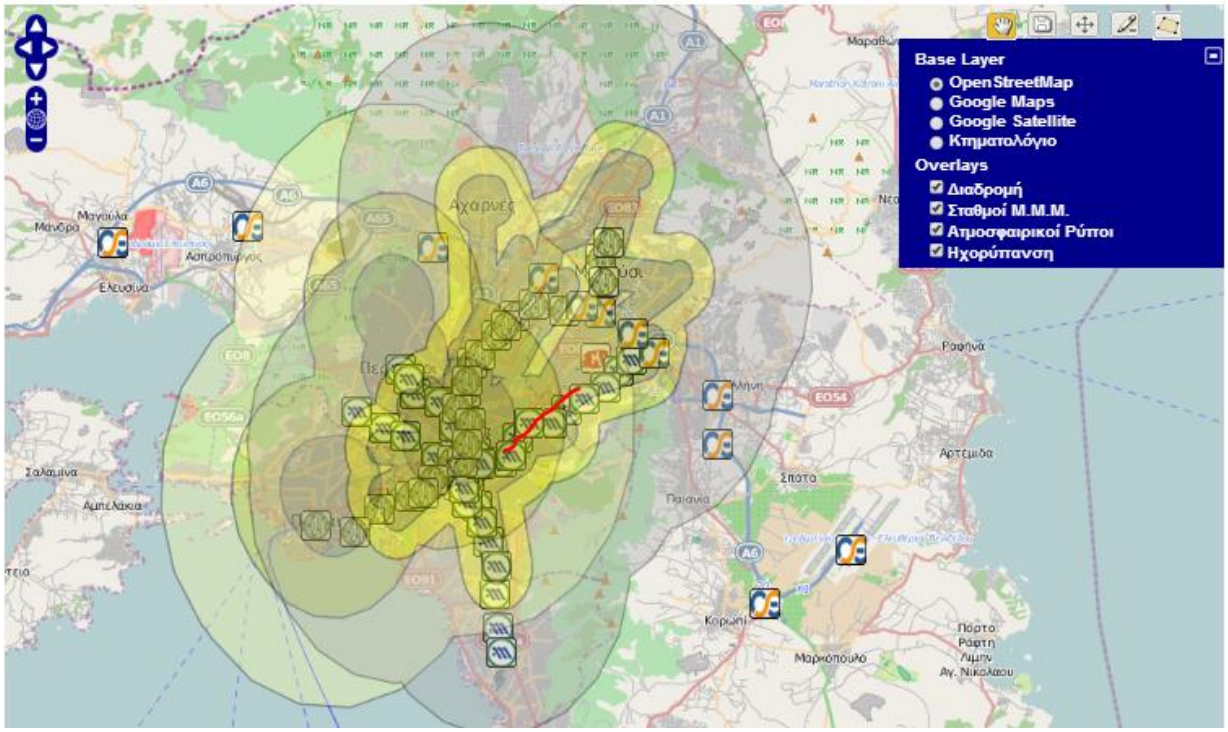
ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΑΓΡΟΝΟΜΩΝ ΚΑΙ ΤΟΠΟΓΡΑΦΩΝ ΜΗΧΑΝΙΚΩΝ

«Πολυ-παραμετρική αξιολόγηση οδικών διαδρομών με χρήση ανοικτών τεχνολογιών web-GIS»



ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΓΚΙΜΤΣΑΣ ΓΙΩΡΓΟΣ

Σεπτέμβριος 2013

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΚΕΦΑΛΑΙΟ 1 – Το Πλαίσιο του Προβλήματος **6**

1.1 Γενικά 6

1.2 Abstract 6

1.3 Περισσότερες Δυνατότητες 7

1.4 Οριοθέτηση του Προβλήματος 7

ΚΕΦΑΛΑΙΟ 2 – Τεχνολογίες και Εργαλεία **9**

2.1 Τεχνολογίες Ιστού 9

2.2 Web Services 9

2.2.1 Πλεονεκτήματα σε σχέση με παλαιότερες Κατανεμημένες Τεχνολογίες 10

2.2.2 Τα τεχνικά χαρακτηριστικά των Web Services 11

2.2.3 Το μοντέλο και η αρχιτεκτονική των Web Services 12

2.2.4 Δυνατότητες των Web Services 13

2.2.5 Εφαρμογές των Web Services 14

2.3 HTML 15

2.3.1 Η ιστορία της HTML 15

2.3.2 Υπερκείμενο 15

2.3.3 Η δομή της HTML 15

2.3.4 Οι ετικέτες στην HTML 17

2.3.5 URL 19

2.3.6 Παράδειγμα HTML 19

2.4 Web GIS 20

2.4.1 Ένα αλληλεπιδραστικό σύστημα βασισμένο στο διαδίκτυο 20

2.4.2 Οι δυνατότητες των Web GIS 21

2.5 Γεωγραφικές Βάσεις Δεδομένων 21

2.5.1 PostgreSQL 21

2.5.2 PostGIS 22

2.5.3 Ο τύπος Geometry και ο μορφότυπος WKT 22

2.5.4 Υποστηριξη Συστημάτων Αναφοράς 24

2.6 XML 25
2.6.1 Η ιστορία της XML 25
2.6.2 Ιδιότητες της XML 26
2.6.3 Elements ή Στοιχεία 29
2.3.4 Οι ετικέτες στην HTML 30
2.6.4 Attributes ή Ιδιότητες 30
2.6.5 Κανόνες ονομασίας 30
2.6.7 DTD 31
2.6.8 XML Schema 32
2.7 Javascript 33
2.7.1 Η ιστορία της Javascript 33
2.7.2 Προγραμματισμός στην πλευρά του πελάτη 34
2.7.3 Δομή της Javascript 34
2.7.4 Τα αντικείμενα στην Javascript 35
2.7.5 Τα χειριστήρια συμβάντος (Event Handlers) της Javascript 38
2.7.6 Παράδειγμα κώδικα Javascript 39
2.7.7 OpenLayers 39
2.8 PHP 40
2.8.1 Η ιστορία της PHP 40
2.8.2 Δομή της PHP 41
2.8.3 Διερμήνευση και Μεταγλώτιση 41
2.8.4 Δυνατότητες της PHP 41
2.8.5 Παράδειγμα PHP κώδικα 43
2.9 GeoServer 43
2.9.1 Τα WFS, WFS-T, WMS και WCS πρωτόκολα 43
2.9.2 Η υπηρεσία WPS 44

ΚΕΦΑΛΑΙΟ 3 – Σχεδίαση & Υλοποίηση διαδικτυακού λογισμικού πολυπαραμετρικής αξιολόγησης διαδρομών **47**
3.1 Ροή Εργασιών 47
3.1.1 APACHE/PHP 48
3.1.2 Postgres/PostGIS 50
3.1.3 GeoServer 51

3.2 Συστατικά στοιχεία λογισμικού που χρησιμοποιούνται 57
3.2.1 QuantumGIS 57
3.2.2 Openlayers 57
3.2.3 Τοπικός Κώδικας 58
3.2.4 MS-PAINT 58
3.2.5 Notepad++ 58
3.3 Διάγραμμα Ροής δεδομένων και συνεργασίας μεταξύ των επιμέρους στοιχείων 59
3.4 Υπηρεσίες που χρησιμοποιήθηκαν/ενσωματώθηκαν έτοιμες 60
3.4.1 Openlayers 60
3.4.2 Google Maps, Bing Maps, Openstreet Maps & Κτηματολόγιο Α.Ε. 60
3.4.3 Google Elevation api 60
3.4.4 Εύρεση των αποστάσεων από τα Μ.Μ.Μ. 60
3.5 Συστατικά στοιχεία λογισμικού που αναπτύχθηκαν 61
3.5.1 Ο υπολογισμός του μήκους των τομών 61
3.5.2 Η αποστολή των XML στον GeoServer 63
3.5.3 Το στήσιμο του OpenLayers Interface 64
3.5.4 Η σύνδεση στο OpenLayers των θεματικών επιπέδων και της διαδρομής 64
3.5.5 Η εμφάνιση των αποτελεσμάτων 64
3.6 Ροή εργασιών κατά την εκτέλεση και η «αρμοδιότητα» εμπλεκόμενων μερών 64

ΚΕΦΑΛΑΙΟ 4 – Εφαρμογή – Παράδειγμα Χρήσης **67**
4.1 Προηγούμενες Εργασίες 67
4.2 Επεκτάσεις - Γενικεύσεις 68
4.3 Σύνοψη των Υπολογιζόμενων Δεικτών 69
4.4 Παράδειγμα Χρήσης Εφαρμογής 70

ΚΕΦΑΛΑΙΟ 5 – Ιδέες Για Περεταίρω Επεκτάσεις **73**
5.1 Προσθήκη νέων παραμέτρων 73
5.2 Ανανέωση Δεδομένων 73
5.3 Ανάπτυξη Εφαρμογής για το χειρισμό του GeoServer 74
5.4 Ανάπτυξη δυναμικών εφαρμογών που να παράγουν διαδρομές 74

ΒΙΒΛΙΟΓΡΑΦΙΑ **75**
ΠΑΡΑΡΤΗΜΑ **76**

ΚΕΦΑΛΑΙΟ 1

Το Πλαίσιο του Προβλήματος

1.1 Γενικά

Ένα σύστημα δρομολόγησης, στην απλούστερη μορφή του, προτείνει τη βέλτιστη, ως προς την απόσταση, διαδρομή ανάμεσα σε δυο γεωγραφικά σημεία. Οι σύγχρονες εφαρμογές δρομολόγησης έχουν εμπλουτιστεί με πληθώρα λειτουργιών – ευκολιών που ανταποκρίνονται στις ολοένα και αυξανόμενες ανάγκες στο νευραλγικό τομέα των μεταφορών. Στο παρελθόν τα συστήματα δρομολόγησης παρείχαν γενικές μόνο πληροφορίες. Ήταν επομένως χρήσιμα μόνο σε επίπεδο στρατηγικού σχεδιασμού. Ακόμη όμως και με τις παρούσες εξελίξεις ένα βασικό επίσης πρόβλημα είναι τα περιορισμένα κριτήρια τα οποία χρησιμοποιούνται που οδηγούν σε αδυναμία σχεδιασμού και επεξεργασίας διαδρομών με ολοκληρωμένη μελέτη και πολύπλευρα κριτήρια.

Σήμερα όμως υπάρχει:

- Η δυνατότητα διασύνδεσης με πληθώρα δεδομένων στη μορφή ψηφιακών χαρτών από το διαδίκτυο
- Η υποδομή που επιτρέπει τη διασύνδεση και την επεξεργασία χωρικών δεδομένων στο διαδίκτυο

Υπάρχουν δηλαδή οι κατάλληλες συνθήκες και τα εργαλεία για την ανάπτυξη ολοκληρωμένων ανοιχτών WEB GIS εφαρμογών που να μπορούν να εξετάζουν - αξιολογουν πολύπαραμετρικά οδικές διαδρομές και μάλλιστα την ευκολία και τα οφέλη μιας δυναμικής εφαρμογής ιστού έναντι μιας στατικής desktop εφαρμογής κυρίως την ταχύτητα δημοσίευσης δεδομένων και την ανεξαρτησία από εξειδικευμένο λογισμικό και εξοπλισμό.

1.2 ABSTRACT

A routing system, in its simplest form, calculates and recommends the fastest (distantwise) route between 2 geographical places. Modern routing applications have been enriched with a variety of functions that respond to the always increasing demands in the neuralgic area of transportation. In the past routing systems provided only generic information. In that way they were only usefull for strategic planning. Even with the present day breakthrouhs a basic problem insists: the very few criteria which are used lead to a weak planning and processing of routing without a complete study and versatile criteria.

But today exist:

- The capability to interconnect with an abundance of data in the form of digitized maps online
- The infrastructure that allows the interconnection and processing of geospatial data online

So today exist the proper conditions and tools to develop comlete open WEB GIS applications that can access using many criteria road routes with the ease and the benefit of a web application to a static desktop application, mainly the speedy publishing of data and the complete independence from specialized software and hardware.

1.3 Περισσότερες δυνατότητες

Παραδείγματα κριτηρίων βελτιστοποίησης ειδικών οδικών διαδρομών είναι:

- **Η ελάχιστη απόσταση από σημεία ενδιαφέροντος**

Παράδειγμα τέτοιου κριτηρίου αποτελεί π.χ. η απόσταση από τους σταθμούς των μέσων μαζικής μεταφοράς.

- **Η ελάχιστη έκθεση σε φαινόμενα**

Παράδειγματα τέτοιου κριτηρίου αποτελεί η έκθεση σε ρύπους, θόρυβο, κίνηση, υψηλές ή χαμηλές θερμοκρασίες.

- **Η βελτιστοποίηση διαδρομής για ειδικούς σκοπούς**

Παράδειγμα τέτοιου τύπου μπορεί να είναι η ήπια ανηφορική κλίση για την βελτιστοποίηση της ευκολίας ενός ποδηλατόδρομου ή για τη μείωση της κατανάλωσης ενός οχήματος.

1.4 Οριοθέτηση του προβλήματος

Η εφαρμογή που αναπτύχθηκε στα πλαίσια αυτής της εργασίας σχεδιάστηκε με τα εξής χαρακτηριστικά:

- Η διαδρομές που θα αξιολογηθούν θα παράγονται είτε από ήδη υπάρχοντα δεδομένα είτε από επί τόπου ψηφιοποίηση με το χέρι
- Τα χαρτογραφικά υπόβαθρα που θα χρησιμοποιηθούν θα είναι ανοιχτά. Θα προέρχονται δηλαδή από υπηρεσίες όπως (OpenStreet maps, Google maps κτλ.)
- Την χρηση map servers για τη δημοσίευση των χωρικών δεδομένων
- Τη χρήση web services για τον υπολογισμό παραμέτρων από το χρήστη
- Τον υπολογισμό για τη διαδρομή χωρικών ιδιοτήτων όπως: απόσταση από σταθμούς Μ.Μ.Μ., την κλίση και τους ρύπους

Τέλος για την επίδειξη των δυνατοτήτων της εφαρμογής θα αναπτυχθεί παράδειγμα πολυ-παραμετρικής αξιολόγησης διαδρομής για το σχεδιασμό ποδηλατοδρόμου στην περιοχή της Αθήνας.

ΚΕΦΑΛΑΙΟ 2

Τεχνολογίες και Εργαλεία

2.1 Τεχνολογίες Ιστού

Το *World Wide Web* (ή σκέτο *Web* όπως συνηθίζουμε να το αποκαλούμε) είναι ένα μεγάλο δίκτυο από υπολογιστές σε όλο τον κόσμο, οι οποίοι μπορούν να επικοινωνούν μεταξύ τους για να μοιραστούν ψηφιακή πληροφορία. Οι πληροφορίες αυτές βρίσκονται σε έγγραφα που ονομάζονται *ιστοσελίδες*. Οι *ιστοσελίδες* είναι αρχεία που βρίσκονται σε υπολογιστές που ονομάζονται Web Servers (οι οποίοι είναι 24 ώρες το 24ωρό συνδεμένοι με το Web ώστε να μπορούμε οποιαδήποτε στιγμή να συνδεθούμε και να ανακτήσουμε μια ιστοσελίδα που βρίσκεται σε αυτούς), ενώ οι υπολογιστές που συνδέονται στους *Web Servers* για να ανακτήσουν τις ιστοσελίδες λέγονται *Web Clients*.

Οι *Web Clients* χρησιμοποιούν ένα πρόγραμμα για να ανακτήσουν τα περιεχόμενα των *ιστοσελίδων* που βρίσκονται στους *Web Servers*. Αυτό το πρόγραμμα ονομάζεται *Web browser* (*πρόγραμμα παρουσίασης ιστοσελίδων* ή *πρόγραμμα πλοήγησης*).

Δύο είναι οι κύριες δουλειές του *Web browser*:

1) να μπορεί να προσπελάσει τις *ιστοσελίδες* μετά από μια αίτηση του χρήστη
2) να εμφανίζει τα περιεχόμενα των *ιστοσελίδων*.

Οι *ιστοσελίδες* περιέχουν οδηγίες για τον τρόπο που θα εμφανίσουν τα περιεχόμενα τους στον *Web browser*. Οι *Web browsers* διαβάζουν τις οδηγίες αυτές και εμφανίζουν τις σελίδες στην οθόνη μας. Οι οδηγίες αυτές είναι γραμμένες στην γλώσσα *HTML*.

2.2 Web Services

Υπάρχουν πολλοί ορισμοί για το τι είναι web services, περίπου όσοι και οι εταιρίες πληροφορικής που αναπτύσουν εργαλεία για τα web services.

Ένας πολύ καλός ορισμός έρχεται από την IBM:

«Τα web services είναι μια τεχνολογία που επιτρέπει στις εφαρμογές να επικοινωνούν μεταξύ τους ανεξαρτήτως πλατφόρμας και γλώσσας προγραμματισμού. Ένα web service είναι μια διεπαφή λογισμικού (software interface) που περιγράφει μια συλλογή από λειτουργίες οι οποίες μπορούν να προσεγγιστούν από το δίκτυο μέσω πρότυπων μηνυμάτων XML. Χρησιμοποιεί πρότυπα βασισμένα στη γλώσσα XML για να περιγράψει μία λειτουργία (operation) προς εκτέλεση και τα δεδομένα προς ανταλλαγή με κάποια άλλη εφαρμογή. Μια ομάδα από web services οι οποίες αλληλεπιδρούν μεταξύ τους καθορίζει μια εφαρμογή web services.»

Η Microsoft μέσα από το MSDN της καταλήγει ότι όλα τα web services έχουν τρία κοινά χαρακτηριστικά:

1. Τα web services εκθέτουν χρήσιμη λειτουργικότητα σε χρήστες του διαδικτύου μέσα από ένα πρότυπο δικτυακό πρωτόκολο. Στις περισσότερες περιπτώσεις αυτό το πρωτόκολλο είναι το SOAP (Simple Object Access Protocol).
2. Τα web services παρέχουν ένα τρόπο να περιγράψουν τις διεπαφές τους με αρκετή λεπτομέρεια ώστε να επιτρέψουν στο χρήστη τους να χτίσει μια εφαρμογή - πελάτη η οποία να επικοινωνήσει μαζί τους. Η περιγραφή συνήθως παρέχεται σε ένα έγγραφο XML το οποίο ονομάζεται έγγραφο WSDL (Web Services Description Language).
3. Τα web services καταχωρούνται ώστε οι δυνητικοί χρήστες να μπορούν να τα βρουν εύκολα. Αυτό γίνεται με το UDDI (Universal Discovery Description and Integration).

Τα web services λοιπόν αποτελούν μία αρχιτεκτονική κατανεμημένων συστημάτων κατασκευασμένη από πολλά διαφορετικά υπολογιστικά συστήματα τα οποία επικοινωνούν μέσω του δικτύου ώστε να δημιουργήσουν ένα σύστημα. Αποτελούνται από ένα σύνολο από πρότυπα τα οποία επιστρέπουν στους υπεύθυνους για την ανάπτυξη (προγραμματιστές - developers) να υλοποιήσουν κατανεμημένες εφαρμογές (χρησιμοποιώντας διαφορετικά εργαλεία από διαφορετικούς προμηθευτές) ώστε να κατασκευάσουν εφαρμογές που χρησιμοποιούν ένα συνδυασμό από ενότητες λογισμικού (software modules) οι οποίες καλούνται από συστήματα που ανήκουν σε διαφορετικά τμήματα ενός οργανισμού ή σε διαφορετικούς οργανισμούς.

2.2.1 Πλεονεκτήματα σε σχέση με παλαιότερες κατανεμημένες τεχνολογίες

1. Ευκολότερος χειρισμός δεδομένων

Παραδοσιακά το κυριότερο πρόβλημα στις κατανεμημένες τεχνολογίες ήταν το λεγόμενο tight-coupling ή στα ελληνικά η ισχυρή συνδεδιμότητα. Μια εφαρμογή που καλούσε μια άλλη απομακρυσμένη ήταν αυστηρά δεμένη με αυτή από την κλήση λειτουργίας (function call) που εκτελούσε και τις παραμέτρους που περνούσε. Στα περισσότερα συστήματα πριν από την έλευση των web services ο τρόπος επικοινωνίας ήταν μια σταθερή διεπαφή με λίγη έως καθόλου ευελεξία ή προσαρμοστικότητα στα περιβάλλοντα ή τις ανάγκες που μεταβάλλονται συνεχώς.

Τα web services χρησιμοποιούν τη γλώσσα XML η οποία μπορεί να περιγράψει οποιαδήποτε δεδομένα σε ένα πραγματικά ανεξάρτητο από πλατφόρμα τρόπο για ανταλλαγή αυτών των δεδομένων μεταξύ συστημάτων. Με αυτόν τον τρόπο οδηγούμαστε σε εφαρμογές με χαλαρή συνδεσιμότητα (loosely-coupled). Επιπλέον τα web services μπορούν να λειτουργήσουν σε πιο αφηρημένο επίπεδο στο οποίο μπορούν να επαναξιολογήσουν, να τροποποιήσουν ή να χειριστούν τύπους δεδομένων δυναμικά κατά περίπτωση. Έτσι σε τεχνικό επίπεδο τα web services μπορούν να χειριστούν δεδομένα πολύ ευκολότερα και να επιτρέψουν στο λογισμικό να επικοινωνεί πιο ελεύθερα.

2. Απλότητα πρωτοκόλλου επικοινωνίας

Τα web services χρησιμοποιούν ως πρωτόκολλο επικοινωνίας το SOAP. Το πρωτόκολλο αυτό είναι πολύ πιο απλό από πρωτόκολλα παλαιότερων τεχνολογιών όπως αυτά που χρησιμοποιούνταν από τα κατανεμημένα περιβάλλοντα CORBA , DCOM, RPC. Έτσι το να δημιουργήσει κανείς μια υλοποίηση SOAP που υπόκειται στα

πρότυπα (standards-compliant) είναι πολύ πιο εύκολο. Σήμερα μπορεί να βρει κανείς υλοποιήσεις του SOAP από τις μεγαλύτερες εταιρίες πληροφορικής αλλά ακόμη και από μεμονομένους προγραμματιστές, πράγμα αδιανόητο για παλαιότερες κατανεμημένες τεχνολογίες.

3. Απλότητα υποδομής

Τα web services λειτουργούν με πρότυπες γλώσσες και πρωτόκολλα όπως η XML, το HTTP και το TCP/IP. Η πλειονότητα των εταιριών έχουν ήδη την δικτυακή υποδομή και τους ανθρώπους με γνώσεις και εμπειρία που τη συντηρούν. Έτσι το κόστος για την εφαρμογή των web services είναι σημαντικά μικρότερο από αυτό των προηγούμενων τεχνολογιών.

4. Ευκολία στην επικοινωνία

Με τις προηγούμενες τεχνολογίες η συνεργασία μεταξύ εταιριών ήταν ένα θέμα διότι κατανεμημένες τεχνολογίες όπως CORBA και DCOM χρησιμοποιούσαν μη πρότυπες πόρτες. Σαν αποτέλεσμα η συνεργασία σήμαινε άνοιγμα "οπών" στα τείχη προστασίας (firewalls), κάτι που πολλές φορές δεν ήταν αποδεκτό από τους ανθρώπους της πληροφορικής σε μια εταιρία αφού έθετε σε κίνδυνο την ασφάλεια των συστημάτων. Το γεγονός αυτό δεν επέτρεπε δυναμική συνεργασία λόγω του ότι απαιτούσε μια χειροκίνητη διαδικασία για τη συνεργασία μιας εταιρίας με τους συνεργάτες της. Τα web services μπορούν να χρησιμοποιήσουν (μεταξύ άλλων) το HTTP ως πρωτόκολλο μεταφοράς και τα περισσότερα τείχη προστασίας επιτρέπουν την πρόσβαση μέσω της θύρας 80 (πρότυπη θύρα για το HTTP). Με αυτόν τον τρόπο οδηγούμαστε σε ευκολότερες και δυναμικές συνεργασίες μεταξύ των συστημάτων των εταιρειών.

5. Διαλειτουργικότητα και ευκολία ανάπτυξης νέων εφαρμογών

Οι προηγούμενες κατανεμημένες τεχνολογίες υπέφεραν από ζητήματα διαλειτουργικότητας διότι κάθε προμηθευτής (vendor) υλοποιούσε το δικό του πρότυπο για distributed object messaging. Με την XML σαν το μόνο πρότυπο στα web services, συστήματα φτιαγμένα από διαφορετικές τεχνολογίες όπως η Java και το .Net μπορούν να επικοινωνήσουν μεταξύ τους. Επιπλέον λόγω της απλότητας της XML είναι πολύ πιο εύκολο να γραφτούν νέες εφαρμογές σε μικρό χρονικό διάστημα.

2.2.2 Τα τεχνικά χαρακτηριστικά των Web Services

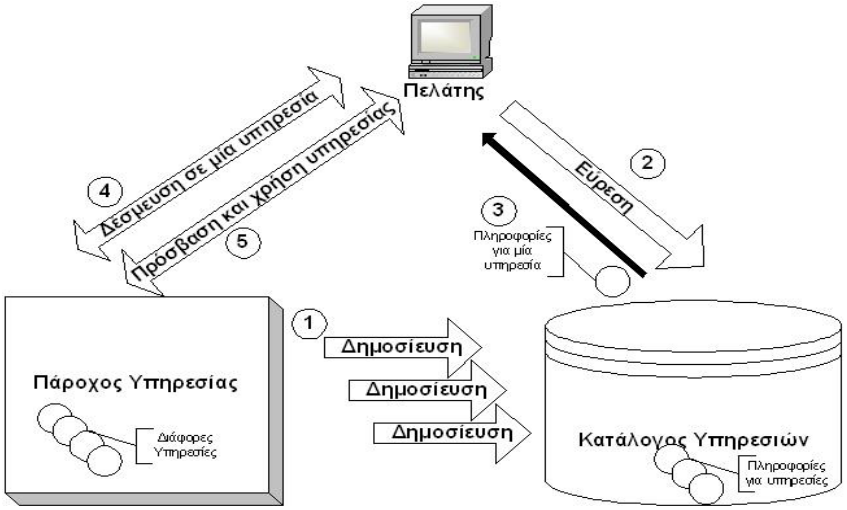
Τα web services προσφέρουν όλα αυτά τα δυναμικά χαρακτηριστικά ώστε να συνδυάσουμε υπηρεσίες σε εφαρμογές, πρέπει όμως πρώτα να χτίσουμε αυτές τις υπηρεσίες. Οι γλώσσες προγραματισμού στην πληροφορική συνεχώς εξελίσονται. Ξεκινώντας δεκαετίες πριν με την ιδέα της function στην οποία παρέχουμε μερικές παραμέτρους, εκτελεί μια λειτουργία με αυτές τις παραμέτρους, και επιστρέφει μια τιμή βασισμένη στους υπολογισμούς που έγιναν. Τελικά, αυτή η πρώτη έννοια εξελίχθηκε σε αντικείμενο (object) όπου κάθε αντικείμενο είχε όχι απλώς έναν αριθμό από λειτουργίες (functions) που μπορεί να εκτελέσει αλλά και τις δικές του ιδιωτικές μεταβλητές (private data variables), αντί να στηρίζεται σε εξωτερικές μεταβλητές του συστήματος που προηγουμένως έκανα πιο περίπλοκη την ανάπτυξη εφαρμογών. Δεδομένου ότι οι εφαρμογές άρχισαν να επικοινωνούν μεταξύ τους, η έννοια του καθορισμού καθολικών διεπαφών (universal interfaces) για αντικείμενα έγινε σημαντική, επιτρέποντας αντικείμενα σε διαφορετικές πλατφόρμες να επικοινωνούν ακόμη και αν είχαν αναπτυχθεί σε διαφορετικές γλώσσες προγραμματισμού και εκτελούνταν σε διαφορετικά λειτουργικά συστήματα.

Στο πιο πρόσφατο βήμα, τα web services προχώρησαν μπροστά με την έννοια των διεπαφών και επικοινωνιών καθορισμένων με XML, ενώνοντας τελικά κάθε είδους εφαρμογή με οποιαδήποτε άλλη, όπως και παρέχοντας την ελευθερία στις εφαρμογές αν αλλάξουν και να εξελιχθούν με το χρόνο, αρκεί να είναι σχεδιασμένες σύμφωνα με την κατάλληλη διεπαφή. Η μεταβλητότητα της XML είναι αυτό που κάνει τα web services διαφορετικά από τεχνολογίες προηγούμενων γενεών. Επιτρέπει το διαχωρισμό της γραμματικής δομής (syntax) και της γραμματικής έννοιας (semantics), και του πώς αυτά υποβάλλονται σε επεξεργασία και κατανοούνται από μία υπηρεσία και το περιβάλλον μέσα στο οποίο υπάρχει. Έτσι λοιπόν τώρα, τα αντικείμενα μπορούν να καθοριστούν σαν υπηρεσίες, οι οποίες επικοινωνούν με άλλες υπηρεσίες σε γραμματική καθορισμένη σε XML, με την οποία κάθε υπηρεσία μεταφράζει και αναλύει το μήνυμα σύμφωνα με μην τοπική της υλοποίησης και το περιβάλλον της. Κατά συνέπεια μια διακτυακή εφαρμογή μπορεί πραγματικά να συντεθεί από πολλαπλές οντότητες διαφόρων υλοποιήσεων και σχεδιασμών εφόσον προσαρμόζονται στους κανόνες που καθορίζονται από την προσανατολισμένη στις υπηρεσίες αρχιτεκτονική τους.

2.2.3 Το μοντέλο και η αρχιτεκτονική των Web Services

Το μοντέλο των web services ακολουθεί το παράδειγμα δημοσίευση (publish), εύρεση(find) και σύνδεση(bind). Στο πρώτο βήμα, ο προμηθευτής της υπηρεσίας δημοσιεύει την υπηρεσία σε ένα κατάλογο υπηρεσιών. Στο δεύτερο βήμα, ο πελάτης ο οποίος ψάχνει για μία υπηρεσία η οποία να καλύπτει τις απαιτήσεις του την αναζητεί στον κατάλογο. Αφού επιτυχημένα βρει πολλαπλές υπηρεσίες επιλέγει μία βάσει των προτιμήσεών του. Τότε μεταφορτώνει την περιγραφή της υπηρεσίας και συνδέεται (δεσμεύεται) με αυτήν ώστε να μπορέσει να καλέσει και να εκτελέσει την υπηρεσία.

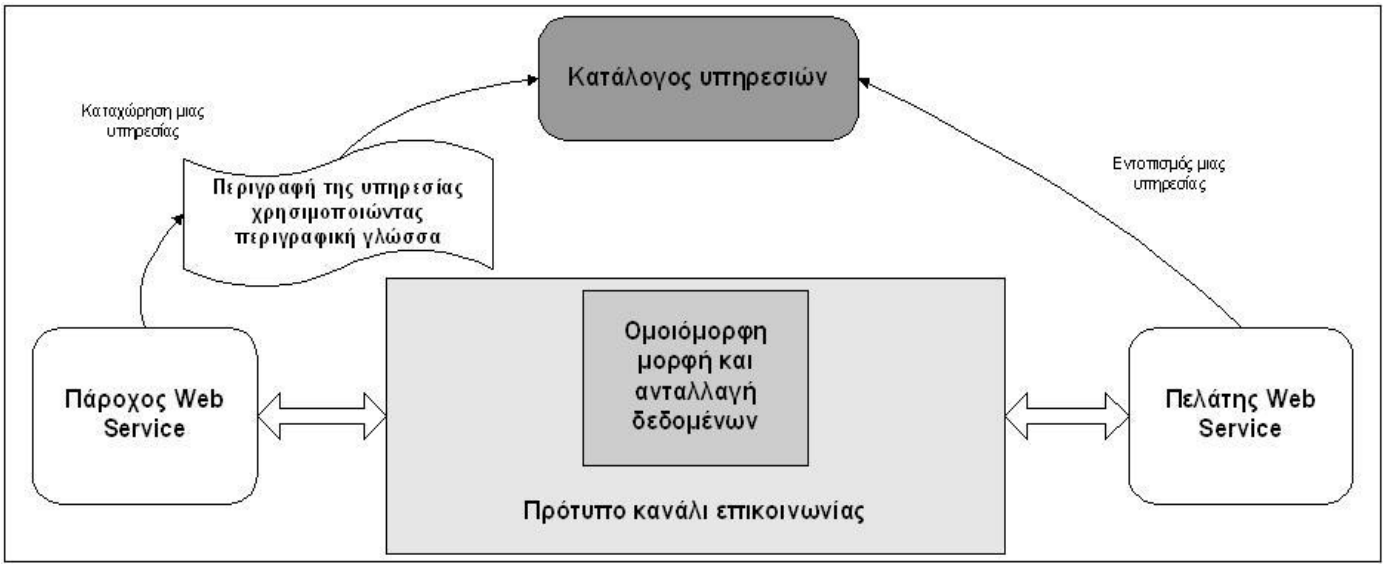
Όταν μιλάμε λοιπόν για μία αρχιτεκτονική προσανατολισμένη στις υπηρεσίες προκύπτουν ορισμένα ζητήματα. Η εφαρμογή που παρέχει την υπηρεσία και η εφαρμογή-πελάτης η οποία χρησιμοποιεί την υπηρεσία μιλάνε μεταξύ τους σε μια κοινή γλώσσα. Έπειτα οι δύο εφαρμογές χρειάζονται ένα τρόπο να εντοπίζουν η μία την άλλη πριν ξεκινήσουν να μιλούν μεταξύ τους. Αυτό αληθεύει ακόμη παραπάνω για τις κατανεμημένες εφαρμογές όπου μία εφαρμογή δεν έχει καμία γνώση της θέσης της άλλης.



Εικόνα 2.1 – Δομή Web Services

Ως εκ τούτου, μπορούμε να πούμε ότι μια βασική αρχιτεκτονική για web services πρέπει να παρέχει:

- Έναν πρότυπο τρόπο για επικοινωνία.
- Ένα ομοιόμορφο μηχανισμό για περιγραφή και ανταλλαγή των δεδομένων.
- Μια πρότυπη περιγραφική γλώσσα (meta language) για να περιγράψει τις υπηρεσίες που προσφέρονται.
- Ένα μηχανισμό για να καταχωρούνται και να εντοπίζονται οι εφαρμογές που βασίζονται σε web services.



Εικόνα 2.2 – Αρχιτεκτονική Web Service

2.2.4 Δυνατότητες των Web Services

Τα web services επιτρέπουν:

- Την αλληλεπίδραση μεταξύ υπηρεσιών σε οποιαδήποτε πλατφόρμα, γραμμένες σε οποιαδήποτε γλωσσα προγραμματισμού.
- Να αντιληφθούμε λειτουργίες εφαρμογών ως εργασίες, οδηγούμενοι σε ανάπτυξη και ροές εργασιών προσανατολισμένες σε εργασίες. Αυτό επιτρέπει μια υψηλότερη αφαίρεση του λογισμικού το οποίο μπορεί να υιοθετηθεί από λιγότερο τεχνικά καταρτισμένους χρήστες.
- Τη χαλαρή συνδεσιμότητα μεταξύ εφαρμογών, πράγμα που σημαίνει ότι αλληλεπιδράσεις μεταξύ υπηρεσιών δε θα χαλάνε κάθε φορά που υπάρχει κάποια αλλαγή στο πώς μία ή περισσότερες υπηρεσίες σχεδιάζονται ή υλοποιούνται.
- Την προσαρμογή ήδη υπάρχουσων εφαρμογών στις μεαβαλλόμενες επιχειρησιακές συνθήκες και ανάγκες των πελατών.
- Να παρέχουμε υπάρχουσες εφαρμογές λογισμικού με διαπαφές υπηρεσιών χωρίς να αλλάξουμε τις αρχικές εφαρμογές, επιτρέποντάς τους να λειτουργούν πλήρως στο περιβάλλον των υπηρεσιών.
- Να εισάγουμε άλλες διοικητικές λειτουργίες ή λειτυοργίες διαχείρισης διαδικασιών όπως η αξιοπιστία, υπευθυνότητα, ασφάλεια, κ.λπ., ανεξάρτητες της αρχικής λειτουργίας μιας εφαρομογής, αυξάνοντας κατά συνέπεια τη μεταβλητότητα και τη χρησιμότητά της στο επιχειρησιακό περιβάλλον.

2.2.5 Εφαρμογές των Web Services

Τα πρώτα web services σκόπευαν να είναι πηγές πληροφορίας τις οποίες μπορεί κανείς πολύ εύκολα να ενσωματώσει στις εφαρμογές του: τιμές μετοχών, προβλέψεις καιρού, αποτελέσματα αθλητικών παχινιδιών κλπ. Έιναι εύκολο να φανταστεί κανείς μια ολόκληρη κατηγορία εφαρμογών που μπορεί να κατασκευάσει ώστε να αναλύει και να συνδυάζει πληροφορία που τον ενδιαφέρει και να την παρουσιάζει με ποικίλους τρόπους. Οι περισσότερες πληροφορίες είναι ήδη διαθέσιμες στον παγκόσμιο ιστό αλλά τα web services θα κάνουν την προγραμματιστική πρόσβαση σε αυτές πιο εύκολη και πιο αξιόπιστη.

Εκθέτοντας ήδη υπάρχουσες εφαρμογές σαν web services θα επιτρέψει στους χρήστες να κατασκευάσουν νέες πιο ισχυρές εφαρμογές οι οποίες χρησιμοποιούν τα web services σαν δομικά στοιχεία. Μία εφαρμογή του προμηθευτή, εκτός από το να εκθέτει τις υπηρεσίες της στον ιστό, θα μπορούσε να χρησιμοποιήσει άλλα web services για να κάνει άλλες εργασίες.

Υπάρχει μία μεγάλη λίστα από έτοιμα web services που θα μπορούσε να χρησιμοποιήσει κανείς, ακόμα και εντελώς δωρεάν. Στον πίνακα που ακολουθεί αναγράφονται κάποια ενδεικτικά web services.

SAINTlogin Cellphone Users Validation	Επιτρέπει το LogIn σε μια ιστοσελίδα χωρίς να απαιτείται η εισαγωή ονόματος και κώδικα χρήστη
Video WebServices	Μηχανή αναζήτησης Βίντεο
Ontok Wikipedia	Μετατρέπει κείμενο ή φράση σε μια ιστοσελίδα σε υπερσύνδεσμο σε άρθρο του Wikipedia
Xignite Tools	Παρέχει εργαλεία συστήματος όπως το ping και το WHOIS
Global Weather	Παρέχει πληροφορίες για τον καιρό σε κάθε μεγάλη πόλη σε όλο κόσμο

Πίνακας 2.1 – Ενδεικτικά Web Services

2.3 HTML

Η **HTML** (ακρωνύμιο του αγγλικού **H**yper**T**ext **M**arkup **L**anguage, δηλαδή. Γλώσσα Σήμανσης Υπερκειμένου) είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων.

Τα στοιχεία της HTML χρησιμοποιούνται για να χτίσουν όλους του ιστότοπους. Η HTML επιτρέπει την ενσωμάτωση εικόνων και άλλων αντικειμένων μέσα στη σελίδα και μπορεί να χρησιμοποιηθεί για να εμφανίσει διαδραστικές φόρμες. Παρέχει τις μεθόδους δημιουργίας δομημένων εγγράφων (δηλαδή εγγράφων που αποτελούνται από το περιεχόμενο που μεταφέρουν και από τον κώδικα μορφοποίησης του περιεχομένου) καθορίζοντας δομικά σημαντικά στοιχεία για το κείμενο, όπως κεφαλίδες, παραγράφους, λίστες, συνδέσμους, παραθέσεις και άλλα. Μπορούν επίσης να ενσωματώνονται σενάρια εντολών σε γλώσσες, όπως η JavaScript, τα οποία επηρεάζουν τη συμπεριφορά των ιστοσελίδων HTML.

2.3.1 Η ιστορία της HTML

Το 1980, ο φυσικός Τιμ Μπέρνερς Λι, ο οποίος εργαζόταν στο CERN, επινόησε το ENQUIRE, ένα σύστημα χρήσης και διαμοιρασμού εγγράφων για τους ερευνητές του CERN, και κατασκεύασε ένα πρωτότυπό του. Αργότερα, το 1989, πρότεινε ένα σύστημα βασισμένο στο διαδίκτυο, το οποίο θα χρησιμοποιούσε υπερκείμενο. Έτσι, έφτιαξε την προδιαγραφή της HTML και έγραψε τον browser και το λογισμικό εξυπηρετητή στα τέλη του 1990.

Η πρώτη δημόσια διαθέσιμη περιγραφή της HTML ήταν ένα έγγραφο με το όνομα Ετικέτες HTML, το οποίο πρωτοαναφέρθηκε στο Διαδίκτυο από τον Μπέρνερς Λι στα τέλη του 1991. Περιέγραφε τα 20 στοιχεία τα οποίο αποτελούσαν τον αρχικό και σχετικά απλό σχεδιασμό της HTML. Εκτός από την ετικέτα υπερσυνδέσμου, οι υπόλοιπες ήταν έντονα επηρεασμένες από την SGMLguid, μια μορφή δημιουργίας τεκμηρίωσης, φτιαγμένη στο CERN και βασισμένη στην SGML.

Μετά που τα πρόχειρα HTML και HTML+ έληξαν, στις αρχές του 1994, το IETF δημιούργησε την Ομάδα Εργασίας για την HTML, η οποία το 1995 ολοκλήρωσε την «HTML 2.0», με την πρόθεση να αποτελέσει την πρώτη προδιαγραφή πάνω στην οποία θα βασίζονταν οι μελλοντικές υλοποιήσεις. Η HTML 2.0 δημοσιεύτηκε ως RFC 1866, και περιείχε ιδέες από τα πρόχειρα HTML και HTML+. Η αρίθμηση 2.0 σκόπευε απλά να ξεχωρίσει την νέα έκδοση από τα πρόχειρα που προηγήθηκαν.

Η περαιτέρω ανάπτυξη κάτω από την επίβλεψη του IETF καθυστέρησε λόγω σύγκρουσης ενδιαφερόντων. Από το 1996 και μετά, οι προδιαγραφές της HTML τηρούνται, μαζί με ανάδραση από τους δημιουργούς λογισμικού, από το World Wide Web Consortium (W3C). Ωστόσο, το 2000 η HTML έγινε επίσης παγκόσμιο πρότυπο (ISO/IEC 15445:2000). Η τελευταία προδιαγραφή της HTML, η HTML 4.01 δημοσιεύτηκε από το W3C το 1999, και το 2001 δημοσιεύτηκαν επίσης και τα λάθη και οι παραλείψεις της (errata).

2.3.2 Υπερκείμενο

Το υπερκείμενο είναι ένα σύνολο πληροφοριών μέσα στο οποίο μπορούμε να κινηθούμε με μη γραμμικό τρόπο με την βοήθεια συνδέσμων. Οι ίδιοι οι σύνδεσμοι υλοποιούν την έννοια του υπερκειμένου.

Εικόνα 2.3 - Υπερκείμενο

2.3.3 Η δομή της HTML

HTML σημαίνει γλώσσα χαρακτηρισμού υπερκειμένου. Η χρήση μιας γλώσσας χαρακτηρισμού σημαίνει ότι γράφεται πρώτα το κείμενο και έπειτα προσθέτονται ειδικά σύμβολα γύρω από τις λέξεις ή από ολόκληρες προτάσεις ώστε να καθοριστεί η εμφάνιση τους στην οθόνη. Τα ειδικά σύμβολα στην HTML λέγονται ετικέτες (tags). Η HTML διαθέτει ένα πεπερασμένο αριθμό ετικετών που μπορούμε να χρησιμοποιήσουμε. Η τελευταία αναθεώρηση του HTML προτύπου είναι η HTML5.

Το αρχείο που περιέχει HTML ετικέτες λέγεται HTML αρχείο και έχει επέκταση .html ή .htm (εκτός βέβαια κι' αν η ιστοσελίδα είναι δυναμική οπότε έχει επεκτάσεις όπως .php. asp, jsp κτλ.). Τα αρχεία αυτά είναι απλά αρχεία κειμένου σε μορφή ASCII και δεν περιέχουν πληροφορίες για το περιβάλλον ή τα προγράμματα με τα οποία θα λειτουργήσουν. Τα αρχεία htm ή html μπορούν να ανοιχτούν για επεξεργασία με οποιονδήποτε επεξεργαστή κειμένου, π.χ. Σημειωματάριο (Notepad) των Windows.

Ένα αρχείο HTML αρχίζει πάντα με την ετικέτα <html> και αποτελείται από δύο ενότητες: την κεφαλή (HEAD) και το κυρίως περιεχόμενο (BODY) ή αλλιώς το "σώμα" της σελίδας.

- **Η ετικέτα <HTML>**

Με την ετικέτα <html> αρχίζουμε πάντα τον κώδικα μας και με την ετικέτα </html> τον τερματίζουμε. Με αυτόν τον τρόπο πληροφορούμε τον browser ότι οι γραμμές που περικλείονται μέσα σε αυτές τις δύο ετικέτες είναι κώδικας γραμμένος σε γλώσσα HTML.

- **Η Ενότητα HEAD**

Η πρώτη ενότητα (ενότητα HEAD) μιας HTML σελίδας ορίζεται με τις ετικέτες <head>...</head> .

Οι ετικέτες που γράφονται στην ενότητα HEAD, αποτελούν τον πρόλογο για την HTML σελίδα. Υπάρχουν μόνο λίγες ετικέτες που γράφονται στην ενότητα αυτή. Η πιο βασική από αυτές είναι η ετικέτα <title>, η οποία καθορίζει τον τίτλο της σελίδας ο οποίος εμφανίζεται στο πάνω μέρος του παραθύρου του web browser.

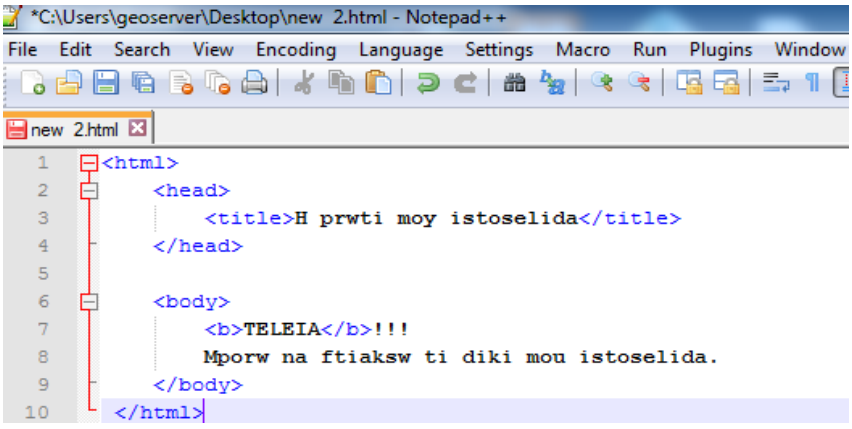
Μια άλλη ετικέτα της ενότητας HEAD είναι η ετικέτα <meta>. Μια από τις λειτουργίες της ετικέτας αυτής είναι να ορίζει το σετ των χαρακτήρων που θα χρησιμοποιήσουμε στην σελίδα.

Ότι γράφουμε μέσα στην ενότητα HEAD δεν εμφανίζεται στην οθόνη του browser!

- **Η Ενότητα BODY**

Η δεύτερη ενότητα (ενότητα body) ορίζεται με τις ετικέτες <body>...</body>

Το ζευγάρι των ετικετών <body> και </body> ορίζει το κυρίως περιεχόμενο της σελίδας μέσα στο οποίο γράφουμε το κείμενο που θέλουμε να εμφανιστεί μαζί με τις HTML ετικέτες που το μορφοποιούν. Στην ενότητα αυτή τοποθετούμε επίσης εικόνες, video και ότι άλλο θέλουμε να εμφανιστεί στην σελίδα.



Εικόνα 2.4 – Δομή της HTML

2.3.4 Οι ετικέτες στην HTML

Οι ετικέτες ελέγχουν την δομή και την μορφή του κειμένου της ιστοσελίδας. Επίσης παρέχουν πληροφορίες προς τον web browser για την σελίδα που πρόκειται να εμφανίσουν, όπως ο τίτλος της σελίδας ή ο συγγραφέας της, κ.α.

Οι HTML ετικέτες γράφονται ανάμεσα στα σύμβολα < και >

πχ. <όνομα-ετικέτας>

Οι περισσότερες HTML ετικέτες αποτελούνται από μια ετικέτα αρχής και μια ετικέτα τέλους και ανάμεσα σε αυτές υπάρχει το κείμενο που χαρακτηρίζεται από τις ετικέτες αυτές. Η ετικέτα τέλους περιέχει τον χαρακτήρα / πριν το όνομα της ετικέτας.

πχ. <όνομα-ετικέτας> ... κείμενο ... </όνομα-ετικέτας>

πχ. Κείμενο με έμφαση

Υπάρχουν ορισμένες ετικέτες που δεν έχουν ετικέτες τέλους. Στις ετικέτες αυτές, πριν από το σύμβολο > τοποθετούμε τον χαρακτήρα /

πχ. <όνομα-ετικέτας />

πχ.

Βασικές ετικέτες:

- **Επικεφαλίδες**
  Οι HTML επικεφαλίδες είναι κείμενο που εμφανίζετε με μεγάλα και έντονα γράμματα. Οι επικεφαλίδες ορίζονται από τις ετικέτες <h1>, <h2>, <h3>, <h4>, <h5> και <h6>. Με την <h1> ορίζουμε την μεγαλύτερη ετικέτα ενώ με την <h6> την μικρότερη.
  Πριν και μετά την επικεφαλίδα εισάγεται αυτόματα στον browser μια κενή γραμμή.
- **Παράγραφοι**
  Οι παράγραφοι ορίζονται από το ζευγάρι ετικετών και .
  Πριν και μετά την παράγραφο εισάγεται αυτόματα στον browser μια κενή γραμμή.
- **Αλλαγή γραμμής**
  Η ετικέτα
 χρησιμοποιείται όταν θέλουμε να τελειώσουμε μια γραμμή κειμένου και να αρχίσουμε μια καινούργια.
- **Οριζόντια Γραμμή**
  Η οριζόντια γραμμή είναι ένα απλό γραφικό που μπορούμε να χρησιμοποιήσουμε στην σελίδα μας (κυρίως σαν διαχωριστικό). Με την ετικέτα <hr> τοποθετούμε μια οριζόντια γραμμή στην σελίδα μας.
- **Σχόλια στην HTML**
  Τα σχόλια χρησιμοποιούνται για να γράφουμε σημειώσεις μέσα στον πηγαίο κώδικα. Δεν εμφανίζονται στην οθόνη του browser. Ένας λόγος για να χρησιμοποιήσουμε σχόλια μέσα σε ένα html αρχείο είναι να γράψουμε την ημερομηνία που δημιουργήσαμε το αρχείο. Ένα σχόλιο αρχίζει με <!-- και τελειώνει με -->.

Ιδιότητες Ετικετών:

Οι ιδιότητες (attributes) των ετικετών είναι τιμές που δίνουν στην ετικέτα διάφορα χαρακτηριστικά. Κάθε μια από αυτές τις τιμές επιδρά διαφορετικά στην εμφάνιση ή την λειτουργία των ετικετών. Μια ιδιότητα μπαίνει αμέσως μετά το όνομα της ετικέτας και αποτελείται από το όνομα της και μια τιμή μέσα σε διπλά εισαγωγικά

Αν και για κάθε ετικέτα υπάρχει μια συγκεκριμένη λίστα διαθέσιμων ιδιοτήτων, υπάρχουν και κάποιες ιδιότητες που μπορούν να εφαρμοστούν σε όλες τις ετικέτες. Αυτές είναι οι παρακάτω :

Ιδιότητες	Τιμή	Περιγραφή	Εξαιρούνται οι ετικέτες
Accesskey	χαρακτήρας	Ορίζει ένα χαρακτήρα του πληκτρολογίου με τον οποίο, όταν θα πατάμε, θα έχουμε άμεση προσπέλαση στο στοιχείο.	
Class	κανόνας μιας κλάσης ή όνομα της κλάσης	Ορίζει το στυλ του στοιχείου.	<base>, <head>, <html>, <meta>, <param>, <script>, <style>, <title>
Dir	ltr (=left to right) ή rtl (=right to left)	Ορίζει την κατεύθυνση του κειμένου. Η γραφή μερικών γλωσσών (κυρίως των ανατολικών χωρών) αποτυπώνεται από τα δεξιά προς τα αριστερά (ltr).	<base>, , <frame>, <frameset>, <hr>, <iframe>, <param>, <script>
Id	όνομα id	ορίζει ένα μοναδικό όνομα (ταυτότητα) για ένα στοιχείο της σελίδας ώστε να μπορούμε να αναφερθούμε σε αυτό μέσα από ένα script (π.χ. JavaScript ή VBScript).	<base>, <head>, <html>, <meta>, <param>, <script>, <style>, <title>
Lang	κωδικός της γλώσσας	Ορίζει τον κωδικό της γλώσσας (GR για τα ελληνικά, EN για τα αγγλικά, κτλ.).	<base>, , <frame>, <frameset>, <hr>, <iframe>, <param>, <script>
Style	κανόνας style	Ορίζει το στυλ του στοιχείου.	<base>, <head>, <html>, <meta>, <param>, <script>, <style>, <title>
Tabindex	αριθμός	Ορίζει την σειρά tab των στοιχείων της σελίδας.	
Title	κείμενο	Ορίζει το κείμενο του πλαισίου το οποίο εμφανίζεται όταν αφήνουμε τον δείκτη του ποντικιού μας επάνω στο στοιχείο (mouseover event).	<base>, <head>, <html>, <meta>, <param>, <script>, <style>, <title>

Πίνακας 2.2 – Ιδιότητες Ετικετών της HTML

2.3.5 URL

Πληκτρολογώντας ένα URL (ή αλλιώς Διεύθυνση) στον browser μπορούμε να έχουμε πρόσβαση στο αρχείο που αυτό "δείχνει" στο Internet. Στον κώδικα HTML, χρησιμοποιούμε URLs για να "δείξουμε", μέσα από ένα html έγγραφο, ένα άλλο html έγγραφο ή άλλη μορφή αρχείου, όπως εικόνες, ήχους, αρχεία pdf κτλ. που βρίσκονται μέσα στο Internet.

Σε έναν σύνδεσμο <a>, για παράδειγμα, τοποθετούμε στην ιδιότητα href το URL που θέλουμε να δείχνει ο σύνδεσμος.

Σε μια εικόνα <img>, τοποθετούμε στην ιδιότητα src το URL το οποίο δείχνει την τοποθεσία που βρίσκεται η εικόνα.

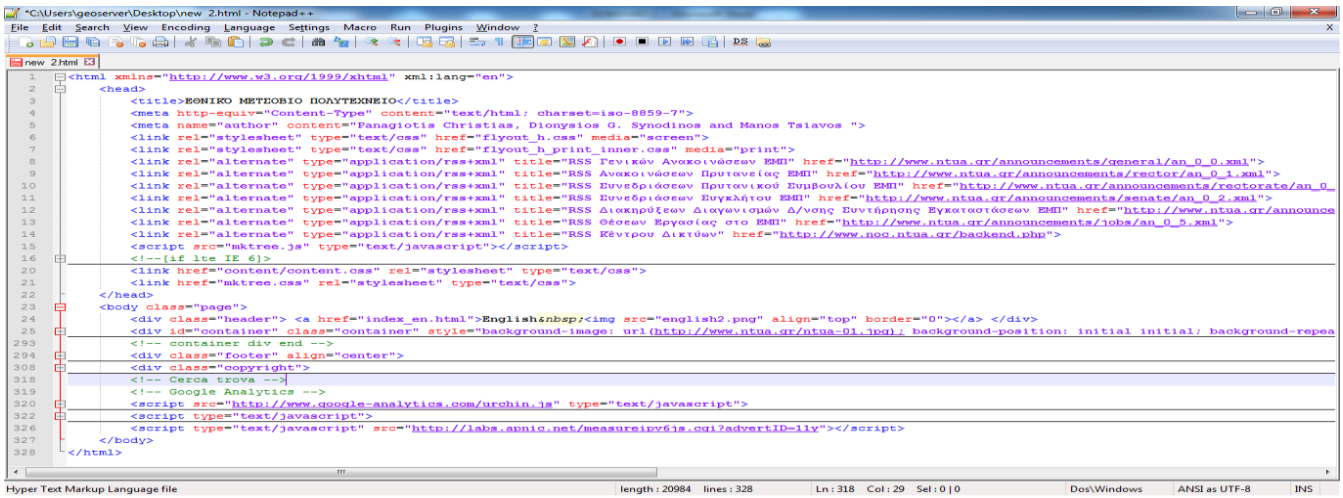
Μια πλήρη Διεύθυνση έχει την παρακάτω σύνταξη:

πρωτόκολλο://host.domain:port/διαδρομή/όνομα αρχείου

- **Πρωτόκολλο:** ορίζει το πρωτόκολλο μεταφοράς του αρχείου. Το πιο συνηθισμένο είναι το http (Hyper Text Transfer Protocol).
- **Host:** το καθορισμένο host για το πρωτόκολλο http είναι το www.
- **Domain:** ορίζει το όνομα του Domain Name, πχ. wlearn.gr, in.gr, disabled.gr
- **Port:** ορίζει τον αριθμό της πόρτας του server από τον οποίον διαβάζουμε το αρχείο. Συνήθως παραλείπεται από το URL. Η προκαθορισμένη πόρτα για το πρωτόκολλο http είναι η 80.
- **Διαδρομή:** ορίζει την διαδρομή που βρίσκεται το αρχείο στον server. Όταν παραλείπουμε την διαδρομή, τότε το αρχείο πρέπει να είναι στο root directory του website.
- **Όνομα αρχείου:** ορίζει το όνομα αρχείου. Σε περίπτωση που αυτό λείπει τότε ο browser ψάχνει για τα αρχεία index.asp ή index.php ή index.html (ανάλογα με τις ρυθμίσεις του server).

2.3.6 Παράδειγμα HTML

Το παρακάτω είναι παράδειγμα HTML κώδικα από την ιστοσελίδα του Ε.Μ.Π.



2.4 Web GIS

Τα Web GIS είναι ένα κεντρικό δίκτυο GIS που χρησιμοποιεί το Internet ως πρωταρχικό μέσο για την παροχή πρόσβασης σε κατανεμημένα δεδομένα και άλλες πληροφορίες, διάσπαρτη χωρική πληροφορία και διεξαγωγή ανάλυσης GIS. Ένα τέτοιο σύστημα επιτρέπει σε μία ποικιλία συσκευών να έχουν πρόσβαση σε χωρικά δεδομένα και εργαλεία επεξεργασίας σε εξυπηρετητές οι οποίοι βρίσκονται οπουδήποτε και σε οποιοδήποτε χρονικό διάστημα.

.

2.4.1 Ένα αλληλεπιδραστικό σύστημα βασισμένο στο διαδίκτυο

Ενώ τα παραδοσιακά επιτραπέζια συστήματα στηρίζονται σε ένα GUI έτσι ώστε οι χρήστες να αλληλεπιδρούν με το πρόγραμμα, τα Web GIS εξαρτώνται από το WWW και τα κατάλληλα addons για την παροχή αλληλεπίδρασης μεταξύ του χρήστη και του προγράμματος. Ως συνέπεια οι χρήστες έχουν την δυνατότητα να διαχειρίζονται δεδομένα και χάρτες ενεργά μέσω των ενσύρματων ή ασύρματων δικτύων. Μάλιστα μπορούν να εκτελέσουν λειτουργίες όπως παροχή χάρτη, σύνταξη χωρικών ερωτημάτων και χωρική ανάλυση χρησιμοποιώντας έναν Web browser ή άλλα προγράμματα.

Τα web GIS λόγω του ότι χρησιμοποιούν το διαδίκτυο, θεωρούνται ως ένα γιγάντιο κατανεμημένο σύστημα έτσι ώστε τα δεδομένα GIS και τα εργαλεία ανάλυσης να μπορούν να βρίσκονται σε διαφορετικούς υπολογιστές (ή εξυπηρετητές) πάνω στο Internet. Οι χρήστες μπορούν να έχουν πρόσβαση σ' αυτά τα δεδομένα και τις. εφαρμογές από οπουδήποτε μέσω ενός ασύρματου ή ενσύρματου δικτύου. Τα γεωχωρικά δεδομένα συνήθως κατανέμονται σε μία συγκεκριμένη τοποθεσία ή μπορεί να βρίσκονται κατανεμημένα σε διαφορετικές περιοχές στο Internet.

Παράλληλα υπάρχει διαθέσιμος ένας αυξημένος αριθμός βάσεων δεδομένων που παρέχονται στο κοινό από δημόσιες ή ιδιωτικές εταιρείες παροχής δεδομένων. Τα κατανεμημένα GIS εκμεταλλεύονται αυτά τα συστήματα και μπορούν να συντάξουν ερωτήματα και να εξάγουν δεδομένα από αυτές τις βάσεις δεδομένων, διατηρώντας τους στην αρχική θέση, παρά να γίνεται το κλασικό κατέβασμα στα τερματικά μηχανήματα και έπειτα ο συνδυασμός αυτών σε τοπικό επίπεδο. Με άλλα λόγια υπάρχει η δυνατότητα να μην κατεβαίνουν τα δεδομένα στον υπολογιστή και μετά να γίνεται η επεξεργασία, αλλά να γίνεται επεξεργασία ταυτόχρονα και ο χρήστης να έχει επαφή μόνο με τα τελικά αποτελέσματα.

Επιπροσθέτως, με τις κατανεμημένες βάσεις δεδομένων, μπορούν και τα εργαλεία χωρικής ανάλυσης να είναι κατανεμημένα μέσω του Internet.

Επειδή ένα web GIS είναι ένα κατανεμημένο σύστημα, οι βάσεις δεδομένων και τα προγράμματα εφαρμογών κείνται σε υπολογιστές που μπορούν να τα διανέμουν. Αυτό το σύστημα διατηρεί τα δεδομένα και τις εφαρμογές επίκαιρα. Με άλλα λόγια το web GIS είναι δυναμικά συνδεδεμένο με τις πηγές δεδομένων. Αυτή η δυναμική φύση του συστήματος επιτρέπει στα web GIS να είναι πιο ευέλικτα στη σύνδεση με πληροφορίες σε πραγματικό χρόνο όπως για παράδειγμα δορυφορικές εικόνες, κυκλοφοριακή κίνηση καθώς και πληροφορίες άμεσης απόκρισης.

2.4.2. Οι δυνατότητες των Web GIS

Τα web GIS είναι ένας ειδικός τύπος εργαλείων GIS που χρησιμοποιούν το Internet ως μέσο για την πρόσβαση και τη μετάδοση δεδομένων και εργαλείων ανάλυσης, για να διεξάγει χωρική ανάλυση και για να δημιουργήσει παρουσιάσεις πολυμέσων GIS. Το web GIS είναι αντικειμενοστραφές, κατανεμημένο και διαλειτουργικό. Ο τελικός χρήστης δεν χρειάζεται απαραίτητα να έχει GIS δεδομένα και λογισμικό εγκατεστημένα στον τοπικό του υπολογιστή, επειδή όλα τα δεδομένα και τα υποσύνολα ανάλυσης μπορεί να είναι διαθέσιμα στους εξυπηρετητές δικτύου και τα οποία μπορούν να ζητηθούν και να αποκτηθούν από τον τελικό χρήστη. Τοπικοί χρήστες σε διαφορετικά λειτουργικά συστήματα είναι ικανοί να έχουν πρόσβαση σε απομακρυσμένα δεδομένα και εργαλεία ανάλυσης σαν αυτά να ήταν αποθηκευμένα σε τοπικό επίπεδο.

2.5 Γεωγραφικές Βάσεις Δεδομένων

Οι γεωγραφικές βάσεις δεδομένων βασίζονται στην ύπαρξη ενός Συστήματος Διαχείρισης Βάσεων Δεδομένων που θα έχει δυνατότητες υποστήριξης χωρικών τύπων (γεωμετριών) όπως σημεία, γραμμές και επιφάνειες. Ένα ΣΔΒΔ μπορεί να υποστηρίζει χωρικούς τύπους αυτόνομα ή να βασίζεται σε μία χωρική επέκταση για αυτό το σκοπό. Οι γεωγραφικές βάσεις δεδομένων μπορεί να περιέχουν διανυσματικά (vector) ή ψηφιδωτά (raster) δεδομένα. Η συνήθης τακτική όμως είναι να εισάγονται στο ΣΔΒΔ μόνο τα διανυσματικά δεδομένα, ενώ τα raster αποθηκεύονται σε διάφορες θέσεις στο σύστημα αρχείων (file system) του η/υ που υποστηρίζει τη λειτουργία του Συστήματος Γεωγραφικών Πληροφοριών.

Η υποστήριξη χωρικών τύπων από τις βάσεις δεδομένων βασίζεται στην ύπαρξη του τύπου δεδομένων geometry, ο οποίος χρησιμοποιείται για να αναπαραστήσει μια ποικιλία χωρικών δεδομένων. Η ύπαρξη επίσης χωρικής υποστήριξης σημαίνει ότι το ΣΔΒΔ πρέπει να υποστηρίζει διάφορα συστήματα προβολής και μετασχηματισμούς μεταξύ τους, χωρικούς τελεστές (π.χ. το αντικείμενο Α επικαλύπτεται με το αντικείμενο Β) συναρτήσεις (π.χ. υπολογισμός του εμβαδού μίας επιφανειακής οντότητας) και χωρικά ευρετήρια.

Όταν δημιουργείται μία Χωρική Βάση Δεδομένων, κάθε οντότητα με χωρική υπόσταση (π.χ. ένα σημείο ενδιαφέροντος), είναι μία εγγραφή σε έναν πίνακα που περιέχει όλα τα αντίστοιχα σημεία, ενώ μεταξύ των πεδίων του πίνακα υφίσταται και ένα ξεχωριστό πεδίο για τη γεωμετρία του αντικειμένου (π.χ. τη θέση του σημείου ενδιαφέροντος, η οποία δίνεται από τις συντεταγμένες του).

2.5.1 PostgreSQL

Το λογισμικό της Βάσης Δεδομένων PostgresSQL, είναι μία από τις δημοφιλέστερες, παγκοσμίως, open source βάσεις δεδομένων με ισχυρή υποστήριξη τύπων γεωγραφικών δεδομένων και μεγάλης κλίμακας εφαρμογές και κοινότητες. Η PostgreSQL λειτουργεί σε Windows (95 / 98 / ME / NT / 2000 / XP), Linux (RedHat / Mandrake / Suse), MacOS X. Παρέχονται γλώσσες προγραμματισμού και interfaces: Perl, Python, C/C++, Embedded SQL, Delphi/Kylix/Pascal, VB, ASP, Java, ODBC, JDBC κ.α. Η διαχείριση γίνεται κυρίως μέσω του PgAdmin III, αλλά και άλλες εφαρμογές τρίτων: (PgAccess , PhpPgAdmin, WinSQL). Αν και η PostgreSQLυποστηρίζει από μόνη της χωρικούς τύπους δεδομένων, αυτοί δεν ακολουθούν το πρότυπο OGC (Open GIS Consortium).

Η PostgreSQL προέρχεται από το πακέτο POSTGRES, το οποίο γράφτηκε στο Πανεπιστήμιο του Μπέρκλεϋ στην Καλιφόρνια των Η.Π.Α.. Αναπτύσσεται πάνω από δύο δεκαετίες και αποτελεί πλέον την πιο προχωρημένη βάση δεδομένων ανοιχτού κώδικα.

Το σχέδιο POSTGRES υπό την ηγεσία του καθηγητή Michael Stonebreaker ξεκίνησε να εφαρμόζεται το 1986. Η POSTGRES έχει αρκετά σημαντικές διανομές από τότε. Η πρώτη δοκιμαστική έκδοση του συστήματος παρουσιάστηκε το 1988 στο συνέδριο ACM-SIGMOD και κυκλοφόρησε σε μικρή ομάδα ατόμων τον Ιούνιο του 1989. Η POSTGRES χρησιμοποιήθηκε για την εφαρμογή ποικίλων εφαρμογών παραγωγής και έρευνας, όπως για παράδειγμα ένα οικονομικό σύστημα ανάλυσης δεδομένων, ένα πακέτο παρακολούθησης της απόδοσης ενός κινητήρα τζετ, βάση δεδομένων εντοπισμού αστεροειδή, μιας ιατρικής βάσης δεδομένων και μερικών συστημάτων γεωγραφικής πληροφορίας. Η POSTGRES χρησιμοποιήθηκε επίσης ως εργαλείο εκπαίδευσης σε ορισμένα πανεπιστήμια.

Το 1994 προστέθηκε διερμηνέας γλώσσας SQL από τους Andrew Yu και Jolly Chen. Υπό την ονομασία Postgres95 απελευθερώθηκε στο διαδίκτυο ως απόγονος ανοιχτού κώδικα του αρχικού κώδικα POSTGRES Berkeley. Ο κώδικας της Postgres95 αποτελούνταν από ANSI C και περιορίστηκε το μέγεθός του κατά 25% σε σχέση με τον κώδικα της POSTGRES. Πολλές εσωτερικές αλλαγές βελτίωσαν την αποδοτικότητα και την δυνατότητα συντήρησής της.

Το 1996 έγινε σαφές ότι η ονομασία Postgres95 δεν θα αντέξει με την πάροδο του χρόνου, γιαυτό και επιλέχθηκε η ονομασία PostgreSQL ώστε να αντικατοπτρίσει την σχέση ανάμεσα στην αρχική POSTGRES και τις πιο πρόσφατες εκδόσεις με συμβατότητα SQL. Την ίδια χρονιά ορίστηκε η αρίθμηση των εκδόσεων να ξεκινά από το νούμερο 6.0, έτσι ώστε να συνεχίζεται από την αρίθμηση που ξεκίνησε από το αρχικό σχέδιο POSTGRES Berkeley.

2.5.2 PostGIS

Είναι μία επέκταση της PostgreSQL για να υποστηρίζει χωρικά δεδομένα, σύμφωνα με το πρότυπο του OGC. Παρέχει ειδικούς τελεστές για τη σύνταξη ερωτημάτων, λειτουργίες συνάθρισης επάνω σε χωρικά δεδομένα καθώς και χωρικές συναρτήσεις. Επιτρέπει επίσης την ανάθεση προβολικών συστημάτων στα χωρικά δεδομένα. Τέλος, μπορεί να χρησιμοποιηθεί για να οπτικοποιηθούν τα δεδομένα μέσω ειδικών εφαρμογών όπως το Quantum GIS και ο GeoServer.

2.5.3 Ο τύπος Geometry και ο μορφότυπος WKT

Ο τύπος Geometry του PostGIS περιέχει όλη τη γεωμετρία (τις συντεταγμένες) του κάθε αντικειμένου. Ο τύπος αυτός είναι γενικός και μπορεί να περιέχει χωρικά αντικείμενα των τριών συνηθέστερων κατηγοριών (σημείο, γραμμή, πολύγωνο), καθώς και συλλογές αυτών των αντικειμένων (δηλαδή πολλά πολύγωνα ή πολλά σημεία μαζί με κάποιες γραμμές κ.ο.κ.).

Η αναπαράσταση του τύπου Geometry σε μορφή πίνακα δε μπορεί να γίνει εύκολα αντιληπτή. Στην πραγματικότητα, κάτι τέτοιο είναι σχεδόν αδύνατο να συμβεί, από τη στιγμή που οι γεωμετρίες αναπαρίστανται από δεκαεξαδικές μορφές όπως η παρακάτω:

"0106000020340800000100000001030000000100000070000008BC3E53F90901C416C1C75
73A103504109A69A955B901C412032DD7CB70350418B00E19E04911C419FDD48A3B90350419
B060D012B911C4166D1451AAB035041FC5482022B911C41178E9219AB03504173B28BD23E91"

Γι αυτό το λόγο, το Open GIS Consortium προτείνει τη χρήση του μορφότυπου WELL-KNOWN TEXT (WKT) ο οποίος υποστηρίζεται από το PostGIS. Με το WKT μπορούν αναπαρασταθούν με έναν εύκολα αντιληπτό τρόπο, σημεία, γραμμές πολύγωνα καθώς και πολλαπλές γεωμετρίες (π.χ. πολυ-πολύγωνα) και συλλογές γεωμετριών. Το PostGIS παρέχει συναρτήσεις που μετατρέπουν το WKT (που μπορεί να διαβαστεί και να γίνει κατανοητό σχετικά εύκολα) σε γεωμετρίες τις παραπάνω μορφής (που δε μπορούν να είναι απευθείας κατανοητές από τους ανθρώπους) και αντίστροφα. Μάλιστα, γεωμετρίες που έχουν συναχθεί σύμφωνα με το WKT format μπορούν να δημιουργηθούν / αναπαρασταθούν σε οποιοδήποτε σύστημα είναι συμβατό με τις κατευθύνσεις που δίνονται από το OGC (π.χ. Oracle, DB2 κ.α.).

Παρακάτω παρατίθενται μερικά παραδείγματα του WKT format:

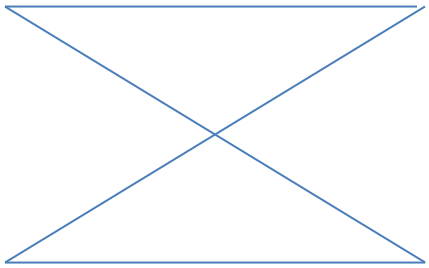
- Σημείο: 'POINT(X Y)'
- Γραμμή: 'LINESTRING(X1 Y1, X2 Y2, ...)'
- Πολύγωνο: 'POLYGON((X1 Y1, X2 Y2, .., X1 Y1), (Xn Yn, Xn+1 Yn+1,.., Xn Yn))'

όπου Xi και Yi οι συντεταγμένες του κάθε σημείου του αντικειμένου.

Το κάθε πολύγωνο μπορεί να περιγράφεται από πολλές σειρές συντεταγμένων μέσα σε παρενθέσεις (X1 Y1, X2 Y2, .., X1 Y1) που αρχίζουν και τελειώνουν στο ίδιο σημείο, από αυτές, η πρώτη σειρά είναι το εξωτερικό όριο του πολυγώνου, ενώ οι υπόλοιπες είναι πιθανά εσωτερικά όρια του πολυγώνου (νησίδες). Προφανώς ένα πολύγωνο μπορεί να μην έχει και καμία νησίδα. Το πολύγωνο επίσης μπορεί να μην είναι απλό (δηλαδή να είναι αυτοτεμνόμενο)

Μία αλλαγή στη σειρά των τεσσάρων σημείων στο παρακάτω σχήμα οδηγεί σε δύο εντελώς διαφορετικά πολύγωνα.

POLYGON ((0 0, 0 1,1 1, 1 0, 1 1))



POLYGON ((0 0, 1 1,0 1, 1 0, 0 0))

Εικόνα 2.6 – Παραδείγματα πολυγώνων με WTK

Επίσης, με το WKT format μπορούν να οριστούν και συλλογές αντικειμένων όπως παρακάτω:

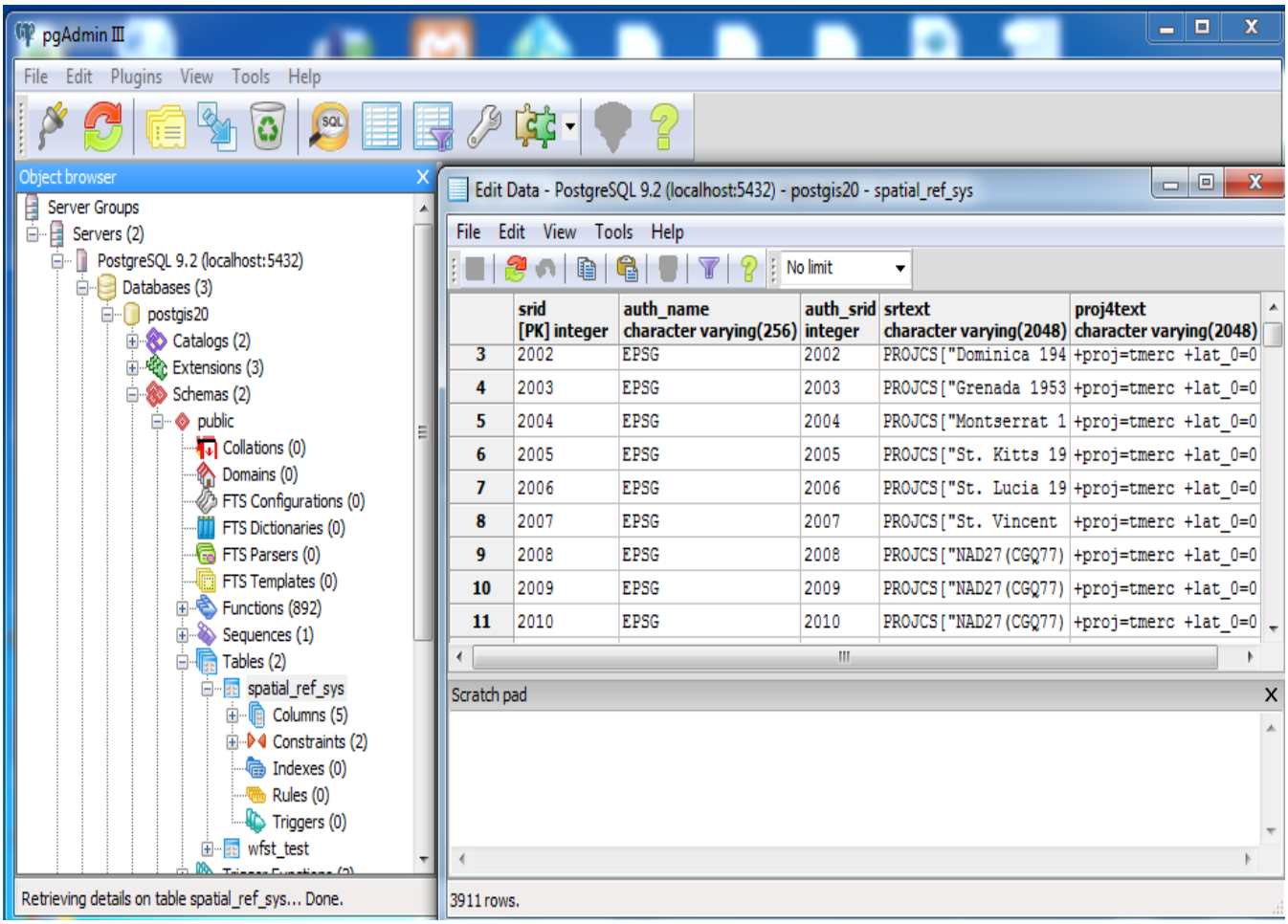
'GEOMETRYCOLLECTION(POINT(X1 Y1),LINESTRING(x2 y2,x3 y3))'

Εκτός από τους παραπάνω απλούς τύπους, υπάρχουν και οι πολλαπλοί τύποι MULTIPOINT, MULTILINESTRING, MULTIPOLYGON, που είναι απλές γενικεύσεις των προηγούμενων απλών τύπων και, π.χ., ο MULTIPOINT περιέχει πολλά σημεία. Ο μετασχηματισμός μεταξύ γεωμετριών και κειμενικής (WKT) μορφής των γεωμετρικών αντικειμένων γίνεται με τις συναρτήσεις GeomFromText.

Ο μετασχηματισμός μεταξύ γεωμετριών και κειμενικής (WKT) μορφής των γεωμετρικών αντικειμένων γίνεται με τις συναρτήσεις GeomFromText και AsText.

2.5.4 Υποστήριξη συστημάτων αναφοράς

Η PostGIS υποστηρίζει πλήθος Γεωγραφικών και Προβολικών συστημάτων συντεταγμένων κάθε ένα από τα οποία περιγράφεται από ένα SRID (από τα αρχικά Spatial Reference Identifier). Το SRID είναι ένας ακέραιος αριθμός (η ταυτότητα) που αντιστοιχεί στο σύστημα συντεταγμένων στο οποίο δίνονται οι συντεταγμένες του συγκεκριμένου αντικειμένου. Αναλυτικά, τα συστήματα συντεταγμένων που υποστηρίζονται από το PostGIS, μπορούν να βρεθούν μέσα στον πίνακα spatial_ref_sys μίας οποιαδήποτε βάσης δεδομένων του PostGIS.



Εικόνα 2.7 – Συστήματα αναφοράς στην PostGIS

2.6 XML

Η eXtensible Markup Language (XML) είναι μια γλώσσα ανεξάρτητη από σύστημα και υλικό για την αναπαράσταση δεδομένων και της μορφής τους σε ένα έγγραφο XML (XML document). Ένα έγγραφο XML στην πιο απλή του μορφή είναι ένα αρχείο κειμένου το οποίο περιέχει δεδομένα μαζί με σήμανση η οποία καθορίζει τη δομή των δεδομένων.

Η XML είναι μια παγκοσμίως συμφωνημένη μεταγλώσσα σήμανσης που χρησιμοποιείται πρώτιστα για την ανταλλαγή πληροφοριών. Η ομορφιά της XML βρίσκεται στο γεγονός ότι είναι επεκτάσιμη. Απλά, η XML είναι ένα σύνολο προκαθορισμένων κανόνων (συντακτικό πλαίσιο) που πρέπει να ακολουθήσουμε κατά τη δόμηση των δεδομένων μας.

Για ένα μεγάλο χρονικό διάστημα, οι προγραμματιστές και οι προμηθετές εφαρμογών κατασκεύαζαν εφαρμογές και συστήματα εγκατεστημένα σε μια επιχείρηση τα οποία επεξεργάζονταν δεδομένα τα οποία μπορούσαν να με το δικό τους ιδιωτικό τρόπο. Αλλά καθώς η ανταλλαγή πληροφορίας μεταξύ εφαρμογών και συστημάτων στις επιχειρήσεις επικρατούσε, έγινε πολύ δύσκολο να αναταλλάσεις δεδομένα διότι τα συστήματα δε σχεδιάστηκαν ώστε να δέχονται δεδομένα από εξωτερικά, άγνωστα συστήματα.

Η XML παρέχει μία πρότυπη και κοινή δομή για τη διανομή δεδομένων μεταξύ ανόμοιων συστημάτων. Επιπλέον η XML έχει ενσωματωμένο ένα μηχανισμό επικύρωσης δεδομένων, ο οποίος εγγυάται ότι η δομή των δεδομένων που λαμβάνεται είναι έγκυρη.

2.6.1 Η ιστορία της XML

Οι ρίζες της XML μπορούν να αναζητηθούν στην εκρηκτική ανάπτυξη του Παγκόσμιου Ιστού στα μέσα της δεκαετίας του 1990 και στους πολέμους των browser που έλαβαν χώρα μεταξύ της Microsoft Corporation και της Netscape Corporation, όπου καθεμιά από αυτές πάλευε για την απόλυτη κυριαρχία.

Καθώς το Διαδίκτυο γινόταν όλο και πιο μεγάλο , και όλο και περισσότεροι χρήστες το χρησιμοποιούσαν, άρχισαν να ανακαλύπτονται από τους προγραμματιστές που χρησιμοποιούσαν HTML, διάφορα προβλήματα:

Ο ίδιος πόρος HTML εμφανιζόταν με διάφορες μορφές , ανάλογα με τον browser που χρησιμοποιούνταν . Αυτό σήμαινε ότι οι σχεδιαστές των ιστοσελίδων έπρεπε το λιγότερο να διπλασιάσουν τις προσπάθειές τους.

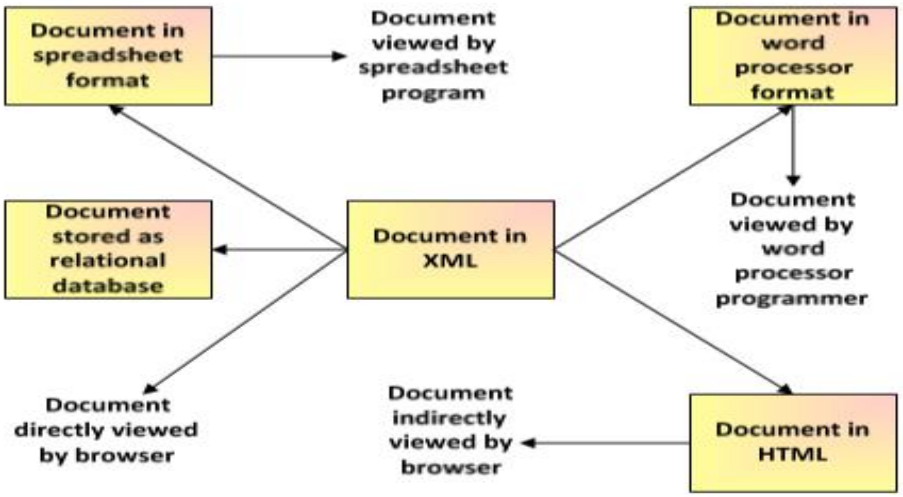
Ορισμένοι κατασκευαστές browser ανέπτυξαν εργαλεία HTML που δεν ήταν αναγνωρίσιμα από άλλους browser.

Ήταν σχεδόν αδύνατον να διακριθεί οποιαδήποτε σημαντική αλλαγή μέσα σε μια ιστοσελίδα: πουθενά αλλού αυτό δεν ήταν πιο προφανές από ότι στην χρήση των μηχανών αναζήτησης. Ακόμη και στα μέσα της δεκαετίας του 1990 πολλοί χρήστες εξέφραζαν την απογοήτευσή τους όσον αφορούσε αυτά τα προγράμματα εξαιτίας του όγκου των ανακτημένων αρχείων που επέστρεφαν οι μηχανές, τα οποία, στην καλύτερη περίπτωση, σχετίζονταν οριακά με την αναζήτηση. Η αιτία αυτής της "φτωχής" απόδοσης δεν ήταν η τεχνολογία των μηχανών αναζήτησης - στην πραγματικότητα επρόκειτο για εξεζητημένα προγράμματα που αποτελούν αποδεικτικό της ευφυΐας των προγραμματιστών - αλλά το ότι τα δεδομένα που επεξεργάζονταν οι σελίδες HTML δεν έδιναν πολλά στοιχεία για το περιεχόμενό τους.

Εξαιτίας αυτών των προβλημάτων η Κοινοπραξία Παγκόσμιου Ιστού, η ομάδα που ελέγχει την διαδικασία τυποποίησης του Ιστού , αποφάσισε το 1996 να αναπτύξει μία σημειακή γλώσσα που μελλοντικά θα υποσκέλιζε την HTML. Οι στόχοι αυτής της γλώσσας ήταν:

- Να χρησιμοποιείται εύκολα στο Internet.
- Να μπορεί να υποστηρίζει πολλές εφαρμογές οι οποίες θα κυμαίνονται από browser μέχρι βάσεις δεδομένων μηχανών αναζήτησης.
- Να είναι συμβατή με την SGML, την γλώσσα επεξεργασίας κειμένου που αποτέλεσε την έμπνευση για την HTML.
- Να μην αποτελεί πολύπλοκη διαδικασία η ανάπτυξη επεξεργαστών κειμένων γραμμένων σε γλώσσες που θα βασίζονταν σε XML, για παράδειγμα θα έπρεπε να είναι εύκολη η εγγραφή ενός προγράμματος για τον έλεγχο της σαφήνειας ενός κειμένου πόρου.
- Ο αριθμός των προαιρετικών εργαλείων της γλώσσας να είναι χαμηλός.
- Να είναι εύκολη η ανάγνωση και κατανόηση των αρχείων XML.
- Να είναι εύκολο να αναπτυχθούν με την χρήση απλών συντακτών, αρχεία γραμμένα σε γλώσσα βασιζόμενη σε XML.

Το 1998 η γλώσσα XML παρουσιάστηκε παγκόσμια ως μία τελευταία υπόδειξη της Κοινοπραξίας Παγκόσμιου Ιστού. Σαν αποτέλεσμα, έγινε μία σταθερά για το Internet. Βασιζόταν στην γλώσσα επεξεργασίας κειμένων SGML, η οποία αποτέλεσε την έμπνευση για την HTML. Ο ρόλος της XML συνοψίζεται στο παρακάτω σχήμα, το οποίο παρουσιάζει την σχέση της με άλλες μορφές αποθήκευσης και παρουσίασης.



Εικόνα 2.8 – Δομή της XML

2.6.2 Ιδιότητες της XML

- **Η XML είναι μία γλώσσα για τη δόμηση δεδομένων**

Με την έννοια δομημένα δεδομένα εννοούμε μία συλλογή στοιχείων δεδομένων όπως είναι για παράδειγμα τα λογιστικά φύλλα, οι κατάλογοι διευθύνσεων, οι παράμετροι διαμόρφωσης, οι οικονομικές συναλλαγές και τα

τεχνικά σχέδια. Η XML είναι, δηλαδή, ένα σύνολο κανόνων (ή διαφορετικά ένα πακέτο κατευθυντήριων γραμμών ή συμβάσεων) για το σχεδιασμό μορφών κειμένου οι οποίες διευκολύνουν τη δόμηση των δεδομένων σας. Η XML διευκολύνει τον υπολογιστή να παράγει δεδομένα, να διαβάζει δεδομένα και να εξασφαλίζει τη σαφήνεια της δομής των δεδομένων. Η XML αποφεύγει τις συνήθεις παγίδες του σχεδιασμού γλωσσών: είναι επεκτάσιμη , ανεξάρτητη συστήματος υλικού και μπορεί να υποστηρίξει διεθνείς και τοπικές προσαρμογές. Η XML είναι πλήρως συμβατή με Unicode όπως αναφέρθηκε και παραπάνω.

- **Η XML θυμίζει την HTML.**

Η XML, όπως η HTML, χρησιμοποιεί ετικέτες (tags) (λέξεις μέσα σε γωνιακές αγκύλες '<' και '>') και γνωρίσματα (τύπου όνομα = " τιμή "). Σε αντίθεση με την HTML η οποία διευκρινίζει τη σημασία κάθε ετικέτας και γνωρίσματος και συχνά προσδιορίζει πως θα εμφανίζεται σε φυλλομετρητή το κείμενο το οποίο περιλαμβάνεται σε αυτά , η XML χρησιμοποιεί ετικέτες μόνο για να οριοθετήσει κομμάτια δεδομένων και αφήνει την ερμηνεία των δεδομένων στην εφαρμογή που τα διαβάζει .

- **Η XML είναι κείμενο αλλά δεν προορίζεται για ανάγνωση.**

Τα προγράμματα που παράγουν λογιστικά φύλλα , καταλόγους διευθύνσεων και άλλα δομημένα δεδομένα αποθηκεύουν, συχνά, τα εν λόγω δεδομένα στο σκληρό δίσκο, χρησιμοποιώντας δυαδική μορφή ή μορφή κειμένου . Ένα από τα πλεονεκτήματα της μορφής κειμένου είναι ότι επιτρέπει στο χρήστη, εάν είναι αναγκαίο , να δει τα δεδομένα χωρίς το πρόγραμμα που τα παρήγαγε . Οι μορφές κειμένου επιτρέπουν, επίσης, στους κατασκευαστές λογισμικού να εκσφαλματώνουν εφαρμογές με μεγαλύτερη ευκολία. Όπως και τα αρχεία HTML, τα αρχεία XML είναι αρχεία κειμένου τα οποία δεν προορίζονται για ανάγνωση αλλά προσφέρουν αυτή τη δυνατότητα στο χρήστη εάν προκύψει ανάγκη. Ωστόσο, οι κανόνες των αρχείων XML είναι αυστηροί σε αντίθεση με τα αρχεία HTML. Η παράληψη μίας ετικέτας ή ένα γνώρισμα δίχως αγκύλες καθιστά άχρηστο το αρχείο XML ενώ η HTML ανέχεται τέτοιου είδους παραλήψεις και συχνά τις επιτρέπει εξολοκλήρου. Η επίσημη προδιαγραφή της XML δεν επιτρέπει σε εφαρμογές να προσπαθούν να μαντέψουν ποιο είναι το πρόγραμμα δημιουργός ενός αρχείου XML με χαμένο σύνδεσμο. Εάν ο σύνδεσμος του αρχείου παρουσιάζει πρόβλημα, η εφαρμογή πρέπει να σταματήσει και να αναφέρει το σφάλμα.

- **Η XML είναι "φλύαρη" γλώσσα.**

Η XML εμφανίζεται υπό μορφή κειμένου και χρησιμοποιεί ετικέτες για την οριοθέτηση των δεδομένων και για τον λόγο αυτό τα αρχεία XML είναι σχεδόν πάντα μεγαλύτερα σε έκταση από συγκρίσιμα αρχεία σε δυαδική μορφή. Πρόκειται για συνειδητή επιλογή των σχεδιαστών της XML. Τα πλεονεκτήματα ενός αρχείου υπό μορφή κειμένου είναι ολοφάνερα και τα μειονεκτήματα αντισταθμίζονται συνήθως σε άλλο επίπεδο. Η χωρητικότητα του σκληρού δίσκου δεν είναι τόσο ακριβή όσο παλαιότερα και προγράμματα όπως το zip και το gzip μπορούν να συμπιέσουν αρχεία αποτελεσματικά και γρήγορα. Επιπρόσθετα, πρωτόκολλα επικοινωνίας όπως τα πρωτόκολλα μόντεμ και το HTTP/1.1, το οποίο είναι το πρωτόκολλο πυρήνας του Ιστού, μπορούν να συμπιέσουν πολύ εύκολα αρχεία με μεγάλη ταχύτητα μεταφοράς και το ίδιο αποτελεσματικά όσο και τα δυαδικά αρχεία .

- **Η XML συνδυάζει διαφορετικές τεχνολογίες.**

Η XML 1.0 είναι η προδιαγραφή που ορίζει τι είναι οι " ετικέτες " και τα " γνωρίσματα ". Πέρα από την XML 1.0, " η οικογένεια XML" είναι ένα διαρκώς αναπτυσσόμενο σύνολο λειτουργικών μονάδων οι οποίες

προσφέρουν χρήσιμες υπηρεσίες για τη διεκπεραίωση σημαντικών έργων τα οποία ανακύπτουν συχνά . Η Xlink περιγράφει έναν προκαθορισμένο τρόπο εισαγωγής υπερσυνδέσμων σε αρχεία XML. Τα XPointer και τα XFragments είναι συντακτικά υπό διαμόρφωση για την υπόδειξη θέσεων ενός εγγράφου XML. Το XPointer μοιάζει λίγο με URL αλλά αντί να υποδεικνύει έγγραφα στον Ιστό, υποδεικνύει κομμάτια πληροφοριών ενός εγγράφου XML. Το CSS, η γλώσσα μορφοποίησης σελίδων, είναι δυνατό να εφαρμοστεί σε XML όπως και σε HTML. Το XSL είναι προηγμένη γλώσσα (advanced language) μορφοποίησης σελίδων. Βασίζεται στο XSLT, μία γλώσσα μετασχηματισμού η οποία χρησιμοποιείται για την αναδιάταξη , την πρόσθεση και την διαγραφή ετικετών και γνωρισμάτων . Το DOM είναι ένα προκαθορισμένο σύνολο λειτουργιών για τη διαχείριση αρχείων XML (και HTML) από μία γλώσσα προγραμματισμού. Τα XML Schemas 1 και 2 επιτρέπουν στους κατασκευαστές λογισμικού να ορίσουν με ακρίβεια τις δομές των δικών τους μορφών XML. Υπάρχουν αρκετά εργαλεία και λειτουργικές μονάδες τα οποία βρίσκονται υπό διαμόρφωση ή είναι ήδη διαθέσιμα.

- **Η XML είναι καινούρια, όχι όμως εντελώς καινούρια.**

Η ανάπτυξη της XML ξεκίνησε το 1996. Από το Φεβρουάριο του 1998 η XML αποτελεί Σύσταση του W3C. Ίσως , λοιπόν να θεωρήσετε ότι η XML δεν έχει ωριμάσει ακόμα τεχνολογικά. Στην πραγματικότητα, όμως, η τεχνολογία XML δεν είναι τόσο καινούρια. Πριν από την XML υπήρχε η SGML, η οποία αναπτύχθηκε στις αρχές της δεκαετίας του '80, τυποποιήθηκε από τον ISO το 1986, και χρησιμοποιήθηκε ευρέως σε προγράμματα με εκτεταμένη τεκμηρίωση. Η ανάπτυξη της HTML ξεκίνησε το 1990. Οι σχεδιαστές της XML επέλεξαν τα καλύτερα τμήματα της SGML, χρησιμοποίησαν την εμπειρία που είχαν αποκτήσει κατά την ανάπτυξη της HTML και παρήγαγαν μία γλώσσα η οποία δεν είναι λιγότερο ισχυρή από την SGML αλλά είναι πιο κανονικοποιημένη και πολύ πιο εύχρηστη. Είναι λοιπόν δύσκολο να διακρίνει κανείς την εξελικτική από την επαναστατική πρόοδο . Αξίζει να σημειωθεί, τέλος, ότι ενώ η SGML χρησιμοποιείται κυρίως για τεχνική τεκμηρίωση, και πολύ λιγότερο για δεδομένα άλλου είδους, για την XML ισχύει ακριβώς το αντίθετο.

- **Η XML οδηγεί την HTML σε XHTML.**

Μία από τις εφαρμογές XML υπάρχει υπό μορφή εγγράφου : πρόκειται για την XHTML του W3C, τη διάδοχο της HTML. Η XHTML διαθέτει αρκετά κοινά στοιχεία με την HTML. Το συντακτικό, όμως, έχει αλλάξει έτσι ώστε να συμβαδίζει με τους κανόνες της XML. Τα έγγραφα με βάση την XML χρησιμοποιούν το συντακτικό της XML, με ορισμένους όμως , περιορισμούς και πρόσθεση σημασίας στο συντακτικό.

- **Η XML επιδέχεται συνδυασμό διαφορετικών μορφών.**

Η XML επιτρέπει στο χρήστη τον ορισμό νέας μορφής εγγράφου προσφέροντάς του τη δυνατότητα να συνδυάσει και να χρησιμοποιήσει άλλες μορφές . Ωστόσο, επειδή δύο διαφορετικές μορφές, οι οποίες έχουν αναπτυχθεί ανεξάρτητα, ενδέχεται να διαθέτουν στοιχεία ή γνωρίσματα με το ίδιο όνομα, πρέπει να αποδοθεί ιδιαίτερη προσοχή κατά το συνδυασμό των δύο μορφών. Για την αποφυγή σύγχυσης ονομάτων κατά το συνδυασμό μορφών, η XML παρέχει ένα μηχανισμό namespace. Παραδείγματα μορφών με βάση την XML οι οποίες χρησιμοποιούν namespaces είναι η XSL και η RDF.

- **Η XML αποτελεί τη βάση του RDF και του Σημασιολογικού Ιστού.**

Ο Σκελετός Περιγραφής Πόρων του W3C (Resource Description Framework) (RDF) είναι μία μορφή κειμένου XML η οποία υποστηρίζει περιγραφή πόρων και εφαρμογές μεταδεδομένων, όπως οι κατάλογοι μουσικής, οι συλλογές φωτογραφιών και οι βιβλιογραφίες. Για παράδειγμα, το RDF έχει τη δυνατότητα αναγνώρισης

προσώπων σε ένα άλμπουμ φωτογραφιών του Ιστού χρησιμοποιώντας πληροφορίες από μία προσωπική λίστα επαφών . Στη συνέχεια , ο πελάτης ηλεκτρονικού ταχυδρομείου (mail client), μπορεί να αποστείλει μηνύματα σε όσους εμφανίζονται στις φωτογραφίες ειδοποιώντας τους ότι οι φωτογραφίες τους έχουν δημοσιευθεί στον Ιστό. Και βέβαια, όπως οι άνθρωποι έχουν συμφωνήσει να χρησιμοποιούν κοινές ονομασίες για τις σημασίες των λέξεων που χρησιμοποιούν όταν επικοινωνούν, έτσι και οι υπολογιστές χρειάζονται μηχανισμούς οι οποίοι να ορίζουν κοινά ονόματα για τους όρους ώστε να είναι εφικτή η αποτελεσματική επικοινωνία . Οι επίσημες περιγραφές όρων που ανήκουν σε ένα συγκεκριμένο νοηματικό πεδίο (για παράδειγμα αυτό των αγορών ή των κατασκευών) ονομάζονται οντολογίες και συνιστούν σημαντικό τμήμα του Σημασιολογικού Ιστού.

- **Η XML δεν χρειάζεται άδεια χρήσης, λειτουργεί ανεξαρτήτως συστήματος και τυγχάνει ευρείας υποστήριξης.**

Η επιλογή της XML προσφέρει πρόσβαση σε μια μεγάλη και διαρκώς αναπτυσσόμενη κοινότητα εργαλείων και ειδικών με μεγάλη εμπειρία στις εν λόγω τεχνολογίες. Αν ο χρήστης διαλέξει την XML είναι σαν να διαλέγει SQL για βάσεις δεδομένων: πρέπει να δημιουργήσει τη δική του βάση δεδομένων και τα δικά του προγράμματα και διαδικασίες για τη διαχείρισή της. Και επειδή η XML δεν χρειάζεται άδεια χρήσης μπορεί να κατασκευάσει πάνω της το δικό του λογισμικό δίχως να πρέπει να πληρώσει. Επίσης, τυγχάνει ευρείας και ολοένα επεκτεινόμενης υποστήριξης που σημαίνει ότι δεν δεσμεύεται ο χρήστης σε ένα μόνο κατασκευαστή. Η XML δεν είναι πάντα η καλύτερη λύση αλλά αξίζει να ληφθεί υπόψη .

2.6.3 Elements ή Στοιχεία

Τα στοιχεία (elements) είναι ετικέτες, όπως και στην HTML, και περιέχουν τιμές. Επιπλέον τα elements είναι δομημένα σαν δένδρο. Ως εκ τούτου έχουμε τα στοιχεία οργανωμένα σε ένα ιεραρχικό τρόπο με ένα στοιχείο-πατέρα και στοιχεία-παιδιά. Τα στοιχεία-παιδιά μπορούν να περιέχουν και αυτά άλλα στοιχεία-παιδιά και ούτω καθ'εξής.

Τα στοιχεία έχουν συγκεκριμένα χαρακτηριστικά. Ορισμένα από αυτά είναι:

1. Τα στοιχεία μπορεί να περιέχουν δεδομένα, οπως το στοιχείο <number> στο παράδειγμα.
2. Αντίστροφα, τα στοιχεία μπορεί να μην περιέχουν δεδομένα αλλά μόνο ιδιότητες, όπως το στοιχείο <shift>.
3. Εναλλακτικά, τα στοιχεία μπορεί να περιέχουν ταυτόχρονα και ιδιότητες αλλά και δεδομένα, αλλά επίσης και στοιχεία-παιδιά, όπως το στοιχείο <phone>.

Τα στοιχεία έχουν κάποιους κανόνες:

1. Όλα τα στοιχεία πρέπει να έχουν ετικέτα κλεισίματος αντίθετα με την HTML όπου υπάρχουν και ετικέτες που δε χρειάζονται κλείσιμο όπως για παράδειγμα η
.
2. Οι ετικέτες των στοιχείων είναι case sensitive δηλαδή υπάρχει διαχωρισμός μεταξύ κεφαλαίων και πεζών και τα ονόματά τους υπακούουν σε κανόνες ονοματολογίας.
3. Τα στοιχεία πρέπει να είναι τοποθετημένα σωστά αντίθετα με την HTML:

HTML : This text is bold and italic

XML : This text is bold and italic

4. Τα έγγραφα της XML πρέπει να έχουν ακριβώς ένα αρχικό στοιχείο (root element).

2.6.4 Attributes ή Ιδιότητες

Οι ιδιότητες (attributes) μας βοηθούν να δώσουμε περισσότερο νόημα και να περιγράψουμε τα στοιχεία μας πιο αποτελεσματικά και με σαφήνεια. Αυτό βοηθάει στο να κάνουμε τα δεδομένα σε ένα έγγραφο XML αυτοπεριγραφικά. Πρέπει πάντα να θυμόμαστε ότι ο κύριος σκοπός των ιδιοτήτων είναι να παρέχουν περισσότερη πληροφορία σχετική με ένα στοιχείο και δεν πρέπει να χρησιμοποιούνται για να περιέχουν τα ίδια τα δεδομένα.

Όπως και τα στοιχεία έτσι και οι ιδιότητες έχουν κάποιους κανόνες:

1. Οι τιμές των ιδιοτήτων πρέπει να είναι εσωκλείωνται σε “” ή σε ‘’.
2. Τα ονόματα των ιδιοτήτων ακολουθούν τους ίδιους κανόνες με αυτά των ετικετών.

2.6.5 Κανόνες ονομασίας

Τα στοιχεία και οι ιδιότητες στην XML πρέπει να ακολουθούν στους παρακάτω κανόνες:

1. Τα ονόματα μπορούν να περιέχουν γράμματα, αριθμούς και άλλους χαρακτήρες.
2. Τα ονόματα δεν πρέπει να ξεκινούν με αριθμό ή χαρακτήρα στίξης.
3. Τα ονόματα δεν πρέπει να ξεκινούν με τα γράμματα xml (ή XML, ή Xml κλπ.).
4. Τα ονόματα δεν μπορούν να περιέχουν κενά.

2.6.6 Καλά διαμορφωμένα έγγραφα (well formed documents)

Ένα «καλά διαμορφωμένο» έγγραφο XML είναι ένα έγγραφο που υπακούει στους κανόνες σύνταξης της XML που αναφέραμε προηγουμένως:

1. Τα έγγραφα XML πρέπει να περιέχουν ένα αρχικό στοιχείο.
2. Τα στοιχεία XML πρέπει να έχουν ετικέτες κλεισίματος.
3. Στις ετικέτες XML υπάρχει διαχωρισμός κεφαλαίων και πεζών.
4. Τα στοιχεία XML πρέπει να είναι σωστα τοποθετημένα.
5. Οι ιδιότητες XML πρέπει να εσωκλείονται πάντα σε “” ή ‘’.

2.6.7 DTD

Όπως σε μία γλώσσα προγραμματισμού πρέπει να ξέρουμε τις προδιαγραφές της γλώσσας, με παρόμοιο τρόπο το Document Type Definition (DTD) είναι μία προδιαγραφή, η οποία πρέπει να ακολουθηθεί όταν δημιουργούμε ένα έγγραφο XML. Επίσης, όπως μία από τις εργασίες του μεταγλωττιστή για κάθε γλώσσα προγραμματισμού είναι να ελέγξει αν η προδιαγραφές ακολουθήθηκαν, με παρόμοιο τρόπο υπάρχουν XML parsers οι οποίοι χρησιμοποιούν το DTD για να ελέγξουν την εγκυρότητα ενός εγγράφου XML5.

Ένα DTD μας βοηθάει να καθορίσουμε τη δομή ενός εγγράφου XML. Μας παρέχει ένα αυστηρό πλαίσιο και κανόνες οι οποίοι θα ακολουθηθούν όταν δημιουργούμε έγγραφα XML. Επιπρόσθετα, το DTD μπορεί να χρησιμοποιηθεί για τον έλεγχο της εγκυρότητας και της ακεραιότητας των δεδομένων που περιέχονται σε ένα έγγραφο XML.

Μερικά χαρακτηριστικά του DTD είναι τα παρακάτω:

1. Το DTD χρησιμοποιείται για να καθορίσει έγκυρα στοιχεία και ιδιότητες που μπορούν να χρησιμοποιηθούν σε ένα έγγραφο XML.
2. Με ένα DTD μπορούμε να καθορίσουμε μια ιεραρχική δομή στοιχείων.
3. Σε ένα DTD μπορεί επίσης να καθοριστεί η διαδοχική οργάνωση μιας συλλογής στοιχείων-παιδιών τα οποία μπορούν να υπάρχουν σε ένα έγγραφο XML.

Ένα DTD μπορεί να χρησιμοποιηθεί απευθείας μέσα σε ένα έγγραφο XML ή μπορεί να υπάρχει εκτός του εγγράφου XML. Στη δεύτερη περίπτωση θα αναφέρεται με ένα δεσμό μέσα στο έγγραφο XML που δείχνει σε αυτό το DTD.

Βασικά το DTD αποτελείται από τα παρακάτω στοιχεία:

Στοιχείο	Περιγραφή
DTD Element	Μεταδεδομένα για ένα στοιχείο. Καθορίζει τι είδους δεδομένα θα έχει το στοιχείο, τον αριθμό των περιστατικών κάθε στοιχείου, τις σχέσεις μεταξύ των στοιχείων και ούτω καθ'εξής.
DTD Attributes	Καθορίζει διάφορους κανόνες και ορισμούς που σχετίζονται με τα δεδομένα.
DTD Entities	Χρησιμοποιείται για να αναφέρει ένα εξωτερικό αρχείο ή για να παρέχει συντομεύσεις σε κοινό κείμενο.

Πίνακας 2.3 – Τα στοιχεία του DTD

Εν ολίγοις, ένα DTD χρησιμοποιείται για να καθορίσει μια δομή εγγράφων με τη διευκρίνιση των λεπτομερειών σχετικά με όλα τα στοιχεία και τις ιδιότητες που πρόκειται να χρησιμοποιηθούν σε ένα έγγραφο XML. Ως εκ τούτου μπορεί να χρησιμοποιηθεί για να ελέγξει την εγκυρότητα ενός εγγράφου XML που υποτίθεται ότι ακολουθεί τους κανόνες που καθορίζονται από αυτό το DTD5.

2.6.8 XML Schema

Το XML Schema είναι μια πιο προηγμένη έκδοση του DTD. Το DTD έχει πολλά μειονεκτήματα σε σχέση με το schema, όπως το ότι δεν υποστηρίζει ισχυρούς τύπους δεδομένων, έχει σύνταξη διαφορετική από την XML και δεν είναι επεκτάσιμο. Το XML Schema παρουσιάστηκε για να υπερνικήσει αυτά τα μειονεκτήματα.

Οι δύο κύριοι στόχοι του W3c XML Schema working group κατά τη διάρκεια του σχεδιασμού του προτύπου του XML Schema ήταν:

1. Να μπορέσουν να εκφράσουν μέσα στο πρότυπο αρχές αντικειμενοστραφούς σχεδιασμού οι οποίες μπορούν να βρεθούν σε όλες τις αντικειμενοστραφείς γλώσσες προγραμματισμού.
2. Να παρέχουν υποστήριξη για σύνθετους τύπους δεδομένων παρόμοια με την υποστήριξη που υπάρχει στις περισσότερες σχεσιακές βάσεις δεδομένων.

Τα κυριότερα χαρακτηριστικά του XML Schema είναι τα παρακάτω:

1. Η σύνταξη είναι όμοια με της XML. Αυτό σημαίνει ότι μπορούμε να επεξεργαστούμε το schema με οποιοδήποτε επεξεργαστή XML.
2. Δεν καθορίζουμε μόνο βασικούς τύπους δεδομένων όπως αλφαρηθμητικό, ακέραιος, πραγματικός και ούτω καθ'εξής αλλά μπορούμε επίσης να καθορίσουμε δικούς μας τύπους δεδομένων. Για παράδειγμα

   <xs:element name="name" type="xs:string" />

   Οι νέοι τύποι που μπορούμε να καθορίσουμε μπορεί να είναι απλοί ή σύνθετοι. Οι σύνθετοι τύποι μπορεί να περιέχουν και άλλα στοιχεία ή και ιδιότητες, ενώ οι απλοί τύποι όχι. Αντίθετα μπορούν να περιέχουν μόνο δεδομένα.
3. Το XML Schema παρέχει επικύρωση βασισμένη στο περιεχόμενο (content-based validation) δηλαδή μπορεί να ορίσει την σειρά με την οποία τα στοιχεία-παιδιά εμφανίζονται. Επίσης παρέχει επικύρωση στους ίδιους τους τύπους δεδομένων.
4. Το XML Schema μας παρέχει τη δυνατότητα να επεκτείνουμε άλλα έγγραφα το οποίο δεν είναι τίποτα άλλο παρά κληρονομικότητα. Αυτό σημαίνει ότι μπορούμε να παράγουμε νέους τύπους δεδομένων βάσει παλαιών τύπων.
5. Το XML Schema παρέχει υποστήριξη για Namespaces (χρησιμοποιώντας URI). Παρέχει σε κάθε στοιχείο ένα μοναδικό αναγνωριστικό, με το οποίο αποφεύγονται συγκρούσεις ονομάτων μεταξύ των στοιχείων.
6. Τέλος το XML Schema είναι εύκολα επεκτάσιμο για να ενσωματώσει και άλλες λειτουργίες στο μέλλον.

Ένα από τα μεγαλύτερα πλεονεκτήματα του XML Schema είναι ότι χρησιμοποιεί XML για τη σύνταξή του. Χρησιμοποιώντας XML, υπάρχοντες XML parsers μπορούν να χρησιμοποιηθούν σε συνδυασμό με ελεγκτές schema για να παρέχουν υπηρεσίες ελέγχου καλής διαμόρφωσης (well-formedness) και επικύρωσης (validation).

Το XML Schema προσφέρει έναν αυτοματοποιημένο μηχανισμό για επικύρωση των εγγράφων XML. Έχει παρατηρηθεί ότι σε τυπικό πρόγραμμα, μέχρι και 60% του κώδικα ξοδεύεται σε έλεγχο δεδομένων. Αν τα δεδομένα μας είναι δομημένα σαν XML, και υπάρχει και ένα schema, μπορούμε να περάσουμε τον έλεγχο των δεδομένων σε ένα ελεγκτή schema (schema validator). Κατά συνέπεια, μπορούμε να μειώσουμε τον κώδικά μας μέχρι και 60%. Επίσης, την επόμενη φορά που θα αλλάξουν οι περιορισμοί των δεδομένων μας ή

προσθέσουμε και άλλα στοιχεία, δεν θα χρειαστεί να γράψουμε νέο κώδικα για τον έλεγχο των δεδομένων μας αλλά απλώς να αλλάξουμε το schema.

2.7 Javascript

Η ανάγκη παράκαμψης της στατικότητας του περιεχομένου της HTML οδήγησε στην ανάπτυξη της φιλοσοφίας του προγραμματισμού από την πλευρά του πελάτη. Η JavaScript είναι μια προγραμματισμού από την πλευρά του πελάτη. Οι εφαρμογές της Javascript στο δυναμικό προγραμματισμό παγκόσμιου ιστού είναι πάρα πολλές. Η Javascript ακολουθεί τη φιλοσοφία των γλωσσών δομημένου προγραμματισμού, και βασίζεται στην υλοποίηση αντικειμένων (objects), γεγονότων (events) και μεθόδων(methods), η χρήση των οποίων συνιστά τον κύριο μηχανισμό αλληλεπίδρα σης της Javascript με τις ιστοσελίδες HTML.

2.7.1 Η ιστορία της Javascript

Η γλώσσα προγραμματισμού JavaScript δημιουργήθηκε αρχικά από τον Brendan Eich της εταιρείας Netscape με την επωνυμία Mocha. Αργότερα, Mocha μετονομάστηκε σε LiveScript, και τελικά σε JavaScript, κυρίως επειδή η ανάπτυξή της επηρεάστηκε περισσότερο από τη γλώσσα προγραμματισμού Java. LiveScript ήταν το επίσημο όνομα της γλώσσας όταν για πρώτη φορά κυκλοφόρησε στην αγορά σε βήτα (beta) εκδόσεις με το πρόγραμμα περιήγησης στο Web, Netscape Navigator εκδοχή 2.0 τον Σεπτέμβριο του 1995. LiveScript μετονομάστηκε σε JavaScript σε μια κοινή ανακοίνωση με την εταιρεία Sun Microsystems στις 4 Δεκεμβρίου, 1995, όταν επεκτάθηκε στην έκδοση του προγράμματος περιήγησης στο Web, Netscape εκδοχή 2.0B3.

Η JavaScript απέκτησε μεγάλη επιτυχία ως γλώσσα στην πλευρά του πελάτη (client-side) για εκτέλεση κώδικα σε ιστοσελίδες, και περιλήφθηκε σε διάφορα προγράμματα περιήγησης στο Web. Κατά συνέπεια, η εταιρεία Microsoft ονόμασε την εφάρμογή της σε JScript για να αποφύγει δύσκολα θέματα εμπορικών σημάτων. JScript πρόσθεσε νέους μεθόδους για να διορθώσει τα Y2K-προβλήματα στην JavaScript, οι οποίοι βασίστηκαν στην java.util.Date τάξη της Java. JScript περιλήφθηκε στο πρόγραμμα Internet Explorer εκδοχή 3.0, το οποίο κυκλοφόρησε τον Αύγουστο του 1996.

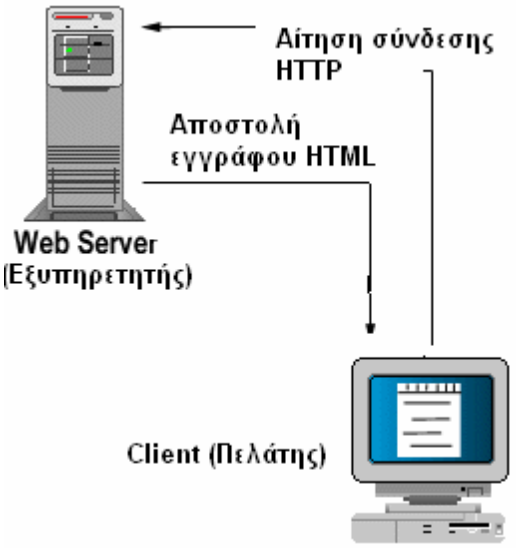
Τον Νοέμβριο του 1996, η Netscape ανακοίνωσε ότι είχε υποβάλει τη γλώσσα JavaScript στο Ecma International (μια οργάνωση της τυποποίησης των γλωσσών προγραμματισμού) για εξέταση ως βιομηχανικό πρότυπο, και στη συνέχεια το έργο είχε ως αποτέλεσμα την τυποποιημένη μορφή που ονομάζεται ECMAScript.

Η JavaScript έχει γίνει μία από τις πιο δημοφιλείς γλώσσες προγραμματισμού ηλεκτρονικών υπολογιστών στον Παγκόσμιο Ιστό (Web). Αρχικά, όμως, πολλοί επαγγελματίες προγραμματιστές υποτίμησαν τη γλώσσα διότι το κοινό της ήταν ερασιτέχνες συγγραφείς ιστοσελίδων και όχι επαγγελματίες προγραμματιστές (και μεταξύ άλλων λόγων). Με με τη χρήση της τεχνολογίας Ajax, η JavaScript γλώσσα επέστρεψε στο προσκήνιο και έφερε πιο επαγγελματική προσοχή προγραμματισμού. Το αποτέλεσμα ήταν ένα καινοτόμο αντίκτυπο στην εξάπλωση των πλαισίων και των βιβλιοθηκών, τη βελτίωση προγραμματισμού με JavaScript, καθώς και αυξημένη χρήση της JavaScript έξω από τα προγράμματα περιήγησης στο Web.

Τον Ιανουάριο του 2009, το έργο CommonJS ιδρύθηκε με στόχο τον καθορισμό ενός κοινού προτύπου βιβλιοθήκης κυρίως για την ανάπτυξη της JavaScript έξω από το πρόγραμμα περιήγησης και μέσα σε άλλες τεχνολογίες (π.χ. server-side).

2.7.2 Προγραμματισμός στην πλευρά του πελάτη

Αν και η HTML γνώρισε αμέσως μεγάλη δημοτικότητα, οι προγραμματιστές ιστοσελίδων αντιμετώπισαν αρκετά νωρίς συγκεκριμένους περιορισμούς. Για παράδειγμα, χρησιμοποιώντας μόνο την HTML, δεν υπάρχει κάποιος τρόπος που να επιτρέπει την εμφάνιση της τρέχουσας ημερομηνίας και ώρας σε κάθε σελίδα. Ουσιαστικά η HTML από μόνη της είναι κατάλληληκαι ιδιαίτερα επιτυχής για τη διανομή, κατόπιν αιτήσεως, σελίδων που έχουν ήδη προετοιμαστεί όσον αφορά το περιεχόμενο, και οι οποίες θεωρούνται στατικες.



Εικόνα 2.9 – Client Based Programming

Η ανάγκη παράκαμψης της στατικότητας του περιεχομένου της HTML, οδήγησε στην ανάπτυξη της φιλοσοφίας του προγραμματισμού από την πλευρά του πελάτη. Κατά τον προγραμματισμό από την πλευρά του πελάτη, η σελίδα HTML εμπλουτίζεται με κατάλληλο κώδικα γραμμένο σε γλώσσα προγραμματισμού, ο οποίος εκτελείται από τον Η/Υ του πελατη, επιτρέποντας τόσο το δυναμικό χειρισμό συμβάντων κατά τη διάρκεια της εμφάνισης της σελίδας, όσο και την αύξηση της διαδραστικότητας μεταξύ του χρήστη και της εφαρμογής.

Οι γλώσσες προγραμματισμού που χρησιμοποιούνται στον προγραμματισμό από την πλευρά του πελάτη, ονομάζονται γλώσσες συγγραφής σεναρίων (scripting languages) και δίνουν τη δυνατότητα επεξεργασίας των δεδομένων της ιστοσελίδας, καθώς και τη διεξαγωγή υπολογισμών και μετασχηματισμών σε αυτά τα δεδομένα, με τη βοήθεια εντολών, συναρτήσεων, αντικειμένων και συμβάντων. Σημαντικότερος εκπρόσωπος των γλωσσών συγγραφής σεναρίων είναι η γλώσσα Javascript.

2.7.3 Δομή της Javascript

Ο κώδικας της JavaScript γράφεται σε καθαρό κείμενο (ASCII μορφή) και ενσωματώνεται μέσα στον κώδικα της HTML, μπορεί δε να εκτελεσθεί αμέσως με το φόρτωμα της σελίδας ή όταν λαμβάνει χώρα ένα συμβάν (event), όπως η πίεση ενός πλήκτρου του ποντικιού ή η τοποθέτηση του ποντικιού πάνω σε ένα αντικείμενο.

Οι εφαρμογές της Javascript στο δυναμικό προγραμματισμό παγκόσμιου ιστού είναι πάρα πολλές. Ως χαρακτηριστικά παραδείγματα αναφέρουμε τον έλεγχο της εγκυρότητας των δεδομένων που πληκτρολογούνται σε μια φόρμα του πελάτη, πριν αυτά αποσταλούν στον εξυπηρετητή, τη διεξαγωγή διάφορων υπολογισμών και μετατροπών μεγεθών και την προσθήκη δυναμικών μενού στις ιστοσελίδες.

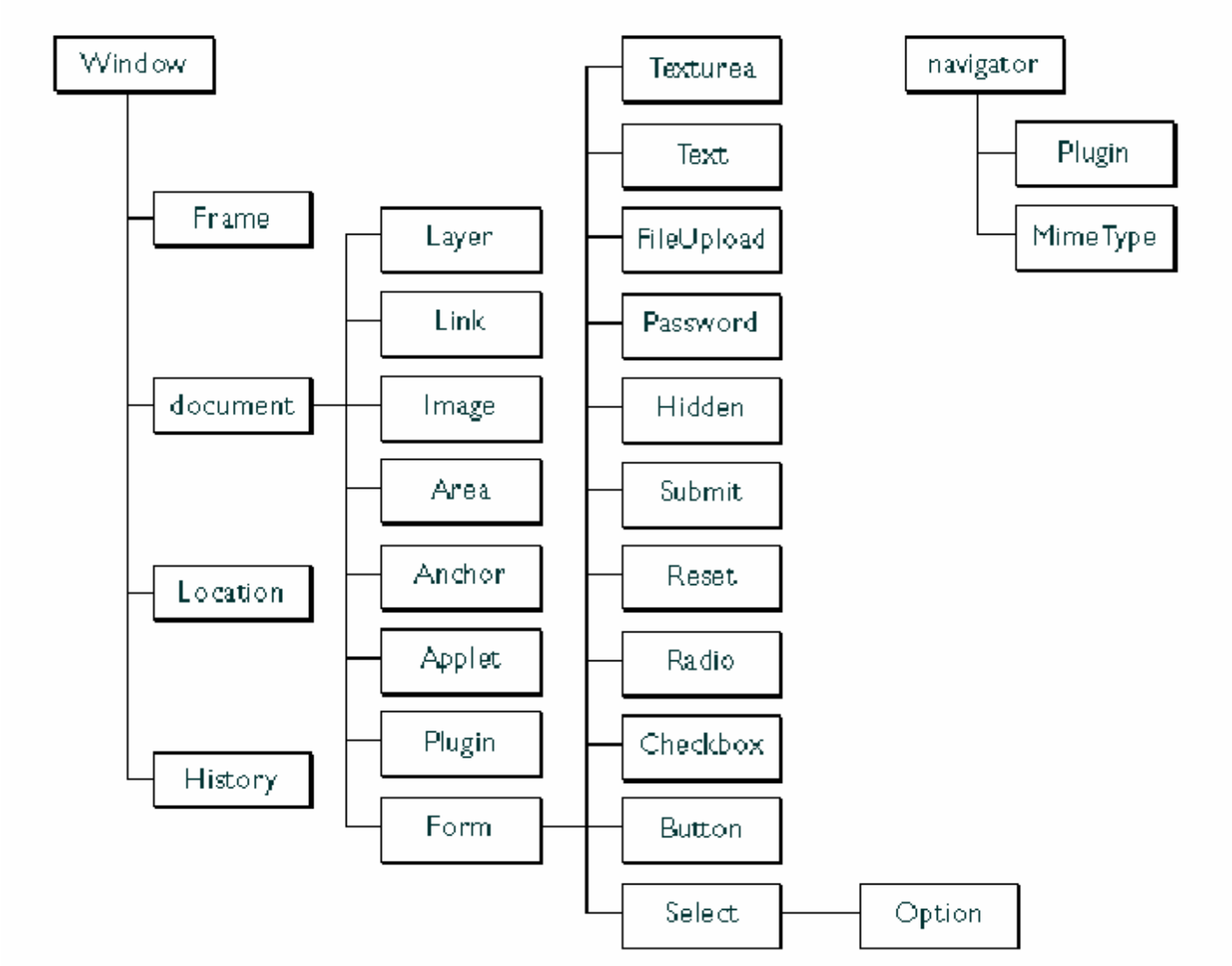
Το σενάριο Javascript ενσωματώνεται στην ιστοσελίδα, ανάμεσα στις ετικέτες <SCRIPT> και </SCRIPT>. Μέσα σε ένα αρχείο HTML μπορούν να υλοποιηθούν πολλά σενάρια Javascript, χρησιμοποιώντας πολλαπλές ετικέτες SCRIPT.

Τα σενάρια Javascript δεν είναι υποχρεωτικά ενσωματωμένα στη σελίδα HTML. Αντίθετα, μπορούν να γραφούν ως ξεχωριστά αρχεία με κατάληξη .js, τα οποία καλούνται και εκτελούνται από τη σελίδα HTML με τη βοήθεια

σχετικής αναφοράς. Σε αυτή την περίπτωση, η ετικέτα SCRIPT συνοδεύεται από την παράμετρο SRC, η οποία χρησιμοποιείται για τη δήλωση του εξωτερικού αρχείου στον κώδικα HTML.

2.7.4 Τα αντικείμενα στην Javascript

Η γλώσσα Javascript βασίζεται στην έννοια του αντικειμένου. Το αντικείμενο είναι μια οντότητα η οποία κατέχει ένα σύνολο από ιδιότητες, οι οποίες μπορούν να τροποποιηθούν με κατάλληλες μεθόδους. Ένα καλό παράδειγμα αντικειμένου είναι το αντικείμενο document (η επιφάνεια του φυλλομετρητή), το οποίο έχει αρκετές ιδιότητες, όπως το χρώμα παρασκηνίου, το χρώμα προσκηνίου, το χρώμα κειμένου και τον τίτλο. Οι περισσότερες ιδιότητες της Javascript σχετίζονται άμεσα ή έμμεσα με αλλαγές μιας ή περισσότερων ιδιοτήτων των αντικειμένων. Το σύνολο των προκαθορισμένων αντικειμένων του φυλλομετρητή είναι γνωστό με το όνομα "Document Object Model" ή DOM και η ιεραρχία τους απεικονίζεται παρακάτω:



Εικόνα 2.10 – Η ιεραρχία αντικειμένων στην Javascript

Η Javascript χρησιμοποιεί την εξής σύνταξη για να αναφερθεί σε κάποια ιδιότητα:

< όνομα αντικειμένου >.< ιδιότητα >

Η έννοια του αντικειμένου είναι στενά συνδεδεμένη και με την έννοια της μεθόδου. Οι μέθοδοι είναι συναρτήσεις ή διαδικασίες που χρησιμοποιούνται για να εκτελούν μια λειτουργία σ' ένα αντικείμενο. Η κλήση και εκτέλεση των μεθόδων των αντικειμένων γίνεται με τη δήλωση αντικείμενο.μέθοδος().

Τα βασικά αντικείμενα του DOM, και οι σχετικές ιδιότητες, μεθόδοι και χειριστήρια συμβάντων που συσχετίζονται με αυτά είναι τα παρακάτω:

- **window** - Δημιουργείται από τον φυλλομετρητή όταν φορτώνεται μια σελίδα. Αποτελεί το αντικείμενο ανωτέρου επιπέδου (top-level object) για τα αντικείμενα document, location και history. Επειδή είναι το προκαθορισμένο αντικείμενο, δεν χρειάζεται να αναφέρουμε το όνομα του παραθύρου (window) όταν αναφερόμαστε στα αντικείμενά του, στις ιδιότητές του ή στις μεθόδους του. Για παράδειγμα, οι δηλώσεις status ("Hallo!") και window.status("Hallo") επιφέρουν το ίδιο αποτέλεσμα, δηλαδή την εμφάνιση ενός μηνύματος στη γραμμή κατάστασης (status line).
- **document** - Είναι ένα αντικείμενο που δημιουργείται από τον φυλλομετρητή όταν φορτώνεται μια ιστοσελίδα, το οποίο περιέχει πληροφορίες για το τρέχον έγγραφο, όπως είναι ο τίτλος του (ιδιότητα title), το χρώμα φόντου (ιδιότητα bgcolour) και οι φόρμες που περιέχει. Αυτές οι ιδιότητες ορίζονται μέσα στις ετικέττες της HTML.
- **frame** - Είναι ένα παράθυρο του φυλλομετρητή στο οποίο εμφανίζεται ένα έγγραφο HTML. Η επιφάνεια του φυλλομετρητή μπορεί να περιέχει πολλά frames τα οποία μπορει να δείχνουν σε διαφορετικά URLs και μπορούν να αποτελούν στόχο (target) από άλλα πλαίσια, όπου όλα μαζί βρίσκονται στο ίδιο παράθυρο. Το κάθε πλαίσιο (frame) αποτελεί μια ιδιότητα του αντικειμένου window.
- **form** - Αποτελεί ιδιότητα του αντικειμένου document. Η κάθε φόρμα σ' ένα έγγραφο αποτελεί ένα ξεχωριστό αντικείμενο (object) στο οποίο μπορούμε να αναφερθούμε μέσω του αντικειμένου form. Το αντικείμενο form είναι ένας πίνακας (array) που δημιουργείται καθώς ορίζονται οι φόρμες (forms) μέσω των HTML tags.
- ***button*** - Είναι ένα αντικείμενο που αποτελεί στοιχείο μιας φόρμας και μπορεί να χρησιμοποιηθεί για να εκτελεσθεί μια ενέργεια (action). Αποτελεί ιδιότητα του αντικειμένουform. Σχετικές ιδιότητες είναι η *name* και η *value*, σχετική μέθοδος είναι η *click* και σχετικό χειριστήριο συμβάντος είναι το *onClick*.
- ***checkbox*** - Είναι ένα στοιχείο φόρμας το οποίο ο χρήστης έχει τη δυνατότητα να κάνει ενεργό ή όχι (on ή off) κάνοντας κλικ σε αυτό. Χρησιμοποιώντας το αντικείμενο *checkbox*, μπορούμε να δούμε αν το στοιχείο είναι επιλεγμένο (ιδιότητα *checked*) καθώς και να έχουμε πρόσβαση στο όνομα (ιδιότητα *name*) και την τιμή του (ιδιότητα *value*). Σχετική μέθοδος είναι η *click* και σχετικό χειριστήριο συμβάντος είναι το *onClick*.
- ***hidden*** - Είναι ένα αντικείμενο κειμένου (text object) που δεν εμφανίζεται (είναι κρυφό) σε μια HTML φόρμα. Τα κρυφά αντικείμενα (hidden objects) μπορούν να χρησιμοποιηθούν μαζί με τα cookies για να μεταβιβάζονται ζευγάρια ονόματος/τιμής (name/valuepairs) για την επικοινωνία μεταξύ πελάτη/εξυπηρετητή. Το αντικείμενο *hidden*, αποτελεί ιδιότητα του αντικειμένου *form* και σχετικές ιδιότητες είναι οι *cookie, defaultValue, name* και *value*.
- ***history*** - Αυτό το αντικείμενο περιέχει πληροφορίες σύνδεσης URL για τις ιστοσελίδες που έχουμε επισκεφθεί νωρίτερα. Αποτελεί ιδιότητα του αντικειμένου *document* και σχετική ιδιότητα είναι η *length*. Σχετικές μέθοδοι είναι οι *back, forward* και *go*.
- ***link*** - Το αντικείμενο αυτό παρέχει πληροφορίες για τους υπάρχοντες συνδέσμους υπερκειμένου. Μπορούμε επίσης να το χρησιμοποιήσουμε για να δημιουργήσουμε καινούργιους συνδέσμους. Αποτελεί ιδιότητα του

αντικειμένου document. Σχετικές ιδιότητες είναι οι *hash, host, hostname, href, length, pathname, port, protocol, search* και *target*. Σχετική μέθοδος είναι η *link* και σχετικά χειριστήρια συμβάντος είναι τα *onClick* και *onMouseOver*.

- ***location*** - Το αντικείμενο αυτό περιέχει πλήρεις πληροφορίες URL για το τρέχον έγγραφο, ενώ η κάθε ιδιότητά του αναφέρεται σε ένα διαφορετικό κομμάτι του URL. Αποτελεί ιδιότητα του αντικειμένου *document*. Σχετικές ιδιότητες είναι οι *hash, host, hostname, href, location, pathname, port, protocol, search* και *target*.
- **Math -** Το αντικείμενο αυτό περιέχει ιδιότητες για μαθηματικές σταθερές και μεθόδους που χρησιμοποιούνται σε συναρτήσεις. Για παράδειγμα, η δήλωση Math.PI αντιπροσωπεύει την τιμή του π σε μια εξίσωση. Σχετικές ιδιότητες είναι οι *E, LN10, LN2, PI, SQRT1_2, SQRT2* και σχετικές μέθοδοι είναι οι *abs, acos, asin, atan, ceil, cos, exp, floor, log, max, min, pow, random, round, sin, sqrt* και *tan.*
- ***navigator*** - Το αντικείμενο αυτό περιέχει πληροφορίες για την τρέχουσα έκδοση του φυλλομετρητή που χρησιμοποιεί ο πελάτης. Σχετικές ιδιότητες είναι οι *appName, appCodeName, appVersion* και *userAgent*.
- ***option*** - Είναι αντικείμενα που δημιουργούνται μέσα στις φόρμες της HTML και παριστάνουν πλήκτρα επιλογής (option buttons). Από ένα σύνολο πλήκτρων επιλογής που ανήκουν στην ίδια λίστα, ο χρήστης μπορεί να επιλέξει μόνο ένα. Σχετικές ιδιότητες είναι οι *checked, defaultChecked, index, length, name* και *value*, σχετική μέθοδος είναι η *click* και σχετικό χειριστήριο συμβάντος είναι το *onClick.*
- ***password*** - Είναι ειδικά πεδία κειμένου της HTML τα οποία ορίζονται μέσα σε μια φόρμα και στα οποία το κείμενο που καταχωρείται από τον χρήστη είναι κρυμμένο συνήθως με το χαρακτήρα "*". Σχετικές ιδιότητες είναι οι *defaultValue, name* και *value* και σχετικές μέθοδοι είναι οι *focus, blur* και *select*.
- ***reset*** - Είναι ένα πλήκτρο εντολής το οποίο ορίζεται μέσα σε μια φόρμα και επαναφέρει τα περιεχόμενα όλων των αντικειμένων της φόρμας στις προκαθορισμένες τους τιμές. Αποτελεί ιδιότητα του αντικειμένου form. Σχετικές ιδιότητες είναι οι *name* και *value*, σχετική μέθοδος είναι η *click* και σχετικό χειριστήριο συμβάντος είναι το *onClick*.
- ***select*** - Είναι μια λίστα επιλογής (selection list) ή λίστα κύλισης (scrolling list) σε μια HTML φόρμα. Μια λίστα επιλογής δίνει τη δυνατότητα στο χρήστη να επιλέξει ένα στοιχείο από μια λίστα, ενώ μια λίστα κύλισης δίνει τη δυνατότητα επιλογής ενός ή περισσοτέρων στοιχείων από μια λίστα. Αποτελεί ιδιότητα του αντικειμένου *form*. Σχετικές ιδιότητες είναι οι *length, name, options* και *selectedIndex*, σχετικές μέθοδοι είναι οι *blur* και *focus* και σχετικά χειριστήρια συμβάντος είναι τα *onBlur, onChange* και *onFocus*.
- ***submit*** - Είναι ένα ειδικό πλήκτρο εντολής που δημιουργείται μέσα σε μια φόρμα και που προκαλεί την υποβολή της στο πρόγραμμα που καθορίζεται από την ιδιότητα *action* της φόρμας. Αποτελεί ιδιότητα του αντικειμένου form. Σχετικές ιδιότητες είναι οι *name* και *value*, σχετική μέθοδος είναι η click και σχετικό χειριστήριο συμβάντος είναι το *onClick*. Τυπική εφαρμογή του πλήκτρου *submit* είναι η υποβολή τιμών μιας φόρμας σε ένα πρόγραμμα του εξυπηρετητή, όπως θα δούμε στο επόμενο κεφάλαιο.
- ***text*** - Είναι ένα πεδίο καταχώρισης κειμένου μίας γραμμής σε μια HTML φόρμα το οποίο δέχεται χαρακτήρες ή και αριθμούς. Αποτελεί ιδιότητα του αντικειμένου form. Σχετικές ιδιότητες είναι οι *defaultValue, name* και *value*, σχετικές μέθοδοι είναι οι *focus, blur* και *select* και σχετικά χειριστήρια συμβάντος είναι τα *onBlur, onChange, onFocus,* και *onSelect.*
- ***textarea*** - Είναι παρόμοιο με το πεδίο κειμένου, με τη διαφορά ότι μπορεί να περιέχει πολλές γραμμές. Αποτελεί ιδιότητα του αντικειμένου form. Σχετικές ιδιότητες είναι οι *defaultValue, name* και *value*, σχετικές μέθοδοι είναι οι *focus, blur* και *select* και σχετικά χειριστήρια συμβάντος είναι τα *onBlur, onChange, onFocus* και *onSelect*.

2.7.5 Τα χειριστήρια συμβάντος (Event Handlers) της Javascript

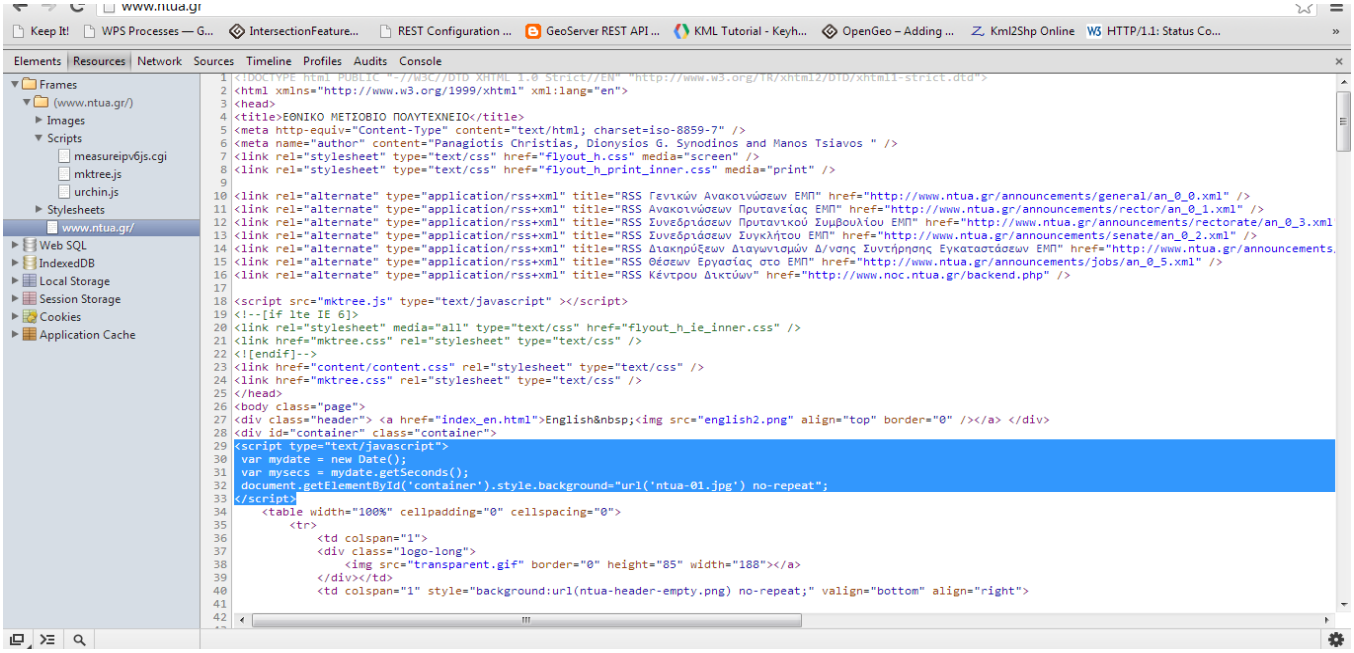
Τα χειριστήρια συμβάντος (event handlers) είναι παράμετροι που επιτρέπουν τη συσχέτιση του κώδικα Javascript με συγκεκριμένα γεγονότα που προκύπτουν από τις κινήσεις του φυλλομετρητή ή του χρήστη. Παρακολουθώντας τις ενέργειες του χρήστη, η JavaScript μπορεί να εκτελεί σενάρια στο Η/Υ του πελάτη, χωρίς να χρειασθεί να μεσολαβήσει ο εξυπηρετητής. Οι ενέργειες του χρήστη σε μια ιστοσελίδα οι οποίες καλύπτονται από τα χειριστήρια συμβάντων, σχετίζονται με την κίνηση του ποντικιού και τις καταχωρήσεις στα στοιχεία (πεδία) των φορμών. Τα κυριότερα χειριστήρια συμβάντων της Javascript συνοψίζονται στον ακόλουθο πίνακα:

Χειριστήριο Συμβάντος	**Κριτήριο ενεργοποίησης**
onBlur	Λαμβάνει χώρα όταν χάνεται η εστίαση (lost focus), δηλαδή όταν απομακρυνόμαστε είτε με κλικ με το ποντίκι είτε πατώντας το πλήκτρο tab σ' ένα πεδίο κειμένου ή σε μια περιοχή κειμένου ή και σ' ένα πλαίσιο λίστας (select) μιας φόρμας.
onChange	Λαμβάνει χώρα όταν τροποποιείται το περιεχόμενο ενός πεδίου κειμένου ή μιας περιοχής κειμένου ή και ενός πλαισίου λίστας μιας φόρμας, πριν ακόμα απομακρυνθούμε από το στοιχείο αυτό.
onClick	Λαμβάνει χώρα όταν κάνουμε κλικ με το ποντίκι σ' ένα αντικείμενο, όπως είναι για παράδειγμα ένα πλήκτρο εντολής (button) ή ένα πλαίσιο ελέγχου (check box)
onFocus	Λαμβάνει χώρα όταν κάνουμε εστίαση (focus) μέσα σ' ένα πεδίο μιας φόρμας.
onLoad	Λαμβάνει χώρα όταν τελειώσει το φόρτωμα μιας ιστοσελίδας σ' ένα παράθυρο ή όταν τελειώσει το φόρτωμα όλων των πλαισίων (frames) που βρίσκονται μέσα σε μια σελίδα.
onMouseOver	Λαμβάνει χώρα όταν ο δείκτης του ποντικιού τοποθετείται πάνω από ένα αντικείμενο ή από ένα σύνδεσμο.
onSelect	Λαμβάνει χώρα όταν επιλέγουμε ένα μέρος ή όλο το κείμενο ενός πεδίου κειμένου (text field) ή μιας περιοχής κειμένου (textarea).
onSubmit	Λαμβάνει χώρα όταν κάνουμε κλικ με το ποντίκι σ' ένα πλήκτρο υποβολής (submit button) για να υποβάλλουμε μια φόρμα.
onUnload	Λαμβάνει χώρα όταν φεύγουμε από ένα έγγραφο (ιστοσελίδα).

Πίνακας 2.4 – Javascript Event Handlers

2.7.6 Παράδειγμα κώδικα Javascript

Ακολουθέι παράδειγμα κώδικα Javascript ενσωματωμένου σε HTML από την ιστοσελίδα του Ε.Μ.Π. :



Εικόνα 2.11 – Παράδειγμα Javascript μέσα σε HTML

2.7.7 OpenLayers

Το OpenLayers είναι μια βιβλιοθήκη Javascript που κάνει δυνατή την οπτικοποίηση δεδομένων στον ιστό. Το OpenLayers διευκολύνει τους προγραμματιστές διαδικτυακών εφαρμογών να ενσωματώσουν στις σελίδες τους δυναμικούς χάρτες, από μια πληθώρα πηγών δεδομένων. Το OpenLayers παρέχει ένα επεκτάσιμο σύνολο χαρτογραφικών εργαλείων και παραθύρων, παρόμοια με το Google Maps API. Όλη η λειτουργικότητα εκτελείται στον περιηγητή, κάτι το οποίο καθιστά το OpenLayers πολύ εύκολο στην εγκατάσταση, χωρίς καθόλου εξάρτηση από τον εξυπηρετητή.

Βασικά Χαρακτηριστικά του OpenLayers:

- Εύκολο στη χρήση μέσω προγραμματιστικών διεπαφών Javascript, σχεδιασμένο να κάνει τον προγραμματισμό εύκολο
- Υποστήριξη διαδεδομένων και καθιερωμένων προτοκόλλων για επικοινωνία με εξυπηρετητές
- Εργαλεία για δημιουργία γραφικών διεπαφών με ευκολία
- Υποστήριξη εμφάνισης δεδομένων στον περιηγητή (με τη χρήση προτύπων SVG, VML, ή τεχνολογιών Canvas), υποστηρίζοντας την ανάπτυξη σύνθετων χαρτών μέσα στον περιηγητή
- Υποστήριξη για φορητές συσκευές (με έμφαση στις συσκευές αφής)
- Δυνατότητα φόρτωσης επιπέδων χαρτών από οποιαδήποτε πηγή (Εμπορικά επίπεδα: Google, Bing, Yahoo)
- Πρότυπα OGC: WMS, WMTS, WFS, WFS-T, GeoRS, GML
- Άλλα: OpenStreetMap (OSM), ArcGI, Images, MapGuide, MapServer, TileCache
- Δυνατότητα να προσπελάσει διανυσματικά δεδομένα και μεταδεδομένα σε οποιοδήποτε πρότυπο:•Atom, ArcXML, GeoJSON, GeoRSS, KML, OSM, SLD, WMTS

2.8 PHP

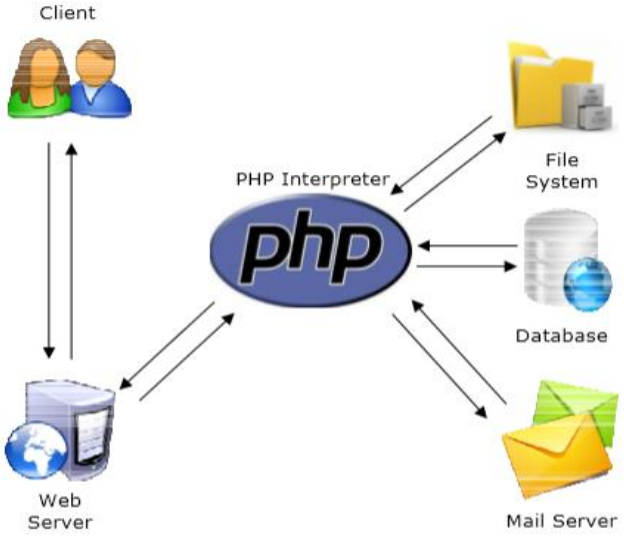
Ο όρος PHP αποτελεί ακρωνύμιο των λέξεων Hypertext Preprocessor δηλαδή προ-επεξεργαστής υπερκειμένου. Πρόκειται για μια scripting γλώσσα υψηλού επιπέδου, που σχεδιάστηκε για τη δημιουργία δυναμικών ιστοσελίδων. Αν και έχει εξελιχθεί σε μια γλώσσα γενικής χρήσης, η κύρια δύναμη της παραμένει στο χώρο του διαδικτυακού προγραμματισμού. Η μεγάλη διαφορά της με πανομοιότυπες γλώσσες προγραμματισμού (π.χ. C, Perl, Python) είναι ότι ο κώδικας εκτελείται στον διακομιστή, παράγοντας και κατά συνέπεια συνδυάζοντας HTML το οποίο αποστέλλεται στη συνέχεια στον πελάτη. Ο πελάτης θα λάβει τα αποτελέσματα από την εφαρμογή αυτών των ενεργειών, αλλά δε θα ξέρει τον κώδικα που κρύβεται από πίσω.

Η εκτέλεση του προγράμματος στον εξυπηρετητή γίνεται με τη βοήθεια του μεταγλωτιστή PHP (PHP parser), ο οποίος εγκαθίσταται στον εξυπηρετητή προσδίδοντάς του τη δυνατότητα εκτέλεσης δυναμικών εφαρμογών PHP. Ο μεταγλωτιστής PHP είναι διαθέσιμος δωρεάν στη διεύθυνση www.php.net, σε διάφορους τύπους, ανάλογα με τον τύπο του εξυπηρετητή και το λειτουργικό σύστημα του υπολογιστή στον οποίο θα γίνει η εγκατάσταση.

Η PHP χαρακτηρίζεται από απλότητα, η οποία παρέχει τις προυποθέσεις γρήγορης εκμάθησης, σε σχέση με άλλες γλώσσες προγραμματισμού από την πλευρά του εξυπηρετητή. Παρόλη την απλότητά της, η PHP προσφέρει προηγμένα χαρακτηριστικά τα οποία συχνά την καθιστούν μία από τις πρώτες επιλογές για επαγγελματικές εφαρμογές. Ένα άλλο σημαντικο χαρακτηριστικό της PHP είναι η υποστήριξη που παρέχει για ένα μεγάλο εύρος βάσεων δεδομένων, στις οποίες περιλαμβάνονται η MySQL, η Oracle, η Informix, η Sybase και πολλές άλλες.

2.8.1 Η ιστορία της PHP

Η ιστορία της PHP ξεκινά από το 1995, όταν ένας φοιτητής, ο Rasmus Lerdorf δημιούργησε χρησιμοποιώντας τη γλώσσα προγραμματισμού Perl ένα απλό script με όνομα php.cgi, για προσωπική χρήση. Το script αυτό είχε σαν σκοπό να διατηρεί μια λίστα στατιστικών για τα άτομα που έβλεπαν το online βιογραφικό του σημείωμα. Αργότερα αυτό το script το διέθεσε και σε φίλους του, οι οποίοι άρχισαν να του ζητούν να προσθέσει περισσότερες δυνατότητες. Η γλώσσα τότε ονομαζόταν PHP/FI από τα αρχικά Personal Home Page/Form Interpreter. Το 1997 η PHP/FI έφθασε στην έκδοση 2.0, βασιζόμενη αυτή τη φορά στη γλώσσα C και αριθμώντας περισσότερα από 50.000 web sites που τη χρησιμοποιούσαν, ενώ αργότερα την ίδια χρονιά οι Andi Gutmans και Zeev Suraski ξαναέγραψαν τη γλώσσα από την αρχή, βασιζόμενοι όμως αρκετά στην PHP/FI 2.0. Έτσι η PHP έφθασε στην έκδοση 3.0 η οποία θύμιζε περισσότερο τη σημερινή μορφή της. Στη συνέχεια, οι Zeev και Andi δημιούργησαν την εταιρεία Zend (από τα αρχικά των ονομάτων τους), η οποία συνεχίζει μέχρι και σήμερα την ανάπτυξη και εξέλιξη της γλώσσας PHP. Ακολούθησε το 1998 η έκδοση 4 της PHP, τον Ιούλιο του 2004 διατέθηκε η έκδοση 5, ενώ αυτή τη στιγμή έχουν ήδη διατεθεί και τα πρώτα snapshots της επερχόμενης PHP 6, για οποιονδήποτε



προγραμματιστή θέλει να τη χρησιμοποιήσει. Σήμερα περισσότερα από 16.000.000 web sites, ποσοστό μεγαλύτερο από το 35% των ιστοσελίδων του Διαδικτύου, χρησιμοποιούν scripts γραμμένα με τη γλώσσα PHP, ενώ το υπόλοιπο 65% το μοιράζονται στατικές σελίδες HTML και όλες οι άλλες γλώσσες προγραμματισμού.

2.8.2 Δομή της PHP

Ο κώδικας PHP γράφεται μέσα στον κώδικα HTML και περικλείεται από τις ετικέτες <?php και ?>. Για να γίνει η επεξεργασία μιας ιστοσελίδας που περιέχει PHP από τον εξυπηρετητή, η σελίδα θα πρέπει να αποθηκευθεί ως αρχείο με κατάληξη .php. Εάν η σελίδα έχει κατάληξη .htm, τότε ο εξυπηρετητής απλά στέλνει τις HTML δηλώσεις στον πελάτη, ο οποίος τις επεξεργάζεται και εμφανίζει το αποτέλεσμα της επεξεργασίας στο φυλλομετρητή. Αν όμως η ιστοσελίδα έχει την κατάληξη .php, τότε αυτή ανοί Εικόνα 2.12 – Δομή της PHP εξυπηρετητή, ο οποίος αναζητά στη σελίδα τμήματα κώδικα PHP. Μόλις βρεθούν τέτοια τμήματα, η εφαρμογή του εξυπηρετητή προχωρά στις απαιτούμενες επεξεργασίες και μετατρέπει το αποτέλεσμά τους σε μορφή HTML, προωθώντας το στον πελάτη.

Η σύνταξη της PHP είναι πολύ κοντά στις εξής γλώσσες: C, C++, Java, JavaScript και Perl. Ένα σενάριο της PHP αποτελείται από μια σειρά εντολών (commands ή statements), η καθε μία από τις οποίες είναι μια οδηγία (instruction) που πρέπει να ακολουθήσει ο Web server πριν προχωρήσει στην επόμενη. Οι εντολές της PHP τερματίζονται πάντα με τον χαρακτήρα semicolon (;). Όπως σε κάθε γλώσσα προγραμματισμού, έτσι και στην PHP βασικά δομικά στοιχεία αποτελούν οι μεταβλητές (Variables) οι τελεστές (operators) και οι συναρτήσεις (functions).

2.8.3 Διερμήνευση και Μεταγλώτιση

Η PHP χρησιμοποιεί μια μίξη από διερμήνευση (interpretation) και μεταγλώττιση (compilation) έτσι ώστε να μπορέσει να δώσει στους προγραμματιστές τον καλύτερο δυνατό συνδυασμό απόδοσης και ευελιξίας . Στο παρασκήνιο , η PHP μεταγλωττίζει το script σε μια σειρά από εντολές (instructions), που είναι γνωστές με τον όρο opcodes, οι οποίες εντολές εκτελούνται μία -μία μέχρι να τελειώσει το script.

Αυτό είναι κάτι διαφορετικό από τις παραδοσιακές γλώσσες που μεταγλωττίζονται , όπως είναι η C++, όπου ο κώδικας μεταγλωττίζεται σε εκτελέσιμο κώδικα μηχανής , ενώ η PHP μεταγλωττίζει εκ νέου το script κάθε φορά που αυτό απαιτείται . Αυτή η συνεχής μεταγλώττιση μπορεί να φαίνεται ως απώλεια χρόνου , αλλά δεν είναι καθόλου κακή καθώς δεν χρειάζεται να κάνουμε συνέχεια εμείς τη μεταγλώττιση των scripts όταν γίνονται κάποιες αλλαγές σ' αυτά .

2.8.4 Δυνατότητες της PHP

Η PHP κυρίως επικεντρώνεται στον προγραμματισμό και τις ενέργειες του διακομιστή (server-side scripting), επομένως μπορεί ο χρήστης να κάνει οτιδήποτε , μπορεί να κάνει οποιοδήποτε άλλο CGI πρόγραμμα . Αλλά η PHP μπορεί να κάνει πολλά περισσότερα .

Υπάρχουν τρεις βασικές περιπτώσεις στις οποίες χρησιμοποιούνται PHP σενάρια :

- **Server-side scripting**
  Είναι το πιο συνηθισμένο . Χρειάζονται τρία πράγματα για να δουλέψει αυτό, ο PHP parser, ένας web server κι ένας web browser. Ο χρήστης πρέπει να " τρέξει " το διακομιστή ιστοσελίδων. Ο χρήστης μπορεί να έχει πρόσβαση στο αποτέλεσμα του προγράμματος PHP μέσω ενός προγράμματος περιήγησης. Όλα αυτά μπορούν να εκτελούνται στο σταθερό υπολογιστή καθώς ο ενδιαφερόμενος θα πειραματίζεται με τον προγραμματισμό σε PHP.
- **Command line scripting**
  Μπορεί να τρέξει ένα PHP πρόγραμμα χωρίς κάποιον διακομιστή ή πρόγραμμα περιήγησης. Χρειάζεται μόνο ένα πρόγραμμα ανάλυσης PHP (PHP parser).
- **Writing desktop applications**
  Η PHP δεν είναι η καλύτερη γλώσσα για τη δημιουργία εφαρμογών επιφάνειας εργασίας με ένα γραφικό περιβάλλον χρήστη, αλλά ο χρήστης που γνωρίζει καλά την PHP μπορεί να χρησιμοποιήσει κάποια προηγμένα PHP χαρακτηριστικά στις εφαρμογές του καθώς επίσης και PHP-GTK για να γράψει κάποια προγράμματα . Η PHP-GTK είναι μία επέκταση της PHP, μη διαθέσιμη όμως στην κεντρική διανομή .

Η PHP μπορεί να χρησιμοποιηθεί σε όλα τα βασικά λειτουργικά συστήματα , όπως Linux, Microsoft Windows, Mac OS X, RISC OS. Η PHP υποστηρίζει επίσης τους περισσότερους διακομιστές δικτύου (web servers), μεταξύ των οποίων Apache, IIS, και πολλοί άλλοι . Η PHP λειτουργεί είτε ως μία λειτουργική μονάδα είτε ως CGI επεξεργαστής .

Επομένως με την PHP, ο προγραμματιστής έχει την ελευθερία να διαλέξει ένα λειτουργικό σύστημα κι έναν διακομιστή δικτύου . Επιπλέον , έχει την ευκαιρία να χρησιμοποιήσει δομημένο προγραμματισμό ή αντικειμενοστραφή προγραμματισμό , ή ένα μίγμα και των δύο .

Με την PHP ο χρήστης δεν περιορίζεται στην εξαγωγή HTML. Οι δυνατότητες της PHP περιλαμβάνουν την εξαγωγή εικόνων, PDF αρχείων , ακόμη και ταινίες Flash. Μπορεί επίσης να εξάγει εύκολα ένα κείμενο , όπως για παράδειγμα XHTML και οποιοδήποτε άλλο XML αρχείο. Η PHP μπορεί να παράγει μόνη της αυτά τα αρχεία και να τα αποθηκεύσει στο σύστημα αρχείων (file system), αντί να τα εκτυπώσει, δημιουργώντας μία κρυφή μνήμη για το δυναμικό περιεχόμενο.

Ένα από τα πιο δυνατά και σημαντικά χαρακτηριστικά της PHP είναι ότι υποστηρίζει ένα ευρύ φάσμα βάσεων δεδομένων. Η δημιουργία μίας ιστοσελίδας στηριζόμενη σε βάσεις δεδομένων είναι πολύ απλή με τη χρήση συγκεκριμένων επεκτάσεων των βάσεων δεδομένων (π.χ. mysql), ή με τη σύνδεση σε οποιαδήποτε βάση δεδομένων υποστηρίζοντας την Open Database Connection μέσω της ODBC επέκτασης .

Η PHP υποστηρίζει επίσης την επικοινωνία με άλλες υπηρεσίες χρησιμοποιώντας πρωτόκολλα όπως LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (στα Windows) και πολλά άλλα. Όσον αφορά τη διασύνδεση , η PHP υποστηρίζει επίσης τη δημιουργία αντικειμένων και τη χρήση αυτών ως PHP αντικείμενα.

Η PHP έχει χρήσιμες δυνατότητες επεξεργασίας κειμένου , οι οποίες περιλαμβάνουν επεκτάσεις και εργαλεία για την ανάλυση και την πρόσβαση του χρήστη σε XML έγγραφα . Η PHP τυποποιεί όλες τις XML επεκτάσεις με σταθερή βάση την libxml2, και επεκτείνει τη δυνατότητα του χρήστη να ορίσει την προσθήκη SimpleXML, XMLReader και XMLWriter υποστήριξης .

2.8.5 Παράδειγμα PHP κώδικα

```
Writing text to a file
<?php
$handle = fopen ("myfile.txt" , 'w+' );
if ($handle ) {
    if (! fwrite ($handle , "Student Name: Mark Fendisen" )) die ("couldn't w
        echo "success writing to file" ;}
?>

Create new file for writing
<?php
$fh = fopen ("myfile.txt" , "w" );
if ($fh == false )  die ("unable to create file" );
?>
```

```
Creating and Calling a PHP Function
<html>
    <body>
        <?php
        //we create a function name my_function
        function my_function ()
            {echo "Hello! How are you?" ;}
        //we call our function like this when we want to use it
        my_function ();
        ?>
    </body>
</html>
```

Εικόνα 2.13 – Παραδείγματα PHP κώδικα

2.9 GeoServer

Ο GeoServer είναι ένας διαδικτυακός εξυπηρετητής που επιτρέπει την δημοσίευση χαρτών και χωρικών δεδομένων από πληθώρα προτύπων σε λογισμικά πελάτες όπως οι περιηγητές διαδικτύου και τα λογισμικά GIS. Αυτό σημαίνει ότι ο χρήστης μπορεί να αποθηκεύσει τα δεδομένα του σε οποιοδήποτε πρότυπο επιθυμεί και οι υπόλοιποι χρήστες δεν χρειάζεται να γνωρίζουν οτιδήποτε σχετικό με τα δεδομένα αυτά. Στην απλούστερη περίπτωση το μόνο που χρειάζονται είναι ένας περιηγητής (web browser) για να μπορούν να βλέπουν τους χάρτες όπως ακριβώς έχουν δημοσιοποιηθεί.

Ο GeoServer είναι η υλοποίηση αναφοράς του προτύπου Open Geospatial Consortium (OGC) Web Feature Service (WFS) και του Web Coverage Service (WCS), καθώς επίσης και ένας πιστοποιημένος εξυπηρετητής υψηλής απόδοσης του προτύπου Web Map Service (WMS). Ο GeoServer αποτελεί ένα κεντρικό πυρήνα του Γεωχωρικού Διαδικτύου (Geospatial Web).

2.9.1 Τα WFS, WFS-T, WMS και WCS πρωτόκολα

Ο GeoServer έχει τη δυνατότητα να σερβίρει δεδομένα από πολλές πηγές δεδομένων:

- **Διανυσματικά:** Shapefiles, Εξωτερικά WFS, PostGIS, ArcSDE, DB2, Oracle Spatial, MySql, SQL Server
- **Ψηφιδωτά:** GeoTiff, JPG και PNG (με το συνοδευτικό world file), πυραμίδες εικόνων, πρότυπα της βιβλιοθήκης GDAL, μωσαικά εικόνων, Oracle GeoRaster

Τα δεδομένα σερβίρονται με το ασφαλές και γρήγορο πρωτόκολλο WMS Η παρουσίαση κάθε θεματικού επιπέδου του χάρτη ελέγχεται από το πρότυπο SLD που επιτρέπει στα χαρακτηριστικά του χάρτη να έχουν χρώμα και σύμβολα. Συνδυάζοντας τους κανόνες αυτούς με τα Φίλτρα του OGC, μπορούν να παραχθούν χάρτες που η παρουσίαση τους εξαρτάται από την κλίμακα θέασης, που επιτρέπει την προσθήκη λεπτομέρειας όσο ο χρήστης

μεγενθύνει το χάρτη. Διαχείριση συγκρούσεων συμβόλων, ομαδοποίηση και προτεραιότητες έχουν επίσης υλοποιηθεί στο λογισμικό αυτό.

Τα πλήρη διανυσματικά δεδομένα μπορούν να σταλούν στους χρήστες με τη χρήση του πρωτόκολλου WFS. Ένας πελάτης WFS μπορεί να κατεβάσει τα διανυσματικά δεδομένα και να τα χρησιμοποιήσει για χαρτοσύνθεση, χωρική ανάλυση και άλλες λειτουργίες. Επίσης, εφόσον ο χρήστης είναι πιστοποιημένος μπορεί να αλλάξει τα δεδομένα και να τα στείλει πίσω στον εξυπηρετητή για αποθήκευση μέσω του πρωτόκολλου WFS-T. Τα δεδομένα μπορούν να σταλούν μέσω του προτύπου GML (συμπιεσμένα) καθώς επίσης και σαν άλλα δημοφιλή πρότυπα όπως τα shapefile και json.

Οι Τιμές των εικονιστικών δεδομένων μπορούν να σταλούν στον πελάτη με τη χρήση του πρωτόκολλου WCS:Ένα λογισμικό GIS μπορεί να ζητήσει από τον GeoServer τα πραγματικά εικονιστικά δεδομένα για τη χρήση σε χωρική ανάλυση. Αυτό επιτρέπει στον χρήστη να δημιουργήσει εφαρμογές που να μοντελοποιούν διαδικασίες που περιγράφουν τα δεδομένα.

Άμεση αλλαγή συστηματων αναφοράς: Ο GeoServer υποστηρίζει τα περισσότερα συστήματα αναφοράς από τη βάση δεδομένων EPSG και επιτρέπει την αλλαγή προβολικού συστήματος ανάλογα με τη ζήτηση από τον πελάτη, επιτρέποντας έτσι σε χρήστες που δεν διαθέτουν λογισμικό αλλαγής χαρτογραφικών προβολών να εκτελούν τη διαδικασία απομακρυσμένα στον εξυπηρετητή.

Ο GeoWebCache είναι ένας πελάτης αποθήκευσης κανονικοποιημένων εικόνων από WMS. Το λογισμικό αυτό παρεμβάλλεται στην επικοινωνία του εξυπηρετητή με τον πελάτη, αποθηκεύει τα τμήματα εικόνων - χαρτών που έχουν ζητηθεί από τους πελάτες, και αναλαμβάνει να σερβίρει εκείνα τα τμήματα που έχουν ήδη δημιουργηθεί, εξοικονομώντας έτσι χρόνο και επεξεργαστική ισχύ για τον εξυπηρετητή. Ο GeoWebCache έχει ήδη ενσωματωθεί μέσα στον GeoServer.

2.9.2 Η υπηρεσία WPS

Η υπηρεσία WPS αν και δεν είναι εξ' αρχής ενσωματωμένη στο GeoServer παρέχεται δωρεάν και απλά απαιτεί «κατέβασμα» μέσα στη βιβλιοθήκη του GeoServer. Η WPS είναι αρκτικόλεξο του αγγλικού Web Processing Service. Χαλαρή μετάφραση αυτού είναι «υπηρεσία επεξεργασίας δεδομένων από τον ιστό». Η WPS είναι μια μέθοδος δημοσίευσης επεξεργασίας χωρικών δεδομένων χωρίς όμως η ίδια η υπηρεσία να καθορίζει ποιες είναι αυτές οι επεξεργασίες. Ο εκάστοτε εξυπηρετητής έχει έτσι τη δυνατότητα να ορίζει τόσο το ποιες υπηρεσίες ενσωματώνει αλλά και πως αυτές τρέχουν.

Ο GeoServer ενσωματώνει υπηρεσίες 2 κατηγοριών:

1. Συναρτήσεις απή την JTS Topology Suite
2. Συναρτήσεις ειδικά φτιαγμένες για τον GeoServer

Η JTS Topology Suite είναι μια Java βιβλιοθήκη συναρτήσεων για την επεξεργασία χωρικών δεδομένων σε δύο διαστάσεις. Η JTS συμορφώνεται με τα πρότυπα του Open Geospatial Consortium (OGC) και περιέχει μια σειρά συναρτήσεων επεξεργασίας χωρικών δεδομένων όπως οι: area (εμβαδό), buffer, intersection (τομή), and simplify (γενίκευση).

Οι συναρτήσεις που είναι ειδικά φριαγμένες για τον GeoServer έχουν να κάνουν κυρίως με τα όρια και την προβολή από ένα σύστημα αναφοράς σε άλλο. Γι' αυτό τον λόγο χρησιμοποιούν μια εσωτερική σύνδεση με τα

WFS/WCS πρωτόκολα χωρίς αυτά να αποτελούν κομμάτι της WPS για την ανάγνωση και την εγγραφή δεδομένων.

Βασικό όφελος από τη WPS είναι η εγγενής δυνατότητα να εκτελεί αλυσιδωτά τις διάφορες επεξεργασίες. Όπως στις γλώσσες προγραμματισμού η μία συνάρτηση μπορεί να καλεί μια άλλη έτσι και στη WPS μπορεί η μια επεξεργασία να καλεί την άλλη και να χρησιμοποιεί έτσι ως δεδομένα αποτελέσματα τα οποία δεν υπάρχουν ως τέτοια στον εξυπηρετητή.

Τα αιτήματα προς την υπηρεσία WPS γίνονται με XML τα οποία περιγράφουν ταυτόχρονα και την επεξεργασία που ζητείται αλλά και τα δεδομένα τα οποία απαιτούνται από τον server.

Τα αποτελέσματα που επιστρέφονται μπορούν να είναι αλφαριθμητικά, διανυσματικά ή ψηφιδωτά δεδομένα.

Ακολουθεί παράδειγμα ενός αιτήματος WPS που υπολογίζει το εμβαδό του πολυγώνου P1:

new 3.xml

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service=
"WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="
http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="
http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="
http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink
" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>JTS:area</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>geom</ows:Identifier>
      <wps:Reference mimeType="text/xml; subtype=gml/3.1.1" xlink:href="
      http://geoserver/wps" method="POST">
        <wps:Body>
          <wps:Execute version="1.0.0" service="WPS">
            <ows:Identifier>gs:CollectGeometries</ows:Identifier>
            <wps:DataInputs>
              <wps:Input>
                <ows:Identifier>features</ows:Identifier>
                <wps:Reference mimeType="text/xml" xlink:href="
                http://geoserver/wfs" method="POST">
                  <wps:Body>
```

new 5.xml

```
                    <wfs:GetFeature service="WFS" version="1.0.0"
                    outputFormat="GML2" xmlns:diplwmatiki="
                    http://localhost/">
                      <wfs:Query typeName="diplwmatiki:P1_new"/>
                    </wfs:GetFeature>
                  </wps:Body>
                </wps:Reference>
              </wps:Input>
            </wps:DataInputs>
            <wps:ResponseForm>
              <wps:RawDataOutput mimeType="text/xml; subtype=gml/3.1.1">
                <ows:Identifier>result</ows:Identifier>
              </wps:RawDataOutput>
            </wps:ResponseForm>
          </wps:Execute>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput>
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>
```

Εικόνα 2.14 – Παράδειγμα WPS αιτήματος

ΚΕΦΑΛΑΙΟ 3

Σχεδίαση & Υλοποίηση διαδικτυακού λογισμικού πολυπαραμετρικής αξιολόγησης διαδρομών

Αποτέλεσμα των εργασιών που περιγράφονται σε αυτό το κεφάλαιο είναι η δημιουργία μιας διαδυκτιακής εφαρμογής η οποία επιτρέπει στον χρήστη να προβαίνει στην πολυπαραμετρική αξιολόγηση διαδρομών. Η εφαρμογή αναπτύσεται πλήρως στον εξυπηρετητή πράγμα που σημαίνει ότι δεν απαιτείται κανένα ειδικό λογισμικό από τη μεριά του χρήστη πέρα από το φυλλομετρητή Google Chrome. Ο χρήστης δεν απαιτείται να έχει καμμιά ειδικευμένη γνώση για το σύνολο των εργασιών που η εφαρμογή εκτελεί. Ο χρήστης απλά σχεδιάζει τη διαδρομή που τον ενδιαφέρει με τα εργαλεία σχεδίασης πάνω στο χάρτη, την αποθηκεύει, επιλέγει την μέγιστη απόσταση από τους σταθμούς του Μετρό και τέλος το θεματικό επίπεδο ως προς το οποίο θέλει να εξετάσει τη διαδρομή και η εφαρμογή επιστρέφει τα αποτελέσματα.

Επιπλέον παρέχονται μια σειρά επιλογών στο χρήστη

- Η δυνατότητα να επιλέξει το είδος του χάρτη πάνω στον οποίο γίνεται η σχεδίαση (μεταξύ των OpenStreetMap, Google Maps, Google Satelite και της Κτηματολόγιο Α.Ε.)
- Η δυνατότητα για εισαγωγή/εξαγωγή στοιχείων σε διάφορα format (GeoJSON, Atom, KML, GeoRSS, GML Version 2&3, WKT και GPX)
- Η δυνατότητα να επιλέξει μεταξύ δύο προβολικών συστημάτων για τα στοιχεία εισόδου/εξόδου

Τα χαρακτηριστικά της εφαρμογής που αναπτύχθηκε είναι ότι:

- Δεν απαιτεί κανένα ειδικευμένο λογισμικό από τον χρήστη
- Δεν απαιτεί καμμιά ειδικευμένη γνώση από τον χρήστη
- Τα προγράμματα και ο κώδικας που χρησιμοποιεί είναι δωρεάν
- Το μεγαλύτερο μέρος των προγραμμάτων και του κώδικα είναι ανοιχτού λογισμικού
- Είναι επεκτάσιμο και μπορεί να εφαρμοστεί πάνω σε οποιοδήποτε WEB GIS εργασία

3.1 Πορεία της εργασίας

Το σύνολο εργασιών έγινε σε λειτουργικό σύστημα Windows 7 και εκτελέστηκε τοπικά. Για μια online εφαρμογή προτιμότερο θα ήταν για λόγους ασφάλειας, ταχύτητας και σταθερότητας το στήσιμο της εφαρμογής σε περιβάλλον Linux.

3.1.1 APACHE/PHP

Πρώτο βήμα ήταν το στήσιμο του τοπικού εξυπηρετητή και η εγκατάσταση της PHP.Χρησιμοποιήθηκε το πακέτο XAMPP 1.7.7 – VC9 που έχει τον APACHE με προεγκατεστημένη την PHP και είναι προρυθμισμένος για τη λειτουργία σε περιβάλλον Windows. Η εγκατάσταση του APACHE έγινε στη διαδρομή C:\xampp. Ο Apache πρέπει να εγκατασταθεί στην θύρα :80, αλλιώς θα πρέπει να γίνουν αντίστοιχες τροποποιήσεις στο σύνολο του συστήματος και του κώδικα. Δημιουργήσαμε επίσης το φάκελο C:\xampp\htdocs\diplwmatiki\... στον οποίο μέσα τοποθετούμε τα πράγματα που ο server μας θα ανεβάσει στο δίκτυο και απαιτούνται για τη λειτουργία του προγράμματος.

Χρειάστηκε να γίνουν κάποιες επιπλέον ρυθμίσεις:

1. Η ρύθμιση της PHP ώστε να επιτρέπει την curl συνάρτηση
   Στο αρχείο \...\xampp\php\php.ini ενεργοποιώ το extension=php_curl.dll
2. Η ρύθμιση του APACHE ώστε να «βλέπει» ως proxy τον Geoserver
   Στο αρχείο \...\ xampp\apache\conf\httpd.conf προσθέτω το παρακάτω κομμάτι κώδικα:

```
<VirtualHost  *:80>
        ProxyRequests Off
        ProxyPreserveHost On

        <Proxy *>
                Order deny,allow
                Allow from all
        </Proxy>

        ProxyPass /geoserver http://localhost:8080/geoserver
        ProxyPassReverse /geoserver http://localhost:8080/geoserver
</VirtualHost>
```

Πίνακας 3.1 – Ρύθμιση Geoserver ως proxy βήμα 1

Στο φάκελο \...\ xampp\cgi-bin\ δημιουργώ ένα αρχείο κειμένου με το όνομα proxy.cgi στο οποίο γράφω το παρακάτω κομμάτι κώδικα:

```
#!/c/Python33/env python

        """This is a blind proxy that we use to get around browser
        restrictions that prevent the Javascript from loading pages not on the
        same server as the Javascript.  This has several problems: it's less
        efficient, it might break some sites, and it's a security risk because
        people can use this proxy to browse the web and possibly do bad stuff
        with it.  It only loads pages via http and https, but it can load any
        content type. It supports GET and POST requests."""

        import urllib2
        import cgi
        import sys, os

        # Designed to prevent Open Proxy type stuff.

        allowedHosts = ['localhost', 'localhost:8080',
                                        'www.openlayers.org', 'openlayers.org',
                'labs.metacarta.com', 'world.freemap.in',
                'prototype.openmnnd.org', 'geo.openplans.org',
                'sigma.openplans.org', 'demo.opengeo.org',
                'www.openstreetmap.org', 'sample.azavea.com',
                'v2.suite.opengeo.org', 'v-swe.uni-muenster.de:8080',
                'vmap0.tiles.osgeo.org', 'www.openrouteservice.org']

        method = os.environ["REQUEST_METHOD"]

        if method == "POST":
           qs = os.environ["QUERY_STRING"]
           d = cgi.parse_qs(qs)
           if d.has_key("url"):
              url = d["url"][0]
           else:
              url = "http://www.openlayers.org"
        else:
           fs = cgi.FieldStorage()
           url = fs.getvalue('url', "http://www.openlayers.org")

        try:
           host = url.split("/")[2]
           if allowedHosts and not host in allowedHosts:
              print "Status: 502 Bad Gateway"
              print "Content-Type: text/plain"
              print
              print "This proxy does not allow you to access that location (%s)." % (host,)
              print
              print os.environ

           elif url.startswith("http://") or url.startswith("https://"):

              if method == "POST":
```

```
            length = int(os.environ["CONTENT_LENGTH"])
            headers = {"Content-Type": os.environ["CONTENT_TYPE"]}
            body = sys.stdin.read(length)
            r = urllib2.Request(url, body, headers)
            y = urllib2.urlopen(r)
        else:
            y = urllib2.urlopen(url)

        # print content type header
        i = y.info()
        if i.has_key("Content-Type"):
            print "Content-Type: %s" % (i["Content-Type"])
        else:
            print "Content-Type: text/plain"
        print

        print y.read()

        y.close()
    else:
        print "Content-Type: text/plain"
        print
        print "Illegal request."

except Exception, E:
    print "Status: 500 Unexpected Error"
    print "Content-Type: text/plain"
    print
        print "Some unexpected error occurred. Error text was:", E
```

Πίνακας 3.2 – Ρύθμιση Geoserver ως proxy βήμα 2

3. Το αρχείο proxy.cgi που μόλις δημιουργήσαμε απαιτεί Python για να το διαβάσει ο APACHE. Γι αυτό το λόγο εγκαταστήσαμε την Python 3.3.2 στη διαδρομή C:\Python33. Σε περίπτωση επιλογής άλλης διαδρομής δεν θα πρέπει να υπάρχει πρόβλημα, καθώς δεν απαιτείται αυστηρή διαδρομή από τον κώδικα.

3.1.2 Postgres/PostGIS

Αρχικά έγινε η εγκατάσταση της Postgresql 9.2.4-1 για Windows. Μέσα απο το StackBuilder κατεβάσαμε και εγκαταστήσαμε την επέκταση PostGIS 2.0. Η εγκατάσταση της Postgres έγινε στη θύρα :5432. Τέλος δημιουργήσαμε ένα πίνακα με ενεργή την επιλογή Template PostGIS 20 και όνομα wfst_test.

Σημειώνεται ότι η αλλαγή θύρας, κωδικού, ονόματος του πίνακα ή ονόματος τής 3ης στήλης του πίνακα (the_geom) απαιτεί αντίστοιχη ρύθμιση του GeoServer, ενώ δεν έχει καμμία σημασία η διαδρομή στην οποία εγκαταστάθηκαν οι Postgres και PostGIS.

Προσοχή πρέπει να δίνεται στο ότι βάση πρέπει να έχει δεδομένα σε ένα μόνο σύστημα αναφοράς και να είναι αυτό που χρησιμοποιείται και στο πρόγραμμα. Αυτό που χρησιμοποιείται στην εφαρμογή μας είναι το EPGS:4326 και στη βάση για MultiLine δεδομένα ξεκινάει με: 0001000000010200000002000000....

3.1.3 GeoServer

Εγκαταστήσαμε την τελευταία σταθερή έκδοση του GeoServer που ήταν η 2.3.4 στη διαδρομή C:\Program Files (x86)\GeoServer 2.3.4. Ο GeoServer πρέπει να εγκατασταθεί στη θύρα :8080 αλλιώς θα πρέπει να γίνουν αντίστοιχες τροποποιήσεις στο σύστημα και στον κώδικα.

Σημειώνεται επίσης ότι :

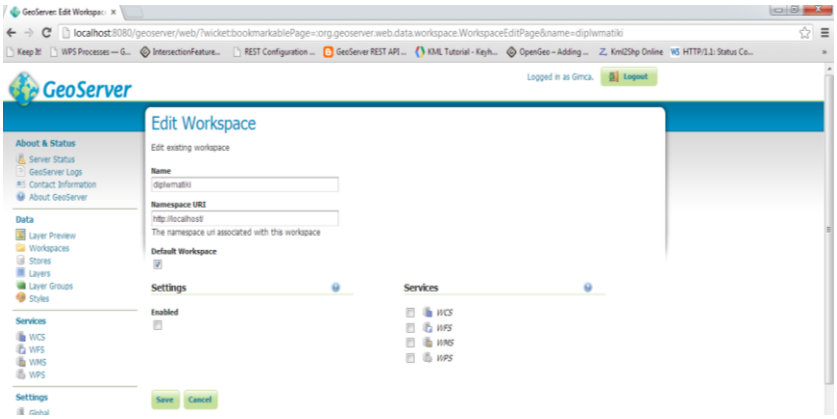
1. **οποιαδήποτε αλλαγή ή χρήση άλλων διαδρομών ή ονομάτων επιπέδων θα πρέπει να συνοδεύεται και από αντίστοιχη τροποίηση των XML αρχείων που περιγράφουν τα δεδομένα, τα WPS αιτήματα και τις διαδρομές των επιπέδων στον κώδικα**.
2. **Η αλλαγή κωδικών πρόσβασης στον GeoServer απαιτεί αλλαγή των αντίστοιχων σημείων κώδικα στα PHP scripts που χρησιμοποιούνται για το ανέβασμα δεδομένων στο GeoServer**.

Για την ενεργοποίηση της υπηρεσίας WPS κατεβάζουμε από την κεντρική σελιδα του GeoServer την WPS προέκταση και κάνουμε extract το αρχείο στην βιβλιοθήκη για εφαρμογές ιστού του GeoServer στη διαδρομή \...\GeoServer 2.3.4\webapps\geoserver\WEB-INF\lib\... και κάνουμε restart to GeoServer.

Ο GeoServer θα χρησιμοποιηθεί για την δημοσίευση τόσο των στατικών θεματικών επιπέδων τα οποία θα χρησιμοποιηθούν για την αξιολόγηση των διαδρομών όσο και του επιπέδου που θα χρησιμοποιηθεί για το σχεδιασμό της διαδρομής καθώς και για τις επεξεργασίες των επιπέδων.

Όλες οι εργασίες στον GeoServer μπορούν να γίνουν είτε μέσω του GUI του GeoServer είτε μέσω προγραμματιστικών μεθόδων. Όταν οι εργασίες που γίνονται δεν απαιτούν την χρήση πολλών αντικειμένων η χρήση του GUI είναι ικανοποιητική, όταν όμως απαιτείται φόρτωση πολλών δεδομένων (layers, styles κτλ.) η χρήση του GUI είναι πολλή κουραστική και χρονοβόρα οπότε απαιτούνται προγραμματιστικές λύσεις. Μια σειρά γλωσσών μπορουν να χρησιμοποιηθούν (Python, Javascript, PHP). Θα δωθούν παραδείγματα των εργασιών που γίνανε μέσω του GUI αλλά και πως θα μπορούσαν να υλοποιηθούν μέσω PHP κώδικα. Τέλος αξίζει να αναφερθεί ότι έχουν στηθεί οι βάσεις και αναπτύσσεται ήδη για μια εφαρμογή REST api για τον GeoServer που θα επιλύει το πρόβλημα της φόρτωσης και μεταχείρισης μεγάλου όγκων δεδομένων χωρίς να απαιτούνται γνώσεις γλωσσών προγραμματισμού.

4. Οι εργασίες στον GeoServer ξεκινούν με τη δημιουργία ενός χώρου εργασίας (workspace).



Εικόνα 3.1 – Edit Workspace

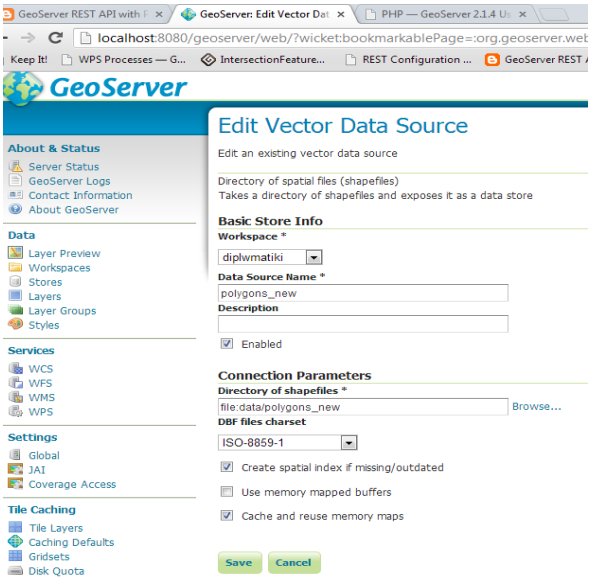
Επιλέγουμε το μενού Workspaces > add a new workspace και μετά βάζουμε το όνομα και το url με το οποίο θα γίνεται η πρόσβαση στον GeoServer και πατάμε Save. Η ενεργοποίηση των υπηρεσιών WCS, WFS, WMS και WPS είναι απαραίτητη μόνο αν έχει επιλεγεί να μην είναι ανοιχτες κεντρικά για όλο το server.

Η ίδια διαδικασία προγραμματιστικά επιτυγχάνεται με τον παρακάτω κώδικα:

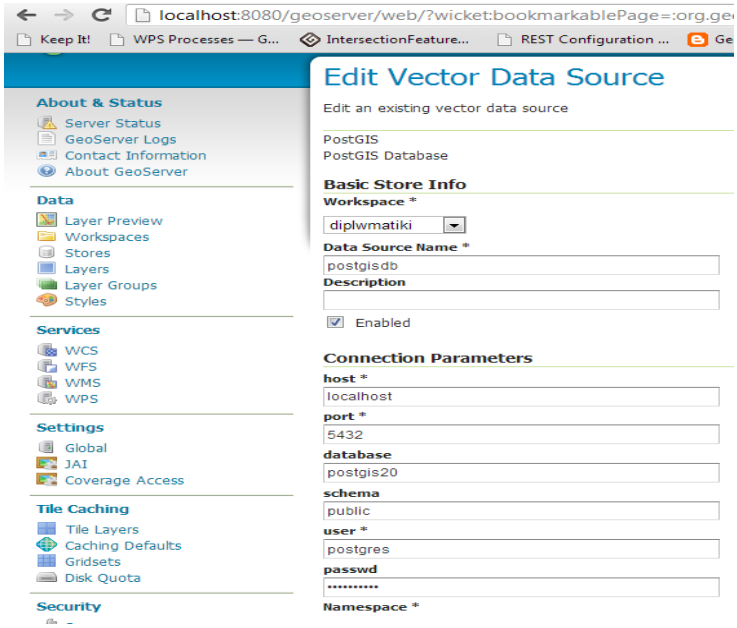
```php
<?php
    // Open log file
    $logfh = fopen("GeoserverPHP.log", 'w') or die("can't open log file");

    // Initiate cURL session
    $service = "http://localhost:8080/geoserver/"; // replace with your URL
    $request = "rest/workspaces"; // to add a new workspace
    $url = $service . $request;
    $ch = curl_init($url);

    // Optional settings for debugging
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true); //option to return string
    curl_setopt($ch, CURLOPT_VERBOSE, true);
    curl_setopt($ch, CURLOPT_STDERR, $logfh); // logs curl messages

    //Required POST request settings
    curl_setopt($ch, CURLOPT_POST, True);
    $passwordStr = "admin:geoserver"; // replace with your username:password
    curl_setopt($ch, CURLOPT_USERPWD, $passwordStr);

    //POST data
    curl_setopt($ch, CURLOPT_HTTPHEADER,
              array("Content-type: application/xml"));
    $xmlStr = "<workspace><name>diplwmatiki</name></workspace>";
    curl_setopt($ch, CURLOPT_POSTFIELDS, $xmlStr);

    //POST return code
    $successCode = 201;

    $buffer = curl_exec($ch); // Execute the curl request

    // Check for errors and process results
    $info = curl_getinfo($ch);
    if ($info['http_code'] != $successCode) {
      $msgStr = "# Unsuccessful cURL request to ";
      $msgStr .= $url." [". $info['http_code']. "]\n";
      fwrite($logfh, $msgStr);
    } else {
      $msgStr = "# Successful cURL request to ".$url."\n";
      fwrite($logfh, $msgStr);
    }
    fwrite($logfh, $buffer."\n");

    curl_close($ch); // free resources if curl handle will not be reused
    fclose($logfh);  // close logfile

?>
```

Πίνακας 3.3 - Edit Workspace με PHP κώδικα

Μετά σειρά έχει η φόρτωση των layers στο server. Αυτό γίνεται με τη δήλωση των αποθηκών δεδομένων (datastores) διαδικασία που «αποκαλύπτει» στο GeoServer τα δεδομένα που υπάρχουν και το είδος τους. Για το layer της διαδρομής απαιτείται η χρήση layer που αποθηκεύεται σε βαση δεδομένων PostGIS ώστε να είναι τροποποιήσιμο (να έχει χώρο για να αποθηκεύεται σε πραγματικό χρόνο). Για τα layers των θεματικών επιπέδων που θα χρησιμοποιηθούν για την παραμετρική αξιολόγηση της διαδρομής εφόσον αυτά είναι στατικά (δεν απαιτείται τροποποίησή τους) και με την παραδοχή ότι είναι δεδομένα που τα έχει ο χειριστής του GeoServer θα χρησιμοποιήσουμε ήδη υπάρχοντα ψηφιοποιημένα δεδομένα που υπάρχουν στο σκληρό που βρίσκεται ο GeoServer με τη μορφή Shapefiles.



Εικόνα 3.2 – Edit Datastore



Εικόνα 3.3 – Edit Datastore (PostGIS)

Επιλέγουμε το μενού Datastores > add a new datastore > directory of shapefiles και εισάγουμε το όνομα του χώρου εργασίας μας (diplwmatiki) το όνομα που θέλουμε το datastore να έχει (polygons_new) και τη διαδρομή που ο φάκελος με τα shapefiles που θέλουμε να ανεβάσουμε έχει (file:data/polygons_new) και πατάμε Save.

Για το layer από την PostGIS επιλέγουμε Datastores > add a new datastore > PostGIS και shapefiles και εισάγουμε το όνομα του χώρου εργασίας μας (diplwmatiki) το όνομα που θέλουμε το datastore να έχει (postgisdb) και τα στοιχεία της PostGIS και πατάμε Save.

Και το ίδιο σε κώδικα:

```php
<?php
    // Open log file
    $logfh = fopen("GeoserverPHP.log", 'w') or die("can't open log file");

    // Initiate cURL session
    $service = "http://localhost:8080/geoserver/"; // replace with your URL
    $request =
"rest/workspaces/diplwmatiki/datastores/polygons_new/external.shp?configure=all";
    $url = $service . $request;
    $ch = curl_init($url);

    // Optional settings for debugging
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true); //option to return string
    curl_setopt($ch, CURLOPT_VERBOSE, true);
    curl_setopt($ch, CURLOPT_STDERR, $logfh); // logs curl messages

    //Required POST request settings
    curl_setopt($ch, CURLOPT_POST, True);
    $passwordStr = "admin:geoserver"; // replace with your username:password
    curl_setopt($ch, CURLOPT_USERPWD, $passwordStr);

    //Required PUT request settings
    $fp = fopen('C:\Program Files (x86)\GeoServer 2.3.4\data_dir\data\polygons_new'
, 'r');
    curl_setopt($ch, CURLOPT_HTTPHEADER, array($data["Content-type: text/plain"]));
    curl_setopt($ch, CURLOPT_PUT, 1);
    curl_setopt($ch, CURLOPT_UPLOAD, true);
    curl_setopt($ch, CURLOPT_REFERER, true);
    curl_setopt($ch, CURLOPT_INFILE, $fp);
    curl_setopt($ch, CURLOPT_INFILESIZE, filesize('C:\Program Files (x86)\GeoServer
2.3.4\data_dir\data\polygons_new');

    //PUT return code
    $successCode = 201;

    $buffer = curl_exec($ch); // Execute the curl request

    // Check for errors and process results
    $info = curl_getinfo($ch);
    if ($info['http_code'] != $successCode) {
      $msgStr = "# Unsuccessful cURL request to ";
      $msgStr .= $url." [". $info['http_code']. "]\n";
      fwrite($logfh, $msgStr);
    } else {
      $msgStr = "# Successful cURL request to ".$url."\n";
      fwrite($logfh, $msgStr);
    }
    fwrite($logfh, $buffer."\n");

    curl_close($ch); // free resources if curl handle will not be reused
    fclose($logfh);  // close logfile

?>
```

Πίνακας 3.4 – Edit Datastore με PHP κώδικα

```php
<?php
    // Open log file
    $logfh = fopen("GeoserverPHP.log", 'w') or die("can't open log file");

    // Initiate cURL session
    $service = "http://localhost:8080/geoserver/rest/"; // replace with your URL
    $request = "workspaces/diplwmatiki/datastores/";

    $url = $service . $request;
    $ch = curl_init($url);

    // Optional settings for debugging
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true); //option to return string
    curl_setopt($ch, CURLOPT_VERBOSE, true);
    curl_setopt($ch, CURLOPT_STDERR, $logfh); // logs curl messages

    //Required POST request settings
    curl_setopt($ch, CURLOPT_POST, True);
    $passwordStr = "admin:geoserver"; // replace with your username:password
    curl_setopt($ch, CURLOPT_USERPWD, $passwordStr);

    //POST data
    curl_setopt($ch, CURLOPT_HTTPHEADER, array("Content-type: text/xml"));
    curl_setopt($ch, CURLOPT_POSTFIELDS, $xml = file_get_contents
("http://localhost/example scripts/upload_postgisdb.xml"));

    //POST return code
    $successCode = 201;

    $buffer = curl_exec($ch); // Execute the curl request

    // Check for errors and process results
    $info = curl_getinfo($ch);
    if ($info['http_code'] != $successCode) {
      $msgStr = "# Unsuccessful cURL request to ";
      $msgStr .= $url." [". $info['http_code']. "]\n";
      fwrite($logfh, $msgStr);
    } else {
      $msgStr = "# Successful cURL request to ".$url."\n";
      fwrite($logfh, $msgStr);
    }
    fwrite($logfh, $buffer."\n");

    curl_close($ch); // free resources if curl handle will not be reused
    fclose($logfh);  // close logfile
?>
```

Πίνακας 3.5 – Edit Datastore με PHP κώδικα (PostGIS)

```
<dataStore>
<name>postgisdb</name>
<connectionParameters>
      <host>localhost</host>
      <port>5432</port>
      <database>test_wfst</database>
      <schema>public</schema>
      <user>postgres</user>
      <passwd>postgres</passwd>
      <dbtype>postgis</dbtype>
</connectionParameters>
</dataStore>
```

Πίνακας 3.6 – Φύλλο XML με τα δεδομένα της PostGIS

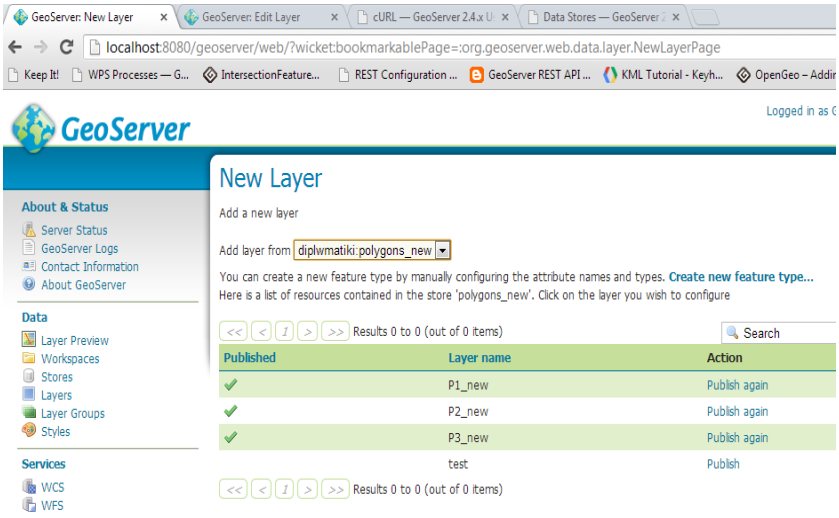
Τελικά πρέπει να γίνει η δημοσίευση αυτών των layers.

Επιλέγουμε Layers > add a new resource > add a layer from – diplwmatiki:polygons_new (ή diplwmatiki:postgisdb)

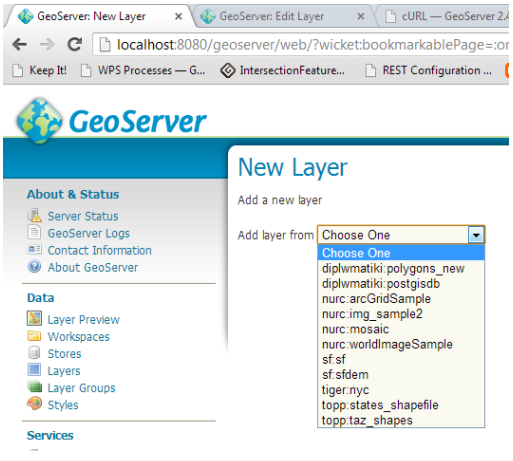
Επιλέγουμε το Layer που θέλουμε (θα έχουν το native όνομα των shapefiles που υπάρχουν στο φάκελο που δηλώσαμε στο server) και πατάμε Publish

Συμπληρώνουμε υποχρεωτικά τα πεδία: Name (όνομα), Native SRS (το σύστημα αναφοράς των δεδομένων – μπορεί να είναι και UNKNOWN), Declared SRS (το σύστημα αναφοράς στο οποίο δηλώνονται ή επαναπροβάλονται τα δεδομενα).

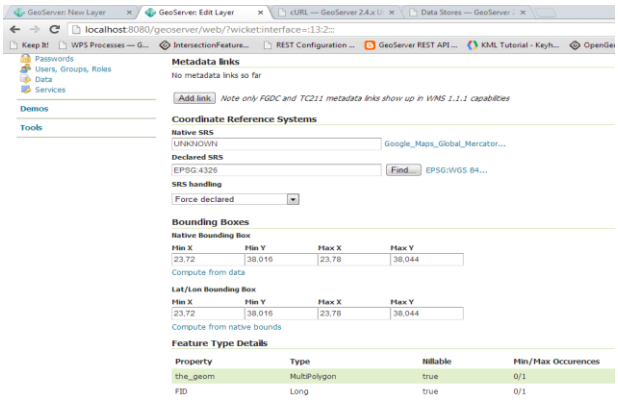
Επιλέγουμε SRS HANDLING > FORCE DECLARED, Native Bounding Box > Compute from data, Lan/Lon Bounding Box > Compute from data



Εικόνα 3.4 α



Εικόνα 3.4 β



Εικόνα 3.4 γ

Εικόνες 3.4 α,β,γ – Δημοσίευση Δεδομένων

Τέλος πηγαίνουμε στο tab Publishing και επιλέγουμε τα στυλιστικά χαραχτηριστικά που θέλουμε το layer να έχει.

Με τη προγραμματιστική μέθοδο τα layers υπάρχουν ήδη έτοιμα για χρήση και τους έχει δωθεί το default style με βάση τη γεωμετρία του κάθε αντικειμένου.

Το σύστημα τώρα είναι στημένο και έτοιμο για να αναπτυχθεί ο κώδικας με τον οποίο

1. Θα προβάλλονται σε περιβάλλον oplenlayers διαδικτυακά τα δεδομένα μας.
2. Θα γίνονται στο server όλες οι επεξεργασίες.

3.2 Συστατικά στοιχεία λογισμικού που χρησιμοποιούνται

Πέρα από το στήσιμο του συστήματος (Goserver, Apache/PHP, Postgres/PostGIS) χρησιμοποιήθηκαν μια σειρά λογισμικών για την περάτωση της εργασίας.

3.2.1 QuantumGIS

Το Quantum GIS (QGIS) είναι ένα λογισμικό GIS ανοιχτού κώδικα, φιλικό στο χρήστη, όπου μπορεί να γίνει απεικόνιση, διαχείριση, επεξεργασία, ανάλυση και σύνθεση χαρτών. Ενσωματώνει ισχυρές αναλυτικές δυνατότητες μέσω της ολοκλήρωσής του με το GRASS. Μπορεί να εκτελεστεί σε Linux, Unix, Mac OSX, και Windows. Υποστηρίζει πολλαπλά διανυσματικά, εικονιστικά πρότυπα αρχείων, πολλαπλούς τύπους βάσεων δεδομένων και αντίστοιχη λειτουργικότητα γι αυτά.

Βασικά Χαρακτηριστικά:

- Γραφικό περιβάλλον διεπαφής που επιτρέπει την αναγνώριση και επιλογή χαρακτηριστικών, την επεξεργασία/οπτικοποίηση/αναζήτηση περιγραφικών χαρακτηριστικών, την άμεση αλλαγή προβολικού συστήματος, τη χρήση συμβόλων χαρακτηριστικών, τις αλλαγές συμβόλων για διανυσματικά και εικονιστικά δεδομένα,.
- Εύκολη προεπισκόπηση πολλών διανυσματικών και εικονιστικών προτύπων ψηφιακών αρχείων όπως πίνακες της βάσης δεδομένων PostgreSQL. Υποστηρίζει τα περισσότερα διανυσματικά πρότυπα: περιλαμβανομένου των ESRI shapefiles, MapInfo, SDTS και GML. Παρέχει υποστήριξη για εικονιστικά δεδομένα όπως Ψηφιακά Μοντέλα Εδάφους, αεροφωτογραφίες και δορυφορικές εικόνες, για πρότυπα του GRASS και για ανάγνωση διαδικτυακών υπηρεσιών θέασης και μεταφόρτωση του OGC (WMS ή WFS).
- Δημιουργία, επεξεργασία και εξαγωγή χωρικών δεδομένων με τη χρήση εργαλείων ψηφιοποίησης του GRASS και το πρότυπο shapefile, του πρόσθετου γεωαναφοράς (plugin).
- Εργαλεία GPS για την εισαγωγή και εξαγωγή αρχείων GPX, τον μετασχηματισμό από άλλα πρότυπα GPS σε GPX , ή η μεταφόρτωση απευθείας αρχείων σε δέκτη GPS
- Η εκτέλεση χωρικών αναλύσεων με τα πρόσθετα fTools και GRASS
- Δημοσιοποίηση στο διαδίκτυο
- Αρχιτεκτονική με πρόσθετα (plugins).

Στο QuantumGIS έγιναν όλες οι απαραίτητες ψηφιοποιήσεις δεδομένων.

3.2.2 Openlayers

Το OpenLayers είναι μια βιβλιοθήκη Javascript που κάνει δυνατή την οπτικοποίηση δεδομένων στο. Το OpenLayers διευκολύνει τους προγραμματιστές διαδικτυακών εφαρμογών να ενσωματώσουν στις σελίδες τους δυναμικούς χάρτες, από μια πληθώρα πηγών δεδομένων. Το OpenLayers παρέχει ένα επεκτάσιμο σύνολο χαρτογραφικών εργαλείων και παραθύρων, παρόμοια με το Google Maps API. Όλη η λειτουργικότητα εκτελείται

στον περιηγητή, κάτι το οποίο καθιστά το OpenLayers πολύ εύκολο στην εγκατάσταση, χωρίς καθόλου εξάρτηση από τον εξυπηρετητή.

Στο OpenLayers γίνονται όλες οι απεικονίσεις των χαρτών, των θεματικών επιπέδων και ο σχεδιασμός των διαδρομών από το χρήστη.

3.2.3 Τοπικός Κώδικας

Για την διεκπαιρέωση των εργασιών που απαιτούνται αναπτύχθηκε τοπικά κώδικας μέρος του οποίου γράφτηκε εξ' αρχής και μέρος του οποίου ενσωματώθηκε είτε με μετατροπές είτε έτοιμο.

Κατ' αρχήν αναπτύχθηκε κώδικας σε για τη ρύθμιση του συστήματος (APACHE/PHP και GeoSever) που περιγράφεται στο κεφάλαια 3.1.1. Αναπτύχθηκε επίσης κώδικας σε PHP που επιτρέπει τον προγραμματιστικό χειρισμό του Geoserver και περιγράφεται στο κεφάλαιο 3.1.3.

Το κυρίως πρόγραμμα (αυτό που χρησιμοποιεί δηλαδή ο χρήστης) είναι γραμένο σε HTML στο οποίο την κύρια δουλειά την κάνουν ενσωματομένα scripts σε Javascript. Η HTML επίσης καλεί εξωτερικά PHP scripts τα οποία αφού επεξεργαστούν κάποια XML φύλλα με δεδομένα επιστρέφουν στο κυρίως πρόγραμμα δεδομένα. Οι λειτουργίες του κώδικα αυτού αναλύονται στα κεφάλαιο 3.4 και 3.5.

3.2.4 MS-PAINT

Αποτελεί σχεδιαστικό λογισμικό ήδη ενσωματομένο στα Windows με το οποίο έγινε η επεξεργασία των εικονιδίων των πλήκτρων που χρησιμοποιεί ο χρήστης για το σχεδιασμό και την αποθήκευση της διαδρομής.

3.2.5 Notepad++

Αν και δεν αποτελεί λογισμικό που απαιτούταν για την εργασία είναι άξιο αναφοράς ως ένα εύχριστο και αξιόπιστο εργαλείο συγγραφής κώδικα. Όλος ο κώδικας που αναπτύχθηκε για την εφαρμογή έγινε σε περιβάλον Notepad++.

Το Notepad++ είναι ένας δωρεάν επεξεργαστής πηγαίου κώδικα και αντικαταστάτης του Σημειωματάριου που υποστηρίζει πολλές γλώσσες. Τρέχει σε περιβάλλον MS Windows, και η χρήση του διέπεται από την Γενική Δημόσια Άδεια GPL. Κατά τη λειτουργία του χρησιμοποιεί τη λιγότερη δυνατή ισχύ του επεξεργαστή.

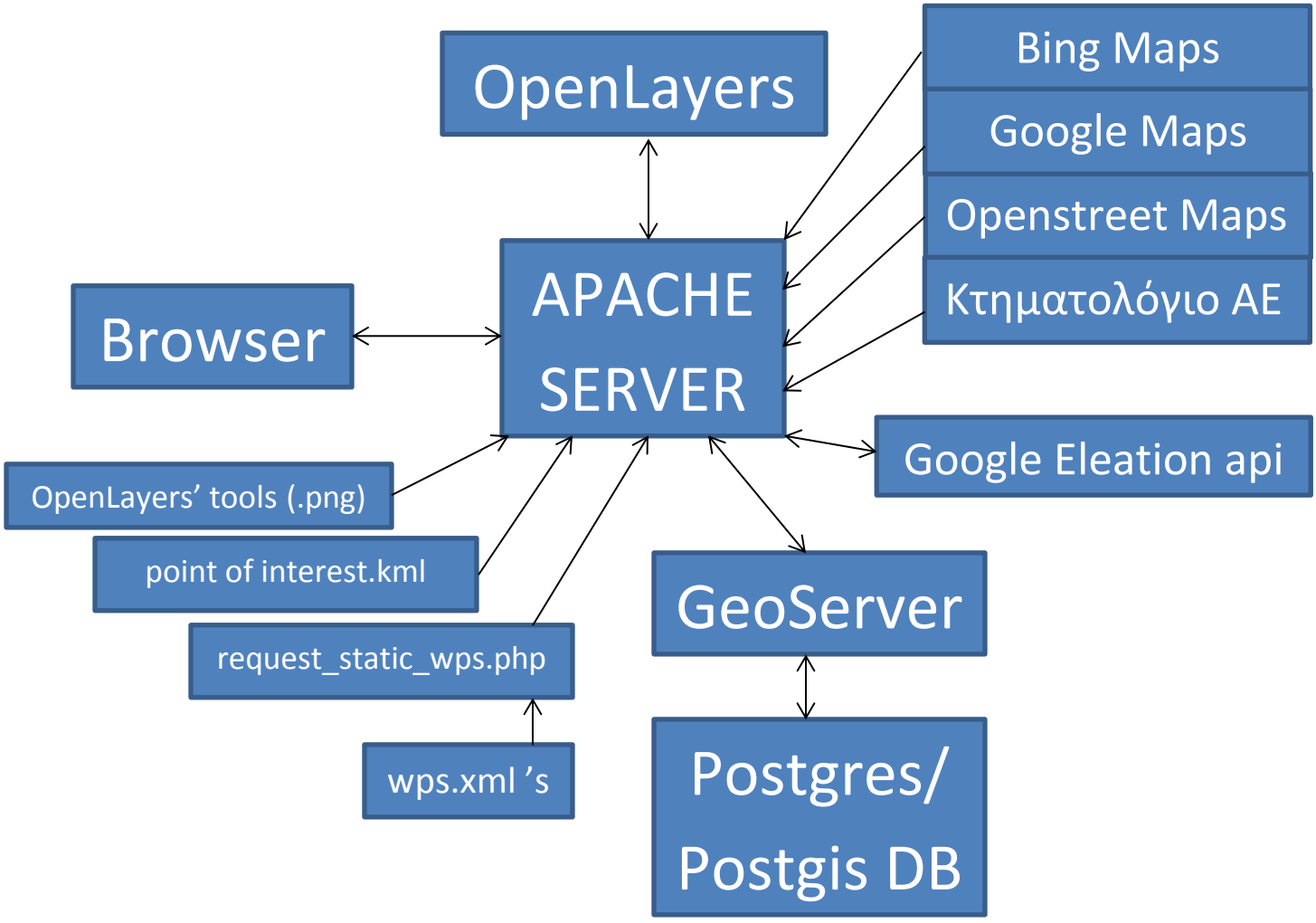
Χαρακτηριστικά:

- Επισήμανση και αναδίπλωση σύνταξης
- Καθορισμός από χρήστη της επισήμανσης και αναδίπλωσης
- Αναζήτηση/Αντικατάσταση PCRE (τυπική έκφραση συμβατή με Perl)
- Περιβάλλον χρήστη πλήρως προσαρμόσιμο: μινιμαλιστικό, καρτέλα με κουμπί κλεισίματος, καρτέλες σε πολλαπλές σειρές, κάθετη καρτέλα και κάθετη λίστα εγγράφου
- Χάρτης εγγράφου
- Αυτόματη συμπλήρωση: συμπλήρωση λέξης, συμπλήρωση λειτουργίας και υπόδειξη παραμέτρων λειτουργίας

- Πολλαπλά έγγραφα (καρτέλα διεπαφής)
- Πολλαπλή προβολή
- WYSIWYG (Εκτύπωση)
- Μεγέθυνση και σμίκρυνση
- Υποστήριξη πολύγλωσσου περιβάλλοντος
- Σελιδοδείκτης
- Εγγραφή και αναπαραγωγή μακροεντολής
- Εκκίνηση με διάφορες παραμέτρους

3.3 Διάγραμμα Ροής δεδομένων και συνεργασίας μεταξύ των επιμέρους στοιχείων

Παρακάτω περιγράφεται η αρχιτεκτονική του συστήματος:



Εικόνα 3.5 – Αρχιτεκτονική του συστήματος

3.4 Υπηρεσίες που χρησιμοποιήθηκαν/ενσωματώθηκαν έτοιμες

Μια σειρά υπηρεσιών που καλύπτουν ένα φάσμα απαιτήσεων της εργασίας ενσωματώθηκαν έτοιμες με κάποια παραμετροποίηση ώστε να εξασφαλιστεί η ομαλή λειτουργία με το σύστημα και τα δεδομένα μας:

3.4.1 Openlayers

Η υπηρεσία OpenLayers χρησιμοποιήθηκε για την προβολή των δεδομένων μας στο χρήστη και ο σχεδιασμός από αυτόν της διαδρομής. Οι ρυθμίσεις για την επικοινωνία της υπηρεσίας με τον browser και τον χρήστη καθώς και με τον Geoserver και τους παροχείς των χαρτών γίνονται με τη χρήση τοπικού κώδικα. Περιγραφή της λειτουργίας υπάρχει στο κεφάλαιο 2.7.7

3.4.2 Google Maps, Bing Maps, Openstreet Maps & Κτηματολόγιο Α.Ε.

Όλες αυτές οι υπηρεσίες είναι πάροχοι χαρτογραφικών υποβάθρων που χρησιμοποιούνται στην εφαρμογή που αναπτύχθηκε. Αξίζει να σημειωθεί ότι το υπόβαθρο της Κτηματολόγιο Α.Ε. είναι ακριβέστερο καθώς πρόκειται για γεωμετρικά διορθωμένες αεροφωτογραφίες και όχι απλά δορυφορικές εικόνες, οι οποίες επειδή επειδή είναι στο σύστημα αναφοράς ΕΓΣΑ ’87 είναι απαραίτητος ο μετασχηματισμός τους για τη χρήση σε συνδιασμό με χάρτες άλλων παρόχων.

3.4.3 Google Elevation api

Πρόκειται για μια υπηρεσία που προσφέρει τη δυνατότητα αναζήτησης υψομέτρων για κάθε σημείο στην επιφάνεια της γης καθώς για περιοχές που δεν υπάρχουν δεδομένα πραγματοποιεί γραμμική παρεμβολή. Ο χρήστης μπορεί να αναζητήσει υψόμετρα κατα μήκος διαδρομής, δίνοντας τη δυνατότητα υπολογισμού υψομετρικών διαφορών κατά μήκος διαδρομών. Σημειώνεται ότι η υπηρεσία έχει περιορισμό κλήσεων 2,500 υψομετρικών αιτημάτων / ημέρα καθώς και σε κάθε υψομετρικό αίτημα ο χρήστης μπορεί να ζητήσει υψομετρική πληροφορία μέχρι 512 περιοχών με ανώτατο όριο τις 25,000 περιοχές / ημέρα.

Τα δεδομένα αυτά πρέπει να μετασχηματιστούν σε γεωδαιτικά, να γίνουν οι κατάλληλοι υπολογισμοί των αποστάσεων πανω στη σφαίρα και να υπολογιστούν οι κλίσεις κατά μήκος της διαδρομής.

Ο τοπικός κώδικας που καλεί την υπηρεσία και ρυθμίζει τα δεδομένα και την εμφάνιση των αποτελεσμάτων και κάνει τους υπολογισμούς ενσωματώθηκε έτοιμος.

3.4.4 Εύρεση των αποστάσεων από τα Μ.Μ.Μ.

Ο κώδικας που υπολογίζει την απόσταση μεταξύ των σταθμών των Μ.Μ.Μ. που έχουμε στη διαδεσή μας και της διαδρομής που σχεδιάζει ο χρήστης και επιστρέφει τα εντός των ορίων αποτελέσματα χρησιμοποιήθηκε ως εφαρμογή αυτούσιος.

3.5 Συστατικά στοιχεία λογισμικού που αναπτύχθηκαν

Το λογισμικό που αναπτύχθηκε χωρίζεται χοντρικά στα εξής μέρη:

1. Η επικοινωνία με τον GeoServer για το ανέβασμα και τη διαχείριση των δεδομένων. Ο κώδικας γράφτηκε σε PHP και περιγράφηκε στο κεφάλαιο 3.1.3.
2. Τα XML που περιγράφουν τα δεδομένα και το ερώτημα στον GeoServer για τον υπολογισμό των μηκών των τομών.
3. Ο PHP κώδικας που επικοινωνεί με τον GeoServer και του στέλνει τα XML.
4. Ο κώδικας σε Javascript που κάνει τη «βασική δουλειά», το στήσιμο και την επικοινωνία των διάφορων υπηρεσιών και κομματιών του συστήματος.
5. Η προβολή των αποτελεσμάτων σε HTML περιβάλλον.

Το σύνολο του κώδικα παρατίθεται στο παράρτημα.

3.5.1 Ο υπολογισμός του μήκους των τομών

Οποιαδήποτε ερώτηση για επεξεργασία δεδομένων προς τον GeoServer γίνεται σε γλώσσα XML, είτε για να περιγράψει τα δεδομένα που εμπλέκονται είτε για να περιγράψει τις ίδιες τις επεξεργασίες που ζητούνται. Η δομή για κάθε τύπο δεδομένων μπορεί να βρεθεί είτε με μια GET ερώτηση στο REST api είτε μέσα από το GUI του GeoServer.

Το ερώτημα που κάνουμε στον Geoserver για τον υπολογισμό του μήκους της τομής της διαδρομής και των θεματικών επιπέδων είναι μια σύνθετη ερώτηση που στον GeoServer επιλύεται ως εξής:

Πρώτα υπολογίζεται η τομή του εκάστοτε θεματικού επίπεδου με τη διαδρομή. Τα θεματικά επίπεδα είναι σύνθετα πολύγωνα (MultiPolygon) τα οποία περιγράφουν το ένα το άλλο – δηλαδή το εξωτερικό όριο του ενός είναι εσωτερικό του άλλου – ενώ η διαδρομή ως σύνθετη τεθλασμένη γραμμή (MultiPolyline). Η εύρεση της τομής γίνεται με το τρέξιμο της εντολής gs:IntersectionFeatureCollection που απαιτεί την είσοδο δύο layers και επιστρέφει ένα νέο που είναι η τομή τους. Η δομή της είναι η εξής:

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>gs:IntersectionFeatureCollection</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>first feature collection</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs"
method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2">
```

```
            <wfs:Query/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>second feature collection</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs"
method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2">
            <wfs:Query/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>
```

Πίνακας 3.7 - gs:IntersectionFeatureCollection

Έπειτα συλλέγεται το αποτέλεσμα και ενοποιείται σε ένα νέο layer ώστε να μπορεί μετά να υπολογιστεί το μήκος του. Αυτό γίνεται με την εντολή gs:CollectGeometries η οποία πέρνει ως δεδομένα ένα οποιποτε σύνολο αντικειμένων και γεωμετριών και τα ενοποιεί σε layer. Η διαδικασία αυτή είναι απαραίτητη γιατί η τομή της διαδρομής με ένα επίπεδο μπορεί να μην είναι ένα μόνο ευθύγραμμο τμήμα αλλά πολλά οπότε και αυτά θα πρέπει από απλές γραμμές (PolyLines) να γίνουν ένα layer με μία γεωμετρία (MultiPolyLines). Η δομή της είναι η εξής:

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>gs:CollectGeometries</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>features</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs"
method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2">
            <wfs:Query/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
  </wps:DataInputs>
```

```
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml; subtype=gml/3.1.1">
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>
```

Πίνακας 3.8 - gs:CollectGeometries

Το μήκος των τομών υπολογίζεται με την εντολή JTS:length. Η JTS:length δέχεται ως δεδομένα είτε γραμμές είτε πολύγωνα επιστρέφοντας μήκος ή περίμετρο αντίστοιχα. Εισάγωντας της ως δεδομένα το αποτέλεσμα της gs:CollectGeometries υπολογίζει το μήκος της διαδρομής μέσα στη βαθμίδα του θεματικού επιπέδου. Η δομή της είναι η εξής:

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>JTS:length</ows:Identifier>
  <wps:DataInputs/>
  <wps:ResponseForm>
    <wps:RawDataOutput>
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>
```

Πίνακας 3.9 - JTS:length

Οι εντολές αυτές ενσωματώνονται η μία μέσα στην άλλη στον τελικό κώδικα έτσι ώστε τα αποτελέσματα της μίας να αποτελούν τα δεδομένα της άλλης. Η ιεραρχία των πράξεων είναι: gs:IntersectionFeatureCollection > gs:CollectGeometries > JTS:length αλλά στον κώδικα τοποθετούνται με ακριβώς την αντίθετη σειρά σε ένα ενιάιο αίτημα και αποθηκεύονται σε ένα .XML αρχείο. Για κάθε ερώτημα που θέλουμε να γίνει υπάρχουν δύο δεδομένα, η γραμμή και το εκάστοτε κομμάτι του εκάστοτε θεματικού επιπέδου. Αυτό σημαίνει ότι θα παραχθούν τόσα .XML αρχεία όσα και τα layers που αποτελούν το κάθε θεματικό επίπεδο.

3.5.2 Η αποστολή των XML στον GeoServer

Για την αποστολή των ερωτημάτων στον GeoServer αναπτύχθηκε PHP κώδικας ο οποίος συνδέεται με τον GeoServer βάζει τους κωδικους και μέσω μιας POST εντολής αποστέλει τα XML αρχεία και του επιστρέφονται τα αποτελέσματα (τα μήκη των τομών) που τα αποθηκεύει πρόχειρα σε κάποιες μεταβλητές.

3.5.3 Το στήσιμο του OpenLayers Interface

Η απουσία μηχανισμού αποθήκευσης των σχεδιαζομενων layers από το OpenLayers καθιστά χωρίς νόημα τη πραγματοποίηση των αιτημάτων προς τον GeoServer γιατί γι' αυτόν δεν θα εμφανίζεται τροποποίηση του layer της διαδρομής και θα τρέχει τα ερωτήματα ως προς την τελευταία αποθηκευμένη διαδρομή. Έπρεπε λοιπόν να αναπτυχθεί μια τεχνική αποθήκευσης και ένα panel εργαλείων ψηφιοποίησης καθώς τα ήδη έτοιμα που παρέχονται δεν επιδέχονται τροποποίησης. Έτσι ήταν απαραίτητο να παραχθούν:

1. Κώδικας που να περιγράφει τα γραφικά των πλήκτρων
2. Κώδικας που να πραγματοποιεί την λειτουργία των πλήκτρων ψηφιοποίησης.
3. Μια τεχνική αποθήκευσης του επιπέδου που ψηφιοποιεί ο χρήστης.
4. Τα εικονίδια των πλήκτρων.

3.5.4 Η σύνδεση στο OpenLayers των θεματικών επιπέδων και της διαδρομής

Η σύνδεση των θεματικών επιπέδων στο OpenLayers γίνεται με την απλή εφαρμογή του WMS πρωτόκολου ώστε απλά να φορτωθούν τα δεδομένα από τον GeoServer στη μηχανή απεικόνισης του OpenLayers

Η σύνδεση του layer της διαδρομής απαιτεί την εφαρμογή του WFS-T πρωτόκολου ώστε να επιτρέπει και την επιστροφή στον GeoServer δεδομένων και την αποθήκευσή τους στο βαση δεδομένων PostGIS ώστε να ακολουθήσουν οι κατάλληλες επεξεργασίες

3.5.5 Η εμφάνιση των αποτελεσμάτων

Τα αποτελέσματα που επιστρέφει ο GeoServer είναι αποθηκευμένα σε μεταβλητές στο PHP script που διεκπαιρεώνει την επικοινωνία με τον GeoServer. Η HTML όμως δεν μπορεί να επεξεργαστεί μεταβλητές της PHP και άρα δεν μπορεί να πει στον Browser τι να τυπώσει. Για το λόγο αυτό αναπτύχθηκε μέσα στον κυρίως κώδικα που είναι σε Javascript μια εφαρμογή AJAX που ανοίγει το PHP script, διαβάζει τις μεταβλητές και επιστρέφει τα δεδομένα τους με τρόπο που η HTML να μπορεί να διαβάζει και τέλος γράφτηκε το απαραίτητο κομμάτι σε HTML που τυπώνει τα αποτελέσματα. Σε ατίθεση με την λογική που υπάρχει με τις κλήσεις – που με το πέρας της ψηφιοποίησης εμφανίζονται τα αποτελέσματα – επιλέχθηκε η χρήση πλήκτρου για να υπενθιμίζει την ανάγκη για αποθήκευση της διαδρομής και όχι απλά το σχεδιασμό της.

3.6 Ροή ελέγχου κατά την εκτέλεση και η «αρμοδιότητα» εμπλεκόμενων μερών

1. Με την κλήση της εφαρμογής από το φυλλομετρητή ο APACHE στέλνει μια σειρά αιτημάτων :
   - Σε βιβλιοθήκες που απαιτούνται για την εκτέλεση της εφαρμογής
   - Στις υπηρεσίες – παρόχους των υποβάθρων (προεπιλεγμενη είναι η υπηρεσία Openstreet Maps)
   - Στον GeoServer που παρέχει τα θεματικά επίπεδα και το layer της διαδρομής.

   Έπειτα ζητάει από το OpenLayers να προβάλει τα δεδομένα. Η διαδικασία αυτή επαναλαμβάνεται κάθε φορά που καλείται η εφαρμογή ή που ζητείται διαφορετικό υπόβαθρο.

2. Έπειτα εισάγεται η διαδρομή. Η διαδικασία αυτή γίνεται με 3 τρόπους:

   - Με ήδη υπάρχοντα δεδομένα που καλούνται απ' ευθείας από την PostgGIS βάση δεδομένων κατά την εκκίνηση της εφαρμογής.
   - Με την ψηφιοποίηση από τον χρήστη νέων δεδομένων.
   - Με την εισαγωγή από το χρήστη νέων έτοιμων δεδομένων.

   Με το πέρας της ψηφιοποίησης της διαδρομής ο APACHE παίρνει τα δεδομένα από το OpenLayers και τα στέλνει στο Google Elevation api το οποίο επιστρέφει είτε το σύνολο των αιτημάτων είτε όσα επιτρέπονται με βάση τους περιορισμούς. Στη δεύτερη περίπτωση εμφανίζεται στο χρήστη μύνημα στον χρήστη και δεν προχωρά η επεξεργασία των υψομετρικών δεδομένων. Στη πρώτη περίπτωση προχωρά κανονικά η επεξεργασία τους που γίνεται τοπικά και εκτυπώνονται τα αποτελέσματα των κλίσεων της διαδρομής και των αποστάσεων από τα Μ.Μ.Μ. στο φυλλομετρητή με βάση τις παραμέτρους που έχει θέσει ο χρήστης. Παράλληλα παράγεται κείμενο που περιγράφει τη διαδρομή και το οποίο μπορεί να εξάγει ο χρήστης.

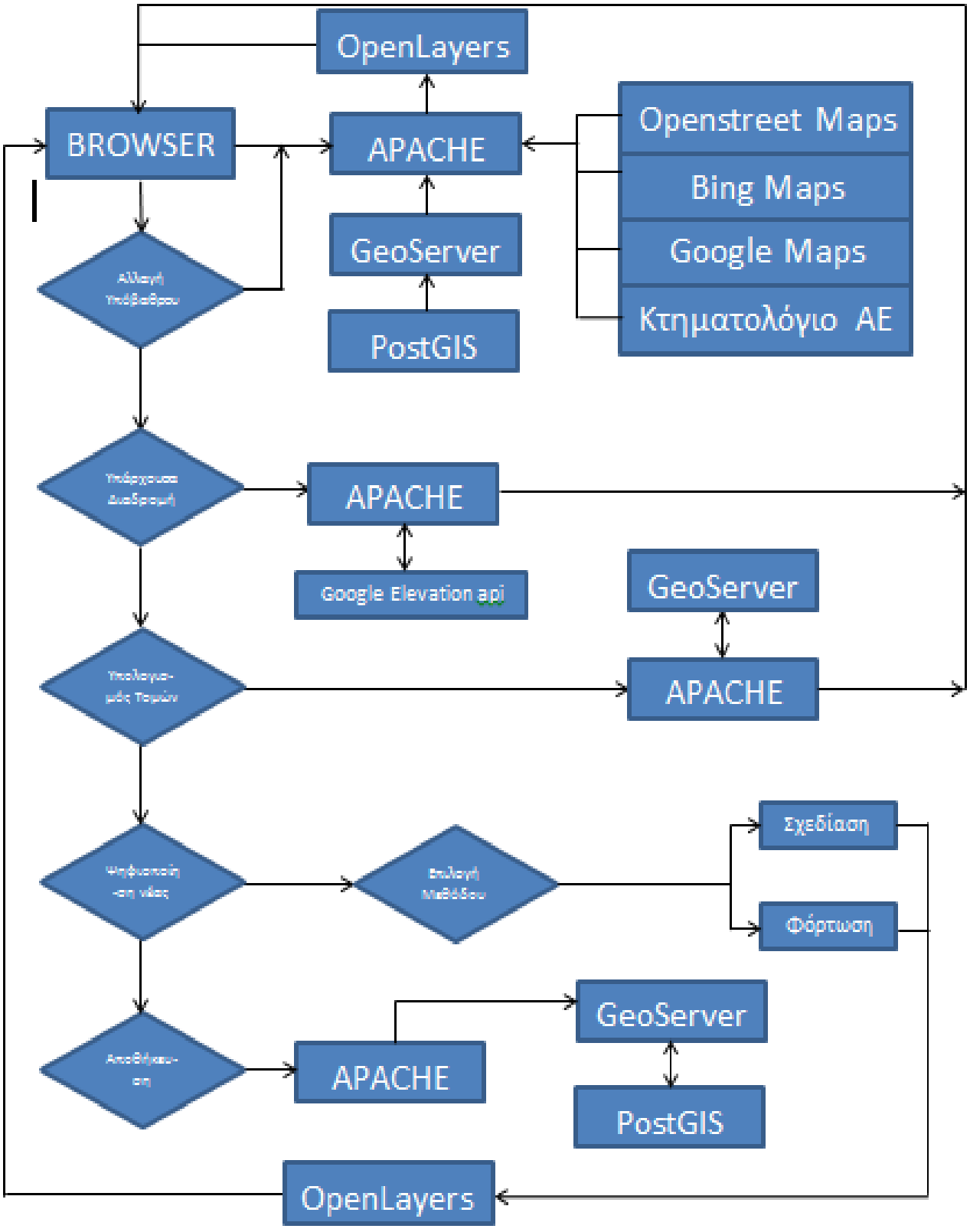
   Η διαδικασία αυτή επαναμβάνεται κάθε φορά που ολοκληρώνεται η διαδικασία ψηφιοποίησης της διαδρομής

3. Η αποθήκευση της ψηφιοποίησης είναι απαραίτητη για να επεξεργαστεί ο GeoServer τα νέα δεδομένα. Αλλιώς αν ζητηθεί ο υπολογισμός των μήκων των τομών ο GeoServer θα χρησιμοποιήσει την τελευταία αποθηκευμένη διαδρομή (ασχέτως τι εμφανίζεται στην οθόνη).

   Κατά τη διαδικασία αυτή ο APACHE παίρνει τα δεδομένα από το OpenLayers και τα στέλνει στο GeoServer ο οποίος ενημερώνει την PostGIS βάση δεδομένων και φορτώνει στον ίδιο τα νέα δεδομένα.

4. Η αίτηση για τον υπολογισμό των τομών στέλνεται στον GeoServer ο οποίος μετά την επεξεργασία επιστρέφει τα αποτελέσματα στον APACHE για να τυπωθούν στο φυλλομετρητή.

Ακολουθεί διάγραμμα ροής που περιγράφει γραφικά την διαδικασία:



Εικόνα 3.6 - Ροή εργασιών κατά την εκτέλεση και η «αρμοδιότητα» εμπλεκόμενων μερών

ΚΕΦΑΛΑΙΟ 4

Εφαρμογή - Παράδειγμα Χρήσης

Για το σκοπό της επίδειξης της εφαρμογής που αναπτύχθηκε θα γίνει μια εργασία αξιολόγησης διαδρομών για την κατασκευή ποδηλατοδρόμων ποδηλατοδρόμων. Θα γίνει αρχικά η παρουσίαση του interface και των εργαλείων της εφαρμογής καθως και των επεξεργασιών και των υπολογισμών που πραγματοποιεί. Η περιοχή στην οποία θα γίνει η ψηφιοποίηση και η αξιολόγηση της διαδρομής είναι τμήμα της οδού Μιχαλακοπούλου στην Αθήνα.

4.1 Προηγούμενες Εργασίες

Προηγούμενη εργασία κομμάτια της οποίας έχουν ενσωματωθεί αυτούσια ή με μετατροπές είναι η διπλωματική εργασία :

«Ανάπτυξη Διαδικτυακής Εφαρμογής για την Αξιολόγηση Παραμέτρων στον Σχεδιασμό Δικτύου Ποδηλατικών Διαδρομών » - ΘΕΟΔΩΡΟΠΟΥΛΟΣ ΑΘΑΝΑΣΙΟΣ – ΓΕΩΡΓΙΟΣ

Η λειτουργίας της εφαρμογής είναι δημιουργία διαδικτυακής υπηρεσίας, η οποία χρησιμοποιεί χαρτογραφικό υπόβαθρο που διανέμεται δωρεάν στο διαδίκτυο (Google, Open Street Maps, Ktimatologio A.E, Bing Maps) και η απεικόνιση συγκεκριμένων γεωγραφικών δεδομένων στο υπόβαθρο αυτό (συγκεκριμένα οι θέσεις των σταθμών του μετρό της Αθήνας, του ηλεκτρικου και του προαστιακού σιδηρόδρομου της Αθήνας). Η εφαρμογή διαθέτει μια εργαλειοθήκη ψηφιοποίησης, δίνοντας την δυνατότητα στον χρήστη να δημιουργεί τα δικά του γεωμετρικά στοιχεία, χρησιμοποιώντας ως χαρτογραφικό υπόβαθρο τους χάρτες που χρησιμοποιεί η εφαρμογή. Πέρα από την δυνατότητα ψηφιοποίησης, θα παρέχεται η δυνατότητα στον χρήστη να εισάγει τα δικά του αρχεία, τα οποία και θα μπορούν να προβληθούν στο χαρτογραφικό υπόβαθρο. Επιπροσθέτως θα παρέχεται στους χρήστες η δυνατότητα μεγέθυνσης, σμίκρυνσης, μετακίνησης και επιλογής στο διαθέσιμο χαρτογραφικό υπόβαθρο. Κατά την ψηφιοποίηση οι συντεταγμένες των γεωμετρικών στοιχείων θα καταγράφονται και θα στέλνονται στην υπηρεσία elevation api της google η οποία και θα επιστρέφει τα υψόμετρα των κορυφών που ψηφιοποιήθηκαν (και των οποίων οι συντεταγμένες καταγράφηκαν). Με χρήση των συνεταγμένων και των αντίστοιχων υψομέτρων θα υπολογίζεται η απόσταση των κορυφών μεταξύ και η αντίστοιχη κλίση των πλευρών του δικτύου (κλίση= υψομετρική διαφορά προς απόσταση). Επιπροσθέτως θα υπολογίζεται η απόσταση του ποδηλατικού δικτύου από τους σταθμούς των Μ.Μ.Μ. Τέλος θα παρέχεται η δυνατότητα αποθήκευσης των αρχείων που δημιούργησε ο χρήστης. Παράδειγμα χρήσης της εφαρμογής μπορεί να βρεθεί στο Κεφάλαιο 7.

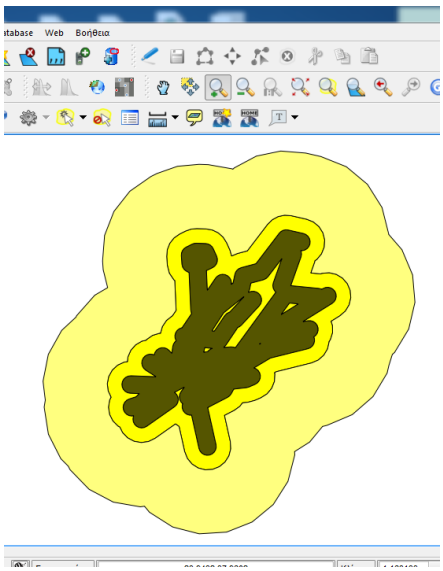
Περιορισμός της συγκεκριμένης εφαρμογής ήταν η αδυναμία επεξεργασίας της διαδρομής με άλλα διανυσματικά ψηφιοποιημένα δεδομένα και η εφαρμογή σε αυτά GIS εργαλείων και μεθόδων.

4.2 Επεκτάσεις - Γενικεύσεις

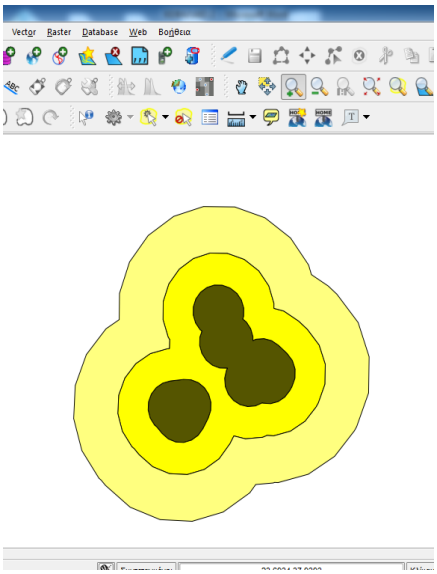
Οι επεκτάσεις που γίνανε πάνω σε αυτή την κεντρική ιδέα είχαν ως στόχο να άρουν αυτόν το περιορισμό και να επιτρέψουν την επεξεργασία της ψηφιοποιημένης διαδρομής με άλλα διανυσματικά στοιχεία ώστε να γίνει δυνατή η αξιολόγηση της διαδρομής με πολλά επιπλέον δεδομένα (πέρα από την κλίση και την απόσταση από τους σταθμούς M.M.M). Αυτό απαιτούσε την αποστολή δεδομένων σε υπηρεσία WEB που να μπορεί να πραγματοποιήσει GIS επεξεργασίες.

Απαραίτητες διαδικασίες ήταν η ανάπτυξη νέων εργαλείων ψηφιοποίησης σε σχέση με τα default εργαλεία του OpenLayers που χρησιμοποιούσε η προϋπάρχουσα εφαρμογή, η σύνδεση των θεματικών επιπέδων που θα χρησιμοποιηθούν για την αξιολόγηση της διαδρομής και η αποστολή των δεδομένων από την εφαρμογή μας στο GeoServer για να γίνουν οι χωρικές επεξεργασίες και τέλος η προβολή των αποτελεσμάτων.

Για το παράδειγμά μας ψηφιοποιήθηκαν 2 θεματικά επίπεδα που αναπαριστούν δεδομένα για την ηχορύπανση και την ατμοσφαιρική ρύπανση στην Αθήνα. Τα δεδομένα αυτά περιγράφονται με ισοδυναμικές καμπύλες. Η υλοποίηση τους γίνεται με πολύγωνα τα οποία το καθένα έχει εσωτερικό όριο του πολυγώνου που περιγράφει και εξωτερικό το πολύγωνο από το οποίο περιγράφεται. Το κάθε θεματικό επίπεδο έχει 3 layers. Παρατίθονται παρακάτω τα θεματικά επίπεδα:



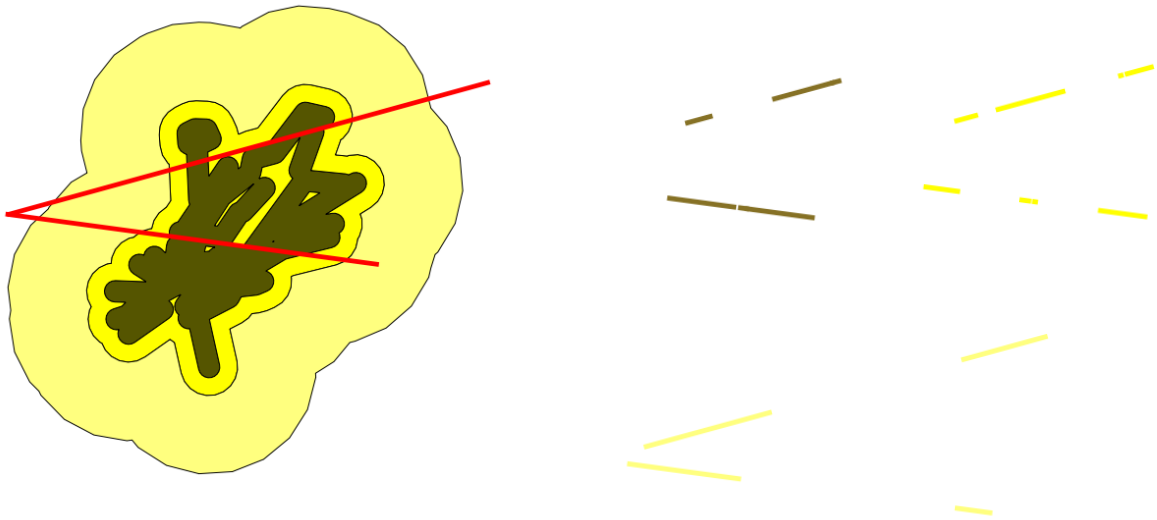
Εικόνα 4.1 - Ηχορύπανση



Εικόνα 4.2 – Ατμοσφαιρικοί Ρύποι

Υπολογίζονται οι τομές της διαδρομής με το κάθε layer του κάθε θεματικού επιπέδου, αθροίζονται τα αποτελέσματα σε ενιαίο layer γεωμετρίας MultiPolyLine και υπολογίζεται το μήκος του κάθε MultiPolyLine που αποτελεί το μήκος που διατρέχει η διαδρομή στην κάθε βαθμίδα του κάθε θεματικού επιπεδου.

Παρατίθεται παράδειγμα μιας διαδρομής και των τομών που παράγονται:



Εικόνα 4.3 – Παράδειγμα υπολογισμού των τομών με το θεματικό επίπεδο «ηχορύπανση»

4.3 Συνόψιση των υπολογιζόμενων δεικτών

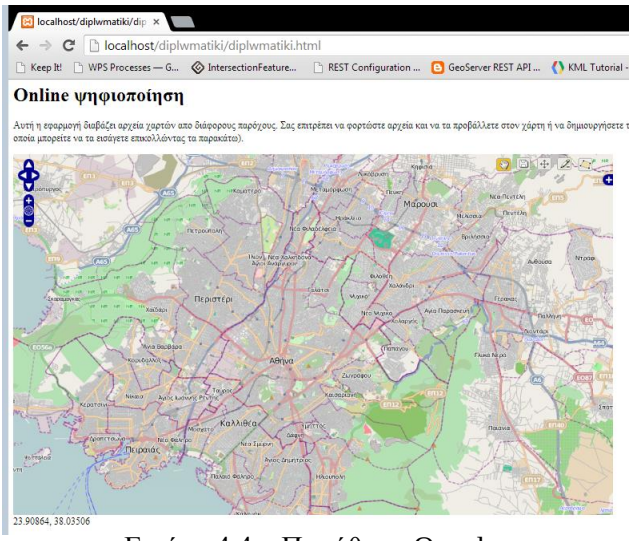
Η εφαρμογή που αναπτύχθηκε επιτρέπει στο χρήστη:

- να χρησιμοποιήσει ήδη υπάρχουσα ή να ψηφιοποιήσει ή ανεβάσει έτοιμη νέα διαδρομή
- να παραχθεί κώδικας σε μια σειρά format που να περιγράφει τη διαδρομή
- να υπολογίσει τη κλίση κάθε ευθύγραμμου τμήματος της διαδρομής και τη μέση κλίση και να τη παραστήσει γραφικά
- να υπολογίσει τους σταθμούς των Μ.Μ.Μ που υπάρχουν μέσα στα όρια που θέτει ο χρήστης
- να υπολογίσει τα μήκη των τομών της διαδρομής με θεματικά επίπεδα που αναπαριστούν διάφορους βαθμούς έκθεσης σε ηχορύπανση και ατμοσφαιρική μόλυνση

Η εφαρμογή παρέχει παράθυρο προβολής των διανυσματικών δεδομένων, μια σειρά από χαρτογραφικά υπόβαθρα και εργαλεία ψηφιοποίησης.

4.4 Παράδειγμα χρήσης της εφαρμογής

Κατα την εκκίνηση της εφαρμογής ο χρήστης βλέπει το εξής:



Εικόνα 4.4 – Παράθυρο Openlayers

Τα εργαλεία της εφαρμογής είναι τα εξής:

Επιτρέπει την κίνηση στο χάρτη, εναλλακτικά μπορεί να χρησιμοποιηθεί το 3^{ο} πλήκτρο του ποντικιού κρατημένο

Εικόνα 4.5

Επιτρέπει τη μεγένθυνση/σμίκρυνση του χάρτη, εναλακτικά μπορεί να χρησιμοποιηθεί η κύλιση της ροδέλας

Εικόνα 4.6

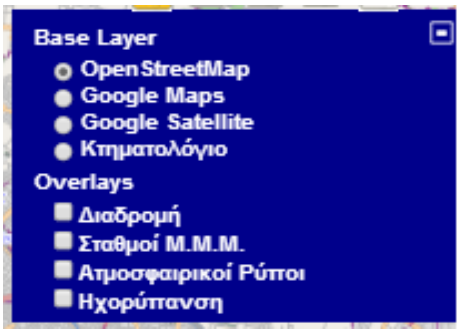
Εικόνα 4.7

Τα εργαλεία ψηφιοποίησης από αριστερά προς δεξιά:

- Εργαλείο μετακίνησης – επιτρέπει την κίνηση πάνω στο χάρτη.
- Εργαλείο αποθήκευσης – αποθηκεύει τη διαδρομή στη PostGIS βάση δεδομένων.
- Εργαλείο τροποποίησης – επιτρέπει τη μετακίνηση των κόμβων της διαδρομής για τη διόρθωση τυχών λαθών.
- Εργαλείο διαγραφής – επιτρέπει τη διαγραφή μιας διαδρομής, ενεργοποιώντας το εργαλείο και επιλεγοντας διαδρομή.
- Εργαλεί σχεδίασης – επιτρέπει το σχεδιασμό μιας διαδρομής, ενεργοποιώντας το εργαλείο και κάνοντας κλικ στο χάρτη δημιουργεί τον πρώτο κομβο, έπειτα κάθε κλικ προσθέτει κόμβο ενώ το τέλος του σχεδιασμού γινεται με διπλό κλικ βάζοντας τον τελευταίο κόμβο.

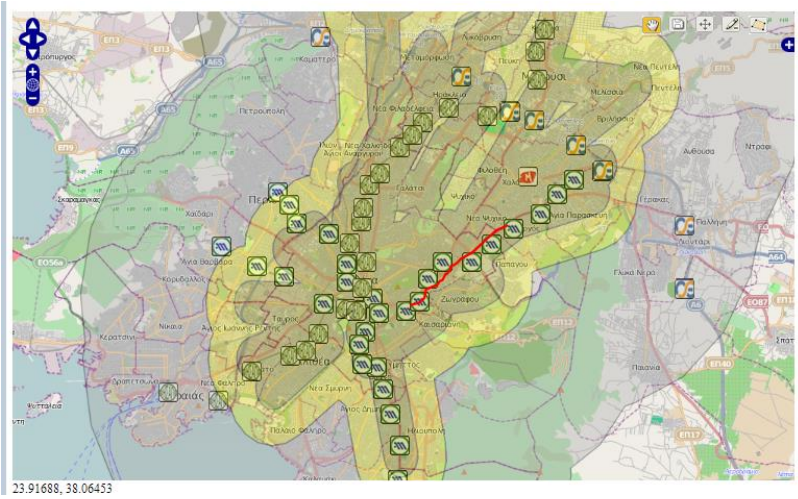
Αναδυόμενο μενού που επιτρέπει:

- Την επιλογή χαρτογραφικού υποβάθρου
- Την ενεργοποίηση / απενεργοποίηση θεματικών επιπέδων



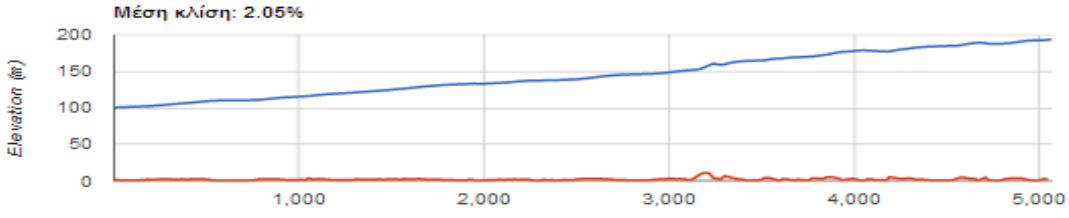
Εικόνα 4.8

Ενεργοποιούμε τα θεματικά επίπεδα Σταθμοί Μ.Μ.Μ. και ηχορύπανση και σχεδιάζουμε μια διαδρομή:



Εικόνα 4.9

Με το πέρας του σχεδιασμού γίνονται οι υπολογισμού των κλίσεων και των αποστάσεων από τα Μ.Μ.Μ. και παράγεται και κείμενο που περιγράφει τη διαδρομή (το default κριτήριο αποστάσεων είναι τα 1000m και το default format είναι GeoJSON). Τέλος αποθηκεύοντας τη διαδρομή και πατώντας το πλήκτρο υπολόγισε τις τομές γίνεται ο υπολογισμός των τομών και επιστρέφονται τα αποτελέσματα.



Εικόνα 4.10 - Κλίσεις

Σταθμός	Απόσταση
Metro-Γραμμές 2-3 - Εθνική Άμυνα / Ethniki Amyna	12 μέτρα
Metro-Γραμμές 2-3 - Κατεχάκη / Katechaki	22 μέτρα
Metro-Γραμμές 2-3 - Μέγαρο Μουσικής / Megaro Mousikis	286 μέτρα
Metro-Γραμμές 2-3 - Ευαγγελισμός / Evangelismos	311 μέτρα
Metro-Γραμμές 2-3 - Χολαργός / Cholargos	444 μέτρα
Metro-Γραμμές 2-3 - Πανόρμου / Panormou	783 μέτρα
Metro-Γραμμές 2-3 - Αμπελόκηποι / Ampelokipi	789 μέτρα

Εικόνα 4.11 – Αποστάσεις από τους σταθμούς Μ.Μ.Μ


```
{
    "type": "Feature",
    "properties": {
    },
    "geometry": {
        "type": "MultiLineString",
        "coordinates": [
            [
                [
                    23.74921115875388,
                    37.97502852747521
                ],
                [
                    23.753159370423997,
                    37.975874254223974
                ],
                [
                    23.755348052979926,
                    37.97749802226702
                ],
                [
                    23.756249275208834,
                    37.980474837044156
                ],
                [
                    23.758998857240105,
                    37.98148963266583
                ],
                [
                    23.760841217042458,
                    37.98189554698527
                ],
```


Εικόνα 4.12 – Αρχείο Εξόδου

Αφού σώσετε τη διαδρομή σας κάντε στο "ΥΠΟΛΟΓΙΣΕ ΤΟΜΕΣ" για ν

Τομή της διαδρομής με το επίπεδο μόλυνση1: 321.00969950366 m

Τομή της διαδρομής με το επίπεδο μόλυνση2: 3840.0879711098 m

Τομή της διαδρομής με το επίπεδο μόλυνση3: 1136.4638543536 m

Τομή της διαδρομής με το επίπεδο ηχορύπανση1: 5297.5615249671 m

Τομή της διαδρομής με το επίπεδο ηχορύπανση2: 0 m

Τομή της διαδρομής με το επίπεδο ηχορύπανση3: 0 m

ΥΠΟΛΟΓΙΣΕ ΤΟΜΕΣ

Εικόνα 4.13 – Υπολογισμός των μηκών των τομών

ΚΕΦΑΛΑΙΟ 5

Ιδέες Για Περεταίρω Επεκτάσεις

Η εφαρμογή που αναπτύχθηκε έχει μεγάλη δυνατότητα παραμετροποίησης και επέκτασης.

Περισσότερο απ' όλα όμως η εφαρμογή αυτή αποτελεί απόδειξη ότι τα μέρη που χρησιμοποιούνται στο σύστημα, οι υπηρεσίες και τα πρωτόκολα επικοινωνίας που εφαρμόζονται δίνουν τη δυνατότητα για ανάπτυξη ολοκληρωμένων WEB GIS εφαρμογών ή ακόμα και πλήρης σουίτας λογισμικού για την ανάπτυξη WEB GIS εφαρμογών με σαφή πρωτερήματα έναντι των Desktop υλοποιήσεων:

- Ταχύτητα διάδοσης των δεδομένων καθώς όλα γίνονται online.
- Τη μη απαίτηση / εξάρτηση εξειδικευμένων λογισμικών και λειτουργικών.
- Το μηδενικό κόστος για το στήσιμο και χρήση του.
- Η λογική του ανοιχτού λογισμικού που επιτρέπει την γρήγορη και πολύπλευρη ανάπτυξη των εργαλείων και των εφαρμογών.

5.1 Προσθήκη νέων παραμέτρων

Η πιο απλή επέκταση αυτής της εφαρμογής θα ήταν η προσθήκη κάποιας νέας παραμέτρου που θα απαιτεί άλλου τύπου επεξεργασίες από τον GeoServer και άρα και την ανάπτυξη εκείνων των XML που θα αξιοποιούν τα κατάλληλα εργαλεία και δεδομένα (π.χ. μια buffer επεξεργασία της διαδρομής για να εξεταστεί αν στην ακτίνα ανοχής υπάρχουν αντικείμενα που θέλουμε ή προσπαθούμε να αποφύγουμε).

5.2 Ανανέωση Δεδομένων

Μια μάλλον αναγκαία επέκταση για τη χρήση της εφαρμογής με πραγματικά δεδομένα online που θα απαιτούσε τη σύνδεση των θεματικών επιπέδων ως προς τα οποία εξετάζουμε τη διαδρομή με κάποιον πάροχο τέτοιων δεδομένων τοσο για λόγους εγκυρότητας των δεδομένων όσο και για συχνή ανανέωση (ακόμη και σε πραγματικό χρόνο αν το επιτρέπει ο πάροχος). Π.χ. Με το που οι κρατικές υπηρεσίες θα ενημερώνανε για τις μεταβολές στη μέση συγκέντρωση ρύπων ή γινόταν νέα μελέτη για δείκτες ηχορύπανσης τα αντίστοιχα θεματικά επίπεδα να ανανεωνόντουσαν αυτόματα.

5.3 Ανάπτυξη Εφαρμογής για το χειρισμό του GeoServer

Η ανάπτυξη μιας εφαρμογής (που μπορεί εύκολα να ενσωματωθεί στην εφαρμογή μας) η οποία θα χειρίζεται με φιλικό τρόπο για το χρήστη το ανέβασμα δεδομένων στον GeoServer και τη δημοσίευσή τους καθώς και το χειρισμό των εργαλείων επεξεργασίας χωρικών δεδομένων που έχει δίνει τη δυνατότητα στον χρήστη να ανεβάζει

τα δικά του δεδομένα και να τα αξιοποιεί καθώς και να αναπτύσει τις δικές του επεξεργασίες για την αξιολόγηση της διαδρομής. Θα μπορούσε κάλλιστα να δίνεται και η δυνατότητα να δημοσιεύει αυτήν την επεξεργασία και να ενσωματώνεται στην εφαρμογή.

Κάτι τέτοιο θα έδινε εντελώς άλλη δυναμική στην εφαρμογή καθώς θα ήταν δυνατό:

- Να αξιοποιούνται τα δεδομένα των χρηστών.
- Να επιτρέπεται στους χρήστες να παράγουν νέους δείκτες αξιολόγης των διαδρομών.
- Οι ίδιοι οι χρήστες να εμπλουτίζουν τη διαδρομή με νέα εργαλεία και μεθόδους αξιολόγησης και να τα χρησιμοποιούν κατα βούληση σε πληθώρα εργασιών.

5.4 Ανάπτυξη δυναμικών εφαρμογών που να παράγουν διαδρομές

Με πυρήνα τη λογική της εφαρμογής μας θα μπορούσαν να αναπτυχθούν δυναμικές εφαρμογές που με βάση τα κριτήρια και τα δεδομένα που εισάγονται να παράγουν βέλτιστη ή πολλά σενάρια διαδρομών που να ικανοποιούν τα κριτήρια που τους δόθηκαν. Εφαρμογές τέτοιου τύπου θα μπορούσαν όχι εύκολα να αντικαταστήσουν την ικανότητα του χρήστη να παίρνει τις αποφασεις αλλά σίγουρα θα μπορούσαν να επιταχύνουν τη διαδικασία απόφασης.

5.5 Εξέταση διαδρομών που κάνει ο χρήστης

Αυτή η εφαρμογή μπορεί να προσαρμοστεί ώστε να χρησιμοποιηθεί όχι για το σχεδισμό βέλτιστων διαδρομών, αλλά για την αξιολόγηση διαδρομών που κάνει ο χρήστης. Οι διαδρομές θα παράγονται από κάποιο GPS (στο αμάξι, στο κινητό κτλ.) του χρήστη που θα αποθηκεύουν τη διαδρομή σε πραγματικό χρόνο – καθώς αυτή γίνεται – και θα μπορεί έπειτα να την επεξεργαστεί στην εφαρμογή η οποία θα του παρέχει τα δεδομένα εκείνα που είναι ο χρήστης χρειάζεται για να προβεί στην αξιολόγηση της διαδρομής που πραγματοποίησε.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Κωνσταντίνος Σιασιάκος – Πληροφορική VI: Δυναμικές Εφαρμογές Παγκόσμιου Ιστού.

Μ. Δρόσος – Σύγκριση Λογισμικών WEB GIS.

Δρ. Ηλίας Φρέντζος – Εισαγωγή στη PostgreSQL / PostGIS.

T. Converse, J. Park, C. Morgan – PhP5 and MySQL Bible, Wiley 2004.

S. Koch – Javascript, Wiley, 2003.

A. Fabio – XML Developpers Guide, McGraw-Hill 2001.

GeoServer 2.3.x User's Guide http://docs.geoserver.org/stable/en/user/

GeoServer 2.3.x Developer's Guide http://docs.geoserver.org/stable/en/developer/

Α. Θεοδωρόπουλος – Ανάπτυξη Διαδικτυακής Εφαρμογής για την Αξιολόγηση Παραμέτρων στον Σχεδιασμό Δικτύου Ποδηλατικών Διαδρομών – Διπλωματική Εργασία

Μ. Στρηλίγκα – Ανάπτυξη Υπηρεσιών Προστιθέμενης Αξίας με Αξιοποίηση Web Services σε Περιβάλλον Ανοικτού Κώδικα στο Διαδίκτυο – Διπλωματική Εργασία

ΙΣΤΟΛΟΓΙΑ:

http://geoserver.org – Η ιστοσελίδα του GeoServer

http://openlayers.org – Η ιστοσελίδα του OpenLayers

http://opengeo.org – Ολοκληρωμένη πλατφόρμα WEB GIS

http://gistutor.com – Παραδείγματα WEB GIS εφαρμογών

https://wiki.state.ma.us/confluence/display/massgis/Getting+Started – Ιστοσελίδα του MassGIS – διαδικτυακή υπηρεσία επεξεργασίας γεωχωρικών δεδομένων

https://developers.google.com/kml/documentation/ – Παραδείγματα KML κώδικα και εφαρμογών

https://php.net – Η ιστοσελίδα της PHP

http://w3schools.com/js – Ιστοσελίδα εκμάθησης Javascript

http://w3schools.com/xml – Ιστοσελίδα εκμάθησης XML

http://w3schools.com/html – Ιστοσελίδα εκμάθησης HTML

http://stackoverflow.com – Ιστότοπος Διαλόγου για Προγραμματισμό

http://w3.org – Διεθνής κοινότητα που αναπτύσει τα Web Standars

http://wikipedia.org – Διαδικτυακή Εγκυκλοπαίδεια

ΠΑΡΑΡΤΗΜΑ

- **Ο κώδικας της εφαρμογής**

```
<html>
<head>

    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <script type="text/javascript"
src="http://openlayers.org/api/OpenLayers.js"></script>
    <script type="text/javascript"
src="http://www.openstreetmap.org/openlayers/OpenStreetMap.js"></script>
    <script type="text/javascript" src="https://www.google.com/jsapi"></script>
    <script type="text/javascript"
src="http://maps.google.com/maps/api/js?sensor=false&libraries=geometry"></scrip
t>

    <style>
        .customEditingToolbar {
            float: left;
            right: 0px;
            height: 30px;
            width: 200px;}

        .customEditingToolbar div {
            float: left;
            margin: 5px;
            width: 24px;
            height: 24px;}

        .olControlNavigationItemActive {
            background-image: url("editing_tool_bar.png");
            background-repeat: no-repeat;
            background-position: -103px -23px;}

        .olControlNavigationItemInactive {
            background-image:url("editing_tool_bar.png");
            background-repeat: no-repeat;
            background-position: -103px -0px;}

        .olControlDrawFeaturePolygonItemInactive {
            background-image: url("editing_tool_bar.png");
            background-repeat: no-repeat;
            background-position: -26px 0px;}

        .olControlDrawFeaturePolygonItemActive {
            background-image: url("editing_tool_bar.png");
            background-repeat: no-repeat;
            background-position: -26px -23px;}

        .olControlModifyFeatureItemActive {
            background-image: url("move_feature_on.png");
            background-repeat: no-repeat;
```

```
            background-position: 0px 1px;}

        .olControlModifyFeatureItemInactive {
            background-image: url("move_feature_off.png");
            background-repeat: no-repeat;
            background-position: 0px 1px;}

        .olControlDeleteFeatureItemActive {
            background-image: url("remove_point_on.png");
            background-repeat: no-repeat;
            background-position: 0px 1px;}

        .olControlDeleteFeatureItemInactive {
            background-image: url("remove_point_off.png");
            background-repeat: no-repeat;
            background-position: 0px 1px;}
    </style>

    <script type="text/javascript">

    //ajax get on request_static_wps
    function loadXMLDoc() {
        var xmlhttp;
        if (window.XMLHttpRequest)
            {// code for IE7+, Firefox, Chrome, Opera, Safari
            xmlhttp=new XMLHttpRequest();}
        else
            {// code for IE6, IE5
            xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");}
        xmlhttp.onreadystatechange=function() {
            if (xmlhttp.readyState==4 && xmlhttp.status==200) {

document.getElementById("myDiv").innerHTML=xmlhttp.responseText;}}
        xmlhttp.open("GET","request_static_wps.php",true);
        xmlhttp.send();}


    //including the apache changes for localhost access
    OpenLayers.ProxyHost = "/cgi-bin/proxy.cgi?url=";

    // Load the Visualization API and the columnchart package.
    google.load("visualization", "1", {packages: ["corechart"]});

    var map;
    var WGS84_google_mercator,formats,wfst_layer;
    var WGS84;
    var elevator, chart;

    function enhmeroshformat() {

        var in_options = {'internalProjection': map.baseLayer.projection,
'externalProjection': new
OpenLayers.Projection(OpenLayers.Util.getElement("inproj").value)};
        var out_options = {'internalProjection': map.baseLayer.projection,
'externalProjection': new
OpenLayers.Projection(OpenLayers.Util.getElement("outproj").value)};
        var gmlOptions = {featureType: "feature",featureNS:
```

```
"http://example.com/feature"};
        var gmlOptionsIn = OpenLayers.Util.extend(OpenLayers.Util.extend({},
gmlOptions), in_options);
        var gmlOptionsOut = OpenLayers.Util.extend(OpenLayers.Util.extend({},
gmlOptions), out_options);
        var kmlOptionsIn = OpenLayers.Util.extend({extractStyles: true},
in_options);

            formats = {'in': {
                wkt: new OpenLayers.Format.WKT(in_options),
                geojson: new OpenLayers.Format.GeoJSON(in_options),
                georss: new OpenLayers.Format.GeoRSS(in_options),
                gml2: new OpenLayers.Format.GML.v2(gmlOptionsIn),
                gml3: new OpenLayers.Format.GML.v3(gmlOptionsIn),
                kml: new OpenLayers.Format.KML(kmlOptionsIn),
                atom: new OpenLayers.Format.Atom(in_options),
                gpx: new OpenLayers.Format.GPX(in_options)
              }, 'out': {
                wkt: new OpenLayers.Format.WKT(out_options),
                geojson: new OpenLayers.Format.GeoJSON(out_options),
                georss: new OpenLayers.Format.GeoRSS(out_options),
                gml2: new OpenLayers.Format.GML.v2(gmlOptionsOut),
                gml3: new OpenLayers.Format.GML.v3(gmlOptionsOut),
                kml: new OpenLayers.Format.KML(out_options),
                atom: new OpenLayers.Format.Atom(out_options),
                gpx: new OpenLayers.Format.GPX(out_options)
              }
            };
    }

    function init() {
        // set up projections
        // World Geodetic System 1984 projection (lon/lat)
        WGS84 = new OpenLayers.Projection("EPSG:4326");

        // WGS84 Google Mercator projection (meters)
        WGS84_google_mercator = new OpenLayers.Projection("EPSG:900913");
        //Initialize the map

        //creates a new openlayers map in
        //the <div> html element with id="map"
        map = new OpenLayers.Map ("map", {
            controls: [
                //allows user pan/zoom ability
                new OpenLayers.Control.Navigation(),
                //displays the pan/zoom tools
                new OpenLayers.Control.PanZoom(),
                //displays a layer switcher
                new OpenLayers.Control.LayerSwitcher(),
                //displays the mouse positions coordinates in a
                //<div> html element with id coordinates
                new OpenLayers.Control.MousePosition({div:
document.getElementById("coordinates")})],
                    projection: WGS84_google_mercator, displayProjection:
WGS84});

        //base layers
```

```
        var openstreetmap = new OpenLayers.Layer.OSM();
        var google_maps = new OpenLayers.Layer.Google("Google Maps",
{numZoomLevels: 20});
        var google_satellite = new OpenLayers.Layer.Google("Google Satellite",
{type: google.maps.MapTypeId.SATELLITE,numZoomLevels: 20});
        KT = new OpenLayers.Layer.WMS("Κτηματολόγιο",
"http://gis.ktimanet.gr/wms/wmsopen/WmsServer.aspx",
                { layers: "basic", reaspect: "true", transparent: 'true' }, {
'buffer': 0 });
                KT.isBaseLayer = true;
                KT.setTileSize(new OpenLayers.Size(1000, 1000));

        ////////////////////////////////////////////////////
        /////fortosh arxeioy kml pou periexei////////////////
        ///toys sta8moys metro hsap kai proastiakou/////////
        ///auto to arxeio 8a xrhsimopoih8ei gia na vre8oun//
        ///oi apostaseis apo thn sxediasmenh grammh ////////
        ////////////////////////////////////////////////////
        kmlLayer = new OpenLayers.Layer.Vector("Σταθμοί M.M.M.", {protocol: new
OpenLayers.Protocol.HTTP({url: "point of interest.kml",format: new
OpenLayers.Format.KML({extractStyles: true, extractAttributes: true,
internalProjection: WGS84_google_mercator})}),strategies: [new
OpenLayers.Strategy.Fixed()]});

        ////////////////////////////////
        //Φόρτωση του editable layer//
        ///////από το Geoserver///////
        ////////////////////////////////
        var saveStrategy = new OpenLayers.Strategy.Save(); //set up a save
strategy

        var wfst_layer = new OpenLayers.Layer.Vector("Διαδρομή", {
            strategies: [new OpenLayers.Strategy.BBOX(), saveStrategy],
            projection: 'EPSG:4326',
            //srsName: "EPSG:4326",
            protocol: new OpenLayers.Protocol.WFS({
                version: "1.1.0",
                srsName: 'EPSG:4326',
                // loading data through localhost url path
                url: "http://localhost/geoserver/wfs",
                featurePrefix: "diplwmatiki",
                featureNS :  "http://localhost/",
                maxExtent: mapextent,
                // layer name
                featureType: "wfst_test",
                // geometry column name
                geometryName: "the_geom",
                schema:
"http://localhost/geoserver/wfs/DescribeFeatureType?version=1.1.0&typename=diplw
matiki:wfst_test"})});

        ////////////////////////////////////
        //set up the modification tools//
        ////////////////////////////////////
        var DeleteFeature = OpenLayers.Class(OpenLayers.Control,{
            initialize:function(layer, options) {
                OpenLayers.Control.prototype.initialize.apply(this, [options]);
```

```
                this.layer = layer;
                this.handler = new OpenLayers.Handler.Feature(
                this, layer, {click: this.clickFeature});},
            clickFeature: function(feature) {
            // if feature doesn't have a fid, destroy it
                if(feature.fid == undefined)
{this.layer.destroyFeatures([feature]);}
                else {feature.state = OpenLayers.State.DELETE;
                    this.layer.events.triggerEvent("afterfeaturemodified",
                    {feature: feature});
                    feature.renderIntent = "select";
                    this.layer.drawFeature(feature);}},
            setMap: function(map) {
            this.handler.setMap(map);
            OpenLayers.Control.prototype.setMap.apply(this, arguments);},
            CLASS_NAME: "OpenLayers.Control.DeleteFeature"});

        var panel = new OpenLayers.Control.Panel({'displayClass':
'customEditingToolbar'});
        var navigate = new OpenLayers.Control.Navigation({title: "Pan Map"});
        var draw = new OpenLayers.Control.DrawFeature(wfst_layer,
OpenLayers.Handler.Path,{
            title: "Draw Feature",
            displayClass: "olControlDrawFeaturePolygon",
            multi: true});
        var edit = new OpenLayers.Control.ModifyFeature(wfst_layer, {
            title: "Modify Feature",
            displayClass: "olControlModifyFeature"});
        var del = new DeleteFeature(wfst_layer, {title: "Delete Feature"});
        var save = new OpenLayers.Control.Button({title: "Save Changes",
trigger: function() {
            saveStrategy.save(); // alert('saved');
            }, displayClass: "olControlSaveFeatures"});
        panel.addControls([navigate, save, edit, del, draw]);
        panel.defaultControl = navigate;
        map.addControl(panel);
        //map.addControl(new OpenLayers.Control.EditingToolbar(wfst_layer));

        ////////////////////////
        //Σύνδεση των πολυγώνων//
        ////////////////////////
        var wms1 = new OpenLayers.Layer.WMS("Ηχορύπανση",
"http://localhost:8080/geoserver/wms?", {
                    layers:
'diplwmatiki:noise1,diplwmatiki:noise2,diplwmatiki:noise3',
                    styles: '',
                    srs: 'EPSG:4326',
                    format: 'image/png',
                    tiled: 'true',
                    tilesOrigin : "143.60260815000004,-43.851764249999995",
transparent: true}, {'opacity': 0.35, 'isBaseLayer': false, 'wrapDateLine':
true});

       var wms = new OpenLayers.Layer.WMS("Ατμοσφαιρικοί Ρύποι",
"http://localhost:8080/geoserver/wms?", {
                    layers:
'diplwmatiki:P3_new,diplwmatiki:P2_new,diplwmatiki:P1_new',
```

```
                    styles: '',
                    srs: 'EPSG:4326',
                    format: 'image/png',
                    tiled: 'true',
                    tilesOrigin : "143.60260815000004,-43.851764249999995",
transparent: true}, {'opacity': 0.35, 'isBaseLayer': false, 'wrapDateLine':
true});

        ////////////////////////////
        //Φόρτωση των layers και//
        //προβολή στο openlayers//
        ////////////////////////////
        map.addLayers([openstreetmap, google_maps,
google_satellite,wfst_layer,kmlLayer,KT,wms,wms1]);
        var mapextent = new OpenLayers.Bounds(23.57500, 37.91600, 23.91750,
38.07500).transform(WGS84, map.getProjectionObject());
        map.zoomToExtent(mapextent);
        ////////////////////////////

        //Create a style object to be used by a style map object
        var vector_style = new OpenLayers.Style({
            'fillColor': '#669933',
            'fillOpacity': .8,
            'strokeColor': 'red',
            'strokeWidth': 3,
            'pointRadius': 8});

        //Create a style map object and set the 'default' intent to the style
object we just created
        var vector_style_map = new OpenLayers.StyleMap({'default':
vector_style});

        //Add the style map to the vector layer
        wfst_layer.styleMap = vector_style_map;
        if(!map.getCenter()){map.zoomToMaxExtent();}


        function modifyChanged(checked) {
            if(checked) {modifyControl.activate();}
            else {modifyControl.deactivate();}}

        // Modifies the 'mode' property.
        function changeMode() {
            var reshape = dijit.byId("reshape").get("checked");
            var resize = dijit.byId("resize").get("checked");
            var rotate = dijit.byId("rotate").get("checked");
            var drag = dijit.byId("drag").get("checked");
            var mode = null;
            if(reshape) {mode |= OpenLayers.Control.ModifyFeature.RESHAPE;}
            if(resize) {mode |= OpenLayers.Control.ModifyFeature.RESIZE;}
            if(rotate) {mode |= OpenLayers.Control.ModifyFeature.ROTATE;}
            if(drag) {mode |= OpenLayers.Control.ModifyFeature.DRAG;}
            modifyControl.deactivate();
            modifyControl.mode = mode;
            modifyControl.activate();}

        // Destroy and create a new control to set the 'geometryType' property.
```

```
        function changeFilter(value) {
            modifyControl.deactivate();
            map.removeControl(modifyControl);
            modifyControl.destroy();
            var geometryTypes = null;
            if(value=="POINT") {geometryTypes = ["OpenLayers.Geometry.Point"];}
            else if(value=="PATH") {geometryTypes =
["OpenLayers.Geometry.LineString"];}
            else if(value=="POLYGON") {geometryTypes =
["OpenLayers.Geometry.Polygon"];}
            modifyControl = new OpenLayers.Control.ModifyFeature(vectorLayer,
{geometryTypes: geometryTypes});
            map.addControl(modifyControl);
            modifyControl.activate();}

        wfst_layer.events.register('featureadded', this, lineAdded);
        var options = {hover: true, onSelect: domhsh};
        var select = new OpenLayers.Control.SelectFeature(wfst_layer, options);
        map.addControl(select);
        new OpenLayers.Control.LayerSwitcher(),
        select.activate();
        enhmeroshformat();

        // Create an ElevationService
        elevator = new google.maps.ElevationService();

        // Create a new chart in the elevation_chart DIV.
        chart = new
google.visualization.LineChart(document.getElementById('elevation_chart'));
    };

    var kmlLayer;

    function lineAdded(e) {
        var feature = e.feature;
        // υψόμετρα
        var nodes =
feature.geometry.clone().transform(WGS84_google_mercator,WGS84).getVertices();
        var locations = [];

        for (var i=0; i<nodes.length; i++) {
            var lon = nodes[i].x;
            var lat = nodes[i].y;
            locations.push(new google.maps.LatLng(lat, lon));}

        // console.log("Added " + locations.length + " locations");
        // Create a PathElevationRequest object using this array.
        // Ask for 256 samples along that path.
        // Initiate the path request.
        var pathRequest = {'path': locations,'samples': 256}

        // elevator.getElevationForLocations({locations: locations},
plotElevation);
        elevator.getElevationAlongPath(pathRequest, plotElevation);

        //εισαγωγή απόστασης από τον χρήστη
        var maxDistance =
```

```
parseFloat(document.getElementById('maxDistance').value);

        if (!maxDistance) {
            // αν ο χρήστης βάλει πολύ μεγάλη τιμή βάζουμε μια προκαθορισμένη
τιμή, π.χ. 5000:
            maxDistance = 5000;
            // και ενημερώνουμε το πεδίο, ώστε να το καταλάβει
            document.getElementById('maxDistance').value = maxDistance;}

        // αποσταση από σταθμό
        var distances = [];
        for(var i=0;i<kmlLayer.features.length;i++) {
            var station = kmlLayer.features[i];
            var distance = feature.geometry.distanceTo(station.geometry);
            // εδώ ασχολούμαστε μόνο με αυτά εντός της απόστασης maxDistance
            if (distance < maxDistance) {
                // εδω κρατάμε σε κάθε σταθμό την απόστασή του από το σημείο
                station.attributes['distanceFromLine'] = distance;
                // δεν χρειάζεται για αυτό που κάνουμε, αλλά μπορεί να θέλουμε
να τα δείξουμε π.χ. πάνω στον χάρτη
                // και εδώ βάζουμε τα (φιλτραρισμένα) distance, τα οποία τώρα
είναι
                // αντικείμενα με πεδία: station: το όνομα του σταθμού και
distance: την απόσταση
                distances.push({'station': station.attributes.name,'distance':
distance});}}

            if (distances.length > 0) {
                // ορίζουμε κατευθείαν την compare, απλά η διαφορά αρκεί
                // το distances αλλαζει το ίδιο, δεν χρειαζεται να ορισεις το
sorteddistances
                distances.sort(function(a,b) {
                // το a, b είναι τώρα αντικείμενα με πεδία: station και distance
                // μας νοιάζει να τα ταξινομήσουμε με βάση το πεδίο distance
                return a.distance - b.distance;});

        // αντι να το φτιαχνουμε κάθε φορά, χρησιμοποιούμε την ίδια θέση στο
HTML και
        var table = document.getElementById("distances");
        // την σβήνουμε κάθε φορά (το innerHTML είναι ένας τρόπος να γραφεις
απευθείας HTML σε ένα Element)
        table.innerHTML = "";
        // βάζουμε τίτλους στις στήλες (οι οποίες ας πούμε είναι ΔΥΟ)
        var header = table.insertRow(-1);
        header.insertCell(-1).innerHTML = 'Σταθμός';
        header.insertCell(-1).innerHTML = 'Απόσταση';

        // και οι γραμμές είναι όσες και τα distances!!!
        for (var i = 0; i < distances.length; i++) {
            var row = table.insertRow(-1);
            // το όνομα του σταθμού
            row.insertCell(-1).innerHTML = distances[i].station;
            // και η απόσταση
            row.insertCell(-1).innerHTML = distances[i].distance.toFixed(0) + '
μέτρα';}

        /*
```

```
        if (distances.length > 0) {
            var closestStation = distances[0].station;
            var minDistance = distances[0].distance;
            window.alert("Ο κοντινότερος σταθμός είναι ο " + closestStation + ",
σε απόσταση " + minDistance.toFixed(0) + " μέτρα");}
        */

        }
    }

    function plotElevation(results, status) {
        if (status == google.maps.ElevationStatus.OK) {
            var sampleDist =
google.maps.geometry.spherical.computeDistanceBetween(results[0].location,
results[1].location);
            var slopes = calculateSlopes(results);
            // Extract the data from which to populate the chart.
            // Because the samples are equidistant, the 'Sample'
            // column here does double duty as distance along the
            // X axis.
            var data = new google.visualization.DataTable();
                data.addColumn('number', 'Distance');
                data.addColumn('number', 'Elevation');
                data.addColumn('number', 'Slope');

        for (var i = 0; i < results.length; i++) {data.addRow([Math.round(i *
sampleDist), results[i].elevation, slopes[i]]);}
        // Draw the chart using the data within its DIV.
        document.getElementById('elevation_chart').style.display = 'block';
        chart.draw(data, {
            width: 800,
            height: 200,
            legend: 'none',
            titleY: 'Elevation (m)',
            title: 'Μέση κλίση: ' + calculateAverage(slopes).toFixed(2) + '%'});
        }
    }

    // elevationData: πίνακας από ζεύγη { location (google.maps.LatLng),
elevation (number) }

    ////////////////////////////
    //υπολογισμός αποστάσεων//
    ////////////////////////////
    function calculateSlopes(elevationData) {
        var distances = [], slopes = [];
        for (i=0; i < elevationData.length-1; i++) {
            distances[i] =
google.maps.geometry.spherical.computeDistanceBetween(elevationData[i].location,
elevationData[i+1].location);
            var heightDiff = Math.abs(elevationData[i+1].elevation-
elevationData[i].elevation);
            slopes[i] = heightDiff / distances[i] * 100;}
    //console.log(distances);
    //console.log(slopes);
    return slopes;}
```

```
    // values: Πίνακας από number
    function calculateAverage(values) {if (!values.length) {return result;}
        var result = 0;
        for (var i=0; i<values.length; i++) {result+=values[i];}
    return result/values.length;}



    function domhsh(feature) {
        var type = document.getElementById("formatType").value;
        //second argument for pretty printing (geojson only)
        var pretty = document.getElementById("prettyPrint").checked;
        var str = formats['out'][type].write(feature, pretty);
        //not a good idea in general, just for this demo
        //str = str.replace(/,/g, ', ');
        document.getElementById('output').value = str;}

    function apodomhsh() {
        var element = document.getElementById('text');
        var type = document.getElementById("formatType").value;
        var features = formats['in'][type].read(element.value);
        var bounds;
        if(features) {
            if(features.constructor != Array) {features = [features];}
            for(var i=0; i<features.length; ++i) {
                if (!bounds) {bounds = features[i].geometry.getBounds();}
                else {bounds.extend(features[i].geometry.getBounds());}}
            wfst_layer.addFeatures(features);
            map.zoomToExtent(bounds);
            var plural = (features.length > 1) ? 's' : '';
            element.value = features.length + ' feature' + plural + ' added';}
        else {element.value = 'Bad input ' + type;}
    }
    </script>
</head>

<body onload="init()">
    <div id="leftcol">
        <h1 id="title">Online ψηφιοποίηση</h1>
        <div id="tags"> </div>
        <p id="shortdesc">
            Αυτή η εφαρμογή διαβάζει αρχεία χαρτών απο διάφορους παρόχους. Σας
επιτρέπει να φορτώστε αρχεία και να τα προβάλλετε στον χάρτη ή να δημιουργήσετε
τα δικά σας. Για τις διαδρομές υπολογίζονται οι κλίσεις και οι αποστάσεις από
προκαθορισμένα σημεία αναφοράς τα οποία μπορείτε να τα εισάγετε επικολλώντας τα
παρακάτω).</p>
        <div id="map" style="width:1000px; height:600px;"></div>
        <div id="coordinates"></div>
        <div id="nodelist"></div>
        <div id="elevation_chart"></div>
        <div id="input">
            <p> Εισάγετε την απόσταση για επιθυμητό μέσο μαζικής μεταφοράς </p>
<input type="text" id="maxDistance" value="1000" />
            <p>Χρησιμοποιείστε το αναδυόμενο μενού για να επιλέξετε το προβολικό
σύστημα εισόδου-εξόδου.Νέα γραμμικά στοιχεία μπορούν να δημιουργηθούν
χρησιμοποιώντας τα εργαλεία ψηφιοποίησης του χάρτη ή κάνοντας επικόλληση αρχείων
παρακάτω.</p>
```

```
            <label for="formatType">Format</label>
                <select name="formatType" id="formatType">
                    <option value="geojson" selected="selected">GeoJSON</option>
                    <option value="atom">Atom</option>
                    <option value="kml">KML</option>
                    <option value="georss">GeoRSS</option>
                    <option value="gml2">GML (v2)</option>
                    <option value="gml3">GML (v3)</option>
                    <option value="wkt">Well-Known Text (WKT)</option>
                    <option value="gpx">GPX</option>
                </select>
                                         
                <label for="prettyPrint">Pretty print</label>
            <input id="prettyPrint" type="checkbox" name="prettyPrint" value="1"
/>
            <br>

            Input Projection: <select id="inproj" onchange='enhmeroshformat()'>
                                <option value="EPSG:4326"
selected="selected">EPSG:4326</option>
                                <option value="EPSG:900913">Spherical
Mercator</option>
                            </select> <br>

            Output Projection: <select id="outproj"
onchange='enhmeroshformat()'>
                                <option value="EPSG:4326"
selected="selected">EPSG:4326</option>
                                <option value="EPSG:900913">Spherical
Mercator</option>
                            </select> <br>

            <textarea rows="6" cols="80" id="text">Εδώ μπορείτε να επικολλήσετε
κείμενο </textarea size=40>
            <br>
            <input type="button" value="Προβολή στον χάρτη"
onclick="apodomhsh();" />
        </div>
    <div id="docs">     </div>
    </div>

    <div id="info">
        <p>Για να προσθέστε σημεία,γραμμές και πολύγωνα χρησιμοποιήστε τα
εργαλεία ψηφιοποίησης που υπάρχουν στον χάρτη.Μετά την δημιουργία τους μπορείτε
να τα επιλέξετε και να δείτε το αρχείο που δημιουργείται.Κάντε το επικόλληση στο
notepad με την αντίστοιχη κατάληξη και θα έχετε ένα πλήρως λειτουργικό
αρχείο.</p>
        <textarea rows="6" cols="80" id="output"size="40"> </textarea>
    </div>
    <table id="distances"></table>

    <div id="info">
        <p>Αφού σώσετε τη διαδρομή σας κάντε στο "ΥΠΟΛΟΓΙΣΕ ΤΟΜΕΣ" για να δείτε
τις τομές της διαδρομής με τα θεματικά επίπεδα<p>
    <div id="myDiv"><h2></h2></div>
    <button type="button" onclick="loadXMLDoc()">ΥΠΟΛΟΓΙΣΕ ΤΟΜΕΣ</button>
    </body></html>
```

- **Το PHP scritp που πραγματοποιεί την επικοινωνία με τον GeoServer**

```php
<?php

    // Open log file
    $logfh = fopen("GeoserverPHP.log", 'w') or die("can't open log file");
    // Initiate cURL session
    $service = "http://localhost:8080/geoserver/"; // replace with your URL
    $request = "wps/";// to add a new workspace
    $url = $service . $request;
    $ch = curl_init($url);
    // Optional settings for debugging
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true); //option to return string
    curl_setopt($ch, CURLOPT_VERBOSE, true);
    curl_setopt($ch, CURLOPT_STDERR, $logfh); // logs curl messages
    //Required POST request settings
    curl_setopt($ch, CURLOPT_POST, True);
    $passwordStr = "Gimca:ggaras1986"; // username:password
    curl_setopt($ch, CURLOPT_USERPWD, $passwordStr);

    //POST data
    curl_setopt($ch, CURLOPT_HTTPHEADER, array("Content-type: application/xml"));
    curl_setopt($ch, CURLOPT_POSTFIELDS, $xml = file_get_contents
("http://localhost/diplwmatiki/wps_p1_l1.xml"));
    $successCode = 201; //POST return code
    $buffer1 = 100000*curl_exec($ch); // Execute the curl request
    // Check for errors and process results
    $info = curl_getinfo($ch);
    if ($info['http_code'] != $successCode) {
        $msgStr = "# Unsuccessful cURL request to ";
        $msgStr .= $url." [". $info['http_code']. "]\n";
        fwrite($logfh, $msgStr);
    }else {
        $msgStr = "# Successful cURL request to ".$url."\n";
        fwrite($logfh, $msgStr); }

    fwrite($logfh, $buffer1."\n");

    //POST data
    curl_setopt($ch, CURLOPT_HTTPHEADER, array("Content-type: application/xml"));
    curl_setopt($ch, CURLOPT_POSTFIELDS, $xml = file_get_contents
("http://localhost/diplwmatiki/wps_p2_l1.xml"));
    $successCode = 201; //POST return code
    $buffer2 = 100000*curl_exec($ch); // Execute the curl request
    // Check for errors and process results
    $info = curl_getinfo($ch);
    if ($info['http_code'] != $successCode) {
        $msgStr = "# Unsuccessful cURL request to ";
        $msgStr .= $url." [". $info['http_code']. "]\n";
        fwrite($logfh, $msgStr);
    }else {
        $msgStr = "# Successful cURL request to ".$url."\n";
        fwrite($logfh, $msgStr); }

    fwrite($logfh, $buffer2."\n");
```

```
    //POST data
    curl_setopt($ch, CURLOPT_HTTPHEADER, array("Content-type: application/xml"));
    curl_setopt($ch, CURLOPT_POSTFIELDS, $xml = file_get_contents
("http://localhost/diplwmatiki/wps_p3_l1.xml"));
    $successCode = 201; //POST return code
    $buffer3 = 100000*curl_exec($ch); // Execute the curl request
    // Check for errors and process results
    $info = curl_getinfo($ch);
    if ($info['http_code'] != $successCode) {
        $msgStr = "# Unsuccessful cURL request to ";
        $msgStr .= $url." [". $info['http_code']. "]\n";
        fwrite($logfh, $msgStr);
    }else {
        $msgStr = "# Successful cURL request to ".$url."\n";
        fwrite($logfh, $msgStr); }

    fwrite($logfh, $buffer3."\n");

    //POST data
    curl_setopt($ch, CURLOPT_HTTPHEADER, array("Content-type: application/xml"));
    curl_setopt($ch, CURLOPT_POSTFIELDS, $xml = file_get_contents
("http://localhost/diplwmatiki/wps_noise1_l1.xml"));
    $successCode = 201; //POST return code
    $buffern1 = 100000*curl_exec($ch); // Execute the curl request
    // Check for errors and process results
    $info = curl_getinfo($ch);
    if ($info['http_code'] != $successCode) {
        $msgStr = "# Unsuccessful cURL request to ";
        $msgStr .= $url." [". $info['http_code']. "]\n";
        fwrite($logfh, $msgStr);
    }else {
        $msgStr = "# Successful cURL request to ".$url."\n";
        fwrite($logfh, $msgStr); }

    fwrite($logfh, $buffern1."\n");

    //POST data
    curl_setopt($ch, CURLOPT_HTTPHEADER, array("Content-type: application/xml"));
    curl_setopt($ch, CURLOPT_POSTFIELDS, $xml = file_get_contents
("http://localhost/diplwmatiki/wps_noise2_l1.xml"));
    $successCode = 201; //POST return code
    $buffern2 = 100000*curl_exec($ch); // Execute the curl request
    // Check for errors and process results
    $info = curl_getinfo($ch);
    if ($info['http_code'] != $successCode) {
        $msgStr = "# Unsuccessful cURL request to ";
        $msgStr .= $url." [". $info['http_code']. "]\n";
        fwrite($logfh, $msgStr);
    }else {
        $msgStr = "# Successful cURL request to ".$url."\n";
        fwrite($logfh, $msgStr); }

    fwrite($logfh, $buffern2."\n");

    //POST data
    curl_setopt($ch, CURLOPT_HTTPHEADER, array("Content-type: application/xml"));
    curl_setopt($ch, CURLOPT_POSTFIELDS, $xml = file_get_contents
```

```
("http://localhost/diplwmatiki/wps_noise3_l1.xml"));
    $successCode = 201; //POST return code
    $buffern3 = 100000*curl_exec($ch); // Execute the curl request
    // Check for errors and process results
    $info = curl_getinfo($ch);
    if ($info['http_code'] != $successCode) {
        $msgStr = "# Unsuccessful cURL request to ";
        $msgStr .= $url." [". $info['http_code']. "]\n";
        fwrite($logfh, $msgStr);
    }else {
        $msgStr = "# Successful cURL request to ".$url."\n";
        fwrite($logfh, $msgStr); }

    fwrite($logfh, $buffern3."\n");
    curl_close($ch); // free resources if curl handle will not be reused

    print "<div><p>Τομή της διαδρομής με το επίπεδο μόλυνση1: $buffer1 m</p><p>Τομή
της διαδρομής με το επίπεδο μόλυνση2: $buffer2 m</p><p>Τομή της διαδρομής με το
επίπεδο μόλυνση3: $buffer3 m</p><></><p>Τομή της διαδρομής με το επίπεδο
ηχορύπανση1: $buffern1 m</p><p>Τομή της διαδρομής με το επίπεδο ηχορύπανση2:
$buffern2 m</p><p>Τομή της διαδρομής με το επίπεδο ηχορύπανση3: $buffern3
m</p></div>";


    fclose($logfh);  // close logfile



?>
```

- **Τα XML που κάνουν τα WPS requests στον GeoServer**

Τομή της Διαδρομής με ηχορύπανση1

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>JTS:length</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>geom</ows:Identifier>
      <wps:Reference mimeType="text/xml; subtype=gml/3.1.1"
xlink:href="http://geoserver/wps" method="POST">
        <wps:Body>
          <wps:Execute version="1.0.0" service="WPS">
```

```
            <ows:Identifier>gs:CollectGeometries</ows:Identifier>
            <wps:DataInputs>
              <wps:Input>
                <ows:Identifier>features</ows:Identifier>
                <wps:Reference mimeType="text/xml"
xlink:href="http://geoserver/wps" method="POST">
                  <wps:Body>
                    <wps:Execute version="1.0.0" service="WPS">

<ows:Identifier>gs:IntersectionFeatureCollection</ows:Identifier>
                      <wps:DataInputs>
                        <wps:Input>
                          <ows:Identifier>first feature collection</ows:Identifier>
                          <wps:Reference mimeType="text/xml"
xlink:href="http://geoserver/wfs" method="POST">
                            <wps:Body>
                              <wfs:GetFeature service="WFS" version="1.0.0"
outputFormat="GML2" xmlns:diplwmatiki="http://localhost/">
                                <wfs:Query typeName="diplwmatiki:noise1"/>
                              </wfs:GetFeature>
                            </wps:Body>
                          </wps:Reference>
                        </wps:Input>
                        <wps:Input>
                          <ows:Identifier>second feature
collection</ows:Identifier>
                          <wps:Reference mimeType="text/xml"
xlink:href="http://geoserver/wfs" method="POST">
                            <wps:Body>
                              <wfs:GetFeature service="WFS" version="1.0.0"
outputFormat="GML2" xmlns:diplwmatiki="http://localhost/">
                                <wfs:Query typeName="diplwmatiki:wfst_test"/>
                              </wfs:GetFeature>
                            </wps:Body>
                          </wps:Reference>
                        </wps:Input>
                        <wps:Input>
                          <ows:Identifier>intersectionMode</ows:Identifier>
                          <wps:Data>
                            <wps:LiteralData>INTERSECTION</wps:LiteralData>
                          </wps:Data>
                        </wps:Input>
                      </wps:DataInputs>
                      <wps:ResponseForm>
                        <wps:RawDataOutput mimeType="text/xml; subtype=wfs-
collection/1.0">
                          <ows:Identifier>result</ows:Identifier>
                        </wps:RawDataOutput>
                      </wps:ResponseForm>
                    </wps:Execute>
                  </wps:Body>
                </wps:Reference>
              </wps:Input>
            </wps:DataInputs>
            <wps:ResponseForm>
              <wps:RawDataOutput mimeType="text/xml; subtype=gml/3.1.1">
                <ows:Identifier>result</ows:Identifier>
```

```
              </wps:RawDataOutput>
            </wps:ResponseForm>
          </wps:Execute>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput>
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>
```

Τομή της Διαδρομής με ηχορύπανση2

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>JTS:length</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>geom</ows:Identifier>
      <wps:Reference mimeType="text/xml; subtype=gml/3.1.1"
xlink:href="http://geoserver/wps" method="POST">
        <wps:Body>
          <wps:Execute version="1.0.0" service="WPS">
            <ows:Identifier>gs:CollectGeometries</ows:Identifier>
            <wps:DataInputs>
              <wps:Input>
                <ows:Identifier>features</ows:Identifier>
                <wps:Reference mimeType="text/xml"
xlink:href="http://geoserver/wps" method="POST">
                  <wps:Body>
                    <wps:Execute version="1.0.0" service="WPS">

<ows:Identifier>gs:IntersectionFeatureCollection</ows:Identifier>
                      <wps:DataInputs>
                        <wps:Input>
                          <ows:Identifier>first feature collection</ows:Identifier>
                          <wps:Reference mimeType="text/xml"
xlink:href="http://geoserver/wfs" method="POST">
                            <wps:Body>
                              <wfs:GetFeature service="WFS" version="1.0.0"
outputFormat="GML2" xmlns:diplwmatiki="http://localhost/">
```

```
                                <wfs:Query typeName="diplwmatiki:noise2"/>
                              </wfs:GetFeature>
                            </wps:Body>
                          </wps:Reference>
                        </wps:Input>
                        <wps:Input>
                          <ows:Identifier>second feature
collection</ows:Identifier>
                          <wps:Reference mimeType="text/xml"
xlink:href="http://geoserver/wfs" method="POST">
                            <wps:Body>
                              <wfs:GetFeature service="WFS" version="1.0.0"
outputFormat="GML2" xmlns:diplwmatiki="http://localhost/">
                                <wfs:Query typeName="diplwmatiki:wfst_test"/>
                              </wfs:GetFeature>
                            </wps:Body>
                          </wps:Reference>
                        </wps:Input>
                        <wps:Input>
                          <ows:Identifier>intersectionMode</ows:Identifier>
                          <wps:Data>
                            <wps:LiteralData>INTERSECTION</wps:LiteralData>
                          </wps:Data>
                        </wps:Input>
                      </wps:DataInputs>
                      <wps:ResponseForm>
                        <wps:RawDataOutput mimeType="text/xml; subtype=wfs-
collection/1.0">
                          <ows:Identifier>result</ows:Identifier>
                        </wps:RawDataOutput>
                      </wps:ResponseForm>
                    </wps:Execute>
                  </wps:Body>
                </wps:Reference>
              </wps:Input>
            </wps:DataInputs>
            <wps:ResponseForm>
              <wps:RawDataOutput mimeType="text/xml; subtype=gml/3.1.1">
                <ows:Identifier>result</ows:Identifier>
              </wps:RawDataOutput>
            </wps:ResponseForm>
          </wps:Execute>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput>
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>
```

Τομή της Διαδρομής με ηχορύπανση3

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>JTS:length</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>geom</ows:Identifier>
      <wps:Reference mimeType="text/xml; subtype=gml/3.1.1"
xlink:href="http://geoserver/wps" method="POST">
        <wps:Body>
          <wps:Execute version="1.0.0" service="WPS">
            <ows:Identifier>gs:CollectGeometries</ows:Identifier>
            <wps:DataInputs>
              <wps:Input>
                <ows:Identifier>features</ows:Identifier>
                <wps:Reference mimeType="text/xml"
xlink:href="http://geoserver/wps" method="POST">
                  <wps:Body>
                    <wps:Execute version="1.0.0" service="WPS">

<ows:Identifier>gs:IntersectionFeatureCollection</ows:Identifier>
                      <wps:DataInputs>
                        <wps:Input>
                          <ows:Identifier>first feature collection</ows:Identifier>
                          <wps:Reference mimeType="text/xml"
xlink:href="http://geoserver/wfs" method="POST">
                            <wps:Body>
                              <wfs:GetFeature service="WFS" version="1.0.0"
outputFormat="GML2" xmlns:diplwmatiki="http://localhost/">
                                <wfs:Query typeName="diplwmatiki:noise3"/>
                              </wfs:GetFeature>
                            </wps:Body>
                          </wps:Reference>
                        </wps:Input>
                        <wps:Input>
                          <ows:Identifier>second feature
collection</ows:Identifier>
                          <wps:Reference mimeType="text/xml"
xlink:href="http://geoserver/wfs" method="POST">
                            <wps:Body>
                              <wfs:GetFeature service="WFS" version="1.0.0"
outputFormat="GML2" xmlns:diplwmatiki="http://localhost/">
                                <wfs:Query typeName="diplwmatiki:wfst_test"/>
                              </wfs:GetFeature>
                            </wps:Body>
                          </wps:Reference>
                        </wps:Input>
                        <wps:Input>
```

```
                        <ows:Identifier>intersectionMode</ows:Identifier>
                        <wps:Data>
                          <wps:LiteralData>INTERSECTION</wps:LiteralData>
                        </wps:Data>
                      </wps:Input>
                    </wps:DataInputs>
                    <wps:ResponseForm>
                      <wps:RawDataOutput mimeType="text/xml; subtype=wfs-
collection/1.0">
                        <ows:Identifier>result</ows:Identifier>
                      </wps:RawDataOutput>
                    </wps:ResponseForm>
                  </wps:Execute>
                </wps:Body>
              </wps:Reference>
            </wps:Input>
          </wps:DataInputs>
          <wps:ResponseForm>
            <wps:RawDataOutput mimeType="text/xml; subtype=gml/3.1.1">
              <ows:Identifier>result</ows:Identifier>
            </wps:RawDataOutput>
          </wps:ResponseForm>
        </wps:Execute>
      </wps:Body>
    </wps:Reference>
  </wps:Input>
 </wps:DataInputs>
 <wps:ResponseForm>
   <wps:RawDataOutput>
     <ows:Identifier>result</ows:Identifier>
   </wps:RawDataOutput>
 </wps:ResponseForm>
</wps:Execute>
```

Τομή της Διαδρομής με ατμοσφαιρική ρύπανση1

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>JTS:length</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>geom</ows:Identifier>
      <wps:Reference mimeType="text/xml; subtype=gml/3.1.1"
xlink:href="http://geoserver/wps" method="POST">
        <wps:Body>
          <wps:Execute version="1.0.0" service="WPS">
```

```
            <ows:Identifier>gs:CollectGeometries</ows:Identifier>
            <wps:DataInputs>
              <wps:Input>
                <ows:Identifier>features</ows:Identifier>
                <wps:Reference mimeType="text/xml"
xlink:href="http://geoserver/wps" method="POST">
                  <wps:Body>
                    <wps:Execute version="1.0.0" service="WPS">

<ows:Identifier>gs:IntersectionFeatureCollection</ows:Identifier>
                      <wps:DataInputs>
                        <wps:Input>
                          <ows:Identifier>first feature collection</ows:Identifier>
                          <wps:Reference mimeType="text/xml"
xlink:href="http://geoserver/wfs" method="POST">
                            <wps:Body>
                              <wfs:GetFeature service="WFS" version="1.0.0"
outputFormat="GML2" xmlns:diplwmatiki="http://localhost/">
                                <wfs:Query typeName="diplwmatiki:P1_new"/>
                              </wfs:GetFeature>
                            </wps:Body>
                          </wps:Reference>
                        </wps:Input>
                        <wps:Input>
                          <ows:Identifier>second feature
collection</ows:Identifier>
                          <wps:Reference mimeType="text/xml"
xlink:href="http://geoserver/wfs" method="POST">
                            <wps:Body>
                              <wfs:GetFeature service="WFS" version="1.0.0"
outputFormat="GML2" xmlns:diplwmatiki="http://localhost/">
                                <wfs:Query typeName="diplwmatiki:wfst_test"/>
                              </wfs:GetFeature>
                            </wps:Body>
                          </wps:Reference>
                        </wps:Input>
                        <wps:Input>
                          <ows:Identifier>intersectionMode</ows:Identifier>
                          <wps:Data>
                            <wps:LiteralData>INTERSECTION</wps:LiteralData>
                          </wps:Data>
                        </wps:Input>
                      </wps:DataInputs>
                      <wps:ResponseForm>
                        <wps:RawDataOutput mimeType="text/xml; subtype=wfs-
collection/1.0">
                          <ows:Identifier>result</ows:Identifier>
                        </wps:RawDataOutput>
                      </wps:ResponseForm>
                    </wps:Execute>
                  </wps:Body>
                </wps:Reference>
              </wps:Input>
            </wps:DataInputs>
            <wps:ResponseForm>
              <wps:RawDataOutput mimeType="text/xml; subtype=gml/3.1.1">
                <ows:Identifier>result</ows:Identifier>
```

```
              </wps:RawDataOutput>
            </wps:ResponseForm>
          </wps:Execute>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput>
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>
```

Τομή της Διαδρομής με ατμοσφαιρική ρύπανση2

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>JTS:length</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>geom</ows:Identifier>
      <wps:Reference mimeType="text/xml; subtype=gml/3.1.1"
xlink:href="http://geoserver/wps" method="POST">
        <wps:Body>
          <wps:Execute version="1.0.0" service="WPS">
            <ows:Identifier>gs:CollectGeometries</ows:Identifier>
            <wps:DataInputs>
              <wps:Input>
                <ows:Identifier>features</ows:Identifier>
                <wps:Reference mimeType="text/xml"
xlink:href="http://geoserver/wps" method="POST">
                  <wps:Body>
                    <wps:Execute version="1.0.0" service="WPS">

<ows:Identifier>gs:IntersectionFeatureCollection</ows:Identifier>
                      <wps:DataInputs>
                        <wps:Input>
                          <ows:Identifier>first feature collection</ows:Identifier>
                          <wps:Reference mimeType="text/xml"
xlink:href="http://geoserver/wfs" method="POST">
                            <wps:Body>
                              <wfs:GetFeature service="WFS" version="1.0.0"
outputFormat="GML2" xmlns:diplwmatiki="http://localhost/">
                                <wfs:Query typeName="diplwmatiki:P2_new"/>
                              </wfs:GetFeature>
```

```
                            </wps:Body>
                          </wps:Reference>
                        </wps:Input>
                        <wps:Input>
                          <ows:Identifier>second feature
collection</ows:Identifier>
                          <wps:Reference mimeType="text/xml"
xlink:href="http://geoserver/wfs" method="POST">
                            <wps:Body>
                              <wfs:GetFeature service="WFS" version="1.0.0"
outputFormat="GML2" xmlns:diplwmatiki="http://localhost/">
                                <wfs:Query typeName="diplwmatiki:wfst_test"/>
                              </wfs:GetFeature>
                            </wps:Body>
                          </wps:Reference>
                        </wps:Input>
                        <wps:Input>
                          <ows:Identifier>intersectionMode</ows:Identifier>
                          <wps:Data>
                            <wps:LiteralData>INTERSECTION</wps:LiteralData>
                          </wps:Data>
                        </wps:Input>
                      </wps:DataInputs>
                      <wps:ResponseForm>
                        <wps:RawDataOutput mimeType="text/xml; subtype=wfs-
collection/1.0">
                          <ows:Identifier>result</ows:Identifier>
                        </wps:RawDataOutput>
                      </wps:ResponseForm>
                    </wps:Execute>
                  </wps:Body>
                </wps:Reference>
              </wps:Input>
            </wps:DataInputs>
            <wps:ResponseForm>
              <wps:RawDataOutput mimeType="text/xml; subtype=gml/3.1.1">
                <ows:Identifier>result</ows:Identifier>
              </wps:RawDataOutput>
            </wps:ResponseForm>
          </wps:Execute>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput>
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>
```

Τομή της Διαδρομής με ατμοσφαιρική ρύπανση3

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>JTS:length</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>geom</ows:Identifier>
      <wps:Reference mimeType="text/xml; subtype=gml/3.1.1"
xlink:href="http://geoserver/wps" method="POST">
        <wps:Body>
          <wps:Execute version="1.0.0" service="WPS">
            <ows:Identifier>gs:CollectGeometries</ows:Identifier>
            <wps:DataInputs>
              <wps:Input>
                <ows:Identifier>features</ows:Identifier>
                <wps:Reference mimeType="text/xml"
xlink:href="http://geoserver/wps" method="POST">
                  <wps:Body>
                    <wps:Execute version="1.0.0" service="WPS">

<ows:Identifier>gs:IntersectionFeatureCollection</ows:Identifier>
                      <wps:DataInputs>
                        <wps:Input>
                          <ows:Identifier>first feature collection</ows:Identifier>
                          <wps:Reference mimeType="text/xml"
xlink:href="http://geoserver/wfs" method="POST">
                            <wps:Body>
                              <wfs:GetFeature service="WFS" version="1.0.0"
outputFormat="GML2" xmlns:diplwmatiki="http://localhost/">
                                <wfs:Query typeName="diplwmatiki:P3_new"/>
                              </wfs:GetFeature>
                            </wps:Body>
                          </wps:Reference>
                        </wps:Input>
                        <wps:Input>
                          <ows:Identifier>second feature
collection</ows:Identifier>
                          <wps:Reference mimeType="text/xml"
xlink:href="http://geoserver/wfs" method="POST">
                            <wps:Body>
                              <wfs:GetFeature service="WFS" version="1.0.0"
outputFormat="GML2" xmlns:diplwmatiki="http://localhost/">
                                <wfs:Query typeName="diplwmatiki:wfst_test"/>
                              </wfs:GetFeature>
                            </wps:Body>
                          </wps:Reference>
                        </wps:Input>
                        <wps:Input>
```

```
                        <ows:Identifier>intersectionMode</ows:Identifier>
                        <wps:Data>
                          <wps:LiteralData>INTERSECTION</wps:LiteralData>
                        </wps:Data>
                      </wps:Input>
                    </wps:DataInputs>
                    <wps:ResponseForm>
                      <wps:RawDataOutput mimeType="text/xml; subtype=wfs-
collection/1.0">
                        <ows:Identifier>result</ows:Identifier>
                      </wps:RawDataOutput>
                    </wps:ResponseForm>
                  </wps:Execute>
                </wps:Body>
              </wps:Reference>
            </wps:Input>
          </wps:DataInputs>
          <wps:ResponseForm>
            <wps:RawDataOutput mimeType="text/xml; subtype=gml/3.1.1">
              <ows:Identifier>result</ows:Identifier>
            </wps:RawDataOutput>
          </wps:ResponseForm>
        </wps:Execute>
      </wps:Body>
    </wps:Reference>
  </wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput>
    <ows:Identifier>result</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
```