



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

**Εφαρμογές Πλοήγησης για Φορητές Συσκευές
με χρήση Web Services**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΓΕΩΡΓΙΟΥ Σ. ΣΤΑΥΡΟΥΛΑΚΗ

Επιβλέπων : Ιωάννης Βασιλείου
Καθηγητής Ε.Μ.Π.

Αθήνα, Νοέμβριος 2006

Η σελίδα αυτή είναι σκόπιμα λευκή.



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Εφαρμογές Πλοήγησης για Φορητές Συσκευές με χρήση Web Services

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΓΕΩΡΓΙΟΥ Σ. ΣΤΑΥΡΟΥΛΑΚΗ

Επιβλέπων : Ιωάννης Βασιλείου
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 3^η Νοεμβρίου 2006.

.....
Ιωάννης Βασιλείου
Καθηγητής Ε.Μ.Π.

.....
Τιμολέων Σελλής
Καθηγητής Ε.Μ.Π.

.....
Νεκτάριος Κοζύρης
Επικουρος Καθηγητής Ε.Μ.Π.

Αθήνα, Νοέμβριος 2006

.....
ΓΕΩΡΓΙΟΣ Σ. ΣΤΑΥΡΟΥΛΑΚΗΣ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Γεώργιος Σ. Σταυρουλάκης, 2006

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Η σελίδα αυτή είναι σκόπιμα λευκή.

Πρόλογος

Ολοκληρώνοντας τη διπλωματική αυτή εργασία, θα ήθελα να ευχαριστήσω όλους όσους μου στάθηκαν κατά τη διάρκεια των σπουδών μου. Ιδιαίτερα θα ήθελα να ευχαριστήσω τον καθηγητή κ. Ιωάννη Βασιλείου, επιβλέποντα την εργασία, για τη συνεχή υποστήριξη την οποία παρείχε.

Επιπλέον, θα ήθελα να ευχαριστήσω θερμά τον υποψήφιο διδάκτορα Σπύρο Αθανασίου, για την καθοδήγηση και τη στήριξη την οποία μου πρόσφερε καθ' όλη τη διάρκεια της εκπόνησης της εργασίας. Επίσης, την Βασιλική Μπεζάτη με την οποία συνεργαστήκαμε άψογα κατά την αρχική θεωρητική μελέτη της εργασίας.

Επιπλέον, νιώθω την ανάγκη να ευχαριστήσω την οικογένειά μου – τους γονείς και τα αδέρφια μου – για την στήριξη που μου παρείχαν όλα αυτά τα χρόνια.

Αθήνα, 16 Οκτωβρίου 2006

Γεώργιος Σταυρουλάκης

Περίληψη

Σκοπός αυτής της διπλωματικής εργασίας είναι ο σχεδιασμός και η ανάπτυξη εφαρμογής πλοήγησης για φορητές συσκευές με χρήση διαδικτυακών υπηρεσιών. Η χρήση διαδικτυακών υπηρεσιών αποτελεί εναλλακτικό τρόπο στην ανάπτυξη εφαρμογών πλοήγησης. Δεν θα είναι πλέον ανάγκη να έχουν οι φορητές συσκευές αποθηκευμένους τους χάρτες των περιοχών που ενδιαφέρουν το χρήστη ούτε και οποιαδήποτε άλλη γεωγραφική πληροφορία (σημεία ενδιαφέροντος κτλ). Οποιαδήποτε τέτοια πληροφορία χρειάζεται η φορητή συσκευή θα τη λαμβάνει καταναλώνοντας τις διαδικτυακές υπηρεσίες που θα αναπτυχθούν.

Για την υλοποίηση των διαδικτυακών υπηρεσιών γίνεται μελέτη και αποτίμηση των υφιστάμενων προτύπων για τη διάθεση χωρικών πληροφοριών μέσω του διαδικτύου που δίνονται από οργανισμούς προτυποποίησης όπως ο Open Geospatial Consortium (OGC) και ο International Organization for Standardization (ISO). Επιπλέον γίνεται μελέτη και αποτίμηση εμπορικά διαθέσιμων διαδικτυακών υπηρεσιών για τη διάθεση χωρικών πληροφοριών. Τέλος, μελετήθηκαν εναλλακτικές μέθοδοι διάθεσης γεωγραφικής πληροφορίας (vector, raster, υβριδικές μέθοδοι, κτλ).

Για την υλοποίηση της εφαρμογής client για την πλοήγηση η οποία εγκαθίσταται σε φορητές συσκευές, έγινε μελέτη αντίστοιχων εφαρμογών πλοήγησης. Μελετήθηκαν επίσης μέθοδοι για την πιστοποίηση των χρηστών του συστήματος, διάφορα προφίλ χρηστών (οδηγός, πεζός κτλ) καθώς και μοντέλο χρέωσης των υπηρεσιών που αναπτύχθηκαν. Κατά την υλοποίηση της εφαρμογής λήφθηκαν υπόψη οι περιορισμένοι πόροι μιας φορητής συσκευής (επεξεργαστής, μνήμη, μέγεθος οθόνης, ταχύτητα σύνδεσης με το Internet). Ο προσδιορισμός της θέσης του χρήστη γίνεται μέσω σύνδεσης με Bluetooth GPS Receiver είτε εναλλακτικά μέσω του server.

Η γλώσσα που χρησιμοποιήθηκε είναι η Java η οποία προσφέρει ανεξαρτησία πλατφόρμας και λειτουργικού συστήματος. Επίσης οι διαδικτυακές υπηρεσίες αναπτύχθηκαν ως SOAP/XML web services γεγονός που προσφέρει στο σύστημα διαλειτουργικότητα με άλλα πληροφοριακά συστήματα.

Λέξεις Κλειδιά: Διαδικτυακές υπηρεσίες, διαλειτουργικότητα, GPS, OGC, OpenLS, XLS, LBS, ISO Metadata Model, SOAP, XML, Java, J2ME, εφαρμογές πλοήγησης, GIS, γεωγραφικά δεδομένα, χωρική πληροφορία.

Η σελίδα αυτή είναι σκόπιμα λευκή.

Abstract

The purpose of this diploma thesis is the design and development of a navigation system for mobile devices using web services. The use of web services composes an alternative way in developing navigation applications. Mobile devices will not need any more to save maps or other geographical information (Points Of Interest etc). Any information like this that the device needs will be received by consuming the web services that we will be developed.

In order to develop the web services, we studied and evaluated standards for geo-spatial web services provided by organizations such as Open Geospatial Consortium (OGC) and International Organization for Standardization (ISO). Furthermore, we evaluated today's commercial geo-spatial web services. Finally, we studied several alternative ways of providing geographical data (vector, raster, hybrid methods etc).

In order to complete the development of the client application for navigation that runs in mobile devices, we searched and studied the features of today's commercial navigation applications. We also studied methods for authentication of the users of the system, several user profiles (driver, pedestrian, traveler, climber, etc) and charging models for the deployed services. During the development of the application we took into consideration the limited resources of a mobile device (CPU, memory, screen size, Internet connection speed). The location of the end user is found either with the use of a Bluetooth GPS Receiver or alternatively through Gateway Service from the server.

The programming language that was used is Java that makes our system platform independent and operating system independent. Moreover, the web services were developed as SOAP/XML web services which offer interoperability to the system.

Keywords: Web services, interoperability, GPS, OGC, OpenLS, XLS, LBS, ISO Metadata Model, SOAP, XML, Java, J2ME, navigation applications, GIS, geographical data, spatial data

Η σελίδα αυτή είναι σκόπιμα λευκή.

Πίνακας περιεχομένων

1	Εισαγωγή.....	1
1.1	Αντικείμενο της διπλωματικής	1
1.2	Οργάνωση του τόμου.....	3
2	Web services για διάθεση χωρικής πληροφορίας	6
2.1	Διαδικτυακές υπηρεσίες (Web Services).....	6
2.1.1	<i>XML</i>	6
2.1.2	<i>Εισαγωγή στα web services</i>	8
2.1.3	<i>Αρχιτεκτονική των web services</i>	9
2.1.4	<i>Τα πρότυπα των web services</i>	10
2.1.5	<i>Java API for XML-based Remote Procedure Call (JAX-RPC)</i>	15
2.2	Michelin viaMichelin.....	16
2.2.1	<i>Λίστα των υπηρεσιών που προσφέρονται:</i>	17
2.3	Microsoft Mappoint	20
2.3.1	<i>Γενικά</i>	20
2.3.2	<i>Λειτουργικότητα</i>	20
2.3.3	<i>Κυριότερες Υπηρεσίες</i>	23
2.4	Google Maps API	25
3	Εφαρμογές πλοήγησης για φορητές συσκευές.....	29
3.1	Συστήματα πλοήγησης GPS για Pocket PC/Palm	29
3.1.1	<i>CoPilot Live Pocket PC</i>	29
3.1.2	<i>Microsoft Pocket Streets</i>	31
3.1.3	<i>Destinator</i>	32
3.1.4	<i>TomTom Navigator</i>	33
3.2	Συστήματα πλοήγησης GPS για κινητά τηλέφωνα (Smartphone).....	35
3.2.1	<i>CoPilot Live Smartphone</i>	35
3.2.2	<i>Microsoft Pocket Streets</i>	36
3.2.3	<i>Destinator SP - Navigation for Smartphones</i>	36

4	Στόχος	40
4.1	Εφαρμογή.....	41
4.1.1	Πρότυπα	41
4.1.2	Server Side.....	41
4.1.3	Client Side.....	48
4.2	Χρήστες	49
4.2.1	Πληροφοριακά συστήματα.....	49
4.2.2	Τελικοί χρήστες (End Users)	49
4.3	Εμπορική Εκμετάλλευση	51
5	Ανάλυση και σχεδίαση.....	55
5.1	Εισαγωγή	55
5.1.1	UML (Unified Modeling Language).....	55
5.1.2	RUP (Rational Unified Process).....	56
5.1.3	YASM (Yet Another Software development Methodology)	56
5.2	Έγγραφο Προδιαγραφών Απαιτήσεων από το Λογισμικό.....	57
5.2.1	Εισαγωγή.....	57
5.2.2	Γενική περιγραφή	59
5.2.3	Απαιτήσεις.....	60
5.2.4	Ανάλυση περιπτώσεων χρήσης	62
5.3	Έγγραφο περιγραφής αρχιτεκτονικού σχεδίου	78
5.3.1	Εισαγωγή.....	78
5.3.2	Αρχιτεκτονικές αποφάσεις.....	80
5.3.3	Αρχιτεκτονικές Όψεις	82
5.4	Έγγραφο περιγραφής του λεπτομερούς σχεδίου.....	96
5.4.1	Εισαγωγή.....	96
5.4.2	Σχεδιαστικές αποφάσεις	97
5.4.3	Σχεδιαστικά μορφήματα	99
5.4.4	Σχεδιαστικές όψεις.....	99
6	Υλοποίηση	115
6.1	Πλατφόρμες και προγραμματιστικά εργαλεία	115
6.1.1	Borland JBuilder X.....	115

6.1.2	<i>Altova XML Spy 5 Enterprise Edition</i>	115
6.1.3	<i>Apache Tomcat</i>	116
6.1.4	<i>Java (J2SE)</i>	116
6.1.5	<i>Java (J2ME)</i>	117
6.1.6	<i>kXML, kSOAP</i>	121
6.1.7	<i>Πρωτόκολλο NMEA 0183</i>	121
6.2	Λεπτομέρειες υλοποίησης.....	123
6.2.1	<i>mobiNav Web Services Server</i>	123
6.2.2	<i>mobiNav Client</i>	126
6.3	Οδηγός Εγκατάστασης.....	128
6.3.1	<i>Server Side</i>	128
6.3.2	<i>Client Side</i>	128
6.4	Εγχειρίδιο Χρήσης	129
7	Επίλογος	136
7.1	Σύνοψη και συμπεράσματα.....	136
7.2	Μελλοντικές επεκτάσεις	138
8	Βιβλιογραφία	141
9	Παράρτημα	146
9.1	Πρότυπα για διάθεση χωρικής πληροφορίας	146
9.1.1	<i>OGC – Open Geospatial Consortium</i>	146
9.1.2	<i>ISO Metadata Model 19115:2003</i>	155

1

Εισαγωγή

1.1 Αντικείμενο της διπλωματικής

Τα διάφορα συστήματα πλοήγησης έχουν κατασκευαστεί με βασικό σκοπό να βοηθήσουν το χρήστη τους να φθάσει στον επιθυμητό προορισμό του. Τα συστήματα αυτά συνδυάζουν πολλές τεχνολογίες και διάφορα πρωτόκολλα για να παρέχουν υπηρεσίες, όπως εύρεση σημείων ενδιαφέροντος, εύρεση διευθύνσεων, παροχή οδηγιών για το πώς να πάει κανείς σε κάποιο προορισμό, real time πλοήγηση, κτλ. Παραδοσιακά, τα συστήματα πλοήγησης διατίθενται με έναν από τους ακόλουθους τρόπους:

- Ως μέρος ενός εξειδικευμένου υλικού (ενσωματωμένα συστήματα) όπως είναι για παράδειγμα διάφορες φορητές συσκευές με ενσωματωμένο δέκτη GPS. Οι συσκευές αυτές είναι κατασκευασμένες αποκλειστικά και μόνο για να παρέχουν υπηρεσίες πλοήγησης, ενώ έχουν αποθηκευμένους όλους τους χάρτες των περιοχών οι οποίες ενδιαφέρουν το χρήστη, τα διάφορα σημεία ενδιαφέροντος καθώς και οποιαδήποτε άλλη γεωγραφική πληροφορία.
- Ως ανεξάρτητο λογισμικό, διαφορετικές εκδόσεις του οποίου μπορούν να εγκατασταθούν σε μια πλειάδα συσκευών όπως PC, laptop, PDA, κινητά τηλέφωνα. Στην περίπτωση αυτή, ο προσδιορισμός της θέσης του χρήστη γίνεται από εξωτερικό δέκτη GPS ο οποίος επικοινωνεί ενσύρματα ή ασύρματα με τη φορητή συσκευή. Επιπλέον, το λογισμικό ενσωματώνει τους χάρτες των περιοχών οι οποίες

ενδιαφέρουν το χρήστη, τα διάφορα σημεία ενδιαφέροντος καθώς και οποιαδήποτε άλλη γεωγραφική πληροφορία.

Σκοπός αυτής της διπλωματικής εργασίας είναι η υλοποίηση ενός εναλλακτικού μοντέλου ανάπτυξης και διάθεσης γεωγραφικών πληροφοριών για εφαρμογές πλοήγησης σε φορητές συσκευές. Συγκεκριμένα, θα αναπτυχθούν δύο ανεξάρτητες εφαρμογές. Από τη μία, θα αναπτυχθούν διαδικτυακές υπηρεσίες οι οποίες θα είναι εγκατεστημένες σε έναν server και θα παρέχουν γεωγραφική πληροφορία στους διάφορους clients. Από την άλλη, θα υλοποιηθεί πρότυπη εφαρμογή client η οποία θα εγκαθίσταται σε φορητές συσκευές και θα καταναλώνει τις παραπάνω διαδικτυακές υπηρεσίες με σκοπό να παρέχει υπηρεσίες πλοήγησης στον τελικό χρήστη.

Από πλευράς server, το σύστημα το οποίο θα κατασκευαστεί θα πρέπει να υλοποιεί όλες τις βασικές υπηρεσίες διάθεσης γεωγραφικής πληροφορίας όπως για παράδειγμα υπηρεσίες υπολογισμού διαδρομής, γεωκωδικοποίησης, εύρεση σημείων ενδιαφέροντος, κτλ. Όλες οι γεωγραφικές πληροφορίες, όπως οι χάρτες για κάποια περιοχή, τα σημεία ενδιαφέροντος κτλ, θα είναι αποθηκευμένες είτε στον ίδιο υπολογιστή όπου θα τρέχουν οι διαδικτυακές υπηρεσίες είτε σε έναν άλλο υπολογιστή που θα παίζει ρόλο GIS server. Για την υλοποίηση των διαδικτυακών υπηρεσιών θα βασιστούμε σε ανοιχτά πρότυπα τα οποία δίνονται από ανεξάρτητους οργανισμούς προτυποποίησης όπως είναι ο Open Geospatial Consortium (OGC) και ο International Organization for Standardization (ISO). Επιπλέον, για να είναι εφικτή η διασύνδεση του server με άλλα πληροφοριακά συστήματα (διαλειτουργικότητα) οι υπηρεσίες θα αναπτυχθούν ως SOAP/XML web services.

Από πλευράς client, η εφαρμογή που θα αναπτυχθεί θα πρέπει να είναι σε θέση να τρέχει σε πολλές φορητές συσκευές και για την ακρίβεια να τρέχει στις περισσότερες Java-enabled συσκευές, από κινητά τηλέφωνα ως PDAs. Για το λόγο αυτό θα χρησιμοποιηθεί η γλώσσα προγραμματισμού Java 2 Micro Edition (J2ME) η οποία παρέχει ανεξαρτησία πλατφόρμας και λειτουργικού συστήματος. Η εφαρμογή client θα παρέχει υπηρεσίες πλοήγησης στο χρήστη.

Σημαντικό χαρακτηριστικό της είναι ότι θα αντλεί το σύνολο της πληροφορίας που απαιτείται μέσω των παραπάνω διαδικτυακών υπηρεσιών, συνεπώς θα πρέπει να μπορεί να καταναλώνει τα παραπάνω SOAP/XML web services. Έτσι, δε χρειάζεται να έχει αποθηκευμένους χάρτες και λοιπές πληροφορίες διότι θα μπορεί να τις λαμβάνει από το server. Αυτό σημαίνει αυτόματα ότι σε περίπτωση ανανέωσης του περιεχομένου των χαρτών ή άλλων γεωγραφικών

πληροφοριών, ο χρήστης δε χρειάζεται να προμηθευτεί νέο λογισμικό ή πακέτο αναβάθμισης του λογισμικού του διότι οι όποιες αλλαγές θα γίνουν μόνο στην πλευρά του server. Επίσης, με τον τρόπο αυτό μειώνουμε αισθητά το μέγεθος της εφαρμογής η οποία πλέον μπορεί να εγκατασταθεί ακόμη και σε κινητά τηλέφωνα με περιορισμένη μνήμη.

Γενικά πάντως, η εφαρμογή θα πρέπει να λαμβάνει υπόψη τους περιορισμένους πόρους μιας φορητής συσκευής (επεξεργαστής, μνήμη, μέγεθος οθόνης, ταχύτητα σύνδεσης με το Internet), καθώς και το συγκεκριμένο περιβάλλον του χρήστη της εφαρμογής (αυτοκίνητο, πεζός, ταξιδιώτης, ορειβάτης, κτλ). Τέλος, ο προσδιορισμός της θέσης του χρήστη θα γίνεται μέσω σύνδεσης με εξωτερική συσκευή GPS αλλά θα πρέπει να υπάρχει επίσης η δυνατότητα εναλλακτικού προσδιορισμού της θέσης του χρήστη με τη βοήθεια του server μέσω του δικτύου κινητής τηλεφωνίας.

Για την υλοποίηση του συστήματος που περιγράψαμε συνοπτικά παραπάνω, θα προηγηθεί θεωρητική μελέτη η οποία θα περιλαμβάνει τα ακόλουθα θέματα:

- Μελέτη και αποτίμηση των υφιστάμενων προτύπων για τη διάθεση χωρικών πληροφοριών μέσω του διαδικτύου (OGC Web Map Services). Μελέτη και αποτίμηση εμπορικά διαθέσιμων διαδικτυακών υπηρεσιών για τη διάθεση χωρικών πληροφοριών (Michelin viaMichelin, Microsoft MapPoint, Google Maps).
- Αποτίμηση εναλλακτικών μεθόδων διάθεσης γεωγραφικής πληροφορίας κυρίως όσον αφορά το format των χαρτών (vector, raster, υβριδικές μέθοδοι, κτλ).
- Αποτίμηση εναλλακτικών μεθόδων πιστοποίησης χρηστών για τη διάθεση των υπηρεσιών.
- Σχεδιασμός και υλοποίηση μοντέλου χρέωσης (charging model) για τη χρήση των υπηρεσιών που θα αναπτυχθούν.

1.2 Οργάνωση του τόμου

Ο τόμος της διπλωματικής εργασίας αποτελείται από 9 κεφάλαια. Στα κεφάλαια αυτά γίνεται αναλυτική περιγραφή της διαδικασίας η οποία ακολουθήθηκε κατά την εκπόνηση της εργασίας. Συγκεκριμένα, περιγράφονται έννοιες σχετικές με το θέμα, όπως για παράδειγμα τα web services. Επιπλέον, φαίνεται η θεωρητική μελέτη η οποία έγινε, παρουσιάζονται τα συμπεράσματα στα οποία φθάσαμε και περιγράφεται ο τρόπος υλοποίησης του συστήματος.

Πιο αναλυτικά, το 1^ο κεφάλαιο είναι εισαγωγικό και περιγράφει σε γενικές γραμμές το αντικείμενο της εργασίας.

Το 2^ο κεφάλαιο αρχικά περιγράφει συνοπτικά την τεχνολογία των web services καθώς και όλες τις σχετικές με αυτά τεχνολογίες. Στη συνέχεια γίνεται μελέτη ορισμένων από τα εμπορικά διαθέσιμα web services για τη διάθεση χωρικής πληροφορίας που υπάρχουν σήμερα. Τα web services τα οποία μελετάμε εδώ είναι το viaMichelin της Michelin, το MapPoint της Microsoft και τέλος το Maps της Google.

Το 3^ο κεφάλαιο περιέχει τη μελέτη ορισμένων δημοφιλών εφαρμογών πλοήγησης που υπάρχουν σήμερα για φορητές συσκευές. Η μελέτη αφορά εφαρμογές για Pocket PC και για Smartphones. Συγκεκριμένα, περιγράφονται τα κυριότερα χαρακτηριστικά εφαρμογών όπως το CoPilot, το PocketStreets, το Destinator, και το TomTom.

Στο 4^ο κεφάλαιο γίνεται ουσιαστικά η αποτίμηση όλων των αναλύσεων των προηγούμενων κεφαλαίων καθώς και της ανάλυσης που βρίσκεται στο παράρτημα, ώστε να βγουν χρήσιμα συμπεράσματα για το πώς ακριβώς θα έπρεπε να είναι η εφαρμογή την οποία θα κατασκευάσουμε, τι χαρακτηριστικά θα πρέπει, δηλαδή, να έχει. Επιπλέον, γίνεται ανάλυση σχετικά με τις κατηγορίες χρηστών οι οποίοι θα χρησιμοποιούν το σύστημα έτσι ώστε να δημιουργήσουμε διαφορετικό προφίλ για κάθε κατηγορία χρήστη. Τέλος, γίνεται αναφορά για πιθανά μοντέλα χρέωσης για τις υπηρεσίες που θα αναπτυχθούν.

Το 5^ο κεφάλαιο αναφέρεται στη ανάλυση και το σχεδιασμό του συστήματος. Έχοντας πλέον μελετήσει θεωρητικά το θέμα, μένει να γίνει σχεδιασμός του συστήματος, να προσδιοριστούν, δηλαδή, οι απαιτήσεις, να δοθεί η αρχιτεκτονική και να γίνει λεπτομερής σχεδίαση του συστήματος. Το κεφάλαιο αυτό, λοιπόν, αρχικά εισάγει τον αναγνώστη στις μεθοδολογίες ανάπτυξης λογισμικού που χρησιμοποιήθηκαν και στη συνέχεια παρατίθενται τα έγγραφα της τεχνολογίας λογισμικού. Συγκεκριμένα περιλαμβάνεται το «*Έγγραφο Προδιαγραφών των απαιτήσεων από το Λογισμικό Σύστημα*», το «*Έγγραφο Περιγραφής της Αρχιτεκτονικής του Συστήματος*» και το «*Έγγραφο Περιγραφής της Λεπτομερούς Σχεδίασης του Συστήματος*».

Το 6^ο κεφάλαιο παρουσιάζει την υλοποίηση της εφαρμογής. Αρχικά, περιγράφονται συνοπτικά οι πλατφόρμες και τα προγραμματιστικά εργαλεία τα οποία χρησιμοποιήθηκαν ενώ στη συνέχεια φαίνονται οι λεπτομέρειες της υλοποίησης με την περιγραφή της δομής του

κώδικα μέσω των κυριότερων πακέτων (packages). Τέλος, παρέχονται οδηγίες για την εγκατάσταση του συστήματος και περιγράφεται ένα σενάριο χρήσης του.

Το 7^ο κεφάλαιο αποτελεί μια σύνοψη της παρούσας διπλωματικής εργασίας και παρατίθενται συμπεράσματα που προέκυψαν. Επιπλέον, αναφέρονται πιθανές μελλοντικές επεκτάσεις του συστήματος που υλοποιήθηκε.

Το 8^ο κεφάλαιο περιλαμβάνει μια παρουσίαση της βιβλιογραφίας η οποία χρησιμοποιήθηκε κατά την εκπόνηση της εργασίας.

Το 9^ο κεφάλαιο αποτελεί το παράρτημα του παρόντος τόμου. Στο παράρτημα φαίνεται η μελέτη των υφιστάμενων προτύπων για τη διάθεση χωρικών πληροφοριών μέσω του διαδικτύου. Συγκεκριμένα, παρουσιάζονται τα περισσότερα από τα πρότυπα που δίνει ο OGC καθώς και τα πρότυπα που δίνει ο ISO σχετικά με τα μετα-δεδομένα των χωρικών πληροφοριών που ανταλλάσσονται.

2

Web services για διάθεση χωρικής πληροφορίας

Στο κεφάλαιο αυτό γίνεται μελέτη και αποτίμηση διαφόρων εμπορικά διαθέσιμων web services που υπάρχουν σήμερα για τη διάθεση χωρικών πληροφοριών. Αρχικά αναλύουμε γενικά την τεχνολογία των διαδικτυακών υπηρεσιών και στη συνέχεια περιγράφουμε τις «Γεωγραφικές» διαδικτυακές υπηρεσίες που παρέχονται από τη Michelin, τη Microsoft και τη Google. Σκοπός μας είναι να βγουν χρήσιμα συμπεράσματα για το τι είδους υπηρεσίες θα πρέπει να παρέχουμε, βασισμένοι σε ορισμένες από τις πιο δημοφιλείς υπηρεσίες που υπάρχουν σήμερα.

2.1 Διαδικτυακές υπηρεσίες (Web Services)

Προτού περιγράψουμε την τεχνολογία των διαδικτυακών υπηρεσιών κάνουμε μια σύντομη εισαγωγή στην XML.

2.1.1 XML

Η XML (Extended Mark-up Language) είναι μια δηλωτική γλώσσα που ήρθε μετά την ASN1. Κύρια διαφορά είναι ότι στην XML χρησιμοποιούνται αποκλειστικά χαρακτήρες (character based, έναντι της ASN1 που είναι bit coded). Επιπλέον η δομή της είναι ταυτοχρόνως πιο απλή αλλά και πιο εύκαμπτη. Από άλλη οπτική γωνία, στην XML ένα κείμενο χρησιμοποιείται για την αναπαράσταση δομημένης πληροφορίας στο διαδίκτυο.

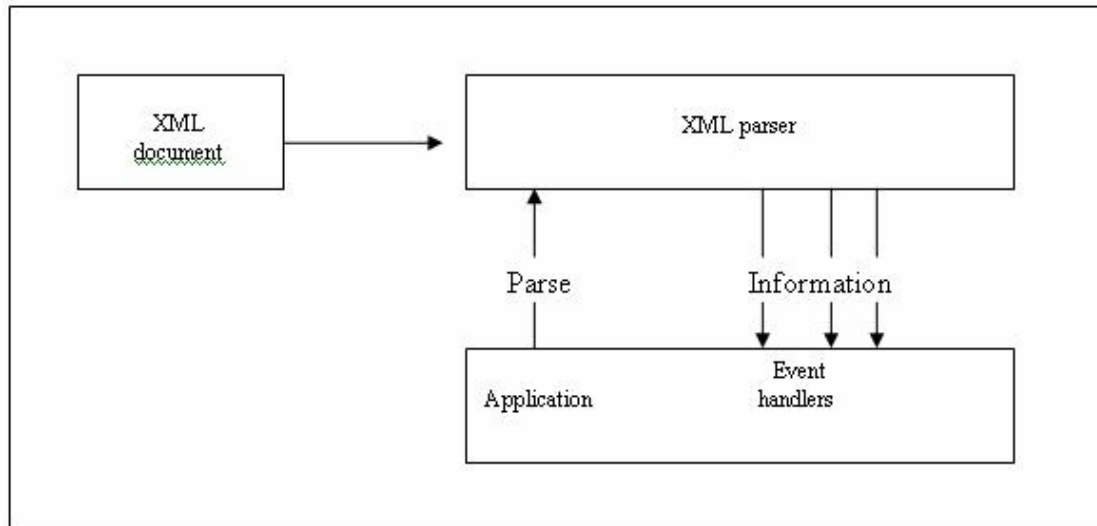
Όπως και στην HTML (Hyper Text Mark-up Language - και η XML - είναι γλώσσα που επισημαίνει επιπλέον πληροφορία αναφερόμενη στο βασικό κείμενο. Στην XML προσδιορίζουμε την πληροφορία αυτή με την βοήθεια ετικετών (tags), δηλ. με ένα αναγνωριστικό περικλειόμενο από <>. Με τις ετικέτες (tags) επιτυγχάνουμε το “markup”. Tags υπάρχουν και στην HTML, π.χ. το ζεύγος ή το μοναδικό
, αυτά όμως είναι προκαθορισμένα και σχετίζονται μόνο με την εμφάνιση του κειμένου. Τα tags της XML είναι προσδιορίσιμα ελεύθερα από τον χρήστη, ο οποίος μπορεί να τους δίδει και όποια σημασία αυτός ορίσει. Για τούτο πρέπει και πάντα να ζευγαρώνονται, δηλ. <text1>my car is here</text1>. Το αναγνωριστικό <text1> </text1> δεν προσδιορίζει αναγκαστικά το πώς θα εμφανισθεί το περικλειόμενο κείμενο (όπως θα έκανε το της HTML), αλλά μπορεί π.χ. να συνδυάζεται με μία ενέργεια, όπως θα καθορίζει δικός μας κώδικας που θα επεξεργασθεί το αρχείο XML. Το ζεύγος <text1> και </text1> ορίζει ένα στοιχείο (element), του οποίου το όνομα είναι το text1 και το περιεχόμενο οτιδήποτε περικλείεται μεταξύ του <text1> (opening tag) και του </text1> (closing tag). Το επόμενο βήμα είναι η ενθυλάκωση (nesting) που δίδει την δύναμη να κατασκευασθούν στοιχεία με πολύπλοκη εσωτερική δενδρική δομή.

Η σημασία της XML έγκειται στο γεγονός ότι διατίθενται έτοιμοι τρόποι σύνδεσης της δομής των κειμένων της με τα πλέον σύγχρονα προγραμματιστικά περιβάλλοντα. Στην διαδικασία αυτή εμπλέκεται πάντα ένας parser. Ο parser κατέχει κεντρικό ρόλο στην αντιστοίχιση και αλληλοσύνδεση της δενδρικής δομής του κειμένου XML με την αντίστοιχη (συνήθως αντικειμενοστραφή) δομή στην χρησιμοποιούμενη γλώσσα, με δεδομένο ότι το κείμενο XML είναι ένας συρμός χαρακτήρων ανταλλασσόμενος μέσω μίας επικοινωνιακής σύνδεσης, ενός αρχείου στον δίσκο, κτλ. Ο parser αποκτά τέτοια σημασία, που θεωρείται πλέον *core technology*. Ενδεικτικά παρουσιάζουμε εν συνεχεία έναν από τους πιο δημοφιλείς parsers που βασίζεται στο Simple API for XML (SAX).

2.1.1.1 XML και SAX

Το Simple API for XML (SAX) προσφέρεται από τον SAX parser, τον οποίο μπορούμε να φανταστούμε σαν έναν σειριακό αναγνώστη (SAX reader) του κειμένου XML, χαρακτήρα προς χαρακτήρα. Ο SAX parser αναγνωρίζοντας τα διάφορα μέρη του XML αρχείου, δημιουργεί αντίστοιχα συμβάντα (events), στα οποία το δικό μας πρόγραμμα οφείλει να ανταποκριθεί σύμφωνα με τις επιθυμίες μας. Ο SAX parser διαβάσει την ταινία (συρμός χαρακτήρων) του XML κειμένου και ανιχνεύει και σταματά σε συγκεκριμένα σημεία. Τούτο

δημιουργεί ένα συμβάν (event) προς τον δικό μας event handler. Μόλις ο event handler αντιμετωπίσει το event, συνεχίζει ο parser μέχρι το επόμενο σημείο. Στο επόμενο σχήμα, φαίνεται η σειριακή είσοδος του κειμένου XML στον SAX parser. Αυτός ελέγχεται από την εφαρμογή στην οποία δηλώνει πίσω (callbacks) συμβάντα, δηλαδή την ανίχνευση των επί μέρους μερών του υπό ανάγνωση XML.



Εικόνα 1: Γενική αρχή SAX parser

2.1.2 Εισαγωγή στα web services

Ένα web service είναι ένα κομμάτι λογισμικού (software module) το οποίο εκτελεί μια ασυνεχή εργασία ή σύνολο εργασιών το οποίο μπορεί να κληθεί πάνω από ένα δίκτυο και ιδιαίτερα το διαδίκτυο. Ο προγραμματιστής μπορεί να δημιουργεί μία εφαρμογή πελάτη (client) η οποία να καλεί μια σειρά από διαδικτυακές υπηρεσίες (web services) μέσω remote procedure calls (RPC) ή μέσω υπηρεσιών μηνυμάτων για να ολοκληρώσει μέρος ή ολόκληρο το λογικό κομμάτι της εφαρμογής. Μια δημοσιευμένη διαδικτυακή υπηρεσία περιγράφει τον εαυτό της έτσι ώστε οι προγραμματιστές να μπορούν να εντοπίσουν την υπηρεσία και να αξιολογήσουν την καταλληλότητά της για τις ανάγκες τους.

Για παράδειγμα, μια εταιρία θα μπορούσε να παρέχει μια διαδικτυακή υπηρεσία στους πελάτες της η οποία να ελέγχει τον κατάλογο των απογραφέντων προϊόντων πριν γίνει μια παραγγελία. Άλλο παράδειγμα είναι η υπηρεσία Federal Express package tracking service την οποία οι πελάτες μπορούν να χρησιμοποιήσουν για να ακολουθήσουν τις αποστολές εμπορευμάτων.

Οι διαδικτυακές υπηρεσίες χρησιμοποιούν το SOAP (Simple Object Access Protocol) για να μεταφέρουν μηνύματα XML και χρησιμοποιούν ως μέσο μεταφοράς των SOAP μηνυμάτων το πρωτόκολλο HTTP. Τα μηνύματα SOAP είναι ουσιαστικά έγγραφα XML τα οποία στέλνονται μεταξύ μιας διαδικτυακής υπηρεσίας και της εφαρμογής που την καλεί.

Οι διαδικτυακές υπηρεσίες μπορούν να γραφούν σε οποιαδήποτε γλώσσα προγραμματισμού και να τρέξουν σε οποιαδήποτε πλατφόρμα. Ένας πελάτης (client) μιας διαδικτυακής υπηρεσίας μπορεί επίσης να είναι γραμμένος σε οποιαδήποτε γλώσσα προγραμματισμού και να τρέχει σε οποιαδήποτε πλατφόρμα. Έτσι, για παράδειγμα, ένας πελάτης γραμμένος στη γλώσσα Delphi ο οποίος τρέχει σε Windows θα μπορούσε να καλέσει μια διαδικτυακή υπηρεσία γραμμένη σε Java η οποία να τρέχει σε Linux.

2.1.3 Αρχιτεκτονική των web services

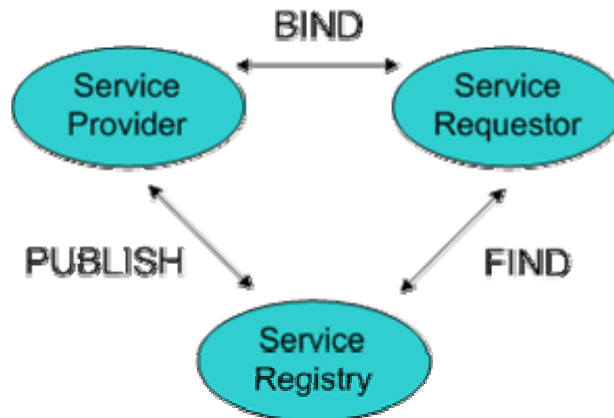
Η αρχιτεκτονική των διαδικτυακών υπηρεσιών επιτρέπει την ανάπτυξη τέτοιων διαδικτυακών υπηρεσιών οι οποίες ενθυλακώνουν όλα τα επίπεδα λειτουργικότητας. Με άλλα λόγια μια διαδικτυακή υπηρεσία μπορεί να είναι πολύ απλή, όπως για παράδειγμα μια υπηρεσία η οποία επιστρέφει την τρέχουσα θερμοκρασία, ή μπορεί να είναι μια πολύπλοκη εφαρμογή. Η αρχιτεκτονική τους, επιπλέον, επιτρέπει σε πολλαπλές διαδικτυακές υπηρεσίες να συνδυάζονται και να δημιουργούν νέα λειτουργικότητα.

Η αρχιτεκτονική των διαδικτυακών υπηρεσιών έχει τρεις ξεχωριστούς ρόλους: Έναν πάροχο (provider), τον καλών την υπηρεσία (requestor), και έναν μεσολαβητή (broker). Ο provider δημιουργεί τη διαδικτυακή υπηρεσία και την κάνει διαθέσιμη σε πελάτες (clients) οι οποίοι επιθυμούν να τη χρησιμοποιήσουν. Ένας requestor είναι μια εφαρμογή client η οποία καταναλώνει την διαδικτυακή υπηρεσία. Η υπηρεσία η οποία καλείται μπορεί με τη σειρά της να καλεί άλλες διαδικτυακές υπηρεσίες παίζοντας το ρόλο πελάτη. Ο broker, όπως για παράδειγμα ένας κατάλογος υπηρεσιών (service registry), παρέχει έναν τρόπο ώστε να επικοινωνούν ο provider και ο requestor μιας διαδικτυακής υπηρεσίας.

Οι τρεις ρόλοι, αυτοί του provider, του requestor και του broker αλληλεπιδρούν μεταξύ τους μέσω των λειτουργιών δημοσίευσης (publish), εύρεσης (find) και σύνδεσης (bind). Ένας provider ενημερώνει τον broker για την ύπαρξη μιας διαδικτυακής υπηρεσίας χρησιμοποιώντας τη διεπαφή δημοσίευσης του broker ώστε να κάνει την υπηρεσία προσβάσιμη στους clients. Οι πληροφορίες που δημοσιεύονται περιγράφουν την υπηρεσία

και καθορίζουν το πού βρίσκεται η υπηρεσία. Ο requestor συμβουλευτεί τον broker για να εντοπίσει μια δημοσιευμένη διαδικτυακή υπηρεσία. Με τις πληροφορίες που παίρνει από τον broker για την υπηρεσία, ο requestor μπορεί πλέον να συνδεθεί, ή να καλέσει τη διαδικτυακή υπηρεσία.

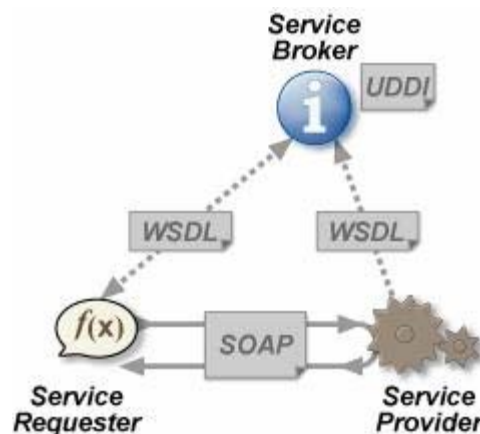
Το ακόλουθο διάγραμμα συνοψίζει το πώς ο provider, ο requestor και ο broker αλληλεπιδρούν μεταξύ τους.



Εικόνα 2: Web Service Roles and Operations

2.1.4 Τα πρότυπα των web services

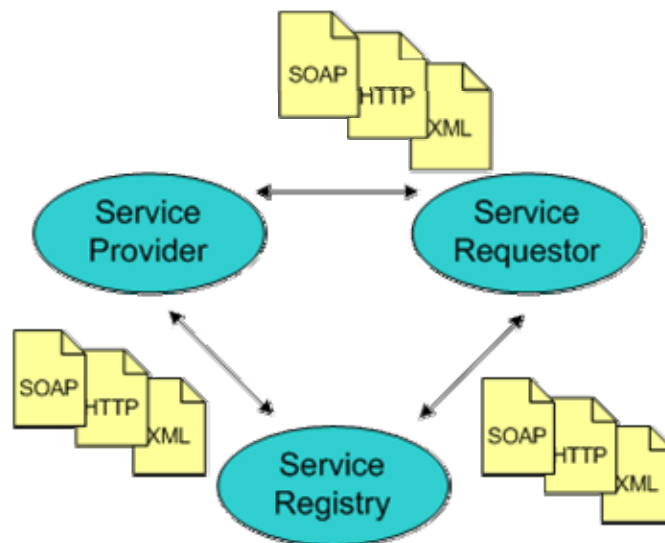
Τα πρότυπα πάνω στα οποία βασίζεται η ανάπτυξη των διαδικτυακών υπηρεσιών είναι εξελισσόμενες τεχνολογίες. Τα κυριότερα από αυτά είναι το SOAP (Simple Object Access Protocol), το WSDL (Web Services Description Language), το UDDI (Universal Description, Discovery and Integration), και το WSIL (Web Services Inspection Language).



Εικόνα 3: Δομή των Web Services

2.1.4.1 Simple Object Access Protocol (SOAP)

Το SOAP είναι ένα transport-independent πρωτόκολλο μηνυμάτων. Κάθε μήνυμα SOAP είναι ένα έγγραφο XML. Το SOAP χρησιμοποιεί μηνύματα μίας κατεύθυνσης παρόλο που γίνεται να συνδυάσει μηνύματα σε ακολουθίες request-and-response μηνυμάτων. Ο ορισμός του SOAP καθορίζει τη μορφή του μηνύματος XML αλλά όχι το περιεχόμενό του και το πώς ακριβώς θα αποσταλεί. Το SOAP, εντούτοις, καθορίζει το πώς τα μηνύματα SOAP δρομολογούνται πάνω από HTTP.



Εικόνα 4: SOAP

Κάθε έγγραφο SOAP έχει ένα root <Envelope> element. Το root element, το πρώτο element μέσα σε ένα έγγραφο XML, περιέχει όλα τα άλλα elements που υπάρχουν μέσα στο έγγραφο. Μέσα στον «φάκελο» (envelope) υπάρχουν δύο μέρη: μια επικεφαλίδα (header) και ένα σώμα (body). Η επικεφαλίδα περιέχει πληροφορίες δρομολόγησης ή context data. Μπορεί να είναι κενό. Το σώμα περιέχει το κυρίως μήνυμα. Μπορεί επίσης να είναι κενό.

Στη συνέχεια παρατίθεται ένα παράδειγμα ενός απλού μηνύματος SOAP το οποίο στέλνεται πάνω από HTTP και ζητά την τρέχουσα τιμή αποθήκης του Borland:

```
POST /StockQuote HTTP/1.1
Host: www.stockquoteserver.com
Content-Type: text/xml; charset="utf-8"
```



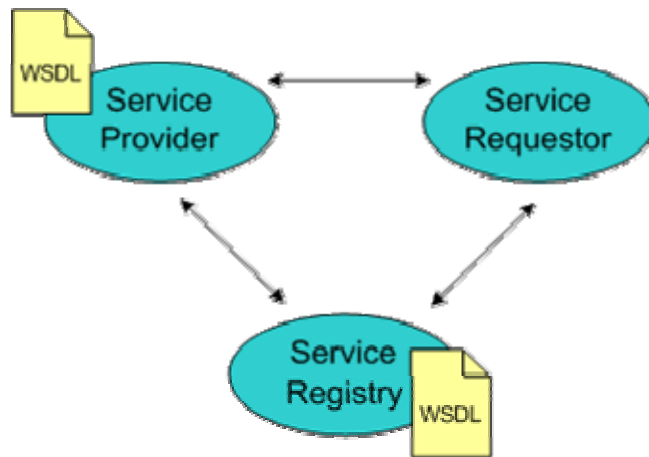
```
Content-Length: nnnn
SOAPAction: "urn:stock-quote-services"

<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <m:GetLastTradePrice xmlns:m="Some-URI">
      <symbol>BORL</symbol>
    </m:GetLastTradePrice>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Για περισσότερες πληροφορίες για το SOAP, δείτε αρχικά τα έγγραφα SOAP στο World Wide Consortium [SP]. Μπορείτε επίσης να επισκεφθείτε τη σελίδα της Apache SOAP [SPA].

2.1.4.2 *Web Services Description Language (WSDL)*

Μια διαδικτυακή υπηρεσία είναι χρήσιμη μόνο εφόσον οποιοσδήποτε μπορεί να βρει τι κάνει και με ποιον τρόπο να την καλέσει. Οι προγραμματιστές πρέπει να γνωρίζουν επαρκείς πληροφορίες για μια διαδικτυακή υπηρεσία έτσι ώστε να μπορούν να γράψουν ένα πρόγραμμα client το οποίο να την καλεί. Το WSDL είναι μια XML-based γλώσσα η οποία χρησιμοποιείται για να ορίσει διαδικτυακές υπηρεσίες και να περιγράψει το πώς μπορεί κανείς να έχει πρόσβαση σε αυτές. Ειδικότερα, περιγράφει το πώς πρέπει να είναι τα δεδομένα και τα μηνύματα που προσφέρει η διαδικτυακή υπηρεσία. Εξετάζοντας το έγγραφο WSDL μιας διαδικτυακής υπηρεσίας, οι προγραμματιστές γνωρίζουν τις μεθόδους που είναι διαθέσιμες και το πώς να τις καλούν χρησιμοποιώντας τις κατάλληλες παραμέτρους.



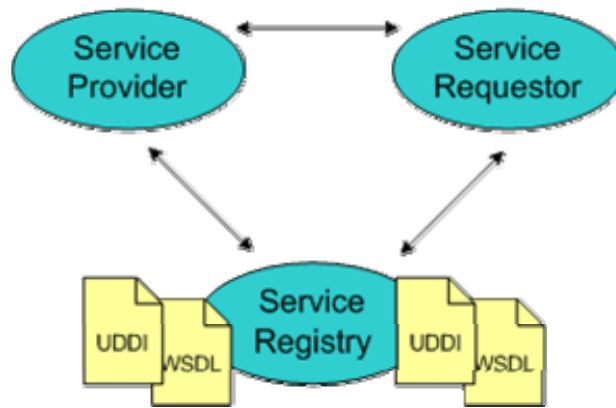
Εικόνα 5: WSDL

Για περισσότερες και σε βάθος πληροφορίες για το WSDL δείτε το Web Services Description Language 1.1 specification στο [WSD].

2.1.4.3 Universal Description, Discovery and Integration (UDDI)

Το UDDI είναι ένα αναπτυσσόμενο πρότυπο για την περιγραφή, δημοσίευση και ανακάλυψη των διαδικτυακών υπηρεσιών που προσφέρει μία εταιρία. Πρόκειται για έναν ορισμό για έναν καταναμημένο κατάλογο πληροφοριών για διαδικτυακές υπηρεσίες. Μόλις αναπτυχθεί μια διαδικτυακή υπηρεσία και δημιουργηθεί και ένα έγγραφο WSDL το οποίο την περιγράφει, πρέπει στη συνέχεια να υπάρχει ένας τρόπος να έρθει η πληροφορία του WSDL στα χέρια των χρηστών οι οποίοι θέλουν να χρησιμοποιήσουν τις διαδικτυακές υπηρεσίες τις οποίες αυτό περιγράφει. Μόλις μια διαδικτυακή υπηρεσία δημοσιεύεται σε έναν κατάλογο UDDI, πιθανοί χρήστες έχουν τη δυνατότητα να ψάξουν και να μάθουν για την ύπαρξη της υπηρεσίας αυτής.

Το περιεχόμενο ενός καταλόγου UDDI είναι παρόμοιο με αυτό ενός τηλεφωνικού καταλόγου. Στον τομέα των «λευκών σελίδων» του καταλόγου υπάρχει πληροφορία όπως το όνομα, η διεύθυνση και ο τηλεφωνικός αριθμός της εταιρίας η οποία παρέχει μία ή περισσότερες διαδικτυακές υπηρεσίες. Ο τομέας των «κίτρινων σελίδων» καθορίζει τον τύπο της εταιρίας και την κατηγοριοποιεί ανά βιομηχανικό τομέα. Οι «πράσινες σελίδες» παρέχουν τα δεδομένα για τις διαδικτυακές υπηρεσίες που παρέχει η εταιρία.

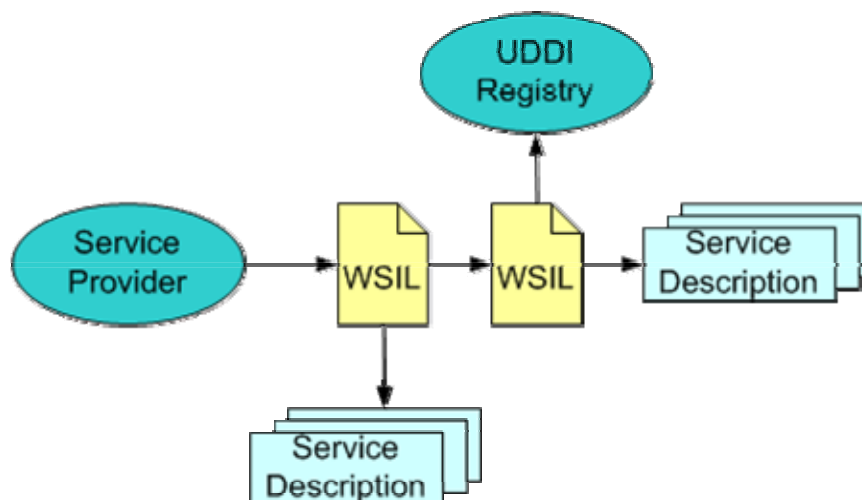


Εικόνα 6: UDDI

Για περισσότερες πληροφορίες για το UDDI δείτε το [UDD].

2.1.4.4 Web Services Inspection Language (WSIL)

Το WSIL, όπως και το UDDI, παρέχει μια μέθοδο για ανακάλυψη υπηρεσιών για τις διαδικτυακές υπηρεσίες. Αντίθετα με το UDDI, το WSIL χρησιμοποιεί ένα από-κεντροποιημένο, κατανεμημένο μοντέλο. Τα έγγραφα WSIL, τα οποία είναι βασικά δείκτες σε λίστες υπηρεσιών, επιτρέπουν στους καταναλωτές των διαδικτυακών υπηρεσιών να βρίσκουν διαθέσιμες υπηρεσίες σε ιστοσελίδες. Ο ορισμός του WSIL παρέχει πρότυπα για τη χρήση XML-formatted εγγράφων για να εξετάζουν μια ιστοσελίδα για υπηρεσίες και ένα σύνολο κανόνων για το πώς οι πληροφορίες αυτές θα γίνονται διαθέσιμες. Ένα έγγραφο WSIL συγκεντρώνει πολλαπλές αναφορές σε προ-υπάρχοντα έγγραφα περιγραφής υπηρεσιών μέσα σε ένα έγγραφο. Το έγγραφο WSIL στη συνέχεια φιλοξενείται από τον πάροχο της υπηρεσίας και έτσι οι καταναλωτές μπορούν να μάθουν για τις διαθέσιμες υπηρεσίες.



Εικόνα 7: WSIL

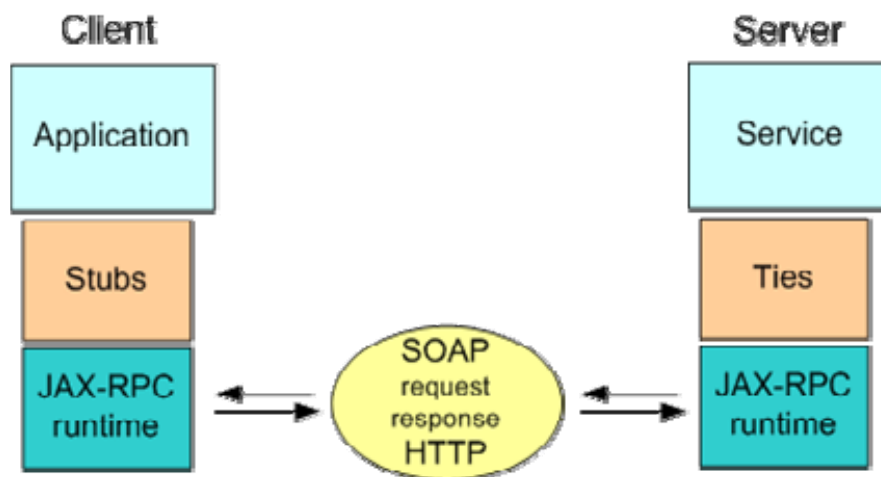
Για περισσότερες πληροφορίες για το Web Services Inspection Language (WS-Inspection) 1.0 specification, επισκεφθείτε το [WSL].

2.1.5 Java API for XML-based Remote Procedure Call (JAX-RPC)

Το JAX-RPC ορίζει ένα Java API το οποίο μπορούν να χρησιμοποιήσουν οι προγραμματιστές Java στις εφαρμογές τους για να αναπτύξουν και να καταναλώνουν διαδικτυακές υπηρεσίες. Χρησιμοποιώντας το JAX-RPC, ένας Java client μπορεί να καταναλώνει μια διαδικτυακή υπηρεσία που τρέχει σε έναν απομακρυσμένο server στο διαδίκτυο, ακόμη και αν η υπηρεσία είναι γραμμένη σε διαφορετική γλώσσα προγραμματισμού και τρέχει σε διαφορετική πλατφόρμα. Μια υπηρεσία JAX-RPC μπορεί επίσης να καταναλωθεί από non-Java clients.

Το JAX-RPC χρησιμοποιεί ένα XML-messaging πρωτόκολλο, όπως είναι το SOAP, για τη μεταφορά μιας απομακρυσμένης κλήσης μιας διαδικασίας πάνω από ένα δίκτυο. Για παράδειγμα, μια διαδικτυακή υπηρεσία η οποία επιστρέφει μια αναφορά αποθεμάτων θα λάμβανε ένα SOAP HTTP request το οποίο θα περιείχε την κλήση μιας μεθόδου από τον client. Χρησιμοποιώντας JAX-RPC, η υπηρεσία εξάγει την κλήση της μεθόδου από το μήνυμα SOAP, τη μεταφράζει σε μία πραγματική κλήση της μεθόδου και την καλεί. Στη συνέχεια η υπηρεσία χρησιμοποιεί JAX-RPC για να μετατρέψει το response της μεθόδου ξανά σε SOAP και να στείλει τα αποτελέσματα πίσω στον client. Ο client λαμβάνει το μήνυμα SOAP και χρησιμοποιεί JAX-RPC για να εξάγει το response.

Το JAX-RPC runtime δημιουργεί stubs και ties, τα οποία είναι κλάσεις που επιτρέπουν επικοινωνία μεταξύ του client και της υπηρεσίας. Ένα stub, το οποίο βρίσκεται στην πλευρά του client, είναι ένα τοπικό αντικείμενο το οποίο αντιπροσωπεύει μια απομακρυσμένη υπηρεσία και ενεργεί ως πληρεξούσιος για την υπηρεσία. Ένα tie, το οποίο είναι στην πλευρά του server, ενεργεί ως πληρεξούσιος στον server.



Εικόνα 8: JAX-RPC

Για περισσότερες πληροφορίες για το JAX-RPC δείτε το [JR] και το JAX-RPC tutorial [JRT].

2.2 *Michelin viaMichelin*

Η ViaMichelin [MIC] είναι μία εταιρία η οποία παρέχει δικτυακές υπηρεσίες για τη διάθεση χωρικής πληροφορίας. Κάποιες χαρακτηριστικές και απαιτητικές υπηρεσίες είναι για παράδειγμα η εύρεση γεωγραφικής θέσης, δημιουργία χάρτη, καθώς και υπηρεσίες εύρεσης διαδρομής (route planning). Οι παραπάνω υπηρεσίες τις οποίες προσφέρει η ViaMichelin είναι προσβάσιμες από οποιαδήποτε πλατφόρμα ανάπτυξης συμπεριλαμβανομένων των Java και .NET.

Ο λόγος για τον οποίο οι ViaMichelin Web Services είναι ανεξάρτητες πλατφόρμες, είναι ότι βασίζονται σε ανοιχτά πρότυπα όπως τα XML/SOAP/WSDL. Το γεγονός αυτό, κάνει τις παραπάνω υπηρεσίες ευέλικτες στο να μπορεί ο οποιοσδήποτε να αναπτύξει και να καθορίσει «υπηρεσίες βοήθειας ταξιδιού». Μπορεί κανείς να διασυνδέει τις web services που παρέχει η ViaMichelin με οποιαδήποτε επιχειρησιακή εφαρμογή όπως για παράδειγμα εποπτεία στόλου οχημάτων (fleet management), τηλεφωνικά κέντρα (call centers), ERP, CRM, ιστοσελίδες ηλεκτρονικού εμπορίου (e-commerce), και πολυκαναλικές εφαρμογές (multi-channel applications). Επιπλέον, μπορεί να γίνει διασύνδεση με βάσεις δεδομένων που περιέχουν πληροφορίες όπως διευθύνσεις πελατών και συντεταγμένες οχημάτων και βρίσκονται στον server της εταιρίας ή της ViaMichelin. Όλα τα παραπάνω σημαίνουν ότι μπορεί κανείς γρήγορα να αναπτύξει αυτές τις υπηρεσίες του σε οποιοδήποτε δίκτυο ή συσκευή όπως για παράδειγμα το Internet, WAP τηλέφωνα, PDAs, SmartPhones.

Οι υπηρεσίες της ViaMichelin παρέχουν προς το παρόν, μια παγκόσμια γεωγραφική κάλυψη όσον αφορά στο περιεχόμενο των χαρτών που περιλαμβάνεται στους server της. Συγκεκριμένα περιλαμβάνονται τα εξής:

- 42 ευρωπαϊκές χώρες και όλη η Βόρεια Αμερική
- 5 εκατομμύρια μίλια στην Ευρώπη και 10 εκατομμύρια μίλια στη Βόρεια Αμερική
- πάνω από ένα εκατομμύριο τοποθεσίες, 90 χιλιάδες εκ των οποίων είναι γεω-κωδικοποιημένες μέχρι και το επίπεδο της οδού.
- Μια βάση δεδομένων με σημεία ενδιαφέροντος (Points of Interest) προσαρμοσμένη σε επαγγελματικές απαιτήσεις.
- Υπηρεσίες και πληροφορίες επιλεγμένες ώστε να ικανοποιούν τις ανάγκες ανθρώπων που συνεχώς μετακινούνται.

Άλλο ένα χαρακτηριστικό των υπηρεσιών που παρέχονται είναι η εγγύηση για την άψογη λειτουργία τους. Το παραπάνω γεγονός οφείλεται στα εξής:

- Οι υπηρεσίες που παρέχονται βρίσκονται σε περισσότερους από εκατό servers.
- Υπάρχει συνεχής έλεγχος
- Γίνονται συνεχώς μετρήσεις για τους χρόνους απόκρισης των υπηρεσιών.

Τέλος, όσον αφορά στο κόστος των υπηρεσιών αυτών, μπορούμε να πούμε ότι η χρήση των υπηρεσιών αυτών έχει το πλεονέκτημα ότι η χρέωση γίνεται μόνο όσο χρησιμοποιούμε την υπηρεσία. Συνεπώς δεν μας ενδιαφέρει το κόστος για την αποθήκευση των χαρτών και άλλων πληροφοριών στους servers, η ανανέωση των πληροφοριών αυτών ή οτιδήποτε άλλο.

2.2.1 Λίστα των υπηρεσιών που προσφέρονται:

Η ViaMichelin προσφέρει τις παρακάτω λειτουργίες:

- Geocoding: Πρόκειται για υπηρεσίες «γεω-κωδικοποίησης». Συγκεκριμένα η λειτουργία αυτή ελέγχει την ορθότητα μιας διεύθυνσης που δίνεται σαν είσοδος και επιστρέφει τις γεωγραφικές συντεταγμένες που της αντιστοιχούν. Επιπλέον εντοπίζεται η χώρα, η πολιτεία, η πόλη, ο δρόμος ή οποιοδήποτε άλλο χαρακτηριστικό που αντιστοιχεί στη διεύθυνση αυτή και το παρουσιάζει σε ένα χάρτη ψάχνοντας συγχρόνως για άλλα μέρη τριγύρω. Η παραπάνω λειτουργία έχει βελτιστοποιηθεί έτσι ώστε να αναγνωρίζει διευθύνσεις σε διαφορετικές χώρες, χρησιμοποιώντας τη μορφοποίηση και τις συμβάσεις

που κάθε χώρα έχει για τις διευθύνσεις.

- **Reverse Geocoding:** Η συγκεκριμένη λειτουργία βρίσκει τη συγκεκριμένη διεύθυνση για ένα μέρος ή γεωγραφικό χαρακτηριστικό από τις γεωγραφικές συντεταγμένες του. Πρόκειται, δηλαδή, για την αντίστροφη της παραπάνω λειτουργίας. Με τη λειτουργία αυτή μπορούμε, για παράδειγμα, να αναγνωρίσουμε πάνω σε χάρτη την ακριβή θέση ενός οχήματος (εντοπισμένο από ένα σύστημα GPS) του οποίου οι συντεταγμένες έχουν σταλεί στην εφαρμογή μας.
- **Route Calculation:** Η λειτουργία αυτή μας δίνει τη δυνατότητα υπολογισμού διαδρομών από ένα σημείο σε ένα άλλο, δίνοντας αναλυτικά πλάνα ταξιδιού και οδηγίες καθοδήγησης. Η λειτουργία αυτή είναι διαθέσιμη στη Ευρώπη και στη Βόρεια Αμερική. Το εύρος των διαθέσιμων πληροφοριών κάνει την υπηρεσία αυτή μια ιδιαίτερα ισχυρή και αξιόπιστη route-planning engine. Οι επιλογές που έχουμε και οι διαθέσιμες πληροφορίες περιλαμβάνουν:
 - Τύπος διαδρομής (κοντινότερη, ταχύτερη, χωρίς διόδια, με τη μέγιστη χρησιμοποίηση των αυτοκινητόδρομων, κτλ)
 - Τύπος οχήματος (μοτοσικλέτα, αυτοκίνητο, HVG, κτλ)
 - Πληροφορίες χρέωσης (τέλη διοδίων ανάλογα και με τον τύπο οχήματος κτλ)
 - Λεπτομέρειες από εποχιακές ή ημερήσιες εκδηλώσεις ή συμβάντα τα οποία πιθανά επηρεάζουν το πλάνο διαδρομής μας (έργα στο δρόμο, festivals, κλειστά περάσματα σε βουνά κτλ)
 - Αναλυτικά πλάνα διαδρομών (χάρτης διαδρομής, οδηγίες οδήγησης, συνολική απόσταση, χρόνος ταξιδιού κτλ) ή περιληπτικά πλάνα διαδρομών (συνολική απόσταση, τύπος δρόμου και χρόνος ταξιδιού). Η περίληψη του πλάνου διαδρομής μπορεί επιπλέον να χρησιμοποιηθεί για την εύρεση κάποιου μέρους το οποίο βρίσκεται πλησιέστερα σε δρόμο.
- **Find POI (Point Of Interest) NEARBY:** Η συγκεκριμένη λειτουργία μας επιτρέπει να αναζητούμε ένα συγκεκριμένο αριθμό διευθύνσεων ή τοποθεσιών πλησιέστερα (σε ευθεία γραμμή) σε μια συγκεκριμένη διεύθυνση ή μέρος ενδιαφέροντος σε μια ορισμένη από το χρήστη ακτίνα αναζήτησης. Για παράδειγμα, η υπηρεσία αυτή μπορεί να ψάξει για έμπορους αυτοκινήτων πλησιέστερα σε μια δοσμένη θέση ή να βρει ανταγωνιστές μιας επιχείρησης οι οποίοι είναι κοντά σε πωλήσεις μαζί της και, σαν αποτέλεσμα, να αναλuthούν περιφέρειες που εξυπηρετούνται λιγότερο.
- **Distance Calculation:** Η λειτουργία αυτή είναι ουσιαστικά μια επέκταση στη Find POI Nearby. Υπολογίζει την απόσταση (σε ευθεία γραμμή) μεταξύ διαφόρων τοποθεσιών,

ταξινομώντας τις με απλή σειρά εγγύτητας ή συνδυάζοντας άλλα κριτήρια (για παράδειγμα μπορούμε να εμφανίσουμε τους πιο κοντινούς εμπορείους πρώτα και μετά, με σειρά προτεραιότητας, τα κοντινότερα μαγαζιά).

- XY to Pixels: Η συνάρτηση αυτή εντοπίζει αυτόματα ένα αντικείμενο με γνωστές γεωγραφικές συντεταγμένες πάνω σε ένα χάρτη χωρίς να πρέπει να χρησιμοποιήσει καθόλου λειτουργία γεωκωδικοποίησης (geocoding). Η λειτουργία της δημιουργίας χάρτη μπορεί να συνδυαστεί έτσι άμεσα με την εμφάνιση λογοτύπων, για παράδειγμα, εντοπίζοντας αντικείμενα σε πραγματικό χρόνο.
- Pixels to XY: Η λειτουργία αυτή εμφανίζει γεωγραφικές συντεταγμένες ενός αντικειμένου πάνω σε ένα χάρτη. Αυτό μας δίνει τη δυνατότητα να έχουμε πλάνα διαδρομών ή υπολογισμό αποστάσεων άμεσα από το χάρτη χωρίς να απαιτείται προηγούμενη γεωκωδικοποίηση του αντικειμένου. Έτσι ελαχιστοποιείται ο χρόνος επεξεργασίας ερωτημάτων με το να μειώνουμε τον απαιτούμενο αριθμό δοσοληψιών δεδομένων.
- Get Best Map Definition: Η συγκεκριμένη λειτουργία μας δίνει τις παραμέτρους που χρειάζονται για τη δημιουργία ενός χάρτη για ένα δεδομένο μέγεθος, κλίμακα και δυνατότητα πλοήγησης (δυνατότητα zoom, κίνησης του χάρτη ανατολικά ή δυτικά κτλ). Ο χάρτης αυτός μπορεί εν συνεχεία να συνδυαστεί με πληροφορία η οποία παρέχεται από ένα σύστημα GIS (Geographical Information System) και να παρουσιαστεί τελικά σε ένα ViaMichelin χάρτη.
- Generate Map: Η υπηρεσία αυτή είναι ουσιαστικά μια επέκταση της Get Best Map Definition. Δημιουργεί έναν εμπλουτισμένο ViaMichelin χάρτη (παρουσιάζοντας τουριστικές, διοικητικές περιοχές, σταθμούς, πάρκα κτλ) σε ένα GIF format για κάθε τύπο πολυμέσων. Αντικείμενα μπορούν να εμφανίζονται στη συνέχεια στο χάρτη με προσαρμοσμένους τύπους λογοτύπων.

Κλείνοντας, αναφέρουμε τις απαιτήσεις συστήματος για να μπορεί κάποιος να χρησιμοποιεί τις παραπάνω web services. Αρχικά, θα πρέπει να υπάρχει ένας υπολογιστής με σύνδεση στο internet. Επιπλέον, θα πρέπει να έχουμε τη δυνατότητα να κωδικοποιούμε μηνύματα μέσω του πρωτοκόλλου SOAP 1.1 και να μπορούμε να τα στείλουμε μέσω του HTTP 1.1. Απαραίτητη είναι, τέλος, η δυνατότητα συντακτικής ανάλυσης XML (XML parsing).

2.3 Microsoft Mappoint

2.3.1 Γενικά

Η διαδικτυακή υπηρεσία MapPoint είναι μια XML-based διαδικτυακή υπηρεσία με ευελιξία όσον αφορά τον προγραμματισμό της και παρέχεται από τη Microsoft. Για την εφαρμογή αυτή έχει χρησιμοποιηθεί η Visual Studio® .NET. Χρησιμοποιείται ευρέως από εταιρίες και ανεξάρτητους προγραμματιστές σε εφαρμογές λογισμικού με στόχο την ολοκλήρωση (integration) σε location-based υπηρεσίες όπως χάρτες, οδηγίες προς οδηγούς και αναζήτηση της πιο κοντινής διαδρομής σε εφαρμογές λογισμικού και διαδικτυακές πύλες (portals).

Επιπροσθέτως, η διαδικτυακή υπηρεσία MapPoint μπορεί να προσφέρει ένα εκτεταμένο σύνολο περιεχομένων που σχετίζονται με την χαρτογράφηση (όπως κατατάξεις επιχειρήσεων καθώς και σημείων ενδιαφέροντος (Points Of Interest-POIs) που είναι πιθανό να διατεθούν ως συστατικό εφαρμογών. Αυτή τη στιγμή υποστηρίζονται από τη δικτυακή υπηρεσία MapPoint περισσότερες από 15.000.000 δοσοληψίες σε καθημερινή βάση.

2.3.2 Λειτουργικότητα

Η διαδικτυακή υπηρεσία MapPoint προσφέρεται μέσω ενός SOAP (Simple Object Access Protocol) API. Κάθε μια από τις προσφερόμενες συναρτήσεις του MapPoint API στέλνει και λαμβάνει ως ορίσματα XML δεδομένα. Η υποστήριξη του SOAP και άλλων ανοιχτών πρότυπων δίνει στη διαδικτυακή υπηρεσία MapPoint το χαρακτηριστικό και προνόμιο της διαλειτουργικότητας (interoperability), με αποτέλεσμα να μπορεί εύκολα να συνδυαστεί και να ολοκληρωθεί με άλλες εφαρμογές.

Το MapPoint Web Service API απαριθμεί τουλάχιστον 66 διαφορετικές κλάσεις και εκατοντάδες ιδιότητες (properties) και απαριθμήσεις (enumerations). Το MapPoint προσφέρει μια πλειάδα υπηρεσιών που τοποθετούν στους χάρτες διευθύνσεις και σημεία ενδιαφέροντος εμπλουτίζοντας τους, κάνουν rendering χαρτών, ορίζουν διαδρομές βάσει σημείων (point-to-point) κ.ο.κ.

- **Common Services.** Η διαδικτυακή υπηρεσία MapPoint εμπεριέχει πέντε βασικές υπηρεσίες που καλούνται Common Services. Αυτές παρέχουν κλάσεις γενικής χρήσης, συναρτήσεις και ιδιότητες.
 - **GetGreatCircleDistances:** επιστρέφει έναν πίνακα με τις αποστάσεις -σε χιλιόμετρα ή μίλια- μεταξύ ενός δοσμένου σημείου και των κυριότερων

σημείων ενδιαφέροντος που εντοπίζονται σε μία περιοχή γύρω από το δοσμένο σημείο (circle distances).

- GetCountryRegionInfo: επιστρέφει το όνομα της χώρας ή της περιφέρειας, τον κωδικό αριθμό της γεωγραφικής οντότητας, τις γεωγραφικές συντεταγμένες (πλάτος και μήκος), τους κωδικούς και τη γλώσσα για μια γεωγραφική οντότητα.
 - GetDataSourceInfo: επιστρέφει τη λειτουργικότητα και το όνομα κάποιας πηγής δεδομένων.
 - GetEntityTypes: επιστρέφει τους τύπους και τις αντίστοιχες ιδιότητες των οντοτήτων που περιέχονται σε κάποια πηγή δεδομένων.
 - GetVersionInfo: επιστρέφει το όνομα και τον αριθμό της τρέχουσας έκδοσης της διαδικτυακής υπηρεσίας MapPoint.
- Find Service. Με την υπηρεσία αυτή πραγματοποιείται η γεωκοδικοποίηση όπως επίσης η αναζήτηση γνωστών σημείων ενδιαφέροντος και επιχειρήσεων. Η υπηρεσία αυτή χρησιμοποιεί οχτώ βασικές συναρτήσεις. Οι κυριότερες είναι:
 - GetLocationInfo: επιστρέφει μια λίστα διευθύνσεων και γεωγραφικών οντοτήτων στις δοσμένες συντεταγμένες
 - Find: επιστρέφει μια λίστα με τις γεωγραφικές οντότητες που βρέθηκαν μετά από αναζήτηση βάσει κάποιων επιλογών.

Συχνά χρησιμοποιούνται και οι ParseAddress και FindAddress:

- ParseAddress διαβάζει (parses) μια διεύθυνση(που εισάγεται σε απλή μορφή string όπως για παράδειγμα 1600 Pennsylvania Avenue) και επιστρέφει ένα αντικείμενο Διεύθυνσης (Address object).
- FindAddress επιστρέφει μια λίστα διευθύνσεων που βρέθηκαν μετά από αναζήτηση βασισμένη σε δοσμένη διεύθυνση, ταξινομημένη με κριτήριο το πόσο ταιριάζουν τα αποτελέσματα με τη δοσμένη διεύθυνση.

Λιγότερο συχνά χρησιμοποιούνται οι:

- FindByID: που επιστρέφει μια λίστα από σημεία ενδιαφέροντος με κριτήριο αναζήτησης το id κάποιας δοσμένης οντότητας,
- FindByProperty: που επιστρέφει μια λίστα με τα σημεία ενδιαφέροντος βάσει των τιμών ορισμένων ιδιοτήτων της οντότητας,

- FindNearby: που επιστρέφει λίστα σημείων ενδιαφέροντος βάσει του τύπου της ζητούμενης γεωγραφικής οντότητας ταξινομημένα με κριτήριο την εγγύτητα τους στο επελεγμένο σημείο και τέλος,
- FindNearRoute: που επιστρέφει μια λίστα με τα σημεία ενδιαφέροντος -βάσει του τύπου της οντότητας. Τα σημεία αυτά βρίσκονται σε κάποια συγκεκριμένη απόσταση από τη θεωρημένη διαδρομή.
- Render Service pushpins. Η υπηρεσία Render απαρτίζεται από τέσσερις συναρτήσεις που προσδιορίζουν και ανακτούν εικόνες-χάρτες με βάση τις γεωγραφικές συντεταγμένες, εισάγουν custom overlays και "push pins" για τη σήμανση συγκεκριμένων συντεταγμένων. Αυτές οι συναρτήσεις παρέχουν αφθονία επιλογών σχετικά με τη διαδικασία χαρτογράφησης. Η κλάση MapSpecification εφοδιάζει την εφαρμογή με όλες τις απαραίτητες επιλογές χαρτογράφησης όσον αφορά τις πηγές δεδομένων για τους χάρτες, τη μορφή των εικόνων, το μέγεθος και την κλίμακα, "pushpins" για τη σήμανση περιοχών στον χάρτη, μια λίστα με τα γεωγραφικά δεδομένα καθώς και όλες τις διαδρομές που πρόκειται να αναπαρασταθούν σε αυτόν.

Οι συναρτήσεις κλειδιά στον Render Server είναι:

- GetMap: επιστρέφει εικόνες-χάρτες, διαφορετικές όψεις/προοπτικές χαρτών και
- GetBestMapView: επιστρέφει την καλύτερη πιθανή όψη του χάρτη για μια επιλεγμένη περιοχή ή κάποιο σύνολο τους. Ως καλύτερη όψη/προοπτική ενός χάρτη ορίζεται από την Microsoft η προοπτική που έχοντας την μεγαλύτερη κλίμακα εμπεριέχει τις περισσότερες επιθυμητές περιοχές.

Ο εξυπηρετητής (server) αυτός περιλαμβάνει 2 συναρτήσεις παροχής βοήθειας.

- ConvertToLatLong που μετατρέπει τις συντεταγμένες εικονοστοιχείων (pixel coordinates) σε γεωγραφικές συντεταγμένες (πλάτος-μήκος) κάποιου χάρτη και
- ConvertToPoint που κάνει το αντίθετο, δηλαδή μετατρέπει τις γεωγραφικές συντεταγμένες σε συντεταγμένες εικονοστοιχείων (pixel coordinates). Αυτό δίνει τη δυνατότητα στον χρήστη να εισάγει ένα είδος αλληλεπίδρασης (interactivity) στο χάρτη.

Για παράδειγμα, είναι δυνατό να αντιλαμβάνεται ο server ποιο σημείο ενδιαφέροντος του χάρτη επιλέγουν οι χρήστες ή ακόμη να γίνεται αίτηση για τη μεταφορά του κέντρου του χάρτη.

- **Route Service.** Με αυτήν προσδιορίζεται το καλύτερο μονοπάτι-διαδρομή μεταξύ δυο σημείων και συντονίζεται η δημιουργία των χαρτών και των οδηγιών (βήμα-βήμα) που ακολουθούν αυτό το μονοπάτι. Είναι μια υπηρεσία πολύ ενδιαφέρουσα μιας και αποτελεί την πιο προγραμματιστικά πολύπλοκη. Υπάρχουν δυο παρόμοιες συναρτήσεις διαθέσιμες:
 - **CalculateRoute:** επιστρέφει μια διαδρομή με βάση τμήματα ή προδιαγραφές διαδρομής.
 - **CalculateSimpleRoute:** επιστρέφει κάποια διαδρομή με βάση κάποιες γεωγραφικές συντεταγμένες για τα σημεία αρχής και τέλους.
- **Customer Data Service.** Η υπηρεσία αυτή είναι η πιο πρόσφατη (Φεβρουάριος 2004) και περιλαμβάνεται στην έκδοση MapPoint Web Service 3.5. Με αυτή την υπηρεσία ο χρήστης έχει τη δυνατότητα να εισάγει στον server της Microsoft γεωγραφικά ή δημογραφικά δεδομένα. Αυτό μπορεί να γίνει είτε με το χέρι, είτε αυτόματα χρησιμοποιώντας μια συνάρτηση που εκτελεί εξαγωγή-μετασχηματισμό-εισαγωγή των δεδομένων με μια συγκεκριμένη σειρά. Αυτό καθιστά τις διάφορες εφαρμογές που χρησιμοποιούν τη διαδικτυακή υπηρεσία MapPoint πιο εύχρηστες μιας και ισχυροποιεί τα εκάστοτε δεδομένα του χρήστη-όπως οι θέσεις των παραρτημάτων μιας εταιρείας, τα site των πελατών, κτλ.- εξισώνοντας τα με αυτά που περιλαμβάνονται στη Βάση Δεδομένων της MapPoint.

2.3.3 Κυριότερες Υπηρεσίες

Η δικτυακή υπηρεσία MapPoint περιλαμβάνει τα παρακάτω χαρακτηριστικά [MPM]:

- **Εύρεση Διευθύνσεων.** Εντοπίζει γεωκωδικοποιημένες διευθύνσεις στην Βόρεια και Νότια Αμερική, καθώς και στην Ευρώπη. Το όνομα της οδού συνδέεται με την πόλη, το κράτος, τον ταχυδρομικό κώδικα ή το όνομα της οδού με την οποία γίνεται διασταύρωση έτσι ώστε το επιστρεφόμενο αποτέλεσμα να είναι σαφές και συγκεκριμένο.
- **Εύρεση περιοχών χωρίς διεύθυνση (Non-Addressable Places).** Αναζητά γεωγραφικές οντότητες όπως πόλεις, ταχυδρομικούς κώδικες, κράτη, χώρες, ποτάμια, λίμνες, αεροδρόμια, σύνορα και άλλα μέρη/οικοδομήματα που δεν μπορούν να οριστούν οδικές διευθύνσεις.
- **Αντιστροφή Γεωκωδικοποίησης (Reverse Geocoding).** Έχει πρόσβαση σε μια λίστα γεωγραφικών οντοτήτων που είναι συσχετισμένα με συγκεκριμένες συντεταγμένες όπως

μια χώρα ή ορισμένη εμπορική περιοχή.

- Εύρεση και Παροχή Σημείων Ενδιαφέροντος (POIs). Οι εταιρείες έχουν τη δυνατότητα να ολοκληρώνουν(integrate) εύκολα και γρήγορα σημεία ενδιαφέροντος που έχουν να κάνουν με κάποια συγκεκριμένη τοποθεσία.

Έτσι παραθέτει σε μικρό χρονικό διάστημα μια λίστα σημείων ενδιαφέροντος με κριτήριο την εγγύτητά τους σε μια επιλεγμένη τοποθεσία. Οι αναζητήσεις αυτές μπορούν να εκτελούνται για σημεία ενδιαφέροντος κατά μήκος μιας συγκεκριμένης δοσμένης διαδρομής. Η δικτυακή υπηρεσία MapPoint εμπεριέχει μια βάση δεδομένων με περισσότερα από 15.000.000 σημεία ενδιαφέροντος (POI), ενώ και οι πελάτες μπορούν να συνεισφέρουν εισάγοντας στη βάση(στον Microsoft server,όπως προαναφέρθηκε) άλλα συνήθη σημεία ενδιαφέροντος.

Για παράδειγμα ξενοδοχεία ή εστιατόρια μπορούν να παρέχουν τη θέση των κοντινότερων ATMs.Ή ακόμα μια κατασκευαστική εταιρία μπορεί να παρέχει μια λίστα με όλα τα πρατήρια υγρών καυσίμων σε μικρή ακτίνα από το κέντρο ανεφοδιασμού.

- Προσαρμογή στις απαιτήσεις του πελάτη (Customizing Locations). Η κωδικοποίηση των γεωγραφικών δεδομένων και η αποθήκευσή τους σε απλές βάσεις δεδομένων για τη χρησιμοποίησή τους από τη MapPoint μπορεί να πραγματοποιηθεί με ιδιαίτερη ευκολία. Είναι γεγονός ότι οι πελάτες προσμετρούν μέχρι και 300 ερευνησίμα γνωρίσματα για κάθε εγγραφή όπως για παράδειγμα, ο τύπος υπηρεσιών ή οι ώρες λειτουργίας.
- Δρομολόγηση (Routing) και Διάθεση οδηγίων. Παρέχει βελτιστοποιημένες οδηγίες για την προσέγγιση κάποιας περιοχής βήμα βήμα. Οι εταιρείες μπορούν να προσφέρουν οδηγίες με τη βοήθεια χαρτών και σταδιακών υποδείξεων για την προσέγγιση κάποιας περιοχής. Για παράδειγμα, μια ασφαλιστική εταιρεία μπορεί πολύ απλά να παρέχει τις εκάστοτε υποδείξεις για τον εντοπισμό θέσεως όπως η διεύθυνση ενός πελάτη ή η θέση ενός ασφαλιστή.
- Πολύγωνα. Οριοθετεί και φωτοσκιάζει μια περιοχή του χάρτη και ακολούθως τη χρησιμοποιεί για την εύρεση σχέσεων με άλλα πολύγωνα, πηγές σημειακών δεδομένων (point data sources), διευθύνσεις και συγκεκριμένες γεωγραφικές θέσεις. Η συνήθης χρήση των πολυγώνων περιλαμβάνει τον προσδιορισμό γειτνιάσεων, περιοχές διανομής, κάλυψη κινητής τηλεφωνίας και προσδιορισμό περιφερειών.

Η υπηρεσία MapPoint διαθέτει ένα εργαλείο δημιουργίας πολυγώνων καθώς και γραμμή εντολών με αρκετά εργαλεία για την μετατροπή ESRI, MapInfo, Autodesk, Intergraph και GML αρχείων σε MWS polygon file format.. Επιπροσθέτως η δικτυακή υπηρεσία

MapPoint διαθέτει επεκτάσεις COM για την εφαρμογή MapPoint 2004 που επιτρέπει στον χρήστη να σχεδιάσει τα δικά του πολύγωνα και να τα σώσει στο XML polygon file format.

- Ενημέρωση σε πραγματικό χρόνο για την κίνηση στους δρόμους (Real-Time Traffic Incidents). Έχει ανά πάσα στιγμή πρόσβαση στα δεδομένα για την τρέχουσα κατάσταση σε περισσότερες από 70 μητροπολιτικές περιοχές των ΗΠΑ όσον αφορά την οδική κυκλοφορία (ατυχήματα, συμφόρηση ή κατασκευαστικά έργα). Τα δεδομένα αυτά εμπεριέχουν χαρακτηριστικά όπως η σοβαρότητα των γεγονότων αυτών, η ροή της κυκλοφοριακής συμφόρησης, η αναμενόμενη διάρκεια και η αναμενόμενη καθυστέρηση.
- Rendering χαρτών. Αποδίδει τους χάρτες βελτιστοποιημένους με διαφορετικό ύψος και μέγεθος γραμματοσειράς ανάλογα με την συσκευή ή την εφαρμογή. Μερικές από αυτές είναι:
 - Line Drive Maps: Οι χρήστες της εφαρμογής αυτής έχουν τη δυνατότητα να απλοποιούν την χαρτογραφημένη διαδρομή απλοποιώντας τον χάρτη και αφαιρώντας τους δρόμους που δε θα ακολουθήσουν. Επιπλέον έχουν τη δυνατότητα να βλέπουν τη διαδρομή τους σε έναν τυπωμένο χάρτη.
 - Night Time Maps: Διαθέτει μαύρο φόντο και δρόμους σε χρώμα ανοιχτό πράσινο γεγονός που καθιστά εύκολη τη χρήση του σε κινητά τηλέφωνα και κατά τη διάρκεια της νύχτας. Ακόμη το μαύρο φόντο είναι βολικό για τα μάτια, καθώς τα απαλλάσσει από τη διαδικασία αναπροσαρμογής τους στις συνθήκες οθόνης και σκοταδιού.
 - Urban Inset Maps: Είναι στατικοί χάρτες που αναγνωρίζουν σημαντικούς δρόμους και σημεία ενδιαφέροντος των κυριότερων περιοχών. Συνήθως χρησιμοποιείται σε εφαρμογές που συνοδεύουν οδοιπόρους (aid travelers).

2.4 Google Maps API

Το Google Maps API [GMA] είναι μια διασύνδεση η οποία αναπτύχθηκε από την εταιρία Google και μας επιτρέπει να ενσωματώνουμε χάρτες Google στις δικές μας ιστοσελίδες χρησιμοποιώντας JavaScript. Μπορεί κανείς με τις λειτουργίες που παρέχονται να σχεδιάσει δείκτες πάνω στο χάρτη, ή να αναπτύξει ακόμα πιο πολύπλοκες εφαρμογές όπως θα δούμε. Προς το παρόν όμως οι υπηρεσίες αυτές που παρέχονται από την Google είναι διαθέσιμες μόνο για web σελίδες και δεν μπορούν να χρησιμοποιηθούν από άλλη εφαρμογή. Δε βασίζονται σε κάποιο ανοιχτό πρότυπο, όπως το SOAP/XML αλλά χρησιμοποιεί JavaScript

όπως προαναφέραμε. Για το λόγο αυτό, ο μόνος τρόπος ενσωμάτωσης τους είναι σε ιστοσελίδες.

Το πλεονέκτημα από τη χρήση των υπηρεσιών αυτών είναι ότι παρέχονται δωρεάν (τουλάχιστον για web sites με μέγιστο αριθμό επισκέψεων 50000/ημέρα). Το μόνο που χρειάζεται να κάνει κάποιος για να χρησιμοποιήσει τις υπηρεσίες αυτές είναι να εγγραφεί στο σύστημα της Google για να του παραχωρηθεί ένας κωδικός πρόσβασης στα API της Google (API key). Εφόσον όμως οι υπηρεσίες παρέχονται δωρεάν από τη Google θα πρέπει όποιος τις χρησιμοποιεί να μην τις διαθέτει στο site του επί πληρωμή αλλά επίσης δωρεάν προς τους επισκέπτες του ιστοχώρου του. Στη συνέχεια παρουσιάζουμε συνοπτικά τα χαρακτηριστικά των υπηρεσιών που παρέχονται στο Google Maps API:

- The GMap class: Πρόκειται για την βασική κλάση. Ένα αντικείμενο της κλάσης GMap αντιστοιχεί σε ένα χάρτη στην σελίδα μας. Μπορεί κανείς να δημιουργήσει όσα στιγμιότυπα αυτής της κλάσης επιθυμεί (ένα για κάθε χάρτη στη σελίδα). Όταν δημιουργούμε ένα νέο στιγμιότυπο χάρτη, καθορίζουμε ένα στοιχείο στη σελίδα το οποίο θα περιέχει το χάρτη. Ο χάρτης στη συνέχεια, σαν μέγεθος του χρησιμοποιεί το μέγεθος του στοιχείου που τον περιλαμβάνει εκτός και αν το ορίσουμε διαφορετικά. Η κλάση GMap παρέχει μεθόδους χειρισμού του κέντρου του χάρτη και του επιπέδου zoom, καθώς και μεθόδους για προσθήκη ή αφαίρεση διαφόρων overlays (όπως για παράδειγμα στιγμιότυπα των κλάσεων GMarker και GPolyline όπως θα δούμε). Επιπλέον, παρέχει μεθόδους που μας δίνουν τη δυνατότητα να ανοίξουμε ένα «παράθυρο πληροφοριών» το οποίο θα περιέχει διάφορες πληροφορίες πάνω στο χάρτη.
- Events: Με τη χρήση των event listeners μπορούμε να εισάγουμε δυναμικά στοιχεία στην εφαρμογή μας. Ένα αντικείμενο της κλάσης αυτής παρέχει έναν αριθμό από γεγονότα (events) και η εφαρμογή μας μπορεί να τα «ακούει» χρησιμοποιώντας τις στατικές μεθόδους GEvent.addListener ή GEvent.bind. Έτσι μπορεί το πρόγραμμα μας για παράδειγμα να εμφανίζει ένα μήνυμα ανάλογα με το κλικ ενός χρήστη πάνω στο χάρτη.
- The Info Window: Κάθε χάρτης έχει ένα μόνο «παράθυρο πληροφοριών», το οποίο εμφανίζει HTML περιεχόμενο σε ένα παράθυρο πάνω από το χάρτη. Το παράθυρο πληροφοριών μοιάζει με ένα «συννεφάκι» σε ένα βιβλίο comic. Αποτελείται από μια περιοχή με τα περιεχόμενα των πληροφοριών η οποία λεπταινεί στην άκρη και γίνεται σαν δείκτης που δείχνει σε ένα καθορισμένο σημείο πάνω στο χάρτη. Αν κάποιος έχει χρησιμοποιήσει το Google Maps ή το Google Local, τότε πολύ πιθανό να έχει δει ένα «παράθυρο πληροφοριών» όταν κάνει κλικ σε ένα εικονίδιο (marker). Άλλο χαρακτηριστικό των παραθύρων αυτών είναι ότι δεν μπορεί κανείς να εμφανίσει περισσότερα από ένα ταυτόχρονα σε ένα δοσμένο χάρτη αλλά μπορεί να μετακινήσει το

παράθυρο και να αλλάξει τα περιεχόμενά του αν αυτό είναι επιθυμητό.

Η βασική μέθοδος για ένα παράθυρο πληροφοριών είναι η `openInfoWindow`, η οποία παίρνει σαν είσοδο ένα σημείο και ένα HTML DOM element. Το παράθυρο πληροφοριών εμφανίζεται με το κείμενό του στο δοσμένο σημείο του χάρτη και εμφανίζει το DOM element στην περιοχή που περιλαμβάνει. Η μέθοδος `openInfoWindowHtml` είναι παρόμοια, αλλά παίρνει ένα HTML string σαν δεύτερο όρισμα αντί για ένα DOM element. Ομοίως, η `openInfoWindowXslt` παίρνει ένα σημείο, ένα XML DOM element, και το URL ενός XSLT αρχείου. Στη συνέχεια εφαρμόζει το μετασχηματισμό XSLT στο XML για να παράγει τα περιεχόμενα του παραθύρου. Η μέθοδος αυτή μεταφέρει το XSLT ασύγχρονα αν δεν έχει ήδη μεταφερθεί από τον browser του χρήστη.

Εκτός από τα παραπάνω, μπορούμε επίσης να εμφανίσουμε ένα παράθυρο πληροφοριών πάνω από ένα overlay όπως για παράδειγμα ένα εικονίδιο (marker). Για να το κάνουμε αυτό περνάμε ως τρίτο όρισμα ένα pixel offset μεταξύ του καθορισμένου σημείου και του κειμένου του παραθύρου πληροφοριών. Η κλάση `GMarker` (την οποία θα δούμε παρακάτω) επιτρέπει μεθόδους `openInfoWindow` οι οποίες χειρίζονται τα pixel offsets αυτόματα βασισμένες στο μέγεθος και σχήμα του εικονιδίου, και συνεπώς δε χρειάζεται να ανησυχεί ο προγραμματιστής για τον υπολογισμό των offsets στην εφαρμογή του.

- **Overlays:** Τα overlays είναι αντικείμενα πάνω στο χάρτη τα οποία είναι «συνδεδεμένα» σε γεωγραφικές συντεταγμένες, και έτσι μετακινούνται όταν ο χρήστης μετακινεί το χάρτη ή κάνει zoom σε αυτόν ή όταν αλλάζει ο τρόπος προβολής (η Google παρέχει διάφορους τρόπους προβολής ενός χάρτη, για παράδειγμα δορυφορική προβολή κτλ). Το Google Maps API περιλαμβάνει δύο τύπους overlays, τα markers που είναι εικονίδια πάνω στο χάρτη και τα polylines που είναι γραμμές που κατασκευάζονται από μία σειρά σημείων.
 - **Markers and Icons:** Ο `GMarker` constructor παίρνει σαν είσοδο ένα εικονίδιο και ένα σημείο και παράγει ένα μικρό σύνολο από events όπως για παράδειγμα το «κλικ» το οποίο μπορούμε να χειριστούμε στη συνέχεια στον κώδικά μας. Το δυσκολότερο κομμάτι στη δημιουργία ενός marker είναι ο καθορισμός του εικονιδίου. Αυτό είναι δύσκολο διότι είναι μεγάλος ο αριθμός των διαφορετικών εικόνων οι οποίες συνθέτουν ένα απλό εικονίδιο στο Maps API. Παρόλα αυτά, αν θέλει κανείς ένα γενικό εικονίδιο μπορεί να δημιουργήσει ένα `GMarker` χωρίς να καθορίσει ένα εικονίδιο.

Τα εικονίδια είναι συνήθως της μορφής πινέζας, με ένα tip το οποίο εμφανίζεται στη θέση που καθορίζεται από τον `GMarker` constructor. Κάθε

εικονίδιο έχει (τουλάχιστον) μια εικόνα στο προσκήνιο και μία εικόνα σαν σκιά. Υπάρχουν και περισσότερες λεπτομέρειες που αφορούν το πώς πρέπει να καθορίζονται τα εικονίδια και μπορούν να αναζητηθούν στο documentation του Google Maps API [GMA].

- Polylines: Ο GPolyline constructor παίρνει σαν είσοδο έναν πίνακα σημείων και δημιουργεί μια σειρά από ευθύγραμμα τμήματα τα οποία ενώνουν αυτά τα σημεία με τη σειρά που δίνονται. Μπορεί κανείς επίσης να καθορίσει το χρώμα, το πάχος και τη φωτεινότητα της γραμμής. Το χρώμα πρέπει να είναι σε δεκαεξαδική μορφή όπως στην HTML. Περισσότερες λεπτομέρειες για τα polylines μπορούν να αναζητηθούν στο documentation του Google Maps API [GMA].
- Controls: Για να χρησιμοποιήσουμε έλεγχο πάνω στο χάρτη, όπως μετακίνηση ή zoom ή οποιονδήποτε άλλο έλεγχο, υπάρχει η μέθοδος addControl. Το Maps API έχει ενσωματωμένους τους εξής ελέγχους που μπορεί κανείς να χρησιμοποιήσει στο χάρτη του:
 - GLargeMapControl: Ένας μεγάλου εύρους κίνησης/zoom έλεγχος που χρησιμοποιείται στο Google Maps.
 - GSmallMapControl: Ένας μικρότερος έλεγχος κίνησης/zoom που χρησιμοποιείται στο Google Local.
 - GSmallZoomControl: Ένας μικρός έλεγχος zoom που χρησιμοποιείται στο Google Maps για να εμφανίσει οδηγίες καθοδήγησης.
 - GMapTypeControl: Έλεγχος για να μπορεί ο χρήστης να αλλάζει τους διάφορους τύπους χαρτών (για παράδειγμα Map και Satellite).
- XML and RPC: Το Google Maps API προσφέρει μια «factory μέθοδο» για τη δημιουργία XmlHttpRequest αντικειμένων τα οποία «δουλεύουν» σε πρόσφατες εκδόσεις των Internet Explorer, Firefox, και Safari. Δίνεται επίσης η δυνατότητα να κάνουμε parsing σε ένα XML έγγραφο με τη στατική μέθοδο GXml.parse, η οποία παίρνει σαν είσοδο ένα XML string. Να σημειώσουμε μόνο ότι το Google Maps API δεν απαιτεί τη χρήση XML ή XmlHttpRequest για να λειτουργήσει καθώς βασίζεται αποκλειστικά σε ένα «καθαρό» JavaScript/DHTML API.

3

Εφαρμογές πλοήγησης για φορητές συσκευές

Στο κεφάλαιο αυτό μελετάμε τις διαθέσιμες εφαρμογές πλοήγησης για φορητές συσκευές που υπάρχουν σήμερα. Σκοπός μας είναι να μελετήσουμε τα διάφορα χαρακτηριστικά τους (χρέωση, χαρακτηριστικά, χρόνοι απόκρισης, ευχρηστία, GUI) και μέσα από τη μελέτη αυτή να οδηγηθούμε στο συμπέρασμα για το πώς θέλουμε να είναι ο client της εφαρμογής μας ο οποίος θα τρέχει σε φορητή συσκευή. Στη συνέχεια ακολουθεί η περιγραφή των διαφόρων συστημάτων πλοήγησης τα οποία τα χωρίσαμε σε δύο κατηγορίες, σε αυτά που τρέχουν σε Pocket PC / Palm και σε αυτά που τρέχουν σε Smartphone.

3.1 Συστήματα πλοήγησης GPS για Pocket PC/Palm

3.1.1 CoPilot Live Pocket PC

Το CoPilot Live Pocket PC 6 μετατρέπει έναν υπολογιστή τσέπης (pocket PC) σε ένα πλήρες σύστημα πλοήγησης. Η πλοήγηση σε δεδομένη διαδρομή, ο αυτόματος επανυπολογισμός διαδρομής, και οι φωνητικές οδηγίες οδήγησης είναι ορισμένες από τις υπηρεσίες τις οποίες παρέχει το CoPilot. Οι παραπάνω υπηρεσίες συνδυάζονται με διάφορα άλλα χαρακτηριστικά έτσι ώστε να κάνουν τα ταξίδια πολύ εύκολα. Ορισμένα από αυτά τα χαρακτηριστικά είναι τα εξής:

CoPILOT^{live} 6 POCKET PC



Εικόνα 9: CoPilot Live Pocket PC

Το CoPilot Live Pocket PC 6 επιτρέπει στο χρήστη να επιλέξει απεικόνιση μεταξύ δισδιάστατου χάρτη και τρισδιάστατου χάρτη αν θέλει να δει το δρόμο μπροστά σε τρισδιάστατη εικόνα. Επιπλέον, υπάρχει ένας wizard ο οποίος βοηθάει το χρήστη βήμα-βήμα στην εισαγωγή διευθύνσεων για τον καθορισμό της διαδρομής που θέλει να ακολουθήσει. Σε περίπτωση οδήγησης κατά τη διάρκεια της νύχτας, υπάρχει night mode το οποίο μειώνει την λάμψη από την ακτινοβολία που εκπέμπει το pocket PC και ξεκουράζει έτσι τα μάτια. Επιλογές για παράκαμψη ή επιλεκτική δρομολόγηση, επιτρέπουν στο χρήστη να αποφύγει δρόμους από τους οποίους δεν θέλει να περάσει και να αποφύγει απροσδόκητο μποτιλιάρισμα. Νέα χαρακτηριστικά περιλαμβάνουν:

- Βελτιωμένο 3D map display με απεικόνιση της απόστασης σε μίλια, μετακινούμενους χάρτες και βέλη κατευθύνσεων τα οποία δείχνουν το δρόμο.
- Finger touch menus και οθόνες επιλογών
- Υποστήριξη απεικόνισης Portrait, Landscape, Square και VGA.
- Προφίλ για δρομολόγηση τα οποία δίνουν στο χρήστη πλήρη έλεγχο στο τι δρόμους να προτιμά το CoPilot για δρομολόγηση (π.χ. μόνο κεντρικές αρτηρίες, αυτοκινητόδρομους κτλ)
- Δυνατότητα να δημιουργήσει ο χρήστης δικές του φωνητικές οδηγίες και να τις μοιραστεί με άλλους.
- Αλλαγή φωνής on the fly μεταξύ των περιλαμβανομένων ανδρικής ή γυναικείας είτε δημιουργημένων από το χρήστη.
- Εύκολη εισαγωγή σημείων ενδιαφέροντος και υποστήριξη μιας ευρείας γκάμας 3rd

party OV2 formats.

- Πλήρης συμβατότητα με το λειτουργικό σύστημα Windows Mobile 5

3.1.2 Microsoft Pocket Streets

Το Microsoft Pocket Streets 2005 μετατρέπει ένα Pocket PC το οποίο βασίζεται σε Microsoft Windows Mobile λειτουργικό σύστημα ή το κινητό τηλέφωνο (smart phone) ενός χρήστη στον προσωπικό του οδηγό σε χιλιάδες πόλεις της βόρειας Αμερικής και της δυτικής Ευρώπης. Το Pocket Streets 2005 προσφέρει οδηγίες πλοήγησης με χάρτη για συσκευές που έχουν Windows Mobile λειτουργικό. Το λογισμικό αυτό δίνει τη δυνατότητα στο χρήστη να δει χάρτες διαφόρων περιοχών, να περιπλανηθεί και να εντοπίσει σημεία ενδιαφέροντος όπως εστιατόρια, ATMs, εμπορικά κέντρα, στάσεις μεταφορικών μέσων, τουριστικά μέρη και άλλα.



Εικόνα 10: Pocket Streets 2005

Έχοντας συνδεδεμένη μια συσκευή GPS με το Pocket PC ή το κινητό τηλέφωνο (smartphone) μπορεί κανείς να δει την θέση του πάνω στο χάρτη του Pocket Streets. Το λογισμικό ελέγχει τη θέση αυτή κάθε δευτερόλεπτο και την απεικονίζει πάνω στο χάρτη. Μπορεί κανείς να έχει το χάρτη κεντραρισμένο μόνιμα στην τρέχουσα θέση του και να βλέπει τις συντεταγμένες του ως γεωγραφικό μήκος και πλάτος καθώς και την ταχύτητα, κατεύθυνση και υψόμετρο.

Το Pocket Streets 2005 υποστηρίζει συσκευές GPS οι οποίες είναι συμβατές με το πρωτόκολλο National Marine Electronics Association (NMEA) και το input/output format τους υποστηρίζει το NMEA 0183 έκδοση 2.0 ή επόμενη. Το Pocket Streets έχει ενσωματωμένους περίπου 300 χάρτες της βορείου Αμερικής και 275 της δυτικής Ευρώπης.

Οι απαιτήσεις συστήματος για το Pocket PC είναι :

- Microsoft Windows CE 3.0 ή επόμενη έκδοση.

- 500 KB διαθέσιμης μνήμης για την εγκατάσταση του λογισμικού
- Επιπλέον μνήμη (περίπου 1 MB ανά χάρτη πόλης) για την εγκατάσταση των χαρτών
- Μια συσκευή GPS η οποία να υποστηρίζει NMEA 0183 2.0 ή επόμενη έκδοση ρυθμισμένο να χρησιμοποιεί COM (σειριακή) θύρα.

3.1.3 *Destinator*

Το Destinator είναι ένα από τα γνωστότερα λογισμικά πλοήγησης για φορητές συσκευές. Η τελευταία έκδοση του λογισμικού είναι το Destinator 6 το οποίο βγήκε στην αγορά τον Αύγουστο του 2006. Το λογισμικό πλοήγησης απευθύνεται σε διάφορες κατηγορίες χρηστών. Οι χρήστες αυτοί είναι άνθρωποι οι οποίοι χρειάζεται να μετακινούνται πολύ, όπως είναι οι πωλητές, οι διανομείς, οι άνθρωποι που ταξιδεύουν συχνά, οι οδηγοί και οι τουρίστες.

Το Destinator 6 για PDAs περιλαμβάνει την ακόλουθη λειτουργικότητα:

- Βελτιωμένη διεπαφή χρήστη: Το user interface της εφαρμογής χαρακτηρίζεται ως διαισθητικό αφού περιλαμβάνει μεγάλα κουμπιά αφής (touch screen buttons) για τα μενού, ένα πλήρες πληκτρολόγιο για την εισαγωγή διευθύνσεων και πολύ καλή ποιότητα από 2Δ / 3Δ χάρτες. Η αυξημένη ευκρίνεια της οθόνης και του χάρτη κάνει ακόμα ευκολότερο το να δει ο χρήστης τις πληροφορίες για την τρέχουσα θέση του και τα ονόματα των οδών.
- Ένα Gigabyte χάρτες των ΗΠΑ και Καναδά.
- Δυνατότητα πρόσβασης σε πληροφορίες πραγματικού χρόνου: Το Destinator 6 δίνει τη δυνατότητα στο χρήστη να έχει πρόσβαση σε πληροφορίες πραγματικού χρόνου όπως είναι η κυκλοφοριακή συμφόρηση στους δρόμους, ο καιρός, διάφορα γεγονότα που πιθανόν συμβαίνουν και διάφορα άλλα.
- SMS πλοήγηση: Με το Destinator 6, οι χρήστες μπορούν εύκολα να στέλνουν στους φίλους και την οικογένειά τους ένα γραπτό μήνυμα με την τρέχουσα θέση τους. Όταν λάβει κανείς το μήνυμα μπορεί να πλοηγηθεί στη θέση του αποστολέα με ένα μόνο κλικ.
- Έξυπνη αναζήτηση: Οι χρήστες μπορούν να εισάγουν μέρος μιας διεύθυνσης μόνο και η συσκευή θα χρησιμοποιήσει έναν αλγόριθμο γρήγορης αναζήτησης ο οποίος αυτόματα συμπληρώνει τη διεύθυνση εισαγωγής και δείχνει τα αποτελέσματα σε μια λίστα.

Μερικά άλλα χαρακτηριστικά του Destinator 6 περιλαμβάνουν:

- Χάρτες 2Δ / 3Δ υψηλής ποιότητας που περιλαμβάνουν ονόματα οδών
- Όψεις μέρας και νύχτας με φωνητικές οδηγίες πλοήγησης και οπτική καθοδήγηση
- Εύκολη πρόσβαση σε εκατομμύρια προ-εγκατεστημένα σημεία ενδιαφέροντος
- Πλοήγηση σε διαφορετικές πολιτείες (για την Αμερική) ή διαφορετικές χώρες χωρίς αλλαγή χάρτη.
- Υποστήριξη Windows Mobile 2003, Windows Mobile 5 (WM5) & WinCE 5

3.1.4 TomTom Navigator

Το λογισμικό TomTom NAVIGATOR 5 μετατρέπει το PDA του χρήστη σε έναν προσωπικό βοηθό ταξιδιού στο δρόμο. Προσφέρει στους χρήστες χαρακτηριστικά για εύκολο και χωρίς άγχος ταξίδι με το αυτοκίνητο, με το ποδήλατο ακόμη και με τα πόδια. Διαθέτει βελτιωμένη λειτουργικότητα όπως τα νέα TomTom PLUS services, σημεία ενδιαφέροντος της επιλογής του χρήστη, αναβαθμισμένο σύστημα πλοήγησης για τον καθορισμό ταξιδιών με περισσότερους του ενός προορισμούς (ενδιάμεσες στάσεις) και ενσωμάτωση του συστήματος στον φάκελο επαφών του PDA. Το TomTom NAVIGATOR 5 διατίθεται με τους τελευταίους χάρτες της TeleAtlas.



Εικόνα 11: TomTom NAVIGATOR software

Ορισμένα από τα χαρακτηριστικά του λογισμικού TomTom NAVIGATOR είναι:

- EASY
 - Ευκολία χρήσης: Απλή λειτουργία οθόνης αφής, διαισθητικό interface

- Πλοήγηση μέσω επαφών: Πλοήγηση απευθείας από το βιβλίο διευθύνσεων του PDA, ενσωματώνοντας το λογισμικό του TomTom στο μενού του PDA.
- Απεικόνιση διαδρομής: Δυνατότητα ελέγχου της υπολογισμένης διαδρομής από το χρήστη πριν ακόμη ξεκινήσει να οδηγή.
- SMART
 - Δυναμική μπάρα status: Δυνατότητα βελτίωσης αναγνωσιμότητας διαμορφώνοντας τη μπάρα status.
 - Συμβουλές: Μέσω των χρήσιμων συμβουλών στην οθόνη ο χρήστης ανακαλύπτει όλα όσα το TomTom NAVIGATOR 5 έχει να προσφέρει.
 - Περιεκτικό guided tour: Δυνατότητα παρακολούθησης όλων των χαρακτηριστικών του λογισμικού εν συντομία.
 - Μικρή και ελαφριά ασύρματη συσκευή GPS
 - Πολυγλωσσία: Φωνητικές οδηγίες (γυναικείες και ανδρικές φωνές) σε περισσότερες από 30 γλώσσες. Το user interface σήμερα περιλαμβάνει 18 γλώσσες.
 - Εκατομμύρια σημεία ενδιαφέροντος: Parking, βενζινάδικα, ξενοδοχεία, εστιατόρια κτλ με δυνατότητα αυτόματης ειδοποίησης κατά την προσέγγιση στο σημείο.
 - Compass Mode: Απεικόνιση πυξίδας στην οθόνη για ακόμη καλύτερο προσανατολισμό κατά την οδήγηση.
- CONNECTED
 - TomTom PLUS services: Οι υπηρεσίες αυτές παρέχουν στο χρήστη πληροφορίες για την τρέχουσα κυκλοφοριακή συμφόρηση στους δρόμους, τον καιρό, καθώς και πληροφορίες από κάμερες ασφαλείας (μόνο για Αγγλία, Γερμανία, Γαλλία και Ολλανδία), επιπλέον φωνές για φωνητικές οδηγίες, σημεία ενδιαφέροντος και πολλά άλλα.
 - Πλοήγηση μέσω επαφών: Πλοήγηση απευθείας από το βιβλίο διευθύνσεων του PDA, ενσωματώνοντας το λογισμικό του TomTom στο μενού του PDA.

Το λογισμικό είναι συμβατό με το PocketPC 2002, το Windows Mobile 2003, το Windows Mobile 2003 Second Edition, και το Palm OS.

3.2 Συστήματα πλοήγησης GPS για κινητά τηλέφωνα (Smartphone)

3.2.1 CoPilot Live Smartphone

Το CoPilot Live Smartphone δίνει δυνατότητα GPS πλοήγησης με πολλά χαρακτηριστικά για κάθε κινητό τηλέφωνο που υποστηρίζει Windows Mobile λογισμικό σύστημα. Χρειάζεται επίσης συσκευή GPS συμβατή με το πρωτόκολλο NMEA.



Εικόνα 12: CoPilot Live Smartphone

Το CoPilot Live εκμεταλλεύεται πλήρως την οθόνη, τον επεξεργαστή και το ηχείο του κινητού τηλεφώνου για να παρέχει στους οδηγούς ακριβείς και αξιόπιστες οδηγίες οδήγησης με λεκτικές ειδοποιήσεις και ευκρινείς απεικονίσεις της διαδρομής στην οθόνη σε κάθε στροφή. Ο χρήστης απλά εισάγει μια οδό, ταχυδρομικό κώδικα, πόλη, επαφή Outlook ή σημείο ενδιαφέροντος και στη συνέχεια ξεκινά η πλοήγηση. Σε περίπτωση που κατά την πλοήγηση ο χρήστης βγει εκτός διαδρομής το CoPilot Live θα υπολογίσει άμεσα μια νέα διαδρομή με νέες οδηγίες. Σε περίπτωση πάλι που ξαφνικά βρεθεί κανείς σε κυκλοφοριακή συμφόρηση σε μια άγνωστη για αυτόν πόλη το CoPilot θα τον βοηθήσει να την αποφύγει περνώντας από άλλους δρόμους. Επιπλέον το CoPilot θα βοηθήσει το χρήστη να αποφύγει σημεία όπου γίνονται έργα στο δρόμο.

Ορισμένα από τα χαρακτηριστικά του λογισμικού περιλαμβάνουν:

- Βελτιωμένο 3D map display με απεικόνιση της απόστασης σε μίλια, μετακινούμενους χάρτες και βέλη κατευθύνσεων τα οποία δείχνουν το δρόμο.
- Προφίλ για δρομολόγηση τα οποία δίνουν στο χρήστη πλήρη έλεγχο στο τι δρόμους να προτιμά το CoPilot για δρομολόγηση (π.χ. μόνο κεντρικές αρτηρίες, αυτοκινητόδρομους κτλ)
- Δυνατότητα να δημιουργήσει ο χρήστης δικές του φωνητικές οδηγίες και να τις μοιραστεί με άλλους.
- Αλλαγή φωνής on the fly μεταξύ των περιλαμβανομένων ανδρικής ή γυναικείας είτε δημιουργημένων από το χρήστη.
- Εύκολη εισαγωγή σημείων ενδιαφέροντος και υποστήριξη μιας ευρείας γκάμας 3rd party OV2 formats.
- Πλήρης συμβατότητα με το λειτουργικό σύστημα Windows Mobile 5

3.2.2 Microsoft Pocket Streets

Για το Microsoft Pocket Streets για κινητό τηλέφωνο ισχύουν όλα όσα έχουμε γράψει για την αντίστοιχη εφαρμογή για pocket PC. Υπάρχουν όμως μερικές επιπλέον πληροφορίες σχετικές με τις απαιτήσεις συστήματος. Έτσι, οι απαιτήσεις συστήματος για ένα smartphone είναι :

- Ένα smartphone βασισμένο σε Microsoft Windows Mobile 2002, 2003 ή επόμενη έκδοση του λειτουργικού συστήματος.
- 400 KB διαθέσιμη μνήμη για την εγκατάσταση του λογισμικού
- Επιπλέον μνήμη (περίπου 1 MB ανά χάρτη πόλης) για την εγκατάσταση των χαρτών (συνιστάται η χρήση storage card).
- Υποστήριξη Bluetooth από το smartphone το οποίο τρέχει Windows Mobile 2003 ή επόμενη έκδοση καθώς και μια συσκευή Bluetooth GPS η οποία υποστηρίζει το πρωτόκολλο NMEA 0183 2.0 ή επόμενη έκδοση.

3.2.3 Destinator SP - Navigation for Smartphones

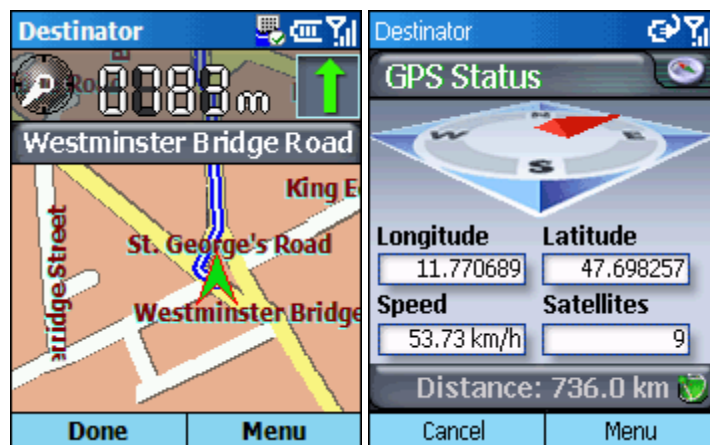
Το Destinator SP είναι ένα λογισμικό πλοήγησης για κινητά τηλέφωνα. Προσφέρει πλήρεις οδηγίες πλοήγησης και σχεδιασμό ταξιδιού. Ο χρήστης μπορεί να αποθηκεύσει εκτεταμένα δεδομένα χαρτών, να σχεδιάσει πολύπλοκες διαδρομές και να έχει οδηγίες οδήγησης ακόμη και κατά τη διάρκεια τηλεφωνικών κλήσεων. Υπάρχουν διάφορες κατηγορίες χρηστών, για παράδειγμα αυτοί που οδηγούν αυτοκίνητο, μοτοσικλέτα, ποδήλατο ή πεζοί χρήστες.



Εικόνα 13: Destinator SP

Τα κυριότερα χαρακτηριστικά του λογισμικού είναι:

- On-board routing χωρίς επιπλέον χρέωση
- Απλή εγκατάσταση
- Λεπτομερής πλοήγηση με καθαρή οπτική και φωνητική καθοδήγηση
- Διαισθητικές όψεις («Στροφή-στροφή», «2Δ χάρτης», «3Δ χάρτης») για εύκολο προσανατολισμό.
- Σχεδίαση διαδρομών χωρίς GPS
- Αποστολή SMS της τρέχουσας θέσης ή του προορισμού
- Υπολογισμός διαδρομής από τις συντεταγμένες που ελήφθησαν από SMS
- Μετάδοση διευθύνσεων και συντεταγμένων σε κάθε SMS-enabled κινητό τηλέφωνο
- Άμεση πλοήγηση σε αποθηκευμένες «αγαπημένες» διευθύνσεις ή επαφές Outlook.
- Προαιρετική δυνατότητα απεικόνισης οδηγιών πλοήγησης κατά τη διάρκεια φωνητικής κλήσης
- Γρήγορος επανυπολογισμός διαδρομής
- Δυνατότητα αποθήκευσης πολλαπλών χαρτών σε μία κάρτα μνήμης



Εικόνα 14: Οθόνες του Destinator SP

Άλλα βασικά χαρακτηριστικά του Destinator SP είναι:

Εύκολος σχεδιασμός ταξιδιού

- Δυνατότητα προ-σχεδιασμού ταξιδιού προς κάθε προορισμό, επιλογή από μια ποικιλία διαδρομών, αποφυγή συμφόρησης ή διοδίων, δρομολόγηση μέσω του αγαπημένου εστιατορίου του χρήστη ή επιλογή ενός νέου σημείου ενδιαφέροντος από μια λίστα από χιλιάδες σημεία
- Απεικόνιση απόστασης και χρόνου που απομένει προς τον προορισμό

Συνομιλία κατά τη διάρκεια του ταξιδιού

- Προαιρετική οπτική απεικόνιση οδηγιών πλοήγησης κατά τη διάρκεια φωνητικής κλήσης.

Ο χρήστης μπορεί να επιλέξει:

- Πλοήγηση μέσω παρακαμπτηρίων οδών
- Πολύ-γλωσσική φωνητική και οπτική καθοδήγηση
- Πλοήγηση απευθείας μέσω της λίστας επαφών

Αποστολή SMS σε φίλους

- Δυνατότητα πρόσκλησης φίλων του χρήστη ή να μοιραστεί μαζί τους διαδρομές.
- Οδηγίες μέσω SMS κατευθείαν στο κινητό τηλέφωνο

Επιλογή όψης

- Πολύ καλό interface φιλικό προς το χρήστη

- 2Δ. 3Δ, «Στροφή-Στροφή» ή κείμενο

Πλοήγηση ανά τον κόσμο

- Ευρεία επιλογή χαρτών για τέσσερις ηπείρους
- Διαθέσιμοι οι τελευταίοι χάρτες για τους καθημερινούς προορισμούς του χρήστη και όχι μόνο.

Οι απαιτήσεις συστήματος για το λογισμικό Destinator SP είναι:

- Microsoft Windows Mobile Smartphone 2003 SE
- Ελάχιστη μνήμη 32 MB RAM
- Εξωτερική κάρτα μνήμης: SD/MMC/MiniSD/RS-MMC
- Διεπαφή για κεραία GPS: Bluetooth, USB ή σειριακή
- Εγγραφή προϊόντος και ενεργοποίηση μέσω Internet

4

Στόχος

Στο παρόν κεφάλαιο γίνεται μια προσπάθεια να εξάγουμε χρήσιμα συμπεράσματα από την παραπάνω θεωρητική μελέτη την οποία κάναμε. Βασισμένοι στα συμπεράσματα αυτά, θα πραγματοποιήσουμε το σχεδιασμό της εφαρμογής μας τόσο για την πλευρά του server όσο και για την πλευρά του client.

Η ανάλυση που θα κάνουμε μπορεί να χωριστεί στις παρακάτω θεματικές ενότητες:

A. Εφαρμογή (client, server side)

- Ποια είναι τα πρότυπα που υλοποιούνται από το σύστημά μας.
- Ποιες τεχνολογίες χρησιμοποιούνται από το σύστημά μας.
- Υπηρεσίες – Ποιες υπηρεσίες είναι διαθέσιμες από το σύστημα.
- Πλατφόρμες – Σε ποιες πλατφόρμες μπορεί να τρέξει το σύστημα που καθορίζουμε.

B. Χρήστες

- Ποιοι χρήστες μπορούν να χρησιμοποιούν το σύστημα. – προφίλ χρηστών.

Γ. Εμπορική εκμετάλλευση

- Χρέωση – Το προϊόν είναι εμπορικά διαθέσιμο; - Μοντέλο χρέωσης.

4.1 Εφαρμογή

Έχοντας μελετήσει διάφορα θέματα σχετικά με τα πρότυπα από τον OGC και τον ISO για τη διάθεση χωρικής πληροφορίας, καθώς και ήδη εμπορικά διαθέσιμες τέτοιες εφαρμογές, είμαστε έτοιμοι να προτείνουμε ένα πλαίσιο στο οποίο μπορούμε να βασιστούμε για τη διάθεση γεωγραφικών υπηρεσιών. Σκοπός είναι να κατασκευαστεί ένα σύστημα το οποίο θα παρέχει στους χρήστες του τη δυνατότητα να ζητούν και να λαμβάνουν γεωγραφικά δεδομένα ανάλογα με τις ανάγκες τους (π.χ. πλοήγηση). Με τον όρο γεωγραφικά δεδομένα αναφερόμαστε σε χάρτες περιοχών, υπολογισμούς διαφόρων διαδρομών, απεικόνιση σημείων ενδιαφέροντος και άλλες σχετικές υπηρεσίες. Επιπλέον, έχοντας μελετήσει τις εμπορικά υφιστάμενες εφαρμογές πλοήγησης οι οποίες τρέχουν σε φορητές συσκευές μπορούμε να προτείνουμε ένα πλαίσιο για το τι χαρακτηριστικά θα πρέπει να έχει ο δικός μας navigation client ο οποίος θα καταναλώνει τα web services του server.

4.1.1 Πρότυπα

Το πλαίσιο που προτείνουμε κάνει χρήση ανοικτών προτύπων που έχουν φτιαχτεί για γεωγραφικές εφαρμογές, όπως αυτά που προτείνονται από τον OGC ή τον ISO και αναλύονται στο Παράρτημα του παρόντος τόμου. Εκμεταλλευόμαστε έτσι όλα εκείνα τα πλεονεκτήματα που έχουμε από τη χρήση των προτύπων αυτών όπως για παράδειγμα η διαλειτουργικότητα. Προτείνεται η χρησιμοποίηση του προτύπου OpenGIS Location Services (OpenLS) το οποίο κάνει έμμεση χρήση του GML (Geography Markup Language), ως το πιο κατάλληλο για ένα σύστημα πλοήγησης το οποίο θα τρέχει από φορητές συσκευές και θα παίρνει τα δεδομένα του από έναν server μέσω web services. Βέβαια, θα πρέπει να γίνει μια μικρή τροποποίηση στο πρότυπο αυτό ώστε αυτό να μπορεί να χρησιμοποιηθεί για SOAP / XML web services διότι τώρα αυτό αναφέρεται σε XML services. Τη διαφορά αυτή θα την εξηγήσουμε στη συνέχεια.

4.1.2 Server Side

Οι υπηρεσίες τις οποίες θα πρέπει να παρέχει ο server είναι τουλάχιστον οι υπηρεσίες πυρήνα όπως αυτές αναφέρονται στο πρότυπο OpenGIS Location Services (OpenLS) του OGC και είναι οι εξής:

- **Directory Service:** Πρόκειται για υπηρεσία online καταλόγου η οποία δίνει τη δυνατότητα στο χρήστη να βρει ένα σημείο κοντά στην τρέχουσα θέση του είτε

γενικότερα ένα σημείο ενδιαφέροντος (Point of Interest).

- Gateway Service: Πρόκειται για μία υπηρεσία μέσω της οποίας ο χρήστης μπορεί να βρει την τρέχουσα θέση του είτε τη θέση κάποιου φίλου του μέσω του κινητού του τηλεφώνου και χωρίς χρήση συσκευής GPS. Πρόκειται δηλαδή για έναν εναλλακτικό τρόπο προσδιορισμού της θέσης ενός χρήστη.
- Location Utility Service (Geocoder/Reverse Geocoder): Πρόκειται για μία υπηρεσία η οποία αναλαμβάνει να κάνει την γεω-κωδικοποίηση και το αντίστροφο, να βρίσκει δηλαδή συντεταγμένες μιας διεύθυνσης εισόδου του χρήστη και το αντίστροφο.
- Presentation Service: Πρόκειται για μια υπηρεσία παρουσίασης ενός τμήματος χάρτη με layers της επιλογής του χρήστη. Επιστρέφει ένα χάρτη σε format επιλογής του χρήστη, αν αυτό υποστηρίζεται με βάση συντεταγμένες επιλογής του χρήστη.
- Route Service: Ίσως η σημαντικότερη υπηρεσία για την εφαρμογή μας. Επιστρέφει τη διαδρομή που πρέπει να ακολουθήσει ο χρήστης από ένα σημείο σε ένα άλλο μαζί με οδηγίες για το πώς θα πάει στον προορισμό του, πληροφορίες για την υπολογισμένη διάρκεια και απόσταση. Η υπηρεσία αυτή είναι η βάση για να γίνει πλοήγηση στο χρήστη μέσω του client.

Να σημειώσουμε στο σημείο αυτό ότι τα γεωγραφικά δεδομένα τα οποία θα επιστρέφει ο server μέσω των παραπάνω services δεν θα είναι κατ' ανάγκη αποθηκευμένα στον ίδιο το server. Μπορεί να είναι αποθηκευμένα σε εξωτερικό GIS server και να τα λαμβάνει από εκεί. Ακολουθεί σχήμα που περιγράφει ακριβώς τη μορφή του συστήματος που προτείνουμε:



Εικόνα 15: Μορφή του προτεινόμενου συστήματος

4.1.2.1 Αρχεία εικόνας

Σε τι format θα πρέπει να επιστρέφεται η τελική εικόνα (χάρτης) που παράγεται από την υπηρεσία Presentation Service; Η απάντηση σε αυτό το ερώτημα όπως γνωρίζουμε δεν είναι μία. Η Presentation Service μπορεί να αποδώσει διάφορους τύπους εικόνας όπως PNG, GIF ή

JPEG, ή ακόμη και SVG. Μετά από σχετική αναζήτηση πληροφοριών [SVO][SVA] για τα πρότυπα αυτά καταλήξαμε στα εξής αποτελέσματα τα οποία συγκεντρώνονται στον παρακάτω πίνακα.

SVG	JPEG / PNG / GIFF
Είναι scalable vector format. Συνεπώς στα αντικείμενα σχεδίασης μπορούμε να κάνουμε zoom ή να τους αλλάξουμε μέγεθος αυθαίρετα χωρίς απώλεια ποιότητας.	Είναι raster formats. Συνεπώς οποιαδήποτε αλλαγή γίνει στο μέγεθος της εικόνας έχουμε απώλεια ποιότητας.
Έχουν plain text format και συνεπώς είναι searchable και μπορεί κανείς να επιλέγει κείμενο πάνω στην εικόνα	Δεν είναι searchable και δεν μπορεί κανείς να επιλέγει κείμενα που εμφανίζονται στην εικόνα
Είναι interactive – Αφού σχεδιαστεί ο χάρτης, μπορεί κανείς να μεταβάλλει κάποια χαρακτηριστικά του	Είναι static – Αφού σχεδιαστεί ο χάρτης είναι μια στατική εικόνα και δεν επιδέχεται αλλαγών
Έχουν γενικά μικρότερο μέγεθος από ότι τα raster formats. Επιδέχονται και συμπίεση.	Επιδέχονται συμπίεση μειώνοντας το μέγεθος τους. Παρόλα αυτά έχουν γενικά μεγαλύτερο μέγεθος από ότι τα vector formats.
Υποστηρίζουν πολύπλοκα animations και μετασχηματισμούς	Δεν υποστηρίζουν πολύπλοκα animations και μετασχηματισμούς
Είναι γενικά μη συμβατά. Δηλαδή, για να δει κανείς SVG αρχεία, χρειάζεται να έχει συνήθως ένα ξεχωριστό plug-in διότι οι περισσότεροι browsers δεν υποστηρίζουν SVG σήμερα.	Είναι πολύ συμβατά. Όλοι οι browsers τα υποστηρίζουν.

SVG	JPEG / PNG / GIFF
Τα σχέδια είναι χωρισμένα σε κομμάτια. Για παράδειγμα, συνήθως αυτό που βλέπει ο χρήστης στην οθόνη μπορεί να είναι διασκορπισμένο σε πολλές τοποθεσίες στο internet. Έτσι μπορεί αλλού να είναι το background κτλ. Αυτό κάνει δύσκολη μερικές φορές την αποθήκευση των εικόνων και την πρόσβαση σε αυτές offline.	Η εικόνα που βλέπουμε στην οθόνη είναι αυτούσια, δεν είναι διασκορπισμένη σε διάφορες τοποθεσίες και μπορούμε εύκολα να την αποθηκεύσουμε και να τη δούμε offline.

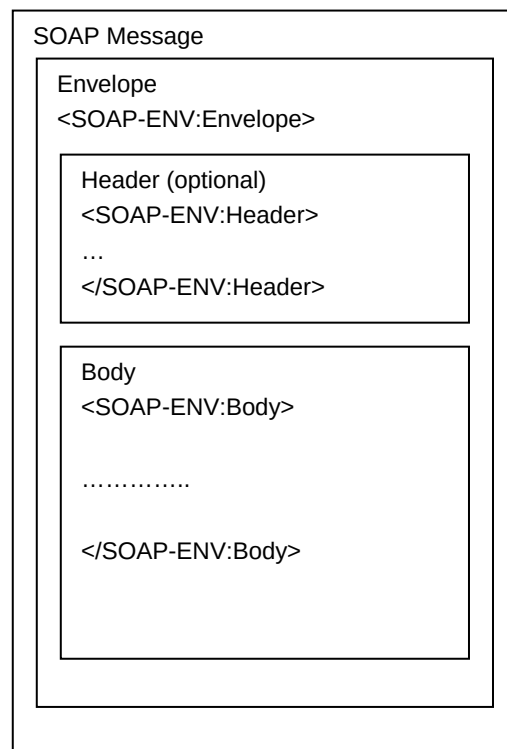
Με βάση τον παραπάνω πίνακα μπορούμε να οδηγηθούμε στο συμπέρασμα ότι η SVG απεικόνιση είναι γενικά καλύτερη για τις γεωγραφικές εφαρμογές. Παρόλα αυτά, οι άλλες τεχνολογίες είναι συγκριτικά πιο ώριμες και συνεπώς δεν μπορούμε να τις απορρίψουμε. Το σύστημά μας λοιπόν πρέπει να υποστηρίζει και τους δύο τύπους απεικόνισης, αλλά καθώς σταδιακά στο μέλλον η SVG θα γίνεται πιο συμβατή και ώριμη τεχνολογία, μπορεί να καθιερωθεί ως αποδεκτό πρότυπο για την απεικόνιση χαρτών από γεωγραφικά συστήματα.

4.1.2.2 SOAP vs XML

Στην έκταση της παρούσας εργασίας έγινε αναφορά και μια στοιχειώδης ανάλυση των γεωγραφικών δικτυακών υπηρεσιών (Geo Web Services) που ήδη υπάρχουν στην αγορά και μονοπωλούν το αγοραστικό ενδιαφέρον του εκάστοτε «καταναλωτή». Η αναφορά σε αυτές τις εφαρμογές είθισται να γίνεται –ατυχώς και γενικευμένα– με τον όρο Web Services, ανεξάρτητα από την τεχνολογία που χρησιμοποιήθηκε για την υλοποίηση και ανάπτυξη της εκάστοτε εφαρμογής.

Στο σημείο αυτό θα πρέπει να επισημάνουμε το γεγονός ότι ο όρος Web Service [JWS] προϋποθέτει την υλοποίηση αυστηρά με χρήση των τεχνολογιών SOAP (Simple Object Access Protocol) και WSDL (Web Service Description Language). Η τεχνολογία SOAP είναι βέβαια XML-based, αλλά επιπλέον επιτρέπει την κατά παραγγελίαν διάρθρωση του κειμένου XML που μεταφέρει την πληροφορία μεταξύ client και server πάνω από HTTP, χωρίς αυτή να υπόκειται στα προκαθορισμένα πρότυπα που χρησιμοποιεί το XML-RPC. Το SOAP μπορεί να δρά σαν serializer: η δομή του αντικειμένου μετατρέπεται προς και από μία σειριακή ροή δεδομένων πάνω στο «σύρμα» μετάδοσης. Η μετατροπή γίνεται σύμφωνα με κανόνες και πρότυπα βασισμένα στην XML και από κοινού συμφωνημένα σε client και server (το πρωτόκολλο SOAP). Μία αναφορά στην διάρθρωση της κωδικοποίησης σε XML

περιλαμβάνεται στο μήνυμα, έτσι ώστε ο δέκτης προκαταρκτικά να γνωρίζει και αποφασίζει αν και πώς θα ασχοληθεί με το αυτό. Γενικά το SOAP διαχωρίζει σαφώς σώμα (body - περιεχόμενο) και το φάκελο (envelope) πού το περιλαμβάνει, και τα δύο εκφρασμένα σε XML. Ένα envelope μπορεί να έχει προαιρετικά μία επικεφαλίδα (header) και body με πολλά άμεσα στοιχεία – παιδιά, τα οποία αποτελούν τα λεγόμενα body blocks. Το SOAP μπορεί να χρησιμοποιηθεί και μέσα στα πλαίσια του RPC (SOAP-RPC.) σαν (πιο εξελιγμένο και δυνατότερο του XML-RPC) και για γενική μεταφορά πληροφορίας άσχετης με τις εξ αποστάσεως κλήσεις (SOAP Messaging).



Εικόνα 16: SOAP Message

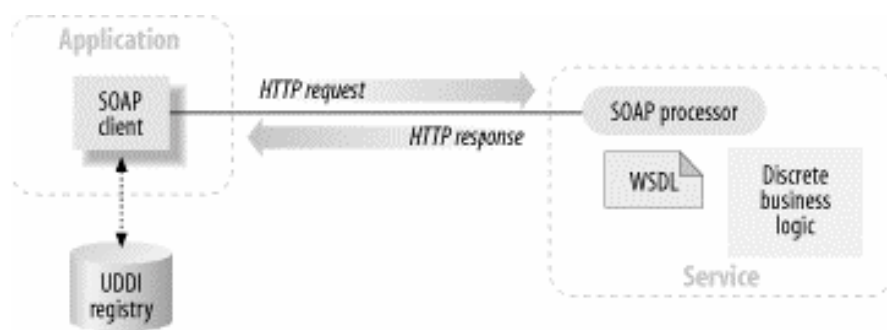
Είναι αλήθεια ότι η τεχνολογία των Web Services έχει αναπτυχθεί με άλματα και οι τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίησή τους ακολούθησαν την ίδια πορεία ανάπτυξης. Αυτές όμως που εδραιώνονται στο χώρο των web services θα πρέπει να πληρούν μια βασική προϋπόθεση και αυτή είναι η υποστήριξη διαλειτουργικότητας (interoperability). Τα τελευταία χρόνια η διαλειτουργικότητα [JWS] έγινε πραγματικότητα με την εισαγωγή του SOAP σε συνδυασμό με τις WSDL (Web Service Description Language) και UDDI (Universal Description, Discovery and Integration).

- Με το SOAP προσφέρεται ένας προτυποποιημένος τρόπος για τη δημιουργία

δομημένου πακέτου με σκοπό τη μεταφορά αρχείων XML μέσω ποικιλίας προτυποποιημένων τεχνολογιών Internet όπως SMTP, HTTP και FTP. Ορίζει επίσης πρότυπα κωδικοποίησης και binding για την κωδικοποίηση κλήσεων που δεν αποτελούν XML RPC. Με αυτό τον τρόπο ετερογενείς χρήστες (πελάτες ή εξυπηρετητές) μπορούν εύκολα να αποκτήσουν διαλειτουργικότητα.

- Η WSDL είναι μια XML τεχνολογία που περιγράφει τη διεπαφή ενός web service σύμφωνα με κάποιο πρότυπο. Η WSDL είναι αρμόδια για την προτυποποίηση του τρόπου με τον οποίο ένα web service τις παραμέτρους εισόδου, εξόδου κτλ. Με τον τρόπο αυτό επιτρέπει στον «πελάτη» να μπορεί να καταλαβαίνει αυτόματα πώς να αλληλεπιδράσει με το εκάστοτε web service.
- Το UDDI προσφέρει έναν κατάλογο παγκόσμιας εμβέλειας με καταγεγραμμένα web services για λόγους που εξυπηρετούν ανάγκες ολοκλήρωσης (integration), ανεύρεσης (discovery) διαθέσιμων web services που ενδιαφέρουν προγραμματιστές κτλ, αλλά και για λόγους διαφήμισης.

Κάθε μια από τις παραπάνω τεχνολογίες προσφέρει ένα πρότυπο για το επόμενο βήμα στην ανάπτυξη ενός web service. Το ζητούμενο όμως στο χώρο των web services είναι η δυναμική ολοκλήρωση των υπηρεσιών που προσφέρουν «άγνωστες» εταιρείες. Αυτό για να γίνει προϋποθέτει τη συνδυασμένη εμπλοκή των τεχνολογιών SOAP, WSDL και UDDI. Στο σχήμα που ακολουθεί φαίνεται η σχέση μεταξύ των τριών τεχνολογιών που αναφέρθηκαν.



Εικόνα 17 Σχέση μεταξύ των τεχνολογιών SOAP, WSDL, UDDI

Η αναφορά που έγινε στα web services που κυκλοφορούν στη σημερινή αγορά, αφορά σε web services (MapPoint, ViaMichelin Web Services, Google Maps) που δεν ακολουθούν τον ακριβή ορισμό του όρου. Μια σωστή περιγραφή τους θα μπορούσε να γίνει με τον όρο XML Services. Η βασική διαφορά που εντοπίζεται μεταξύ Web και XML Services είναι ότι τα

δεύτερα δεν χρησιμοποιούν τον φάκελο (envelope) που ορίζει το SOAP, αλλά γίνεται μεταφορά XML μέσω του διαδικτυακού πρωτοκόλλου HTTP.

Η εφαρμογή που προτείνεται θα ακολουθήσει τον ακριβή ορισμό των Web Services και θα σχεδιαστεί / υλοποιηθεί βάσει των προδιαγραφών που αυτό προβλέπει.

Η αγορά έχει έως τώρα υποστηρίξει την «τεχνολογία» των XML Services διότι, κάνει τις εφαρμογές πιο απλές, άρα και πιο «ελαφριές». Είναι συνεπώς πιο εύκολο να αναπτυχθούν και αναπτύσσονται σε μέγεθος (ιδανικά μικρό) που επιτρέπει την εύκολη ενσωμάτωσή τους σε web browser. Αντίθετα, στα Web Services το SOAP κάνει τις εφαρμογές πιο πολύπλοκες στο να αναπτυχθούν, επαρκώς πιο μεγάλες σε μέγεθος μιας και οποιοδήποτε XML αρχείο περιβάλλεται από τον φάκελο (envelope) που ορίζει υποχρεωτικά το SOAP και συμπερασματικά πιο χρονοβόρες.

Από την άλλη όμως τα XML Services ΔΕΝ είναι Web Services και δεν ακολουθούν τις προδιαγραφές του WS-I [WSI]. Συνεπώς δεν παρέχουν το βασικό χαρακτηριστικό της διαλειτουργικότητας, κάτι που αποτελεί θεμελιώδες στοιχείο των Web Services που βασίζονται στο πρότυπο SOAP. Επακόλουθο αποτελεί η έλλειψη ικανότητας αυτών των εφαρμογών να μπορούν να λειτουργούν και να προσφέρουν τις υπηρεσίες τους σε οποιαδήποτε πλατφόρμα και ανεξαρτήτως γλώσσας προγραμματισμού, κάτι που είναι προφανές και δεδομένο για τα Web Services. Ακόμη το SOAP αν και πιο πολύπλοκο στην εκμάθηση και χρήση του είναι επεκτάσιμο και υποστηρίζει επιτυχέστερα και σύνθετα δεδομένα. Επιπροσθέτως όντας στην κορυφή του ενδιαφέροντος είναι συνεχώς εξελίξιμο και αναβαθμίσιμο.

Όσον αφορά στην ασφάλεια (security), την πιστοποίηση (authentication) και την απόκρυψη (encryption) που επιτυγχάνεται στην κάθε περίπτωση, τα XML Services μιας και δεν αποτελούν Web Services δεν υποστηρίζουν τα πλεονεκτήματα που εντάσσονται ως χαρακτηριστικά στα δεύτερα. Πιο συγκεκριμένα με απλά XML Services δεν μπορεί να ακολουθούν τα πρότυπα WS-Security και WS-Encryption. Αυτά είναι χαρακτηριστικά που προκύπτουν από τη χρήση κάποιων κατάλληλων SOAP επεκτάσεων κατά τη δόμηση και υλοποίηση των Web Services ώστε να είναι ασφαλή. Τα χαρακτηριστικά του προτύπου WS-Security (σύνολο SOAP επεκτάσεων που επιφέρουν την ασφάλεια) είναι ευέλικτα και σχεδιάζονται έτσι ώστε να αποτελούν τη βάση για το «χτίσιμο» μοντέλων ασφαλείας (security models), όπως τα PKI, Kerberos και SSL.

4.1.3 Client Side

Με βάση τις σύγχρονες εφαρμογές πλοήγησης για φορητές συσκευές, τις οποίες μελετήσαμε, μπορούμε να επισημάνουμε κάποια από τα βασικότερα χαρακτηριστικά τα οποία θα έπρεπε να έχει ο client που θα κατασκευάσουμε. Ο client μας, λοιπόν, πρέπει να παρέχει τουλάχιστον τα εξής:

- Πλοήγηση σε προορισμό της επιλογής του χρήστη
- Οδηγίες «Στροφή-στροφή» γραπτές
- Φωνητικές οδηγίες
- Απεικόνιση πυξίδας
- Περιήγηση σε σημεία ενδιαφέροντος
- Περιήγηση σε διευθύνσεις επιλογής του χρήστη
- Δυνατότητα σύνδεσης με Bluetooth GPS Receiver που να υποστηρίζει το πρωτόκολλο NMEA.
- Εύχρηστο μενού επιλογών (λίγα κλικ για τις βασικές υπηρεσίες)
- Γραφικό περιβάλλον, φιλικό προς το χρήστη

Τα παραπάνω χαρακτηριστικά είναι εκείνα τα οποία τα έχουν γενικά όλα τα navigation systems τα οποία μελετήσαμε. Παρόλα αυτά, το σύστημά μας θα πρέπει να έχει κάποια ξεχωριστά χαρακτηριστικά που αφορούν κυρίως στον τρόπο σχεδίασής του. Θα πρέπει λοιπόν να ισχύουν τα εξής:

- Ανεξαρτησία πλατφόρμας και λειτουργικού συστήματος. Η εφαρμογή θα πρέπει να είναι μεταφέρσιμη, να μπορεί να τρέξει δηλαδή σε οποιαδήποτε φορητή συσκευή και με οποιοδήποτε λειτουργικό σύστημα. Για το λόγο αυτό προτείνεται η Java (J2ME) ως γλώσσα ανάπτυξης του συστήματος.
- Μικρό μέγεθος εφαρμογής ώστε να μπορεί να τρέχει και στις πιο περιορισμένες από πλευράς πόρων φορητές συσκευές. Αυτό είναι εφικτό διότι όλα τα δεδομένα όπως είδαμε μπορούν να είναι αποθηκευμένα στον server και να λαμβάνονται μέσω web services από τον client. Στον client πρέπει να είναι αποθηκευμένες μόνο οι κλάσεις απαραίτητες για τη λειτουργία του και ίσως ορισμένα resources όπως είναι για παράδειγμα αρχεία ήχου για τις φωνητικές οδηγίες. Οτιδήποτε σχετικό με γεωγραφική πληροφορία θα λαμβάνεται από το server. Θα πρέπει σε αυτό το σημείο να γίνει μελέτη για το χρόνο απόκρισης της εφαρμογής ώστε αυτός να είναι αποδεκτός για εφαρμογές πραγματικού χρόνου όπως είναι η πλοήγηση.

4.2 Χρήστες

Τους χρήστες των υπηρεσιών που θα αναπτυχθούν αποτελούν οι έως τώρα αναφερθέντες πελάτες-clients. Αυτοί χωρίζονται σε δύο γενικευμένες κατηγορίες που αναλύονται περαιτέρω ακολούθως. Αυτές είναι οι εξής: τα εκάστοτε «πληροφοριακά συστήματα» και οι «τελικοί χρήστες» (end users).

4.2.1 Πληροφοριακά συστήματα

Εν γένει κάποιο πληροφοριακό σύστημα μπορεί να λειτουργεί και ως πάροχος και ως χρήστης την ίδια ακριβώς στιγμή. Δηλαδή είναι σύνηθες κάποιο πληροφοριακό σύστημα που αποτελεί τον πάροχο των γεωγραφικών πληροφοριών για ένα χρήστη, να μην διαθέτει από μόνο του όλη τη γεωγραφική πληροφορία που απαιτούν οι ανάγκες του πελάτη-χρήστη, αλλά να την προμηθεύεται από κάποιο άλλο πληροφοριακό σύστημα, παρέχοντας δηλαδή μια υπηρεσία προστιθέμενης αξίας (Value Added Services). Στην προκειμένη περίπτωση το πρώτο πληροφοριακό σύστημα λειτουργεί σαν χρήστης για το δεύτερο.

4.2.2 Τελικοί χρήστες (End Users)

Με τον όρο αυτό αναφέρονται όλες οι αυτόνομες εφαρμογές, που κάνουν χρήση των προσφερόμενων υπηρεσιών, χωρίς να προσφέρουν με τη σειρά τους άλλες υπηρεσίες. Οι τελικοί χρήστες χωρίζονται με τη σειρά τους σε κατηγορίες προφίλ ανάλογα με τις ανάγκες τους. Συνεπώς τα πιθανά προφίλ χρήστη δεν είναι δεδομένα, αλλά είναι κάτι που επαφίεται στις επιθυμίες των παρόχων των υπηρεσιών. Μερικά από τα πλέον συνηθισμένα, είναι:

A. Οδηγός

Αυτό το προφίλ είναι ίσως από τα δημοφιλέστερα στην παρούσα αγορά και δυνητικά περιλαμβάνει το σύνολο των οδηγών. Οδηγός μπορεί να σημαίνει απλός οδηγός αυτοκινήτου, οδηγός ταξί, οδηγός φορτηγού ή ακόμα και ασθενοφόρου. Το κοινό χαρακτηριστικό μεταξύ των παραπάνω υποκατηγοριών οδηγού είναι ο στόχος για τον εντοπισμό της βέλτιστης διαδρομής μεταξύ δύο σημείων A, B και η παροχή των αντίστοιχων οδηγιών. Η ιδιότητα που διαχωρίζει την κατηγορία «οδηγός» σε υποκατηγορίες είναι το κριτήριο που κάνει μια διαδρομή βέλτιστη. Για παράδειγμα η διαδρομή γίνεται βέλτιστη για έναν οδηγό ασθενοφόρου όταν αυτή περιλαμβάνει τους εγγύτερους χωρίς κίνηση δρόμους (περιλαμβάνοντας και μονόδρομους αν αυτό είναι σωτήριο). Από την άλλη, έναν οδηγό ταξί

δεν τον ενδιαφέρει να συμπεριλάβει δρόμους μικρής διέλευσης, αφού εκεί δεν είναι πιθανό να εντοπίσει πελάτες.

Β. Τουρίστας

Το προφίλ του τουρίστα θα μπορούσαμε να πούμε ότι είναι μια εμπλουτισμένη έκδοση του προφίλ του οδηγού, μιας και αυτό δίνει τη διαδρομή μεταξύ δύο σημείων Α, Β, αλλά εμπεριέχει και οδηγό της εκάστοτε πόλης απαρτιζόμενο από σημεία ενδιαφέροντος όπως μουσεία, πλατείες, αξιοθέατα κτλ και τρόπους για της εξασφάλιση της εύκολης διαμονής και μετακίνησης του στην πόλη, παρέχοντας τα σημεία των σταθμών μέσων μεταφοράς (μετρό, λεωφορείων κτλ) και πιθανότατα και τα δρομολόγια τους. Δηλαδή στόχος της εφαρμογής όσον αφορά σε έναν τουρίστα είναι να προταθεί η βέλτιστη διαδρομή βάσει τουριστικών σημείων ενδιαφέροντος.

Γ. Πεζός

Το προφίλ για έναν πεζό διαφέρει με τα προηγούμενα στον τρόπο με τον οποίο δίνονται οι οδηγίες για τη διαδρομή. Δηλαδή για παράδειγμα αντί να δίνονται οδηγίες του στυλ «σε 300 μέτρα στρίψτε δεξιά», με το προφίλ πεζού αυτό θα μετατραπεί σε «μετά το σταθμό του μετρό στρίψτε δεξιά». Μ' αυτόν τον τρόπο ο πεζός δε χρειάζεται να μετράει απόσταση ώστε να ακολουθήσει τη σωστή πορεία, αλλά να μπορεί να χρησιμοποιεί την εφαρμογή ασυναίσθητα, απλά περπατώντας στο δρόμο, με τον ίδιο τρόπο που θα ακολουθούσε τις οδηγίες που θα παρείχε ένας άνθρωπος.

Δ. Ναυσιπλόος

Το προφίλ αυτό διαφέρει αρκετά με τα έως τώρα αναφερθέντα μιας και σε αυτό το προφίλ η χαρτογράφηση γίνεται σε θάλασσα, όπου και δίνεται η έμφαση. Το στοιχείο της ξηράς υπάρχει βέβαια, όμως μόνο για να θέτει τα όρια του χώρου πλεύσης. Δηλαδή στα νησιά δε χαρτογραφούνται δρόμοι και σημεία ενδιαφέροντος παρά μόνο οι ακτές, τα ακρωτήρια, οι κολποί, τα λιμάνια (φυσικά ή τεχνητά) και βεβαίως οι φάροι. Στοιχεία δηλαδή που η παρατήρηση και η γνώση της ύπαρξης τους θα έκανε την πλεύση των ναυσιπλόων κάθε κατηγορίας (επιβατηγό, εμπορικό, ιστιοπλοϊκό κτλ) ασφαλή και βέλτιστη.

Πρέπει εδώ να αναφέρουμε και την πιθανή υλοποίηση κάποιας εφαρμογής που θα συνδυάζει και τη χαρτογράφηση της θαλάσσιας περιοχής, αλλά και της ξηράς. Αυτή θα απευθύνεται κυρίως σε ταξιδιώτες που ανήκουν ταυτόχρονα και στην κατηγορία του ναυσιπλόου και του τουρίστα (π.χ. ιστιοπλόος), στον οποίο είναι απαραίτητη μια εφαρμογή που θα κάνει την

πλεύση ασφαλή και τη διαμονή ευχάριστη, παρέχοντας σημεία ενδιαφέροντος όπως πολιτισμικά στοιχεία (π.χ. του νησιού), εστιατόρια, χώρους διασκέδασης.

Βέβαια η λίστα με τα πιθανά προφίλ χρηστών μπορεί να επεκταθεί ανάλογα με την επιθυμία του καταναλωτή-χρήστη και τις ανάγκες της εκάστοτε αγοράς. Μερικά από τα πιο εξειδικευμένα προφίλ που μπορούν να προταθούν είναι:

- Πεζοπόρος / ορειβάτης
- Μεταφορική εταιρεία
- Αγρότες

4.3 Εμπορική Εκμετάλλευση

Η εμπορική εκμετάλλευση των γεωγραφικών πληροφοριών είναι μια δυνατότητα που είναι υλοποιήσιμη σήμερα και εφαρμόζεται στα εμπορικά συστήματα που παρουσιάσαμε σε προηγούμενη ενότητα. Οι πληρωμές μέσω του Διαδικτύου και οι υπηρεσίες παραγγελιών, τα μοντέλα πληρωμής και οι συμφωνίες παραχώρησης αδειών είναι μέρος αυτού του είδους χρήσης. Έτσι οι υπηρεσίες μέσω ασύρματων συσκευών για τον εντοπισμό, την εξαγωγή και την αποστολή γεωγραφικών δεδομένων θα πρέπει να μπορούν να εκτελέσουν και τις ανάλογες διαδικασίες πληρωμής.

Σε αυτό ακριβώς το σημείο θα πρέπει ίσως να δοθεί έμφαση στο γεγονός ότι ανάλογα με το πού δίνεται το βάρος στην εκμετάλλευση της εφαρμογής/υπηρεσίας, οι τρόποι πληρωμής μπορεί να ωφελούν είτε τον πάροχο, είτε τον καταναλωτή. Μπορεί ακόμα να εξυπηρετούν διαφορετικούς σκοπούς και πόρους και ανάλογα με αυτούς να διαμορφώνεται το μοντέλο πληρωμής.

Παρακάτω θα γίνει μια προσπάθεια κάλυψης των βασικότερων μοντέλων πληρωμής [MPM][EBU] για παροχή των δικτυακών υπηρεσιών γεωγραφικής πληροφορίας.

- Συνδρομητικά προγράμματα: Απευθύνονται κυρίως σε χρήστες που στηρίζουν τη δουλειά τους σε τέτοιες υπηρεσίες και επομένως η χρήση τους δεν αποτελεί πολυτέλεια, αλλά πηγή εισοδήματος όπως για παράδειγμα οι οδηγοί ταξί για τους οποίους θα είναι καθημερινά χρήσιμη μια τέτοια εφαρμογή.
- Ανά KB (όγκος διακινούμενης πληροφορίας): Ο τρόπος αυτός πληρωμής είναι απλός και προφανής. Ανάλογα με το αποτέλεσμα της ερώτησης σε μια βάση δεδομένων, μεταφέρεται πληροφορία της οποίας ο όγκος μετράται σε κάποιον αριθμό KB. Ο

όγκος διακινούμενης πληροφορίας υπολογίζεται αθροιστικά ανάλογα με τα τον όγκο που προκύπτει από το κάθε πληροφοριακό σύστημα (Value Added Service) που εμπλέκεται στην υποβληθείσα ερώτηση. Όμως ο όγκος του αποτελέσματος, που είναι και ο όγκος που διακινείται είναι ο ίδιος, είτε το ερώτημα ήταν απλό, είτε ήταν σύνθετο. Δηλαδή ίδιος όγκος, άρα και ίδιο κόστος πληρωμής σε διαφορετικό κόστος επεξεργασίας. Συνεπώς αυτό το μοντέλο πληρωμής θα είναι κατάλληλο μόνο όταν η ερώτηση απευθύνεται σε βελτιστοποιημένα αρχεία, όπου τα δεδομένα είναι οργανωμένα βάσει κάποιων ήδη υπαρχόντων στατιστικών αποτελεσμάτων που έχουν να κάνουν με τη συχνότητα συγκεκριμένων ερωτήσεων σε μια ΒΔ ή ακόμη σε εφαρμογές υλοποιημένες με έξυπνους αλγόριθμους προανάκτησης (pre-fetching). Το προτέρημα αυτών των αλγορίθμων στην προκειμένη περίπτωση είναι ότι διαθέτουν μνήμη και μπορούν να «θυμούνται» τις τελευταίες ερωτήσεις στη ΒΔ με τη λογική ότι ερωτήσεις που έγιναν θα ξαναγίνουν.

Το πρόβλημα που εντοπίζεται σε αυτό το μοντέλο πληρωμής έγκειται στην έλλειψη εμπιστοσύνης από την πλευρά του πελάτη. Η εφαρμογή που σχεδίασε κάποιος πάροχος δεν είναι απαραίτητο να είναι και η βέλτιστη με αποτέλεσμα ο πελάτης να χρεώνεται όγκο πληροφορίας που δεν είναι ωφέλιμος.

- Άπαξ χρέωση: Δηλαδή εδώ μιλάμε για άπαξ κοστολόγηση ανά πρόσβαση στη δικτυακή υπηρεσία, ό,τι κι αν περιλαμβάνει η απάντηση, όσοι υπολογισμοί και να χρειαστούν για τη διεξαγωγή της. Το μοντέλο αυτό συνεπάγεται τον κίνδυνο ο χρήστης να πληρώνει ελάχιστα για πολύπλοκη επεξεργασία της ερώτησης που έθεσε στην εφαρμογή (γεγονός που είναι ασύμφορο για τον πάροχο) ή από την άλλη μεριά τον κίνδυνο για τον χρήστη να πληρώνει μια πολύ απλή ερώτηση στο κόστος μιας πολύπλοκης, η οποία ορίστηκε ως χρέωση ανά πρόσβαση (γεγονός συμφέρον για τον πάροχο). Βλέπουμε λοιπόν, ότι αυτός ο τρόπος κοστολόγησης είναι αρκετά δύσκολο να ισορροπήσει τα συμφέροντα παρόχου και καταναλωτή με πιθανότερο αποτέλεσμα την αποτυχία εδραίωσης της εφαρμογής στην αγορά λόγω αποτροπής του καταναλωτή από την αγορά και χρήση του. Μιλάμε δηλαδή και πάλι για πρόβλημα εμπιστοσύνης.
- Ανά ερώτημα στη ΒΔ (query): Κατά αυτό τον τρόπο επιτυγχάνεται μια πιο δίκαιη κοστολόγηση της παροχής υπηρεσιών, αφού έτσι ο χρήστης πληρώνει ακριβώς το τίμημα της εργασίας που του παρέχεται. Κάθε ερώτημα αποτελείται από άλλα θεμελιώδη κι έτσι το κόστος υπολογίζεται αθροιστικά ανάλογα με τα τον αριθμό των ερωτημάτων-μονάδων που επιχειρεί η ΒΔ. Το συγκεκριμένο μοντέλο πληρωμής, όμως θα μπορούσε να χαρακτηριστεί αδρομερές και δε συστήνεται αυτοτελές παρά μόνο στη περίπτωση που τα ερωτήματα είναι απολύτως συγκεκριμένα και το κόστος

τους εκ των προτέρων γνωστό για τον πάροχο.

- **Ανά επιτυχή απάντηση:** Στο μοντέλο αυτό ο χρήστης υποχρεούται να πληρώσει μόνο όταν η ερώτηση, η επεξεργασία και η απάντηση δόθηκε επιτυχώς, δηλαδή μονάχα όταν η διαδικασία έγινε ολοκληρωμένα και το σύστημα την έφερε εις πέρας. Αυτός ο τρόπος κοστολόγησης εξυπηρετεί ιδιαίτερα τον χρήστη μιας και κανένας πάροχος δεν είναι σε θέση να εξασφαλίσει μια απάντηση για κάθε ερώτηση του χρήστη, είτε γιατί το σύστημα μπορεί κάπου να εκτελέσει κάποιο σφάλμα, είτε γιατί απλά τα δεδομένα που περιλαμβάνει δεν είναι επαρκή για την παραγωγή του ζητούμενου αποτελέσματος. Όμως αυτό συνεπάγεται ότι η πράξεις και οι υπολογισμοί που θα εκτελεί το σύστημα χωρίς επιτυχία θα επιβαρύνουν το σύστημα χωρίς κάποιο αντίτιμο. Το παραπάνω μοντέλο θα μπορούσε να υλοποιηθεί σε καλά δομημένες και επαρκώς εμπλουτισμένες ΒΔ με συνεπαγόμενο μηδαμινό ποσοστό αποτυχίας.
- **Δωρεάν πρόσβαση σε περιεχόμενο:** Ο δυνατότερος εναλλακτικός τρόπος, ο οποίος μπορεί να αυξήσει τον αριθμό των χρηστών σε πολύ μικρό χρονικό διάστημα, χωρίς όμως να εγγυηθεί πίστη του καταναλωτή. Ο κατακερματισμός του περιεχόμενου σε υποκατηγορίες με περισσότερες λεπτομέρειες μας δίνει τη δυνατότητα της περαιτέρω χρέωσης, πέρα του δωρεάν περιεχομένου.
- **Δωρεάν για τους χρήστες:** Αν και το συγκεκριμένο μοντέλο φαίνεται -με την πρώτη ματιά- ακατόρθωτο από την πλευρά του παρόχου, δεν είναι. Σε αυτό το μοντέλο το κόστος πληρώνεται από τις εταιρείες και τους οργανισμούς που μπορούν να απολαμβάνουν τα οφέλη της διαφημίσεις των προϊόντων και των υπηρεσιών που προσφέρουν. Οι διαφημίσεις αυτές μπορεί να συγκαταλεχθούν στα περιεχόμενα της εφαρμογής ως σημεία ενδιαφέροντος κατά μήκος μιας διαδρομής.
- **Υβριδικά μοντέλα χρέωσης:** Όπως έγινε κατανοητό κανένα από τα προαναφερθέντα μοντέλα χρέωσης δεν μπορεί να χαρακτηριστεί ως κοινώς αποδεκτό, γιατί κάθε ένα μπορεί έχει τη βέλτιστη απόδοση ανάλογα με τις ανάγκες του χρήστη, την προσφορά του εκάστοτε παρόχου και τη δομή και οργάνωση των ΒΔ που διαχειρίζονται τα πληροφοριακά συστήματα που εμπλέκονται στη εκτέλεση μιας επερώτησης και τη μεταφορά δεδομένων. Έτσι σε πρώτη φάση τουλάχιστον, καλό είναι να ακολουθηθούν συνδυαστικά διάφοροι τρόποι χρέωσης των χρηστών, όπως για παράδειγμα την δωρεάν πρόσβαση σε περιεχόμενο μέχρι ενός σημείου, όπου μπαίνει σε λειτουργία είτε χρέωση ανά KB / ανά ερώτημα για συγκεκριμένα θεμελιώδη ερωτήματα (με την έννοια της μονάδας), είτε χρέωση βάσει συνδρομής σε συνδυασμό με ταυτόχρονη μείωση του κόστους αυτού με τον εμπλουτισμό της εφαρμογής με διαφημίσεις που θα λειτουργούν ως σημεία ενδιαφέροντος χρήσιμα

στον χρήστη (ανάλογα με το προφίλ του).

5

Ανάλυση και σχεδίαση

Στο παρόν κεφάλαιο περιγράφεται η ανάλυση και η σχεδίαση του λογισμικού συστήματος του οποίου την γενική περιγραφή δώσαμε και στην προηγούμενη ενότητα. Η ανάλυση και σχεδίαση αυτή γίνεται με τη βοήθεια των εγγράφων τεχνολογίας λογισμικού για το προς ανάπτυξη σύστημα. Αρχικά κάνουμε μια εισαγωγή με την περιγραφή της γλώσσας μοντελοποίησης UML καθώς και των μεθοδολογιών ανάπτυξης λογισμικού RUP και YASM οι οποίες χρησιμοποιήθηκαν. Στη συνέχεια παραθέτουμε τα έγγραφα της τεχνολογίας λογισμικού που προέκυψαν. Σημειώνουμε στο σημείο αυτό ότι για τα περισσότερα από τα σχέδια που έγιναν για τις ανάγκες των εγγράφων της τεχνολογίας λογισμικού χρησιμοποιήθηκε το πρόγραμμα Microsoft Office Visio 2003.

5.1 Εισαγωγή

5.1.1 UML (Unified Modeling Language)

Η ενοποιημένη γλώσσα μοντελοποίησης UML (Unified Modeling Language) είναι μια γραφηματική (graphical) γλώσσα μοντελοποίησης γενικής χρήσης. Προέκυψε από την ενοποίηση πολλών υπαρχόντων και διαφορετικών μεταξύ τους γλωσσών μοντελοποίησης, που είχαν χρησιμοποιηθεί στην πράξη στο πλαίσιο διαφόρων μεθοδολογιών ανάπτυξης λογισμικών συστημάτων. Συμπεριλαμβάνει ότι καλό είχαν αυτές οι προηγούμενες γλώσσες

μοντελοποίησης. Ήλθε να καλύψει την ανάγκη ύπαρξης μιας μοναδικής γλώσσας μοντελοποίησης με όλα τα καλά που αυτό συνεπάγεται. Έτσι, δικαιολογείται και η ευρεία αποδοχή της σήμερα. Βέβαια, πρωταρχικά απευθύνεται στη μοντελοποίηση λογισμικών συστημάτων. Όμως, μπορεί να χρησιμοποιηθεί και για μοντελοποίηση άλλων κατηγοριών συστημάτων, όπως των τεχνητών συστημάτων. Η UML είναι τώρα βιομηχανικό πρότυπο και υποστηρίζεται από το μεγάλο οργανισμό OMG [<http://www.omg.org/>], ανήκει δε στην κατηγορία των αντικειμενοστραφών γλωσσών μοντελοποίησης.

5.1.2 RUP (Rational Unified Process)

Η μεθοδολογία RUP (Rational Unified Process) αποτελεί μια βιομηχανική μεθοδολογία λογισμικού η οποία αναπτύχθηκε από την εταιρεία Rational Software Corporation. Η RUP βασίζεται σε ένα σύνολο από αρχές ανάπτυξης λογισμικού όπως για παράδειγμα:

- Ανάπτυξη λογισμικού επαναληπτικά, δηλαδή τα μέρη του λογισμικού αναπτύσσονται, ελέγχονται και συμπληρώνονται διαρκώς.
- Χειρισμός των απαιτήσεων των χρηστών, δηλαδή καθορισμός των αναγκών των χρηστών και αλλαγή των απαιτήσεων όταν οι ανάγκες αυτές αλλάζουν
- Χρήση αρχιτεκτονικής με components
- Οπτική μοντελοποίηση του λογισμικού
- Επαλήθευση της ποιότητας του λογισμικού
- Έλεγχος των αλλαγών στο λογισμικό

5.1.3 YASM (Yet Another Software development Methodology)

Η YASM (Yet Another Software development Methodology) είναι μια εκπαιδευτική μεθοδολογία ανάπτυξης λογισμικών συστημάτων που αναπτύχθηκε στο Εργαστήριο Λογισμικής Μηχανικής (Software Engineering Lab) του ΕΜΠ.

Περίληπτικά, η YASM έχει τη δική της φιλοσοφία που βασίζεται στα επόμενα σημεία:

- 1) Τα λογισμικά συστήματα είναι συστατικά των υπολογιστικών συστημάτων και αυτά είναι υποσυστήματα των τεχνητών συστημάτων.

- 2) Τα λογισμικά συστήματα αυτοματοποιούν την εκτέλεση δεδομενικών εργασιών, οι οποίες πάντοτε προκύπτουν ως εκτελέσεις λύσεων δεδομενικών προβλημάτων.
- 3) Τα λογισμικά συστήματα είναι γλωσσολογικά προϊόντα.
- 4) Η λογισμική εργασία (η εργασία που απαιτείται για την ανάπτυξη των λογισμικών συστημάτων) είναι μια πνευματική εργασία που μόνο μερικώς μπορεί να υποστηριχθεί από εργαλεία και βέβαια δεν είναι αυτοματοποιήσιμη, τουλάχιστον έως τώρα.
- 5) Για την ανάπτυξη ενός λογισμικού συστήματος απαιτούνται τρεις λύσεις: η θεωρητική, η σχεδιαστική και η τεχνολογική και αντίστοιχα τρεις γλώσσες: η αρχική, η σχεδιαστική και η κωδική, για την περιγραφή αυτών των λύσεων.
- 6) Ο μετασχηματισμός της θεωρητικής δεδομενικής λύσης σε τεχνολογική λύση στην ουσία ισοδυναμεί με μετάφραση.
- 7) Η ανάπτυξη ενός λογισμικού συστήματος χωρίζεται σε δύο μέρη: στο Μπροστινό που είναι ένα και στο Πισινό που μπορεί να είναι πολλά (πολλές διαφορετικές υλοποιήσεις). Το Μπροστινό είναι ανεξάρτητο της Τεχνολογίας που θα χρησιμοποιηθεί ενώ το Πισινό είναι εξαρτημένο από αυτήν.
- 8) Η YASM είναι ανεξάρτητη της Σχολής Προγραμματισμού (διαδικασιακής, αντικειμενοστραφούς, ψηφιδοστρεφούς) που θα χρησιμοποιηθεί.

5.2 Έγγραφο Προδιαγραφών Απαιτήσεων από το Λογισμικό

5.2.1 Εισαγωγή

5.2.1.1 Σκοπός

Σκοπός αυτού του εγγράφου είναι η παρουσίαση των προδιαγραφών των απαιτήσεων για το λογισμικό σύστημα “mobiNav”. Το λογισμικό σύστημα “mobiNav” υλοποιεί αφενός εφαρμογή εξυπηρετητή (server) που παρέχει διαδικτυακές υπηρεσίες για τη διάχυση χωρικής πληροφορίας και αφετέρου εφαρμογή πελάτη (client) η οποία εγκαθίσταται σε ασύρματες φορητές συσκευές, καταναλώνει τις παραπάνω υπηρεσίες και παρέχει υπηρεσίες πλοήγησης στον τελικό χρήστη. Ο χρήστης του συστήματος θα έχει τη δυνατότητα να καθορίσει μια διαδρομή για την οποία θέλει να μάθει κάποιες πληροφορίες (διάρκεια, απόσταση, οδηγίες

οδήγησης), ή ακόμα και να λάβει φωνητικές οδηγίες για την πλοήγησή του. Τέλος, θα μπορεί να παρακολουθεί τη θέση του στο χάρτη σε πραγματικό χρόνο καθώς θα κινείται.

Όλες οι πληροφορίες σχετικά με τους διάφορους προορισμούς, όπως το που βρίσκονται διάφορα σημεία ενδιαφέροντος, διευθύνσεις, κτλ, θα βρίσκονται αποθηκευμένες σε έναν απομακρυσμένο εξυπηρετητή ΓΣΠ (GIS server) ο οποίος θα τις παρέχει μέσω των διαδικτυακών υπηρεσιών που θα αναπτυχθούν. Το ίδιο ισχύει και για τους χάρτες των διαφόρων περιοχών. Το μόνο που θα είναι αποθηκευμένο στην ασύρματη φορητή συσκευή είναι η εφαρμογή client.

5.2.1.2 Εμβέλεια

Το παρόν έγγραφο παρουσιάζει τις περιπτώσεις χρήσης του συστήματος και περιλαμβάνει περαιτέρω ανάλυση αυτών με τη βοήθεια κατάλληλων διαγραμμάτων σε UML. Μέσα από τα διαγράμματα αυτά, παρουσιάζεται η συμμετοχή και των δυο βασικών τμημάτων του λογισμικού, του τμήματος της πλευράς του εξυπηρετητή (server side) και της πλευράς του πελάτη (client side). Το έγγραφο απευθύνεται σε οποιονδήποτε επιθυμεί να χρησιμοποιεί τις παραπάνω υπηρεσίες, καθώς και στον αρχιτέκτονα και κατασκευαστή του συστήματος.

5.2.1.3 Ορισμοί, Ακρωνύμια, Συντομογραφίες

Web Services: Διαδικτυακές υπηρεσίες

GPS: Geographical Positioning System

POI: Point of Interest (Σημείο ενδιαφέροντος)

GPRS: General Packet Radio Service

SSL: Secure Socket Layer

OGC: Open Geospatial Consortium

5.2.1.4 Αναφορές

Χρησιμοποιήθηκαν οι πηγές που αναφέρονται στο κεφάλαιο για τη Βιβλιογραφία.

5.2.2 Γενική περιγραφή

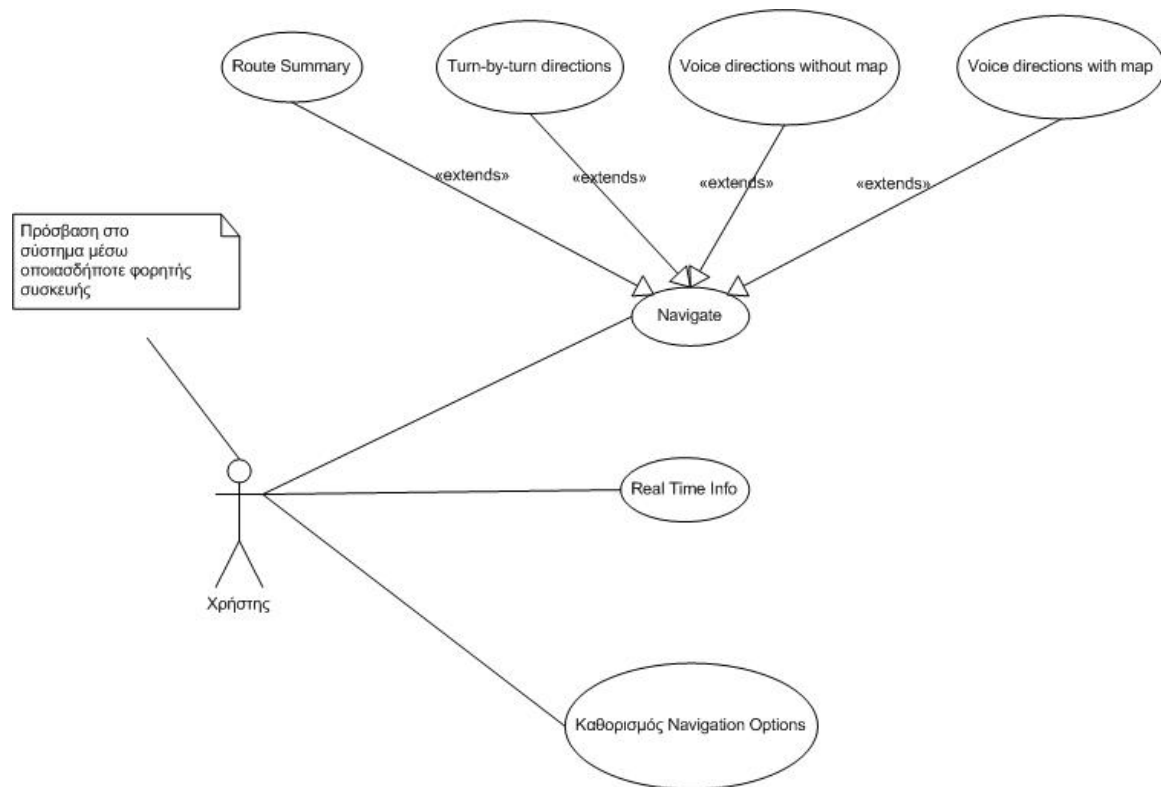
5.2.2.1 Μοντέλο περιπτώσεων χρήσης

Το μοντέλο περιπτώσεων χρήσης περιγράφει μια σειρά από ενέργειες που εκτελεί το σύστημα ώστε να αποδώσει χρήσιμα αποτελέσματα σε ένα ρόλο του συστήματος. Με τον όρο «ρόλος του συστήματος» αναφέρεται ένας χρήστης ή ένα εξωτερικό σύστημα (λογισμικό) που αλληλεπιδρά με το παρόν σύστημα λογισμικού για να λάβει κάποιες πληροφορίες.

Το ενδιαμέσο σύστημα λογισμικού “**mobiNav Web Services Server**” υλοποιεί τέσσερις διαδικτυακές υπηρεσίες (Web Services) οι οποίες είναι οι βασικές υπηρεσίες όπως καθορίζονται από το πρότυπο OpenLS του OGC (OpenLS core services). Τις υπηρεσίες αυτές καλεί το client side του λογισμικού μας “**mobiNav Client**” ώστε να παρέχει στον τελικό χρήστη υπηρεσίες πλοήγησης (πληροφορίες για τη διαδρομή, οδηγίες οδήγησης, φωνητικές οδηγίες, χάρτες διαδρομής), καθώς και παρουσίασης τους σε χάρτη σε πραγματικό χρόνο. Με βάση τα παραπάνω δημιουργούνται οι εξής έξι (6) περιπτώσεις χρήσης του συστήματός μας:

- Route Summary (Πληροφορίες για τη διαδρομή)
- Turn-by-turn directions (Οδηγίες οδήγησης)
- Voice Directions without map (Φωνητικές οδηγίες χωρίς χάρτη)
- Voice Directions with map (Φωνητικές οδηγίες με χάρτη)
- Real Time Info (Απεικόνιση σε χάρτη σε πραγματικό χρόνο)
- Navigation Options (Επιλογές πλοήγησης)

Οι παραπάνω περιπτώσεις χρήσης του συστήματός μας φαίνονται στο παρακάτω διάγραμμα. Στη συνέχεια του κειμένου θα γίνει ανάλυση της καθεμίας από τις περιπτώσεις χρήσης με τη βοήθεια διαγραμμάτων δραστηριοτήτων (activity diagrams).



Εικόνα 18: Use case diagram

Όπως φαίνεται στο παραπάνω διάγραμμα, οι τέσσερις πρώτες περιπτώσεις υπάγονται στο σύνολό τους σε μία γενική περίπτωση χρήσης του συστήματός μας που αφορά στην πλοήγηση (navigation).

5.2.3 Απαιτήσεις

5.2.3.1 Προδιαγραφές περιπτώσεων χρήσης

Κάθε περίπτωση χρήσης αναλύεται και τεκμηριώνεται στη συνέχεια του παρόντος εγγράφου στην ενότητα «Ανάλυση Περιπτώσεων Χρήσης».

5.2.3.2 Λειτουργικότητα

Ακολούθως περιγράφονται τα γνωρίσματα που απαιτείται να έχει το λογισμικό του client από πλευράς λειτουργικότητας προς το χρήστη. Τα γνωρίσματα αυτά σχετίζονται με τις περιπτώσεις χρήσης που προαναφέρθηκαν.

User Interface

Απαιτείται η ύπαρξη GUI (Graphical User Interface) μέσω του οποίου ο χρήστης θα έχει πρόσβαση στις υπηρεσίες. Το GUI θα πρέπει να είναι ενιαίο για την πρόσβαση σε όλες τις διατιθέμενες υπηρεσίες. Ο χρήστης θα δίνει τις απαραίτητες πληροφορίες στην εφαρμογή client μέσω των διαφόρων φορμών του GUI. Οι πληροφορίες αυτές θα έχουν σχέση τόσο με τον καθορισμό του τρόπου λειτουργίας του συστήματος (π.χ. με απεικόνιση σε χάρτη της διαδρομής ή όχι), όσο και με τον καθορισμό των διαφόρων παραμέτρων για κλήση κάπου web service (π.χ. καθορισμός προορισμού προκειμένου να λάβουμε οδηγίες για μία διαδρομή). Στη συνέχεια, η εφαρμογή θα είναι σε θέση να καλεί τα διάφορα web services που παρέχονται από τον server και να παρουσιάζει τα αποτελέσματα τα οποία λαμβάνει σε γραφικό επίσης περιβάλλον, φιλικό προς το χρήστη.

5.2.3.3 Απαιτήσεις χρήσης

Το λογισμικό θα πρέπει να είναι απλό στη χρήση και να μην απαιτεί κάποια ιδιαίτερη εκμάθηση ή ιδιαίτερες γνώσεις για κάποιον που θέλει να το χρησιμοποιήσει. Για το λόγο αυτό, όπως προαναφέραμε, το GUI του συστήματος θα πρέπει να είναι φιλικό προς το χρήστη και να βασίζεται σε γραφικό περιβάλλον.

5.2.3.4 Απαιτήσεις αξιοπιστίας

Για να είναι το σύστημα διαθέσιμο διαρκώς στο χρήστη, θα πρέπει να είναι διαθέσιμα τα web services που υλοποιούνται στον server μας. Η παραπάνω απαίτηση προϋποθέτει την διαθεσιμότητα του δικτύου GPRS για τη σύνδεση της φορητής συσκευής με τον server μας, καθώς και τη διαθεσιμότητα της βάσης δεδομένων που είναι αποθηκευμένες όλες οι γεωγραφικές πληροφορίες. Πρέπει, δηλαδή, να είναι διαθέσιμος μονίμως και ο GIS server.

5.2.3.5 Απαιτήσεις ασφάλειας

Το σύστημά μας θα πρέπει να εγγυάται πιστοποίηση εξυπηρετητή και πελάτη (authentication) για να γίνει ανταλλαγή δεδομένων. Ο χρήστης για να μπορεί να χρησιμοποιεί τις υπηρεσίες θα πρέπει αρχικά να πιστοποιείται. Έτσι, οι δικτυακές υπηρεσίες θα πρέπει να παρέχονται πάνω από ασφαλές κανάλι επικοινωνίας SSL.

5.2.3.6 Απαιτήσεις επιδόσεων

Το σύστημά μας θα πρέπει να αρχικοποιείται γρήγορα και επιπλέον ο χρόνος απόκρισης του θα πρέπει να είναι σχετικά μικρός. Έτσι, πρέπει να γίνεται σχετικά γρήγορα η αποστολή και λήψη μηνυμάτων από client και server καθώς και η εξαγωγή στοιχείων από αυτά (parsing). Η απαιτήσεις αυτές οφείλονται στο γεγονός ότι η εφαρμογή θα δίνει οδηγίες σε πραγματικό χρόνο γεγονός που απαιτεί ταχύτητα στην επεξεργασία των δεδομένων.

Επιπλέον, η εφαρμογή μας θα πρέπει να τρέχει σε οποιαδήποτε φορητή συσκευή, να είναι δηλαδή ανεξάρτητη πλατφόρμας. Επειδή οι περισσότερες φορητές συσκευές έχουν περιορισμένους πόρους (επεξεργαστή, μνήμη, μέγεθος οθόνης, ταχύτητα σύνδεσης με το Internet), η εφαρμογή client καθώς και τα μηνύματα που θα ανταλλάσσονται με τον server θα πρέπει να είναι «ελαφριά» (lightweight).

5.2.3.7 Πρότυπα

Το σύστημα θα πρέπει να βασίζεται σε ανοικτά πρότυπα, όπως για παράδειγμα στα πρότυπα του OGC για διάθεση γεωγραφικών πληροφοριών και στα πρότυπα που δίνει ο οργανισμός πιστοποίησης ISO. Επιπλέον οι διαδικτυακές υπηρεσίες θα πρέπει να είναι SOAP/XML Web Services.

5.2.4 Ανάλυση περιπτώσεων χρήσης

5.2.4.1 Περίπτωση χρήσης 1: Navigate

5.2.4.1.1 Περιγραφή περίπτωσης χρήσης

Η συγκεκριμένη περίπτωση χρήσης αποτελεί τη βασική περίπτωση χρήσης του συστήματός μας. Αφορά στην πλοήγηση του χρήστη σε διαδρομή της επιλογής του. Η περίπτωση χρήσης αυτή είναι γενική και περιλαμβάνει τέσσερις υποπερίπτώσεις όπως εξάλλου φαίνεται και στο διάγραμμα περιπτώσεων χρήσης που παραθέσαμε παραπάνω. Οι υποπερίπτώσεις αυτές είναι οι εξής:

- Route Summary: Ο χρήστης επιθυμεί να μάθει μόνο τον συνολικό χρόνο και απόσταση για μία διαδρομή.

- Turn-by-turn directions: Ο χρήστης επιθυμεί να μάθει οδηγίες οδήγησης για το πώς θα πάει από μία αφετηρία σε κάποιον προορισμό.
- Voice directions without map: Ο χρήστης επιθυμεί πλοήγηση από το σύστημα με φωνητικές οδηγίες χωρίς απεικόνιση της θέσης του σε χάρτη.
- Voice directions with map: Ο χρήστης επιθυμεί πλοήγηση από το σύστημα με φωνητικές οδηγίες με ταυτόχρονη απεικόνιση της θέσης του σε χάρτη.

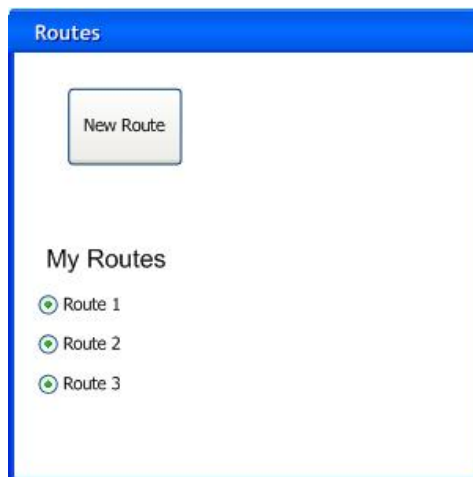
Ο χρήστης καθορίζει τον τρόπο λειτουργίας του συστήματος σε μία από τις παραπάνω περιπτώσεις μέσω των επιλογών πλοήγησης (Navigation Options) όπως θα δούμε στη συνέχεια.

Για την περίπτωση χρήσης “Navigate” ο χρήστης κάνει κλικ στο κουμπί “Navigation” στην οθόνη “Welcome Options Screen” η οποία θα έχει την εξής μορφή.



Εικόνα 19: Welcome Options Screen

Στη συνέχεια, ο χρήστης καθορίζει τη διαδρομή πλοήγησης είτε επιλέγοντας μία διαδρομή αποθηκευμένη στη φορητή συσκευή στο “my Routes”, είτε ορίζοντας μια νέα διαδρομή όπως φαίνεται στην παρακάτω οθόνη.



Εικόνα 20: Route Screen

Για να ορίσει ο χρήστης μια νέα διαδρομή, παρέχονται οι εξής δυνατότητες μέσω της εφαρμογής μας:

- Επιλογή σημείων από λίστα με αποθηκευμένα σημεία (my POIs).
- Εύρεση νέου POI μέσω του Directory Service που παρέχεται από τον server.
- Εύρεση νέας διεύθυνσης μέσω του Geocoding Service που παρέχεται από τον server.

Έτσι, για τον καθορισμό μιας νέας διαδρομής το User Interface της εφαρμογής θα πρέπει να παρέχει επιπλέον τις εξής οθόνες:

- New Route Screen: Ο χρήστης επιλέγει να ορίσει αφετηρία, ενδιάμεσες στάσεις, προορισμό για τη διαδρομή.
- POI Screen: Ο χρήστης επιλέγει ένα σημείο ενδιαφέροντος από τη λίστα των “my POIs” είτε επιλέγει να βρει ένα νέο σημείο ενδιαφέροντος ή μια διεύθυνση.
- Directory Options Screen: Ο χρήστης επιλέγει τύπο σημείου ενδιαφέροντος προς αναζήτηση.
- Geocode Options Screen: Ο χρήστης εισάγει μία διεύθυνση προς αναζήτηση.

Η μορφή που θα πρέπει να έχουν οι οθόνες αυτές παρουσιάζεται στη συνέχεια.



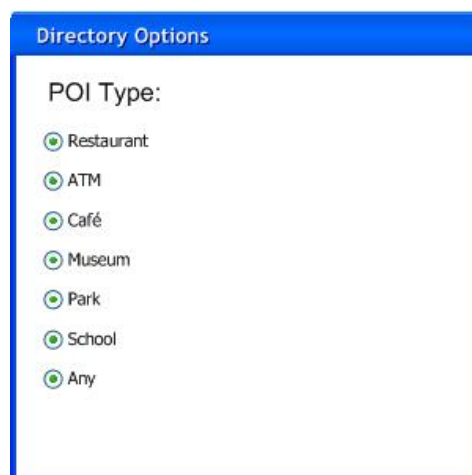
The 'New Route' screen features a blue header bar with the title 'New Route'. Below the header, there are three vertically stacked text input fields. The first field is labeled 'Start Point:', the second is labeled 'Via Points:', and the third is labeled 'End Point:'. Each label is positioned to the left of its corresponding input box.

Εικόνα 21: New Route Screen



The 'Points of Interest' screen has a blue header bar with the title 'Points of Interest'. Below the header, there are two buttons: 'Directory Service: Find a new POI' and 'Geocoding Service: Find a new Address'. Below these buttons, the text 'My POIs' is displayed, followed by a list of three items: 'POI 1', 'POI 2', and 'POI 3'. Each item is preceded by a green circular icon containing a white dot.

Εικόνα 22: POI Screen



The 'Directory Options' screen features a blue header bar with the title 'Directory Options'. Below the header, the text 'POI Type:' is displayed. Underneath, there is a list of seven options: 'Restaurant', 'ATM', 'Café', 'Museum', 'Park', 'School', and 'Any'. Each option is preceded by a green circular icon containing a white dot.

Εικόνα 23: Directory Options Screen



Εικόνα 24: Geocode Options Screen

5.2.4.1.2 Ροή γεγονότων

Η βασική ροή γεγονότων περιγράφει τη διαδικασία κατά την οποία η περίπτωση χρήσης εκτελείται και ολοκληρώνεται με επιτυχία. Σε περίπτωση σφάλματος κατά την βασική ροή έχουμε εναλλακτικές ροές γεγονότων που περιγράφουν το τι ακολουθεί ανά περίπτωση.

Προτού περιγράψουμε τη βασική ροή για την περίπτωση χρήσης “Navigate” θα πρέπει να παρουσιάσουμε τρεις ροές γεγονότων, τις «Εύρεση Νέου POI», «Εύρεση διεύθυνσης», «Καθορισμός Διαδρομής». Οι ροές αυτές θα χρησιμοποιηθούν από τη βασική ροή της περίπτωσης χρήσης “Navigate”.

Ροή «Εύρεση Νέου POI»

1. Ο χρήστης επιλέγει από τη λίστα με τους διάφορους τύπους POI, τον τύπο του σημείου ενδιαφέροντος στο οποίο επιθυμεί να πλοηγηθεί (βλ. Directory Options Screen παραπάνω).
2. Το σύστημα προσδιορίζει τη θέση του χρήστη. Αυτό γίνεται είτε μέσω εξωτερικής συσκευής Bluetooth GPS Receiver, είτε μέσω της υπηρεσίας Gateway Service του server.
3. Αποστολή μηνύματος κλήσης της υπηρεσίας “Directory Service” με παραμέτρους τις συντεταγμένες θέσης που έχουν προσδιοριστεί και τον τύπο POI επιλογής του χρήστη.
4. Λήψη από τον server των στοιχείων που εστάλησαν με το μήνυμα μέσω parsing του μηνύματος.
5. Δημιουργία και αποστολή απάντησης στον client με τη λίστα των POI που βρέθηκαν.

6. Εμφάνιση της λίστας των POI στην οθόνη της φορητής συσκευής για επιλογή από το χρήστη. Ο χρήστης μπορεί να επιλέξει ένα σημείο ενδιαφέροντος από τη λίστα για αποθήκευση (my POIs) είτε για χρήση προς καθορισμό διαδρομής.

Ροή «Εύρεση Διεύθυνσης»

1. Ο χρήστης εισάγει τη διεύθυνση την οποία θέλει να βρει για να πλοηγηθεί (βλ. Geocode Options Screen παραπάνω).
2. Αποστολή μηνύματος κλήσης της υπηρεσίας “Geocode Service” με παράμετρο τη διεύθυνση επιλογής του χρήστη.
3. Λήψη από τον server των στοιχείων που εστάλησαν με το μήνυμα μέσω parsing του μηνύματος.
4. Δημιουργία και αποστολή απάντησης στον client με την διεύθυνση που βρέθηκε με βάση την αρχική επιλογή του χρήστη.
5. Εμφάνιση της διεύθυνσης που βρέθηκε στην οθόνη της φορητής συσκευής για επιλογή από το χρήστη. Ο χρήστης μπορεί να επιλέξει τη διεύθυνση για αποθήκευση (my POIs) είτε για χρήση προς καθορισμό διαδρομής.

Ροή «Καθορισμός Διαδρομής»

1. Ο χρήστης καθορίζει αφετηρία για τη διαδρομή (start Point όπως φαίνεται στην οθόνη New Route Screen παραπάνω). Αυτό γίνεται είτε μέσω εύρεσης νέου POI ή εύρεσης διεύθυνσης, είτε επιλέγοντας από τα my POIs.
2. Με όμοιο με το βήμα 1 τρόπο, ο χρήστης καθορίζει ενδιάμεσες στάσεις (via Points) καθώς και προορισμό (end Point) για τη διαδρομή.
3. Ο χρήστης μπορεί να αποθηκεύσει τη διαδρομή (my Routes) είτε να τη χρησιμοποιήσει άμεσα για πλοήγηση.

Στη συνέχεια ακολουθεί η βασική ροή για την περίπτωση χρήσης “Navigate” η οποία κάνει χρήση όπως είπαμε των παραπάνω ροών.

Βασική Ροή “Navigate”

1. Ο χρήστης κάνει κλικ στο κουμπί “Navigation” στην αρχική οθόνη (βλ. οθόνη Welcome Options Screen παραπάνω).
2. Ο χρήστης καθορίζει τη διαδρομή στην οποία θέλει να πλοηγηθεί (βλ ροή «Καθορισμός διαδρομής» παραπάνω) είτε επιλέγει άμεσα μία από τις αποθηκευμένες στο my Routes.

3. Αποστολή στον server μηνύματος κλήσης της υπηρεσίας “Route Service” με παράμετρο τα καθορισμένα σημεία για τη διαδρομή (αφετηρία, ενδιάμεσες στάσεις, προορισμός) και τις επιλογές του χρήστη στο “Navigation Options” που θα δούμε στη συνέχεια.
4. Λήψη στον server των στοιχείων μέσω parsing του μηνύματος και αποστολή στον client μηνύματος απάντησης με τη διαδρομή αναλυτικά.
5. Λήψη στον client της απάντησης, parsing του μηνύματος και ανάλογα με την επιλογή λειτουργίας του συστήματος από το χρήστη (στο Navigation Options) έχουμε τις εξής υποπεριπτώσεις χρήσης:
 - Route Summary: Εμφάνιση στην οθόνη των στοιχείων του συνολικού χρόνου και συνολικής απόστασης για τη διαδρομή.
 - Turn-by-turn directions: Εμφάνιση στην οθόνη των οδηγιών οδήγησης για τη διαδρομή.
 - Voice Directions Without Map: Πλοήγηση με φωνητικές οδηγίες και παρακολούθηση της θέσης του χρήστη ώστε ανά πάσα στιγμή να βρίσκεται εντός διαδρομής. Παράλληλα, στην οθόνη εμφανίζονται γραπτές οι οδηγίες οδήγησης μαζί με μία πυξίδα η οποία δείχνει τη σχετική θέση τρέχουσας θέσης – προορισμού.
 - Voice Directions With Map: Όμοια με την παραπάνω περίπτωση. Αντί όμως της πυξίδας εδώ εμφανίζεται η θέση του χρήστη πάνω στο χάρτη. Οι χάρτες της διαδρομής δεν είναι αποθηκευμένοι στη φορητή συσκευή αλλά λαμβάνονται κάθε φορά καλώντας την υπηρεσία “Presentation Service” στο server.

Εναλλακτική ροή 1: Σφάλμα κατά το parsing του μηνύματος στο server.

1. Δημιουργία και αποστολή μηνύματος σφάλματος από τον server προς τον client με παραμέτρους την περιγραφή και τον κωδικό του σφάλματος.
2. Λήψη μηνύματος σφάλματος από τον client και εμφάνισή του στην οθόνη.
3. Έξοδος από την εφαρμογή.

Εναλλακτική ροή 2: Σφάλμα κατά το parsing του μηνύματος στον client.

1. Εμφάνιση του σφάλματος στην οθόνη.
2. Έξοδος από την εφαρμογή.

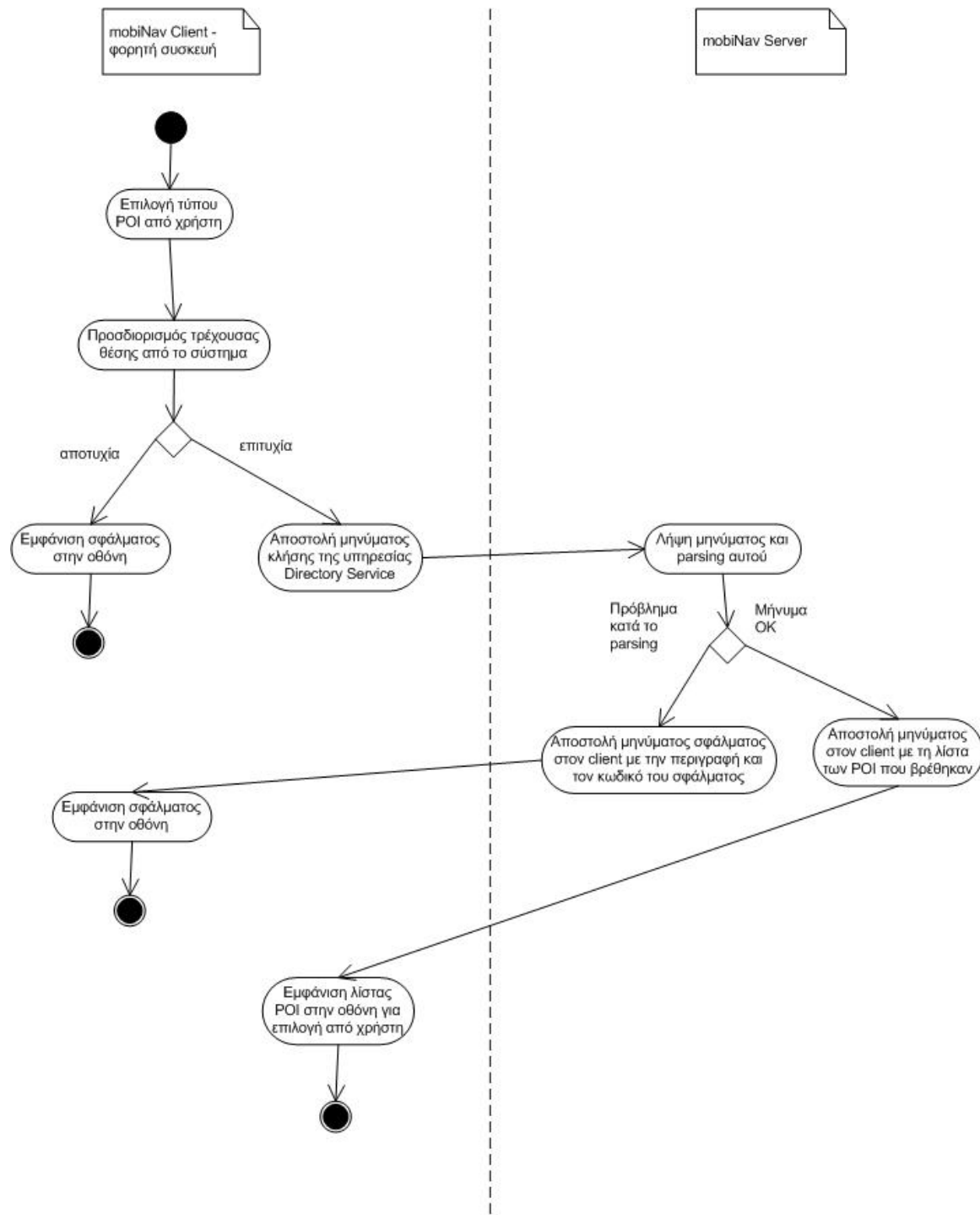
Εναλλακτική ροή 3: Σφάλμα κατά την πλοήγηση (ο χρήστης βγαίνει εκτός διαδρομής).

1. Δημιουργία και αποστολή μηνύματος κλήσης της υπηρεσίας “Route Service” για

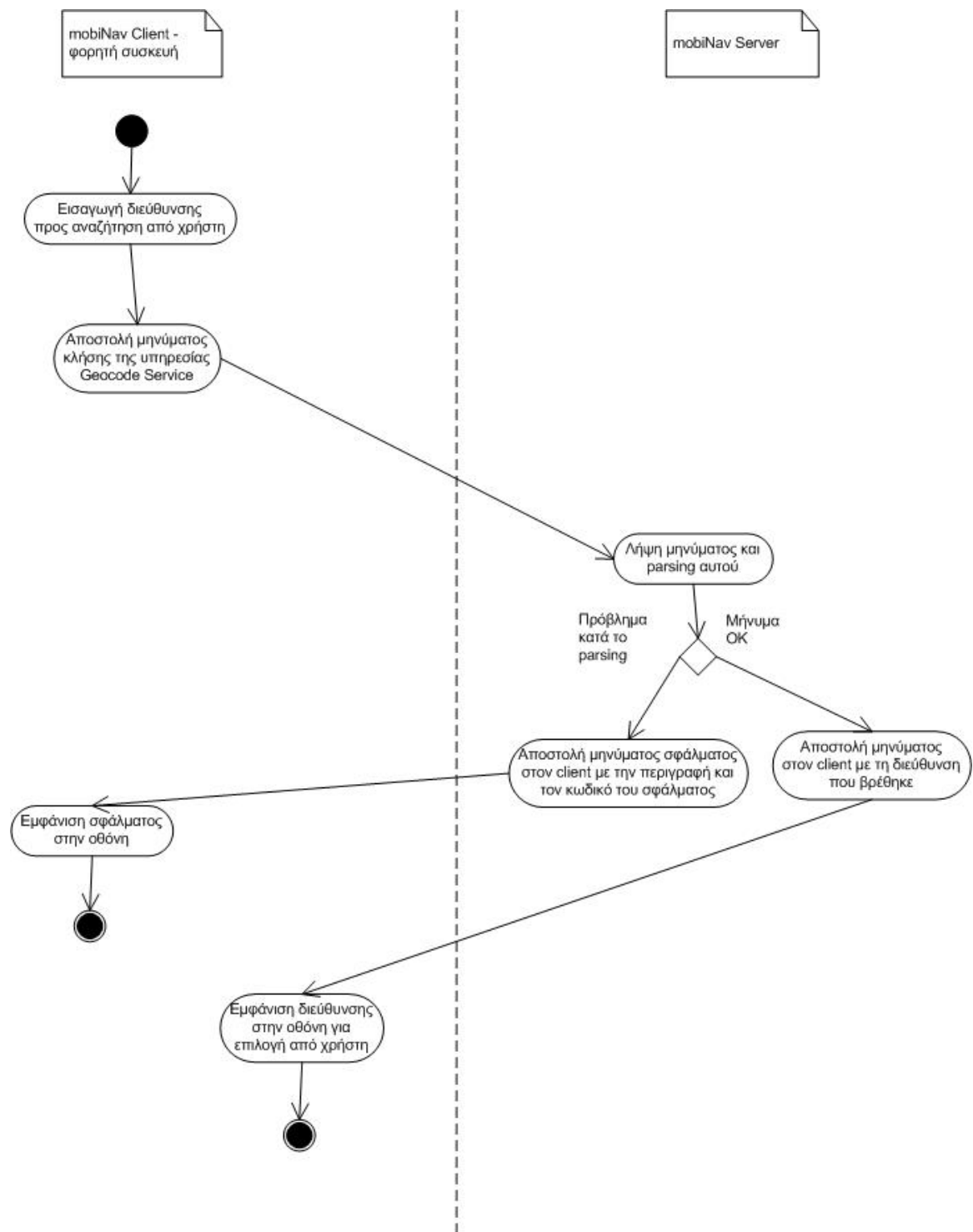
υπολογισμό νέας διαδρομής με παραμέτρους όμοιες με το βήμα 3 της βασικής ροής με μόνη διαφορά την αλλαγή αφετηρίας με την τρέχουσα θέση.

2. Επιστροφή στο βήμα 4 της βασικής ροής για υπολογισμό νέας διαδρομής πλέον.

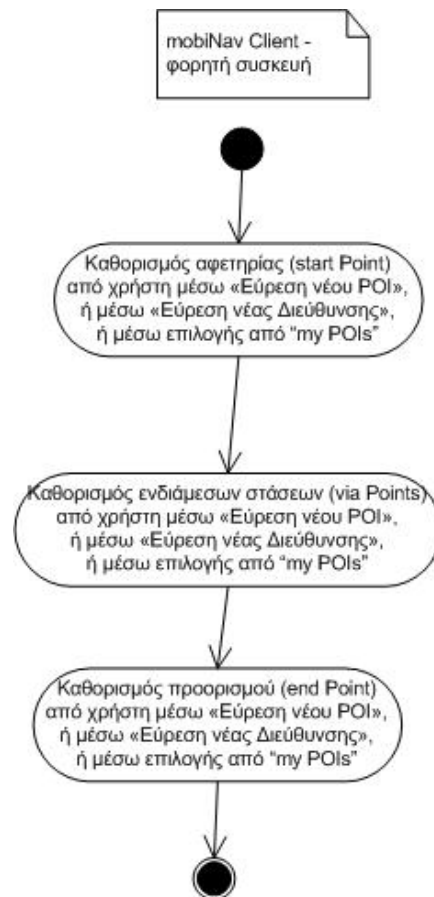
Συγκεντρωτικά όλες οι παραπάνω ροές φαίνονται στα παρακάτω διαγράμματα ροής:



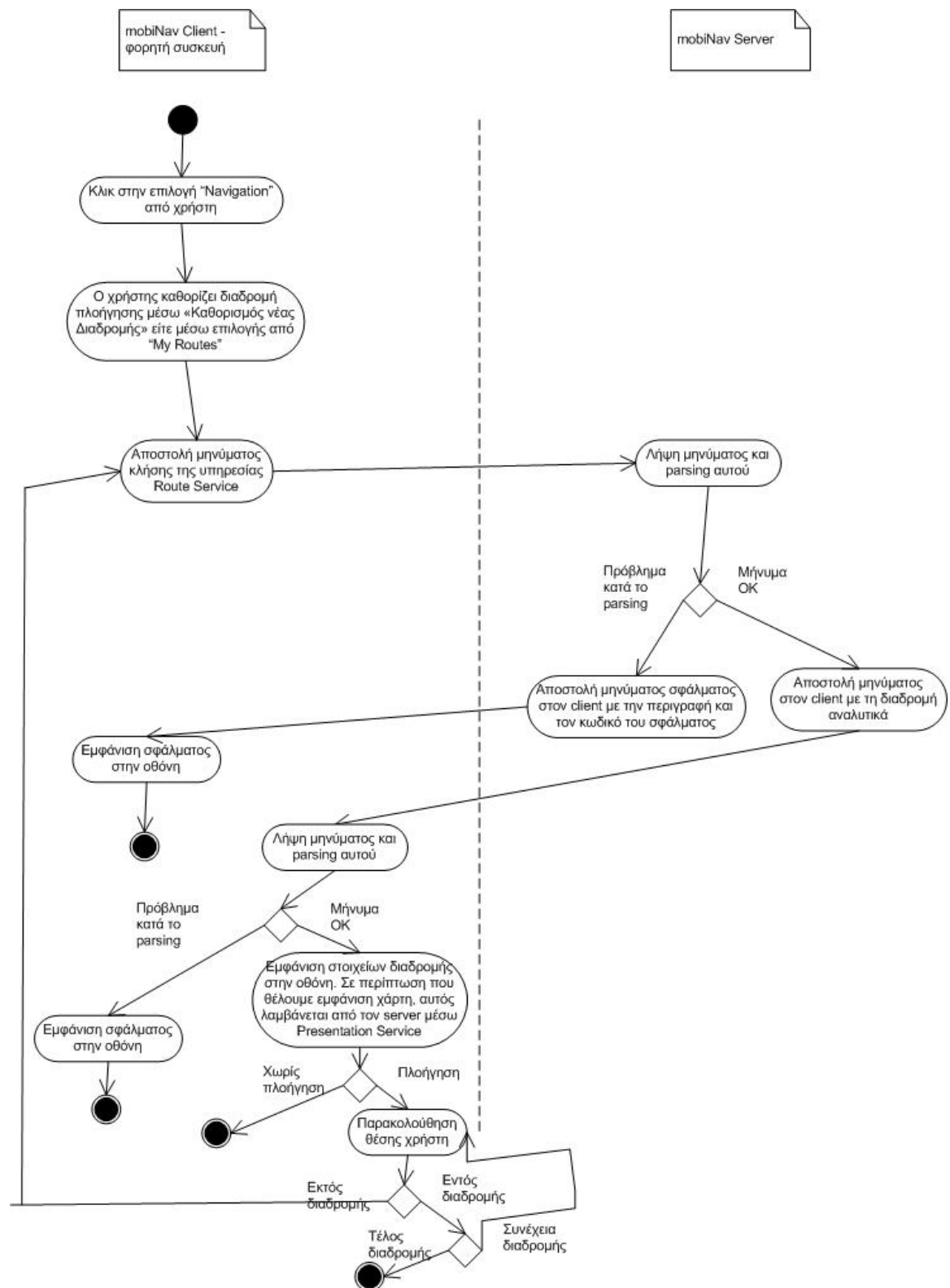
Εικόνα 25: Activity Diagram – Εύρεση Νέου POI



Εικόνα 26: Activity Diagram – Εύρεση Διεύθυνσης



Εικόνα 27: Activity Diagram – Καθορισμός Διαδρομής



Εικόνα 28: Activity Diagram – Navigate

5.2.4.2 Περίπτωση χρήσης 2: Real Time Info

5.2.4.2.1 Περιγραφή περίπτωσης χρήσης

Η περίπτωση χρήσης “Real Time Info” του συστήματός μας αφορά ουσιαστικά την περίπτωση κατά την οποία ο χρήστης θέλει να χρησιμοποιήσει το σύστημα για να δει αρχικά και στη συνέχεια να παρακολουθεί τη θέση του στο χάρτη, σε πραγματικό χρόνο.

Για να χρησιμοποιήσει ο χρήστης το σύστημα κατ’ αυτό τον τρόπο αρκεί στην αρχική οθόνη (βλ. Welcome Options Screen παραπάνω) να επιλέξει να πατήσει το κουμπί “Real Time Info”. Μόλις ο χρήστης κάνει αυτή την επιλογή, το σύστημα αρχικά προσδιορίζει την τρέχουσα θέση του χρήστη και στη συνέχεια καλώντας την υπηρεσία “Presentation Service” ζητά και λαμβάνει το αντίστοιχο τμήμα χάρτη για παρουσίαση στο χρήστη. Στη συνέχεια εξακολουθεί και παρακολουθεί τη θέση του χρήστη καθώς αυτός μετακινείται και επαναλαμβάνει όλη την παραπάνω διαδικασία. Όλα όσα περιγράψαμε εδώ, φαίνονται αναλυτικότερα μέσω των διαφόρων ροών γεγονότων.

5.2.4.2.2 Ροή γεγονότων

Βασική Ροή

1. Ο χρήστης κάνει κλικ στο κουμπί “Real Time Info” στην αρχική οθόνη (βλ οθόνη Welcome Options Screen παραπάνω).
2. Το σύστημα προσδιορίζει τη θέση του χρήστη. Αυτό γίνεται είτε μέσω εξωτερικής συσκευής Bluetooth GPS Receiver, είτε μέσω της υπηρεσίας Gateway Service του server.
3. Αποστολή στον server μηνύματος κλήσης της υπηρεσίας “Presentation Service” με αίτηση για το χάρτη της γύρω περιοχής από τη θέση του χρήστη (διαστάσεις χάρτη τουλάχιστον ίσες με τις διαστάσεις της οθόνης).
4. Λήψη στον server, parsing μηνύματος, δημιουργία και αποστολή απάντησης στον client με το ζητούμενο χάρτη.
5. Λήψη από τον client της απάντησης, parsing μηνύματος.
6. Απεικόνιση της θέσης του χρήστη πάνω στο χάρτη στην οθόνη. Στη συνέχεια γίνεται παρακολούθηση της θέσης του χρήστη. Μόλις ο χρήστης μετακινείται έξω από την περιοχή του χάρτη τον οποίο έχει η φορητή συσκευή, επιστρέφουμε στο βήμα 3 για αίτηση νέου χάρτη.

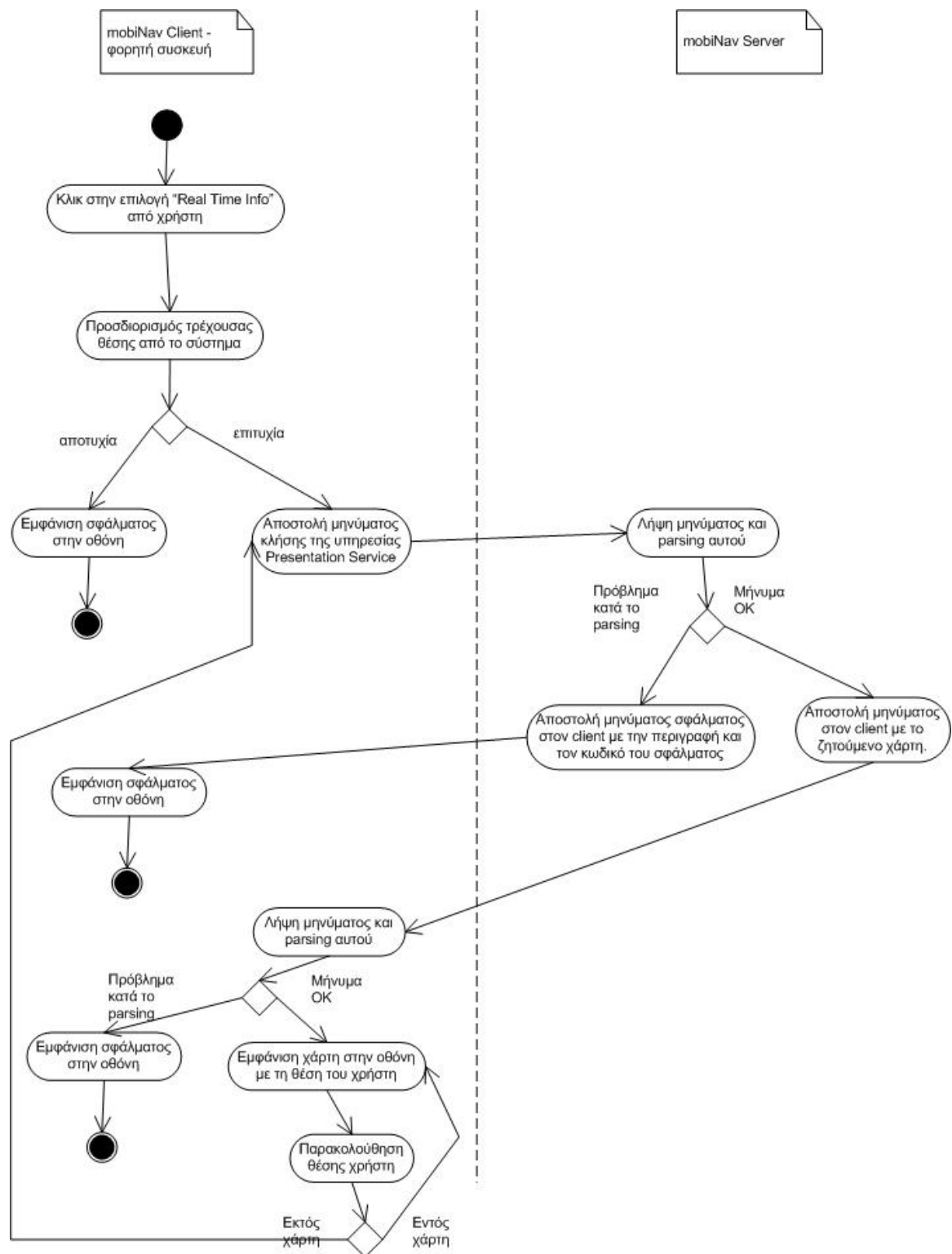
Εναλλακτική ροή 1: Σφάλμα κατά το parsing του μηνύματος στο server.

1. Δημιουργία και αποστολή μηνύματος σφάλματος από τον server προς τον client με παραμέτρους την περιγραφή και τον κωδικό του σφάλματος.
2. Λήψη μηνύματος σφάλματος από τον client και εμφάνισή του στην οθόνη.
3. Έξοδος από την εφαρμογή.

Εναλλακτική ροή 2: Σφάλμα κατά το parsing του μηνύματος στον client.

1. Εμφάνιση του σφάλματος στην οθόνη.
2. Έξοδος από την εφαρμογή.

Συγκεντρωτικά όλες οι παραπάνω ροές γεγονότων φαίνονται στο ακόλουθο διάγραμμα ροής:



Εικόνα 29: Activity Diagram – Real Time Info

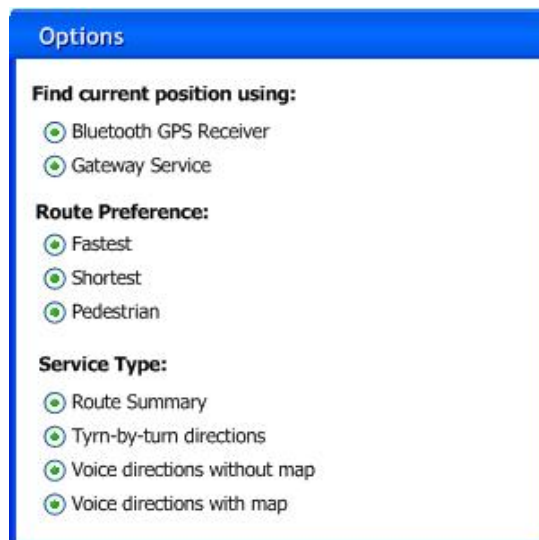
5.2.4.3 Περίπτωση χρήσης 3: *Navigation Options*

5.2.4.3.1 Περιγραφή περίπτωσης χρήσης

Η περίπτωση χρήσης “Navigation Options” αφορά στην περίπτωση που ο χρήστης καθορίζει τις επιλογές για τον τρόπο λειτουργίας του συστήματος. Οι επιλογές αυτές είναι τρεις.

- Η πρώτη έχει να κάνει με το πώς το σύστημα θα προσδιορίζει την τρέχουσα θέση του. Ο χρήστης μπορεί να επιλέξει να προσδιορίζεται η θέση του μέσω Bluetooth GPS Receiver, είτε μέσω της υπηρεσίας “Gateway Service” που παρέχεται από το server.
- Η δεύτερη επιλογή αφορά στην προτίμηση του χρήστη όταν ζητά διαδρομές πλοήγησης. Οι επιλογές που έχει είναι να ζητά τη γρηγορότερη διαδρομή από πλευράς χρόνου, τη συντομότερη από πλευράς απόστασης, ή τέλος διαδρομή για πεζό χρήστη.
- Η τρίτη επιλογή που μπορεί να κάνει ο χρήστης αφορά στη λειτουργία του συστήματος πλοήγησης και μπορεί να επιλέξει έναν από τους τέσσερις τρόπους λειτουργίας που αναλύσαμε παραπάνω: Route Summary, Turn-by-turn directions, Voice Directions Without Map, Voice Directions With Map.

Όλες οι παραπάνω επιλογές του χρήστη αποθηκεύονται στη φορητή συσκευή και κάθε φορά που εκκινεί η εφαρμογή φορτώνονται οι επιλογές που είχε κάνει ο χρήστης την τελευταία φορά. Τις επιλογές αυτές μπορεί να τις κάνει ο χρήστης επιλέγοντας το κουμπί “Navigation Options” στην αρχική σελίδα (βλ. Welcome Options Screen παραπάνω). Αφού επιλέξει το κουμπί αυτό, στη συνέχεια πρέπει να συμπληρώσει μια φόρμα της παρακάτω μορφής.



Εικόνα 30: Options Screen

5.2.4.3.2 Ροή γεγονότων

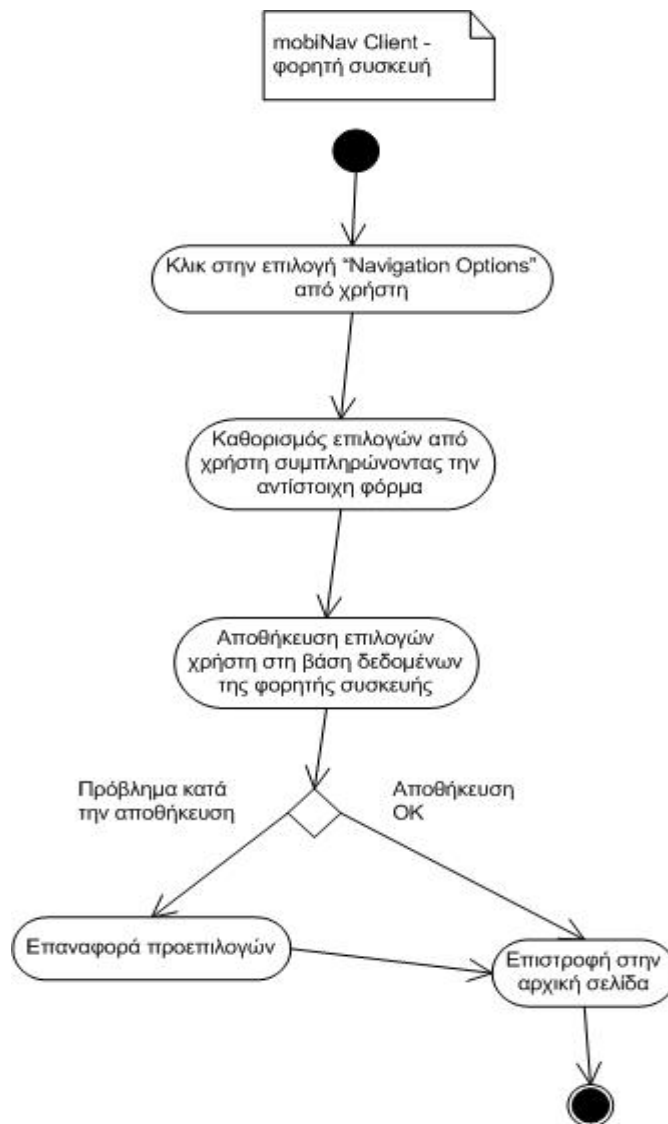
Βασική Ροή

1. Ο χρήστης κάνει κλικ στο κουμπί “Navigation Options” στην αρχική οθόνη (βλ οθόνη Welcome Options Screen παραπάνω).
2. Ο χρήστης συμπληρώνει με τις επιλογές του τη φόρμα της οποίας η μορφή φαίνεται παραπάνω (Options Screen).
3. Οι επιλογές του χρήστη αποθηκεύονται στη βάση δεδομένων της φορητής συσκευής.
4. Επιστροφή στην αρχική σελίδα (Welcome Options Screen).

Εναλλακτική Ροή: Σφάλμα κατά την αποθήκευση των επιλογών του χρήστη.

1. Επαναφέρονται οι προεπιλεγμένες από το σύστημα τιμές για όλες τις επιλογές.
2. Επιστροφή στην αρχική σελίδα (Welcome Options Screen).

Συγκεντρωτικά, οι παραπάνω ροές γεγονότων φαίνονται στο ακόλουθο διάγραμμα ροής:



Εικόνα 31: Activity Diagram – Navigation Options

5.3 Έγγραφο περιγραφής αρχιτεκτονικού σχεδίου

5.3.1 Εισαγωγή

Στο συγκεκριμένο έγγραφο γίνεται παρουσίαση της τεκμηρίωσης κατά YASM του Πισινού Μέρους ανάπτυξης του συστήματος “mobiNav”. Η υλοποίηση του συστήματος στηρίζεται σε δύο υπολογιστές συνδεδεμένους μεταξύ τους και μία φορητή συσκευή η οποία συνδέεται μέσω του δικτύου GPRS με τον έναν από τους υπολογιστές. Για την ακρίβεια, η φορητή συσκευή (κινητό τηλέφωνο, PDA) είναι η συσκευή στην οποία τρέχει ο client του

συστήματός μας, ενώ ο ένας από τους παραπάνω υπολογιστές παίζει ρόλο του application server για το σύστημά μας και είναι ο υπολογιστής στον οποίο τρέχουν τα web services που αναπτύσσουμε. Τέλος, ο δεύτερος υπολογιστής που αναφέραμε ουσιαστικά είναι εξωτερικός του συστήματός μας και είναι ο υπολογιστής στον οποίο είναι αποθηκευμένες όλες οι γεωγραφικές πληροφορίες τις οποίες διαθέτουμε μέσω των web services στον client. Είναι ουσιαστικά δηλαδή ένας GIS Server. Η τεκμηρίωση που ακολουθεί παρέχει μια αναλυτική περιγραφή του αρχιτεκτονικού σχεδίου του συστήματος “mobiNav” μέσω της περιγραφής των σημαντικότερων όψεών του.

5.3.1.1 Σκοπός

Σκοπός του εγγράφου αυτού είναι η παρουσίαση της αρχιτεκτονικής του λογισμικού συστήματος “mobiNav”, το οποίο παρέχει υπηρεσίες πλοήγησης στους χρήστες του. Στο παρόν έγγραφο δίνεται από μακροσκοπική σκοπιά η σχεδίαση του λογισμικού συστήματος καθώς και η απεικόνιση των απαιτήσεων που έχουν τεθεί στο έγγραφο «Προδιαγραφών Απαιτήσεων από το Λογισμικό», σε αρχιτεκτονικές όψεις. Είναι σημαντικό να σημειώσουμε ότι η περιγραφή του αρχιτεκτονικού σχεδίου αφορά μόνο στο κομμάτι που αποτελεί τον πυρήνα του συστήματός μας (φορητή συσκευή, web services server) και όχι στις εξωτερικές του συνιστώσες (GIS server).

5.3.1.2 Ενδιαφερόμενοι

Αποδέκτες του εγγράφου είναι αφενός οι χρήστες των υπηρεσιών που θα παρέχονται, και αφετέρου οι κατασκευαστές του λογισμικού συστήματος “mobiNav”.

5.3.1.3 Ορισμοί, ακρωνύμια και συντομεύσεις

Web Services: Διαδικτυακές υπηρεσίες

GPS: Geographical Positioning System

POI: Point of Interest (Σημείο ενδιαφέροντος)

GPRS: General Packet Radio Service

SSL: Secure Socket Layer

OGC: Open Geospatial Consortium

5.3.1.4 Αναφορές

Το έγγραφο «Προδιαγραφών Απαιτήσεων από το Λογισμικό Σύστημα» το οποίο παρουσιάστηκε στην προηγούμενη παράγραφο καθώς και η σχετική βιβλιογραφία στο αντίστοιχο κεφάλαιο του παρόντος τόμου.

5.3.2 Αρχιτεκτονικές αποφάσεις

Στη συνέχεια δίνονται οι σημαντικότερες αρχιτεκτονικές αποφάσεις που χρειάστηκε να ληφθούν για την υλοποίηση του λογισμικού συστήματος.

5.3.2.1 Πλατφόρμα (υλισμική, λογισμική)

Ως υλισμική πλατφόρμα, το σύστημά μας αποτελείται από έναν προσωπικό υπολογιστή (PC) και μία φορητή συσκευή (PDA ή κινητό τηλέφωνο). Ο προσωπικός υπολογιστής λειτουργεί ως server της εφαρμογής μας και παρέχει τα web services τα οποία αναπτύσσουμε. Η φορητή συσκευή αποτελεί τον πελάτη (client) του συστήματος, ο οποίος συνδέεται με τον server μέσω του δικτύου GPRS και καλεί τα web services τα οποία παρέχονται για να λάβει τα δεδομένα που ζητά ο χρήστης. Η φορητή αυτή συσκευή θα πρέπει να έχει σύστημα προσδιορισμού θέσης μέσω GPS είτε δυνατότητα σύνδεσης μέσω Bluetooth με εξωτερική συσκευή Bluetooth GPS Receiver. Σε αντίθετη περίπτωση ο προσδιορισμός θέσης θα γίνεται μέσω αντίστοιχου web service το οποίο θα παρέχεται από τον server αλλά θα έχουμε μειωμένη ακρίβεια στο αποτέλεσμα. Τέλος, αναφέρουμε την ανάγκη της σύνδεσης του προσωπικού υπολογιστή με έναν εξωτερικό GIS server ο οποίος θα έχει αποθηκευμένη όλη τη γεωγραφική πληροφορία που χρειάζεται να παρέχεται μέσω των web services. Μια εποπτική εικόνα των υλισμικών συνιστωσών δίνεται στη συνέχεια μέσω της πλατφορμικής όψης.

Ως λογισμική πλατφόρμα ανάπτυξης και εκτέλεσης χρησιμοποιούνται:

- ✓ Στον client: Η φορητή συσκευή θα πρέπει να μπορεί να εκτελέσει προγράμματα Java, συνεπώς θα πρέπει να υπάρχει Java Virtual Machine (JVM) και επιπλέον θα πρέπει η συγκεκριμένη JVM να υποστηρίζει τα εξής JSRs : JSR – 82 για Bluetooth, JSR – 172 για web services, xml parsing, JSR – 135 για mobile media API (για τις φωνητικές οδηγίες).

- ✓ Στον server: Στον server θα τρέχει ο Apache Tomcat (έκδοση 5.5) για την παραγωγή και αποστολή των SOAP/ XML μηνυμάτων ως responses στα requests του client. Επιπλέον χρησιμοποιείται μια μηχανή Java (συγκεκριμένα η Java Virtual Machine 1.5.0_06) για την εκτέλεση του κώδικα Java του sever (classes κτλ).

Να σημειώσουμε στο σημείο αυτό ότι το λειτουργικό σύστημα του client όσο και του server μπορεί να είναι οποιοδήποτε.

Για τη συντήρηση θα χρησιμοποιηθεί η ίδια λογισμική πλατφόρμα με αυτήν της ανάπτυξης και της εκτέλεσης. Δεν προβλέπεται χρησιμοποίηση κάποιου ιδιαίτερου λογισμικού για τη συντήρηση.

5.3.2.2 Σχολή / Γλώσσα προγραμματισμού

Στην ανάπτυξη του εν λόγω λογισμικού χρησιμοποιήθηκε η αντικειμενοστραφής σχολή προγραμματισμού. Ως γλώσσα προγραμματισμού θα χρησιμοποιηθεί η Java, και συγκεκριμένα η J2SE στην πλευρά του server και η J2ME στην πλευρά του client. Ο λόγος που επιλέχθηκε η Java είναι ότι η συγκεκριμένη γλώσσα υποστηρίζεται από όλους τους προσωπικούς υπολογιστές και σχεδόν από όλες τις φορητές συσκευές. Έτσι, το σύστημα που θα αναπτυχθεί θα μπορεί να τρέχει σε οποιαδήποτε πλατφόρμα και πάνω από οποιοδήποτε λειτουργικό σύστημα.

5.3.2.3 Κωδική γλώσσα

Η κωδική γλώσσα είναι η Java μαζί με τις διάφορες διαπροσωπίες (interfaces) που απαιτούνται από τις απαραίτητες βιβλιοθήκες για την υποστήριξη γραφικού περιβάλλοντος από πλευράς client, την επικοινωνία μεταξύ client και server, την επικοινωνία του client με εξωτερική συσκευή Bluetooth GPS Receiver (JSR 82), την υποστήριξη κλήσης web services από φορητή συσκευή (JSR 172) κτλ.

5.3.2.4 Ρυθμός

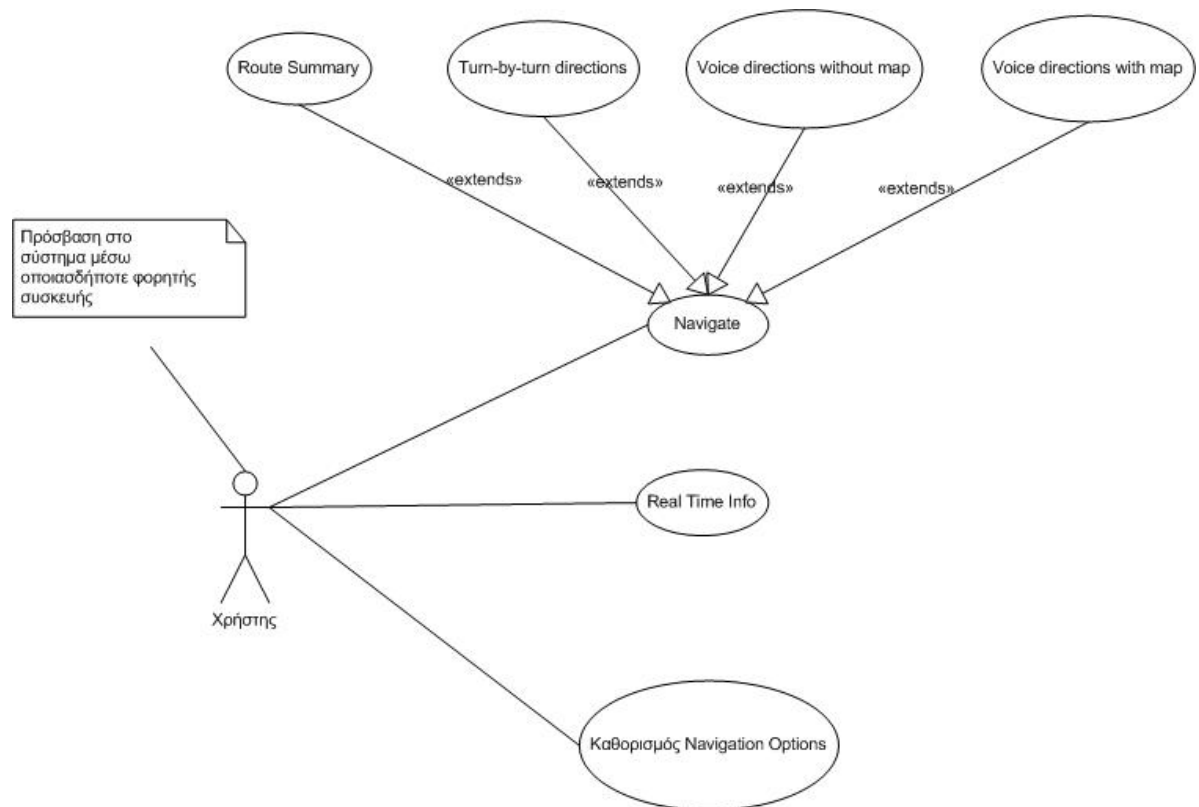
Ο κύριος ρυθμός που χρησιμοποιείται από το σύστημά μας είναι ο event – driven, δηλαδή η εκτέλεση των διαφόρων διεργασιών πυροδοτείται από γεγονότα. Συγκεκριμένα, ο χρήστης ελέγχει την εκτέλεση των διεργασιών στον application server μέσω της γραφικής

διαπροσωπείας (Graphical User Interface) που του προσφέρει το σύστημα. Για τις περισσότερες από τις υπηρεσίες ο χρήστης εισάγει στοιχεία σε διάφορες φόρμες και τα αποστέλλει με το πάτημα ενός κουμπιού της φορητής συσκευής στον server στον οποίο γίνονται οι κλήσεις των διάφορων web services. Υπάρχουν βέβαια περιπτώσεις στις οποίες ο client από μόνος του καλεί υπηρεσίες, αλλά και πάλι η κλήση γίνεται εξαιτίας κάποιας ενέργειας του χρήστη. Χαρακτηριστικό παράδειγμα αποτελεί η περίπτωση στην οποία ο client δίνει οδηγίες πλοήγησης στον χρήστη και ο χρήστης παρεκκλίνει της πορείας του αλλάζοντας διαδρομή. Τότε αυτόματα ο client ζητά από τον server υπολογισμό νέας διαδρομής καλώντας το αντίστοιχο service.

5.3.3 Αρχιτεκτονικές Όψεις

5.3.3.1 Περιβαλλοντική / Υπηρεσιακή

Στην περιβαλλοντική / υπηρεσιακή όψη φαίνονται οι δράστες (actors) του συστήματος καθώς και οι υπηρεσίες τις οποίες αυτό προσφέρει. Θα πρέπει να σημειώσουμε στο σημείο αυτό ότι με τον όρο υπηρεσίες εδώ εννοούμε τις υπηρεσίες που παρέχει το client – side της εφαρμογής προς τον τελικό χρήστη και όχι τις υπηρεσίες τις οποίες υλοποιεί ο server και τις οποίες καλεί ο client. Δηλαδή, υπηρεσίες προς το χρήστη του συστήματός μας εδώ θεωρούνται οι υπηρεσίες πλοήγησης, απεικόνισης σε χάρτη της θέσης του χρήστη σε πραγματικό χρόνο και τον ορισμό επιλογών πλοήγησης. Οι υπηρεσίες αυτές βασίζονται φυσικά στα τέσσερα web services που υλοποιούνται στον server και δεν φαίνονται εδώ. Ακολουθεί το διάγραμμα περιπτώσεων χρήσης (use case diagram) που αποτελεί την περιβαλλοντική / υπηρεσιακή όψη του συστήματος.

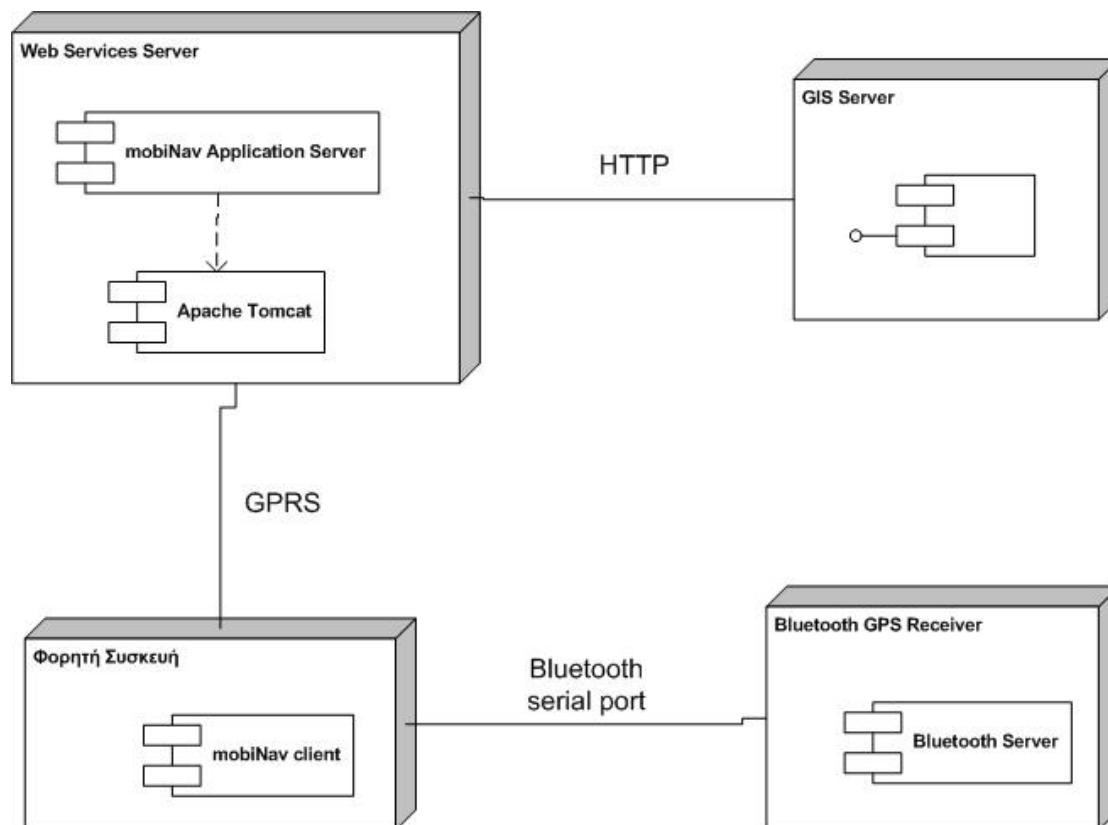


Εικόνα 32: Use Case Diagram

Περιγραφή και ανάλυση του παραπάνω διαγράμματος έγινε στο «Έγγραφο Προδιαγραφών Απαιτήσεων από το Λειτουργικό Σύστημα».

5.3.3.2 Πλατφορμική

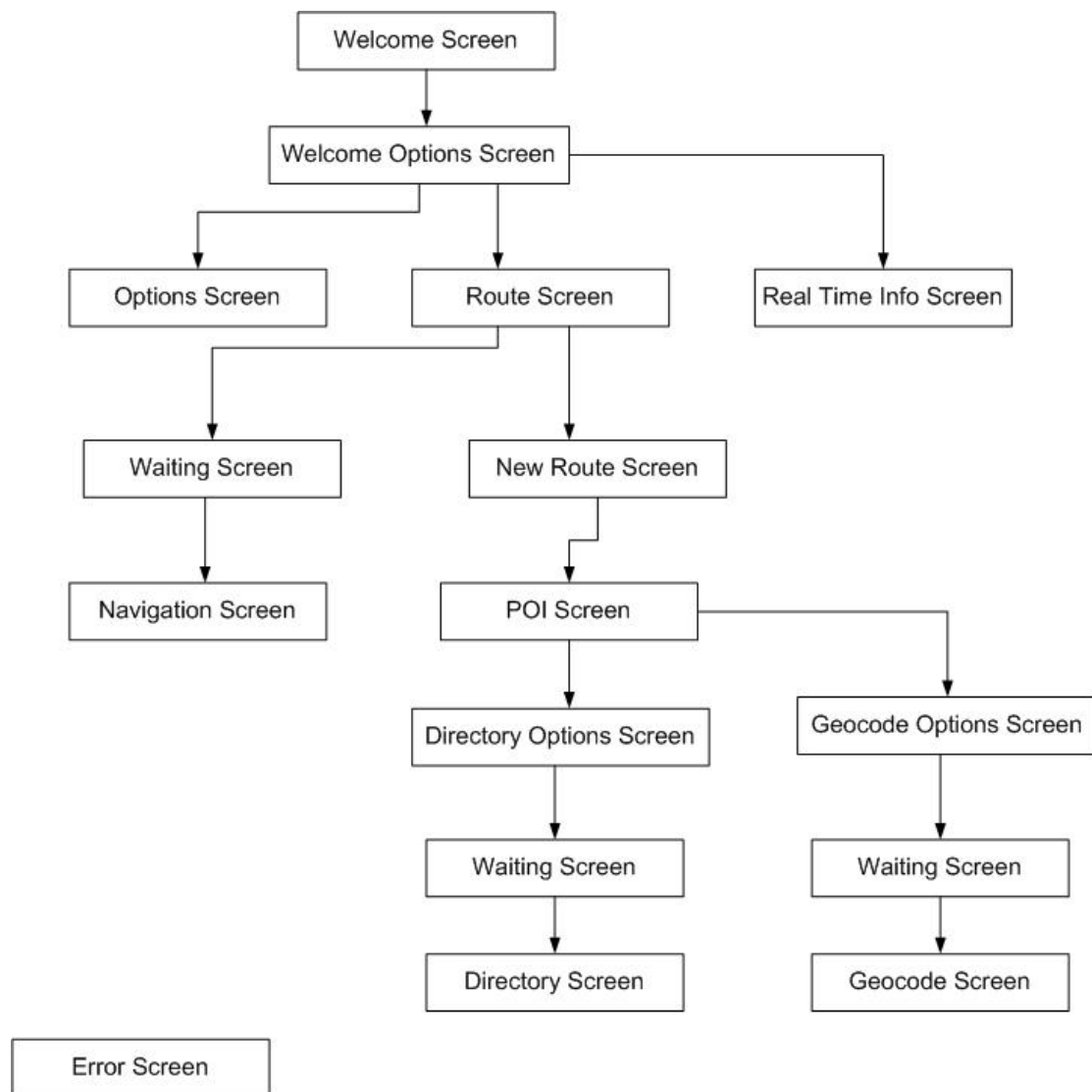
Στο επόμενο διάγραμμα παρουσιάζουμε την πλατφορμική όψη του συστήματός μας. Η πλατφορμική όψη περιγράφει με ένα παραταξιακό διάγραμμα (deployment diagram) της UML, τους υπολογιστές καθώς και τις άλλες συσκευές που λαμβάνουν μέρος στην εκτέλεση του mobiNav καθώς και τα πρωτόκολλα της μεταξύ τους επικοινωνίας. Ακολουθεί λοιπόν το παραταξιακό διάγραμμα με βάση τα όσα ειπώθηκαν στην ενότητα για λογισμική και υλισμική πλατφόρμα.



Εικόνα 33: Deployment Diagram

5.3.3.3 Χρηστική

Σε αυτή την ενότητα παρουσιάζονται οι διάφορες οθόνες που στην πραγματικότητα θα χρησιμοποιεί ο χρήστης του συστήματος mobiNav για να του παρέχονται οι διάφορες υπηρεσίες. Παρουσιάζεται έτσι το σύστημα από τη σκοπιά του τρόπου χρήσης του. Στο σχήμα που ακολουθεί φαίνεται το σύνολο των οθονών του User Interface του συστήματος και ακολουθεί η περιγραφή τους.



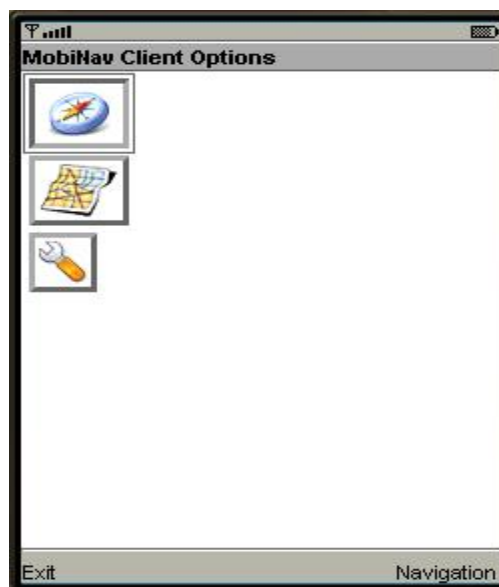
Εικόνα 34: Οθόνες του συστήματος mobiNav

Η οθόνη που ακολουθεί εμφανίζεται στην εκκίνηση της εφαρμογής για μερικά δευτερόλεπτα στο χρήστη και στη συνέχεια αυτόματα μεταβαίνει στην επόμενη οθόνη η οποία είναι ουσιαστικά η αρχική οθόνη του συστήματός μας.



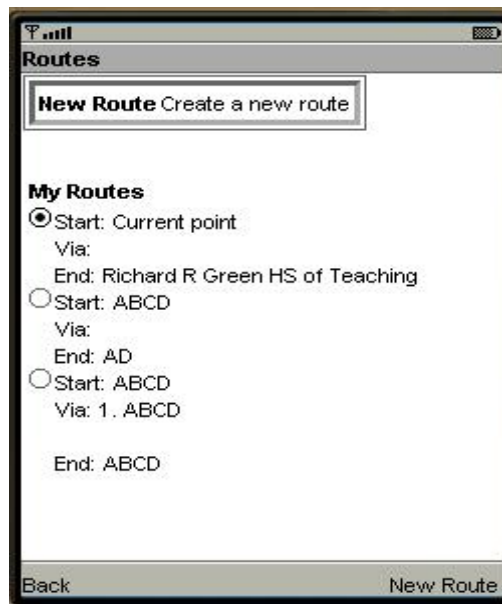
Εικόνα 35: Χρηστική όψη - Οθόνη Welcome Screen

Ακολουθεί η αρχική οθόνη του συστήματός μας από την οποία ο χρήστης μπορεί να επιλέξει τι θέλει να κάνει (Navigation, Real Time Info, Navigation Options).



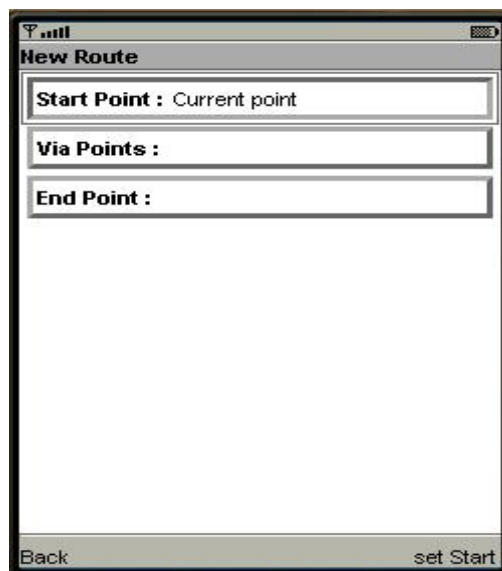
Εικόνα 36: Χρηστική όψη - Οθόνη Welcome Options Screen

Στην περίπτωση που ο χρήστης επιλέξει να πλοηγηθεί (Navigation) τότε ακολουθεί η εξής οθόνη καθορισμού διαδρομής.



Εικόνα 37: Χρηστική όψη - Οθόνη Route Screen

Στη συνέχεια είτε επιλέγει μια διαδρομή από το “My Routes” και ξεκινά η πλοήγηση είτε επιλέγει καθορισμό νέας διαδρομής οπότε εμφανίζεται η επόμενη οθόνη:



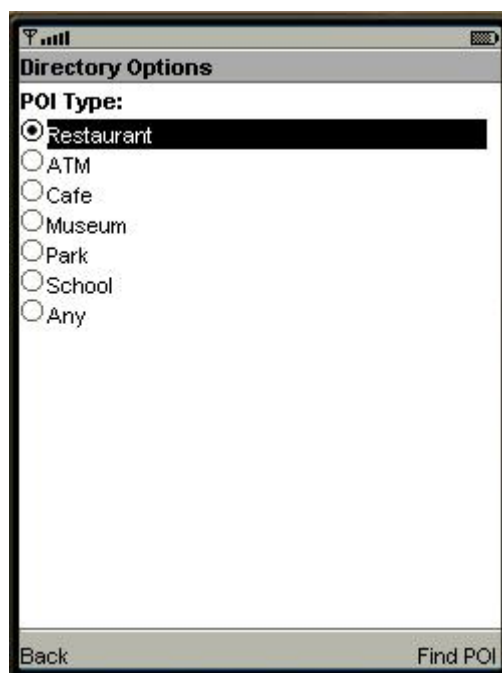
Εικόνα 38: Χρηστική όψη - Οθόνη New Route Screen

Στη συνέχεια μπορεί να επιλέξει να καθορίσει αφετηρία, ενδιάμεσες στάσεις, προορισμό με τη βοήθεια της παρακάτω οθόνης.



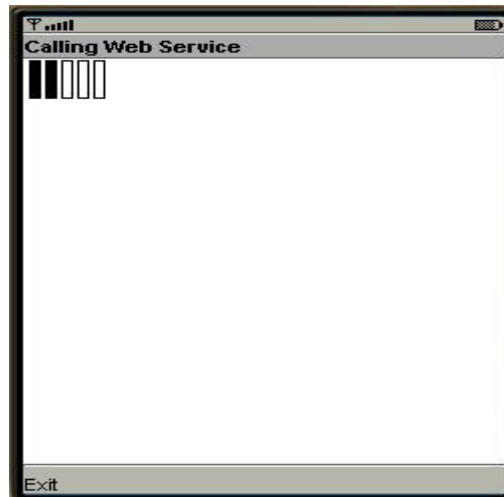
Εικόνα 39: Χρηστική όψη - Οθόνη POI Screen

Ο χρήστης εδώ μπορεί να επιλέξει ένα αποθηκευμένο σημείο στα “My POIs” είτε να βρει ένα σημείο ενδιαφέροντος κοντά, είτε κάποια διεύθυνση. Τα παραπάνω τα κάνει με κλήσεις στις υπηρεσίες Directory και Geocode Service αντίστοιχα. Για να γίνουν αυτές οι κλήσεις υπάρχουν οι ακόλουθες οθόνες προς το χρήστη.



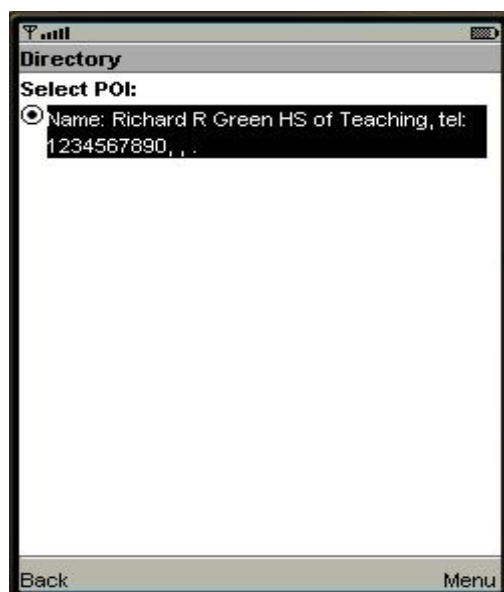
Εικόνα 40: Χρηστική όψη - Οθόνη Directory Options Screen

Αφού ο χρήστης επιλέξει τύπο POI προς αναζήτηση, καλείται η υπηρεσία “Directory Service” και στο μεταξύ, ο χρήστης βλέπει μια οθόνη αναμονής:



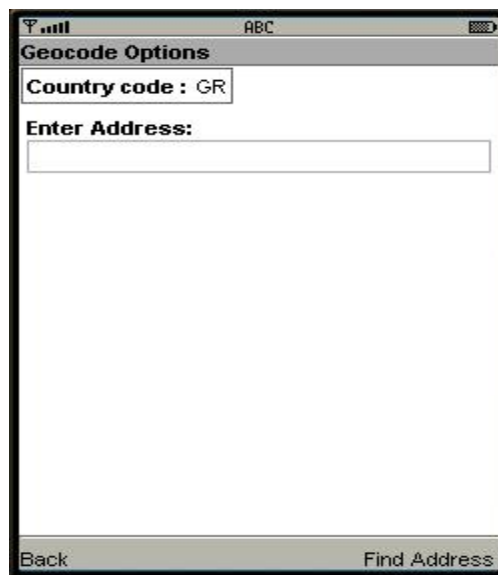
Εικόνα 41: Χρηστική όψη - Οθόνη Waiting Screen

Η οθόνη Waiting Screen εμφανίζεται κάθε φορά που έχουμε κλήση ενός web service και χρειάζεται ο χρήστης να περιμένει να δει κάποιο αποτέλεσμα. Μόλις είναι όλα έτοιμα, εμφανίζεται η παρακάτω οθόνη με τα αποτελέσματα της κλήσης της υπηρεσίας:



Εικόνα 42: Χρηστική όψη - Οθόνη Directory Screen

Ομοίως έχουμε αναζήτηση για κάποια διεύθυνση με τις ακόλουθες οθόνες.

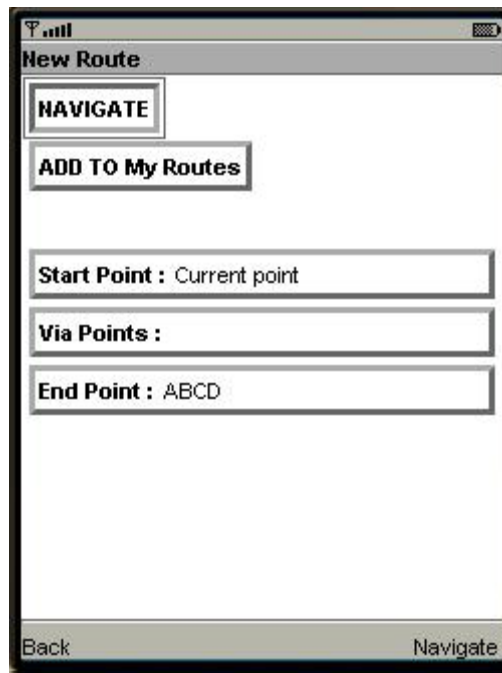


Εικόνα 43: Χρηστική όψη - Οθόνη Geocode Options Screen



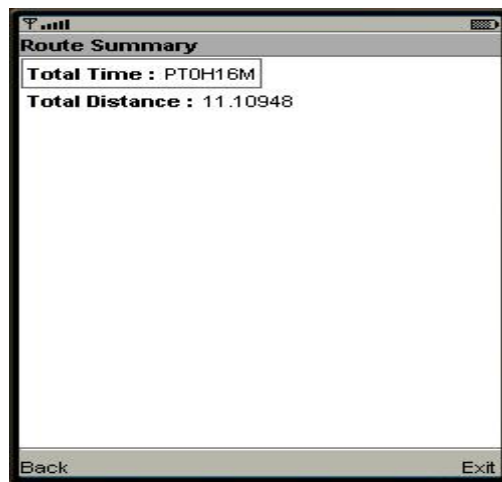
Εικόνα 44 : Χρηστική όψη - Οθόνη Geocode Screen

Όταν καθοριστεί η νέα διαδρομή ο χρήστης βλέπει την παρακάτω οθόνη η οποία είναι ίδια με την New Route Screen όμως πλέον ο χρήστης μπορεί να πατήσει κουμπί για έναρξη πλοήγησης είτε να αποθηκεύσει τη διαδρομή στο My Routes.

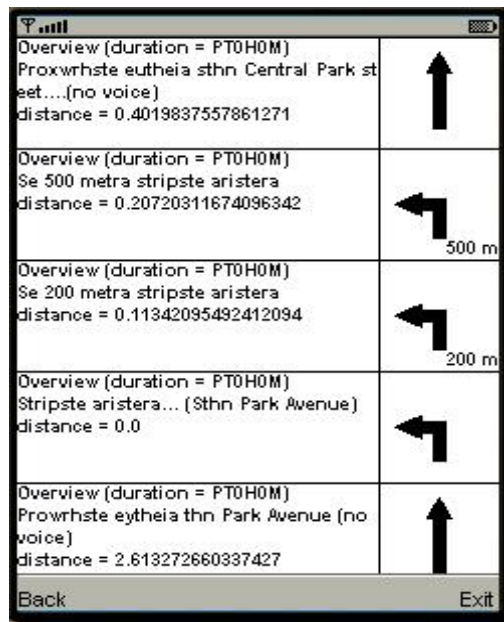


Εικόνα 45: Χρηστική όψη - Οθόνη New Route Screen 2

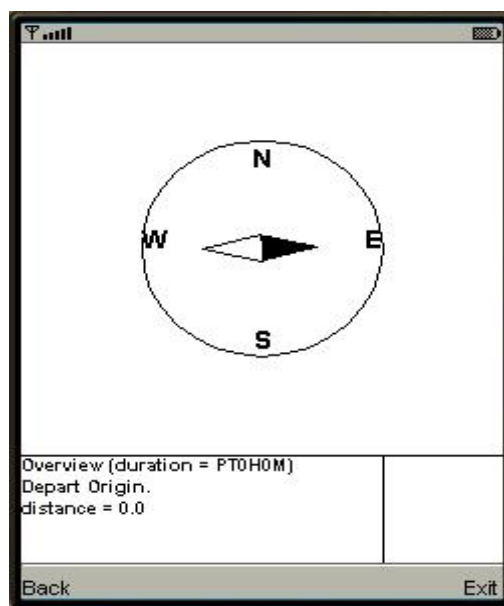
Σε περίπτωση που ο χρήστης επιλέξει πλοήγηση θα γίνει κλήση του web service “Route Service” και ανάλογα με τον τύπο υπηρεσίας που έχει επιλέξει ο χρήστης θα εμφανιστεί τελικά μία από τις τέσσερις ακόλουθες οθόνες.



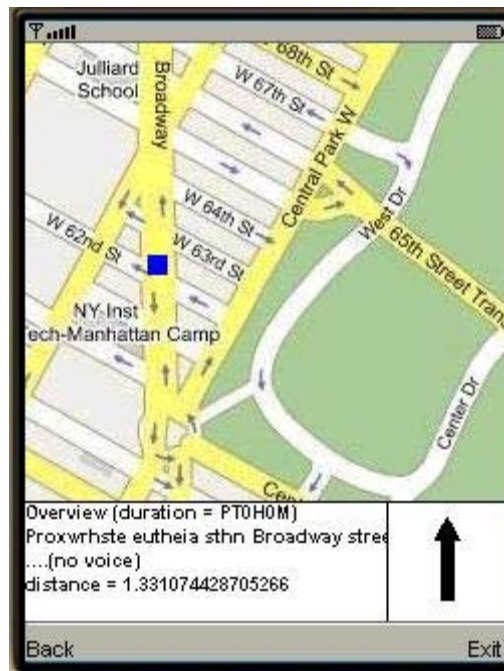
Εικόνα 46: Χρηστική όψη - Οθόνη Navigation Screen 1 (Route Summary)



Εικόνα 47: Χρηστική όψη - Οθόνη Navigation Screen 2 (Turn-by-turn directions)



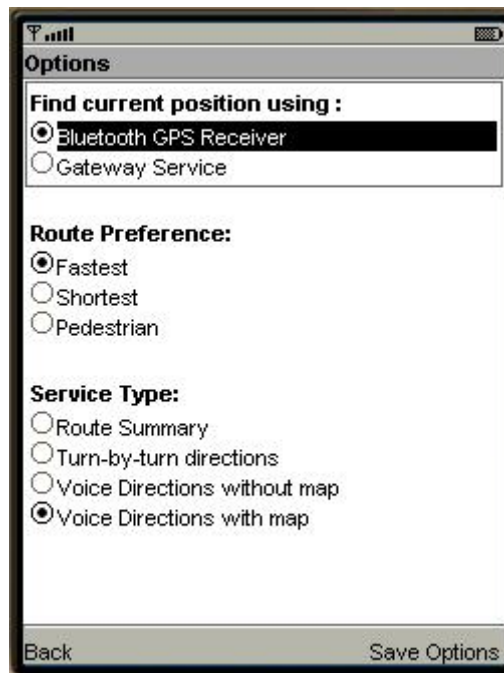
Εικόνα 48: Χρηστική όψη - Οθόνη Navigation Screen 4 (Voice Directions Without Map)



Εικόνα 49: Χρηστική όψη - Οθόνη Navigation Screen 4 (Voice Directions With Map)

Αναφέρουμε στο σημείο αυτό ότι αν ο χρήστης επιλέξει στην αρχική οθόνη “Real Time Info” η οθόνη που θα παρουσιαστεί είναι πανομοιότυπη με την Navigation Screen 4 παραπάνω με μόνη διαφορά ότι δε θα έχει οδηγίες οδήγησης μέσα αλλά θα έχει μόνο το χάρτη.

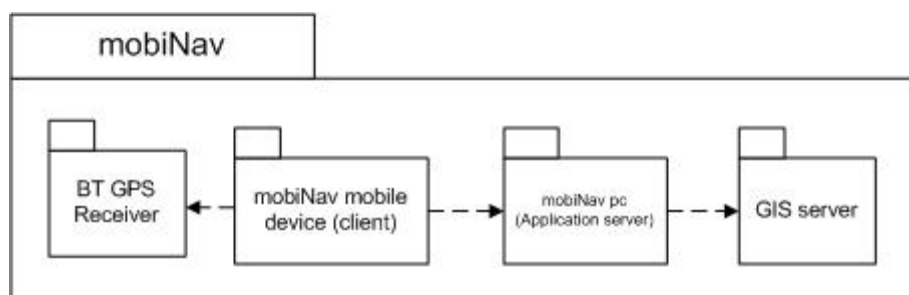
Τέλος, όταν ο χρήστης επιλέγει να καθορίζει τις επιλογές πλοήγησης (Navigation Options) στην αρχική σελίδα, τότε καλείται να συμπληρώσει την παρακάτω φόρμα για να αποθηκευτούν οι επιλογές του.



Εικόνα 50: Χρηστική όψη - Οθόνη Options Screen

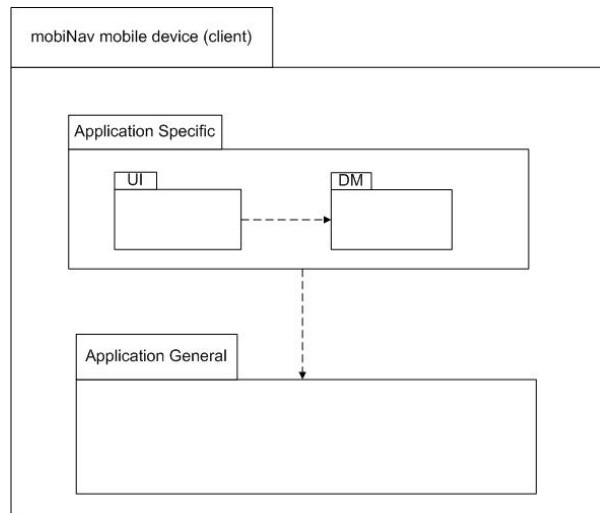
5.3.3.4 Δομική

Σε αυτή την όψη φαίνονται τα μέρη του mobiNav σε μακροσκοπικό επίπεδο καθώς και ποιες είναι οι εξαρτήσεις τους. Σημειώνουμε πάλι ότι στο παρακάτω διάγραμμα φαίνεται η δομή μόνο του πυρήνα του συστήματός μας και δεν περιλαμβάνεται η δομική αποσύνθεση του εξωτερικού GIS Server.



Εικόνα 51: Δομική όψη συστήματος mobiNav (επίπεδο 0)

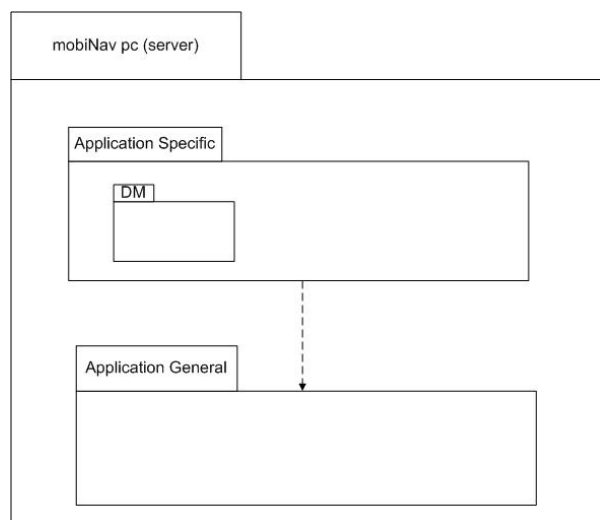
Στο επίπεδο 0 της δομικής όψης φαίνονται οι υπολογιστές του mobiNav και οι εξαρτήσεις τους. Όπως παρατηρούμε η φορητή συσκευή μπορεί να λαμβάνει πληροφορίες τόσο από το εξωτερικό Bluetooth GPS Receiver όσο και από τον server του συστήματός μας. Ο server μας με τη σειρά του λαμβάνει πληροφορίες από τον εξωτερικό GIS server.



Εικόνα 52: Δομική όψη του client του συστήματος mobiNav (επίπεδο 1)

Στην παραπάνω όψη, το ειδικό μέρος (application specific) αποτελείται από τα παρακάτω πακέτα (packages):

- UI (User Interface), με το οποίο γίνεται η επικοινωνία με το χρήστη. Το UI αποτελείται από το σύνολο των οθονών που περιγράψαμε στη χρηστική όψη παραπάνω.
- DM (Data Management), το οποίο περιλαμβάνει όλες εκείνες τις κλάσεις (Java classes) που είναι υπεύθυνες για τον χειρισμό των δεδομένων, τόσο αυτών που ανταλλάσσονται μεταξύ client και server, όσο και εκείνων τα οποία επεξεργάζεται ο client τοπικά.

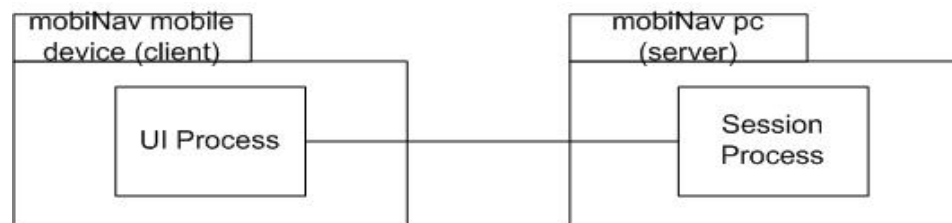


Εικόνα 53: Δομική όψη του server του συστήματος mobiNav (επίπεδο 1)

Όπως παρατηρούμε από την παραπάνω δομική όψη για τον server, ο server περιλαμβάνει κλάσεις μόνο για Data Management και δεν έχει διαπροσωπεία προς τον χρήστη (User Interface). Αυτό συμβαίνει διότι ο χρήστης δεν επικοινωνεί άμεσα με τον server αλλά επικοινωνεί μαζί του έμμεσα μέσω του client.

5.3.3.5 Εκτελεστική

Σε αυτή την όψη φαίνεται ο τρόπος λειτουργίας του λογισμικού που τρέχει σε κάθε υπολογιστή. Εμφανίζεται ως ένα σύνολο εκτελέσιμων οντοτήτων (run time entities), που όλες άμεσα ή έμμεσα υποστηρίζουν την αυτοματοποίηση των δεδομενικών εργασιών του συστήματος, συνεργαζόμενες μεταξύ τους και καταγράφονται ως ένα διάγραμμα που οι κόμβοι του είναι ενεργές κλάσεις (active classes) της UML.



Εικόνα 54: Εκτελεστική όψη mobiNav

5.4 Έγγραφο περιγραφής του λεπτομερούς σχεδίου

5.4.1 Εισαγωγή

5.4.1.1 Σκοπός

Σκοπός αυτού του εγγράφου είναι η παρουσίαση του Λεπτομερούς Σχεδίου του συστήματος mobiNav.

5.4.1.2 Ενδιαφερόμενοι

Ενδιαφερόμενοι αυτού του εγγράφου είναι ο αρχιτέκτων του συστήματος, οι σχεδιαστές οι προγραμματιστές και οι ελεγκτές.

5.4.1.3 Βιβλιογραφία

Τα έγγραφα «Έγγραφο Προδιαγραφών Απαιτήσεων από το Λογισμικό Σύστημα», «Έγγραφο Περιγραφής της Αρχιτεκτονικής του Συστήματος».

5.4.2 Σχεδιαστικές αποφάσεις

Οι σχεδιαστικές αποφάσεις που πήραμε είναι οι εξής:

- Από τα πολλά και διαφορετικά πρότυπα που δίνει ο Open Geospatial Consortium (OGC) για τη διάθεση χωρικής πληροφορίας επιλέξαμε το **OpenGIS Location Services (OpenLS): Core Services, OGC #05-016**. Το πρότυπο αυτό περιγράφει τα βασικά web services (core services) τα οποία θα πρέπει να υλοποιεί ένας server ο οποίος παρέχει γεωγραφικές πληροφορίες (χάρτες, διαδρομές, σημεία ενδιαφέροντος κτλ) προς τους clients. Τα βασικά αυτά web services είναι τα εξής: **Directory Service, Gateway Service, Location Utility Service (Geocoder/Reverse Geocoder), Presentation Service, Route Service**. Ο server μας λοιπόν παρέχει όλες τις παραπάνω υπηρεσίες και τα μηνύματα που ανταλλάσσονται για την κλήση των υπηρεσιών αυτών είναι xml έγγραφα τα οποία ακολουθούν τα xml schemas που δίνονται σε αυτό το πρότυπο του OGC. Για την ακρίβεια, χρησιμοποιείται ένα ειδικό schema για τα xml αρχεία το οποίο δίνει ο OGC με την ονομασία **XLS (XML for Location Services)**. Σημειώνουμε επίσης, πως οι υπηρεσίες που θα παρέχουμε αποφασίσαμε να είναι SOAP/XML Web Services δηλαδή τα μηνύματα ανταλλάσσονται μέσω SOAP.
- Επειδή έχουμε ανταλλαγή γεωγραφικής πληροφορίας μεταξύ client και server, οι συντεταγμένες θέσης οι οποίες αναφέρονται σε κάποιο σημείο στην επιφάνεια της γης ανταλλάσσονται ως (γεωγραφικό μήκος, γεωγραφικό πλάτος) ή (longitude, latitude). Η μορφή αυτή της πληροφορίας ενώ είναι βολική για να προσδιορίσουμε τη θέση ενός σημείου πάνω στη σφαιρική επιφάνεια της γης, δε βολεύει στο να γίνουν διάφοροι υπολογισμοί όπως η απόσταση μεταξύ δύο σημείων κτλ. Έτσι αποφασίσαμε να μετατρέπουμε τη δυνάδα τιμών (longitude, latitude) σε (x,y) συντεταγμένες σε καρτεσιανό επίπεδο όταν θέλουμε να κάνουμε διάφορους υπολογισμούς. Για λόγους απλότητας αφενός, καθώς και για λόγους περιορισμένων πόρων από τη φορητή συσκευή στην εκτέλεση πολύπλοκων υπολογισμών με πραγματικούς αριθμούς, αποφασίσαμε η μετατροπή αυτή να γίνεται προσεγγιστικά. Χρησιμοποιούμε λοιπόν

τους παρακάτω προσεγγιστικούς τύπους: $x = 111000 * \cos(\text{latitude}) * \text{longitude}$, $y = 111000 * \text{latitude}$, όπου τα x , y που λαμβάνουμε είναι σε μέτρα.

- Όταν έχουμε μεταφορά χάρτη από τον server προς τον client μετά από αίτηση του δεύτερου, αποφασίσαμε ο χάρτης να μεταφέρεται και αυτός μέσα σε μήνυμα SOAP όπως άλλωστε και όλα τα υπόλοιπα δεδομένα. Εάν ο χάρτης είναι σε binary format θα μεταφέρεται ως string κωδικοποιημένο σε base 64.
- Στην περίπτωση που ο client ζητάει διαδρομή από το server για πλοήγηση με χάρτη, οι χάρτες της διαδρομής αποφασίστηκε να μη μεταφέρονται εξ αρχής με τη διαδρομή αλλά όποτε χρειάζεται ο client κάποιο χάρτη να τον λαμβάνει με νέα αίτηση μέσω Presentation Service. Αυτή η απόφαση έγινε, από τη μία για λόγους ταχύτητας απόκρισης του συστήματος στην αρχή της πλοήγησης και από την άλλη για να μην κατεβάζει ο client πολλά δεδομένα τα οποία τελικά μπορεί να μη χρειαστεί. Για παράδειγμα, αν κατά την πλοήγηση ο χρήστης βρεθεί εκτός διαδρομής και πρέπει να ξανακαλέσει service για λήψη νέας διαδρομής θα έπρεπε να κατεβάσει νέους χάρτες αφού πιθανό οι προηγούμενοι να μην του κάνουν για την απεικόνιση της νέας διαδρομής.
- Σχετικά με τις φωνητικές οδηγίες τις οποίες θα δίνει ο client κατά την πλοήγηση, αποφασίσαμε τα εξής. Όλα τα σχετικά αρχεία ήχου θα είναι αποθηκευμένα ως resources στην εφαρμογή, δηλαδή στη φορητή συσκευή. Το ποια αρχεία θα παίζει κάθε φορά το κινητό κατά την πλοήγηση θα αποφασίζεται μετά από parsing μιας εντολής (μηνύματος) η οποία θα περιλαμβάνεται στο περιεχόμενο του sub-element Instruction του element RouteInstruction του XLS εγγράφου response του Route Service (βλ. OpenGIS Location Services (OpenLS): Core Services, OGC #05-016). Δηλαδή μια οδηγία (Instruction) που θα επιστρέφει ο server θα πρέπει να είναι της εξής μορφής: [Εντολή];[Γραπτή Οδηγία], όπου [Εντολή] είναι η εντολή την οποία κάνει parsing η εφαρμογή client και καθορίζει ποια από τα αρχεία ήχου θα πρέπει να παιχτούν. Η [Γραπτή Οδηγία] είναι η οδηγία σε μορφή κειμένου η οποία είναι αναγνώσιμη μόνο από τον άνθρωπο και όχι από το λογισμικό και θα εμφανίζεται ως έχει στην οθόνη. Η [Εντολή] τώρα μπορεί να έχει τη μία από τις εξής τιμές: DEPART, GO_STRAIGHT, CIRCLE, TURN_LEFT, TURN_RIGHT, APP_DEST, REACH_DEST, των οποίων η σημασία είναι προφανής. Σε περίπτωση που υπάρχει προσδιορισμός απόστασης ή εξόδου από μία κυκλική διαδρομή αυτός ακολουθεί μετά την παραπάνω τιμή χωρισμένος με κόμμα. Για παράδειγμα για την οδηγία «στρίψτε δεξιά σε 200 μέτρα» η εντολή θα είναι TURN_RIGHT,200;.

5.4.3 Σχεδιαστικά μορφήματα

Δεν χρησιμοποιήθηκαν σχεδιαστικά μορφήματα.

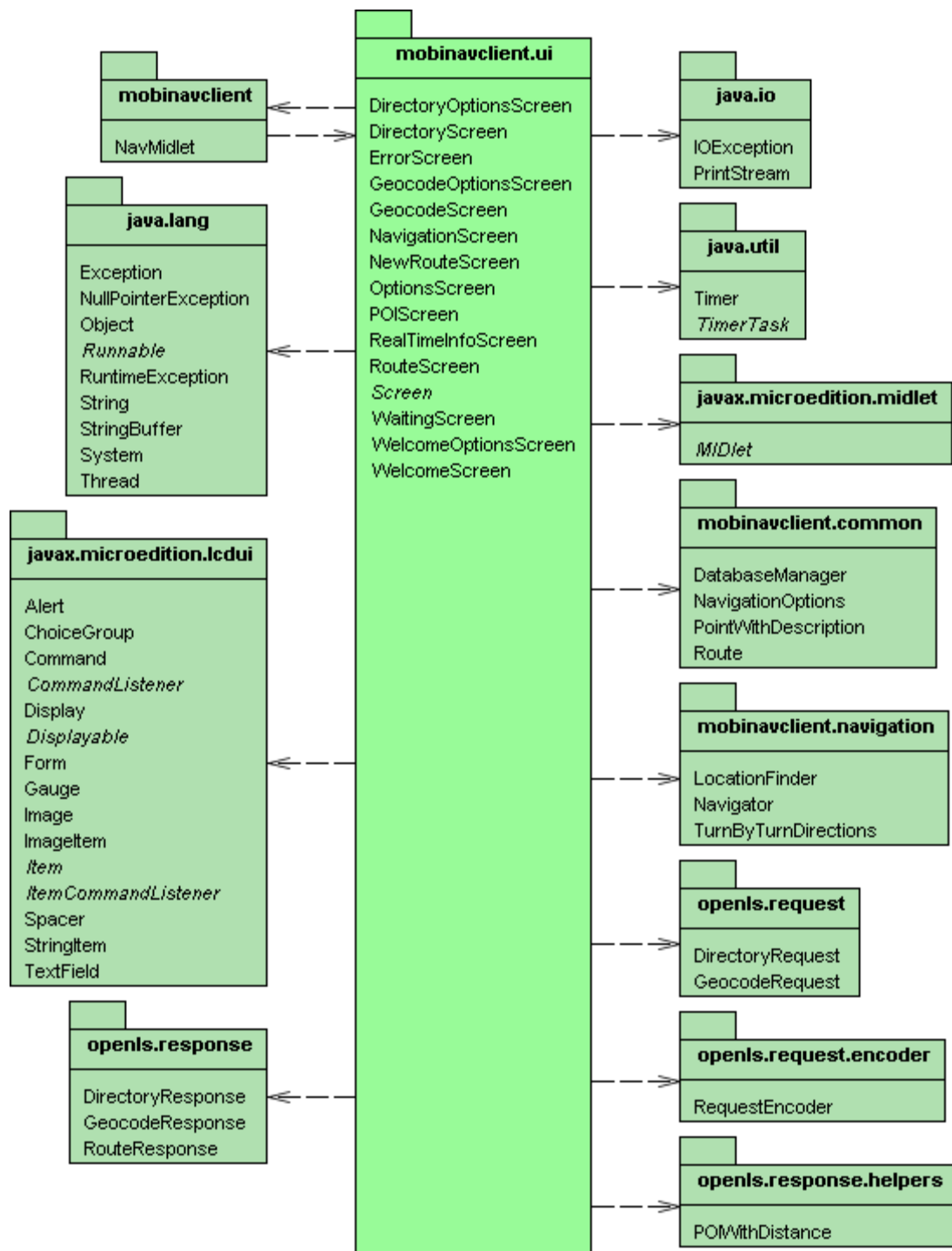
5.4.4 Σχεδιαστικές όψεις

5.4.4.1 Αποσυνθετική

Η αποσυνθετική όψη των υπολογιστών του συστήματός μας όπως αυτοί ορίστηκαν και περιγράφηκαν στο «Εγγραφο Περιγραφής της Αρχιτεκτονικής του Συστήματος» φαίνονται στη συνέχεια.

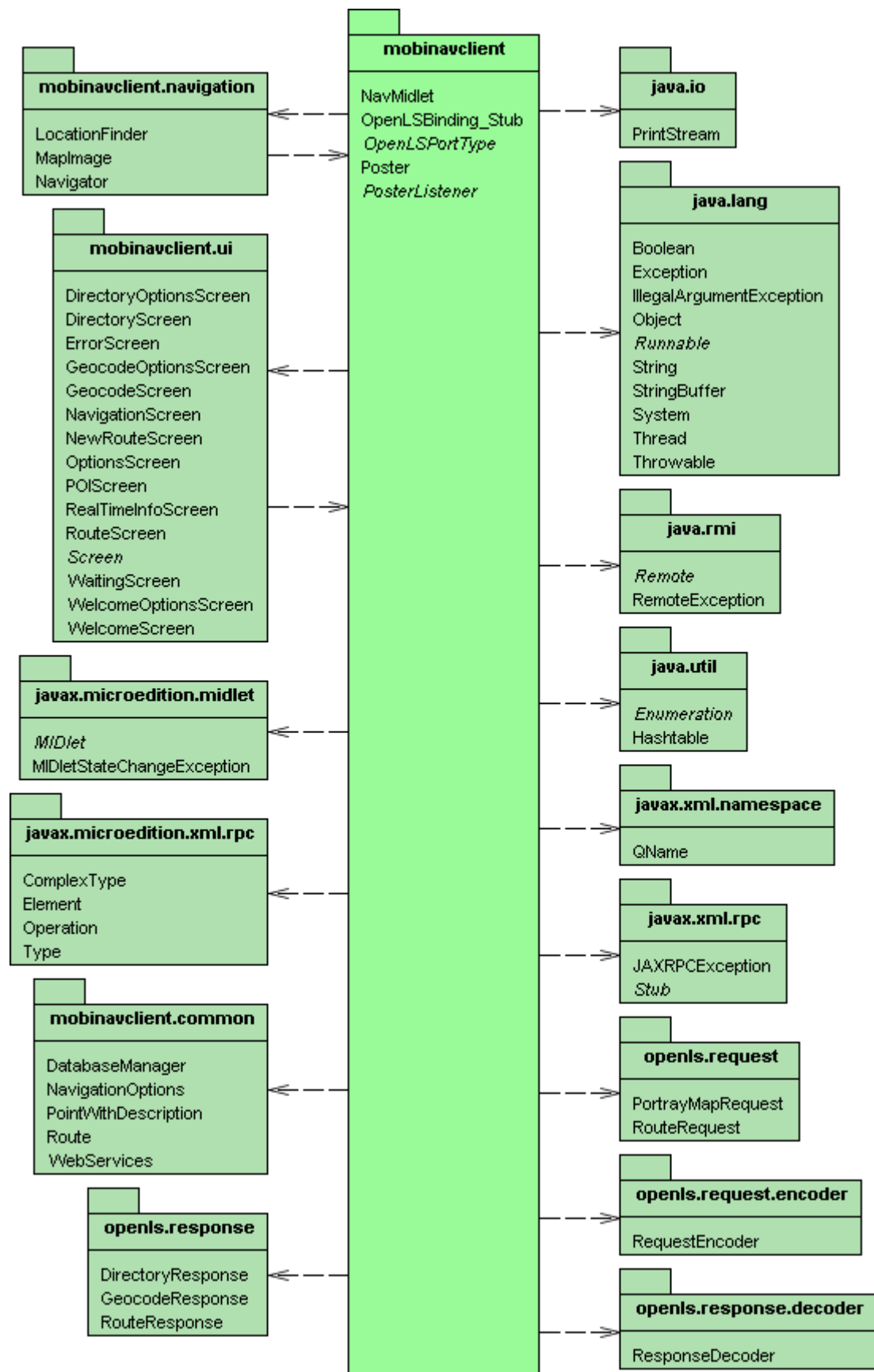
Φορητή Συσκευή (client)

Η αρχιτεκτονική οντότητα UI (User Interface) του ειδικού μέρους αποσυντίθεται στις Java classes που φαίνονται στο ακόλουθο UML διάγραμμα.



Εικόνα 55: Αποσυνθετική όψη του UI του client του συστήματος mobiNav

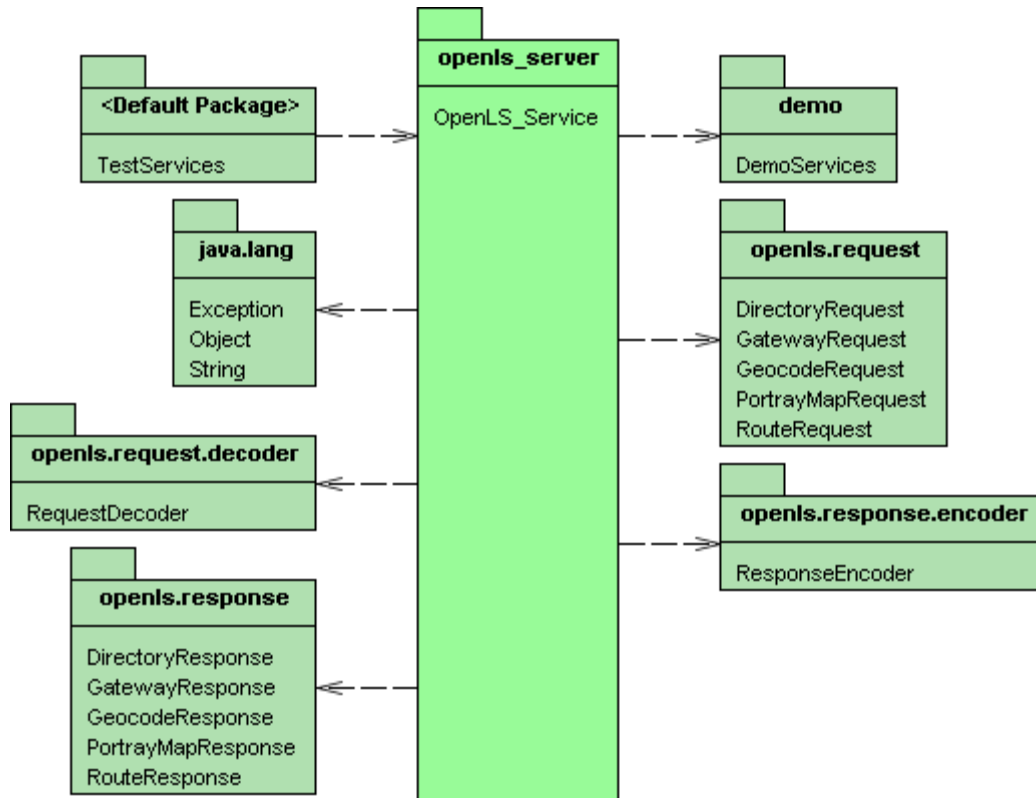
Η αρχιτεκτονική οντότητα DM (Data Management) του ειδικού μέρους του client αποσυντίθεται και αυτή σε Java classes οι οποίες φαίνονται στο ακόλουθο διάγραμμα. Στο διάγραμμα αυτό η κλάση NavMidler είναι η κύρια κλάση της εφαρμογής ενώ οι υπόλοιπες στο κεντρικό package (OpenLSBinding_Stub, OpenLSPortType, Poster, PosterListener) χρειάζονται για την κλήση των web services προς τον server.



Εικόνα 56: Αποσυνθετική όψη του DM του client του συστήματος mobiNav

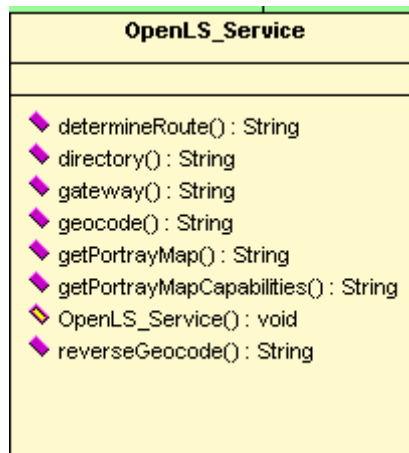
Υπολογιστής Συστήματος mobiNav (server)

Ο server όπως έχουμε ήδη περιγράψει αποτελείται μόνο από μία αρχιτεκτονική οντότητα, την DM (Data Management) στο ειδικό μέρος. Η αποσύνθεση της οντότητας αυτής φαίνεται στο διάγραμμα που ακολουθεί.



Εικόνα 57: Αποσυνθετική όψη του DM του server του συστήματος mobiNav

Για να φανούν οι υπηρεσίες τις οποίες παρέχει ο server δίνουμε την αποσυνθετική όψη της κύριας κλάσεις του server, της `OpenLS_Service.class`



Εικόνα 58: Αποσυνθετική όψη της κλάσης OpenLS_Service

Φαίνεται ότι τα ορίσματα που δέχονται όλα τα web services είναι τύπου string. Αυτό σημαίνει ότι τα xml έγγραφα ανταλλάσσονται ως strings μεταξύ client και server.

5.4.4.2 Δεδομενική

Στη δεδομενική όψη φαίνονται τα XML schemas στα οποία υπακούν όλα τα μηνύματα τα οποία ανταλλάσσονται μεταξύ client και server ως requests – responses για τα διάφορα web services. Φαίνεται επίσης το πώς πρέπει να είναι τα SOAP messages τα οποία ανταλλάσσονται.

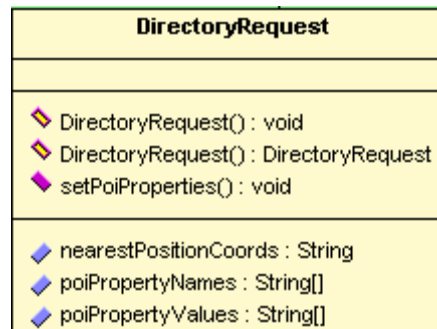
5.4.4.2.1 Δεδομένα εφαρμογής

XML schemas

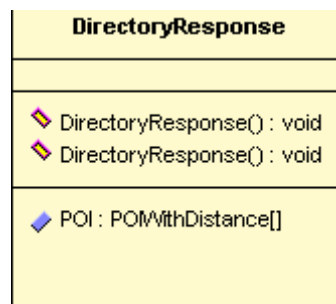
Τα XML schemas στα οποία υπακούν όλα τα μηνύματα για τις κλήσεις των web services καθώς και αυτά που αντιστοιχούν στα responses του server, είναι όλα τα schemas τα οποία δίνει ο OGC στο πρότυπο OpenGIS Location Services (OpenLS): Core Services, OGC #05-016. Φυσικά αυτά τα schemas καλύπτουν ένα πολύ μεγάλο φάσμα από πιθανές περιπτώσεις χρήσης ενός συστήματος διάθεσης γεωγραφικής πληροφορίας. Για το λόγο αυτό, στην εφαρμογή μας θα χρησιμοποιήσουμε μονάχα ένα υποσύνολο αυτών των περιπτώσεων. Αυτό σημαίνει ότι τα έγγραφα θα είναι φυσικά συμβατά με τα schemas που δίνει ο OGC αλλά θα χρησιμοποιούν μόνο ένα μέρος αυτών.

Τα schemas τα οποία χρησιμοποιεί η εφαρμογή μας δίνονται, λοιπόν, από τον OGC [OGC] και δεν κρίνεται σκόπιμο να τα παραθέσουμε εδώ. Αντίθετα, θα παραθέσουμε διαγράμματα

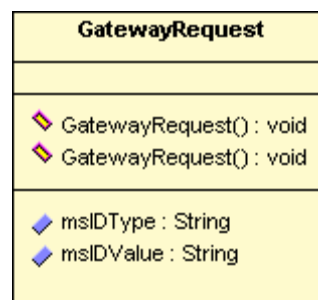
των κλάσεων οι οποίες θα αντιστοιχίζονται με τα δεδομένα τα οποία χρειάζεται ο client να δώσει ώστε να κατασκευαστούν τα XML αρχεία – μηνύματα κλήσης των διαφόρων web services και αυτά τα οποία ο server πρέπει να δώσει ώστε να κατασκευαστούν τα XML αρχεία – responses. Οι κλάσεις αυτές φαίνονται στα παρακάτω διαγράμματα UML. Η σημασία των κλάσεων αυτών καθώς και των πεδίων τους είναι προφανής από τα ονόματα που τους έχουν δοθεί.



Εικόνα 59: Directory Request



Εικόνα 60: Directory Response



Εικόνα 61: Gateway Request

GatewayResponse
 GatewayResponse() : void  GatewayResponse() : void
 msIDType : String  msIDValue : String  positionCoords : String

Εικόνα 62: Gateway Response

GeocodeRequest
 GeocodeRequest() : void  GeocodeRequest() : void
 countryCode : String  freeFormAddress : String

Εικόνα 63: Geocode Request

GeocodeResponse
 GeocodeResponse() : void  GeocodeResponse() : void
 countryCode : String  freeFormAddress : String  geocodedCoords : String

Εικόνα 64: Geocode Response

PortrayMapRequest	
 bbmaxCoords : String  bbminCoords : String  bgColor : String	
 getBBmaxCoords() : String  getBBminCoords() : String  getBGColor() : String  PortrayMapRequest() : void  PortrayMapRequest() : PortrayMapRequest  setBBmaxCoords() : void  setBBminCoords() : void  setBGColor() : void	
 basemapFilter : String  content : String  format : String  height : String  layername : String[]  transparent : String  width : String	

Εικόνα 65: Portray Map Request

PortrayMapResponse	
 PortrayMapResponse() : void  PortrayMapResponse() : PortrayMapResponse	
 map : Map[]	

Εικόνα 66: Portray Map Response

Η παραπάνω κλάση χρησιμοποιεί τη βοηθητική κλάση Map η οποία φαίνεται στο επόμενο διάγραμμα.

Map	
	bbmaxCoords : String
	bbminCoords : String
	getBBmaxCoords() : String
	getBBminCoords() : String
	Map() : void
	Map() : void
	setBBmaxCoords() : void
	setBBminCoords() : void
	data : byte[]
	format : String
	height : String
	url : String
	width : String

Εικόνα 67: Map

ReverseGeocodeRequest	
	ReverseGeocodeRequest() : void
	ReverseGeocodeRequest() : ReverseGeocodeRequest
	positionCoords : String

Εικόνα 68: Reverse Geocode Request

ReverseGeocodeResponse	
	ReverseGeocodeResponse() : void
	ReverseGeocodeResponse() : void
	countryCode : String
	freeFormAddress : String
	positionCoords : String

Εικόνα 69: Reverse Geocode Response

RouteRequest	
	map : PortrayMapRequest
	getGeometryRequest() : boolean
	getMapRequest() : PortrayMapRequest
	getRouteInstructionsRequest() : boolean
	getRouteMapRequest() : boolean
	RouteRequest() : void
	RouteRequest() : RouteRequest
	setMapRequest() : void
	endPointCoords : String
	geometryRequest : boolean
	preference : String
	routeInstructionsRequest : boolean
	routeInstructionsRequestFormat : String
	routeInstructionsRequestProvideBBBox : String
	routeInstructionsRequestProvideGeometry : String
	routeMapRequest : boolean
	startPointCoords : String
	viaPointCoords : String[]

Εικόνα 70: Route Request

RouteResponse	
	RouteResponse() : void
	RouteResponse() : void
	routeGeometryCoords : String[]
	routeInstructions : RouteInstruction[]
	routeInstructionsListLang : String
	routeMap : Map[]
	routeSummaryBBmaxCoords : String
	routeSummaryBBminCoords : String
	routeSummaryTotalDistance : String
	routeSummaryTotalTime : String

Εικόνα 71: Route Response

Η κλάση Route Response χρησιμοποιεί δύο βοηθητικές κλάσεις, τη Map την οποία παρουσιάσαμε παραπάνω, και τη RouteInstruction η οποία ακολουθεί.

RouteInstruction	
	bbmaxCoords : String
	bbminCoords : String
	getBBmaxCoords() : String
	getBBminCoords() : String
	RouteInstruction() : void
	RouteInstruction() : void
	setBBmaxCoords() : void
	setBBminCoords() : void
	description : String
	distanceValue : String
	duration : String
	instruction : String
	routeInstructionGeometryCoords : String[]

Εικόνα 72: Route Instruction

Με βάση τις παραπάνω κλάσεις λοιπόν, κατασκευάζονται τα διάφορα requests – responses τα οποία συμφωνούν με τα XML schemas του OGC. Για την κατασκευή των xml αρχείων και το αντίστροφο (κατασκευή αντικειμένων των παραπάνω κλάσεων μετά από parsing του xml) θα πρέπει να κατασκευαστούν ειδικές κλάσεις για το σκοπό αυτό (RequestEncoder, RequestDecoder, ResponseEncoder, ResponseDecoder).

Στο σημείο αυτό μένει να φανεί η μορφή των SOAP messages τα οποία ανταλλάσσονται.

SOAP messages

Τα μηνύματα SOAP request πρέπει να είναι στην εξής μορφή:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:tns="http://openls_server">
  <soap:Body>
  <tns:[Service Name]>
  <request xmlns="">String (Request XLS message)</request>
  </tns:[Service Name]>
  </soap:Body>
</soap:Envelope>
```

```
</soap:Body>
</soap:Envelope>
```

Στην περιοχή [Service Name] πρέπει να μπαίνει κάθε φορά το όνομα της υπηρεσίας (συνάρτησης) την οποία ο client επιθυμεί να καλέσει. Στην περιοχή “String (Request XLS message)” πρέπει να τοποθετείται το XML (για την ακρίβεια XLS) αρχείο που θα περιέχει όλες τις παραμέτρους που χρειάζεται το service. Το XML αρχείο θα δίνεται ως string και φυσικά θα είναι συμβατό με τα schemas του OGC.

Τα μηνύματα SOAP response πρέπει να είναι στην εξής μορφή:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:[Service Name]Response
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="http://openls_server">
      <ns1:[Service Name]Return xsi:type="xsd:string">String (Response
XLS message)</ns1:[Service Name]Return>
    </ns1:[Service Name]Response>
  </soapenv:Body>
</soapenv:Envelope>
```

Όμοια και εδώ, όπου [Service Name] τοποθετούμε το όνομα του service το οποίο επιστρέφει απάντηση στον client και όπου “String (Response XLS message)” τοποθετεί ο server την απάντηση ως XML (XLS) έγγραφο με τη μορφή string. Το XML αυτό είναι φυσικά συμβατό με τα OGC XML schemas.

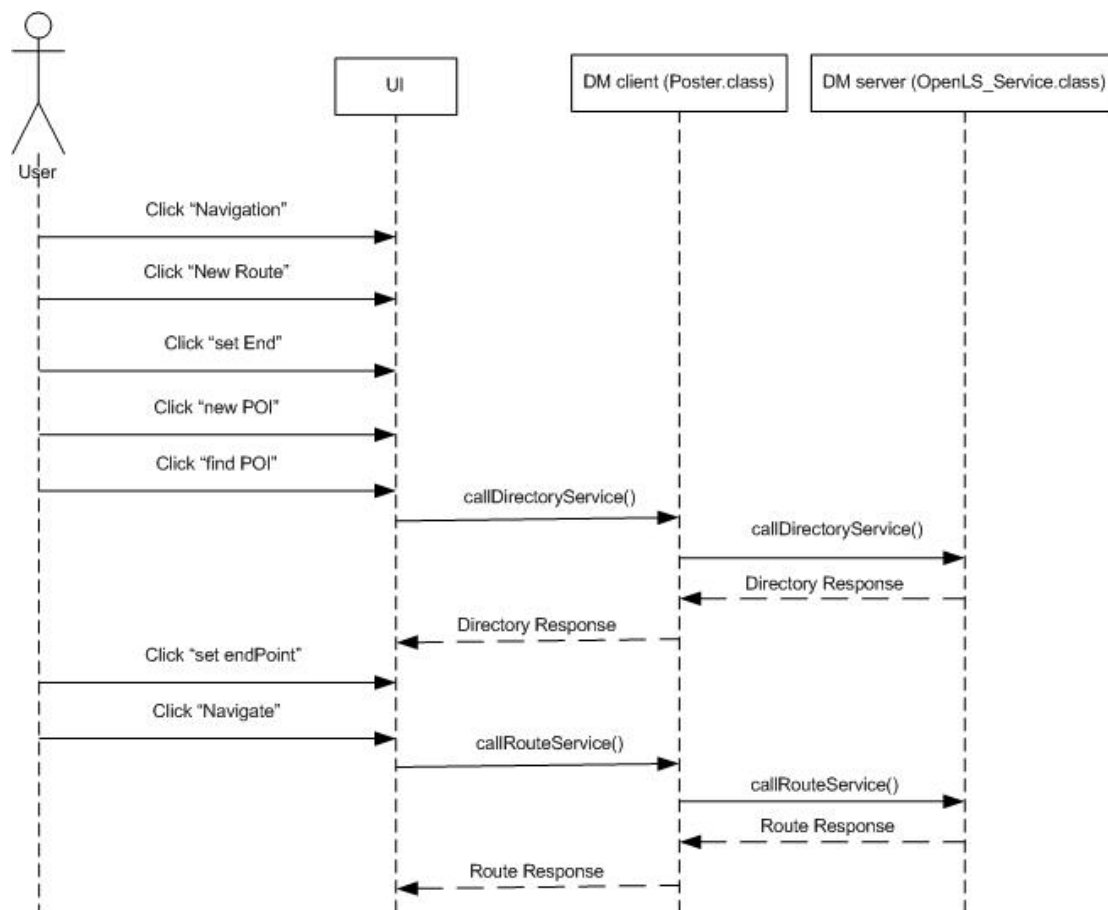
5.4.4.2.2 Βάση Δεδομένων

Η βάση δεδομένων από την οποία αντλεί το σύστημά μας γεωγραφική πληροφορία είναι εξωτερική και υπάρχει σε ένα GIS server. Για το λόγο αυτό δεν γίνεται ανάλυση βάσης δεδομένων στο παρόν έγγραφο.

5.4.4.3 Συνεργατική

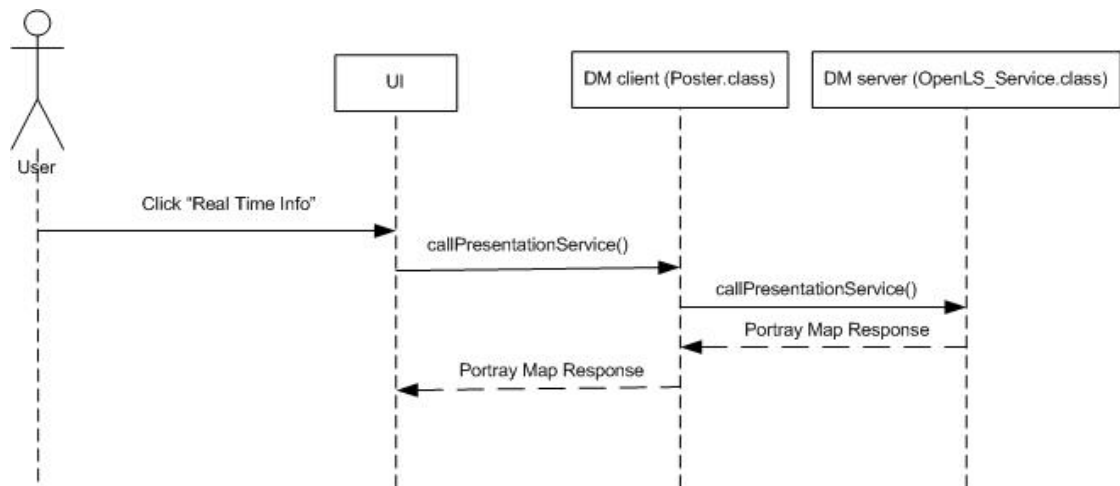
5.4.4.3.1 Ακολουθιακά Διαγράμματα

Οι εργασίες που εκτελούνται από το λογισμικό σύστημα mobiNav είναι: Πλοήγηση χρήστη (Navigate), Παρακολούθηση θέσης χρήστη σε πραγματικό χρόνο (Real Time Info) και τέλος ορισμός επιλογών πλοήγησης (Navigation Options). Οι παραπάνω εργασίες έχουν αναλυθεί σε προηγούμενο έγγραφο («Έγγραφο Προδιαγραφών Απαιτήσεων από το Λογισμικό Σύστημα»). Εδώ παρουσιάζουμε ακολουθιακά διαγράμματα για καθεμία περίπτωση χρήσης του συστήματος. Αρχικά παρουσιάζουμε το ακολουθιακό διάγραμμα της εργασίας “Navigate”.

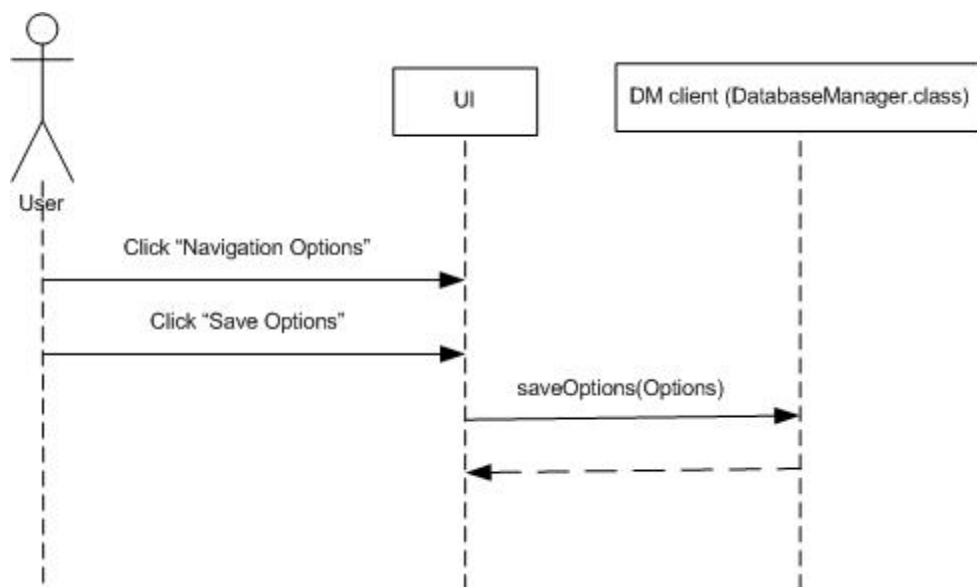


Εικόνα 73: Ακολουθιακό διάγραμμα “Navigate”

Το παραπάνω διάγραμμα δείχνει αναλυτικά τη διαδικασία για μια περίπτωση πλοήγησης και τις οντότητες του συστήματος οι οποίες χρησιμοποιούνται. Καθώς ο χρήστης κάνει κλικ στα διάφορα κουμπιά αλλάζουν οι οθόνες του User Interface με σειρά την οποία έχουμε ήδη περιγράψει σε προηγούμενο έγγραφο («Έγγραφο Περιγραφής της Αρχιτεκτονικής») και εμπλέκεται η αρχιτεκτονική οντότητα UI (User Interface). Μόλις καλείται κάποιο web service, στη διαδικασία εισάγονται κλάσεις από το DM του client καθώς και το DM του server και τελικά τα αποτελέσματα εμφανίζονται πάλι στην οθόνη μέσω της οντότητας UI. Με βάση την ίδια λογική, στη συνέχεια βλέπουμε τα ακολουθιακά διαγράμματα για τις περιπτώσεις “Real Time Info” και “Navigation Options”.



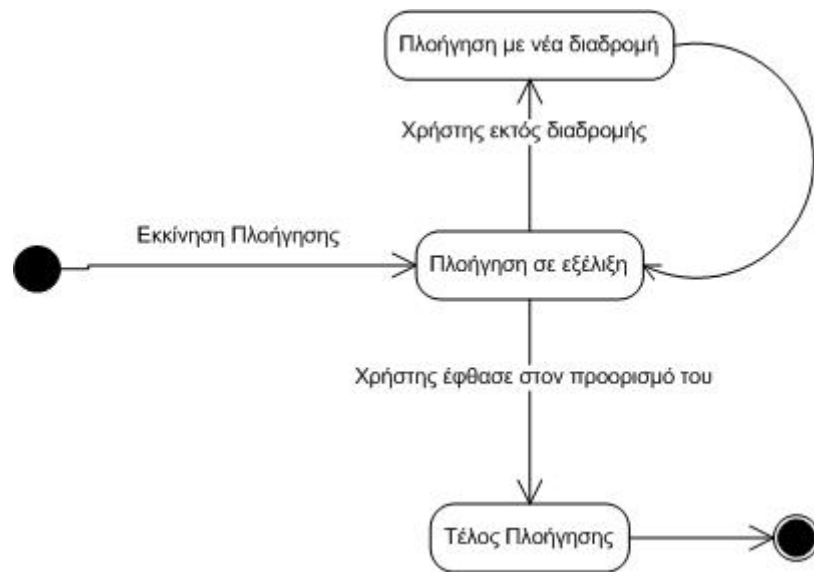
Εικόνα 74: Ακολουθιακό διάγραμμα “Real Time Info”



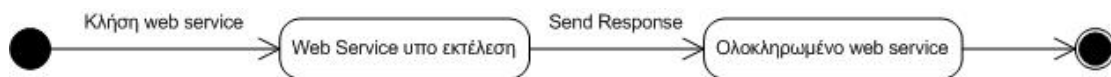
Εικόνα 75: Ακολουθιακό διάγραμμα “Navigation Options”

5.4.4.3.2 Διαγράμματα Καταστάσεων

Με τα διαγράμματα καταστάσεων περιγράφεται ο κύκλος ζωής μιας οντότητας και η σειρά με την οποία η οντότητα εκτελεί τις λειτουργίες της. Στην περίπτωση του λογισμικού συστήματος mobiNav δύο βασικές οντότητες είναι αφενός η «Πλοήγηση» και αφετέρου η «Διαδικτυακή υπηρεσία» ή “web service”. Παρουσιάζουμε λοιπόν διαγράμματα καταστάσεων για τις δύο αυτές οντότητες.



Εικόνα 76: Διάγραμμα καταστάσεων για την οντότητα «Πλοήγηση»



Εικόνα 77: Διάγραμμα καταστάσεων για την οντότητα «web service»

5.4.4.4 Διαπροσωπειακή

Σε αυτή την όψη παρουσιάζονται όλες οι σχεδιαστικές οντότητες για τις οποίες θα γραφεί κώδικας μαζί με τις διαπροσωπείες τους. Οι κλάσεις αυτές έχουν ήδη περιγραφεί σε προηγούμενες ενότητες του παρόντος εγγράφου (Αποσυνθετική και Δεδομενική όψη). Αφενός οι κλάσεις για τα αρχιτεκτονικές οντότητες UI, DM του client και DM του server περιγράφηκαν στην ενότητα «Αποσυνθετική Όψη» και αφετέρου οι κλάσεις που αφορούν στα δεδομένα που ανταλλάσσονται μέσω XML strings στην ενότητα «Δεδομενική Όψη».

5.4.4.5 Αλγοριθμική

Δεν εφαρμόζεται κάποιος αλγόριθμος που κρίνεται σκόπιμο να αναφερθεί.

6

Υλοποίηση

Στο κεφάλαιο αυτό παρουσιάζουμε τον τρόπο με τον οποίο υλοποιήσαμε το λογισμικό σύστημα που αναλύσαμε παραπάνω. Επιπλέον, παρουσιάζουμε τις πλατφόρμες, τα εργαλεία, τις τεχνολογίες και τα πρωτόκολλα τα οποία χρησιμοποιήσαμε κατά την υλοποίηση του συστήματος.

6.1 Πλατφόρμες και προγραμματιστικά εργαλεία

6.1.1 Borland JBuilder X

Ως βασική πλατφόρμα ανάπτυξης της εφαρμογής χρησιμοποιήθηκε το Borland JBuilder X Enterprise, έκδοση 10.0.176.0, που αναπτύχθηκε από την Borland (www.borland.com) Μία δοκιμαστική έκδοση του Borland JBuilder X Enterprise βρίσκεται διαθέσιμη στη σελίδα:

http://www.borland.com/products/downloads/download_jbuilder.html

6.1.2 Altova XML Spy 5 Enterprise Edition

Ως περιβάλλον για την ανάπτυξη, τροποποίηση των διαφόρων αρχείων xml, xml schemas που χρειάστηκε η εφαρμογή χρησιμοποιήσαμε το XML Spy 5 Enterprise Edition, version 5, rel. 3.

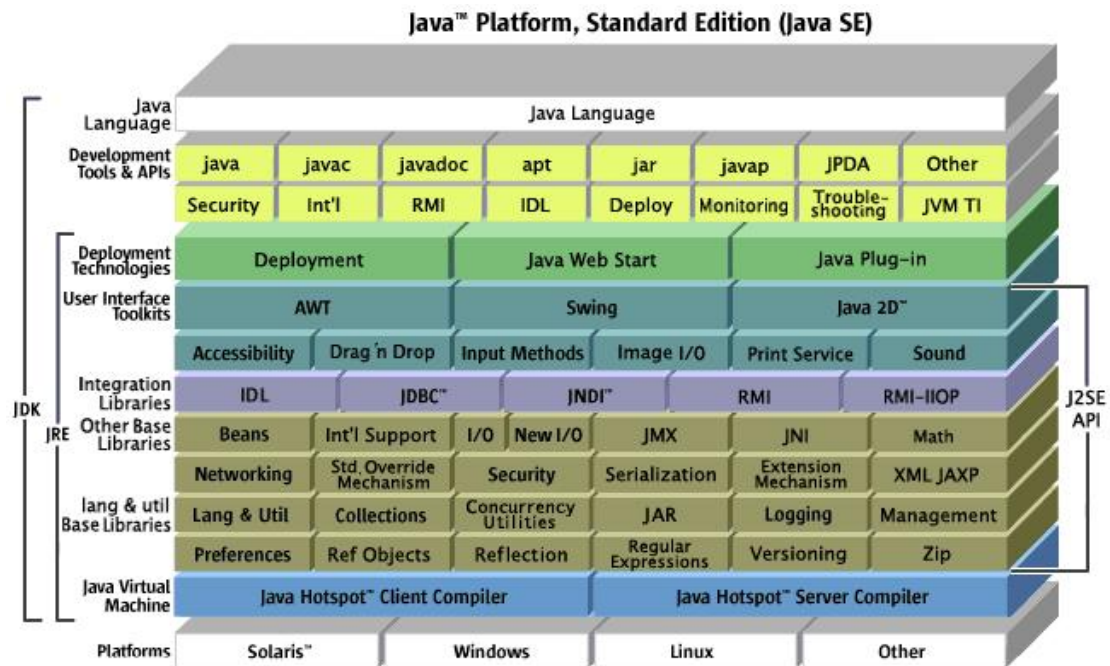
Το XML Spy, το οποίο αναπτύχθηκε από την Altova (www.altova.com) παρέχει ένα γραφικό περιβάλλον για την μοντελοποίηση, επεξεργασία, τροποποίηση και εκσφαλμάτωση (debugging) XML εγγράφων. Μια δοκιμαστική έκδοση του XML Spy βρίσκεται διαθέσιμη στη σελίδα: http://www.altova.com/download/xmlspy/xml_editor_enterprise.html

6.1.3 Apache Tomcat

Ως εξυπηρετητής του διαδικτύου στον οποίο εγκαταστάθηκαν τα web services, χρησιμοποιήθηκε ο Apache Tomcat, στην έκδοση 4.1 η οποία είναι ενσωματωμένη στον JBuilder. Φυσικά, η εφαρμογή μπορεί να δουλέψει και με μεγαλύτερες εκδόσεις του Apache Tomcat. Ο Tomcat είναι ένα servlet container ή servlet engine. Υποστηρίζει servlets και JSPs καθώς και στατικές σελίδες. Αναπτύχθηκε από την Apache Software Foundation (www.apache.org) και διατίθεται δωρεάν στο διαδίκτυο στη σελίδα tomcat.apache.org/index.html.

6.1.4 Java (J2SE)

Η γλώσσα προγραμματισμού που χρησιμοποιήσαμε είναι η Java και πιο συγκεκριμένα η έκδοση Java 1.4.2_01-b06, η οποία μπορεί να βρεθεί στο site της sun (www.java.sun.com) και είναι επιπλέον ενσωματωμένη στο JBuilder. Η Java 2 Standard Edition (J2SE) είναι η πλατφόρμα ανάπτυξης που χρησιμοποιήσαμε για την ανάπτυξη των web services στον server. Η J2SE μας επιτρέπει να αναπτύξουμε εφαρμογές Java για desktops και servers, καθώς επίσης και εφαρμογές για ενσωματωμένα (embedded) περιβάλλοντα και περιβάλλοντα πραγματικού χρόνου (real time). Η J2SE περιλαμβάνει επίσης κλάσεις οι οποίες υποστηρίζουν την ανάπτυξη web services και αποτελεί θεμέλιο για τη δημιουργία της Java Platform Enterprise Edition (Java EE). Οι τεχνολογίες που περιλαμβάνει η J2SE φαίνονται στο ακόλουθο σχήμα.



Εικόνα 78: Java Platform, Standard Edition

6.1.5 Java (J2ME)

Η Java 2 Micro Edition (J2ME) είναι η πλατφόρμα ανάπτυξης εφαρμογών για μικρές συσκευές. Παρέχει ένα εύρωστο, ευέλικτο περιβάλλον για την ανάπτυξη εφαρμογών που τρέχουν σε μια ευρεία γκάμα από μικρές συσκευές, όπως κινητά τηλέφωνα ή PDAs. Οι εφαρμογές που γράφονται σε J2ME είναι ανεξάρτητες από τη συσκευή στην οποία τρέχουν. Το μόνο που χρειάζεται είναι η συσκευή να υποστηρίζει Java. Η J2ME έχει αναπτυχθεί σε εκατομμύρια συσκευές, υποστηρίζεται από τους περισσότερους κατασκευαστές φορητών συσκευών και χρησιμοποιείται από πολλές εταιρείες ανά τον κόσμο.

Η Java ME μπορεί να χωριστεί στις διαμορφώσεις – configurations, τα προφίλ – profiles, και τα προαιρετικά πακέτα – optional packages.

6.1.5.1 Διαμορφώσεις – Configurations

Οι διαμορφώσεις είναι προδιαγραφές που αφορούν στον καθορισμό μιας virtual machine και ενός βασικού συνόλου βιβλιοθηκών που παρέχουν τα απαραίτητα APIs τα οποία μπορούν να χρησιμοποιηθούν από μια ομάδα συσκευών. Παρέχουν τη βασική λειτουργικότητα για μια γκάμα συσκευών οι οποίες έχουν παρόμοια χαρακτηριστικά. Για παράδειγμα, μια

διαμόρφωση μπορεί να έχει σχεδιαστεί για συσκευές οι οποίες έχουν λιγότερο από 512 KB μνήμης και περιορισμένη συνδεσιμότητα. Η virtual machine την οποία αναφέραμε παραπάνω, είναι είτε μια πλήρης Java Virtual Machine (JVM) είτε ένα υποσύνολο αυτής. Το σύνολο των APIs είναι συνήθως ένα υποσύνολο των APIs της Java SE. Προς το παρόν, υπάρχουν δύο διαμορφώσεις για την Java ME: η Connected Limited Device Configuration (CLDC) και η Connected Device Configuration (CDC).

6.1.5.2 Προφίλ – Profiles

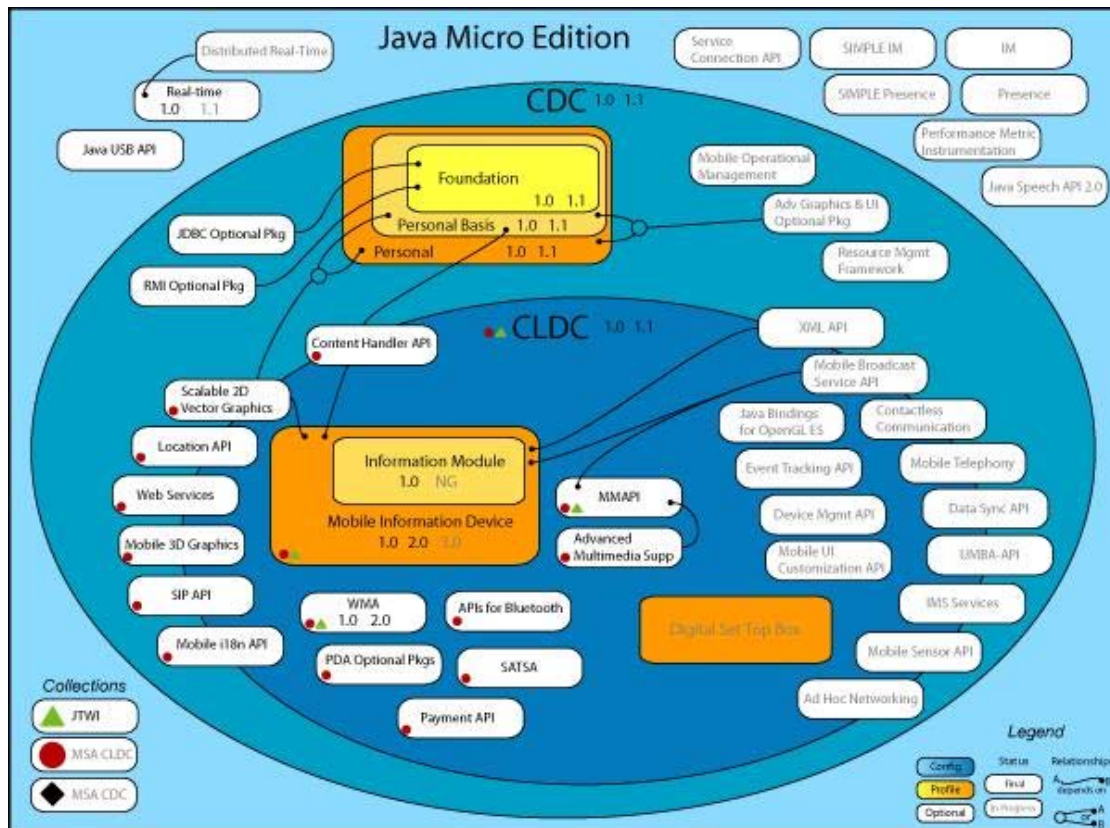
Τα προφίλ συμπληρώνουν μια διαμόρφωση, προσθέτοντας συγκεκριμένα APIs για να δημιουργήσουν ένα runtime environment για να μπορούν να τρέχουν εφαρμογές σε μια συγκεκριμένη κατηγορία συσκευών. Ένα προφίλ είναι ένα σύνολο από APIs υψηλότερου επιπέδου τα οποία ορίζουν το μοντέλο κύκλου ζωής της εφαρμογής, τη διεπαφή χρήστη, τη μόνιμη αποθήκευση και την πρόσβαση στις ιδιαίτερες ιδιότητες μιας συσκευής. Ένα ευρέως αποδεκτό παράδειγμα είναι ο συνδυασμός της διαμόρφωσης CLDC με το προφίλ Mobile Information Device Profile (MIDP) για την παροχή μιας ολοκληρωμένου περιβάλλοντος εφαρμογών Java για κινητά τηλέφωνα και άλλες συσκευές με παρόμοια χαρακτηριστικά.

6.1.5.3 Προαιρετικά πακέτα – optional packages

Τα προαιρετικά πακέτα επεκτείνουν την πλατφόρμα Java ME προσθέτοντας λειτουργικότητα στην στοίβα τεχνολογιών η οποία περιλαμβάνει την διαμόρφωση CLDC ή CDC και ένα σχετικό προφίλ. Έχοντας δημιουργηθεί για να καλύψουν πολύ ειδικές απαιτήσεις εφαρμογής, τα προαιρετικά πακέτα προσφέρουν APIs για τη χρησιμοποίηση υπαρχόντων και νέων τεχνολογιών όπως η συνδεσιμότητα με βάσεις δεδομένων, αποστολή μηνυμάτων ασύρματα, πολυμέσα, 3D γραφικά, και web services.

6.1.5.4 Η πλατφόρμα Java ME σήμερα

Η τρέχουσα σχέση μεταξύ των διαμορφώσεων, προφίλ και προαιρετικών πακέτων που υπάρχουν φαίνεται στο παρακάτω διάγραμμα.



Εικόνα 79: Java Platform, Micro Edition

Όπως φαίνεται στο διάγραμμα αυτό, η Java ME έχει δύο βασικές διαμορφώσεις. Η πρώτη είναι η Connected Limited Device Configuration (CLDC). Η διαμόρφωση αυτή είναι για μικρές ασύρματες συσκευές με περιορισμένη επεξεργαστική ισχύ, μνήμη και γραφική έξοδο, όπως τα κινητά τηλέφωνα, ή τα PDAs. Οι απαιτήσεις του CLDC σε υλικό είναι εξαιρετικά περιορισμένες:

1. 16bit ή 32bit επεξεργαστής 16MHz ή ανώτερος
2. 160 KB σταθερής μνήμης για την αποθήκευση των βιβλιοθηκών και της εικονικής μηχανής Java
3. Τουλάχιστον 192KB συνολικής μνήμη για την πλατφόρμα Java
4. Διασύνδεση σε κάποιο δίκτυο, συχνά περιορισμένου εύρους ζώνης.

Η έκδοση 1.0 του CLDC (Java Specification Request – JSR 30) ήταν η πρώτη έκδοση, που παρείχε εξαιρετικά περιορισμένες δυνατότητες σε τέτοιου είδους μικρές συσκευές. Η τρέχουσα έκδοση 1.1 (JSR 139) επεκτείνει την έκδοση 1.0, και προσθέτει αρκετές επεκτάσεις, όπως οι πράξεις κινητής υποδιαστολής

Το προφίλ MIDP το οποίο φαίνεται στο παραπάνω διάγραμμα, βασίζεται πάνω στο CLDC και ήταν το πρώτο προφίλ που κατασκευάστηκε και συνεπώς το πρώτο Java ME περιβάλλον

εφαρμογών. Το MIDP προσδιορίζει κινητά τηλέφωνα, ενώ οι εφαρμογές που γράφονται πάνω σε αυτό αποκαλούνται MIDlets. Η έκδοση 1.0 (JSR 37) ήταν η πρώτη έκδοση του MIDP, και προσέφερε βασικές λειτουργίες, όπως η διασύνδεση του χρήστη και η ασφάλεια του δικτύου. Η τρέχουσα έκδοση 2.0 προσφέρει βελτιωμένη διεπαφή χρήστη, δυνατότητες πολυμέσων και δημιουργία παιχνιδιών, και ακόμα βελτιωμένες δυνατότητες δικτυακής διασύνδεσης.

Η άλλη βασική διαμόρφωση που παρατηρούμε στο διάγραμμα εκτός της CLDC είναι η Connected Device Configuration (CDC). Η διαμόρφωση αυτή είναι για μεγαλύτερες συσκευές από πλευράς μνήμης και ταχύτητας επεξεργασίας καθώς και από πλευράς συνδεσιμότητας. Παραδείγματα τέτοιων συσκευών είναι τα set-top boxes, Internet appliances, embedded servers καθώς και “high-end mobile devices”.

Στην εφαρμογή μας χρησιμοποιήσαμε το συνδυασμό MIDP – CLDC ο οποίος είναι πολύ διαδεδομένος, έτσι ώστε η εφαρμογή μας να τρέχει σε μικρές φορητές συσκευές με περιορισμένους πόρους. Επιπλέον, από τα προαιρετικά πακέτα χρησιμοποιήσαμε τα εξής:

- **Web Services (JSR 172):** Το JSR 172 είναι το standard API για την πρόσβαση σε Web Services από μια J2ME-enabled φορητή συσκευή. Παρέχει βιβλιοθήκες κατάλληλες για τη συγγραφή εφαρμογών μέσω των οποίων μπορεί μια φορητή συσκευή να καλέσει web services καθώς και να κάνει parsing σε XML έγγραφα, μέσω ενός ελαφριού (lightweight) XML parser τον οποίο παρέχει. Συγκεκριμένα, παρέχει τον SAX (Simple API for XML) parser, γνωστό και από τη Java SE, προσαρμοσμένο όμως κατάλληλα στα χαρακτηριστικά φορητών συσκευών περιορισμένων πόρων.
- **Bluetooth (JSR 082):** Το JSR 082 είναι το API μέσω του οποίου καθίσταται εφικτή η πρόσβαση στις ιδιότητες της Bluetooth συνδεσιμότητας μιας φορητής συσκευής μέσω Java. Συγκεκριμένα το Bluetooth API παρέχει μια διεπαφή προς την ασύρματη σύνδεση μέσω Bluetooth μιας φορητής συσκευής, περιλαμβανομένων των λειτουργιών ανακάλυψης συσκευών (device discovery) προς σύνδεση και ανταλλαγής δεδομένων (data exchange).
- **Mobile Media API (JSR 135):** Το Mobile Media API (MMAPI) είναι ένα προαιρετικό πακέτο το οποίο υποστηρίζει εφαρμογές πολυμέσων σε συσκευές που υποστηρίζουν J2ME. Αυτό το πακέτο, το οποίο ορίστηκε από τη Java Community Process (JCP) στο JSR 135, είναι πολύ ευέλικτο. Σχεδιάστηκε για να τρέχει με οποιοδήποτε πρωτόκολλο και format. Για παράδειγμα, δεν καθορίζει ότι η εφαρμογή πρέπει να υποστηρίζει συγκεκριμένα πρωτόκολλα μεταφοράς δεδομένων όπως το HTTP ή το Real Time Transport Protocol (RTP), ή συγκεκριμένα media format όπως MP3, MIDI, ή MPEG-4.

6.1.6 kXML, kSOAP

Πολλές από τις συσκευές κινητών τηλεφώνων σήμερα δεν υποστηρίζουν ακόμα το JSR 172 για τα web services το οποίο χρησιμοποιήθηκε. Έτσι έπρεπε να βρούμε ένα εναλλακτικό τρόπο να καλούμε SOAP/XML web services μέσω των κινητών τηλεφώνων χωρίς τη βοήθεια του JSR 172. Τη λύση σε αυτό το πρόβλημα την έδωσαν τα πακέτα kSOAP, kXML τα οποία αναπτύχθηκαν από την Enhyadra ακριβώς για το σκοπό αυτό. Έτσι η εφαρμογή μας δίνεται σε δύο εκδόσεις, σε έκδοση που κάνει χρήση JSR 172, και σε έκδοση η οποία χρησιμοποιεί kSOAP, kXML για συσκευές που δεν υποστηρίζουν JSR 172.

Το kXML project παρέχει έναν XML pull parser και writer κατάλληλους για κάθε πλατφόρμα Java συμπεριλαμβανομένης και της J2ME (CLDC / MIDP / CDC). Εξαιτίας του μικρού μεγέθους του, είναι ιδιαιτέρως κατάλληλο για Applets ή εφαρμογές Java που τρέχουν σε φορητές συσκευές όπως είναι τα Palm Pilots ή κινητά τηλέφωνα που υποστηρίζουν MIDP. Ορισμένα από τα χαρακτηριστικά του kXML είναι:

- Υποστήριξη XML Namespace
- “Relaxed” mode για parsing HTML ή άλλα SGML formats
- Μικρό «ίχνος» μνήμης
- Ένας pull-based parser για απολοποιημένο parsing ακόμη και πολύπλοκων αρχείων XML (nested / modularized XML structures).
- Υποστήριξη εγγραφής XML συμπεριλαμβανομένου του χειρισμού namespace
- Προαιρετικό kDOM
- Προαιρετική υποστήριξη WAP (WBXML/WML)

Το kSOAP με τη σειρά του, είναι ένα SOAP API κατάλληλο για τη Java 2 Micro Edition, βασισμένο στο kXML το οποίο περιγράψαμε παραπάνω. Εξαιτίας του μικρού «ίχνους» μνήμης που έχει και το kSOAP, μπορεί να χρησιμοποιηθεί για την ανάπτυξη ακόμα και SOAP – enabled Java Applets.

6.1.7 Πρωτόκολλο NMEA 0183

Η National Marine Electronics Association (NMEA) των Ηνωμένων Πολιτειών της Αμερικής δημοσιεύει τα πρωτόκολλα για τη διεπαφή ναυτικών ηλεκτρονικών συσκευών. Η χρήση αυτών των πρωτοκόλλων επεκτάθηκε και πλέον χρησιμοποιείται ως έξοδος από οποιαδήποτε συσκευή μπορεί να προσφέρει δεδομένα καθοδήγησης ή εντοπισμού θέσης. Το πιο συχνά

χρησιμοποιούμενο πρωτόκολλό της είναι το NMEA 0183 έκδοση 3.01 που δόθηκε στη δημοσιότητα τον Ιανουάριο του 2002.

Η έξοδος του NMEA 0183 είναι συμβατή με το RS232 της σειριακής θύρας του υπολογιστή. Με την εξέλιξη της τεχνολογίας δημιουργήθηκαν ασύρματες συσκευές Bluetooth, οι οποίες δημιουργούν για την έξοδο NMEA σειριακή θύρα πάνω από την σύνδεση Bluetooth (Bluetooth Serial Port Protocol – BTSPP).

Η ταχύτητα μεταφοράς των δεδομένων είναι 4800 bps, με 8 bits δεδομένων και κανένα bit ισοτιμίας. Το πρωτόκολλο NMEA 0183 χρησιμοποιεί ένα απλό ASCII, σειριακής επικοινωνίας πρωτόκολλο το οποίο καθορίζει το πώς τα δεδομένα μεταφέρονται μέσα σε μια πρόταση από τον πομπό σε έναν ή περισσότερους δέκτες. Το πρότυπο επίσης καθορίζει τον τύπο των περιεχομένων κάθε πρότασης (μήνυμα) έτσι ώστε όλοι οι δέκτες να έχουν τη δυνατότητα να κάνουν parsing στο μήνυμα αποτελεσματικά:

- Κάθε μήνυμα ξεκινάει με το χαρακτήρα δολάριο («\$»)
- Οι επόμενοι πέντε χαρακτήρες καθορίζουν τον τύπο του μηνύματος
- Όλα τα επόμενα πεδία δεδομένων τα οποία ακολουθούν χωρίζονται με κόμμα.
- Ο πρώτος χαρακτήρας ο οποίος ακολουθεί αμέσως μετά το τελευταίο πεδίο δεδομένων είναι ένας αστερίσκος
- Ο αστερίσκος ακολουθείται από ένα διψήφιο checksum.

Ακολουθεί ένα παράδειγμα μιας πρότασης του πρωτοκόλλου NMEA:

```
Waypoint Arrival Alarm  
$GPAAM,A,A,0.10,N,WPTNME*43
```

Όπου:

AAM	Arrival Alarm
A	Arrival circle entered
A	Perpendicular passed
0.10	Circle radius
N	Nautical miles
WPTNME	Waypoint name
*43	Checksum data

Το παραπάνω πρωτόκολλο χρειάστηκε για το parsing και την εξαγωγή των δεδομένων που λαμβάνουμε από τη συσκευή Bluetooth GPS Receiver στο κινητό τηλέφωνο.

6.2 Λεπτομέρειες υλοποίησης

Στο σημείο αυτό περιγράφουμε εν συντομία την διαδικασία υλοποίησης της εφαρμογής μας, ο κώδικας της οποίας δίνεται στο CD το οποίο συνοδεύει τη διπλωματική εργασία. Τα projects στον JBuilder που κατασκευάσαμε για την εφαρμογή είναι τα εξής δύο: OpenLS_Services.jpx και mobinavClient.jpx τα οποία υλοποιούν την server side και client side του συστήματός μας αντίστοιχα.

6.2.1 *mobiNav Web Services Server*

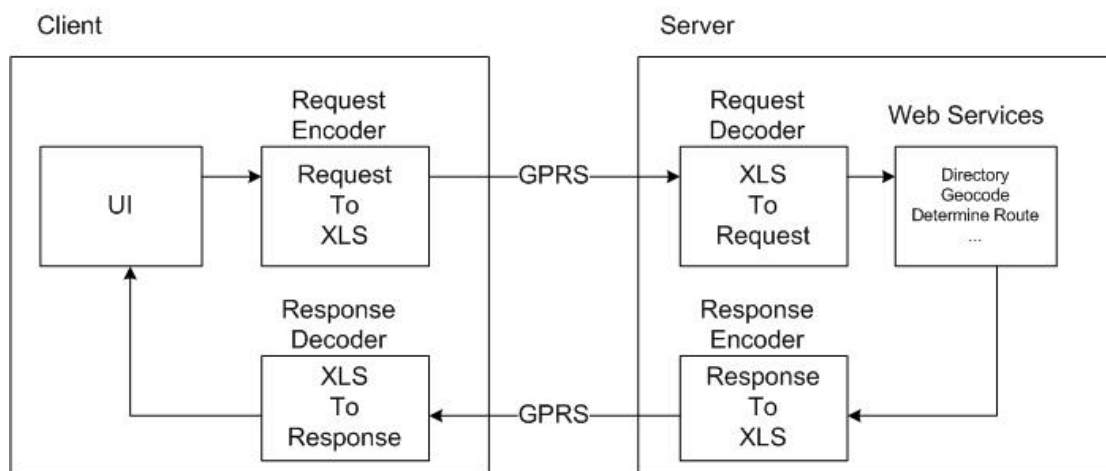
Κατά την υλοποίηση του server της εφαρμογής (OpenLS_Services.jpx) κατασκευάσαμε τα ακόλουθα βασικά packages:

- **package opens:** Το package αυτό περιέχει κλάσεις οι οποίες υλοποιούν τους request και response encoders και decoders. Όλες οι κλάσεις που υλοποιήσαμε εδώ βασίζονται αποκλειστικά στο πρότυπο που δίνει ο OGC “OpenGIS Location Services (OpenLS): Core Services” OGC 05-016. Τα schemas που δίνει αυτό το πρότυπο είναι πολύ ωραία δομημένα. Περιέχουν πολλές εντολές import για να επαναχρησιμοποιούν elements που έχουν ορίσει σε άλλα .xsd αρχεία. Περιέχουν επίσης πολλούς αφηρημένους τύπους δεδομένων (abstract types). Αυτού του είδους η δόμηση είναι βολική για τον άνθρωπο όμως δεν είναι εφικτό στη μορφή αυτή να δημιουργήσουμε κλάσεις που αντιστοιχούν σε αυτά τα schemas (διαδικασία data binding) αυτόματα μέσω προγραμμάτων που για αυτή τη δουλειά.

Ένα τέτοιο data binding framework το οποίο είναι ενσωματωμένο στο JBuilder για την παραγωγή κλάσεων Java από XML schemas είναι το Castor. Το Castor όμως, προς το παρόν τουλάχιστον, δεν μπορεί να κάνει data binding σε αρχεία που περιέχουν εντολές import ούτε υποστηρίζει abstract types. Συνεπώς, για να χρησιμοποιήσουμε το Castor ως data binding framework θα έπρεπε να τροποποιήσουμε τα schemas του OGC σε άλλα, ισοδύναμα με τα πρώτα, κατάλληλα όμως για επεξεργασία (χωρίς εντολές import και abstract types). Παρόλα αυτά, προτιμήσαμε να μη χρησιμοποιήσουμε κάποιο data binding framework τύπου Castor αλλά να κατασκευάσουμε εμείς από το μηδέν τις κλάσεις που θα αντιστοιχούν στα διάφορα schemas. Η επιλογή που κάναμε οφείλεται στο ότι αφενός η τροποποίηση

των schemas θα ήταν εξαιρετικά επίπονη και με κίνδυνο λάθους και αφετέρου για ομοιομορφία με το κομμάτι του client. Οι κλάσεις που δημιουργεί το Castor είναι πάρα πολλές και δεν είναι η βέλτιστη λύση επειδή τις δημιουργεί για εκτέλεση από PC που δεν έχει περιορισμένους πόρους όπως μια φορητή συσκευή (client). Έτσι, κατασκευάσαμε τις κλάσεις για τα request elements των schemas στο package opensls.request και τις κλάσεις για τα response elements των schemas στο package opensls.response. Φυσικά οι κλάσεις αυτές δεν αντιστοιχίζονται πλήρως με τα schemas (όπως θα τα έκανε π.χ. ο Castor) αλλά μόνο με ένα υποσύνολο αυτών το οποίο μας αρκεί για το σύστημά μας. Μέσα στα υπό-packages opensls.request / response.encoder / decoder περιέχονται κλάσεις που υλοποιούν τους request/response encoders/decoders. Οι encoders παίρνουν σαν είσοδο μια κλάση response ή request ανάλογα και παράγουν το αντίστοιχο XML (XSL σωστότερα) βασισμένο στο πρότυπο OpenLS του OGC. Την αντίστροφη διαδικασία εκτελεί ο decoder.

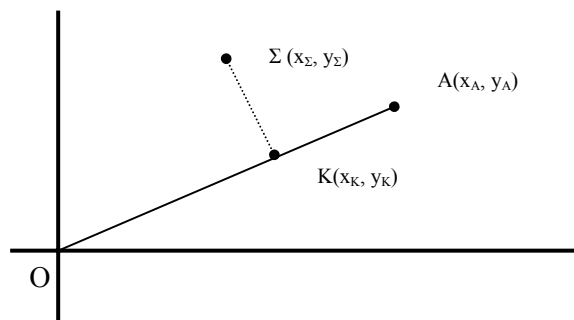
Ο decoder για να λειτουργήσει παίρνει ένα έγγραφο xml και αφού κάνει parsing εξάγει τις παραμέτρους στην αντίστοιχη κλάση. Ως XML parser χρησιμοποιήθηκε ο πολύ γνωστός SAX parser. Η επιλογή αυτή οφείλεται για λόγους ταχύτητας απόκρισης (είναι γνωστό ότι ο SAX parser είναι κατά πολύ ταχύτερος από άλλους parsers, π.χ. DOM). Επιπλέον επιλέξαμε SAX parser γιατί δεν απαιτεί πολλούς πόρους από το σύστημα και υποστηρίζεται από τη J2ME μέσω του JSR-172 επομένως μπορούσαμε να χρησιμοποιήσουμε ίδιας μορφής κώδικα και στον client (στην έκδοση που χρησιμοποιεί JSR-172). Ο ρόλος των encoders / decoders στην εφαρμογή φαίνεται καλύτερα σε σχηματική αναπαράσταση του συστήματός μας η οποία φυσικά είναι βασισμένη στα έγγραφα της τεχνολογίας λογισμικού που έχουμε παραθέσει παραπάνω.



Εικόνα 80: Το σύστημα mobiNav (από τη σκοπιά των request-response encoders / decoders)

- **package coord:** Περιέχει κλάσεις για χειρισμό γεωγραφικών δεδομένων (σημείων, ευθειών, bounding boxes κτλ). Οι διάφοροι υπολογισμοί πάνω στα γεωγραφικά δεδομένα έγιναν με βάση αναλυτική γεωμετρία. Για όλους τους υπολογισμούς, έπρεπε πρώτα να μετατρέπουμε τα δεδομένα σε καρτεσιανό επίπεδο (x, y) όταν αυτά δίνονταν ως γεωγραφικό μήκος, γεωγραφικό πλάτος. Έτσι για τη μετατροπή μεταξύ x,y και longitude, latitude χρησιμοποιήθηκαν οι προσεγγιστικοί τύποι: $x = 111000 * \cos(\text{latitude}) * \text{longitude}$, $y = 111000 * \text{latitude}$, όπου τα x, y που λαμβάνουμε είναι σε μέτρα.

Με τα δεδομένα σε καρτεσιανό επίπεδο ήταν εύκολο να βρούμε την απόσταση μεταξύ δύο σημείων σε μέτρα ως $\text{distance} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$. Τέλος, αυτό που αξίζει να σημειώσουμε εδώ είναι ο τρόπος λειτουργίας της συνάρτησης `getClosestPointOnLine()` η οποία παίρνει σαν παράμετρο ένα σημείο και μία ευθεία και επιστρέφει το σημείο εκείνο της ευθείας το οποίο είναι πιο κοντά στο δοθέν σημείο. Έχουμε το ακόλουθο βοηθητικό σχήμα.



Δίνονται τα σημεία με συντεταγμένες (x_1, y_1) , (x_2, y_2) τα οποία προσδιορίζουν την ευθεία και στο σχήμα προσδιορίζουν το σημείο A ως $x_A = x_2 - x_1$ και $y_A = y_2 - y_1$. Δίνεται επίσης το σημείο $\Sigma(x_\Sigma, y_\Sigma)$ και ζητείται το $K(x_K, y_K)$. Μέσα από τη μελέτη του σχήματος εύκολα προκύπτει ότι οι συντεταγμένες x_K, y_K δίνονται από τις σχέσεις : Αν $x_A = 0$ τότε προφανώς $x_K = x_1 = x_2$ και $y_K = y_1 + y_\Sigma$. Αν τώρα $x_A \neq 0$ τότε έχουμε: $a = y_A / x_A$, $x_K = x_1 + (a * y_\Sigma - x_\Sigma) / (a * a - 1)$, $y_K = y_1 + a * x_K$.

- **package demo:** Εδώ έχουμε κλάσεις για μια εφαρμογή demo για τον server ώστε να δοκιμάσουμε το σύστημά μας χωρίς πραγματικό GIS server αλλά με εικονικό ο οποίος δίνει όμως πραγματικά στοιχεία για μια περιοχή της επιλογής μας. Η επιλογή της περιοχής στην παρούσα εφαρμογή έγινε από το Google maps. Χρησιμοποιήσαμε δεδομένα χάρτη (εικόνα σε μορφή jpeg) και δημιουργήσαμε οδηγίες οδήγησης για μια υποτιθέμενη διαδρομή στη συγκεκριμένη περιοχή. Ο χάρτης της περιοχής πρέπει

να βρίσκεται στη θέση C:\DemoServices\map.jpg ώστε το σύστημα να δουλέψει σωστά. Οι οδηγίες οδήγησης βρίσκονται στη συνάρτηση getDemoInstructions() της κλάσης DemoResponseCalculations.java. Με τον τρόπο αυτό μπορέσαμε να δοκιμάσουμε το σύστημα χωρίς να είναι απαραίτητο να έχουμε πραγματικά δεδομένα.

- **package opensls_server:** Περιέχει μόνο μία κλάση, την OpenLS_Service η οποία υλοποιεί τα web services που παρέχουμε. Αυτά είναι τα core services από το πρότυπο “OpenGIS Location Services (OpenLS): Core Services” OGC 05-016. Συγκεκριμένα οι συναρτήσεις που υλοποιούν τα web services είναι: Για το Route Service έχουμε την determineRoute(), για το Directory Service την directory(), για το Gateway Service την gateway(), για το Location Utility Service τις geocode(), reverseGeocode(), και τέλος για το Presentation Service τις getPortrayMap(), getPortrayMapCapabilities(). Στο υπό-package opensls_server.generated υπάρχουν κλάσεις που παράγει αυτόματα ο JBuilder για τα services αυτά. Οι κλάσεις αυτές που παράγονται αφορούν κυρίως στο χειρισμό των δεδομένων που ανταλλάσσονται, την ενθυλάκωσή τους σε SOAP messages κτλ.

6.2.2 *mobiNav Client*

Κατά την υλοποίηση του client της εφαρμογής (mobinavClient.jpx) κατασκευάσαμε τα ακόλουθα βασικά packages:

- **package opensls:** Το package αυτό είναι πανομοιότυπο σε κώδικα με το αντίστοιχο του server όπως περιγράψαμε παραπάνω στην έκδοση του client που χρησιμοποιεί JSR-172. Ως XML parser χρησιμοποιήσαμε τον SAX parser όπως ήδη αναφέραμε ο οποίος δίνεται με το JSR-172 για web services σε φορητές συσκευές. Αυτή η επιλογή οφείλεται από τη μια για να έχουμε πανομοιότυπο κώδικα με το server, και από την άλλη γιατί ούτως ή άλλως θα χρησιμοποιούσαμε το JSR-172 για να καταναλώνουμε web services μέσω της φορητής συσκευής. Στην δεύτερη έκδοση του client μας, βέβαια, στην οποία δε χρησιμοποιήσαμε JSR-172 για να τρέχει η εφαρμογή από φορητές συσκευές που δεν το υποστηρίζουν, ο κώδικας που αφορά στο parsing των XML αρχείων διαφέρει ελαφρώς. Μετά την εισαγωγή των κλάσεων της kXML τροποποιήσαμε τους XML parsers ώστε να χρησιμοποιούν το kXML API. Πρέπει να σημειώσουμε εδώ πάντως, ότι στην ουσία, από το package opensls, ο server χρησιμοποιεί τις κλάσεις που αφορούν στους request decoder, response encoder, ενώ ο client εκείνες που αφορούν στους response decoder, request encoder και μόνο.

- **package coord:** Όμοια και αυτό το package είναι πανομοιότυπο σε κώδικα με το αντίστοιχο package του server.
- **package mobinavclient:** Εδώ περιέχονται ουσιαστικά όλες οι κλάσεις για τον client μας. Η βασική κλάση του client είναι η NavMidlet, ενώ οι κλάσεις OpenLSBinding_Stub, OpenLSPortType, Poster, PosterListener χρησιμοποιούνται για την κλήση των web services με βάση το JSR – 172. Ο τρόπος λειτουργίας των κλάσεων αυτών περιγράφεται από το έγγραφο “MIDP: Web Services Developers Guide”, version 1.0, της NOKIA. Στο έγγραφο αυτό περιγράφεται ένα απλό παράδειγμα ενός mobile client που καταναλώνει web services με το JSR – 172 και το χρησιμοποιήσαμε ως πρότυπο για την ανάπτυξη του δικού μας κώδικα. Την ίδια δομή κώδικα έχουμε και στην έκδοση του client με kSOAP (δηλαδή χωρίς JSR-172) αλλά με μικρές τροποποιήσεις στις κλάσεις που καλούν τα web services ώστε να χρησιμοποιούν το kSOAP API. Στο υπό-package mobinavclient.ui έχουμε τις κλάσεις για το User Interface του χρήστη, είναι δηλαδή οι διάφορες οθόνες της εφαρμογής όπως αυτές περιγράφηκαν στα έγγραφα της τεχνολογίας λογισμικού.

Στο υπό-package mobinavclient.common έχουμε τοποθετήσει κάποιες βοηθητικές κλάσεις οι οποίες χρησιμοποιούνται από διάφορα σημεία μέσα στην εφαρμογή και αφορούν στη διαχείριση της βάσης δεδομένων της φορητής συσκευής μέσω του Record Management System (RMS), τις επιλογές πλοήγησης του χρήστη, τα ονόματα και τη διεύθυνση κλήσης των web services κτλ. Στο υπό-package mobinavclient.util έχουμε μια βοηθητική κλάση για το διαχωρισμό String σε tokens η οποία δεν υπάρχει στο J2ME και έτσι την κατασκευάσαμε εμείς. Τέλος, ένα πολύ σημαντικό package είναι το mobinavclient.navigation το οποίο περιέχει όλες εκείνες τις κλάσεις που υλοποιούν τη διαδικασία πλοήγησης, από την εύρεση της τρέχουσας θέσης μέχρι και την εκφώνηση των φωνητικών οδηγιών οι οποίες είναι αποθηκευμένες στη φορητή συσκευή ως αρχεία WAV. Να σημειώσουμε στο σημείο αυτό ότι όλες οι εικόνες που εμφανίζονται στην εφαρμογή εκτός από το χάρτη ο οποίος έρχεται έτοιμος από το server, σχεδιάζονται από την εφαρμογή. Αυτό σημαίνει ότι τόσο οι πυξίδα όσο και οι οδηγίες οδήγησης που εμφανίζονται είναι αποτέλεσμα σχεδίασης μετά από υπολογισμούς στην εφαρμογή. Έτσι έχουμε scalability όταν η εφαρμογή τρέχει σε διαφορετικές συσκευές ανάλογα με το μέγεθος της οθόνης της συσκευής.

6.3 Οδηγός Εγκατάστασης

6.3.1 Server Side

Απαιτήσεις Συστήματος

Για να εγκατασταθούν οι διαδικτυακές υπηρεσίες το σύστημα θα πρέπει να έχει εγκατεστημένο τον Apache Tomcat, στην έκδοση 4.1 ή μεγαλύτερη και να υποστηρίζει Java και συγκεκριμένα τουλάχιστον την έκδοση Java 1.4.2_01-b06. Εναλλακτικά, αρκεί να υπάρχει JBuilder X Enterprise, έκδοση 10.0.176.0 της Borland.

Εγκατάσταση

Για την εγκατάσταση των διαδικτυακών υπηρεσιών θα πρέπει απλά να μεταφέρουμε το φάκελο με τις κλάσεις που έχουν δημιουργηθεί για τις διαδικτυακές υπηρεσίες, ο οποίος βρίσκεται στο φάκελο “OpenLS_Services” στο CD, στη θέση “TOMCAT_HOME\webapps” και να εκκινήσουμε τον tomcat server.

Στην περίπτωση όμως που έχουμε εγκατεστημένο τον JBuilder δε χρειάζεται καμμία αντιγραφή αρχείων. Αρκεί να ανοίξουμε το project του JBuilder που βρίσκεται στο φάκελο “OpenLS_Services ” στο CD που δίδεται με τον παρόν τόμο. Στη συνέχεια απλά τρέχουμε το project και τα υπόλοιπα τα αναλαμβάνει ο JBuilder.

6.3.2 Client Side

Απαιτήσεις Συστήματος

Η φορητή συσκευή στην οποία θα εγκατασταθεί η εφαρμογή πλοήγησης θα πρέπει να υποστηρίζει Java και συγκεκριμένα J2ME (MIDP 2.0 / CLDC 1.1). Επιπλέον θα πρέπει να υποστηρίζει τα πακέτα JSR 082 και JSR 135 και προαιρετικά το JSR 172.

Εγκατάσταση

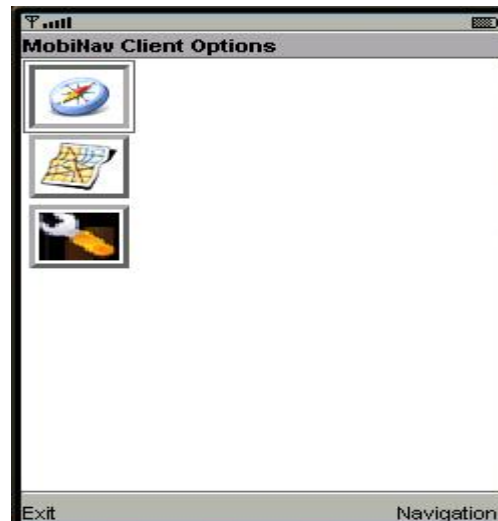
Συνδέουμε ενσύρματα ή ασύρματα τη φορητή συσκευή με τον ηλεκτρονικό υπολογιστή στον οποίο έχουμε εισάγει το CD που διατίθεται με τον παρόν τόμο. Αντιγράφουμε το αρχείο mobinavclient.jar το οποίο βρίσκεται στο φάκελο “mobinavClient” από το CD στη φορητή συσκευή.

Στην περίπτωση που η φορητή συσκευή υποστηρίζει JSR 172, επιλέγουμε την έκδοση 1 του client που είναι η μικρότερη σε μέγεθος. Σε αντίθετη περίπτωση, για συσκευές που δεν υποστηρίζουν JSR 172 επιλέγουμε την έκδοση 2 του mobinavclient.jar η οποία είναι μεγαλύτερη σε μέγεθος. Τρέχουμε το αρχείο που μεταφέραμε στη φορητή συσκευή και η εφαρμογή ξεκινά.

6.4 Εγχειρίδιο Χρήσης

Στη συγκεκριμένη ενότητα παρουσιάζεται ένα σενάριο χρήσης του συστήματός μας και συγκεκριμένα η βασική περίπτωση κατά την οποία ο χρήστης χρησιμοποιεί το σύστημα για πλοήγηση με φωνητικές οδηγίες και απεικόνιση της θέσης του σε χάρτη. Στην περίπτωση αυτή, λοιπόν, από πλευράς χρήστη έχουμε τις ακόλουθες ενέργειες.

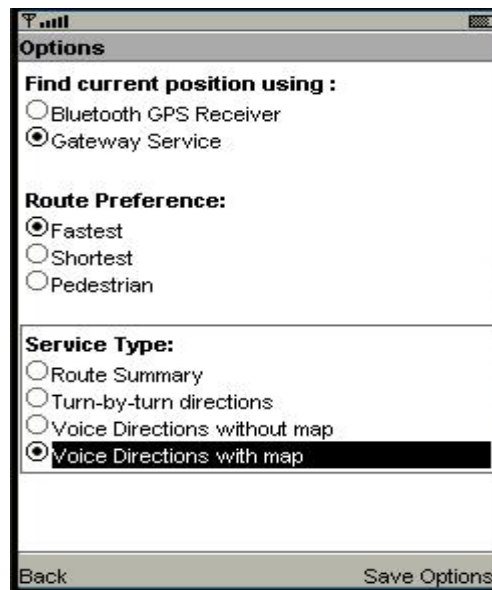
Αφού ο χρήστης εκκινήσει την εφαρμογή, εμφανίζεται στην οθόνη το όνομα του συστήματος (mobiNav) και έπειτα μεταφέρεται ο χρήστης στην οθόνη επιλογών του client η οποία φαίνεται στη συνέχεια.



Εικόνα 81: Οθόνη επιλογών client

Στην οθόνη αυτή ο χρήστης αρχικά θα πρέπει να ρυθμίσει τα settings για το σύστημα. Για να το κάνει αυτό, κάνει scroll down με τα βέλη του κινητού τηλεφώνου και επιλέγει (μέσω του select button) το τρίτο εικονίδιο (αυτό με το κλειδί) το οποίο αντιστοιχεί στις επιλογές (options). Με την επιλογή αυτή ο χρήστης μεταφέρεται στην ακόλουθη οθόνη. Εκεί πρέπει να

επιλέξει τις επιλογές που φαίνονται στην οθόνη στο σχήμα που ακολουθεί και είναι: Gateway Service, Fastest, Voice Directions with map.



Εικόνα 82: Οθόνη Options

Έπειτα πρέπει να σώσει τις επιλογές που έκανε κάνοντας κλικ στο κουμπί “Save Options” στην παραπάνω οθόνη. Με τον τρόπο αυτό αποθηκεύονται οι επιλογές του χρήστη και μεταφέρεται άμεσα πίσω στην προηγούμενη οθόνη (Οθόνη επιλογών Client). Εκεί ο χρήστης επιλέγει το πρώτο εικονίδιο (αυτό με την πυξίδα) το οποίο αφορά στην πλοήγηση. Με την επιλογή του χρήστη εμφανίζεται η παρακάτω οθόνη. Στην οθόνη αυτή ο χρήστης μπορεί να επιλέξει να ορίσει νέα διαδρομή είτε να πλοηγηθεί σε μία αποθηκευμένη διαδρομή στο “My Routes”.



Εικόνα 83: Οθόνη Routes

Στο συγκεκριμένο σενάριο ο χρήστης ορίζει μια νέα διαδρομή επιλέγοντας το κουμπί “New Route”. Μετά την επιλογή του αυτή στην οθόνη του κινητού εμφανίζεται επιλογή ορισμού σημείων αφετηρίας, ενδιάμεσων στάσεων και προορισμού (οθόνη New Route παρακάτω). Ο χρήστης επιλέγει να ορίσει προορισμό κάνοντας κλικ στο κουμπί “set End” αφού πρώτα έχει κάνει scroll down στο κουμπί “End Point”.



Εικόνα 84: Οθόνη New Route

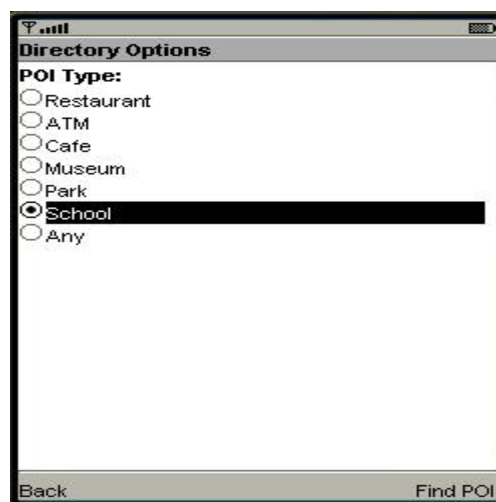
Η οθόνη που ακολουθεί στη συνέχεια, επιτρέπει στο χρήστη να επιλέξει κάποιο σημείο για προορισμό. Για το συγκεκριμένο σενάριο ο χρήστης επιλέγει την υπηρεσία “Directory

Service” για να βρει σημείο ενδιαφέροντος και να το θέσει ως προορισμό του. Ο χρήστης κάνει κλικ στο κουμπί “New POI”.



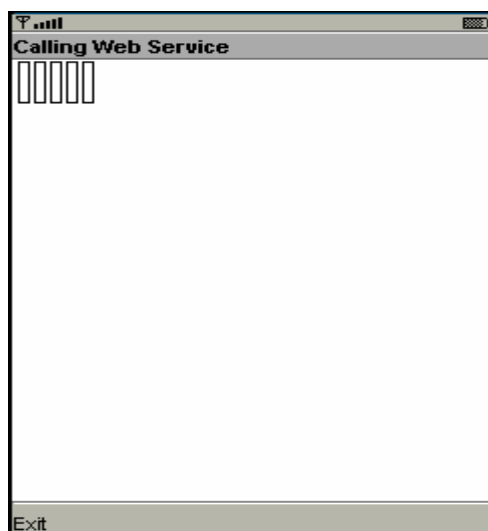
Εικόνα 85: Οθόνη Points of Interest

Με την επιλογή του αυτή ο χρήστης έχει τη δυνατότητα να επιλέξει τύπο σημείου ενδιαφέροντος (οθόνη Directory Options παρακάτω).



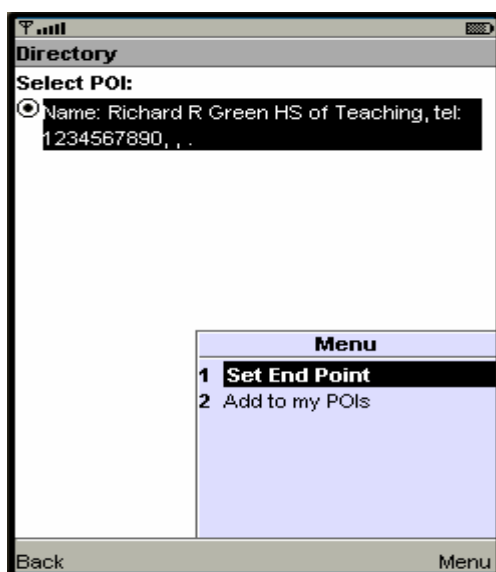
Εικόνα 86: Οθόνη Directory Options

Ο χρήστης επιλέγει κάποιον από τους τύπους σημείων (π.χ. School) και κάνει κλικ στο κουμπί “Find POI” στην οθόνη αυτή. Ακολουθεί για μερικά δευτερόλεπτα η παρακάτω οθόνη.



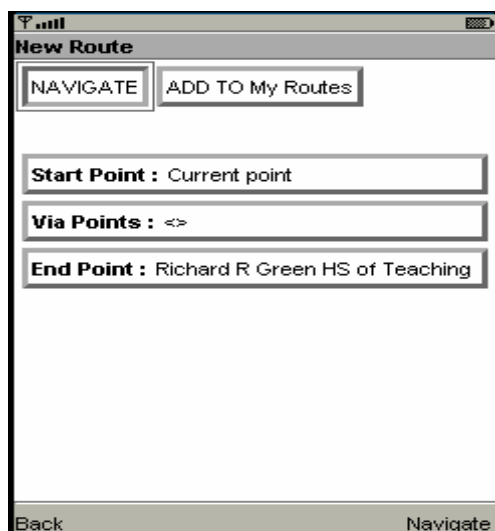
Εικόνα 87: Οθόνη Waiting

Τελικά καταλήγουμε την οθόνη αποτελέσματος (οθόνη Directory) με τα σημεία ενδιαφέροντος (POIs) που έχουν βρεθεί.



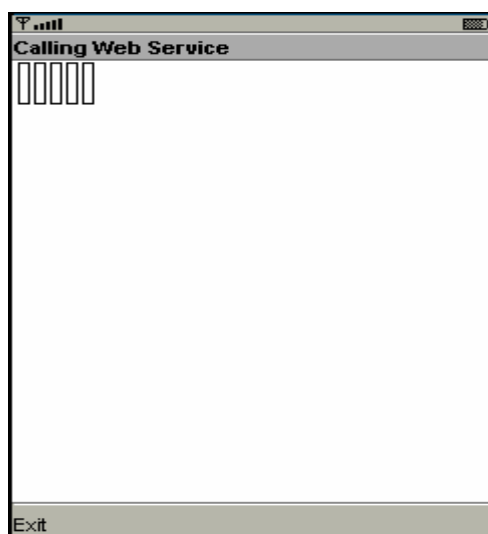
Εικόνα 88: Οθόνη Directory

Ο χρήστης επιλέγει ένα από αυτά και κάνει κλικ στο κουμπί “Menu” και ακολούθως “Set End Point” μέσω του πλήκτρου select του κινητού τηλεφώνου. Επαναφέρεται έτσι στην οθόνη New Route έχει όμως τώρα δυνατότητα να κάνει κλικ σε δύο επιπλέον κουμπιά (“Navigate” και “Add to My Routes”) όπως φαίνεται παρακάτω.



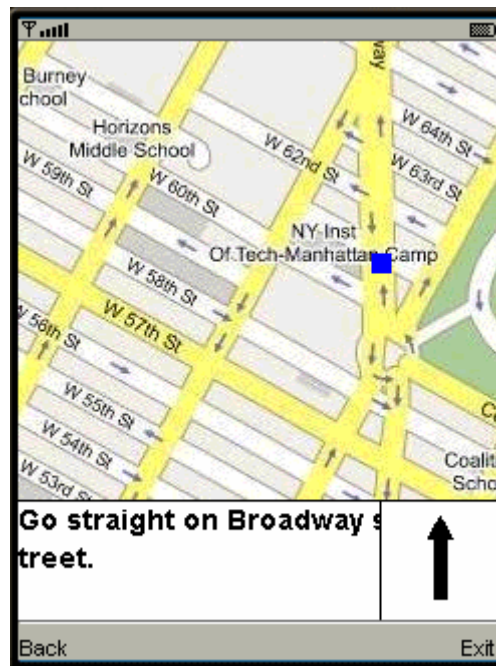
Εικόνα 89: Οθόνη New Route

Για το συγκεκριμένο σενάριο ο χρήστης επιλέγει “Navigate” και φαίνεται η παρακάτω οθόνη για μερικά δευτερόλεπτα.



Εικόνα 90: Οθόνη Waiting

Τελικά καταλήγουμε στην τελική οθόνη όπου γίνεται πλοήγηση (demo) και ο χρήστης έχει δυνατότητα να επιστρέψει στην οθόνη Routes με κλικ στο κουμπί “Back” είτε να σταματήσει την εκτέλεση της εφαρμογής κάνοντας κλικ στο κουμπί “Exit”. Στην οθόνη αυτή απεικονίζεται η τρέχουσα θέση του χρήστη με μπλε χρώμα πάνω στο χάρτη και η τρέχουσα γραπτή οδηγία. Κατά τη διάρκεια της πλοήγησης οδηγίες αλλαγής κατεύθυνσης δίνονται και φωνητικά.



Εικόνα 91: Οθόνη Navigation

Για περισσότερες λεπτομέρειες για τη χρήση του συστήματος, τις διάφορες οθόνες του και αναλυτική περιγραφή όλων των περιπτώσεων χρήστης μπορείτε να απευθυνθείτε στο κεφάλαιο «Ανάλυση και σχεδίαση του συστήματος» του παρόντος τόμου.

7

Επίλογος

7.1 Σύνοψη και συμπεράσματα

Ανακεφαλαιώνοντας, μπορούμε να ισχυριστούμε ότι σε γενικές γραμμές ο αρχικός στόχος που είχαμε θέσει επετεύχθη. Υλοποιήθηκε γενικά το σύστημά το οποίο είχαμε σχεδιάσει τόσο από πλευράς client όσο και από πλευράς server.

Από πλευράς server έχουμε υλοποιήσει τις πέντε βασικότερες διαδικτυακές υπηρεσίες ή υπηρεσίες-πυρήνα κατά το πρότυπο “OpenGIS Location Services (OpenLS): Core Services”, OGC 05-016. Έτσι έχουμε την παροχή των εξής υπηρεσιών:

- Directory Service για να λαμβάνουμε τα διάφορα σημεία ενδιαφέροντος.
- Gateway Service για να λαμβάνουμε την τρέχουσα θέση του κινητού τηλεφώνου σε περίπτωση που δεν υπάρχει Bluetooth GPS Receiver.
- Location Utility Service (Geocoder / Reverse Geocoder) για να βρίσκουμε διευθύνσεις πάνω στο χάρτη και το αντίστροφο.
- Presentation Service για να λαμβάνουμε τους χάρτες, σε JPEG format προς το παρόν μόνο.
- Route Service για να λαμβάνουμε τις διάφορες υπολογισμένες διαδρομές με οδηγίες

οδήγησης και άλλες πληροφορίες.

Φυσικά όλες οι παραπάνω υπηρεσίες είναι υλοποιημένες ως SOAP/XML web services.

Από την άλλη πλευρά, έχουμε υλοποιήσει εφαρμογή client σε J2ME και έτσι μπορεί να εγκαθίσταται σε οποιοδήποτε Java enabled κινητό τηλέφωνο. Η εφαρμογή αυτή έγινε σε δύο διαφορετικές εκδόσεις, μία με χρήση JSR 172 και μία με χρήση kSOAP και kXML για την κλήση web services. Οι δύο διαφορετικές εκδόσεις έγιναν διότι το JSR 172 δεν υποστηρίζεται από πολλά κινητά σήμερα και έτσι έπρεπε να βρούμε εναλλακτικό τρόπο για κλήση των web services. Το κινητό λοιπόν είναι σε θέση να καλεί τα παραπάνω SOAP/XML web services που κατασκευάσαμε, είναι επιπλέον σε θέση να συνδέεται με Bluetooth GPS Receiver όταν υποστηρίζει JSR 082 και τέλος μπορεί και παίζει φωνητικές οδηγίες όταν υποστηρίζει το JSR 135. Ο client λοιπόν, με τις παραπάνω ιδιότητες, παρέχει στον τελικό χρήστη τις εξής υπηρεσίες:

- Εκτιμώμενος χρόνος και απόσταση για μία διαδρομή
- Οδηγίες «Στροφή – Στροφή»
- Πλοήγηση με φωνητικές οδηγίες χωρίς χάρτη
- Πλοήγηση με φωνητικές οδηγίες με χάρτη
- Απεικόνιση της θέσης του χρήστη στο χάρτη σε πραγματικό χρόνο

Κατά τη σχεδίαση των παραπάνω εφαρμογών παρουσιάστηκαν διάφορα προβλήματα, ορισμένα εκ των οποίων ξεπεράστηκαν με ευκολία και άλλα δυσκολότερα. Όσον αφορά στις διαδικτυακές υπηρεσίες τις οποίες υλοποιήσαμε, μπορούμε να πούμε ότι τα εργαλεία τα οποία παρέχει ο JBuilder που χρησιμοποιήσαμε μας βοήθησαν πολύ, τόσο στην ανάπτυξη των υπηρεσιών, όσο και στο deploy αυτών στον ενσωματωμένο Tomcat server. Το μόνο πρόβλημα που συναντήσαμε κατά την ανάπτυξη των web services ήταν ότι δεν μπορούσαμε αυτόματα να κάνουμε Data Binding μέσω του Castor στα XML schemas του OGC όπως έχουμε ήδη περιγράψει στο κεφάλαιο 7.2. Επιπλέον, παρουσιάστηκε το πρόβλημα του πώς να μεταφέρουμε το χάρτη ο οποίος ήταν σε JPEG format μέσα σε XML αρχείο. Η λύση δόθηκε με τη μετατροπή του σε string κωδικοποιημένο σε base 64.

Κατά την υλοποίηση του client παρουσιάστηκαν περισσότερα προβλήματα. Παρατηρήσαμε ότι ενώ τα βασικά πακέτα που υλοποιούν το J2ME υποστηρίζονται γενικά από τις περισσότερες φορητές συσκευές σήμερα, δε συμβαίνει το ίδιο και με τα διάφορα προαιρετικά

πακέτα που χρησιμοποιήσαμε, όπως τα JSRs 172, 082, 135. Εναλλακτική λύση για τη μη χρήση των προαιρετικών αυτών πακέτων βρέθηκε μόνο για την περίπτωση του JSR 172 για τα web services. Στην περίπτωση αυτή, κατασκευάσαμε δεύτερη έκδοση του client που χρησιμοποιεί τα πακέτα kSOAP, kXML, πιο ώριμες τεχνολογίες στον τομέα των web services, XML parsing. Βέβαια είχαμε το κόστος της πρόσθεσης επιπλέον κλάσεων στην εφαρμογή γεγονός που μεγαλώνει το μέγεθός της αλλά τώρα δεν χρειάζεται το κινητό τηλέφωνο να υποστηρίζει JSR 172.

Αφού ολοκληρώσαμε την εφαρμογή τη δοκιμάσαμε σε δύο διαφορετικές φορητές συσκευές, στο κινητό τηλέφωνο Sony Ericsson k700 και στο PDA του εργαστηρίου. Και στις δύο περιπτώσεις είχαμε κάποιο πρόβλημα στις απαιτητικές από πλευράς μνήμης περιπτώσεις της πλοήγησης με πυξίδα και πλοήγησης με χάρτη. Το πρόβλημα ήταν ότι η εφαρμογή σταματούσε να εκτελείται ξαφνικά στις περιπτώσεις αυτές, πιθανότατα λόγω έλλειψης διαθέσιμης μνήμης. Επειδή σε αυτές τις δύο περιπτώσεις γίνεται επαναληπτική κλήση web services, οδηγηθήκαμε στο συμπέρασμα ότι οι φορητές συσκευές σήμερα από πλευράς JVM (Java Virtual Machine) δεν είναι πολύ ώριμες στον τομέα των web services.

Παρόλα αυτά, πρέπει να σημειώσουμε ότι η εφαρμογή δούλεψε ικανοποιητικά στον emulator, τόσο από πλευράς server όσο και από πλευράς client με τη μόνη διαφορά ότι τα δεδομένα που χρησιμοποιήθηκαν ήταν πειραματικά. Προσομοιώνουν βέβαια τα πραγματικά δεδομένα τα οποία θα έστελνε ένας GIS server αλλά δεν δοκιμάστηκε η εφαρμογή πλήρως ακόμη στην πράξη, λόγω τεχνικών δυσκολιών, για τους λόγους που αναφέραμε παραπάνω αλλά και διότι δεν υπήρχε έτοιμος GIS server συμβατός με το πρότυπο του OGC που χρησιμοποιήσαμε για να γίνει δοκιμή.

7.2 Μελλοντικές επεκτάσεις

Το σύστημα το οποίο υλοποιήσαμε μπορούμε να πούμε ότι αποτελεί μια εναλλακτική προσέγγιση στην κατασκευή εφαρμογών πλοήγησης. Βέβαια, πρέπει να γίνουν αρκετές επεκτάσεις πάνω σε αυτή την αρχική έκδοση του συστήματος έτσι ώστε το σύστημα αυτό να μπορέσει να είναι μία πλήρης εφαρμογή πλοήγησης και να έχουν εκπληρωθεί όλοι οι στόχοι που είχαν αρχικά τεθεί.

Αρχικά θα πρέπει να γίνει δοκιμή της υπάρχουσας έκδοσης του συστήματος στην πράξη, με πραγματικά δηλαδή δεδομένα. Για να γίνει αυτό εφικτό πρέπει να λυθεί πρώτα το πρόβλημα

με τη σύνδεση Bluetooth μέσω Java. Αυτό σημαίνει ότι πρέπει να βρεθεί συσκευή που να υποστηρίζει το JSR 082 ή να βρεθεί εναλλακτικός τρόπος σύνδεσης. Για παράδειγμα να γίνεται η σύνδεση Bluetooth με native κώδικα του κινητού (για παράδειγμα C++) και η κλήση της native μεθόδου να γίνεται μέσω JNI.

Επιπλέον, πρέπει να γίνει σύνδεση με GIS server τέτοιο ώστε να επιστρέφει τα δεδομένα σε format βασισμένο στο πρότυπο του OGC που χρησιμοποιήσαμε. Σε διαφορετική περίπτωση, θα πρέπει να κατασκευαστεί ένας μετατροπέας των δεδομένων στο επιθυμητό format οπότε έτσι μπορούμε να συνδεθούμε σε οποιοδήποτε GIS server.

Από πλευράς βελτιώσεων του υπάρχοντος κώδικα, μπορούν να γίνουν οι εξής επεκτάσεις:

- Διαχείριση δεδομένων και ιδίως χαρτών από το Record Management System (RMS). Μείωση των κλήσεων σε web services μέσω caching δεδομένων.
- Βελτίωση της απεικόνισης των χαρτών στην οθόνη μέσω pre-fetching κομματιών χάρτη στα οποία εκτιμάται ότι θα μεταβεί ο χρήστης στα επόμενα δευτερόλεπτα κατά την οδήγηση.
- Υποστήριξη παραπάνω format χαρτών εκτός από JPEG. Καλό είναι να δοθεί έμφαση στο vector format (π.χ. SVG) το οποίο θα βελτιώσει την εικόνα που φαίνεται στην οθόνη σε διάφορες κλίμακες. Θα μειωθεί επίσης το μέγεθος των δεδομένων που διακινούνται μέσω των web services.
- Υλοποίηση διαφόρων προφίλ χρηστών ορισμένα από τα οποία έχουμε ήδη περιγράψει στο κεφάλαιο 4.
- Υλοποίηση μοντέλου χρέωσης των υπηρεσιών με βάση ίσως κάποιο από τα μοντέλα χρέωσης που έχουμε περιγράψει στο κεφάλαιο 4.
- Υλοποίηση authentication χρηστών για χρήση των υπηρεσιών.
- Επέκταση ίσως των δυνατοτήτων του συστήματος τόσο στον server όσο και στον client με χρήση του πλήρους προτύπου του OGC και όχι μόνο του υποσυνόλου που χρησιμοποιήσαμε.
- Πρόβλεψη ενεργειών από πλευράς εφαρμογής client στην περίπτωση που έχουμε εισερχόμενη κλήση κατά τη διάρκεια για παράδειγμα μίας πλοήγησης.
- Μείωση του μεγέθους της εφαρμογής με μία απλή μείωση του μεγέθους των αρχείων ήχου σε συμπίεσμένη μορφή (π.χ. .MP3) με αλλαγή στον κώδικα ώστε να μπορεί να παίζει τέτοια συμπίεσμένα αρχεία.

- Προσθήκη επιπλέον features στον client όπως για παράδειγμα δυνατότητα πλοήγησης μέσω SMS και άλλα χαρακτηριστικά όπως σε άλλες σύγχρονες εφαρμογές πλοήγησης.
- Πιθανή επέκταση του προτύπου του OGC που χρησιμοποιήθηκε για να είναι κατάλληλο για τις ανάγκες πεζών χρηστών του συστήματος με την προσθήκη φωνητικών οδηγιών ακόμη και με τα ονόματα των οδών ή και των σημείων ενδιαφέροντος αποθηκευμένων στον server και πρόβλεψη ενός XML schema για τη μεταφορά τους στον client μέσω web services.
- Γενική βελτίωση του User Interface του χρήστη ώστε να γίνει όσο το δυνατόν πιο εύχρηστο.

Αυτές είναι ίσως οι κυριότερες επεκτάσεις που μπορούν να γίνουν στο μέλλον ώστε να έχουμε ένα ολοκληρωμένο σύστημα πλοήγησης. Βέβαια πρέπει επίσης να γίνουν πειράματα με τους χρόνους απόκρισης του συστήματος, ιδίως στην περίπτωση απότομης κλιμάκωσης του συστήματος. Τέλος, πρέπει να δούμε κατά πόσο μπορεί να ανταποκριθεί ο server σε περίπτωση ταυτόχρονων αιτήσεων από πολλούς clients και αν θα λειτουργούν όλα όπως αναμένεται.

8

Βιβλιογραφία

- [Ala02] Nadine Alameh. “Prototyping a Raster Image Re-projection Web Service: Challenges and Lessons”, Submitted to EOGeo 2002
- [ALK] ALK Technologies home page : <http://www.alk.com/>
CoPilot
- [ALT] Altova web site: www.altova.com
- [ASF] Apache Software Foundation web site: www.apache.org
- [Atk01] Rob Atkinson. “Gazetteer (Gaz)”, OGC Document #01-036, March 15, 2001
- [Bea04] Jeff de La Beaujardiere. “Web Map Service (WMS1.3)”, OGC Document #04-024, August 2, 2004
- [BOR] Borland web site: www.borland.com
- [CDL+03] Simon Cox, Paul Daisey, Ron Lake, Clemens Portele, Arliss Whiteside. “Geography Markup Language (GML3.0)”, OGC Document #02-023r4, January 29, 2003
- [Cox03] Simon Cox. “Observations and Measurements (O&M)”, OGC Document #03-022r3, February 4, 2003
- [DCY+] Liping Di, Aijun Chen, Wenli Yang and Peisheng Zhao. “The Integration of Grid Technology with OGC Web Services (OWS) in NWGIS for NASA

EOS Data”

- [DES] Destinator Technologies home page: www.destinatortechologies.com/
- [DR01] Allan Doyle, Carl Reed. “Introduction to OGC Web Services”, May 30, 2001
- [EBU] http://www.e2b.gr/pressroom/news_coverage/e-business_march_01.htm
- [Eva03] John Evans. “Web Coverage Service (WCS)”, OGC Document #03-065r6, October 16, 2003
- [Fal04] Sandra Fallman. “Using WFS to build GIS support”, April 7, 2004
- [FK03] Stephane Fellah, Steven Keens. “OWS Messaging Framework (OMF)”, OGC Document #03-029, January 20, 2003
- [GMA] Google Maps API documentation:
<http://www.google.com/apis/maps/documentation/>
- [ISO] <http://www.iso.ch/iso/en/>
- [JME] Java ME web site www.java.sun.com/javame
- [JR] JAX-RPC: <http://java.sun.com/xml/jaxrpc/>
- [JRT] JAX-RPC tutorial:
<http://java.sun.com/webservices/docs/1.0/tutorial/doc/JAXRPC.html>
- [JSE] Java SE web site www.java.sun.com/javase
- [JSR] JSR : <http://jcp.org/en/jsr/overview>
- [JWS] Java Web Services. David Chappell, Tyler Jewell. Publisher: O'Reilly. First Edition March 2002
- [Kol04] Kris Kolodziej. “OpenGIS Web Map Server Cookbook”, OGC Document #03-050r1, November 4, 2004
- [KSO] kSOAP web site: <http://ksoap.objectweb.org/>
- [KXM] kXML web site: <http://kxml.objectweb.org/>
- [Lak01] Ron Lake. “Location Organizer Folder (LOF)”, OGC Document #01-037, March 30, 2001
- [Lan02] Jeff Lansing. “Coverage Portrayal Service (CPS)”, OGC Document #02-019r1, February 28, 2002
- [Mab05] Marwa Mabrouk. “OpenGIS Location Services (OpenLS): Core Services [Parts 1-5] (OLS Core)”, OGC Document #05-016, May 2, 2005
- [Mar01] Serge Margoulies. “Geocoder (GeoC)”, OGC Document #01-026r1, March 28, 2001

- [McC02] Tom McCarty. “Sensor Collection Service (SCS)”, OGC Document #02-028, April 19, 2002
- [MIC] viaMichelin Web Services (Επίσημο site): <http://ws.viamichelin.com/wswebsite/gbr/jsp/hme/MaHomePage.jsp>
- [MMP] Επίσημο site της Microsoft MapPoint: <http://www.microsoft.com/mappoint>
- [MP] <http://www.microsoft.com/mappoint/products/webservice/default.aspx>
- [MPM] Περιοδικό για το mappoint: <http://www.mp2kmag.com/>
- [MPM] Mobile Payments in M-Commerce, September 2002: <http://ebusinessforum.gr/content/downloads/MobilePaymentsinMCommrce.pdf>
- [MPS] Microsoft PocketStreets 2005 web site : <http://www.microsoft.com/windowsmobile/downloads/pocketstreets/default.msp>
- [NAG] Οργάνωση συστήματος & δεδομένων της Εθνικής Υποδομής Γεωγραφικών Πληροφοριών: <http://ontogeo.ntua.gr/nagii/>
- [Neb04] Doug Nebert. “Catalogue Services (CAT)”, OGC Document #04-021r2, August 2, 2004
- [NGM] Preparing for the New Geospatial Metadata Horizon, ESRI User Conference San Diego, CA • July 9, 2003
- [NME] NMEA web site : <http://www.nmea.org/>
- [NMW] NMEA – Wikipedia: <http://en.wikipedia.org/wiki/NMEA>
- [NOK05] “MIDP: Web Services Developers Guide”, version 1.0, NOKIA, Oct 2005
- [OAS] 01-111 - OpenGIS Abstract Specification, Topic 11: OpenGIS Metadata (ISO/TC 211 DIS 19115), version 5
- [OGC] Open Geospatial Consortium : <http://www.opengeospatial.org/>
- [Per04] George Percivall. “OWS1.2 Image Handling Design (Image Handling)”, OGC Document # 04-051, September 26, 2004
- [Rei01] Louis Reich. “Web Registry Server (WRS)”, OGC Document #01-024r1, January 26, 2001
- [RUP] Rational Unified Process - Wikipedia: http://en.wikipedia.org/wiki/Rational_Unified_Process
- [Ryd99] Keith Ryden. “Simple Features - SQL (SFS)”, OGC Document #99-049, May 5, 1999

- [SAA+] Ahmet Sayar, Mehmet S. Aktas, Galip Aydin, Geoffrey Fox and Marlon Pierce. “Developing a Web Service-Compatible Map Server for Geophysical Applications”
- [Sch05] Peter Schut. “Web Processing Service (WPS)”, OGC Document #05-007, January 24, 2005
- [SDM] <http://www.codeproject.com/netcf/CfSmartDeviceMap.asp>
- [Sin01] Raj Singh. “Web Terrain Server (WTS)”, OGC Document #01-061, August 24, 2001
- [SIT] Standards of ISO Technical Committee 211, από το επίσημο site του Geographical Survey Institute: http://www.gsi.go.jp/PCGIAP/bangalore/bangalore_isorep.pdf
- [Sko05] Εμμανουήλ Στ. Σκορδαλάκη. “Λογισμική Μηχανική, Μια εισαγωγή”, ΕΜΠ, Οκτ. 2005
- [Son05] Jerome Sonnet. “Web Map Context Documents (WMC)”, OGC Document #05-005, May 3, 2005
- [SP] Έγγραφο SOAP στο World Wide Consortium: <http://www.w3.org/2002/ws/>
- [SPA] Σελίδα της Apache για SOAP: <http://ws.apache.org/soap/>
- [Sta04] Γ. Ι. Στασινόπουλος, “XML και σχετικές τεχνολογίες”. Δεκ. 2004
- [SUN] Sun web site www.java.sun.com
- [SVA] SVG advantages: <http://www.svggeneral.com/content.asp?contentid=87>
- [SVO] SVG vs Other Formats: http://www.gpsvisualizer.com/faq.html#output_format
- [TOM] TomTom official web site : <http://www.tomtom.com/>
- [TWS] <http://webteam.waikato.ac.nz/Talks/WebServices/slide15-0.html>
- [UDD] UDDI: <http://www.uddi.org>
- [Vre05] Peter Vretanos. “Web Feature Service (WFS)”, OGC Document #04-094, May 3, 2005
- [Whi04] Arliss Whiteside. “OWS1.2 Image Handling Requirements (Image Handling Req)”, OGC Document #04-052, September 26, 2004
- [Whi05] Arliss Whiteside. “OGC Web Services Common Specification (Common)”, OGC Document #05-008c1, May 3, 2005
- [WIK] <http://en.wikipedia.org/>
- [WMF+05] Arliss Whiteside, Markus U. Müller, Stephane Fellah, Frank Warmerdam.

“Web Coordinate Transformation Service (WCTS)”, OGC Document #05-013, April 13, 2005

[WSD] WSDL: <http://www.w3.org/TR/wsdl>

[WSI] Επίσημο site οργάνισµού WS-I (WebServices-Interoperability):
<http://www.ws-i.org/>

[WSL] WSIL: <http://www-106.ibm.com/developerworks/webservices/library/ws-wsilspec.html>

9

Παράρτημα

9.1 Πρότυπα για διάθεση χωρικής πληροφορίας

Στο κεφάλαιο αυτό γίνεται μελέτη των προτύπων που υπάρχουν σήμερα για τη διάθεση χωρικών πληροφοριών μέσω του διαδικτύου. Συγκεκριμένα μελετάμε τα πρότυπα τα οποία δίνονται από τους οργανισμούς OGC και ISO.

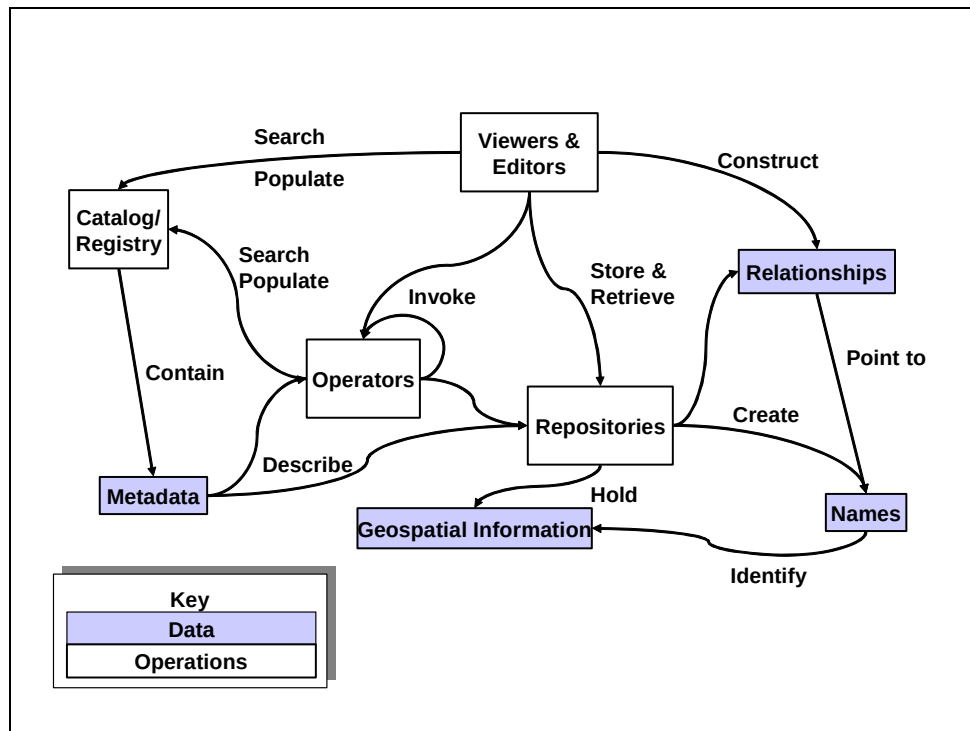
9.1.1 OGC – Open Geospatial Consortium

Ο Open Geospatial Consortium [OGC], είναι ένας μη κερδοσκοπικός, διεθνής, εθελοντικός οργανισμός, ο οποίος σκοπό έχει την ανάπτυξη προτύπων για υπηρεσίες βασισμένες στη γεωγραφική πληροφορία (geospatial and location based services). Το όραμα του οργανισμού είναι «ένας κόσμος στον οποίο όλοι έχουν όφελος από τη γεωγραφική πληροφορία και οι υπηρεσίες είναι διαθέσιμες σε κάθε δίκτυο, εφαρμογή ή πλατφόρμα». Είναι λοιπόν έμφυτη σε αυτό το όραμα, η απαίτηση για πρότυπα, καθώς και οι στρατηγικές εκείνες ώστε αυτά τα πρότυπα να γίνουν ένα αναπόσπαστο κομμάτι της επιχειρηματικής διαδικασίας και σχεδιασμού.

Τα μέλη του οργανισμού είναι διάφορες ιδιωτικές επιχειρήσεις, κυβερνητικοί οργανισμοί και πανεπιστήμια. Όλοι συμμετέχουν σε μία διαδικασία για τη δημιουργία, έλεγχο και δημοσίευση των διαφόρων προτύπων για συστήματα διάθεσης γεωγραφικής πληροφορίας (GIS) και άλλες επικρατούσες τεχνολογίες. Έτσι αναπτύσσεται ένας συνεχώς αυξανόμενος αριθμός προδιαγραφών για να εξυπηρετήσουν την ανάγκη για διαλειτουργικότητα στον τομέα των συστημάτων για τη διάθεση γεωγραφικής πληροφορίας.

Για τους σκοπούς της παρούσης διπλωματικής εργασίας, εστιάζουμε περισσότερο στις προδιαγραφές που αφορούν σε Web Services που έχουν οριστεί από τον OGC. Τα OGC Web Services (OWS) [DR01] επιτρέπουν στα συστήματα επεξεργασίας γεωγραφικής πληροφορίας να επικοινωνούν μεταξύ τους διαμέσου του Διαδικτύου χρησιμοποιώντας γνωστές τεχνολογίες όπως η XML, το HTTP, κτλ. Οι OWS παρέχουν ένα πλαίσιο διαλειτουργικότητας, ανεξάρτητο φορέα, για web-based ανακάλυψη, πρόσβαση, ολοκλήρωση, ανάλυση, εκμετάλλευση και οπτικοποίηση πολλαπλών πηγών γεωγραφικής πληροφορίας, καθώς και πληροφορίας προερχόμενης από αισθητήρες (sensors). Μπορούμε να πούμε ότι οι OWS είναι ένα πλέγμα από υπηρεσίες επεξεργασίας γεωγραφικής πληροφορίας οι οποίες μπορούν να συνδεθούν μεταξύ τους σε δυναμικές, αλυσίδες για τη δημιουργία δυναμικών εφαρμογών.

Οι υπηρεσίες OWS θα επιτρέπουν στις μελλοντικές εφαρμογές να «συναρμολογούνται» κατά βούληση από επιμέρους, δικτυακά προσβάσιμες υπηρεσίες διάθεσης και επεξεργασίας γεωγραφικής πληροφορίας. Ένα OGC Web Service μπορεί έτσι να χρησιμοποιηθεί σαν ένα «μαύρο κουτί» το οποίο εκτελεί κάποιες λειτουργίες, όπως για παράδειγμα παροχή πληροφοριών οδήγησης. Όπως θα δούμε, οι υπηρεσίες αυτές μπορούν να περιγράψουν τη λειτουργία ή λειτουργίες που μπορούν να εκτελούν μέσω μετά-πληροφορίας. Έτσι είναι εφικτή η αναζήτηση υπηρεσιών και η κατανόηση του τι μπορεί να κάνει μια δοσμένη υπηρεσία. Αυτές οι υπηρεσίες είναι διευθυνσιοδοτημένες από ένα URI και είναι εξορισμού προσβάσιμες από το διαδίκτυο. Στο παρακάτω σχήμα φαίνεται ένα γενικό αρχιτεκτονικό σχήμα των OGC Web Services. Το σχήμα αυτό παρουσιάζει τις γενικές κλάσεις των υπηρεσιών που συμμετέχουν σε διάφορες λειτουργίες επεξεργασίας χωρικών δεδομένων. Επιπλέον περιγράφει τις ιδιότητες που πρέπει να έχουν οι υπηρεσίες που ανήκουν σε αυτές τις κλάσεις για να μπορούν να συνδεθούν σε χρήσιμες εφαρμογές.



Εικόνα 92: Μερικές κλάσεις και ιδιότητες της γενικής αρχιτεκτονικής των OWS

Στη συνέχεια παρουσιάζουμε μια λίστα από τις σημαντικότερες προδιαγραφές του OGC στην οποία περιλαμβάνονται φυσικά και δικτυακές υπηρεσίες:

- **WMS – Web Map Service:** Η υπηρεσία WMS [Bea04] παράγει δυναμικά χάρτες βασισμένη σε γεωγραφική πληροφορία. Αυτό το διεθνές standard ορίζει ως «χάρτη» την απεικόνιση γεωγραφικής πληροφορίας σαν ένα αρχείο ψηφιακής εικόνας κατάλληλο για εμφάνιση σε οθόνη υπολογιστή. Ένας χάρτης δεν είναι μόνο τα δεδομένα. Οι χάρτες που παράγει η υπηρεσία WMS αποδίδονται γενικά σε μορφή εικόνας όπως PNG, GIF ή JPEG, ή μερικές φορές σαν διανυσματικά γραφικά όπως το Scalable Vector Graphics (SVG) ή ακόμη και σε Web Computer Graphics Metafile (WebCGM) format. Η υπηρεσία αυτή ορίζει τρεις λειτουργίες:
 - ο Επιστρέφει μετά-πληροφορία επιπέδου υπηρεσίας
 - ο Επιστρέφει ένα χάρτη ο οποίος έχει καλά καθορισμένες γεωγραφικές παραμέτρους και παραμέτρους διαστάσεων
 - ο Επιστρέφει πληροφορία για συγκεκριμένα χαρακτηριστικά πάνω σε ένα χάρτη (προαιρετικά).

Οι λειτουργίες αυτές του Web Map Service μπορούν να «κληθούν» χρησιμοποιώντας έναν web browser και υποβάλλοντας requests σε μορφή Uniform Resource Locator (URL). Το περιεχόμενο αυτών των URL εξαρτάται από το ποια λειτουργία ζητάμε να εκτελεστεί. Συγκεκριμένα, όταν ζητάμε ένα χάρτη το URL δείχνει τι πληροφορία θα

εμφανιστεί στο χάρτη, ποιο τμήμα της γης πρόκειται να σχεδιαστεί, το επιθυμητό σύστημα αναφοράς συντεταγμένων και τις διαστάσεις της εικόνας που θα απεικονιστεί. Όταν δύο ή περισσότερες εικόνες (χάρτες) παράγονται με τις ίδιες γεωγραφικές παραμέτρους και παραμέτρους διαστάσεων, τα αποτελέσματα μπορούν να συντεθούν με επικάλυψη του ενός πάνω στο άλλο δημιουργώντας ένα σύνθετο χάρτη. Τα format εικόνων που χρησιμοποιούνται (όπως GIF ή PNG) υποστηρίζουν διαφανή background επιτρέποντας έτσι στις εικόνες που επικαλύπτονται από άλλες να είναι ορατές. Επιπροσθέτως, μεμονωμένοι χάρτες μπορούν να ζητηθούν από διαφορετικούς servers. Με τον τρόπο αυτό είναι εφικτή η δημιουργία ενός δικτύου από κατανεμημένους map servers από τους οποίους οι clients μπορούν να ζητούν απλούς χάρτες και να κατασκευάζουν σύνθετους χάρτες με προσαρμοσμένα χαρακτηριστικά.

- **WFS – Web Feature Service:** Η υπηρεσία Web Feature Service [Vre05] είναι ουσιαστικά μια διεπαφή (interface) η οποία επιτρέπει στις αιτήσεις για γεωγραφικά χαρακτηριστικά διαμέσου του διαδικτύου να έχουν υψηλή διαλειτουργικότητα. Χρησιμοποιεί μια XML-based γλώσσα (την GML, που παρουσιάζουμε στη συνέχεια) για ανταλλαγή δεδομένων. Η προδιαγραφή της WFS καθορίζει interfaces για την περιγραφή λειτουργιών χειρισμού δεδομένων για γεωγραφικά χαρακτηριστικά. Οι λειτουργίες αυτές για τον χειρισμό δεδομένων είναι η δημιουργία, η διαγραφή, η ενημέρωση ενός χαρακτηριστικού (feature instance) καθώς και δημιουργία ερωτημάτων βασισμένων σε γεωγραφικούς και μη γεωγραφικούς περιορισμούς. Η βασική υπηρεσία WFS επιτρέπει μόνο να θέτουμε ερωτήσεις και μας επιστρέφει διάφορα χαρακτηριστικά ανάλογα με αυτό που ζητάμε. Υπάρχει όμως και η transactional WFS η οποία επιτρέπει τις υπόλοιπες λειτουργίες όπως δημιουργία, διαγραφή και ενημέρωση ενός χαρακτηριστικού. Για τις διάφορες WFS λειτουργίες που αναφέραμε υπάρχουν δύο τρόποι κωδικοποίησης. Ο ένας είναι μέσω XML (HTTP POST/SOAP) και ο άλλος είναι μέσω Keyword-Value pairs (HTTP GET/REST). Πάντως σε κάθε περίπτωση βασικό πρωτόκολλο επικοινωνίας είναι το HTTP. Όσον αφορά στα δεδομένα που ανταλλάσσονται μεταξύ client και ενός WFS server ή γλώσσα που χρησιμοποιείται είναι η Geography Markup Language (GML) όπως προαναφέραμε. Η GML κωδικοποιεί δεδομένα για συντεταγμένες, γεωμετρικά δεδομένα (π.χ. πολύγωνα), καθώς και δεδομένα τοπολογίας (για παράδειγμα το δεδομένο «ο δρόμος Α καταλήγει στο δρόμο Β»). Περισσότερες λεπτομέρειες σχετικά με την WFS και θέματα υλοποίησής της μπορούν να αναζητηθούν στα σχετικά specifications που δίνει ο OGC.
- **WCS – Web Coverage Service:** Η υπηρεσία WCS [Eva03] επεκτείνει το interface του Web Map Server για να μας δώσει πρόσβαση σε γεωχωρικά επιστρώματα (coverages) τα οποία περιέχουν τιμές ή ιδιότητες γεωγραφικών τοποθεσιών, αντί για στατικούς χάρτες

σαν εικόνες. Χρειάζεται να έχει κανείς πρόσβαση σε τέτοιου είδους γεωγραφική πληροφορία (πέρα από χρήση σε απλούς viewers) για client-side απεικόνιση, multi-valued coverages, και για χρήση ως είσοδο σε επιστημονικά μοντέλα και άλλους clients.

- **CPS – Coverage Portrayal Service:** Σκοπός της συγκεκριμένης υπηρεσίας [Lan02] είναι να μας δώσει μια διεπαφή για την παραγωγή εικόνας από δεδομένα επιστρωμάτων. Ουσιαστικά δίνει επιπλέον πληροφορία πάνω σε ένα αντικείμενο που επιστρέφει η υπηρεσία WCS. Διασυνδέει τους WCS clients με τις WMS υπηρεσίες χρησιμοποιώντας τη γλώσσα Styled Layer Descriptor (SLD) ως γλώσσα υπηρεσίας. Οι διεπαφές της υπηρεσίας CPS είναι ελαφρά παραλλαγμένες σε σχέση με τις αντίστοιχες της WMS.
- **CS-W – Catalog Service Web (Catalog Interface):** Οι υπηρεσίες καταλόγου (Catalog Services) [Neb04] μας δίνουν τη δυνατότητα να δημοσιεύουμε και να αναζητούμε συλλογές από περιγραφική πληροφορία (metadata) για δεδομένα, υπηρεσίες, καθώς και άλλα αντικείμενα με σχετική πληροφορία. Τα μεταδεδομένα στους καταλόγους αντιπροσωπεύουν χαρακτηριστικά, πάνω στα οποία μπορούμε να κάνουμε ερωτήματα και να τα παρουσιάσουμε για αξιολόγηση και περαιτέρω επεξεργασία τόσο από ανθρώπους όσο και από λογισμικό. Οι υπηρεσίες καταλόγου είναι απαραίτητες για να υποστηρίξουν την «ανακάλυψη» καταχωρημένης πληροφορίας εντός μιας συνεργαζόμενης κοινότητας. Στις προδιαγραφές που δίνει ο OGC καθορίζονται τα αφηρημένα μοντέλα καθώς και τα μοντέλα υλοποίησης που απαιτούνται για τη δημοσίευση και την πρόσβαση ψηφιακών καταλόγων.
- **SFS – Simple Features – SQL:** Τα Simple Features [Ryd99] είναι ένα OpenGIS πρότυπο το οποίο καθορίζει το πώς πρέπει να αποθηκεύονται ψηφιακά τα γεωγραφικά δεδομένα (σημείο, γραμμή, πολύγωνο κτλ) με χωρικά και μη χωρικά χαρακτηριστικά. Τα Simple Features βασίζονται σε 2D γεωμετρία με γραμμική παρεμβολή μεταξύ κορυφών. Σε γενικές γραμμές, η γεωμετρία δύο διαστάσεων είναι απλή αν δεν περιέχει self-intersection. Οι προδιαγραφές των Simple Features καθορίζουν διάφορους χωρικούς τελεστές, οι οποίοι μπορούν να χρησιμοποιηθούν για να παράγει κανείς νέα γεωμετρικά σχήματα βασισμένος στα ήδη υπάρχοντα.
- **GML –Geography Markup Language:** Η GML [CDL+03] είναι μια XML-based γλώσσα, καθορισμένη για να εκφράζει γεωγραφικά χαρακτηριστικά. Εξυπηρετεί σαν γλώσσα μοντελοποίησης για γεωγραφικά συστήματα καθώς επίσης και σαν ένα ανοιχτό format ανταλλαγής για γεωγραφικές συναλλαγές στο Internet. Η GML παρέχει μια ποικιλία αντικειμένων για γεωγραφική περιγραφή, όπως γεωγραφικά χαρακτηριστικά,

συστήματα αναφοράς συντεταγμένων, γεωμετρία, τοπολογία, χρόνο, μονάδες μέτρησης και καθολικές τιμές. Έτσι, η ψηφιακή αναπαράσταση του πραγματικού κόσμου μπορεί να γίνει μέσω ενός συνόλου από χαρακτηριστικά. Ένα χαρακτηριστικό έχει ένα σύνολο από ιδιότητες, όπου κάθε ιδιότητα αποτελείται σαν μια τριάδα (όνομα, τύπος, τιμή). Ο αριθμός των ιδιοτήτων που μπορεί να έχει ένα χαρακτηριστικό, καθορίζεται από τον ορισμό του τύπου του. Μπορούμε επιπλέον να έχουμε ένα χαρακτηριστικό το οποίο να αποτελείται από ένα σύνολο από άλλα χαρακτηριστικά. Τέλος, στη GML, τα γεωγραφικά χαρακτηριστικά έχουν σαν υποτύπους τα επιστρώματα (coverages) και τα στοιχεία παρατηρήσεων (observations). Περισσότερες λεπτομέρειες για τη γλώσσα GML μπορούν να αναζητηθούν στα αντίστοιχα specifications του OGC.

- **OpenGIS Location Services (OpenLS):** Οι υπηρεσίες OpenLS [Mab05] είναι γνωστές επίσης ως GeoMobility Server (GMS), μια ανοικτή πλατφόρμα για εφαρμογές υπηρεσιών βασισμένων σε τοποθεσία. Ουσιαστικά οι υπηρεσίες OpenLS έχουν στόχο τον καθορισμό πρόσβασης σε υπηρεσίες πυρήνα (βασικές υπηρεσίες) και στους αφηρημένους τύπους δεδομένων που περιέχει ο GeoMobility Server. Οι βασικές υπηρεσίες που παρέχονται από τον GMS είναι οι εξής: Directory Service, Gateway Service, Location Utility Service (Geocoder/Reverse Geocoder), Presentation Service, Route Service.
- **Gazetteer:** Πρόκειται για μια υπηρεσία [Atk01] που δίνει πρόσβαση σε μία αρχή για τα ονόματα των τοποθεσιών, ουσιαστικά ένα «γεωγραφικό λεξικό». Η αρχή αυτή (Gazetteer) επιστρέφει τις γεωμετρίες που σχετίζονται με τις παραπάνω τοποθεσίες.
- **Geocoder:** Πρόκειται για τη γνωστή υπηρεσία γεωκωδικοποίησης [Mar01]. Παίρνει σαν είσοδο ένα string το οποίο περιέχει μια διεύθυνση κάποιον κωδικό ή άλλους γεωγραφικούς όρους και επιστρέφει τη γεωγραφική τοποθεσία που τους αντιστοιχεί (συνήθως σαν γεωγραφικές συντεταγμένες ενός σημείου).
- **Location Organizer Folder (LOF):** Πρόκειται για ένα πρότυπο [Lak01] το οποίο καθορίζει μια δομή για την οργάνωση πληροφορίας σχετικά με συγκεκριμένα γεγονότα ή γεγονότα ενδιαφέροντος.
- **Observations and Measurements (O&M):** Το πρότυπο αυτό [Cox03] περιγράφει ένα πλαίσιο και τρόπο κωδικοποίησης για τις παρατηρήσεις και τις μετρήσεις.
- **OGC Web Services Common:** Το πρότυπο αυτό [Whi05] καθορίζει τα χαρακτηριστικά τα οποία είναι, ή θα έπρεπε να είναι, κοινά σε όλα τα πρότυπα που αφορούν στις

υπηρεσίες OGC Web Services. Με τον όρο κοινά χαρακτηριστικά εννοούμε τα περιεχόμενα των λειτουργιών ερωταποκρίσεων (requests – responses), τις παραμέτρους τους, και τον τρόπο κωδικοποίησής τους. Το πρότυπο αυτό στην παρούσα μορφή του περιλαμβάνει τις υπηρεσίες WMS, WFS, WCS που αναφέραμε παραπάνω.

- **Web Coordinate Transformation Service (WCTS):** Η υπηρεσία αυτή [WMF+05] παρέχει διεπαφές για εντοπισμό θέσης, συστήματα συντεταγμένων, και μετατροπή συντεταγμένων, οι οποίες μπορούν να χρησιμοποιηθούν από γεωχωρικές εφαρμογές και άλλες υπηρεσίες. Η μετατροπή γεωχωρικών δεδομένων από ένα σύστημα συντεταγμένων αναφοράς σε ένα άλλο είναι συχνά απαραίτητη όταν χρησιμοποιούνται δεδομένα από διαφορετικές πηγές σε μια εφαρμογή. Αυτό συμβαίνει γιατί συχνά τα δεδομένα αποθηκεύονται σε διαφορετικά συστήματα συντεταγμένων ανάλογα με την εφαρμογή. Έτσι, για να χρησιμοποιήσουμε και να συνδυάσουμε δεδομένα από διαφορετικές εφαρμογές πρέπει πρώτα να τα μετατρέψουμε στο ίδιο σύστημα συντεταγμένων. Η υπηρεσία WCTS, λοιπόν, μας δίνει τη λύση σε αυτό το πρόβλημα περιλαμβάνοντας όλων των τύπων μετασχηματισμούς και μετατροπές. Η υπηρεσία παίρνει σαν είσοδο ψηφιακά χαρακτηριστικά ή επιστρώματα σε ένα σύστημα συντεταγμένων και δίνει σαν έξοδο τα ίδια χαρακτηριστικά όμως σε άλλο σύστημα συντεταγμένων.
- **Web Map Context (WMC):** Το πρότυπο αυτό [Son05] «συνοδεύει» το OGC Web Map Service και καθορίζει το πώς μπορεί να περιγραφεί μια ειδική ομαδοποίηση ενός ή περισσότερων χαρτών από έναν ή περισσότερους map servers. Η περιγραφή αυτή της ομαδοποίησης πρέπει να είναι σε ένα μεταφέρσιμο, ανεξάρτητο πλατφόρμας format για αποθήκευση σε ένα χώρο αποθήκευσης ή για μετάδοση ανάμεσα σε clients. Η περιγραφή αυτή είναι γνωστή και σαν “Web Map Context Document” ή απλούστερα “Context”. Προς το παρόν, τα παραπάνω έγγραφα (Context Documents) έχουν σχεδιαστεί για διασυνδέσεις υπηρεσίας WMS.
- **OWS Messaging Framework(OMF):** Πρόκειται για ένα πλαίσιο [FK03] για το πώς πρέπει να είναι τα μηνύματα που ανταλλάσσονται για να διεξαχθούν συνδέσεις μεταξύ των υπηρεσιών OGC Web Services. Το πλαίσιο αυτό είναι ανεξάρτητο από κάθε πρωτόκολλο μεταφοράς και κάθε κωδικοποίηση μηνυμάτων. Χρησιμοποιώντας το OMF, ένας σχεδιαστής υπηρεσιών μπορεί να επικεντρωθεί μόνο στον ορισμό και τη ροή των μηνυμάτων για κάθε ενέργεια που υποστηρίζεται από την υπηρεσία, χωρίς να χρειάζεται να ανησυχεί για την μεταφορά διανομή των μηνυμάτων αυτών. Είναι προφανές ότι με τη χρήση του OMF θα απλοποιηθούν πού οι υλοποιήσεις των υπηρεσιών OGC Web Services και θα γίνει εφικτή η διασύνδεση υπηρεσιών.

- **OWS1.2 Image Handling Design – Requirements:** Πρόκειται για δύο πρότυπα [Per04][Whi04] τα οποία καθορίζουν αφενός τη σχεδίαση του συστήματος για διαχείριση εικόνων σε υπηρεσίες OWS και αφετέρου τις απαιτήσεις για την υποστήριξη συναρτήσεων χειρισμού εικόνων. Η σχεδίαση του συστήματος που προαναφέραμε καθορίζει δύο κύριες υπηρεσίες: Την Image Archive Service και την Image Catalogue Service. Οι διεπαφές για τις δύο αυτές υπηρεσίες καθορίζονται χρησιμοποιώντας προηγούμενα καθορισμένες διεπαφές υπηρεσιών OWS.
- **Sensor Collection Service (SCS):** Η βασική λειτουργία της υπηρεσίας SCS [McC02] είναι να παρέχει μια διαδικτυακή διεπαφή προς έναν αισθητήρα, συλλογή αισθητήρων ή πληρεξούσιο αισθητήρα. Ως αισθητήρες (sensors), ορίζονται συσκευές που μετρούν φυσικές ποσότητες.
- **Web Processing Service (WPS):** Η υπηρεσία WPS [Sch05] παρέχει πρόσβαση σε υπολογισμούς ή μοντέλα τα οποία ενεργούν πάνω σε χωρικά δεδομένα. Τα δεδομένα τα οποία χρειάζεται η υπηρεσία μπορούν να είναι διαθέσιμα τοπικά, ή να μεταφέρονται μέσω δικτύου χρησιμοποιώντας πρότυπα ανταλλαγής δεδομένων όπως είναι η Geography Markup Language (GML) ή το Geolinked Data Access Service (GDAS).
- **Web Registry Service (WRS):** Η υπηρεσία καταχώρησης (Registry Service) [Rei01] ορίζει ένα μηχανισμό για ταξινόμηση, καταχώρηση, περιγραφή, αναζήτηση, διατήρηση και πρόσβαση στις πληροφορίες από τις διάφορες πηγές δεδομένων. Η υπηρεσία OGC Service Registry παρέχει μεθόδους για το χειρισμό ενός αποθηκευτικού χώρου (repository). Στις καταχωρήσεις έχουμε πρόσβαση μέσω μιας εφαρμογής για το σκοπό αυτό (Registry Client).
- **Web Terrain Server (WTS):** Ο σκοπός του WTS [Sin01] είναι να παράγει προοπτικές όψεις από γεωαναφερόμενα δεδομένα – τυπικά από τρισδιάστατα επιστρώματα.

Η χρησιμοποίηση των παραπάνω προτύπων στις γεωγραφικές εφαρμογές είναι πολύ σημαντική διότι επιφέρει πολλά πλεονεκτήματα τα οποία συνοψίζονται στα εξής:

- Η χρήση προτύπων μας διευκολύνει στο να κατανέμουμε τα γεωχωρικά δεδομένα σε διάφορες πλατφόρμες, λειτουργικά συστήματα, γλώσσες προγραμματισμού κτλ.
- Είναι ευκολότερο για τους προγραμματιστές εφαρμογών να εντάσουν τη γεωχωρική λειτουργικότητα και τα δεδομένα στις δικές τους εφαρμογές.

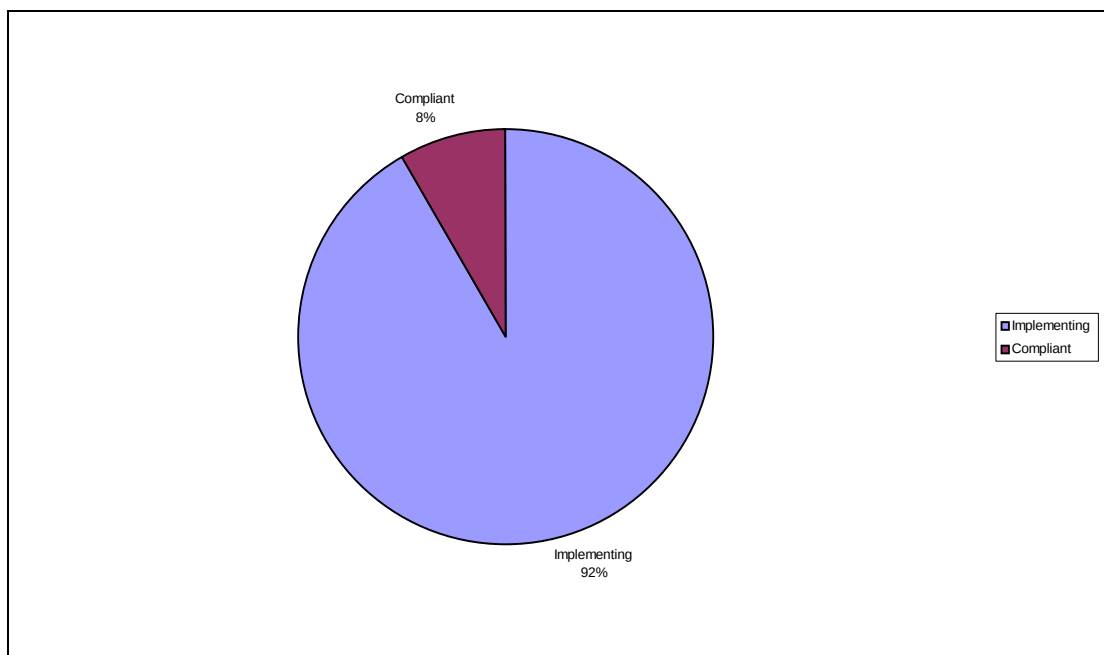
- Μπορούμε να εκμεταλλευτούμε τις τεράστιες υποδομές που έχουν φτιαχτεί για να υλοποιηθεί η αρχιτεκτονική των δικτυακών υπηρεσιών – περιλαμβανομένων των εργαλείων ανάπτυξης, εξυπηρετητών εφαρμογών, πρωτοκόλλων μηνυμάτων, υποδομών ασφαλείας κτλ.

Παρόλα τα πλεονεκτήματα που προφανώς έχουμε με τη χρησιμοποίηση των standards, δεν είναι πάντα εύκολη η μετατροπή τους σε web services. Έχουν γίνει διάφορες προσπάθειες για την πλήρη χρησιμοποίηση των προτύπων που προτείνονται αλλά δεν ήταν πάντα εφικτό. Τις περισσότερες φορές χρειαζόταν κάποιες μετατροπές ή προσθήκες για να πάρουμε το επιθυμητό αποτέλεσμα. Για παράδειγμα, σε ορισμένα standards δεν προβλέπεται ακόμη η χρησιμοποίηση SOAP/XML Web Services και επομένως όποιος θέλει να χρησιμοποιήσει την τεχνολογία αυτή πρέπει να κατασκευάσει τα αντίστοιχα interfaces στην εφαρμογή του.

Ορισμένες φορές έχει γίνει η προσπάθεια να καθοριστούν βασικές απαιτήσεις που χρειάζονται για την κατασκευή και χρήση μιας κατανεμημένης υποδομής γεωεπεξεργασίας. Η εμπειρία έδειξε ότι δεν υπάρχει μια κοινώς αποδεκτή λύση η οποία να είναι ιδανική για όλους τους χρήστες, όλους τους σκοπούς και όλες τις εφαρμογές. Υπάρχει πάντως η ανάγκη για απλές λύσεις για διεπαφές δικτυακών υπηρεσιών και αποτελεσματικής διασύνδεσης συστημάτων. Η ανάγκη αυτή οδήγησε στην προσπάθεια προτυποποίησης που γίνεται από τον OGC. Τα standards όμως που προτείνονται, αναθεωρούνται καθημερινά και τροποποιούνται με σκοπό τη βελτίωση τους. Πολλές φορές δημιουργούνται νέα standards όπου υπάρχει ανάγκη. Όλες αυτές οι βελτιώσεις – εισαγωγές νέων προτύπων προκύπτουν κυρίως από την εφαρμογή στην πράξη των ήδη υπαρχόντων προτύπων.

Πολλές εταιρίες κατασκευάζουν καθημερινά γεωγραφικά συστήματα προσπαθώντας να είναι συμβατά με τα standards που έχουν προταθεί. Έτσι φαίνεται στην πράξη αν αυτά δουλεύουν καλά ή αν θέλουν βελτίωση. Ακόμη γίνεται εμφανές αν υπάρχουν πεδία για τα οποία δεν έχουν ακόμη προβλεφθεί πρότυπα. Το ρόλο του ελέγχου των προϊόντων για το κατά πόσο είναι συμβατά με τα πρότυπα τον έχει αναλάβει ο ίδιος ο OGC μέσω ενός προγράμματος, του Compliance Testing το οποίο λειτουργεί ως εξής. Ελέγχεται αρχικά το σύστημα που έχει κατασκευαστεί ως προς το ποια από τα πρότυπα έχει υλοποιήσει. Στη συνέχεια, γίνεται έλεγχος διαλειτουργικότητας για το κατά πόσο το σύστημα αυτό μπορεί να λειτουργήσει σωστά σε ένα περιβάλλον με άλλα αντίστοιχα συστήματα που έχουν κατασκευαστεί. Αν το σύστημα καταφέρει να περάσει και τις δύο φάσεις τότε μπορεί η εταιρία που το κατασκεύασε να πάρει αντίστοιχη πιστοποίηση από τον OGC. Στο παρακάτω διάγραμμα παραθέτουμε κάποια στατιστικά στοιχεία που δείχνουν το ποσοστό των προϊόντων που βρίσκονται σε

φάση ανάπτυξης τουλάχιστον ενός standard και το ποσοστό εκείνων τα οποία είναι ήδη συμβατά με κάποιο standard.



Εικόνα 93: Ποσοστό προϊόντων συμβατών με τουλάχιστον ένα standard του OGC

Από το παραπάνω διάγραμμα, φαίνεται ότι, προς το παρόν τουλάχιστον, λίγα μόνο προϊόντα είναι εκείνα που έχουν περάσει το Compliance Test. Αυτό δείχνει από τη μία, ότι τα κριτήρια για να περάσει ένα σύστημα το Compliance Test είναι αυστηρά, αλλά από την άλλη φαίνεται και το γεγονός ότι η μετατροπή των standards σε υλοποιημένα προγράμματα δεν είναι μια απλή διαδικασία ακόμη. Πρέπει να γίνουν, λοιπόν, πολλές βελτιώσεις στο μέλλον για να διευκολυνθούν τα πράγματα στον τομέα αυτό.

Αυτό που αξίζει να σημειώσουμε σαν τελικό συμπέρασμα είναι ότι για να είναι ευρέως αποδεκτά τα GIS συστήματα στο μέλλον θα πρέπει η αρχιτεκτονική τους να χαρακτηρίζεται από απλότητα, σε συνδυασμό με δυνατότητα κλιμάκωσης και ευελιξία. Ουσιαστική ώθηση προς αυτή την κατεύθυνση θα δώσουν τα ευρέως αποδεκτά πρότυπα που θα προκύψουν.

9.1.2 ISO Metadata Model 19115:2003

9.1.2.1 Η ανάγκη για μεταδεδομένα

Η συνεχής άνοδος της τεχνολογίας, σε συνδυασμό με την ανάδειξη της σημασίας που έχει η γεωγραφία και ο τρόπος με τον οποίο τα αντικείμενα συνδέονται χωροταξικά, προκάλεσε

παγκοσμίως, την όλο και πιο διευρυμένη χρήση των ψηφιακών γεωγραφικών πληροφοριακών συστημάτων. Όλο και περισσότερα άτομα από το ευρύ κοινό και όχι μόνο από το χώρο των γεωγραφικών επιστημών, ασχολούνται με τη δημιουργία, τον εμπλουτισμό και την τροποποίηση των ψηφιακών γεωγραφικών πληροφοριών. Καθώς το πλήθος, η πολυπλοκότητα και η ποικιλία των γεωγραφικών δεδομένων αυξάνει, προέκυψε η ανάγκη ανάπτυξης μιας μεθόδου που θα εξασφαλίζει την ερμηνεία όλων αυτών των δεδομένων από κάθε σκοπιά.

Τα ψηφιακά γεωγραφικά δεδομένα είναι μια προσπάθεια για τη μοντελοποίηση, την περιγραφή και την απεικόνιση του πραγματικού κόσμου από πληροφοριακά συστήματα. Κάθε περιγραφή της πραγματικότητας είναι πάντα αφαιρετική, πάντα μερική και πάντα μία από πολλές πιθανές «όψεις» της πραγματικότητας. Αυτή η «όψη» ή το μοντέλο του πραγματικού κόσμου δεν είναι μια ακριβής και πανομοιότυπη αναπαράστασή του. Ορισμένα πράγματα έχουν υποστεί προσέγγιση, άλλα απλοποίηση, ενώ άλλα έχουν αγνοηθεί τελείως. Τέλεια, ολοκληρωμένα και ορθά δεδομένα υπάρχουν σπάνια. Για τη διασφάλιση της μη διαστρέβλωσης των δεδομένων, πρέπει να γίνεται πλήρης τεκμηρίωση των υποθέσεων και των περιορισμών που επενεργούν σ' αυτά κατά τη δημιουργία τους.

Τα μεταδεδομένα (metadata) επιτρέπουν την πλήρη περιγραφή ενός συνόλου δεδομένων, με τέτοιο τρόπο ώστε οι χρήστες να μπορούν να καταλάβουν τις υποθέσεις και τους περιορισμούς που έγιναν, καθώς και να αξιολογήσουν την καταλληλότητα των δεδομένων που τους διατίθενται σε σχέση με τη χρήση που αυτοί επιδιώκουν.

Τα γεωγραφικά δεδομένα παράγονται από ένα άτομο ή έναν οργανισμό και χρησιμοποιούνται από πολλούς διαφορετικούς χρήστες. Η επιτυχής τεκμηρίωση των γεωγραφικών δεδομένων παρέχει στους χρήστες τη δυνατότητα καλύτερης κατανόησης τους, καθώς επίσης τους καθιστά ικανούς να τα χρησιμοποιήσουν κατάλληλα. Καθώς τα γεωγραφικά δεδομένα χρησιμοποιούνται όλο και περισσότερο, η σωστή τεκμηρίωση τους θα εξασφαλίσει την καλύτερη δυνατή γνώση των υπαρχόντων χαρακτηριστικών τους και θα επιτρέψει την καλύτερη διαχείριση για τη δημιουργία, αποθήκευση, ενημέρωση και επαναχρησιμοποίηση των δεδομένων.

9.1.2.2 ISO

Ο Διεθνής Οργανισμός Προτυποποίησης ISO (International Organization for Standardization) [ISO] είναι ένα διεθνές σώμα για τον καθορισμό προτύπων στο οποίο συμμετέχουν μεγάλες

εταιρείες και τουλάχιστον ένας οργανισμός προτυποποίησης από κάθε χώρα που αποτελεί μέλος του ISO.

Το Διεθνές Πρότυπο ISO 19115 ετοιμάστηκε από την Τεχνική Επιτροπή ISO/TC 211, Geographic information/Geomatics. Περιλαμβάνει 11 παραρτήματα από τα οποία τα παραρτήματα Α έως Ε είναι κανονιστικά (normative: ο όρος αναφέρεται σε πρότυπα που έχουν εκτιμηθεί -από την πλειοψηφία- ως αποδεκτά), ενώ τα παραρτήματα F έως K είναι πληροφοριακά (informative: ο όρος αφορά στα πρότυπα που δεν ανήκουν στην κατηγορία normative, αλλά συνεισφέρουν στη σωστή χρήση και εκτέλεση ενός προτύπου).

9.1.2.3 ISO 19115:2003

Ο αντικειμενικός σκοπός του ISO 19115:2003 [OAS][ISO] είναι η παροχή μιας δομής για την περιγραφή ψηφιακών γεωγραφικών δεδομένων. Αυτό το πρότυπο προορίζεται για χρήση τόσο από αναλυτές & προγραμματιστές πληροφοριακών συστημάτων, όσο και από απλούς χρήστες, με σκοπό να κατανοήσουν τις βασικές αρχές και τις γενικές απαιτήσεις για την προτυποποίηση των γεωγραφικών πληροφοριών.

Το ISO 19115:2003 καθορίζει το σχήμα (schema) που απαιτείται για την περιγραφή γεωγραφικών δεδομένων και υπηρεσιών. Παρέχει πληροφορίες σχετικές με την αναγνώριση, την έκταση, την ποιότητα, το χωρικό και χρονικό σχήμα, την χωρική αναφορά και τη διανομή των ψηφιακών γεωγραφικών δεδομένων, καθώς και τις διαδικασίες επέκτασής τους. Λεπτομερέστερα μεταδεδομένα για ειδικούς τύπους γεωγραφικών δεδομένων ορίζονται σε άλλες σειρές του προτύπου ISO 19100 (series standards).

Ειδικότερα, το ISO 19115:2003:

- Εφοδιάζει τους δημιουργούς των δεδομένων με τις απαραίτητες πληροφορίες για το σωστό χαρακτηρισμό των γεωγραφικών δεδομένων.
- Διευκολύνει την οργάνωση και διαχείριση των μεταδεδομένων των γεωγραφικών δεδομένων.
- Καθιστά ικανούς τους χρήστες να χρησιμοποιούν γεωγραφικά δεδομένα με τον πιο αποτελεσματικό τρόπο, γνωρίζοντας τα βασικά τους χαρακτηριστικά.
- Διευκολύνει την εξεύρεση, την εξόρυξη και την επαναχρησιμοποίηση δεδομένων. Έτσι,

οι χρήστες μπορούν εύκολα να εντοπίζουν, να προσεγγίζουν, να αξιολογούν, να αγοράζουν και να χρησιμοποιούν γεωγραφικά δεδομένα.

- Δίνει τη δυνατότητα στους χρήστες να αποφασίζουν αν τα γεωγραφικά δεδομένα που εντοπίζουν κάπου θα τους είναι χρήσιμα.

Επιπλέον, το ISO 19115:2003 ορίζει:

- Υποχρεωτικά και προαιρετικά, τμήματα (sections), οντότητες (entities) και στοιχεία (elements) μεταδεδομένων.
- Το ελάχιστο σύνολο μεταδεδομένων που απαιτούνται για την εξυπηρέτηση κάθε εφαρμογής μεταδεδομένων (εύρεση δεδομένων, καθορισμός της μορφής δεδομένων που θα χρησιμοποιηθούν, πρόσβαση σε δεδομένα και μεταφορά δεδομένων).
- Προαιρετικά στοιχεία μεταδεδομένων – ώστε να είναι δυνατή μια πιο εκτεταμένη περιγραφή γεωγραφικών εφαρμογών, αν αυτό είναι απαραίτητο.
- Μια μέθοδο για την επέκταση μεταδεδομένων έτσι ώστε να προσαρμόζονται σε εξειδικευμένες ανάγκες.

Εξαιτίας των παραπάνω, το 19115:2003 είναι κατάλληλο για πολλές διαφορετικές μορφές γεωγραφικών δεδομένων όπως: χάρτες, διαγράμματα, αλλά και μη γεωγραφικά δεδομένα, όπως απλά έγγραφα.

9.1.2.3.1 Metadata Set Information

Ο πυρήνας του ISO 19115 είναι το Metadata entity set information, που αποτελείται από τα ακόλουθα σύνολα μοντέλων δεδομένων:

- Identification information. Περιλαμβάνει πληροφορίες με τις οποίες επιτυγχάνεται η ταυτοποίηση κάποιας ή κάποιων πηγών πληροφοριών (resources) με μοναδικό τρόπο.
- Constraint information. Εμπεριέχει περιορισμούς σχετικά με την πρόσβαση και τη χρήση μιας πηγής πληροφοριών ή ενός συνόλου μεταδεδομένων.
- Data quality information. Παρουσιάζει την εμβέλεια και την καταγωγή των δεδομένων (για παράδειγμα, πληροφορίες σχετικές με γεγονότα ή πηγές δεδομένων που

χρησιμοποιήθηκαν κατά την σύνταξη των δεδομένων), ενώ παραθέτει πληροφορίες σχετικές με την ποιότητα των δεδομένων.

- Maintenance information. Ορίζει την εμβέλεια (scope) των δεδομένων και τη συχνότητα ενημέρωσης τους.
- Spatial representation information. Περιλαμβάνει πληροφορίες που αφορούν στους μηχανισμούς (grid ή vector) που χρησιμοποιούνται για την παρουσίαση χωρικών πληροφοριών σε ένα σύνολο δεδομένων.
- Reference system information. Γίνεται περιγραφή των χωρικών (spatial) και χρονικών (temporal) συστημάτων αναφοράς (reference systems).
- Content information. Δίνονται πληροφορίες για τους καταλόγους γνωρισμάτων που χρησιμοποιούνται ή/και πληροφορίες για την περιγραφή ενός coverage dataset.
- Portrayal catalogue information. Περικλείει πληροφορίες για την αναγνώριση του καταλόγου αναπαράστασης (portrayal catalogue) που χρησιμοποιήθηκε
- Distribution information. Έχει πληροφορίες για τον πάροχο και επιλογές για τη λήψη δεδομένων από μια πηγή πληροφοριών.
- Metadata extension information. Δίνονται πληροφορίες για τις επεκτασεις των μεταδεδομένων, όπως αυτές ορίστηκαν από το χρήστη.
- Application schema information. Έχει πληροφορίες για το σχήμα εφαρμογής (application schema) που χρησιμοποιήθηκε για το σχηματισμό του εκάστοτε συνόλου δεδομένων.
- Extent information. Περιλαμβάνει στοιχεία δεδομένων που περιγράφουν το χωρικό και χρονικό μέγεθος κάποιας αναφερόμενης οντότητας.
- Citation and responsible party information. Περιέχει τύπους δεδομένων για την παράθεση κάποιας πηγής πληροφοριών (σύνολο δεδομένων, χαρακτηριστικά, πηγή, δημοσίευση, κτλ), καθώς και πληροφορίες για την υπεύθυνη ομάδα (πάροχος, συντάκτης, κτλ) μιας πηγής πληροφοριών.

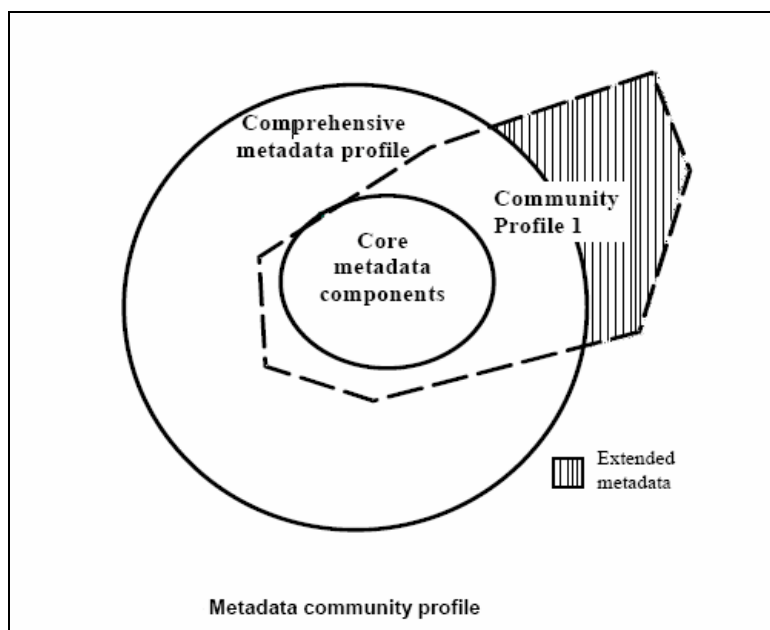
Στο ακόλουθο σχήμα παρατίθεται η περιγραφόμενη οργάνωση του Metadata entity set information υπό τη μορφή UML Package Diagram.

κάποια προτεινόμενη επέκταση και η δημιουργία ενός προφίλ κοινότητας (community profile).

Το ISO 19115 ορίζει περίπου 300 στοιχεία μεταδεδομένων με τα περισσότερα από αυτά να κατατάσσονται στην κατηγορία «προαιρετικά». Είναι σαφώς ορισμένα έτσι ώστε να βοηθούν τους χρήστες στην ακριβή κατανόηση του τι περιγράφουν. Μεμονωμένες κοινότητες, χώρες ή οργανισμοί μπορούν να αναπτύξουν ένα προφίλ (community profile) στο διεθνές πρότυπο. Επιλέγεται ένα σύνολο στοιχείων μεταδεδομένων και κατόπιν ορίζεται ως υποχρεωτικό για το συγκεκριμένο προφίλ. Ένα δοσμένο στοιχείο (π.χ. η «τιμή» σε ένα σύνολο δεδομένων) μπορεί να οριστεί ως «υποχρεωτικό» για μία συγκεκριμένη κοινότητα, η οποία πάντα θα χρειάζεται την αναφορά αυτού του στοιχείου. Μια ομάδα-κοινότητα χρηστών μπορεί να επιθυμεί τον ορισμό στοιχείων μεταδεδομένων που δεν προβλέπονται από το διεθνές πρότυπο. Ωστόσο, τα προστιθέμενα στοιχεία δε θα είναι γνωστά έξω από τα όρια της κοινότητας, εκτός και αν δημοσιευθούν.

Ένα προφίλ πρέπει να ορίζει πεδίο μεγεθών και πεδίο ορισμού για όλα τα στοιχεία μεταδεδομένων. Η επιλογή προτυποποιημένων πεδίων ορισμού σε κάποια κοινότητα είναι σημαντική, γιατί επιτρέπει τη διαλειτουργικότητα μεταξύ διαφορετικών συστημάτων. Περισσότερες πληροφορίες σχετικά με τα προφίλ κοινότητας υπάρχουν στο ISO 19106.

Στο επόμενο σχήμα παρουσιάζεται η σχέση μεταξύ των βασικών συστατικών του προτύπου (core metadata components), του συνόλου των συστατικών του προτύπου (comprehensive metadata profile) και των κατά περίπτωση εθνικών, τοπικών και οργανωτικών προφίλ.



Εικόνα 95: Σχέση μεταξύ των βασικών συστατικών του προτύπου (core metadata components), του συνόλου των συστατικών του προτύπου (comprehensive metadata profile) και των κατά περίπτωση εθνικών, τοπικών και οργανωτικών προφίλ

9.1.2.3.4 *Comprehensive dataset metadata profile – XML DTD*

Το πλήρες σύνολο των μεταδεδομένων του προτύπου είναι διαθέσιμο υπό τη μορφή ενός συνόλου από XML DTD. Το ISO 19115 κάνει αναφορές σε πολλά στοιχεία που ορίζονται σε άλλα πρότυπα. Όταν γίνεται αναφορά σε ένα τέτοιο στοιχείο (του οποίου ο ορισμός γίνεται σε άλλο πρότυπο), γίνεται χρήση της XML τιμής “ANY”. Κάτω από τη γραμμή αυτή υπάρχει σχόλιο που υποδεικνύει σε ποιο στοιχείο αναφέρεται και σε ποιο πρότυπο ορίζεται.

Αρκετά από τα στοιχεία μεταδεδομένων δεν είναι μοναδικά ονομασμένα. Για το λόγο αυτό όταν δημιουργήθηκαν τα DTD που χρησιμοποιήθηκαν στο πρότυπο αυτό ήταν απαραίτητη η μετονομασία κάποιων στοιχείων ώστε να είναι μοναδικά. Για το σκοπό αυτό ένας αλφαβητικός χαρακτήρας και μια παύλα προστέθηκε στην αρχή του ονόματος κάθε τέτοιου στοιχείου. Ένα στοιχείο με το ρόλο του «ληξίαρχου» προστίθεται σε όλα τα στοιχεία που έχουν μία λίστα κωδικών όπως το πεδίο ορισμού τους. Αυτό το στοιχείο είναι που θα δώσει την τοποθεσία του αρχείου που αποτελεί την πηγή της λίστας κωδικών.

9.1.2.3.5 Παράδειγμα

Ακολουθεί σαν παράδειγμα εκτέλεσης ένα τμήμα αρχείου δεδομένων XML που έχει αναπτυχθεί ως DTD.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE MD_Metadata SYSTEM "19115-DatasetMetadat.dtd">
<MD_Metadata id="root">
  <fileIdentifier>19115-ex-vmap0</fileIdentifier>
  <language>en</language>
  <characterSet>
    <registrar/>
    <MD_CharacterSetCode value="ucs2"/>
  </characterSet>
  <contact>
    <organisationName>National Imagery and Mapping Agency</organisationName>
    <contactInfo>
      <address>
        <deliveryPoint>4600 Sangamore Road</deliveryPoint>
        <city>Bethesda</city>
        <administrativeArea>Maryland</administrativeArea>
        <postalCode>20816-5003</postalCode>
        <country>United States</country>
      </address>
      <onlineResource>
        <linkage>http://www.nima.mil</linkage>
      </onlineResource>
    </contactInfo>
    <role>
      <registrar/>
      <CI_RoleCode value="pointOfContact"/>
    </role>
  </contact>
  <dateStamp>20000526</DateStamp>
  <metadataStandardName>ISO 19115 Geographic Information - Metadata</metadataStandardName>
  <metadataStandardVersion>DIS</metadataStandardVersion>
  <distributionInfo>
    <distributionFormat>
      <b-name>VPF</b-name>
      <version>1.0</version>
    </distributionFormat>
    <transferOptions>
      <onLine>
        <linkage>http://www.nima.mil</linkage>
      </onLine>
    </transferOptions>
  </distributionInfo>
```

Εικόνα 96: Παράδειγμα τμήματος αρχείου XML που έχει αναπτυχθεί ως DTD

Το περιεχόμενο (contact) της περιοχής που έχει επισημανθεί με χρώμα κόκκινο ορίζεται στο πακέτο Citation and responsible party information που φαίνεται στο UML διάγραμμα που προηγήθηκε σύμφωνα με το CI_ResponsibleParty. Το περιεχόμενο (distributionInfo) της περιοχής σε χρώμα πράσινο ορίζεται στο πακέτο Distribution information του ευρύτερου MD_Metadata.