

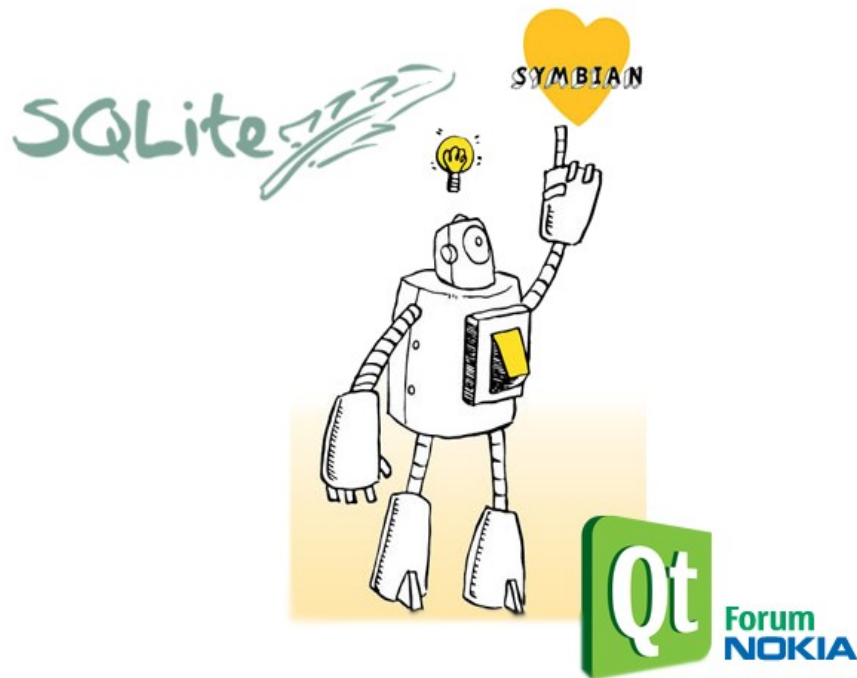


ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη διαδραστικής εφαρμογής για Smartphone



Όν/νυμο φοιτητή: Ελευθέριος Γεωργίου

Αρ. Μητρώου: 05/2864

Επιβλέπων καθηγητής:

Μιχάλης Σαλαμπάσης

ΘΕΣΣΑΛΟΝΙΚΗ 2011

Πτυχιακή εργασία του φοιτητή Ελευθέριου Γεωργίου

Πρόλογος

Στην ψηφιακή εποχή που ζούμε τα πάντα γύρω μας υπάρχουν και συμβαίνουν λόγω της ύπαρξης των δικτύων και των τεχνολογιών που τα περιβάλλουν. Κακώς ή καλώς χωρίς τα δίκτυα, είμαστε “κλινικά νεκροί”. Όπου και να κοιτάξουμε γύρω μας από την χρήση ενός απλού κινητού τηλεφώνου μέχρι την επικοινωνία μέσω δορυφόρων, τα πάντα είναι δίκτυα.

Σε αυτόν τον αιώνα που είμαστε το παρόν και το μέλλον είναι οι φορητές συσκευές και η ψηφιοποίηση των πάντων. Μία μεγάλη κατηγορία φορητών συσκευών και γεμάτη εκπλήξεις για το μέλλον είναι τα smartphones. Αυτό με έκανε να μπω στην διαδικασία να ψάξω το τι είναι αυτά και πως θα μπορούσα να προγραμματίσω πάνω σε αυτά. Η πρόταση της πτυχιακής του κ.Μιχάλη Σαλαμπάση ήταν πολύ ενδιαφέρουσα και ελκυστική και ήταν μια καλή αφορμή για την περιηγησή μου και του, μέσα στον “κόσμο” των smartphones. Η επιλογή του Symbian έγινε διότι ήταν η πρωτοπόρα την περίοδο της ανάθεσης της πτυχιακής (μπορεί τώρα να έχουν αλλάξει κάπως τα δεδομένα).

Το περιβάλλον ανάπτυξης της εφαρμογής ήταν ένα μεγάλο βοήθημα για την εκπόνηση αυτής της πτυχιακής. Με το Qt είχα και προγενέστερη επαφή στο εργαστηριακό μάθημα Μεθοδολογίες Προγ/μου I με τον καθηγητή κ. Θεοδόσιο Χειμωνίδη. Και μου ήταν κάπως οικείο το περιβάλλον αυτό και η δική του συμβολή ήταν πολύ ιδανική στην παρούσα στιγμή.

Σε αυτή την πτυχιακή θα αναλυθούν το τι είναι mobile computing, το τι είναι smartphone, θα μάθουμε πληροφορίες γύρω από την ιστορία των smartphones, τα διάφορα mobile OS. Ακόμα πληροφορίες για το Symbian OS και το Qt. Θα μάθουμε για το “πάντρεμα” του Symbian και του Qt για την δημιουργία μιας εφαρμογής.

Κλείνοντας, καθημερινά τα πάντα γύρω μας αλλάζουν μαζί και οι τεχνολογίες και όλο και περισσότερο ο Η/Υ μπαίνει στην ζωή μας, επομένως είναι επιτακτικό της εποχής μας η δημιουργία εφαρμογών σε φορητές συσκευές ευέλικτες αλλά και ταυτόχρονα τόσο δυνατές όπως μία desktop εφαρμογή.

Πτυχιακή εργασία του φοιτητή Ελευθέριου Γεωργίου

Περίληψη

Στον 21ο αιώνα που ζούμε τα πάντα γύρω μας αποτελούνται από φορητές συσκευές. Οι φορητές συσκευές είναι το αποτέλεσμα από την διαρκείς ανάπτυξη των δικτύων και των τεχνολογιών τους, του mobile computing. Ένα πολύ καλό παράδειγμα και ευρέως γνωστό είναι τα ασύρματα δίκτυα, όπου μπορούμε να τα βρούμε παντού και πλέον τα θεωρούμαι αναπόσπαστο κομμάτι. Από την απλή πλοήγηση μας στο διαδίκτυο μέσω ενός φορητού υπολογιστή μέχρι την άμεση αγορά πραγμάτων μέσω αυτού.

Στις μέρες μας, οι φορητές συσκευές που είναι ευρέως διαδεδομένες είναι τα smartphones. Στα απλά ελληνικά είναι το “έξυπνο τηλέφωνο”. Είναι ο “καρπός” του mobile computing. Είναι συσκευές που δεν είναι απλά τηλέφωνα αλλά στην πραγματικότητα είναι μια μικρογραφία ενός Η/Υ. Η ιστορία αυτών ανέρχεται πίσω στο 20ο αιώνα, μόλις το 1992 με το πρώτο smartphone της IBM το Simon. Από τότε μέχρι σήμερα έχουν αλλάξει πολλά και η μορφή όσο και η υπόσταση αυτών των “θαυμάτων” είναι τεράστια.

Για την υπόσταση των smartphones, όπως και στους υπολογιστές πρέπει να υπάρχει και ένα λειτουργικό σύστημα για να γίνεται η διαχείριση των πόρων για την συσκευή. Υπάρχουν πολλά λειτουργικά συστήματα, αλλά στην πτυχιακή μας θα μας απασχολήσει το Symbian. Το Symbian είναι ένα λειτουργικό σύστημα που είναι της εταιρίας Nokia. Έχει απίστευτες δυνατότητες και μαζί με το περιβάλλον εργασίας του Qt δίνουν στους προγραμματιστές το απαραίτητο κίνητρο για δημιουργία εφαρμογών που στο παρελθόν φάνταζαν απίθανες να υλοποιηθούν σε ένα τόσο μικρό οπτικά αντικείμενο.

Κλείνοντας, με την χρήση του Qt και του Qt Mobility, έχουμε την δυνατότητα να φτιάχνουμε εφαρμογές που αναφέρονται είτε σε Symbian os, είτε σε Windows OS, είτε σε Linux, και πάντα με το ίδιο κομμάτι κώδικα. Στο τελευταίο κεφάλαιο της πτυχιακής υπάρχει η υλοποίηση ενός πολυλεξικού για Symbian.

Τέλος, οι φορητές συσκευές είναι το μέλλον της ανθρωπότητας καθώς έχουν φτιαχτεί για να κάνουν την ζωή μας και την καθημερινότητά μας πιο απλή και χωρίς χάσιμο χρόνου.

Abstract

In the 21st century where we live in, everything around us consist of mobile devices. The mobile devices are the result of the ongoing development of networks and their technologies, the mobile computing. For example, wireless networks, which we can find them everywhere. Using them, for a simply browsing on net via a laptop computer to purchase things directly through them.

Nowadays, one very important piece of mobile devices consist of smartphones, in greek means “smart phone”. It is the “fruit” of mobile computing. Smartphones are which are not just a simple phone but is actually a miniature of a PC. The story of these around back to 20th century, was 1992 when first smartphone came to life IBM's Simon. Since then much has changed and the format and the substance of these “miracles”.

In order to have a smartphone we must have and an operating system to manage system resources. There are many operating systems which support on different devices, but on our document we will work with the Symbian OS. Symbian OS is a Nokia's operating system for smartphones. It has incredible possibilities and with the Qt Creator give to developers the necessary motivation to create applications which may seemed unreachable on the past years.

Finally, using Qt with Qt Mobility, we can create application fos Symbian OS, Windows OS, or Linux, where ever we want with the same source code. At the last chapter of this document there is a reference for a lexicon implementation for Symbian.

To sum up, the mobile devices are the future of humanity and make our lives easier without wasting time.

Ευρετήριο περιεχομένων

Πρόλογος.....	3
Περίληψη.....	5
Abstract.....	6
Ευρετήριο Σχημάτων και Πινάκων.....	11
1 Mobile computing.....	15
1.1 Εισαγωγή.....	15
1.2 Τεχνολογίες του mobile computing.....	18
1.2.1 Ασύρματες τεχνολογίες (Wireless Technologies).....	19
1.2.1.1 In-Room δίκτυα.....	19
1.2.1.2 Wireless LAN.....	20
1.2.1.3 Broadband Wireless networks.....	22
1.2.1.4 Wide Area Wireless/Radio networks.....	23
1.2.1.5 Satellite-Based networks.....	23
1.2.1.6 Cellular networks.....	24
1.2.2 Συστήματα εντοπισμού θέσης.....	26
1.2.2.1 Satellite Positioning Systems.....	26
1.2.2.2 Cellular Positioning Systems.....	26
1.2.2.3 Indoor Positioning Systems.....	27
1.3 Επίλογος.....	28
2 Smartphone και mobile λειτουργικά συστήματα.....	31
2.1 Τι είναι το Smartphone;.....	31
2.2 Ιστορική αναδρομή.....	33
2.2.1 Ιστορική αναδρομή με εικόνες.....	36
2.3 Mobile λειτουργικά συστήματα (Operating Systems OS).....	39
2.3.1 Android OS.....	39
2.3.2 BlackBerry OS.....	40
2.3.3 iOS.....	42
2.3.4 Palm webOS.....	44
2.3.5 Windows Mobile.....	45
2.3.6 Bada.....	46

2.4 Επίλογος.....	47
3 Symbian OS και Qt.....	49
3.1 Symbian OS.....	49
3.1.1 Τεχνολογία Symbian.....	52
3.1.2 Symbian - Εξέλιξη προς την πρωτοποριακή πλατφόρμα.....	53
3.1.3 Symbian – Casual Application developer.....	55
3.2 Qt.....	56
3.2.1 Ένα Cross-platform περιβάλλον ανάπτυξης.....	56
3.2.2 Qt στο Mobile Domain.....	57
3.2.3 Άδεια χρήσης του Qt.....	60
3.2.3.1 Qt Commercial Developer License.....	60
3.2.3.2 Qt GNU LGPL v2.1.....	61
3.2.3.3 Qt GNU GPL v3.0.....	61
3.3 Επίλογος.....	63
4 Qt Mobility.....	65
4.1 Εισαγωγή.....	65
4.2 Bearer Management (Φορέας Διαχείρισης).....	67
4.2.1 Δημιουργία ενός network session.....	68
4.3 Camera.....	69
4.4 Contacts (Επαφές).....	69
4.4.1 Δημιουργία μιας καινούριας επαφής.....	70
4.5 Document Gallery.....	71
4.6 Feedback (Ανατροφοδότηση).....	72
4.7 Location (Τοποθεσία).....	73
4.7.1 Τοποθεσία χρήστη και συνεχής ενημέρωση.....	75
4.8 Messaging (Μηνύματα).....	75
4.8.1 Δημιουργία και αποστολή ενός email μηνύματος.....	76
4.9 Multimedia (Πολυμέσα).....	77
4.9.1 Αναπαραγωγή ενός αρχείου ήχου.....	77
4.10 Organizer.....	77
4.11 Publish & Subscribe.....	78

4.12 Qt Service Framework.....	78
4.13 Sensors (Αισθητήρες).....	79
4.13.1 Κοινές συμβάσεις.....	79
4.15 Telephony Events (Γεγονότα του τηλεφώνου).....	82
4.16 Versit.....	82
4.17 Επίλογος.....	84
5 Εφαρμογή : “Πολυλεξικό.....	87
5.1 Εισαγωγή στην SQLite.....	87
5.2 Nokia Qt SDK 1.0.....	88
5.2.1 Απαιτήσεις συστήματος.....	88
5.2.2 Περιβάλλον ανάπτυξη εφαρμογών Qt Creator.....	89
5.2.3 Ρύθμιση περιβάλλοντος ανάπτυξης για Symbian.....	90
5.3 Περί της εφαρμογής - “Πολυλεξικό.....	91
5.3.1 Περιγραφή της βάσης.....	91
5.3.1.1 Περιγραφή πινάκων της βάσης.....	91
5.3.2 Εφαρμογή “Πολυλεξικό.....	92
5.3.2.1 Καταλήξεις αρχείων της εφαρμογής του Qt.....	92
5.3.2.2 Κάποια κομμάτια κώδικα (επεξήγηση).....	96
5.4 Extensions αρχείων για Symbian OS.....	106
5.5 Επίλογος.....	107
Βιβλιογραφία.....	109

Πτυχιακή εργασία του φοιτητή Ελευθέριου Γεωργίου

Ευρετήριο Σχημάτων και Πινάκων

Διάγραμμα 3.2.2	
Οι δυνατότητες της cross-platform Qt σε διαφορετικές κινητές και μη πλατφόρμες.....	59
Διάγραμμα 5.5	
Symbain apps publishing.....	108
Εικόνα 1.2.1.2α	
Wi-Fi Alliance.....	21
Εικόνα 1.2.1.2β	
Wi-Fi Certified.....	21
Εικόνα 1.2.1.2γ	
Wi-Fi Certified.....	21
Εικόνα 1.3α	
Mobile computing γύρω μας καθημερινά.....	28
Εικόνα 1.3β	
Mobile computing σε μία αποθήκη.....	29
Εικόνα 2.3.1α	
Android 2.2.....	40
Εικόνα 2.3.1β	
Android Emulator.....	40
Εικόνα 2.3.2α	
BlackBerry trackwheel ή trackball.....	41
Εικόνα 2.3.2β	
BlackBerry trackpad και touch-screen.....	41
Εικόνα 2.3.3α	
Portrait και Landscape mode.....	42
Εικόνα 2.3.4α	
Palm Pre.....	44
Εικόνα 2.3.4β	
Palm Pixi.....	44
Εικόνα 2.3.5α	
Windows Mobile 6.5.3.....	45

Εικόνα 2.3.6α	
Samsung Wave S8500.....	46
Εικόνα 3.2.1α	
Qt διαδικασία ανάπτυξης πρόγραμμα μία φορά και τη χρήση πολλαπλές πλατφόρμες.....	56
Εικόνα 3.2.1β	
Εμφάνιση των Qt κουμπιών σε διαφορετικές πλατφόρμες.....	57
Εικόνα 3.2.2α	
To Greenphone.....	58
Εικόνα 3.2.2β	
To user interface του Greenphone.....	58
Εικόνα 3.3	
Symbian C++ και Qt ως ένα.....	63
Εικόνα 4.13.1β	
6 κατευθύνσεις.....	80
Εικόνα 4.13.α	
Καρτεσιανό σύστημα συντεταγμένων.....	79
Εικόνα 4.13γ	
Περιστροφή γύρω από έναν άξονα.....	80
Εικόνα 4.13δ	
Όψη μιας συσκευής.....	80
Εικόνα 4.17	
Qt Mobility APIs.....	84
Εικόνα 5.2.2α	
Qt Creator.....	89
Εικόνα 5.2.2β	
Qt Creator(λεπτομέρειες).....	89
Εικόνα 5.2.3	
Συσκευή συνδεδεμένη.....	91
Εικόνα 5.3.2.1α	
Introduction.ui.....	95

Εικόνα 5.3.2.1γ	
LexiconForm.ui.....	95
Εικόνα 5.3.2β	
About.ui.....	95
Εικόνα 5.5	
Built σε περιβάλλον Symbian – χρήση του Qt.....	107
Πίνακας 1.2.1.6	
Από την 1G στην 4G γενιά.....	25
Πίνακας 1.2.2.2	
Σύγκριση κυψελοειδών συστημάτων εντοπισμού θέσης.....	27
Πίνακας 2.2.1α	
Ιστορική αναδρομή.....	36
Πίνακας 2.2.1β	
Ιστορική αναδρομή (Συνέχεια).....	37
Πίνακας 2.2.1γ	
Ιστορική αναδρομή (Συνέχεια).....	38
Πίνακας 2.4α	
Μοίρασμα των mobile OS στην αγορά (πηγή Gartner).....	47
Πίνακας 2.4β	
Σύγκριση mobile OS (1).....	47
Πίνακας 2.4β	
Σύγκριση mobile OS (2).....	48
Πίνακας 3.1: Psion Συσκευές.....	51
Πίνακας 3.2.3.....	62
Πίνακας 4.1	
Τα API του Qt Mobility v1.1.....	67
Πίνακας 4.17 (1).....	85
Πίνακας 4.17 (2).....	85

Πτυχιακή εργασία του φοιτητή Ελευθέριου Γεωργίου

1

Mobile computing

Στις μέρες μας, οι φορητές (mobile) συσκευές είναι το παν, σχεδόν τα πάντα γίνονται με την ασύρματη τεχνολογία, η οποία εξελίσσεται και εισχωρεί στην ζωή μας όλο και περισσότερο. Η καθημερινότητα των ανθρώπων βασίζεται στην φορητότητα των συσκευών και στις υπηρεσίες που προσφέρουν. Το κεφάλαιο αυτό έχει ως σκοπό να μας εισάγει στην έννοια του mobile computing και στις διάφορες mobile computing τεχνολογίες που χρησιμοποιούνται ευρέως και σχεδόν σε ότι υπάρχει γύρω μας, όπου και σε ότι κοιτάζουμε.

1.1 Εισαγωγή

Μέσα στα τελευταία χρόνια, η χρήση συσκευών που χρησιμοποιούν την ασύρματη τεχνολογία έχει αυξηθεί σε υπερβολικά μεγάλο βαθμό. Όλο και περισσότεροι άνθρωποι είναι σε θέση να έχουν πρόσβαση σε πληροφοριακά συστήματα τα οποία βρίσκονται σε ενσύρματα δίκτυα (wired networks) οπουδήποτε και οποτεδήποτε, χρησιμοποιώντας φορητού μεγέθους ασύρματες υπολογιστικές συσκευές (wireless computing devices) που τροφοδοτούνται από μπαταρίες. Παραδείγματα τέτοιων συσκευών είναι τα: notebooks, tablet PCs, PDAs (**P**ersonal **D**igital **A**ssistants). Αυτές οι φορητές υπολογιστικές συσκευές επικοινωνούν με έναν κεντρικό server μέσω ενός ασύρματου καναλιού επικοινωνίας και γίνονται ένα ακόμα κομμάτι του υπάρχοντος υπολογιστικού περιβάλλοντος (computing environment).

Επομένως, οι mobile χρήστες μπορούν να έχουν πρόσβαση σε πληροφορίες που βρίσκονται σε ένα στατικό δίκτυο (static network) ενώ αυτοί “ταξιδεύουν”, κυριολεκτικά ή μεταφορικά, με την έννοια της καθημερινής μετακίνησης. Αυτός ο τύπος του computing είναι γνωστός και ως mobile computing. Το mobile computing παρέχει data-intensive εφαρμογές με χρήσιμες πτυχές της ασύρματης τεχνολογίας

και για να υποστηρίξει η mobile τεχνολογία τέτοιου είδους εφαρμογές τις αναφέρει ως mobile databases.

Οι πάροχοι κινητών υπηρεσιών έχουν δημιουργήσει έναν αριθμό από υπηρεσίες, συμπεριλαμβανομένων: υπηρεσίες πρόγνωση καιρού, ειδήσεις, πληροφορίες δεικτών χρηματιστηρίου, συνάλλαγμα, αποτελέσματα εκλογών, τουριστικές υπηρεσίες, δρομολογίων αεροπλάνων, ερωτήματα για το που βρίσκομαι (location-dependent query), και καθοδήγηση (route guidance). Τώρα, για να αξιοποιηθούν οι δυνατότητες των ασυρμάτων πληροφοριών υπηρεσίες ένας αριθμός από ζητήματα και προκλήσεις δημιουργήθηκαν. Μερικά από αυτά είναι: διαχείριση mobile data, διαχείριση μνήμης (cache), υποδομή ασυρμάτων δικτύων, location-dependent διαχείριση δεδομένων, θέματα διαχείρισης ενέργειας, και θέματα μετάδοσης δεδομένων (data broadcasting). Όσο αναφορά, location-dependent ερωτήματα σύντομα θα έχει μεγάλο κοινό και θα παρουσιάσει ενδιαφέρον. Αυτό έχει ως συνέπεια, η παροχή αποδοτικών και αποτελεσματικών location-dependent mobile υπηρεσιών θα είναι άκρως απαραίτητη.

Παρά την πολυπλοκότητα, χάρις αυτών που εμπλέκονται για τις υπηρεσίες location-dependent πληροφοριών, πρέπει να κατανοήσουμε κάποια χαρακτηριστικά του mobile computing environment, όπως είναι:

- Περιορισμός πόρων στις mobile συσκευές : Οι mobile συσκευές γίνονται όλο και μικρότερες και ελαφρύτερες. Αυτό γίνεται με σκοπό να είναι πιο ελκυστικές και η δυνατότητα μεταφοράς πιο εφικτή. Αλλά αυτός ο σχεδιασμός συσκευών επίκειται σε κάποιους συμβιβασμούς, συμπεριλαμβανομένων: χαμηλή διάρκεια ζωής της μπαταρίας, “χαμηλή” υπολογιστική δύναμη, περιορισμό στην ικανότητα αποθήκευσης. Πιο ειδικά, με την διάρκεια ζωής μια μπαταρίας, το προσδόκιμο ζωής ενός συσσωρευτή (πχ νικελίου-καδμίου, ιόντων λιθίου) εκτιμάται ότι θα αυξηθεί ο χρόνος της πραγματικής χρήσης στο 15% μέσα στα επόμενα χρόνια. Επιπλέον, πρέπει να σημειωθεί ότι η ασύρματη μετάδοση δεδομένων απαιτεί μεγαλύτερη ποσότητα ηλεκτρικής ενέργειας, περίπου 10 φορές μεγαλύτερη από την ενέργεια που απαιτείται για την λήψη δεδομένων.
- Χαμηλού εύρους ζώνης του δικτύου (low network bandwidth) : Οι χρήστες mobile συσκευών μπορούν να συνδεθούν με το δίκτυο σταθερής

τηλεφωνίας μέσω διαφόρων ασυρμάτων δικτύων επικοινωνιών, συμπεριλαμβανομένων: ασύρματο ραδιόφωνο, ασύρματο τοπικό δίκτυο (LAN), ασύρματα κυψελοειδή, δορυφόρο και πολλά άλλα. Καθένα από αυτά τα ασύρματα δίκτυα προσφέρει διαφορετική ικανότητα εύρους ζώνης (bandwidth). Επομένως, αυτό το ασύρματο εύρος ζώνης είναι πολύ μικρό σε σχέση με αυτό που προσφέρει ένα σταθερό δίκτυο, όπως είναι το ATM (**Asynchronous Transfer Mode**), που μπορεί να παρέχει ταχύτητες που φτάνουν και τα 155Mbps. Σχεδιάζοντας, εφαρμογές που έχουν πρόσβαση σε δεδομένα μέσω ενός δικτύου, ο χρόνος απόκρισης γίνεται ένα σημαντικό ζήτημα στον τομέα του mobile computing.

- Κόστος ασύμμετρης (asymmetric) επικοινωνίας : Η διαφορετική bandwidth capacity μεταξύ του downstream επικοινωνίας και του upstream επικοινωνίας, έχει δημιουργήσει ένα καινούριο περιβάλλον που αποκαλείται Asymmetric Communication Environment. Στην πραγματικότητα υπάρχουν δύο περιπτώσεις που μπορεί να υπ' έρθει ασύμμετρη επικοινωνία:
 - α) Να οφείλεται στην ικανότητα των φυσικών συσκευών. Για παράδειγμα, οι servers έχουν πολλοί ισχυρούς πομπούς εκπομπής, ενώ οι mobile clients έχουν πολύ μικρή ικανότητα μεταφοράς.
 - β) Να οφείλεται στα πρότυπα που χρησιμοποιούν οι εφαρμογές για την ροή των πληροφοριών. Για παράδειγμα, σε μια κατάσταση που ο αριθμός των servers είναι πολύ μικρότερος από αυτό των clients, έχουμε ασύμμετρη επικοινωνία, και αυτό επειδή δεν υπάρχει αρκετή χωρητικότητα για να χειριστεί τις ταυτόχρονες αιτήσεις από τους πολλαπλούς clients.
- Ανομοιογένεια των mobile συσκευών : Οι βιομηχανίες mobile συσκευών έχουν αναπτύξει μια μεγάλη ποικιλία από mobile συσκευές, όπως laptops, tablet PCs, καθώς και κινητά τηλέφωνα. Ωστόσο, οι συσκευές αυτές έχουν και διαφορετικά χαρακτηριστικά και δυνατότητες, όπως είναι το λειτουργικό σύστημα, η υπολογιστική ισχύ, display και network capacity. Επομένως, αυτή η ανομοιογένεια προκαλεί κάποιες προκλήσεις όσο αναφορά την διαχείριση και διανομή των δεδομένων από του πάροχους mobile υπηρεσιών.

- Κινητικότητα (Mobility) : Η ασύρματη τεχνολογία επιτρέπει στους mobile χρήστες να κινούνται ελεύθερα και ανεξάρτητα από το ένα μέρος στο άλλο. Μια handoff υπηρεσία συμβαίνει όταν ένας χρήστης μετακινηθεί από μία περιοχή κυψέλης του δικτύου σε κάποια άλλη. Επομένως, είναι απαραίτητο να διασφαλιστεί ότι η μεταβίβαση αυτή θα γίνει ομαλά και με διαφάνεια για τους χρήστες.
- Οι συχνές αποσυνδέσεις : Οι mobile χρήστες συχνά αποσυνδέονται από το δίκτυο. Αυτό μπορεί να οφείλεται σε πολλούς και διάφορους λόγους. Κάποιοι λόγοι μπορεί να είναι: αποτυχίες σήματος, μη κάλυψη δικτύου και της εξοικονόμησης ενέργειας. Ο τελευταίος λόγος είναι και προς όφελος μας και προσφέρει και πλεονεκτήματα, δεδομένου ότι η συσκευή σε ενεργή λειτουργία απαιτεί χιλιάδες φορές περισσότερη ενέργεια από ότι όταν είναι σε κατάσταση εξοικονόμησης ενέργειας. Ακόμα μπορεί, η ασύρματη σύνδεση να εξασθενεί περαιτέρω, λόγω της απόστασης του πελάτη από τον σταθμό βάσης ή ο πελάτης να βρίσκεται σε κίνηση.

Τώρα, σύμφωνα με τα παραπάνω χαρακτηριστικά, είναι απαραίτητο να υπάρχει ένας αποτελεσματικός μηχανισμός για την επεξεργασία των mobile data. Αυτό περιλαμβάνει, βεβαίως, την πλήρη κατανόηση των mobile computing τεχνολογιών και το περιβάλλον των mobile data. Επομένως, η γνώση για την επεξεργασία των mobile data, συμπεριλαμβανομένης και της κατανόησης της πολυπλοκότητας των mobile queries, είναι πολύ σημαντική. Τέλος, προκειμένου να μπουν όλα αυτά σε μια σειρά, πρέπει να δούμε πως αποκρίνεται το κινητό στην επεξεργασία δεδομένων, όταν τεθεί σε λειτουργία, συμπεριλαμβανομένης και την χρήση και άλλων mobile εφαρμογών.

1.2 Τεχνολογίες του mobile computing

Η επεξεργασία των mobile data οφείλεται στην εξέλιξη στις ασύρματες τεχνολογίες, καθώς και στα συστήματα εντοπισμού θέσης. Αυτές οι δύο μεγάλες ενότητες θα αναλυθούν εκτενώς παρακάτω αρχίζοντας από τις ασύρματες τεχνολογίες.

1.2.1 Ασύρματες τεχνολογίες (Wireless Technologies)

Όπως είναι γνωστό υπάρχουν ασύρματες τεχνολογίες που χρησιμοποιούνται σε κλειστούς και ανοιχτούς χώρους Τα οποία περιλαμβάνουν : 1) In-Room δίκτυα, 2) Ασύρματο LAN (Wireless LAN), 3) Ασύρματα ευρυζωνικά δίκτυα (Broadband Wireless networks), 4) Ευρείας περιοχής ασύρματα δίκτυα (Wide Area Wireless/Radio networks), 5) Δίκτυα βασισμένα σε δορυφόρο (Satellite-based networks) και 6) Κυψελοειδή δίκτυα (Cellular networks).

1.2.1.1 In-Room δίκτυα

Στα In-Room δίκτυα δίνεται η δυνατότητα στις mobile συσκευές να επικοινωνούν με άλλες συσκευές μέσω ενός ασυρμάτου δικτύου μικρής εμβέλειας. Γενικά, υπάρχουν δύο είδη In-Room δικτύων: σε υπέρυθρης και ράδιο συχνότητας.

Χρησιμοποιώντας την υπέρυθρη συχνότητα, η ασύρματη κάλυψη του δικτύου μπορεί να φτάσει και τα 50 μέτρα, με bandwidth που μπορεί να φτάσει μέχρι και τα 1Mbps. Το πιο συνηθέστερο πρότυπο που χρησιμοποιείται για την τεχνολογία του υπέρυθρου δικτύου είναι το IrDA (**I**nfrared **D**ata **A**ssociation), ένας οργανισμός industry-sponsored, η οποία ιδρύθηκε το 1993 και δημιούργησε πολλά διεθνή πρότυπα (standards) για το υλικά και λογισμικά που χρησιμοποιούν υπέρυθρες συνδέσεις. IrDA είναι μια point-to-point, στενής γωνίας, χρησιμοποιώντας το ad-hoc πρότυπο μετάδοσης και έχει σχεδιαστεί με τέτοιο τρόπο ώστε να λειτουργεί σε απόσταση από 0 έως 1 μέτρο και ταχύτητες 9600bps έως 16Mbps. Στο πρότυπο IrDA 1.1, το μέγιστο μέγεθος το δεδομένων που μπορεί να μεταδοθεί είναι 2048 bytes και η μέγιστη ταχύτητα είναι 4Mbps. Ακόμα, IrDA είναι η ίδια τεχνολογία που χρησιμοποιείται για μια τηλεόρασης μέσω ενός τηλεχειριστηρίου. Σε γενικές γραμμές, προσφέρει την τεχνολογία της ασύρματης συνδεσιμότητας σε συσκευές, που κανονικά θα χρησιμοποιούσαν καλώδια για σύνδεση.

Το άλλο In-Room δίκτυο είναι το ράδιο συχνότητας (radio frequency). Το πιο κοινό πρότυπο που χρησιμοποιεί αυτό το είδος δικτύου είναι το Bluetooth. Το Bluetooth είναι μία ασύρματη τεχνολογία μεγάλης ταχύτητας και χαμηλής ισχύος εκπομπής μικροκύματα. Σχεδιάστηκαν με σκοπό να προσφέρουν σύνδεση σε κινητά τηλέφωνα, laptops, PDAs με άλλες φορητές συσκευές με την βοήθεια ή μη του χρήστη.

Σε αντίθεση με τις υπέρυθρες, το Bluetooth δεν απαιτεί line-of-sight τοποθέτηση των συνδεδεμένων συσκευών. Η τεχνολογία αυτή χρησιμοποιείται έτσι ώστε να υπάρξουν βελτιώσεις σε υπάρχοντες ασύρματες LAN τεχνικές, αλλά έγινε περισσότερο γνωστή για το χαμηλό κόστος και το μικρό μέγεθος. Όταν οι συσκευές έχουν την δυνατότητα σύνδεσης μέσω Bluetooth και έρχονται σε εμβέλεια που “βλέπει” η μία συσκευή την άλλη, γίνεται αμέσως η μεταφορά της διεύθυνσης της συσκευής και δημιουργείται ένα μικρό δίκτυο, χωρίς να εμπλέκεται πουθενά ο χρήστης. Τέλος, η ασύρματη κάλυψη του δικτύου κυμαίνεται από 1 έως και 100 μέτρα και η ταχύτητα μετάδοσης των δεδομένων μπορεί να φτάσει και μέχρι και τα 3Mbps.

1.2.1.2 Wireless LAN

Ένα ασύρματο τοπικό δίκτυο είναι ένα δίκτυο που παρέχει ευρύ ασύρματο εύρος ζώνης και χαμηλή κινητικότητα των πελατών (low mobility clients). Η τεχνολογία αυτή ξεπερνάει το φάσμα που μπορεί να φτάσουν οι υπέρυθρες και το Bluetooth με την βελτίωση ότι η κάλυψη (η διάμετρος) του δικτύου μπορεί να φτάσει και τα 200 μέτρα. Παρέχει χαμηλή κινητικότητα, υψηλού ρυθμού δεδομένα (high-data-rate) μεταφορά δεδομένων, μέσα στην περιορισμένη-κλειστή περιοχή. Ο στόχος των ασυρμάτων δικτύων είναι η γεφύρωση με τα συμβατικά ενσύρματα δίκτυα.

Υπάρχουν πάρα πολλά διαθέσιμα πρότυπα για το WLAN. Το πιο επιτυχές πρότυπο σήμερα και είναι επιφανειακά παρόμοιο με το Ethernet, το IEEE 802.11 πρότυπο. Το πρότυπο αυτό έχει μία σειρά από πρωτόκολλα. Ωστόσο, υπάρχουν τρεις τύποι του IEEE 802.11 που χρησιμοποιούνται ευρέως είναι το IEEE 802.11a, IEEE 802.11b και το IEEE 802.11g. Οι προδιαγραφές του 802.11 είναι ένας μέρος μιας οικογένειας συνεχούς εξελισσόμενων προτύπων ασυρμάτων δικτύων, γνωστή και ως 802.11 οικογένεια. Η συγκεκριμένη προδιαγραφή σύμφωνα με την οποία ένα δίκτυο Wi-Fi λειτουργεί ονομάζεται και “γεύση” του δικτύου.

Το Wi-Fi (**Wireless Fidelity**) είναι ένας όρος για ορισμένα είδη ασύρματων τοπικών δικτύων (WLAN) που χρησιμοποιεί τις προδιαγραφές στην οικογένεια 802.11. Ο όρος Wi-Fi δημιουργήθηκε από μία οργάνωση που ονομάζεται Wi-Fi Alliance (εικόνα 1.2.1.2α), η οποία εποπτεύει τις δοκιμές που πιστοποιούν την

λειτουργικότητα των προϊόντων. Ένα προϊόν που περνάει τα τεστ της συμμαχίας δίνεται η ετικέτα “Wi-Fi certified” (σήμα κατατεθέν, εικόνα 1.2.1.2β,γ). Αρχικά, η πιστοποίηση αυτή εφαρμόζονταν μόνο στα προϊόντα που χρησιμοποιούσαν το πρότυπο 802.11b. Σήμερα εφαρμόζεται και σε προϊόντα που χρησιμοποιούν κάθε πρότυπο 802.11.



Το Wi-Fi έχει βρει αποδοχή σε πολλές επιχειρήσεις, οργανισμούς, σχολεία, ιδρύματα, σπίτια ως μια εναλλακτική λύση σε ένα ενσύρματο LAN. Στις μέρες μας, πολλά ξενοδοχεία, καφετέριες, fast-food και άλλες εγκαταστάσεις προσφέρουν πρόσβαση στο κοινό μέσω του Wi-Fi. Οι θέσεις αυτές είναι και γνωστές και ως hot spots. Πολλοί χρεώνουν την ημερήσια ή ωριαία πληρωμή για την πρόσβαση, αλλά μερικοί το προσφέρουν και δωρεάν. Μια διασυνδεδεμένη περιοχή που αποτελείται από hot spots και σημεία πρόσβασης στο δίκτυο ονομάζεται και hot zone.

Το τρέχον πρότυπο IEEE 802.11 είναι γνωστό για την έλλειψη ενός βιώσιμου μηχανισμού ασφαλείας. Σε ένα μη επαρκώς προστατευμένο Wi-Fi δίκτυο είναι ευάλωτο επειδή μπορεί να έχουν πρόσβαση μη εξουσιοδοτημένα άτομα – χρήστες που χρησιμοποιούν την ελεύθερη πρόσβαση στο διαδίκτυο. Οτιδήποτε είναι συνδεδεμένο σε ένα ασύρματο LAN πρέπει να χρησιμοποιεί δικλίδες ασφαλείας, όπως είναι η προστασία της ιδιωτικής ζώνης WEP (**W**ired **E**quivalent **P**rivacy) πρότυπο κρυπτογράφησης, το πιο πρόσφατο WPA (**W**i-Fi **P**rotected **A**ccess), IPsec (**I**nternet **S**ecurity **P**rotocol) ή ένα εικονικό ιδιωτικό δίκτυο (VPN (**V**irtual **P**ivate **N**etwork)).

Υπάρχει και άλλο πρότυπο ασύρματου LAN το HiperLAN, το οποίο κυρίως χρησιμοποιείται σε Ευρωπαϊκές χώρες. Υπάρχουν δύο προδιαγραφές: HiperLAN/1 και HiperLAN/2. Και τα δύο έχουν εγκριθεί από τον Ευρωπαϊκό Ινστιτούτο Τηλεπικοινωνιακών Προτύπων (ESTI). Τα πρότυπα HiperLAN προσφέρουν ικανότητες και δυνατότητες παρόμοιες με αυτές που προσφέρουν τα πρότυπα 802.11. Το HiperLAN/1 παρέχει επικοινωνίες με ταχύτητα από 20Mbps έως και 5GHz στο φάσμα των ραδιοσυχνοτήτων (RF). Το HiperLAN/2 ορίζεται και ως ένα ευέλικτο πρότυπο Radio LAN που έχει σχεδιαστεί με τέτοιο τρόπο, για να παρέχει πρόσβαση με ταχύτητες έως και 54Mbps σε μία ποικιλία δικτύων, συμπεριλαμβανομένων των 3G δικτύων, ATM και IP-based δίκτυα, καθώς επίσης και τα δίκτυα ιδιωτικής χρήσης ως ασύρματο LAN σύστημα.

Βασικές εφαρμογές περιλαμβάνουν δεδομένα, φωνή και βίντεο, με συγκεκριμένες QoS (**Quality of Services**) παραμέτρους που λαμβάνονται υπόψη. Τα HiperLAN/2 μπορούν να εγκατασταθούν σε γραφεία, αίθουσες διδασκαλίας, σπίτια, εργοστάσια, hot spots περιοχές όπως οι αίθουσες εκδηλώσεων, όπου η μετάδοση ραδιοφωνικών σημάτων είναι μια αποτελεσματική εναλλακτική λύση ή ένα συμπλήρωμα στο είδη ενσύρματης τεχνολογίας. Τέλος, αξίζει να σημειωθεί ότι το HiperLAN/2 έχει αναπτυχθεί σε συνδυασμό με τον σύνδεσμο ιαπωνικών προτύπων, τον σύνδεσμο Association of Radio Industries and Broadcasting.

1.2.1.3 Broadband Wireless networks

Οι ασύρματες ευρυζωνικές συνδέσεις είναι μία ασύρματη τεχνολογία που επιτρέπει την ταυτόχρονη ασύρματη παράδοση φωνής, δεδομένων, και βίντεο που έχει εμφανιστεί πρόσφατα στις μητροπολιτικές περιοχές. Αυτή η ασύρματη τεχνολογία είναι κυρίως διαθέσιμη στις μητροπολιτικές περιοχές, διότι είναι απαραίτητο να υπάρχει σαφής “όραση” μεταξύ του πομπού και των mobile computing συσκευές. Υπάρχουν δύο τύποι αυτής της τεχνολογίας: το LMDS (**Local Multi-point Distribution Service**) και το MMDS (**Multi-channel Multi-point Distribution Service**). Το LMDS χρησιμοποιεί μία υψηλού εύρους ζώνη ασύρματης συχνότητα που κυμαίνεται μεταξύ των 20-31GHz, ενώ το MMDS χρησιμοποιεί ένα χαμηλού εύρους ζώνη ασύρματης συχνότητα στα 2GHz και έχει κάλυψη μέχρι και 50 χιλιόμετρα.

Επιπλέον, ασύρματο ευρυζωνικό δίκτυο, το οποίο είναι χτισμένο με βάση το πρότυπο IEEE 802, είναι το WiMAX. Το WiMAX σχηματίστηκε τον Απρίλιο του 2001, είναι ένας συνασπισμός (coalition) της ασύρματης βιομηχανίας, τα μέλη της οποίας οργανώνονται για την προώθηση του IEEE 802.16, πρότυπα για τα ασύρματα ευρυζωνικής πρόσβασης δίκτυα. Η τεχνολογία του WiMAX 802.16 αναμένεται να επιτρέπουν στις εφαρμογές πολυμέσων με ασύρματη σύνδεση να έχουν κάλυψη μέχρι και 50 χιλιόμετρα. Κλείνοντας, ο κύριος στόχος του WiMAX είναι να προωθηθεί και να πιστοποιηθεί η συμβατότητα και λειτουργικότητα των συσκευών που χρησιμοποιούν την 802.16 προδιαγραφή, και για την ανάπτυξη τέτοιων συσκευών στην αγορά.

1.2.1.4 Wide Area Wireless/Radio networks

Ασύρματο δίκτυο ευρείας περιοχής έχει σχεδιαστεί για να παρέχει μετάδοση δεδομένων και η υποδομή της αποτελείται από σταθμούς βάσης, κέντρα διαχείρισης δικτύων, και διακόπτες για την μετάδοση των δεδομένων. Τα χαρακτηριστικά του ασύρματου δικτύου ευρείας περιοχής είναι η μεγάλη κινητικότητα, και η μεγάλου εύρους και χαμηλού ρυθμού μετάδοση δεδομένων μέσω ψηφιακής επικοινωνίας. Αυτός ο τύπος δικτύων μπορεί να χωριστεί σε δύο κατηγορίες: το δημόσιο και το ιδιωτικό ραδιοφωνικό (radio) δίκτυο.

Η πρώτη κατηγορία αναφέρεται σε ασύρματες επικοινωνίες δεδομένων που παρέχονται στο κοινό από τους φορείς παροχής υπηρεσιών και ο μέσος ρυθμός δεδομένων είναι 4800bps στα 19,2Kbps, ενώ στην άλλη κατηγορία, το ιδιωτικό ασύρματο δίκτυο, παρέχονται από ιδιωτικές εταιρίες για τους δικούς τις σκοπούς.

1.2.1.5 Satellite-Based networks

Το δορυφορικό δίκτυο έχει χρησιμοποιηθεί για αναμετάδοση φωνής, βίντεο ή δεδομένων από την δεκαετία του '60. Τα χαρακτηριστικά αυτού του δικτύου είναι ότι έχει ευρεία κάλυψη φάσματος, ακριβό, αμφίδρομη επικοινωνία και χαμηλής ποιότητας υπηρεσίες φωνής. Γενικότερα, διαθέτει ευρεία κάλυψη, που εκτείνεται από τον ωκεανό μέχρι και τις πιο απομακρυσμένες περιοχές της γης. Παρέχει αμφίδρομη επικοινωνία, ωστόσο, έχει χαμηλή ποιότητα φωνής ή περιορισμένα στοιχεία. Τέλος, είναι δαπανηρή για την παροχή αυτού του είδους δικτύου.

Υπάρχουν τρεις κοινοί όροι που χρησιμοποιούνται σε αυτά τα δορυφορικά δίκτυα και έχουν να κάνουν με την απόσταση, την χωρική σχέση των δορυφόρων από την Γη: οι GEOS (**GEO**stationary **Satellites**), οι MEOS (**Medium Earth Orbit Satellites**), και οι LEOS (**Low Earth Orbit Satellites**). Οι GEOS, MEOS, LEOS βρίσκονται σε υψόμετρα 35.768 χιλιόμετρα, 100.000 χιλιόμετρα και 1.000 χιλιόμετρα αντίστοιχα.

1.2.1.6 Cellular networks

Το κυψελοειδές δίκτυο έχει εξελιχθεί από την πρώτη γενιά (1G) έως και την τέταρτη γενιά (4G).

Η πρώτη γενιά(1G) των κυψελοειδών συστημάτων εμφανίστηκε στις αρχές τις δεκαετίας του 1980 και βασίζεται στην αναλογική τεχνολογία. Τα κύρια χαρακτηριστικά του είναι χαμηλή χωρητικότητα, έλλειψη ασφάλειας, και ακατάλληλο για μη φωνητικές εφαρμογές. Η φωνή μεταδίδεται μέσω FM και ο ρυθμός μετάδοσης των δεδομένων είναι 1,2 – 9,6Kbps.

Μετά την πρώτη γενιά αναλογικών mobile συστημάτων, η δεύτερη γενιά (2G) ψηφιακών mobile συστημάτων εισήχθησαν περίπου το 1991. Παρείχαν μεγαλύτερη χωρητικότητα και χαμηλότερο κόστος στους φορείς εκμετάλλευσης δικτύων, ενώ στους χρήστες προσέφεραν σύντομα μηνύματα και υπηρεσίες χαμηλού ρυθμού δεδομένων, προσθέτοντας και υπηρεσίες ομιλίας. Το 2G σηματοδότησε την άφιξη των ψηφιακών τεχνικών διαμόρφωσης(digital modulation techniques), ώστε να αυξηθεί η χωρητικότητα, να υπάρχει καλύτερη ποιότητα ομιλίας, να ενισχυθούν τα χαρακτηριστικά που έχουν να κάνουν με την ασφάλεια, και να δημιουργηθούν πιο αποτελεσματικά τερματικά. Προσφέρει ρυθμό μεταφοράς δεδομένων που κυμαίνεται από 9 έως 14Kbps. Τα 2G συστήματα είναι το GSM (**G**lobal **S**ystem for **M**obile **C**ommunications), το TDMA (**T**ime **D**ivision **M**ultiple **A**ccess), το PDC (**P**ersonal **D**igital **C**ellular) το CDMA (**C**ode **D**ivision **M**ultiple **A**ccess). Το GSM χρησιμοποιείται στα περισσότερα μέρη του κόσμου εκτός από την Ιαπωνία που χρησιμοποιείται το PDC.

Η 2,5G γενιά είναι μια βελτίωση της 2G. Έχει την ικανότητα να χρησιμοποιεί την packet-switched λύση στο GPRS (**G**eneral **P**acket **R**adio **S**ystem). Το GPRS προσφέρει την δυνατότητα να είσαι πάντα σε απευθείας σύνδεση και να

πληρώνεις μόνο για τα δεδομένα που πραγματικά μεταφέρονται. Ο ρυθμός μετάδοσης των δεδομένων κυμαίνεται έως και 20Kbps και δίδεται η δυνατότητα να προστεθούν ελκυστικές υπηρεσίες.

Η τρίτη γενιά (3G) αναπτύχθηκε το 1992. Η στροφή στο 3G απαιτεί πολλές προσπάθειες. Οι προσπάθειες της ITU έγιναν μέσω του ITM-2000 που οδήγησε σε μία σειρά συστάσεων. Οι οποίες συστάσεις αφορούσαν το εύρος ζώνης του χρήστη, τον πλούτο των προσφερόμενων υπηρεσιών (multimedia υπηρεσίες), και η ευελιξία. Παραδείγματα 3G, το UMTS (**U**niversal **M**obile **T**elecommunications **S**ystem), το CDMA2000. Αυτή η γενιά έχει τρεις κατηγορίες από ρυθμό μετάδοσης δεδομένων:

1. 2,4Mbps σε σταθερές χρήστες (σταθερές θέσεις)
2. 384Kbps των πεζών χρήστες (ταχύτητα : 3 m/hr)
3. 144Kbps έως οχημάτων των χρηστών (ταχύτητα : 60 m/hr)

Η επόμενη γενιά του 3G ασύρματο δίκτυο είναι το 3.5G με 3Mbps/secs ρυθμό μετάδοσης δεδομένων. Η τέταρτη γενιά (4G) δεν έχει επίσημα κυκλοφορήσει ακόμα, αλλά αναμένεται ότι αυτή η γενιά θα υποστηρίξει απαιτήσεις μέχρι και 1Gbps. Στον πίνακα 1.2.1.6 γίνεται σύγκριση μεταξύ αυτών των τεσσάρων γενεών.

<i>1G</i>	<i>2G</i>	<i>3G</i>	<i>4G</i>
Βασική mobility	Βελτιωμένη mobility "roaming"*	Συνεχές και αδιάκοπο roaming	IP-based mobility
Βασικές υπηρεσίες	Ποικίλες υπηρεσίες – ανταλλαγή δεδομένων	Υπηρεσίες concept and model	Εξαιρετικά μεγάλες ταχύτητες δεδομένων
Μη συμβατό	Επικεφαλής σε παγκόσμιο επίπεδο	Παγκόσμια λύση	Τέλεια σύγκλιση τηλεπικοινωνιών και επικοινωνία δεδομένων

Πίνακας 1.2.1.6: Από την 1G στην 4G γενιά

*roaming: Η υπηρεσία περιαγωγής (roaming) προσφέρει την δυνατότητα στους χρήστες να χρησιμοποιούν το κινητό τους τηλέφωνο στο εξωτερικό, σε οποιοδήποτε δίκτυο χώρας, με το οποίο συνεργάζεται ο κάθε παροχέας κινητών υπηρεσιών.

1.2.2 Συστήματα εντοπισμού θέσης

Τα ισχύοντα συστήματα εντοπισμού θέσης καλύπτουν τις τεχνολογίες εντοπισμού θέσης που χρησιμοποιούνται για τα εσωτερικά και υπαίθρια δίκτυα. Αυτά καλύπτουν: 1) Δορυφορικών συστημάτων εντοπισμού θέσης, 2) Κυψελοειδή συστήματα θέσης και 3) τα Indoor συστήματα εντοπισμού θέσης.

1.2.2.1 Satellite Positioning Systems

Το GPS (**G**lobal **P**ositioning **S**ystem) είναι το πιο δημοφιλές δορυφορικό σύστημα εντοπισμού θέσης. Παρέχει δύο βασικά είδη υπηρεσιών: α) την SPS (**S**tandard **P**ositioning **S**ystem) και β) την PPS (**P**recise **P**ositioning **S**ystem). Η SPS είναι μία υπηρεσία εντοπισμού θέσης και χρονισμού με επίκεντρο τον χρήστη της πόλης, ενώ η PPS είναι μια υπηρεσία που χρησιμοποιείται κυρίως για στρατιωτικές εφαρμογές και κύρια χαρακτηριστικά είναι η θέση, η ακρίβεια και η ταχύτητα, η οποία περιορίζεται μόνο σε εξουσιοδοτημένους χρήστες και μόνο. Υπάρχει και ένα άλλο σύστημα εντοπισμού θέσης που ονομάζεται Galileo, το οποίο λειτουργεί από το 2009.

1.2.2.2 Cellular Positioning Systems

Το κυψελοειδές σύστημα εντοπισμού θέσης ενσωματώνει το GPS έτσι ώστε το κυψελοειδές δίκτυο να παρέχει τερματικά με την βοήθεια και την διόρθωση των δορυφόρων. Μερικά παραδείγματα αυτού του συστήματος εντοπισμού θέσης για την δεύτερη γενιά του κυψελοειδούς δικτύου (GSM) είναι το Cell-Id σε συνδυασμό με βελτίωση του χρονισμού (timing advance), το E-OTD (**E**nhanced **O**bserved **T**ime **D**ifference), U-TDoA (**U**plink **T**ime **D**ifference **o**f **A**rrival), και το A-GPS (**A**ssisted **G**PS). Η εισαγωγή του Cell-Id και του A-GPS στα δίκτυα GSM είναι σχετικά απλή, ενώ η E-OTD και U-TDoA περιλαμβάνει σημαντικές αλλαγές και επεκτάσεις.

Ακόμα, μερικά παραδείγματα της 3G γενιάς κυψελοειδούς συστήματος εντοπισμού θέσης είναι οι μέθοδοι Cell-based, το OTDoA-IPDL (**O**bserved **T**ime **D**ifference **o**f **A**rrival with **I**dle **P**eriod **D**ownlink), και το A-GPS.

Το A-GPS είναι μια υβριδική λύση που χρησιμοποιεί πληροφορίες από δορυφόρους και δίκτυο. Επιτρέπει σε μια συσκευή που περιέχει GPS δέκτη να

τοποθετηθεί γρήγορα και με μεγάλη ακρίβεια. Το A-GPS βρίσκεται στον σταθμό βάσης και τροφοδοτεί με πληροφορίες της mobile computing συσκευής. Τέλος, είναι ακριβής, με μεγάλη ευαισθησία, μειώνει τον χρόνο απόκρισης της θέσης, και χρησιμοποιεί λιγότερη ενέργεια στον GPS sever.

	Ακρίβεια (Accuracy)			Συνοχή	Απόδοση
	Αγροτικές	Προαστιακές	Αστικές		
Cell-Id	>10km	2-10km	50-100km	Φτωχή	Καλή
E-TOD και OTDoA	50-150m	50-250m	50-300m	Μέτρια	Μέτρια
U-TDoA	50-120m	40-50m	40-50m	Μέτρια	Μέτρια
A-GPS	10-40m	20-100m	30-150m	Καλή	Καλή

Πίνακας 1.2.2.2: Σύγκριση κυψελοειδών συστημάτων εντοπισμού θέσης

1.2.2.3 Indoor Positioning Systems

Τα εσωτερικά συστήματα εντοπισμού θέσης λειτουργούν μέσα σε ένα εσωτερικό ή τοπικό περιβάλλον, όπως εμπορικά κέντρα ή κτίρια. Υπάρχουν τέσσερις εσωτερικούς χώρους με βάση τα συστήματα εντοπισμού θέσης: (i) WLAN-based, (ii) **R**adio **F**requency **I**dentification (RFID)-based, (iii) υπέρυθρη βάση, και (iv) με βάση το υπερηχογράφημα.

WLAN-based εσωτερικό σύστημα εντοπισμού θέσης είναι το πιο δημοφιλές, και χρησιμοποιεί IEEE802.11 συσκευές.

Η RFID που βασίζεται σε μια αναδυόμενη τεχνολογία η οποία είναι δημοφιλής για χρήση σε εφαρμογές όπως η διαχείριση περιουσιακών στοιχείων, ταυτοποίηση του προϊόντος, και αυτοματισμού εργοστασίων.

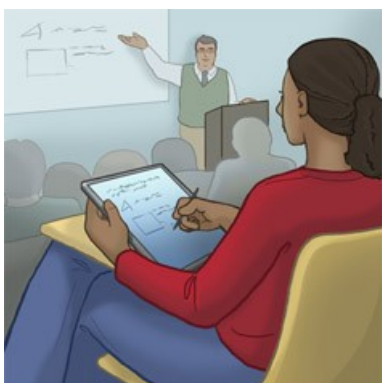
Τα υπέρυθρα συστήματα εντοπισμού θέσης χρησιμοποιούν την τεχνολογία υπέρυθρων. Ένα παράδειγμα είναι το υπέρυθρο σύστημα εντοπισμού θέσης είναι η Xerox ParcTab και το έργο WIPS (**W**ireless **I**ndoor **P**ositioning **S**ystems).

Με βάση το υπερηχογράφημα σύστημα εντοπισμού θέσης χρησιμοποιεί υπερήχους, και συνδυάζει εξαιρετικά ήχους και ραδιοφωνικά δίκτυα, και ένα καλό παράδειγμα είναι το Active Bat.

1.3 Επίλογος

Σε αυτό το κεφάλαιο έγινε μία αναλυτική περιγραφή του mobile computing και των διαφόρων τεχνολογιών που υπάρχουν γύρω από αυτή. Το mobile computing συνδέεται με το υλικό του mobility, δεδομένα και λογισμικά σε εφαρμογές πληροφορικής.

Το mobile computing έχει καταστεί δυνατό με την σύγκλιση των κινητών επικοινωνιών και τεχνολογιών πληροφορικής, οι οποίες περιλαμβάνουν τα κινητά τηλέφωνα, PDAs, WLAN, ασύρματα δίκτυα ευρείας ζώνης και ασύρματα ATMs. Η αυξανόμενη μικρογράφηση όλων των συστημάτων κάνει την χρήση του mobile computing πραγματικότητα. Εν κατακλείδι, το mobile computing – η χρήση φορητών υπολογιστικών με την δυνατότητα ασύρματης δικτύωσης έχει ξεσηκώσει ήδη τον τρόπο που χρησιμοποιούνται οι ηλεκτρονικοί υπολογιστές σήμερα. Στην παρακάτω εικόνα φαίνεται πως χρησιμοποιείται το mobile computing σήμερα.



Taking notes in a class



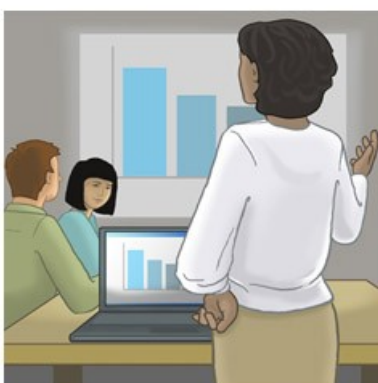
Reading cached e-mail during a flight



Updating patient records between appointments



Browsing the web while reclining on a sofa



Giving a business presentation



Watching a movie in a hotel room

Εικόνα 1.3α: Mobile computing γύρω μας καθημερινά



Εικόνα 1.3β: Mobile computing σε μία αποθήκη

Όπως φαίνεται στην εικόνα 1.3α υπάρχει ένας άνθρωπος που διαβάζει τα email του σε ώρα πώλησης, αυτή η φορητή συσκευή μπορούμε να πούμε ότι είναι ένα smartphone. Στο επόμενο κεφάλαιο θα εισαχθούμε στο τι είναι smartphone και ποιες λειτουργίες υπάρχουν γύρω από αυτά.

Πτυχιακή εργασία του φοιτητή Ελευθέριου Γεωργίου

2

Smartphone και mobile λειτουργικά συστήματα

Το mobile computing είναι παντού στις μέρες μας. Ένας μεγάλος αριθμός από cell phones, PDA's, και άλλες φορητές συσκευές έχουν ξεπεράσει τον αριθμό των υπολογιστών παγκοσμίως. Εκτιμάτε, ότι περίπου 1,8 εκατομμύρια συσκευές υπάρχουν στον κόσμο, οι οποίες διαχειρίζονται πληθώρα διεργασιών και λειτουργιών. Λειτουργίες όπως, επικοινωνία μέσω φωνής, αποστολή /λήψη μηνυμάτων, πλοήγηση στο διαδίκτυο, παιχνίδια και πολλά άλλα.

Καθώς οι τεχνολογίες των ασυρμάτων , όπως είναι τα 3G κυψελοειδή δίκτυα, ασύρματα LAN's, Bluetooth, WiMax και Ultra-Wideband συνεχίζουν να ωριμάζουν, δίνουν σε αυτές τις mobile συσκευές την δύναμη να μπορούμε να βρούμε το mobile computing οπουδήποτε, οποτεδήποτε και σε οποιαδήποτε συσκευή.

Αυτό το κεφάλαιο θα έχει ως αρχή να μας εισάγει στο τι είναι το smartphone. Έπειτα θα δούμε την εξελικτική πορεία του και το πως και πόσο ραγδαία εξελίχθηκε αυτή η συσκευή μέσα στα χρόνια, από τη μέρα της “γέννησής” της μέχρι και σήμερα. Στο τέλος του κεφαλαίου, θα υπάρχει μια εκτενής αναφορά στα λειτουργικά συστήματα που υπάρχουν και χρησιμοποιούν τα smartphones.

2.1 Τι είναι το Smartphone;

Η παγκόσμια βιομηχανία της ασύρματης κινητής τηλεφωνίας κινείται γρηγορότερα από παραδοσιακές, voice-based κυψελοειδούς υπηρεσίες τηλεφώνου που συνδυάζουν υπηρεσίες φωνής και δεδομένων. Έτσι δημιουργήθηκε η ανάγκη έτσι ώστε τα κινητά τηλέφωνα να έχουν γρηγορότερη πρόσβαση σε δεδομένα και να επεκτείνουν τις υπηρεσίες τους, που έχουν σχέση με των υψηλών ταχυτήτων ασύρματα δίκτυα, χρησιμοποιώντας ποικίλες

τεχνολογίες που αναφέρονται στα ασύρματα δίκτυα. Για παράδειγμα, οι 2,5G/3G ασύρματες υπηρεσίες υπάρχουν και χρησιμοποιούνται από ένα αριθμό συνδρομητών που συνεχώς αυξάνεται ραγδαία. Ακόμα ο αριθμός το Wi-Fi hotspots και τα ασύρματα LAN's συνεχίζουν να αυξάνονται ανεξέλεγκτα.

Επομένως το μήνυμα είναι σαφές, η τάση της αγοράς είναι προφανής: Cell Phones, PDA's (**P**ersonal **D**igital **A**ssistant), και φορητές ηλεκτρονικές συσκευές. Συσκευές που θα είναι απλές και φορητές και θα λειτουργούν ως παγκόσμιες προσωπικές επικοινωνιακές και υπολογιστικές πλατφόρμες, γενικά όπως αποκαλείται **smartphone**. Πραγματικά, η αγορά έχει δει μία ραγδαία αύξηση στις πωλήσεις των smartphones σε σχέση με την αργή αύξηση των πωλήσεων PDAs που πωλούνται παγκοσμίως.

Γενικά, ένα smartphone είναι ένα ισχυρό, με πολλές λειτουργίες (multi-function) κινητό τηλέφωνο που ενσωματώνει πληθώρα λειτουργιών ενός PDA, όπως είναι το *προσωπικό καθημερινό πρόγραμμα* (personal schedule), *ημερολόγιο* (calendar) και *βιβλίο διευθύνσεων* (address book), ωστόσο έχουν την δυνατότητα πρόσβασης σε *υπηρεσίες διαδικτύου* και σε εφαρμογές χρησιμοποιώντας είτε ένα πληκτρολόγιο είτε μία ακίδα (stylus). Επί προσθέτως, δίνετε η δυνατότητα στους χρήστες ενώ κάνουν ένα τηλεφώνημα, ταυτόχρονα, να έχουν πρόσβαση στο διαδίκτυο, να ελέγχουν τα email τους, να δημιουργούν έγγραφα, να παίζουν online παιχνίδια, να αναβαθμίζουν το προσωπικό τους καθημερινό πρόγραμμα, ή να έχουν πρόσβαση σε ένα επιχειρηματικό δίκτυο μέσω ενός VPN (**V**irtual **P**rivate **N**etwork). Ωστόσο, η ασύρματη πρόσβαση στο διαδίκτυο είναι δυνατή και μέσω και κυψελοειδή ασύρματα δίκτυα (cellular wireless network), όπως είναι GSM/GPRS, CDMA, CDMA2000, ή WCDMA σύμφωνα με άλλους.

Κλείνοντας, λόγω της συνεχώς αυξανόμενης ζήτησης για δεδομένα, όλο και περισσότερα cell phones και smartphones είναι εξοπλισμένα με ισχυρούς επεξεργαστές για κινητά , όπως είναι ο ARM επεξεργαστής, 64 – 128 MB μνήμη κινητού, από 512MB – 4 GB flash μνήμη και σκληρούς δίσκους από 2 – 32GB πλέον. Μερικά παραδείγματα είναι το Omnia II της Samsung, το iPhone.

2.2 Ιστορική αναδρομή

Το πρώτο smartphone ονομάστηκε *Simon*, το οποίο σχεδιάστηκε από την IBM το 1992 και εμφανίστηκε ως προϊόν “αφηρημένης” έννοιας την ίδια χρονιά στην COMDEX (**C**omputer **D**ealers' **E**xhibition), μία έκθεση της βιομηχανίας των υπολογιστών που έλαβε χώρα στο Las Vegas της Nevada.

Το Simon βγήκε στην αγορά, διαθέσιμο για το κοινό το 1993 και πωλούνταν από την εταιρία BellSouth, σε 190 αμερικάνικες πόλεις και 15 πολιτείες και η αρχική τιμή του ήταν \$899. Πιο συγκεκριμένα εκτός από το να είναι ένα απλό κινητό τηλέφωνο (mobile phone), περιείχε ακόμα *ημερολόγιο* (calendar), *βιβλίο διευθύνσεων* (address book), *παγκόσμιο ρολόι* (world clock), *αριθμομηχανή* (calculator), *σημειωματάριο* (note pad), *υπηρεσία email* , είχε την δυνατότητα να *στέλνει και να λαμβάνει φαξ* και τέλος *παιχνίδια*. Μία ακόμα καινοτομία ήταν ότι το Simon δεν είχε κλασσικά φυσικά κουμπιά έτσι ώστε να υπάρχει η δυνατότητα να κάνει κάποιος την διαχείρισή του μέσω αυτών. Αντιθέτως, οι καταναλωτές – χρήστες είχαν την δυνατότητα μέσω μιας οθόνης αφής (touch-screen) να επιλέξουν αριθμούς κινητών τηλεφώνων με τα δάχτυλά τους ή να δημιουργήσουν τηλεκτυπο ή αλλιώς φαξ (facsimile) και memos μέσω μίας προαιρετικής ακίδας (stylus). Η εισαγωγή του κειμένου (text) γινόταν από μια μοναδική on-screen “predictive” πληκτρολόγιο ή QWERTY πληκτρολόγιο. Τέλος, το Simon περιείχε μία προαιρετική PCMCIA memory card και εμφανίστηκε στην ταινία “*The Net*”.

Το Simon αποτέλεσε το εναρκτήριο λάκτισμα για την δημιουργία ενός νέου και συνεχώς βελτιωμένου προϊόντος από πολλές εταιρίες. Η σειρά Nokia Communicator ήταν από τα πρώτα smartphone της Nokia που άρχισε με το Nokia 900 , που βγήκε στην αγορά το 1996. Αυτό το διακριτικό smartphone, με style που έμοιαζε με αυτό των υπολογιστών palmtop, ήταν αποτέλεσμα μίας ομαδικής προσπάθειας ενός πρώτου επιτυχημένου και ακριβού PDA, από τον Hewlett Packard, με συνδυασμό ενός bestselling κινητού της Nokia την περίοδο εκείνη, και ο συνδυασμός αυτός καθόρισε τις δύο αυτές συσκευές. Και έτσι ξεκίνησε η εξέλιξη των όχι πλέον απλών κινητών τηλεφώνων.

Το Nokia 9210 ήταν το πρώτο της σειράς Communicator που είχε έγχρωμη οθόνη και το πρώτο αληθινό smartphone μιας και είχε ένα ανοικτό λειτουργικό σύστημα, το 9500 Communicator ήταν της ίδιας σειράς της Nokia που περιείχε

κάμερα και Wi-Fi. Επίσης, το 9300 Communicator ήταν ένα ογκώδης κινητό τηλέφωνο και άνοιγε στην μέση και γινόταν ένα πολύ μικρό notebook διαστάσεων οθόνης 640×220, και τέλος το τελευταίο E90 περιέχει και GPS. Η σειρά Nokia Communicator είναι πολύ αξιοπρόσεκτη και για το γεγονός ότι υπήρξε η πιο ακριβή σειρά smartphone μιας κυρίαρχης εταιρίας, κυρίως 20% και μερικές φορές 40% ακριβότερα από τα επόμενα ακριβά smartphone από οποιαδήποτε κύριο κατασκευαστή.

Στο παρασκήνιο υπάρχουν και άλλες εταιρίες που κατασκευάζουν κινητά τηλέφωνα με μία “αφηρημένη” έννοια. Το 1997 η Ericsson έβγαλε στο κοινό, το πρώτο της smartphone, GS88. Στην συνέχεια το υποδέχτηκε το R380 το 2000, το οποίο έχει και οθόνη αφής, και ήταν το πρώτο το οποίο χρησιμοποιούσε Symbian OS. Και τέλος, η Ericsson δημιουργεί το P800 το 2002, το πρώτο smartphone με κάμερα. Το 2001, η Microsoft ανακοινώνει το Windows CE Pocket PC OS, το οποίο αποκαλούνταν και “Microsoft Windows Powered Smartphone 2002”. Και έτσι αρχικά η Microsoft όρισε τα Windows Smartphone προϊόντα με την έλλειψη οθόνης αφής αλλά προσφέροντας μία οθόνη με μειωμένη ανάλυση, η οποία υπήρχε στις ήδη Pocket PC συσκευές.

Συνεχίζοντας, στις αρχές του 2002 η Handspring βγάζει στην αγορά το Palm OS Theo smartphone, το οποίο χρησιμοποιεί ένας πλήρες πληκτρολόγιο το οποίο συνδυάζει wireless web browsing, email, ημερολόγιο, και οργάνωση επαφών μέσω εφαρμογών κινητού τρίτων, οι οποίες μπορούν να “κατεβούν” ή να συγχρονιστούν από ή μέσω ενός ηλεκτρονικού υπολογιστή. Ακόμα κατά το ίδιο έτος η RIM παραθέτει στην αγορά το BlackBerry, ήταν το πρώτο smartphone βελτιστοποιημένο για ασύρματη χρήση email και είχε κατορθώσει να έχει 32 εκατομμύρια συνδρομητές μέχρι τον Δεκέμβρη του 2009. Το 2007, η Nokia λανσάρει στην αγορά το Nokia N95 το οποίο ενσωμάτωνε ένα μεγάλο εύρος από χαρακτηριστικά όπως: GPS, μία κάμερα 5 megapixel με αυτόματη εστίαση και LED flash, 3G, Wi-Fi και TV-out. Στα επόμενα χρόνια αυτά τα χαρακτηριστικά θα αποτελέσουν standard για κάθε νέο smartphone.

Λίγο αργότερα τον ίδιο χρόνο, μπήκε στην κατασκευή smartphone και η εταιρία Apple Inc., η οποία εμφάνισε το δικό της πρώτο smartphone, με το όνομα *iPhone*. Ήταν εξαιρετικά ακριβό, κόστιζε περίπου \$500. Ήταν, επίσης, από τα

πρώτα smartphone που όλες οι λειτουργίες και ο χειρισμός γινόταν αποκλειστικά από μία οθόνη αφής, τα άλλα ήταν το LG Prada και το HTC Touch. Υπήρξε το πρώτο κινητό τηλέφωνο το οποίο χρησιμοποιούσε multi-touch interface, όμως με την άφιξη του υπήρξε μία αμφισβήτηση αν πραγματικά ήταν smartphone ή όχι, καθώς η πρώτη γενιά smartphone είχε έλλειψη στην χρήση επισήμων εφαρμογές τρίτων. Έπειτα, δημιουργήθηκε μια διαδικασία που είχε το όνομα jailbreaking και έδωσε σε χρήστες να χρησιμοποιούν εφαρμογές τρίτων οι οποίες όμως δεν είναι επίσημες. Τέλος, το iPhone δεν υποστήριζε 3G μιας και υπήρχε έλλειψη 3G σήματος στις Ηνωμένες Πολιτείες εκείνη την περίοδο.

Έπειτα, το 2008 ένα καινούριο λογισμικό αποκαλούμενο Android OS έρχεται στο προσκήνιο. Το Android είναι μία πλατφόρμα ανοικτού κώδικα (open source) που υποστηρίζεται από την Google, μαζί με κύριους hardware και software developers, αναφέροντας μερικούς όπως είναι η Intel, HTC, ARM, Motorola και το eBay, από την Open Handset Alliance. Το πρώτο κινητό που χρησιμοποιήθηκε το Android OS ήταν το HTC Dream. Το software suite που περιείχε το κινητό ήταν αποκλειστικά εφαρμογές που ανήκουν στην Google όπως είναι το Maps, Calendar, Gmail και το Chrome Lite πλήρη HTML web browser της Google. Τέλος, οι εφαρμογές τρίτων ήταν διαθέσιμες μέσω του Android Market, όπου υπάρχουν και ελεύθερες και μέσω πληρωμής εφαρμογές.

Τον Ιούλιο του 2008, η Apple εισήγαγε την δεύτερη γενιά του iPhone, το οποίο είχε μία πιο μειωμένη τιμή και υποστήριζε 3G. Επίσης, δημιούργησε και το Apple Store όπου υπάρχουν ελεύθερες και μέσω πληρωμής εφαρμογές. Το Apple Store μπορεί να μεταφέρει εφαρμογές smartphone που έχουν προγραμματιστεί από τρίτους κατευθείαν στο iPhone ή στο iPod μέσω Wi-Fi ή μέσω κυψελοειδούς δικτύου (cellular network) χωρίς να χρειάζεται να μεσολαβήσει ηλεκτρονικός υπολογιστής για να κατεβούν οι εφαρμογές αυτές. Ακόμα, το Apple Store είχε και έχει μεγάλη επιτυχία και μέχρι τον Απρίλιο του 2010 φιλοξενούσε περισσότερες από 185.000 εφαρμογές και μέχρι τον Ιανουάριο του 2010 είχαν κατεβεί περισσότερες από 3 εκατομμύρια εφαρμογές.

Τέλος, στις άλλες πλατφόρμες υπάρχει η δυνατότητα να κατεβάσει ο καθένας εφαρμογές από οποιοδήποτε website. Ωστόσο, μερικές εταιρίες έχουν λανσάρει πρόσφατα τα δικά τους app store. Η RIM λάνσαρε το δικό της app store και το

ονόμασε BlackBerry App World, τον Απρίλιο του 2009. Η Nokia λάνσαρε το δικό της Oni Store, τον Μάιο 2009. Η Palm λάνσαρε το Palm App Catalog, τον Ιούνιο του 2009 και η Microsoft λάνσαρε το Windows Marketplace for Mobile, τον Οκτώβριο του 2009.




Τελευταία, τον Ιανουάριο του 2010, η Google λάνσαρε το Nexus One που χρησιμοποιεί το Android OS. Όμως, αν και το Android OS έχει multi-touch δυνατότητες, είχε αφαιρέσει αυτό χαρακτηριστικό από Nexus One, αλλά το επανέφερε μέσω μίας αναβάθμισης firmware τις 2 Φεβρουαρίου του 2010.

2.2.1 Ιστορική αναδρομή με εικόνες

<i>Χρονολογίες</i>	1992	1996	1997	2000
<i>Κινητά</i>				
Simon				
Nokia 9000				
Ericsson GS88				
Nokia 9210				









Πίνακας 2.2.1α: Ιστορική αναδρομή

Πτυχιακή εργασία του φοιτητή Ελευθέριου Γεωργίου

Χρονολογίες Κινητά	2000	2002	2004	2005
Ericsson R380				
Sony Ericsson P800				
Palm Theo				
BlackBerry				
Nokia 9500				
Nokia 9300				

Πίνακας 2.2.1β: Ιστορική αναδρομή (Συνέχεια)

Πτυχιακή εργασία του φοιτητή Ελευθέριου Γεωργίου

<i>Χρονολογίες Κινητά</i>	<i>2007</i>	<i>2008</i>	<i>2010</i>
Nokia E90			
Nokia N95			
iPhone			
LG Prada (KE850)			
HTC Touch			
HTC Dream			
iPhone 3G			
Nexus One			

Πίνακας 2.2.1ν: Ιστορική αναδρομή (Συνέχεια)

2.3 Mobile λειτουργικά συστήματα (Operating Systems OS)

Υπάρχουν πολλά λειτουργικά συστήματα που μπορεί να βρεθούν στα smartphones. Συμπεριλαμβανομένων των Android, BlackBerry OS, iOS, Symbian OS (θα αναφερθούμε στο κεφάλαιο 3), webOS, bada, και Windows Mobile. Το λειτουργικό σύστημα του Android χτίστηκε πάνω σε Linux, ενώ το iOS προέρχεται από τα λειτουργικά συστήματα BSD (Berkeley Software Distribution) και NeXTSTEP, τα οποία όλα σχετίζονται με το Unix. Όμως παρακάτω θα γίνει εκτενείς αναφορά για τα περισσότερα αναφερθέντα mobile λειτουργικά συστήματα.



2.3.1 Android OS

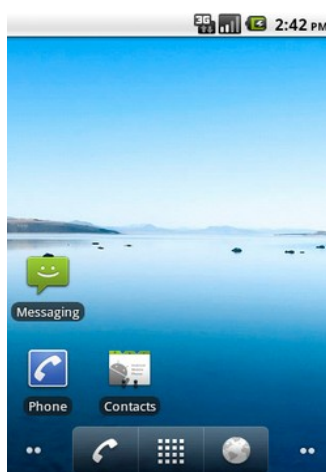
Το Android OS είναι ένα mobile λειτουργικό σύστημα το οποίο αναπτύχθηκε από την Google και βασίζεται πάνω στο Linux kernel και σε GNU (Unix-Like OS) λογισμικό. Αρχικά αναπτύχθηκε από την εταιρία Android Inc, μια εταιρία που αγοράστηκε από την Google, και έπειτα ενσωματώθηκε στην Open Handset Alliance.

Το Android έχει μία τεράστια κοινότητα από προγραμματιστές που γράφουν εφαρμογές “apps”, που επεκτείνει την λειτουργία των συσκευών. Υπάρχουν περίπου 70.000 apps και εκτιμάτε ότι θα ξεπεραστεί ο αριθμός των 100.000 που έχουν υποβληθεί. Οι προγραμματιστές μπορούν να γράψουν εφαρμογές χρησιμοποιώντας την Java, ως γλώσσα προγραμματισμού, και δίνει την δυνατότητα να ελέγχουν τις συσκευές μέσω Java libraries που είναι Google-developed. Ωστόσο, υπάρχει και μια υλοποίηση από την Ruby, η οποία είναι υπό εξέλιξη, και επιτρέπει ολόκληρες Android εφαρμογές να γραφτούν σε Ruby.

Η παρουσίαση του Android OS και η διανομή του έγινε μαζί με την ίδρυση της Open Handset Alliance στις 5 Νοεμβρίου 2007, μιας συμμαχίας μεταξύ 78 hardware, software και εταιρίες τηλεπικοινωνιών που είναι αφιερωμένη στην προώθηση open standards για τις mobile συσκευές. Η Google έκδοσε τον περισσότερο Android code κάτω από την Apache Licence, μια free software και open licence.

Το Android OS software stack (ένα σεντ από προγράμματα που συνδυάζονται

όλα μαζί για να επιφέρουν ένα αποτέλεσμα), αποτελείται από Java εφαρμογές που τρέχουν βασισμένες σε ένα Java-based object oriented application framework πάνω από Java core libraries που τρέχουν σε μια εικονική μηχανή Dalvik virtual machine, που έχει δοθεί ο χαρακτηρισμός JIT(**J**ust-**I**n-**T**ime) υλοποίηση. Ωστόσο, υπάρχουν και libraries που είναι γραμμένες σε C και αυτές περιλαμβάνουν: surface manager, OpenCore media framework, SQLite database, OpenGL ES 2.0 3D graphics API, WebKit layout engine, SGL graphics engine, SSL (**S**ecure **S**ocket **L**ayer), και Bionic libc (GNU C library). Τέλος, το Android OS περιέχει 12 εκατομμύρια γραμμές κώδικα, από αυτές περίπου, οι 3 εκατομμύρια γραμμές είναι XML (**E**xtensible **M**arkup **L**anguage), οι 2,8 εκατομμύρια γραμμές είναι γραμμένες σε C, 2,1 εκατομμύρια γραμμές σε Java, και 1,75 εκατομμύρια γραμμές είναι, τέλος, γραμμένες σε C++.



Εικόνα 2.3.1α:
Android 2.2



Εικόνα 2.3.1β:
Android Emulator



2.3.2 BlackBerry OS

BlackBerry OS είναι ιδιόκτητο mobile λειτουργικό σύστημα, το οποίο αναπτύχθηκε από την RIM (**R**esearch **I**n **M**otion), για την δική της BlackBerry Series, μια σειρά από smartphones συσκευές.

Αυτό το λειτουργικό σύστημα παρέχει multitasking (πολλές διεργασίες μαζί) και υποστηρίζει συγκεκριμένες συσκευές εισόδου (input devices), τις οποίες έχει υιοθετήσει η RIM και τις χρησιμοποιεί στα smartphones. Όπως είναι το trackwheel,

trackball (εικόνα 2.3.2α) και τώρα πρόσφατα το trackpad και το touch-screen(εικόνα 2.3.2β).



Εικόνα 2.3.2α:
BlackBerry trackwheel ή trackball



Εικόνα 2.3.2β: BlackBerry
trackpad και touch-screen

Η πλατφόρμα του BlackBerry έγινε γνωστή για την υποστήριξή του στο εταιρικό email μέσω του MIDP 1.0 (**M**obile **I**nformation **D**evice **P**rofile) και τώρα τελευταία μέσω του MIDP 2.0. Το MIDP 2.0 επιτρέπει τον πλήρη συγχρονισμό και ενεργοποίηση ασύρματα με το Windows Exchange, Lotus Domino, ή Novell GroupWise email, ημερολόγιο, tasks, σημειώσεις, και επαφές, όταν χρησιμοποιείτε μαζί με τον BlackBerry Enterprise Server. Επιπλέον, το λειτουργικό σύστημα υποστηρίζει και WAP 1.2 (**W**ireless **A**pplication **P**rotocol).

Τώρα, όσο αναφορά την αναβάθμιση του λειτουργικού συστήματος η RIM προφέρει μία υπηρεσία, την BlackBerry OTASL (**O**n **T**he **A**ir **S**oftware **L**oading), όπου η πιθανή λήψη κάποιων αναβαθμίσεων γίνεται αυτόματα μέσω της ασύρματης μεταφοράς που προσφέρει αυτή η υπηρεσία.

Τέλος, οι εφαρμογές από τρίτους μπορούν να γραφούν χρησιμοποιώντας το BlackBerry API (**A**pplication **P**rogramming **I**nterface) σε C++, όμως για λειτουργήσουν κανονικά πρέπει να φέρουν ψηφιακή υπογραφή (digitally signed). Κλείνοντας, έγινε γνωστό τον Απρίλιο του 2010 από την RIM ότι το φθινόπωρο του 2010 θα είναι έτοιμο το νέο βελτιωμένο λειτουργικό σύστημα BlackBerry OS 6.



2.3.3 iOS

Το iOS είναι το λειτουργικό σύστημα της Apple. Αυτό το λειτουργικό σύστημα δεν αφορά μόνο το iPhone αλλά έχει επεκταθεί έτσι ώστε να βρίσκει εφαρμογή στα iPod Touch και iPad. Η Apple έχει αυστηρούς κανόνες, και δεν επιτρέπει το λειτουργικό της σύστημα να βρίσκεται και να “τρέχει” σε υλικό (hardware) τρίτων.

Η βασική ιδέα του user interface του iOS βασίζεται στην έννοια του άμεσου χειρισμού (direct manipulation) και χρησιμοποιώντας multi-touch χειρονομίες. Τα interface control elements αποτελούνται από μπάρες ολίσθησης (sliders), διακόπτες (switchers) και κουμπιά (buttons). Η απόκριση στις εντολές εισόδου του χρήστη είναι άμεσες και προσφέρει ένα ευχάριστο (fluid) interface. Η αλληλεπίδραση με το OS περιλαμβάνει χειρονομίες όπως, το χτύπημα(swiping), την πληκτρολόγηση (tapping), το pinching και το reverse pinching. Τα δύο τελευταία αναφέρονται σε κινήσεις- χειρονομίες που γίνονται με πολλαπλά δάχτυλα, από τον χρήστη, όπως για παράδειγμα το zoom-in zoom-out σε μια εικόνα.

Ακόμα υπάρχουν εσωτερικοί αισθητήρες ή αλλιώς καλύτερα επιταχυνσίμετρα (accelerometer), τα οποία χρησιμοποιούνται από πολλές εφαρμογές, έτσι ώστε να καταλαβαίνουν πότε κουνιέται η συσκευή ή πότε γυρίζει μέσα στις τρεις διαστάσεις του. Για παράδειγμα, μία κίνηση undo, ή μια κίνηση για να αλλάξουμε κάποιο τραγούδι που ακούμε, ωστόσο υπάρχει και η πιο απλή αλλαγή που βασίζεται στην τριών διαστάσεων κίνηση, που είναι η αλλαγή από portrait σε landscape mode (εικόνα 2.3.3α).



Εικόνα 2.3.3α: Portrait και Landscape mode

Τώρα, το iOS προέρχεται από το Mac OS X, το οποίο το μοιράζεται με το ίδρυμα Darwin, και επομένως είναι και αυτό ένα Unix-Like λειτουργικό σύστημα από την φύση του. Το iOS αποτελείται από τέσσερα “αφηρημένα επίπεδα(layers), τα οποία είναι τα εξής:

1. το Core OS layer : το επίπεδο αυτό παρέχει του χαμηλού επιπέδου (low-level) χαρακτηριστικά όπου οι περισσότερες τεχνολογίες του iOS είναι χτισμένες πάνω σε αυτό.
2. το Core Services layer : το επίπεδο αυτό παρέχει τις θεμελιώδεις υπηρεσίες συστημάτων που χρειάζονται και χρησιμοποιούν όλες οι εφαρμογές που δουλεύουν με το iOS.
3. το Media layer : το επίπεδο αυτό παρέχει όλες τις τεχνολογίες που σχετίζονται με τα γραφικά (graphics), τον ήχο (audio) και το βίντεο, έτσι ώστε να δημιουργείτε μια καλύτερη εμπειρία πολυμέσων. Επιπλέον, αυτές οι τεχνολογίες έχουν σχεδιαστεί με τέτοιο τρόπο έτσι ώστε να είναι πιο εύκολος σχεδιασμός εφαρμογών, που να δείχνουν και να ακούγονται ωραία. Αυτό το υψηλού επιπέδου (high-level) framework του iOS δίνει την δυνατότητα στους προγραμματιστές να φτιάχνουν βελτιωμένα γραφικά (advanced graphics) και κινούμενες εικόνες (animations) πάρα πολύ γρήγορα.
4. το Cocoa Touch layer : το επίπεδο αυτό παρέχει όλα αυτά τα εργαλεία έτσι ώστε να χτιστεί μία iOS εφαρμογή.

Περισσότερες πληροφορίες για αυτά τα επίπεδα, βρίσκονται στην παρακάτω website:http://developer.apple.com/library/ios/#documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/Introduction/Introduction.html#//apple_ref/doc/uid/TP40007898-CH1-SW1.

Το iOS χρησιμοποιεί κατά προσέγγιση τα 500MB της μνήμης του κινητού, αλλά αυτό ποικίλει από κινητό σε κινητό. Τέλος, οι προγραμματιστές έχουν την δυνατότητα προγραμματίσουν εφαρμογές, που βασίζονται σε αυτό το λειτουργικό σύστημα, που είναι γραμμένες σε C++, C, και Objective-C, και χρησιμοποιώντας το αντίστοιχο SDK (**S**oftware **D**evelopment **K**it), που έχει την ιδιαιτερότητα να τρέχει μόνο σε Mac OS.

2.3.4 Palm webOS

Το Palm webOS είναι ένα mobile λειτουργικό σύστημα που ανήκει στην εταιρία Palm και τρέχει σε Linux kernel. Το Palm Pre (εικόνα 2.3.4α) ήταν το πρώτο smartphone που χρησιμοποίησε ο λειτουργικό σύστημα το webOS, τα οποία και τα δύο προβλήθηκαν για πρώτη φορά στον κοινό στις 8 Ιανουαρίου του 2009, στο Consumer Electronics Show που έλαβε χώρα στο Las Vegas. Ωστόσο, η διάθεσή τους στο κοινό και την αγορά έγινε στις 6 Ιουνίου του 2009. Η δεύτερη συσκευή που χρησιμοποίησε αυτό το λειτουργικό ήταν το Palm Pixi (εικόνα 2.3.4β) , το οποίο διατέθηκε στην αγορά στις 15 Νοεμβρίου του 2009.

Το webOS χαρακτηρίζεται από την δυνατότητα που δίνει στην απευθείας σύνδεση με κοινωνικά δίκτυα (social networks) και την ολοκλήρωση του web 2.0. Όσο αφορά τον προγραμματισμό εφαρμογών αυτό μπορεί να γίνει μέσω των γλωσσών προγραμματισμού C και C++.



Εικόνα 2.3.4α: Palm Pre



Εικόνα 2.3.4β: Palm Pixi



2.3.5 Windows Mobile

Το Windows Mobile είναι ένα λειτουργικό σύστημα το οποίο έχει αναπτυχθεί από την Microsoft και χρησιμοποιείται ευρέως σε smartphones και άλλες mobile συσκευές. Η τελευταία τωρινή έκδοση ονομάζεται “Windows Phone 6.5”, και βασίζεται στο Windows CE 5.2 kernel. Χαρακτηρίζεται από βασικές εφαρμογές που έχουν δημιουργηθεί μέσω του Microsoft Windows API, και έχουν σχεδιαστεί με τέτοιο τρόπο έτσι ώστε να είναι φιλίες με τις desktop εφαρμογές των Windows.

Όσο αναφορά τον προγραμματισμό εφαρμογών, οι προγραμματιστές έχουν την δυνατότητα να προγραμματίζουν εφαρμογές, βασισμένοι στην γλώσσα προγραμματισμού C++. Η Microsoft έδωσε στο κοινό το δικό της SDK (**S**oftware **D**evelopment **K**it), το οποίο τρέχει με την βοήθεια του Visual Studio, δηλαδή επί της ουσίας οι εφαρμογές είναι γραμμένες σε Visual C++. Ωστόσο, υπάρχουν και εναλλακτικά άλλα περιβάλλοντα για να αναπτυχθούν οι εφαρμογές από τους προγραμματιστές όπως είναι τα : Lazarus, Lexico, NS Basic και Basic4ppc.

Κλείνοντας, η Microsoft στις 5 Ιουνίου του 2009 έδωσε την δυνατότητα στους προγραμματιστές να αναρτήσουν τις εφαρμογές τους, δηλαδή οι λεγόμενες εφαρμογές τρίτων, όπου μπορεί να είναι είτε ελεύθερες είτε με πληρωμή εφαρμογές, στο Windows Marketplace.



*Εικόνα 2.3.5α:
Windows Mobile 6.5.3*



2.3.6 Bada

Το Bada είναι ένα λειτουργικό σύστημα το οποίο αναπτύχθηκε από την Samsung Electronics. Έχει σχεδιαστεί με τέτοιο τρόπο έτσι ώστε να καλύπτει και τις ανάγκες των smartphones αλλά και άλλων mobile συσκευών. Η Samsung ισχυρίζεται ότι το συγκεκριμένο λειτουργικό σύστημα, θα είναι η δυνατή εφικτή πλατφόρμα για να γίνει η μετατροπή απλών mobile συσκευών σε smartphones. Το όνομα “bada” προέρχεται από μια κορεάτικη λέξη που σημαίνει ωκεανός ή θάλασσα.

Το πρώτο κινητό που χρησιμοποίησε ποτέ το λειτουργικό αυτό ήταν το Samsung Wave S8500 (εικόνα 2.3.6α), το οποίο βγήκε στην αγορά στις 1 Ιουνίου του 2010, και η εταιρία πούλησε πάνω από 1 εκατομμύριο συσκευές σε λίγο παραπάνω από 1 μήνα.

Ο προγραμματισμός των εφαρμογών βασίζεται στην C++. Η Samsung έδωσε το δικό της SDK (**S**oftware **D**evelopment **K**it), αλλά σε δοκιμαστική (beta) έκδοση, μόλις τον Μάιο του 2010. Για να προσελκύσει προγραμματιστές, οργάνωσε ένα διαγωνισμό τον bada Developer Challenge με έπαθλο τον πόσο των \$2,700,000. Τέλος, τον Αύγουστο του 2010 έδωσε στο κοινό μια ολοκληρωμένη μορφή του SDK την έκδοση 1.0



Εικόνα 2.3.6α: Samsung Wave S8500

2.4 Επίλογος

Σε αυτό το κεφάλαιο έγινε μία ιστορική αναδρομή για τα smartphone και πως από το πρώτο Simon smartphone φτάσαμε στα σημερινά “τεχνολογικά θαύματα”, στις φορητές συσκευές – smartphone με δυνατότητες ενός υπολογιστή και σε λειτουργίες αλλά και σε υλικό πλέον. Πέρα από την ιστορική αναδρομή έγινε και μία εκτενής ανάλυση όλων των mobile OS που υπάρχουν σήμερα στις αγορές και με τα διάφορα χαρακτηριστικά αυτών. Ο παρακάτω πίνακας φαίνεται πως μοιράζονται τα διάφορα mobile OS στην παγκόσμια αγορά smartphone.

Χρόνος	Symbian	Android	RIM	iOS	Microsoft	Άλλα OS
2010	37.6%	22.7%	16%	15.7%	4.2%	3.8%
2009	46.9%	3.9%	19.9%	14.4%	8.7%	6.1%
2008	52.4%	0.5%	16.6%	8.2%	11.8%	10.5%
2007	63.5%	--	9.6%	2.7%	12%	12.1%

Πίνακας 2.4α Μοίρασμα των mobile OS στην αγορά (πηγή Gartner)

Στους παρακάτω πίνακες υπάρχει μία συνοπτική περιγραφή των mobile OS με τα κάποια χαρακτηριστικά αυτών, που σχετίζονται με την πληροφορική κυρίως.

Χαρακτηριστικά	iOS	Android	webOS	Windows Mobile	Windows Phone
Εταιρία	Apple	Open Handset Alliance (Google)	HP/Palm	Microsoft	Microsoft
Έκδοση	4.3	2.3.3	1.4.5	6.5.3	7.0.7390.0
Οικογένεια OS	Mac OS X/Unix-like	Linux	Linux	Windows CE 5.2	Windows CE 7
CPU αρχιτεκτονική	ARM	ARM, MIPS, Power Architecture, x86	ARM	ARM	ARM
Προγ/σμός με	C, C++, Objective-c	C, C++, Java	C	C++	.NET (Silverlight/XNA)

Πίνακας 2.4β Σύγκριση mobile OS (1)

Χαρακτηριστικά	Blackberry OS	Symbian	Maemo	MeeGo	Bada
Εταιρία	RIM	Symbian Foyndation	Nokia	Linux Foundation	Samsung
Έκδοση	6.0.0	9.5	5.0	1.1.2	1.2
Οικογένεια OS	Mobile OS	Mobile OS	Linux	Linux	Linux
CPU αρχιτεκτονική	ARM	ARM	ARM	ARM, x86	ARM
Προγ/σμός με	Java	C++	C/C++	C++	C++

Πίνακας 2.4β Σύγκριση mobile OS (2)

Μετά από αυτή την σύγκριση των mobile OS και το μοίρασμα αυτών μέσα στην παγκόσμια αγορά μπορούμε να πούμε ότι το Symbian OS είναι ένα πολύ δυνατό και γεμάτο υποσχέσεις λειτουργικό σύστημα για smartphones. Στο επόμενο κεφάλαιο θα εισαχθούμε και θα αναλύσουμε λεπτομέρειες του Symbian OS και θα αναλύσουμε ένα περιβάλλον ανάπτυξης λογισμικού, το περιβάλλον το Qt.

3






Symbian OS και Qt




Σε αυτό το κεφάλαιο θα γίνει μία σύντομη εισαγωγή στο “πάντρεμα” των Qt και του Symbian σε μια πλατφόρμα. Θα σημειωθεί επίσης γιατί θα πρέπει να επιλέγουν Qt για Symbian. Το κεφάλαιο αυτό εξηγεί επίσης το σκεπτικό γιατί Qt για Symbian είναι μια ενδιαφέρουσα λύση για ένα ευρύ φάσμα των κινητών προγραμματιστές - από αρχάριους μέχρι και τους πιο ειδικούς - με την πλήρη λειτουργικότητα της πλατφόρμας Symbian και απολαμβάνοντας τον εύκολο προγραμματισμό μέσω του Qt σε συνδυασμό με τις δυνατότητες cross-platform της.

3.1 Symbian OS

Το Symbian OS είναι μία προηγμένη mobile development πλατφόρμα ανάπτυξης λογισμικού που χρησιμοποιείται ευρέως στα κινητά τηλέφωνα. Στηριζόμενο στην C++, ένα μικροπυρήνα (mikrokernel), και εκτεταμένη χρήση της σε αντικειμενοστραφούς σχεδίασης εφαρμογών, έδωσε την δυνατότητα στην ανάπτυξη μίας σειράς συσκευών και έτσι γεννήθηκε το Symbian, το πιο ευρέως χρησιμοποιημένη πλατφόρμα στα smartphones σήμερα.

Η απαρχή του Symbian OS μπορεί να αναχθεί στην δεκαετία του 1990 και στις Psion κινητές συσκευές. Όντας η εξέλιξη της τεχνολογίας, εκείνη την εποχή, οι συσκευές Psion ήταν μέτριες ως προς τις επιδόσεις και τις δυνατότητες με τα σημερινά δεδομένα, και την ικανότητα των Symbian OS για να στηρίξουν την εισαγωγή νέων χαρακτηριστικών που περιμένει κανείς από τα κινητά τηλέφωνα, μπορεί όμως να ληφθεί ως ένα αφιέρωμα στους αρχικούς σχεδιαστές. Το δρόμος από το λειτουργικό σύστημα των συσκευών Psion στην πλατφόρμα Symbian έχει περάσει από διάφορες φάσεις, από τις ρίζες της στο Psion και τώρα στην νεοδημιουργηθείσα Symbian Foundation.

 <p><i>Organiser II</i></p>	<p>Launched: 1986 Memory: 8K, 16K, 32K or 64K Screen: 480 x 160 pixels Variants: CM, XP, LZ</p>
 <p><i>Series 3</i></p>	<p>Launched: 1991 Memory: 128K or 256K Screen: 240 x 80 pixels</p>
 <p><i>Series 3a, 3c, 3mx</i></p>	<p>Launched: 1993 Memory: 128K, 256K, 512K, 1MB or 2MB Screen: 480 x 160 pixels</p>
 <p><i>Siena</i></p>	<p>Launched: 1996 Memory: 512K or 1MB Screen: 240 x 160 pixels</p>
 <p><i>Series 5</i></p>	<p>Launched: 1997 Memory: 4MB or 8MB Screen: 640 x 240 pixels</p>
 <p><i>Series 7, netBook</i></p>	<p>Launched: 1999 Memory: 16MB (Series 7), 32MB (netBook). Screen: 640 x 480 pixels. Colour</p>

 <p>Series 5mx</p>	<p>Launched: 1999 Memory: 16MB Screen: 640 x 240 pixels Variants: Psion 5mx Pro , Ericsson Mc218</p>
 <p>Revo, Revo Plus</p>	<p>Launched: 1999 Memory: 8MB (Revo), 16MB (Revo Plus) Screen: 480 x 160 pixels Variants: Diamond Mako</p>
 <p>Psion PLC</p>	<p>Launched: 2000 Memory: Screen: Variants:</p>

Πίνακας 3.1: Psion Συσκευές

Η προέλευση του Symbian OS βρίσκονται στα σχέδια του Psion PLC, και περιλαμβάνει κυρίως Sibo, ή EPOC16, η οποία βασιζόταν στην C, και αργότερα EPOC32, η οποία μετονομάστηκε ως Symbian στη μετέπειτα ζωή του. Η εταιρεία αναπτύσσει επίσης SDKs (**S**oftware **D**evelopment **K**its) για να καταστεί δυνατή τρίτους η ανάπτυξη εφαρμογών, κάτι το οποίο αποδείχθηκε ότι ήταν σημαντική κατά τα επόμενα βήματα για την τρέχουσα πλατφόρμα για smartphones.

Η εταιρεία Symbian, η αρχική έδρα του Symbian OS, όπως το γνωρίζουμε σήμερα, ιδρύθηκε το 1998 για να αναπτύξει μια φορητή πλατφόρμα για τα smartphones. Αρχικά που ανήκουν σε μια σειρά μεγάλων εταιρειών κινητής συσκευής - Ericsson, η Matsushita, η Motorola, Nokia και Psion - ο στόχος ήταν να χορηγήσει άδεια Symbian OS με τους κατασκευαστές των προηγμένων 2,5 G και 3G κινητών τηλεφώνων, με μεγάλο μέρος της πνευματικής ιδιοκτησίας που προέρχονται από Psion PLC . Ο τρόπος ανάπτυξης ήταν τέτοιος που η εταιρεία Symbian επικεντρώθηκε στο βασικό λειτουργικό σύστημα και στο κυρίως framework, λαμβάνοντας υπόψη ότι οι δικαιούχοι ανάπτυξης εφαρμογών διαφέρουν από device-specific παραλλαγές, συμπεριλαμβανομένων των Symbian phone-like συσκευών, τουλάχιστον αρχικά, και για UIQ (**U**ser **I**nterface **Q**uartz) επικοινωνίας βασισμένες σε μια οθόνη αφής.

Η δημιουργία του Symbian Foundation ανακοινώθηκε στις 24 Ιουνίου 2008

από σημαντικά ονόματα στον κλάδο των κινητών, συμπεριλαμβανομένων των Nokia, Sony Ericsson, Motorola και NTT DoCoMo. Ο στόχος του ιδρύματος είναι να δημιουργηθεί μια πιο αποδεδειγμένη, ανοικτή και πλήρη κινητή πλατφόρμα λογισμικού στον κόσμο. Όσον αφορά την τεχνολογία, ο στόχος είναι να ενοποιήσει το Symbian OS, το S60, το UIQ και το MOAP (**M**obile **O**riented **A**pplications **P**latform) (S) σε μια πλατφόρμα ανοικτού κώδικα για διάφορες κινητές συσκευές.

3.1.1 Τεχνολογία Symbian

Οι σχεδιαστικές αποφάσεις του Symbian OS αντανakλούν την προέλευσή του που αναφέρθηκαν ανωτέρω. Πρόκειται για ένα λειτουργικό σύστημα που στοχεύει ειδικά για συστήματα με περιορισμένους πόρους μνήμης και σε γενικές γραμμές, μετά το state-of-the design τέχνη από τη δεκαετία του 1990. Οι πιο χαρακτηριστικές σχεδιαστικές αποφάσεις έχουν ως εξής.

Symbian OS βασίζεται στην αρχιτεκτονική μικροπυρήνα. Ο όρος μικροπυρήνα συνδέεται συνήθως με ένα σχέδιο, όπου όλοι οι διαχειριστές των πόρων που έχουν πρόσβαση στο υποκείμενο υλικό τρέχουν σε διαφορετικές διαδικασίες (processes) και ο πυρήνας του λειτουργικού συστήματος παρέχει μόνο τον ελάχιστο προγραμματισμό (scheduling) και τον χειρισμό των διακοπών (interrupt). Επιπλέον, είναι σύνηθες για τον μικροπυρήνα να εφαρμόσει τον μηχανισμό message-passing που επιτρέπει στους διαχειριστές πόρων να επικοινωνούν χρησιμοποιώντας τις εγκαταστάσεις του τις οποίες ο πυρήνας παρέχει. Κατά το σχεδιασμό του Symbian OS, η εισαγωγή ενός μικροπυρήνα υπήρξε ένας σημαντικός στόχος.

Η χρήση της προσέγγισης μικροπυρήνα είχε ως αποτέλεσμα τον καθορισμό των ειδικών εξαρτημάτων λογισμικού, σκοπός του οποίου είναι να διαχειριστεί διαφορετικούς τύπους πόρων. Είναι συνήθες σε προσεγγίσεις μικροπυρήνα, οι εν λόγω διαχειριστές πόρων αναφέρονται ως διακομιστές (servers). Κάθε πόρος του συστήματος συνοψίζεται σε ένα διακομιστή, η οποία είναι υπεύθυνη για τη διαχείριση των πόρων. Όταν ο πόρος χρησιμοποιείται, ο πελάτης πρώτα επικοινωνεί με τον διακομιστή, και στη συνέχεια δημιουργεί μια “συνάντηση” (session) με τον διακομιστή. Μόλις εδραιωθεί αυτό το session, ο πελάτης μπορεί να χρησιμοποιήσει τους πόρους και τις υπηρεσίες που προσφέρονται από το

διακομιστή. Επίσης, η διαχείριση σφαλμάτων είναι ενσωματωμένος στο σύστημα. Εάν ένας server είναι “νεκρός”, τα μηνύματα λάθους περνάνε στους πελάτες, και αν ένας πελάτης “πεθαίνει”, ο διακομιστής θα πρέπει να είναι σε θέση να αποδεσμεύσει τους διατιθέμενους πόρους.

Ο σχεδιασμός του Symbian OS βασίζεται σε μεγάλο βαθμό στον object orientation. Μια σειρά από εκτεταμένα frameworks χρησιμοποιούνται για διάφορες εργασίες, περιλαμβανομένης της ανάπτυξης εφαρμογών (application framework) και των γενικών χειρισμού συμβάντων (active objects), τα οποία θα εξεταστούν στη συνέχεια. Η τελευταία συνδέεται επίσης με τη διαχείριση των πόρων, όπως servers - δηλαδή, οι διαχειριστές των πόρων - επικοινωνεί με τους πελάτες τους χρησιμοποιώντας μηνύματα, τη λήψη των οποίων έχει ως αποτέλεσμα τη δημιουργία ενός συμβάντος. Επιπλέον, από τις εγκαταστάσεις του προγραμματισμού για C ++, η γλώσσα που χρησιμοποιείται κυρίως στο Symbian OS, ήταν υπανάπτυκτες κατά τη στιγμή του σχεδιασμού, πολλές αποκλειστικές συμβάσεις θεσπίστηκαν για την αντιμετώπιση αυτών των περιπτώσεων. Με την πάροδο του χρόνου, η βελτίωση της off-the-shelf C++ συστημάτων έχει κατ' αρχήν, καταστήσει μερικές από αυτές τις συμβάσεις άχρηστες, αλλά η απομάκρυνσή τους από την πραγματική βάση κώδικα μπορεί να είναι δύσκολη.

3.1.2 Symbian - Εξέλιξη προς την πρωτοποριακή πλατφόρμα

Όπως και με κάθε μεγάλης διάρκειας ζωής λειτουργικό σύστημα, παραδοχές σχετικά με τον διαθέσιμο εξοπλισμό απαιτούσαν ανασχεδιασμό και την επανεξέταση των πολλών τμημάτων του Symbian OS. Δεδομένου ότι το κύριο framework του Symbian OS είναι τα smartphones (“έξυπνα τηλέφωνα”), είναι φυσικό ότι η εξέλιξη των υποτιθέμενων εγκαταστάσεων τους υπήρξε επίσης η κινητήρια δύναμη της εξέλιξης λειτουργικό σύστημα Symbian OS. Σημαντικές αλλαγές στο λειτουργικό σύστημα του Symbian OS περιλαμβάνουν τουλάχιστον τα εξής θέματα.

Οι hardware εγκαταστάσεις έχουν αλλάξει ριζικά. Οι αρχικές συσκευές Psion βασιζόντουσαν σε δίσκους RAM που ήταν γρήγορες, αλλά απαιτούνταν ένα ξεχωριστό αντίγραφο ασφαλείας κατά τη διάρκεια της εγγραφής. Ωστόσο, οι κινητές συσκευές συνήθως χρησιμοποιούν μνήμη flash αντί για δίσκους. Λόγω των

φυσικών χαρακτηριστικών της μνήμης flash, ένα νέο αρχείο απαιτείται σε κάθε περίπτωση.

Το μέγεθος της μνήμης σε διαφορετικές συσκευές κινητών έχει αυξηθεί τρομακτικά. Πράγματι, δεν είναι ασυνήθιστο να βρει κανείς εκατοντάδες megabytes διαθέσιμης μνήμης και gigabytes του δίσκου, το οποίο ήταν το κύριο επίτευγμα στην κατάσταση της τέχνης στις αρχές της δεκαετίας του 2000 όταν το πρώτο Symbian-based κινητό τηλέφωνο τέθηκε σε εφαρμογή. Κατά συνέπεια, πολλές εκτιμήσεις σχετικά με τη χρήση της μνήμης που ήταν μια αναγκαιότητα, στα πρώτα κινητά τηλέφωνα που χρησιμοποιούν λειτουργικό σύστημα Symbian OS δεν είναι πλέον αναγκαία, λαμβανομένου ιδίως υπόψη τα καθήκοντα του προγραμματιστή εφαρμογών. Για έναν, όμως, κατασκευαστή συσκευής, ωστόσο, τα ζητήματα αυτά εξακολουθούν να ισχύουν.

Παράλληλα με την αύξηση του μεγέθους της μνήμης, έχουμε και βελτίωση στις δυνατότητες επεξεργασίας. Εκτός από την αύξηση στην συχνότητα χρονισμού, έχουν όλο και περισσότερο τη δυνατότητα περιφερειακά να εισαχθούν. Σε γενικές γραμμές, η προσέγγιση μικροπυρήνα του Symbian είναι κατάλληλη για αυτό, αλλά διαφορετικές επεκτάσεις απαιτούν προφανώς προσπάθειες μηχανικής για την ένταξη τους στην πλήρη Symbian OS-based συστήματα.

Ο αριθμός των διαφόρων υποσυστημάτων που μπορεί να συνδεθούν στα τωρινά κινητά τηλέφωνα έχει αυξηθεί. Την ίδια στιγμή, μεμονωμένα υποσυστήματα έχουν γίνει με απίστευτο βαθμό πολύπλοκα. Κατά συνέπεια, η επίδρασή τους στην υποστήριξη για όλων αυτών έχει προστεθεί και η πολυπλοκότητα του Symbian OS. Για παράδειγμα, η διασύνδεση μεταξύ του Symbian OS και των βασικών πόρων του υλικού, το οποίο αρχικά αποτελείτο από λίγες μόνο διακομιστές έχει εξελιχθεί σε ένα σύνθετο framework, όπου η plug-in αρχιτεκτονική χρησιμοποιείται για την μεταβλητότητα μεταξύ των διαφόρων συνιστωσών του υλικού.

Η εισαγωγή της ασφάλειας της πλατφόρμα έχει απαιτήσει σημαντικό επανασχεδιασμό του Symbian OS. Κατά συνέπεια, αυτή η φάση σηματοδοτεί επίσης μια σημαντική διαταραχή στην εξέλιξη του Symbian OS. Για έναν application developer, αυτό σηματοδοτεί επίσης μια σημαντική παραβίαση της συμβατότητας μεταξύ διαφορετικών συσκευών σε γενικές γραμμές, για να μην αναφέρουμε binary συμβατότητα, το οποίο επίσης παραβιάζεται.

Τέλος, η ικανότητα του Symbian OS να στηρίζει τα παραπάνω είναι ένας φόρος τιμής στον αρχικό σχεδιασμό που εξακολουθεί να αποτελεί τον πυρήνα του λειτουργικού συστήματος. Ειδικότερα, η αρχιτεκτονική μικροπυρήνα έχει αποδειχθεί ότι είναι μια ευέλικτη πλατφόρμα για σύνθετες εφαρμογές, και την ενσωμάτωση των πόρων στους servers προσέφερε μια πρακτική προσέγγιση για τη διαχείριση του συνεχώς αυξανόμενου αριθμού των νέων πόρων που συνδέονται με τα κινητά τηλέφωνα. Από την άλλη πλευρά, η εξέλιξη της πλατφόρμας έχει οδηγήσει σε επιπλοκές που είναι ορατά από έναν απλό προγραμματιστή, ιδίως με τη μορφή της τεκμηρίωσης.

3.1.3 Symbian – Casual Application developer

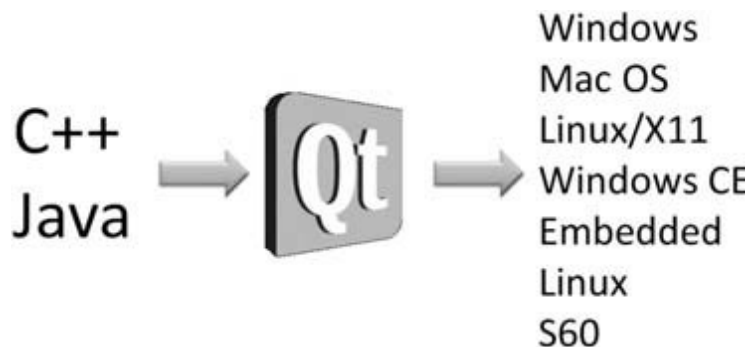
Ενώ οι διευκολύνσεις του Symbian OS έχουν βελτιωθεί, για ένα απλό application developer του Symbian OS έχουν παραμείνει κάπως προκλητικές. Οι προκλήσεις που μπορεί να σχετίζεται με δύο ιδιαίτερα σχεδιαστικές επιλογές: η Symbian αρχιτεκτονική της εφαρμογής, το οποίο πρόκειται να χρησιμοποιηθεί ως βάση για την ανάπτυξη εφαρμογών με γραφική διεπαφή χρήστη (GUI) και μια σειρά από platform-specific λεπτομέρειες, οι οποίες μπορεί να θεωρηθούν ως περιπτές και περιπλοκές από έναν προγραμματιστή εφαρμογής. Αυτό θεωρείται απογοητευτικό όταν η ανάπτυξη εφαρμογών για Symbian OS γίνεται μόνο περιστασιακά. Εκτός από την Python και το Open C / C ++, Qt για Symbian είναι μια ενδιαφέρουσα λύση. Το Qt είναι μια εφαρμογή cross-platform framework ανάπτυξης εφαρμογών που χρησιμοποιείται ευρέως για την ανάπτυξη προγραμμάτων GUI. Παρόλο που συνήθως συνδέονται με τον προγραμματισμό GUI, το σύστημα περιλαμβάνει επίσης πολλές άλλες συνιστώσες, όπως threads και η διασύνδεση με τα υποσυστήματα όπως η βάση δεδομένων SQL και ανάλυσης XML. Εσωτερικά το Qt βασίζεται σε C ++, αλλά περιλαμβάνει πολλές no-standard επεκτάσεις σε εφαρμογή από ένα πρόσθετο προεπεξεργαστή, η οποία, ωστόσο, δημιουργεί πρότυπο C ++ κώδικα. Τέλος, το Qt τρέχει σε όλες τις σημαντικές πλατφόρμες και υποστηρίζει τη διεθνοποίηση.

3.2 Qt

Qt , όπως προφέρεται πιο χαριτωμένα, είναι ένα προϊόν της QtSoftware που ονομάστηκε αρχικά Trolltech (που ιδρύθηκε το 1994). Με βάση την αρχική ιδέα της δημιουργίας ενός object-oriented συστήματος, το πρώτο πρωτότυπο ήταν έτοιμο το 1993, και Trolltech ιδρύθηκε ένα χρόνο αργότερα στη Νορβηγία. Στις 20 Μαΐου 1995, η πρώτη δημόσια διανομή του ξεκίνησε με το όνομα Qt 0.9. Το όνομα ήταν αποτέλεσμα της nice-looking γράμμα “Q” στον επεξεργαστή κειμένου Emacs Harvald και το “t” από την λέξη “toolkit”, εμπνευσμένο από την τεχνολογία Xt (X Toolkit). Κατά τη διάρκεια των τελευταίων ετών, το Qt έχει χρησιμοποιηθεί από πολλούς πελάτες, όπως το Google, το Skype, τη Volvo και πολλούς άλλους.

3.2.1 Ένα Cross-platform περιβάλλον ανάπτυξης

Η κύρια ιδέα πίσω από το Qt ήταν η δυνατότητα να προγραμματίσετε μια εφαρμογή σε C ++ ή Java μια φορά και να αναπτύξετε αυτό σε πολλά desktop και ενσωματωμένα λειτουργικά συστήματα χωρίς αλλαγή του κώδικα σε όλα (Εικόνα 3.2.1α). Από την πλευρά των προγραμματιστών το API του Qt παραμένει το ίδιο σε όλες τις υποστηριζόμενες πλατφόρμες, με την προϋπόθεση ότι οι προγραμματιστές χρησιμοποιούν κοινές API μόνο και δεν χρησιμοποιούν device-specific interfaces, όπως αυτά που προσφέρονται στον τομέα των κινητών συσκευών μόνο. Επιπλέον, το ίδιο toolchain είναι διαθέσιμο για τα διαφορετικά υπολογιστικά συστήματα για να υποστηρίξει ανάλογες εμπειρίες ανάπτυξης, η οποία με τη σειρά του οδηγεί σε αύξηση της παραγωγικότητας κατά τη διάρκεια διαφορετικών toolchains.



Εικόνα 3.2.1α: Qt διαδικασία ανάπτυξης: πρόγραμμα μία φορά και τη χρήση πολλαπλές πλατφόρμες.

Όταν χρησιμοποιείται το Qt οι εφαρμογές που προκύπτουν θα πρέπει να μοιάζουν με ένα εγγενές πρόγραμμα της συγκεκριμένης πλατφόρμας προκειμένου να εξασφαλιστεί η φιλικότητα προς το χρήστη, δηλαδή μια πιο προσαρμοστική εμφάνιση και αίσθηση. Στην Εικόνα 3.2.1β φαίνεται η εμφάνιση των Qt κουμπιών σε πολλαπλές πλατφόρμες. Σε αντίθεση με την ενότητα SWING της Java όταν οι εφαρμογές θα πρέπει επίσης να διατρέξουν για όλες τις διαφορετικές πλατφόρμες, η διεπαφή χρήστη θα φαινόταν παντού το ίδιο, που θα μπορούσε να προκαλέσει εκνευρισμό στον πελάτη που χρησιμοποιείται για μια συγκεκριμένη χρήση. Το Qt έρχεται με μια διαισθητική βιβλιοθήκη κατηγορίας και ολοκληρωμένη υποστήριξη που προσφέρει εργαλεία ανάπτυξης για C ++ και την ανάπτυξη σε Java. Εκτός από το API και την cross-compilation υποστήριξη, και σε ορισμένες περιπτώσεις το πιο σημαντικό, εργαλεία υποστήριξης ανάπτυξης εισήχθησαν στο Qt, όπως Designer Qt ή το Qt Linguist, τα οποία θα αναληθούν σε επόμενο κεφάλαιο.



Εικόνα 3.2.1β: Εμφάνιση των Qt κουμπιών σε διαφορετικές πλατφόρμες

3.2.2 Qt στο Mobile Domain

Στην αρχή το Qt είχε ως στόχο τα Windows, τα Mac OS (αρχής γενομένης με το Qt 3,0) και τα λειτουργικά συστήματα Linux μόνο. Ωστόσο, η προσέγγιση ήταν τόσο ελκυστική ώστε, που σύντομα επεκτάθηκαν και σε άλλα ενσωματωμένα συστήματα, το 2000. Τέλος, το 2006 εισήγαγε την Trolltech Greenphone, το οποίο είναι ένα πλήρως χαρακτηρισμένο κινητό τηλέφωνο που βασίζεται στο Linux (Εικόνα 3.2.2α). Το Greenphone είχε ήδη μια οθόνη αφής και μια σειρά από ασύρματες διεπαφές. Στην εικόνα 3.2.2β φαίνεται το user interface του Greenphone.



Εικόνα 3.2.2α: Το Greenphone



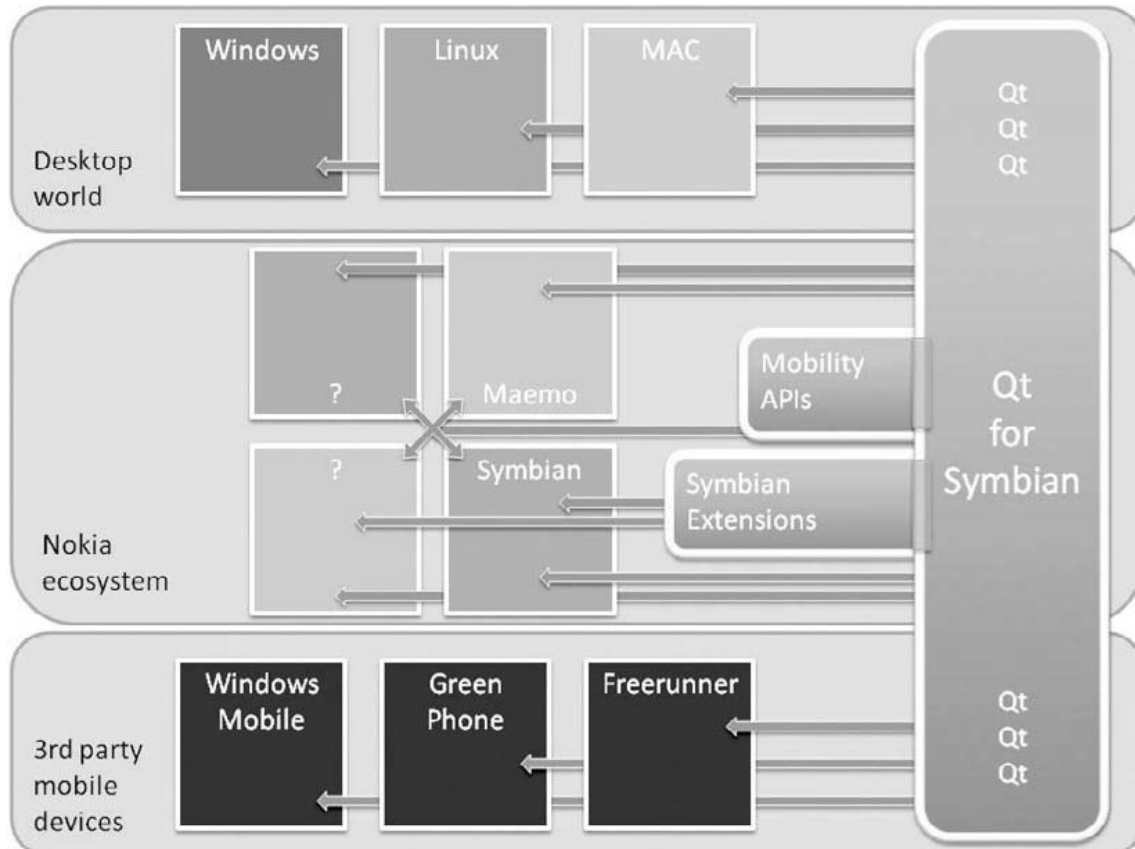
Εικόνα 3.2.2β: Το user interface του Greenphone

Το 2008 η Nokia εξαγόρασε την Trolltech και εισήγαγε το Qt στις πλατφόρμες της, αρχής γενομένης με τις πλατφόρμες Symbian και Linux. Μαζί με την τεχνολογία της Nokia έγινε προσιτή και η πρόσβαση σε μια μεγάλη και ευκίνητη κοινότητα προγραμματισμού. Αλλά προγραμματισμού, όπως αυτό είναι ένας δρόμος διπλής κατεύθυνσης, Qt προγραμματιστές μπορούν να έχουν πρόσβαση σε εκατομμύρια mobile domains, με εκατομμύρια καινούργιες συσκευές ετησίως. Εκτός από τις ήδη υπάρχουσες γλώσσες προγραμματισμού Java, Symbian, C++, Python ή Flash, η εισαγωγή του Qt και του toolchain της, στις κινητές συσκευές θα μειώσει τον αριθμό των εμποδίων για τη δημιουργία μίας mobile εφαρμογής.

Το όραμα της Qt έχει να την καθιερώσει παντού - που θα καλύπτει τόσο desktop εφαρμογές όσο και εφαρμογές που αφορούν τον mobile τομέα. Στον mobile τομέα το Qt είναι διαθέσιμο για Linux-enabled συσκευές και Nokia έχει ωθήσει το Qt πάνω στη Symbian πλατφόρμα. Για να μπορέσει το Qt να χρησιμοποιήσει το πλήρες φάσμα των κινητών λειτουργιών, όπως αυτές που δεν είναι διαθέσιμες στην επιφάνεια εργασίας παγκοσμίως, το νέο Symbian API πρέπει να εισαχθούν. Αυτά τα APIs είναι υπεύθυνοι για τη χρήση πληροφοριών εντοπισμού, μηνυμάτων όπως τα SMS και MMS, φωτογραφικές μηχανές, ενσωματωμένο αισθητήρες και πολλά άλλα. Φυσικά, είναι δυνατή η δημιουργία

κινητών εφαρμογών για την πλατφόρμα Symbian χωρίς τη χρήση του νεοεισαχθέντα API, αλλά προφανώς τα APIs έχουν τη δυνατότητα να δημιουργήσει πραγματικά κινητές εφαρμογές για τις υπηρεσίες εντοπισμού θέσης, της κοινωνικής κινητά δίκτυα και πολλές άλλες.

Στο διάγραμμα 3.2.2 απεικονίζεται η εφαρμογή του λογισμικού του Qt για τις διαφορετικές πλατφόρμες. Όποια γλώσσα του Qt χρησιμοποιείται, ο κώδικας θα πρέπει να “δουλέψει” σε οποιαδήποτε πλατφόρμα, σε επιτραπέζιους και φορητούς υπολογιστές, το Nokia οικοσύστημα και σε mobile συσκευές τρίτων. Από την άλλη πλευρά, τα Symbian APIs μπορούν να χρησιμοποιηθούν μόνο για την πλατφόρμα Symbian και μπορεί να επεκταθεί αργότερα και για άλλες πλατφόρμες του οικοσυστήματος της Nokia. Μια τέτοια προσέγγιση είναι παρόμοια με τη ενσωματωμένη Qt προσέγγιση, όπου οι λειτουργίες όπως το τηλέφωνο που σχετίζονται με την πρόσβαση των ημερολογίων και επαφών ή τη χρήση ασύρματων μικρής εμβέλειας Bluetooth περιορίζεται στο ενσωματωμένο περιβάλλον.



Διάγραμμα 3.2.2: Οι δυνατότητες της cross-platform Qt σε διαφορετικές κινητές και μη πλατφόρμες.

3.2.3 Άδεια χρήσης του Qt

Το πιο σημαντικό θέμα για τους mobile developers είναι η συμφωνία παραχώρησης άδειας εκμετάλλευσης. Το Qt προσφέρει διαφορετικά υποδείγματα άδειας χρήσης. Ενώ οι εμπορικές και GPL εκδόσεις ήταν παρόντες από την αρχή, η LGPL έχει μόνο προστεθεί προσφάτως. Το τελευταίο επιτρέπει στους προγραμματιστές να αποκομίζουν κέρδη από τις εφαρμογές τους χωρίς να κάνουν τον πηγαίο κωδικά τους γνωστό στο κοινό. Το Qt είναι διαθέσιμο σύμφωνα με τις ακόλουθες άδειες χρήσης:

5. *Qt Commercial Developer License*
6. *Qt GNU LGPL v2.1*
7. *Qt GNU GPL v3.0*

3.2.3.1 Qt Commercial Developer License

Η Qt Commercial Developer License είναι η σωστή άδεια χρήσης για την ανάπτυξη των ιδιόκτητων ή / και εμπορικού λογισμικού με χρήση του Qt, όπου δεν θέλετε να μοιραστείτε τον πηγαίο κώδικα.

Θα πρέπει να αγοραστεί μία Qt Commercial Developer License από την ίδια την εταιρεία ή από έναν από τους εξουσιοδοτημένους μεταπωλητές της, πριν να αρχίσει να αναπτύσσεται ένα εμπορικό λογισμικό. Τέλος η Qt Commercial Developer License περιλαμβάνει έναν περιορισμό που εμποδίζει ο συνδυασμός των κώδικα που αναπτύχθηκε με το Qt GNU LGPL v. 2.1 ή GNU GPL κατά 3,0 εκδόσεις αδειών με εμπορική άδεια χρήσης κωδικό Qt.

Εμπορική Δικαιώματα και runtimes

Για κανονικές εφαρμογές υπολογιστών γραφείου, δεν υπάρχουν δικαιώματα, οι άδειες εκτέλεσης, ή άλλα πρόσθετα έξοδα. Για τα σταθερά συστήματα ή συσκευές όπου η κύρια λειτουργία βασίζεται στο Qt, ένα χρόνο λειτουργίας των αδειών μπορεί να ισχύουν.

Αγορά και Μεταβίβαση Qt Commercial Developer License

Η άδεια πωλείται σε μια βάση ανά-developer και ανατίθενται για ένα άτομο. Για να παραγγελθούν μία ή περισσότερες Qt άδειες σε εμπορικούς developers πρέπει να επικοινωνήσετε με την εταιρεία. Οι Qt άδειες μπορούν να

μεταβιβάζονται, αλλά μόνο κάθε έξι μήνες και εντός του ίδιου οργανισμού. Για να μεταφερθεί μια άδεια, ή για να συζητήσει μια πιο ευέλικτη συμφωνία αδειοδότησης, πάλι πρέπει να επικοινωνήσετε με την εταιρεία.

Όφελος για το Qt

Το όφελος για το Qt είναι ότι με τα τέλη αδειών θα βοηθήσει να πληρώσει για την περαιτέρω ανάπτυξη του περιβάλλοντος του Qt.

3.2.3.2 Qt GNU LGPL v2.1

Αυτή η έκδοση του Qt είναι κατάλληλο για την ανάπτυξη της Qt εφαρμογών (ιδιόκτητα ή open source) με τον όρο ότι μπορούν να συμμορφωθούν με τους όρους και τις προϋποθέσεις που περιλαμβάνονται στην έκδοση GNU LGPL v2.1.

Όφελος για το Qt

Το όφελος για το Qt είναι ότι κάθε μεταβολή του Qt πρέπει να διατίθενται υπό τους όρους της παρούσας άδειας.

3.2.3.3 Qt GNU GPL v3.0

Αυτή η έκδοση του Qt είναι κατάλληλη για ανάπτυξη Qt εφαρμογών που επιθυμεί να χρησιμοποιήσει ο κάθε χρήστης, σε συνδυασμό με λογισμικό το οποίο υπόκειται στους όρους της GNU General Public License v3.0 ή εάν είναι πρόθυμος ο developer να συμμορφωθούν με τους όρους της GNU General Public License v3.0.

Όφελος για το Qt

Το όφελος για το Qt είναι ότι κάθε μεταβολή του Qt πρέπει να διατίθενται υπό τους όρους της παρούσας άδειας.

Τέλος, στον πίνακα 3.2.3 φαίνονται οι συμφωνίες παραχώρησης αδειών εκμετάλλευσης σε σχέση με τον άλλον.

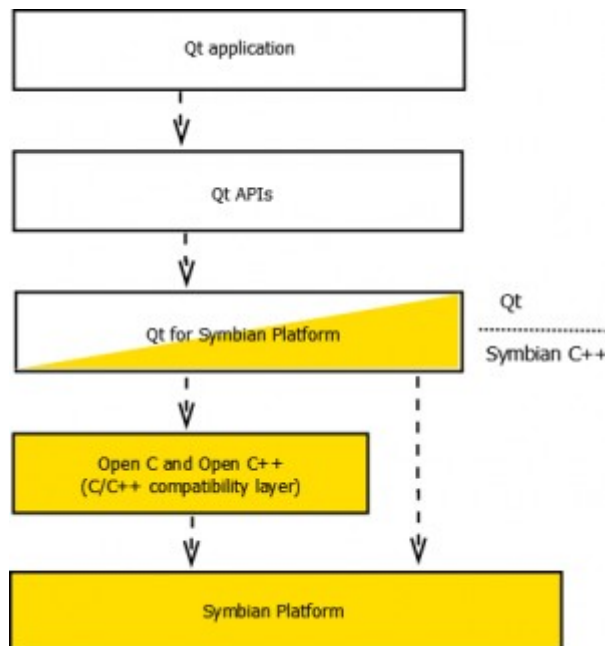
	Commercial	LGPL	GPL
Κόστος άδειας	Άδεια, τέλος που επιβάλλεται	Χωρίς τέλη άδειας	Χωρίς τέλη άδειας
Πρέπει να παρέχει τις αλλαγές του πηγαίου κώδικα της Qt	Όχι, οι τροποποιήσεις μπορεί να είναι κλειστές	Ο πηγαίος κώδικας πρέπει να παρέχονται	Ο πηγαίος κώδικας πρέπει να παρέχονται
Μπορεί να δημιουργήσει αποκλειστικές εφαρμογές	Ναι – Ο πηγαίος κώδικας δεν πρέπει να δημοσιοποιηθεί	Ναι, σύμφωνα με την LGPL v2.1 όρους	Όχι, οι εφαρμογές είναι αντικείμενα που υπόκεινται στον GPL και ο πηγαίος κώδικας πρέπει να διατίθενται
Παρέχονται ενημερώσεις	Ναι, η άμεση πρόσκληση που αποστέλλεται σε άτομα με έγκυρη υποστήριξη και συμφωνία ενημέρωση	Ναι, τίθενται στη διάθεση	Ναι, τίθενται στη διάθεση
Υποστήριξη	Ναι, για τα άτομα με έγκυρη υποστήριξη και συμφωνία ενημέρωση	Δεν περιλαμβάνονται αλλά διατίθενται ξεχωριστά για την αγορά	Δεν περιλαμβάνονται αλλά διατίθενται ξεχωριστά για την αγορά
Επιβάρυνση για Runtimes	Ναι, για κάποιες ενσωματωμένες χρήσεις	Όχι	Όχι

Πίνακας 3.2.3

3.3 Επίλογος

Σε αυτό το κεφάλαιο είδαμε την ιστορία του Symbian, την πλατφόρμα του Symbian OS και είδαμε γενικές πληροφορίες για το περιβάλλον ανάπτυξης εφαρμογών του Qt.

Το Qt είναι ένα περιβάλλον ανάπτυξης εφαρμογών το οποίο βοηθά τους προγραμματιστές να προγραμματίζουν εφαρμογές που μπορεί να είναι desktop εφαρμογές αλλά και εφαρμογές που αναφέρονται σε smartphone πλατφόρμες. Ακόμα το Qt δίνει δυνατότητα στους προγραμματιστές με τον ίδιο κώδικα η εφαρμογή τους να “παίζει” παντού. Το τελευταίο διάστημα αυτή η πλατφόρμα έχει λύσει τα χέρια των προγραμματιστών και προσφέρει σε αυτούς την δυνατότητα να προγραμματίζουν εφαρμογές που σχετίζονται με το Symbian OS και το Windows Mobile OS. Στο επόμενο κεφάλαιο θα δούμε το σχετικά με το Qt Mobility το οποίο σχετίζεται με τις διάφορα APIs που αναφέρονται στις mobile OS πλατφόρμες και προσφέρει στους προγραμματιστές να έχουν πρόσβαση στις διάφορες του smartphone. Από την προσθήκη, τροποποίηση μίας επαφής και την αποστολή ενός κείμενο μέσω SMS μέχρι την πρόσβαση του σε διάφορα “τσιπάκια” των smartphones όπως είναι οι αισθητήρες του.



Εικόνα 3.3: Symbian C++ και Qt ως ένα

Πτυχιακή εργασία του φοιτητή Ελευθέριου Γεωργίου

4

Qt Mobility

Στο κεφάλαιο αυτό παρουσιάζεται το Qt Mobility API v1.1. Ο στόχος αυτών των APIs είναι να καταστεί δυνατή η χρήση του κινητού τηλεφώνου σε λειτουργιών Qt εφαρμογών. Τα APIs είναι εύκολο στη χρήση και έχουν σχεδιαστεί για χρήση cross-platform.

4.1 Εισαγωγή

Όπως φαίνεται, το Qt framework εισάγει ένα ισχυρό μέσο για την δημιουργία cross-platform εφαρμογών με φιλικό και εύχρηστο περιβάλλον για τον χρήστη. Επιπλέον, το Qt παρέχει APIs που βρίσκουν εφαρμογή σε πολλά σημαντικότερα χαρακτηριστικά του συστήματος, συμπεριλαμβανομένων των εγκαταστάσεων όπως η δικτύωση παράδειγμα, το αρχείο που συνδέονται με τη λειτουργικότητα και το multi-threading, για να αναφέρουμε μόνο μερικά. Από τη φύση τους, αυτές τις δυνατότητες του συστήματος να υπάρχουν σε όλες τις πλατφόρμες – συμπεριλαμβανομένων τόσο των φορητών και desktop περιβαλλόντων. Συνεπώς, είναι φυσικό να παρέχονται για την ανάπτυξη cross-platform, ως αναπόσπαστο μέρος του γενικού Qt.

Ωστόσο, οι φορητές συσκευές όπως τα smartphones έχουν πολλές ιδιαιτερότητες. Αυτές ξεφύγουν από τις κοινές βιβλιοθήκες του Qt cross-platform. Για παράδειγμα, οι πληροφορίες τοποθεσίας και mobile συστήματα μηνυμάτων είναι μερικά χαρακτηριστικά που είναι χαρακτηριστικά που ισχύουν για φορητά συστήματα μόνο. Δεδομένου ότι το Qt υποστηρίζεται τώρα από μια σειρά κινητές πλατφόρμες, ένα νέο σύνολο API για χρήση mobile χαρακτηριστικών σε μια μορφή cross-platform καθίσταται πρακτική ανάγκη. Το Qt πακέτο υλοποιεί ένα σύνολο από APIs που παρέχουν πρόσβαση στα πιο συχνά χρησιμοποιούμενα mobile χαρακτηριστικά χωρίς να αναγκάζει τον developer της εφαρμογή να εξαρτάται από

την πλατφόρμα εγγενή κώδικα όπως το Symbian C++.

Το Qt Mobility v1.1 παρέχει ένα σύνολο από νέα API το Qt. Αυτά τα APIs παρέχουν πρόσβαση σε λειτουργίες που είναι γνωστά από άλλα mobile συστήματα προγραμματισμού, όπως η Python ή Mobile Java. Επιπλέον, όπως σε οποιαδήποτε άλλη mobility βιβλιοθήκη, αυτά τα APIs επιτρέπουν στον προγραμματιστή να χρησιμοποιεί τις mobile δυνατότητες που σχετίζονται με ευκολία από ένα ενιαίο framework και τις εφαρμόζουν σε κινητά τηλέφωνα, φορητούς και επιτραπέζιους υπολογιστές, με την προϋπόθεση βέβαια ότι οι συναφείς ευκολίες είναι διαθέσιμες στη συσκευή.

Τα APIs του Qt Mobility είναι πραγματικά API cross-platform. Ως εκ τούτου, το framework δεν παρέχει μόνο πολλές πτυχές της mobile εμπειρίας στην ανάπτυξη εφαρμογών με την απλούστευση χρήση των συναφών τεχνολογιών, αλλά και έχει εφαρμογή πέρα από την mobile συσκευή. Κατά συνέπεια, ο πηγαίος κώδικας μπορεί να επαναχρησιμοποιηθεί πέρα από τα σύνορα του Symbian. Στη συνέχεια θα παρουσιασθούν τα διάφορα μέρη του Qt Mobility API.

Στον παρακάτω πίνακα φαίνονται τα API του Qt Mobliity εν συντομία:

Bearer Management	API για τον έλεγχο της κατάστασης του συστήματος.
Camera	API για τον έλεγχο των συσκευών λήψης εικόνων
Contacts	API που επιτρέπει στους πελάτες να ζητούν στοιχεία επαφής από την τοπική ή απομακρυσμένη backend.
Document Gallery	API για την πλοήγηση και αναζήτηση εγγράφων με χρήση meta-data τους.
Feedback	Ένα API που επιτρέπει τον πελάτη για τον έλεγχο των δονήσεων της συσκευής (όταν ένας δονητής χρησιμοποιείται) είτε τα “piezo feedback” ανατροφοδότηση από την οθόνη.
Location	API που παρέχει μια βιβλιοθήκη για την θέση, τη διαχείριση ορόσημο και χαρτογράφηση και πλοήγηση.
Messaging	API που επιτρέπει την πρόσβαση σε υπηρεσίες μηνυμάτων.
Multimedia	Ένα σύνολο από APIs για να παίξει και

	καταγράφουν media και να διαχειρίζονται μια συλλογή από media περιεχόμενα.
Organizer	API που επιτρέπει στους πελάτες να ζητήσουν το ημερολόγιο, το χρονοδιάγραμμα και τα προσωπικά δεδομένα από τα τοπικά ή απομακρυσμένα backend.
Publish and Subscribe	API που επιτρέπει στις εφαρμογές να διαβάζουν τιμές, να περιηγούνται και να εγγραφούν σε αλλαγή κοινοποιήσεων.
Qt Service Framework	Μια σειρά από Qt APIs που επιτρέπει στους πελάτες να ανακαλύψουν και υπόσταση σε αυθαίρετες υπηρεσίες.
Sensors	API που παρέχει πρόσβαση σε αισθητήρες.
System Information	Ένα σύνολο από APIs για να ανακαλύψετε το σύστημα σχετικές πληροφορίες και δυνατότητες.
Telephony Events	API που επιτρέπει την πρόσβαση σε τηλεφωνικές υπηρεσίες που σχετίζονται με γεγονότα τηλεφώνου.
Versit	API για την εισαγωγή και την εξαγωγή προς τη μορφή vCard και iCalendar.

Πίνακας 4.1: Τα API του Qt Mobility v1.1

4.2 Bearer Management (Φορέας Διαχείρισης)

Ο θεμελιώδης στόχος του bearer management είναι να μειώσει τις ανησυχίες των developer στην ανάπτυξη, όταν τον εντοπισμό των βέλτιστων διαθέσιμων σύνδεση από μια λίστα πιθανών φορέων IP και 3G συνδέσεων. Ο χρήστης μπορεί να επιλέξει την καλύτερη σύνδεση, ή η σύνδεση μπορεί να επιλέγεται με διαφανή διαδικασία, έτσι ώστε η WLAN περιαγωγή (roaming) μπορεί να συμβεί απρόσκοπτα.

Αυτό το API ελέγχει την κατάσταση σύνδεσης του συστήματος., ο χρήστης μπορεί να αρχίσει ή να σταματήσει τις διεπαφές επικοινωνίας και να περιφέρεται (roam) μεταξύ σημείων πρόσβασης με διαφάνεια.

Μερικά παραδείγματα από τη λειτουργία αυτού του API αναφέρονται παρακάτω:

- On-demand χρήση του κατάλληλου σημείου πρόσβασης. Όταν ένας χρήστης ξεκινά να χρησιμοποιήσετε ένα πρόγραμμα περιήγησης το κατάλληλο σημείο πρόσβασης έχει επιλεγεί από τις διαθέσιμες αυτά και η σύνδεση γίνεται με διαφάνεια.
- Εφαρμογές όπως το ηλεκτρονικό ταχυδρομείο ή IM περιφέρονται αυτόματα μεταξύ κυτταρικού πακέτων δεδομένων και WLAN. Ο προγραμματιστής της εφαρμογής έχει τον έλεγχο και αν χρειαστεί η εφαρμογή μπορεί να κλείσει ομαλά τις TCP συνδέσεις πάνω από το παλιό κομιστή μετά τη δημιουργία νέων συνδέσεων TCP πάνω από το νέο φορέα, για παράδειγμα.
- Ο developer της εφαρμογής μπορεί να δημιουργήσει ένα περιβάλλον εργασίας χρήστη ρυθμίσεις, όπου ο χρήστης μπορεί να επιλέξει ένα συγκεκριμένο συνδυασμό που θα χρησιμοποιηθεί με την εφαρμογή.
- Μια εφαρμογή διαχείρισης συνδέσεων μπορούν να εφαρμοστούν με το Qt. Η εφαρμογή αυτή απαριθμεί διαθέσιμες συνδέσεις, δείχνει μετρητές και επιτρέπει στο χρήστη να συνδεθεί ή να αποσυνδεθεί.

4.2.1 Δημιουργία ενός network session

Το `QNetworkConfigurationManager` είναι μια κλάση που διαχειρίζεται τις διαμορφώσεις του δικτύου που παρέχει το σύστημα. Η κλάση `QNetworkConfiguration` παρέχει μία αφαίρεση των συνθέσεων του σημείου πρόσβασης. Το ακόλουθο παράδειγμα κώδικα δείχνει πώς μια σύνοδο δίκτυο μπορεί να αποδειχθεί χωρίς καμία παρέμβαση από τον χρήστη:

```
// Set Internet Access Point

QNetworkConfigurationManager manager;

const bool canStartIAP = (manager.capabilities() &
QNetworkConfigurationManager::CanStartAndStopInterfaces);

// Is there default access point, use it

QNetworkConfiguration cfg = manager.defaultConfiguration();

if (!cfg.isValid() || (!canStartIAP && cfg.state() != QNetworkConfiguration::Active))

{
```

```
QMessageBox::information(this, tr("Network"), tr("No Access Point found."));  
  
return;  
  
}  
  
session = new QNetworkSession(cfg, this);  
  
session->open();  
  
session->waitForOpened(-1); // msec is -1, this function will not time out.
```

4.3 Camera

Το API για την κάμερα επιτρέπει σε υψηλό επίπεδο ελέγχου των διαφόρων πτυχών των σταθερών εικόνων και βίντεο. Η κάμερα είναι ένα μέρος των API πολυμέσων και η σχέση αυτή είναι εμφανής όταν παρατηρηθεί ότι ορισμένες κατηγορίες βασικών κλάσεων είναι subclassed από ορισμένες Multimedia base κλάσεις συμπεριλαμβανομένων *QMediaObject* και *QMediaControl*.

```
camera = new QCamera;  
  
viewFinder = new QCameraViewfinder();  
  
viewFinder->show();  
  
camera->setViewfinder(viewFinder);  
  
imageCapture = new QStillImageCapture(camera);  
  
camera->start();  
  
//on half pressed shutter button  
  
camera->lock();  
  
//on shutter button pressed  
  
imageCapture->capture("imagefile.jpg");  
  
//on shutter button released  
  
camera->unlock();
```

4.4 Contacts (Επαφές)

Η θεμελιώδης χρήση για ένα κινητό τηλέφωνο και για την ίδρυση των επικοινωνιών είναι οι επαφές. Συνήθως, αυτό γίνεται με μια ήδη υπάρχουσα

επαφή. Κατά συνέπεια, μία από τις ειδικές περιπτώσεις χρήσεις μιας mobile συσκευής είναι η διαχείριση των επαφών. Επομένως, αυτό είναι ένα από τα πιο ουσιώδεις επεκτάσεις του Qt Mobility από τη σκοπιά του προγραμματιστή.

Το API για τις επαφές ορίζει τη δομή και την ανάκτηση των δεδομένων επαφής από την τοπική ή απομακρυσμένη backend. Ακόμη, το API παρέχει λειτουργίες όπως τη δημιουργία, την επεξεργασία, τη λίστα, να διαγραφεί και αναζήτησης στοιχεία επικοινωνίας, ανεξάρτητα από το αν τα δεδομένα που είναι αποθηκευμένα τοπικά είτε εξ αποστάσεως.

4.4.1 Δημιουργία μιας καινούριας επαφής

Μια επαφή μπορεί να δημιουργηθεί με τη δημιουργία μιας εμφάνισης του αντικειμένου *QContact*, προσθέτοντας τα στοιχεία επικοινωνίας και να το αποθηκευθεί στην βάση δεδομένων επαφών με την κλάση *QContactManager*. Αυτό γίνεται στο ακόλουθο κώδικα:

```
QContactManager* contactManager = new QcontactManager(this);

QContact homer;

// Create name detail

QContactName name;

name.setFirst("Ελευθέριος");

name.setLast("Γεωργίου");

name.setCustomLabel("Elgeor");

homer.saveDetail(&name);

// Create phone number detail

QContactPhoneNumber number;

number.setContexts(QContactDetail::ContextHome);

number.setSubTypes(QContactPhoneNumber::SubTypeMobile);

number.setNumber("555112233");

homer.saveDetail(&number);
```

```
homer.setPreferredDetail("DialAction", number);  
  
// Create address detail  
  
QContactAddress address;  
  
address.setCountry("Ελλάδα");  
  
address.setRegion("Σκιάθος");  
  
address.setPostCode("37002");  
  
homer.saveDetail(&address);  
  
// Save the contact to the contacts database  
  
contactManager->saveContact(&homer);  
  
}
```

4.5 Document Gallery

Το Document Gallery API παρέχει ένα API για την πλοήγηση και επερώτηση των εγγράφων που χρησιμοποιούν τα meta-data τους. Μας επιτρέπει ενημερώνουμε τις απόψεις επιλογή των μέσων του μια συσκευή αναπαραγωγής μουσικής ή την προβολή εικόνων. Προβλέπει επίσης API για την αναζήτηση και την επεξεργασία των meta-data των μεμονωμένων εγγράφων.

Συνήθως όλα τα αρχεία σε καταλόγους χρηστών πρέπει να είναι διαθέσιμες στη συλλογή εγγράφων. Οι βασικοί τύποι είναι ήχου, βίντεο, εικόνας, και εγγράφου, αλλά τα αρχεία μπορεί επίσης να χαρακτηριστούν ως ένα αρχείο, φάκελο, κείμενο ή playlist. Υπάρχουν επίσης ορισμένα meta-types που αποτελείται από τα meta-data των αρχείων, αυτές είναι οι καλλιτέχνη (Artist), άλμπουμ (Album), AudioGenre και PhotoAlbum.

Το Document Gallery API αποτελείται από ένα σύνολο ασύγχρονων αιτήσεων (*QGalleryAbstractRequest*) κλάσεις που μιλούν στο αρχείο του συστήματος παροχής indexing υπηρεσιών, αυτό θα είναι το tracker για την πλατφόρμα του Maemo, τα meta-data του συστήματος (MDS) για Symbian και δυνητικά για το Windows Search, ή Spotlight για τα Macs. Υπάρχουν τέσσερα αιτήματα, το *QGalleryQueryRequest* το οποίο παίρνει ένα type, ένα στοιχείο γονέα, καθώς και ορισμένα κριτήρια φιλτραρίσματος (*QGalleryFilter*) και επιστρέφει meta-data για

όλα τα στοιχεία που ταιριάζουν σε μια *QGalleryItemList*. Το *QGalleryItemRequest* παίρνει ένα στοιχείο ταυτότητας και επιστρέφει meta-data για ακριβώς αυτό το στοιχείο. Το *QGalleryUrlRequest* παίρνει το URL ενός στοιχείου και να επιλύει το αναγνωριστικό και τον τύπο του εν λόγω στοιχείου. Το *QGalleryRemoveRequest* παίρνει ένα στοιχείο ταυτότητας και αφαιρεί το στοιχείο αυτό από τη συλλογή.

Οι αιτήσεις που λειτουργούν με εφαρμογές των *QAbstractGallery*. Η default υλοποίηση είναι η *QDocumentGallery*, αλλά είναι δυνατόν να δημιουργηθούν εναλλακτικές εφαρμογές που μιλούν σε ένα διακομιστή μέσω ενημέρωσης ή των υπηρεσιών web.

Το κύριο σημείο διεπαφής με στοιχεία σε μια συλλογή είναι η *QGalleryItemList* κλάση που προβλέπει accessors για την ανάγνωση και την γραφή των meta-data από ένα σύνολο αντικειμένων. Ένα *QGalleryItemList* δεν μπορεί να δημιουργηθεί άμεσα, αντί για παράδειγμα, πρέπει να ζητηθεί από μια συλλογή με μία από τις κατηγορίες αιτήματος gallery.

4.6 Feedback (Ανατροφοδότηση)

Το Feedback API επιτρέπει στους προγραμματιστές για την υλοποίηση της εφαρμογής τους απτή ανάδραση στις αιτήσεις τους. Ακόμα το API παρέχει πληροφορίες για διάφορους τύπους απτής ανάδρασης, και επιτρέπει προσαρμοσμένες εφαρμογές αυτών των τύπων ανατροφοδότησης.

Η κλάση *QFeedbackEffect* παρέχει μια απλή διεπαφή για να επιτρέψει στους προγραμματιστές να ξεκινήσουν, να σταματήσουν ή να διακόψουν προσωρινά την αναπαραγωγή ενός αποτελέσματος. Πρόκειται για μια abstract κλάση που εφαρμόζεται από *QFeedbackHapricsEffect* και *QFeedbackFileEffect*. Επιπλέον, επιτρέπει τη στιγμιαία αναπαραγωγή αφής αποτελέσματα που ταιριάζουν με το θέμα του συστήματος.

Το *QFeedbackHapricsEffect* είναι ένα προσαρμοσμένο εφέ που επιτρέπει σε ένα πρόγραμμα να παίζει απτική επίδραση σε έναν actuator. Ένα τέτοιο αποτέλεσμα μπορεί να έχει διάρκεια, η ένταση, και την επιλογή της αναπαραγωγής (συμπεριλαμβανομένου του χρόνου επίθεσης και την ένταση, και fade χρόνου και ένταση), καθώς και ένα χρονικό διάστημα, αν είναι ένα περιοδικό αποτέλεσμα.

Το *QFeedbackFileEffect* είναι haptic tune το οποίο είναι αποθηκευμένο σε ένα

αρχείο που μπορεί να παιχθεί ξανά. Αυτά τα τραγούδια θα πρέπει να δημιουργηθεί από μία εφαρμογή τρίτου που μπορεί να δημιουργήσει αρχεία μελωδίας που είναι κατανοητά από τον *QFeedbackFileInterface* plugin το οποίο χρησιμοποιείται για να παίξει πίσω το αποτέλεσμα στη συσκευή.

Στις περισσότερες περιπτώσεις χρήσης, οι *QFeedbackEffect*, *QFeedbackHapticsEffect* και *QFeedbackFileEffect* κλάσεις είναι οι μόνες κατηγορίες που οι προγραμματιστές τις χρησιμοποιούν στα προγράμματά τους. Υπάρχουν κάποιες ειδικές περιπτώσεις, ωστόσο, εφόσον επιθυμούν να εφαρμόσουν οι προγραμματιστές το δικό τους plugin ανατροφοδότησης. Το Qt Mobility Feedback API επιτρέπει σε third party προγραμματιστές την εφαρμογή *QFeedbackInterface* σε ένα plugin που μπορεί να φορτωθεί από το plugin μηχανισμό φόρτωσης για να παιχθούν τα εφέ ανάδρασης.

Σημειώστε ότι σε οποιαδήποτε εφαρμογή, μόνο η υψηλής προτεραιότητας plugin για καθένα από τα τρία είδη της ανατροφοδότησης θα χρησιμοποιηθούν για την παροχή feedback effects εκτός από τα plugins ανάδρασης αρχείου (όπου μόνο η υψηλής προτεραιότητα plugin για κάθε υποστηριζόμενη mime-type χρησιμοποιείται). Οι κλάσεις διεπαφής που πρέπει να εφαρμοστούν από ένα plugin για την παροχή feedback effects στους πελάτες είναι οι εξής:

- *QFeedbackHapticsInterface* για *QFeedbackHapticsEffect*
- *QFeedbackFileInterface* για *QFeedbackFile*
- *QFeedbackThemeInterface* για το ενσωματωμένο εφέ θέματος (*QfeedbackEffect::ThemeEffect*).

Σε γενικές γραμμές, η συσκευή θα δώσει στην εφαρμογή του καθενός από τα τρία διαφορετικά interfaces, που θα είναι η προεπιλεγμένη plugins που χρησιμοποιούνται για τα διάφορα *QFeedbackEffects*. Πρέπει να σημειωθεί ότι τα plugin του API αυτού είναι υπό ανάπτυξη και υπόκεινται σε αλλαγές συνεχώς.

4.7 Location (Τοποθεσία)

Ένας από τους πιο γεμάτο υποσχέσεις mobile computing. Δεδομένου ότι η τοποθεσία είναι όλο και πιο σημαντικό κομμάτι του framework για πολλές

εφαρμογές, πολλές φορητές συσκευές παρέχουν πρόσβαση σε πληροφορίες εντοπισμού θέσης. Στην περίπτωση του Qt, το API αυτό είναι επίσης μια σημαντική mobile επέκταση.

Το Location API συμπυκνώνει βασικές γεωγραφικές πληροφορίες για τον χρήστη που προέρχονται από δορυφόρο ή από άλλες πηγές, συμπεριλαμβανομένων γεωγραφικό πλάτος και μήκος, που φέρουν, την ταχύτητα και το υψόμετρο. Επιτρέπει μια σειρά γεωγραφικών εφαρμογών όπως χάρτες. Οι πληροφορίες που παρέχονται περιλαμβάνουν τα εξής:

- Η ημερομηνία και η ώρα κατά την οποία η θέση έχει αναφερθεί.
- Η ταχύτητα της συσκευής που ανέφερε τη θέση της.
- Το υψόμετρο των υποβαλλόμενων θέση (ύψος πάνω από την επιφάνεια της θάλασσας).
- Ο θέση της συσκευής σε μοίρες, σε σχέση με το πραγματικό βορρά.

Οι πηγές δεδομένων, αυτού του API, που δημιουργούνται με τη δημιουργία της υπο-κλάσης *QGeoPositionInfoSource* και παρέχοντας αντικείμενα *QGeoPositionInfo* μέσω του *QGeoPositionInfoSource::positionUpdated()* σήματος. Οι πελάτες που απαιτούν δεδομένα που σχετίζονται με την τοποθεσία μπορούν να συνδεθούν στο *positionUpdated()* σήμα και να καλέσουν την μέθοδο *startUpdates()* ή *requestUpdate()* για να προκαλέσει την κατανομή των δεδομένων θέσης. Οι κυριότερες κλάσεις του location API είναι:

- *QGeoAreaMonitor*: Επιτρέπει την ανίχνευση των αλλαγών εγγύτητας για ένα συγκεκριμένο σύνολο συντεταγμένες.
- *QGeoCoordinate*: Καθορίζει τη γεωγραφική θέση στην επιφάνεια της Γης.
- *QGeoPositionInfo*: Περιέχει πληροφορίες που συγκεντρώνονται σε παγκόσμια θέση, κατεύθυνση και ταχύτητα σε μια συγκεκριμένη χρονική στιγμή.
- *QGeoPositionInfoSource*: Abstract κλάση βάσης για την κατανομή των θέσης ενημερώσεις.
- *QGeoSatelliteInfo*: Περιέχει βασικές πληροφορίες σχετικά με κάποιον δορυφόρο.
- *QGeoSatelliteInfoSource*: Abstract κλάση βάσης για την κατανομή των ενημερώσεων πληροφοριών μέσω δορυφόρου.

4.7.1 Τοποθεσία χρήστη και συνεχής ενημέρωση

Στο ακόλουθο παράδειγμα κώδικα παίρνει τις μεταβολές της θέσης του χρήστη. Η κλάση *QGeoPositionInfoSource* χρησιμοποιείται για να πάρουμε την τοποθεσία της συσκευής. Εάν η πηγή υφίσταται, μπορεί να χρησιμοποιήσει είτε τα δεδομένα μέσω δορυφόρου ή κάποια άλλη μέθοδο εντοπισμού θέσης. Η κλάση χρήστη πρέπει να δημιουργήσει τις ενημερωμένες εκδόσεις κώδικα και να ζητήσει την κλήση της μεθόδου *startUpdates()*. Οι πληροφορίες εντοπισμού θέσης στη συνέχεια τελικά θα περάσει από την μέθοδο *positioningUpdated()* με μια παράμετρο που περιέχει το γεωγραφικό μήκος και πλάτος, υψόμετρο, κλπ., όπως πληροφορίες για τη θέση. Αυτό γίνεται με τον ακόλουθο κώδικα:

```
// Get the default positioning source. If exists, then request updates.

QGeoPositionInfoSource *source = QGeoPositionInfoSource::
createDefaultSource();

if (source){
    connect(source, SIGNAL(positionUpdated(QGeoPositionInfo)), this,
SLOT(handlePositionUpdated(QGeoPositionInfo)));

    source->startUpdates();}

// Custom slot for getting the updates

void handlePositionUpdated(const QGeoPositionInfo &info)

{

    double latitude = info.coordinate().latitude();

    double longitude = info.coordinate().longitude()

double altitude = info.coordinate().altitude();

};
```

4.8 Messaging (Μηνύματα)

Μια κοινή διεπαφή για τον χειρισμό των SMS, MMS και μηνυμάτων ηλεκτρονικού ταχυδρομείου που δίνεται από την ανταλλαγή μηνυμάτων μέσω αυτού του API. Το API παρέχει πρόσβαση σε πολυάριθμες δραστηριότητες που

συνδέονται με την ανταλλαγή μηνυμάτων. Επιτρέπει υπηρεσίες μηνυμάτων για την αναζήτηση και ταξινόμηση μηνυμάτων, κοινοποίηση μεταβολών στα μηνύματα, αποθήκευση, αποστολή μηνυμάτων με ή χωρίς συνημμένα, ανάκτηση δεδομένων από μηνύματα και να ξεκινήσει η προτιμώμενη εφαρμογή μηνυμάτων του πελάτη, είτε για να εμφανισθεί ένα υπάρχον μήνυμα, ή για να συνδεθεί με ένα μήνυμα.

4.8.1 Δημιουργία και αποστολή ενός email μηνύματος

Η δημιουργία του μηνύματος είναι πολύ απλή. Το *QMessage* αποτελεί το αντικείμενο μήνυμα, το οποίο μπορεί να είναι διαφόρων τύπων, όπως ένα μήνυμα ηλεκτρονικού ταχυδρομείου, MMS ή μήνυμα SMS. Τα απαιτούμενα πεδία μήνυμα, το σώμα, τα συνημμένα και τα άλλα δεδομένα που μπορούν να προστεθούν εξαρτάται από τον τύπο του μηνύματος. Το ακόλουθο απόσπασμα κώδικα δημιουργεί ένα νέο μήνυμα email και το στέλνει σε μια καθορισμένη διεύθυνση ηλεκτρονικού ταχυδρομείου. Αυτό γίνεται με την κλάση *QMessageServiceAction*, η οποία μπορεί επίσης να χρησιμοποιηθεί για την ανάκτηση των μηνυμάτων και της διαβίβασης στοιχείων και άλλων κατάλληλων μηνυμάτων που σχετίζονται με messaging-related δράσεις:

```
QMessageServiceAction* serviceAction = new QMessageServiceAction( this );  
  
// Create a new email message  
  
QMessage message;  
  
message.setType(QMessage::Email);  
  
// Add required fields  
  
message.setTo(QMessageAddress("elgeor@it.teithe.gr",QMessageAddress::Email));  
  
message.setSubject("Pictures from our holidays :)");  
  
// Set message body  
  
message.setBody("Καλημέρα!");  
  
// Add attachments  
  
QStringList attachments;  
  
attachments.append("Picture1.jpg");
```

```
message.appendAttachments(attachments);  
  
// Send the message  
  
serviceAction->send(message);
```

4.9 Multimedia (Πολυμέσα)

Τα πολυμέσα έχουν γίνει ένα τυποποιημένο χαρακτηριστικό γνώρισμα στα κινητά τηλέφωνα. Το Qt Mobility API παρέχει πρόσβαση σε μια βιβλιοθήκη πολυμέσων, η οποία προσφέρει έναν εύκολο τρόπο για αναπαραγωγή και εγγραφή ήχου και βίντεο σε διάφορες μορφές. Εκτός από την αναπαραγωγή και την καταγραφή, υπάρχουν και άλλα χαρακτηριστικά που μπορούν επίσης να προσεγγιστούν χρησιμοποιώντας το API αυτό. Για παράδειγμα, το ραδιόφωνο FM μπορεί να χρησιμοποιηθεί μέσα από αυτό το API. Επιπλέον, προβολές διαφανειών μπορεί να γίνει με αυτό το API.

4.9.1 Αναπαραγωγή ενός αρχείου ήχου

Στον ακόλουθο κώδικα γίνεται η αναπαραγωγή ενός MP3 τραγουδιού σε μια απομακρυσμένη ιστοσελίδα. Το API στέλνει σήματα για την πρόοδο της αναπαραγωγής αρχείων πολυμέσων. Το σήμα `positionChanged()` εκπέμπεται με μια παράμετρος που καθορίζει θέση αναπαραγωγής σε χιλιοστά του δευτερολέπτου από την αρχή του αρχείου ήχου. Η διάρκεια μέθοδος `duration()` επιστρέφει το συνολικό χρόνο αναπαραγωγής του αρχείου αναπαραγωγής:

```
QMediaPlayer* mediaPlayer = new QMediaPlayer;  
  
connect(mediaPlayer, SIGNAL(positionChanged(qint64)), this,  
        SLOT(myPositionChangedHandler(qint64)));  
  
mediaPlayer->setMedia(QUrl::fromLocalFile("http://music.com/song.mp3"));  
  
mediaPlayer->setVolume(50);  
  
mediaPlayer->play();
```

4.10 Organizer

Το Organizer API παρέχει στους πελάτες τη δυνατότητα να έχουν πρόσβαση στο ημερολόγιο, το χρονοδιάγραμμα και τα προσωπικά τους δεδομένα σε μια πλατφόρμα-ανεξάρτητη. Αυτό επιτυγχάνεται με τον καθορισμό γενικών

πληροφοριών προσωπικού χαρακτήρα, αφαιρέσεις που μπορεί να περιγραφούν επαρκώς το ημερολόγιο και τον προγραμματισμό των δεδομένων που είναι αποθηκευμένα σε οποιαδήποτε πλατφόρμα. Λόγω της φύσης cross-platform του API, και η δυνατότητα για τους προγραμματιστές να γράψουν ανεξάρτητα από την πλατφόρμα εφαρμογές ενός *QOrganizerItemManager* που μπορεί να ενοποιήσει ένα ή περισσότερες συγκεκριμένες πλατφόρμες, ημερολόγιο ή backends προγραμματισμό, προβλέπεται ότι η σημασιολογία και ιδιορρυθμίες των υποκείμενων αποθηκεύσεις δεδομένων σε οποιαδήποτε πλατφόρμα μπορεί να είναι εντελώς αδιαφανή από τη σκοπιά των Qt-based client cross-platform εφαρμογών.

4.11 Publish & Subscribe

Publish & Subscribe είναι μια ευρέως χρησιμοποιούμενο παράδειγμα ανταλλαγής μηνυμάτων, όπου η επικοινωνία είναι αποσυνδεδεμένη από δύο παραγωγούς (εκδότες) και καταναλωτές (συνδρομητές) των δεδομένων. Η ανακοίνωση πραγματοποιείται ασύγχρονα, πιο συχνά χρησιμοποιώντας ένα πρόσθετο αντικείμενο δεδομένων.

Το API αυτό επιτρέπει τη θεματική πρόσβαση στις πληροφορίες σε μελλοντικές εκδόσεις. Τα δεδομένα είναι οργανωμένα σε μια δομή που μοιάζει ως ένα δέντρο, με τα δεδομένα να μπορούν να γίνουν «σκιά» από άλλα δεδομένα χρησιμοποιώντας το «ίδιο κλειδί», ένα string που προσδιορίζει το φύλλο ή κόμβο. Το πλαίσιο είναι ένα παράδειγμα μιας οντολογίας, σε καθορισμένη δέσμη των σχέσεων. Δεδομένου ότι οι αλλαγές στο πλαίσιο, οι τιμές των αντικειμένων, αλλά και οι σχέσεις παραμένουν οι ίδιες. Προορίζεται να είναι μια τεχνολογία ευρείας διάδοσης.

4.12 Qt Service Framework

Είναι κοινό γνωστό για τις mobile συσκευές ότι παρέχουν συγκεκριμένων device-specific υπηρεσίες. Ως εκ τούτου, μία από τις πιο ενδιαφέρουσες επεκτάσεις κινητών τηλεφώνων του Qt είναι μια πλατφόρμα-ανεξάρτητη μέθοδος για να ανακαλύπτει αυτές τις υπηρεσίες. Εντός του πεδίου εφαρμογής του Qt Mobility API, στο service framework του API, το οποίο ορίζει ένα ενιαίο τρόπο για

την εξεύρεση, την εφαρμογή και την πρόσβαση σε υπηρεσίες σε πολλαπλές πλατφόρμες. Όσον αφορά το framework, η υπηρεσία είναι ένα ανεξάρτητο στοιχείο που επιτρέπει στον πελάτη να εκτελεί μια σαφώς καθορισμένη λειτουργία.

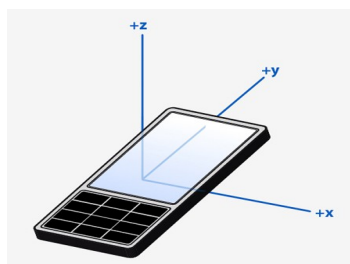
Οι υπηρεσίες που υλοποιούνται ως plug-ins εγκατεστημένα στη συσκευή για να μπορεί να κοιτάξει για τη στήριξη των εξωτερικών υπηρεσιών που εκτελείται σε έναν κεντρικό server. Επιπλέον, επειδή το service framework είναι ουσιαστικά μια abstraction layer, η αίτηση δεν χρειάζεται να ασχολείται με το υποκείμενο πρωτόκολλο, όπου οι servers είναι, τις ιδιαιτερότητες του υλικού με τη δικτύωση και άλλες λεπτομέρειες χαμηλού επιπέδου.

4.13 Sensors (Αισθητήρες)

Το API για τους αισθητήρες παρέχει πρόσβαση στους αισθητήρες. Αυτό καλύπτει τόσο του υψηλού επιπέδου αισθητήρες, όπως ο προσανατολισμός της οθόνης (πορτρέτο, τοπίο) και του χαμηλού επιπέδου σε πραγματικό χρόνο αισθητήρες, όπως δεδομένα που έχουν σχέση με το επιταχυνσιόμετρο (accelerometer).

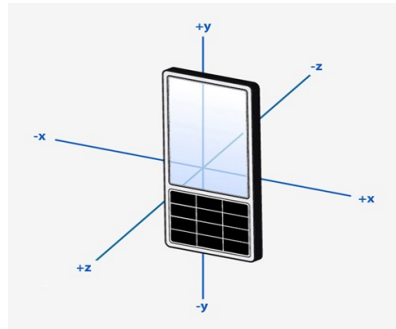
4.13.1 Κοινές συμβάσεις

Εάν δεν ορίζεται διαφορετικά, οι αισθητήρες πρέπει να χρησιμοποιούν το δεξιό χέρι καρτεσιανό σύστημα συντεταγμένων.



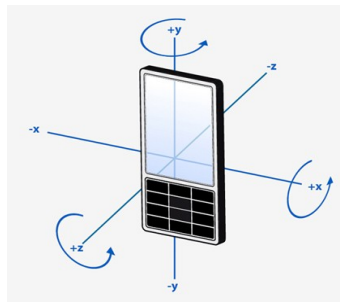
*Εικόνα 4.13.α:
Καρτεσιανό σύστημα
συντεταγμένων*

Για να καταστεί δυνατή για τις μετρήσεις σε όλες τις 6 κατευθύνσεις, χρησιμοποιούνται αρνητικές τιμές.



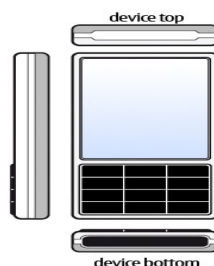
Εικόνα 4.13.1β: 6 κατευθύνσεις

Όταν υπάρχει περιστροφή γύρω από έναν άξονα, η περιστροφή πρέπει να εκφράζεται ως περιστροφή του δεξιού χεριού.



Εικόνα 4.13γ: Περιστροφή γύρω από έναν άξονα

Σε γενικές γραμμές, τα δεδομένα του αισθητήρα είναι προσανατολισμένη προς το επάνω μέρος της συσκευής. Οι τιμές που θα εμφανίζονται στην οθόνη μπορεί να χρειαστεί να μετατραπούν έτσι ώστε να ταιριάζουν με τον προσανατολισμό του UI. Ένας αισθητήρας μπορεί να καθορίζει τα δεδομένα του όταν υπάρχει περιστροφή του UI.



Εικόνα 4.13δ: Όψη μιας συσκευής

4.14 System Information (Πληροφορίες συστήματος)

Μέσα σε κάθε mobile συσκευή, υπάρχουν πολλές πληροφορίες σχετικά με το ίδιο. Για παράδειγμα, πληροφορίες που περιλαμβάνουν λεπτομέρειες όπως το λογισμικό, την συνδεσιμότητα, τα χαρακτηριστικά του υλικού, και ούτω καθεξής. Για τον προγραμματιστή, αυτή η πληροφορία είναι πολύ σημαντική, δεδομένου ότι οι αιτήσεις μπορεί να χρειαστούν ορισμένα χαρακτηριστικά του υλικού, έτσι ώστε να είναι χρήσιμες, ή να ζητούν πληροφορίες σχετικές με το σύστημα, προκειμένου να τους προσαρμοστεί το κάθε πρόγραμμα.

Το API αυτό παρέχει πρόσβαση σε πληροφορίες του συστήματος και απευθύνονται στα ακόλουθα:

Version (Έκδοση): περιέχει πληροφορίες για μια σειρά από το λογισμικό υποστήριξης που σχετίζεται με τη συσκευή. Για παράδειγμα, οι πληροφορίες είναι διαθέσιμες από το λειτουργικό σύστημα και το firmware για την έκδοση του WebKit, Qt και το πλαίσιο παροχής υπηρεσιών.

Features (Χαρακτηριστικά): τα χαρακτηριστικά γνωρίσματα που περιλαμβάνουν υποσυστήματα, όπως η κάμερα, τα Bluetooth, το GPS, το ραδιόφωνο FM, και ούτω καθεξής.

Network (Δίκτυο): διατηρεί πληροφορίες σχετικά με τη σύνδεση δικτύου (π.χ. διεύθυνση MAC) και τον τύπο του δικτύου, όπως το GSM, CDMA, WCDMA, WiFi, Ethernet και άλλα.

Display Information (Εμφάνιση πληροφοριών): παρέχει πρόσβαση σε απεικόνιση πληροφοριών (φωτεινότητα/ βάθος χρώματος) από το σύστημα.

Storage Information (Πληροφορίες αποθήκευσης): πληροφορίες που περιγράφουν αναλυτικά την παρουσία των διαφόρων συσκευών αποθήκευσης, όπως εσωτερικές, αφαιρούμενη, CD-ROM ή ακόμη και αν δεν υπάρχουν συσκευές αποθήκευσης.

Device Information (Πληροφορίες συσκευής): παρέχει πρόσβαση σε πληροφορίες για τη συσκευή από το σύστημα.

Battery Status (Κατάσταση μπαταρίας): δίνει την κατάσταση της ενέργειας της μπαταρίας σε συγκεκριμένα επίπεδα.

Power State (Ισχύς): λεπτομέρειες πώς το τηλέφωνο αυτή τη στιγμή δουλεύει και αν είναι φορτισμένο ή όχι.

Profile (Προφίλ): επιτρέπει στον προγραμματιστή να ελέγχει τις ρυθμίσεις του προφίλ με αυτό το API για αθόρυβη, με δόνηση , η κανονική και άλλους τρόπους.

SIM: υποδηλώνει την παρουσία μιας κάρτα SIM, διπλής κάρτας SIM ή κλειδωμένη κάρτα SIM που να μπορεί να ανακτηθεί.

Input Methods (Μέθοδοι εισαγωγής): καθορίζει το είδος της μεθόδου εισαγωγής, όπως keys/buttons, πληκτρολόγιο QWERTY, οθόνη αφής και μόνο, ή multi-touch.

Screensaver (Προστασία οθόνης): παρέχει πρόσβαση στην προφύλαξη οθόνης και απόφραξης.

4.15 Telephony Events (Γεγονότα του τηλεφώνου)

Η βιβλιοθήκη που ασχολείται με τα γεγονότα του τηλεφώνου παρέχει ένα σύνολο λειτουργιών για την πρόσβαση σε υπάρχουσες κλήσεις και τις πληροφορίες τους .Χρησιμοποιώντας αυτό το API ο προγραμματιστής είναι σε θέση να κάνει ερωτήματα για τη τρέχουσα κατάσταση τηλεφωνίας,π.χ. δεν υπάρχει κλήση, ήχους, κλήση σε εξέλιξη και αν η κλήση είναι σε αναμονή. Τα σήματα, υπό την προϋπόθεση να ενημερώσει τις υπόλοιπες εφαρμογές για τις αλλαγές της κλήση, π.χ. Κουδούνισμα, να μπει σε αναμονή, τερματισμός της κλήσης. Η εφαρμογή μπορεί να ρωτήσει πληροφορίες, όπως καλέστε χρησιμοποιώντας αυτό το API, π.χ. Τύπο κλήσης, τον αριθμό τηλεφώνου, SIP id (Session Initiation Protocol),σύνδεση με την επαφή του καλούντος. Οι βασικές κλάσεις αυτού του API είναι:

- *QTelephonyCallInfo*: Αντιπροσωπεύει μια κλήση όπως τον τύπο, την κατάσταση και άλλες πληροφορίες της κλήσης.
- *QTelephonyCallList*: Περιέχει μια λίστα των ενεργών κλήσεων. Αυτό το αντικείμενο μπορεί να κοινοποιήσει άλλες εφαρμογές για τις αλλαγές της κλήσης.

4.16 Versit

Το Versit API παρέχει μια βιβλιοθήκη για να μετατρέψει *QContacts* προς και από σε αρχεία vCard, καθώς και για τη μετατροπή *QOrganizerItems* προς και από αρχεία iCalendar.

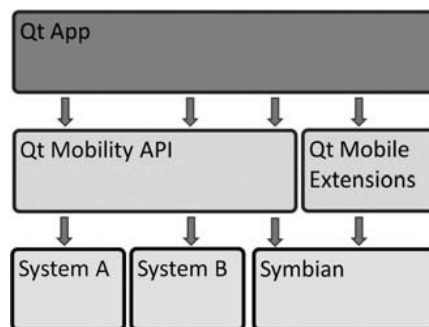
Πιο ιδιαίτερα τα vCard και iCalendar μοιράζονται την ίδια δομή, που εκπροσωπούνται στην abstract μορφή με μια κοινή κλάση, δηλαδή την *QVersitDocument*. Τα vCard και iCalendar αρχεία μπορούν να αναλυθούν σε μορφή *QVersitDocument* χρησιμοποιώντας *QVersitReader*. Τα *QVersitDocument* αντικείμενα μπορούν να γραφτούν σε μια I/O συσκευή, χρησιμοποιώντας την κλάση *QVersitWriter*.

Ένα αντικείμενο *QVersitDocument* μπορεί να αποτελούν είτε το vCard ή το iCalendar. Το *QVersitDocuments* εκπροσωπεί τα vCards μπορεί να μετατραπεί σε τα *QContacts* χρησιμοποιώντας το *QVersitContactImporter*. Ακόμα το *QVersitDocuments* εκπροσωπεί τα iCalendars μπορεί να μετατραπεί σε *QOrganizerItems* χρησιμοποιώντας το *QVersitOrganizerImporter*. Κλείνοντας το *QContacts* και το *QOrganizerItems* μπορούν να μετατραπούν σε *QVersitDocuments* χρησιμοποιώντας *QVersitContactExporter* και *QVersitOrganizerExporter*, αντίστοιχα.

Τέλος, το *QVersitReader* και *QVersitWriter* υποστηρίζουν την ανάγνωση και τη γραφή σε vCard 2.1, 3.0 και σε vCard iCalendar 2,0 μορφή. Πρέπει να σημειωθεί ότι το Organizer API είναι ακόμα υπό ανάπτυξη, έτσι ώστε το Qt Versit κατηγοριών API που σχετίζεται με την εισαγωγή και εξαγωγή ειδών διοργάνωσης μπορεί να αλλάξουν. Κλείνοντας, το Versit ® είναι σήμα κατατεθέν του Internet Mail Consortium.

4.17 Επίλογος

Κλείνοντας και ανακεφαλαιώνοντας σε αυτό το κεφάλαιο είδαμε τα πάντα γύρω από το Qt Mobility και τα APIs του. Η χρήση αυτού είναι το κίνητρο που δίνει στους προγραμματιστές ότι μπορεί να χρησιμοποιηθεί σε οποιοδήποτε σύστημα, όπως φαίνεται στην εικόνα 4.17. Άλλα συστήματα όπως το Windows Mobile ή Maemo μπορεί να μην υποστηρίζουν πλήρως από τα APIs όπως φαίνεται στο πίνακα 4.17, αλλά γίνονται σημαντικά βήματα έτσι ώστε καινούργια APIs να προστεθούν και να μειώσουν – αφανίσουν τις διάφορες ασυμβατότητες.



Εικόνα 4.17: Qt Mobility APIs

Υποστήριξη πλατφόρμας

Χρώμα	Εξήγηση
	Ένα λειτουργικό σύστημα υποστήριξης για το API στην πλατφόρμα είναι πλήρης.
	Ένα λειτουργικό σύστημα υποστήριξης για το API στην πλατφόρμα γίνεται επεξεργασία του, ωστόσο αυτό δεν είναι λειτουργικά πλήρης.
	Ένα λειτουργικό σύστημα υποστήριξης για το API στην πλατφόρμα γίνεται επεξεργασία του. Στο σημείο αυτό απέχει πολύ από την πλήρη μορφή του ή δεν υπάρχει καμία πλατφόρμα με ειδικό κωδικό στο εσωτερικό του πηγαίου κώδικα του QDF.
	Ένα λειτουργικό σύστημα υποστήριξης για το API στην πλατφόρμα στο οποίο δεν έχει δουλέψει ακόμα κανείς. Είναι δυνατό για άλλους να εφαρμόσουν κάτι και να υλοποιηθεί η υποστήριξη

1) Το Qt Mobility API ανάπτυξης εστιάζεται σε Nokia πλατφόρμες. Αυτές οι πλατφόρμες αποτελούν μέρος της συνεχιζόμενης QA, το οποίο περιλαμβάνει μονάδα κάλυψης ελέγχου ανά API και αυστηρό έλεγχο μέσω κοινής ωφελείας τεστ και άλλες διαδικασιών δοκιμής του συστήματος.

Πτυχιακή εργασία του φοιτητή Ελευθέριου Γεωργίου

	API Maturity Level	Tier 1 Platforms					
		S60 3rd Edition, Feature Pack 1	S60 3rd Edition, Feature Pack 2	S60 5th Edition	Symbian	Maemo 5	Harmattan
Service Framework (in-process)	FINAL						
Messaging	FINAL						
Bearer Management	FINAL						
Publish and Subscribe	FINAL						
Contacts	FINAL						
Location	FINAL						
Multimedia	FINAL						
System Information	FINAL						
Sensors	FINAL						
Versit(vCard)	FINAL						
Versit(Organizer)	FINAL						
Camera	FINAL						
Service Framework(OOP)	FINAL						
Organizer	FINAL						
Landmarks	FINAL						
Document Gallery	FINAL	*)	*)	*)			
Maps/Navigation	FINAL						
Feedback	FINAL						

Πίνακας 4.17 (1)

2) Αυτές οι πλατφόρμες δεν είναι ήδη υπό ανάπτυξη ή QA. Εισφορές σε αυτές τις πλατφόρμες είναι ευπρόσδεκτες.

	API Maturity Level	Tier 2 Platforms		
		Windows XP/Vista	Linux	Mac OS X
Service Framework (in-process)	FINAL			
Messaging	FINAL			
Bearer Management	FINAL			
Publish and Subscribe	FINAL			
Contacts	FINAL			
Location	FINAL			
Multimedia	FINAL			
System Information	FINAL			
Sensors	FINAL			
Versit(vCard)	FINAL			
Versit(Organizer)	FINAL			
Camera	FINAL			
Service Framework(OOP)	FINAL			
Organizer	FINAL			
Landmarks	FINAL			
Document Gallery	FINAL			
Maps/Navigation	FINAL			
Feedback	FINAL			

Πίνακας 4.17 (2)

* : Το document gallery backend για S60 3.1, 3.2 και 5.0 δεν είναι ενεργοποιημένο στην προ-χτισμένο πακέτα οφείλεται σε θέματα εξάρτησης βιβλιοθήκης σε κάποιες συσκευές.

Πτυχιακή εργασία του φοιτητή Ελευθέριου Γεωργίου

5

Εφαρμογή : “Πολυλεξικό”

Στο κεφάλαιο αυτό παρουσιάζεται η εφαρμογή “Πολυλεξικό”, MultiDictionary. Θα αναλυθούν θέματα όπως SQLite, Qt (περιβάλλον ανάπτυξης της εφαρμογής), η ανάπτυξη της εφαρμογής, καθώς και τα extension του Symbian OS και του Qt.



5.1 Εισαγωγή στην SQLite

Η SQLite είναι μια in-process βιβλιοθήκη που υλοποιεί ένα αυτοτελές, serverless, χωρίς διαμόρφωση, μηχανισμό συναλλαγής βάσης δεδομένων SQL. Ο κωδικός για της SQLite είναι δωρεάν και είναι ελεύθερο για χρήση για οποιοδήποτε σκοπό, εμπορικό ή ιδιωτικό. Η SQLite συμβαίνει σήμερα σε περισσότερες εφαρμογές από ότι μπορούμε να βασιστούμε.

Ακόμα, η SQLite είναι μια ενσωματωμένη μηχανή βάσης δεδομένων SQL. Αντίθετα από τις περισσότερες άλλες βάσεις δεδομένων SQL, η SQLite δεν έχει μια ξεχωριστή διαδικασία διακομιστή. Έχει την δυνατότητα να διαβάζει και γράφει απευθείας σε συνηθισμένα αρχεία στο δίσκο. Μια πλήρης βάση δεδομένων SQL με πολλούς πίνακες, δείκτες, προκαλεί, και οι απόψεις, περιέχεται σε ένα μοναδικό αρχείο στο δίσκο. Η μορφή του αρχείου βάσης δεδομένων είναι cross-platform - μπορείτε να αντιγράψετε ελεύθερα μια βάση δεδομένων μεταξύ των 32-bit και 64-bit συστήματα.

Τέλος, η SQLite είναι μια συμπαγής βιβλιοθήκη. Με όλα τα χαρακτηριστικά ενεργοποιημένα, το μέγεθος της βιβλιοθήκη μπορεί να είναι μικρότερο από 300KiB, ανάλογα με τις ρυθμίσεις βελτιστοποίησης του compiler. Κλείνοντας, η SQLite είναι δημοφιλής στα κινητά τηλέφωνα, PDAs, και MP3 players. Υπάρχει μια αντίστροφη σχέση μεταξύ της χρήσης μνήμης και της ταχύτητας. Η SQLite τρέχει συνήθως γρηγορότερα τόσο περισσότερη μνήμη σας το δώσει. Παρ' όλα αυτά, οι επιδόσεις είναι συνήθως αρκετά καλές ακόμα και με χαμηλή μνήμη περιβάλλοντα.



5.2 Nokia Qt SDK 1.0

Το SDK που χρειαζόμαστε για να κάνουμε μία εφαρμογή για το Symbian OS είναι το Nokia Qt SDK 1.0 , υπάρχει και η beta έκδοση 1.1. Το Qt SDK είναι ένα εύκολο στην εγκατάσταση και στην χρήση περιβάλλον ανάπτυξης για τους Qt προγραμματιστές εφαρμογών για πλατφόρμες κινητών συσκευών, όπως είναι η Symbian και το Maemo TM. Η εγκατάσταση και ρύθμιση του είναι πολύ εύκολη : Απλά κατεβάστε το αρχείο εγκατάστασης για την πλατφόρμα σας από την σελίδα <http://www.forum.nokia.com/Develop/Qt/> , οι οδηγίες εγκατάστασης βρίσκονται στην σελίδα http://www.forum.nokia.com/Develop/Qt/Getting_started/.

5.2.1 Απαιτήσεις συστήματος

Τα υποστηριζόμενα λειτουργικά συστήματα είναι τα εξής:

- 32 και 64-bit Microsoft Windows XP Service Pack 2, Windows Vista, και Windows 7.
- Apple Mac OS X 10.6 ή πιο πρόσφατο.
- 32 και 64-bit (K)Ubuntu Linux 8.04 ή πιο πρόσφατο, με τα ακόλουθα απαραίτητα στοιχεία:

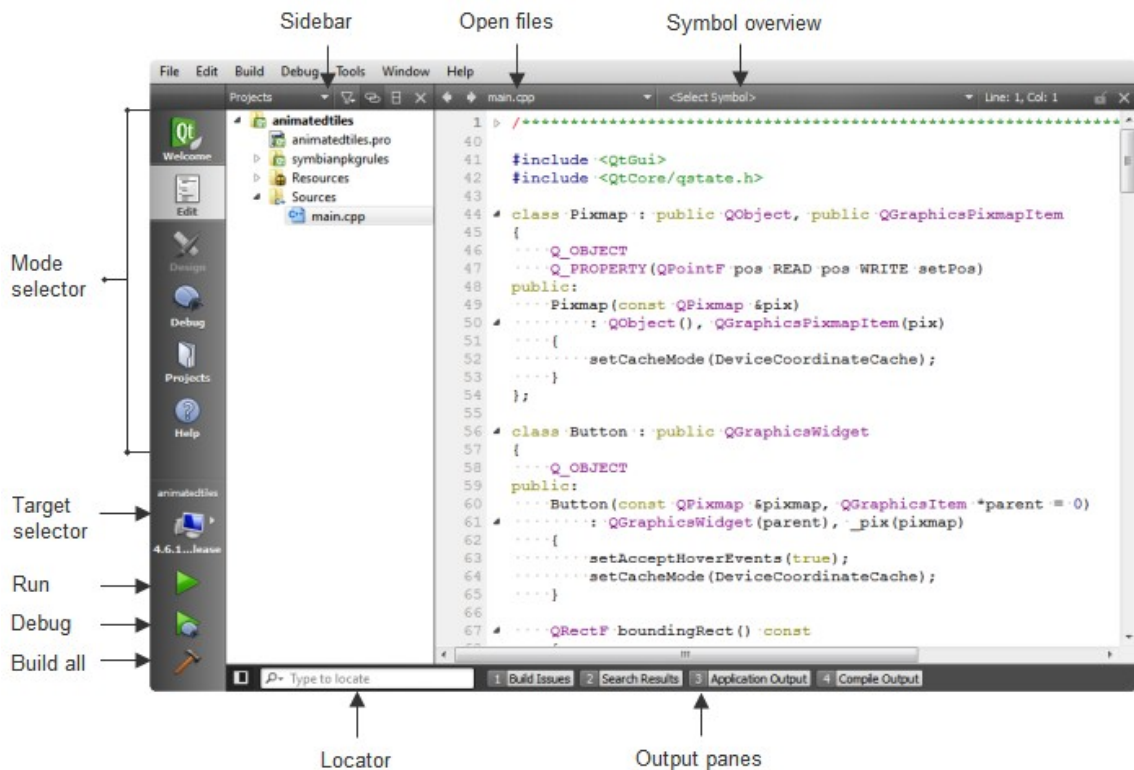
g++	Make
libSM-dev	libxrender-dev
libxext-dev	libfreetype6-dev
libxcursor-dev	libxfixes-dev
libxi-dev	libxrandr-dev
libglib2.0-dev	libfontconfig1-dev
libxft-dev	libx11-dev
	libgl-dev και libglu-dev (QtOpenGL)

Για την εγκατάσταση χρειάζεστε να έχετε περίπου 7GB ελεύθερο χώρο στον σκληρό σας δίσκο.

5.2.2 Περιβάλλον ανάπτυξη εφαρμογών Qt Creator



Εικόνα 5.2.2α: Qt Creator



Εικόνα 5.2.2β: Qt Creator(λεπτομέρειες)

5.2.3 Ρύθμιση περιβάλλοντος ανάπτυξης για Symbian

Hardware και software απαιτήσεις

Τα Windows είναι η μοναδική πλατφόρμα ανάπτυξης για το Symbian που υποστηρίζεται αυτή τη στιγμή. Για την ανάπτυξη και την εκτέλεση εφαρμογών για τη συσκευή, χρειαζόμαστε τα εξής:

- Τους *Nokia USB drivers* που υποστηρίζεται το PC Suite ή το Ovi Suite.
- Το *App TRK* εφαρμογή για τη συσκευή
- Το πακέτο *qt_installer.sis* πρέπει να εγκατασταθεί στη συσκευή, που παραδίδεται με το Qt SDK.
- Το *Qt Mobility APIs* πρέπει να εγκατασταθεί στη συσκευή, εάν θα χρησιμοποιήσει κάτι από αυτά η εφαρμογή.

Εγκατάσταση απαραίτητων εφαρμογών στην συσκευή

Το Nokia Qt SDK δημιουργεί συντομεύσεις για την εγκατάσταση των απαιτούμενων εφαρμογών για τις Symbian συσκευές:

1. Σύνδεση της συσκευή στον υπολογιστή ανάπτυξης με ένα καλώδιο USB στο PC Suite Mode. Εάν δεν έχετε χρησιμοποιήσει προηγουμένως τη συσκευή με το Ovi Suite ή το PC Suite, όλα τα απαραίτητα προγράμματα οδήγησης εγκαθίστανται αυτόματα. Αυτό παίρνει περίπου ένα λεπτό.

2. Επιλογή **Έναρξη>Nokia Qt SDK>Symbian>Install Qt to Symbian device** και ακολουθήστε τις οδηγίες στην οθόνη για να το εγκαταστήσετε στη συσκευή.

3. Επιλογή **Έναρξη>Nokia Qt SDK>Symbian>Install Qt Mobility to Symbian device** και ακολουθήστε τις οδηγίες στην οθόνη για να το εγκαταστήσετε στη συσκευή.

4. Επιλογή **Έναρξη>Nokia Qt SDK>Symbian>Install TRK to Symbian device** και ακολουθήστε τις οδηγίες στην οθόνη για να το εγκαταστήσετε την TRK debugging εφαρμογή στη συσκευή.

Building και running εφαρμογές

Μπορείτε να δοκιμάσουμε την εφαρμογή μας σε μια συσκευή με την κατασκευή και την λειτουργία της από το Qt Creator για Symbian συσκευή.

1. Συνδέστε τη συσκευή στον υπολογιστή ανάπτυξης μέσω ενός καλωδίου USB. Ο επιλογέας εμφανίζεται ένα πράσινο σημάδι ελέγχου όταν μια συσκευή είναι συνδεδεμένη.



Εικόνα 5.2.3:
Συσκευή
συνδεδεμένη

2. Ξεκινήστε το App TRK εφαρμογή στη συσκευή σας και να απενεργοποιήσετε τη σύνδεση Bluetooth.

3. Επιλέξτε Επιλογές για να επιλέξετε USB ως τύπο σύνδεσης.

4. Κάντε κλικ στο κουμπί *Run* στο Qt Creator.

5.3 Περί της εφαρμογής - “Πολυλεξικό”

5.3.1 Περιγραφή της βάσης

Η βάση μου δόθηκε από τον επιβλέπων της πτυχιακής μου κ. Μιχάλη Σαλαμπάση που ήταν σε μορφή Microsoft access και η μόνη “αλλαγή” στην βάση είναι η μετατροπή της σε μορφή SQLite3. Το όνομα του αρχείου είναι MultiDictionary.db, είναι αρχείο τύπου SQLite3 και το πρόγραμμα που χρησιμοποιήθηκε για την διαχείριση της βάσης είναι το *SQLite2009 Pro*, (σελίδα για κατέβασμα του προγράμματος <http://www.sqlite.org/cvstrac/wiki?p=ManagementTools>).

5.3.1.1 Περιγραφή πινάκων της βάσης

Πίνακας lexicons:

lexicons(IDLexicon: INT, LexiconName: TEXT). Ο πίνακας αυτός περιλαμβάνει όλα λεξικά. Το IDLexicon είναι ένας μοναδικός αριθμός που αναφέρεται στο λεξικό, και το LexiconName είναι το όνομα, ή η περιγραφή του λεξικού.

Πίνακας KlisiHTML:

KlisiHTML(idKlisi: INT, Counter: INT, klisi: TEXT). Ο πίνακας αυτός περιλαμβάνει όλους τους κλιτικούς πίνακες των λεξικών. Το idKlisi είναι ο

μοναδικός αριθμός που αναφέρεται στους κλιτικούς πίνακες. Το πεδίο Counter χρησιμοποιείται για να μπορούμε με συνδυασμό με το idKlisi την δυνατότητα όταν αναφερόμαστε σε ένα ρήμα στην αναζήτησή μας, πχ. “τρέχω”, να μπορώ να πάρω τους κλιτικούς πίνακες του ρήματος που είναι οι 8 χρόνοι, επομένως έχω το ίδιο idKlisi αλλά 8 διαφορετικά Counter. Τέλος, το πεδίο klisi περιέχει τους κλιτικούς πίνακες σε μορφή HTML tables.


Πίνακας interpret:


interpret(id: INT, interpretation: TEXT, klisi: INT, word: TEXT, search: TEXT, LexiconType: INT). Ο πίνακας αυτός περιλαμβάνει όλες τις περιγραφές των λέξεων από όλα τα λεξικά. Το πεδίο id είναι ο μοναδικός αριθμός που αναφέρεται στην κάθε λέξη. Το πεδίο interpretation είναι η περιγραφή της κάθε λέξης. Το πεδίο klisi είναι ο αριθμός που αντιστοιχεί το idKlisi του πίνακα KlisiHTML(δεν είναι ξένο κλειδί τον πίνακα interpret!). Στην συνέχεια, το πεδίο word αποτελεί την λέξη όπως είναι γραμμένη κανονικά, δηλαδή με τόνους, κεφαλαία κτλ. (μας βοηθάει για την σωστή παρουσίαση των λέξεων). Το πεδίο search, όπως λέει και το όνομα του είναι το πεδίο που βασιζόμαστε για να κάνουμε την αναζήτησή μας. Περιέχει τις λέξεις χωρίς τόνους, με μικρούς χαρακτήρες και χωρίς σημεία στίξης. Τέλος, το πεδίο LexiconType προσδιορίζει το id του λεξικού (δεν είναι ξένο κλειδί στο πίνακα interpret!).

5.3.2 Εφαρμογή “Πολυλεξικό” με χρήση του Qt

Σε αυτή την παράγραφο, θα σχολιαστούν, αναλυθούν τα αρχεία του Qt που έχουν να κάνουν με την εφαρμογή και γενικά θα γίνει μία εκτενής αναφορά για την εφαρμογή.

5.3.2.1 Καταλήξεις αρχείων της εφαρμογής του Qt

*.pro  : αρχείο που δημιουργήθηκε με το λογισμικό Qt SDK, το οποίο επιτρέπει στους προγραμματιστές να δημιουργούν cross-platform εφαρμογές, περιέχει όλες τις πληροφορίες που είναι απαραίτητες για να γίνει compile μια εφαρμογή Qt με την εντολή qmake του Qt.

*.ui  : αποθηκεύει τις ρυθμίσεις του user interface για ένα πρόγραμμα, αποθηκεύονται σε μορφή XML και περιέχει τους ορισμούς των Qt widgets με

υποδοχές και τα σήματα, μπορούν να προβληθούν σε ένα βασικό πρόγραμμα επεξεργασίας κειμένου ή ανοιχτούν με ένα πρόγραμμα σχεδίασης UI. Τα UI αρχεία μπορούν να δημιουργηθούν χρησιμοποιώντας το Qt Designer και το Qt Creator, οι οποίες αποτελούν μέρος του Qt SDK.

.qrc: αρχείο XML που χρησιμοποιείται από τα προγράμματα που δημιουργήθηκαν με χρήση του Qt, περιέχει έναν κατάλογο των πόρων αίτησης, όπως τα αρχεία εικόνας. Ακόμα περιλαμβάνεται ως μέρος της δέσμης της εφαρμογής όταν η εφαρμογή Qt είναι built. Οι Qt εφαρμογές χρησιμοποιούν τα QRC αρχεία για να εντοπίσουν τους πόρους για την εφαρμογή κατά τη διάρκεια της εκτέλεσης. Τα μονοπάτια των αρχείων που αναφέρεται σε ένα αρχείο QRC είναι σχετικά με τον κατάλογο στον οποίο βρίσκεται το αρχείο QRC.

.h: header αρχείο, αναφέρεται σε ένα C++ έγγραφο πηγαίο κώδικα, μπορεί να περιέχει μεταβλητές, σταθερές, και λειτουργίες που χρησιμοποιούνται από άλλα αρχεία στο πλαίσιο ενός σχεδίου προγραμματισμού. Επίσης με επιτρέπει να χρησιμοποιούνται συχνά και να εγγράφονται μόνο μία φορά και αναφέρεται από άλλα αρχεία προέλευσης, όταν χρειάζεται.

.cpp: πρόγραμμα του πηγαίο κώδικα γραμμένο σε C++, μπορεί να είναι ένα αυτόνομο πρόγραμμα ή ένα από τα πολλά αρχεία που αναφέρεται στο πλαίσιο ενός σχεδίου προγραμματισμού, μπορεί να αντιμετωπιστεί με ένα πρόγραμμα επεξεργασίας κειμένου, αλλά τα περισσότερα προγράμματα ανάπτυξης λογισμικού θα εμφανίσει το κείμενο με τις χρήσιμες επισήμανση σύνταξης.

Τώρα, στην εφαρμογή μας στο MultiDicrionary.pro υπάρχουν τα εξής:

```
1 QT += sql core gui webkit
2 TARGET =
3 TEMPLATE = app
4HEADERS += About.h Introduction.h LexiconForm.h
5 FORMS += About.ui Introduction.ui LexiconForm.ui
6 SOURCES += About.cpp Introduction.cpp LexiconForm.cpp main.cpp
7 RESOURCES += MultiDictionary.qrc
8 CONFIG += mobility
```

```
9 MOBILITY =
10 symbian{
    ICON = dictionaryIcon.svg
    TARGET.UID3 = 0xe1d36f4e
    # TARGET.CAPABILITY +=
    TARGET.EPOCSTACKSIZE = 0x14000
    TARGET.EPOCHEAPSIZE = 0x20000 0x1000000 #Min 128kb, Max 16Mb}
```

Γραμμή 1: QT, Qt-specific configuration επιλογές. Τα στοιχεία που είναι μετά το ίσον (=) αναφέρονται στα διάφορα modules που χρησιμοποιήθηκαν στην εφαρμογή. Από default μπαίνουν το core και gui, το μεν core αναφέρεται στο QtCore , ενώ το δε gui αναφέρεται QtGui module. Το sql αναφέρεται στο QSql module, μιας χρησιμοποιήθηκαν κλάσεις της Qt SQL βιβλιοθήκης για την δυνατότητα εμφάνισης αποτελεσμάτων από την βάση. Τέλος, το webkit αναφέρεται στο QtWebKit module, και χρησιμοποιήθηκε για να παρουσιαστούν τα αποτελέσματα που έχουν να κάνουν με τον πίνακα KlisiHTML (περισσότερες λεπτομέρειες θα αναφερθούν παρακάτω).

Γραμμή 4: Περιλαμβάνει τα αρχεία headers που χρησιμοποιεί η εφαρμογή.

Γραμμή 5: Περιλαμβάνει τα αρχεία ui που χρησιμοποιήθηκαν στην εφαρμογή.

Γραμμή 6: Περιλαμβάνει τα αρχεία cpp που χρησιμοποιεί η εφαρμογή.

Γραμμή 7: Περιλαμβάνει τα αρχεία qrc και αποτελεί στην ουσία τους πόρους που χρησιμοποιεί η εφαρμογή, όπως εικόνες της εφαρμογής.

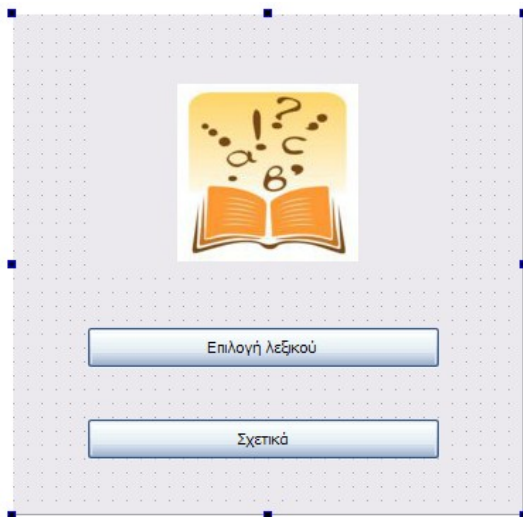
Γραμμή 8: Γενικά configuration για την εφαρμογή μας. Το mobility μας δείχνει ότι η εφαρμογή χρησιμοποιεί το Qt Mobility.

Γραμμή 9: Περιλαμβάνει τι από το Qt Mobility χρησιμοποιεί. Για παράδειγμα αν μετά το ίσον (=) υπήρχε το *systeminfo* θα σήμαινε ότι η εφαρμογή μας χρησιμοποιεί πληροφορίες της mobile συσκευής.

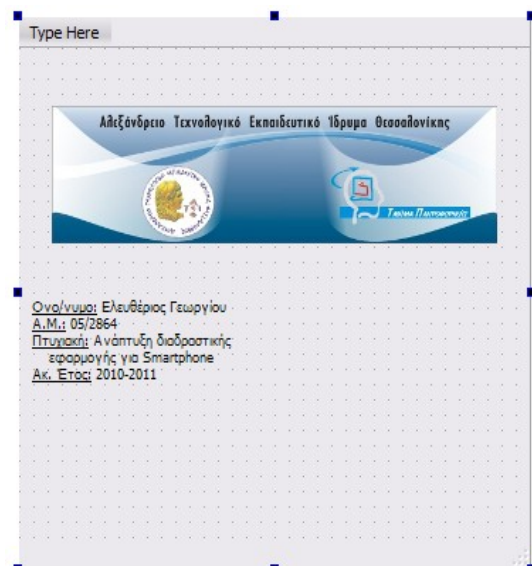
Γραμμή 10: Περικλείει πληροφορίες που έχουν να κάνουν καθαρά με την mobile συσκευή symbian. Στον tag ICON γράφουν το αρχείο .svg που θα αποτελέσει και η εικόνα της εφαρμογής μας. Το αρχείο svg είναι τα αρχικά των λέξεων **Scalable Vector Graphics**, και η ιδανική αναλογία του εικονιδίου αυτού είναι 44x44. Το TARGET.UID3 αποτελεί το μοναδικό αναγνωριστικό για την εφαρμογή

μας έτσι ώστε να γίνει αντιληπτό από το κινητό μας που χρησιμοποιεί Symbian OS. Εάν η τιμή UID3 δεν ορίζεται, τότε μια τυχαία τιμή από το debug φάσμα χρησιμοποιείται, δηλαδή από 0xE0000000 μέχρι το 0xEFFFFFFF. Πρέπει να σημειωθεί ότι η προεπιλεγμένη τιμή είναι μόνο για πολύ περιορισμένη διανομή. Θα πρέπει να αποκτηθεί ένα μοναδικό UID από τη Symbian για να είναι σε θέση να υπογράψει την αίτηση για περαιτέρω διανομή. Τέλος, τα TARGET.EPOCSTACKSIZE και TARGET.EPOCSTACKSIZE ορίζουν το μέγεθος της heap και stack.

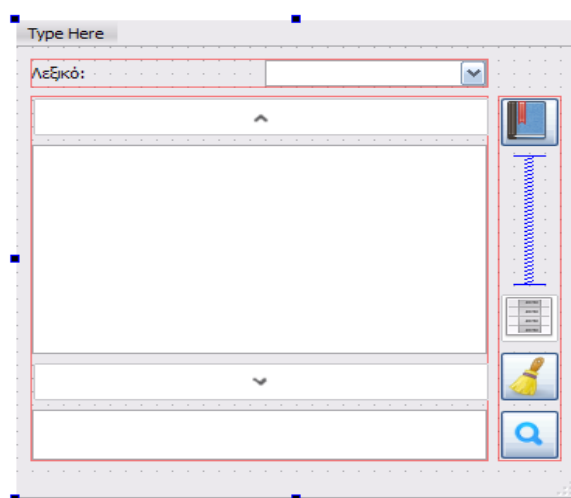
Στην συνέχεια θα παρουσιαστούν τα αρχεία ui που χρησιμοποιήθηκαν στην εφαρμογή μας σε design mode. Και είναι οι ακόλουθες εικόνες:



Εικόνα 5.3.2.1α: Introduction.ui



Εικόνα 5.3.2β: About.ui



Εικόνα 5.3.2.1γ: LexiconForm.ui

Introduction.ui: Αποτελείται από *QLabel* που περιέχει την εικόνα και δύο *QPushButton*. Στο action clicked() το *QPushButton* “Επιλογή λεξικού” εμφανίζεται το *LexiconForm.ui* και με το *QPushButton* “Σχετικά” εμφανίζεται το *About.ui*.

LexiconForm.ui: Αποτελείται από τις εξής λειτουργίες(*QPushButton*)

- Επιλογή λεξικού:



- Προβολή κλιτικών πινάκων:



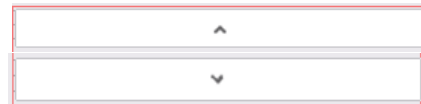
- Καθαρισμός:



- Αναζήτηση:



- Scroll προς τα πάνω και προς τα κάτω στην λίστα των λέξεων:



Ακόμα αποτελείται από μια *QListWidget* που είναι το κεντρικό άσπρο πλαίσιο όπου εκεί θα εμφανίζονται τα αποτελέσματα της αναζήτησης. Επίσης έχει και ένα *QComboBox* όπου θα εμφανίζονται τα διαθέσιμα λεξικά.

About.ui: Αποτελείται από δύο *QLabel* ένα όπου είναι η εικόνα και το άλλο που είναι τα διάφορα στοιχεία της εφαρμογής.

5.3.2.2 Κάποια κομμάτια κώδικα (επεξήγηση)

Σύνδεση με την βάση MultiDictionary.db:

```
1void LexiconForm::setupModel() {  
2    dir = new QDir();  
3#if defined(Q_WS_S60)  
4    // For mobile  
5    dir->setPath("E:");  
6#else  
7    // For PC
```



```
8  dir->setPath("C:/");
9#endif
10 QString path(dir->path().append(QDir::separator()).append("MultiDictionary.db"));
11 path = QDir::toNativeSeparators(path);
12 db = QSqlDatabase::addDatabase("QSQLITE");
13 db.setDatabaseName(path);
14 if (!db.open()) {
15     QMessageBox::critical(0, errors,errorMesS, QMessageBox::Ok);
16     return;
17 }
18 model = new QSqlRelationalTableModel(this);
19 model->setTable("interpret");
20 model->setSort(6,Qt::AscendingOrder);
21 lexiconType = model->fieldIndex("LexiconType");
22 model->setRelation(lexiconType,QSqlRelation("lexicons", "IDLexicon", "LexiconName"));
23 model->select();
}
```

Γραμμή 2: Δημιουργία ενός καινούργιου Directory.

Γραμμή 3: Αυτό το if defined με το # (κάγκελο) μπροστά μπορεί να ορίσει κανείς λειτουργίες που αντιστοιχούν σε κάτι συγκεκριμένο. Για παράδειγμα Q_WS_S60 αναφέρομαι σε μία συσκευή Symbian S60 και η default λειτουργία αυτής είναι ότι μιλάμε για H/Y, όπως φαίνεται στην γραμμή 6. Ο driver E: αντιστοιχεί στον αφαιρούμενο δίσκο της συσκευής, ο driver C: αντιστοιχεί στην μνήμη του κινητού.

Γραμμή 10: Ορίζεται το full path της βάσης MultiDictionary.db.

Γραμμή 11: Η μέθοδος *toNativeSeparatos()* επιστέφει για παράδειγμα στα windows *toNativeSeparatos(c:/ptyxiakh/MultiDictionary.db)* θα επιστρέψει "c:\ptyxiakh\MultiDictionary.db".

Γραμμή 12: Ορίζεται τον driver της βάσης μου. Η συγκεκριμένη εφαρμογή χρησιμοποιεί τον SQLite, επομένως ο αντίστοιχος driver είναι ο *QSQLITE*.

Γραμμή 13: Ορίζεται το όνομα (διαδρομή) της βάσεις που θα χρησιμοποιηθεί.

Γραμμή 14: Γίνεται έλεγχος αν μπορεί να υπάρξει connection με την βάση. Αν για κάποιο λόγο δεν υπάρξει σύνδεση εμφανίζεται ένα μήνυμα που ενημερώνει τον χρήστη για το πρόβλημα, αλλιώς γίνεται η σύνδεση με την βάση.

Γραμμή 18: Η κλάση *QsqlRelationTableModel* μας επιτρέπει να χρησιμοποιήσουμε ένα πίνακα ως επεξεργάσιμο μοντέλο δεδομένων, και μας δίνει την δυνατότητα να εφαρμόσουμε και τα ξένα κλειδιά (foreign keys).

Γραμμή 19: Ορίζουμε στο μοντέλο μας τον πίνακα *interpret*.

Γραμμή 20: Γίνεται ταξινόμηση σύμφωνα με έκτη στήλη του πίνακα σε αύξουσα σειρά.

Γραμμή 21: Στην μεταβλητή *lexiconType* τύπου *int* δύνουμε την τιμή που αντιστοιχεί στο πεδίο του πίνακα της στήλης *LexiconType*.

Γραμμή 22: Δημιουργία σχέσης μεταξύ του πίνακα *interpret* και *lexicons*, στην ουσία η γραμμή αυτή μας λέει ότι το *lexiconType* είναι ξένο κλειδί στον πίνακα *interpret* και αντιστοιχίζεται με το *IDLexicon* του πίνακα *lexicons* και αυτή η “σχέση” μου επιστρέφει το *LexiconName*.

Γραμμή 23: Κάνει *select* τον πίνακα που ορίστηκε στην γραμμή 19 και επιστρέφει *true* αν είναι επιτυχής, αλλιώς *false*.

Εισαγωγή λεξικών σε *QComboBox*:

```
1 QSqlTableModel *relModel = model->relationModel(lexiconType);  
2 lex_ui->lexTypeQComboBox->setModel(relModel);  
3 lex_ui->lexTypeQComboBox->setModelColumn(relModel->fieldIndex("LexiconName"));
```

Γραμμή 1: Ορίζουμε ένα μοντέλο το οποίο αναφέρεται στην στήλη *lexiconType* του πίνακα *interpret*.

Γραμμή 2: Ορίζεται στο *QComboBox* το μοντέλο που μόλις δημιουργήθηκε.

Γραμμή 3: Ορίζεται ότι το *QComboBox* θα παίρνει τιμές από το πεδίο *LexiconName* του πίνακα *lexicons*.

Αναζήτηση λέξης μέσα από ένα λεξικό:

```
1  if(word != ""){
2      createSearchLoader();
3      QVector<uint> t = word.toUcs4();
4      progress->setValue(this->index);
5      QMap<int, int>::const_iterator i = mLexType.constBegin();
6      while (i != mLexType.constEnd()) {
7          if(i.key()==t.value(0)){
8              ibegin = i.value();
9              ++i;
10             iend = i.value();
11             --i;
12         }
13         i++;
14     }
15     query.exec("SELECT a.word, a.interpretation, a.klisi, a.id FROM interpret a INNER JOIN
lexicons l ON a.LexiconType = l.IDLexicon WHERE a.LexiconType = " +
QString::number(lexiconIndex) + " AND a.search LIKE '" + word + "%' AND id BETWEEN"+
QString::number(ibegin)+" AND "+QString::number(iend));
16     if(query.at() == -2){
17         lex_ui->wordsQListWidget->addItem(nodataS);
18     } else{
19         hideMe();
20         while(query.next()){
21             if(this->index == 0){
22                 id_find = query.value(3).toInt();
23             }

```

Πτυχιακή εργασία του φοιτητή Ελευθέριου Γεωργίου

```
24     if(this->index > 50){
25         break;
26     }else{
27         progress->setValue(this->index);
28         map_desc->insert(query.value(0).toString(),query.value(1).toString());
29         map_klisi->insert(query.value(0).toString(),query.value(2).toInt());
30         id_end_find = query.value(3).toInt();
31         this->index++;
32     }
33 }
34 from = id_find;
35 to = id_end_find;
36 QMap<QString, QString>::const_iterator i = map_desc->constBegin();
37 while (i != map_desc->constEnd()) {
38     lex_ui->wordsQListWidget->addItem(i.key());
39     ++i;
40 }
40 progress->setValue(100);
41 lex_ui->moreQToolButton->setEnabled(true);
42 lex_ui->lessQToolButton->setEnabled(true);
43 showMe();
44 }
45 } else {
46     lex_ui->wordsQListWidget->addItem(nodataS);
47 }
```

Γραμμή 1: Έλεγχος αν η λέξη που εισήγαγε ο χρήστης είναι διάφορη του κενού.

Γραμμή 3: Μετατροπή των γραμμάτων της λέξης σε μία λίστα από unicode αριθμούς, πχ το “α” είναι 945.

Γραμμή 5-13: Με βάση ένα map όπου βρίσκεται η αντιστοίχιση κάποιων id με τα unicode των γραμμάτων γίνεται έλεγχος αν υπάρχει το πρώτο στοιχείο της λίστα μέσα σε αυτό το map.

Γραμμή 15: Γίνεται ένα select ερώτημα για να πάρουμε τις απαραίτητες πληροφορίες για την λέξη που αναζήτησε ο χρήστης.

Γραμμή 16: Η μέθοδος `at()` επιστρέφει έναν ακέραιο αριθμό αν αυτός ο αριθμός είναι `-2` (*After the last record.*) σημαίνει ότι το select ερώτημά μας δεν έχει δεδομένα να μας επιστρέψει.

Γραμμή 17: Αφού το ερώτημά μας δεν μας έχει επιστρέψει κάποιο αποτέλεσμα πρέπει να ενημερώσουμε τον χρήστη, επομένως προσθέτουμε ένα στοιχείο στην `QListWidget` του `LexiconForm.ui` ενημερώνοντας τον χρήστη με κατάλληλο κείμενο ότι δεν βρέθηκαν αποτελέσματα.

Γραμμή 18-33: Είναι το σημείο του κώδικα που μας λέει τι συμβαίνει όταν υπάρχουν αποτελέσματα αλλά με κάποιο τρόπο πρέπει να τα πάρουμε.

Γραμμή 20: Γίνεται μια επαναληπτική διαδικασία όσο υπάρχουν αποτελέσματα, αλλά αν τα αποτελέσματα είναι πάνω από 50 διακόπτεται η διαδικασία αυτή βιαίως με την εντολή `break` (γραμμή 25) και συμβαίνουν τα εξής πράγματα:

- Γραμμή 22: Εισαγωγή στην μεταβλητή την τέταρτη τιμή του select ερωτήματος (`a.id`).
- Γραμμή 28: Εισαγωγή σε ένα `map_desc` τις τιμές `a.word` και `a.interpretation`, δηλαδή την πρώτη και δεύτερη τιμή του select ερωτήματος.
- Γραμμή 29: Εισαγωγή σε ένα `map_klisi` τις τιμές `a.word` και `a.klisi`, δηλαδή την πρώτη και τρίτη τιμή του select ερωτήματος.

Γραμμή 36-40: Εισαγωγή των αποτελεσμάτων στην `QListWidget` του `LexiconForm.ui` για να δει τα αποτελέσματα ο χρήστης.

Γραμμή 41,42: Κάνουμε ενεργά τα κουμπιά για να “περπατούμε” μέσα στο λεξικό.

Γραμμή 46: Αν η λέξη που εισάγει προς αναζήτηση ο χρήστης είναι “κενή”,

τότε προσθέτουμε ένα στοιχείο στην *QListWidget* του *LexiconForm.ui* ενημερώνοντας τον χρήστη με κατάλληλο κείμενο ότι δεν βρέθηκαν αποτελέσματα.

Επιλογή λέξης και εμφάνιση περιγραφή της λέξης:

```
1 void LexiconForm::on_wordsQListWdget_itemClicked(QListWidgetItem* item){
2     QString description;
3     QMap<QString, QString>::const_iterator i = map_desc->find(item->text());
4     while (i != map_desc->end() && i.key() == item->text()) {
5         description = i.value();
6         ++i;
7     }
8     showDescription(item->text(),description);
9 }
10 void LexiconForm::showDescription(QString word,QString desc){
11     QMessageBox *messageBox = new QMessageBox();
12     messageBox->setText(word);
13     ...
14     messageBox->setInformativeText(desc+"\n");
15     messageBox->setStandardButtons(QMessageBox::Ok);
16     messageBox->setDefaultButton(QMessageBox::Ok);
17     ...
18     return;
19 }
20 }
```

Γραμμή 1: Κλίση της μεθόδου όταν επιλεγεί ένα στοιχείο από την λίστα (*QListWidget*).

Γραμμή 2-7: Εύρεση της περιγραφής της λέξης μέσα από το *map_desc*.

Γραμμή 8: κλίση της μεθόδου *showDescription(QString, QString)*.

Γραμμή 10-20: Εμφάνιση ενός *QMessageBox* με τίτλο την λέξη και περιγραφή την περιγραφή της λέξης. Έχει ως κουμπί λειτουργίας το OK που στην ουσία μας επιστρέφει στην αρχική μας φόρμα.

Εμφάνιση κλιτικών πινάκων μίας λέξης:

```
1 void LexiconForm::showKlisiTables(QMap<int, QString> *map){
2     web_win = new QFrame();
3     QByteArray web_win_title ="Κλιτικοί πίνακες";
4     QString web_win_titleS = codec->toUnicode(web_win_title);
5     web_win->setWindowTitle(web_win_titleS);
6     web_klisi_QComboBox = new QComboBox();
7     map_web_xronoi = xronos();
8     QMap<int, QString>::const_iterator y = map_web_xronoi.constBegin();
9     while (y != map_web_xronoi.constEnd()) {
10         web_klisi_QComboBox->addItem(y.value());
11         ++y;
12     }
13     view = new QWebView();
14     nextButton = new QPushButton(">>");
15     previousButton = new QPushButton("<<");
16     webLayout = new QGridLayout();
17     indexLabel = new QLabel();
18     web_menu = new QMenu(web_win);
19     webLayout->addWidget(web_klisi_QComboBox, 0, 1, 1, 2);
20     webLayout->addWidget(view, 1, 1, 1, 2);
21     webLayout->addWidget(previousButton, 2, 1, 1, 1);
```

```
22  webLayout->addWidget(nextButton, 2, 2, 1, 1);
23  web_backAction = new QAction(tr("Back"), this);
24  web_backAction->setMenu(m_menu);
25  web_backAction->setSoftKeyRole(QAction::PositiveSoftKey);
26  web_win->connect(web_backAction,SIGNAL(triggered()),this,SLOT(web_goBack()));
27  web_win->addAction(web_backAction);
28  QMap<int, QString>::iterator i = map->find(1);
29  indexLabel->setText(QString::number(i.key()));
30  view->setHtml(i.value());
31  web_klisi_QComboBox->setCurrentIndex(indexLabel->text().toInt() - 1);
32  web_win->setLayout(webLayout);
...
33  web_win->connect(nextButton, SIGNAL(clicked()), this, SLOT(next()));
34  web_win->connect(previousButton, SIGNAL(clicked()), this, SLOT(previous()));
35  web_win->connect(web_klisi_QComboBox,SIGNAL(activated(int)), this,
SLOT(xronos_selection(int)));
...
```

Γραμμή 2: Δημιουργία ενός καινούργιου *QFrame*.

Γραμμή 5: Ορισμός τίτλου στον παράθυρο.

Γραμμή 6: Δημιουργία *QComboBox*.

Γραμμή 8-12: Εισαγωγή τιμών στο *QComboBox*.

Γραμμή 13: Δημιουργία *QWebView*.

Γραμμή 14,15: Δημιουργία *QPushButton*.

Γραμμή 16: Δημιουργία *QGridLayout*.

Γραμμή 18: Δημιουργία *QMenu*.

Γραμμή 19-22: Εισαγωγή των παραπάνω στο *QGridLayout*.

Γραμμή 23: Δημιουργία *QAction*.

Γραμμή 30: Εισαγωγή τιμής στον *QWebView*.

Γραμμή 26, 33-35: Σύνδεση των διαφόρων στοιχείων με κάποιες

Πτυχιακή εργασία του φοιτητή Ελευθέριου Γεωργίου

συγκεκριμένες λειτουργίες, *back()*, *next()*, *previous()*, *xronos_selection(int)*.

5.3.2.3 Η εφαρμογή μέσα από τον προσομοιωτή (Εικόνες)



5.4 Extensions αρχείων για Symbian OS

Όπως αναφέρθηκε λίγο παραπάνω (βλέπε *Building και running εφαρμογές*) μετά από το compile του προγράμματός μας στον φάκελο που βρίσκεται το project μας δημιουργούνται κάποια αρχεία τα οποία είναι αυτά που δίνουν στην εφαρμογή μας τον “αέρα” μιας mobile εφαρμογής για Symbian. Παρακάτω θα περιγραφούν τα διάφορα extension των αρχείων του Symbian που σε άλλη περίπτωση θα βρίσκαμε αν καθόμασταν να γράψουμε μία εφαρμογή από το μηδέν χωρίς την χρήση του περιβάλλοντος του Qt. Τα extensions είναι τα ακόλουθα:

**_reg.rss*: Είναι ένα ειδικό Symbian αρχείο πόρων που περιέχει πληροφορίες σχετικά με την εφαρμογή που απαιτείται από το Symbian για να δει έναν πρέπει να ξεκινήσει μία εφαρμογή ή να ανοίξει κάποιο shell για να κάνει τις απαραίτητες ενέργειες.

**.loc*: Το .loc αρχείο μπορεί να χρησιμοποιηθεί για τον εντοπισμό, εάν η αίτηση πρέπει να υποστηρίξει πολλαπλές γλώσσες.

**.rss*: Ένα standard Symbian αρχείο πόρων. Ένα αρχείο πόρος που χρησιμοποιείται για τον καθορισμό ορισμένων στοιχείων UI, όπως μενού, παραθύρων διαλόγου, εικονίδια εφαρμογών και λεζάντες.

**.inf/*.mmp*: Η συνιστώσα ορισμός (bld.inf) αρχείο, και ο ορισμός του έργου (*.mmp) είναι αρχείο που χρησιμοποιείται για να περιγράψει τα αρχεία που περιέχονται σε ένα project Symbian. Για παράδειγμα, χρησιμοποιούνται κατά την κατασκευή του project από την εντολή prompt. Χρησιμοποιώντας το Qt για Symbian εφαρμογές αυτά τα αρχεία δημιουργούνται αυτόματα από το .pro αρχείο και θα πρέπει επομένως να χρησιμοποιούν το .pro αρχείο. κατά την αλλαγή των ρυθμίσεων του project.

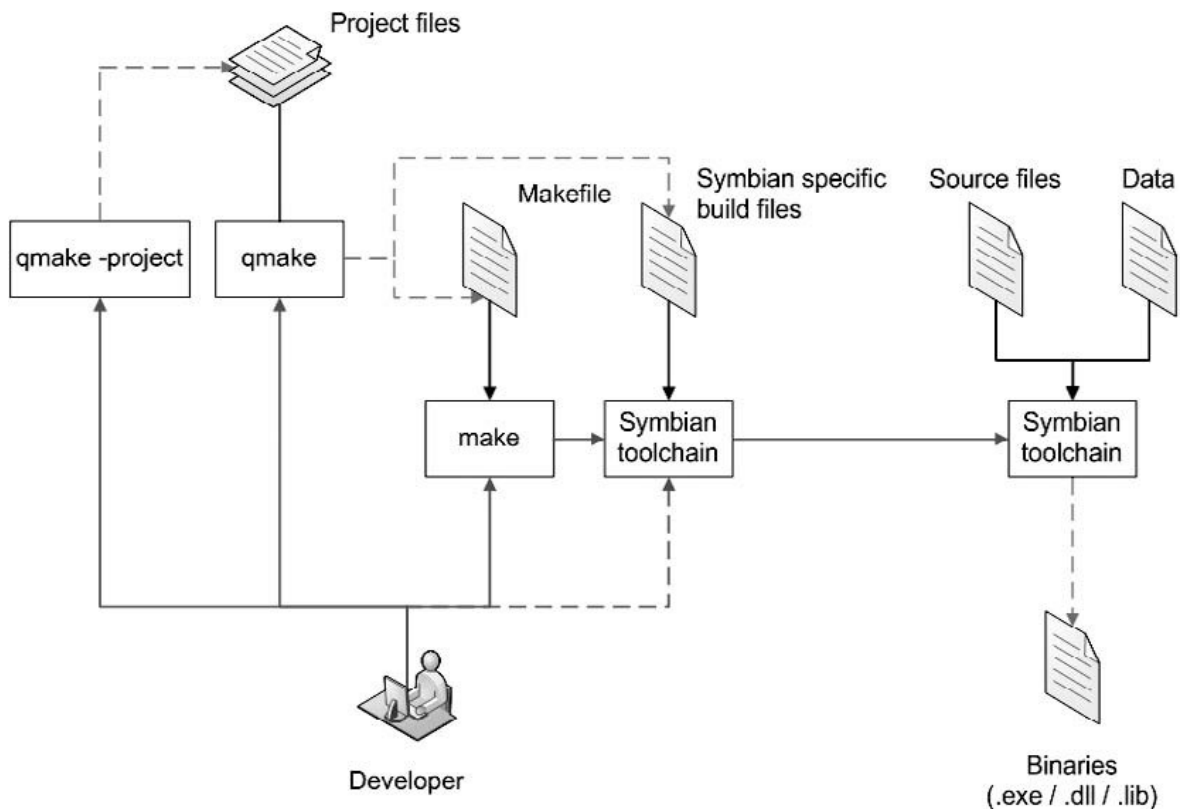
**.pkg*: Ένα Symbian αρχείο του πακέτου, το οποίο χρησιμοποιείται κατά τη δημιουργία Symbian αρχείου εγκατάστασης .sis.

**.sis*: Αρχείο installer για συσκευές που χρησιμοποιούν το λειτουργικό σύστημα Symbian OS.

**.sisx*: Τα .sisx αρχεία περιέχουν αρχεία, πακέτα εγκατάστασης που χρησιμοποιείται από το λειτουργικό σύστημα Symbian OS.

5.5 Επίλογος

Σε αυτό το κεφάλαιο είδαμε την υλοποίηση της εφαρμογής “Πολυλεξικό”, η οποία έγινε στο στο περιβάλλον ανάπτυξης εφαρμογών Qt. Το εργαλείο του Qt μας δίνει την δυνατότητα με τον ίδιο κώδικα και ορίζοντας μέσα στον κωδικό μας τι θα γίνεται όταν “τρέχει” η εφαρμογής μας ως desktop application και τι όταν τρέχει ως mobile application.



Εικόνα 5.5 Built σε περιβάλλον Symbian – χρήση του Qt

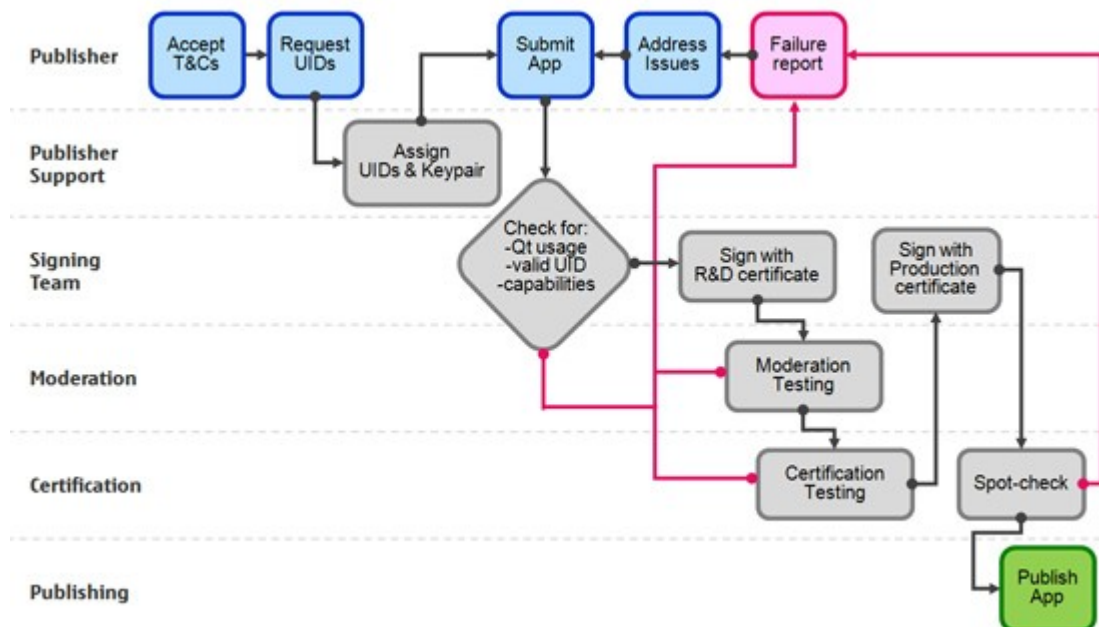
Τα Qt/Symbian project χρησιμοποιούν το underlying Symbian toolchain, αν και τα project “κατασκευάζονται” με διαφορετικό τρόπο. Ξεκινώντας, τα πρότυπα εργαλεία για το built του Qt χρησιμοποιούνται ως περιτύλιγμα γύρω από τα αντίστοιχα εργαλεία του Symbian. Αυτό σημαίνει ότι τα standard Qt αρχεία, όπως το *.pro χρησιμοποιούνται αντί των Symbian αρχείων του project. Επιπλέον, αυτά τα αρχεία είναι “χτισμένα” με τον ίδιο τρόπο που είναι “χτισμένα” και μη-Symbian Qt εφαρμογές, χρησιμοποιώντας τις εντολές `make` και `qmake`. Αυτή η υλοποίηση μεταξύ του Symbian και του Qt απεικονίζεται στην εικόνα 5.5.

Ολοκληρώνοντας αυτό το κεφάλαιο θα ήταν καλό να επισημανθούν

πληροφορίες που αφορούν γενικά πληροφορίες που μπορεί κανείς να βρει εφαρμογές για Symbian και πως μπορεί κανείς να εμπορευθεί Qt εφαρμογές για Symbian.

Το Oni Store είναι το ψηφιακό μαγαζί της Nokia, όπως οι χρήστες – πελάτες μπορούν να κατεβάσουν mobile παιχνίδια, εφαρμογές, video, εικόνες, ring tones για τις Nokia συσκευές. Μερικά από αυτά είναι δωρεάν, και άλλα πρέπει κανείς να χρησιμοποιήσει πιστωτική κάρτα για να έχει την δυνατότητα να έχει την πλήρη μορφή αυτού που θέλει να αγοράσει – προμηθευθεί.

Στην επόμενο σχεδιάγραμμα φαίνεται τι πρέπει να γίνει μέχρι να εκδοθεί η εφαρμογή του ως προς πώληση ή δωρεάν διάθεση.



Διάγραμμα 5.5: Symbain apps publishing

Σε αυτή την ιστοσελίδα υπάρχουν περισσότερες λεπτομέρειες για την έκδοση των Symbian εφαρμογών. <http://www.forum.nokia.com/Distribute/> . Επίσης, στην ιστοσελίδα http://www.forum.nokia.com/Distribute/Packaging_and_signing.xhtml υπάρχουν οι όροι και το τι πρέπει κανείς να κάνει ακριβώς για να εκδώσει την εφαρμογή του.

Βιβλιογραφία

Gehrke Johannes and Raghu Ramakrishnan (2002), *ΣΥΣΤΗΜΑΤΑ ΔΙΑΧΕΙΡΗΣΗΣ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ 2η Έκδοση Τόμος Α*, Εκδόσεις ΤΖΙΟΛΑ, Θεσσαλονίκη.

Robert Lafore (2005), *Αντικειμενοστραφής προγραμματισμός με τη C++*, Εκδόσεις Κλειδάριθμος.

Aubert Michael (2008), *Quick Recipes on Symbian OS Mastering C++ Smartphone Development*, Wiley.

Baijian Yang and Pei Zheng and Lionel Ni (2007), *Professional Microsoft Smartphone Programming*, Wrox Publishing

Blanchette Jasmin and SummerfieldMark (2008), *C++ GUI Programming with Qt 4, Second Edition*, Prentice Hall.

Ezust Alan and Ezust Paul (2007), *An introduction to design patterns in C++ with Qt 4*, Pearson Education.

Fitzek Frank H.P. And Mikkonen Tommi and Torp Tony (2010), *Qt for Symbian*,Wiley.

Litovski Ivan and Maynard Richard (2010), *Inside Symbian SQL A Mobile Developer's Guide to SQLite*, Wiley.

David Taniar (2009), *Mobile Computing Concepts, Methodologies, Tools, and Applications*, Monash University, Australia.