένων.
- Έπειτα να ταξινομηθούν τα δεδομένα σε 16 διαφορετικά tables(όσες και οι διάφορες κατηγορίες) στο database ανάλογα με την κατηγορία που ανήκουν.

Στο 1ο στάδιο χρησιμοποιήθηκε το εργαλείο **osmium** με το flag add-locations-to-ways. Δίνοντας ως input το αρχικό μου osm.pbf αρχείο εκείνο επιστρέφει ως output ένα αντίγραφο του πρώτου, με την διαφορά ότι παίρνει τις συντεταγμένες των nodes και τις προσθέτει στα αντίστοιχα ways. Ακολουθώντας την παραπάνω λογική τα ways πλέον δεν αποτελούνται μόνο από ένα σύνολο από nodes, αλλά περιέχουν και την δικιά τους γεωχωρική πληροφορία. Χρησιμοποιήθηκε ένας PbfReader με τον οποίο διαβάζουμε το παραγόμενο από το osmium osm.pbf αρχείο με την χρήση πολλών νημάτων(Threads). Καθώς γίνεται το διάβασμα συγκεντρώνουμε τις πληροφορίες για - id, name, Geometry - και τις εισάγουμε στο αντίστοιχο table που ανήκει σύμφωνα με την κατηγορία.

Μέχρι και για τα ways ακολουθήθηκε η παραπάνω διαδικασία. Τα relations όμως δεν έχουν την γεωγραφική πληροφορία που χρειάζεται, αλλά αναφορές στα id των ways και των nodes από τα οποία απαρτίζονται. Γι' αυτό ενώ διαβάζουμε την πληροφορία των ways και των nodes αποθηκεύεται ένα αντίγραφό για καθένα από αυτούς σε ένα βοηθητικό table που λέγεται osm_all_data. Όταν διαβάζονται τα relations βλέπουμε τα id στα οποία αναφέρονται και έπειτα εξάγεται η πληροφορία που χρειάζεται από το osm_all_data table που δημιουργήθηκε προηγουμένως. Αφού αποσπαστεί εξ' ολοκλήρου η γεωγραφική πληροφορία, μπορεί να προστεθεί το κάθε ένα relation και στο κατάλληλο table-κατηγορία όπου ανήκει.

Στο τέλος αυτής της διαδικασίας θα έχουν με επιτυχία περάσει όλα τα OSM δεδομένα μαζί με τις γεωμετρίες τους στα διαφορετικά tables.

3.3.1 OSM γεωγραφική πληροφορία σε Γεωμετρίες

Το OSM δεν περιέχει την γεωχωρική πληροφορία σε μορφή γεωμετρίας πάνω στον χάρτη. Ουσιαστικά, αυτό με το οποίο μας προμηθεύει είναι σημεία και συλλογές σημείων. Γι' αυτό πριν αποθηκευτεί η πληροφορία στην βάση μας θα πρέπει να αποκωδικοποιηθεί ο συνδυασμός από τα σημεία που μας προσφέρει το OSM και να δημιουργηθεί η αντίστοιχη γεωμετρία.

Η κάθε γεωμετρία παράγεται ως εξής:

Στην περίπτωση που είναι ένα node τότε η γεωμετρία του περιγράφεται απλά ως ένα Point με τις συντεταγμένες που δίνονται.

Εάν είναι η οντότητα τύπου way τότε η υπάρχουν 2 περιπτώσεις. Η πρώτη είναι να περιγράφεται ως ένα απλό line. Η δεύτερη είναι πως εάν αυτό το line είναι δακτυλίδι, Δηλαδή ενώνεται η αρχή και το τέλος του τότε αυτό σε γεωμετρία περιγράφεται ως ένα πολύγωνο.

Τέλος, στην περίπτωση των relations χρησιμοποιώντας την JTS βιβλιοθήκη ανάλογα τις γεωμετρίες από τις οποίες αποτελείται, γίνεται ο συνδυασμός αυτών σε μία. Στην δυσμενέστερη περίπτωση εάν οι γεωμετρίες δεν συνδυάζονται μεταξύ τους το αποτέλεσμα θα είναι ένα geometry collection.

3.3.2 Λόγοι απόρριψης αυτής της λογικής

Η παραπάνω λογική έκανε την δουλεία όπως προβλεπόταν και τα αποτελέσματά της έβγαιναν σωστά. Τα προβλήματα που είχε ήταν τα εξής:

- Η διαδικασία αποδείχθηκε αρκετά χρονοβόρα. Παρόλο που χρησιμοποιήθηκε παραλληλία, για ένα μικρό δείγμα από OSM data, αυτό της Ελλάδας(200 MB) έκανε πολύ ώρα. Η καθυστέρηση προκλήθηκε από ένα σύνολο παραγόντων αλλά ο κυριότερος ήταν ο τρόπος διαχείρισης των relations. Για κάθε ένα μέρος από ένα relation έπρεπε να γίνει query στην βάση για να αποσπαστεί η πληροφορία από το osm_all_data table. Πρακτικά το μεγαλύτερο μέρος του χρόνου καταλάμβανε η επικοινωνία, δηλαδή τα queries προς την βάση για την δημιουργία των relations.
- Η χρήση του εργαλείου osmium απλά για την δημιουργία ενός αντίγραφου του αρχικού με την έξτρα πληροφορία στα ways. Εάν επρόκειτο για ένα μικρό αρχείο δεν θα υπήρχε κάποιο πρόβλημα αλλά για δεδομένα απ' όλο τον πλανήτη θα έπρεπε να διπλασιάσουμε τον αποθηκευτικό χώρο.
- Δεν υπήρχε κάποιος τρόπος να ξεχωρίσει αργότερα η νεότερη πληροφορία. Δυσμενέστερη έκβαση αποτελούσε η περίπτωση όπου θα έπρεπε να ξανακατέβει όλο το updated .osm.pbf αρχείο εξαρχής και να ξαναπεράσει ολόκληρο στην βάση μαζί με τις αλλαγές.

3.4 Υλοποίηση του Update Process v2

Η διαδικασία μπορεί να επιμεριστεί σε 2 καταστάσεις:

- **Initialization**: Όπου περιγράφει την διαδικασία που θα εκτελεστεί κατά την πρώτη φορά εκτέλεσης του προγράμματος.
- **Update**: Όπου αυτή η διαδικασία θα ακολουθείται όλες τις επόμενες φορές μετά το Initialize.

Ουσιαστικά το initialization αποτελεί μια κατάσταση κατά την οποία αρχικοποιείται η διαδικασία που θα ακολουθηθεί.

3.4.1 Initialization-Προετοιμασία (Setup)

Κατά το Initialization πρέπει να στηθούν κάποια προαπαιτούμενα με σκοπό την μετέπειτα εξέλιξη της διαδικασίας.

- Να φορτωθούν τα δεδομένα του YAGO2geo στην βάση δεδομένων.
- Να κατέβει το πιο πρόσφατο αρχείο με τα πιο πρόσφατα OSM δεδομένα από τον Geofabrik server. Προτιμάται το osm.pbf format.

3.4.1.1 YAGO2geo Δεδομένα

Πρέπει να περάσουν τα δεδομένα του YAGO2geo στην βάση. Αυτό το βήμα χρειάζεται για να γίνει η διασύνδεση καθώς δεν προστίθενται νέες οντότητες και δεν διαγράφονται οι ήδη υπάρχουσες. Σκοπός είναι οι οντότητες που εξάγονται από το OSM να ενημερώσουν, με τις όποιες αλλαγές υπέστησαν, τις αντίστοιχες οντότητες του YAGO2geo. Για αυτό χρειάζεται να ανέβουν στην βάση τα δεδομένα του YAGO2geo έτσι ώστε να λάβουμε υπόψη μας από την πλευρά του OSM μόνο εκείνες τις οντότητες που έχουν κοινό osm id με εκείνες του YAGO2geo.

```
<http://yago-knowledge.org/resource/geoentity_Garden_Avenue_Park_7306206>
        a       <http://kr.di.uoa.gr/yago2geo/ontology/OSM_park> ;
        <http://kr.di.uoa.gr/yago2geo/ontology/hasOSM_ID>
                "way/388257242" ;
        <http://kr.di.uoa.gr/yago2geo/ontology/hasOSM_Name>
                "Garden Avenue Park" ;
        <http://www.opengis.net/ont/geosparql#hasGeometry>
                <http://kr.di.uoa.gr/yago2geo/resource/Geometry_osm_9e915b29-0593-
3d7f-8513-8c8958feff91> .

<http://kr.di.uoa.gr/yago2geo/resource/Geometry_osm_faf85217-60ac-3e27-bf09-7b3931
0ffd25>
        <http://www.opengis.net/ont/geosparql#asWKT>
                "<http://www.opengis.net/def/crs/EPSG/0/4326> POLYGON ((-117.68961
81 35.640303200000005, -117.6884916 35.6404729, -117.68788400000001 35.6403495, -1
17.68783970000001 35.6394545, -117.68788400000001 35.6393928, -117.68897890000001
35.6393362, -117.68906120000001 35.639361900000004, -117.6891688 35.63980940000000
4, -117.689289 35.6399586, -117.6894979 35.6399792, -117.6896181 35.64030320000000
5))"^^<http://www.opengis.net/ont/geosparql#wktLiteral> .
```

3.1 Format of YAGO2geo.ttl file

Για να γίνει η φόρτωση αυτών των δεδομένων θα χρησιμοποιηθεί το .ttl αρχείο που περιέχει τα δεδομένα του YAGO2geo σε μορφή turtle. Για να εξάγουμε τις πληροφορίες των triples από το αρχείο ακολουθείται η εξής λογική: Αντί να φορτώνουμε το αρχείο στην μνήμη, προσπελαύνουμε το αρχείο, συλλέγουμε τις απαραίτητες πληροφορίες για την κάθε οντότητα και εισάγονται στην βάση, πριν προχωρήσουμε στην επόμενη οντότητα. Με αυτόν τον τρόπο αποφεύγουμε να φορτώσουμε όλα τα δεδομένα στην μνήμη που μπορεί να οδηγήσει σε Memory Errors.

Για την καλύτερη αναπαράσταση της πληροφορίας δημιουργούμε ένα table yago_all_data το οποίο θα αποθηκεύσει όλες τις πληροφορίες αλλά και τις αλλαγές αργότερα. Στα columns με κατάληξη _updated κρατείται η ενημερωμένη από το OSM πληροφορία.

```
                    Table "public.yago_all_data"
     Column      |          Type           | Collation | Nullable | Default
-----------------+-------------------------+-----------+----------+---------
 id              | text                    |           | not null |
 name            | text                    |           |          |
 name_updated    | text                    |           |          |
 category        | text                    |           |          |
 category_updated | text                   |           |          |
 link            | text                    |           |          |
 geometry        | text                    |           |          |
 flag            | integer                 |           |          |
 geom_prev       | geometry(Geometry,4326) |           |          |
 geom_updated    | geometry(Geometry,4326) |           |          |
Indexes:
    "yago_all_data_pkey" PRIMARY KEY, btree (id)
```

3.2 yago_all_data table

Το πρόβλημα που συναντάται από την σειριακή προσπέλαση (εικόνα 3.1) είναι πως το triple που περιγράφει την γεωμετρία δεν αντιστοιχεί στην οντότητα που περιγράφεται από πάνω. Αυτό σημαίνει πως οι γεωμετρίες δεν είναι μαζί με τις υπόλοιπες πληροφορίες της οντότητας στην οποία αντιστοιχούν, αλλά διασκορπισμένες μες στο αρχείο. Γι' αυτό υπάρχει ένα βοηθητικό table yago_geos το οποίο αποθηκεύει τις γεωμετρίες και έπειτα τις αντιστοιχεί στο yago_all_data αφού φορτωθούν όλες από το Turtle αρχείο.

3.4.2 Initialization-Ανέβασμα των OSM δεδομένων στην PostgreSQL

Κατά το Initialization state πρέπει να δοθεί ως input ο file.osm.pbf που περιέχει όλη την OSM πληροφορία.

Τα γεωχωρικά δεδομένα από το OSM δεν αναπαριστούν την γεωμετρία ή κάποιο σχήμα. Το εργαλείο **osm2psql** μετατρέπει το γενικότερο OSM schema (nodes, ways, relations) σε πραγματικές γεωμετρίες για την κάθε οντότητα και τις εισάγει στην βάση δεδομένων. Για να φορτώσει τα δεδομένα στην βάση δημιουργεί έξι tables:

- Τα planet_osm_nodes, planet_osm_ways, planet_osm_rels tables που αναφέρονται στην OSM πληροφορία όπως ακριβώς αποδίδεται από το osm αρχείο.
- Τα planet_osm_point, planet_osm_line, planet_osm_polygon που αναφέρονται στις γεωμετρίες που σχηματίζονται από τα προηγούμενα τρία tables.

Το osm2pgsql μετατρέπει τα nodes, ways και relations σε γεωμετρίες ως εξής:

- Εάν κάποιο node χαρακτηρίζεται από tags τότε μπαίνει στα planet_osm_point.
- Εάν κάποιο way είναι μια γραμμή μπαίνει στα planet_osm_line.
- Εάν κάποιο way αποτελεί μια γραμμή μορφής δακτυλιδιού, δηλαδή η αρχή του να συμπίπτει με το τέλος του, τότε εισάγεται στο planet_osm_polygon.
- Τέλος συνδυάζοντας τα μέλη ενός relation προκύπτει είτε μια γεωμετρία όπως πριν είτε μια multi-γεωμετρία.

Τα tables με τις γεωμετρίες (τα 3 τελευταία) έχουν την εξής δομή:

Πίνακας 3.1 planet_osm_tables' Format

osm_id	Tags' keys	ways

Το osm_id περιγράφει το id που αντιστοιχεί σε αυτήν την osm οντότητα (έναν αριθμό). Εάν πρόκειται για relation το id είναι αρνητικό.

Tags' keys είναι το πλήθος των columns με όνομα όλα τα keys που έχουν οι οντότητες των πινάκων. Κάθε οντότητα που έχει κάποιο συγκεκριμένο tag (key-value pair) συμπληρώνει στο αντίστοιχο key column με την τιμή value αυτού του tag.

Το ways column έχει τις παραγόμενες γεωμετρίες που μας παρέχει το osm2pgsql.

Πίνακας 3.2 planet_osm_table Example

osm_id	**name**	**Natural**	**Water**	**Highway**	**Way**
287382	Μαραθώνα	water	Lake	-	POLYGON(...)
287383	Εθνική οδός	-	-	Primary	LINESTRING(...)

Στο παράδειγμα αυτό η λίμνη Μαραθώνα έχει κενό στο column highway επειδή δεν έχει κάποιο tag με key highway που την χαρακτηρίζει. Βέβαια το table έχει το column highway γιατί παρακάτω η οντότητα Εθνική Οδός χαρακτηρίζεται από tag το οποίο έχει ως key το "highway" και value το "primary".

3.4.2.1 Osm2pgsql

Το εργαλείο αυτό όπως και τα προηγούμενα ανήκει επίσης στην γενικότερη κατηγορία των OSM tools. Η βασική δουλεία που επιτελεί το osm2pgsql είναι να εισάγει γεωχωρικά δεδομένα από το OSM σε μια PostgreSQL/ PostGIS βάση δεδομένων. Έχει την δυνατότητα επίσης να διαχειρίζεται και PBF format αρχεία.

Τεχνικά χαρακτηριστικά

Τι μπορεί να κάνει το osm2pgsql;

- Μετατρέπει το OSM αρχείο σε μια PostgreSQL βάση δεδομένων

- Μετατροπή των OSM tags σε στήλες στην βάση δεδομένων που μπορούν να διαμορφωθούν από το style αρχείο.
- Ικανό να διαβάσει .gz, .bz2, .pbf και .o5m αρχεία
- Μπορεί να εμπλουτίσει με αλλαγές την βάση δεδομένων ώστε να κρατείται ενήμερη.

Λειτουργία του Osm2pgsql:

1. Το osm2pgsql συνδέεται στην βάση δεδομένων και δημιουργεί τα ακόλουθα 4 tables: planet_osm_point, planet_osm_line, planet_osm_roads and planet_osm_polygon. Στην περίπτωση που χρησιμοποιείται η --slim λειτουργία, θα δημιουργηθούν τρία ακόμα tables: planet_osm_nodes, planet_osm_ways and planet_osm_rels.
2. Τρέχει έναν parser στο αρχείο εισόδου και επεξεργάζεται τα nodes, ways και relations.
3. Εάν ένα node έχει κάποιο tag που εμπεριέχεται στο style αρχείο τότε προστίθεται σοτ planet_osm_point. Εάν δεν υπάρχει το tag τότε απλά κρατιούνται οι συντεταγμένες του.
4. Τα ways μετατρέπονται σε Well Known Text (WKT) γεωμετρίες χρησιμοποιώντας τις συντεταγμένες των nodes που διαβάστηκαν προηγουμένως. Το WKT αποτελεί ουσιαστικά έναν τρόπο αναπαράστασης γεωμετριών ως αλφαριθμητικά δεδομένα (string).
5. Εάν ένα way έχει κάποιο tag στο style αρχείο δηλωμένο ως "polygon" και σχηματίζει ένα κλειστό δακτυλίδι τότε προστίθεται στο planet_osm_polygon table.
6. Έπειτα τα relations προσπελαύνονται. Το osm2pgsql έχει μια ειδική μεταχείριση για αυτές και διαμορφώνει την κατάλληλη γεωμετρία με βάση τα μέλη της.
7. Τέλος γίνεται η χρήση ευρετηρίων για την αύξηση της ταχύτητας των queries.

Μερικά από τα modes που μπορεί να υποστηρίξει το osm2pgsql:

-a,--append

Προσθέτει το OSM αρχείο χωρίς να αφαιρέσει τα ήδη υπάρχοντα δεδομένα. Με αυτόν τον τρόπο θα μπορούμε να ενημερώσουμε την βάση με την χρήση diff αρχείων. (**used in Update state**)

-c,--create

Αντικαθιστά την ήδη υπάρχουσα πληροφορία στην βάση δεδομένων. (**used in Initialization state**)

-s, --slim

Αποθηεύει προσωρινά δεδομένα στην βάση. Στην περίπτωση που δεν χρησιμοποιηθεί αυτή η λειτουργία όλη η προσωρινή μνήμη θα φορτωθεί στην RAM. Με το slim mode μπορούμε να φορτώσουμε τα δεδομένα στο σύστημα ακόμα και με περιορισμένη μνήμη. **Αυτή η λειτουργία είναι απαραίτητη στην περίπτωση που θέλουμε μελλοντικές ενημερώσεις με --append**. (**used**)

--drop

Διαγράφει τα tables που δημιουργούνται από τη slim λειτουργία μόλις η διαδικασία του osm2pgsql ολοκληρωθεί. Στην περίπτωση που χρειάζεται να ενημερώσουμε την βάση θα πρέπει να διατηρήσουμε την ύπαρξη των tables, καθώς αυτά χρειάζονται για επεξεργασία των diff files.(**not used**)

-S,--style /path/to/style

Ορίζει το style αρχείο που θα χρησιμοποιηθεί. Αυτό το αρχείο ορίζει τα tags που θα οριστούν ως columns στο data base και ποια από αυτά θα απορριφθούν. Defaults to /usr/share/osm2pgsql/default.style. (**used**)

-l,--latlong

Αποθηκεύει τα δεδομένα σε όρους latitude longitude. (**used**)

Performance options

-C,--cache num

Μόνο για τη slim λειτουργία διευκρινίζει το μέγεθος της RAM για την αποθήκευση nodes. Όσο περισσότερα nodes αποθηκεύονται στην cache τόσο πιο γρήγορα θα γίνει η εισαγωγή.(**used 75% of total RAM**)

--number-processes num

Διευκρινίζει το πλήθος των παράλληλων διεργασιών που χρησιμοποιούνται για διάφορες λειτουργίες. (**used (# of cores / 2) - 1**)

3.4.2.2 Το φιλτράρισμα

Το YAGO2geo το οποίο και θέλουμε να ενημερώσουμε χρησιμοποιώντας πληροφορία προερχόμενη από το OSM, έχει συγκεκριμένη οντολογία. Τα δεδομένα του αποτελούνται μόνο από συγκεκριμένες κατηγορίες. Αυτές είναι το osm_bay, osm_beach, osm_canal, osm_city, osm_forest, osm_island, osm_lagoon, osm_lake, osm_locality, osm_nature_reserve, osm_oxbow, osm_park, osm_resevoir, osm_stream, osm_town, osm_village.

Το OSM παρέχει ένα σύνολο από tags που περιγράφουν και δίνουν πληροφορίες σχετικά με τις οντότητες, όπως είναι το όνομα. Επιπλέον συνδυασμοί διάφορων tags δηλώνουν την ευρύτερη κατηγορία στην οποία ανήκει η κάθε οντότητα. Για παράδειγμα εάν μια οντότητα έχει τα tags natural:water και water:lake τότε ορίζεται πως αυτή η οντότητα αποτελεί Lake.

Από τις δύο παραπάνω παραγράφους μπορούμε να καταλήξουμε σε ένα σύνολο συνδυασμών από tags του OSM για να περιγράψουμε μόνο εκείνες τις κατηγορίες που το YAGO2geo χρειάζεται.

Πίνακας 3.3 General Category-OSM Tags

Category	***Tags***
osm_bay	natural:bay
osm_beach	natural:beach
osm_canal	waterway:canal
osm_city	place:city
osm_forest	natural:wood | landuse:fores
osm_island	place:island
osm_lagoon	natural:water & water:lagoon
osm_lake	natural:water & water:lake
osm_locality	place:locality
osm_nature_reserve	leisure:nature_reserve
osm_oxbow	natural:water & water:oxbow
osm_park	leisure:park
osm_resevoir	landuse:reservoir | natural:water & water:reservoir
osm_stream	waterway:stream
osm_town	place:town
osm_village	place:village

Το OSM ως μια πλούσια και crowdsourced πηγή δεδομένων μας προσφέρει πληθώρα πληροφοριών χωρίς όμως όλες να μας είναι απαραίτητες. Έτσι δημιουργείται η ανάγκη να φιλτράρουμε τα tags ώστε να επιλεχθούν μόνο τα επιθυμητά δεδομένα.

Το εργαλείο osm2pgsql μπορεί να φιλτράρει τα tags keys columns των geometry tables με την χρήση του --style flag. Μετά το --style μπορεί κάποιος να παραθέσει ένα αρχείο της μορφής file.style. Σε αυτό το αρχείο ορίζονται ποια tags' keys είναι χρήσιμα και ποια όχι . Με αυτόν τον τρόπο δεν μειώνεται μόνο το πλήθος των columns απορρίπτοντας αυτά που δεν μας χρειάζονται- αλλά και το γενικότερο πλήθος από τις οντότητες, καθώς εάν μία οντότητα έχει tags' keys μόνο από αυτά που έχουν απορριφθεί τότε και η ίδια η οντότητα δεν εισέρχεται στα tables με τις γεωμετρίες.

Σαν αποτέλεσμα αυτής της περικοπής για ένα μικρό αρχείο αυτό της Ελλάδας (200MB):

- Εάν το εργαλείο χρησιμοποιηθεί χωρίς το style τότε θα παραχθούν ~2M γραμμές και πολύγωνα.
- Εάν χρησιμοποιηθεί το --style τότε θα παραχθούν ~450K από τα προηγούμενα.

osm_all_data table

Στα tables planet_osm_point, planet_osm_line και planet_osm_polygon της βάσης εμπεριέχεται οι γεωμετρίες των osm δεδομένων με συγκεκριμένα tags' keys. Επειδή το YAGO2geo, στην περίπτωση των OSM δεδομένων, νοιάζεται μόνο για τις γεωμετρίες δεν μας ενδιαφέρει το περιεχόμενο του planet_osm_points. Συγκεντρώνουμε την χρήσιμη πληροφορία των 2 tables lines και polygons σε ένα table που ονομάζεται osm_all_data. Η παραπάνω στρατηγική ακολουθείται για τους εξής λόγους:

- Ο κυριότερος λόγος είναι για να φιλτραριστούν τα δεδομένα ακόμα περισσότερο. Χρησιμοποιώντας το style του osm2pgsql απορρίπτονται όλα τα tags' keys τα οποία δεν χρειάζονται. Όμως το YAGO2geo χρειάζεται συγκεκριμένα pairs of keys and **values.** Για παράδειγμα εάν είναι απαραίτητο το tag natural:water μπορεί να κρατήσουμε το key "natural" αλλά υπάρχουν και οντότητες με άσχετα από τα αναγκαία values. Π.χ. το key "natural" μπορεί να έχει ως value "rock" το οποίο δεν εμπεριέχεται στα αναγκαία tag που χρειάζονται οι συγκεκριμένες κατηγορίες του YAGO2geo (πίνακας 3.3).
- Ο δεύτερος λόγος είναι ότι το osm_all_data table λειτουργεί ως cache. Κρατάει όλα τα entities και μόλις ένα update έρθει τότε μαρκάρει μόνο εκείνα τα rows που έχουν αλλαγές. Αυτή η λειτουργία γίνεται με την χρήση ενός column ονόματι "flag". Εάν μια οντότητα έχει αλλάξει ή είναι η πρώτη φορά που εισέρχεται στο table τότε παίρνει την τιμή 1 στο column flag, αλλιώς θα πάρει την τιμή 0. Έπειτα όταν έρθει η ώρα του ενημέρωσης θα λαμβάνονται υπόψη μόνο αυτά που έχουν flag=1.
- Ο τρίτος λόγος που γίνεται η συγκέντρωση της πληροφορίας στο osm_all_data είναι επειδή τα osm id με τα id του YAGO2geo δεν έχουν την ίδια μορφή. Τα osm id αποτελούνται από έναν αριθμό, ο οποίος όταν η οντότητα προέρχεται από ένα relation τότε ο αριθμός αυτός είναι αρνητικός. Τα YAGO2geo id έχουν την μορφή node/id_num, way/id_num και relation/id_num. Γι' αυτό προκειμένου να γίνει αργότερα το interlink θα πρέπει να ταιριάξουν τα 2 id format.

Σαν αποτέλεσμα από αυτό, από τις 450K οντότητες των γραμμών και των πολύγονων μένουν μόνο 45K οντότητες στο osm_all_data.

3.4.3 Update- Προετοιμασία (Setup)

Κατά το Update setup ακολουθούνται τα παρακάτω βήματα.

- Κατεβάζουμε μόνο τις πιο πρόσφατες αλλαγές των OSM δεδομένων.
- Χρησιμοποιώντας τις καινούριες αυτές αλλαγές ενημερώνονται τα OSM δεδομένα που έχουν κατέβει από το Initialize-setup.

Κατέβασμα των πιο πρόσφατων OSM δεδομένων

Για να κατεβάσουμε τις πιο πρόσφατες αλλαγές των OSM δεδομένων θα γίνει η χρήση του εργαλείου **osmupdate.** Το εργαλείο αυτό ανήκει στην ευρύτερη οικογένεια των OSM εργαλείων η οποία αποτελεί ένα σύνολο από εργαλεία με τα οποία μπορεί κάποιος να διαχειριστεί και να επεξεργαστεί OSM δεδομένα.

3.4.3.1 Osmupdate

Το εργαλείο αυτό κατεβάζει και αθροίζει τα OSM Changefiles διάφορων κατηγοριών (minutely, hourly, daily). Τα OSM changefiles ή diff files είναι οι αλλαγές που έχουν υποστεί τα OSM δεδομένα κατά μια συγκεκριμένη χρονική περίοδο. Τέτοια files προσφέρονται και από τον Geofabrik server και δίνουν την δυνατότητα να μην χρειαστεί να κατεβάσει κάποιος όλο το αρχείο εκ νέου, στην περίπτωση που θέλει την πιο πρόσφατη εκδοχή του, αλλά να κατεβάσει μόνο τις αλλαγές αυτού.

Η όλη διαδικασία στηρίζεται σε ένα χαρακτηριστικό το οποίο εμπεριέχεται μέσα σε αυτά τα αρχεία και ονομάζεται timestamp. Το timestamp κάθε OSM αρχείου είναι μια χρονική σφραγίδα που περιγράφει μια συγκεκριμένη ημερομηνία και ώρα. Αυτή η ημερομηνία-ώρα δηλώνει τις έως και ποιες αλλαγές εμπεριέχει το αρχείο. Έχοντας αυτήν την πληροφορία το εργαλείο θα κατεβάσει τις μεταγενέστερες από το timestamp πληροφορίες και με αυτές μπορεί είτε να ενημερώσει το ίδιο το OSM αρχείο είτε να συγκεντρώσει όλες τις αλλαγές σε ένα συμπιεσμένο αρχείο. Ακολουθείται η 2η εκδοχή.

Τεχνικά χαρακτηριστικά

Το osmupdate παίρνει σαν όρισμα το osm.pbf αρχείο που έχουμε και αναγνωρίζει το timestamp του. Υπάρχει η δυνατότητα να οριστεί το επιθυμητό timestamp, αλλά στην περίπτωσή μας δεν χρειάζεται. Έπειτα κατεβάζει όσες αλλαγές έχουν συμβεί και τις συμπιέζει σε έναν φάκελο τον **changes.osc.gz**.

Τέλος αυτό που θα έπρεπε να σημειωθεί είναι η προέλευση αυτών των αρχείων. Εάν πρόκειται για κάποιο αρχείο το οποίο δεν αποτελεί όλον τον κόσμο, η πηγή την οποία θα συμβουλευτούμε είναι ο Geofabrik server. Αυτός προσφέρει τα δεδομένα του κόσμου τμηματοποιημένα. Εάν χρησιμοποιηθεί το osmupdate για ολόκληρο τον κόσμο τότε θα πρέπει να συμβουλευτούμε τον σύνδεσμο[1] από όπου προσφέρεται όλα τα diff files και σε εκδοχές με διαφορετικές χρονικές περιόδους. Για παράδειγμα υπάρχουν τα changefiles ανά λεπτό, ανά ώρα καθώς και τα changefiles ανά μέρα. Οι διάφορες κατηγορίες υπάρχουν για να δώσουν την δυνατότητα σε εφαρμογές να είναι όσο το δυνατόν περισσότερο σύγχρονες με τις αλλαγές της πηγής την ώρα που γίνονται.

Το osmupdate δίνει αυτήν την δυνατότητα για την επιλογή της πηγής από όπου θα κατέβουν τα diff files μέσω του flag --base-url. Εκεί ορίζεται ο σύνδεσμος που θα χρησιμοποιήσει το εργαλείο για να κατεβάσει τα αρχεία. Σαν default ορίζεται το planet.openstreetmap.org και σε αυτήν την περίπτωση μπορεί κάποιος να διαλέξει τί χρονικής- κατηγορίας θα είναι τα αρχεία χρησιμοποιώντας τα flags --day --hour –minute αντίστοιχα.

Το Geofabrik δεν υποστηρίζει την τελευταία λειτουργικότητα. Τα διάφορα τμηματοποιημένα diff files έχουν μόνο ανά μέρα εκδοχές. Επίσης με την επιλογή --keep-tempfiles κρατάμε τα diff files που έχει κατεβάσει το osmupdate(μετά την εκτέλεση διαγράφει τα diff files) και εάν χρειαστεί να ξανακατέβει κάποιο από αυτά θα δει πρώτα εάν υπάρχει σε αυτήν την cache ώστε να αποφύγει την διαδικασία.

3.4.3.2 Ενημέρωση του planet.osm.pbf αρχείου

Για να ενημερωθεί ένα αρχείο outdated με τις επιθυμητές αλλαγές που έχουν κατέβει, θα γίνει η χρήση ενός εργαλείου το οποίο θα προέρχεται και αυτό από την OSM tools οικογένεια

[1]https://planet.openstreetmap.org/

το **osmconvert**. Αυτό το εργαλείο χρησιμοποιείται και εσωτερικά από το osmupdate στην περίπτωση που θέλουμε να ενσωματώσουμε τις αλλαγές που κατεβάζει κατευθείαν στο .osm.pbf file. Γενικά το εργαλείο αυτό χρησιμοποιείται για την μετατροπή και την επεξεργασία των OpenStreetMap files. Παρόμοιο εργαλείο με τον ίδιο σκοπό και μεγαλύτερη λειτουργικότητα είναι το osmosis. Ο λόγος για τον οποίο κατέληξα στην χρήση του osmconvert είναι πως εκτός από το να ενσωματώσει τα updates στο OSM αρχείο, αλλάζει αυτόματα και το timestamp του αρχείου με βάση και τα τελευταία updates που δέχτηκε. Έτσι με αυτόν τον τρόπο την επόμενη φορά που θα γίνει update το osmupdate θα χρησιμοποιήσει το ανανεωμένο timestamp και δεν θα χρειαστεί να κατεβάσει και να ενσωματώσει παλιότερη πληροφορία στο changes.osc.gz.

Το μειονέκτημα που έχει το osmconvert είναι πως δεν μπορεί να ενημερώσει το ήδη υπάρχον αρχείο. Στην περίπτωση που το output είναι το ίδιο OSM αρχείο με το input τότε απλώς θα διαγράψει το input. Θα χρειαστεί η δημιουργία ενός καινούριου αρχείου στο οποίο θα συνδυαστούν οι αλλαγές και η OSM πληροφορία. Όταν τελειώσει η όλη διαδικασία τότε το νέο ανανεωμένο αρχείο θα κάνει overwrite το παλιότερο.

3.4.4 Update-Ανέβασμα των OSM δεδομένων στην PostgreSQL

Κατά το update state χρειάζεται να δοθούν μόνο οι αλλαγές (changes.osc.gz) στην βάση δεδομένων μας.

Προκειμένου να φορτώσει τα OSM δεδομένα η update κατάσταση χρησιμοποιεί και αυτή το εργαλείο **osm2pgsql**. Το εργαλείο αυτό έχει την δυνατότητα πέρα από osm πληροφορία να δέχεται και diff files χρησιμοποιώντας την --append λειτουργία. Αφού δεχτεί κάποιο diff file τότε εφαρμόζει τις αλλαγές που περιγράφει στα planet_osm_nodes/ways/rels tables και έπειτα μεταφέρει τις αλλαγές αυτές και στα tables με τις γεωμετρίες. Μετά από αυτό η πληροφορία περνάει πάλι στο osm_all_data table και ελέγχονται μόνο οι αλλαγές. Γενικά η ίδια λογική που ακολουθείται στο φιλτράρισμα από το *Initialization-Ανέβασμα των OSM δεδομένων στην PostgreSQL* ακολουθείται και εδώ.

3.4.5 Ενημέρωση της yago_all_data πληροφορίας

Η επιθυμητή OSM και YAGO2geo πληροφορία έχει εισαχθεί στην βάση δεδομένων στα osm_all_data και yago_all_data tables αντίστοιχα. Κάθε οντότητα από αυτά τα tables διακρίνεται από ένα ξεχωριστό id της μορφής (way|relation)/*num*.Για την διαδικασία του update θα εκτελεστεί ένα query το οποίο θα κάνει join τα δυο αυτά tables με βάση το προηγούμενο id. Μόνο οι OSM οντότητες που έχουν κοινό id με κάποια οντότητα από το yago_all_data table θα ληφθούν υπόψη. Με αυτόν τον τρόπο δεν μας ενδιαφέρουν οι οντότητες των οποίων το id δεν υπάρχει στο YAGO2geo, καθώς σκοπός μας είναι αν ενημερώσουμε την ήδη υπάρχουσα. Επιπλέον στο osm_all_data εμπεριέχεται το column flag. Αυτό το flag ξεχωρίζει τις ενημερωμένες οντότητες(flag=1) από τις μη(flag=0).

Οπότε join μεταξύ των δύο tables θα γίνει μεταξύ **μόνο** των αλλαγμένων OSM οντοτήτων και των αντίστοιχων YAGO2geo οντοτήτων με βάση πάντα το id. Η νέα πληροφορία θα εισαχθεί στο table του yago_all_data στο updated_*column* που της αντιστοιχεί(πχ εάν για μια οντότητα ενημερωθεί η γεωμετρία τότε η νέα πληροφορία θα εισαχθεί στο column updated_geos). Αυτό γίνεται προκειμένου να κρατηθεί η παλιά και νέα πληροφορία ώστε να συγκριθούν. Όταν ολοκληρωθεί η διαδικασία τότε θα καλείται ένα query το οποίο θα μεταφέρει την ενημερωμένη πληροφορία στην αντίστοιχη στήλη και θα μηδενίζει το

updated_*column*. Η όλη διαδικασία γίνεται μέσω ενός μόνο query εντός της βάσης ως join μεταξύ δύο tables και γι' αυτόν τον λόγο μπορεί να χαρακτηριστεί ως βελτιστοποιημένη. Όπως αναφέρθηκε στην αρχή δεν προστίθενται νέες οντότητες που δεν υπάρχουν στο yago_all_data ούτε διαγράφονται υπάρχουσες. Αυτό που γίνεται είναι η ενημέρωση των υπαρχόντων YAGO2geo δεδομένων.

3.4.6 Ενημέρωση του YAGO2geo Γνωσιακού Γράφου

Πλέον έχουμε περάσει όλες τις αλλαγές στο yago_all_data table και ακολουθεί η αναβάθμιση του KB. Η καινούρια πληροφορία διατηρείται στα *columns*_updated ενώ η παλιά στα αντίστοιχα *columns* του table (name-name_updated, category-category_updated, geom_prev-geom_updated). Τα *columns*_ updated δεν έχουν τιμή για κάθε οντότητα μόνο για αυτές που έχουν υποστεί κάποια αλλαγή από την πλευρά του OSM. Το link υποδηλώνει το Subject που το YAGO2geo αντιστοιχεί στην κάθε οντότητα (πχ Στην εικόνα 3.1 το subject της οντότητας αυτής είναι το geoentity_Garden_Avenue_Park_7306206). Αποθηκεύοντας και διατηρώντας το αναγνωριστικό αυτό μπορούμε να γνωρίζουμε ακριβώς σε ποια YAGO2geo οντότητα αναφερόμαστε. Επιπλέον η γεωμετρία στο RDF schema αποθηκεύεται με βάση την οντολογία της geoSPARQL, δηλαδή ως εξής:

subject hasGeometry geo_id

geo_id asWKT Polygon(…)

Από το παραπάνω παρατηρούμε πως η πραγματική πληροφορία της γεωμετρίας περιγράφεται από το δεύτερο triple. Εάν η πληροφορία της γεωμετρίας αλλάξει τότε θα πρέπει απλά να ενημερώσουμε το object του δεύτερου triple. Για να έχουμε πρόσβαση στο δεύτερο triple αποθηκεύεται το *geo_id* στο column *geometry* του yago_all_data table για την κάθε οντότητα. Έτσι στην περίπτωση που θέλουμε να γίνει κάποια αλλαγή στις γεωμετρίες μπορούμε να αναφερθούμε απευθείας στο δεύτερο triple

Προκειμένου να γίνει το update στα RDF triples του KB θα πρέπει να χρησιμοποιήσουμε την SPARQL query language. Στέλνοντας τα κατάλληλα queries στο KB θα μπορέσουμε να επεξεργαστούμε τα RDF δεδομένα και εν τέλει να εντάξουμε τις αλλαγές που έχουμε διακρίνει. Τα queries αυτά θα χωρίζονται σε 3 σκέλη:

- Το DELETE όπου διαγράφεται το triple που περιγράφει την παλιά πληροφορία
- Το INSERT όπου εισάγεται στο KB η ενημερωμένη πληροφορία υπό μορφή triple
- Το WHERE όπου περιγράφονται τα subjects στα οποία θα εφαρμοστούν τα DELETE και INSERT

Για να δεχτεί ο Γνωσιακός Γράφος τα queries στήνεται σε ένα SPARQL endpoint με την χρήση της Strabon[18] εφαρμογής. Αφού το KB είναι up τότε μπορούμε να του στείλουμε τα update queries που επιθυμούμε ως ένα Post request και αυτός να τα εκτελέσει.

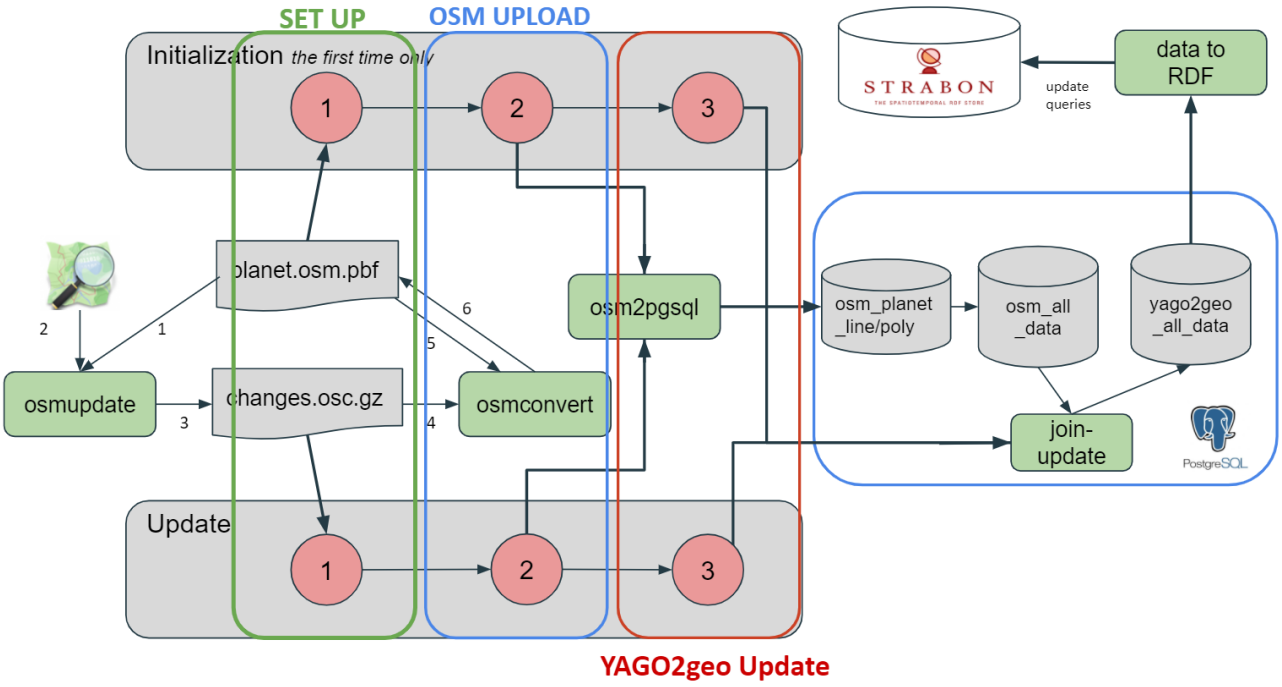
```
PREFIX ykr: <http://yago-knowledge.org/resource/>
PREFIX geosp:<http://www.opengis.net/ont/geosparql#>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX yrec:<http://kr.di.uoa.gr/yago2geo/resource/>
PREFIX yont: <http://kr.di.uoa.gr/yago2geo/ontology/>

DELETE {yrec:Geometry_osm_558b9c5c-85d9-346a-b996-4137acfa7ecd geosp:asWKT ?t.
yrec:Geometry_osm_8d1ed6a5-078f-37ba-9a77-7289b876adf7 geosp:asWKT ?t.
<http://yago-knowledge.org/resource/National_Garden,_Athens> rdf:type yont:OSM_park.
yrec:Geometry_osm_c1d06a84-11b0-3b50-b376-930d8d662daf geosp:asWKT ?t.
}
INSERT {yrec:Geometry_osm_558b9c5c-85d9-346a-b996-4137acfa7ecd geosp:asWKT
      "<http://www.opengis.net/def/crs/EPSG/0/4326> POLYGON((...))"^^<http://www.opengis.net/ont/geosparql#wktLiteral>.
yrec:Geometry_osm_8d1ed6a5-078f-37ba-9a77-7289b876adf7 geosp:asWKT
      "<http://www.opengis.net/def/crs/EPSG/0/4326> POLYGON((...))"^^<http://www.opengis.net/ont/geosparql#wktLiteral>.
<http://yago-knowledge.org/resource/National_Garden,_Athens> rdf:type yont:OSM_forest.
yrec:Geometry_osm_c1d06a84-11b0-3b50-b376-930d8d662daf geosp:asWKT
      "<http://www.opengis.net/def/crs/EPSG/0/4326> POLYGON((...))"^^<http://www.opengis.net/ont/geosparql#wktLiteral>.
}
WHERE { ?s ?p ?o.
?o geosp:asWKT ?t.
FILTER( ?s= ykr:geoentity_Alsos_Pagkratiou_10345875 || ?s= ykr:geoentity_Parko_Iroon_10345877 ||
    ?s= <http://yago-knowledge.org/resource/National_Garden,_Athens>)
}
```

3.3 Update Sparql Query

Στο query του παραδείγματος γίνονται update 3 γεωμετρίες από 3 διαφορετικές οντότητες, αλλά και η κατηγορίας μιας εξ αυτών. Για να γίνει το update στην SPARQL χρειάζεται πρώτα να διαγραφούν τα παλιά triples και έπειτα να εισαχθούν τα καινούρια. Το τελευταίο subject δεν χρησιμοποιεί prefix:prefixName γιατί εσωτερικά στο prefix name υπάρχει το “,” και η μορφή αυτή δεν είναι αποδεκτή από την SPARQL.

3.4.7 Σχηματική Απεικόνιση της Update Διαδικασίας



3.4The whole Process

Συνοψίζοντας από όλα όσα είδαμε παραπάνω η διαδικασία μπορεί διακριθεί σε 2 καταστάσεις: στο initialization όπου γίνεται την πρώτη φορά και το update που γίνεται για κάθε επόμενη εκτέλεση. Κάθε μία από αυτές τις καταστάσεις μπορεί να απλοποιηθεί σε 3 βασικά βήματα: Set up, OSM Upload, YAGO2geo Update.

- **Initialization**
  - *Set up*: Κατέβασμα των OSM δεδομένων σε μορφή osm.pbf αρχείου. Το osmupdate κατεβάζει τις ενημερώσεις του αρχείου και τις συμπιέζει στο changes.osc.gz. Έπειτα το osmconvert δέχεται ως είσοδο το changes.osc.gz και ενημερώνει με τις αλλαγές αυτές το αρχικό .osm.pbf αρχείο. Κατέβασμα των YAGO2geo δεδομένων και εισαγωγή τους στην βάση.
  - *OSM Upload*: Το ενημερωμένο αρχείο εισέρχεται ως είσοδο στο osm2pgsql ώστε να το εισάγει στην postgres βάση δεδομένων.
  - *YAGO2geo Update*: Τα φιλτραρισμένα OSM δεδομένα στο osm_all_data table κάνουν update τα YAGO2geo δεδομένα (YAGO2geo_all_data table)
- **Update**:
  - *Set up*: Κατέβασμα των αλλαγών στο changes.osc.gz και ενσωμάτωσή τους στο ήδη κατεβασμένο osm.pbf αρχείο.
  - *OSM Upload*: Με το osm2pgsql εφαρμογή των αλλαγών στην βάση.
  - *YAGO2geo Update*: Ενημέρωση του yago2geo_all_data table χρησιμοποιώντας τις αλλαγές του osm_all_data.

Το εργαλείο προορίζεται για χρήση σε εβδομαδιαία βάση ώστε να ενημερώνει το KB.Αφού οι αλλαγές περάσουν στο yago2geo_all_data table περνάνε έπειτα στο remote KB.

4 FUTURE WORK

Εν κατακλείδι, στην πτυχιακή αυτή αναπτύχθηκε ένα σύστημα το οποίο στοχεύει στην σταδιακή ενημέρωση του YAGO2geo με γεωχωρική πληροφορία προερχόμενη από το OSM. Αυτή η διαδικασία θα μπορεί να εκτελείται ανά κάποια χρονικά διαστήματα, μερών, να συλλέγει τις ενημερώσεις που χρειάζεται και έπειτα να αναβαθμίζει τον Γνωσιακό Γράφο που είναι σε κάποιο remote endpoint.

Η διαδικασία αυτή έχει στηθεί προκειμένου να γίνεται ανά κάποια χρονικά διαστήματα μερών, αλλά θα μπορούσε να επεκταθεί η χρήση της και για update ανά ώρα ή και γιατί όχι ανά λεπτό. Με αυτόν τον τρόπο η ενημέρωση των δεδομένων θα γινόταν σχετικά άμεσα και εάν κάτι άλλαζε στην πηγή, θα άλλαζε σχεδόν απευθείας-με καθυστέρηση της τάξεως του λεπτού- και από την πλευρά του YAGO2geo.

Βέβαια ως προέκταση των παραπάνω θα μπορούσε να είναι η ενημέρωση της πληροφορίας του YAGO2geo σε πραγματικό χρόνο. Δηλαδή την στιγμή που αλλάζει κάτι στην OSM πληροφορία η αλλαγή να μεταφέρεται και στο YAGO2geo. Αυτό δύναται να πραγματοποιηθεί με κάποιο Live streaming όπου τα δεδομένα ενώ κατευθύνονται για την ενημέρωση των OSM δεδομένων θα διασταυρώνονται και θα γίνεται η απόκτησή τους παράλληλα και από την πλευρά του KB. Αφού πάρουμε την αλλαγή αυτή να ενσωματώνεται με τον υπόλοιπο RDF Γράφο του YAGO2geo και να γίνεται η αντίστοιχη ενημέρωση. Αυτό θα μπορούσε να οδηγήσει στην δημιουργία ενός εργαλείου το οποίο θα διαθέτει ουσιαστικά τα δεδομένα του OSM στα πλαίσια ενός καλά σχεδιασμένου Γνωσιακού Γράφου (Knowledge Base) με ακρίβεια στη γεωχωρική πληροφορία. Βέβαια η πληροφορία αυτή δεν θα περιέχει όλα τα δεδομένα του OSM γιατί τότε θα είχαμε έναν ανεπιθύμητα μεγάλο και θορυβώδη όγκο δεδομένων. Ιδανικά θα έπρεπε να φιλτράρονται με γνώμονα την πρότυπη διαδικασία του YAGO εξασφαλίζοντας έτσι τη διατήρηση της ποιότητάς τους. Αυτό πρακτικά σημαίνει την απόρριψη των αλλαγών σε οντότητες που δεν μας ενδιαφέρουν.

Μια άλλη προέκταση του εργαλείου θα μπορούσε να είναι η εισαγωγή νέων και διαγραφή των παλιών δεδομένων του YAGO2geo, καθώς τώρα η λειτουργία του εργαλείου περιορίζεται στην ενημέρωσή τους. Για να επιτευχθεί η εισαγωγή νέων οντοτήτων είναι απαραίτητο να ακολουθηθεί ο ίδιος έλεγχος ο οποίος διεξήχθη για τα δεδομένα που εισήχθησαν στο YAGO2geo εξ αρχής[11]. Αλλιώς θα υποβαθμιστεί η ποιότητα των δεδομένων εάν είναι να εισέρχεται κάθε στοιχείο που συναντάται.

Τέλος το YAGO2geo συγκεντρώνει την προσοχή του σε συγκεκριμένες κατηγορίες που εμπεριέχουν όσο το δυνατόν πιο σταθερές γεωμετρίες (λίμνες, ποτάμια). Υπάρχει η δυνατότητα να γίνει επέκταση αυτών των κατηγοριών ώστε να διαθέτουμε μια πολύ μεγαλύτερη ποικιλία δεδομένων. Θα ήταν δυνητικά εφικτό να γίνει για παράδειγμα η ένταξη οδών σε αυτόν ώστε να εμπλουτιστεί ακόμα περισσότερο ο Γράφος.

ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ

RDF	Resource Description Framework
OWL	Web Ontology Language
W3C	World Wide Web Consortium
SPARQL	SPARQL Protocol and RDF Query
YAGO	Yet Another Great Ontology
OSM	OpenStreetMap
LGD	LinkedGeoData
KB	Knowledge Base
PBF	Protocolbuffer Binary Format
ETL	Extract-Transform-Load
URI	Uniform Resource Identifier
Turtle	Terse RDF Triple Language
SQL	Structured Query Language
WKT	Well Known Text

ΑΝΑΦΟΡΕΣ

[1] TIM BERNERS-LEE, JAMES HENDLER and ORA LASSILA, 'THE SEMANTIC WEB', Scientific American,Vol. 284, No. 5, MAY 2001, pp. 34-43.

[2] Paulheim, Heiko, Journal: Semantic Web, vol. 8, no. 3, pp. 489-508, 2017

[3] A. Jentzsch, 'Linked Open Data Cloud', in X.media.press, 2014, pp. 209–219.

[4] F. Erxleben, M. Günther, M. Krötzsch, J. Mendez, and D. Vrandečić, 'Introducing Wikidata to the LinkedData Web', in Lecture Notes in Computer Science, 2014, pp. 50–65.

[5] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives, DBpedia: a nucleus for a web of open data, in: Proceedings of the 6th International Semantic Web Conference, 2007

[6] T. Rebele, F. Suchanek, J. Hoffart, J. Biega, E. Kuzey, and G. Weikum, 'YAGO: A Multilingual Knowledge Base from Wikipedia, Wordnet, and Geonames', in Lecture Notes in Computer Science, 2016, pp. 177-185.

[7] Johannes Hoffart et al. "YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia". In: Artificial Intelligence 194 (2013), pp. 28–61.

[8] T. Pellissier Tanon, G. Weikum, F. Suchanek, 'YAGO 4: A Reason-able Knowledge Base ', in Lecture Notes in Computer Science, 2020, pp. 583-596.

[9] Corby O., Dieng R., Hébert C. (2000) A Conceptual Graph Model for W3C Resource Description Framework. In: Ganter B., Mineau G.W. (eds) Conceptual Structures: Logical, Linguistic, and Computational Issues. ICCS 2000. Lecture Notes in Computer Science, vol 1867. Springer, Berlin, Heidelberg.

[10] M. Haklay and P. Weber, "OpenStreetMap: User-Generated Street Maps," in IEEE Pervasive Computing, vol. 7, no. 4, pp. 12-18, Oct.-Dec. 2008, doi: 10.1109/MPRV.2008.80.

[11] Nikolaos Karalis, Georgios Mandilaras, and Manolis Koubarakis. "Extending the YAGO2 Knowledge Graph with Precise Geospatial Knowledge". In: International Semantic Web Conference. Springer. 2019, pp. 181–197.

[12] Manolis Koubarakis and Kostis Kyzirakos. "Modeling and querying metadata in the semantic sensor web: The model stRDF and the query language stSPARQL". In: Extended Semantic Web Conference. Springer. 2010, pp. 425–439.

[13] Matthew Perry and John Herring. "OGC GeoSPARQL-A geographic query language for RDF data". In: OGC implementation standard 40 (2012).

[14] Hellmann S., Stadler C., Lehmann J., Auer S. (2009) DBpedia Live Extraction. In: Meersman R., Dillon T., Herrero P. (eds) On the Move to Meaningful Internet Systems: OTM 2009. OTM 2009. Lecture Notes in Computer Science, vol 5871. Springer, Berlin, Heidelberg.

[15] Auer S., Lehmann J., Hellmann S. (2009) LinkedGeoData: Adding a Spatial Dimension to the Web of Data. In: Bernstein A. et al. (eds) The Semantic Web - ISWC 2009. ISWC 2009. Lecture Notes in Computer Science, vol 5823. Springer, Berlin, Heidelberg.

[16] Stadler, Claus et al. 'LinkedGeoData: A Core for a Web of Spatial Open Data'. 1 Jan. 2012 : 333 – 354.

[17] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, Sebastian Hellmann, DBpedia - A crystallization point for the Web of Data, Journal of Web Semantics, Volume 7, Issue 3, 2009, Pages 154-165.

[18] Kyzirakos K. et al. (2013) The Spatiotemporal RDF Store Strabon. In: Nascimento M.A. et al. (eds) Advances in Spatial and Temporal Databases. SSTD 2013. Lecture Notes in Computer Science, vol 8098. Springer, Berlin, Heidelberg.

[19] Mankovski S. (2016) W3C. In: Liu L., Özsu M. (eds) Encyclopedia of Database Systems. Springer, New York, NY.

[20] Polleres A. (2014) SPARQL. In: Alhajj R., Rokne J. (eds) Encyclopedia of Social Network Analysis and Mining. Springer, New York, NY.