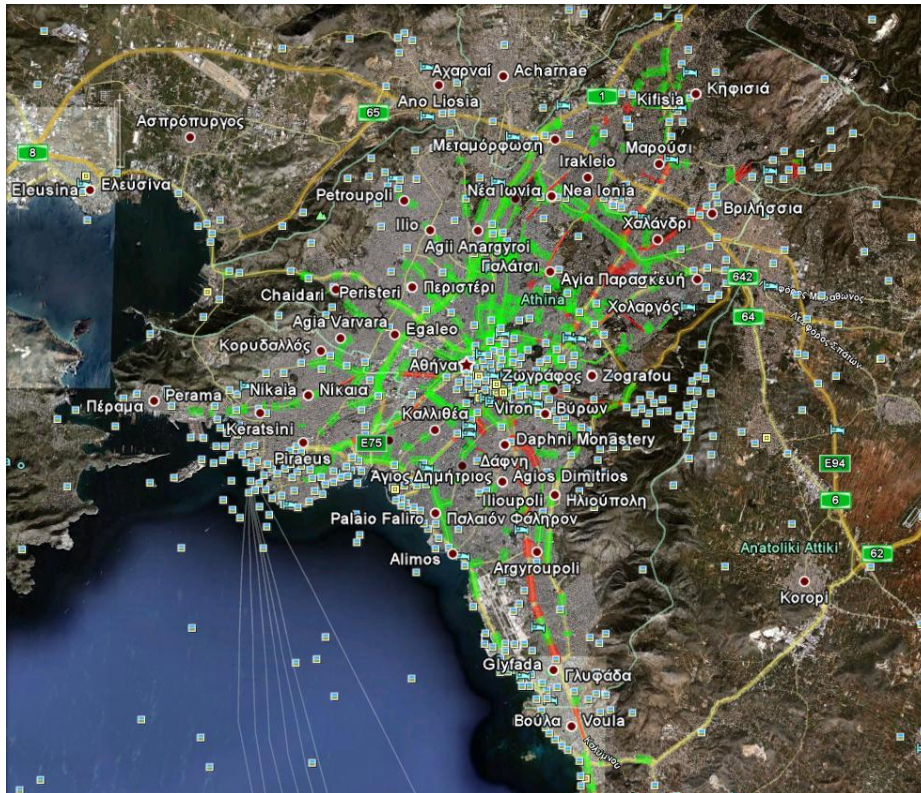




ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
Σχολή Αγρονόμων & Τοπογράφων Μηχανικών
ΓΕΩΠΛΗΡΟΦΟΡΙΚΗ
ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ

Εποπτεία Κυκλοφοριακού Φόρτου σε Πραγματικό Χρόνο



Κυριάκος Γ. Γεωργούσης

ΑΘΗΝΑ
ΙΟΥΛΙΟΣ 2010

**Εποπτεία Κυκλοφοριακού Φόρτου
σε Πραγματικό Χρόνο**



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

Σχολή Αγρονόμων & Τοπογράφων Μηχανικών

**ΓΕΩΠΛΗΡΟΦΟΡΙΚΗ
ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ**

Εποπτεία Κυκλοφοριακού Φόρτου σε Πραγματικό Χρόνο

Κυριάκος Γ. Γεωργούσης

*Μεταπτυχιακή διπλωματική εργασία
Δ.Π.Μ.Σ. «Γεωπληροφορική», 2008 - 2010*

Επιβλέπων: Καθηγητής Τίμος Σελλής

**ΑΘΗΝΑ
ΙΟΥΛΙΟΣ 2010**

Τίτλος εργασίας: ΕΠΟΠΤΕΙΑ ΚΥΚΛΟΦΟΡΙΑΚΟΥ ΦΟΡΤΟΥ
ΣΕ ΠΡΑΓΜΑΤΙΚΟ ΧΡΟΝΟ

Εθνικό Μετσόβιο Πολυτεχνείο
Διατμηματικό Πρόγραμμα Μεταπτυχιακών Σπουδών
«Γεωπληροφορική», ακαδ. έτος 2008-2010

Επιβλέπων: Καθηγητής Τίμος Σελλής
(Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών)

Κ. Γ. Γεωργούσης, 2009-2010
Ηλεκτρονικό ταχυδρομείο: kerkgeo@gmail.com

Επιτρέπεται χωρίς περιορισμούς η αναπαραγωγή του συνόλου ή μέρους αυτής της εργασίας σε έντυπη, ηλεκτρονική ή οποιαδήποτε άλλη μορφή, αποκλειστικά και μόνο για εκπαιδευτική χρήση, υπό την προϋπόθεση ότι τα αντίγραφα δεν θα διακινηθούν για οποιασδήποτε μορφής άμεσο ή έμμεσο εμπορικό όφελος και θα αναφέρονται η πηγή, ο τίτλος και η ημερομηνία της παρούσας έκδοσης.

στους γονείς μου

Ευχαριστίες

Με την ολοκλήρωση της μεταπτυχιακής διπλωματικής εργασίας μου, θα ήθελα να ευχαριστήσω ιδιαίτερα τον κ. Κώστα Πατρούμπα, για την πολύτιμη βοήθεια που μου πρόσφερε, για όλο τον πολύτιμο χρόνο του που μου αφιέρωσε και για την χρήσιμη καθοδήγηση αλλά και τις υποδείξεις του καθ' όλη την διάρκεια της εργασίας και της συγγραφής της. Ένα μεγάλο ευχαριστώ στον καθηγητή μου κ. Τίμο Σελλή, για την εμπιστοσύνη του στην ανάθεση της μεταπτυχιακής εργασίας. Επίσης θα ήθελα να ευχαριστήσω και τον συμφοιτητή μου, Σπύρο Πουράνη, τον οποίο ταλαιπώρησα ζητώντας του βοήθεια κατά την διάρκεια των μετακινήσεων και τον οποίο ευχαριστώ εγκάρδια.

Ακόμη θα ήθελα να ευχαριστήσω θερμά όλους τους καθηγητές μου για τον εμπλουτισμό των γνώσεων μου και για την καθοδήγηση τους. Στο σημείο αυτό θα ήθελα επίσης να ευχαριστήσω βαθύτατα όλη την οικογένεια μου, αλλά και κάθε έναν ξεχωριστά, γιατί εκτός από αφανείς ήρωες και συμπαραστάτες της προσπάθειας αυτής, με στήριξαν με όποιο τρόπο εφικτό μπορούσαν και μου προσέφεραν αμέριστη βοήθεια, όμως πάνω από όλα μου συμπαραστάθηκαν ως σωστοί και υποδειγματικοί γονείς. Κλείνοντας ένα μεγάλο ευχαριστώ στην αδερφή μου, την γιαγιά μου και την κοπέλα μου, οι οποίες ήταν συνεχώς δίπλα μου και με βοήθησαν με τον δικό τους τρόπο.

ΣΑΣ ΕΥΧΑΡΙΣΤΩ ΘΕΡΜΑ ΟΛΟΥΣ...!!!

Κ. Γεωργούσης

Ιούλιος 2010

Εποπτεία κυκλοφοριακού φόρτου σε πραγματικό χρόνο

Σύνοψη

Στον σύγχρονο κόσμο, η απευθείας παρακολούθηση κινούμενων αντικειμένων αξιοποιείται από πολλές μοντέρνες εφαρμογές, όπως υπηρεσίες εντοπισμού, συστήματα πλοήγησης για οχήματα, κ.ά. Το σύστημα επεξεργασίας οφείλει να αντεπεξέρχεται σε χωρική πληροφορία που μεταδίδεται συνεχώς από πολυάριθμες κινούμενες πηγές, συγκροτώντας με τις αλληπάλληλες ανανεώσεις των στιγμάτων τους ένα εφήμερο ρεύμα δεδομένων.

Σκοπός της παρούσας μεταπτυχιακής εργασίας είναι η υλοποίηση μιας απλής και εύχρηστης εφαρμογής για την υποθετική καταγραφή, δυναμική επεξεργασία και χαρτογραφική απεικόνιση του κυκλοφοριακού φόρτου στο οδικό δίκτυο της Αθήνας. Συγχρόνως, στόχος είναι η ανάδειξη της σημασίας των παραθύρων στις εφαρμογές ρευμάτων δεδομένων, φανερώνοντας πόσο επιτακτική είναι η χρήση τους σε ερωτήματα διαρκείας που χρησιμοποιούν βασικούς τελεστές της σχεσιακής άλγεβρας.

Κρίνοντας από τη συγκέντρωση των κινούμενων οχημάτων, η οποία μπορεί να ανιχνευτεί εύκολα με κάποιο σύστημα προσδιορισμού θέσης (π.χ. GPS), από την παρατηρούμενη θέση τους πάνω στο οδικό δίκτυο, την μέση ταχύτητα κίνησής τους σε κάθε οδικό άξονα και χρονική περίοδο, καθώς και τις διαθέσιμες λωρίδες του οδικού άξονα, επιβάλλεται σε πραγματικό χρόνο ένα κυμαινόμενο κόστος ανά όχημα, το οποίο ονομάζεται «τέλος συμφόρησης», ως αποτρεπτικό της χρήσεως του αυτοκινήτου σε ώρες αιχμής για βεβαρημένους άξονες του οδικού δικτύου.

Ο σχεδιασμός της εφαρμογής αφορά διακριτές θέσεις οχημάτων που αποστέλλονται στην εφαρμογή ως εισερχόμενο ρεύμα δεδομένων και βασίζεται στην παρακολούθηση και επεξεργασία των κινήσεων αυτών των σημειακών αντικειμένων. Η διαχείριση των ρευμάτων κινούμενων αντικειμένων αποσκοπεί στην επεξεργασία των στιγμάτων των παρακολουθούμενων οχημάτων σε πραγματικό χρόνο, για την εξαγωγή χρήσιμων πληροφοριών για τα οδικά τμήματα του δικτύου, καθώς και την παραγωγή ορισμένων διαρκώς ανανεώσιμων στατιστικών μεγεθών. Στο οδικό δίκτυο, τα κινούμενα οχήματα μοντελοποιούνται ως σημεία δύο διαστάσεων και συσχετίζονται με ένα χρονόσημο για κάθε θέση που κατέχουν, μιας και το στίγμα τους αλλάζει δυναμικά με τον χρόνο. Η ενημέρωση για την αλλαγή των θέσεων των οχημάτων υποθετικά ανιχνεύεται και συλλέγεται από σύγχρονες συσκευές και μεταδίδεται σε πραγματικό χρόνο σε έναν κεντρικό επεξεργαστή, ο οποίος παρακολουθεί και ελέγχει την κίνηση των παρακολουθούμενων οχημάτων.

Τέλος, προκειμένου να καταστεί εφικτή η επεξεργασία τόσο μεγάλου όγκου ρεύματος δεδομένων που καταγράφει την κατάσταση χιλιάδων οχημάτων στο δίκτυο, χρησιμοποιούνται κατάλληλα ερωτήματα διαρκείας που εκτελούνται επί μακρών και επιστρέφουν συνεχώς απαντήσεις σε πραγματικό χρόνο.

National Technical University of Athens
School of Rural & Surveying Engineering

POST-GRADUATE PROGRAMME «GEOINFORMATICS»

Diploma post-graduate dissertation
K. Georgousis

Athens, July 2010

Monitoring of traffic load in real time

Abstract

Recently, direct monitoring of *moving objects* is being used by several modern applications, such as location-based services, navigation systems for vehicles, etc. Hence, the processing mechanism has to cope with spatial information that is transmitted continuously by many moving sources, constituting a transient *data stream* of positional updates.

This thesis aims at implementing a simple, yet functional application for online processing and cartographic rendering of *traffic load* in the road network of Athens, assuming that positional information becomes available from the moving vehicles. It also points out the importance of *window* specification in data stream applications, revealing how imperative is their use in continuous queries that involve basic operators of relational algebra.

Judging from the concentration of moving vehicles, which can be detected easily with a positioning system (e.g. GPS), by their observed locations on the road network, by their average movement speed in each road axis and the respective time period, as well as the available lanes of each road axis, a so-called "*congestion charge*" is calculated in *real time*. This charge is a varying amount per vehicle, as a deterrent of car usage in peak hours for heavily loaded road segments.

This application handles *discrete point locations* of vehicles arriving as an input data stream. The streaming mechanism processes these positional updates in real time, in order to produce useful information for each road segment, as well as some periodically renewable statistics. Moving vehicles are modelled as two-dimensional points with a timestamp, since their position changes dynamically with time. Location updates can be detected and collected by means of certain devices and subsequently transmitted in real time to a central processor, which is responsible for vehicle monitoring.

Finally, in order to enable efficient processing of such a voluminous data stream that captures the status of thousands of vehicles on the network, certain *continuous queries* are being executed, each returning incremental response to the users in real time.

Πίνακας Περιεχομένων

Κεφάλαιο 1

| | |
|---|----------|
| Εισαγωγή..... | 1 |
| 1.1 Αντικείμενο εργασίας..... | 1 |
| 1.2 Προσέγγιση της εργασίας..... | 1 |
| 1.3 Οργάνωση του κειμένου της εργασίας..... | 3 |

Κεφάλαιο 2

| | |
|--|----------|
| Διαχείριση ρευμάτων δεδομένων..... | 5 |
| 2.1 Εισαγωγή..... | 5 |
| 2.2 Μοντέλο ρεύματος δεδομένων..... | 6 |
| 2.2.1 Κανόνες για την επεξεργασία ρευμάτων δεδομένων..... | 7 |
| 2.3 Ανεπάρκεια συμβατικών ΣΔΒΔ..... | 8 |
| 2.4 Ερωτήματα σε ρεύματα δεδομένων..... | 11 |
| 2.4.1 Ερωτήματα διαρκείας βάσει του χρόνου ή του περιεχομένου..... | 12 |
| 2.4.2 Ερωτήματα συνάθροισης σε ρεύματα δεδομένων..... | 12 |
| 2.5 Παράθυρα σε ρεύματα δεδομένων..... | 13 |
| 2.5.1 Παράθυρα οροσήμου..... | 14 |
| 2.5.2 Κυλιόμενα παράθυρα..... | 15 |
| 2.6 Γλώσσες ερωταποκρίσεων ρευμάτων δεδομένων..... | 20 |
| 2.6.2 Προτεινόμενες γλώσσες ερωταποκρίσεων..... | 20 |
| 2.7 Κυριότερα συστήματα διαχείρισης ρευμάτων δεδομένων..... | 21 |
| 2.7.1 AURORA..... | 21 |
| 2.7.2 GIGASCOPE..... | 23 |
| 2.7.3 NiagaraCQ..... | 24 |
| 2.7.4 STREAM (STanford stREam datA Management)..... | 24 |
| 2.7.5 TelegraphCQ..... | 26 |

Κεφάλαιο 3

| | |
|--|-----------|
| Διαχείριση κινούμενων αντικειμένων..... | 31 |
| 3.1 Εισαγωγή..... | 31 |
| 3.2 Τύποι χωρικών αντικειμένων..... | 32 |
| 3.3 Μοντέλα αναπαράστασης..... | 32 |
| 3.3.1 Αφηρημένο μοντέλο..... | 33 |
| 3.3.2 Διακριτό μοντέλο..... | 34 |
| 3.4 Χωρικοί τελεστές..... | 34 |
| 3.5 Τροχιές κινούμενων αντικειμένων..... | 37 |
| 3.5.1 Κατηγορίες κίνησης..... | 38 |
| 3.5.2 Αβεβαιότητα αναπαράστασης τροχιάς..... | 39 |
| 3.6 Κατηγορίες ερωτημάτων..... | 41 |
| 3.6.1 Ερωτήματα δέσης..... | 41 |
| 3.6.2 Ερωτήματα τροχιάς..... | 43 |
| 3.6.3 Σύνθετα ερωτήματα..... | 44 |
| 3.7 Δεικτοδότηση - Ευρετήρια..... | 44 |

| | |
|--|------------|
| 3.8 Ζητήματα προς επίλυση..... | 46 |
| 3.9 Διαχείριση ρευμάτων κινούμενων αντικειμένων..... | 47 |
| | |
| Κεφάλαιο 4 | |
| Διαδικτυακοί Χάρτες..... | 49 |
| 4.1 Εισαγωγή..... | 49 |
| 4.2 Τύποι διαδικτυακών χαρτών..... | 50 |
| 4.2.1 Στατικοί διαδικτυακοί χάρτες..... | 52 |
| 4.2.2 Δυναμικά παραγόμενοι διαδικτυακοί χάρτες..... | 52 |
| 4.2.3 Κατανεμημένοι διαδικτυακοί χάρτες..... | 53 |
| 4.2.4 Διαδικτυακοί χάρτες με κίνηση..... | 53 |
| 4.2.5 Διαδικτυακοί χάρτες σε πραγματικό χρόνο..... | 54 |
| 4.2.6 Εξατομικευμένοι διαδικτυακοί χάρτες..... | 55 |
| 4.2.7 Διαδραστικοί διαδικτυακοί χάρτες..... | 55 |
| 4.2.8 Αναλυτικοί διαδικτυακοί χάρτες..... | 56 |
| 4.2.9 Συνεργατικοί διαδικτυακοί χάρτες..... | 57 |
| 4.3 Αξιολόγηση διαδικτυακών χαρτών..... | 58 |
| 4.4 Ιστορική αναδρομή..... | 60 |
| 4.5 Τεχνολογίες διαδικτυακής χαρτογράφησης..... | 63 |
| 4.5.1 Πρωτόκολλο Web Map Service (WMS)..... | 64 |
| 4.5.2 Πρωτόκολλο Web Feature Service (WFS)..... | 65 |
| 4.5.3 Πρωτόκολλο Web Coverage Service (WCS)..... | 67 |
| | |
| Κεφάλαιο 5 | |
| Ανάλυση Απαιτήσεων Και Σχεδιασμός Εφαρμογής..... | 69 |
| 5.1 Εισαγωγή..... | 69 |
| 5.2 Αρχιτεκτονική εφαρμογής..... | 69 |
| 5.3 Περιγραφή των λειτουργιών του εξυπηρετητή δεδομένων..... | 72 |
| 5.3.1 Περιγραφή στατικών δεδομένων εφαρμογής..... | 72 |
| 5.3.2 Περιγραφή δυναμικών δεδομένων εφαρμογής..... | 77 |
| 5.3.3 Περιγραφή ερωτημάτων διαρκείας της εφαρμογής..... | 84 |
| 5.3.4 Περιγραφή συναρτήσεων PL/pgSQL της εφαρμογής..... | 87 |
| 5.4 Χαρτογραφική απεικόνιση μέσω διαδικτυακού εξυπηρετητή..... | 90 |
| 5.4.1 Παραγωγή των αποτελεσμάτων..... | 91 |
| 5.4.2 Περιοδική ανανέωση και απεικόνιση των αποτελεσμάτων..... | 94 |
| 5.5 Συγγενικές εφαρμογές..... | 97 |
| | |
| Κεφάλαιο 6 | |
| Συμπεράσματα & Προτάσεις..... | 99 |
| 6.1 Συμπεράσματα..... | 99 |
| 6.2 Προτάσεις - Μελλοντικές επεκτάσεις..... | 100 |
| | |
| Παράρτημα | 103 |
| Παράρτημα Α: Δομή και περιεχόμενο γεωγραφικού αρχείου..... | 103 |
| Παράρτημα Β: Βάση δεδομένων..... | 103 |
| Παράρτημα Γ: Κώδικες εφαρμογής..... | 114 |
| Παράρτημα Δ: Αποτελέσματα αρχείων KML και XML..... | 133 |

Βιβλιογραφία – Αναφορές.....137

Γλωσσάριο Όρων.....141

Κατάλογος Σχημάτων

Κεφάλαιο 2

| | | |
|-----------|--|---------|
| Σχήμα 2.1 | Διάγραμμα επεξεργασίας ρευμάτων δεδομένων. | Σελ. 7 |
| Σχήμα 2.2 | Βασική αρχιτεκτονική των συστημάτων διαχείρισης βάσεων δεδομένων και των μηχανών επεξεργασίας ρευμάτων. | Σελ. 10 |
| Σχήμα 2.3 | Γενικό παράδειγμα εφαρμογής παραμέτρων RANGE και SLIDE. | Σελ. 15 |
| Σχήμα 2.4 | Παράδειγμα χρονικού παραθύρου, με RANGE = 20sec και SLIDE = 10sec. | Σελ. 17 |
| Σχήμα 2.5 | Παράδειγμα παραθύρου πλειάδων, με N = 6 πλειάδες. | Σελ. 18 |
| Σχήμα 2.6 | Παράδειγμα μεριστικού παραθύρου, με N = 3 πλειάδες για κάθε ομάδα που ορίζεται από το γνώρισμα L = {id}. | Σελ. 19 |
| Σχήμα 2.7 | Αρχιτεκτονική του συστήματος Aurora. | Σελ. 22 |
| Σχήμα 2.8 | Η αρχιτεκτονική του συστήματος STREAM. | Σελ. 25 |
| Σχήμα 2.9 | Η αρχιτεκτονική του συστήματος TelegraphCQ. | Σελ. 28 |

Κεφάλαιο 3

| | | |
|-----------|---|---------|
| Σχήμα 3.1 | Αφηρημένη αναπαράσταση κινούμενου σημείου. | Σελ. 33 |
| Σχήμα 3.2 | Διακριτές αναπαραστάσεις για κινούμενο σημείο και κινούμενη περιοχή. | Σελ. 34 |
| Σχήμα 3.3 | Παράδειγμα ελάχιστων περιβαλλόντων παραλληλόγραμμων σε αντικείμενα. | Σελ. 35 |
| Σχήμα 3.4 | Διάφοροι χωρικοί τελεστές. | Σελ. 36 |
| Σχήμα 3.5 | Αναπαράσταση τροχιάς ενός κινούμενου αντικειμένου στον τρισδιάστατο χώρο (x, y, t). | Σελ. 38 |
| Σχήμα 3.6 | Μοντέλο αβεβαιότητας κινούμενου αντικειμένου. | Σελ. 40 |
| Σχήμα 3.7 | Κατηγορίες ερωτημάτων σε κινούμενα αντικείμενα. | Σελ. 42 |

Κεφάλαιο 4

| | | |
|------------|--|---------|
| Σχήμα 4.1 | Ταξινόμηση διαδικτυακών χαρτών σε στατικούς και δυναμικούς. | Σελ. 51 |
| Σχήμα 4.2 | Ταξινόμηση διαδικτυακών χαρτών σύμφωνα με τον βαθμό πολυπλοκότητας. | Σελ. 51 |
| Σχήμα 4.3 | Στατικός χάρτης USGS DRG. | Σελ. 52 |
| Σχήμα 4.4 | Animated χάρτης καταγραφής των ψηφοφοριών της Γερουσίας των ΗΠΑ. | Σελ. 54 |
| Σχήμα 4.5 | Προγνωστικός χάρτης συνολικής νεφοκάλυψης TCC της Ελλάδος, από Ε.Μ.Υ. '2010-06-19 21:00:30'. | Σελ. 55 |
| Σχήμα 4.6 | Παράδειγμα ενός αναλυτικού διαδικτυακού χάρτη (CentreMap). | Σελ. 57 |
| Σχήμα 4.7 | Συσχέτιση μεταξύ πελατών / εξυπηρετητών και OGC πρωτοκόλλων. | Σελ. 63 |
| Σχήμα 4.8 | Παράδειγμα αιτήματος «Get Map». | Σελ. 64 |
| Σχήμα 4.9 | Παράδειγμα υπηρεσίας WFS. | Σελ. 66 |
| Σχήμα 4.10 | Υπηρεσία WFS από την European Space Agency Fire. | Σελ. 67 |
| Σχήμα 4.11 | Παράδειγμα υπηρεσίας WCS, αίτημα «GetCoverage». | Σελ. 68 |

Κεφάλαιο 5

| | | |
|-----------|--|---------|
| Σχήμα 5.1 | Αρχιτεκτονικό διάγραμμα της εφαρμογής. | Σελ. 70 |
| Σχήμα 5.2 | Χαρτογραφικό υπόβαθρο του οδικού κυκλοφοριακού δικτύου της Αθήνας. | Σελ. 73 |
| Σχήμα 5.3 | Απεικόνιση KML αρχείου μέσω GoogleEarth. | Σελ. 93 |
| Σχήμα 5.4 | Στιγμιότυπο εφαρμογής για χρονική στιγμή '2010-01-12 12:05:30'. | Σελ. 95 |
| Σχήμα 5.5 | Στιγμιότυπο εφαρμογής για χρονική στιγμή '2010-01-12 12:07:00'. | Σελ. 95 |
| Σχήμα 5.6 | Στιγμιότυπο εφαρμογής για χρονική στιγμή '2010-01-12 12:07:30'. | Σελ. 96 |
| Σχήμα 5.7 | Στιγμιότυπο εφαρμογής για χρονική στιγμή '2010-01-12 12:08:00'. | Σελ. 96 |

Παράρτηματα

| | | |
|-----------|--|----------|
| Σχήμα Β.1 | Διάγραμμα απεικόνισης πινάκων της βάσης δεδομένων της εφαρμογής. | Σελ. 106 |
| Σχήμα Β.2 | Διάγραμμα απεικόνισης πινάκων επέκτασης της βάσης δεδομένων. | Σελ. 113 |

Κατάλογος Πινάκων

Κεφάλαιο 2

| | | |
|-------------|---|---------|
| Πίνακας 2.1 | Αντιπαραβολή χαρακτηριστικών μεταξύ Συστημάτων Διαχείρισης Βάσεων Δεδομένων και Ρευμάτων Δεδομένων. | Σελ. 10 |
| Πίνακας 2.2 | Επεξήγηση τελεστών συνάθροισης | Σελ. 13 |

Κεφάλαιο 5

| | | |
|-------------|---|---------|
| Πίνακας 5.1 | Εκτιμώμενη απόδοση λωρίδων ανάλογα με την κατηγορία του δρόμου. | Σελ. 75 |
| Πίνακας 5.2 | Εκτιμώμενη απόδοση τέλους συμφόρησης, ανάλογα την κατηγορία του δρόμου και τη χρονικής περιόδου μετακίνησης των οχημάτων. | Σελ. 76 |
| Πίνακας 5.3 | Εκτιμώμενο επίπεδο υπηρεσιών του οδικού δικτύου. | Σελ. 76 |
| Πίνακας 5.4 | Στυλ μορφοποίησης απεικόνισης των οδικών αξόνων. | Σελ. 92 |

Κεφάλαιο 1

Εισαγωγή

1.1 Αντικείμενο εργασίας

Στον σύγχρονο κόσμο, η απευθείας παρακολούθηση κινούμενων αντικειμένων (π.χ. άνθρωποι, ζώα ή συσκευές) έχει δώσει ώθηση σε πολλές μοντέρνες εφαρμογές, όπως υπηρεσίες εντοπισμού, εποπτεία φυσικού περιβάλλοντος, συστήματα πλοήγησης για οχήματα, κ.ά. Σε παρόμοιες περιπτώσεις, το σύστημα επεξεργασίας οφείλει να αντεπεξέρχεται σε χωρική πληροφορία που μεταδίδεται συνεχώς από πολυάριθμες κινούμενες πηγές, συγκροτώντας με τις αλληπάλληλες ανανεώσεις των στιγμάτων τους ένα εφήμερο ρεύμα δεδομένων.

Η εργασία αποσκοπεί στην υλοποίηση μιας απλής και εύχρηστης εφαρμογής για την υποδεικτική καταγραφή, δυναμική επεξεργασία και χαρτογραφική απεικόνιση του κυκλοφοριακού φόρτου στο οδικό δίκτυο της Αθήνας. Είναι γνωστό ότι η αυξανόμενη κυκλοφορία οχημάτων σε συνδυασμό με την έντονη αστικοποίηση, υποβαθμίζει την ποιότητα ζωής των κατοίκων ιδίως σε μεγάλα πολεοδομικά συγκροτήματα. Διάφορες πόλεις, όπως για παράδειγμα το Λονδίνο, η Νέα Υόρκη, το Μιλάνο, κ.α., έχουν αναπτύξει και ήδη εφαρμόσει ή εξετάζουν πολύ σοβαρά την επιβολή ενός συστήματος πληρωμής διοδίων ανάλογα με την τρέχουσα συμφόρηση στους κύριους οδικούς άξονες τους, το οποίο ονομάζεται «τέλος συμφόρησης» (congestion charge).

Κρίνοντας από τη συγκέντρωση των κινούμενων οχημάτων, η οποία μπορεί να ανιχνευτεί εύκολα με κάποιο σύστημα προσδιορισμού θέσης (π.χ. GPS), από την παρατηρούμενη θέση τους πάνω στο οδικό δίκτυο, την μέση ταχύτητα κίνησής τους σε κάθε οδικό άξονα και χρονική περίοδο, καθώς και τις διαδέσιμες λωρίδες του οδικού άξονα, επιβάλλεται σε πραγματικό χρόνο (online) ένα κυμαινόμενο κόστος ανά όχημα, ως αποτρεπτικό της χρήσεως του αυτοκινήτου σε ώρες αιχμής για βεβαρημένους άξονες του οδικού δικτύου.

1.2 Προσέγγιση της εργασίας

Αντί της αποθήκευσης σε ένα σύστημα βάσεων δεδομένων, τα μαζικά μεταβαλλόμενα στοιχεία απαιτούν επεξεργασία σε πραγματικό χρόνο. Καθώς τα νέα στίγματα των οχημάτων καταφθάνουν διαρκώς, έγκυρες απαντήσεις πρέπει να δίνονται σχεδόν αμέσως στα ερωτήματα διάρκειάς που έχουν διατυπωθεί. Παράλληλα, πρέπει να λαμβάνεται υπόψη ότι οι θέσεις των οχημάτων που εξετάζονται και αποστέλλονται στην εφαρμογή ως εισερχόμενο ρεύμα δεδομένων είναι διακριτές και ο σχεδιασμός της εφαρμογής βασίζεται στην παρακολούθηση και επεξεργασία των κινήσεων αυτών των σημειακών αντικειμένων.

Επομένως η προσέγγιση της εργασίας δίνει έμφαση στην χωρική επεξεργασία σημειακών θέσεων οχημάτων, την συσχέτισή τους με τους αντίστοιχους οδικούς άξονες και την διαρκή ανανέωση ορισμένων επιλεγμένων στατιστικών μεγεθών.

Στο οδικό δίκτυο, τα κινούμενα οχήματα μοντελοποιούνται ως σημεία δύο διαστάσεων (x,y), τα οποία συνδέονται με μία συγκεκριμένη χρονική στιγμή για κάθε θέση που κατέχουν, μιας και η θέση τους αλλάζει δυναμικά με τον χρόνο. Η ενημέρωση για την αλλαγή των θέσεων των οχημάτων, υποθετικά ανιχνεύεται και συλλέγεται από σύγχρονες συσκευές, όπως GPS, PDA, κ.ά., και μεταδίδεται σε πραγματικό χρόνο σε έναν κεντρικό επεξεργαστή, ο οποίος παρακολουθεί και ελέγχει την κίνηση των παρακολουθούμενων οχημάτων.

Από την διαχείριση και επεξεργασία αυτών των χωρο-χρονικών ρευμάτων δεδομένων που αφορούν τις σημειακές θέσεις των παρακολουθούμενων οχημάτων σε πραγματικό χρόνο, εξάγονται χρήσιμες πληροφορίες για τα οδικά τμήματα του δικτύου, καθώς και κάποια διαρκή ανανεώσιμα στατιστικά μεγέθη:

- A. Η συγκέντρωση των κινούμενων οχημάτων, όπως ανιχνεύεται από την παρατηρούμενη θέση τους πάνω στο δίκτυο.
- B. Η τροχιά των οχημάτων με παράλληλη καταγραφή των οδικών αξόνων που κινούνται και την διάρκεια παραμονής τους σε αυτούς, όπως ανιχνεύεται από την παρατηρούμενη θέση τους πάνω στο δίκτυο και την διαδρομή που προκύπτει από την κίνησή τους (δυναμική αλλαγή των θέσεων).
- Γ. Ο κυκλοφοριακός φόρτος των οδικών αξόνων με βάση την κατηγορία του δρόμου στον οποίο ανήκουν.
- Δ. Οι μέσες ταχύτητες κίνησης των οχημάτων, ανά οδικό άξονα και χρονική περίοδο.

Σύμφωνα με τα παραπάνω, επιβάλλεται σε πραγματικό χρόνο ένα κυμαινόμενο κόστος ανά όχημα, για συγκεκριμένους οδικούς άξονες. Οι χρεώσεις αυτές καταγράφονται ανά όχημα και γίνεται δυναμική ενημέρωση σε τακτά χρονικά διαστήματα κατά την διάρκεια της ημέρας στο συνολικό κόστος χρήσης του αυτοκινήτου, με απώτερο σκοπό να δίνεται η δυνατότητα στους οδηγούς να γνωρίζουν το συνολικό κόστος χρέωσης των μετακινήσεών τους για κάθε ημέρα της επιλογής τους.

Προκειμένου όμως να καταστεί εφικτή η επεξεργασία τόσο μεγάλου όγκου ρεύματος δεδομένων (*data stream*) που καταγράφει την κατάσταση χιλιάδων οχημάτων στο δίκτυο, χρησιμοποιούνται κατάλληλα ερωτήματα διαρκείας (*continuous queries*), τα οποία εκτελούνται επί μακρών και επιστρέφουν συνεχώς απαντήσεις σε πραγματικό χρόνο. Οι φάσεις εκπόνησης της εργασίας είναι οι εξής:

- Ανάλυση των απαιτήσεων για ένα απλουστευμένο σενάριο εφαρμογής τέλους συμφόρησης, λαμβάνοντας υπόψη κυρίως χωρικές παραμέτρους (π.χ.

- Οργάνωση του σχήματος μιας χωρικής βάσης δεδομένων σε περιβάλλον TelegraphCQ, το οποίο προσφέρει αξιοποίηση της χωρικής υποδομής της PostgreSQL. Στην βάση δεδομένων αποτυπώνονται τόσο στατικοί πίνακες (γεωμετρία δικτύου), όσο και δυναμικά ρεύματα δεδομένων (στίγματα οχημάτων), καθώς και παράγωγα ρεύματα και πίνακες (τροχιές ρευμάτων, κυκλοφοριακός φόρτος οδικών αξόνων, χρεώσεις οχημάτων).
- Διατύπωση κατάλληλων ερωτημάτων διαρκείας σε μίαν επέκταση της SQL (CQL- Continuous Query Language) για την χωρική επεξεργασία των θέσεων των κινούμενων οχημάτων και την ανανέωση στατιστικών στοιχείων (π.χ. οχήματα ανά οδικό άξονα, ταχύτητα ανά οδικό άξονα και χρονική περίοδο, κλπ.). Τα συγκεκριμένα ερωτήματα προσδιορίζουν παράθυρα (*windows*), ώστε να απομονώνουν κάθε φορά τις πιο πρόσφατες μετρήσεις (π.χ. στίγματα θέσεων των οχημάτων κατά το τελευταίο ημίωρο).
- Χρήση ενός απλού μαθηματικού τύπου, για τον υπολογισμό του επιβαλλόμενου τέλους συμμόρφωσης (ποσό ανά όχημα) ανάλογα με τις συνθήκες κυκλοφορίας των οχημάτων στο οδικό δίκτυο της πόλης, εφαρμόζοντας απλούς κανόνες απόφασης (χρονική περίοδος και επίπεδο συμμόρφωσης).
- Απεικόνιση των αποτελεσμάτων σε χάρτη, με την βοήθεια εργαλείων Java/Javascript, PHP και Html. Η απεικόνιση των αποτελεσμάτων σε χάρτη γίνεται σε περιβάλλον GoogleMaps, όπου τα εικονιζόμενα στοιχεία είναι γραμμικά.

1.3 Οργάνωση του κειμένου της εργασίας

Η συγκεκριμένη εργασία προσεγγίζει το θέμα της διαχείρισης και διαδικτυακής απεικόνισης των ρευμάτων χωρικών δεδομένων, τόσο σε επίπεδο θεωρητικής μελέτης όσο και σε επίπεδο υλοποίησης. Έτσι τα κεφάλαια 2 έως 4 καλύπτουν το θεωρητικό κομμάτι της εργασίας ενώ το κεφάλαιο 5 αναφέρεται στην παρούσα υλοποίηση της εφαρμογής. Συγκεκριμένα, ο τρόπος διάρθρωσης της εργασίας είναι ο ακόλουθος:

Στο 2^ο κεφάλαιο γίνεται μία σύντομη ανασκόπηση της διεθνούς βιβλιογραφίας σχετικά με την διαχείριση και επεξεργασία των ρευμάτων δεδομένων, ώστε να γίνει αντιληπτό πώς αυτές οι συνεχόμενες ροές δεδομένων, σχετίζονται με το πρόβλημα και προσδιορίζουν την φύση του.

Στο 3^ο κεφάλαιο γίνεται μία σύντομη ανασκόπηση της διεθνούς βιβλιογραφίας σχετικά με την διαχείριση κινούμενων αντικειμένων, ώστε να γίνει αντιληπτό πώς γίνεται η διαχείρισή τους μέσω ρευμάτων.

Στο 4^ο κεφάλαιο γίνεται μία σύντομη ανασκόπηση σχετικά με τους διαδικτυακούς χάρτες και τις τεχνολογίες διαδικτυακής χαρτογράφησης (πρωτόκολλα WMS, WFS, WCS). Περιγράφονται οι διαφορετικοί τύποι των διαδικτυακών χαρτών, η οπτικοποίηση που προσφέρεται μέσω αυτών και η αξιολόγησή τους.

Στο 5^ο κεφάλαιο αναλύονται οι απαιτήσεις της εφαρμογής και παρουσιάζεται ο σχεδιασμός της. Αναλύεται η αρχιτεκτονική του συστήματος, περιγράφονται οι λειτουργίες του και παρουσιάζονται αναλυτικά τα εμπλεκόμενα δεδομένα που σχετίζονται με την εφαρμογή και την προσδιορίζουν.

Τέλος, στο 6^ο κεφάλαιο παρατίθενται τα συμπεράσματα που προκύπτουν από την υλοποίηση της εφαρμογής, την αξιολόγηση του σχεδιασμού της χωρικής βάσης και των δυνατοτήτων της στην ανταπόκριση των ερωτημάτων διαρκείας. Επίσης προτείνονται ορισμένες χρήσιμες μελλοντικές επεκτάσεις της εφαρμογής.

Κεφάλαιο 2

Διαχείριση ρευμάτων δεδομένων

2.1 Εισαγωγή

Τα συμβατικά Συστήματα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ, DataBase Management Systems - DBMS) είναι σχεδιασμένα για την αποθήκευση αρκετά μεγάλου όγκου δεδομένων, πάνω στα οποία οι χρήστες του συστήματος έχουν την εποπτεία με την υποβολή διαφόρων ειδών ερωτημάτων (*queries*) ή με την εκτέλεση ορισμένων ενεργειών, όπως είναι η εισαγωγή νέων στοιχείων, η διαγραφή ή η τροποποίηση παλαιότερων.

Το ενδιαφέρον στα ΣΔΒΔ εστιάζεται τις περισσότερες φορές στα πιο πρόσφατα στοιχεία που περιέχει η βάση δεδομένων, χωρίς να υπάρχει ενδιαφέρον για τα προηγούμενα στιγμιότυπά της. Ένα τυπικό ΣΔΒΔ είναι ικανό να παρέχει ακριβείς απαντήσεις στα ερωτήματα που του υποβάλλονται, εφόσον όλα τα στοιχεία που χρειάζονται στους υπολογισμούς για την απόκριση στα ερωτήματα είναι άμεσα διαθέσιμα. Συνήθως, ορισμένου τύπου ερωτήματα για να απαντηθούν χρειάζονται αρκετό χρόνο, επομένως αν υπάρχει απαίτηση για απόκριση σε πραγματικό χρόνο (*real-time*), αυτή δεν μπορεί να καλυφθεί πάντοτε.

Επειδή όμως, οι εφαρμογές που έχουν ως αντικείμενο την παρακολούθηση (*monitoring applications*) των δεδομένων υπερέχουν και παρουσιάζουν περισσότερο ενδιαφέρον, οι προδιαγραφές των τυπικών ΣΔΒΔ είναι αρκετά περιοριστικές όσον αφορά την διαχείριση αυτών. Πρόκειται για εφαρμογές όπου τα στοιχεία παρουσιάζονται όχι πλέον με τη μορφή στατικών σχέσεων (*relations*), αλλά ως δεδομένα συνεχούς ροής μέσα σε ένα δίκτυο, δηλαδή ως ένα **ρεύμα δεδομένων (*data stream*)**. Χαρακτηριστικά παραδείγματα τέτοιων εφαρμογών είναι συστήματα εποπτείας δικτύων υπολογιστών, συστήματα παρακολούθησης της διακύμανσης των οικονομικών μεγεθών (π.χ. χρηματιστηριακοί δείκτες), συστήματα διαχείρισης διαφόρων δικτύων αισθητήρων, συστήματα στρατιωτικών εφαρμογών, κ.ά.

Οι εφαρμογές αυτές εστιάζουν στις πιο πρόσφατες πληροφορίες που εισέρχονται στο σύστημα με τα εξής κοινά χαρακτηριστικά:

- ❖ Η διαχείριση της εισερχόμενης πληροφορίας δυναμικά με τη μορφή ρευμάτων δεδομένων, με έμφαση στην πιο πρόσφατη πληροφορία.
- ❖ Η διατύπωση **ερωτημάτων διαρκείας (*continuous queries*)**, τα οποία απαιτούν σε πραγματικό χρόνο επεξεργασία και επιστροφή απαντήσεων.

2.2 Μοντέλο ρεύματος δεδομένων

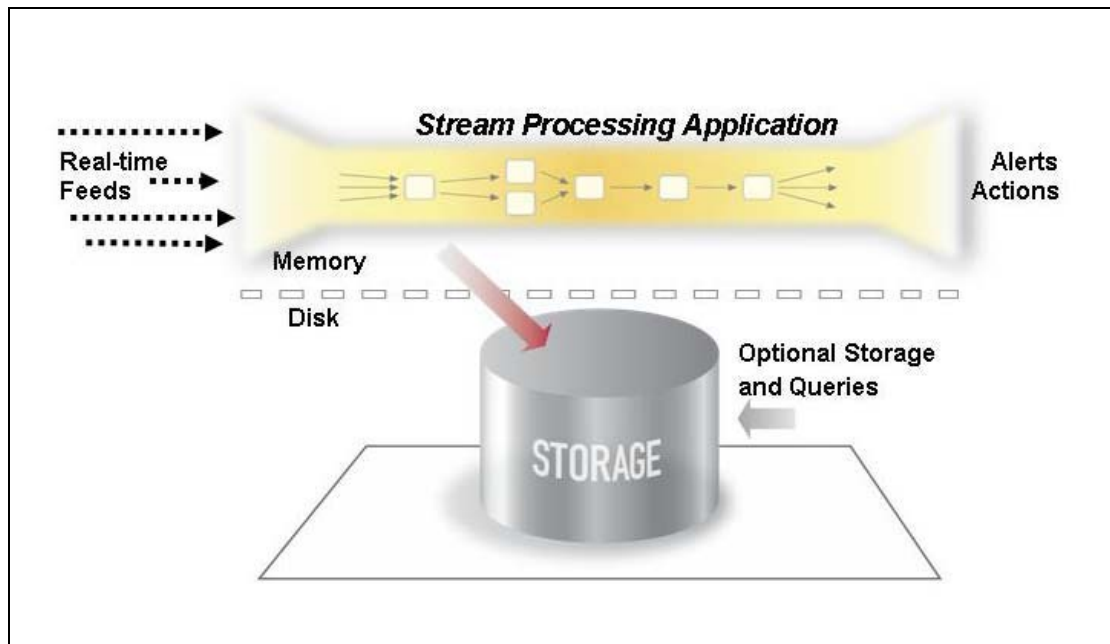
Ένα ρεύμα δεδομένων μπορεί να θεωρηθεί ως μια ακολουθία στοιχείων τα οποία παράγονται διαρκώς από μια πηγή. Ανάλογα λοιπόν με την εφαρμογή, αυτή η πηγή δεδομένων μπορεί να είναι ένας αισθητήρας μέτρησης κάποιου φυσικού μεγέθους (π.χ. αισθητήρας θερμοκρασίας σε έναν μετεωρολογικό σταθμό), μια μετοχή κάποιας εταιρίας της οποίας η διακύμανση της τιμής της παρακολουθείται διαρκώς από το χρηματιστήριο, κ.ά.

Διαμέσου του ρεύματος μπορούν να μεταφέρονται στοιχεία τα οποία είναι δυνατόν να έχουν τη γνωστή μορφή των *σχεσιακών πλειάδων (relational tuples)*, όπως για παράδειγμα οι καταγραφές των τηλεφωνικών κλήσεων, οι καταγραφές των μετρήσεων από διαφορετικούς αισθητήρες, κ.α.. Η διαφορά είναι πως αυτές οι πλειάδες δεν καταχωρούνται σε στατικούς πίνακες όπως συμβαίνει στα τυπικά ΣΔΒΔ, αλλά καταφθάνουν διαρκώς σε πραγματικό χρόνο με απρόβλεπτο έως πάρα πολύ γρήγορο ρυθμό, χωρίς περιορισμούς ως προς το μέγεθος ή τις ιδιότητές τους. Συνεπώς, ένα τουλάχιστον τμήμα ή ακόμα και το σύνολο των δεδομένων εισόδου δεν είναι διαθέσιμα για επεξεργασία από το δίσκο ή την κύρια μνήμη, αλλά καταφθάνουν με τη μορφή ενός ή περισσοτέρων ρευμάτων δεδομένων.

Στις πλειάδες των ρευμάτων δεδομένων υπάρχει καταγεγραμμένη η πηγή προέλευσης των στοιχείων, καθώς και ένα *χρονόσημο (timestamp)*, ώστε να είναι γνωστή η χρονική στιγμή που καταφθάνει ή παράγεται η πληροφορία. Το χρονόσημο ή η διάταξη των πλειάδων είναι εξαιρετικά πολύτιμες παράμετροι για τα ερωτήματα που μπορεί να τεθούν στα δεδομένα. Δεν υπάρχει δυνατότητα ελέγχου στην σειρά με την οποία καταφθάνουν τα δεδομένα από τις εξωτερικές πηγές στο σύστημα για επεξεργασία, είτε πρόκειται για ένα ρεύμα είτε για συνένωση στοιχείων από πολλαπλά ρεύματα. Όταν ολοκληρωθεί η επεξεργασία μιας πλειάδας του ρεύματος, αυτή απορρίπτεται ή αρχειοθετείται με τη μορφή *περιλήψεων (data synopses, summaries)*, που σημαίνει ότι το συγκεκριμένο στοιχείο δεν μπορεί πλέον να ανακτηθεί εύκολα, εκτός και εάν έχει φυλαχθεί σκόπιμα στην κύρια μνήμη του συστήματος. Το διαθέσιμο μέγεθος της κύριας μνήμης είναι τις περισσότερες φορές πολύ μικρό για να καταφέρει να διατηρήσει όλο τον όγκο που μεταφέρει ένα ρεύμα δεδομένων και συνήθως διατηρείται κάποιιο μικρό τμήμα των πιο πρόσφατων περιεχομένων ή κάποιας μορφής συνοπτική πληροφορία (βλ. σχήμα 2.1).

Επίσης υπάρχει η περίπτωση τα δεδομένα που εισέρχονται στο σύστημα να έχουν συγχωνευτεί με παρόμοια στοιχεία από διαφορετικές πηγές δεδομένων: λ.χ. είναι δυνατόν να καταφθάνουν στο σύστημα στοιχεία από πολλαπλούς αισθητήρες και για διαφορετικές χρονικές περιόδους ή η περίπτωση πολυδιάστατων ρευμάτων δεδομένων, όπου τα στοιχεία αναφέρονται σε πολλαπλές πηγές δεδομένων.

Συνήθως μαζί με τα ρεύματα δεδομένων υπάρχουν στο σύστημα και άλλα δεδομένα σε μορφή αποθηκευμένων σχέσεων, οπότε πολύ συχνά είναι απαραίτητη η σύνδεση (*join*) μεταξύ δυναμικών ρευμάτων δεδομένων και στατικών σχεσιακών δεδομένων.



Σχήμα 2.1: Διάγραμμα επεξεργασίας ρευμάτων δεδομένων. (Πηγή: [SCZ05])

Ειδικά σε ό,τι αφορά τα σχεσιακά δεδομένα, γίνεται η υπόθεση ότι τα περιεχόμενά τους παραμένουν σε σημαντικό βαθμό αμετάβλητα, ώστε να απλοποιείται σημαντικά το ζήτημα της διαχείρισης δοσοληψιών στο σύστημα, αφού σπάνια θα συμβαίνουν ταυτόχρονα μεταβολές τόσο στις στατικές σχέσεις όσο και στην επεξεργασία των ρευμάτων δεδομένων.

2.2.1 Κανόνες για την επεξεργασία ρευμάτων δεδομένων

Υπάρχουν ορισμένες απαιτήσεις τις οποίες οποιοδήποτε λογισμικό συστημάτων πρέπει να είναι σε θέση να μπορεί να καλύπτει, ώστε να υπερέχει στην επεξεργασία ρευμάτων σε πραγματικό χρόνο. Έχει προταθεί [SCZ05] μία κατηγοριοποίηση των απαιτήσεων, η οποία χρησιμεύει στην αξιολόγηση, από οποιονδήποτε χρήστη, της επεξεργασίας ρευμάτων που πραγματοποιεί το σύστημα το οποίο χρησιμοποιεί. Απώτερος σκοπός των κανόνων είναι η σύγκριση ανάμεσα στα σχεσιακά ΣΔΒΔ και της αναλυτικής επεξεργασίας σε πραγματικό χρόνο.

Κανόνας 1^{ος} - Συνεχής ροή δεδομένων. Η πρώτη απαίτηση για ένα σύστημα επεξεργασίας ρευμάτων σε πραγματικό χρόνο είναι να υποβληθούν σε επεξεργασία τα μηνύματα «in-stream», χωρίς οποιαδήποτε απαίτηση να αποθηκευτούν για να εκτελεστεί οποιαδήποτε λειτουργία ή ακολουθία διαδικασιών.

Κανόνας 2^{ος} - Ερωτήματα SQL στα ρεύματα. Πρέπει να υποστηρίζεται από το σύστημα μια υψηλού επιπέδου γλώσσα «StreamSQL» με ενσωματωμένους επεκτάσιμους τελεστές.

Κανόνας 3^{ος} - Διαχείριση ατελειών των ρευμάτων (Καθυστερήσεις, ελλείψεις και μη σωστή διάταξη των δεδομένων). Πρέπει να υπάρχουν ενσωματωμένοι μηχανισμοί για την παροχή ευκαμψίας του συστήματος στις ενδεχόμενες «ατέλειες» που μπορεί να παρουσιάσει ένα ρεύμα, όπως ελλείψεις

στοιχείων και μη σωστή διάταξή τους, προβλήματα που παρουσιάζονται συνήθως στα ρεύματα δεδομένων στον πραγματικό κόσμο.

Κανόνας 4^{ος} - Παραγωγή προβλέψιμων αποτελεσμάτων. Μια μηχανή επεξεργασίας ρευμάτων δεδομένων πρέπει να μπορεί να εγγυάται ντετερμινιστικά και επαναλαμβανόμενα αποτελέσματα.

Κανόνας 5^{ος} - Ενοποίηση στατικών και δυναμικών δεδομένων. Ένα σύστημα επεξεργασίας ρευμάτων δεδομένων πρέπει να έχει την ικανότητα να μπορεί να αποθηκεύει αποτελεσματικά, να παρέχει πρόσβαση, και να μπορεί να τροποποιεί την κατάσταση των πληροφοριών, και να τις συνδυάζει με τα επίκαιρα δεδομένα που καταφθάνουν από τα ρεύματα. Για τη σωστή ενοποίηση των δεδομένων, το σύστημα πρέπει να μπορεί να χρησιμοποιεί μια ομοιόμορφη γλώσσα όταν επεξεργάζεται κάθε δεδομένο διαφορετικού τύπου.

Κανόνας 6^{ος} - Εγγύηση ασφάλειας και διαθεσιμότητας των δεδομένων. Το σύστημα οφείλει να εξασφαλίζει ότι οι εφαρμογές εκτελούνται και είναι διαθέσιμες, και ότι η ακεραιότητα των στοιχείων διατηρείται πάντα.

Κανόνας 7^{ος} - Αυτόματη κατανομή των εφαρμογών. Ένα σύστημα επεξεργασίας ρευμάτων δεδομένων πρέπει να μπορεί να κατανέμει την επεξεργασία των δεδομένων σε πολλαπλάσιους επεξεργαστές και τις μηχανές, ώστε να επιτυγχάνει αυξητικά μία κλιμακωσιμότητα. Η διανομή αυτή πρέπει να είναι αυτόματη και διαφανής.

Κανόνας 8^{ος} - Ακαριαία επεξεργασία και ανταπόκριση συστήματος. Ένα σύστημα επεξεργασίας ρευμάτων δεδομένων πρέπει να έχει μια ιδιαίτερα βελτιστοποιημένη, ελάχιστα-υπερφορτωμένη μηχανή εκτέλεσης ώστε να μπορεί να παραδίδει σε πραγματικό χρόνο τις απαντήσεις σε εφαρμογές που απαιτούν μεγάλο όγκο δεδομένων.

2.3 Ανεπάρκεια συμβατικών ΣΔΒΔ

Το μοντέλο για την επεξεργασία των δεδομένων που επικρατεί στα γνωστά Συστήματα Διαχείρισης Βάσεων Δεδομένων δεν μπορεί να ανταποκριθεί στις απαιτήσεις των εφαρμογών που κάνουν χρήση ρευμάτων δεδομένων. Σε αυτό το μοντέλο οι χρήστες «αντλούν» αποτελέσματα από τα υπάρχοντα δεδομένα στο σύστημα (*pull model*), διατυπώνοντας ερωτήματα στιγμιοτύπου (*one-time* ή *snapshot queries*) σε αυτά. Αντίθετα, στο νέο μοντέλο που διαμορφώνεται στις εφαρμογές ρευμάτων δεδομένων, τα δεδομένα δεν είναι από την αρχή αποθηκευμένα στατικά στο σύστημα, αλλά παράγονται διαρκώς νέα στοιχεία με ρυθμό τον οποίο καθορίζουν οι πηγές δεδομένων, πάνω στις οποίες ο χρήστης ή το σύστημα δεν μπορεί να επέμβει. Επίσης, τα ερωτήματα διαρκείας που διατυπώνονται στο σύστημα παραμένουν ενεργά για αρκετά μεγάλο χρονικό διάστημα και καλούνται να παρέχουν σωστά αποτελέσματα σε κάθε άφιξη στοιχείων από τα ρεύματα που εισέρχονται στο σύστημα.

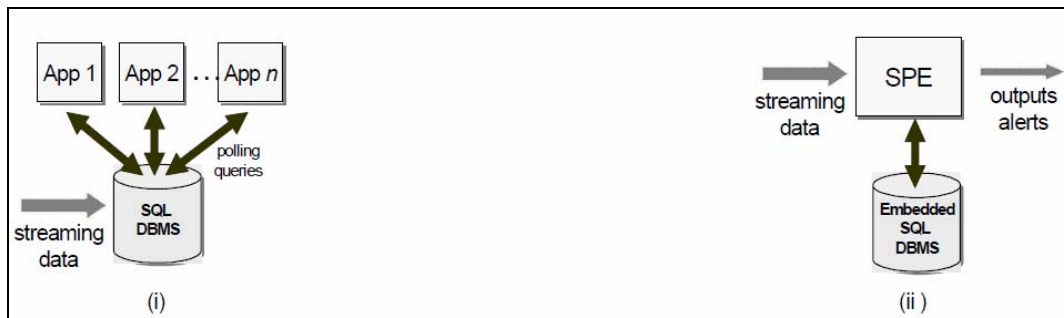
Συνεπώς, στο νέο μοντέλο, τα δεδομένα «σπρώχνουν» απαντήσεις στον χρήστη και στα ερωτήματα διαρκείας που ο ίδιος υπέβαλε (*push model*). Για χαμηλούς ρυθμούς άφιξης στοιχείων υπό την μορφή ρεύματος, τα συμβατικά ΣΔΒΔ θα αντιμετώπιζαν το πρόβλημα με την συντήρηση υλοποιημένων όψεων (*materialized views*). Ο χαμηλός ρυθμός άφιξης αποτελεί σημαντικό αλλά

απαραίτητο περιορισμό, μιας και σημαίνει λιγότερο συχνές ενημερώσεις των υλοποιημένων όψεων από το σύστημα. Οι περισσότερες όμως εφαρμογές παρουσιάζουν απαιτήσεις άμεσης επεξεργασίας των δεδομένων που λαμβάνουν σε πραγματικό χρόνο, επειδή συχνά έχουν μικρά περιθώρια απόκρισης στην δημιουργία κρίσιμων συνθηκών που σηματοδοτούνται από τα εξαγόμενα αποτελέσματα. Έτσι δεν είναι αποδεκτή η λύση των υλοποιημένων όψεων, όπου ο χρόνος απόκρισης του συστήματος μπορεί να είναι αρκετά μεγάλος ανάλογα με την χρονική στιγμή εμφάνισης χαμηλού φόρτου εργασίας. Μάλιστα τα επιδιωκόμενα περιθώρια απόκρισης του συστήματος μπορεί να είναι τόσο μικρά σε χρονική διάρκεια, ώστε το κόστος αποθήκευσης της πληροφορίας σε σκληρούς δίσκους να είναι πολύ υψηλό. Επιπλέον η απαίτηση άμεσης επεξεργασίας σε πραγματικό χρόνο, προσανατολίζει τον σχεδιασμό του συστήματος στην χρήση κατά το δυνατό περισσότερο της κύριας μνήμης, το οποίο εναντιώνεται στον σχεδιασμό των συμβατικών ΣΔΒΔ.

Η παρουσία ερωτημάτων διαρκείας αλλάζει και τον τρόπο με τον οποίο επιχειρείται η βελτιστοποίηση των ερωτημάτων. Στο μοντέλο που καθιερώνουν τα ρεύματα δεδομένων πρέπει να λαμβάνονται υπόψη τα ερωτήματα διαρκείας, τα οποία παραμένουν ενεργά στο σύστημα για συγκεκριμένο χρονικό διάστημα. Κατά συνέπεια, γίνεται αντιληπτό ότι δεν είναι αρκετή οποιαδήποτε προσπάθεια προσαρμογής των συμβατικών ΣΔΒΔ στις νέες απαιτήσεις, αφού ο σχεδιασμός τους πραγματοποιήθηκε με στόχο την αντιμετώπιση διαφορετικών συνθηκών, ενώ παράλληλα αρκετές από τις θεμελιώδεις αρχές που επικρατούν σε αυτά τα συστήματα έρχονται σε αντίφαση με το μοντέλο της επεξεργασίας των εισερχόμενων δεδομένων σε πραγματικό χρόνο.

Συνοψίζοντας, τα συστήματα διαχείρισης βάσεων δεδομένων (DBMSs) χρησιμοποιούνται ευρέως λόγω της δυνατότητάς τους να αποθηκεύουν μεγάλα σύνολα δεδομένων και της αποτελεσματικότητάς τους σε ερωτήματα που θέτουν οι χρήστες. Στο σχήμα 2.2(i) εξηγείται η βασική αρχιτεκτονική των ΣΔΒΔ. Τα δεδομένα εισέρχονται στο ΣΔΒΔ άμεσα ή μέσω μιας εφαρμογής. Μια συλλογή εφαρμογών μπορεί έπειτα να χειριστεί τα δεδομένα. Ο χρήστης μπορεί να χρησιμοποιήσει αυτές τις προ-εγκατεστημένες εφαρμογές και μπορεί επίσης να κωδικοποιήσει πρόσθετες σε μια γλώσσα γενικού σκοπού όπως η C++ ή η Java, χρησιμοποιώντας την ενσωματωμένη SQL του συστήματος. Αντίθετα οι μηχανές επεξεργασίας ρευμάτων (SPEs) σχεδιάζονται συγκεκριμένα για την διαχείριση ρευμάτων δεδομένων. Η βασική αρχιτεκτονική τους παρουσιάζεται στο σχήμα 2.2(ii). Οι μηχανές επεξεργασίας ρευμάτων εκτελούν την επεξεργασία μέσω SQL στα εισερχόμενα στοιχεία των ρευμάτων, χωρίς απαραίτητως να τα αποθηκεύουν.

Σαφώς, για να αποθηκευτεί από τον χρήστη μία ορισμένη κατάσταση όταν χρειάζεται, μπορεί να χρησιμοποιηθεί μια συμβατική βάση δεδομένων SQL η οποία είναι ενσωματωμένη στο σύστημα για περισσότερη αποδοτικότητα. Οι μηχανές επεξεργασίας ρευμάτων χρησιμοποιούν ειδικούς τελεστές (π.χ. χρονικά παράθυρα) για να εκφράζουν τη λογική επεξεργασίας τους προς τα ρεύματα.



Σχήμα 2.2: Βασική αρχιτεκτονική των συστημάτων διαχείρισης βάσεων δεδομένων και των μηχανών επεξεργασίας ρευμάτων. (Πηγή: [SCZ05])

Αντίστοιχα, στον πίνακα 2.1 που ακολουθεί, συνοψίζονται επιγραμματικά τα κυριότερα χαρακτηριστικά των συστημάτων διαχείρισης ρευμάτων δεδομένων σε αντιπαράθεση με τα χαρακτηριστικά των συμβατικών συστημάτων διαχείρισης βάσεων δεδομένων.

| Σύστημα διαχείρισης βάσεων δεδομένων | Σύστημα διαχείρισης ρευμάτων δεδομένων |
|--|---|
| Στατικές σχέσεις | Εφήμερα ρεύματα δεδομένων |
| Ερωτήματα στιγμιότυπου | Ερωτήματα διάρκειας |
| Τυχαία προσπέλαση πλειάδων | Διαδοχική προσπέλαση πλειάδων |
| «Απεριόριστος» αποθηκευτικός χώρος | Πεπερασμένη ποσότητα κύριας μνήμης |
| Μόνο το τρέχον στιγμιότυπο της βάσης δεδομένων έχει σημασία | Η σειρά άφιξης των πλειάδων στο σύστημα έχει σημασία |
| Ο χώρος όπου τηρούνται τα δεδομένα είναι «παθητικός» (απλώς για την αποθήκευση). (pull-model) | Τα πρόσφατα δεδομένα τηρούνται στη μνήμη, και η άφιξή τους ενεργοποιεί τα ερωτήματα. (push model) |
| Σχετικά χαμηλός ρυθμός αλλαγών στα δεδομένα, έπειτα από ενέργειες των χρηστών | Πιθανόν να καταφθάνουν ραγδαία ογκώδη δεδομένα (πολλά GB από τις πηγές δεδομένων) |
| Τα δεδομένα θεωρούνται ακριβή | Τα δεδομένα πιθανόν να είναι παρωχημένα ή να περιέχουν ανακρίβειες και ελλείψεις |
| Η απάντηση δεν απαιτείται σε πραγματικό χρόνο, αλλά θα είναι ακριβής | Η απάντηση απαιτείται σε πραγματικό χρόνο, έστω και προσεγγιστικά |
| Η εκτέλεση των προσχεδίων ερωτημάτων εξαρτάται από τον επεξεργαστή και το φυσικό σχεδιασμό της βάσης δεδομένων | Ο ρυθμός άφιξης των δεδομένων και τα χαρακτηριστικά τους θεωρούνται μεταβαλλόμενα και μη προβλέψιμα |

Πίνακας 2.1: Αντιπαραβολή χαρακτηριστικών μεταξύ Συστημάτων Διαχείρισης Βάσεων Δεδομένων και Ρευμάτων Δεδομένων. (Πηγή: [Pat03])

2.4 Ερωτήματα σε ρεύματα δεδομένων

Τα ερωτήματα τα οποία τίθενται σε συστήματα διαχείρισης ρεύματα δεδομένων παρουσιάζουν αρκετές ομοιότητες με τα αντίστοιχα σε συμβατικά ΣΔΒΔ. Ωστόσο, υπάρχουν δύο χαρακτηριστικές διακρίσεις. Η πρώτη διάκριση αφορά σε ερωτήματα που τίθενται μόνο μια φορά και σε ερωτήματα τα οποία εκτελούνται για κάποιο ορισμένο χρονικό διάστημα [BBD+02]. Ουσιαστικά διαχωρίζονται σε ερωτήματα στιγμιότυπου (one-time queries) και ερωτήματα διαρκείας (continuous queries).

Ερωτήματα στιγμιότυπου (snapshot ή one-time queries). Η κατηγορία περιλαμβάνει ερωτήματα που τίθενται σε παραδοσιακά ΣΔΒΔ και πρόκειται για ερωτήματα τα οποία αξιολογούνται μία φορά πάνω σε ένα στιγμιότυπο του συνόλου των δεδομένων που περιέχει η βάση δεδομένων, με την απάντηση να επιστρέφεται στο χρήστη του συστήματος. Τέτοια είναι τα ερωτήματα που συνέχεια διεκπεραιώνονται από τις συμβατικές ΒΔ, όπου ο χρήστης πρέπει να ενεργήσει ώστε να αντλήσει τις πληροφορίες που τον ενδιαφέρουν (pull model).

Ερωτήματα διαρκείας (continuous queries). Είναι κατηγορία που περιλαμβάνει τα ερωτήματα που τίθενται σε συστήματα διαχείρισης ρευμάτων δεδομένων και πρόκειται για ερωτήματα των οποίων οι απαντήσεις υπολογίζονται συνεχώς όσο τα δεδομένα των ρευμάτων συνεχίζουν να καταφθάνουν στο σύστημα. Αυτού του είδους τα ερωτήματα εκτελούνται μονίμως επί μεγάλο χρονικό διάστημα, μέχρι ο χρήστης να τα αποσύρει ή να αναστείλει τον υπολογισμό τους. Το ίδιο το σύστημα αναλαμβάνει να προωθήσει τα νέα αποτελέσματα προς το χρήστη (push model), όποτε αυτά γίνουν διαθέσιμα, χωρίς εκείνος να είναι υποχρεωμένος να υποβάλλει αλληπάλληλα το ίδιο ερώτημα. Τα ερωτήματα διαρκείας είναι η πιο ενδιαφέρουσα κατηγορία ερωτημάτων στα ρεύματα δεδομένων. Η απάντηση σε ένα ερώτημα διαρκείας παράγεται κατά τη διάρκεια του χρόνου, απεικονίζοντας πάντα τα δεδομένα των ρευμάτων που έχουν εισέλθει στο σύστημα μέχρι εκείνη την χρονική στιγμή. Οι απαντήσεις των ερωτημάτων διαρκείας μπορούν να αποθηκεύονται και να ενημερώνονται μέχρι να εισέλθουν νέα δεδομένα στο σύστημα ή μπορούν να παράγονται ως ρεύματα δεδομένων οι ίδιες. Για παράδειγμα, τα ερωτήματα συνάθροισης (aggregation queries) μπορούν να περιλαμβάνουν συχνές αλλαγές στις πλειάδες των απαντήσεων, υπαγορεύοντας την προσέγγιση της αποθήκευσης των απαντήσεων, ενώ τα ερωτήματα σύνδεσης (join queries) είναι μονοτονικά και μπορούν να παράγουν γρήγορες και απεριόριστες απαντήσεις, υπαγορεύοντας την προσέγγιση οι απαντήσεις να παράγονται ως ρεύματα.

Η δεύτερη διάκριση αφορά σε ερωτήματα που είναι γνωστά εκ των προτέρων και σε ερωτήματα τα οποία τίθενται όταν έχουν ήδη εισέλθει στο σύστημα ορισμένα δεδομένα [BBD+02]. Ειδικότερα, υπάρχουν:

Προκαθορισμένα ερωτήματα (predefined queries). Ένα προκαθορισμένο ερώτημα είναι εκείνο που παρέχεται στο σύστημα διαχείρισης ρευμάτων δεδομένων προτού φθάσουν οποιαδήποτε σχετικά δεδομένα από τα ρεύματα στο σύστημα. Τα προκαθορισμένα ερωτήματα είναι γενικά ερωτήματα διαρκείας, αν και προγραμματισμένα ερωτήματα στιγμιότυπου μπορούν επίσης να προκαθοριστούν.

Μη προβλέψιμα ερωτήματα (ad hoc queries). Τα μη προβλέψιμα ερωτήματα, είναι εκείνα που εκδίδονται σε πραγματικό χρόνο, αφού έχουν αρχίσει ήδη να εισέρχονται στο σύστημα δεδομένα από τα ρεύματα. Μπορούν να είναι είτε ερωτήματα στιγμιότυπου είτε ερωτήματα διαρκείας. Επίσης τέτοιου είδους ερωτήματα περιπλέκουν τον σχεδιασμό ενός συστήματος διαχείρισης ρευμάτων δεδομένων, επειδή κυρίως δεν είναι γνωστά εκ των προτέρων, έτσι ώστε να ληφθούν υπόψη κατά το στάδιο της βελτιστοποίησης ερωτημάτων (query optimization), της ταυτοποίησης των κοινών υπο-εκφράσεων μεταξύ των ερωτημάτων, κ.λπ. Επιπλέον, είναι πιθανό η σωστή απάντηση σε ένα μη προβλέψιμο ερώτημα να απαιτεί αναφορά σε δεδομένα τα οποία που είχαν φθάσει ήδη στο σύστημα μέσω των ρευμάτων, δηλαδή ιστορικά στοιχεία, που ενδεχομένως να έχουν πλέον απορριφθεί.

2.4.1 Ερωτήματα διαρκείας βάσει του χρόνου ή του περιεχομένου

Τα ερωτήματα διαρκείας επιβάλλεται να ανταποκρίνονται είτε σε κάθε μεταβολή των στοιχείων (change based) όποτε αυτά γίνονται διαθέσιμα, είτε να εκτελούνται ανά τακτά χρονικά διαστήματα (time-based) όπως τα προσδιορίζει ο χρήστης. Η πρώτη κατηγορία ερωτημάτων, πλεονεκτεί ως προς τον χρόνο απόκρισης, αλλά ενδεχομένως δεσμεύει πολύτιμους πόρους του συστήματος όταν δεν χρειάζονται άμεσες απαντήσεις. Από την άλλη πλευρά, όταν τα ερωτήματα διαρκείας εκτελούνται ανά τακτά χρονικά διαστήματα, το σύστημα μπορεί να τα υποστηρίξει περισσότερο αποτελεσματικά, εμφανίζοντας δυνατότητες κλιμάκωσης (scalability). Βεβαίως, παρουσιάζονται προβλήματα στο ζήτημα της βελτιστοποίησης των ερωτημάτων όταν αυτά χαρακτηρίζονται από διαφορετικές συχνότητες εκτέλεσης ή όταν τα χρονικά διαστήματα είναι αλληλεπικαλυπτόμενα.

2.4.2 Ερωτήματα συνάθροισης σε ρεύματα δεδομένων

Συνήθως η πληροφορία που καταφτάνει μέσω των ρευμάτων δεδομένων είναι αρκετά λεπτομερής και πολλές φορές σε βαθμό που δεν ενδιαφέρει όλους τους χρήστες. Για παράδειγμα, εάν ένας αισθητήρας στέλνει τις μετρήσεις για την θερμοκρασία κάθε μισό λεπτό, είναι πολύ πιθανόν να μην υπάρχουν μεταβολές στην τιμή μεταξύ των διαδοχικών τιμών της θερμοκρασίας. Αντιθέτως, περισσότερο ενδιαφέρον και σημασία έχουν κάποια στατιστικά στοιχεία, όπως για παράδειγμα η μέση ωριαία θερμοκρασία ή η μέγιστη τιμή που παρατηρήθηκε τις τελευταίες 24 ώρες. Επομένως, τα ερωτήματα συνάθροισης αποτελούν σπουδαίο εργαλείο για την επεξεργασία των ρευμάτων, αφού παρέχουν την τάση μεταβολής των στοιχείων τους. Αυτά τα ερωτήματα κάνουν χρήση ειδικών τελεστών συνάθροισης, όπως COUNT(), AVG(), MAX(), MIN(), SUM(), κ.λπ. Στον πίνακα 2.2 που ακολουθεί, φαίνεται η επεξήγηση των πιο σημαντικών τελεστών συνάθροισης.

| Λειτουργία | Περιγραφή |
|----------------------------------|--|
| AVG() | Μέσος όρος των τιμών σε μια στήλη. Η στήλη μπορεί να περιέχει μόνο αριθμητικά στοιχεία. |
| COUNT() , COUNT(*) | Το πλήθος των τιμών σε μια στήλη (εάν διευκρινίζεται το όνομα της) ή όλων των σειρών σε έναν πίνακα ή μια ομάδα (εάν διευκρινίζετε *). Το COUNT() αγνοεί NULL τιμές, αλλά το COUNT(*) τις περιλαμβάνει στο πλήθος. |
| MAX() | Υψηλότερη τιμή σε μια στήλη (τελευταίο γράμμα αλφαριθμητικά για τους τύπους στοιχείων κειμένων). Αγνοεί τις NULL τιμές. |
| MIN() | Χαμηλότερη τιμή σε μια στήλη (πρώτο γράμμα αλφαριθμητικά για τους τύπους στοιχείων κειμένων). Αγνοεί τις NULL τιμές. |
| SUM() | Άθροισμα των τιμών σε μια στήλη. Η στήλη μπορεί να περιέχει μόνο αριθμητικά στοιχεία. |

Πίνακας 2.2: Επεξήγηση τελεστών συνάθροισης.

Όσον αφορά τα ερωτήματα συνάθροισης, ενδιαφέρον κυρίως παρουσιάζουν: (α) τα ερωτήματα συνάθροισης για μεμονωμένα στοιχεία (*point aggregates*), στα οποία ζητούνται τα συγκεντρωτικά αποτελέσματα για ένα συγκεκριμένο στοιχείο που καταγράφεται στο ρεύμα, π.χ. «Να υπολογιστεί η μέση ταχύτητα κίνησης του οχήματος #X τις τελευταίες 5 ώρες» και (β) τα ερωτήματα συνάθροισης για σειρά στοιχείων (*range aggregates*), στα οποία ομαδοποιούνται τα δεδομένα σε ομάδες οι οποίες προσδιορίζονται από το γεγονός ότι κάποια χαρακτηριστικά τους κινούνται σε ένα ορισμένο διάστημα τιμών, π.χ. «Ζητείται ο συνολικός αριθμός των οχημάτων τα οποία εισήλθαν στον δακτύλιο της Αθήνας σε ένα καθορισμένο χρονικό διάστημα».

2.5 Παράθυρα σε ρεύματα δεδομένων

Ένα ΣΔΒΔ είναι αδύνατον να μπορεί να αποθηκεύσει και να λάβει υπόψη του όλη την ιστορική εξέλιξη των ρευμάτων εισόδου. Δεδομένου ότι συχνά καταφεύγει σε προσεγγιστικές απαντήσεις των ερωτημάτων, δεν είναι απαραίτητο να εξετάσει το σύνολο των πλειάδων που έχει περάσει ως εκείνη τη στιγμή από το σύστημα, αλλά μάλλον μόνο ένα συγκεκριμένο τμήμα τους, εντός ενός παραθύρου επί των πλέον πρόσφατων στοιχείων. Έτσι, σε κάθε χρονική στιγμή, το παράθυρο περιλαμβάνει ένα σύνολο διαδοχικών πλειάδων επί των οποίων θα εφαρμοστεί το ερώτημα. Η οριοθέτηση των χρονικών παραθύρων γίνεται με τη χρήση κατάλληλων χρονοσήμων στις πλειάδες του ρεύματος. Κάθε πλειάδα συνοδεύεται από μια χρονική ένδειξη η οποία συνήθως δηλώνει τη χρονική στιγμή παραγωγής της. Το σύστημα οργανώνει τις πλειάδες που λαμβάνει με βάση το χρονόσημό τους (π.χ. σε αύξουσα σειρά), σχηματίζοντας έτσι το παράθυρο.

Το εύρος του παραθύρου σχηματίζεται είτε με βάση τη διάσταση του χρόνου από τα χρονόσημα (π.χ. πλειάδες των τελευταίων 5 λεπτών) είτε με βάση το πλήθος των πλειάδων. Είναι σαφές ότι στην πρώτη περίπτωση δεν είναι

προκαθορισμένο το μέγεθος μνήμης που απαιτείται σε αντίθεση με τη δεύτερη όπου διατηρείται σταθερό το πλήθος των πλειάδων ανεξαρτήτως εύρους των χρονοσήμων.

Συνεπώς, η χρήση των παραθύρων στα ερωτήματα διαρκείας συνίσταται στην απόσπαση πεπερασμένου πλήθους στοιχείων από το ρεύμα δίνοντας πρόσβαση διαρκώς σε περιορισμένο αλλά σαφώς προσδιορισμένο τμήμα του ρεύματος. Τα περιεχόμενά τους τοποθετούνται χρονικά κοντά στα πιο πρόσφατα στοιχεία των ρευμάτων, ενώ επαναπροσδιορίζονται με κάθε νέα πλειάδα που φτάνει σε αυτά, ανάλογα με τους κανόνες και τις παραμέτρους κάθε παραθύρου. Υπάρχουν διάφορα είδη παραθύρων πάνω σε ένα ρεύμα δεδομένων, με το ενδιαφέρον να στρέφεται κυρίως στα παράθυρα σταθερής αρχής ή παράθυρα ορόσημου (*landmark windows*) και στα κυλιόμενα παράθυρα (*sliding windows*).

Τα παράθυρα αποτελούν πολύ σημαντικό και αναπόσπαστο κομμάτι της έκφρασης ερωτημάτων προς το σύστημα και είναι αδύνατον να απαντηθούν ερωτήματα τα οποία αναφέρονται σε όλη την ιστορία του ρεύματος. Γενικά η έμφαση στις περισσότερες εφαρμογές δίνεται στο πρόσφατο παρελθόν και όχι στο απώτερο. Με την εισαγωγή παραθύρων, ο ίδιος ο χρήστης της εφαρμογής αποφασίζει την εμβέλεια του ερωτήματος η οποία μπορεί να παραβληθεί με τον παρελθοντικό ορίζοντα του παραθύρου αν χρειάζεται.

Οι παραθυρικοί τελεστές απαντώνται σε ερωτήματα διαρκείας με στόχο να τροφοδοτήσουν προβληματικούς τελεστές που χρησιμοποιούν όλα τα δεδομένα εισόδου τους για την επιστροφή πλήρους απάντησης, όπως π.χ. οι συναδροιστικοί τελεστές πλήθους (COUNT), αθροίσματος (SUM), μέσου όρου (AVG) κ.τ.λ. Όμως, δεν είναι ούτε εφικτή ούτε επιθυμητή, η διατήρηση και η επεξεργασία όλων των δεδομένων. Η τροφοδότηση των προβληματικών τελεστών με παράθυρα έχει ως αποτέλεσμα την απεμπλοκή τους, με αντίτιμο την αποδοχή προσεγγίσεων στις επιστρεφόμενες απαντήσεις. Οι τελεστές αυτοί επεξεργάζονται πλέον τα στοιχεία που περιέχονται σε κάθε παράθυρο, εστιάζοντας και αυτοί με την σειρά τους στην πρόσφατη πληροφορία.

2.5.1 Παράθυρα οροσήμου

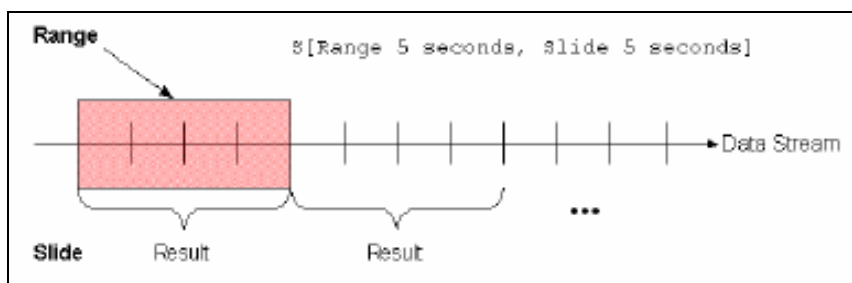
Τέτοιας μορφής παράθυρα διατηρούν πάντοτε σταθερό ένα από τα δύο άκρα τους, «σημαδεύοντας» αυθαίρετα ένα χρονικό σημείο ενδιαφέροντος. Το πιο σύνηδες είναι να έχουν ως σταθερή αφετηρία κάποιο χρονόσημο, αλλά το πέρας τους παρακολουθεί τη χρονική εξέλιξη των πλειάδων του ρεύματος. Άρα, το νεότερο άκρο του παραθύρου προχωρεί παράλληλα με το χρόνο, ταυτιζόμενο με την παρούσα χρονική στιγμή, ώστε να καλύπτει συνεχώς την έλευση νέων στοιχείων και το εύρος του παραθύρου αυξάνεται διαρκώς όπως και ο αριθμός των πλειάδων που περιλαμβάνει. Επομένως τα περιεχόμενα αυτών των παραθύρων, προσδιορίζονται βάσει χρονοσήμου, ενώ το βήμα τους είναι μοναδιαίο σε επίπεδο πλειάδας. Για να την εφαρμογή τους σε ένα ερώτημα διαρκείας χρειάζεται πρώτα να οριστεί σαφώς το χρονικό διάστημα εφαρμογής, δηλαδή η εμβέλεια του παραθύρου.

2.5.2 Κυλιόμενα παράθυρα

Πρόκειται για έναν σχετικά πολύπλοκο τύπο παραθύρων, ο οποίος εφαρμόζεται συχνά στις εφαρμογές ρευμάτων δεδομένων. Τα περιεχόμενά τους αναφέρονται σε συγκεκριμένο χρονικό διάστημα, ενώ γενικότερα τα παράθυρα αυτού του τύπου εμφανίζουν τις ακόλουθες ιδιότητες: (α) τα άκρα τους είναι και τα δύο μεταβλητά, δηλαδή έχουν αφετηρία και πέρασ που κινούνται ταυτόχρονα παρακολουθώντας την χρονική εξέλιξη των στοιχείων που συρρέουν σύστημα, (β) τα περιεχόμενά τους προσδιορίζονται βάσει χρονοσήμου, έτσι παλαιότερα δεδομένα απορρίπτονται και καινούργια εισέρχονται με κυμαινόμενο ρυθμό και (γ) το βήμα μπορεί να είναι είτε μοναδιαίο είτε κατ' άλματα, πάντοτε όμως σε επίπεδο χρονοσήμου. Επομένως το εύρος αυτών των παραθύρων παραμένει σταθερό, αλλά το πλήθος των πλειάδων και τα περιεχόμενα που βρίσκονται εντός δεν διατηρούνται αμετάβλητα. Για τον ορισμό τους χρησιμοποιούνται οι εξής παράμετροι:

- ✓ Παράμετρος εύρους (RANGE), με την οποία καθορίζεται η μέγιστη διαφορά χρονοσήμων που μπορεί να εμφανίζεται στις περιεχόμενες πλειάδες του παραθύρου.
- ✓ Παράμετρος βήματος ή κύλισης (SLIDE), η οποία προσδιορίζει τον ρυθμό με τον οποίο ανανεώνονται τα περιεχόμενά του.
- ✓ Παράμετρος χρόνου έναρξης εφαρμογής του παραθύρου (START AT), με την οποία καθορίζεται η χρονική στιγμή που θα ξεκινήσει το παράθυρο να επενεργεί.

Όπως φαίνεται στο σχήμα 2.3, οι παράμετροι RANGE και SLIDE καθορίζουν πόσα δεδομένα περιλαμβάνονται στο παράθυρο, και συγκεκριμένα η παράμετρος RANGE καθορίζει τη χρονική περίοδο (π.χ. 5 δευτερόλεπτα) των δεδομένων που κοιτάζει το παράθυρο για επεξεργασία, ενώ η παράμετρος SLIDE καθορίζει τη χρονική περίοδο (π.χ. 5 δευτερόλεπτα) που το παράθυρο κυλάει και δίνει τα αποτελέσματα.



Σχήμα 2.3: Γενικό παράδειγμα εφαρμογής παραμέτρων RANGE και SLIDE.
(Πηγή: [CQL09])

Τα κυλιόμενα παράθυρα μπορούν να διακριθούν περαιτέρω σε τρεις κατηγορίες: (α) χρονικά παράθυρα (*Time - based windows*), (β) παράθυρα πλειάδων (*Tuple - based windows*) και (γ) μεριστικά παράθυρα (*Partitioned windows*).

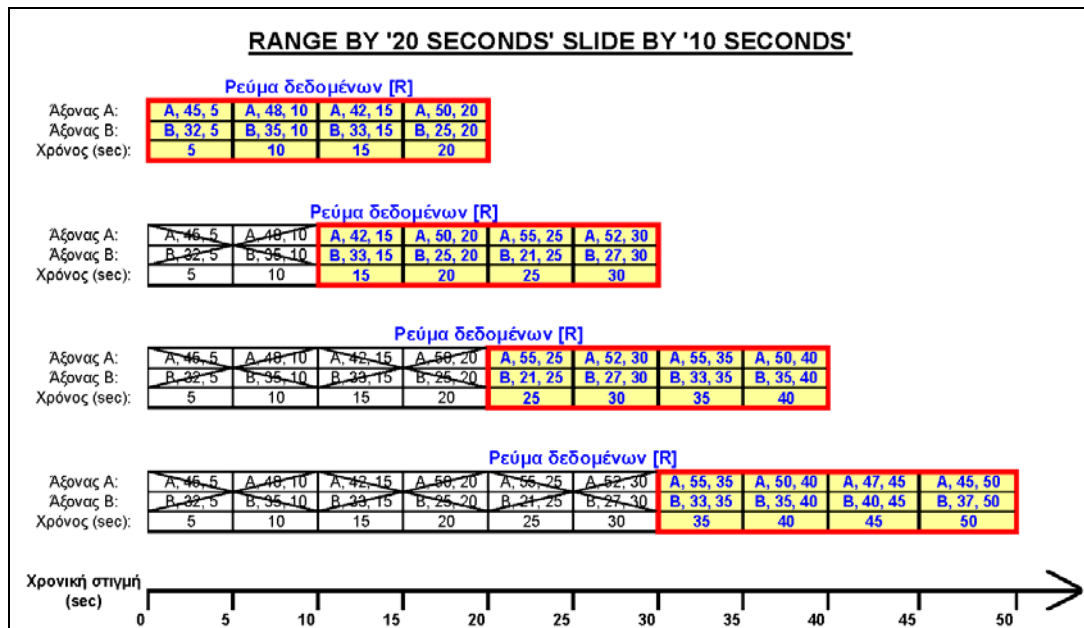
Χρονικά παράθυρα. Εφαρμόζονται πάνω στα περιεχόμενα του ρεύματος, παίρνουν ως παράμετρο ένα συγκεκριμένο χρονικό διάστημα T και καθορίζονται πάνω στο ρεύμα από αυτήν την παράμετρο εύρους (**RANGE T**). Διαισθητικά, αυτό το παράθυρο καθορίζει τη σχέση παραγωγής του κατά τη διάρκεια του χρόνου με την ολίσθηση ενός χρονικού διαστήματος μεγέθους T χρονικών στιγμών πάνω στο ρεύμα εισόδου. Αν τυχόν $T = 1$, το παράθυρο αναφέρεται μόνο στις πλειάδες με το πιο πρόσφατο χρονόσημο, ενώ αν στο ερώτημα χρειαστεί να γίνει αναφορά σε όλες τις πλειάδες του ρεύματος τότε $T = \infty$.

Τα παράθυρα αυτά παρουσιάζουν τις εξής χαρακτηριστικές ιδιότητες: (α) τα άκρα τους είναι και τα δύο μεταβλητά, (β) χρησιμοποιούν ως βάση προσδιορισμού των περιεχομένων τους ένα χρονικό διάστημα και (γ) το βήμα τους είναι μοναδιαίο σε επίπεδο χρόνου. Τα στοιχεία εντός του παραθύρου υπολογίζονται ως εξής: ξεκινώντας από την παρούσα χρονική στιγμή και προχωρώντας παράλληλα με την εξέλιξη του χρόνου, επιλέγονται συνεχώς οι πλειάδες του ρεύματος με χρονόσημο εντός του εύρους του καθορισμένου χρονικού διαστήματος του παραθύρου (παράμετρος εύρους - **RANGE**), ενώ το παράθυρο μετακινείται κάθε φορά με βήμα μοναδιαίο στην επιλεγμένη μονάδα του χρόνου ή σύμφωνα με ένα καθορισμένο χρονικό διάστημα δηλαδή τον ρυθμό ανανέωσης των περιεχομένων του παραθύρου (παράμετρος κύλισης - **SLIDE**).

Παράδειγμα. Έστω ότι υπάρχει ένα ρεύμα δεδομένων (**R**), το οποίο μεταφέρει την εξής πληροφορία για δύο οδικούς άξονες: τον κωδικό κάθε άξονα, τον οδικό φόρτο κάθε άξονα και το χρονόσημο εκπομπής των δεδομένων, λ.χ. κάθε 5 δευτερόλεπτα. Σύμφωνα με τα δεδομένα τίθεται το εξής ερώτημα: «Βρες τον μέσο φόρτο των δύο αξόνων κάθε 10 δευτερόλεπτα, εξετάζοντας δεδομένα περιόδου 20 δευτερολέπτων, αρχίζοντας από την χρονική στιγμή '2010-01-12 12:00:00'».

```
SELECT AxisID, AVG(R.Load) AS Axis_Load
FROM Axes_load AS R [RANGE BY '20 SECONDS'
SLIDE BY '10 SECONDS'
START AT '2010-01-12 12:00:00']
GROUP BY R.AxisID;
```

Όπως φαίνεται και από το σχήμα 2.4, το ερώτημα διαρκείας το οποίο εκτελείται πάνω στο ρεύμα **R**, επιστρέφει κάθε 10 δευτερόλεπτα τον μέσο όρο του φόρτου (**AVG**) καθενός οδικού άξονα, ο οποίος όμως υπολογίζεται από δεδομένα περιόδου 30 δευτερολέπτων, ξεκινώντας την εφαρμογή του χρονικού παραθύρου από το καθορισμένο χρονόσημο ('2010-01-12 12:00:00').

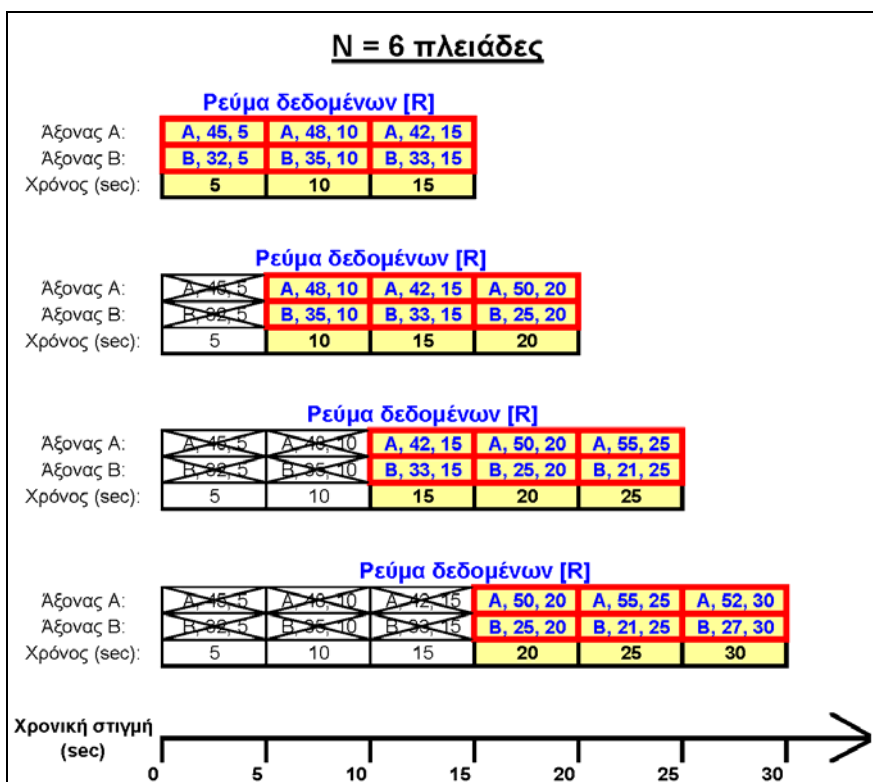


Σχήμα 2.4: Παράδειγμα χρονικού παραθύρου, με RANGE = 20sec και SLIDE = 10sec.

Παράθυρα πλειάδων. Τέτοια παράθυρα ανά πάσα χρονική στιγμή εστιάζουν στις N τελευταίες πλειάδες του ρεύματος. Τα συγκεκριμένα παράθυρα παρουσιάζουν τις εξής ιδιότητες: (α) τα άκρα τους είναι και τα δύο μεταβλητά, (β) χρησιμοποιούν ως βάση προσδιορισμού των περιεχομένων τους το πλήθος των πλειάδων και (γ) το βήμα τους είναι μοναδιαίο σε επίπεδο πλειάδας. Τα στοιχεία εντός του παραθύρου υπολογίζονται ως εξής: ξεκινώντας από την παρούσα χρονική στιγμή και προχωρώντας προς το παρελθόν (αντίθετα προς την εξέλιξη του χρόνου), επιλέγονται συνεχώς πλειάδες μέχρι να καλυφθεί ο ζητούμενος αριθμός των N πλειάδων. Ενδέχεται βέβαια το πλήθος των πλειάδων με ίδιο χρονόσημο να υπερβαίνει ή ακόμη και να υπολείπεται του αριθμού των πλειάδων N που προκαθορίστηκε, οπότε η επιλογή θα πρέπει να γίνει κατά μη ντετερμινιστικό τρόπο.

Παράδειγμα. Έστω ότι υπάρχει ένα ρεύμα δεδομένων (R), το οποίο μεταφέρει την εξής πληροφορία για δύο οδικούς άξονες: τον κωδικό κάθε άξονα, τον οδικό φόρτο κάθε άξονα και το χρονόσημο εκπομπής των δεδομένων, λ.χ. κάθε 5 δευτερόλεπτα. Επίσης κάθε 5 δευτερόλεπτα έρχονται δύο πλειάδες κάθε φορά, μία για τον οδικό άξονα A και μία για τον οδικό άξονα B. Σύμφωνα με τα δεδομένα τίθεται το εξής ερώτημα: «Βρες τον μέσο φόρτο των δύο αξόνων, εξετάζοντας συνολικά δεδομένα $N = 6$ πλειάδων».

```
SELECT AxisID, AVG(R.Load) AS Axis_Load
FROM Axes_load AS R [ROWS '6']
GROUP BY R.AxisID;
```

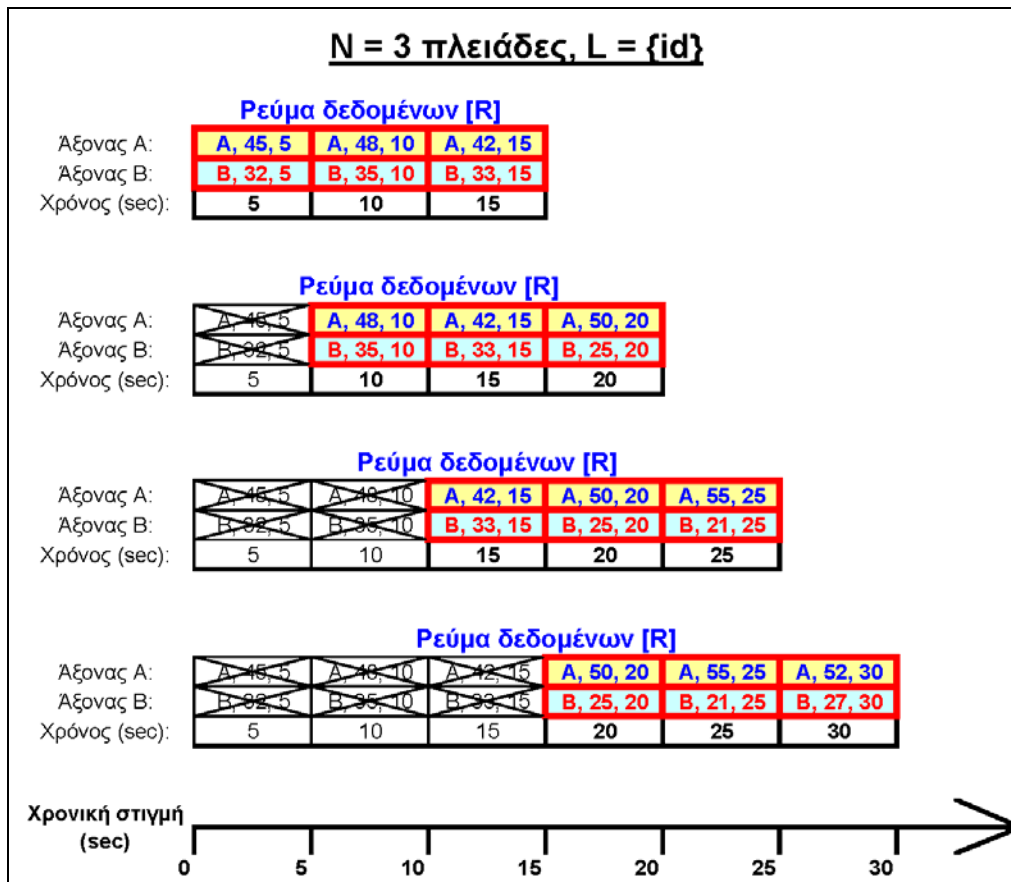


Σχήμα 2.5: Παράδειγμα παραδύρου πλειάδων, με $N = 6$ πλειάδες.

Όπως φαίνεται και από το σχήμα 2.5, το ερώτημα διαρκείας το οποίο εκτελείται πάνω στο ρεύμα R , επιστρέφει τον μέσο όρο του φόρτου (AVG) κάθε ενός οδικού άξονα, ο οποίος όμως υπολογίζεται από συνολικά δεδομένα 6 πλειάδων και στην προκειμένη περίπτωση 3 πλειάδες για τον οδικό άξονα A και 3 πλειάδες για τον οδικό άξονα B.

Μεριστικά παράθυρα. Διαχωρίζουν το ρεύμα σε υπορεύματα βάσει της λίστας γνωρισμάτων ομαδοποίησης $L = \{A_1, A_2, \dots, A_n\}$. Για κάθε υπορεύμα, υπολογίζεται ένα κυλιόμενο παράθυρο μεγέθους N πλειάδων και τα επιμέρους αποτελέσματα συνενώνονται για να παραχθούν τα τελικά περιεχόμενα του παραδύρου. Ουσιαστικά, κάποια πλειάδα του ρεύματος με συγκεκριμένες τιμές για τα γνωρίσματα ομαδοποίησης θα συμμετέχει στο παράθυρο, μόνον εφόσον το χρονόσημο την κατατάσσει μεταξύ των πλέον πρόσφατων πλειάδων της ομάδας με ίδιες τιμές στα συγκεκριμένα γνωρίσματα. Αξίζει να σημειωθεί ότι μετά την ομαδοποίηση δεν εφαρμόζεται κάποια συνάρτηση συνάθροισης (π.χ. SUM, AVG κ.ά.) όπως στη σχεσιακή άλγεβρα, αλλά απλώς καταμετρώνται οι πλειάδες κάθε ομάδας.

Τα παράθυρα αυτά παρουσιάζουν τις εξής χαρακτηριστικές ιδιότητες: (α) τα άκρα τους είναι και τα δύο μεταβλητά, (β) χρησιμοποιούν ως βάση προσδιορισμού των περιεχομένων τους το πλήθος των πλειάδων και (γ) το βήμα τους είναι μοναδιαίο σε επίπεδο πλειάδας. Τα στοιχεία εντός του παραδύρου υπολογίζονται ως εξής: ξεκινώντας από την παρούσα χρονική στιγμή και προχωρώντας προς το παρελθόν, (αντίθετα προς την εξέλιξη του χρόνου), επιλέγονται συνεχώς για κάθε διαφορετική τιμή της λίστας γνωρισμάτων ομαδοποίησης L οι πιο πρόσφατες N πλειάδες.



Σχήμα 2.6: Παράδειγμα μεριστικού παραθύρου, με $N = 3$ πλειάδες για κάθε ομάδα που ορίζεται από το γνώρισμα $L = \{id\}$.

Παράδειγμα. Έστω ότι υπάρχει ένα ρεύμα δεδομένων (R), το οποίο μεταφέρει την εξής πληροφορία για δύο οδικούς άξονες: τον κωδικό κάθε άξονα, τον οδικό φόρτο κάθε άξονα και το χρονόσημο εκπομπής των δεδομένων, λ.χ. κάθε 5 δευτερόλεπτα. Επίσης κάθε 5 δευτερόλεπτα έρχονται δύο πλειάδες κάθε φορά, μία για τον οδικό άξονα A και μία για τον οδικό άξονα B. Σύμφωνα με τα δεδομένα τίθεται το εξής ερώτημα: «Βρες τον μέσο φόρτο των δύο αξόνων, εξετάζοντας τα $N = 3$ πιο πρόσφατα στοιχεία ανά άξονα (AxisID)».

```
SELECT AxisID, AVG(R.Load) AS Axis_Load
FROM Axes_load AS R [PARTITION BY 'AxisID' ROWS '3'];
```

Όπως φαίνεται και από το σχήμα 2.6, το ερώτημα διαρκείας το οποίο εκτελείται πάνω στο ρεύμα R, επιστρέφει τον μέσο όρο του φόρτου (AVG) κάθε ενός οδικού άξονα, ο οποίος όμως υπολογίζεται από δεδομένα 3 πλειάδων για κάθε άξονα. Πάνω στα στοιχεία αυτού του ρεύματος εφαρμόζεται μεριστικό παράθυρο μεγέθους $N = 3$ πλειάδων, με λίστα γνωρισμάτων ομαδοποίησης τον κωδικό κάθε άξονα, δηλαδή $L = \{AxisID\}$.

2.6 Γλώσσες ερωταποκρίσεων ρευμάτων δεδομένων

Η διατύπωση των ερωτημάτων διαρκείας στις εφαρμογές ρευμάτων δεδομένων, συνήθως πραγματοποιείται μέσω κάποιας δηλωτικής (*declarative*) γλώσσας μορφής SQL, αφήνοντας στο σύστημα την επιλογή του κατάλληλου φυσικού προσχεδίου εκτέλεσης (*physical query plan*). Κατά τον τρόπο αυτό, καθορίζεται το είδος και η διάταξη των τελεστών που θα χρησιμοποιηθούν κατά την εκτέλεση του ερωτήματος. Εναλλακτικά, μπορεί να χρησιμοποιηθεί κάποια διαδικαστική (*procedural*) γλώσσα για τον άμεσο προσδιορισμό του φυσικού προσχεδίου εκτέλεσης από τον χρήστη, πιθανόν και με την βοήθεια κάποιας κατάλληλης γραφικής διεπαφής χρήστη (*Graphical User Interface*).

Όπως όμως αναφέρθηκε, τα ερωτήματα διαρκείας εστιάζουν συνήθως στην επεξεργασία της πιο πρόσφατης πληροφορίας, εγείροντας την ανάγκη αφενός για χρονική σήμανση των στοιχείων, αφετέρου για την υλοποίηση κατάλληλων τελεστών που θα αποσπούν και θα επεξεργάζονται τα πλέον πρόσφατα στοιχεία. Η χρονική σήμανση επιτυγχάνεται με την προσθήκη ενός πεδίου χρονοσήμου στις πλειάδες των ρευμάτων, ενώ εισάγεται η έννοια των παραδύρων για να δηλωθεί η έμφαση των ερωτημάτων στην πιο πρόσφατη πληροφορία.

2.6.1 Προτεινόμενες γλώσσες ερωταποκρίσεων

Τα παράθυρα αποτελούν βασικό και καινοτόμο μηχανισμό των προτεινόμενων στην βιβλιογραφία ΣΔΡΔ (*Aurora, Gigascope, STREAM, TelegraphCQ*). Στα πλαίσια υλοποίησης αυτών των συστημάτων έχουν προταθεί οι εξής γλώσσες ερωταποκρίσεων:

SQuAl (Stream Query Algebra) [ACC+03b]. Στο σύστημα *Aurora*, τα ερωτήματα που διατυπώνονται στην γλώσσα *SQuAl* μετασχηματίζονται σε γράφους κατάλληλα συνδεδεμένων τόξων και κουτιών. Οι γράφοι μπορούν εναλλακτικά να σχηματιστούν μέσω κατάλληλης γραφικής διεπαφής. Τα κουτιά αναπαριστούν τους παρεχόμενους τελεστές, ενώ τα τόξα αντιστοιχούν σε ρεύματα δεδομένων και χρησιμοποιούνται για τις συνδέσεις μεταξύ κουτιών. Οι τελεστές λειτουργούν με ρεύματα δεδομένων, ενώ η επεξεργασία στατικών σχέσεων υποστηρίζεται μόνο κατά έμμεσο τρόπο. Τα παράθυρα αποτελούν έμφυτο χαρακτηριστικό κάθε τελεστή που τα χρειάζεται και ορίζονται μέσω κατάλληλων παραμέτρων του τελεστή κατά την διατύπωση του ερωτήματος.

GSQL [CJSS03]. Στο σύστημα *Gigascope*, η γλώσσα αυτή έχει μορφή γλώσσας SQL, αλλά εφαρμόζεται αποκλειστικά σε ρεύματα δεδομένων. Κάθε ρεύμα θεωρείται ότι διαθέτει ένα ή περισσότερα γνωρίσματα βάσει των οποίων μπορεί να οριστεί κάποια χρονική διάταξη. Κατά την διατύπωση των ερωτημάτων ορίζονται πρωτόκολλα με την βοήθεια των οποίων: α) ερμηνεύονται τα περιεχόμενα των εισερχόμενων ρευμάτων, β) καθορίζεται ο τρόπος προσδιορισμού της διάταξής τους και γ) προσομοιώνεται η λειτουργία των παραδύρων. Η προσομοίωση των παραδύρων πραγματοποιείται με συνθήκες επιλογής πάνω στα γνωρίσματα που καθορίζουν την διάταξη των στοιχείων σύμφωνα με το επιλεγμένο πρωτόκολλο. Τέλος, λαμβάνεται ιδιαίτερη μέριμνα για

την ενημέρωση των αποτελεσμάτων ακόμα και κατά την απουσία εισερχόμενων στοιχείων πληροφορίας.

CQL (Continuous Query Language) [ABW03]. Στο σύστημα STREAM, η CQL αποτελεί επέκταση της γλώσσας SQL στην οποία υποστηρίζεται η επεξεργασία τόσο ρευμάτων δεδομένων όσο και στατικών σχέσεων. Η CQL αποτελείται από τρία δομικά στοιχεία: α) την σχεσιακή γλώσσα ερωταποκρίσεων SQL για τους τελεστές σχεσιακών πινάκων β) μία γλώσσα προσδιορισμού παραθύρων η οποία ακολουθεί συντακτικά στοιχεία της SQL-99 για την μετατροπή των ρευμάτων σε σχεσιακούς πίνακες και γ) τρεις ειδικούς τελεστές για την μετατροπή σχεσιακών πινάκων σε ρεύματα: ISTREAM για τις νέες πλειάδες του προσωρινού σχεσιακού πίνακα, DSTREAM για τις πλειάδες που διαγράφηκαν από τον πίνακα και RSTREAM για την παροχή όλων των πλειάδων του προσωρινού πίνακα.

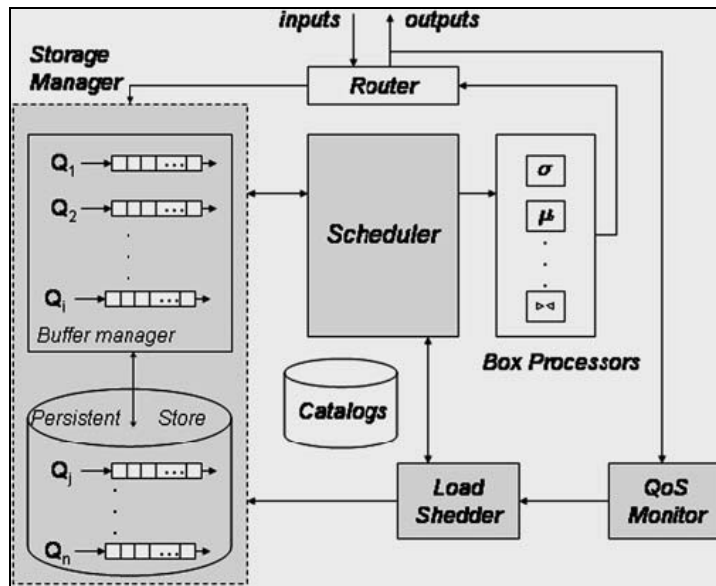
StreaQueL (Stream Query Language) [CCD+03, GO03]. Το σύστημα TelegraphCQ, υποστηρίζει μία δηλωτική γλώσσα, η οποία αποδίδει ιδιαίτερη έμφαση στον προσδιορισμό διαφόρων τύπων παραθύρων. Κατά βάση, στην τυπική σύνταξη ενός ερωτήματος σε SQL προσαρτάται μια ειδική δομή σύνταξης (RANGE BY 'time' SLIDE BY 'time' START AT 'time'), δηλώνοντας το παράθυρο που θα χρησιμοποιηθεί για την απομόνωση των στοιχείων των αντίστοιχων ρευμάτων. Πέρα από τα ρεύματα, υποστηρίζεται η επεξεργασία και σχεσιακών πινάκων.

2.7 Κυριότερα συστήματα διαχείρισης ρευμάτων δεδομένων

Στην ενότητα αυτή παρουσιάζονται συνοπτικά τα κυριότερα συστήματα τα οποία ασχολούνται με τα ρεύματα δεδομένων ή υποστηρίζουν τα ερωτήματα διαρκείας. Τέσσερα από αυτά (Aurora, NiagaraCQ, STREAM, TelegraphCQ), έχουν εμφανιστεί πιο πρόσφατα, υιοθετώντας μια ολοκληρωμένη οπτική αντιμετώπισης των ιδιαίτερων θεμάτων που ανακύπτουν σχετικά με τα ρεύματα δεδομένων και προτείνουν μερικές πρωτότυπες ιδέες ή αλγορίθμους. Οι σχετικές δημοσιεύσεις, αποτέλεσαν σημαντικό τμήμα της βιβλιογραφικής μελέτης που πραγματοποιήθηκε.

2.7.1 AURORA

Το σύστημα AURORA είναι ένα γενικού σκοπού Σύστημα Διαχείρισης Ρευμάτων Δεδομένων που σχεδιάζεται και υλοποιείται από το 2001 με τη σύμπραξη των Πανεπιστημίων Brandeis και Brown καθώς και του M.I.T [ACC+03b, wwwAUR]. Για την επεξεργασία των ρευμάτων, προτείνεται μια προσέγγιση βασισμένη σε ένα γραφικό περιβάλλον με «κουτιά» και «βέλη», θυμίζοντας ένα διάγραμμα ροής δεδομένων αναπαριστώντας το προσχέδιο εκτέλεσης κάποιου ερωτήματος διαρκείας. Οι πρωτογενείς τελεστές («κουτιά») χρησιμοποιούνται για να συνδέσουν το ερώτημα που πρέπει να υποβληθεί στο σύστημα, ενώ βάσει αυτών υλοποιούνται και άλλοι, πιο πολύπλοκοι, όπως οι γνωστοί τελεστές συγχώνευσης (merge), σύνδεσης (join), επιλογής (filter), κατάργησης (drop) πλειάδων κ.ά.



Σχήμα 2.7: Αρχιτεκτονική του συστήματος Aurora. (Πηγή: [ACC+03b])

Οι χρήστες μπορούν επίσης να διατυπώνουν τα ερωτήματά τους σε μια δηλωτική γλώσσα (όπως μια επέκταση της SQL), έπειτα όμως είναι απαραίτητος ο μετασχηματισμός τους στην αναπαράσταση που αντιλαμβάνεται το σύστημα, δηλαδή με «κουτιά» και «βέλη».

Ο χρονοπρογραμματιστής (scheduler) αποτελεί το συνδετικό κρίκο όλων των τμημάτων της αρχιτεκτονικής του συστήματος, όπως εικονίζεται στο σχήμα 2.7. Αυτός αποφασίζει ποια «κουτιά» θα τρέξουν στη συνέχεια, προσδιορίζοντας τόσο τον αριθμό των πλειάδων που θα δοθούν για επεξεργασία σε κάποιο τελεστή, όσο και η πορεία των στοιχείων διαμέσου των τελεστών προς την έξοδο. Οι πλειάδες που καταλήγουν στην έξοδο παρακολουθούνται μονίμως από τον Επόπτη Ποιότητας (QoS Monitor), επισημαίνοντας στον χρονοπρογραμματιστή χρήσιμα στοιχεία για τις επιδόσεις του συστήματος.

Υπάρχει επίσης ένας διαχειριστής αποθήκευσης (Storage Manager) που αναλαμβάνει να οδηγήσει τις πλειάδες σε ενδιάμεσες ουρές (buffer queues). Αυτό θα συμβεί όταν διαπιστωθεί ότι εξαντλείται η διαθέσιμη ποσότητα μνήμης, κάτι που δεν μπορεί να αποκλειστεί όταν τα ιστορικά στοιχεία που συσσωρεύονται στα σημεία σύνδεσης διογκωθούν υπερβολικά. Τα στοιχεία του ρεύματος διέρχονται μέσα από το δίκτυο των τελεστών, που μπορεί να θεωρηθεί ως ένας άκυκλος κατευθυνόμενος γράφος εκτελούμενων λειτουργιών. Σε κάθε εισερχόμενο στοιχείο προσδίδεται ένα μοναδικό αναγνωριστικό συστήματος τύπου χρονοσήμου, το οποίο χρησιμοποιείται για την εποπτεία της παρεχόμενης ποιότητας υπηρεσίας από το σύστημα.

Τελικά τα στοιχεία καταλήγουν στην έξοδο, οι πλειάδες της οποίας τροφοδοτούν κάποια εφαρμογή κατά ασύγχρονο τρόπο, ανάλογα δηλαδή προς το ρυθμό παραγωγής τους. Οι χρήστες έχουν την ευχέρεια να ορίσουν διάφορες απαιτήσεις σχετικά με την ποιότητα υπηρεσιών (Quality of Service QoS) των ερωτημάτων που θέτουν στο σύστημα. Το σύστημα με την σειρά του αντεπεξέρχεται στις απαιτήσεις αυτές με την δημιουργία και παρακολούθηση διαγραμμάτων ποιότητας υπηρεσίας (QoS graphs) όσον αφορά τους χρόνους

απόκρισης, την ακρίβεια των αποτελεσμάτων και την βαρύτητα των εξαγόμενων απαντήσεων. Ο χρήστης έχει την δυνατότητα να μεταβάλλει την προτεραιότητα εκτέλεσης των ερωτημάτων, με στόχο την μεγιστοποίηση του αθροιστικού μεγέθους της ποιότητας υπηρεσιών (QoS) που λαμβάνεται, συνυπολογίζοντας όλους τους τελεστές («κουτιά»).

Οι προδιαγραφές των χρηστών και τα διαγράμματα ποιότητας χρησιμεύουν στη συνέχεια για να αποφασιστεί με ποιο τρόπο και σε ποιες περιστάσεις θα αποβληθεί κάποιο μέρος των δεδομένων, ελαφρύνοντας το φόρτο του συστήματος (*load shedding*) επηρεάζοντας την ακρίβεια των απαντήσεων. Έτσι, εάν το σύστημα διαπιστώσει ότι η τεχνική αυτή δεν αποδίδει αρκετά, μπορεί να προσπαθήσει να αναδιατάξει το δίκτυο των τελεστών, χρησιμοποιώντας γνωστές μεθόδους βελτιστοποίησης, όπως αυτές που στηρίζονται στην αντιμεταθετικότητα τελεστών.

2.7.2 GIGASCOPE

Πρόκειται για ένα σύστημα που αναπτύσσεται από την AT&T σε συνεργασία με το Πανεπιστήμιο Carnegie Mellon [CJSS03]. Το Gigascope χρησιμοποιείται στη διαχείριση δικτύου τηλεπικοινωνιών ή υπολογιστών και εφαρμόζεται στην εποπτεία δικτύων οπτικών ινών υψηλών ταχυτήτων, με ικανότητα μεταφοράς εκατομμυρίων πακέτων ανά δευτερόλεπτο. Το σύστημα αυτό έχει χρησιμοποιηθεί σε διάφορες εφαρμογές, όπως ανάλυση δικτύων, ανάλυση πρωτοκόλλων, ανίχνευση μη εξουσιοδοτημένης διείσδυσης σε δίκτυο, καθώς και σε ερευνητικούς σκοπούς (π.χ. video streams). Στο σύστημα αυτό η διακινούμενη πληροφορία παίρνει την μορφή ρεύματος δεδομένων αποκλείοντας την επεξεργασία στατικών σχέσεων.

Προκειμένου να είναι εφικτή η επεξεργασία των ρευμάτων εισόδου και η συνακόλουθη μετατροπή τους σε ρεύματα εξόδου, σε όλες τις πλειάδες προσκολλάται χρονόσημο έπειτα από την εκτέλεση των σχετικών ερωτημάτων διαρκείας. Ειδικά τα ερωτήματα που άπτονται της ανάλυσης του δικτύου κάνουν ρητή αναφορά στο χρονικό προσδιορισμό των στοιχείων. Σε αντίθεση με παρόμοιες εφαρμογές τηλεπικοινωνιών που χρησιμοποιούν διαδικαστικές γλώσσες ερωταποκρίσεων, στο Gigascope προτιμήθηκε για λόγους ευελιξίας μια παραλλαγή της SQL (GSQL).

Πρόκειται για μια συνεπτυγμένη μορφή της SQL (π.χ. δεν επιτρέπονται *outer joins*), αλλά με ορισμένους πρόσθετους τελεστές (όπως ο τελεστής *merge* για τη συγχώνευση ρευμάτων δεδομένων, διατηρώντας όμως τη διάταξη των χρονοσήμων). Οι χρήστες, μέσω ενός μηχανισμού δήλωσης όψεων (*views*) έχουν τη δυνατότητα ορισμού συναρτήσεων ή τελεστών για εξειδικευμένες λειτουργίες, οι οποίες κατόπιν μπορούν να τεθούν στη διάθεση και των υπολοίπων χρηστών. Όταν υποβάλλονται ερωτήματα στο σύστημα, περνούν από τη διαδικασία μεταγλώττισης (*compiler*) και εξάγονται τμήματα κώδικα σε C και C++, τα οποία και τελικά εκτελούνται, επιστρέφοντας στους χρήστες τα εξαγόμενα ρεύματα δεδομένων. Τα αποτελέσματα των ερωτημάτων μπορούν να διοχετευτούν σε αποθήκες δεδομένων (*data warehouses*) για περαιτέρω επεξεργασία.

2.7.3 NiagaraCQ

Το NiagaraCQ είναι ένα από τα υποσυστήματα του πρωτότυπου προγράμματος Niagara που σχεδιάζεται στο Πανεπιστήμιο του Wisconsin από το 1999, με γνώμονα την εξυπηρέτηση εφαρμογών Internet. Στόχος του είναι η ανάπτυξη σε περιβάλλον JAVA ενός κατανεμημένου συστήματος βάσεων δεδομένων σε ένα περιβάλλον όπου τα ερωτήματα υποβάλλονται και αποσύρονται δυναμικά, ειδικά για κατανεμημένα δεδομένα σε μορφή XML με χρήση μιας γλώσσας ερωταποκρίσεων όπως η XML-QL.

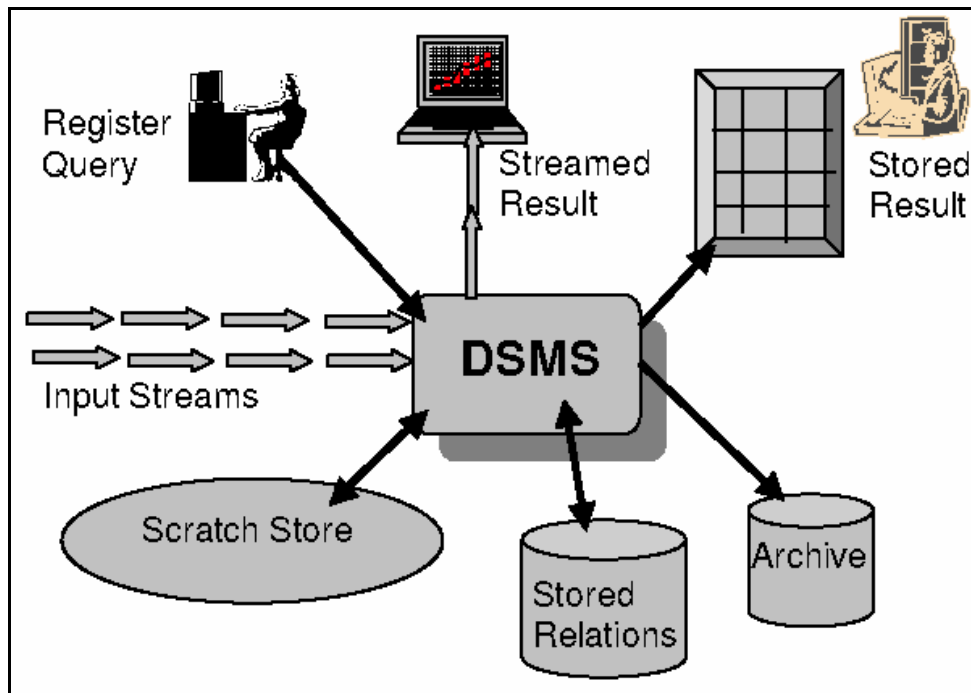
Θεωρητικά, ένας μεγάλος αριθμός χρηστών θα πρέπει να έχει τη δυνατότητα να θέσει ερωτήματα διαρκείας σε αυτή τη γλώσσα, οπότε το σύστημα θα πρέπει να μπορεί να υποστηρίξει έναν κλιμακούμενο αριθμό από triggers, δηλαδή περίπλοκα ερωτήματα σε δεδομένα Internet. Προκειμένου λοιπόν να ανταπεξέρχεται στις απαιτήσεις εκατομμυρίων ερωτημάτων, πρέπει να μπορεί να τα ομαδοποιήσει βάσει των κοινών τους στοιχείων, αφού είναι αναμενόμενο ότι πολλά ερωτήματα θα έχουν μεταξύ τους αρκετές ομοιότητες.

2.7.4 STREAM (STanford stREam datA Management)

Το STREAM είναι ένα Σύστημα Διαχείρισης Ρευμάτων Δεδομένων γενικού σκοπού, το οποίο αναπτύσσεται από το 2001 στο Πανεπιστήμιο Stanford [ABW03, BBD+02, wwwSTR]. Εγκαταλείποντας οποιαδήποτε απόπειρα προσαρμογής κάποιου υπάρχοντος ΣΔΒΔ ώστε να ανταποκρίνεται στις απαιτήσεις που θέτουν τα χαρακτηριστικά των ρευμάτων δεδομένων και η φύση των ερωτημάτων διαρκείας, οι προσπάθειες επικεντρώθηκαν στη συγκρότηση ενός ολοκληρωμένου πρωτότυπου ΣΔΡΔ. Η ανάπτυξη έχει τρεις κυρίως στόχους:

- A. Ένα ευέλικτο τρόπο διεπαφής (interface) προκειμένου να διευκολύνεται η ανάγνωση και η εγγραφή ρευμάτων δεδομένων, ως μέρος μιας ιεραρχικής διαχείρισης του αποθηκευτικού χώρου.
- B. Την αποτελεσματική επεξεργασία των ερωτημάτων διαρκείας που διατυπώνονται σε SQL ή με τελεστές της σχεσιακής άλγεβρας, συμπεριλαμβανομένων των συναθροιστικών.
- Γ. Ένα περιβάλλον API για την υποβολή των ερωτημάτων διαρκείας και τη λήψη των απαντήσεων σ' αυτά.

Στο σχήμα 2.8, παρουσιάζεται σε απλοποιημένη μορφή η αρχιτεκτονική σύλληψη του συστήματος. Παρόλο που βασική προτεραιότητα αποτελεί η online επεξεργασία των δεδομένων, δεν μπορεί να αποκλειστεί το ενδεχόμενο ορισμένες εφαρμογές να προϋποδέτουν μόνιμη αρχειοθέτηση (Archive) κάποιων στοιχείων για μεταγενέστερη επεξεργασία. Επιπλέον, διάφορα ερωτήματα διαρκείας τυπικά απαιτούν την τήρηση κάποιας ενδιάμεσης κατάστασης (Scratch Store), η οποία μπορεί να φυλάσσεται στη μνήμη (τακτική που ακολουθεί το STREAM) ή ακόμη και στο δίσκο.



Σχήμα 2.8: Η αρχιτεκτονική του συστήματος STREAM. (Πηγή: [Pat04b])

Οι χρήστες έχουν δυνατότητα υποβολής ερωτημάτων διαρκείας, των οποίων η εκτέλεση παρατείνεται μέχρις ότου απενεργοποιηθούν. Τα αποτελέσματα που προκύπτουν μπορεί να διοχετεύονται σε εφαρμογές ως άλλα ρεύματα δεδομένων (Streamed Result), αλλά μπορεί να θεωρηθούν και ως κάποιας μορφής υλοποιημένες όψεις, δηλαδή σχεσιακοί πίνακες που ενημερώνονται με την πάροδο του χρόνου (Stored Result).

Τα ερωτήματα υποβάλλονται με χρήση της ειδικά διαμορφωμένης δηλωτικής γλώσσας ερωταποκρίσεων CQL (Continuous Query Language). Συντακτικά, η CQL είναι υπερσύνολο της SQL, με προσθήκη εξειδικευμένων δομών για την υποστήριξη κυλιόμενων παραθύρων και δειγματοληψίας δεδομένων (sampling). Σημαντικό στοιχείο της γλώσσας αποτελεί το γεγονός ότι η σημασιολογία των ερωτημάτων διαρκείας επιβάλλεται να αντιμετωπίζεται με παρόμοιο τρόπο τόσο τα δεδομένα των ρευμάτων όσο κι εκείνα που αντλούνται από στατικές σχέσεις.

Όταν ένα ερώτημα διαρκείας υποβάλλεται στο σύστημα, μετασχηματίζεται στο κατάλληλο προσχέδιο εκτέλεσης, διαφορετικό για κάθε ερώτημα. Εναλλακτικά, τα προσχέδια μπορούν να υποβληθούν στο σύστημα απευθείας, με χρήση ενός γραφικού περιβάλλοντος, παρακάμπτοντας τη διατύπωση ερωτημάτων με την CQL. Αυτό το περιβάλλον προσφέρει τη δυνατότητα συγχώνευσης παρόμοιων προσχεδίων εκτέλεσης ή έστω κάποιων μερών τους. Το γραφικό περιβάλλον στηρίζεται στο γεγονός ότι τα ερωτήματα διαρκείας μπορούν να παρασταθούν με τη βοήθεια δομών στην κύρια μνήμη και να τοποθετηθούν σε αρχεία XML. Συνεπώς, κάποιοι ειδικά εξουσιοδοτημένοι χρήστες μπορούν να επέμβουν σ' αυτά τα αρχεία, δημιουργώντας, τροποποιώντας ή μεταφέροντας στοιχεία από το ένα στο άλλο, πριν τα δέσουν στο σύστημα.

Ένα προσχέδιο εκτέλεσης ερωτήματος αποτελείται από τελεστές αλληλοσυνδεόμενους με ουρές και τροφοδοτούμενους από συνόψεις δεδομένων. Οι τελεστές δέχονται δεδομένα από τις ουρές εισόδου, επεξεργάζονται τις πλειάδες βάσει της σημασιολογίας τους και παραδίδουν τις πλειάδες του αποτελέσματος σε μια - μοναδική για τον κάθε τελεστή - ουρά εξόδου. Εκτός από τους γνωστούς σχεσιακούς τελεστές, για ορισμένους έχουν αναπτυχθεί και οι παραδυρικές εκδοχές τους (λ.χ. για τη σύνδεση ρευμάτων), καθώς και τελεστές δειγματοληψίας. Οι ενδιάμεσες ουρές καθορίζουν τα μονοπάτια που ακολουθούν οι πλειάδες κατά τη διάρκεια της εκτέλεσής τους.

Τέλος, οι συνόψεις (synopses) δεδομένων χρησιμοποιούνται για να εξάγουν κάποιες περιλήψεις στοιχείων των ρευμάτων που έχουν παρέλθει από ορισμένους ενδιάμεσους τελεστές. Αυτές οι συνόψεις, που συνήθως βασίζονται σε κυλιόμενα παράθυρα, θα αξιοποιηθούν κατά το μελλοντικό υπολογισμό που θα διεξάγει ο τελεστής (λ.χ. εάν πρόκειται για τελεστή σύνδεσης, μπορεί να διατηρούνται συνοπτικά στοιχεία για το καθένα από τα δύο ρεύματα που πρόκειται να συσχετιστούν). Η εκτέλεση των ερωτημάτων ρυθμίζεται από έναν καθολικό χρονοπρογραμματιστή (global scheduler).

Η οπτικοποίηση πληροφοριών σχετικών με τα ερωτήματα, την εκτέλεσή τους και την κατανομή των πόρων του συστήματος, είναι πολύ σημαντική για τους διαχειριστές του συστήματος. Με αυτόν τον τρόπο, μπορούν να ρυθμίσουν κατάλληλα την απόδοση του ΣΔΡΔ, αν και το ίδιο από μόνο του θα πρέπει να ανταποκρίνεται σε μεταβαλλόμενες συνθήκες, που προκύπτουν τόσο από το πλήθος των ερωτημάτων, όσο και από τα χαρακτηριστικά των ρευμάτων. Οι παρεχόμενες δυνατότητες αναφέρονται στην τροποποίηση - κατά το χρόνο εκτέλεσης - της κατανομής της μνήμης (π.χ. μεταξύ συνόψεων), της δομής των προσχεδίων εκτέλεσης (π.χ. αλλάζοντας τον τύπο της σύνοψης που χρησιμοποιείται από κάποιον τελεστή σύνδεσης), καθώς και της πολιτικής χρονοδρομολόγησης ανάμεσα στις διαθέσιμες εναλλακτικές. Η μνήμη του συστήματος μοιράζεται δυναμικά μεταξύ των συνόψεων και των ουρών στα σχέδια εκτέλεσης των ερωτημάτων, μαζί με την ενδιάμεση μνήμη (buffers) που διευκολύνουν το χειρισμό ρευμάτων που καταφτάνουν στο σύστημα, καθώς και μιας μορφής λανθάνουσας μνήμης (cache) που χρησιμοποιείται για δεδομένα που τηρούνται στο δίσκο.

Συμπερασματικά, κεντρικό πρόβλημα στο σύστημα STREAM αποτελεί η αποτελεσματική εκτέλεση των ερωτημάτων διάρκειας σε καθεστώς περιορισμένης ποσότητας μνήμης. Ως επί το πλείστον, το ενδιαφέρον εστιάζεται στον υπολογισμό προσεγγιστικών απαντήσεων και στην ανάλυση των απαιτήσεων σε μνήμη των ερωτημάτων που τίθενται.

2.7.5 TelegraphCQ

Το σύστημα αυτό αποτελεί μετεξέλιξη του πρωτοτύπου Telegraph που συντονίζεται από το Πανεπιστήμιο Berkeley ήδη από το 2000, με κύριο στόχο την ανάπτυξη μιας αρχιτεκτονικής προσαρμοζόμενης στη ροή των δεδομένων κυρίως σε δικτυακά περιβάλλοντα, με έμφαση στα δίκτυα αισθητήρων [CCD+03, wwwTCQ]. Ωστόσο, η αναγκαιότητα αντιμετώπισης των ζητημάτων που

ανακύπτουν ως προς το χειρισμό ρευμάτων δεδομένων οδήγησε σε εκ νέου σχεδιασμό και (από το 2002) στην υλοποίηση του TelegraphCQ, διαχωρίζοντας το από το ευρύτερο αντικείμενο του Telegraph.

Η οπτική της προσέγγισης των ρευμάτων δεδομένων στο TelegraphCQ παρουσιάζει μια πολύ ενδιαφέρουσα πρωτοτυπία: θεωρείται ότι δεν είναι μόνο τα δεδομένα που εμφανίζουν μεταβλητότητα και ρέουν μέσα στο δίκτυο, αλλά και τα ίδια τα ερωτήματα διαρκείας μπορούν να παρομοιαστούν με ρεύματα, μιας και τόσο ο αριθμός τους όσο και η δομή τους αλλάζει κατά απρόβλεπτο τρόπο με την πάροδο του χρόνου.

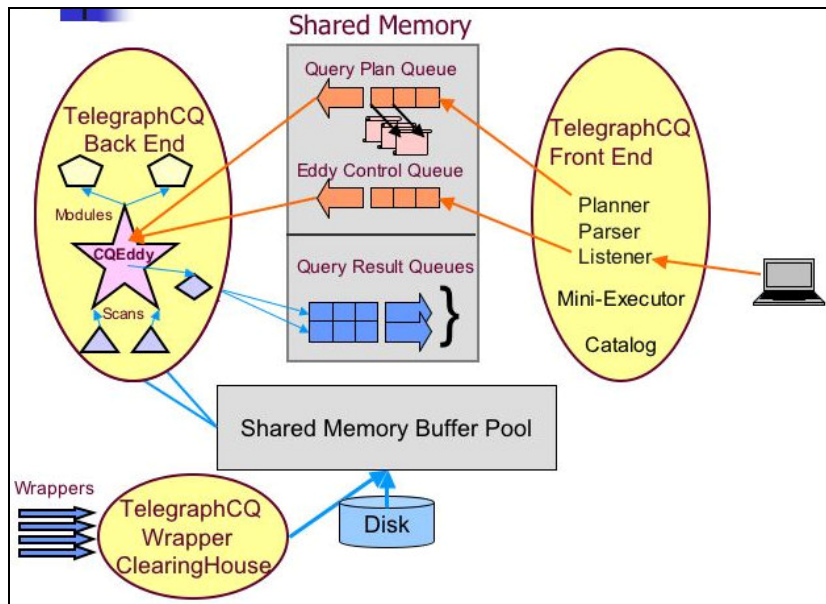
Η κινητικότητα λοιπόν τόσο των δεδομένων όσο και των ερωτημάτων που τα αφορούν έδωσαν το έναυσμα ώστε η προσέγγιση που ακολουθείται από το TelegraphCQ να περιστραφεί γύρω από την προσαρμοστικότητα τελεστών (adaptivity of operators). Η διαχείριση των δεδομένων στο σύστημα αποβλέπει στην ικανότητα του να εξελίσσεται και να προσαρμόζεται ταχέως σε δραστικές αλλαγές που αναφέρονται στη διαθεσιμότητα δεδομένων, στο περιεχόμενό τους, στα χαρακτηριστικά του ίδιου του συστήματος ή του δικτύου που το τροφοδοτεί, και φυσικά στις απαιτήσεις των χρηστών.

Η ανάπτυξη στηρίχθηκε αρχικά στην προσαρμογή στοιχείων της αρχιτεκτονικής της PostgreSQL προγραμματίζοντας σε C/C++, ώστε να καταστεί εφικτή η από κοινού επεξεργασία ερωτημάτων διαρκείας επί ρευμάτων δεδομένων. Πολλά τμήματα του κώδικα της PostgreSQL χρησιμοποιήθηκαν, άλλα με ελάχιστες και άλλα με σημαντικές τροποποιήσεις, κυρίως αυτά που αφορούν το εξωτερικό μέρος (Front End) της επικοινωνίας με τους χρήστες και τα ερωτήματα που θέτουν στο σύστημα.

Στο σχήμα 2.9, με τις ελλείψεις απεικονίζονται οι κύριες διαδικασίες που συναποτελούν τον εξυπηρετητή (server) του TelegraphCQ (Front End, Executor, Wrapper), ενώ προβλέπεται ειδικός χώρος κοινής μνήμης (shared memory).

Στο εξωτερικό επίπεδο, ο Postmaster της PostgreSQL ξεκινάει κάποια νέα διεργασία μόλις αντιληφθεί αίτημα για νέα σύνδεση με το σύστημα. Επειδή κάθε σύνδεση μπορεί να έχει πολλούς ανοιχτούς δρομείς (cursors), χρησιμοποιείται ένας αντιπρόσωπος (proxy) για την συγκέντρωση των διάσπαρτων ερωτημάτων από τους χρήστες, οπότε είναι δυνατόν να ενεργοποιηθούν πολλοί δρομείς (cursors) με μια μόνο σύνδεση. Ο Ακροατής (Listener) υποδέχεται τα ερωτήματα διαρκείας, τα οποία αναλύονται συντακτικά (Parser) και βελτιστοποιούνται (Optimizer) προκειμένου να προκύψει ένα προσαρμοζόμενο προσχέδιο εκτέλεσης (adaptive query plan).

Κάθε προσχέδιο περιλαμβάνει και τελεστές που έχουν ήδη αναπτυχθεί στα πλαίσια του Telegraph, με κύριο στόχο την προσαρμοστικότητα στις εκάστοτε καταστάσεις. Έτσι, τα Eddies είναι τα υποτμήματα (modules) που αποφασίζουν με ποιο τρόπο τα δεδομένα πρέπει να διοχετεύονται αδιαλείπτως στους τελεστές των ερωτημάτων πλειάδα προς πλειάδα. Τα Flux (Fault-tolerant Loadbalancing eXchange) δρομολογούν τις πλειάδες μεταξύ των επεξεργαστών μιας συστοιχίας (cluster) ώστε να επιτύχουν παραλληλισμό της εκτέλεσης με εξισορρόπηση φόρτου (load balancing) και ανοχή σε σφάλματα (fault-tolerance).



Σχήμα 2.9: Η αρχιτεκτονική του συστήματος TelegraphCQ. (Πηγή: [Pat04a])

Αυτά τα πρόσθετα υποτιμήματα δεν μπορούν να ξεχωρίσουν αρχιτεκτονικά από τους άλλους συμβατικούς τελεστές, αφού απλώς απορροφούν και στη συνέχεια παράγουν πλειάδες. Ωστόσο, ο συνδετικός κρίκος τους είναι τα Fjords, μιας μορφής API που επιτρέπουν την ενδοεπικοινωνία μεταξύ αυτών των τμημάτων, ώστε να σχηματιστεί το τελικό προσχέδιο εκτέλεσης του ερωτήματος. Έπειτα, τα προσχέδια διοχετεύονται δυναμικά σε μια ουρά που έχει δημιουργηθεί στο κοινό τμήμα μνήμης του συστήματος. Απ' εκεί, ο Εκτελεστής (Executor) επιλέγει συνεχώς τα πλέον πρόσφατα προσχέδια και τα προσθέτει στα ήδη εκτελούμενα ερωτήματα. Τα αποτελέσματα της επεξεργασίας περιέρονται σε ουρές εξόδου στο κοινό τμήμα της μνήμης, απ' όπου ο Ακροατής (Listener) τα κατευθύνει στον αντιπρόσωπο (proxy) για να διανεμηθούν τελικά στους χρήστες.

Εφόσον αναμένεται ότι το σύστημα θα υποστηρίξει ένα μεγάλο αριθμό ερωτημάτων, δεν κρίνεται σκόπιμο να δημιουργηθεί καινούργια διεργασία (thread) για το καθένα. Εν τούτοις, κρίνεται επιθυμητό η παρουσία πολλαπλών διεργασιών οι οποίες θα μπορούν να εκτελεστούν παράλληλα, να εντάσσεται στις δυνατότητες του TelegraphCQ. Κάθε διεργασία αποτελεί και ένα διαφορετικό «Αντικείμενο Εκτέλεσης» (Execution Object), που περιλαμβάνει έναν χρονοπρογραμματιστή (scheduler), μια ή περισσότερες ουρές γεγονότων (event queues) και μια σειρά από μη προβλέψιμες αφηρημένες «Ενότητες Αποστολής» (Dispatch Units), δηλαδή οντότητες με κοινά στοιχεία που θα σταλούν προς εκτέλεση στο σύστημα. Γι' αυτό το λόγο, στον Εκτελεστή (Executor) τα ερωτήματα διακρίνονται σε κατηγορίες, ανάλογα με την ευχέρεια που εμφανίζουν για κοινή εκτέλεση με άλλα παρόμοια, βάσει επικαλύψεων στα ρεύματα δεδομένων και τους πίνακες που εμπλέκονται στο καθένα.

Τέλος, υπάρχει ο μηχανισμός του Wrapper στον οποίο ανατίθεται η προσκόμιση των δεδομένων των ρευμάτων στο σύστημα. Οι κύριες δυσκολίες στο σημείο αυτό, είναι αφενός μεν η αποφυγή ανασταλτικών φαινομένων, π.χ. επιβράδυνση της εκτέλεσης εξαιτίας χρονοβόρων αναμονών για δεδομένα, αφετέρου δε η ελάττωση των προσπελάσεων στο δίσκο. Γι' αυτό, το συγκεκριμένο

υποσύστημα του TelegraphCQ εκτελείται ως χωριστή διαδικασία, με χρήση ειδικών μη ανασταλτικών δομών, όπως τα Fjords, ενεργοποιώντας μια σειρά διεργασιών ώστε τα δεδομένα να εισάγονται και να εξάγονται αποτελεσματικά. Κατ' αυτόν τον τρόπο, επιτυγχάνεται δυνατότητα διαχείρισης διαφόρων τύπων πηγών δεδομένων, δηλαδή στοιχείων που είτε απαιτούνται από το σύστημα (pull model) είτε προσκομίζονται σ' αυτό (push model). Όταν τα δεδομένα εισέλθουν μέσω του Wrapper προωθούνται στον Εκτελεστή για επεξεργασία με τη διαμεσολάβηση των λεγόμενων Streamers. Οι τελευταίοι τα μετατρέπουν σε πλειάδες στις δομές της ενδιάμεσης μνήμης (buffer pool), κι αν ο χώρος δεν επαρκεί, τα καταχωρούν ακόμη και στο δίσκο. Οι πλειάδες που προκύπτουν προσπελαύνονται από τον Εκτελεστή μέσω ενός τελεστή σάρωσης (scanner) που η συμπεριφορά του ελέγχεται από τις δομές παραθύρων που έχουν ενσωματωθεί στα ερωτήματα.

Κεφάλαιο 3

Διαχείριση κινούμενων αντικειμένων

3.1 Εισαγωγή

Η μεγάλη τεχνολογική εξέλιξη στο χώρο των συστημάτων γεωγραφικού εντοπισμού, κυρίως GPS και ασύρματες τηλεπικοινωνίες, έχει δώσει ώθηση σε πολλές εφαρμογές στον τομέα των βάσεων δεδομένων, προσφέροντας πλέον αρκετά μεγαλύτερη ακρίβεια σε πολύ μικρότερο κόστος. Αυτές οι εφαρμογές καλούνται να κάνουν διαχείριση χωροχρονικών φαινομένων, δηλαδή αντικειμένων με χωρική υπόσταση των οποίων η θέση ή η έκταση μεταβάλλεται με την πάροδο του χρόνου.

Ο τομέας των *Βάσεων Δεδομένων Κινούμενων Αντικειμένων* (*Moving Object Databases - MODs*) είναι ένας σημαντικός χώρος έρευνας που έχει προσελκύσει ιδιαίτερο ενδιαφέρον και στόχος τους είναι η επέκταση της τεχνολογίας των απλών βάσεων δεδομένων προκειμένου να υποστηρίξουν την αναπαράσταση και τα ερωτήματα κινούμενων αντικειμένων και της τροχιάς τους. Οι βάσεις κινούμενων αντικειμένων μπορούν να χρησιμοποιηθούν σε αρκετές διαδικασίες υποστήριξης αποφάσεων, όπως π.χ. εκτίμηση και πρόβλεψη κυκλοφοριακής κίνησης, ανάλυση συνθηκών κυκλοφοριακής συμφόρησης, συστήματα διαχείρισης στόλου, κ.λπ.

Στη βιβλιογραφία των χωροχρονικών βάσεων δεδομένων, υπάρχουν εργασίες σχετικά με τις παρούσες και μελλοντικές θέσεις των κινούμενων αντικειμένων και εργασίες για τις παλαιότερες θέσεις αντικειμένων. Τα συστήματα χωροχρονικών βάσεων δεδομένων αποτελούν στην ουσία συστήματα κινούμενων αντικειμένων με ιδιαίτερο ενδιαφέρον, προσφέροντας τη δυνατότητα να συνενωθούν οι ιδέες από τα συγγενικά πεδία των χωρικών και των χρονικών βάσεων δεδομένων. Η μεταβολή της γεωμετρίας μπορεί να συμβαίνει είτε σε διακριτά στάδια (π.χ. τα όρια ενός γεωτεμαχίου) είτε κατά συνεχή τρόπο. Ένα από τα βασικά ζητήματα αφορά στην κινητικότητα της πληροφορίας, τόσο από την πλευρά των αλλαγών στα δεδομένα, όσο και από την άποψη των χρηστών των οποίων οι μη καθορισμένες απαιτήσεις πρέπει να ικανοποιούνται. Επομένως, η αλλαγή της θέσης των αντικειμένων πρέπει να μελετηθεί αναλυτικά, δηλαδή να μοντελοποιηθεί με κάποιο τρόπο η διαρκής κίνησή τους, λαμβάνοντας υπόψη ότι είναι ανεπαρκής η απόδοσή της με διαφορετική τιμή για τη χρονική στιγμή και διαφορετική τιμή για τις χωρικές συνιστώσες.

Τα κινούμενα αντικείμενα είναι γεωμετρίες και μπορεί να είναι σημεία, γραμμές, επιφάνειες ή όγκοι χρονικά μεταβαλλόμενοι, ενώ τροχιά είναι η περιγραφή της κίνησης των αντικειμένων. Καθώς ο γεωγραφικός χώρος είναι συνεχής, η φυσική κίνηση περιγράφεται από μία συνεχή αλλαγή θέσης, δηλαδή μία συνάρτηση από το χρόνο στο γεωγραφικό χώρο. Η κίνηση υποδηλώνει και μία χρονική συνιστώσα, δεδομένου ότι δεν μπορεί να γίνει αντιληπτή η κίνηση παρά μόνον μέσω σύγκρισης δύο διαφορετικών χρονικών στιγμών. Συνεπώς, η τροχιά

ενός κινούμενου αντικειμένου δεν είναι απλώς παράθεση μερικών ευθυγράμμων τμημάτων, αλλά αποκτά ευρύτερη σημασιολογία, αποτελούμενη από οντότητες με πρόσθετα χαρακτηριστικά. Οι δομές που χρησιμοποιούνται για την τήρηση των τροχιών των αντικειμένων είναι σκόπιμο να διασφαλίζουν τη συνέχειά τους, δηλαδή την αλληλοδιαδοχή των τμημάτων που τα αποτελούν. Φυσικά, μια τέτοιου είδους μοντελοποίηση εξαρτάται όχι μόνο από τη μορφή των ίδιων των αντικειμένων, αλλά και από τον τύπο των ερωτημάτων που θέτουν οι χρήστες στο σύστημα.

3.2 Τύποι χωρικών αντικειμένων

Υπάρχουν δύο πράγματα στο πραγματικό κόσμο τα οποία χρήζουν μοντελοποίησης: (α) το πρώτο είναι ο ίδιος ο χώρος (π.χ. ο χωρισμός ενός κράτους σε νομούς) και (β) το δεύτερο είναι διάφορα αντικείμενα στο χώρο (π.χ. πόλεις, ποταμοί, δρόμοι, κ.τλ.), καθώς και πιθανές συνδέσεις ανάμεσα τους όπως είναι μια σιδηροδρομική διασταύρωση. Για την μοντελοποίηση των αντικειμένων στο χώρο χρειάζονται:

- Σημεία
- Γραμμές
- Πολύγωνα ή περιοχές (δηλαδή σχήματα με εμβαδό)

Τα σημεία αντιπροσωπεύουν την περίπτωση στην οποία μόνο η θέση στο χώρο παίζει ρόλο και όχι η έκταση. Οι γραμμές αποτελούν τον τρόπο σύνδεσης των αντικειμένων και άρα είναι κατάλληλες για την αναπαράσταση δρόμων, ποταμών, κ.τλ. Τέλος, τα πολύγωνα αναπαριστούν τα αντικείμενα στα οποία έχει σημασία και η έκτασή τους στο χώρο εκτός της θέσης τους. Όσον αφορά τη μοντελοποίηση του ίδιου του χώρου, χρειάζεται ένας τρόπος να χωριστεί σε τεμάχια και ένας τρόπος να χωριστεί σε δίκτυα.

Επομένως γίνεται αντιληπτό αμέσως, ότι τα χωρικά δεδομένα είναι πολύπλοκες δομές, οι οποίες μπορούν να αποτελούνται είτε από ένα απλό σημείο, είτε από πάρα πολλά σημεία αυθαίρετα τοποθετημένα στο χώρο. Κατά συνέπεια, ορισμένες φορές είναι αδύνατο η αποθήκευση τέτοιων δεδομένων σε συμβατικούς σχεσιακούς πίνακες. Επίσης, τα δεδομένα και κατ' επέκταση οι βάσεις δεδομένων τείνουν να είναι πολύ μεγάλα σε όγκο. Επομένως είναι απαραίτητη η αποθήκευσή τους στη δευτερεύουσα μνήμη (δίσκο).

3.3 Μοντέλα αναπαράστασης

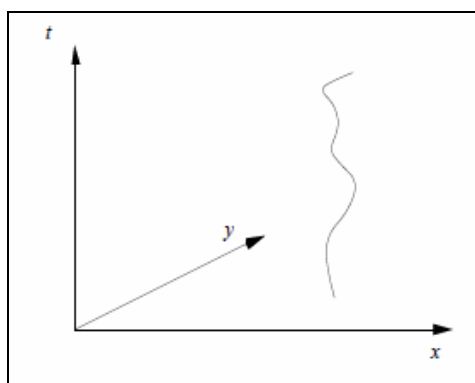
Τα αντικείμενα που μπορούν να καταγραφούν σε μια χωρική βάση δεδομένων είναι: (α) το σημείο, για το οποίο μόνο η θέση έχει ενδιαφέρον, (β) η γραμμή, με την έννοια μιας καμπύλης στο χώρο και (γ) η περιοχή, για αντικείμενα που κατέχουν μια έκταση.

Οι χωροχρονικές βάσεις δεδομένων, καταγράφοντας τα ίδια αντικείμενα, θεωρούν ότι κινούνται στο χώρο. Δηλαδή ότι τα σημεία και οι γραμμές μπορεί απλώς να μετακινούνται, αλλά οι περιοχές είναι δυνατόν να μετατοπίζονται ή και να υφίστανται αλλαγές στο σχήμα τους, δηλαδή να συρρικνώνονται ή να επεκτείνονται. Το ενδιαφέρον εστιάζεται κυρίως στα κινούμενα σημειακά και πολυγωνικά αντικείμενα και λιγότερο στα γραμμικά, καθώς οι γραμμές δεν θεωρούνται πρωτογενείς οντότητες, αλλά μπορούν να αντιμετωπιστούν ως προβολές της κίνησης άλλων αντικειμένων.

Η μοντελοποίηση των αντικειμένων μπορεί να στηριχτεί σε αφηρημένους τύπους δεδομένων (*abstract data types*), που αφορούν τόσο τα ίδια τα κινούμενα αντικείμενα όσο και βοηθητικές έννοιες χρήσιμες κατά τον ορισμό λειτουργιών πάνω στους τύπους δεδομένων. Το σύνολο των τύπων των δεδομένων και των λειτουργιών που ορίζονται μπορούν να αποτελέσουν μια άλγεβρα κινούμενων αντικειμένων και να ενταχθούν κατόπιν σε μια κατάλληλη γλώσσα ερωταποκρίσεων κάποιου ΣΔΒΔ, ώστε να προκύψει ένα πλήρες μοντέλο. Ωστόσο, ένα αφηρημένο μοντέλο παρά τη δυνατότητά του να συλλάβει ολόκληρη την εξέλιξη ενός φαινομένου, δεν μπορεί να αναπαρασταθεί υπολογιστικά στο σύνολό του, οπότε η περαιτέρω διατύπωση ενός διακριτού μοντέλου κρίνεται απολύτως απαραίτητη, περιορίζοντας την κίνηση σε πεπερασμένα μόνο στιγμιότυπα. Για παράδειγμα μια κινούμενη περιοχή μπορεί να οριστεί είτε ως μια συνεχής συνάρτηση του χρόνου με τιμές τις αντίστοιχες περιοχές (αφηρημένο μοντέλο), είτε να αναπαρασταθεί ως ένα πολυεδρικό σχήμα στις τρεις διαστάσεις (διακριτό μοντέλο). Στην πραγματικότητα και τα δύο μοντέλα κρίνονται απαραίτητα, αναπτύσσοντας πρώτα το αφηρημένο μοντέλο για τη γενική μελέτη των φαινομένων, που σχετίζονται με κινούμενα αντικείμενα και ύστερα προχωρώντας στο διακριτό για την υλοποίηση.

3.3.1 Αφηρημένο μοντέλο

Η διατύπωση ενός αφηρημένου μοντέλου της κίνησης των αντικειμένων εξασφαλίζει την κλειστότητα (*closure*) των οριζόμενων τύπων καθώς και τη συνέπεια, απλότητα και εκφραστική ικανότητα στον αποτελεσματικό χειρισμό απείρων συνόλων, αδιαφορώντας για τη δυνατότητα πεπερασμένης αναπαράστασής τους με κάποιο μαθηματικό τρόπο (βλ. σχήμα 3.1).



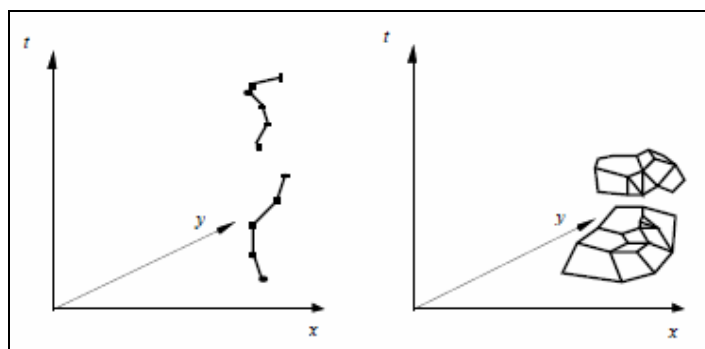
Σχήμα 3.1: Αφηρημένη αναπαράσταση κινούμενου σημείου. (Πηγή: [EGSV98])

Το μοντέλο επιχειρεί να περιγράψει τη θέση ενός σημειακού αντικειμένου μέχρι την τρέχουσα στιγμή, εκλαμβάνοντας την κίνησή του ως μετασχηματισμό (mapping) από το άπειρο πεδίο του χρόνου στο επίσης άπειρο πεδίο του χώρου.

3.3.2 Διακριτό μοντέλο

Ένα διακριτό μοντέλο της κίνησης αντικειμένων είναι στην ουσία μια ατελής προσέγγιση της κίνησης, εκφράζοντας πεπερασμένες αναπαραστάσεις κάποιων στιγμιοτύπων ενός άπειρου φαινομένου και αποδίδοντας το με ένα υποσύνολο των τιμών του αντίστοιχου αφηρημένου μοντέλου. Αντίστοιχο ζήτημα έχει τεθεί και στις χωρικές βάσεις δεδομένων, όπου το σχήμα των οντοτήτων αποδίδεται συνήθως από γραμμικές προσεγγίσεις τους, δηλαδή, μια περιοχή να εμφανίζεται ως πολύγωνο ή μια καμπύλη ως πολυγραμμή (polyline). Κατά συνέπεια, η κίνηση ενός σημειακού αντικειμένου αποδίδεται από μια πολυγραμμή στο τρισδιάστατο σύστημα αξόνων (x, y, t) , ενώ αντίστοιχα μια κινούμενη περιοχή από ένα σύνολο πολυέδρων (βλ. σχήμα 3.2).

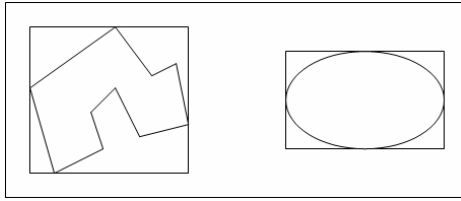
Ο ορισμός τύπων δεδομένων και λειτουργιών στο διακριτό μοντέλο αποδεικνύεται πολύ περισσότερο περίπλοκος σε σχέση με το αφηρημένο μοντέλο, εξαιτίας της αναγκαιότητας προσεγγίσεων. Πέραν της αυξημένης πολυπλοκότητας, η δημιουργία του διακριτού μοντέλου συνεπάγεται δυσεπίλυτα ζητήματα ως προς την αναπαράσταση διαφόρων τύπων δεδομένων.



Σχήμα 3.2: Διακριτές αναπαραστάσεις για κινούμενο σημείο και κινούμενη περιοχή. (Πηγή: [EGSV98])

3.4 Χωρικοί τελεστές

Πολύ συχνά, λόγω της πολυπλοκότητας των γεωμετριών των αντικειμένων σε μια χωρική βάση δεδομένων, οι τελεστές στην αρχή προσπαθούν να εντοπίσουν κάποιες αρχικές λύσεις χρησιμοποιώντας τα περιβάλλοντα παραλληλόγραμμα (bounding box), τα οποία περικλείουν μέσα τους την σύνθετη γεωμετρία (βλ. σχήμα 3.3). Αυτά έχουν το ελάχιστο εμβαδό και για αυτό ονομάζονται ελάχιστα περιβάλλοντα παραλληλόγραμμα. Αφού εφαρμοστούν τα ελάχιστα συνοριακά κουτιά στα αντικείμενα, οι τελεστές ελέγχουν μία προς μία τις εφικτές λύσεις χρησιμοποιώντας την ακριβή γεωμετρία των αντικειμένων.



Σχήμα 3.3: Παράδειγμα ελάχιστων περιβαλλόντων παραλληλόγραμμων σε αντικείμενα.

Μεταξύ των διαδικασιών που προσφέρονται από τις χωρικές άλγεβρες, οι χωρικοί τελεστές είναι οι σημαντικότεροι. Επίσης καθιστούν πιθανό το ενδεχόμενο να ζητήσουν όλα τα χωρικά αντικείμενα σε ένα ερώτημα αντικειμένου, όπως για παράδειγμα όλα τα αντικείμενα μέσα σε ένα παράθυρο. Μπορούν να διαχωριστούν οι ακόλουθες κατηγορίες τελεστών:

Οι τοπολογικοί τελεστές, όπως π.χ. γειτνίαση (adjacent), εντός (inside), χωρίς συνάφεια (disjoint), είναι αμετάβλητες κάτω από τοπολογικούς μετασχηματισμούς όπως είναι η μεγέθυνση - σμίκρυνση και η περιστροφή.

Τελεστές κατεύθυνσης, για παράδειγμα, πάνω (above), κάτω (below), ή βόρεια από (north of), νοτιοδυτικά από (southwest of), κ.λπ.

Μετρικοί τελεστές, για παράδειγμα η απόσταση να είναι μικρότερη από 100 μέτρα (distance < 100m).

Μεταξύ αυτών, οι τοπολογικοί τελεστές είναι οι πιο θεμελιώδεις και έχουν μελετηθεί σε κάποιο βάθος (βλ. σχήμα 3.4). Παρακάτω παρατίθενται οι πιο συχνά συναντώμενοι τελεστές χωρικών δεδομένων, μαζί με την επεξήγησή τους:

Τελεστής ισότητας (exact match query). Χρησιμοποιείται όταν δίνεται ένα χωρικό αντικείμενο με μία γεωμετρία και ζητείται να βρεθούν όλα τα αντικείμενα με την ίδια ακριβώς γεωμετρία.

Τελεστής σημείου (point query). Χρησιμοποιείται όταν δίνεται ένα σημείο στο χώρο και ζητείται να βρεθούν όλα τα αντικείμενα που το περιέχουν.

Τελεστής περιοχής (window query, range query). Χρησιμοποιείται όταν δίνεται ένα παράθυρο n-διαστάσεων και ζητείται να βρεθούν όλα τα αντικείμενα που έχουν ένα τουλάχιστον κοινό σημείο με την περιοχή (“χωρικό παράθυρο”).

Τελεστής τομής (intersection query, overlap query). Χρησιμοποιείται όταν δίνεται ένα χωρικό αντικείμενο και ζητείται να βρεθούν όλα τα αντικείμενα που έχουν ένα τουλάχιστον κοινό σημείο με αυτό.

Τελεστής παθητικού εγκλεισμού (enclosure query). Χρησιμοποιείται όταν δίνεται ένα χωρικό αντικείμενο και ζητείται να βρεθούν όλα τα αντικείμενα που το περιέχουν.

Τελεστής ενεργητικού εγκλεισμού (containment query). Χρησιμοποιείται όταν δίνεται ένα χωρικό αντικείμενο και ζητείται να βρεθούν όλα τα αντικείμενα που αυτό περιέχει.

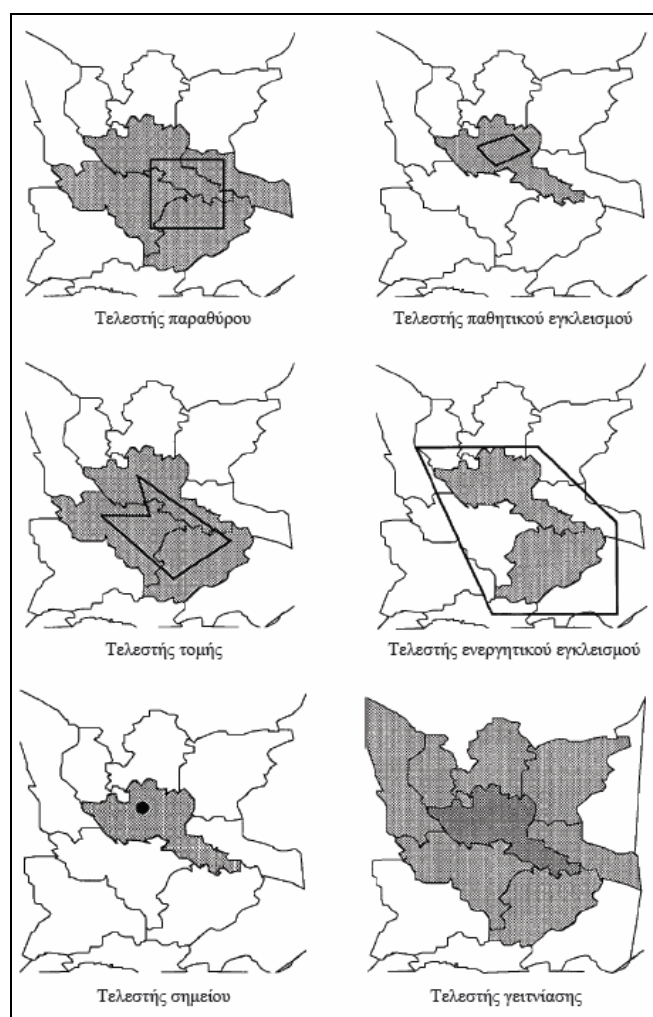
Τελεστής γειτνίασης (adjacent query). Χρησιμοποιείται όταν δίνεται ένα χωρικό αντικείμενο και ζητείται να βρεθούν όλα τα αντικείμενα με τα οποία δεν έχει κανένα κοινό εσωτερικό σημείο, αλλά έχει κάποια κοινά εξωτερικά σημεία, δηλαδή σημεία με τα οποία εφάπτεται κατά κάποιο τρόπο. Ουσιαστικά ζητείται να βρεθούν τα αντικείμενα που συνορεύουν με το δοσμένο.

Τελεστής εγγύτερου γείτονα (nearest-neighbor query). Χρησιμοποιείται όταν δίνεται ένα χωρικό αντικείμενο και ζητείται να βρεθεί ένα αντικείμενο το

οποίο έχει την μικρότερη απόσταση από αυτό ή ορισμένα k αντικείμενα τα οποία είναι πιο κοντά στο δοσμένο. Η απόσταση μεταξύ χωρικών αντικειμένων συνήθως καθορίζεται ως η απόσταση μεταξύ των κοντινότερων σημείων τους. Κοινές συναρτήσεις για την απόσταση μεταξύ σημείων, αποτελούν οι Ευκλείδεια απόσταση και η απόσταση Μανχάταν.

Τελεστής σύνδεσης (spatial join query). Χρησιμοποιείται όταν υπάρχει η ανάγκη σύνδεσης ενός αντικειμένου με βάση κάποιο χωρικό χαρακτηριστικό των δεδομένων μιας βάσης δεδομένων. Ουσιαστικά χρησιμοποιείται όταν υπάρχουν δύο δοσμένα σύνολα αντικειμένων, και ζητείται να βρεθούν τα ζεύγη των αντικειμένων από το ένα και το άλλο που ικανοποιούν μια προκαθορισμένη συνθήκη (π.χ. απόσταση μικρότερη από 50m), η οποία μπορεί να είναι κάποιο ερώτημα ή να περιέχει άλλα ερωτήματα.

Όλοι οι παραπάνω τελεστές χρησιμοποιούνται σε ερωτήματα, τα οποία εκφράζονται στην άλγεβρα και στο ΣΔΒΔ μέσω των λειτουργιών που έχουν αναπτυχθεί. Αντίστοιχα, μπορούν να επεκταθούν στο πεδίο του χρόνου για κινούμενα χωρικά αντικείμενα μέσω των αντίστοιχων λειτουργιών. Η χωρική διάσταση μπορεί να αναφέρεται είτε στο παρελθόν είτε στο παρόν είτε στο άμεσο μέλλον υποθέτοντας και κάποια τεχνική για την πρόβλεψη.



Σχήμα 3.4: Διάφοροι χωρικοί τελεστές. [Πηγή: GG98]

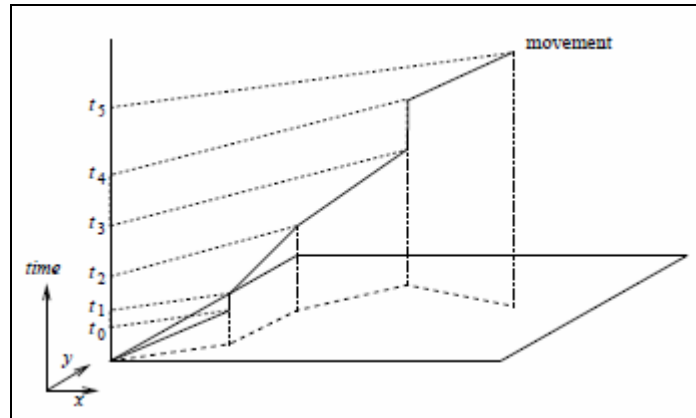
3.5 Τροχιές κινούμενων αντικειμένων

Μία ιστορική τροχιά μπορεί απλά να οριστεί ως μία συνάρτηση από το χρόνο προς ένα γεωγραφικό χώρο, όμως η περιγραφή, η αναπαράσταση και η αντιμετώπισή της είναι πιο περίπλοκα θέματα. Μία τροχιά είναι η καταγραφή της κίνησης κάποιου αντικειμένου, δηλαδή η καταγραφή των θέσεων του αντικειμένου σε συγκεκριμένες χρονικές στιγμές και η καμπύλη της τροχιάς προκύπτει εφαρμόζοντας μεθόδους παρεμβολής στα σημεία καταγραφής. Ωστόσο, οποιαδήποτε μέθοδο παρεμβολής και αν χρησιμοποιηθεί, η καμπύλη που προκύπτει είναι απλώς μία δυνητική αναπαράσταση της πραγματικής τροχιάς και η αβεβαιότητά της εξαρτάται από τα πιθανά σφάλματα που ενυπάρχουν κατά την αρχική καταγραφή των θέσεων του αντικειμένου. Στη βιβλιογραφία έχει προταθεί πλήθος μοντέλων για τη διαμόρφωση και την κατάλληλη αντιμετώπιση αυτής της αβεβαιότητας.

Αξιολογώντας το γεγονός ότι τα δεδομένα θέσης μπορεί να μεταβάλλονται χρονικά, η αντίστοιχη βάση δεδομένων πρέπει να περιλαμβάνει όλο το ιστορικό της εξέλιξης και το ΣΔΒΔ πρέπει να έχει τη δυνατότητα να ανατρέξει στο παρελθόν σε οποιοδήποτε χρονικό αποτύπωμα και να μπορεί να ανακαλέσει τις πληροφορίες της βάσης δεδομένων σε εκείνη τη χρονική στιγμή. Τέτοια ζητήματα αντιμετωπίζονται επιτυχώς από τις βάσεις δεδομένων κινούμενων αντικειμένων.

Καταγράφοντας λοιπόν τις διαδοχικές θέσεις ενός κινούμενου αντικειμένου προκύπτει η τροχιά του. Προκειμένου να είναι ακριβής η αναπαράσταση της πορείας του, πρέπει η θέση του αντικειμένου να καταγράφεται συνεχώς με την πάροδο του χρόνου, το οποίο είναι πρακτικά αδύνατο. Οι σημειακές θέσεις των αντικειμένων εκ των πραγμάτων λαμβάνονται από τις συσκευές καταγραφής σε διακριτές χρονικές στιγμές (π.χ. κάθε λίγα δευτερόλεπτα), επομένως η τροχιά αντιμετωπίζεται ως δειγματοληψία της πραγματικής κίνησης. Οι συντεταγμένες των θέσεων συνήθως είναι συνδεδεμένες με τα χρονόσημα στα οποία σημειώθηκαν, η κίνηση ενός αντικειμένου στο επίπεδο μπορεί να αναπαρασταθεί με την τροχιά του σε τρισδιάστατο σύστημα αξόνων (βλ. σχήμα 3.6), το οποίο συντίθεται από δύο χωρικές (x,y) συντεταγμένες και μία χρονική (t) συντεταγμένη.

Μια πρώτη πρακτική προσέγγιση για την αναπαράσταση της τροχιάς του αντικειμένου, είναι η αποθήκευση του δείγματος θέσεων με τη μορφή εγγραφών σε έναν πίνακα βάσης δεδομένων. Ωστόσο, κάτι τέτοιο σημαίνει ότι η κίνηση του αντικειμένου μεταξύ των σημείων δειγματοληψίας είναι απροσδιόριστη. Συνεπώς, γίνεται απόπειρα αναπαράστασης ολόκληρης της κίνησης των αντικειμένων με κάποια μέθοδο, όπως η μέθοδος της παρεμβολής. Μόνον έτσι είναι δυνατόν να απαντηθούν ερωτήματα που αναφέρονται σε χρονικές στιγμές για τις οποίες δεν υπάρχουν καταγεγραμμένες θέσεις. Εφόσον η συνεχής καταγραφή της τροχιάς είναι πρακτικά αδύνατη, η εύρεση της θέσης αντικειμένου για κάποια ενδιάμεση χρονική στιγμή προκύπτει με παρεμβολή, μιας και η γραμμική παρεμβολή κρίνεται επαρκής για τις περισσότερες εφαρμογές. Εναλλακτικά, πιο πολύπλοκες τεχνικές, όπως τα πολυώνυμα splines, μπορούν να χρησιμοποιηθούν για την καλύτερη εκτίμηση της θέσης του αντικειμένου σε χρονικές στιγμές για τις οποίες δεν υπάρχει πληροφορία.



Σχήμα 3.5: Αναπαράσταση τροχιάς ενός κινούμενου αντικειμένου στον τρισδιάστατο χώρο (x, y, t) . (Πηγή: [Pfo02])

Θεωρώντας γραμμική παρεμβολή, τα γνωστά σημεία θεωρούνται άκρα ευθυγράμμων τμημάτων και η τροχιά προσεγγίζεται από μία τεθλασμένη πολυγραμμή (polyline) που απεικονίζει την τροχιά του αντικειμένου στον τρισδιάστατο χώρο (x, y, t) . Αυτή η πολυγραμμή αποθηκεύεται τελικά σε μια χωρική βάση δεδομένων και οι συντεταγμένες του χώρου και του χρόνου συνδυάζονται ώστε να αποτελέσουν κοινό σύστημα αναφοράς. Όπως φαίνεται στο σχήμα 3.5, η διακεκομμένη γραμμή συμβολίζει την προβολή (το ίχνος) της τροχιάς του κινούμενου αντικειμένου στο χωρικό επίπεδο (x, y) .

Συνοψίζοντας, υπάρχουν τρία είδη κίνησης των αντικειμένων και αντίστοιχα των τροχιών που πρέπει να είναι σε θέση να μπορεί να καταγράφει ένα σύστημα βάσεων κινούμενων αντικειμένων: (α) την κίνηση των αντικειμένων στο παρελθόν, δηλαδή τις ιστορικές τροχιές τους, (β) την κίνηση στο παρόν, δηλαδή την τρέχουσα τροχιά τους και (γ) την κίνηση των αντικειμένων στο μέλλον, δηλαδή την πρόβλεψη της τροχιάς τους. Μεγάλο ενδιαφέρον στο πεδίο των κινούμενων αντικειμένων παρουσιάζει η πρόβλεψη μελλοντικών θέσεων του κινούμενου αντικειμένου. Η συστηματική πρόβλεψη θέσης αποτελεί μεγάλη πρόκληση και πρέπει να λαμβάνει υπόψη πολλές παραμέτρους.

3.5.1 Κατηγορίες κίνησης

Η κίνηση των αντικειμένων (σημειακών, γραμμικών, επιφανειών) είναι πολύ πιθανόν να διέπεται από περιορισμούς, αφού δεν είναι λογικό να υποτεθεί ότι αυτά κινούνται ανεμπόδιστα σε έναν δυσδιάστατο ευκλείδειο χώρο. Έτσι, οι κινήσεις μπορούν να διακριθούν σε τρεις βασικές κατηγορίες:

Κίνηση χωρίς περιορισμούς. Η κίνηση χωρίς περιορισμούς, όπως για παράδειγμα η κίνηση των σκαφών στη θάλασσα, είναι περίπτωση η οποία εξετάζεται κυρίως από τις μεθόδους προσπέλασης χωροχρονικών στοιχείων και σπάνια ενδιαφέρει τα πραγματικά συστήματα, αφού προϋποθέτει ως υπόβαθρο ένα τέλειο ευκλείδειο επίπεδο.

Κίνηση υπό περιορισμούς. Η κίνηση υπό περιορισμούς, όπως για παράδειγμα η κίνηση των πεζών ή των οχημάτων, είναι περίπτωση στην οποία

θεωρείται ότι υφίστανται φυσικά εμπόδια που δυσχεραίνουν ή απαγορεύουν τη μετάβαση (π.χ. κτίρια ή υδάτινες επιφάνειες). Αυτά τα εμπόδια θεωρούνται ως χώρος, όπου υπάρχουν περιορισμένες ή ανύπαρκτες δυνατότητες κίνησης, πάντοτε σε σχέση με τις δύο χωρικές διαστάσεις, ενώ η χρονική διάσταση παραμένει ανεπηρέαστη. Τα εμπόδια που προβάλλονται στην κίνηση των αντικειμένων εκτός από στατικά μπορεί να είναι δυναμικά, δηλαδή για παράδειγμα η κυκλοφοριακή συμφόρηση σε ένα τμήμα του οδικού δικτύου της Αθήνας μπορεί να θεωρηθεί ως ανασταλτικός παράγοντας της ομαλής εξέλιξης της τροχιάς των οχημάτων.

Κίνηση σε δίκτυα μεταφορών. Η κίνηση σε δίκτυα μεταφορών, όπως για παράδειγμα σε οδικό ή σε σιδηροδρομικό δίκτυο, είναι ουσιαστικά μια υποπερίπτωση της κίνησης υπό περιορισμούς, αφού η μετατόπιση των αντικειμένων συμβαίνει αποκλειστικά πάνω στο δίκτυο και όχι γενικά στο δυσδιάστατο επίπεδο. Σε αυτήν την περίπτωση, οι χρήστες δεν ενδιαφέρονται τόσο πολύ για τις απόλυτες συντεταγμένες της θέσης των αντικειμένων, ούτε για την έκταση που καταλαμβάνουν στο χώρο τα στοιχεία του δικτύου (π.χ. το πλάτος των δρόμων). Σημασία έχει κυρίως η σχετική θέση των κινούμενων αντικειμένων ως προς τα τμήματα του δικτύου. Επιπλέον, τόσο οι προελεύσεις όσο και οι προορισμοί των κινούμενων αντικειμένων είναι σημεία του δικτύου και μπορούν να αξιοποιηθούν συγκεκριμένα πρότυπα κίνησης σε αυτά.

Συνεπώς, ο διαχωρισμός της κίνησης στις ανωτέρω κατηγορίες διευκολύνει σημαντικά το έργο της δεικτοδότησης (indexing) των αντικειμένων, μέσω των ενδεδειγμένων δομών δεδομένων που έχουν προταθεί για την κάθε περίπτωση και αξίζει να σημειωθεί ότι η μέθοδος που θα επιλεγεί για τη δεικτοδότηση εξαρτάται κατά σημαντικό βαθμό από τον τύπο των ερωτημάτων των χρηστών, τους οποίους εξυπηρετεί το σύστημα.

3.5.2 Αβεβαιότητα αναπαράστασης τροχιάς

Η αβεβαιότητα (uncertainty), ορίζεται ως η μέτρηση της διαφοράς μεταξύ των πραγματικών περιεχομένων μιας βάσης δεδομένων και των περιεχομένων που θα είχε δημιουργήσει ο χρήστης ή η εφαρμογή μέσω παρατήρησης της πραγματικότητας. Στοιχεία τα οποία μπορούν να αποτελέσουν πηγή αβεβαιότητας είναι τα εξής:

- Ατελής παρατήρηση του πραγματικού κόσμου
- Ελλιπής γλώσσα αναπαράστασης
- Άγνοια ή έλλειψη αποτελεσματικότητας

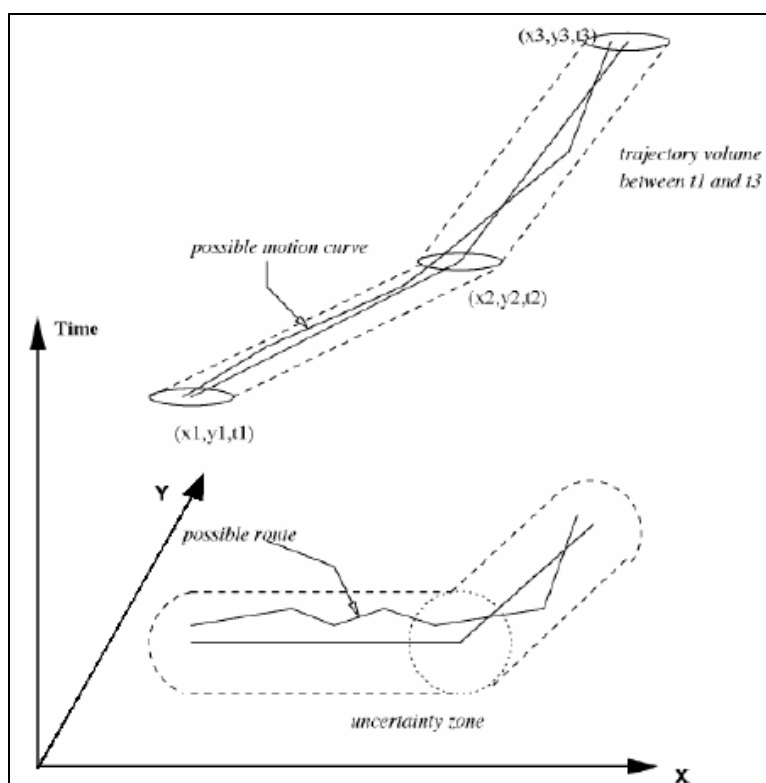
Η θέση ενός κινούμενου αντικειμένου, όπως καταγράφεται στο σύστημα, είναι εξ ορισμού ανακριβής, εφόσον είναι αδύνατο να ταυτίζεται με την πραγματική θέση σε κάθε χρονική στιγμή και γι' αυτό εμφανίζεται πάντοτε ορισμένη απόκλιση. Αυτό δημιουργεί προβλήματα σε ερωτήματα που μπορεί να τεθούν. Επιπρόσθετα, είναι υπαρκτή η πιθανότητα κάποιο αντικείμενο να

αποσυνδεθεί προσωρινά από το κεντρικό σύστημα και να μην μπορεί να στείλει πληροφορίες σχετικά με την εξέλιξη της κίνησής του.

Επομένως, εφόσον η τροχιά ενός κινούμενου αντικειμένου προκύπτει βάσει δείγματος μόνο των θέσεων από τις οποίες διήλθε, είναι προφανές ότι η αναπαράσταση της κίνησης περιέχει σημαντικό βαθμό αβεβαιότητας. Ακόμη και αν είναι εντελώς γνωστές δύο διαδοχικές σημειακές θέσεις της πορείας του αντικειμένου, δεν είναι σίγουρο ότι η ενδιάμεση διαδρομή ήταν όντως ένα ευθύγραμμο τμήμα, όπως υποτίθεται βάσει της μεθόδου της γραμμικής παρεμβολής, αλλά μπορεί ενδεχομένως να είναι μια τεθλασμένη ή μια καμπύλη (βλ. σχήμα 3.6).

Συνεπώς, η τροχιά, με τον τρόπο που καταγράφεται, διαφέρει από την πραγματική κίνηση, εξαιτίας του σφάλματος δειγματοληψίας, η ποσοτικοποίηση του οποίου αποσκοπεί στον περιορισμό της έκτασης των πιθανών θέσεων όπου το αντικείμενο μπορεί να είχε κινηθεί ενδιάμεσα.

Εκτός αυτού, δεν μπορεί να δοθεί εγγύηση ότι τα ίδια τα σημεία δειγματοληψίας είναι απολύτως ακριβή. Είναι αναπόφευκτο ότι κατά τη διαδικασία της καταγραφής των συντεταγμένων υπεισέρχονται σφάλματα μέτρησης, κυρίως από τις συσκευές GPS, αλλά κι άλλες αιτίες όπως π.χ. θόρυβος, παρεμβολές, διακοπές στη μετάδοση κλπ. Έτσι η ίδια η θέση του αντικειμένου δεν μπορεί να προσδιοριστεί επακριβώς, αλλά κυμαίνεται εντός κάποιου περιωρίου σφάλματος. Το σφάλμα δεν είναι φυσικά το ίδιο για όλα τα σημεία, αλλά επηρεάζεται ευθέως από το ρυθμό δειγματοληψίας, δηλαδή τη συχνότητα καταγραφής των συντεταγμένων: όσο συχνότερη είναι η ύπαρξη μετρήσεων των θέσεων, τόσο πληρέστερη είναι η γνώση της πορείας των αντικειμένων και τόσο μικρότερο το σφάλμα της τροχιάς.



Σχήμα 3.6: Μοντέλο αβεβαιότητας κινούμενου αντικειμένου. (Πηγή: [Fre08])

Κατ' επέκταση, η πυκνότητα των σημείων εξαρτάται και μπορεί να ρυθμιστεί ανάλογα με την ταχύτητα κίνησης αλλά και την μέγιστη επιτρεπτή απόσταση των δειγμάτων που λαμβάνονται.

Μια λύση στο πρόβλημα της αβεβαιότητας στην αναπαράσταση των δεδομένων, είναι να καθοριστεί η μέγιστη επιτρεπόμενη ακρίβεια των δεδομένων τροχιάς που θα τροφοδοτούν την βάση κινούμενων αντικειμένων, βάσει της απαιτούμενης ακρίβειας στα αποτελέσματα των ερωτημάτων. Κατόπιν, μπορούν οι χρήστες μέσω του συστήματος να επιλέξουν τη χρήση της κατάλληλης, περισσότερο ή λιγότερο ακριβούς, που σημαίνει επίσης και περισσότερο ή λιγότερο δαπανηρής μεθόδου εντοπισμού που θα χρησιμοποιηθεί για τα δεδομένα που θα τροφοδοτήσουν το σύστημα. Συνεπώς όλες οι παραπάνω απαιτήσεις θα μπορούσαν να ικανοποιηθούν μέσω ενός μοντέλου που προβλέπει το σφάλμα που εισάγεται στα αποτελέσματα των ερωτημάτων βάσει γνωστών ιδιοτήτων των ερωτημάτων και του συνόλου των δεδομένων, χωρίς να εκτελείται το ερώτημα.

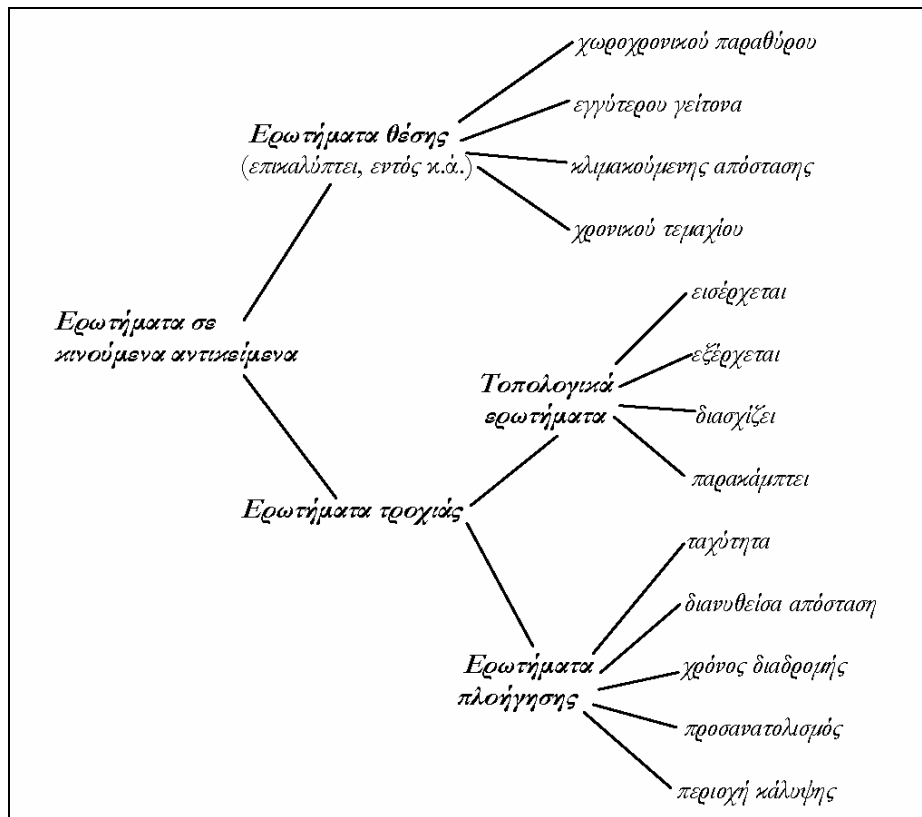
3.6 Κατηγορίες ερωτημάτων

Εφόσον η κύρια πληροφορία που καταγράφεται από το σύστημα αναφέρεται στη θέση των κινούμενων αντικειμένων, προφανώς οι χρήστες θα πρέπει να μπορούν να λαμβάνουν απαντήσεις σε σχετικά ερωτήματα θέσης (*location-based* ή *coordinated-based queries*). Υπάρχουν όμως και ερωτήματα τα οποία διερευνούν ολόκληρη την τροχιά των κινούμενων αντικειμένων ή έστω συμπαγή μέρη της (*trajectory-based queries*) και αναφέρονται είτε στην τοπολογία της κίνησης (*τοπολογικά ερωτήματα – topological queries*), είτε σε δευτερογενείς πληροφορίες (π.χ. ταχύτητα) οι οποίες σχετίζονται με την πλοήγηση των αντικειμένων (*ερωτήματα πλοήγησης – navigational queries*).

Το σύστημα πρέπει να είναι σε θέση να απαντήσει και σε καθαρά χωρικά ή χρονικά ερωτήματα, αγνοώντας κατά περίπτωση τις άλλες διαστάσεις και οφείλει να είναι σε θέση καταγράψει την ιστορική εξέλιξη της κίνησης κάποιου κινούμενου αντικειμένου, οπότε πρέπει να υπάρχει εξασφάλιση της συνέπειας των διαδοχικών θέσεων του.

3.6.1 Ερωτήματα θέσης

Οι κύριοι τύποι ερωτημάτων που υποβάλλονται σε χωροχρονικές βάσεις δεδομένων στηρίζονται στη εξέταση των συντεταγμένων των σημειακών θέσεων με τις χρονικές στιγμές που έχουν καταγραφεί και ονομάζονται ερωτήματα θέσης. Τα ίδια ερωτήματα μπορούν να εφαρμοστούν και στα συστήματα κινούμενων αντικειμένων, θεωρώντας την τροχιά τους ως ακολουθία σημειακών θέσεων. Τα ερωτήματα θέσης αφορούν την κύρια πληροφορία που καταγράφουν τα συστήματα και διακρίνονται σε:



Σχήμα 3.7: Κατηγορίες ερωτημάτων σε κινούμενα αντικείμενα. (Πηγή: [Pat03])

Ερωτήματα χωροχρονικού παραθύρου (range queries). Εφαρμόζεται ένα χωρικό παράθυρο συγκεκριμένων διαστάσεων και ένα χρονικό διάστημα ορισμένου εύρους, ώστε να επιλέγονται τα κατάλληλα τμήματα τροχιών που το τέμνουν.

Ερωτήματα εγγύτερου γείτονα (nearest neighbor queries). Εύρεση των κοντινότερων αντικειμένων (καθορισμένου αριθμού), για δεδομένο χρονικό διάστημα ή χρονική στιγμή, στο αντικείμενο ενδιαφέροντος ή εύρεση των αντικειμένων που βρίσκονται γύρω του εντός ορισμένης ακτίνας. Η βασική διαφοροποίηση σε σχέση με παρόμοια ερωτήματα σε χωρικές βάσεις δεδομένων έγκειται ακριβώς στη χρονική διάσταση, αφού τα χρονόσημα (timestamps) των αντικειμένων που εξετάζονται θα πρέπει να συμβαδίζουν. Οι γείτονες είναι αυτοί οι οποίοι βρίσκονται κοντά με βάση κάποια απόσταση (ευκλείδεια, manhattan, κ.α.) σε αντίστοιχα χρονόσημα.

Ερωτήματα κλιμακούμενης απόστασης (sorted-distance queries). Αναζήτηση των κοντινότερων αντικειμένων (μη καθορισμένου αριθμού) στο αντικείμενο ενδιαφέροντος ή σε ένα σημείο αναφοράς. Επιπλέον, το αποτέλεσμα θα πρέπει να επιστρέφει τον κατάλογο των αντικειμένων αλλά και τις σχετικές αποστάσεις σε διάταξη.

Ερωτήματα χρονικού τεμαχίου (time-slice queries). Είναι ερωτήματα με τα οποία αναζητούνται οι θέσεις των κινούμενων αντικειμένων για ένα συγκεκριμένο χρονικό παράθυρο ή μια συγκεκριμένη χρονική στιγμή στο παρελθόν. Ουσιαστικά, πρόκειται για υποπερίπτωση των ερωτημάτων χωροχρονικού παραθύρου, όπου το παράθυρο έχει καθορισμένο ή μηδενικό χρονικό εύρος.

3.6.2 Ερωτήματα τροχιάς

Τα ερωτήματα τροχιάς διακρίνονται σε τοπολογικά ερωτήματα και ερωτήματα πλοήγησης. Τα τοπολογικά ερωτήματα είναι ερωτήματα που αφορούν την συμπεριφορά της τροχιάς στο χρόνο σε σχέση με κάποια χωρική έκταση, για παράδειγμα ένα κινούμενο αντικείμενο μπορεί να εισέρχεται σε ένα χωρικό παράθυρο ή να εξέρχεται από αυτό ή μπορεί να το διασχίζει ή να το παρακάμπτει. Αντίθετα τα ερωτήματα πλοήγησης αφορούν διάφορες παραμέτρους της τροχιάς του αντικειμένου, όπως η ταχύτητα, η διανυθείσα απόσταση, κ.τλ.

A. Ερωτήματα τοπολογίας

Τα τοπολογικά ερωτήματα αφορούν τις χωροχρονικές σχέσεις της τροχιάς των αντικειμένων με άλλα γειτονικά. Η σημασιολογία τους βασίζεται στο γνωστό μοντέλο τοπολογικών σχέσεων (9-intersection model) που υιοθετείται από τα χωρικά συστήματα βάσεων δεδομένων, φυσικά επεκτείνοντας το με τη χρονική διάσταση. Έτσι, εκτός των βασικών κατηγορημάτων για ένα αντικείμενο, όπως: εντός (*inside*), επικαλύπτει (*overlap*), καλύπτεται (*covered by*), καλύπτει (*covers*), περιλαμβάνει (*contains*), συναντά (*meet*), ισούται (*equal*), και χωρίς συνάφεια (*disjoint*), εξετάζονται τα σύνδετα κατηγορήματα: εισέρχεται (*enter*), εξέρχεται (*leave*), διασχίζει (*cross*) και παρακάμπτει (*bypass*).

Για την διερεύνηση της πιθανής σχέσης της τροχιάς ενός αντικειμένου με μια περιοχή του χώρου, αναγκαστικά ελέγχονται περισσότερα από ένα τμήματα της τροχιάς του αντικειμένου. Ο κύριος τύπος ερωτημάτων που εμπίπτει στην κατηγορία αυτή είναι τα ερωτήματα χωροχρονικής σύνδεσης (*spatiotemporal join*), με χαρακτηριστικότερη περίπτωση τη διερεύνηση της εγγύτητας (*proximity*) δύο ή περισσότερων τροχιών αντικειμένων.

B. Ερωτήματα πλοήγησης

Η διαφοροποίηση αυτών των ερωτημάτων από τα τοπολογικά είναι ότι οι απαντήσεις που αναζητούνται δεν είναι καταχωρημένες (όπως π.χ. οι συντεταγμένες των θέσεων σε διακριτές χρονικές στιγμές), αντιθέτως υπολογίζονται βάσει της τροχιάς του αντικειμένου. Για παράδειγμα ένα μέγεθος το οποίο μπορεί να υπολογιστεί είναι η ταχύτητα (υπολογίζεται ως πηλίκο της απόστασης που διανύθηκε προς το αντίστοιχο χρονικό διάστημα). Έτσι, π.χ. εάν ζητείται η μέση ή η μέγιστη ταχύτητα κατά μήκος της τροχιάς, υπολογίζονται αρχικά οι ταχύτητες κίνησης του αντικειμένου σε όλα τα επιμέρους τμήματα και ύστερα υπολογίζεται ο μέσος όρος ή η μέγιστη τιμή.

Άλλα παράγωγα μεγέθη που έχουν χρησιμότητα σε εφαρμογές κινούμενων αντικειμένων είναι η διανυθείσα απόσταση και ο χρόνος διαδρομής. Βάσει αυτών, των μεγεθών είναι δυνατόν να διαπιστωθεί κατά πόσον το αντικείμενο συνεχίζει την πορεία του ή παραμένει ακίνητο. Επίσης, ο προσανατολισμός της κίνησης ενός αντικειμένου αντιπροσωπεύεται από το διάνυσμα μεταξύ δύο συγκεκριμένων θέσεων, ενώ η περιοχή κάλυψης προκύπτει από το κυρτό περίγραμμα της τροχιάς.

3.6.3 Σύνθετα ερωτήματα

Οι παραπάνω τύποι ερωτημάτων μπορούν να συνδυαστούν και να παράγουν σύνθετα ερωτήματα. Ερωτήματα που παρέχουν τη δυνατότητα στους χρήστες να υποβάλλουν συνδυασμούς ερωτημάτων, στην προσπάθειά τους να λάβουν απαντήσεις σε σχέση με τις τροχιές των αντικειμένων. Κάτι τέτοιο εμπλέκει το ζήτημα της επιλογής ορισμένων τροχιών για τα αντικείμενα ενδιαφέροντος, καθώς και επιμέρους τμημάτων τους μέσω σύνθετων κριτηρίων.

3.7 Δεικτοδότηση - Ευρετήρια

Σε εφαρμογές κινούμενων αντικειμένων στο χώρο, αποδίδεται μεγάλη σημασία στην έγκαιρη και διαρκή παρακολούθηση κρίσιμων παραμέτρων της κίνησης (θέση, ταχύτητα, προσανατολισμός κ.ά.), ώστε να παρέχεται στους χρήστες των συστημάτων η κατάλληλη ενημέρωση. Όμως η εποπτεία σε πραγματικό χρόνο ενός μεγάλου αριθμού κινούμενων αντικειμένων θέτει δύσκολα προβλήματα σχετικά με την αποτελεσματική προσπέλαση της πληροφορίας και κυρίως λόγω του μεγάλου όγκου της πληροφορίας, η χρήση δίσκου είναι αναπόφευκτη. Συνεπώς η παρουσία τεραστίου πλήθους δεδομένων, αποτελεί σημαντικό πρόβλημα για τις βάσεις κινούμενων αντικειμένων και ο μόνος τρόπος αντιμετώπισης αυτής της κατάστασης είναι η εκμετάλλευση εξειδικευμένων μεθόδων προσπέλασης που χρησιμοποιούνται για τη χωροχρονική δεικτοδότηση τροχιών.

Επομένως η επινόηση κατάλληλων δομών δεδομένων, δηλαδή δεικτών ή ευρετηρίων (*indexes*) για την απρόσκοπτη επεξεργασία των σχετικών ερωτημάτων είναι άκρως αναγκαία και απαραίτητη. Οι τεχνικές που έχουν προταθεί ακολουθούν κυρίως δυο τάσεις: (α) η πρώτη τάση είναι αυτή που συμπεριφέρεται στην χρονική διάσταση ως μία ακόμα διάσταση (*time-oblivious*), ενώ (β) η δεύτερη αναφέρεται ως κινητικές δομές δεδομένων (*kinetic data structures*) και επιχειρεί την κατασκευή ενός δυναμικού δείκτη στην κύρια μνήμη για την παρακολούθηση της κίνησης σημειακών αντικειμένων.

Υπάρχουν ποικίλες απαιτήσεις που οι πολυδιάστατες μέθοδοι προσπέλασης πρέπει να καλύψουν, βασισμένες στις ιδιότητες των χωρικών δεδομένων και των εφαρμογών τους [GG98]:

Δυναμική. Δεδομένου ότι τα αντικείμενα εισάγονται και διαγράφονται από τη βάση δεδομένων σε οποιαδήποτε δεδομένη διάταξη, οι μέθοδοι προσπέλασης πρέπει συνεχώς να παρακολουθούν τις αλλαγές.

Δευτεροβάθμια/τριτοβάθμια διαχείριση αποθήκευσης. Παρά την ανάπτυξη των κύριων μνημών, δεν είναι συχνά δυνατό να κρατηθεί η πλήρης βάση δεδομένων στην κύρια μνήμη. Επομένως, οι μέθοδοι προσπέλασης πρέπει να ενσωματώσουν δευτεροβάθμια και τριτοβάθμια αποθήκευση.

Ευρύτερες υποστηρικτικές διαδικασίες. Οι μέθοδοι προσπέλασης δεν πρέπει να υποστηρίζουν μόνο έναν ιδιαίτερο τύπο λειτουργίας, όπως η ανάκτηση, εις βάρος άλλων διαδικασιών, όπως η διαγραφή.

Ανεξαρτησία από την εισαγωγή δεδομένων και την σειρά εισόδου τους. Οι μέθοδοι προσπέλασης πρέπει να διατηρούν την αποδοτικότητά τους ακόμα και όταν τα δεδομένα εισόδου είναι ιδιαίτερα μεταβλητά ή η σειρά εισαγωγής τους αλλάζει. Αυτό το σημείο είναι ιδιαίτερα σημαντικό για τα στοιχεία που διανέμονται διαφορετικά κατά μήκος των διάφορων διαστάσεων.

Απλότητα. Οι πολύπλοκες μέθοδοι προσπέλασης με πολλές ειδικές περιπτώσεις, είναι συχνά επιρρεπείς σε λάθη για να χρησιμοποιηθούν και όχι αρκετά δυνατές για εφαρμογές μεγάλης κλίμακας.

Επεκτασιμότητα. Οι μέθοδοι προσπέλασης πρέπει να μπορούν να προσαρμόζονται καλά στην αύξηση του όγκου αποθήκευσης των δεδομένων στις βάσεις.

Αποδοτικότητα χρόνου. Οι χωρικές αναζητήσεις πρέπει να είναι γρήγορες. Ένας σημαντικός στόχος είναι να επιτυγχάνονται τα χαρακτηριστικά απόδοσης των μονοδιάστατων Β-δένδρων. Αρχικά οι μέθοδοι προσπέλασης πρέπει να μπορούν να εγγυηθούν στη χειρότερη περίπτωση μια λογαριθμική απόδοση αναζήτησης για όλες τις πιθανές διανομές δεδομένων εισόδου, ανεξάρτητα από την σειρά εισαγωγής του και έπειτα, αυτή η απόδοση της χειρότερης περίπτωσης πρέπει να ισχύει για οποιοδήποτε συνδυασμό των χαρακτηριστικών γνωρισμάτων των δεδομένων.

Αποδοτικότητα χώρου. Ένας δείκτης πρέπει να είναι μικρός σε μέγεθος έναντι των δεδομένων που διαχειρίζεται και να μπορεί να εγγυάται μια σταθερή και εκμεταλλεύσιμη δομή αποθήκευσης.

Συντονισμός και αποκατάσταση. Στις σύγχρονες βάσεις δεδομένων, στις οποίες πολλοί χρήστες ενημερώνουν ταυτόχρονα, ανακτούν, και εισάγουν δεδομένα, οι μέθοδοι προσπέλασης πρέπει να παρέχουν δυνατές τεχνικές για τη διαχείριση των συναλλαγών, χωρίς σημαντικές απώλειες απόδοσης.

Ελάχιστη επίδραση. Η ένταξη μιας μεθόδου προσπέλασης σε ένα σύστημα βάσεων δεδομένων πρέπει να ασκεί ελάχιστη επίδραση στα υπάρχοντα μέρη του συστήματος.

Από τα παραπάνω προκύπτει, ότι κύριος στόχος της δεικτοδότησης, είναι να παρέχεται η δυνατότητα σε πολλούς χρήστες να υποβάλλουν ταυτόχρονα διαφορετικά ερωτήματα σχετικά με τη θέση, την τροχιά ή την τοπολογία ή να διερευνούν τα πρότυπα της κίνησης των αντικειμένων, σε καθεστώς διαρκών αλλαγών στα περιεχόμενα. Επομένως, η συνήθως δενδρική δομή δεικτών πρέπει να μπορεί να ανταποκρίνεται σε καταστάσεις όπου τα ερωτήματα εναλλάσσονται με ταυτόχρονες ενημερώσεις της τροχιάς των αντικειμένων. Η αναγκαιότητα των δεικτών θεωρείται αναμφισβήτητη, δεδομένου ότι ο ρυθμός μεταγωγής δεδομένων μεταξύ κύριας μνήμης και δίσκου δεν μπορεί να συναγωνιστεί τις ολοένα βελτιούμενες ταχύτητες των επεξεργαστών ή τις αυξανόμενες χωρητικότητες των σκληρών δίσκων. Οι αποτελεσματικές δομές δεικτοδότησης προσπαθούν να παρακάμψουν την εντεινόμενη δυσχέρεια μεταφοράς δεδομένων από και προς το δίσκο.

Η προσέγγιση της τροχιάς κατά γραμμικό τρόπο σε τρεις χωροχρονικές διαστάσεις προϋποθέτει ότι η σχετική δομή δεδομένων δεν χρειάζεται να ενημερώνεται παρά μόνον όταν η κίνηση του αντικειμένου αλλάζει. Επομένως, αναμένεται ότι έχει καταγραφεί πλήρως όλη η ιστορική εξέλιξη της τροχιάς και

είναι γνωστά εκ των προτέρων όλα τα σημεία που την αποτελούν. Ωστόσο, το δυναμικό πρόβλημα μετατρέπεται σε στατικό με μια επιπλέον διάσταση, επιβραδύνοντας το χρόνο εκτέλεσης των αλγορίθμων.

Η δεικτοδότηση της τρέχουσας ή των μελλοντικών θέσεων των κινούμενων αντικειμένων πρακτικά βασίζεται σε ήδη εφαρμοζόμενες τεχνικές για χωρικά στοιχεία, με κυρίαρχο το R-δένδρο και τις εκάστοτε παραλλαγές του. Μια τέτοια προσέγγιση κρίνεται βολική κυρίως για την απάντηση ερωτημάτων, αλλά μόνο όταν ο αριθμός των ενημερώσεων ανά αντικείμενο είναι σχετικά περιορισμένος εντός συγκεκριμένων χρονικών διαστημάτων. Επίσης, οι δομές αυτές στηρίζονται στην μέθοδο της διαγραφής και της επανεισαγωγής του αντικειμένου που εμφανίζει μεταβολή στην τροχιά του, καθιστώντας δύσκολο το χειρισμό τους σε περιπτώσεις όπου πρέπει να αποθηκευτούν πρόσθετες πληροφορίες ως προς τη μελλοντική εξέλιξη της τροχιάς.

Στον τομέα της χωροχρονικής δεικτοδότησης κυριαρχεί η παρουσία των R-δέντρων [Gut94], με τις παραλλαγές και τις προεκτάσεις τους. Στην πράξη κάτι τέτοιο είναι αναμενόμενο, δεδομένου ότι τα R-δέντρα είναι ιδιαίτερα δημοφιλή στις χωρικές βάσεις δεδομένων. Συνεπώς, η δεικτοδότηση των κινούμενων αντικειμένων βασίζεται κυρίως σε υπάρχουσες τεχνικές για χωρικά αντικείμενα, με κυρίαρχο το R-tree. Παραλλαγές, όπως το STR-tree (Spatio-temporal K-tree), το TB-tree (Trajectory-Bundle Tree), το TPR-tree (Time-Parameterized R-tree), το R^{EXP}-tree και το STAR-tree προσπαθούν να καλύψουν την ανεπάρκεια του R-tree στη δεικτοδότηση κινούμενων αντικειμένων.

Συνοψίζοντας πρέπει να μπορούν να υπάρχουν σε ένα σύστημα τρία επιθυμητά είδη δεικτοδότησης των τροχιών των αντικειμένων: (α) για το παρελθόν, δηλαδή για τις ιστορικές τροχιές τους, (β) για το παρόν, δηλαδή την τρέχουσα θέση τους (για την εξυπηρέτηση των τρεχόντων ερωτημάτων της βάσης) και (γ) για το μέλλον, δηλαδή την πρόβλεψη της τροχιάς τους.

3.8 Ζητήματα προς επίλυση

Το μοντέλο των κινούμενων αντικειμένων και οι σχετικές δομές δεδομένων εξακολουθούν να αποτελούν γόνιμο πεδίο έρευνας, καθώς η ένταξη τέτοιων οντοτήτων σε συμβατικά ΣΔΒΔ απαιτείται όλο και περισσότερο τα τελευταία χρόνια, ενώ παράλληλα ενισχύεται και από την εξέλιξη της τεχνολογίας στο χώρο των συστημάτων γεωγραφικού εντοπισμού, των ασύρματων τηλεπικοινωνιών και των δικτύων. Επομένως προκύπτουν τα ακόλουθα ζητήματα που χρήζουν επίλυσης:

Αποτελεσματικές δομές δεικτοδότησης. Η εξεύρεση πιο αποτελεσματικών δομών δεικτοδότησης για αποτελεσματικότερη επεξεργασία διαφόρων τύπων ερωτημάτων αποτελεί σημαντικό ζήτημα, επειδή η μελέτη της αβεβαιότητας που υπεισέρχεται στις μετρήσεις και την αναπαράσταση της τροχιάς καταδεικνύει αδυναμία ολοκληρωτικής γνώσης της κίνησης.

Απόκριση σε πραγματικό χρόνο. Πρακτικά, τα ερωτήματα που είναι δυνατόν να τεθούν σε ένα πραγματικό σύστημα είναι κατά κανόνα περισσότερο πολύπλοκα από τα ερωτήματα παραδύρου ή εγγύτερου γείτονα. Για παράδειγμα

μπορεί να ζητηθεί η πυκνότητα των σημείων σε κάποια αυθαίρετη χρονική στιγμή ή να βρεθούν τα αντικείμενα που γειτονεύουν με κάποιο συγκεκριμένο σημείο εφόσον ο αριθμός τους υπερβαίνει ένα καθορισμένο κατώφλι. Το πρώτο ερώτημα έχει προφανή χρησιμότητα σε εφαρμογές ελέγχου κυκλοφορίας, ενώ το δεύτερο είναι κρίσιμο για εμπορικά συστήματα που παρέχουν υπηρεσίες γεωγραφικού εντοπισμού (location-based services). Κατά συνέπεια, η επεξεργασία ερωτημάτων καθίσταται ακόμη περισσότερο απαιτητική, προκειμένου οι απαντήσεις να διοχετεύονται ακαριαία στους χρήστες, εντείνοντας την ανάγκη για δομές δεδομένων που να επιτυγχάνουν απόκριση σε πραγματικό χρόνο, οπότε ίσως θα πρέπει να εξεταστεί η αξιοποίηση της κύριας μνήμης.

Αποκέντρωση επεξεργασίας. Η διαβίβαση των αποτελεσμάτων γίνεται από τον κεντρικό υπολογιστή προς τη συσκευή που έθεσε το ερώτημα. Όταν η κλίμακα του συστήματος αυξάνεται κατά πολύ, τότε ο υπολογισμός της τροχιάς ίσως πρέπει να κατανεμηθεί μεταξύ πολλών επεξεργαστών. Η επεκτασιμότητα του συστήματος δυσχεραίνεται εξαιτίας φυσικών περιορισμών, κυρίως σε ότι αφορά την επικοινωνία μεταξύ μεγάλου αριθμού κόμβων του δικτύου, την επεξεργαστική ισχύ των μηχανημάτων, καθώς και την ενεργειακή αυτονομία των συσκευών. Συνεπώς, αντί να συσσωρεύονται όλες οι πληροφορίες σε κάποιο κεντρικό υπολογιστή, είναι προτιμότερο ο καταμερισμός του φόρτου μεταξύ κάποιων κόμβων, καθέναν από τους οποίους να αναλαμβάνει την παρακολούθηση μιας ομάδας αντικειμένων, όπως και την επικοινωνία με τους άλλους κόμβους.

3.9 Διαχείριση ρευμάτων κινούμενων αντικειμένων

Όσον αφορά τα χωρικά κινούμενα αντικείμενα, το μοντέλο του ρεύματος δεδομένων αποδίδει αρκετά ικανοποιητικά την ουσία του φαινομένου. Όπως έχει ήδη αναφερθεί, η αποθήκευση των δεδομένων σε στατικές σχέσεις συμφέρει υπό την προϋπόθεση ότι δεν συμβαίνουν συχνές αλλαγές ή προσθήκες στα δεδομένα. Οι ίδιοι οι χρήστες δεν επεμβαίνουν στην εισαγωγή των στοιχείων, η μετάδοση των οποίων πιθανόν εμφανίζει περιοδικές διακοπές ή έντονους ρυθμούς. Έτσι, ο μεγάλος όγκος δεδομένων που είναι πιθανόν να συνεχίσει να καταφθάνει αποτυπώνοντας τη διαρκή κίνηση των αντικειμένων, αποτελεί μεγάλο πρόβλημα ως προς την αποθήκευσή του σε συμβατικούς πίνακες μιας βάσης δεδομένων. Η χρήση της μνήμης για τη διατήρηση των πρόσφατων τμημάτων της τροχιάς για κάθε αντικείμενο και η συντήρηση περιλήψεων για τη συνολική πορεία, προφανώς ανταποκρίνεται καλύτερα τόσο στην εξοικονόμηση πόρων του συστήματος όσο και στις πιθανές απαιτήσεις των χρηστών. Βεβαίως, στα συστήματα κινούμενων αντικειμένων, οι πληροφορίες μεταβάλλονται διαρκώς (κι όχι μόνο η θέση των αντικειμένων στο χώρο), κατά κύριο λόγο με προσθήκες νέων στοιχείων, οπότε δεν είναι καθόλου αναγκαίο ούτε και πρακτικό να υποβάλλονται πολλαπλές φορές τα ίδια ερωτήματα για την διερεύνηση των αλλαγών στα περιεχόμενα των στοιχείων. Συμφέρει περισσότερο να υφίστανται μονίμως κάποια ερωτήματα διαρκείας τα οποία ενεργοποιούνται ακαριαία από τις συντελούμενες αλλαγές, ενώ οι τυχόν μεταβολές στο αποτέλεσμα να προωθούνται προς τους χρήστες σε πραγματικό χρόνο.

Ήδη έχει γίνει αναφορά στους τύπους των ερωτημάτων που εμπλέκουν κινούμενα αντικείμενα. Η βασική διαφοροποίηση στην προκειμένη περίπτωση έγκειται στην υποστήριξη της επί μακρών εκτέλεσης ερωτημάτων διαρκείας. Φυσικά, βασικό ρόλο στη σημασιολογία τους παίζει η χρονική στιγμή της υποβολής, ο προσδιορισμός της διάρκειας εκτέλεσης, όπως και η τυχόν αναφορά τους ως προς συγκεκριμένα χρονόσημα. Για τη διευκόλυνση της επεξεργασίας, η κύρια μνήμη ως χώρος τήρησης της πληροφορίας είναι πολύ ταχύτερη και ευέλικτη από το σκληρό δίσκο, ειδικά με την τάση που υπάρχει τα τελευταία χρόνια η χωρητικότητά της διαρκώς να αυξάνει και το κόστος της να ελαττώνεται. Επομένως, η επεξεργασία των ενημερώσεων των στοιχείων για τα κινούμενα αντικείμενα, όπως και των αντίστοιχων ερωτημάτων, είναι δυνατόν να γίνεται στη μνήμη, αξιοποιώντας την όσο το δυνατόν περισσότερο. Από τα παραπάνω γίνεται κατανοητό με τα κινούμενα αντικείμενα δεν υπάρχει η δυνατότητα υλοποίησης ευρετηρίων στα δεδομένα, επειδή πρέπει να διατηρούνται στην μνήμη του συστήματος και ύστερα από επεξεργασία να προκύπτει η απάντηση σε ερωτήματα διαρκείας.

Για την αντιμετώπιση του πλεονασμού δεδομένων ή του υπέρμετρου όγκου των στοιχείων που αναγκαστικά συσσωρεύονται, αντίστοιχα προς τις περιλήψεις δεδομένων που υιοθετούνται για τα ρεύματα δεδομένων, θα ήταν χρήσιμο να δημιουργηθούν κάποια σκίσα τροχιάς. Σ' αυτές τις δομές, θα μπορούσε να καταγραφεί η γενική πορεία των αντικειμένων, ώστε να επιταχυνθεί η εκτέλεση των ερωτημάτων και να υπάρχει δυνατότητα απάντησης σε ερωτήματα που τίθενται αφού τα στοιχεία έχουν παρέλθει από το σύστημα. Συνεπώς, το κύριο ενδιαφέρον των περισσότερων εφαρμογών παρακολούθησης κινούμενων αντικειμένων εστιάζεται στην τρέχουσα εξέλιξη της πορείας και κατά συνέπεια, αποδίδεται ιδιαίτερη βαρύτητα στις πλέον πρόσφατες θέσεις που έχουν καταγραφεί, ενώ τα παλαιότερα στοιχεία χαρακτηρίζονται παρωχημένα. Είναι λοιπόν απολύτως βάσιμος ο ισχυρισμός ότι το φαινόμενο αντιστοιχεί σε ρεύματα τροχιάς αντικειμένων (trajectory streams).

Κεφάλαιο 4

Διαδικτυακοί χάρτες

4.1 Εισαγωγή

Η διαδικτυακή χαρτογράφηση (*web mapping*) είναι η διαδικασία του σχεδιασμού, της υλοποίησης, της δημιουργίας και παράδοσης χαρτών στο διαδίκτυο. Ενώ η διαδικτυακή χαρτογράφηση κυρίως ασχολείται με τεχνολογικά θέματα, η διαδικτυακή χαρτογραφία (*web cartography*) μελετά επιπλέον θεωρητικές πτυχές, όπως: τη χρήση των διαδικτυακών χαρτών, την αξιολόγηση και τη βελτιστοποίηση των τεχνικών και των ροών εργασίας, την ικανότητα αξιοποίησης των διαδικτυακών χαρτών, τις κοινωνικές πτυχές και πολλά άλλα.

Τα διαδικτυακά συστήματα γεωγραφικών πληροφοριών (*web GIS*) είναι παρόμοια με τη διαδικτυακή χαρτογράφηση αλλά με έμφαση στην ανάλυση, στην επεξεργασία συγκεκριμένων γεωγραφικών δεδομένων του έργου και ερευνητικών πτυχών. Συχνά οι όροι *web GIS* και *web mapping* χρησιμοποιούνται ως συνώνυμα, ακόμη και αν δεν σημαίνουν ακριβώς το ίδιο. Στην πραγματικότητα, τα σύνορα μεταξύ διαδικτυακών χαρτών (*web maps*) και *web GIS* είναι θολά. Οι διαδικτυακοί χάρτες αποτελούν συχνά μέσα παρουσίασης στο *web GIS* και κερδίζουν όλο και περισσότερο αναλυτικές ικανότητες.

Ειδική περίπτωση των διαδικτυακών χαρτών είναι οι χάρτες για κινητά τηλέφωνα, οι οποίοι εμφανίζονται σε κινητές υπολογιστικές συσκευές, όπως σε κινητά τηλέφωνα, σε έξυπνα τηλέφωνα, σε ψηφιακούς προσωπικούς βοηθούς (*PDA*s), σε γεωγραφικά συστήματα εντοπισμού θέσης (*GPS*) και άλλες συσκευές. Αν οι χάρτες σε αυτές τις συσκευές εμφανίζονται με ένα πρόγραμμα περιήγησης διαδικτύου για κινητά (*mobile web browser*) ή με ένα διαδικτυακό πράκτορα (*web user agent*), μπορούν να θεωρηθούν ως κινητοί διαδικτυακοί χάρτες (*mobile web maps*). Αν στους *mobile web maps* εμφανίζονται επίσης ευαίσθητες πληροφορίες που αφορούν το περιβάλλον και την τοποθεσία, όπως είναι τα σημεία ενδιαφέροντος (*Points Of Interest - POI*), τότε ο όρος υπηρεσίες εντοπισμού (*location-based services*) χρησιμοποιείται συχνά.

Η χρήση του διαδικτύου ως μέσο διάδοσης των χαρτών, μπορεί να θεωρηθεί ως ένα σημαντικό βήμα προόδου στη χαρτογραφία και ανοίγει πολλές νέες δυνατότητες, όπως χάρτες σε πραγματικό χρόνο, φθηνότερη διάδοση, συχνότερες και φθηνότερες ενημερώσεις των δεδομένων και του λογισμικού, εξατομικευμένο περιεχόμενο στους χάρτες, καταναμημένες πηγές δεδομένων και διαμοιρασμό των γεωγραφικών πληροφοριών. Εμπλέκει επίσης πολλές προκλήσεις που οφείλονται σε τεχνικούς περιορισμούς (χαμηλή ανάλυση απεικόνισης και περιορισμένη ταχύτητα εύρους ζώνης, ιδίως σε κινητές υπολογιστικές συσκευές, εκ των οποίων πολλές είναι μικρού φυσικού μεγέθους και χρησιμοποιούν αργές ασύρματες συνδέσεις διαδικτύου), σε θέματα πνευματικών δικαιωμάτων και ασφάλειας, σε θέματα αξιοπιστίας και σε θέματα τεχνικής πολυπλοκότητας. Ενώ οι πρώτοι διαδικτυακοί χάρτες ήταν κυρίως

στατικοί, λόγω τεχνικών περιορισμών, οι σημερινοί διαδικτυακοί χάρτες μπορούν να είναι πλήρως διαδραστικοί και να ενσωματώνουν πολλαπλά μέσα. Αυτό σημαίνει ότι και τα δύο, web mapping και web cartography, πρέπει επίσης να ασχοληθούν με την διαδραστικότητα, τη χρηστικότητα και τα ζητήματα των πολυμέσων.

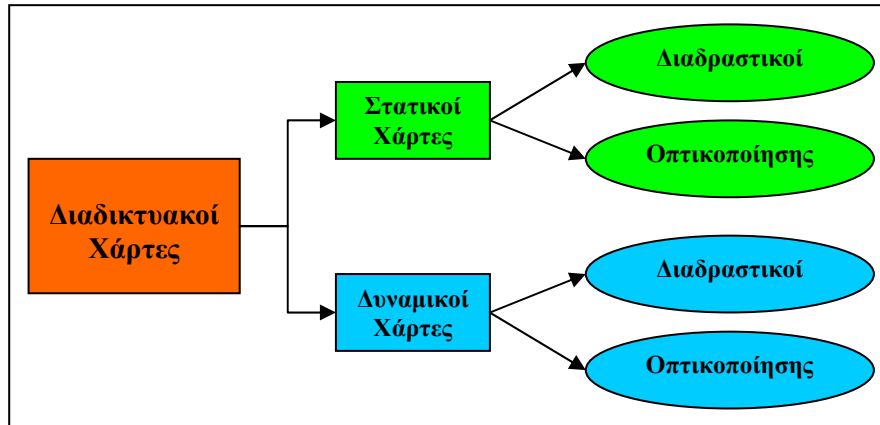
Η έλευση της διαδικτυακής χαρτογράφησης μπορεί να θεωρηθεί ως μια σημαντική νέα τάση στη χαρτογραφία. Προηγουμένως, η χαρτογραφία ήταν περιορισμένη σε λίγες εταιρείες, ινστιτούτα και οργανισμούς χαρτογράφησης, απαιτώντας δαπανηρό και περίπλοκο hardware υλικό και λογισμικό, καθώς και εξειδικευμένους χαρτογράφους και μηχανικούς γεωπληροφορικής. Με τη διαδικτυακή χαρτογράφηση, ελεύθερα διαθέσιμες τεχνολογίες χαρτογράφησης και γεωγραφικών δεδομένων επέτρεψαν σε κάθε ειδικευμένο πρόσωπο την παραγωγή διαδικτυακών χαρτών, με την ακρίβεια σε κόστος γεωγραφικά δεδομένα και την τεχνική πολυπλοκότητα (εναρμόνιση των δεδομένων, ελλιπή πρότυπα) να είναι δύο από τα εναπομείναντα εμπόδια που εμποδίζουν τη διαδικτυακή χαρτογράφηση από την πλήρη εξέλιξή της.

Η φτηνή και εύκολη μεταφορά των γεωγραφικών δεδομένων σε ολόκληρο το διαδίκτυο επιτρέπει την ενοποίηση των κατανεμημένων πηγών δεδομένων, ανοίγοντας ευκαιρίες που υπερβαίνουν τις δυνατότητες των μη ενοποιημένων μέσων αποθήκευσης δεδομένων. Οποιοσδήποτε με ελάχιστη τεχνογνωσία και υποδομή μπορεί να εξελιχθεί σε φορέα παροχής γεωγραφικών δεδομένων. Αυτά τα στοιχεία μπορούν να θεωρηθούν και ως ένα πλεονέκτημα και ως ένα μειονέκτημα, γιατί ενώ επιτρέπει σε όλους να παράγουν χάρτες, θέτει επίσης γεωγραφικά δεδομένα στα χέρια ανεκπαιδευτων ανθρώπων που ενδεχομένως παραβιάζουν τις χαρτογραφικές και γεωγραφικές αρχές και εισάγουν ατέλειες κατά την προετοιμασία, ανάλυση και παρουσίαση των γεωγραφικών και χαρτογραφικών στοιχείων. Η μόρφωση του κοινού σχετικά με τη γεωγραφική ανάλυση και τις χαρτογραφικές μεθόδους και αρχές, πρέπει συνεπώς να αποτελεί προτεραιότητα.

4.2 Τύποι διαδικτυακών χαρτών

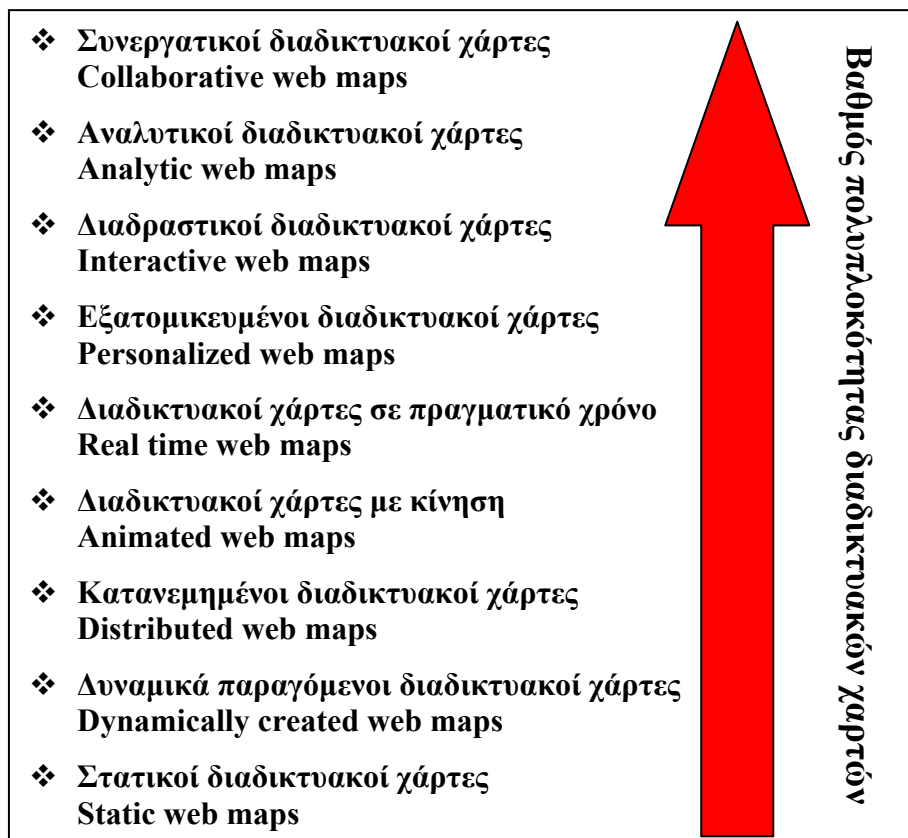
Μια πρώτη ταξινόμηση των διαδικτυακών χαρτών που μπορεί να γίνει είναι αυτή μεταξύ των στατικών και δυναμικών χαρτών. Καθεμιά από αυτές τις κατηγορίες υποδιαιρείται περαιτέρω σε διαδραστικούς (*interactive*) και για οπτικοποίηση μόνο (*view only*) διαδικτυακούς χάρτες (βλ. σχήμα 4.1). Ο πιο κοινός χάρτης που βρίσκεται στο παγκόσμιο ιστό είναι ο μη διαδραστικός στατικός χάρτης. Οι στατικοί χάρτες προσφέρουν τις παραδοσιακές εικόνες χαρτών και συχνά οι πηγές για αυτούς τους χάρτες αποτελούν τα αρχικά χαρτογραφικά προϊόντα, τα οποία σαρώνονται και τοποθετούνται στις ιστοσελίδες του παγκόσμιου ιστού.

Ωστόσο, σήμερα, υπό το πρίσμα της αύξησης του αριθμού των διαφορετικών τύπων διαδικτυακών χαρτών που υπάρχουν, η ταξινόμηση αυτή χρειάζεται αναθεώρηση. Πλέον υπάρχουν πρόσθετες δυνατότητες όσον αφορά την κατανομή των πηγών δεδομένων, χάρτες κατόπιν συνεργασίας, εξατομικευμένοι χάρτες, κλπ.



Σχήμα 4.1: Ταξινόμηση διαδικτυακών χαρτών σε στατικούς και δυναμικούς.

Σύμφωνα με τον χαρτογράφο Jan-Menno Kraak [wwwWMG], ο οποίος ανέπτυξε την πρώτη ταξινόμηση των διαδικτυακών χαρτών το 2001, οι διαδικτυακοί χάρτες μπορούν να διαχωριστούν με βάση το επίπεδο πολυπλοκότητάς τους όπως φαίνεται στο σχήμα 4.2. Όπως με όλα τα συστήματα ταξινόμησης, ούτε αυτό το σύστημα είναι τέλειο και είναι αρκετά δύσκολο να διαχωριστεί ποιος διαδικτυακός χάρτης είναι περισσότερο πολύπλοκος από του υπόλοιπους, ενώ παράλληλα υπάρχει και το πρόβλημα ότι δεν μπορούν να είναι τακτοποιημένοι όλοι οι χάρτες σε μια και μόνο ενιαία κατηγορία. Εντούτοις, αυτή η ταξινόμηση παρέχει μια καλή αφετηρία για την περιγραφή και ανάλυση των διάφορων τύπων χαρτών που μπορούν να βρεθούν στον παγκόσμιο ιστό σήμερα.



Σχήμα 4.2: Ταξινόμηση διαδικτυακών χαρτών σύμφωνα με τον βαθμό πολυπλοκότητας.

4.2.1 Στατικοί διαδικτυακοί χάρτες

Οι στατικοί χάρτες, όπως υποδηλώνει το όνομά τους, είναι ουσιαστικά τα ψηφιακά αντίγραφα των γνωστών εντύπων χαρτών. Ενώ ο χρήστης έχει την δυνατότητα να μεγεθύνει μέσα ή έξω στο χάρτη, το περιεχόμενο του χάρτη είναι στατικό. Η κλίμακα του χάρτη δεν μπορεί να αλλάξει, οι περιοχές στις άκρες του χάρτη δεν μπορούν να απεικονισθούν και τα περιεχόμενα του χάρτη δεν έχουν την δυνατότητα να μεταβάλλονται από ενεργά σε ανενεργά και αντίστροφα. Αυτός ο τύπος διαδικτυακού χάρτη, είναι συχνά προϊόν είτε ψηφιακής σάρωσης ενός χάρτη έντυπης μορφής, οι οποίοι δεν είχαν σχεδιαστεί ως χάρτες για απεικόνιση σε οθόνη, είτε προϊόν εξαγωγής ενός προγράμματος GIS. Οι έντυποι χάρτες έχουν πολύ υψηλότερη ανάλυση και πυκνότητα πληροφοριών από τις χαρακτηριστικές οθόνες των υπολογιστών του ίδιου φυσικού μεγέθους, οπότε θα μπορούσαν να είναι μη αναγνώσιμοι όταν εμφανίζονται στις οθόνες με λάθος ανάλυση.



Σχήμα 4.3: Στατικός χάρτης USGS DRG. (Πηγή: [wwwWWM])

Αυτού του είδους οι χάρτες (βλ. σχήμα 4.3) είναι μόνο για οπτικοποίηση χωρίς κίνηση (animation) και διαδραστικότητα. Δημιουργούνται μόνο μία φορά, τις περισσότερες φορές χειρωνακτικά και ενημερώνονται σπάνια. Τυπικές μορφές γραφικών για στατικούς διαδικτυακούς χάρτες είναι PNG, JPEG, GIF, ή TIFF (π.χ. DRG) για ψηφιδωτά αρχεία (raster), SVG, PDF ή SWF για διανυσματικά αρχεία (vector).

4.2.2 Δυναμικά παραγόμενοι διαδικτυακοί χάρτες

Σε αντίθεση με τους στατικούς διαδικτυακούς χάρτες, αυτού του είδους οι χάρτες δημιουργούνται κατ' απαίτηση και παράγονται κάθε φορά δυναμικά (on the fly) όποτε ο χρήστης φορτώνει την σχετική ιστοσελίδα, συχνά από δυναμικές

βάσεις δεδομένων. Ο διαδικτυακός εξυπηρετητής (web server) δημιουργεί τον χάρτη χρησιμοποιώντας ένα διαδικτυακό εξυπηρετητή χαρτών ή ένα αυτόνομο λογισμικό. Η επιλογή της δημοσίευσης ενός στατικού ή ενός δυναμικού διαδικτυακού χάρτη, πρέπει να εξαρτάται από το πόσο συχνά τα δεδομένα που απεικονίζονται διαμέσου αυτού του χάρτη αλλάζουν.

Για παράδειγμα, κάποιος χρήστης μπορεί να θέλει να απεικονίσει: (α) τον πληθυσμό των πόλεων της χώρας του, που συντάσσονται μετά από μία απογραφή και (β) τις θέσεις ορισμένων οχημάτων επάνω στο κυκλοφοριακό δίκτυο της πόλης του. Το σενάριο Α απαιτεί έναν στατικό χάρτη, δεδομένου ότι τα στοιχεία πληθυσμών και τα γεωγραφικά διαμερίσματα στα οποία συλλέχθηκαν τα δεδομένα, αντιπροσωπεύουν ένα έγκυρο στιγμιότυπο. Αφότου η απογραφή είναι πλήρης, τα στοιχεία δεν αλλάζουν. Αντίθετα στο σενάριο Β η παραγωγή ενός ακριβή στατικού χάρτη των θέσεων των οχημάτων είναι αδύνατον, δεδομένου ότι οι θέσεις τους αλλάζουν συνεχώς, οπότε προφανώς, αυτό το σενάριο απαιτεί τη χρήση ενός δυναμικά παραγόμενου χάρτη.

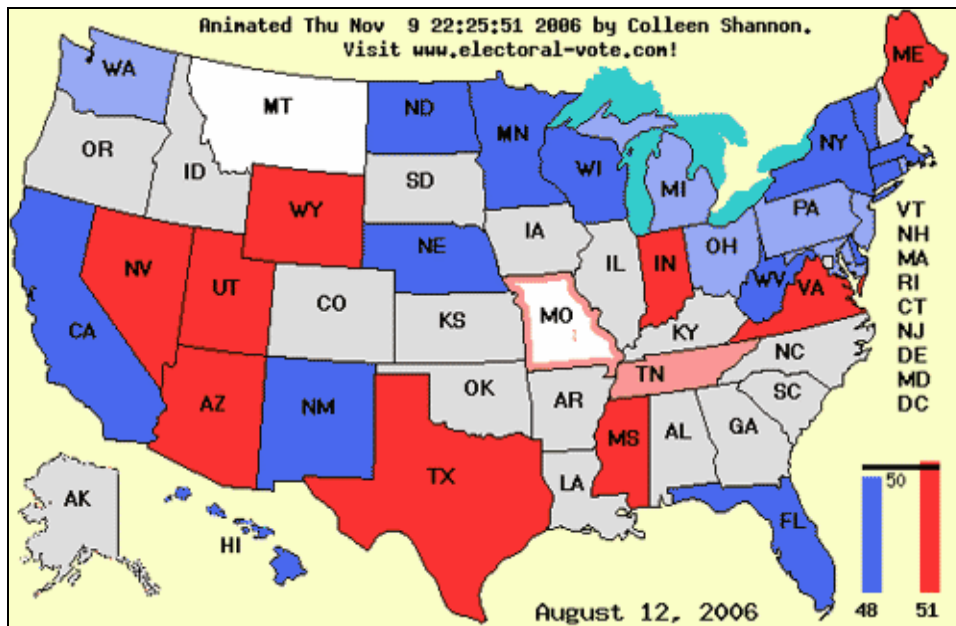
4.2.3 Κατανεμημένοι διαδικτυακοί χάρτες

Αυτή η κατηγορία των διαδικτυακών χαρτών, αφορά εκείνους που συντάσσονται από κατανεμημένες πηγές δεδομένων και αποτελεί στην ουσία μια υποκατηγορία των δυναμικά παραγόμενων χαρτών. Το ιδιαίτερο χαρακτηριστικό γνώρισμα αυτής της κατηγορίας είναι ότι μερικά από τα δεδομένα που χρησιμοποιούνται για την παραγωγή του χάρτη προέρχονται και από άλλους κεντρικούς υπολογιστές. Αυτό μπορεί να περιλαμβάνει τη χρήση των επίσημων πρωτοκόλλων διανομής δεδομένων, όπως η υπηρεσία διαδικτυακών χαρτών (WMS – Web Map Service) της ανοικτής γεωχωρικής κοινοπραξίας (OGC – Open Geospatial Consortium), η υπηρεσία διαδικτυακών χαρακτηριστικών γνωρισμάτων (WFS – Web Feature Service) ή τα πρότυπα υπηρεσιών διαδικτυακών ψηφιδωτών αρχείων (WCS – Web Coverage Service).

Το WMS πρωτόκολλο προσφέρει μία τυποποιημένη μέθοδο για την πρόσβαση σε χάρτες άλλων εξυπηρετητών (servers). Οι WMS εξυπηρετητές μπορούν να συλλέξουν αυτές τις διαφορετικές πηγές, να ανασχεδιάσουν τα επίπεδα του χάρτη, εφόσον είναι αναγκαίο, και να τα στείλουν πίσω ως μία σύνθετη εικόνα που περιέχει όλα τα απαιτούμενα επίπεδα του χάρτη. Ένας εξυπηρετητής μπορεί να προσφέρει έναν τοπογραφικό χάρτη ως βάση, ενώ οι άλλοι εξυπηρετητές μπορούν να προσφέρουν τα θεματικά επίπεδα του χάρτη.

4.2.4 Διαδικτυακοί χάρτες με κίνηση

Οι κινούμενοι χάρτες (animated maps) δείχνουν αλλαγές στο χάρτη με την πάροδο του χρόνου από την κίνηση των γραφικών ή χρονικών μεταβλητών, δηλαδή συνδέονται με την απεικόνιση των στιγμιοτύπων της ίδιας μεταβλητής του χάρτη, η οποία λαμβάνεται σε διαφορετικούς χρόνους. Διάφορες μορφές δεδομένων, πολυμέσων και τεχνολογιών επιτρέπουν την απεικόνιση των κινούμενων χαρτών στο διαδίκτυο, όπως: SVG, Adobe Flash, Java, Quicktime, κλπ., με διαφορετικούς βαθμούς αλληλεπίδρασης.



Σχήμα 4.4: Animated χάρτης καταγραφής των ψηφοφοριών της Γερουσίας των ΗΠΑ. (Πηγή: [wwwWMG])

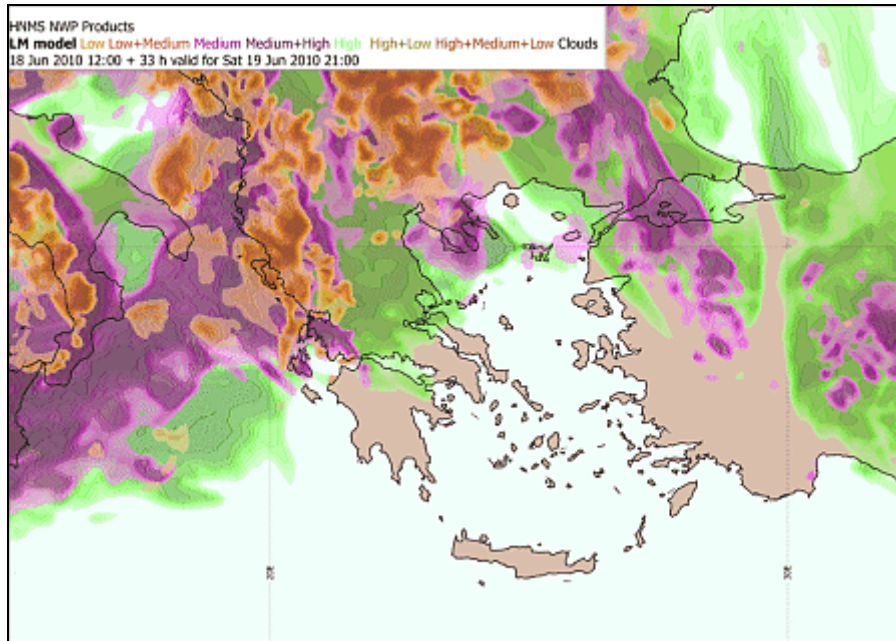
Παραδείγματα διαδικτυακών χαρτών με κίνηση είναι οι χάρτες του καιρού, οι χάρτες που εμφανίζουν δυναμικά τα φυσικά ή άλλα φαινόμενα (όπως τα θαλάσσια ρεύματα, τα πρότυπα κίνησης των ανέμων, κ.λπ.).

Ένα καλό παράδειγμα αποτελεί ο χάρτης του σχήματος 4.4, στον οποίο παρουσιάζονται τα αποτελέσματα των ψηφοφοριών εκστρατείας για την Αμερικανική Γερουσία το 2006, που διεξάγονται λίγες εβδομάδες πριν τις εκλογές.

4.2.5 Διαδικτυακοί χάρτες σε πραγματικό χρόνο

Οι χάρτες σε πραγματικό χρόνο δείχνουν την κατάσταση ενός φαινομένου σε σχεδόν πραγματικό χρόνο (μόνο με λίγα δευτερόλεπτα ή λεπτά καθυστέρηση). Η συλλογή των δεδομένων που απαιτείται για έναν τέτοιο χάρτη είναι συνήθως αυτοματοποιημένη, μέσω της χρήσης αισθητήρων, και οι χάρτες δημιουργούνται ή ενημερώνονται σε τακτά χρονικά διαστήματα ή αμέσως μετά από τη ζήτηση του χρήστη.

Εκτός από την κατηγορία των διαδικτυακών χαρτών που περιλαμβάνουν κίνηση, οι χάρτες του καιρού (βλ. σχήμα 4.5), είναι επίσης εκείνοι που προβάλλονται περισσότερο στην κατηγορία διαδικτυακών χαρτών πραγματικού χρόνου. Ορισμένα άλλα παραδείγματα τέτοιων χαρτών, είναι οι κυκλοφοριακοί χάρτες ή οι χάρτες που παράγονται από τα συστήματα παρακολούθησης οχημάτων, μέσω της χρήσης των αισθητήρων όπως εκείνων που εγκαθίστανται κατά μήκος των αστικών οδικών δικτύων για να ανιχνεύουν την οδική κυκλοφορία ή των φορητών συσκευών GPS.



Σχήμα 4.5: Προγνωστικός χάρτης συνολικής νεφοκάλυψης TCC της Ελλάδος, από Ε.Μ.Υ. '2010-06-19 21:00:00'. (Πηγή: [www.EMY.gov.gr])

4.2.6 Εξατομικευμένοι διαδικτυακοί χάρτες

Αυτή η κατηγορία διαδικτυακών χαρτών, αναφέρεται σε ιστοσελίδες που επιτρέπουν στο χρήστη να μπορεί να έχει την δυνατότητα εισαγωγής πληροφορίας όσον αφορά την απεικόνιση του χάρτη (εκτός του μεγέθους απεικόνισης). Οι εξατομικευμένοι διαδικτυακοί χάρτες επιτρέπουν στο χρήστη επιλεκτικά να εφαρμόσει το δικό του φίλτράρισμα στα στοιχεία, να προβάλλει το δικό του περιεχόμενο και να εφαρμόσει το δικό του προσωπικό στυλ διαμόρφωσης και συμβολισμών επάνω στον χάρτη. Ουσιαστικά επιτρέπεται στον χρήστη η εισαγωγή πληροφορίας στον χάρτη η οποία, μπορεί να περιλαμβάνει τα χρώματα της απεικόνισης, τη μέθοδο ταξινόμησης, τον τίτλο, το υπόμνημα, την γραφική κλίμακα ή οποιοδήποτε άλλο στοιχείο του χάρτη πάνω στο οποίο μπορεί κάποιος να έχει τον έλεγχο κατά την διαδικασία κατασκευής ενός χάρτη μέσω ενός προγράμματος GIS.

Το πρότυπο OGC παρέχει το πρότυπο SLD (Styled Layer Description) που ενδέχεται να αποσταλεί σε ένα εξυπηρετητή WMS για την εφαρμογή των επιμέρους στυλ διαμόρφωσης. Αυτό συνεπάγεται ότι το περιεχόμενο και η δομή των δεδομένων του απομακρυσμένου εξυπηρετητή WMS πρέπει να είναι διαμορφωμένη κατάλληλα.

4.2.7 Διαδραστικοί διαδικτυακοί χάρτες

Ενώ οι πρώτοι χάρτες στον παγκόσμιο ιστό ήταν στην συντριπτική πλειοψηφία τους στατικοί, στην σημερινή εποχή οι διαδραστικοί χάρτες είναι η πιο συνηθισμένη μορφή διαδικτυακών χαρτών. Η διαδραστικότητα έρχεται με

πολλές μορφές, με πιο σύνηδη την δυνατότητα να μπορεί ο χρήστης να αλλάζει την κλίμακα απεικόνισης του χάρτη, μέσω της μεγέθυνσης ή της σμίκρυνσης του (zooming) και να μπορεί να πλοηγείται σε οποιαδήποτε περιοχή του επιθυμεί (panning).

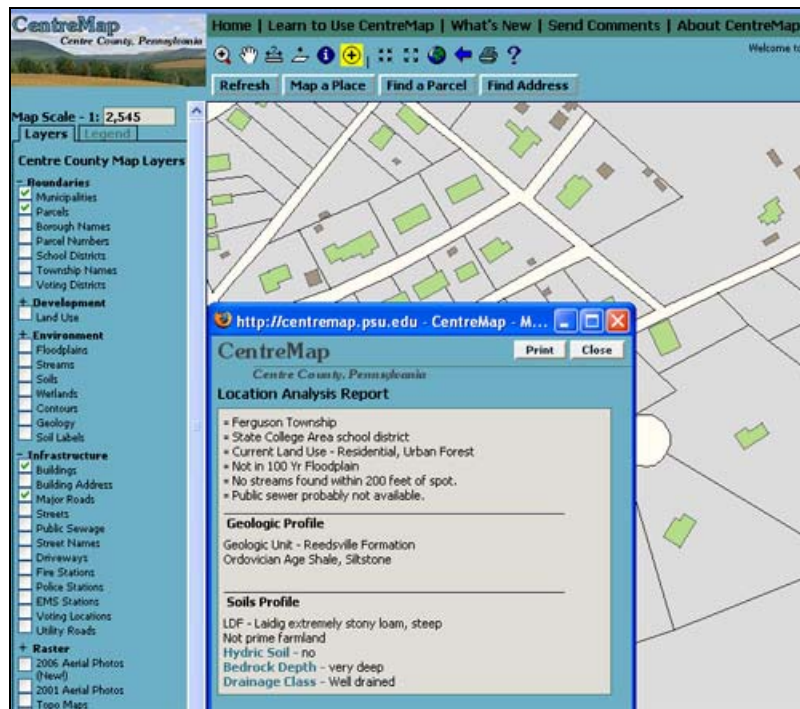
Η διαδραστικότητα είναι ένα από τα σημαντικότερα πλεονεκτήματα των απεικονισμένων χαρτών σε οθόνη και των διαδικτυακών χαρτών, η οποία βοηθά στην αντιστάθμιση των μειονεκτημάτων που προκύπτουν από την οθόνη του υπολογιστή. Ακόμη βοηθά στην διερεύνηση του χάρτη, στην αλλαγή των παραμέτρων του, στην πλοήγησή του από το χρήστη και στην διαδραστικότητα του, αποκαλύπτει περισσότερες πληροφορίες στο χρήστη, συνδέεται με άλλους πόρους, κ.τλ. Από τεχνική άποψη, αυτό επιτυγχάνεται μέσω του συνδυασμού των γεγονότων, scripting και χειρισμούς μέσω DOM. Οι πιο συνήθεις μορφές διαδραστικότητας διαδικτυακών χαρτών παρέχουν τις εξής δυνατότητες στο χρήστη:

- Ενεργοποίηση / απενεργοποίηση των επιπέδων του χάρτη.
- Λήψη αναλυτικών πληροφοριών για τα χαρακτηριστικά γνωρίσματα του χάρτη.
- Πλοήγηση στους ιστοχώρους που συνδέονται με τα χαρακτηριστικά του χάρτη μέσω υπερσυνδέσεων.
- Παραγωγή ενός πλήρους εξατομικευμένου χάρτη.
- Εξερεύνηση προτύπων και συσχετίσεων των δεδομένων, χρησιμοποιώντας εργαλεία απεικόνισης που συνδέονται δυναμικά με το χάρτη.

4.2.8 Αναλυτικοί διαδικτυακοί χάρτες

Η εποχή "WebGIS" καθορίστηκε από τον πολλαπλασιασμό των εφαρμογών που επέτρεψε στο κοινό να εκτελεί αναλύσεις οι οποίες παλαιότερα ήταν διαθέσιμες μόνο στους επαγγελματίες των GIS. Συνεπώς, οι αναλυτικοί διαδικτυακοί χάρτες προσφέρουν GIS ανάλυση, είτε με παρεχόμενα γεωγραφικά δεδομένα, είτε με γεωγραφικά δεδομένα τα οποία φορτώνονται από το χάρτη του χρήστη. Όπως αναφέρθηκε, η διαχωριστική γραμμή μεταξύ των αναλυτικών διαδικτυακών χαρτών και των web GIS είναι θολή. Συχνά, τα μέρη της ανάλυσης πραγματοποιούνται από ένα GIS εξυπηρετητή (server), ενώ ο πελάτης (client) εμφανίζει το αποτέλεσμα της ανάλυσης, όμως όσο οι διαδικτυακοί πελάτες κερδίζουν όλο και περισσότερες δυνατότητες, αυτή η κατανομή καθηκόντων μπορεί να μετατοπιστεί σταδιακά.

Η πιο κοινή αναλυτική λειτουργία που εκτελείται από τις διαδικτυακές χαρτογραφικές εφαρμογές είναι η εύρεση διαδρομών, στην οποία ο χρήστης είναι σε θέση να διευκρινίσει μια αρχική διεύθυνση και μια τελική διεύθυνση, ώστε κατόπιν ο κεντρικός υπολογιστής που διαχειρίζεται τον χάρτη να αναλύσει το οδικό δίκτυο για να καθορίσει την καλύτερη δυνατή διαδρομή. Στους χρήστες παρέχεται η επιλογή να επιλέξουν είτε τη γρηγορότερη διαδρομή (σε χρόνο) είτε την συντομότερη διαδρομή (σε απόσταση), ενώ παράλληλα είναι δυνατό να καθορίσουν ποιους δρόμους θέλουν να αποφύγουν.



Σχήμα 4.6: Παράδειγμα ενός αναλυτικού διαδικτυακού χάρτη (CentreMap).
(Πηγή: [wwwCMAP])

4.2.9 Συνεργατικοί διαδικτυακοί χάρτες

Η τελευταία κατηγορία διαδικτυακών χαρτών περιλαμβάνει εκείνους τους χάρτες οι οποίοι δημιουργούνται και διατηρούνται από ένα κατανεμημένο δίκτυο ανθρώπων. Τέτοιες ιστοσελίδες είναι παρόμοιες σε σκεπτικό με την Wikipedia, όπου διάφοροι άνθρωποι συνεργάζονται για να δημιουργήσουν και να βελτιώσουν τους χάρτες στο διαδίκτυο. Ένα από τα καλύτερα γνωστά προγράμματα διαδικτυακής χαρτογράφησης μέσω συνεργασίας είναι το OpenStreetMap, ένα πρόγραμμα του οποίου στόχος είναι να δημιουργηθεί ένας ελεύθερος διαμορφώσιμος (ανά πάσα χρονική στιγμή) χάρτης όλων των οδικών δικτύων παγκοσμίως και στο οποίο οποιοσδήποτε ενδιαφερόμενος πολίτης μπορεί να εγγραφεί στην ιστοσελίδα και να προσθέσει την δική του πληροφορία.

Τεχνικά, πρόκειται για μια εφαρμογή που επιτρέπει ταυτόχρονη επεξεργασία σε όλο το διαδίκτυο, η οποία πρέπει να εξασφαλίζει ότι τα γεωμετρικά χαρακτηριστικά που επεξεργάζεται ένα άτομο είναι κλειδωμένα, ώστε να μην είναι δυνατή η επεξεργασία από άλλα άτομα ταυτόχρονα και συγχρόνως να γίνεται ένας ελάχιστος έλεγχος της ποιότητας, πριν τα δεδομένα αυτά δημοσιοποιηθούν στον παγκόσμιο ιστό. Ορισμένα δημοφιλή από τα προγράμματα συνεργασίας για παραγωγή τέτοιων χαρτών είναι το OpenStreetMap και το WikiMapia.

4.3 Αξιολόγηση διαδικτυακών χαρτών

Εύκολη παροχή ενημερωμένων πληροφοριών. Η εύκολη παροχή ενημερωμένων πληροφοριών μέσω των διαδικτυακών χαρτών, ειδικά αν οι χάρτες δημιουργούνται αυτόματα από βάσεις δεδομένων, τότε η απεικόνιση των πληροφοριών πάνω σε αυτούς μπορεί να γίνεται σε σχεδόν πραγματικό χρόνο. Το σημαντικότερο όμως είναι ότι δεν χρειάζεται πλέον να τυπώνονται, να επεξεργάζονται και να διανέμονται, όπως γινόταν παλαιότερα. Ορισμένα σύνηδη παραδείγματα είναι: (α) ένας χάρτης στον οποίο εμφανίζονται τα εκλογικά αποτελέσματα αμέσως μόλις είναι διαθέσιμα ή (β) ένας χάρτης στον οποίο εμφανίζεται η κατάσταση της οδικής κυκλοφορίας σχεδόν σε πραγματικό χρόνο χρησιμοποιώντας δεδομένα οδικής κυκλοφορίας.

Φθηνή υποδομή. Η φθηνή υποδομή των διαδικτυακών χαρτών σε software και hardware. Το hardware του διαδικτυακού εξυπηρετητή είναι φθηνά διαθέσιμο και υπάρχουν πολλά εργαλεία ανοικτού κώδικα για την παραγωγή διαδικτυακών χαρτών.

Εύκολη διανομή. Η εύκολη διανομή των ενημερωμένων εκδόσεων των προϊόντων. Επειδή οι διαδικτυακοί χάρτες διανέμουν και λογική αλλά και δεδομένα σε κάθε αίτημα ή φόρτωσή τους, οι ενημερώσεις του προϊόντος μπορούν να γίνονται κάθε φορά που ο χρήστης φορτώνει την εφαρμογή. Στην παραδοσιακή χαρτογραφία, όταν πρόκειται για έντυπους χάρτες ή διαδραστικούς χάρτες οι οποίοι διανέμονται με μη διαδικτυακά (offline) μέσα (CD, DVD, κλπ.), η ενημέρωση ενός χάρτη υπόκειται σε σοβαρές προσπάθειες, προκαλώντας την επανεκτύπωση και αναδιανομή των μέσων. Με τους διαδικτυακούς χάρτες, η ενημέρωση των δεδομένων και των προϊόντων είναι ευκολότερη, φθηνότερη, γρηγορότερη και ασφαλώς συχνότερη.

Υποστήριξη. Η δυνατότητα λειτουργίας σε προγράμματα περιήγησης του διαδικτύου (browsers) και σε διάφορα λειτουργικά συστήματα. Εάν οι διαδικτυακοί χάρτες έχουν υλοποιηθεί με βάση εργαλεία ανοικτού κώδικα, το υποκείμενο λειτουργικό σύστημα και το πρόγραμμα περιήγησης του διαδικτύου δεν αποτελούν πρόβλημα.

Συνδυασμός κατανεμημένων πηγών δεδομένων. Χρησιμοποιώντας εργαλεία ανοικτού κώδικα και τεκμηριωμένους APIs μπορεί εύκολα να γίνει ενσωμάτωση διαφορετικών πηγών δεδομένων, εάν το σύστημα προβολής, η κλίμακα του χάρτη και η ποιότητα των δεδομένων ταιριάζουν. Η χρήση των κεντρικών πηγών δεδομένων καταργεί την επιβάρυνση για τους μεμονωμένους οργανισμούς να διατηρούν αντίγραφα των ίδιων δεδομένων. Η αρνητική πλευρά είναι ότι πρέπει ο οποιοσδήποτε να βασίζεται και να εμπιστεύεται τις εξωτερικές πηγές δεδομένων.

Δυνατότητα εξατομίκευσης. Με τη χρήση προφίλ χρηστών, προσωπικών φίλτρων και προσωπικών στυλ διαμόρφωσης και προσωπικής συμβολοποίησης, οι χρήστες μπορούν να ρυθμίσουν και να σχεδιάσουν τους δικούς τους χάρτες, εάν τα διαδικτυακά συστήματα χαρτογράφησης υποστηρίζουν την εξατομίκευση.

Δυνατότητα χαρτογράφησης σε συνεργασία. Παρόμοια φιλοσοφία με το έργο της Wikipedia, οι τεχνολογίες διαδικτυακής χαρτογράφησης, όπως DHTML/Ajax, SVG, Java, Adobe Flash, κ.λπ. επιτρέπουν την απόκτηση διανεμημένων δεδομένων και προσπαθειών συνεργασίας μεταξύ των ανθρώπων.

Παραδείγματα τέτοιου είδους έργων είναι το OpenStreetMap ή η κοινότητα του Google Earth. Όπως και με άλλα ανοικτά προγράμματα (open projects), η διασφάλιση της ποιότητας είναι πολύ σημαντική και πρωταρχικής σημασίας.

Υποστήριξη υπερσυνδέσεων των διαδικτυακών χαρτών με άλλες πληροφορίες στο διαδίκτυο. Ακριβώς όπως κάθε άλλη ιστοσελίδα ή wiki, οι διαδικτυακοί χάρτες μπορούν να δράσουν ως ένα ευρετήριο για άλλες πληροφορίες στον παγκόσμιο ιστό. Οποιαδήποτε ευαίσθητη περιοχή σε ένα χάρτη, ένα κείμενο ετικέτα, κλπ. μπορούν να παρέχουν υπερσυνδέσεις σε πρόσθετες πληροφορίες. Για παράδειγμα, ένας χάρτης ο οποίος δείχνει τις επιλογές των μέσων μαζικής μεταφοράς μπορεί άμεσα να συνδέεται με το αντίστοιχο τμήμα που δείχνει διαδικτυακά τα δρομολόγια των τρένων.

Εύκολη ενσωμάτωση πολυμέσων. Τα προγράμματα περιήγησης του διαδικτύου πλέον υποστηρίζουν την αναπαραγωγή βίντεο, ήχου και κινούμενης εικόνας (SVG, SWF, Quicktime, κ.τλ).

Αξιοπιστία. Η αξιοπιστία του διαδικτύου και της υποδομής των διαδικτυακών εξυπηρετητών δεν είναι ακόμη αρκετά καλή. Ειδικά εάν ένας διαδικτυακός χάρτης βασίζεται στις εξωτερικές - κατανεμημένες πηγές δεδομένων, ο αρχικός δημιουργός του χάρτη συχνά δεν μπορεί να εγγυηθεί τη διαθεσιμότητα των πληροφοριών.

Ακριβό κόστος δεδομένων. Τα γεωγραφικά δεδομένα είναι ακριβά σε κόστος. Σε αντίθεση με τις ΗΠΑ, όπου τα γεωγραφικά δεδομένα συλλέγονται από κρατικούς φορείς και είναι συνήθως διαθέσιμα δωρεάν ή πολύ φθηνά, τα γεωγραφικά δεδομένα είναι συνήθως πολύ ακριβά στην Ευρώπη ή σε άλλα μέρη του κόσμου.

Περιορισμένος χώρος οθόνης Η/Υ. Όπως και με τους άλλους χάρτες που βασίζονται στην οθόνη, οι διαδικτυακοί χάρτες έχουν το πρόβλημα του περιορισμένου χώρου της οθόνης. Αυτό είναι ιδιαίτερα πρόβλημα για τους κινητούς διαδικτυακούς χάρτες και για υπηρεσίες εντοπισμού, όπου οι χάρτες πρέπει να εμφανίζονται σε πολύ μικρές οθόνες με ανάλυση πολύ χαμηλή. Η τεχνολογική πρόοδος όμως βοηθάει να ξεπεραστούν τέτοιου είδους περιορισμοί.

Ποιότητα και ακρίβεια. Πολλοί διαδικτυακοί χάρτες είναι κακής ποιότητας, τόσο σε επίπεδο συμβολισμών, όσο και σε περιεχόμενο και σε ακρίβεια δεδομένων.

Πολύπλοκη ανάπτυξη. Παρά την αυξανόμενη διαθεσιμότητα των ελεύθερων και εμπορικών εργαλείων αλλά και μέσων για την δημιουργία διαδικτυακής χαρτογράφησης και εφαρμογών web GIS, εξακολουθεί να είναι πολύπλοκη η δημιουργία των διαδραστικών διαδικτυακών χαρτών. Πολλές τεχνολογίες, προγράμματα, υπηρεσίες και πηγές δεδομένων πρέπει να συγκεντρωθούν και να ενοποιηθούν για την παραγωγή τέτοιου είδους χαρτών.

Πνευματικά δικαιώματα. Πολλοί άνθρωποι εξακολουθούν να είναι απρόθυμοι να δημοσιεύσουν γεωγραφικά δεδομένα, ιδίως υπό το πρίσμα ότι τα γεωγραφικά δεδομένα είναι ακριβά σε ορισμένα μέρη του κόσμου. Φοβούνται τις παραβιάσεις των δικαιωμάτων πνευματικής ιδιοκτησίας από τα άλλα άτομα τα οποία χρησιμοποιούν τα δεδομένα τους χωρίς τις κατάλληλες αιτήσεις για άδεια.

Προστασία προσωπικών δεδομένων. Με τις λεπτομερείς πληροφορίες που διατίθενται και σε συνδυασμό με τις κατανεμημένες πηγές δεδομένων, είναι δυνατόν να αποκαλυφθούν και να συνδυαστούν πολλές ιδιωτικές και προσωπικές

πληροφορίες μεμονωμένων ατόμων. Για παράδειγμα, ιδιότητες και προσωπικές περιουσίες (ακίνητα) φυσικών προσώπων είναι πλέον προσβάσιμες μέσω υψηλής ανάλυσης αεροφωτογραφιών και δορυφορικών εικόνων σε ολόκληρο τον κόσμο.

4.4 Ιστορική αναδρομή

Η ενότητα αυτή περιέχει μερικές από τις χρονικές στιγμές - ορόσημα της διαδικτυακής χαρτογράφησης και των διαδικτυακών υπηρεσιών χαρτογράφησης. Επειδή όμως, η διαδικτυακή χαρτογράφηση εξαρτάται και από τις τεχνολογίες του διαδικτύου, η ενότητα αυτή περιλαμβάνει επίσης ορισμένες χρονικές στιγμές - ορόσημα της ανάπτυξης των τεχνολογιών του διαδικτύου [wwwWWM].

❖ Περίοδος 1989 – 1995:

- 1989-1909: *Γέννηση του WWW*, το WWW εφευρέθηκε στο CERN για την ανταλλαγή ερευνητικών εγγράφων.
- 1990-12: *Πρώτος Web Browser και Web Server*, ο Tim Berners-Lee έγραψε το πρώτο πρόγραμμα περιήγησης στο web και τον web server.
- 1991-04: *Πρωτόκολλο HTTP 0.9*, ο αρχικός σχεδιασμός του πρωτόκολλου HTTP για την επικοινωνία μεταξύ του browser και του server.
- 1991-06: *ViolaWWW 0.8 Browser*, το πρώτο δημοφιλές πρόγραμμα περιήγησης στο web. Γραμμένο για X11 πάνω σε Unix.
- 1991-08: Το πρόγραμμα WWW ανακοινώθηκε δημοσίως, θεωρείται ως η ημερομηνία ντεμπούτο του Παγκόσμιου Ιστού. Ανακοινώθηκε στην ομάδα ειδήσεων alt.hypertext.
- 1992-06: *Πρωτόκολλο HTTP 1.0*, έκδοση 1.0 του πρωτοκόλλου HTTP. Εισάγει την POST μέθοδο και τις εξακολουθητικές συνδέσεις (persistent connections).
- 1993-04: Το CERN ανακοίνωσε τον Παγκόσμιο Ιστό ως δωρεάν, το CERN ανακοίνωσε ότι η πρόσβαση στο διαδίκτυο θα είναι ελεύθερη για όλους. Το διαδίκτυο προσείλκυσε κρίσιμη μάζα ανθρώπων.
- 1993-06: *HTML 1.0*. Η πρώτη έκδοση του HTML , που δημοσιεύθηκε από τον T. Berners-Lee και Dan Connolly.
- 1993-07: *Xerox PARC Map Viewer* , ο πρώτος mapserver που βασίζεται σε CGI/Perl , επιτρέποντας την επαναπροβολή του στυλ και τον καθορισμό της έκτασης του χάρτη.
- 1994-10: *Netscape Browser 0.9 (Mosaic)*, η πρώτη έκδοση του εξαιρετικά δημοφιλούς προγράμματος περιήγησης Netscape Navigator .
- 1995-03: *Java 1.0*, η πρώτη δημόσια έκδοση της Java.
- 1995-11: *HTML 2.0*, παρουσίασε υποδείγματα, φόρτωση αρχείων, διεθνοποίηση και εικόνες χάρτη από την πλευρά του πελάτη (client-side image maps).
- 1995-12: *Javascript 1.0*, παρουσίασε το πρώτο script βασισμένο στη διαδραστικότητα.
- 1995: *MapGuide* , την πρώτη φορά παρουσιάστηκε ως Argus MapGuide.

❖ Περίοδος 1996 – 2000:

- 1996-01: *JDK 1.0*, πρώτη έκδοση του Sun JDK.
- 1996-02: Mapquest, η πρώτη διάσημη διαδικτυακή υπηρεσία χαρτογράφησης διευδύνσεων και δρομολόγησης, με χαρτογραφική απόδοση.
- 1996-06: *Multimap*, το Ηνωμένο Βασίλειο με βάση το δικτυακό τόπο Multimap ξεκίνησε προσφέροντας διαδικτυακή χαρτογράφηση, δρομολόγηση και υπηρεσίες εντοπισμού (location based services).
- 1996-11: Geomedia WebMap 1.0, η πρώτη έκδοση του Geomedia WebMap, που ήδη υποστηρίζει διανυσματικά γραφικά μέσω της χρήσης των ActiveCGM.
- 1996-πτώση: *MapGuide*, η Autodesk απέκτησε την Argus Technologies και παρουσίασε τον Autodesk MapGuide 2.0.
- 1996-12: Macromedia Flash 1.0, η πρώτη έκδοση του Flash plugin της Macromedia.
- 1997-01: HTML 3.2, παρουσίασε πίνακες, applets, script στοιχεία, στοιχεία πολυμέσων, flowtext around images, κ.λπ.
- 1997-03: Η Νορβηγική εταιρεία Mapnet ξεκινά την εφαρμογή για τον ιστότοπο www.epi.no με ενεργό στρώμα POI για τις λίστες περιουσιακών ακινήτων.
- 1997-07: UMN MapServer 1.0, αναπτύχθηκε στο πλαίσιο του έργου ForNet της NASA.
- 1997-12: HTML 4.0, παρουσίασε το στυλ με CSS , την απόλυτη και σχετική θέση των στοιχείων, τα πλαίσια , τα στοιχεία αντικειμένων, κ.α.
- 1998-06: Terraserver *HIIA*, κυκλοφόρησε μία διαδικτυακή υπηρεσία χάρτη που εξυπηρετεί αεροφωτογραφίες (κυρίως ασπρόμαυρες) και USGS DRGs. Ένα από τα πρώτα δημοφιλή WMS. Η υπηρεσία αυτή είναι μια κοινή προσπάθεια της USGS, της Microsoft και της HP.
- 1998-07: UMN MapServer 2.0, προστέθηκε η στήριξη της αλλαγής προβολικού συστήματος (PROJ.4).
- 1998-08: MapObjects Internet Map Server, η είσοδος της ESRI στην διαδικτυακή χαρτογράφηση.
- 1999-03: Το πρωτόκολλο HTTP 1.1, έκδοση 1.1 του πρωτοκόλλου HTTP. Εισάγει την αίτηση pipelining για πολλαπλές συνδέσεις μεταξύ του εξυπηρετητή και του πελάτη.
- 2000-02: ArcIMS 3.0, η πρώτη δημόσια έκδοση της ESRI του ArcIMS.
- 2000-06: Γεωγραφικό Δίκτυο της ESRI, η ESRI ίδρυσε το Γεωγραφικό Δίκτυο για τη διανομή των δεδομένων και των υπηρεσιών διαδικτυακής χαρτογράφησης.
- 2000-06: UMN MapServer 3.0, αναπτύχθηκε στο πλαίσιο του έργου TerraSIP της NASA. Αυτή είναι και η πρώτη δημόσια, ανοικτού κώδικα έκδοση UMN Mapserver. Προστέθηκε υποστήριξη ψηφιδωτών και υποστήριξη για γραμματοσειρές TrueType (Free Type).
- 2000-08: Flash Player 5, παρουσιάζει το ActionScript 1.0 (ECMAScript συμβατότητα).

❖ Περίοδος 2001 – 2005:

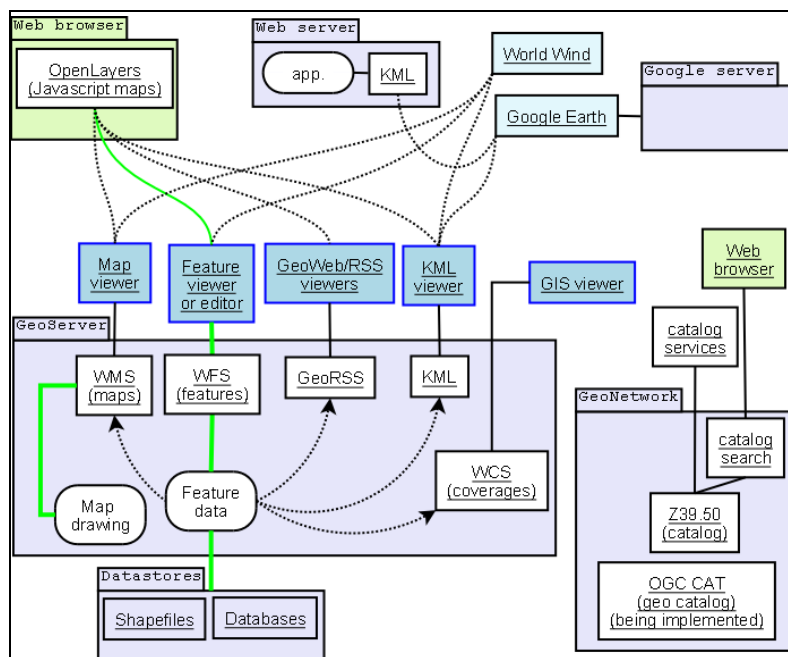
- 2001-06: MapScript 1.0 για τον UMN MapServer, προσθέτει μεγάλη ευελιξία σε UMN MapServer επιλύσεις.
- 2001-09: SVG 1.0 W3C υπόδειξη, SVG (Scalable Vector Graphics) 1.0 έγινε υπόδειξη του W3C.
- 2002-06: UMN MapServer 3.5, προστέθηκε υποστήριξη για PostGIS και ArcSDE. Η έκδοση 3.6 προσθέτει αρχική υποστήριξη OGC WMS.
- 2002-07: ArcIMS 4.0, έκδοση 4 του ArcIMS web map server.
- 2003-01: SVG 1.1 W3C υπόδειξη, το SVG 1.1 έγινε υπόδειξη του W3C. Αυτό παρουσιάζει τα προφίλ των κινητών SVG Tiny και SVG Basic.
- 2003-06: *NASA World Wind*, εκδίδεται η *NASA World Wind*. Ένας ανοικτός εικονικός κόσμος που φορτώνει δεδομένα από κατανεμημένους πόρους μέσω του διαδικτύου. Η επιφάνεια και τα κτίρια, μπορούν να αναπαρασταθούν με 3 διαστάσεις. Η γλώσσα επισήμανσης (βασισμένη στο πρότυπο XML) επιτρέπει στους χρήστες να ενσωματώσουν το δικό τους προσωπικό περιεχόμενο. Αυτή η εικονική υδρόγειος χρειάζεται ειδικό λογισμικό και δεν εκτελείται σε ένα web browser.
- 2003-07: UMN MapServer 4.0, προσθέτει 24bit υποστήριξη εξόδου διανυσματικών και υποστήριξη για PDF και SWF.
- 2003-09: *Flash Player 7*, παρουσιάζει το ActionScript 2.0 (ECMAScript 2.0 συμβατότητα (βελτιωμένο προσανατολισμό αντικειμένων)). Επίσης, αρχική υποστήριξη αναπαραγωγής βίντεο.
- 2004-07: Ιδρύθηκε το OpenStreetMap από τον Steve Coast. Το OSM είναι ένα πρόγραμμα συνεργασίας βασισμένο στο διαδίκτυο για τη δημιουργία ενός παγκόσμιου χάρτη με ελεύθερη άδεια.
- 2005-01: Ο Nikolas Schiller δημιουργεί τον διαδραστικό «Ιδρυτικό Χάρτη» του κέντρου της πόλης της Washington, DC.
- 2005-02: *Google Maps*, η πρώτη έκδοση του Google Maps. Βασισμένο σε ψηφιδωτή επιφάνεια οργανωμένη με γραφικό περίγραμμα τετραδικού δέντρου και η φόρτωση των στοιχείων να γίνεται με XML Http Requests. Η εφαρμογή αυτή χαρτογράφησης έγινε ιδιαίτερα δημοφιλής στον παγκόσμιο ιστό, επειδή επέτρεψε σε άλλους ανθρώπους να ενσωματώσουν τις υπηρεσίες του Google map στη δική τους ιστοσελίδα.
- 2005-04: UMN MapServer 4.6, προσθέτει υποστήριξη για το SVG.
- 2005-06: *To Google Earth*, η πρώτη έκδοση του Google Earth κυκλοφόρησε με βάση την εικονική μεταφορά του πλανήτη. Η επιφάνεια και τα κτίρια, μπορούν να αναπαρασταθούν σε 3 διαστάσεις. Η γλώσσα επισήμανσης KML (βασισμένη στο πρότυπο XML) επιτρέπει στους χρήστες να ενσωματώσουν το δικό τους προσωπικό περιεχόμενο. Αυτή η εικονική υδρόγειος χρειάζεται ειδικό λογισμικό και δεν εκτελείται σε ένα web browser.
- 2005-01: *Firefox 1.5*, η πρώτη έκδοση του Firefox με την εγγενή υποστήριξη SVG. Υποστηρίζει Scripting αλλά όχι animation.

❖ Περίοδος 2006 – 2009:

- 2006-05: Το ξεκίνημα της Wikimapia.
- 2006-06: Opera 9, η Opera κυκλοφορεί την έκδοση 9 με την εκτεταμένη υποστήριξη SVG (συμπεριλαμβανομένων scripting και animation).
- 2006-08: SVG 1.2 Mobile Candidate Recommendation, αυτό το κινητό προφίλ του SVG παρουσιάζει βελτιωμένη υποστήριξη πολυμέσων και πολλά χαρακτηριστικά που απαιτούνται για την κατασκευή στο διαδίκτυο εμπλουτισμένων διαδικτυακών εφαρμογών .
- 2009-01: Δημιουργία των Ovi Maps, από την Nokia, για τα smart phones της.

4.5 Τεχνολογίες διαδικτυακής χαρτογράφησης

Τα τελευταία χρόνια ο αριθμός των τεχνολογιών που έχουν παραχθεί για την υλοποίηση έργων διαδικτυακής χαρτογράφησης είναι αρκετά μεγάλος. Πλέον, μπορεί να χρησιμοποιηθεί για την υλοποίηση έργων διαδικτυακής χαρτογράφησης οποιοδήποτε περιβάλλον και γλώσσα προγραμματισμού, καθώς και πλαίσιο εργασίας από πλευράς εξυπηρετητή (server side). Σε κάθε περίπτωση οι δύο τεχνολογίες, δηλαδή από την πλευρά του εξυπηρετητή και από την πλευρά του πελάτη πρέπει να χρησιμοποιούνται. Επίσης, όπως είναι γνωστό το OGC (Open Geospatial Consortium) [wwwOGC] έχει προδιαγράψει τους βασικούς τρόπους διάθεσης των γεωγραφικών και χαρτογραφικών δεδομένων στον παγκόσμιο ιστό με μία σειρά τυποποιημένων υπηρεσιών και προτύπων, όπως τα WMS, WFS, WCS, WMC, SLD, GML, KML κ.α. (βλ. σχήμα 4.7). Συνεπώς γίνεται μια αναφορά στα πρωτόκολλα WMS, WFS και WCS, τα οποία είναι και οι πιο δημοφιλείς διαδικτυακές υπηρεσίες για χάρτες (web map services).

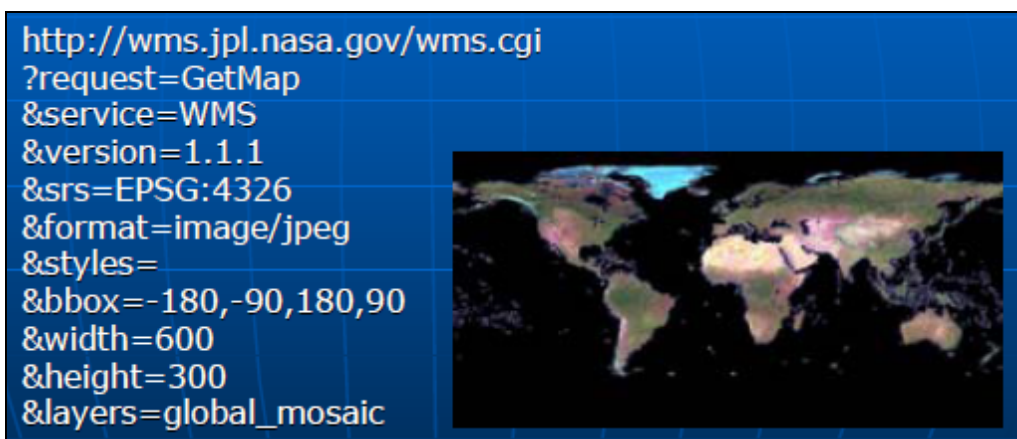


Σχήμα 4.7: Συσχέτιση μεταξύ πελατών / εξυπηρετητών και OGC πρωτοκόλλων.
(Πηγή: [wwwWOGC])

4.5.1 Πρωτόκολλο Web Map Service (WMS)

Με τον όρο *Web Map Service (WMS)*, εννοείται ένα πρωτόκολλο το οποίο παράγει χάρτες με γεωαναφερόμενα δεδομένα στο διαδίκτυο. Το συγκεκριμένο πρωτόκολλο αναπτύχθηκε και δημοσιεύθηκε αρχικά από το OGC το 1999. Η υπηρεσία WMS αποτελεί μια μέθοδο αποστολής χαρτών σε μορφή εικόνας (GIF, PNG, BMP, κ.τλ.) ή περιστασιακά σε διανυσματική μορφή γραφικών στοιχείων (Scalable Vector Graphics - SVG) ή σε μορφή Web Computer Graphics Metafile (WebCGM), στον παγκόσμιο ιστό, όπου οι χάρτες χρησιμοποιούνται για την οπτικοποίηση των δεδομένων και δεν αποτελούν τα δεδομένα καθ' αυτά. Οι προδιαγραφές του OGC προσδιορίζουν τρεις διαφορετικούς τύπους αιτημάτων που παρέχονται μέσω WMS [wwwWMS]:

- Αίτημα «**GetMap**», με το οποίο ο πελάτης ζητάει ένα χάρτη από ένα εξυπηρετητή που διαθέτει την υπηρεσία WMS και του επιστρέφει ένα χάρτη με ακριβή χωρικά και γεωμετρικά χαρακτηριστικά (βλ. σχήμα 4.8). Το αίτημα συνοδεύεται από μια σειρά παραμέτρων, που ορίζουν το περιεχόμενο του χάρτη, τη χωρική του έκταση, το προβολικό του σύστημα, τη μορφή της εικόνας (format), κ.α.
- Αίτημα «**GetCapabilities**», με το οποίο ο πελάτης ζητάει από τον εξυπηρετητή να τον ενημερώσει για τα διαθέσιμα δεδομένα και ο εξυπηρετητής απαντά με ένα τυποποιημένο κείμενο XML, που περιγράφει τα διαθέσιμα θεματικά επίπεδα και τα μεταδεδομένα τους.
- Αίτημα «**GetFeatureInfo**», με το οποίο ο πελάτης αναζητά τις τιμές των γνωρισμάτων ενός αντικειμένου, που επιλέγει με το ποντίκι του Η/Υ πάνω στο χάρτη που είναι υπό μορφή εικόνας. Αυτό το αίτημα είναι προαιρετικό και επιστρέφει πληροφορίες σχετικά με συγκεκριμένα επιλεγμένα στοιχεία που απεικονίζονται σε έναν χάρτη.



Σχήμα 4.8: Παράδειγμα αιτήματος «Get Map». (Πηγή: [Ste07])

Το πρότυπο OGC προσδιορίζει ένα συντακτικό για URL's που καλεί κάθε ένα από αυτά τα αιτήματα του πελάτη. Επιπλέον προσδιορίζεται μία γλώσσα XML για τα μεταδεδομένα που περιγράφουν τις δικτυακές υπηρεσίες. Όταν ένας χρήστης ζητά από μία υπηρεσία WMS έναν χάρτη, τότε πρέπει να προσδιορίσει τις εξής παραμέτρους:

- ο Ποια πληροφορία επιθυμεί να απεικονιστεί στον χάρτη (θεματικά επίπεδα).
- ο Τον τρόπο απεικόνισης (στυλ διαμόρφωσης) του χάρτη.
- ο Το τμήμα της γήινης επιφάνειας που θα περιλαμβάνει ο χάρτης.
- ο Το σύστημα προβολής των συντεταγμένων.
- ο Τη μορφή εικόνας του χάρτη (π.χ. JPEG, BMP, TIFF, PNG, κ.α.).
- ο Τις διαστάσεις του χάρτη.
- ο Την διαφάνεια και το χρώμα του φόντου.

Συνεπώς, μέσω του πρωτοκόλλου WMS, ο χρήστης - πελάτης της διαδικτυακής υπηρεσίας, καθορίζοντας τις παραπάνω παραμέτρους παίρνει ως επιστροφή από τον εξυπηρετητή έναν χάρτη σε μορφή εικόνας που ο ίδιος έχει προσδιορίσει και συγχρόνως ένα τυποποιημένο κείμενο σε μορφή XML, GML ή KML, την πληροφορία των δεδομένων που απεικονίζονται πάνω στον χάρτη.

4.5.2 Πρωτόκολλο Web Feature Service (WFS)

Με τον όρο *Web Feature Service (WFS)*, εννοείται ένα πρωτόκολλο το οποίο ορίζει τις προδιαγραφές με τις οποίες ένας πελάτης - χρήστης μπορεί να περιγράφει και να αποστέλλει ερωτήματα και διεργασίες διαχείρισης δεδομένων πάνω σε γεωγραφικά γνωρίσματα των δεδομένων, χρησιμοποιώντας πάλι το διαδίκτυο (HTTP) ως πλατφόρμα διανομής, αλλά αυτή την φορά έτσι ώστε να επιστρέφονται στον χρήστη - πελάτη τα δεδομένα καθ' αυτά αντί για αναπαραστάσεις τους σε χάρτη όπως με το WMS. Στην ουσία η υπηρεσία WFS αποτελεί μια μέθοδο διάχυσης γεωγραφικών οντοτήτων στον παγκόσμιο ιστό και διανυσματικά δεδομένα σε μορφή GML, σε αντίθεση με την υπηρεσία WMS, που επιστρέφει χάρτες σε μορφή εικόνας (GIF, PNG, κ.λπ.). Οι διεργασίες διαχείρισης δεδομένων περιλαμβάνουν τη δυνατότητα:

- ο Επιλογής ή υποβολής ερωτήματος στα χαρακτηριστικά γνωρίσματα των δεδομένων, βασισμένη σε χωρικά και μη χωρικά κριτήρια.
- ο Δημιουργίας ενός νέου στιγμιότυπου των χαρακτηριστικών γνωρισμάτων (feature instance).
- ο Διαγραφής ενός στιγμιότυπου των χαρακτηριστικών γνωρισμάτων.
- ο Ενημέρωσης ενός στιγμιότυπου των χαρακτηριστικών γνωρισμάτων.

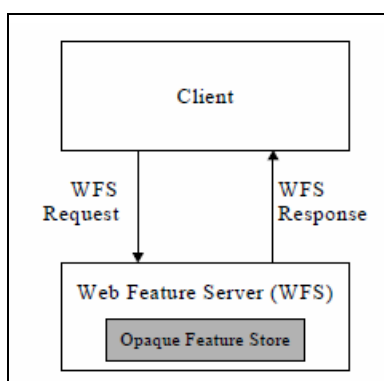
Η πρώτη διεργασία είναι υποχρεωτική και πρέπει να υποστηρίζεται σε κάθε σύστημα υπηρεσιών WFS. Οι υπόλοιπες διεργασίες είναι προαιρετικές. Όταν ένα σύστημα υπηρεσιών WFS υποστηρίζει και τις προαιρετικές διεργασίες τότε λέμε ότι μπορεί να εκτελεί και συναλλαγές και ευθυγραμμίζεται με την

προδιαγραφή WFS-T (*Web Feature Service – Transactional*). Οι προδιαγραφές του OGC προσδιορίζουν τρεις διαφορετικούς τύπους αιτημάτων που παρέχονται μέσω WFS:

- Αίτημα «**GetFeature**», με το οποίο ο πελάτης ζητάει ένα σύνολο γεωγραφικών οντοτήτων σε μορφή GML, από έναν εξυπηρετητή που διαθέτει την υπηρεσία WFS, και το αίτημα του συνοδεύεται από μια σειρά παραμέτρων που ορίζουν τα περιεχόμενα του συνόλου.
- Αίτημα «**GetCapabilities**», με το οποίο ο πελάτης ζητάει παροχή και λήψη πληροφοριών για γεωγραφικές οντότητες που παρέχει ο εξυπηρετητής WFS.
- Αίτημα «**DescribeFeatureType**», με το οποίο ο πελάτης ζητάει πληροφορίες για το σχήμα των γεωγραφικών δεδομένων.

Ένα αίτημα υπηρεσιών WFS αποτελείται από μια περιγραφή ενός ερωτήματος ή διαδικασιών μετασχηματισμού δεδομένων που πρόκειται να εφαρμοστούν σε ένα ή περισσότερα χαρακτηριστικά γνωρίσματα. Όπως φαίνεται διαγραμματικά στο σχήμα 4.9, το αίτημα δημιουργείται στο περιβάλλον του χρήστη και αποστέλλεται στον διαδικτυακό εξυπηρετητή μέσω πρωτοκόλλου HTTP είτε σαν εντολή URL είτε σαν μήνυμα σε γλώσσα GML (*Geography Markup Language*). Ο διαδικτυακός εξυπηρετητής χαρακτηριστικών γνωρισμάτων διαβάζει και στην συνέχεια εκτελεί το αίτημα. Όταν το αίτημα εκτελεστεί, αποστέλλεται πίσω στον χρήστη η απάντηση που επιβεβαιώνει την επιτυχή εκτέλεση και περιλαμβάνει τα δεδομένα που ζητήθηκαν ή τροποποιήθηκαν. Οι πληροφορίες που αποτελούν την απάντηση και αποστέλλονται πίσω στον χρήστη είναι επίσης κωδικοποιημένες σε γλώσσα GML. Το πρότυπο WFS του OGC επιτρέπει σε έναν χρήστη να ανακτήσει γεωμετρικές και χωρικές πληροφορίες από πολλαπλές υπηρεσίες WFS.

Συνεπώς, μέσω του πρωτοκόλλου WFS, στον χρήστη – πελάτη της διαδικτυακής υπηρεσίας, επιστρέφονται τα γεωγραφικά χαρακτηριστικά, όπως γεωμετρικά και θεματικά γνωρίσματα και όχι ο χάρτης σε μορφή εικόνας, όπως συμβαίνει με το πρωτόκολλο WMS.



Σχήμα 4.9: Παράδειγμα υπηρεσίας WFS. (Πηγή: [wwwWFS])



Σχήμα 4.10: Υπηρεσία WFS από την European Space Agency Fire. (Πηγή: [Ste07])

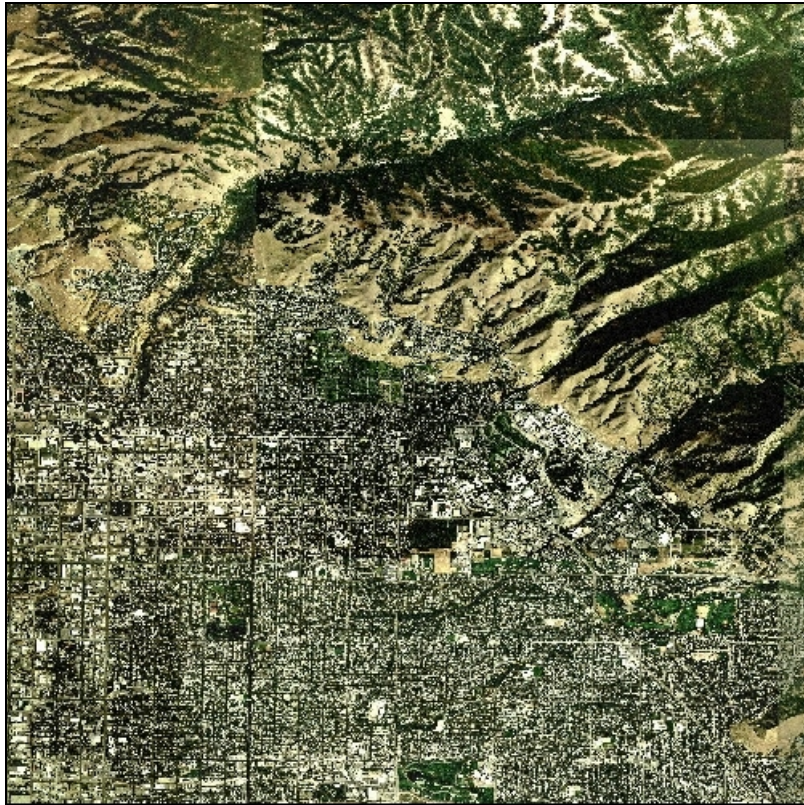
4.5.3 Πρωτόκολλο Web Coverage Service (WCS)

Με τον όρο *Web Coverage Service (WCS)*, εννοείται μία υπηρεσία η οποία αποτελεί μια μέθοδο δημοσιοποίησης των γεωχωρικών δεδομένων στον παγκόσμιο ιστό ως επιστρώματα (coverages), τα οποία περιέχουν τις τιμές ή τις ιδιότητες των γεωγραφικών θέσεων, υποστηρίζοντας τη δικτυωμένη ανταλλαγή αυτών των γεωχωρικών επιστρωμάτων.

Αντίθετα με τους στατικούς χάρτες σε μορφή εικόνας της υπηρεσίας WMS, τα επιστρώματα περιέχουν τιμές ή ιδιότητες στις γεωγραφικές που απεικονίζουν. Δηλαδή αντίθετα από την υπηρεσία WMS, η οποία φιλτράρει και απεικονίζει τα χωρικά δεδομένα για να επιστρέψει στατικούς χάρτες, η υπηρεσία WCS παρέχει την πρόσβαση σε άδικτες γεωχωρικές πληροφορίες και επιστρέφει γεωμετρικά χαρακτηριστικά και θεματικά δεδομένα. Η υπηρεσία WCS αποτελείται από τρεις τύπους αιτημάτων [wwwWCS]:

- Αίτημα «**GetCapabilities**», το οποίο επιστρέφει ένα έγγραφο XML περιγράφοντας την υπηρεσία και τις συλλογές των δεδομένων, από τις οποίες οι πελάτες μπορούν να ζητήσουν τα επιστρώματα.
- Αίτημα «**GetCoverage**», μιας υπηρεσίας WCS το οποίο απαντάται κανονικά αφότου το αίτημα «**GetCapabilities**» έχει καθορίσει ποια ερωτήματα επιτρέπονται και ποια δεδομένα είναι διαθέσιμα. Το αίτημα «**GetCoverage**» επιστρέφει τις τιμές ή τις ιδιότητες των γεωγραφικών θέσεων, τις οποίες συσσωρεύει σε μία γνωστή μορφή εικόνας για επιστρώματα, όπως DTED, GeoTIFF, HDF-EOS, ή NITF. Η σύνταξη και η σημασιολογία του αιτήματος, είναι παρόμοια με το αίτημα «**GetMap**» της υπηρεσίας WMS, αλλά διάφορες επεκτάσεις υποστηρίζουν την ανάκτηση των επιστρωμάτων παρά των στατικών χαρτών.
- Αίτημα «**DescribeCoverageType**».

Συνεπώς, μέσω του πρωτόκολλου WCS, ο χρήστης - πελάτης της διαδικτυακής υπηρεσίας, λαμβάνει έναν χάρτη σε μορφή εικόνας από πρωτογενές υλικό το οποίο είναι σε ψηφιδωτή μορφή (raster). Επίσης τα επιστρώματα είναι αντικείμενα (ή εικόνες) σε μια γεωγραφική περιοχή, ενώ η υπηρεσία WMS επιστρέφει μόνο μια εικόνα, στην οποία οι τελικοί χρήστες δεν μπορούν να παρέμβουν ή να αναλύσουν χωρικά. Παραδείγματα επιστρωμάτων είναι οι δορυφορικές εικόνες (βλ. σχήμα 4.11), που χρησιμοποιούνται στην τηλεπισκόπηση για ανάλυση της φασματικής υπογραφής στις θέσεις του γεωγραφικού χώρου, οι αεροφωτογραφίες εδαφοκάλυψης ή βλάστησης και τα ψηφιακά μοντέλα εδάφους.



Σχήμα 4.11: Παράδειγμα υπηρεσίας WCS, αίτημα «GetCoverage». (Πηγή: [wwwDGR])

Κεφάλαιο 5

Ανάλυση απαιτήσεων και σχεδιασμός εφαρμογής

5.1 Εισαγωγή

Στον σύγχρονο κόσμο, η απευθείας παρακολούθηση κινούμενων αντικειμένων (π.χ. άνθρωποι, ζώα ή συσκευές) έχει δώσει ώθηση σε πολλές μοντέρνες εφαρμογές, όπως υπηρεσίες εντοπισμού, εποπτεία φυσικού περιβάλλοντος, συστήματα πλοήγησης για οχήματα, κ.α. Σε παρόμοιες περιπτώσεις, το σύστημα επεξεργασίας οφείλει να αντεπεξέρχεται σε χωρική πληροφορία που μεταδίδεται συνεχώς από πολυάριθμες κινούμενες πηγές, συγκροτώντας με τις αλληπάλληλες ανανεώσεις των στιγμάτων τους ένα εφήμερο ρεύμα δεδομένων. Αντί της αποθήκευσης σε ένα σύστημα βάσεων δεδομένων, τέτοια μαζικά μεταβαλλόμενα στοιχεία απαιτούν επεξεργασία σε πραγματικό χρόνο. Για παράδειγμα, στην διαχείριση στόλου οχημάτων, οι ελεγκτές θέλουν να ειδοποιούνται όταν τα οχήματα εισέρχονται εντός του δακτυλίου, ενώ οι χρήστες - πελάτες επιθυμούν να εντοπίσουν πόσο είναι ο κυκλοφοριακός φόρτος σε ορισμένους οδικούς άξονες που τους ενδιαφέρουν. Καθώς νέα στίγματα συνεχώς καταφθάνουν, έγκυρες απαντήσεις σε πολύ λίγο χρονικό διάστημα πρέπει να δίνονται σε τέτοια ερωτήματα διαρκείας που έχουν τεθεί από τους χρήστες.

Στο κεφάλαιο αυτό περιγράφεται η αρχιτεκτονική του συστήματος και ο σχεδιασμός της χαρτογραφικής εφαρμογής. Αναλύονται οι απαιτήσεις της εφαρμογής για ένα απλουστευμένο σενάριο επιβολής τέλους συμφόρησης (congestion charge), λαμβάνοντας υπόψη κυρίως χωρικές παραμέτρους (π.χ. την πυκνότητα των οχημάτων, την ιεράρχηση του οδικού δικτύου, το πλήθος των λωρίδων ανά οδικό άξονα, κ.α.), περιγράφονται αναλυτικά οι λειτουργίες της και παρουσιάζονται τα εμπλεκόμενα δεδομένα που σχετίζονται με το πρόβλημα και το προσδιορίζουν. Δύο βασικά στοιχεία της εφαρμογής που πρέπει να αναφερθούν είναι ότι: (α) οι θέσεις των αντικειμένων που αποστέλλονται στην εφαρμογή ως εισερχόμενο ρεύμα δεδομένων, δηλαδή στην προκειμένη περίπτωση τα στίγματα των κινούμενων οχημάτων, είναι διακριτές και όχι συνεχείς και (β) ο σχεδιασμός της εφαρμογής βασίζεται στην παρακολούθηση και επεξεργασία των κινήσεων σημειακών αντικειμένων.

5.2 Αρχιτεκτονική εφαρμογής

Μία γενική αρχιτεκτονική άποψη της χαρτογραφικής εφαρμογής και των επιπέδων που την συνδέτουν, δίνεται μέσω της απεικόνισης του συνολικού σχεδιασμού του συστήματος στο παρακάτω διάγραμμα (βλ. σχήμα 5.1). Η χαρτογραφική εφαρμογή αναπτύχθηκε με γνώμονα την διαχείριση, την επεξεργασία και την απεικόνιση των χωρικών ρευμάτων δεδομένων, για την εποπτεία του κυκλοφοριακού φόρτου στην Αθήνα. Η εφαρμογή σχεδιάστηκε και υλοποιήθηκε ως μία σύνδεση κυρίως δύο επιπέδων:



Σχήμα 5.1: Αρχιτεκτονικό διάγραμμα της εφαρμογής.

Α. «Εξυπηρετητής δεδομένων». Αποτελεί το επίπεδο στο οποίο γίνεται η διαχείριση των δεδομένων και το οποίο δεν είναι προσιτό από τον χρήστη ή τον πελάτη, ο οποίος δεν μπορεί να επεμβαίει με οποιοδήποτε τρόπο σε αυτό. Το επίπεδο αυτό χωρίζεται σε δύο μέρη επεξεργασίας: (α) το ρεύμα δεδομένων το οποίο εισέρχεται στο σύστημα και (β) την εκτέλεση των χωροχρονικών ερωτημάτων διαρκείας και την επιστροφή των αποτελεσμάτων τους. Βασική λειτουργία και για τα δύο μέρη αποτελεί ο μηχανισμός επεξεργασίας των ρευμάτων δεδομένων TelegraphCQ, ο οποίος προσφέρει αυξημένες προηγμένες δυνατότητες διαχείρισης μεγάλου όγκου ανανεώσεων των στιγμάτων των κινούμενων οχημάτων σε πραγματικό χρόνο, καθώς και υποστήριξη πολλαπλών ερωτημάτων διαρκείας. Ειδικότερα:

1^η φάση επεξεργασίας. Όλα τα χωροχρονικά δεδομένα διαχειρίζονται από το σύστημα TelegraphCQ v2.1, το οποίο είναι μία φημισμένη μηχανή επεξεργασίας και διαχείρισης ρευμάτων δεδομένων αξιοποιώντας την χωρική υποδομή της PostgreSQL 7.3.2. Το σύστημα αυτό επιτρέπει την αποτελεσματική εκτέλεση ερωτημάτων διαρκείας, τα οποία μπορούν να τρέχουν παράλληλα έναντι μιας κλιμακώσιμης ροής όγκου δεδομένων. Στα πλαίσια προσομοίωσης της τροφοδοσίας του συστήματος με ένα ρεύμα δεδομένων, το οποίο μεταφέρει κάθε χρονική στιγμή την θέση των οχημάτων αλλά και το χρονόσημο παραγωγής των συντεταγμένων τους (x, y, t), γίνεται χρήση ειδικών wrappers που

διαβάζουν έτοιμα εξωτερικά προπαρασκευασμένα δεδομένα και τα στέλνουν για επεξεργασία στο σύστημα με ένα προκαθορισμένο ρυθμό άφιξης. Την επεξεργασία του ρεύματος δεδομένων που μεταφέρει τις διακριτές θέσεις των σημειακών κινούμενων οχημάτων επάνω στο οδικό δίκτυο, αναλαμβάνει το σύστημα TelegraphCQ, το οποίο δίνει ως έξοδο ορισμένα παραγόμενα ρεύματα.

2^η φάση επεξεργασίας. Με την βοήθεια της PL/pgSQL, διατυπώνονται μόνιμα χωροχρονικά ερωτήματα διαρκείας με κατάλληλα χρονικά παράθυρα, τα οποία συνεχώς εκτελούνται από το σύστημα TelegraphCQ, το οποίο επιστρέφει ως έξοδο σε συγκεκριμένη χρονική στιγμή, τα αποτελέσματα των ερωτημάτων. Η αξιοποίηση της χωρικής υποδομής που προσφέρει η PostgreSQL, διευκολύνει σε μεγάλο βαθμό την σύνταξη και την εκτέλεση των ερωτημάτων.

- B. «Διαδικτυακός εξυπηρετητής».** Αποτελεί το επίπεδο στο οποίο γίνεται η οπτικοποίηση των αποτελεσμάτων των ερωτημάτων διαρκείας πάνω σε ένα διαδικτυακό χάρτη (GoogleMaps). Το επίπεδο αυτό είναι προσιτό από τον χρήστη ή τον πελάτη μόνο οπτικά και συνεχώς “ακούει” το επίπεδο του εξυπηρετητή δεδομένων, για την περιοδική ανανέωση των αποτελεσμάτων που οπτικοποιούνται στην διάρκεια του χρόνου (πχ. ο φόρτος της κυκλοφοριακής κίνησης ανά οδικό άξονα). Η οπτικοποίηση στο επίπεδο αυτό επιτυγχάνεται και υλοποιείται με την βοήθεια εργαλείων της Javascript, PHP και HTML, ενώ η απεικόνιση των αποτελεσμάτων σε χάρτη γίνεται μέσω περιβάλλοντος GoogleMaps. Επίσης τα συγκεκριμένα εργαλεία όχι μόνον καλύπτουν τυπικές λειτουργίες του χάρτη, αλλά επίσης διευκολύνουν δραστικά και την παρουσίαση των αποτελεσμάτων των χωροχρονικών ερωτημάτων υπό μορφή πινάκων.

Πολύ βασικό ζήτημα είναι η σύνδεση και ο τρόπος επικοινωνίας μεταξύ των δύο επιπέδων. Ο εξυπηρετητής δεδομένων είναι συνδεδεμένος με τον διαδικτυακό εξυπηρετητή μέσου δικτύου. Σύμφωνα με όσα αναφέρθηκαν προηγουμένως, αφού προκύψουν από το σύστημα τα παραγόμενα ρεύματα, Ο διαδικτυακός εξυπηρετητής υποβάλλει αίτημα (PHP) μέσω Javascript στον εξυπηρετητή δεδομένων για την προσπέλαση των δεδομένων μέσω των ερωτημάτων, ώστε να αποκτήσει τα αποτελέσματα τους υπό μορφή XML για εμφάνιση σε πίνακα και υπό μορφή KML για χαρτογραφική οπτικοποίηση πάνω στον χάρτη.

Επίσης, ο διαδικτυακός εξυπηρετητής μέσω των εργαλείων που προσφέρει η PHP, κάθε 15 δευτερόλεπτα ζητάει από τον εξυπηρετητή δεδομένων τα ανανεωμένα αποτελέσματα των ερωτημάτων διαρκείας και εκείνος με την σειρά του τα αποστέλλει. Έτσι, με αυτόν τον τρόπο εκτελείται μία συνεχής περιοδική ανανέωση των αποτελεσμάτων της εφαρμογής, η οποία έχει πάντα ως τελικό αποδέκτη τον χρήστη.

Οι δοκιμές της εφαρμογής πραγματοποιήθηκαν σε Η/Υ με κεντρική μονάδα επεξεργασίας 1.67 GHz Athlon XP 2000+, μνήμη 768MB και λειτουργικό σύστημα Ubuntu 9.10.

5.3 Περιγραφή των λειτουργιών του εξυπηρετητή δεδομένων

Ενώ άλλες συγγενικές ερευνητικές προσπάθειες επικεντρώνονται κυρίως στην εξέταση των τρεχόντων στιγμάτων των αντικειμένων, αγνοώντας την σημασία του ιστορικού της κίνησής τους, με την συγκεκριμένη εφαρμογή γίνεται απόπειρα να μελετηθούν οι τροχιές εν εξελίξει και όχι απλώς οι μεμονωμένες θέσεις, επιδιώκοντας να καλυφθούν και να απαντηθούν ερωτήματα διαρκείας κινούμενων αντικειμένων, για κυμαινόμενες χρονικές περιόδους.

Διαπιστώνεται, ότι για τα ρεύματα δεδομένων η έννοια των κυλιόμενων παραθύρων παρέχει ένα βολικό μέσο προσδιορισμού των πρόσφατων τμημάτων για τροχιές. Εξάλλου, κάποια χωρικά ερωτήματα πιθανόν να εμπλέκουν κινούμενες ή/και στατικές οντότητες όσον αφορά τον έλεγχο τοπολογικών συσχετίσεων.

Επισημαίνεται ότι οι θέσεις των οχημάτων που εξετάζονται και αποστέλλονται στην εφαρμογή ως εισερχόμενο ρεύμα δεδομένων είναι διακριτές και ότι ο σχεδιασμός της εφαρμογής βασίζεται στην παρακολούθηση και επεξεργασία των κινήσεων αυτών των σημειακών αντικειμένων.

Στην συνέχεια της ενότητας περιγράφονται αναλυτικά ξεχωριστά οι λειτουργίες που απαιτείται να εκτελεί ο εξυπηρετητής δεδομένων (βλ. ενότητα 5.2). Συγκεκριμένα ακολουθεί μία περιγραφή του τρόπου οργάνωσης του σχήματος της χωρικής βάσης δεδομένων σε περιβάλλον TelegraphCQ, το οποίο προσφέρει αξιοποίηση της χωρικής υποδομής της PostgreSQL. Στην βάση δεδομένων αποτυπώνονται τόσο οι στατικοί πίνακες (π.χ. γεωμετρία δικτύου), όσο και το δυναμικό ρεύμα δεδομένων (θέσεις οχημάτων), καθώς και κάποιοι παράγωγοι δυναμικοί πίνακες που προκύπτουν, μαζί με έναν παράγωγο στατικό πίνακα (χρεώσεις οχημάτων). Επίσης, γίνεται η περιγραφή των ερωτημάτων διαρκείας σε μίαν επέκταση της SQL, την CQL για την χωρική επεξεργασία των θέσεων των κινούμενων οχημάτων και την περιοδική ανανέωση ορισμένων στατιστικών στοιχείων. Τέλος, περιγράφονται οι τρεις βασικές συναρτήσεις σε γλώσσα προγραμματισμού PL/pgSQL που δημιουργήθηκαν για την συνεχή περιοδική εκτέλεση (κάθε 15 δευτερόλεπτα) συγκεκριμένων καθορισμένων μόνιμων χωροχρονικών ερωτημάτων διαρκείας με τα κατάλληλα χρονικά παράθυρα.

5.3.1 Περιγραφή στατικών δεδομένων εφαρμογής

Για τις ανάγκες της υλοποίησης της χαρτογραφικής εφαρμογής δόθηκε έτοιμο το βασικό γεωγραφικό υπόβαθρο των χωρικών δεδομένων, δηλαδή ο χάρτης με το οδικό κυκλοφοριακό δίκτυο της Αθήνας, στον οποίο απεικονίζονται οι γεωγραφικές πληροφορίες που χρησιμοποιούνται και αξιοποιούνται κατάλληλα από το σύστημα. Αναλυτικά η δομή και το περιεχόμενο του γεωγραφικού αρχείου παρατίθεται στο παράρτημα Α. Επειδή το γεωγραφικό αρχείο εξαρχής περιείχε μόνο τα οδικά τμήματα, πραγματοποιήθηκε κατάλληλη επεξεργασία του μέσω του προγράμματος ArcMap 9.3 του ArcGIS, ώστε να δομηθούν τα τμήματα αυτά του οδικού δικτύου σε άξονες σύμφωνα με το καθορισμένο μοντέλο δεδομένων. Επίσης το σύστημα συντεταγμένων του οδικού δικτύου από ΕΓΣΑ'87 μετατράπηκε στο Παγκόσμιο Γεωδαιτικό Σύστημα WGS'84.



Σχήμα 5.2: Χαρτογραφικό υπόβαθρο του οδικού κυκλοφοριακού δικτύου της Αθήνας.

Συνεπώς, το οδικό δίκτυο της Αθήνας (βλ. σχήμα 5.2), το οποίο αποτελεί τον βασικό κορμό των γεωγραφικών δεδομένων, συγκροτείται στην βάση δεδομένων του συστήματος από τους εξής τύπους χωρικών δεδομένων:

Γραμμικά τμήματα δρόμων. Τμήματα τα οποία συνδέουν το κύριο κυκλοφοριακό οδικό δίκτυο της Αθήνας. Πρόκειται για μεμονωμένα τμήματα δρόμων τα οποία συνδέουν τους δρόμους της Αθήνας, οι οποίοι ταξινομούνται σε 5 κατηγορίες.

Γραμμικοί άξονες δρόμων. Άξονες οι οποίοι συνδέουν το κύριο κυκλοφοριακό οδικό δίκτυο της Αθήνας. Πρόκειται για ακολουθίες συνεχόμενων οδικών τμημάτων τα οποία σχηματίζουν οδικούς άξονες.

Η βάση στατικών δεδομένων της εφαρμογής αποτελείται από τις παραπάνω χωρικές οντότητες και μία μη χωρική (αφορά τις χρεώσεις), οι οποίες διαθέτουν χωρικές και περιγραφικές ιδιότητες. Όλες αυτές οι οντότητες είναι καταχωρημένες στους ανάλογους στατικούς πίνακες και σχηματίζουν την βάση

των στατικών δεδομένων της χαρτογραφικής εφαρμογής. Στο παράρτημα Β, απεικονίζονται διαγραμματικά οι οντότητες στους ανάλογους πίνακες, μαζί με τα χωρικά και περιγραφικά χαρακτηριστικά τους.

Όσον αφορά τον καθορισμό των στατικών πινάκων, δημιουργείται αρχικά ένα ειδικό σχήμα (traffic) για την προσαρμογή όλων των χωρικών και μη στοιχείων, στα οποία γίνεται προσπέλαση από την εφαρμογή. Η υλοποίηση των πινάκων πραγματοποιείται με τη χρήση της δηλωτικής γλώσσας SQL που είναι ενσωματωμένη στη PostgreSQL και αποτελεί την δημοφιλέστερη γλώσσα που χρησιμοποιείται για την πρόσβαση στις βάσεις δεδομένων. Επομένως ο ορισμός των πινάκων γίνεται σύμφωνα με τη γλώσσα ορισμού DDL (Data Definition Language) και πραγματοποιείται με χρήση της εντολής CREATE TABLE (βλ. παράρτημα Β).

Στα χωρικά δεδομένα, εκτός της γεωμετρίας τους, καταχωρούνται ορισμένες ιδιότητες, εκ των οποίων κάποιες είναι απαραίτητες για τον σχεδιασμό της χωρικής βάσης, ενώ κάποιες άλλες προστίθενται για την μετέπειτα εξαγωγή πληροφορίας μέσω των λογικών ερωτημάτων που πρέπει να υποστηρίζονται από την εφαρμογή. Κάθε πίνακας περιλαμβάνει ορισμένα χαρακτηριστικά γνωρίσματα (πχ. απλά, σύνθετα ή πλειότιμα), όπου ορισμένα από αυτά αποτελούν τα γνωρίσματα-κλειδιά τους. Η βάση στατικών δεδομένων της εφαρμογής αποτελείται από τους ακόλουθους πίνακες με χωρικές και περιγραφικές ιδιότητες:

Ο πίνακας «**Οδικά τμήματα**» (Road segments) έχει ως χαρακτηριστικά γνωρίσματα τα εξής:

- *Κωδικός (αύξων αριθμός γεωμετρικού συνδέσμου) για κάθε μεμονωμένο οδικό τμήμα που αποτελεί και το πρωτεύον κλειδί του συγκεκριμένου πίνακα.*
- *Κωδικός του οδικού άξονα όπου ανήκει το οδικό τμήμα.*
- *Ονομασία του δρόμου όπου ανήκει το οδικό τμήμα (ελληνικά).*
- *Ονομασία του δρόμου όπου ανήκει το οδικό τμήμα στα λατινικά (μεταγραφή σύμφωνα με τα πρότυπα που ορίζονται από τον ΕΛΟΤ).*
- *Κατηγορία του οδικού τμήματος μέσω ενός κωδικού ιεράρχησης.*
 - 1: Αυτοκινητόδρομος
 - 2: Άξονας ταχείας κυκλοφορίας
 - 3: Κύρια αρτηρία
 - 4: Δευτερεύουσα αρτηρία
 - 5: Συλλεκτήρια οδός
- *Χαρακτηρισμός μονόδρομων*
 - FT: Η κυκλοφορία των οχημάτων ακολουθεί τη φορά της ψηφιοποίησης του συνδέσμου
 - TF: Η κίνηση διεξάγεται ανάποδα από τη φορά της ψηφιοποίησης
 - NO: Αμφίδρομη κυκλοφορία
- *Γεωμετρικό μήκος του οδικού τμήματος (σε μέτρα).*
- *Αριθμός λωρίδων του οδικού τμήματος (βλ. πίνακα 5.1).*
- *Γεωμετρικός τύπος (στη συγκεκριμένη περίπτωση είναι γραμμικός).*

| Κωδικός κατηγορίας δρόμου | Λωρίδες δρόμου |
|---------------------------|----------------|
| 1 | Οκτώ |
| 2 | Έξι |
| 3 | Τέσσερις |
| 4 | Δύο |
| 5 | Μία |

Πίνακας 5.1: Εκτιμώμενη απόδοση λωρίδων ανάλογα με την κατηγορία του δρόμου.

Ο πίνακας «Οδικοί άξονες» (Road axes) έχει ως χαρακτηριστικά γνωρίσματα τα εξής:

- Κωδικός κάθε οδικού άξονα που αποτελεί και το πρωτεύον κλειδί του πίνακα.
- Συντελεστής κυκλοφοριακού φόρτου που αναλογεί σε κάθε οδικό άξονα, ανάλογα το μήκος και τον αριθμό λωρίδων που διαθέτει (χωρητικότητα οδοστρώματος του οδικού άξονα).
- Ονομασία του δρόμου όπου ανήκει ο οδικός άξονας (ελληνικά).
- Ονομασία του δρόμου όπου ανήκει ο οδικός άξονας στα λατινικά (μεταγραφή σύμφωνα με τα πρότυπα που ορίζονται από τον ΕΛΟΤ).

Ο πίνακας «Χρεώσεις» (Charges) έχει ως χαρακτηριστικά γνωρίσματα τα εξής:

- Χρόνος έναρξης περιόδου στην οποία το όχημα χρεώνεται την ανάλογη χρέωση.
- Χρόνος λήξης περιόδου στην οποία το όχημα χρεώνεται την ανάλογη χρέωση.
- Ελάχιστος φόρτος της αντίστοιχης περιόδου στην οποία το όχημα χρεώνεται την ανάλογη χρέωση.
- Μέγιστος φόρτος της αντίστοιχης περιόδου στην οποία το όχημα χρεώνεται την ανάλογη χρέωση.
- Επιβαλλόμενη χρέωση του οχήματος. Πρόκειται για το τέλος κυκλοφοριακής συμφόρησης που χρεώνεται σε κάθε όχημα ανάλογα τον κυκλοφοριακό φόρτο του οδικού άξονα την χρονική στιγμή που τον χρησιμοποιεί (βλ. πίνακα 5.2).
- Κλάση κατηγορίας του κυκλοφοριακού φόρτου. Πρόκειται για το επίπεδο εξυπηρέτησης που παρέχει το οδικό δίκτυο στα οχήματα, το οποίο αλλάζει ανάλογα με τις τιμές του ελάχιστου και μέγιστου φόρτου (βλ. πίνακα 5.3).

Σημειώνεται, ότι και τα τέσσερα χαρακτηριστικά γνωρίσματα, ο χρόνος έναρξης και λήξης περιόδου, μαζί με τον ελάχιστο και μέγιστο φόρτο της αντίστοιχης περιόδου, αποτελούν το πρωτεύον κλειδί του πίνακα «Χρεώσεις».

| Χρόνος έναρξης | Χρόνος λήξης | Ελάχιστος φόρτος | Μέγιστος φόρτος | Χρέωση (Τέλος συμφόρησης) | Κατηγορία δρόμου |
|----------------|--------------|------------------|-----------------|---------------------------|------------------|
| 11:00:00 | 13:00:00 | 0 | 19 | 0 | A |
| 11:00:00 | 13:00:00 | 19 | 32 | 0.5 | B |
| 11:00:00 | 13:00:00 | 32 | 48 | 1 | C |
| 11:00:00 | 13:00:00 | 48 | 68 | 1.5 | D |
| 11:00:00 | 13:00:00 | 68 | 108 | 2 | E |
| 11:00:00 | 13:00:00 | 108 | 200 | 3 | F |

Πίνακας 5.2: Εκτιμώμενη απόδοση τέλους συμφόρησης, ανάλογα την κατηγορία του δρόμου και τη χρονικής περιόδου μετακίνησης των οχημάτων.

| Επίπεδο εξυπηρέτησης οδικής κυκλοφορίας (Level Of Service) | | | | |
|--|--|------------------------------------|--|---|
| Κλάση κατηγορίας φόρτου | Περιγραφή κυκλοφοριακής ροής | Εκτιμώμενη καθυστέρηση κυκλοφορίας | Ελάχιστος φόρτος (Οχήματα / Km / Λωρίδα) | Μέγιστος φόρτος (Οχήματα / Km / Λωρίδα) |
| A | Η ροή είναι ίση ή παραπάνω από το όριο ταχύτητας. Οι αυτοκινητιστές έχουν πλήρη κινητικότητα μεταξύ των λωρίδων του οδικού δικτύου. | Λίγη ή καθόλου | 0 | 19 |
| B | Ελαφρώς κορεσμένη ροή με κάποια ικανότητα ελιγμών στις λωρίδες. Δύο αυτοκινητιστές μπορεί να αναγκάζονται να οδηγούν δίπλα-δίπλα, γεγονός που περιορίζει την αλλαγή λωρίδων. | Μικρή | 19 | 32 |
| C | Η δυνατότητα διάσχισης ή αλλαγής λωρίδων δεν διασφαλίζεται. Οι περισσότεροι πεπειραμένοι οδηγοί είναι άνετοι και η ταχύτητα διατηρείται, αλλά οι δρόμοι είναι κοντά στην μέγιστη κυκλοφοριακή ικανότητα ροής οχημάτων. Αυτό είναι ο στόχος για τις περισσότερες αστικές εθνικές οδούς. | Μεσαία | 32 | 48 |
| D | Οι ταχύτητες είναι κάπως μειωμένες και οι αυτοκινητιστές περιορίζονται από άλλα οχήματα. Χαρακτηριστικές αστικές συνθήκες εθνικών οδών σε ώρες αιχμής. | Μεγάλη | 48 | 68 |
| E | Η ροή γίνεται ακανόνιστη, η ταχύτητα ποικίλλει και φθάνει σπάνια το όριο ταχύτητας. Αυτό θεωρείται αποτυχία του κυκλοφοριακού συστήματος. | Πολύ μεγάλη | 68 | 108 |
| F | Η ροή έχει συχνές πτώσεις ταχύτητας έως σχεδόν μηδενική κίνηση οχημάτων. Ο χρόνος ταξιδιού είναι απρόβλεπτος. | Συμφόρηση | 108 | 200 |

Πίνακας 5.3: Εκτιμώμενο επίπεδο υπηρεσιών του οδικού δικτύου.

Στον πίνακα 5.2, δίνεται μία αυθαίρετη κοστολόγηση της χρήσης των οδικών αξόνων ανάλογα με την χρονική περίοδο που χρησιμοποιείται, τον κυκλοφοριακό φόρτο που υπάρχει στον άξονα την χρονική στιγμή που χρησιμοποιείται από το κινούμενο όχημα και φυσικά από την κλάση της κατηγορίας που κατατάσσεται εκείνη την χρονική στιγμή ο οδικός άξονας σε σχέση με το κυκλοφοριακό φόρτο του. Συμπληρωματικά, στον πίνακα 5.3, περιγράφεται το επίπεδο των εκτιμώμενων υπηρεσιών του οδικού δικτύου και λειτουργεί κυρίως ως ένας δείκτης μέτρησης της έντασης της κυκλοφοριακής συμφόρησης [Tca10].

Συνοψίζοντας, η αποτύπωση των στατικών πινάκων που αποτελούν τα στατικά δεδομένα της εφαρμογής, οργανώνονται κατ' αυτόν τον τρόπο στο σχήμα της χωρικής βάσης δεδομένων σε περιβάλλον TelegraphCQ. Αυτό το σχήμα της χωρικής βάσης δεδομένων, αποτελεί το βασικό γεωγραφικό υπόβαθρο της κίνησης των οχημάτων, τα οποία παρακολουθούνται και αποστέλλουν τις θέσεις τους πάνω στο οδικό κυκλοφοριακό δίκτυο της Αθήνας, για επεξεργασία από το σύστημα σε πραγματικό χρόνο.

Στο παράρτημα Β δίνεται μία πιθανή επέκταση της βάσης δεδομένων με περισσότερους στατικούς πίνακες, οι οποίοι θα μπορούσαν να επεκτείνουν τον σχεδιασμό της χωρικής βάσης και να χρησιμοποιηθούν από την εφαρμογή για περαιτέρω εξαγωγή πληροφορίας μέσω των λογικών ερωτημάτων.

5.3.2 Περιγραφή δυναμικών δεδομένων εφαρμογής

A. Ρεύμα θέσεων κινούμενων οχημάτων

Τα κινούμενα οχήματα μοντελοποιούνται ως δισδιάστατα σημεία (x,y) που συνδέονται με τις χρονικές ενδείξεις (t), γνωστές επίσης ως θέσεις με χρονική σήμανση, δεδομένου ότι η θέση τους μπορεί να αλλάξει δυναμικά με την παρέλευση του χρόνου. Υποτίθεται ότι οι ανανεώσεις των θέσεων των οχημάτων συλλέγονται από κινούμενες πηγές χρησιμοποιώντας σύγχρονες συσκευές, όπως GPS, PDA, κ.λπ., οι οποίες επικοινωνούν σε πραγματικό χρόνο με τον κεντρικό επεξεργαστή του συστήματος.

Όλο το ενδιαφέρον εστιάζεται στην διαχείριση αυτών των χωροχρονικών δεδομένων μόλις διατεθούν στη μηχανή επεξεργασίας, προκειμένου να δοθούν σε πραγματικό χρόνο οι απαντήσεις στα ερωτήματα διαρκείας που έχουν διατυπωθεί.

B. Δήλωση δυναμικού πίνακα στο TelegraphCQ

Το TelegraphCQ επεκτείνει την γλώσσα ορισμού δεδομένων (Data Definition Language – DDL), η οποία αποτελεί τμήμα της SQL, με ειδικές εντολές για τη διαχείριση ρευμάτων δεδομένων (CREATE STREAM, ALTER STREAM, DROP STREAM). Όλοι οι τύποι δεδομένων που προσφέρονται από την PostgreSQL 7.3.2 εξακολουθούν να υποστηρίζονται, όμως ένα πρόσθετο χαρακτηριστικό γνώρισμα που ονομάζεται “tcqtime” πρέπει να δηλώνεται, έτσι ώστε να διαχωρίζονται τα ρεύματα δεδομένων από τους συμβατικούς σχεσιακούς πίνακες. Από προεπιλογή του συστήματος το οριζόμενο πεδίο

«TIMESTAMP COLUMN», θα περιέχει την τιμή της χρονικής σήμανσης (timestamp) που αποδίδεται σε κάθε εισερχόμενο στοιχείο δεδομένων του ρεύματος.

Επίσης, το TelegraphCQ υποστηρίζει τόσο *αρχειοθετημένα (archived)* όσο και μη *αρχειοθετημένα (unarchived)* ρεύματα δεδομένων. Ένα αρχειοθετημένο ρεύμα δεδομένων υποστηρίζεται από την αποθήκευσή του στο σκληρό δίσκο του συστήματος, ώστε να στέλνει αποτελεσματικά τις προσαρτημένες μόνο πλειάδες στο σύστημα. Αντιθέτως ένα μη αρχειοθετημένο ρεύμα δεδομένων χρησιμοποιεί τις ουρές της κοινής μνήμης του συστήματος για την επικοινωνία των δεδομένων μεταξύ των πηγών εισαγωγής των δεδομένων και της μηχανής εκτέλεσης των εργασιών. Ωστόσο, παρατηρείται ότι τα σύνθετα χωροχρονικά ερωτήματα τα οποία εκφράζονται με οριζόμενες από το χρήστη PL/pgSQL συναρτήσεις, λειτουργούν μόνο για τα αρχειοθετημένα ρεύματα δεδομένων. Επομένως, στην υλοποίηση της εφαρμογής όλα τα ρεύματα δεδομένων ορίζονται ως *αρχειοθετημένα*.

Όσον αφορά τον καθορισμό των ρευμάτων δεδομένων, δημιουργείται αρχικά ένα ειδικό σχήμα (traffic) για την προσαρμογή όλων των χωρικών στοιχείων, στα οποία γίνεται προσπέλαση από την εφαρμογή. Εφόσον στο TelegraphCQ υποστηρίζονται όλοι οι τύποι δεδομένων που προσφέρονται από την PostgreSQL, υποστηρίζονται και οι απαραίτητοι γεωμετρικοί τύποι δεδομένων (σημεία, γραμμές, πολύγωνα) οι οποίοι μπορούν να χρησιμοποιηθούν για τις χωρικές ιδιότητες. Σε αυτήν την περίπτωση, κάθε εισερχόμενο στίγμα των οχημάτων είναι ουσιαστικά ένα σημείο και ως εκ τούτου, μπορεί να καθοριστεί ένα ρεύμα δεδομένων για τα οχήματα που κινούνται με βάση τα στοιχεία και τα στίγματα που καταγράφονται από αυτά. Το ρεύμα αυτό δηλώνεται μέσω ενός δυναμικού πίνακα (βλ. παράρτημα Β), ως εξής:

Πίνακας «Τοποθεσίες κινούμενων οχημάτων»

```
CREATE STREAM traffic.Locations
(VehicleID CHAR(7),
 Position POINT,
 SegmentID INTEGER,
 T INTEGER,
 Tcqtime TIMESTAMP TIMESTAMPCOLUMN)
TYPE ARCHIVED;
```

Αυτός ο δυναμικός πίνακας «*Τοποθεσίες κινούμενων οχημάτων*» (Locations) έχει ως χαρακτηριστικά γνωρίσματα τα εξής:

- Κωδικός του κινούμενου οχήματος.
- Γεωμετρικός τύπος (στη συγκεκριμένη περίπτωση είναι σημειακός).
- Κωδικός του οδικού τμήματος στο οποίο κινείται το όχημα.
- Χρονική στιγμή (ως ακέραιος).
- Χρονόσημο, το οποίο ορίζεται για κάθε θέση του κινούμενου οχήματος.

Γίνεται η υπόθεση ότι η γεωγραφική αναπαράσταση όλων των εισερχόμενων δεδομένων προσαρμόζεται με τη σημασιολογία της PostgreSQL, διαφορετικά, αριθμητικές στήλες θα μπορούσαν να χρησιμοποιηθούν για να κρατήσουν τις χωρικές συντεταγμένες. Οι χρονικές ενδείξεις απεικονίζουν τον πραγματικό χρόνο όπου κάθε στοιχείο καταγράφηκε και μπορούν να χρησιμοποιηθούν για χρονοσήμανση (timestamping). Οι τιμές του χρονοσήμου προσδιορίζονται από το TelegraphCQ, ώστε να ολοκληρώνουν τη διάταξη μεταξύ των πλειάδων του ρεύματος. Σε περίπτωση που τα χρονόσημα δεν υπάρχουν στα εισερχόμενα στοιχεία, χρησιμοποιείται ο τρέχων χρόνος του συστήματος (χρόνος συναλλαγής).

Για να είναι σε θέση να προωθηθούν τα δεδομένα μέσα στο συγκεκριμένο ρεύμα δεδομένων, πρέπει να υπάρχει ένας κατάλληλος wrapper που θα αναλάβει αυτή την εργασία (βλ. παράρτημα Β). Για παράδειγμα, μπορεί να οριστεί ένας wrapper ο οποίος θα χειρίζεται τις τιμές των παραμέτρων οι οποίες είναι διαχωρισμένες με ερωτηματικό (;), αναμένοντας ότι τα εισερχόμενα δεδομένα θα είναι συμβατά με την διαμόρφωση των χαρακτηριστικών γνωρισμάτων του πίνακα. Έπειτα, συνδέεται το εισερχόμενο ρεύμα θέσεων με τον wrapper που έχει δημιουργηθεί.

Τονίζεται ότι κανένα ενδιάμεσο αποτέλεσμα στοιχείων ή ερωτημάτων δεν εισάγεται στους σχεσιακούς πίνακες. Ακόμα και αν η PostgreSQL υποστηρίζει την δομή των R-δέντρων και GiST, κανένα χωρικό ευρετήριο δεν χτίζεται ή διατηρείται για τα ρεύματα δεδομένων.

Γ. Παραγωγή πειραματικών δεδομένων

Στα πλαίσια της μελέτης των κινούμενων αντικειμένων επιχειρήθηκε η δημιουργία υποθετικών τροχιών, βάσει ενός ψηφιακού χάρτη του πολυεδομικού συγκροτήματος Αθηνών. Το συγκεκριμένο ψηφιακό υπόβαθρο του βασικού οδικού δικτύου της Αθήνας προέρχεται από την εταιρεία ΕΡΑΤΟΣΘΕΝΗΣ Α.Ε., κατόπιν ψηφιοποίησης χαρτών κλίμακας 1:5000 και τηρείται σε διανυσματική μορφή. Η γεωγραφική βάση δεδομένων καλύπτει συνολικά 55 δήμους σε μια έκταση περίπου 250 km². Οι οδικοί άξονες που περιλαμβάνονται διακρίνονται σε κατηγορίες (λεωφόροι ταχείας κυκλοφορίας, πρωτεύουσες και δευτερεύουσες αρτηρίες, βοηθητικοί δρόμοι), καλύπτοντας το σύνολο του κυκλοφοριακού δικτύου της Αττικής. Τα περιγραφικά στοιχεία του υποβάθρου περιλαμβάνουν, μεταξύ άλλων, τη μέση ταχύτητα κίνησης των οχημάτων στη διάρκεια της ημέρας, η οποία έχει προκύψει από επιτόπιες μετρήσεις. Αυτό είναι και το κύριο στοιχείο το οποίο αξιοποιείται για τον υπολογισμό του μέσου χρόνου διαδρομής ενός οχήματος κατά μήκος όλων των συνδέσμων του δικτύου, καθιστώντας τη συγκεκριμένη γεωγραφική βάση δεδομένων κατάλληλη για υπολογισμό της βέλτιστης διαδρομής μέσα στο οδικό δίκτυο της πόλης.

Επομένως, εάν οριστούν δύο ή περισσότερα σημεία πάνω στο χάρτη είναι εφικτός ο υπολογισμός της συντομότερης διαδρομής βάσει του χρόνου που απαιτείται, υπό την προϋπόθεση ότι είναι διαθέσιμο ένα λογισμικό με ενσωματωμένους αλγόριθμους ανάλυσης δικτύου.

Με χρήση του λογισμικού ArcView 3.2 και της επέκτασής του ArcView Network Analyst, δημιουργήθηκαν συνολικά 10000 τροχιές ισάριθμων

κινούμενων οχημάτων. Ως προέλευση και προορισμός κάθε διαδρομής επιλέχθηκαν ζεύγη κόμβων του δικτύου κατά τυχαίο τρόπο (με μια γεννήτρια ψευδοτυχαίων αριθμών). Η τροχιά που συνδέει τα δύο άκρα της πορείας κάθε αντικειμένου, όταν υπολογιστεί, απεικονίζεται στο λογισμικό ως μια ενιαία πολυγραμμική. Προκειμένου όμως η καταγραφή της διαδρομής να συμφωνεί με το μοντέλο του ρεύματος τροχιάς, έγινε δειγματοληψία με ρυθμό ένα σημείο ανά 15 δευτερόλεπτα για κάθε όχημα. Τελικά, προέκυψε ένα αρχείο τροχιών με εγγραφές που φέρουν την ταυτότητα του αντικειμένου, τις συντεταγμένες και το χρονόσημο, όπως περιγράφηκε για τον δυναμικό πίνακα `traffic.Locations`.

Δ. Πειραματικά δεδομένα θέσεων κινούμενων οχημάτων

Σημαντικό υπόβαθρο για την σωστή και επιτυχή εκτέλεση της εφαρμογής αποτελούν τα δεδομένα τα οποία πρόκειται να εισαχθούν στην βάση. Τα δεδομένα που τροφοδοτούν το ρεύμα δεδομένων που αφορά τα κινούμενα οχήματα στο `TelegraphCQ`, παράγονται όπως αναφέρθηκε από προσομοιώσεις της κίνησης των οχημάτων κατά μήκος του οδικού δικτύου της Αθήνας. Όλα τα αντικείμενα μετατοπίζονται με την ίδια ταχύτητα κατά μήκος ενός δεδομένου οδικού τμήματος. Η προσομοίωση αφορά 10000 κινούμενα οχήματα, τα οποία μεταδίδουν την θέση τους κάθε 15 δευτερόλεπτα, για ένα χρονικό διάστημα 50 λεπτών. Αρχικά, όλα τα οχήματα είναι ακινητοποιημένα, αλλά βαθμιαία όλο και περισσότερα αρχίζουν να κινούνται κατά μήκος του οδικού δικτύου. Η διαδρομή κίνησης των οχημάτων που επιλέχθηκε είναι από την περιφέρεια προς το κέντρο του οδικού δικτύου. Ο εσωτερικός χρόνος άφιξης των ενημερώσεων των στιγμάτων των οχημάτων είναι σταθερός για όλα τα οχήματα στο σύνολο των δεδομένων, υποθέτοντας την εντατικότερη κατάσταση με όλα τα οχήματα να δηλώνουν την ενημέρωση των θέσεων τους ταυτόχρονα σε συγκεκριμένο χρονικό διάστημα, δηλαδή κάθε 15 δευτερόλεπτα.

Επομένως, το σύστημα πρέπει ταυτόχρονα να αντιμετωπίσει τις ογκώδεις ενημερώσεις θέσεων, οι οποίες προκαλούν την ανανέωση των αποτελεσμάτων για όλες τις ενεργές ερωτήσεις.

Επίσης γίνεται η παραδοχή ότι καμία ατέλεια ρευμάτων δεν εμφανίζεται, όπως καθυστερήσεις στην άφιξη των στιγμάτων των οχημάτων ή διπλά στίγματα με την ίδια τιμή χρονόσημου, έτσι ώστε να αποφευχθεί οποιαδήποτε τροποποίηση στις κατασκευασμένες διαδρομές και να συντηρηθεί έτσι η συνοχή μεταξύ των τμημάτων των ρευμάτων που αφορούν τις τροχιές των κινούμενων αντικειμένων. Στατικές πληροφορίες (όπως ο τύπος του οχήματος, το χρώμα του αμαξώματος, κ.λπ.) θα μπορούσαν να αποθηκευτούν σε έναν χαρακτηριστικό σχεσιακό πίνακα (`traffic.vehicles`) στην `PostgreSQL`, δεδομένου ότι αυτά τα δεδομένα δεν αναμένεται να αναμεταδίδονται σε πραγματικό χρόνο μαζί με τις κρίσιμες χωροχρονικές πληροφορίες (βλ. παράρτημα Β).

Ε. Τροφодότηση δεδομένων θέσεων οχημάτων στο ρεύμα

Το αρχείο `traffic10k_wgs84.log`, είναι ένα αρχείο τύπου `CSV`, το οποίο περιέχει μια σειρά από θέσεις με το αντίστοιχο χρονόσημο (σε σύστημα συντεταγμένων `WGS'84`), οι οποίες θα μπορούσαν να έχουν συλλεχθεί από την

κίνηση των οχημάτων ή να έχουν παραχθεί από οποιαδήποτε κατάλληλη γεννήτρια. Για παράδειγμα, ένα τμήμα του αρχείου είναι το ακόλουθο:

```
1;(23.63982836,37.95762300);1179;0;'2010-01-12 12:00:00 EET'  
2;(23.82474624,38.00441091);255;0;'2010-01-12 12:00:00 EET'  
3;(23.87519077,38.05723635);2993;0;'2010-01-12 12:00:00 EET'  
1;(23.63980474,37.95884590);5045;1;'2010-01-12 12:00:15 EET'  
2;(23.82372909,38.00465157);3660;1;'2010-01-12 12:00:15 EET'  
3;(23.87380901,38.05689299);7887;1;'2010-01-12 12:00:15 EET'  
1;(23.63976309,37.96006830);5045;2;'2010-01-12 12:00:30 EET'  
2;(23.82278523,38.00502463);3660;2;'2010-01-12 12:00:30 EET'  
3;(23.87242512,38.05655642);7887;2;'2010-01-12 12:00:30 EET'
```

Οι τιμές σε κάθε εισερχόμενη εγγραφή πρέπει να αντιστοιχούν στο σχήμα των χαρακτηριστικών γνωρισμάτων που ορίζονται για το προκαθορισμένο ρεύμα `traffic.Locations`. Προκειμένου να μεταδοθούν τα δεδομένα στο `TelegraphCQ` και να τροφοδοτηθεί το αντίστοιχο ρεύμα δεδομένων, εκτελείται η ανάλογη εντολή σε περιβάλλον λειτουργικού συστήματος Linux (βλ. παράρτημα Β).

Επίσης δόθηκε έτοιμο το αρχείο `sliderate.pl`, (βλ. παράρτημα Γ), που είναι αρχείο τύπου Perl και το οποίο συνδέεται και τροφοδοτεί με δεδομένα τον αντίστοιχο `wrapper` του `TelegraphCQ`. Αυτός ο `wrapper` είναι μια διαδικασία η οποία ακούει τα αιτήματα σύνδεσης σε συγκεκριμένη θύρα (στη θύρα 5533), από τις πηγές δεδομένων και προωθεί τις πλειάδες στο ρεύμα χρησιμοποιώντας τον συνδεδεμένο τύπο `wrapper` του ρεύματος που στην εφαρμογή είναι ο `trafficwrapper` (βλ. παράρτημα Β).

Εντούτοις, τα δεδομένα ανακτώνται με ένα σταθερό καθορισμένο ρυθμό (10000 πλειάδες ανά 15 δευτερόλεπτα) προσφέροντας τη δυνατότητα το σύστημα να μιμηθεί την επεξεργασία άφιξης του ρεύματος δεδομένων σε πραγματικό χρόνο, πάντα όμως στο καθορισμένο ρυθμό άφιξης των δεδομένων. Το ρεύμα `traffic.Locations` είναι το βασικό ρεύμα για όλα τα δεδομένα θέσης των οχημάτων και τροφοδοτεί το σύστημα με όλα τα απαραίτητα δεδομένα, τα οποία πρέπει να εισέρχονται κανονικά στο σύστημα `TelegraphCQ`, ώστε τα ερωτήματα διαρκείας που ενδιαφέρουν την εφαρμογή να μπορούν να διατυπώνονται στο αντίστοιχο ρεύμα.

ΣΤ. Παράγωγα ρεύματα και πίνακες

Δεδομένου ότι η εφαρμογή πρέπει να μπορεί να διαχειρίζεται και να επεξεργάζεται τις τροχιές των κινούμενων οχημάτων, πρέπει επίσης να μπορεί να διατηρεί την εξελισσόμενη πληροφορία της κίνησης των οχημάτων. Για τον λόγο αυτόν, δημιουργείται ένα ρεύμα δεδομένων των γραμμικών μετατοπίσεων (`displacements`), τα οποία συνδέουν τις διαδοχικές θέσεις που καταγράφονται για κάθε όχημα (τις τελευταίες 2 θέσεις κάθε φορά). Λόγω του τύπου των δεδομένων που είναι διαθέσιμος στην `PostgreSQL`, στο παραγόμενο ρεύμα τα δεδομένα για τις μετατοπίσεις κάθε οχήματος μπορούν να υπολογιστούν σε πραγματικό χρόνο από το αρχικό ρεύμα `traffic.Locations` και έπειτα να εισαχθούν σε αυτό. Το παράγωγο ρεύμα αυτό δηλώνεται μέσω ενός δυναμικού πίνακα (βλ. παράρτημα Β) στο `TelegraphCQ`, ως εξής:

Πίνακας «Τροχιές κινούμενων οχημάτων»

```
CREATE STREAM traffic.Trajectories
(VehicleID      CHAR(7) ,
 AxisID         INTEGER,
 Displacement   DOUBLE PRECISION,
 Speed DOUBLE   PRECISION,
 Apply_charge   BOOLEAN,
 tcqtime       TIMESTAMP TIMESTAMPCOLUMN)
TYPE ARCHIVED;
```

Αυτός ο παράγωγος πίνακας «Τροχιές κινούμενων οχημάτων» (Trajectories) έχει ως χαρακτηριστικά γνωρίσματα τα εξής:

- Κωδικός του κινούμενου οχήματος.
- Μέγεθος της μετατόπισης του κινούμενου οχήματος μεταξύ δύο θέσεων (από την προηγούμενη θέση στην τωρινή του).
- Ταχύτητα κίνησης του οχήματος.
- Δείκτης επιβολής χρέωσης ή μη επιβολής χρέωσης του οχήματος με το αντίστοιχο τέλος συμφόρησης για τον οδικό άξονα στον οποίο κινείται και χρησιμοποιεί για κάθε χρονική στιγμή.
- Χρονόσημο που αντιστοιχεί στην θέση του κινούμενου οχήματος.

Το παραγόμενο ρεύμα traffic.Trajectories χρειάζεται για την διατήρηση της πληροφορίας που αφορά: (α) τον οδικό άξονα στον οποίο βρίσκεται και κινείται το κάθε όχημα σε κάθε χρονική στιγμή, (β) την ταχύτητα κίνησης του κάθε οχήματος σε κάθε χρονική στιγμή και (γ) την επιβολή χρέωσης ή όχι σε κάθε κινούμενο όχημα για την διέλευσή του από κάθε οδικό άξονα για κάθε χρονική στιγμή. Εφόσον το όχημα εξακολουθεί να κινείται στον ίδιο οδικό άξονα δεν επιβάλλεται καινούργια επιπρόσθετη χρέωση. Το παραγόμενο ρεύμα με τις τροχιές των κινούμενων οχημάτων γεμίζει με δεδομένα τα οποία επεξεργάζονται, υπολογίζονται και αποστέλλονται μέσω της συνάρτησης createTrajectory, η οποία εξετάζεται στην ακόλουθη ενότητα 5.3.4.

Επίσης, η εφαρμογή πρέπει να μπορεί να υπολογίζει και τον κυκλοφοριακό φόρτο σε κάθε οδικό άξονα του οδικού δικτύου. Για να το καταφέρει αυτό, πρέπει να διατηρεί την εξελισσόμενη πληροφορία του φόρτου των οδικών αξόνων. Γι' αυτό τον λόγο δημιουργείται ακόμη ένα παραγόμενο ρεύμα (traffic.Roadaxes_load), το οποίο χρειάζεται για την διατήρηση της πληροφορίας που αφορά: (α) το κυκλοφοριακό φόρτο για κάθε οδικό άξονα σε κάθε χρονική στιγμή, (β) την κλάση της κατηγορίας στην οποία ανήκει κάθε οδικός άξονας σε κάθε χρονική στιγμή, δηλαδή το επίπεδο εξυπηρέτησης που προσφέρει, καθώς και (γ) την χρέωση που πρέπει να επιβάλλεται σε κάθε κινούμενο όχημα για την διέλευσή του από κάθε οδικό άξονα για κάθε χρονική στιγμή. Το παράγωγο ρεύμα δηλώνεται μέσω ενός δυναμικού πίνακα (βλ. παράρτημα Β) στο TelegraphCQ, ως εξής:

Πίνακας «Φόρτος οδικών αξόνων»

```
CREATE STREAM traffic.Roadaxes_load
(AxisID          INTEGER,
 Vehicle_count   INTEGER,
 Load DOUBLE     PRECISION,
 Charge          DOUBLE PRECISION,
 Load_class      CHAR(1),
 Tcqtime        TIMESTAMP TIMESTAMPCOLUMN)
TYPE ARCHIVED;
```

Αυτός ο παράγωγος πίνακας «Φόρτος οδικών αξόνων» (Road axes load) έχει ως χαρακτηριστικά γνωρίσματα τα εξής:

- Κωδικός του οδικού άξονα.
- Αριθμός των οχημάτων τα οποία κινούνται σε μία δεδομένη χρονική στιγμή επάνω σε κάθε οδικό άξονα.
- Τιμή του φόρτου του οδικού άξονα.
- Χρέωση που πρέπει να επιβληθεί ανάλογα την χρονική στιγμή που διέρχεται από τον άξονα το κινούμενο όχημα (σύμφωνα με τις τιμές που ορίζονται στο στατικό πίνακα «χρεώσεις οχημάτων», όπως φαίνεται στο πίνακα 5.2).
- Κλάση της κατηγορίας του κυκλοφοριακού φόρτου που υπάρχει στον οδικό άξονα (βλ. πίνακα 5.3)
- Χρονόσημο που αντιστοιχεί στην θέση του κινούμενου οχήματος.

Το παραγόμενο ρεύμα με τους φόρτους των οδικών αξόνων, γεμίζει με δεδομένα τα οποία επεξεργάζονται, υπολογίζονται και αποστέλλονται μέσω της συνάρτησης computeLoad, η οποία εξετάζεται στην ακόλουθη ενότητα 5.3.4.

Τέλος, η εφαρμογή πρέπει να μπορεί να υπολογίζει την χρέωση κάθε κινούμενου οχήματος, ανάλογα την χρονική περίοδο και τους οδικούς άξονες που χρησιμοποιεί για την κίνησή του στο οδικό δίκτυο και να επιβάλλεται το αντίστοιχο τέλος κυκλοφοριακής συμφόρησης. Για να το καταφέρει αυτό πρέπει να μπορεί να διατηρεί την εξελισσόμενη πληροφορία της χρέωσης των κινούμενων οχημάτων, επειδή στο τέλος της ημέρας σε κάθε κινούμενο όχημα έχουν επιβληθεί πολλές χρεώσεις.

Γι' αυτό δημιουργείται ένας παραγόμενος πίνακας (traffic.Congestion_charges) και όχι ρεύμα (επειδή πρέπει η πληροφορία να διατηρείται αποθηκευμένη), ο οποίος χρειάζεται για την διατήρηση της πληροφορίας που αφορά τις χρεώσεις που επιβάλλονται σε κάθε κινούμενο όχημα κατά την διάρκεια της ημέρας. Ο παράγωγος πίνακας δηλώνεται μέσω ενός δυναμικού πίνακα (βλ. παράρτημα Β) στο TelegraphCQ, ως εξής:

Πίνακας «Χρεώσεις οχημάτων»

```
CREATE TABLE traffic.Congestion_charges
(VehicleID          CHAR(7),
 Congestion_charge  DOUBLE PRECISION,
 TS                 TIMESTAMP NOT NULL);
```

Αυτός ο παράγωγος πίνακας «Χρεώσεις οχημάτων» (Congestion charges) έχει ως χαρακτηριστικά γνωρίσματα τα εξής:

- Κωδικός του κινούμενου οχήματος.
- Συνολική χρέωση των τελών συμφόρησης που επιβάλλονται στο όχημα.
- Χρονική στιγμή κατά την οποία αποδίδεται η συνολική χρέωση του οχήματος.

Ο παραγόμενος πίνακας με τις χρεώσεις των οχημάτων γεμίζει κατόπιν με δεδομένα τα οποία επεξεργάζονται, υπολογίζονται και αποστέλλονται μέσω της συνάρτησης computeCharge, η οποία εξετάζεται στην ακόλουθη ενότητα 5.3.4.

Είναι σημαντικό να αναφερθεί ότι, το πρωτεύον κλειδί δεν δημιουργείται σκόπιμα στα παραγόμενα ρεύματα και στον παραγόμενο πίνακα, προκειμένου να αποφευχθούν οι καθυστερήσεις κατά τη διάρκεια των εισαγωγών. Όσον αφορά τα ερωτήματα διαρκείας, μπορούν χωρίς πρόβλημα να εμπλέκουν τα παράγωγα ρεύματα και τον παράγωγο πίνακα, τα οποία τροφοδοτούνται με πλειάδες οι οποίες παράγονται μετά από την επεξεργασία των αρχικών δεδομένων.

Κάθε παράγωγος πίνακας περιλαμβάνει ορισμένα χαρακτηριστικά γνωρίσματα, τα οποία παρατίθενται στην συνέχεια για καθένα ξεχωριστά, μαζί με την επεξήγηση τους, ενώ το αντίστοιχο διάγραμμα απεικόνισης τους υπάρχει διαθέσιμο και περιλαμβάνεται στο παράρτημα Β.

5.3.3 Περιγραφή ερωτημάτων διαρκείας της εφαρμογής

Η επεξεργασία των ρευμάτων επιτυγχάνεται με τη χρήση διάφορων τύπων χωροχρονικών ερωτημάτων διαρκείας, π.χ., αναζήτηση αντικειμένων μέσα σε μια περιοχή. Τα αποτελέσματα των ερωτημάτων είναι πάντα επίκαιρα, με τις πιο πρόσφατα λαμβανόμενες ενημερώσεις των θέσεων των αντικειμένων. Στην πραγματικότητα, οι χρήστες μπορούν να ορίσουν παράθυρα, δηλαδή, αναπτύσσοντας τις περιόδους που είναι πιο ενδιαφέρουσες για τους υπολογισμούς τους (π.χ., η εξέταση των θέσεων που λαμβάνονται κατά τη διάρκεια του τελευταίου 20λεπτου), αντί της εξέτασης των ενδεχομένως απεριόριστων τροχιών του αντικειμένου που συσσωρεύονται κατά τη διάρκεια μεγάλων περιόδων.

Όπως αναφέρθηκε και σε προηγούμενη ενότητα, σκοπός της εφαρμογής είναι η διατύπωση κατάλληλων ερωτημάτων διαρκείας σε γλώσσα CQL (Continuous Query Language), για την χωρική επεξεργασία των θέσεων των κινούμενων οχημάτων και την ανανέωση διάφορων σημαντικών στατιστικών στοιχείων (π.χ. οχήματα ανά οδικό άξονα, ταχύτητα ανά οδικό άξονα και χρονική περίοδο, κλπ.). Τα συγκεκριμένα ερωτήματα προσδιορίζουν παράθυρα, ώστε να

απομονώνουν κάθε φορά τις πιο πρόσφατες μετρήσεις (π.χ. στίγματα θέσεων των οχημάτων κατά το τελευταίο ημίωρο). Επειδή όμως απώτερος στόχος της εφαρμογής είναι η επιβολή ενός τέλους κυκλοφοριακής συμμόρφωσης στα οχήματα για την χρήση των οδικών αξόνων, ανάλογα την χρονική περίοδο που διέρχονται από αυτούς, η χρήση ενός απλού μαθηματικού τύπου, για τον υπολογισμό αυτού του επιβαλλόμενου τέλους συμμόρφωσης (ποσό ανά όχημα) ανάλογα με τις συνθήκες κυκλοφορίας των οχημάτων στο οδικό δίκτυο της πόλης, εφαρμόζεται χρησιμοποιώντας απλούς κανόνες απόφασης (χρονική περίοδος και επίπεδο συμμόρφωσης).

Στο TelegraphCQ, όλα τα ερωτήματα διαρκείας πρέπει να καθορίζουν ένα κατάλληλο παράθυρο πάνω από κάθε ρεύμα δεδομένων το οποίο συμμετέχει στην SQL διατύπωση τους. Με λίγα λόγια, ανά πάσα χρονική στιγμή, ένα παράθυρο επιστρέφει συνεχόμενα ένα πεπερασμένο σύνολο των πρόσφατων πλειάδων από το συνολικό ρεύμα. Αυτό το τμήμα των εισερχόμενων δεδομένων χρησιμοποιείται στη συνέχεια για την αξιολόγηση των αιτημάτων και την παραγωγή αποτελεσμάτων που αντιστοιχούν σε εκείνη την χρονική στιγμή. Μόνο τα κυλιόμενα παράθυρα υποστηρίζονται από το TelegraphCQ, έτσι ώστε να περιλαμβάνονται μόνο τα φρέσκα στοιχεία και τα παλαιότερα να απορρίπτονται από το περιεχόμενό τους. Η έκφραση ενός κυλιόμενου παραθύρου απαιτεί τον καθορισμό του χρονικού πεδίου εφαρμογής (RANGE BY «time»), το βήμα της μετακίνησης (SLIDE BY «time») και προαιρετικά τον χρόνο της έναρξης (START AT «time»).

Επισημαίνεται ότι όλα τα ερωτήματα διαρκείας τα οποία διατυπώνονται στο σύστημα TelegraphCQ, περικλείονται αναγκαστικά μέσα σε ένα κέρσορα ο οποίος είναι υπεύθυνος για την προώθηση των αποτελεσμάτων που παράγονται. Με την εντολή «DECLARE» δηλώνεται ο κέρσορας και με την εντολή «FETCH FORWARD» δηλώνονται πόσα αποτελέσματα επιστρέφει ο κέρσορας (βλ. παράρτημα Β).

Για την έκφραση σωστών χωρικών δηλώσεων στα ερωτήματα, γίνεται χρήση των τυπικών γεωμετρικών συναρτήσεων και των αντίστοιχων τύπων δεδομένων που προσφέρει η PostgreSQL. Τα ερωτήματα γενικά χωρίζονται σε χωρικά, χωροχρονικά και ανάλογα με τους τελεστές ή την δυναμικότητα των αντικειμένων εμπεριέχουν, σε στατικά ή κινούμενα. Τα χωροχρονικά ερωτήματα είναι ίδια με εκείνα των χωρικών, μόνο που ισχύουν για καθορισμένα χρονικά παράθυρα.

Επίσης, πρέπει να λαμβάνεται υπόψη ότι το ρεύμα λαμβάνει τις ανανεώσεις των θέσεων των οχημάτων κάθε 15 δευτερόλεπτα. Επομένως, τα ερωτήματα που διατυπώνονται αφορούν κυρίως την χωρική επεξεργασία των θέσεων των κινούμενων οχημάτων και την ανανέωση ορισμένων βασικών στατιστικών μεγεθών.

Παρακάτω περιγράφονται τα ερωτήματα που απαντώνται επιτυχώς από την εφαρμογή, μέσω των οποίων επιβεβαιώνεται η λειτουργικότητα της χωρικής βάσης και επιβεβαιώνεται ο έλεγχος καλής λειτουργίας της εφαρμογής. Αυτά τα βασικά ερωτήματα ενσωματώνονται μέσω χρήσης εργαλείων Java/Javascript στην ροή εργασιών της εφαρμογής (βλ. ενότητα 5.4). Επειδή το σύστημα αναφοράς είναι το WGS'84, οι συντεταγμένες των γεωμετρικών στοιχείων είναι

σε φ και λ, και όχι σε X και Y. Επιπλέον ερωτήματα που θα μπορούσαν να διατυπωθούν παρατίθενται στο παράρτημα Β.

Top10 vehicles charges query: «Βρες τα 10 οχήματα που έχουν την μεγαλύτερη χρέωση, τοποθετημένα κατά φθίνουσα σειρά».

```
SELECT VehicleID, MAX(ts) AS maxtime, SUM(Congestion_charge) AS
Congestion_charge
FROM traffic.Congestion_charges
GROUP BY VehicleID
ORDER BY SUM(Congestion_charge) DESC LIMIT 10;
```

Το ερώτημα, επιστρέφει την συνολική χρέωση του επιβαλλόμενου τέλους συμφόρησης στα κινούμενα οχήματα σε κάθε δεδομένη χρονική στιγμή, αλλά όχι για όλα τα οχήματα, μόνο για τα 10 οχήματα τα οποία παρουσιάζουν την μεγαλύτερη χρέωση μέχρι εκείνη την δεδομένη χρονική στιγμή. Συνεπώς εντοπίζονται τα 10 οχήματα που έχουν την μεγαλύτερη χρέωση, τοποθετούνται κατά φθίνουσα σειρά και παράλληλα εμφανίζεται η συνολική χρέωση καθενός από τα 10 οχήματα μαζί με την μέγιστη χρονική στιγμή. Επισημαίνεται ότι το χρονόσημο των κινούμενων οχημάτων μπορεί να αναφέρεται σε προηγούμενη χρονική στιγμή, αν την χρονική στιγμή που ζητούνται τα αποτελέσματα δεν έχει επιβληθεί νέα χρέωση ώστε να προκύψει νέα τιμή συνολικής χρέωσης, αλλά λόγω όμως της υπάρχουσας χρέωσης συγκαταλέγονται στα 10 πρώτα.

Το ερώτημα αυτό αποτελεί βασικό ερώτημα της εφαρμογής και τα αποτελέσματα του παρατίθενται στον χρήστη μέσω του διαδικτυακού εξυπηρετητή σε πινακοποιημένη μορφή (βλ. παράρτημα Δ). Ο πίνακας από τον οποίο προέρχονται τα στοιχεία είναι στατικός και όχι δυναμικός (ρεύμα). Τα αποτελέσματα του ερωτήματος πρέπει να πινακοποιούνται περιοδικά κάθε 15 δευτερόλεπτα για να απεικονίζονται στο διαδικτυακό περιβάλλον που έχει ως αποδέκτη τον χρήστη – πελάτη. Συνεπώς το ερώτημα πρέπει να καλείται μέσα από μία διαδικασία PHP και με την βοήθεια των εργαλείων της Javascript να επιστρέφει αποτελέσματα κάθε 15 δευτερόλεπτα, ώστε να είναι δυνατή η περιοδική ανανέωση των αποτελεσμάτων του στο διαδικτυακό περιβάλλον (βλ. ενότητα 5.4.1).

Top5 axes load query: «Βρες τους 5 άξονες που έχουν τον μεγαλύτερο φόρτο, τοποθετημένους κατά φθίνουσα σειρά».

```
SELECT R.AxisID, AVG(R.Load) AS Axis_load, WTIME(*)
FROM traffic.Roadaxes_load AS R [RANGE BY '2 MINUTES'
SLIDE BY '30 SECONDS'
START AT '2010-01-12 12:02:00']

GROUP BY R.AxisID
ORDER BY AVG(R.Load) DESC LIMIT 5;
```

Το ερώτημα, επιστρέφει ως αποτέλεσμα κάθε 30 δευτερόλεπτα, 5 οδικούς άξονες σε φθίνουσα σειρά, οι οποίοι έχουν τις μεγαλύτερες τιμές κυκλοφοριακού φόρτου, λαμβάνοντας υπόψη του ένα χρονικό παράθυρο 2 λεπτών, το οποίο ξεκινάει να εφαρμόζεται από την χρονική στιγμή 12:02:00 της 01-12-2010.

Επίσης το σύστημα TelegraphCQ προσφέρει την συναθροιστική συνάρτηση WTIME(*), η οποία επιστρέφει πάντα το πιο πρόσφατο χρονόσημο των περιεχομένων του τρέχοντος παράθυρου. Επισημαίνεται ότι ενώ το ερώτημα λειτουργεί κανονικά τελικώς δεν εκτελείται με αυτήν την διατύπωση στη ροή εργασιών, λόγω του προβλήματος που αναφέρεται στην ενότητα 5.4.2.

Top10 axes vehicles speeds query: «Βρες τους 10 άξονες που διαθέτουν την μεγαλύτερη ταχύτητα κίνησης οχημάτων, τοποθετημένους κατά φθίνουσα σειρά».

```
SELECT AxisID, avg(Speed) AS Average_speed, wtime(*) AS TS
FROM traffic.Trajectories [RANGE BY '1 MINUTE'
                           SLIDE BY '15 SECONDS'
                           START AT '2010-01-12 12:01:00']

GROUP BY AxisID
ORDER BY avg(Speed) DESC LIMIT 10);
```

Το ερώτημα, κάθε 15 δευτερόλεπτα, επιστρέφει τους οδικούς άξονες με την μεγαλύτερη μέση ταχύτητα διέλευσης στα κινούμενα οχήματα σε κάθε δεδομένη χρονική στιγμή, αλλά όχι για όλους τους άξονες, μόνο για τους 10 άξονες οι οποίοι παρουσιάζουν την μεγαλύτερη μέση ταχύτητα διέλευσης μέχρι εκείνη την δεδομένη χρονική στιγμή, λαμβάνοντας υπόψη του ένα χρονικό παράθυρο ενός λεπτού, το οποίο ξεκινάει να εφαρμόζεται από την χρονική στιγμή 12:01:00 της 01-12-2010. Συνεπώς εντοπίζονται οι 10 οδικοί άξονες που διαθέτουν την μεγαλύτερη μέση ταχύτητα διέλευσης στα κινούμενα οχήματα, τοποθετούνται κατά φθίνουσα σειρά και παράλληλα εμφανίζεται η μέση ταχύτητα κίνησης καθενός από τους 10 άξονες μαζί με το αντίστοιχο χρονόσημο.

Επίσης το σύστημα TelegraphCQ προσφέρει την συναθροιστική συνάρτηση WTIME(*), η οποία επιστρέφει πάντα το πιο πρόσφατο χρονόσημο των περιεχομένων του τρέχοντος παράθυρου.

Το ερώτημα αυτό αποτελεί βασικό ερώτημα της εφαρμογής και τα αποτελέσματα του παρατίθενται στον χρήστη μέσω του διαδικτυακού εξυπηρετητή σε πινακοποιημένη μορφή (βλ. παράρτημα Δ). Ο πίνακας από τον οποίο προέρχονται τα στοιχεία είναι δυναμικός (ρεύμα). Τα αποτελέσματα του ερωτήματος πρέπει να πινακοποιούνται περιοδικά κάθε 15 δευτερόλεπτα για να απεικονίζονται στο διαδικτυακό περιβάλλον που έχει ως αποδέκτη τον χρήστη – πελάτη. Συνεπώς το ερώτημα πρέπει να καλείται μέσα από μία διαδικασία Java να επιστρέφει αποτελέσματα κάθε 15 δευτερόλεπτα, ώστε να είναι δυνατή η περιοδική ανανέωση των αποτελεσμάτων του στο διαδικτυακό περιβάλλον με την βοήθεια των εργαλείων της Javascript (βλ. ενότητα 5.4.1).

5.3.4 Περιγραφή συναρτήσεων PL/pgSQL της εφαρμογής

Με την βοήθεια της PL/pgSQL, συντίθενται μόνιμα χωροχρονικά ερωτήματα διαρκείας με κατάλληλα χρονικά παράθυρα, τα οποία συνεχώς εκτελούνται από το σύστημα TelegraphCQ, το οποίο επιστρέφει ως έξοδο σε συγκεκριμένη χρονική στιγμή, τα τρέχοντα αποτελέσματα των ερωτημάτων που τίθενται.

Οι τρεις βασικές συναρτήσεις που δημιουργήθηκαν σε γλώσσα προγραμματισμού PL/pgSQL, είναι οι εξής:

Συνάρτηση createTrajectory

Αυτή η συνάρτηση για να λειτουργήσει δέχεται ως δεδομένο εισόδου μία συγκεκριμένη χρονική στιγμή. Κατόπιν δομείται το ακόλουθο ερώτημα διαρκείας, με την εφαρμογή ενός κέρσορα (βλ. παράρτημα Γ), για την προσπέλαση των δεδομένων από την βάση:

```
SELECT L.vehicleID, L.position, L.SegmentID, L.tcqtime
FROM traffic.Locations AS L [RANGE BY '30 SECONDS'
                             SLIDE BY '15 SECONDS']
WHERE L.tcqtime <= ts
AND L.tcqtime >= ts - interval '15 seconds'
ORDER BY L.vehicleID, L.tcqtime;
```

Για όλα τα κινούμενα οχήματα, κάθε 15 δευτερόλεπτα, το ερώτημα επιστρέφει την θέση που έχουν και το οδικό τμήμα στο οποίο κινούνται σε κάθε δεδομένη χρονική στιγμή, καθώς και το χρονόσημο, λαμβάνοντας υπόψη του ένα χρονικό παράθυρο 30 δευτερολέπτων, όσο είναι και το χρονικό διάστημα εκπομπής δύο θέσεων των κινούμενων οχημάτων προς το σύστημα.

Εφόσον υπάρχει διαθέσιμη η προηγούμενη και η τωρινή θέση του κινούμενου οχήματος, υπολογίζεται το αντίστοιχο γραμμικό τμήμα της τροχιάς την οποία διαγράφει κάθε κινούμενο όχημα σε κάθε δεδομένη χρονική στιγμή, λαμβάνοντας την συγκεκριμένη χρονική στιγμή από το χρονόσημο που υπάρχει στις θέσεις των οχημάτων από το εισερχόμενο ρεύμα δεδομένων traffic.Locations. Εφόσον υπολογιστεί η μετατόπιση του οχήματος (σε μέτρα) από την μία χρονική στιγμή στην άλλη, στην συνέχεια υπολογίζεται αν το όχημα κινείται στον ίδιο οδικό άξονα ή έχει μεταβεί σε κάποιο καινούργιο. Ανάλογα με την απάντηση δίνεται και η αντίστοιχη τιμή για την επιβολή χρέωσης ή μη επιβολής στο κινούμενο όχημα για την δεδομένη χρονική στιγμή.

Τέλος, αφού είναι γνωστή η μετατόπιση, υπολογίζεται από τον γνωστό μαθηματικό τύπο η ταχύτητα κίνησης του οχήματος (σε km/h) για την δεδομένη χρονική στιγμή.

Τα αποτελέσματα που εξάγονται από την λειτουργία της συνάρτησης, χρησιμεύουν στην τροφοδότηση του παράγωγου ρεύματος traffic.Trajectories με τα αναμενόμενα δεδομένα, δηλαδή τα αποτελέσματα της συνάρτησης εξάγονται αμέσως στο αντίστοιχο ρεύμα που αφορά τις τροχιές των οχημάτων.

Συνάρτηση computeLoad

Αυτή η συνάρτηση για να λειτουργήσει δέχεται ως δεδομένο εισόδου μία συγκεκριμένη χρονική στιγμή. Κατόπιν δομείται το ακόλουθο ερώτημα διαρκείας, με την εφαρμογή ενός κέρσορα (βλ. παράρτημα Γ), για την προσπέλαση των δεδομένων από την βάση:

```

SELECT T.AxisID, COUNT(T.VehicleID) AS Vehicles_count,
A.Load_coefficient
FROM traffic.Trajectories AS T [RANGE BY '15 SECONDS'
                               SLIDE BY '15 SECONDS'],
     traffic.Road_axes A
WHERE T.tcqtime = ts
AND T.AxisID = A.AxisID
GROUP BY T.AxisID, A.Load_coefficient
HAVING COUNT(T.VehicleID) > 0;

```

Για όλους τους οδικούς άξονες, κάθε 15 δευτερόλεπτα, το ερώτημα επιστρέφει τον αριθμό των οχημάτων που κινούνται σε κάθε δεδομένη χρονική στιγμή, καθώς και τον συντελεστή του κυκλοφοριακού φόρτου που αναλογεί σε κάθε οδικό άξονα (χωρητικότητα οδοστρώματος κάθε άξονα ανάλογα με το μήκος και τον αριθμό λωρίδων που διαθέτει), λαμβάνοντας υπόψη του ένα χρονικό παράθυρο 15 δευτερολέπτων, όσο είναι και το χρονικό διάστημα εκπομπής των δέσεων των κινούμενων οχημάτων προς το σύστημα.

Εφόσον υπάρχει διαθέσιμος ο αριθμός των οχημάτων που κινούνται πάνω σε κάθε οδικό άξονα, υπολογίζεται ο αντίστοιχος κυκλοφοριακός φόρτος κάθε οδικού άξονα που προκαλούν τα κινούμενα οχήματα σε κάθε δεδομένη χρονική στιγμή, λαμβάνοντας δεδομένα για αυτά από το παράγωγο ρεύμα δεδομένων traffic.Trajectories και από τον στατικό πίνακα traffic.Road_axes.

Ο κυκλοφοριακός φόρτος (A) υπολογίζεται με βάση τον εξής απλό μαθηματικό τύπο: $A = B \times \Gamma$, όπου B ο αριθμός των κινούμενων οχημάτων σε κάθε άξονα και Γ ο αντίστοιχος συντελεστής του κυκλοφοριακού φόρτου που του αναλογεί. Εφόσον υπολογιστεί ο κυκλοφοριακός φόρτος (σε Οχήματα/Km/Λωρίδα) κάθε οδικού άξονα για μία δεδομένη χρονική στιγμή, στην συνέχεια υπολογίζεται το ποσό της χρέωσης (Δ) που πρέπει να επιβληθεί σε κάθε διερχόμενο όχημα των οδικών αξόνων. Ο υπολογισμός της χρέωσης (τέλος κυκλοφοριακής συμφόρησης) που επιβάλλεται για την χρήση κάθε οδικού άξονα από τα κινούμενα οχήματα είναι κυμαινόμενη και προκύπτει ανάλογα την χρονική περίοδο που εμπίπτει το χρονόσημο των οχημάτων και ανάλογα με τα όρια στα οποία εμπίπτει ο κυκλοφοριακός φόρτος του οδικού άξονα, δηλαδή τον οριζόμενο μέγιστο και ελάχιστο κυκλοφοριακό φόρτο για την δεδομένη χρονική περίοδο (βλ. πίνακες 5.2 και 5.3).

Τέλος, υπολογίζεται και η αντίστοιχη κλάση της κατηγορίας που ανήκει κάθε οδικός άξονας, με βάση τον κυκλοφοριακό φόρτο που του αναλογεί για την δεδομένη χρονική στιγμή.

Τα αποτελέσματα που εξάγονται από την λειτουργία της συνάρτησης, χρησιμεύουν στην τροφοδότηση του παράγωγου ρεύματος traffic.Roadaxes_load με τα αναμενόμενα δεδομένα, δηλαδή τα αποτελέσματα της συνάρτησης εξάγονται αμέσως στο αντίστοιχο ρεύμα που αφορά τον κυκλοφοριακό φόρτο σε κάθε άξονα του οδικού δικτύου.

Συνάρτηση computeCharge

Αυτή η συνάρτηση για να λειτουργήσει δέχεται ως δεδομένο εισόδου μία συγκεκριμένη χρονική στιγμή. Κατόπιν δομείται το ακόλουθο ερώτημα διαρκείας,

με την εφαρμογή ενός κέρσορα (βλ. παράρτημα Γ), για την προσπέλαση των δεδομένων από την βάση:

```
SELECT T.VehicleID, L.Charge
FROM (SELECT * FROM traffic.Roadaxes_load WHERE tcqtime = ts) AS
L, (SELECT * FROM traffic.Trajectories WHERE tcqtime = ts) AS T
WHERE T.AxisID = L.AxisID
AND T.Apply_charge = TRUE;
```

Για όλα τα κινούμενα οχήματα, για κάθε χρονική στιγμή, το ερώτημα επιστρέφει την χρέωση που επιβάλλεται σε κάθε κινούμενο όχημα για κάθε οδικό άξονα από τον οποίο διέρχεται και χρησιμοποιεί στο οδικό δίκτυο, λαμβάνοντας δεδομένα για κοινές δεδομένες χρονικές στιγμές, από τα παραγόμενα ρεύματα δεδομένων `traffic.Trajectories` και `traffic.Roadaxes_load`.

Συνεπώς μέσω της συνάρτησης καταχωρείται η χρέωση (εφόσον δεν είναι μηδενική) κάθε κινούμενου οχήματος, ανάλογα την χρονική περίοδο και τους οδικούς άξονες που χρησιμοποιεί για την κίνηση του στο οδικό δίκτυο, δηλαδή το αντίστοιχο τέλος κυκλοφοριακής συμφόρησης κάθε οχήματος για κάθε δεδομένη χρονική στιγμή. Τα αποτελέσματα που εξάγονται από την λειτουργία της συνάρτησης, χρησιμεύουν στην τροφοδότηση του παράγωγου ρεύματος `traffic.Congestion_charges` με τα αναμενόμενα δεδομένα.

Τα αποτελέσματα της συνάρτησης εξάγονται αμέσως στο αντίστοιχο ρεύμα που αφορά την χρέωση των κινούμενων οχημάτων με το ανάλογο τέλος συμφόρησης που αντιστοιχεί σε κάθε άξονα του οδικού δικτύου.

Επίσης δημιουργήθηκε μία βοηθητική συνάρτηση `processData` η οποία εκτελεί για μία συγκεκριμένη χρονική στιγμή που δίνεται ως δεδομένο εισόδου, τις τρεις συναρτήσεις `createTrajectory`, `computeLoad` και `computeCharge`, για την ίδια χρονική στιγμή ως δεδομένο εισόδου και σε αυτές. Στόχος της συνάρτησης είναι η ενσωμάτωση αυτών των τριών συναρτήσεων στην διαδικασία που εκτελεί ο κώδικας Java (βλ. παράρτημα Γ).

5.4 Χαρτογραφική απεικόνιση μέσω διαδικτυακού εξυπηρετητή

Ζητούμενο της εφαρμογής είναι η χαρτογραφική απεικόνιση του κυκλοφοριακού φόρτου των οδικών αξόνων, εξαιτίας της κίνησης των οχημάτων στο οδικό δίκτυο της Αθήνας. Σκοπός όμως, είναι η χαρτογραφική απεικόνιση να γίνεται σε διαδικτυακό περιβάλλον, επάνω σε χάρτη αξιοποιώντας το ελεύθερο περιβάλλον που προσφέρεται μέσω GoogleMaps. Επίσης, ζητούμενο είναι να απεικονίζονται υπό μορφή πίνακα η μέση ταχύτητα κίνησης των οχημάτων σε κάθε οδικό άξονα αλλά και οι χρεώσεις που επιβάλλονται σε κάθε χρονική στιγμή στα κινούμενα οχήματα του δικτύου.

Επισημαίνεται ότι ο διαδικτυακός εξυπηρετητής δεν είναι προσιτός από τον χρήστη, παρά μόνο οπτικά, αφήνοντας τον χρήστη απλά παθητικό χωρίς να μπορεί να επεμβεί με οποιοδήποτε τρόπο σε αυτόν.

Στην συνέχεια της ενότητας περιγράφονται καθεμιά από τις λειτουργίες που απαιτείται να εκτελεί ο εξυπηρετητής δεδομένων, ώστε να γίνεται με επιτυχία η απεικόνιση των ζητούμενων αποτελεσμάτων.

5.4.1 Παραγωγή των αποτελεσμάτων

Το πρώτο βήμα της διαδικασίας που εκτελείται στον διαδικτυακό εξυπηρετητή, είναι η παραγωγή των επιθυμητών αποτελεσμάτων που παράγονται από την λειτουργία του συστήματος TelegraphCQ, τα οποία είναι εκείνα που απεικονίζονται στους χρήστες της εφαρμογής.

Συνεπώς, στο βήμα αυτό καθορίζονται δύο βασικά ερωτήματα:

- A. Ποια είναι τα αποτελέσματα που πρέπει να αποστέλλονται στον διαδικτυακό εξυπηρετητή προς απεικόνιση;
- B. Ποιος είναι ο τρόπος με τον οποίο αυτά τα αποτελέσματα μεταδίδονται από τον εξυπηρετητή δεδομένων στον διαδικτυακό εξυπηρετητή;

Όσον αφορά το πρώτο ερώτημα, σκοπός της εφαρμογής είναι η απεικόνιση του κυκλοφοριακού φόρτου σε πραγματικό χρόνο και ταυτόχρονα η επιβολή ενός συμβολικού τέλους συμφόρησης στα κινούμενα οχήματα στο οδικό της Αθήνας, Συνεπώς από το σύστημα παράγονται τα εξής αποτελέσματα (βλ. ενότητα 5.3.3), τα οποία τελικά είναι και εκείνα που απεικονίζονται:

- Ο κυκλοφοριακός φόρτος για κάθε οδικό άξονα του δικτύου, σε κάθε δεδομένη χρονική στιγμή.
- Η ταχύτητα κίνησης των οχημάτων για τους 10 πιο βεβαρημένους οδικό άξονα του δικτύου, σε κάθε δεδομένη χρονική στιγμή.
- Η χρέωση των 10 πιο χρεωμένων οχημάτων του τέλους συμφόρησης για κάθε οδικό άξονα που χρησιμοποιούν και διέρχονται για την μετακίνησή τους, σε κάθε δεδομένη χρονική στιγμή.

Όσον αφορά το δεύτερο ερώτημα, πρόκειται για την διαδικασία που ακολουθείται για την προσπέλαση των δεδομένων, από την βάση του εξυπηρετητή δεδομένων μέσω των ερωτημάτων διαρκείας. Η εξαγωγή των αποτελεσμάτων από την βάση γίνεται μέσω ρουτινών PHP ή Java. Συγκεκριμένα, τα αποτελέσματα τα οποία αφορούν (α) τον κυκλοφοριακό φόρτο για κάθε οδικό άξονα και (β) την χρέωση των οχημάτων επιβάλλοντας ένα καθορισμένο τέλος συμφόρησης σε αυτά, ανακτώνται μέσω ρουτινών PHP, οι οποίες ενεργοποιούνται από τον διαδικτυακό εξυπηρετητή αλλά εκτελούνται στον εξυπηρετητή δεδομένων και συγκεκριμένα στο TelegraphCQ.

Αντίθετα, τα αποτελέσματα που αφορούν τη μέση ταχύτητα κίνησης των οχημάτων για κάθε οδικό άξονα, αποκτώνται μέσω ρουτίνας Java, η οποία βρίσκεται και ενεργοποιείται από τον εξυπηρετητή δεδομένων αλλά αποστέλλονται στον διαδικτυακό εξυπηρετητή για απεικόνιση στην ιστοσελίδα.

Ανάκτηση κυκλοφοριακού φόρτου οδικών αξόνων

Αυτή η διαδικασία PHP αρχικά πραγματοποιεί την απαραίτητη σύνδεση με την βάση δεδομένων. Κατόπιν δομείται το ακόλουθο ερώτημα διαρκείας, με την εφαρμογή ενός κέρσορα (βλ. παράρτημα Β), για την προσπέλαση των δεδομένων από την βάση:

```
SELECT R.Segment, L.AxisID, L.Load_class
FROM traffic.Road_segments AS R, traffic.Roadaxes_load AS L
[RANGE BY '15 SECONDS' SLIDE BY '15 SECONDS']
WHERE R.AxisID=L.AxisID;
```

Για όλους τους οδικούς άξονες, κάθε 15 δευτερόλεπτα, το ερώτημα επιστρέφει την κλάση της κατηγορίας του φόρτου που έχουν σε κάθε δεδομένη χρονική στιγμή, λαμβάνοντας υπόψη του ένα χρονικό παράθυρο 15 δευτερολέπτων, όσο είναι και το χρονικό διάστημα εκπομπής των θέσεων των κινούμενων οχημάτων προς το σύστημα.

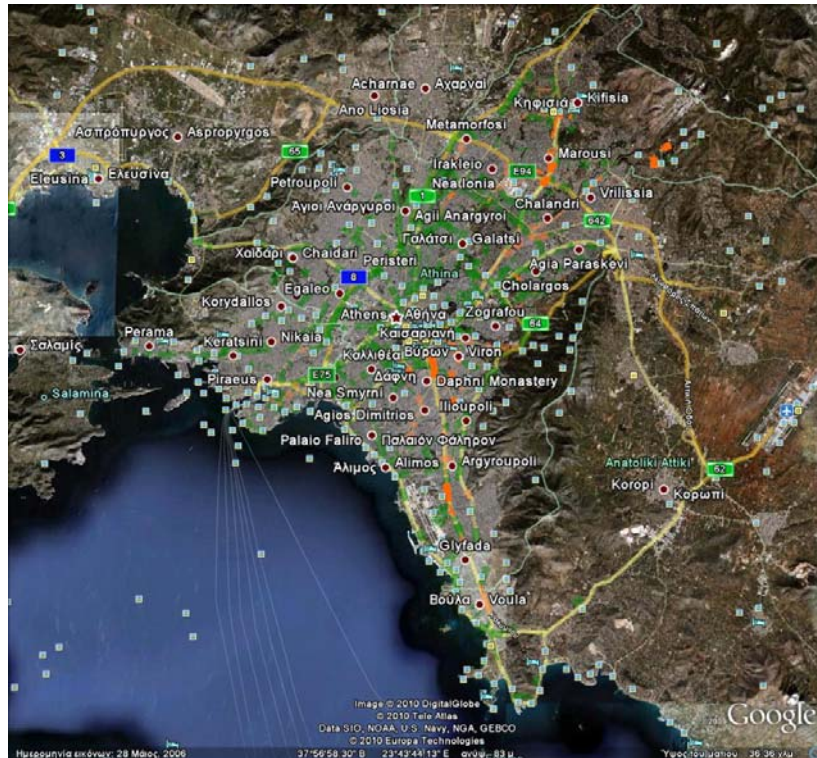
Στην συνέχεια γίνεται η δόμηση ενός KML αρχείου σύμφωνα με τα πρότυπα των KML αρχείων και εισάγονται ανάλογα με την κλάση της κατηγορίας που ανήκει κάθε οδικός άξονας του δικτύου, διαφορετικό στυλ μορφοποίησης, δηλαδή ξεχωριστό χρώμα και πάχος γραμμής, όπως φαίνεται στον ακόλουθο πίνακα 5.4.

Τέλος, κάθε 15 δευτερόλεπτα εκτελείται η εξαγωγή των αποτελεσμάτων, σύμφωνα με το ερώτημα διαρκείας, στο KML αρχείο όπου και αποθηκεύονται. Επισημαίνεται ότι το KML αρχείο αποθηκεύεται στον διαδικτυακό εξυπηρετητή και αναμένει να κληθεί προς απεικόνιση στην ιστοσελίδα.

Στο παράρτημα Γ, υπάρχει ο κώδικας PHP με την ονομασία `Traffic_lines.php` και στο παράρτημα Δ υπάρχει ένα τμήμα των αποτελεσμάτων τα οποία παράγονται από το συγκεκριμένο PHP αρχείο και είναι αποθηκευμένα σε κώδικα υπό μορφή αρχείου KML. Στο παρακάτω σχήμα 5.3, απεικονίζεται γραφικά μέσω της εφαρμογής GoogleEarth το αρχείο KML, το οποίο έχει παραχθεί για κάποια δεδομένη χρονική στιγμή.

| Κλάση κατηγορίας | Απόχρωση | Πάχος γραμμής |
|------------------|-----------------|---------------|
| A | Πράσινο | 9 |
| B | Κίτρινο ανοιχτό | 9 |
| C | Κίτρινο | 9 |
| D | Πορτοκαλί | 9 |
| E | Κόκκινο | 9 |
| F | Κόκκινο σκούρο | 9 |

Πίνακας 5.4: Στυλ μορφοποίησης απεικόνισης των οδικών αξόνων.



Σχήμα 5.3: Απεικόνιση KML αρχείου μέσω GoogleEarth.

Ανάκτηση χρέωσης των οχημάτων

Αυτή η διαδικασία PHP αρχικά πραγματοποιεί την απαραίτητη σύνδεση με την βάση δεδομένων. Κατόπιν δομείται το λογικό ερώτημα *Top10 vehicles charges query* (βλ. ενότητα 5.3.3), για την προσπέλαση των δεδομένων από την βάση.

Στην συνέχεια γίνεται η δόμηση ενός XML αρχείου σύμφωνα με τα πρότυπα των XML αρχείων Τέλος, εκτελείται η εξαγωγή των αποτελεσμάτων, σύμφωνα με το ερώτημα, στο XML αρχείο όπου και αποθηκεύονται. Επισημαίνεται, ότι το XML αρχείο αποθηκεύεται στον διαδικτυακό εξυπηρετητή και αναμένει να κληθεί προς απεικόνιση στην ιστοσελίδα.

Στο παράρτημα Γ, υπάρχει ο κώδικας PHP με την ονομασία *top10_vehiclescharges.php* και στο παράρτημα Δ υπάρχουν ενδεικτικά τα αποτελέσματα τα οποία παράγονται από το συγκεκριμένο PHP αρχείο και είναι αποθηκευμένα σε κώδικα υπό μορφή αρχείου XML.

Ανάκτηση μέσης ταχύτητας κίνησης των οχημάτων στους οδικούς άξονες

Αυτή η διαδικασία σε Java αρχικά πραγματοποιεί την απαραίτητη σύνδεση με την βάση δεδομένων. Κατόπιν δομείται το ερώτημα διαρκείας *Top10 axes vehicles speeds query* (βλ. ενότητα 5.3.3), με την εφαρμογή ενός κέρσορα (βλ. παράρτημα Β), για την προσπέλαση των δεδομένων από την βάση.

Στην συνέχεια γίνεται η δόμηση ενός XML αρχείου σύμφωνα με τα πρότυπα των XML αρχείων Τέλος, εκτελείται η εξαγωγή των αποτελεσμάτων,

σύμφωνα με το ερώτημα, στο XML αρχείο όπου και αποθηκεύονται. Επισημαίνεται, ότι το XML αρχείο παράγεται στον εξυπηρετητή δεδομένων και αποστέλλεται στον διαδικτυακό εξυπηρετητή προς απεικόνιση στην ιστοσελίδα. Στο παράρτημα Γ, υπάρχει ο κώδικας Java με την ονομασία TrafficProcessing.java, στον οποίο περιλαμβάνεται σε ένα μέρος η διαδικασία που αναλύθηκε καθώς και η παραγωγή του XML αρχείου, ενώ στο παράρτημα Δ υπάρχουν ενδεικτικά τα αποτελέσματα τα οποία παράγονται από το συγκεκριμένο Java αρχείο και είναι αποθηκευμένα σε κώδικα υπό μορφή αρχείου XML.

5.4.2 Περιοδική ανανέωση και απεικόνιση των αποτελεσμάτων

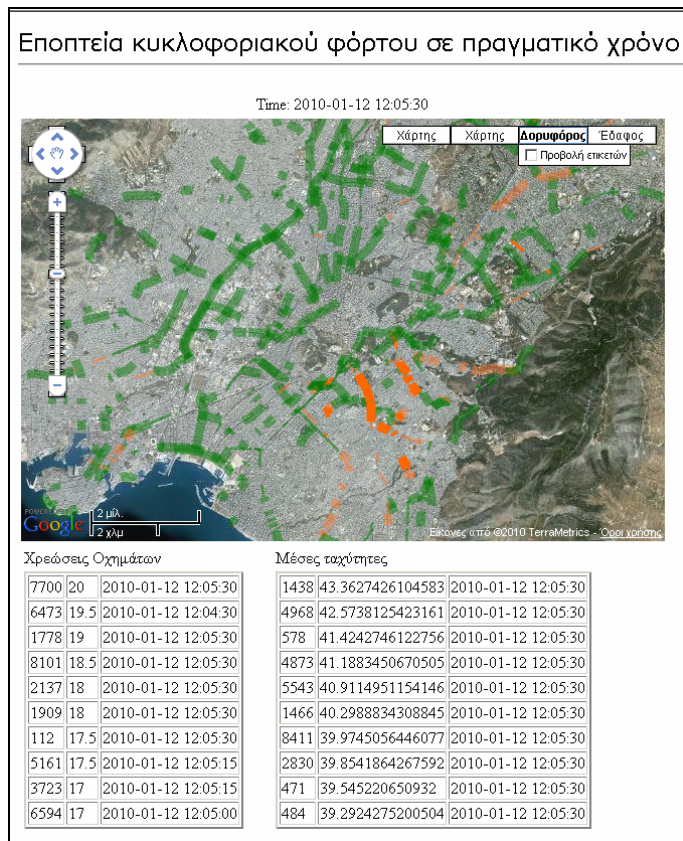
Ζητούμενο είναι η περιοδική ανανέωση όλων των αποτελεσμάτων, τα οποία απεικονίζονται στον διαδικτυακό εξυπηρετητή και έχουν ως αποδέκτη τον κάθε χρήστη της εφαρμογής.

Όσον αφορά τα αποτελέσματα για τον κυκλοφοριακό φόρτο των οδικών αξόνων και την χρέωση των κινούμενων οχημάτων, υπεύθυνη για αυτήν την περιοδική ανανέωση είναι η Javascript, η οποία εκτελείται από τον διαδικτυακό εξυπηρετητή και μέσω της οποίας οι ρουτίνες PHP εκτελούνται περιοδικά κάθε 15 δευτερόλεπτα. Ο αντίστοιχος κώδικας της Javascript βρίσκεται ενσωματωμένος στον κώδικα HTML, μέσω του οποίου γίνεται η απεικόνιση των αποτελεσμάτων στην ιστοσελίδα (βλ. παράρτημα Γ).

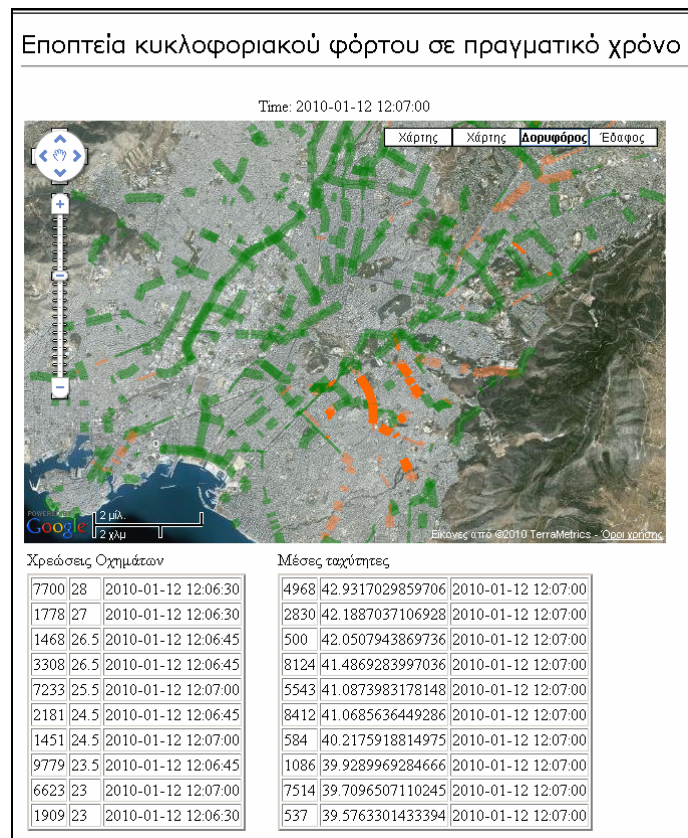
Αντίστοιχα, η Java αναλαμβάνει την περιοδική ανανέωση (κάθε 15 δευτερόλεπτα) των αποτελεσμάτων για την μέση ταχύτητα κίνησης των οχημάτων στους οδικούς άξονες (βλ. ενότητα 5.4.1) και όχι η HTML. Αυτό συμβαίνει λόγω της αρχιτεκτονικής της ροής εργασιών της εφαρμογής. Όταν δοκιμάστηκε τα αποτελέσματα να εξάγονται μέσω ρουτίνας PHP και η περιοδική ανανέωση τους να γίνεται από την Javascript, η οποία είναι ενσωματωμένη μέσα στο κώδικα της HTML, το σύστημα TelegraphCQ δεν μπορούσε να ανταποκριθεί και διακοπτόταν η εισροή των δεδομένων θέσης των οχημάτων.

Σημαντικό επίσης, είναι να αναφερθεί ότι οι λειτουργίες που εκτελεί ο εξυπηρετητής δεδομένων (βλ. ενότητα 5.3), εφαρμόζονται περιοδικά κάθε 15 δευτερόλεπτα μέσω του κώδικα Java, ο οποίος χρησιμοποιείται ταυτόχρονα και για την περιοδική παραγωγή των αποτελεσμάτων που αφορούν την μέση ταχύτητα κίνησης των οχημάτων στους οδικούς άξονες (βλ. παράρτημα Γ),

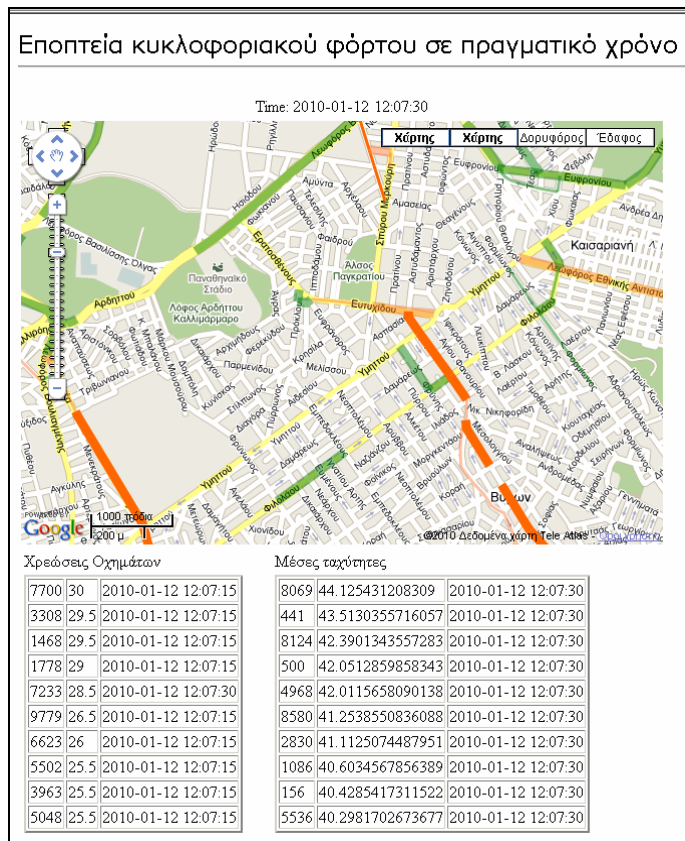
Τέλος, όσον αφορά την απεικόνιση των αποτελεσμάτων, αυτή γίνεται σε ιστοσελίδα του διαδικτύου περιοδικά κάθε 15 δευτερόλεπτα, είτε προκύπτουν νέα αποτελέσματα προς απεικόνιση είτε όχι. Η απεικόνιση του κυκλοφοριακού φόρτου γίνεται χαρτογραφικά πάνω σε χάρτη περιβάλλοντος GoogleMaps, όπου τα εικονιζόμενα στοιχεία είναι γραμμικά, με διαφορετικά χρώματα και διαφορετικό πάχος γραμμών, ενώ η απεικόνιση των αποτελεσμάτων των 10 οδικών αξόνων που έχουν την μεγαλύτερη μέση ταχύτητα κίνησης των οχημάτων και των 10 κινούμενων οχημάτων με την μεγαλύτερη χρέωση του επιβαλλόμενου τέλους συμφόρησης γίνεται υπό μορφή πίνακα. Στην συνέχεια παρατίθενται ορισμένα στιγμιότυπα της εφαρμογής σε τέσσερις διαφορετικές χρονικές στιγμές, ενώ η απεικόνιση πραγματοποιείται και σε διαφορετικά υπόβαθρα του χάρτη που προσφέρονται μέσω GoogleMaps (χάρτης, δορυφόρος, έδαφος):



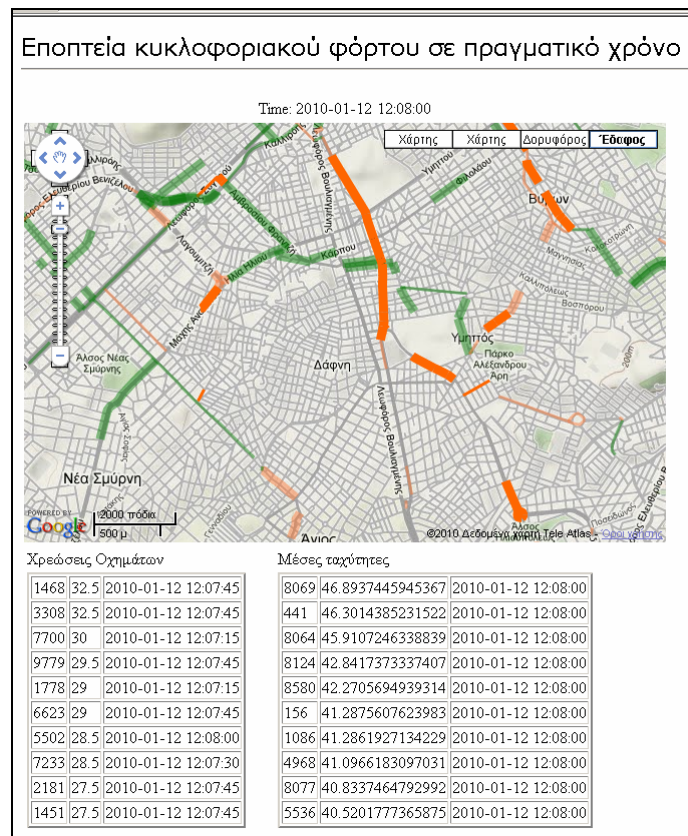
Σχήμα 5.4: Στιγμιότυπο εφαρμογής για χρονική στιγμή '2010-01-12 12:05:30'.



Σχήμα 5.5: Στιγμιότυπο εφαρμογής για χρονική στιγμή '2010-01-12 12:07:00'.



Σχήμα 5.6: Στιγμιότυπο εφαρμογής για χρονική στιγμή '2010-01-12 12:07:30'.



Σχήμα 5.7: Στιγμιότυπο εφαρμογής για χρονική στιγμή '2010-01-12 12:08:00'.

5.5 Συγγενικές εφαρμογές

Υπάρχουν ορισμένες συγγενικές εφαρμογές στις οποίες αξίζει να γίνει μία σύντομη αναφορά και περιγραφή του τρόπου λειτουργίας τους και των υπηρεσιών που προσφέρουν.

Δύο ελληνικοί διαδικτυακοί τόποι, η Ναυτεμπορική και η Καθημερινή, παρέχουν μέσω εφάμιλλων εφαρμογών πληροφορίες σχετικά με την οδική κυκλοφορία και τις μετακινήσεις των οχημάτων στον οδικό ιστό της Αθήνας [www.NAFT, www.KATH]. Και οι δύο διαδικτυακές εφαρμογές απεικονίζουν τον τρέχοντα κυκλοφοριακό φόρτο στους οδικούς άξονες της πόλης για κάθε χρονική στιγμή, καθώς και τα αντίστοιχα οδικά συμβάντα που λαμβάνουν χώρα στον οδικό ιστό. Ο επισκέπτης έχει τη δυνατότητα να αναζητήσει πληροφόρηση σχετικά με πάσης φύσεως συμβάντα, όπως τροχαία ατυχήματα, πορείες, αθλητικές εκδηλώσεις, εργασίες οδού, μποτιλιάρισμα, κ.ά. σε δρόμους ενδιαφέροντός του. Επίσης, μπορεί να λάβει δυναμική πληροφόρηση για την κυκλοφοριακή κατάσταση για επιλεγμένες διαδρομές, καθώς και πληροφορίες σχετικά με το χρόνο διαδρομής και την μέση ταχύτητα κίνησης.

Όσον αφορά την διαδικτυακή εφαρμογή της Καθημερινής, παρέχονται ορισμένες πρόσθετες δυνατότητες από τη συγκεκριμένη υπηρεσία: (α) η λειτουργία της εύρεσης διαδρομών, (β) η αναζήτηση διεύθυνσης και (γ) η αναζήτηση σημείων ενδιαφέροντος. Η λειτουργία της εύρεσης διαδρομών επιτρέπει στον επισκέπτη την αναζήτηση οποιασδήποτε διαδρομής στην Ελλάδα για επιλεγμένα σημεία αφετηρίας και προορισμού (σε επίπεδο διεύθυνσης / δρόμου). Επίσης, ο χρήστης έχει την δυνατότητα να αναζητά διευθύνσεις και σημεία ενδιαφέροντος, ενώ παράλληλα η υπηρεσία επιτρέπει την εύρεση κοντινών σημείων ενδιαφέροντος (πρατήρια βενζίνης, ξενοδοχεία, εστιατόρια, φαρμακεία, κ.α.) κατά μήκος των αναζητηθέντων διαδρομών, καθώς επίσης και γύρω από συγκεκριμένα σημεία ή περιοχές.

Τα δεδομένα που χρησιμοποιούνται και από τις δύο διαδικτυακές εφαρμογές, προέρχονται από το Υπουργείο Μεταφορών και Επικοινωνιών της Ελλάδος. Η συλλογή αυτών των κυκλοφοριακών δεδομένων γίνεται δυναμικά, με την βοήθεια αυτόματων μετρητών στους οδικούς άξονες της πόλης και μέσω των καμερών διαχείρισης κυκλοφορίας.

Μία άλλη συγγενική εφαρμογή είναι η ITS (Intelligent Transportation Systems), η οποία αναπτύχθηκε στο Βασιλικό Ινστιτούτο Τεχνολογίας της Στοκχόλμης, στην Σουηδία [KRG10]. Πρόκειται για εφαρμογή διαχείρισης και επεξεργασίας κυκλοφοριακής πληροφορίας σε πραγματικό χρόνο. Τα ρεύματα δεδομένων προέρχονται από ενσωματωμένους δέκτες GPS σε φορτηγά και ταξί, τα οποία κινούνται στην πόλη της Στοκχόλμης.

Σκοπός της εφαρμογής αυτής είναι ο υπολογισμός της ταχύτητας και της κυκλοφοριακής ροής για όλες τις διαφορετικές οδούς στην πόλη (διαρκώς ανανεωμένες), ο όγκος της κίνησης σε διάφορες περιοχές της πόλης, η εκτίμηση του χρόνου διαδρομής μεταξύ διαφορετικών σημείων της πόλης, η εύρεση των συντομότερων διαδρομών δυναμικά σύμφωνα με τις τρέχουσες κυκλοφοριακές συνθήκες, κ.λπ. Το σύστημα που έχει αναπτυχθεί πάνω σε πλατφόρμα της IBM, μπορεί να χειριστεί μεγάλους όγκους εισερχόμενων δεδομένων (περισσότερα από

120000 εισερχόμενα στήγματα ανά δευτερόλεπτο), σε συνδυασμό με έναν χάρτη που περιέχει πάνω από 600.000 γραμμικά οδικά τμήματα. Επίσης, έχει την δυνατότητα να παραγάγει συνεχώς διαφορετικά είδη στατιστικών κυκλοφορίας και απαντήσεων σε ερωτήματα χρηστών.

ΚΕΦΑΛΑΙΟ 6

ΣΥΜΠΕΡΑΣΜΑΤΑ & ΠΡΟΤΑΣΕΙΣ

6.1 Συμπεράσματα

Στην παρούσα μεταπτυχιακή εργασία, επιχειρήθηκε η δημιουργία μιας χαρτογραφικής εφαρμογής, η οποία να λειτουργεί και να επεξεργάζεται δεδομένα σημειακών κινούμενων αντικειμένων σε πραγματικό χρόνο. Υπάρχει μεγάλη ανάγκη για την διαχείριση του κυκλοφοριακού φόρτου σε πραγματικό χρόνο στις σύγχρονες μεγαλουπόλεις μέσω συστημάτων και δομών οι οποίες εκμεταλλεύονται την δυναμική των ρευμάτων δεδομένων και των ερωτημάτων διαρκείας.

Σκοπός ήταν η απεικόνιση σε πραγματικό χρόνο του κυκλοφοριακού φόρτου στο οδικό δίκτυο της Αθήνας και συγχρόνως η ανάδειξη της σημασίας των παραθύρων στις εφαρμογές ρευμάτων δεδομένων, φανερώνοντας πόσο επιτακτική είναι η χρήση τους σε ερωτήματα διαρκείας που χρησιμοποιούν βασικούς τελεστές της σχεσιακής άλγεβρας.

Το επίπεδο προσέγγισης στις επιστρεφόμενες απαντήσεις είναι καλά προσδιορισμένο από την σημασιολογία των παραθύρων και άμεσα αντιληπτό από τους χρήστες. Πρέπει ωστόσο να σημειωθεί ότι όπως στην πλειοψηφία των εφαρμογών ρευμάτων δεδομένων, η επιδιωκόμενη επεξεργασία επικεντρώνεται στην πιο πρόσφατη πληροφορία. Η εφαρμογή των παραθύρων σε αυτήν την περίπτωση έρχεται περισσότερο να εκπληρώσει αυτήν την απαίτηση παρά να επηρεάσει την ακρίβεια των απαντήσεων.

Οι δοκιμές με τα συνδυαστικά πειραματικά δεδομένα που προσομοιώνουν την κίνηση οχημάτων στο οδικό δίκτυο της Αθήνας, καταδεικνύουν ότι η εφαρμογή η οποία βασίζεται στο σύστημα TelegraphCQ: (α) αντέχει σε ρυθμούς εισροής δεδομένων τουλάχιστον 10000 οχήματα ανά 15 δευτερόλεπτα, (β) υποστηρίζει τα απαιτητικά αιτήματα που διατυπώνονται, (γ) επιτρέπει τον ορισμό παράγωγων ρευμάτων και πολύπλοκων ερωτημάτων και (δ) απεικονίζει εγκαίρως τον κυκλοφοριακό φόρτο των οδικών αξόνων.

Ωστόσο, παρατηρήθηκε ότι τα σύνθετα χωροχρονικά ερωτήματα που εκφράζονται με την βοήθεια των καθορισμένων συναρτήσεων PL/pgSQL, λειτουργούν μόνο για αρχειοδετημένα ρεύματα.

Επιγραμματικά λοιπόν, η παρούσα υλοποίηση της εφαρμογής μπορεί να χαρακτηριστεί ως ο σκελετός ενός συστήματος διαχείρισης ρευμάτων δεδομένων με διαθέσιμα τα βασικά του στοιχεία και αποτελεί την βάση για περαιτέρω επέκταση όπως αναλύεται στην συνέχεια.

6.2 Προτάσεις - Μελλοντικές επεκτάσεις

Λόγω της αυξημένης ζήτησης για την ύπαρξη «έξυπνων» συστημάτων διαχείρισης του κυκλοφοριακού φόρτου σε πραγματικό χρόνο στις μεγαλουπόλεις, η συγκεκριμένη εφαρμογή θα μπορούσε να επεκταθεί αρκετά περισσότερο ώστε να καλύπτει και ορισμένες επιπλέον ανάγκες των χρηστών του οδικού δικτύου αλλά και των διαχειριστών της κυκλοφοριακής κίνησης.

Ακολουθούν ορισμένες χρήσιμες προτάσεις και μελλοντικές επεκτάσεις, οι οποίες θα μπορούσαν να εφαρμοστούν κατόπιν αναθεώρησης ή τροποποίησης ορισμένων λειτουργιών και διαδικασιών της εφαρμογής, για την πληρέστερη ενημέρωση των οδηγών των κινούμενων οχημάτων, καθώς και για την δικαιότερη επιβολή χρέωσης αυτών με το αντίστοιχο τέλος συμφόρησης.

Δυνατότητα προϋπολογισμού της συνολικής χρέωσης. Στους οδηγούς των κινούμενων οχημάτων να παρέχεται η δυνατότητα της εκ των προτέρων γνώσης της συνολικής χρέωσής τους. Επιλέγοντας να κινηθούν βάσει ορισμένων συγκεκριμένων υποθετικών διαδρομών, η χρέωση να μην είναι σταθερή αλλά δυναμική ανάλογα με τον εκάστοτε φόρτο των οδικών τμημάτων που τις συνδέτουν και την ώρα που επιλέγεται για την μετακίνησή τους.

Δυνατότητα χρονικής εκτίμησης διαδρομών. Να παρέχεται η δυνατότητα της εκτίμησης του χρόνου που απαιτείται για την μετακίνηση ενός οχήματος από ένα σημείο της πόλης σε ένα άλλο.

Ρουτίνες βέλτιστης διαδρομής. Να υπάρχουν ενσωματωμένες στην εφαρμογή στοχαστικές ρουτίνες ελάχιστης διαδρομής (shortest path), οι οποίες να βασίζονται στις τρέχουσες κυκλοφοριακές συνθήκες. Τέτοιου είδους ρουτίνες θα βοηθούσαν τους οδηγούς να αποφασίσουν και να επιλέξουν την βέλτιστη διαδρομή για την μετακίνησή τους με τα οχήματα.

Υποτροπή οδηγών. Να λαμβάνεται υπόψη η προστασία των οδηγών των οποίων παρακολουθούνται τα οχήματα τους όταν κινούνται στο οδικό δίκτυο της πόλης, ώστε αν υπάρξουν συνεχόμενες υψηλές χρεώσεις (πάνω από ένα καθορισμένο χρηματικό όριο) για αρκετές μέρες, να ενημερώνονται μέσω μιας συμβουλευτικής υποτροπής για την επανεξέταση και αλλαγή των διαδρομών που χρησιμοποιούν στις καθημερινές μετακινήσεις τους, ιδιαίτερα για ώρες αιχμής.

Διαχωρισμός του δακτυλίου. Να υπάρχει διαφορετικό κοστολόγιο χρέωσης για την κίνηση των οχημάτων στους οδικούς άξονες που περιλαμβάνονται στο δακτύλιο της πόλης, απ' ό,τι στους οδικούς άξονες οι οποίοι βρίσκονται εκτός δακτυλίου.

Επανεξέταση υπολογισμού της χρέωσης. Κάθε κινούμενο όχημα να μην χρεώνεται απλά κατά την είσοδό του στον οδικό άξονα, αλλά να λαμβάνεται υπόψη και ο χρόνος παραμονής και κίνησης του κατά μήκος του άξονα. Επίσης, αν και είναι αρκετά δύσκολο, θα μπορούσε να εφαρμοστεί κάποιος αλγόριθμος ο οποίος όσο το όχημα θα χρησιμοποιούσε τον οδικό άξονα για την μετακίνησή του,

θα υπολόγιζε συνεχώς την αντίστοιχη χρέωση που πρέπει να του επιβληθεί, ανάλογα με τον μέσο κυκλοφοριακό φόρτο που θα παρουσίαζε ο οδικός άξονας (δυναμικά) καθ' όλη την χρονική περίοδο κίνησης του οχήματος σε αυτόν.

Απεικόνιση κυκλοφοριακού φόρτου σε κόμβους. Να παρέχεται η δυνατότητα από το σύστημα να απεικονίζεται ο κυκλοφοριακός φόρτος στους κύριους οδικούς κόμβους της πόλης, λαμβάνοντας υπόψη τον φόρτο των συμβαλλόμενων οδών σε αυτούς. Μέσω αυτής της απεικόνισης, οι οδηγοί θα γνωρίζουν ποιους κόμβους του οδικού δικτύου πρέπει να αποφύγουν κατά την μετακίνησή τους με τα οχήματα.

Παραρτήματα

Παράρτημα Α: Δομή και περιεχόμενο γεωγραφικού αρχείου

Η δομή και το περιεχόμενο του γεωγραφικού αρχείου είναι η εξής:

- Θεματικό επίπεδο: Οδικά τμήματα του οδικού δικτύου της Αθήνας.
- Τύπος αρχείου: ESRI shapefile (shp)
- Περιεχόμενο: Βασικό οδικό δίκτυο πολεοδομικού συγκροτήματος Αθηνών
- Δομή: Γραμμική
- Πρωτογενείς πηγές: Χάρτες κλίμακας 1:5000
- Σύστημα αναφοράς: ΕΓΣΑ'87 (συντεταγμένες σε μέτρα)
- Προβολή: Εγκάρσια Μερκατορική
- Ελλειψοειδές αναφοράς: GRS 80
- Πλήθος εγγραφών: 8589

Παράρτημα Β: Βάση δεδομένων

❖ Βάση δεδομένων της εφαρμογής

Για τον καθορισμό της βάσης δεδομένων, δημιουργείται αρχικά ένα ειδικό σχήμα (traffic) για την προσαρμογή όλων των χωρικών και μη στοιχείων, στα οποία γίνεται προσπέλαση από την εφαρμογή.

Η υλοποίηση των πινάκων πραγματοποιείται με τη χρήση της δηλωτικής γλώσσας SQL που είναι ενσωματωμένη στη PostgreSQL και αποτελεί την δημοφιλέστερη γλώσσα που χρησιμοποιείται για την πρόσβαση στις βάσεις δεδομένων.

Επομένως ο ορισμός των πινάκων σύμφωνα με τη γλώσσα ορισμού DDL (Data Definition Language) πραγματοποιείται κατόπιν της χρήσης της εντολής CREATE TABLE, εφόσον δημιουργείται αρχικά ένα ειδικό σχήμα (traffic) για την προσαρμογή όλων των πινάκων στα οποία γίνεται προσπέλαση από την εφαρμογή.

Επίσης, το TelegraphCQ επεκτείνει την γλώσσα ορισμού δεδομένων (Data Definition Language – DDL), η οποία αποτελεί τμήμα της SQL, με ειδικές εντολές για τη διαχείριση ρευμάτων δεδομένων (CREATE STREAM, ALTER STREAM, DROP STREAM).

Δημιουργία σχήματος της βάσης δεδομένων (Traffic)

```
CREATE SCHEMA traffic;
```

A. Στατικοί πίνακες

Πίνακας «Οδικά τμήματα»

```
CREATE TABLE traffic.Road_segments
(Segment          PATH,
 Road_segmentID   INTEGER,
 AxisID           INTEGER NOT NULL,
 Roadname_gr      VARCHAR(80) NOT NULL,
 Roadname_lt      VARCHAR(50) NOT NULL,
 Category         SMALLINT NOT NULL,
 One_way          CHAR(2) NOT NULL,
 Length           DECIMAL(10,3) NOT NULL,
 Lanes            SMALLINT);
```

```
CREATE UNIQUE INDEX rdsegs_pk ON
traffic.Road_segments(Road_SegmentID);
CREATE INDEX rdsegs_idx ON traffic.Road_segments USING RTREE
(polygon(segment));
```

Πίνακας «Οδικοί άξονες»

```
CREATE TABLE traffic.Road_axes
(AxisID           INTEGER CONSTRAINT rdaxes_pk PRIMARY KEY,
 Load_coefficient DECIMAL(10,3),
 Axis_namegr      VARCHAR(80),
 Axis_namel      VARCHAR(50));
```

Πίνακας «Χρεώσεις»

```
CREATE TABLE traffic.Charges
(Start_time      TIME WITHOUT TIME ZONE NOT NULL,
 End_time        TIME WITHOUT TIME ZONE NOT NULL,
 Load_min        DOUBLE PRECISION NOT NULL,
 Load_max        DOUBLE PRECISION NOT NULL,
 Charge          DOUBLE PRECISION,
 Load_class      CHAR(1),
```

```
CONSTRAINT charges_pk PRIMARY KEY (Start_time, End_time,
Load_min, Load_max));
```

B. Δυναμικός πίνακας (Ρεύμα δεδομένων θέσης κινούμενων οχημάτων)

Δημιουργία wrapper

```
CREATE WRAPPER trafficwrapper (
(init=csv_init,
next=csv_next,
done=csv_done)
WITH WRAPPERINFO 'DELIMITERS2=;,DEFAULT_DELIMITERS=';
```

Πίνακας «Τοποθεσίες κινούμενων οχημάτων»

```
CREATE STREAM traffic.Locations
(VehicleID CHAR(7),
 Position POINT,
 SegmentID INTEGER,
 T INTEGER,
 Tcqtime TIMESTAMP TIMESTAMPCOLUMN)
TYPE ARCHIVED;
```

Σύνδεση wrapper με το ρεύμα δεδομένων

```
ALTER STREAM traffic.Locations
ADD WRAPPER trafficwrapper;
```

Γ. Παράγωγοι πίνακες (Παραγόμενα ρεύματα και πίνακας)

Πίνακας «Τροχιές κινούμενων οχημάτων»

```
CREATE STREAM traffic.Trajectories
(VehicleID CHAR(7),
 AxisID INTEGER,
 Displacement DOUBLE PRECISION,
 Speed DOUBLE PRECISION,
 Apply_charge BOOLEAN,
 tcqtime TIMESTAMP TIMESTAMPCOLUMN)
TYPE ARCHIVED;
```

Πίνακας «Φόρτος οδικών αξόνων»

```
CREATE STREAM traffic.Roadaxes_load
(AxisID INTEGER,
 Vehicle_count INTEGER,
 Load DOUBLE PRECISION,
 Charge DOUBLE PRECISION,
 Load_class CHAR(1),
 Tcqtime TIMESTAMP TIMESTAMPCOLUMN)
TYPE ARCHIVED;
```

Πίνακας «Χρεώσεις οχημάτων»

```
CREATE TABLE traffic.Congestion_charges
(VehicleID CHAR(7),
 Congestion_charge DOUBLE PRECISION,
 TS TIMESTAMP NOT NULL);
```

| Road segments | | | | | | | | |
|--------------------|-------------------|------------------|-------------|--------------|------------|---------|--------|-------|
| Segment | Road segmentID | AxisID | Roadname gr | Roadname It | Category | One way | Length | Lanes |
| Road axes | | | | | | | | |
| AxisID | Axis_nameIt | Load_coefficient | | | | | | |
| Charges | | | | | | | | |
| Start time | End time | Load min | Load max | Charge | Load class | | | |
| Locations | | | | | | | | |
| VehicleID | Position | SegmentID | T | Tcptime | | | | |
| Trajectories | | | | | | | | |
| VehicleID | AxisID | Displacement | Speed | Apply_charge | Tcptime | | | |
| Road axes load | | | | | | | | |
| AxisID | Vehicle count | Load | Charge | Load class | Tcptime | | | |
| Congestion charges | | | | | | | | |
| VehicleID | Congestion_charge | TS | | | | | | |

Σχήμα Β.1: Διάγραμμα απεικόνισης πινάκων της βάσης δεδομένων της εφαρμογής.

❖ Εντολή τροφοδότησης δυναμικού ρεύματος με δεδομένα

Προκειμένου να μεταδοθούν τα δεδομένα στο TelegraphCQ και να τροφοδοτηθεί το αντίστοιχο ρεύμα δεδομένων, εκτελείται η ακόλουθη εντολή, σε περιβάλλον λειτουργικού συστήματος Linux:

```
cat traffic10k_wgs84.log | sliderate.pl localhost 5533
trafficwrapper,traffic.Locations 10000 15
```

❖ Δομή κέρσορα για ερωτήματα διάρκειας

Όλα τα ερωτήματα διάρκειας τα οποία διατυπώνονται στο σύστημα TelegraphCQ, περιλαμβάνονται αναγκαστικά μέσα σε ένα κέρσορα ο οποίος είναι υπεύθυνος για την προώθηση των αποτελεσμάτων που παράγονται. Με την εντολή «DECLARE» δηλώνεται ο κέρσορας και με την εντολή «FETCH FORWARD» δηλώνονται πόσα αποτελέσματα πρέπει να επιστρέφει ο κέρσορας. Παρακάτω ακολουθεί παράδειγμα εφαρμογής ενός κέρσορα (π.χ. του portion1) σε κάποιο ερώτημα:

```
BEGIN WORK;

DECLARE portion1 CURSOR FOR
SELECT ...
FROM ... [RANGE BY '...' SLIDE BY '...']
WHERE ...;

FETCH FORWARD ... IN portion1;
CLOSE portion1;
COMMIT WORK;
```

❖ Επιπλέον ερωτήματα

Η δομή και η μοντελοποίηση που εφαρμόστηκε στην βάση δεδομένων του συστήματος TelegraphCQ, επιτρέπει την διατύπωση ορισμένων επιπλέον σημαντικών χωροχρονικών ερωτημάτων γενικού ενδιαφέροντος. Αυτά τα επιπλέον ερωτήματα περιγράφονται σε γλώσσα SQL και CQL, για τα απλά χωρικά και χωροχρονικά αντίστοιχα ερωτήματα, τα οποία μπορούν να απαντηθούν επιτυχώς από την εφαρμογή. Μέσα από τα ερωτήματα επιβεβαιώνεται η λειτουργικότητα της χωρικής βάσης και η διατύπωση τους προσφέρει έναν έλεγχο καλής λειτουργίας της εφαρμογής. Σημειώνεται, ότι επειδή το σύστημα αναφοράς των συντεταγμένων είναι το WGS'84, οι συντεταγμένες των γεωμετρικών χωρικών στοιχείων είναι σε φ και λ, και όχι σε X και Y.

Q1. Spatiotemporal range window: «Εντόπισε όλα τα οχήματα που βρίσκονται τα τελευταία 10 λεπτά εντός του οδικού άξονα “8174”».

```
SELECT VehicleID, Tcqtime
FROM traffic.Trajectories [RANGE BY '10 MINUTES'
                           SLIDE BY '1 MINUTE']
WHERE AxisID = 8174;
```

Το ερώτημα επιστρέφει ως αποτέλεσμα κάθε ένα λεπτό, τα οχήματα που βρίσκονται στον οδικό άξονα με κωδικό “8174”, λαμβάνοντας υπόψη του ένα χρονικό παράθυρο 10 λεπτών.

Q2. Spatiotemporal range window: «Εντόπισε όλα τα οχήματα που πέρασαν τα τελευταία 10 λεπτά από τον οδικό άξονα 8174».

```
SELECT T.VehicleID, T.Tcqtime
FROM traffic.Trajectories [RANGE BY '10 MINUTES'
                           SLIDE BY '10 MINUTES']
WHERE AxisID = 8174;
```

Το ερώτημα επιστρέφει ως αποτέλεσμα κάθε 10 λεπτά, τα οχήματα τα οποία διέσχισαν τον οδικό άξονα με κωδικό “8174”, λαμβάνοντας υπόψη του ένα χρονικό παράθυρο 10 λεπτών. (επάλληλο παράθυρο - tumbling window)

Q3. Spatial containment query: «Βρες όλα τα οχήματα που εντοπίζονται τώρα εντός του δακτυλίου στο κέντρο της Αθήνας».

```
SELECT L.VehicleID, RA.AxisID, RA.Axis_name, L.Tcqtime
FROM traffic.Locations AS L [RANGE BY '15 SECONDS'
                             SLIDE BY '15 SECONDS'],
     traffic.Road_segments AS RS, traffic.Road_axes AS RA
WHERE (L.Position @
polygon'(23.730975,37.992308,23.714093,37.984883,23.698052,37.9
68771,23.708069,37.962887,23.730625,37.957774,23.747507,37.9647
08,23.766981,37.986914)')=TRUE
AND L.SegmentID = RS.Road_segmentid
AND RS.AxisID = RA.AxisID;
```

Στις χωρικές βάσεις δεδομένων, ένα ερώτημα παραθύρου επιστρέφει τα χαρακτηριστικά γνωρίσματα που εμπίπτουν μέσα σε μία πολυγωνική περιοχή ενδιαφέροντος. Στην προκειμένη περίπτωση που πρόκειται για επεξεργασία ρεύματος θέσης οχημάτων, εφαρμόζεται απλά μία παραδυρική κατασκευή ενάντια σε οποιεσδήποτε θέσεις λαμβάνονται πέρα από το πιο πρόσφατο χρονόσημο και ελέγχεται αυτό το πεπερασμένο σύνολο σημείων για χωρικό εγκλεισμό του (τελεστής @) σε σχέση με μία στάσιμη περιοχή ενδιαφέροντος, η οποία καθορίζεται σύμφωνα με τη σημασιολογία της PostgreSQL.

Το ερώτημα επιστρέφει ως αποτέλεσμα κάθε 15 δευτερόλεπτα, τα οχήματα τα οποία εντοπίζονται εντός των ορίων του δακτυλίου στο κέντρο της Αθήνας, ξεκινώντας την χρονική στιγμή που εκτελείται το ερώτημα και λαμβάνοντας υπόψη του ένα χρονικό παράθυρο 15 δευτερολέπτων. Με τον πρώτο τρόπο υπολογίζεται η απάντηση κάνοντας χρήση των χωρικών ορίων του δακτυλίου, ενώ με τον δεύτερο τρόπο υπολογίζεται η απάντηση κάνοντας χρήση των περιγραφικών χαρακτηριστικών των οδικών αξόνων, δηλαδή αν είναι μέσα στον δακτύλιο ή αποτελούν όριο του.

Q4. Aggregate query: «Ειδοποίησε όταν περισσότερα από 15 οχήματα εντοπιστούν ταυτόχρονα εντός του δακτυλίου της Αθήνας».

```
SELECT Tcqtime, COUNT(VehicleID) AS Count
FROM traffic.Locations [RANGE BY '1 MINUTE'
                        SLIDE BY '15 SECONDS'
                        START AT '2010-01-12 12:01:00']
WHERE (Position @
polygon'(23.730975,37.992308,23.714093,37.984883,23.698052,37.968771,
23.708069,37.962887,23.730625,37.957774,23.747507,37.964708,23.766981
,37.986914)')=TRUE
GROUP BY Tcqtime
HAVING COUNT(VehicleID) >= 15;
```

Στην προκειμένη περίπτωση εφαρμόζεται απλά μία παραδυρική κατασκευή ενάντια σε οποιεσδήποτε θέσεις λαμβάνονται πέρα από το πιο πρόσφατο χρονόσημο και ελέγχεται αυτό το πεπερασμένο σύνολο σημείων για χωρικό εγκλεισμό του (τελεστής @) σε σχέση με μία στάσιμη περιοχή ενδιαφέροντος, η οποία καθορίζεται σύμφωνα με τη σημασιολογία της PostgreSQL.

Το ερώτημα επιστρέφει ως αποτέλεσμα κάθε 15 δευτερόλεπτα, τον αριθμό των οχημάτων τα οποία εντοπίζονται εντός των ορίων του δακτυλίου στο κέντρο της Αθήνας όταν ξεπερνούν σε αριθμό τα 15, ξεκινώντας την χρονική στιγμή 12:01:00 της 01-12-2010 και λαμβάνοντας υπόψη του ένα χρονικό παράθυρο ενός λεπτού. Με τον πρώτο τρόπο υπολογίζεται η απάντηση κάνοντας χρήση των χωρικών ορίων του δακτυλίου, ενώ με τον δεύτερο τρόπο υπολογίζεται η απάντηση κάνοντας χρήση των περιγραφικών χαρακτηριστικών των οδικών αξόνων, δηλαδή αν είναι μέσα στον δακτύλιο ή αποτελούν όριο του.

Q5. Periodic aggregate query: «Κάθε λεπτό υπολόγισε τον τρέχοντα αριθμό οχημάτων εντός του δακτυλίου».

```
SELECT COUNT(VehicleID) AS Vehicles_count, Tcqtime
FROM traffic.Locations [RANGE BY '1 MINUTE'
                        SLIDE BY '1 MINUTE'
                        START AT '2010-01-12 12:01:00']

WHERE (Position @
polygon'(23.730975,37.992308,23.714093,37.984883,23.698052,37.9
68771,23.708069,37.962887,23.730625,37.957774,23.747507,37.9647
08,23.766981,37.986914)')=TRUE
GROUP BY Tcqtime;
```

Στην προκειμένη περίπτωση εφαρμόζεται απλά μία παραδυρική κατασκευή ενάντια σε οποιεσδήποτε θέσεις λαμβάνονται πέρα από το πιο πρόσφατο χρονόσημο και ελέγχεται αυτό το πεπερασμένο σύνολο σημείων για χωρικό εγκλεισμό του (τελεστής @) σε σχέση με μία στάσιμη περιοχή ενδιαφέροντος, η οποία καθορίζεται σύμφωνα με τη σημασιολογία της PostgreSQL.

Το ερώτημα επιστρέφει ως αποτέλεσμα κάθε ένα λεπτό, τον αριθμό των οχημάτων τα οποία εντοπίζονται εντός των ορίων του δακτυλίου στο κέντρο της Αθήνας, ξεκινώντας την χρονική στιγμή 12:01:00 της 01-12-2010 και λαμβάνοντας υπόψη του ένα χρονικό παράθυρο ενός λεπτού. Με τον πρώτο τρόπο υπολογίζεται η απάντηση κάνοντας χρήση των χωρικών ορίων του δακτυλίου, ενώ με τον δεύτερο τρόπο υπολογίζεται η απάντηση κάνοντας χρήση των περιγραφικών χαρακτηριστικών των οδικών αξόνων, δηλαδή αν είναι μέσα στον δακτύλιο ή αποτελούν όριο του.

Q6. Query with complex spatial predicates. «Εντόπισε όλα τα οχήματα που διασχίζουν τα όρια του δακτυλίου».

```
SELECT VehicleID, MAX(Tcqtime), COUNT(VehicleID)
FROM traffic.Locations [RANGE BY '30 SECONDS'
                        SLIDE BY '15 SECONDS'
                        START AT '2010-01-12 12:01:00']

WHERE (Position @
polygon'(23.730975,37.992308,23.714093,37.984883,23.698052,37.9
68771,23.708069,37.962887,23.730625,37.957774,23.747507,37.9647
08,23.766981,37.986914)')=TRUE
GROUP BY VehicleID
HAVING COUNT(VehicleID) = 1;
```

Στην προκειμένη περίπτωση εφαρμόζεται απλά μία παραδυρική κατασκευή ενάντια σε οποιεσδήποτε θέσεις λαμβάνονται πέρα από το πιο πρόσφατο χρονόσημο και ελέγχεται αυτό το πεπερασμένο σύνολο σημείων για χωρικό εγκλεισμό του (τελεστής @) σε σχέση με μία στάσιμη περιοχή ενδιαφέροντος, η οποία καθορίζεται σύμφωνα με τη σημασιολογία της PostgreSQL. Το ερώτημα επιστρέφει ως αποτέλεσμα κάθε 15 δευτερόλεπτα, τον αριθμό των οχημάτων τα οποία διασχίζουν τα όρια του δακτυλίου στο κέντρο της Αθήνας, ξεκινώντας την χρονική στιγμή 12:01:00 της 01-12-2010 και λαμβάνοντας υπόψη του ένα χρονικό παράθυρο 30 δευτερολέπτων.

Q7. Currently moving objects: «Εντόπισε όλα τα οχήματα που αυτή τη στιγμή είναι εν κινήσει».

```
SELECT VehicleID, Tcqtime
FROM traffic.Trajectories [RANGE BY '15 SECONDS'
                           SLIDE BY '15 SECONDS']
WHERE Speed > 0;
```

Το ερώτημα επιστρέφει ως αποτέλεσμα κάθε 15 δευτερόλεπτα, τον αριθμό των οχημάτων τα βρίσκονται σε κατάσταση κίνησης, λαμβάνοντας υπόψη του ένα χρονικό παράθυρο 15 δευτερολέπτων.

Q8. Trajectory query: «Βρες όλα τα οδικά τμήματα που έχει διανύσει το όχημα “40” και υπολόγισε το αντίστοιχο συνολικό χρονικό διάστημα που έχει μεσολαβήσει για την διέλευση του από τα τμήματα αυτά και τα συνολικά διανυθέντα χιλιόμετρα».

```
SELECT SUM(Displacement) AS Total_meters, AGE(MAX(Tcqtime) -
MIN(Tcqtime)) AS Total_time, AxisID
FROM traffic.Trajectories [RANGE BY '30 MINUTES'
                           SLIDE BY '10 MINUTES']
WHERE VehicleID = 40
GROUP BY AxisID;
```

Το ερώτημα επιστρέφει ως αποτέλεσμα κάθε 10 λεπτά, όλα τα οδικά τμήματα που έχει διανύσει το όχημα με κωδικό “40”, το αντίστοιχο συνολικό χρονικό διάστημα που έχει μεσολαβήσει για την διέλευση του από τα τμήματα αυτά καθώς και το συνολικό άθροισμα των χιλιομέτρων που έχει διανύσει, λαμβάνοντας υπόψη του ένα χρονικό παράθυρο 30 λεπτών.

Q9. Windowed stream-join query: «Εντόπισε τα οχήματα που κάποια στιγμή κατά τα τελευταία 3 λεπτά, βρίσκονταν κοντά στον “ΕΥΑΓΓΕΛΙΣΜΟ” λιγότερο από 250 μέτρα (με συντεταγμένες)».

```
SELECT VehicleID, Tcqtime
FROM traffic.Locations [RANGE BY '3 MINUTES'
                        SLIDE BY '1 MINUTE']
WHERE (Position <-> point'(23.746325,37.978083)') <= 250;
```

Το ερώτημα επιστρέφει ως αποτέλεσμα κάθε ένα λεπτό, όλα τα οχήματα τα οποία βρέθηκαν σε απόσταση ίση ή μικρότερη των 250 μέτρων από το νοσοκομείο “ΕΥΑΓΓΕΛΙΣΜΟΣ”, λαμβάνοντας υπόψη του ένα χρονικό παράθυρο 3 λεπτών. Ουσιαστικά κάθε λεπτό επιστρέφονται τα οχήματα τα οποία εντός μιας χρονικής διάρκειας 3 λεπτών, βρέθηκαν ή βρίσκονται ακόμα εντός μίας απόστασης μικρότερης ή ίσης των 250 μέτρων. Σημειώνεται ότι το νοσοκομείο δεν απεικονίζεται ως πολύγωνο με όρια, αλλά ως σημείο με συγκεκριμένες συντεταγμένες (x,y) και ότι η γεωμετρική λειτουργία (τελεστής <->) υπολογίζει την Ευκλείδεια απόσταση.

Q10. Traffic load query: «Βρες ποιοι άξονες του οδικού δικτύου έχουν κυκλοφοριακό φόρτο πάνω από 50».

```
SELECT RL.AxisID, RA.Axis_name||t, RL.Load
FROM traffic.Roadaxes_load AS RL [RANGE BY '5 MINUTES'
                                SLIDE BY '30 SECONDS'],
     traffic.Road_axes RA
WHERE RL.AxisID = RA.AxisID
AND RL.Load > 50;
```

Το ερώτημα επιστρέφει ως αποτέλεσμα κάθε 30 δευτερόλεπτα, τους οδικούς άξονες οι οποίοι έχουν τιμή κυκλοφοριακού φόρτου πάνω από 50, λαμβάνοντας υπόψη του ένα χρονικό παράθυρο 5 λεπτών.

Q11. Congestion charge query: «Υπολόγισε πόση ήταν η συνολική χρέωση του οχήματος “105” την ημέρα '2010-01-12'».

```
SELECT VehicleID, SUM(Congestion_charge) AS Total_charge
FROM traffic.Congestion_charges
WHERE VehicleID = 105
AND SUBSTRING(ts,1,10) = '2010-01-12'
GROUP BY VehicleID;
```

Το ερώτημα επιστρέφει ως αποτέλεσμα, την συνολική χρέωση του οχήματος με κωδικό “105”, για την ημέρα με ημερομηνία 01-12-2010.

Q12. Congestion charge query: «Υπολόγισε πόση είναι η συνολική χρέωση του οχήματος 105 τώρα».

```
SELECT VehicleID, SUM(Congestion_charge) AS Total_charge
FROM traffic.Congestion_charges
WHERE VehicleID = 105
GROUP BY VehicleID;
```

Το ερώτημα επιστρέφει ως αποτέλεσμα, την συνολική χρέωση του οχήματος με κωδικό “105”, που έχει χρεωθεί στο συγκεκριμένο όχημα έως την συγκεκριμένη χρονική στιγμή που εκτελείται το ερώτημα.

❖ Πιθανή επέκταση της βάσης δεδομένων της εφαρμογής

Δίνεται μία πιθανή επέκταση της βάσης δεδομένων με περισσότερους στατικούς πίνακες, οι οποίοι θα μπορούσαν να επεκτείνουν τον σχεδιασμό της χωρικής βάσης και να χρησιμοποιηθούν από την εφαρμογή στην μετέπειτα εξαγωγή πληροφορίας μέσω των λογικών ερωτημάτων. Συνεπώς, η βάση δεδομένων του συστήματος θα μπορούσε να συγκροτείται και από τους εξής τύπους χωρικών δεδομένων:

Σημειακές θέσεις γενικού ενδιαφέροντος. Θέσεις πάνω στο οδικό δίκτυο, οι οποίες μπορεί να αφορούν συγκεκριμένα κτίρια και καταστήματα (πχ.

νοσοκομεία, στάδια, εμπορικά κέντρα, σιδηροδρομικοί σταθμοί, κ.α.) ή συγκεκριμένους χώρους (π.χ. πάρκα, πλατείες, κ.α.).

Πολυγωνικές ζώνες ιδιαίτερου ενδιαφέροντος. Ζώνες οι οποίες καθορίζουν τα όρια συγκεκριμένων περιοχών, όπως για παράδειγμα τα όρια των γειτονικών δήμων, το λιμάνι, κ.α., καθώς και ο δακτύλιος στο κέντρο της Αθήνας όπου ισχύουν και εφαρμόζονται στα διερχόμενα οχήματα προκαθορισμένες κυκλοφοριακές ρυθμίσεις.

Η επέκταση της βάσης δεδομένων της εφαρμογής θα αποτελείται από τις παραπάνω χωρικές οντότητες και μία μη χωρική (αφορά τις ιδιότητες των κινούμενων οχημάτων), οι οποίες διαθέτουν χωρικές και περιγραφικές ιδιότητες. Όλες αυτές οι οντότητες θα είναι καταχωρημένες στους ανάλογους στατικούς πίνακες.

Πίνακες με χωρικές και περιγραφικές ιδιότητες:

- Σημεία ενδιαφέροντος (Landmarks)
- Περιοχές ενδιαφέροντος (Zones)

Πίνακας με περιγραφικές ιδιότητες:

- Κινούμενα Οχήματα (Vehicles)

Ο πίνακας «**Σημεία ενδιαφέροντος**» θα έχει ως χαρακτηριστικά γνωρίσματα τα εξής:

- Κωδικός του σημείου ενδιαφέροντος που θα αποτελεί και το πρωτεύον κλειδί του συγκεκριμένου πίνακα.
- Ονομασία του σημείου.
- Γεωμετρικός τύπος (στη συγκεκριμένη περίπτωση θα είναι σημειακός).

Ο πίνακας «**Περιοχές ενδιαφέροντος**» θα έχει ως χαρακτηριστικά γνωρίσματα τα εξής:

- Κωδικός της ζώνης της περιοχής ενδιαφέροντος που θα αποτελεί και το πρωτεύον κλειδί του συγκεκριμένου πίνακα
- Ονομασία της περιοχής.
- Γεωμετρικός τύπος (στη συγκεκριμένη περίπτωση θα είναι πολυγωνικός).

Ο πίνακας «Κινούμενα οχήματα» θα έχει ως χαρακτηριστικά γνωρίσματα τα εξής:

- Κωδικός του κινούμενου οχήματος που θα αποτελεί και το πρωτεύον κλειδί του συγκεκριμένου πίνακα (συνήθως προτιμάται η πινακίδα με τον αριθμό κυκλοφορίας του οχήματος.
- Εταιρία κατασκευής του οχήματος.
- Τύπος του οχήματος (π.χ. μοτοσικλέτα, επιβατικό, φορτηγό κ.λπ.).
- Χρώμα του οχήματος.

Η υλοποίηση των στατικών αυτών πινάκων, θα πραγματοποιείται με τη χρήση της δηλωτικής γλώσσας SQL που είναι ενσωματωμένη στη PostgreSQL. Συνεπώς ο ορισμός των πινάκων σύμφωνα με τη γλώσσα ορισμού DDL (Data Definition Language) και κατόπιν της χρήσης της εντολής CREATE TABLE, οι οποίοι θα υπάρχουν και αυτοί στο ειδικό σχήμα (traffic) ώστε να γίνεται ενιαία προσπέλαση από την εφαρμογή, θα είναι ο εξής:

Πίνακας σημείων ενδιαφέροντος (Landmarks)

```
CREATE TABLE traffic.Landmarks
(Landmark          POINT,
 LandmarkID        INTEGER CONSTRAINT landmarks_pk PRIMARY KEY,
 Landmark_name     VARCHAR(30) NOT NULL);
```

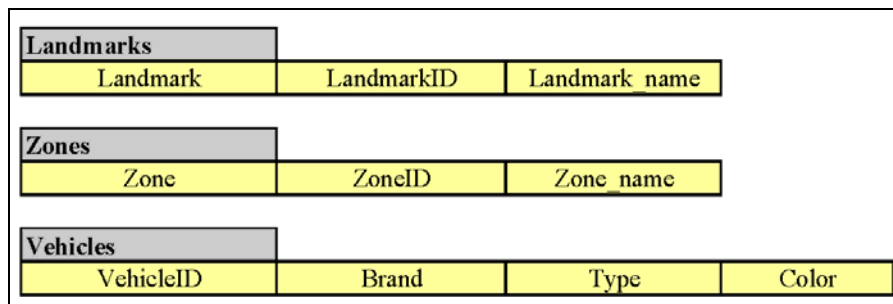
Πίνακας περιοχών ενδιαφέροντος (Zones)

```
CREATE TABLE traffic.Zones
(Zone              POLYGON,
 ZoneID            INTEGER CONSTRAINT zones_pk PRIMARY KEY,
 Zone_name         VARCHAR(30) NOT NULL);
```

```
CREATE INDEX zones_idx ON traffic.zones USING RTREE (zone);
```

Κινούμενα Οχήματα (Vehicles)

```
CREATE TABLE traffic.Vehicles
(VehicleID        CHAR(7) CONSTRAINT vehicles_pk PRIMARY KEY,
 Brand            VARCHAR(20) NOT NULL,
 Type            VARCHAR(10) NOT NULL,
 Color           VARCHAR(10) NOT NULL);
```



Σχήμα Β.2: Διάγραμμα απεικόνισης πινάκων επέκτασης της βάσης δεδομένων.

Παράρτημα Γ: Κώδικες εφαρμογής

❖ Κώδικας συναρτήσεων PL/pgSQL

A. Συνάρτηση createTrajectory:

```
CREATE OR REPLACE FUNCTION createTrajectory(timestamp) RETURNS
INTEGER AS '
DECLARE
    curID1 char(7);
    curID2 char(7);
    curPoint1 point;
    curPoint2 point;
    curDispl double precision;
    curTimestamp1 timestamp;
    curTimestamp2 timestamp;
    curSegment1 integer;
    curSegment2 integer;
    curAxis1 integer;
    curAxis2 integer;
    curApplycharge varchar(5);

    curPast2Positions CURSOR (ts timestamp) IS
        SELECT L.vehicleID, L.position, L.SegmentID, L.tcqtime
        FROM traffic.Locations AS L [RANGE BY ''30 SECONDS''
        SLIDE BY ''15 SECONDS'']
        WHERE L.tcqtime <= ts AND L.tcqtime >= ts - interval ''15
        seconds''
        ORDER BY L.vehicleID, L.tcqtime;

BEGIN

    OPEN curPast2Positions($1);

    LOOP
        FETCH curPast2Positions INTO curID1, curPoint1,
        curSegment1, curTimestamp1;
        FETCH curPast2Positions INTO curID2, curPoint2,
        curSegment2, curTimestamp2;

        IF NOT FOUND THEN
            EXIT;
        END IF;

        curDispl := curPoint1 <-> curPoint2;

        SELECT INTO curAxis1 AxisID
        FROM traffic.Road_segments
        WHERE Road_segmentID = curSegment1;

        SELECT INTO curAxis2 AxisID
        FROM traffic.Road_segments
        WHERE Road_segmentID = curSegment2;

        IF curAxis1 = curAxis2 THEN
            curApplycharge := ''FALSE'';
        ELSE
            curApplycharge := ''TRUE'';
        END IF;
    
```

```

EXECUTE 'INSERT INTO traffic.Trajectories VALUES('' ||
curID2 || ', ' || curAxis2 || ', ' ||
(curDispl*111319) || ', ' ||
(curDispl*111*3600)/(date_part('second',
age(curTimestamp2, curTimestamp1))) || ', ' ||
curApplycharge || ', ' || quote_literal(curTimestamp2)
|| ')'';

END LOOP;

CLOSE curPast2Positions;

RETURN 1;

END;
' LANGUAGE 'plpgsql';

```

B. Σύναρτηση computeLoad:

```

CREATE OR REPLACE FUNCTION computeLoad(timestamp) RETURNS
INTEGER AS '
DECLARE
    curAxisID integer;
    curVehiclescount integer;
    curTimestamp timestamp;
    curLoadcoefficient double precision;
    curLoad double precision;
    curCharge double precision;
    curLoadclass char(1);
    rowResult traffic.Charges%ROWTYPE;

    curAxesload CURSOR (ts timestamp) IS
        SELECT T.AxisID, COUNT(T.VehicleID) AS Vehicles_count,
        A.Load_coefficient
        FROM traffic.Trajectories AS T [RANGE BY '15 SECONDS'
        SLIDE BY '15 SECONDS'], traffic.Road_axes A
        WHERE T.tcqtime = ts
        AND T.AxisID = A.AxisID
        GROUP BY T.AxisID, A.Load_coefficient
        HAVING COUNT(T.VehicleID) > 0;

BEGIN

    OPEN curAxesload($1);

    LOOP
        FETCH curAxesload INTO curAxisID, curVehiclescount,
        curLoadcoefficient;

        IF NOT FOUND THEN
            EXIT;
        END IF;

        curLoad := curVehiclescount * curLoadcoefficient;

        SELECT Charge, Load_class INTO curCharge, curLoadclass
        FROM traffic.Charges
        WHERE curLoad >= Load_min AND curLoad <= Load_max

```

```

AND SUBSTRING($1,12,8) BETWEEN Start_time AND End_time;

IF curCharge IS NULL OR curLoadClass IS NULL THEN
curCharge := 0;
curLoadClass := 'A';
END IF;

EXECUTE 'INSERT INTO traffic.Roadaxes_load VALUES('
|| curAxisID || ',' || curVehiclescount || ',' ||
curLoad || ',' || curCharge || ',' ||
quote_literal(curLoadClass) || ',' ||
quote_literal($1) || ')';

END LOOP;

CLOSE curAxesload;

RETURN 1;

END;
' LANGUAGE 'plpgsql';

```

Γ. Συνάρτηση computeCharge:

```

CREATE OR REPLACE FUNCTION computeCharge(timestamp) RETURNS
INTEGER AS '
DECLARE
    curVehicleID char(7);
    curCharge double precision;

    curChargecongestion CURSOR (ts timestamp) IS
    SELECT T.VehicleID, L.Charge
    FROM (SELECT * FROM traffic.Roadaxes_load WHERE tcqtime =
ts) AS L, (SELECT * FROM traffic.Trajectories WHERE
tcqtime = ts) AS T
    WHERE T.AxisID = L.AxisID
    AND T.Apply_charge = TRUE;

BEGIN

    OPEN curChargecongestion($1);

    LOOP
        FETCH curChargecongestion INTO curVehicleID, curCharge;

        IF NOT FOUND THEN
            EXIT;
        END IF;

        IF curCharge > 0 THEN
            EXECUTE 'INSERT INTO traffic.Congestion_charges
VALUES(' || curVehicleID || ',' || curCharge ||
',' || quote_literal($1) || ')';
        END IF;

    END LOOP;

    CLOSE curChargecongestion;

```



```

RETURN 1;

END;
' LANGUAGE 'plpgsql';

```

Δ. Συνάρτηση `processData`:

```

CREATE OR REPLACE FUNCTION processData(timestamp) RETURNS
INTEGER AS '
BEGIN
    EXECUTE 'SELECT createTrajectory('' || quote_literal($1)
    || '')'';
    EXECUTE 'SELECT computeLoad('' || quote_literal($1) ||
    '')'';
    EXECUTE 'SELECT computeCharge('' || quote_literal($1) ||
    '')'';

    RETURN 1;
END;
' LANGUAGE 'plpgsql';

```

❖ Κώδικας Java

Αρχείο `TrafficProcessing.java`:

```

//Title: TrafficProcessing.java
//Authors: Kostas Patroumpas, Evi Kefallinou
//Description: Main class and entry point to the stream
processing application.
//      It connects to TelegraphCQ to create basic
streams and periodically update data.
//Created on: 01/04/2005
//Last modified: 11/06/2010

import java.io.*;
import java.sql.*;
import jcifs.smb.NtlmPasswordAuthentication;
import jcifs.smb.SmbFile;
import jcifs.smb.SmbFileOutputStream;

public class TrafficProcessing
{
    //Basic application parameters (read from INI file)
    String rate, applDir, geodataDir, streamdataDir, tempDir,
streamFile;

    int ratePeriod, refreshPeriod, sleepTime;
    long startTime;

    java.sql.Connection conn1;
    java.sql.Connection conn2;

    Timestamp ts;

    //Handle JDBC connection to TelegraphCQ
    java.sql.Connection conTCQ;
    Connect connect = new Connect();

```

```

//-----

//TrafficProcessing constructor
public TrafficProcessing()
{

    //Read application settings and parameters from the INI
file
    readINI();

    sleepTime = refreshPeriod;

    //Test connection to TelegraphCQ
    conTCQ = connect.connectTCQ();
System.out.println("Starting connection...");
    //Create basic streams that maintain incoming locations,
trajectories and traffic data
    execSQLcommand("DROP STREAM traffic.Locations");
    execSQLcommand("CREATE STREAM traffic.Locations
(VehicleID CHAR(7), Position POINT, SegmentID INTEGER, t
INTEGER, tcqtime TIMESTAMP TIMESTAMPCOLUMN) TYPE ARCHIVED");
    execSQLcommand("ALTER STREAM traffic.Locations ADD
WRAPPER trafficwrapper");

    execSQLcommand("DROP STREAM traffic.Trajectories");
    execSQLcommand("CREATE STREAM traffic.Trajectories
(VehicleID CHAR(7), AxisID INTEGER, Displacement DOUBLE
PRECISION, Speed DOUBLE PRECISION, Apply_charge BOOLEAN,
tcqtime TIMESTAMP TIMESTAMPCOLUMN) TYPE ARCHIVED");

    execSQLcommand("DROP STREAM traffic.Roadaxes_load");
    execSQLcommand("CREATE STREAM traffic.Roadaxes_load
(AxisID INTEGER, Vehicle_count INTEGER, Load DOUBLE PRECISION,
Charge DOUBLE PRECISION, Load_class CHAR(1), Tcqtime TIMESTAMP
TIMESTAMPCOLUMN) TYPE ARCHIVED");

    execSQLcommand("DELETE FROM traffic.Congestion_charges");

    System.out.println("Basic streams have been created in
TelegraphCQ");
/*
    //Create a JDBC connection to be used for a CQ:
Q5_axesload
    Connect connect1;           //Connection object to
TelegraphCQ

    //Create a new connection to TelegraphCQ
    connect1 = new Connect();File myfile = new
File(System.getProperty("user.dir") + "/.gvfs/" + share + " on
" + server + "/audit.txt");
    conn1 = connect1.connectTCQ();

    declareCursor(conn1, "DECLARE Q5_axesload CURSOR FOR
SELECT AxisID, AVG(Load) AS Axis_load, wtime(*) AS TS
FROM traffic.Roadaxes_load [RANGE BY '1 MINUTE' SLIDE BY
'15 SECONDS' START AT '2010-01-12 12:01:00'] GROUP BY AxisID
ORDER BY AVG(Load) DESC LIMIT 5");

System.out.println("Cursor1 declared...");

*/

```

```

        //Create a JDBC connection to be used for a CQ:
Q10_axesvehspeeds
        Connect connect2;          //Connection object to
TelegraphCQ

        //Create a new connection to TelegraphCQ
connect2 = new Connect();
conn2 = connect2.connectTCQ();

        declareCursor(conn2, "DECLARE Q10_axesvehspeeds CURSOR
FOR SELECT AxisID, avg(Speed) AS Average_speed, wtime(*) AS TS
FROM traffic.Trajectories [RANGE BY '1 MINUTE' SLIDE BY '15
SECONDS' START AT '2010-01-12 12:01:00'] GROUP BY AxisID ORDER
BY avg(Speed) DESC LIMIT 10");

System.out.println("Cursor2 declared...");

        try{
                //Push streaming data into TelegraphCQ
                Runtime r1 = Runtime.getRuntime();

                String cmd1[] = {"gnome-terminal", "--zoom", "0.6",
"-e", "--command", "sh -c 'cat "+streamdataDir+"/"+streamFile+"
| "+streamdataDir+"/sliderate.pl localhost 5533
trafficwrapper,traffic.Locations "+rate + " " +ratePeriod
+";bash'"};

System.out.println(cmd1[5]);
                Process p1=r1.exec(cmd1);

                }
                catch(Exception e)
                { e.printStackTrace(); };

                //Periodically call functions to update traffic data
                ProcRefresh proc;
                proc = new ProcRefresh();

        } //End of constructor

        //-----

        //Execute a DDL command in TelegraphCQ (used to create base
streams during initialisation)
        public void execSQLcommand(String strSQL)
        {
                java.sql.Connection conn;
                Connect connect;          //Connection object to
TelegraphCQ

                //Create a new connection to TelegraphCQ
                connect = new Connect();
                conn = connect.connectTCQ();
                if (conn == null)
                        System.out.println("Cannot establish connection to
TelegraphCQ!");

                //Attempt to submit this SQL command to TelegraphCQ
                try
                {

```

```

        java.sql.Statement s;
        s = conn.createStatement();
        s.execute(strSQL);
        conn.close();
    }
    catch (Exception e)
    {
        System.out.println(strSQL + "\n" + " : Command could
not be executed in TelegraphCQ. Please check again its SQL
syntax.");
        e.printStackTrace();
    }
}

//-----

//Execute a DDL command in TelegraphCQ (used to create base
streams during initialisation)
public void declareCursor(java.sql.Connection conn, String
strSQL)
{
    if (conn == null)
        System.out.println("Cannot establish connection to
TelegraphCQ!");

    //Attempt to submit this SQL command to TelegraphCQ
    try
    {
        java.sql.Statement s;
        s = conn.createStatement();
        s.executeUpdate("BEGIN");
        s.executeUpdate(strSQL);
    }
    catch (Exception e)
    {
        System.out.println(strSQL + "\n" + " : Command could
not be executed in TelegraphCQ. Please check again its SQL
syntax.");
        e.printStackTrace();
    }
}

//-----

//Execute a DML command in TelegraphCQ (used to select data
from a fetch)
public void fetchData(java.sql.Connection conn, String
strSQL)
{
    if (conn == null)
        System.out.println("Cannot establish connection to
TelegraphCQ!");

    //Attempt to submit this SQL command to TelegraphCQ
    try
    {

        //Connection to an XML file in Web server (connection
through SAMBA)
        String user = "kerk:kerk";

```

```

        NtlmPasswordAuthentication auth = new
NtlmPasswordAuthentication(user);
        String path =
"smb://gaia/kerk/top10_axesvehspeeds.xml";
        SmbFile sFile = new SmbFile(path, auth);
        SmbFileOutputStream sfos = new
SmbFileOutputStream(sFile);
        String resXML = "<Items>";

        java.sql.Statement s;
        s = conn.createStatement();

        java.sql.ResultSet r ;
        r= s.executeQuery(strSQL);

        while (r.next())
        {
            //Type indicative attribute values
            resXML = resXML + "<Result AxisID='" +
r.getString(1) + "' Average_speed='" + r.getString(2) + "'
TS='" + r.getString(3) + "' />";
        }
        resXML = resXML + "</Items>";

        //Write XML stream into output file
        sfos.write(resXML.getBytes());
//        System.out.println("Batch completed!");

    }
    catch (Exception e)
    {
        System.out.println(strSQL + "\n" + " : Command could
not be executed in TelegraphCQ. Please check again its SQL
syntax.");
        e.printStackTrace();
    }
}

//-----

//Read INI file and initialize basic application parameters
void readINI()
{
    String[] vParam;
    try
    {
        LineNumberInputStream aLNIS;
        aLNIS = new LineNumberInputStream(new
FileInputStream("TrafficProcessing.ini"));
        DataInputStream s = new DataInputStream(aLNIS);
        String myLine;
        while ( (myLine = s.readLine()) != null)
        {
            if(myLine.equals("[STREAMDATA]"))
            {
                myLine = s.readLine();
                vParam = myLine.split("=");
                streamdataDir = vParam[1];
            }
            else if(myLine.equals("[TEMP]"))

```

```

        {
            myLine = s.readLine();
            vParam = myLine.split("=");
            tempDir = vParam[1];
        }
        else if(myLine.equals("[STREAMFILE]"))
        {
            myLine = s.readLine();
            vParam = myLine.split("=");
            streamFile = vParam[1];
        }
        else if(myLine.equals("[RATE]"))
        {
            myLine = s.readLine();
            vParam = myLine.split("=");
            rate = vParam[1];
        }
        else if(myLine.equals("[PERIOD]"))
//Refresh period for calling functions
        {
            myLine = s.readLine();
            vParam = myLine.split("=");
            ratePeriod = Integer.parseInt(vParam[1]);
            refreshPeriod = ratePeriod * 1000;
        }
        else if(myLine.equals("[TIMESTAMP]"))
//Initial timestamp of the incoming stream
        {
            myLine = s.readLine();
            vParam = myLine.split("=");
            ts =
java.sql.Timestamp.valueOf(vParam[1]);
        }
    }
    catch (IOException e) // catch possible io errors
from readLine()
    {
        System.out.println("Got an IOException error!");
        e.printStackTrace();
    }
} //end of ReadINI()

//-----

//This class periodically executes certain SQL commands
class ProcRefresh implements Runnable
{
    private Thread thread;

    //Constructor
    public ProcRefresh ()
    {
        //Start a new thread that will take care of the
application refresh
        thread = new Thread(this);
        thread.start();
    }

    //Executes when the thread is activated
    public void run()

```

```

        {
            //Run periodically
            while(true)
            {
                try
                {
                    thread.sleep(sleepTime);
                    startTime = System.currentTimeMillis();

                    //Execute a batch of SQL commands (essentially
                    run functions that feed traffic volumes)
                    execSQLcommand("SELECT
processData('"+ts+"'");

                    java.util.Date now = new java.util.Date(); //
                    Get clock time
                    System.out.println(now + " Data updated for
                    timestamp " + ts);

                    //Update timestamp value
                    ts.setTime(ts.getTime() + refreshPeriod);

                    java.sql.Timestamp sts =
                    Timestamp.valueOf("2010-01-12 12:02:00");

                    //Execute FETCH SQL command
                    if (ts.after(sts))
                    {
                        //
                        fetchData(conn1, "FETCH FORWARD 5 IN
                        Q5_axesload");
                        //
                        System.out.println(now + "Data fetched for
                        query Q5_axes_load");
                        fetchData(conn2, "FETCH FORWARD 10 IN
                        Q10_axesvehspeeds");
                        System.out.println(now + "Data fetched for
                        query Q10_axes_veh_speeds");
                    }
                    //Adjust sleep time for next iteration
                    sleepTime = refreshPeriod -
                    (int)(System.currentTimeMillis() - startTime);
                    if (sleepTime < 0)
                        sleepTime = 0;
                }
                catch(Exception e)
                {
                    System.out.println("Thread problem!!!");
                    //
                    execSQLcommand("CLOSE Q5_axesload");
                    execSQLcommand("CLOSE Q10_axesvehspeeds");
                    execSQLcommand("COMMIT");
                }
            }
        } //end of run

    } //end of ProcRefresh

//-----

//Main simply creates a new instance of this application
public static void main(String args[])
{

```

```

        System.out.println("Starting application for processing
traffic volumes");
        System.out.println("Source code partially taken from
Trajectory Stream Visualizer version 1.0");
        System.out.println("Development (c) 2004-2010 Kostas
Patroumpas, Evi Kefallinou");
        TrafficProcessing mainApp = new TrafficProcessing();
    }
} //end of class TrafficProcessing

```

❖ Κώδικας ρουτινών PHP

A. Αρχείο `Traffic_lines.php`:

```

<?php

//Max execution time (in seconds)
set_time_limit(240);

// Connecting, selecting database
include 'TrafficDB_conf.php';

//Get important parameters submitted with the web form
$timeSleep = 15; //Seconds to wait before resuming
(parameter in the php call)

//Start a new transaction block for this continuous query
$queryb = "BEGIN WORK;";
$resultb = pg_query($queryb) or die('Query failed: ' .
pg_last_error());

//Construct the continuous SQL query within a cursor as
prescribed by TelegraphCQ
$querys = "SELECT R.Segment, L.AxisID, L.Load_class
FROM traffic.Road_segments AS R,
traffic.Roadaxes_load AS L [RANGE BY '15 SECONDS' SLIDE BY '15
SECONDS']
WHERE R.AxisID=L.AxisID";

$queryd = "DECLARE Q1_axes CURSOR FOR " . $querys;
$resultd = pg_query($queryd) or die('Query failed: ' .
pg_last_error());

//*****Requires CURSOR!!!!!! FETCH*****
//Repetitive query evaluation (until time limit is exceeded)
while (true)
{
    $bgttime = time();

    // Start KML file, create parent node
    $dom = new DOMDocument('1.0','UTF-8');

    //Create the root KML element and append it to the
Document
    $node = $dom->createElementNS('http://earth.google.com/kml/2.1','kml');
    $parNode = $dom->appendChild($node);
}

```



```

$dNode = $dom->createElement('Document');
$docNode = $parNode->appendChild($dNode);

//*****CREATE ADDITIONAL STYLES*****
$sNode = $dom->createElement('Style');
$sNode->setAttribute("id", "classA");

$styleNode = $docNode->appendChild($sNode);
$node = $dom->createElement("LineStyle");
$lnNode = $styleNode->appendChild($node);

$colorNode = $dom->createElement('color', "7d008000");
$widthNode = $dom->createElement('width', "9");

$lnNode->appendChild($colorNode);
$lnNode->appendChild($widthNode);

$sNode = $dom->createElement('Style');
$sNode->setAttribute("id", "classB");

$styleNode = $docNode->appendChild($sNode);
$node = $dom->createElement("LineStyle");
$lnNode = $styleNode->appendChild($node);

$colorNode = $dom->createElement('color', "5014F0D2");
$widthNode = $dom->createElement('width', "9");

$lnNode->appendChild($colorNode);
$lnNode->appendChild($widthNode);

$sNode = $dom->createElement('Style');
$sNode->setAttribute("id", "classC");

$styleNode = $docNode->appendChild($sNode);
$node = $dom->createElement("LineStyle");
$lnNode = $styleNode->appendChild($node);

$colorNode = $dom->createElement('color', "5014F0FF");
$widthNode = $dom->createElement('width', "9");

$lnNode->appendChild($colorNode);
$lnNode->appendChild($widthNode);

$sNode = $dom->createElement('Style');
$sNode->setAttribute("id", "classD");

$styleNode = $docNode->appendChild($sNode);
$node = $dom->createElement("LineStyle");
$lnNode = $styleNode->appendChild($node);

$colorNode = $dom->createElement('color', "7d4080ff");
$widthNode = $dom->createElement('width', "9");

$lnNode->appendChild($colorNode);
$lnNode->appendChild($widthNode);

```

```

    $sNode = $dom->createElement('Style');
    $sNode->setAttribute("id", "classE");

    $styleNode = $docNode->appendChild($sNode);
    $node = $dom->createElement("LineStyle");
    $lnNode = $styleNode->appendChild($node);

    $colorNode = $dom->createElement('color', "ff0267fe");
    $widthNode = $dom->createElement('width', "9");

    $lnNode->appendChild($colorNode);
    $lnNode->appendChild($widthNode);

    $sNode = $dom->createElement('Style');
    $sNode->setAttribute("id", "classF");

    $styleNode = $docNode->appendChild($sNode);
    $node = $dom->createElement("LineStyle");
    $lnNode = $styleNode->appendChild($node);

    $colorNode = $dom->createElement('color', "501400A0");
    $widthNode = $dom->createElement('width', "9");

    $lnNode->appendChild($colorNode);
    $lnNode->appendChild($widthNode);
//*****END OF ADDITIONAL STYLES*****

    //Create a Folder element and append it to the KML
element
    $fnode = $dom->createElement('Folder');
    $folderNode = $docNode->appendChild($fnode);

    while((time() - $bgtime) < $timeSleep)
    {
        //Fetch resulting tuples one by one
        $qFetch = "FETCH FORWARD 1 IN Q1_axes";
        $resFetch = pg_query($qFetch); // or die('Query
failed: ' . pg_get_notify($dbconn));
        // if (!!$resFetch) &&
if ($row = pg_fetch_array($resFetch, null, PGSQL_ASSOC))
        {
            //Export record as a node into the KML file
            $node = $dom->createElement("Placemark");
            $resultNode = $folderNode-
>appendChild($node);
            $resultNode->setAttribute("id",
$row['axisid']);

            //Choose the style corresponding to this line
            $curStyleNode = $dom-
>createElement('styleUrl', "#class" . $row['load_class']);
            $resultNode->appendChild($curStyleNode);

            //Create a linestring element
            $lineNode = $dom-
>createElement('LineString');
            $resultNode->appendChild($lineNode);

            $seg = $row['segment'];
            $seg = str_replace(' ', ', ', $seg);

```

```

        $seg = str_replace('(', ' ', $seg);
        $seg = str_replace(')', ' ,0', $seg);

        $coorNode = $dom-
>createElement('coordinates', $seg);
        $lineNode->appendChild($coorNode);
    }

    //Free resultset
    pg_free_result($resFetch);
}

//Save the entire KML file (overwrite previous one)
$dom->save('traffic_lines.kml');
}

//Close cursor
$qDrop = "CLOSE Q1_axes;";
$rDrop = pg_query($qDrop) or die('Query failed: ' .
pg_last_error());

//Commit transaction block
$queryc = "COMMIT WORK;";
$resultc = pg_query($queryc) or die('Query failed: ' .
pg_last_error());

//Closing database connection
pg_close($dbconn);

?>

```

B. Αρχείο top10_vehiclescharges.php:

```

<?php

// Connecting, selecting database
include 'TrafficDB_conf.php';

$query = "SELECT VehicleID, MAX(ts) AS maxtime,
SUM(Congestion_charge) AS Congestion_charge
FROM traffic.Congestion_charges
GROUP BY VehicleID
ORDER BY SUM(Congestion_charge) DESC LIMIT 10";

$result = pg_query($query) or die('Query failed: ' .
pg_last_error());

// Start XML file, create parent node
$dom = new DOMDocument('1.0','UTF-8');

//Create the root Items element and append it to the
Document
$node = $dom->createElement("Items");
$itemNode = $dom->appendChild($node);

while ($row = pg_fetch_array($result, null, PGSQL_ASSOC))
{
    //Export record as a node into the XML file
    $node = $dom->createElement("Result");
}

```

```

        $resultNode = $itemsNode->appendChild($node);
        $resultNode->setAttribute("VehicleID",
$row['vehicleid']);
        $resultNode->setAttribute("Congestion_charge",
$row['congestion_charge']);
        $resultNode->setAttribute("ts", $row['maxtime']);
    }

    //Save the entire XML file
    $dom->save('top10_vehcharges.xml');

    // Free resultset
    pg_free_result($result);

?>

```

❖ Κώδικας HTML/Javascript

Αρχείο TrafficLoadMap.html:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<!-- Title: AthensStreamingTraffic1.html -->
<!-- Description: Draw an external KML file on GoogleMaps. -->
<!-- Created by Kostas Patroumpas -->
<!-- Date: 8/1/2010 -->
<!-- Last modified: 16/7/2010 -->
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:v="urn:schemas-microsoft-com:vml">
    <HEAD>
        <meta content="text/html; charset=utf-8" http-
equiv="content-type">
        <script type="text/javascript"
src="http://gaia.dbnet.ece.ntua.gr/TSV/scripts/jquery-
1.3.2.min.js"></script>
        <script
src="http://maps.google.com/maps?file=api&v=2&key=ABQIAAAA3
3z4DFR3uYZniPtFjtkkDBS17_Fibqpsi4gxnrBL4ptJVteRkRS2fuHnj72RO9bw
Z8g_p96v_Nb7xQ" type="text/javascript"></script>
        <script type="text/javascript">

var map;
var banner;
var curTime;
var geoXml;

//Application initialization
function init()
{
    if (GBrowserIsCompatible())
    {
        map = new GMap2(document.getElementById("map"));
        map.setMapType(G_SATELLITE_MAP);
        map.getDragObject().setDraggableCursor("move");
        map.setCenter(new GLatLng(37.9765, 23.7292),12);
        map.addControl(new GScaleControl());
        map.enableScrollWheelZoom();
    }
}

```

```

        map.disableDoubleClickZoom(); //Disable this
effect
        map.addControl(new GLargeMapControl());
        map.addControl(new GMapTypeControl());
        map.setUIToDefault();

        //Create connection to TelegraphCQ and declare cursors
for continuous queries
        jQuery.get(
'http://gaia.dbnet.ece.ntua.gr/TSV/traffic_lines.php?rand='+ (new
Date()).valueOf() , function ( data ) {});

        // Refresh map function
        function refreshMap(map)
        {
            if (status == "stopped")
window.setTimeout(function(){ refreshMap(map)},5000);

            if (geoXml != null) {map.removeOverlay(geoXml);}
            geoXml = new
GGeoXml("http://gaia.dbnet.ece.ntua.gr/TSV/traffic_lines.kml?ra
nd="+ (new Date()).valueOf());
            map.addOverlay(geoXml);

            jQuery.get(
'http://gaia.dbnet.ece.ntua.gr/TSV/top10_vehiclescharges.php?ra
nd='+ (new Date()).valueOf() , function ( data )
                { //Create a table

                    writeLoadTable("top10_vehcharges.xml","tabLoad1");
                });

            writeLoadTable("http://gaia.dbnet.ece.ntua.gr/TSV/kerk/to
p10_axesvehspeeds.xml","tabLoad2");

            writeTimestamp("http://gaia.dbnet.ece.ntua.gr/TSV/kerk/to
p10_axesvehspeeds.xml");

            //Reloads map every second
            window.setTimeout(function(){
refreshMap(map)},15000);
            }
            refreshMap(map);
        }
    }

function writeTimestamp(xmlFile)
{
if (window.XMLHttpRequest)
    {
        xmlhttp=new XMLHttpRequest();
    }
else // Internet Explorer 5/6
    {
        xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
xmlhttp.open("GET",xmlFile+"?RANDOM="+Math.random(),false);
xmlhttp.send("");
}

```

```

xmlDoc=xhttp.responseXML;

var xmlTable=xmlDoc.getElementsByTagName("Result");
//Get timestamp value from table
curTime = xmlTable[1].getAttribute("TS");
document.getElementById("txtTime").innerText = "Time: " +
curTime;

}

function writeLoadTable(xmlFile, tabDiv)
{
if (window.XMLHttpRequest)
{
xhttp=new XMLHttpRequest();
}
else // Internet Explorer 5/6
{
xhttp=new ActiveXObject("Microsoft.XMLHTTP");
}
xhttp.open("GET",xmlFile+"?RANDOM="+Math.random(),false);
xhttp.send("");
xmlDoc=xhttp.responseXML;

var xmlTable=xmlDoc.getElementsByTagName("Result");

//Create a table structure that will be filled with
streaming values
var res1 = document.getElementById(tabDiv);

//Remove all attribute values from the respective
tabular spreadsheet
while(res1.hasChildNodes())
{ res1.removeChild(res1.lastChild); }

var tab = document.createElement('table');
var tbo = document.createElement('tbody');
var row, cell;

//Iterate through all actual records and display their
contents
for (var i = 0; i < xmlTable.length; i++)
{
//First table row is used to display attribute
(field) names
row = document.createElement('tr');
//Iterate through all columns of this record
for(var j = 0; j < xmlTable[i].attributes.length
;j++)
{ //Create a new cell for each attribute value
cell = document.createElement('td');

cell.appendChild(document.createTextNode(xmlTable[i].getA
ttribute(xmlTable[i].attributes[j].nodeName)));
row.appendChild(cell);
}
tbo.appendChild(row);
}
tab.appendChild(tbo);
tab.setAttribute("border", "2"); //Set cell border

```


❖ Κώδικας αρχείου τύπου Perl

Αρχείο sliderate.pl:

```
#!/usr/bin/perl
use FileHandle;
use Socket;

$usage = "source.pl host port sourcename
  where
    host = The name or IP address of the TelegraphCQ host to
connect to
    port = The port number on which the TCQ host is listening
    wrappername = The name of the TCQ wrapper to push data into
";

if ($#ARGV<3) {
    print "Missing one or more parameters.\n Usage: $usage";
    exit;
}
$host = $ARGV[0];
$port = $ARGV[1];
$sourcename = $ARGV[2];

$numitems= $ARGV[3];
$slot= $ARGV[4];

for $f (@ARGV) {
    $i++;
    if ($f eq "-h" || $f eq "--help") { print "Usage: $usage";
    exit; }
}
#open the socket
$iaddr = inet_aton($host);
$paddr = sockaddr_in($port,$iaddr);
$proto = getprotobyname('tcp');
socket(SOCK, PF_INET, SOCK_STREAM, $proto) || die "Couldn't
open socket: $!";
connect(SOCK, $paddr) || die "connect: $!";
autoflush SOCK,1;
$namestr =
sprintf("%4d%s\x0",length($sourcename)+1,$sourcename);
print SOCK $namestr;

$starttime=time;
$count=0;
while (<STDIN>) {
    $str = $_;
    # print($str);
    print(SOCK $str);
    $count++;
    if (($count % $numitems) eq 0)
        { if (($count % (5 * $numitems)) eq 0) {print($str); }
        $currtime=time;
        $diff=$starttime + $slot - $currtime;
        if ($diff gt 0) { sleep $diff; }
        $starttime= time;
        }
}
close(SOCK);
```


Παράρτημα Δ: Αποτελέσματα αρχείων KML και XML

A. Αρχείο top10_vehcharges.xml:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <Items>
-   <Result VehicleID="1468" Congestion_charge="45.5" ts="2010-01-12 12:11:00" />
-   <Result VehicleID="9779" Congestion_charge="42.5" ts="2010-01-12 12:11:00" />
-   <Result VehicleID="9343" Congestion_charge="41" ts="2010-01-12 12:10:30" />
-   <Result VehicleID="1460" Congestion_charge="40.5" ts="2010-01-12 12:11:15" />
-   <Result VehicleID="5229" Congestion_charge="39.5" ts="2010-01-12 12:11:15" />
-   <Result VehicleID="9743" Congestion_charge="38.5" ts="2010-01-12 12:11:00" />
-   <Result VehicleID="3308" Congestion_charge="36.5" ts="2010-01-12 12:09:30" />
-   <Result VehicleID="7700" Congestion_charge="36.5" ts="2010-01-12 12:11:15" />
-   <Result VehicleID="6616" Congestion_charge="36" ts="2010-01-12 12:11:00" />
-   <Result VehicleID="7072" Congestion_charge="36" ts="2010-01-12 12:10:30" />
- </Items>
```

B. Αρχείο traffic_lines.kml:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <kml xmlns="http://earth.google.com/kml/2.1">
- <Document>
- <Style id="classA">
- <LineStyle>
- <color>7d008000</color>
- <width>9</width>
- </LineStyle>
- </Style>
- <Style id="classB">
- <LineStyle>
- <color>5014F0D2</color>
- <width>9</width>
- </LineStyle>
- </Style>
- <Style id="classC">
- <LineStyle>
- <color>5014F0FF</color>
- <width>9</width>
- </LineStyle>
- </Style>
- <Style id="classD">
- <LineStyle>
- <color>7d4080ff</color>
- <width>9</width>
- </LineStyle>
- </Style>
- <Style id="classE">
- <LineStyle>
```

```

    <color>ff0267fe</color>
    <width>9</width>
  </LineStyle>
</Style>
- <Style id="classF">
- <LineStyle>
- <color>501400A0</color>
  <width>9</width>
  </LineStyle>
</Style>
- <Folder>
- <Placemark id="2">
  <styleUrl>#classA</styleUrl>
- <LineString>
  <coordinates>23.8083871655425,38.0735179396314,0
23.8084174417295,38.0736576862841,0</coordinates>
  </LineString>
</Placemark>
- <Placemark id="3">
  <styleUrl>#classC</styleUrl>
- <LineString>
  <coordinates>23.8073354544812,38.0703752835181,0
23.8072313659359,38.0705328357016,0</coordinates>
  </LineString>
</Placemark>
- <Placemark id="4">
  <styleUrl>#classA</styleUrl>
- <LineString>
  <coordinates>23.7425070844869,38.0429560466188,0
23.7434283786813,38.0440891628533,0
23.7436765936809,38.044400644653,0
23.7440336607818,38.044852060795,0
23.7443754962613,38.0452809107993,0
23.7447088350792,38.0456962221901,0
23.7452241492805,38.0463237270097,0
23.7454507788373,38.046599107141,0
23.745821557175,38.0470054835956,0
23.7463138019047,38.0476149081258,0</coordinates>
  </LineString>
</Placemark>
- <Placemark id="4">
  <styleUrl>#classA</styleUrl>
- <LineString>
  <coordinates>23.7463138019047,38.0476149081258,0
23.7465838444085,38.0479038988324,0
23.7468719911965,38.048210953407,0</coordinates>
  </LineString>
</Placemark>
- <Placemark id="4">
  <styleUrl>#classA</styleUrl>
- <LineString>
  <coordinates>23.7468890248852,38.0482290155991,0
23.7472259840419,38.0485316679937,0
23.7474471924225,38.0487304240903,0
23.7475575880308,38.048838814291,0
23.7478217242106,38.0490872320135,0
23.7479513075305,38.0492091821326,0
23.7481533023152,38.0494033893101,0
23.7483737726373,38.0496111548167,0
23.7487397961276,38.0499589288059,0

```

```

23.748850287856,38.0500402797107,0
23.7491743156587,38.0502753031518,0</coordinates>
  </LineString>
</Placemark>
- <Placemark id="4">
  <styleUrl>#classA</styleUrl>
- <LineString>
  <coordinates>23.74211487352,38.0424775103377,0
23.7425070844869,38.0429560466188,0</coordinates>
  </LineString>
</Placemark>
- <Placemark id="4">
  <styleUrl>#classA</styleUrl>
- <LineString>
  <coordinates>23.7468719911965,38.048210953407,0
23.7468890248852,38.0482290155991,0</coordinates>
  </LineString>
</Placemark>
- <Placemark id="4">
  <styleUrl>#classA</styleUrl>
- <LineString>
  <coordinates>23.7491743156587,38.0502753031518,0
23.7495928296133,38.050690782103,0</coordinates>
  </LineString>
</Placemark>
</Folder>
</Document>
</kml>

```

Γ. Αρχείο top10_axesvehspeeds.xml:

```

- <Items>
  <Result AxisID="1467" Average_speed="50.5096779222774"
TS="2010-01-12 12:02:00" />
  <Result AxisID="4960" Average_speed="43.8078646984168"
TS="2010-01-12 12:02:00" />
  <Result AxisID="3237" Average_speed="41.0596544658487"
TS="2010-01-12 12:02:00" />
  <Result AxisID="4939" Average_speed="41.0368578428062"
TS="2010-01-12 12:02:00" />
  <Result AxisID="5542" Average_speed="40.7219052096205"
TS="2010-01-12 12:02:00" />
  <Result AxisID="7603" Average_speed="40.4078389997145"
TS="2010-01-12 12:02:00" />
  <Result AxisID="1470" Average_speed="40.3928730141454"
TS="2010-01-12 12:02:00" />
  <Result AxisID="4898" Average_speed="40.2703303704951"
TS="2010-01-12 12:02:00" />
  <Result AxisID="5547" Average_speed="40.1226208210578"
TS="2010-01-12 12:02:00" />
  <Result AxisID="1159" Average_speed="39.6781224751161"
TS="2010-01-12 12:02:00" />
</Items>

```


Βιβλιογραφία - Αναφορές

❖ Άρθρα για ρεύματα δεδομένων και κινούμενα αντικείμενα

[ACC+03b] D.J. Abadi, D. Carney, U. Cetintemel, M. Cherniack, C. Convey, S. Lee, M. Stonebraker, N. Tatbul, and S. Zdonik. Aurora: A New Model and Architecture for Data Stream Management. *VLDB Journal*, 12(2):120-139, August 2003.

[ABW03] A. Arasu, S. Babu, and J. Widom. An Abstract Semantics and Concrete Language for Continuous Queries over Streams and Relations. In *Proceedings of the 9th International Conference on Data Base Programming Languages (DBPL'03)*, Berlin, Germany, September 2003.

[ACG+04] A. Arasu, M. Cherniack, E. Galvez, D. Maier, A. S. Maskey, E. Ryzkina, M. Stonebraker, and R. Tibbetts. Linear Road: A Stream Data Management Benchmark. In *Proceedings of the 30th International Conference on Very Large Data Bases (VLDB)*, pp.480-491, Toronto, Canada, September 2004.

[BBD+02] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. Models and Issues in Data Stream Systems. In *Proceedings of the 21st ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'02)*, pp.1-16, Madison, Wisconsin, May 2002.

[CCD+03] S. Chandrasekaran, O. Cooper, A. Deshpande, M.J.Franklin, J.M. Hellerstein, W. Hong, S. Krishnamurthy, S.R. Madden, V. Raman, F. Reiss, and M.A. Shah. TelegraphCQ: Continuous Dataflow Processing for an Uncertain World. In *Proceedings of the First Biennial Conference on Innovative Data Systems Research (CIDR)*, Asilomar, California, January 2003.

[CJSS03] C. Cranor, T. Johnson, O. Spataschek, and V. Shkapenyuk. Gigascope: A Stream Database for Network Applications. In *Proceedings of the 22nd ACM SIGMOD International Conference on Management of Data*, pp. 647-651, San Diego, California, USA June 2003.

[CQL09] Oracle® CEP. CQL Language Reference 11g Release1 (v.11.1.1). E12048-01. May 2009

[EGSV98] M. Erwig, R.H. Gutting, M.M. Schneider, and M. Vazirgiannis. Abstractand Discrete Modelling of Spatio-Temporal Data Types. In *Proceedings of the 6th ACM Symposium on Geographic Information Systems*, Washington DC, pp.131-136, November 1998.

- [Fre08] Η. Φρέντζος. Διδακτορική διατριβή: «Διαχείριση Δεδομένων Τροχιών Κινούμενων Αντικειμένων». Αθήνα, Ιούλιος 2008.
- [GO03] L. Golab, and M. Tamer Ozsu. Issues in Data Stream Management. *ACM SIGMOD Record*, 32(2):5-14, June 2003.
- [Gut94] R. H. Gutting. An Introduction to Spatial Database Systems. *VLDB Journal*, Special Issue on Spatial Database Systems, 3 (4): 357-399, October 1994.
- [GG98] V. Gaede, and O. Gunther. Multidimensional Access Methods. *ACM Computing Surveys*, 30 (2): 170-231, 1998.
- [Kol08] Ν. Κολιός. Διπλωματική εργασία: «Διαχείριση Δυναμικών Μεταβολών σε Χωρικά Δεδομένα με Χρήση διαδικτυακών Υπηρεσιών». Αθήνα, Δεκέμβριος 2008.
- [KRG10] Η. Koutsopoulos, M. Rahmani, B. Guc. Real-Time Traffic Information Management using Stream Computing. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 2010.
- [Kyp06] Εμ. Κυπραίος. Διπλωματική εργασία: «Επεξεργασία Ερωτημάτων Διαρκείας σε Ρεύματα Κινούμενων Αντικειμένων». Αθήνα, Ιούλιος 2006.
- [MAHP03] M.F. Mokbel, W.G. Aref, S.E. Hambrusch, and S. Prabhakar. Towards Scalable Location-aware Services: Requirements and Research Issues. In *Proceedings of the 11th ACM International Symposium on Advances in Geographic Information Systems (GIS'03)*, pp. 110-117, New Orleans, Louisiana, USA, November 2003.
- [Pat03] Κ. Πατρούμπας. Μεταπτυχιακή εργασία: «Συστήματα ρευμάτων δεδομένων για κινούμενα αντικείμενα». Αθήνα, Μάιος 2003.
- [Pat04] Κ. Πατρούμπας. Προσδιορισμός Παραθύρων σε Ρεύματα Δεδομένων. Αθήνα, Δεκέμβριος 2004.
- [Pat04a] Κ. Πατρούμπας. Εκτελώντας ερωτήματα διαρκείας με το TelegraphCQ: Πρώτες απόπειρες με κινούμενα αντικείμενα. *Τεχνική αναφορά*, Φεβρουάριος 2004.
- [Pat04b] Κ. Πατρούμπας. Ερωτήματα διαρκείας σε τροχιές κινούμενων αντικειμένων: Πρώτες δοκιμές με το σύστημα STREAM. *Τεχνική αναφορά*, Φεβρουάριος 2004.
- [Pfo02] D. Pfooser. Indexing the Trajectories of Moving Objects. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, June 2002.

[Pot05] Μ. Ποταμιάς. Διπλωματική εργασία: «Συμπύεση Δεδομένων Τροχιάς Κινούμενων Αντικειμένων». Αθήνα, Οκτώβριος 2005.

[PS04] K. Patroumpas and T. Sellis. Managing Trajectories of Moving Objects as Data Streams. In *Proceedings of the 2nd Workshop on Spatio-Temporal Database Management (STDBM'04)*, Toronto, Canada, August 2004.

[PS10] K. Patroumpas and T. Sellis. Maintaining consistent results of continuous queries under diverse window specifications. Article in press *Information Systems*, 2010.

[SCZ05] M. Stonebraker, U. Cetintemel, and S. Zdonik. The 8 Requirements of Real-Time Stream Processing. *ACM SIGMOD Record*, 34(4):42-47, December 2005.

[Ste07] Em. Stefanakis. Geographic Applications and XML. Αθήνα, 2007.

[Ste09] Em. Στεφανάκης. Τεχνολογίες Δημοσιοποίησης Χαρτογραφικού Περιεχομένου στον Παγκόσμιο Ιστό. Αθήνα, Μάρτιος 2009.

[The03] Ύ. Theodoridis. Ten Benchmark Database Queries for Location-based Services. *Computer Journal*, 46(6): 713-725, 2003.

[Tzw05] Η. Τζωρτζακάκης. Διπλωματική εργασία: «Προσδιορισμός Παραθύρων σε Ρεύματα Δεδομένων». Αθήνα, Οκτώβριος 2005.

[WXCJ98] O. Wolfson, B. Xu, S. Chamberlain, L. Jiang. Moving Objects Databases: Issues and Solutions. In *Proceedings of 10th International Conference on Scientific and Statistical Database Management (SSDB 1998)*, pp. 111-122, Capri, Italy, July 1998.

❖ Άρθρα για κυκλοφοριακό φόρτο

[MKR07] T. V. Mathew and K. V. Krishna Rao. Introduction to Transportation Engineering 35.1. NPTEL, May 2007

[Tca10] Transportation Cost and Benefit Analysis II – Congestion Costs. Victoria Transport Policy Institute (www.vtpi.org), April 2010

[Tra06] A. A. Trani. Quick Review of Highway Level of Service. 2006

❖ Διαδικτυακοί τόποι

[wwwOGC] Open Geospatial Consortium: <http://www.opengeospatial.org/standards>

[wwwWMS] Web Map Service: <http://www.opengeospatial.org/standards/wms>

[wwwWFS] Web Feature Service: <http://www.opengeospatial.org/standards/wfs>

[wwwWCS] Web Coverage Service: <http://www.opengeospatial.org/standards/wcs>

[wwwOSG] Open Source Geospatial Foundation: <http://www.osgeo.org>

[wwwAUR] Aurora: <http://www.cs.brown.edu/research/aurora/>

[wwwSTR] Stream: <http://www-db.stanford.edu/stream/>

[wwwTCQ] TelegraphCQ: <http://telegraph.cs.berkeley.edu/telegraphcq/v0.2/>

[wwwPHP] <http://articles.sitepoint.com/print/accessing-postgresql-php>

[wwwWWM] Wikipedia: http://en.wikipedia.org/wiki/Web_mapping

[wwwWGM] Wikipedia: http://en.wikipedia.org/wiki/Google_Maps

[wwwWMG] https://www.e-education.psu.edu/geog863/resources/l3_p4.html

[wwwMCS] <http://kartoweb.itc.nl/webcartography/webmaps/classification.htm>

[wwwWEMA] Wapedia: http://wapedia.mobi/en/Web_mapping

[wwwWOGC] http://en.wikipedia.org/wiki/Open_Geospatial_Consortium

[wwwEMY] http://www.hnms.gr/hnms/greek/forecast/forecast_maps_index_html

[wwwCMAP] CentreMap: <http://centremap.psu.edu/code/mapindex.asp>

[wwwDGR] Spatial Data Infrastructures: <http://www.deegree.org>

[wwwNAFT] Ναυτεμπορική: <http://www.naftemporiki.gr/traffic/>

[wwwKATH] Καθημερινή: <http://e-map.kathimerini.gr/default.aspx>

Γλωσσάριο Όρων

| | |
|--|------------------------------|
| Αβεβαιότητα | Uncertainty |
| Αρχειοδετημένα ρεύματα δεδομένων | Archived data streams |
| Αφήρημένοι τύποι δεδομένων | Abstract data types |
| Βάσεις Δεδομένων Κινούμενων Αντικειμένων | Moving Object Databases |
| Βελτιστοποίηση ερωτημάτων | Query optimization |
| Γλώσσα ερωτημάτων διαρκείας | Continuous Query Language |
| Γραφική διεπαφή χρήστη | Graphical User Interface |
| Δεικτοδότηση – ευρετήρια | Indexing |
| Διαδικτυακός εξυπηρετητής | Web server |
| Εξυπηρετητής δεδομένων | Data server |
| Ερωτήματα διαρκείας | Continuous queries |
| Ερωτήματα στιγμιοτύπου | One-time or Snapshot queries |
| Ερωτήματα συνάθροιση | Aggregation queries |
| Κινούμενα αντικείμενα | Moving objects |
| Κλιμακωσιμότητα | Scalability |
| Κυλιόμενα παράθυρα | Sliding windows |
| Μεριστικά παράθυρα | Partitioned windows |
| Μη αρχειοδετημένα ρεύματα δεδομένων | Unarchived data streams |
| Μη προβλέψιμα ερωτήματα | Ad hoc queries |
| Μηχανή επεξεργασίας ρευμάτων | Spatial Processing Engine |
| Παράθυρα οροσήμευ | Landmark windows |
| Παράθυρα πλειάδων | Tuple – based windows |
| Πελάτης – χρήστης | Client |
| Περιβάλλοντα παραλληλόγραμμα | Bounding box |
| Περιλήψεις δεδομένων | Data synopses or Summaries |
| Ποιότητα υπηρεσιών | Quality of Service QoS |
| Προκαθορισμένα ερωτήματα | Predefined queries |
| Προσαρμοστικότητα τελεστών | Adaptivity of operators |
| Ρεύμα δεδομένων | Data stream |
| Ρεύματα τροχιάς αντικειμένων | Trajectory streams |
| Συστήματα Διαχείρισης Βάσεων Δεδομένων | Database Management Systems |
| Τέλος συμφόρησης | Congestion charge |
| Υλοποιημένες όψεις | Materialized views |
| Υπηρεσία διαδικτυακών γνωρισμάτων | Web Feature |
| Υπηρεσία διαδικτυακών χαρτών | Web Map Service |
| Υπηρεσίες γεωγραφικού εντοπισμού | Location-based services |
| Υπηρεσιών διαδικτυακών ψηφιδωτών αρχείων | Web Coverage Service |
| Φόρτος συστήματος | Load shedding |
| Χρονικά παράθυρα | Time - based windows |
| Χρονοσήμανση | Timestamping |
| Χρονόσημο | Timestamp |