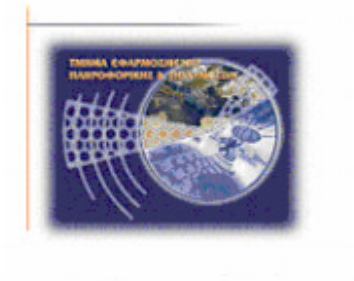
Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχόλη Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής και Πολυμέσων



Πτυχιακή Εργασία

Τίτλος: Σύγχρονη Συγχρονισμένη Συνεργασία Υποστηριζόμενη από κινητές συσκευές

Μιλτιάδης Κονσολάκης (Α.Μ.: 2006)

Επιβλέπων Καθηγητής: Ακουμιανάκης Δημοσθένης

Ημερομηνία Παρουσίασης: 17/03/2011

Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Δημοσθένη Ακουμιανάκη για την δυνατότητα που μου έδωσε να ασχοληθώ με ένα τόσο καινοτόμο και ενδιαφέρον θέμα. Επίσης, θα ήθελα να ευχαριστήσω τον συνεργάτη και φίλο Γιώργο Βελλή για την πολύτιμη καθοδήγηση και βοήθεια κατά την υλοποίηση και συγγραφή αυτής της πτυχιακής. Ακόμα ένα ευχαριστώ στα μελή του εργαστηριού iSTLab για την όποια βοήθεια τους. Στη συνέχεια θέλω να ευχαριστήσω την οικογένειά μου, προπάντων, για την δυνατότητα που μου προσέφερε να πραγματοποιήσω τις σπουδές μου με κάθε πολυτέλεια καθώς επίσης την αξιοσημείωτη συμπαράσταση που μου παρείχαν κατά τη διάρκεια εκπόνησης της πτυχιακής εργασίας μου. Τέλος, ευχαριστώ πολύ τα άτομα του φιλικού μου περιβάλλοντος, οι οποίοι με στήριξαν ψυχολογικά και ανέχτηκαν την απουσία μου για αξιόλογο χρονικό διάστημα.

Abstract

This thesis deals with the subject of cooperation supported by a computer (Computer Supported Cooperative Work: CSCW) and specifically in modern distributed (synchronous distributed) and the promotion of opportunities arising from the rapid growth observed in the field of mobile computing devices. More specifically focuses on the rapidly emerging and promising, in comparison to previous generations of mobile devices, category of smart mobile phones in which the use of two-dimensional graphic images transport (2D graphical metaphors) is studied as an appropriate mean of distributed multiple use interaction with the aim of a possibly higher degree of exploitation of the increased computing capabilities offered. In this perspective a series of new non-conventional interactive objects are developed and presented, while considering ways to optimize and integrate appropriate structures to assist transfer social elements aiming at supporting awareness among participating users in a collaborative session. Also the basic concepts of synchronous computer – supported co operations are studied and analyzed (ex. floor management etc) and also appropriate software structures are further developed and presented for their support.

Furthermore given the increased complexity that accompanies the software development for multiple devices, taking into account the inherent extensive variety of different configurations at the hardware level, operating system software supplied and the need of users in their ability to operate transparently specific applications regardless of the specific context of use, this paper proposes to address these advantages of an alternative model of software development, especially model-based software development.

Σύνοψη

Η συγκεκριμένη πτυχιακή εργασία καταπιάνεται με το γνωστικό αντικείμενο της συνεργασίας υποστηριζόμενης από υπολογιστή (Computer Supported Cooperative Work: CSCW) και δει της σύγχρονης κατανεμημένης (synchronous distributed) στα πλαίσια της ανάδειξης των δυνατοτήτων που εγείρονται από τη ραγδαία ανάπτυξη που παρατηρείται στο πεδίο των κινητών υπολογιστικών συσκευών. Πιο συγκεκριμένα εστιάζει στη νέα ταχύτατα ανερχόμενη και πολλά υποσχόμενη, σε σχέση με τις προηγούμενες γενιές κινητών συσκευών, κατηγορία των έξυπνων κινητών τηλεφώνων (smart phones) στα πλαίσια της οποίας μελετάται η χρήση δυσδιάστατων εικονικών γραφικών μεταφορών (2D graphical metaphors) ως ενδεδειγμένων μέσων κατανεμημένης πολυχρηστικής (multiuser) αλληλεπίδρασης έχοντας ως στόχο την όσο το δυνατόν μεγαλύτερου βαθμού εκμετάλλευση των αυξημένων προσφερόμενων υπολογιστικών δυνατοτήτων τους. Υπό αυτή την οπτική γωνία αναπτύχθηκαν και παρουσιάζονται μια σειρά νέων μη συμβατικών αλληλεπιδραστικών αντικειμένων, εξετάζοντας παράλληλα τρόπους ανάδειξης και ενσωμάτωσης κατάλληλων δομών υποβοήθησης μεταφοράς κοινωνικών στοιχείων στοχεύοντας στην υποστήριξη ενημερότητας (awareness) μεταξύ συμμετεχόντων χρηστών στα πλαίσια μιας συνεργατικής συνόδου. Επίσης μελετώνται και αναλύονται βασικές έννοιες της συγχρονισμένης συνεργασίας υποστηριζόμενης από υπολογιστή (βλ. διαχείριση αντιγράφων αντικειμένων, διαχείριση δαπέδου, κοκ.) ενώ επιπλέον αναπτύσσονται και παρουσιάζονται κατάλληλες εξειδικευμένες δομές λογισμικού για την υποστήριξή τους.

Παράλληλα δεδομένης της αυξημένης πολυπλοκότητας που συνοδεύει την ανάπτυξη λογισμικού για πολλαπλές συσκευές, λαμβάνοντας υπόψη την εκ φύσεως εκτεταμένη ποικιλία διαφορετικών διαμορφώσεων τόσο σε επίπεδο υλικού, λειτουργικού συστήματος, παρεχόμενου λογισμικού όσο και της ανάγκης των χρηστών όσον αφορά τη δυνατότητά τους να εκτελούν διαφανώς συγκεκριμένες-ενδεδειγμένες εφαρμογές ασχέτως των εκάστοτε πλαισίων χρήσης (contexts of use), η συγκεκριμένη εργασία προτείνει την αντιμετώπιση των παραπάνω εκμεταλλευόμενη τα κληροδοτηθέντα πλεονεκτήματα ενός εναλλακτικού μοντέλου ανάπτυξης λογισμικού και συγκεκριμένα της ανάπτυξης λογισμικού βάσει μοντέλων (model-based development).

Πίνακας περιεχομένων

Πίνακας Εικόνων

1. Εισαγωγή

Αυτή η πτυχιακή σχετίζεται με το γνωστικό αντικείμενο της συνεργασίας υποστηριζόμενης από υπολογιστή (Computer Supported Cooperative Work) και δει της σύγχρονης δίνοντας έμφαση στη συνεργασία υποστηριζόμενη από έξυπνες κινητές συσκευές. Ωστόσο δεδομένης της ολοένα αυξανόμενης χρήσης έξυπνων υπολογιστικών συσκευών και της διεισδυτικότητάς τους ως μέρος των καθημερινών μας πρακτικών μια νέα πραγματικότητα διαμορφώνεται όπου η έμφαση δίνεται σε επίπεδο διαφανούς πρόσβασης σε υπηρεσίες και όχι σε επίπεδο εφαρμογών ως μέσου πρόσβασης σε αυτές εγείροντας πλήθος απαιτήσεων σε επίπεδο ανάπτυξης. Συγκεκριμένα δεδομένης της ανομοιογένειας που χαρακτηρίζει τα δυνητικά περιβάλλοντα χρήσης (contexts of use), στα πλαίσια των οποίων καλούνται να εκτελεστούν πλήθος εφαρμογών, ανομοιογένειες τόσο σε επίπεδο υλικού, λειτουργικού συστήματος και εγγενούς λογισμικού υποστήριξης (υπό τη μορφή toolkits, π.χ.: GUI toolkit, κ.α.), εγείρουν πλήθος απαιτήσεων καθώς πλήθος παραμέτρων πρέπει να ληφθούν υπόψη κατά την ανάπτυξη λογισμικού γεννώντας εύλογα ερωτήματα σχετικά με την επάρκεια και ικανότητα των συμβατικών μεθόδων ανάπτυξης λογισμικού να επιληφθούν και ανταπεξέλθουν επιτυχώς αυτών. Ενδεικτικό είναι το παράδειγμα όπου συνήθως το πλήθος των περιβαλλόντων χρήσης που πρέπει να υποστηριχθούν από μια συγκεκριμένη εφαρμογή αντιστοιχεί συνήθως σε ανάγκη για γνώση αντίστοιχου πλήθους γλωσσών προγραμματισμού. Αν και η sun Microsystems (νυν Oracle) αναγνωρίζοντας το πρόβλημα επέκτεινε τη φιλοσοφία της java: “compile once run everywhere”, και σε επίπεδο κινητών συσκευών, κάνοντας ένα βήμα για τη διευκόλυνση της ανάπτυξη λογισμικού για κινητά, εισάγοντας την J2ME (από το: Java2 Mobile Edition), ακόμα το εν λόγω πεδίο υποφέρει είτε λόγω της έλλειψης υποστήριξης διερμηνευτή J2ME από πλήθος λειτουργικών συστημάτων, είτε δεδομένων των περιορισμών που τίθενται για την ανάπτυξη λογισμικού από την εκάστοτε πλατφόρμα (βλ. κυρίως ελλείψεις σε είδος, πλήθος και ποιότητα προσφερόμενων διαδραστικών αντικειμένων ανά πλατφόρμα και δυνατότητες συσκευών). Ένα άλλο κύριο πρόβλημα εστιάζεται στην ανάγκη για αναβάθμιση και ενσωμάτωση νέων απαιτήσεων-δυνατοτήτων σε υπάρχουσες πολύ-περιβαλλοντικές (multi-context) εφαρμογές. Η κατάσταση περιπλέκεται ακόμα περισσότερο αν συμπεριλάβουμε την ανάγκη δημιουργίας και υποστήριξης σύγχρονων συνεργατικών συστημάτων δεδομένης της εκ φύσεων αυξημένης δυσκολίας και πολυπλοκότητας αυτών.

Υπό το πρίσμα που περιγράφηκε παραπάνω η εν λόγω πτυχιακή εργασία προτείνει την αντιμετώπιση της ανάπτυξη σύγχρονου συνεργατικού λογισμικού για κινητές συσκευές, και δει έξυπνα (smart phones), μελετώντας, εξετάζοντας και προτείνοντας, εναλλακτικούς τρόπους ανάπτυξης λογισμικού ενδεδειγμένους για τέτοιου είδους απαιτητικά περιβάλλοντα, σε αντιδιαστολή με το προβληματικό εκ φύσεως επικρατών προγραμματιστικό στυλ, αυτό του χαμηλού επιπέδου βασιζόμενου σε εργαλειοθήκες (toolkit-based programming).

2. Συνεργασία Υποστηριζόμενη από Υπολογιστή

Η συνεργασία υποστηριζόμενη από υπολογιστή χαρακτηρίζει ένα διεπιστημονικό χώρο στον οποίο εμπλέκονται τόσο επιστήμες της πληροφορικής όπως: αλληλεπίδραση ανθρώπου υπολογιστή (Human Computer Interaction), τεχνολογία λογισμικού (Software Engineering), Κατανεμημένα Συστήματα (Distributed Computing) όπως και κοινωνικών επιστημών, κυρίως λόγω της ανάγκης για μελέτη του τρόπου οργάνωσης και λειτουργίας εικονικών ομάδων με στόχο την ανάδειξη θεωρητικών μοντέλων που ερμηνεύουν και υποβοηθούν τις εικονικές αλληλεπιδράσεις μεταξύ των χρηστών καθώς και της ανάδειξης και υπόδειξης κατάλληλων κοινωνικών στοιχείων που πρέπει να μεταφέρονται-κατανέμονται μεταξύ των χρηστών με στόχο την επιτυχημένη υποστήριξη ενημερότητας μεταξύ των συνεργαζόμενων χρηστών. Ως εκ τούτου θεωρείται εκ φύσεως ένα ιδιαίτερα απαιτητικό πεδίο τόσο λόγω των τεχνολογικών δυσκολιών που εγείρονται, όσο και λόγω της διεπιστημονικής φύσης αυτού.

Ένας από τους πιο δημοφιλείς τρόπους ταξινόμησης συνεργατικών συστημάτων είναι αυτός της χωροχρονικής κατανομής των συνεργατικών δραστηριοτήτων. Συγκεκριμένα δεδομένης της χωρικής κατανομής οι εφαρμογές διαχωρίζεται σε κατανεμημένες (δηλ. distributed) ή εκ του σύνεγγυς (δηλ. collocated), ενώ υπό το δεύτερο κριτήριο, αυτό του χρόνου, διαχωρίζονται σε σύγχρονες ή ασύγχρονες ανάλογα με τον αν οι αλλαγές γίνονται άμεσα αντιληπτές μεταξύ των συμμετεχόντων ή όχι αντίστοιχα. Εμείς ωστόσο θα εστιάσουμε στη σύγχρονη κατανεμημένη συνεργασία. Αξίζει να σημειωθεί ότι οι τεχνολογικές προκλήσεις διαφέρουν σε σχετικά μεγάλο βαθμό σε σχέση με το είδος υποστηριζόμενης συνεργασίας και είναι δύσκολο να υποστηριχθούν πολλές μορφές μαζί, παρά το γεγονός ότι κατά καιρούς έχουν αναφερθεί διάφορες προσπάθειες (π.χ.: δυνατότητα εναλλαγής μεταξύ σύγχρονης και ασύγχρονης ενημέρωση).

Σήμερα υπάρχουν πολλά είδη σύγχρονων συνεργατικών εφαρμογών, ωστόσο οι προσφερόμενες υπηρεσίες, το επίπεδο συνδεσιμότητας (coupling-level) και οι τεχνολογίες που χρησιμοποιούνται για την υλοποίηση αυτών μπορεί να διαφέρουν σε σημαντικό βαθμό ανάλογα με τη φύση της επικοινωνίας. Ενδεικτικό παράδειγμα αποτελεί ο διαμοιρασμός πληροφορίας σε επίπεδο εφαρμογής (application sharing), όπου αυτό που επιχειρείται, στόχος δηλαδή είναι η μετατροπή κλασικών εφαρμογών προγραμματισμένων για single-user αποκλειστικά χρήση σε πολυχρηστικές (π.χ.: multiuser Word, multiuser MSPaint, κοκ). Άλλο παράδειγμα αποτελούν οι πού δημοφιλείς screen sharing εφαρμογές πολύ χρήσιμες ειδικότερα σε περιπτώσεις απομακρυσμένης υποβοήθησης (remote assistance), όπου το επίπεδο διαμοιρασμού έγκειται σε επίπεδο ολόκληρης ή μέρους της οθόνης. Ωστόσο, αξίζει να ενδεικτικά να σημειωθεί ότι πολλοί είναι οι τεχνολογικής φύσης περιορισμοί ανάλογα με το επίπεδο διαμοιρασμού, όπως αυτό της ανάγκης για κοινό μέγεθος οθόνης, λειτουργικού συστήματος, εκδόσεις του λογισμικού, κοκ.

Στα πλαίσια του αντικειμένου της παρούσης εργασίας ιδιαίτερο ενδιαφέρον παρουσιάζουν οι περιπτώσεις διαμοιρασμού σε επίπεδο widget και μοντέλου(σε όρους Model View Controller

αρχιτεκτονικής). Στη συνέχεια παρουσιάζονται τεχνολογικές συνιστώσες που αφορούν συγκεκριμένα το διαμοιρασμό σε επίπεδο τόσο widget, αλλά και μοντέλου, δεδομένης της κατεύθυνσης της πτυχιακής αλλά και των επιθυμητών εφαρμογών προς υποστήριξη.

2.1 Διαχείριση Συνόδων (Session Management)

Στόχος των συνόδων (sessions) είναι η ομαδοποίηση τόσο των στιγμιότυπων των διαμοιραζόμενων μεταξύ των εμπλεκόμενων χρηστών αντικειμένων (shared objects), αλλά και η οργάνωση των χρηστών που προσπελαύνουν αυτά.

Δεδομένου ότι η ύπαρξη συνεργασίας προϋποθέτει την ύπαρξη κοινόχρηστων αντικειμένων οργανωμένων και προσβάσιμων μεταξύ μελών μιας συγκεκριμένης ομάδας χρηστών, ουσιαστικά μια σύνοδος λειτουργεί και μπορεί εναλλακτικά να οριστεί σαν ένα εικονικό αντικείμενο συμπερίληψης (κοινόχρηστος εικονικός υποδοχέας: shared virtual container), εκτελούμενο υπό κανονικές συνθήκες στην πλευρά του server (server-side), υποβοηθώντας παράλληλα με αυτό τον τρόπο την καθοριστική για την υλοποίηση της συνεργασίας δυνατότητα συγχρονισμού μεταξύ των χρηστών βάσει της 'ομάδας' στην οποία ανήκουν(-της οποίας είναι μέλη). Άρα είναι προφανές ότι κάθε σύνοδος συσχετίζεται με συγκεκριμένα αντικείμενα, apriori καθορισμένα, με δυνατότητα ωστόσο runtime μεταβολών ανάλογα με την περίπτωση τα οποία κάθε φορά έχουν νόημα στα πλαίσια των προβλημάτων/στόχων που κάθε σύνοδος προσπαθεί να λύσει.

Στα πλαίσια ενός σύγχρονου συνεργατικού συστήματος μπορεί να υπάρχουν κατά κανόνα ταυτόχρονα καμία, μια ή και περισσότερες σύνοδοι, καθόλου, μερικώς ή πλήρως χρονικά επικαλυπτόμενες. Η υπολογιστική εξασφάλιση της ύπαρξης και γενικότερης λειτουργικότητας των συνόδων εξασφαλίζεται από την υλοποίηση και ύπαρξη ενός αντικειμένου το οποίο ονομάζεται διαχειριστής συνόδων(session manager). Στις υποχρεώσεις-δυνατότητες ενός τέτοιου διαχειριστή έγκειται η δυνατότητα ορισμού μιας συνόδου μέσω των παραμέτρων αυτής (π.χ. ημ/νια εκκίνησης για προκαθορισμένες, προβλεπόμενη ημερομηνία λήξης ή διάρκεια, κοκ.), ορισμού επιτρεπόμενων μελών (registration policies), δυνατότητας παροχής προγραμματιστικής πρόσβασης(δημοσιοποιώντας το ανάλογο API) στα κοινόχρηστα αντικείμενα, κοκ, όπως επίσης και καταστροφής.

2.2 Διαχείριση Δαπέδου (Floor Management)

Η χρησιμότητα – αξία της διαχείρισης δαπέδου αναδεικνύεται στα πλαίσια κοινόχρηστων χώρων εργασίας (shared work spaces) που δυνητικώς φιλοξενούν στιγμιότυπα διαμοιραζόμενων μεταξύ κατανεμημένων χρηστών αντικειμένων ώστε να διασφαλιστεί η μη ταυτόχρονη πρόσβαση σε 'εκτεθειμένων' σε αυτή 'κρίσιμων' δεδομένων/αντικειμένων.

Αξίζει ωστόσο να σημειωθεί ότι δεν αποτελεί κληροδοτηθείσα υποχρέωση κάθε συνεργατικού συστήματος η δημιουργία και ύπαρξη της έννοιας της διαχείρισης δαπέδου. Ενδεικτικό παράδειγμα αποτελούν τα πολυχρηστικά συστήματα ζωγραφικής (multiuser painting), όπου κάθε χρήστης δύναται

να ενεργήσει σε οποιοδήποτε διαμοιραζόμενο σημείο του συνεργατικού χώρου ασχέτως των ενεργειών και του τρέχοντος σημείου ενδιαφέροντος των άλλως δυνητικώς συμμετεχόντων χρηστών.

Ωστόσο σε αρκετές περιπτώσεις είναι επιθυμητό οι ενέργειες ενός χρήστη να προϋποθέτουν τον καθολικό έλεγχο-κυριότητα(floor control) σε επίπεδο είτε κοινόχρηστου χώρου ή μεμονωμένων αντικειμένων. Η απόκτηση του ελέγχου-κυριότητας ενός κοινόχρηστου αντικειμένου μπορεί να γίνεται, ανάλογα με την εκάστοτε εφαρμογή, με βάσει πλήθος μεθόδων μεταξύ των οποίων οι:

- First In First Served: ο έλεγχος δίνεται με βάσει τη σειρά που ζητήθηκε (δημοκρατική λειτουργία)
- Role Based: Αυτός που έχει 'ισχυρότερο' ρόλο(π.χ. Administrator vs Simple User) εκτοπίζει βίαια αυτούς με το λιγότερο ισχυρό ρόλο
- Mixed: Π.χ. μεταξύ χρηστών που έχουν τον ίδιο ρόλο ισχύει η πρώτη μέθοδος, ενώ σε αντίθετη η δεύτερη

Αξίζει να σημειωθεί πως μπορεί να υπάρχει και δίδεται αρκετές φορές η δυνατότητα εν εξελίξει (runtime) αλλαγής του καθεστώτος (Floor policy) βάσει του οποίου διαμοιράζεται ο έλεγχος.

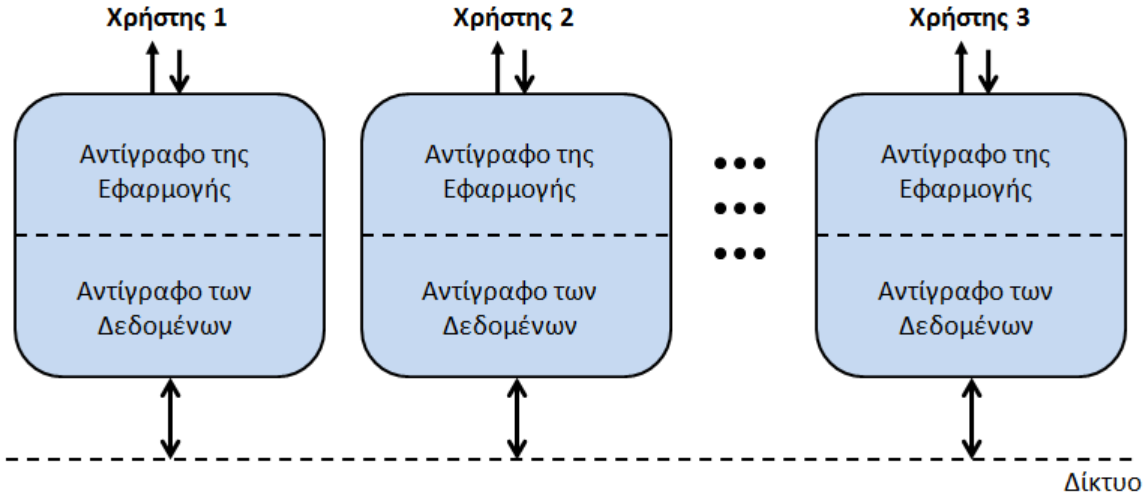
2.3 Αρχιτεκτονικές (Architectures)

Η αρχιτεκτονική ενός σύγχρονου συνεργατικού συστήματος μπορεί κατά κανόνα να ανήκει σε μια εκ των τριών ακόλουθων κατηγοριών: αντιγραφής (replicated), κεντρικοποιημένη (centralized) και υβριδική (hybrid).

Ωστόσο για να γίνει ορατή και αντιληπτή η διαφοροποίηση αυτών, πρέπει να ορίσουμε ποια είναι τα τμήματα-επίπεδα μιας εφαρμογής που ως λειτουργικότητα μπορούν να κατατμηθούν. Τα επίπεδα στα οποία μπορεί να χωριστεί η λειτουργικότητα μιας εφαρμογής είναι αυτό της όψης (View), το οποίο σχετίζεται με εκείνες τις πληροφορίες που αφορούν το rendering της διεπαφή και μόνο, του μοντέλου (Model), στο οποίο αποθηκεύεται η up-to-date κατάσταση που πρόκειται να εμφανιστεί καταλλήλως από την εκάστοτε συσχετιζόμενη με αυτό όψη (View) και τον ελεγκτή (Controller) του οποίου ρόλος είναι να μεταφέρει αλλαγές και συνεπώς να ενημερώνει και να κρατά συγχρονισμένα τα δεδομένα που είναι να εμφανιστούν (που ορίζουν την κατάσταση της γραφικής εφαρμογής - application's state) με τα δεδομένα που εν τέλει εμφανίζονται στο χρήστη (γίνονται δηλ. rendered).

Υπό αυτή τη θεώρηση, στις αρχιτεκτονικές αντιγράφων (replicated architectures) η φιλοσοφία εστιάζεται στην αντιγραφή των τριών λειτουργικών συνιστωσών (δηλ. των view, controller και model), σε κάθε κατανεμημένο κόμβο, το οποίο μεταφράζεται στο ότι κάθε κατανεμημένη πολυχρηστική εφαρμογή/κόμβος, οφείλει να κρατάει ένα δικό του τοπικό αντίγραφο των κοινά διαμοιραζόμενων πληροφοριών. Π.χ. στην περίπτωση που το διαμοιραζόμενο αντικείμενο είναι ένα στιγμιότυπο της κλάσης κουμπί (Button), αυτό το οποίο θα περίμενε κανείς να γίνει replicated είναι μια λογική μεταβλητή (Boolean variable) η οποία θα αντιστοιχεί στην τρέχουσα κατάσταση του κουμπιού, δηλαδή

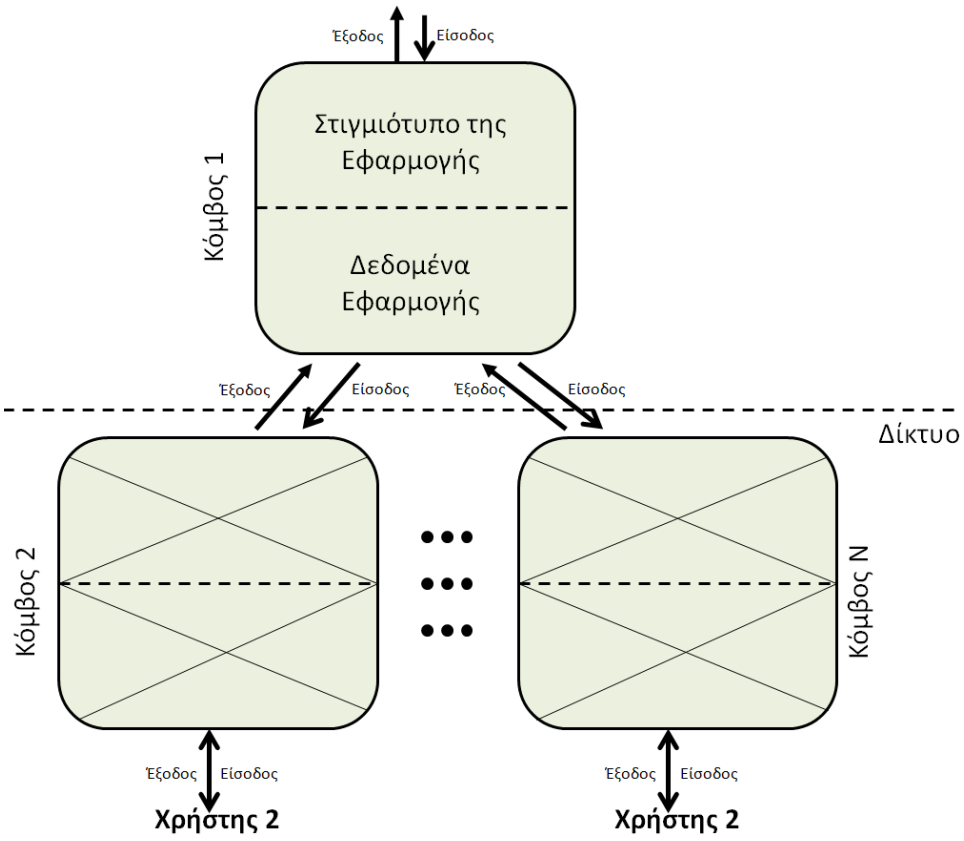
το αν είναι πατημένο(var == true) ή όχι (var == false). Συνεπάγεται λοιπόν ότι κάθε client, που μετέχει στα πλαίσια της ίδιας συνεργατικής συνόδου, είναι υποχρεωμένος να εκτελεί και εκτελεί ένα δικό του τοπικό (local) στιγμιότυπο (instance) της εφαρμογής η οποία είναι υποχρεωτικό να είναι ήδη εγκατεστημένη σε κάθε ένα client ξεχωριστά. Στο παρακάτω σχήμα (Εικόνα 1) παρουσιάζεται ένα παράδειγμα της αρχιτεκτονικής αντιγράφων.



Εικόνα 1: Παράδειγμα αρχιτεκτονικής Αντιγράφων

Από την άλλη πλευρά, αυτή των κεντρικοποιημένων αρχιτεκτονικών υφίσταται ένα μοναδικό στιγμιότυπο (single instance) μιας εφαρμογής που είναι εγκατεστημένη και εκτελείται σε αποκλειστικά ένα υπολογιστικό κόμβο, από τους συνολικά διαθέσιμους στους οποίους δεν είναι υποχρεωτική η ύπαρξη της ανάλογης κοινόχρηστης εφαρμογής. Παραδείγματος χάριν στην περίπτωση ενός κεντρικοποιημένου πολυχρηστικού κειμενογράφου (centralized multiuser word processor) είναι απαραίτητος ο ορισμός ενός κύριου (master) ή αλλιώς κεντρικού (centralized) κόμβου, στα πλαίσια του οποίου είναι υποχρεωτική η ύπαρξη και δυνατότητα απευθείας εκτέλεσης της εν λόγω εφαρμογής (προϋποθέτοντας ότι είναι τοπικά εγκατεστημένη), με τους υπόλοιπους (κόμβους) να συνδέονται απευθείας σε αυτόν (κεντρικό κόμβο) στον οποίο είναι απόλυτα εξαρτώμενοι. Το τελευταίο μεταφράζεται ως εξής, η είσοδος κάθε κόμβου μεταβιβάζεται και επεξεργάζεται αποκλειστικά και μόνο στον ορισμένο ως κεντρικό κόμβο και αφού επεξεργαστεί επιστρέφεται-μεταβιβάζεται μέσω του δικτύου η έξοδος αυτού στον εξαρτημένο (slave) κόμβο όπου και παρουσιάζεται. Με τη βοήθεια της Εικόνα 2: Παράδειγμα Κεντρικοποιημένης αρχιτεκτονικήςΕικόνα 2, παρουσιάζεται η δομή και φιλοσοφία τέτοιου είδους συστημάτων. Ωστόσο είναι προφανές ότι στα μειονεκτήματα αυτής της ‘μεθόδου’ εισάγονται μεγάλες χρονικές καθυστερήσεις, μεταξύ εισόδου του χρήστη και ανταποκρισιμότητας του συστήματος, ανάλογες του τύπου, της χωρητικότητας και τρέχοντος φόρτου του εκάστοτε δικτύου. Επίσης στις κεντρικοποιημένες αρχιτεκτονικές δεν υποστηρίζεται ελαστικότητα όσον αφορά την εν εξελίξει (runtime) δυνατότητα ενός χρήστη να εναλλάσσεται καταστάσεων όσον αφορά την κατάσταση της σύνδεσής του (συνδεδεμένος ή αποσυνδεμένος). Το τελευταίο μπορεί να

είναι αδιάφορο σε πλήθος εφαρμογών μη δημιουργώντας κανένα απολύτως πρόβλημα, ωστόσο σε περιβάλλοντα όπου πρέπει να συμπεριληφθούν-υποστηριχθούν κινητοί χρήστες (οι οποίοι μπορεί να εναλλάσσουν πολύ εύκολα την κατάστασή τους ανάλογα του δικτύου μεταξύ ενεργής και ανενεργής) πλήθος απαιτήσεων εγείρεται αναδεικνύοντας κληροδοτηθέντες περιορισμούς της εν λόγω αρχιτεκτονικής. Στα μειονεκτήματα των κεντρικοποιημένων αρχιτεκτονικών συγκαταλέγεται και ο υψηλός φόρτος δικτύου που συνήθως επιβάλουν - προϋποθέτουν. Στα πλεονεκτήματά της συγκαταλέγεται η εύκολη διαχείριση ουσιαστικών απαιτήσεων όπως της διαχείρισης δαπέδου, διατήρησης συγχρονισμού και συνεπώς συνοχής (consistency) μεταξύ των κατανεμημένων κόμβων. Ενδεικτικές εφαρμογές που στηρίζονται σε αυτή την αρχιτεκτονική είναι αυτές της απομακρυσμένης υποβοήθησης μέσω διαμοιρασμού οθόνης (remote assistance screen sharing, κ.α.).



Εικόνα 2: Παράδειγμα Κεντρικοποιημένης αρχιτεκτονικής

Ωστόσο, και η αρχιτεκτονική αντιγράφων που αναφέρθηκε προηγουμένως δεν στερείται μειονεκτημάτων όπως αυξημένης πολυπλοκότητας στην εξασφάλιση διατήρησης συγχρονισμού και συνεπώς συνέπειας μεταξύ των κατανεμημένων κόμβων όπως επίσης δυσκολίας στην υλοποίηση μηχανισμών αποφυγής ταυτόχρονης πρόσβασης (floor managers) σε κοινόχρηστους πόρους σε συνδυασμό με συνήθως πολύ ειδικευμένες μεθόδους επίλυσης ασυνεπειών σε περίπτωση αναπόφευκτης ύπαρξης. Ένας άλλος περιορισμός αυτών είναι η προϋπόθεση και απαίτηση εγκατάστασης της κοινόχρηστης εφαρμογής σε κάθε ένα κόμβο εξασφαλίζοντας παράλληλα τη μεταξύ τους συμβατότητα

(π.χ. ίδια έκδοση, κοινές δυνατότητες, ίδια πρόσθετα, κοκ.). Ωστόσο υπερτερεί της κεντρικοποιημένης αρχιτεκτονικής όσον αφορά τη δυνατότητα υποστήριξης ταυτόχρονα, είτε σε επίπεδο εφαρμογής είτε κοινόχρηστων αντικειμένων, τον ορισμό της κατάστασής τους ως συνδεδεμένης (connected) και μη (disconnected) για ένα χρονικό διάστημα που μπορεί να είναι τυχαίο και ανεξάρτητο.

Το ποια αρχιτεκτονική θα επιλεγεί είναι αποκλειστικά θέμα τόσο της εφαρμογής που θα αναπτυχθεί – υποστηριχθεί όσο και της ιεραρχίας απαιτήσεων και όχι της συνολικά συγκριτικά υπερτέρησης κάποιας εκ των δύο. Ωστόσο ένα βήμα πιο πέρα προς αυτή την κατεύθυνση γίνεται από τις υβριδικές αρχιτεκτονικές που δεν είναι τίποτα άλλο από μίξεις των παραπάνω.

Τέλος, όσον αφορά τη συγκεκριμένη λειτουργική συνιστώσα των σύγχρονων συνεργατικών συστημάτων, αξίζει να σημειωθεί πως παρά το γεγονός ότι η συγκειμένη ενότητα καλύπτει βασικές μόνο έννοιες μια πιο εκτενής αναφορά θα ξέφευγε από το στόχο της παρούσης εργασίας.

2.4 Ενημερότητα (Awareness)

Η ενημερότητα (awareness) αφορά τη διαθεσιμότητα και δυνατότητα μετάδοσης, μέσω των εκάστοτε διαθέσιμων εικονικών μέσων, πληροφοριών που αφορούν την 'κατάσταση'(status) των λοιπών χρηστών την εκάστοτε χρονική στιγμή, έχοντας ως στόχο τον καλύτερο δυνατό συντονισμό μεταξύ τους και συνεπώς την καλύτερη, ποιοτικότερη και πιο αποτελεσματική συνεργασία.

Η ύπαρξη ενημερότητας είναι είτε αυτονόητη (βλ.: πρόσωπο με πρόσωπο επικοινωνία: 'Face to Face communication'), είτε αυθύπαρκτη όσον αφορά συγκεκριμένες μορφές συνεργασίας(βλ. π.χ.: collocated collaboration, videoconferencing, κ.α.), μερικώς υποστηριζόμενη εκ φύσεως (βλ. screen-sharing) ή και παντελώς απούσα (βλ. π.χ.: application sharing, όπου μονοχρηστικές εφαρμογές 'προσαρμόζονται' σε multiuser συνθήκες). Σε περιβάλλοντα διαμοιραζόμενου εικονικού χώρου (shared workspace environments), όπως αυτά που μελετώνται στα πλαίσια της παρούσης πτυχιακής, χρειάζεται ειδική μέριμνα για την υποστήριξή τους.

Αξίζει να σημειωθεί ότι παρά το γεγονός ότι το θέμα της ενημερότητας έχει απασχολήσει και συνεχίζει να απασχολεί την ερευνητική κοινότητα, δεν έχει δοθεί κανένας μέχρι στιγμής καθολικός και ευρέως αποδεκτός ορισμός του είδους της πληροφορίας που πρέπει κάθε φορά να μεταφέρεται ώστε να επιτυγχάνεται ο μέγιστος βαθμός ποιοτικά δυνατής ενημερότητας που να προσιδιάζει δηλαδή όσο το δυνατόν περισσότερο την πρόσωπο με πρόσωπο επικοινωνία.

Όσον αφορά τη συνεργασία βασιζόμενη σε εικονικούς διαμοιραζόμενους χώρους (virtual workspaces), που είναι και το αντικείμενο της τρέχουσας μελέτης, το είδος των πληροφοριών που φαίνεται πως είναι καταλληλότερο να μεταφέρεται μεταξύ των συμμετεχόντων-συνεργαζόμενων χρηστών και που κερδίζει συνεχώς έδαφος είναι αυτό που περιγράφεται από τους Gutwin & Greenberg στο [4)]. Συγκεκριμένα αναφέρει πως τα βασικά ερωτήματα στα οποία πρέπει να μπορεί να απαντάει ένας χρήστης αναλύονται στα: α) Ποιος χρήστης(ιδιότητα χρήστη), β) Κάνει τι (ποια ενέργεια εκτελεί

μεταξύ των διαθέσιμων επιλογών την εκάστοτε χρονική στιγμή) γ) Και που (αντικείμενο εστίασης: Current Object of Focus), με αναφορά σε ποιο αντικείμενο. Ενδεικτικά παρατίθενται οι παρακάτω δύο πίνακες (Εικόνα 3,Εικόνα 4) που συνοψίζουν το είδος των κατηγοριών των ερωτημάτων στις οποίες πρέπει να μπορεί να 'απαντά' μια πολυχρηστική διεπαφή (multiuser UI), με το είδος των απαιτούμενων απαντήσεων.

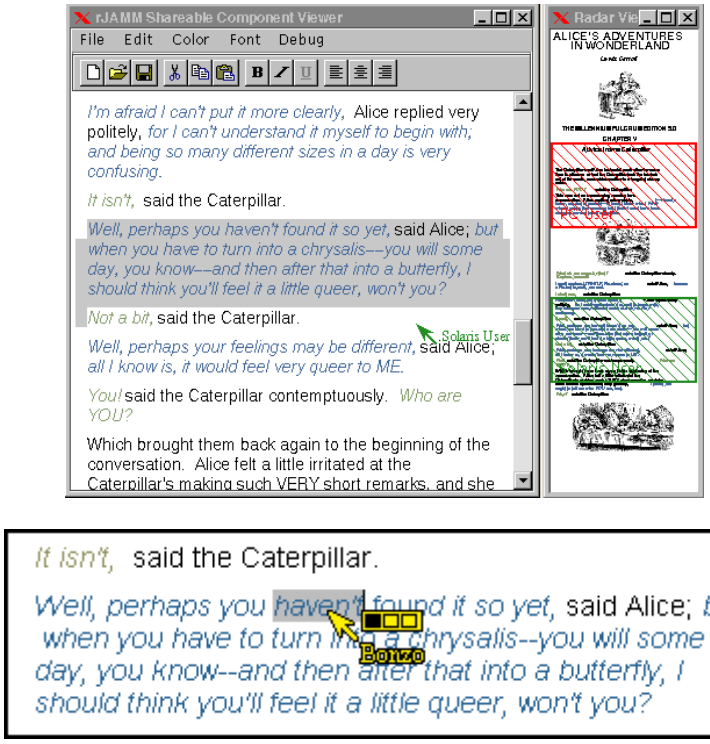
Category	Element	Specific questions
Who	Presence	Is anyone in the workspace?
	Identity	Who is participating? Who is that?
What	Authorship	Who is doing that?
	Action	What are they doing?
	Intention	What goal is that action part of?
	Artefact	What object are they working on?
Where	Location	Where are they working?
	Gaze	Where are they looking?
	View	Where can they see?
	Reach	Where can they reach?

Εικόνα 3 : Στοιχεία του χώρου εργασίας με την παρούσα κατάσταση

Category	Element	Specific questions
How	Action history	How did that action happen?
	Artefact history	How did this artefact come to be in this state?
When	Event History	When did that event happen?
Who (past)	Presence history	Who was here, and when?
Where (past)	Location History	Where has a person been?
What (past)	Action history	What has a person been doing?

Εικόνα 4 : Στοιχεία του χώρου εργασίας με το παρελθόν

Στην **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.**, παρατηρούμε ενδεικτικούς μηχανισμούς υποστήριξης της ενημερότητας εταίρων (radar view πάνω όψη και διάγραμμα δραστηριότητα κάτω όψη) σε περιβάλλον κοινόχρηστου συνεργατικού χώρου εργασίας (collaborative virtual workspace). Και στις δύο περιπτώσεις τ συνεργατικό περιβάλλον διατηρεί πληροφορία σχετικά με τους εμπλεκόμενους και το τι κάνουν (ή τι αλλαγές επιφέρουν) στο κοινόχρηστο αντικείμενο, τις οποίες και δημοσιοποιούν με κατάλληλους μηχανισμούς διάδρασης.



Εικόνα 5: Παραδείγματα μηχανισμών ενημερότητας

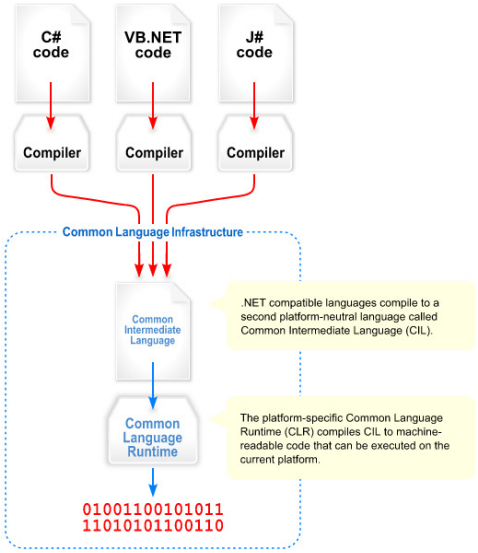
3. Ανάπτυξη Λογισμικού για Κινητές Συσκευές

Σε αυτό το κεφάλαιο θα περιγράψουμε τις πιο δημοφιλείς τεχνολογίες με τις όποιες μπορούμε να αναπτύξουμε λογισμικό για έξυπνες κινητές συσκευές, καθώς επίσης κάποια εκ των πλεονεκτημάτων και περιορισμών που τίθενται από την εκάστοτε αρχιτεκτονική της κάθε πλατφόρμας.

3.1 Windows Mobile OS (VB, C#, J#)

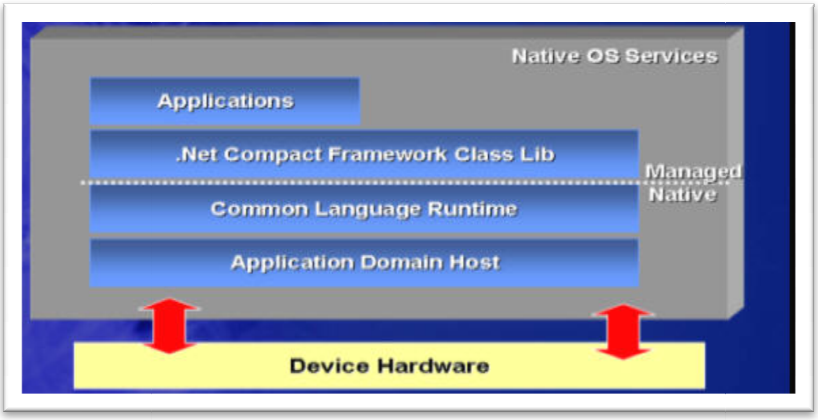
Το widows mobile είναι η εκδοχή του πιο δημοφιλούς λειτουργικού συστήματος για προσωπικούς υπολογιστές, τα MS Windows, για κινητές συσκευές. Η τελευταία έκδοση του εν λόγω λειτουργικού συστήματος είναι η "Windows Mobile 7" και αποτελεί την τελευταία προσπάθεια της εταιρίας, όπως επίσημα έχει δηλωθεί, να ανταπεξέλθει στην ανταγωνιστική αγορά των λειτουργικών συστημάτων για έξυπνα κινητά μιας και ήδη φαίνεται να έχει μείνει πολύ πίσω των εξελίξεων και δυνατοτήτων που προσφέρουν τα ανταγωνιστικά λειτουργικά συστήματα (βλ.: Google Android, Apple iOS).

Η τρόποι ανάπτυξης λογισμικού για το εν λόγω λειτουργικό σύστημα αποτελούν επέκταση της φιλοσοφίας της εταιρίας που το αναπτύσσει (Microsoft) και που διέπει τις αντίστοιχες εκδόσεις της για σταθερούς υπολογιστές. Συγκεκριμένα οι προγραμματιστές καλούνται να χρησιμοποιήσουν μια ή περισσότερες εκ των υποστηριζόμενων βάσει της πλατφόρμας .Net γλωσσών προγραμματισμού. Η .Net είναι ένα πλαίσιο εργασίας (framework), ενδεδειγμένο αποκλειστικά για τα λειτουργικά συστήματα των Windows, το οποίο υποστηρίζει μέσω κατάλληλων δομών τη συγγραφή και διαλειτουργικότητα προγραμμάτων υλοποιημένων σε όποια εκ των υποστηριζόμενων από αυτό γλωσσών προγραμματισμού (όπως: VB.Net, C#, j#, κα.). Το .Net Framework αποτελείται – περιλαμβάνει το CLR (Common Language Runtime) απαραίτητο αντικείμενο (component) – εικονική μηχανή (virtual machine), ενδεδειγμένη για την απρόσκοπτη εκτέλεση προγραμμάτων γραμμένων σε υποστηριζόμενες από αυτό γλωσσών προγραμματισμού.



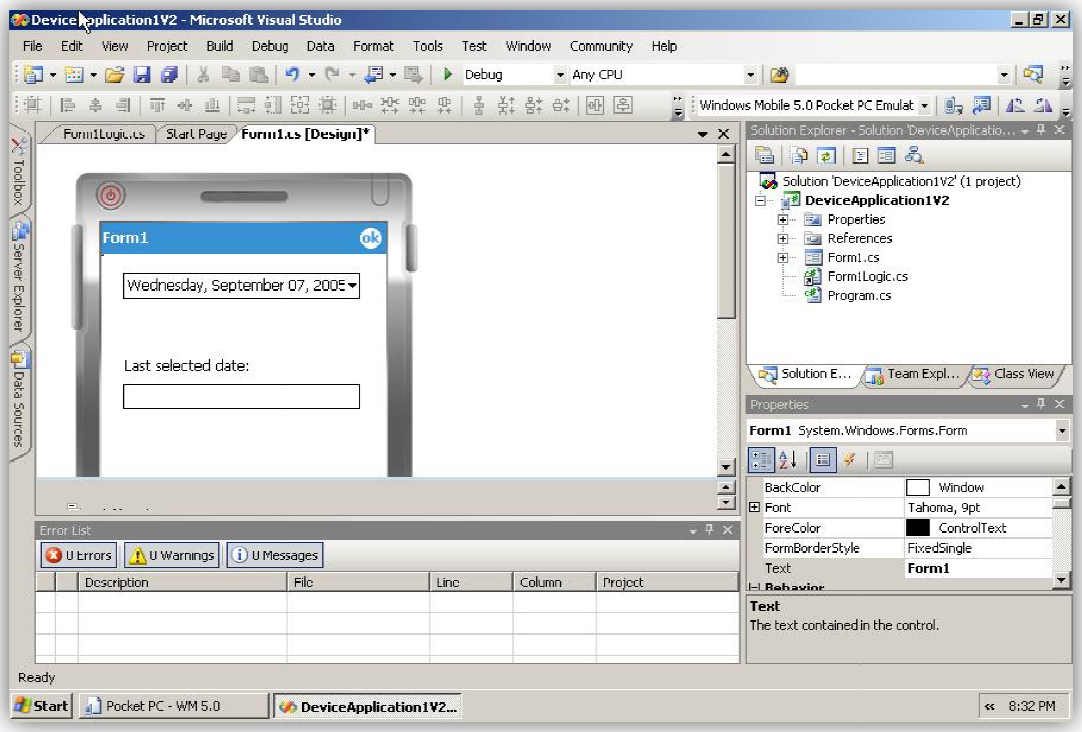
Εικόνα 6: Επισκόπηση της Αρχιτεκτονικής του .Net Framework

Στην Εικόνα 6, απεικονίζεται η θέση του CLR στην αρχιτεκτονική του .Net, με το CIL (Common Language Infrastructure) να αποτελεί την ενδιάμεση αφαιρετική δομή στην οποία γίνονται compiled τα προγράμματα της .Net ώστε να υποστηριχθεί η δυνητική διαλειτουργικότητά των. Το .NET Framework παρέχει επίσης πλήθος βιβλιοθηκών μεταξύ των οποίων εργαλειοθηκών για τη ανάπτυξη διεπαφών χρήστη υπολογιστή, τη διευκόλυνση/αυτοματισμό/τυποποίηση της πρόσβασης σε δεδομένα (Input/Output), τη σύνδεση σε πλήθος βάσεων δεδομένων (Database connectivity), την κρυπτογραφία (cryptography), τη δικτυακή επικοινωνία (networking) κ.α.. Ωστόσο υπό το πρίσμα των περιορισμών που τίθενται από τις κινητές συσκευές (βλ.: κινητές συσκευές, προσωπικούς ψηφιακούς βοηθούς, κινητά τηλεφώνα, κλπ.) και την ανάπτυξη λογισμικού σε αυτές η Microsoft προτείνει μια εκδοχή του .Net σχεδιασμένου να λειτουργεί με περιορισμένους πόρους, το οποίο ονομάζει .NET Compact Framework. Αρκετές βιβλιοθήκες είναι ίδιες με την κανονική πλατφόρμα ωστόσο έχουν διαφοροποιηθεί κάποιες έτσι ώστε να υποστηρίζουν φορητές συσκευές καθώς επίσης έχουν μειωθεί ώστε να χρησιμοποιούν μικρότερη χωρητικότητα. Στην Εικόνα 7 βλέπουμε την αρχιτεκτονική του compact .NET Framework.



Εικόνα 7 : Επισκόπηση αρχιτεκτονικής του Compact .NET Framework

Προκειμένου να υλοποιήσουμε εφαρμογές βασισμένες στην εν λόγο πλατφόρμα χρησιμοποιούμε το Visual Studio .NET (όπως φαίνεται στην Εικόνα 8) το οποίο παρέχεται από την Microsoft και δεν είναι δωρεάν. Η κατασκευή γραφικών διεπαφών είναι ίδια με αυτήν της έκδοσης για προσωπικούς υπολογιστές, βάση της όποιας μπορούμε να δημιουργήσουμε κουμπιά, πλαίσια κειμένου κ.α. πάνω σε μια φόρμα. Το βασικό μειονέκτημα είναι το ότι η δημιουργία διεπαφών ενδεδειγμένων για λειτουργία μέσω οθονών αφής δεν αποτελεί εύκολη διαδικασία. Προκειμένου να γίνει κάτι τέτοιο πρέπει να χρησιμοποιήσουμε επιπλέον βιβλιοθήκες που παρέχουν τέτοιου είδους δυνατότητες.



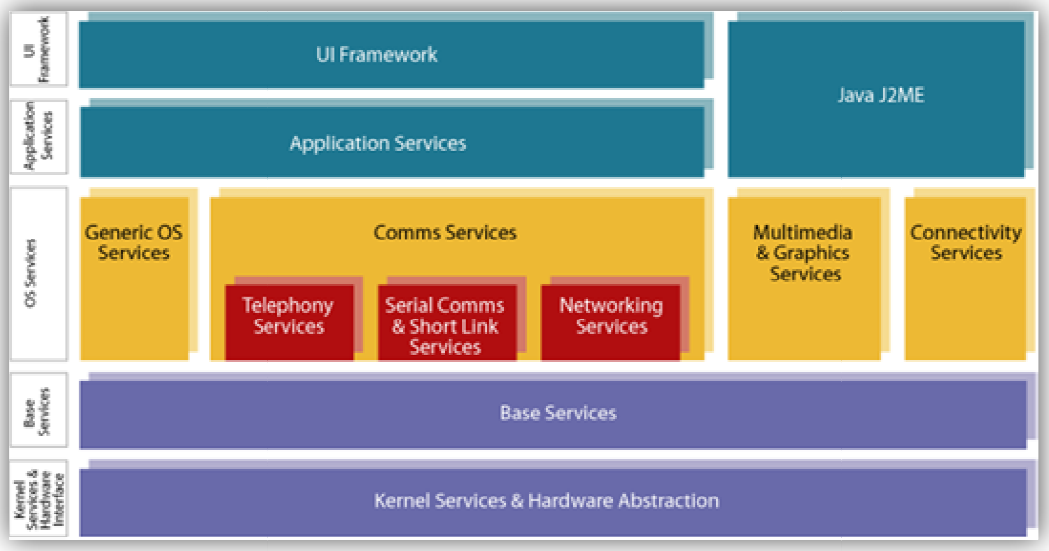
Εικόνα 8 : Visual Studio .NET IDE

Η τελευταία έκδοση τον Microsoft Windows Mobile (τα Windows Phone 7) τα οποία κυκλοφορήσαν στης αρχές του 2010 αφορούν έξυπνες φορητές συσκευές και έχουν σχεδιαστή για να καλύψουν και εκμεταλλευτούν τις προηγμένες δυνατότητες αυτών γεγονός που σημαίνει ότι οι εφαρμογές που φτιάχνονται για την τελευταία έκδοση δεν θα δύνανται να τρέξουν στις παλιότερες εκδόσεις (no backwards compatibility supported) του λειτουργικού όπως και το αντίθετο. Στην **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.** παρουσιάζεται ένα παράδειγμα διεπαφής στο περιβάλλον των Windows Phone 7 στα πλαίσια μιας τυχαίας έξυπνης φορητή συσκευής που το υποστηρίζει.

Εικόνα 9 : Windows Phone 7 σε φορητή συσκευή

3.2 Symbian OS (C++)

Το Symbian είναι ένα ανοιχτού κώδικα λειτουργικό σύστημα το οποίο σχεδιάστηκε για έξυπνες φορητές συσκευές και παρέχεται από την Nokia. Στην S60 5η έκδοση, περιλαμβάνεται η βιβλιοθήκη με την οποία μπορούμε να σχεδιάζουμε γραφικές διεπαφές σε αντίθεση με όλες τις προηγούμενες εκδόσεις στις οποίες αποτελούσε ένα πρόσθετο σύστημα. Το εν λόγω λειτουργικό σύστημα λειτουργεί αποκλειστικά σε επεξεργαστές ARM. Στην Εικόνα 10 βλέπουμε την αρχιτεκτονική του Symbian OS.



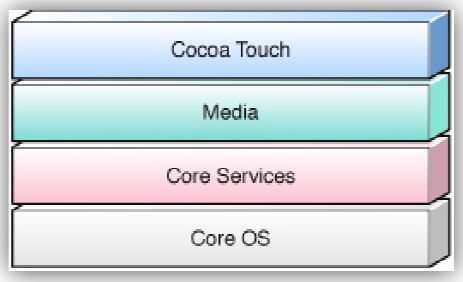
Εικόνα 10 : Αρχιτεκτονική Symbian OS

Για να μπορέσουμε να γράψουμε εφαρμογές για Symbian χρησιμοποιούμε την C++ με το Qt μέσω του Qt Creator ή του Carbide. Το Qt υποστηρίζει την 3η και την 5η έκδοση του Symbian καθώς και την νέα πλατφόρμα που υποστηρίζεται και από το MeeGo. Ένα εναλλακτικός τρόπος για την ανάπτυξη εφαρμογών βασίζεται στη χρήση της γλώσσας Python, Adobe Flash ή της Java ME. Παλιότερα το λειτουργικό σύστημα Symbian χρησιμοποιούσε την C++ με το Carbide C++ IDE ως βασική πλατφόρμα ανάπτυξης εφαρμογών. Στην **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.** βλέπουμε μια γραφική διεπαφή του λειτουργικού Symbian.

Εικόνα 11 : Γραφική διεπαφή Symbian OS

3.3 iOS (Objective C, C, C++)

Το iOS γνωστό ως iPhone OS είναι το λειτουργικό σύστημα της Apple. Αρχικά υλοποιήθηκε για το iPhone το οποίο ανήκει στην κατηγορία των έξυπνων κινητών συσκευών και έπειτα υποστήριξε και άλλες συσκευές όπως το iPod touch, iPad και το Apple TV. Η Apple δεν έχει δώσει την αδεία για να εγκατασταθεί το λειτουργικό σύστημα σε άλλες συσκευές από τρίτους κατασκευαστές. Η γραφική διεπαφή χρήστη (GUI) είναι βασισμένη στην έννοια του άμεσου χειρισμού, με την χρήση πολλαπλής αφής. Το iOS προέρχεται από την Mac OS X η οποία ακολουθεί το Darwin και συνεπώς είναι ένα λειτουργικό σύστημα τύπου Unix. Έχουμε τέσσερα επίπεδα αρχιτεκτονικής τα οποία φαίνονται στην Εικόνα 12.



Εικόνα 12 : Αρχιτεκτονική iOS

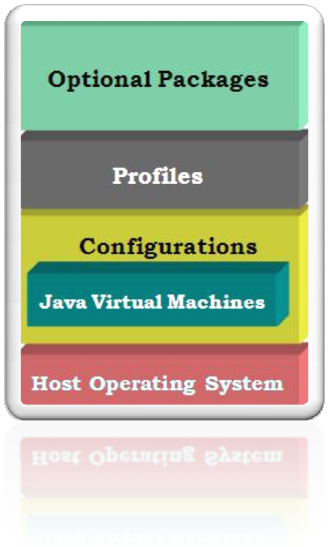
Η τελευταία έκδοση είναι η iPhone 4 στην οποία υποστηρίζεται το multitasking. Στις προηγούμενες εκδόσεις η δυνατότητα αυτή δεν περιλαμβάνονταν καθώς η Apple πίστευε ότι με την εκτέλεση πολλών εφαρμογών ταυτόχρονα θα υπήρχε μεγαλύτερη κατανάλωση μπαταριάς. Στην **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.** παρουσιάζεται μια ενδεικτική γραφική διεπαφή του iPhone OS.

Εικόνα 13 : Γραφική διεπαφή iOS

Οι εφαρμογές θα πρέπει να έχουν δημιουργηθεί σύμφωνα με το iOS αλλά και με την αρχιτεκτονική του ARM επεξεργαστή. Προκειμένου να κατασκευάσουμε δικές μας εφαρμογές χρειαζόμαστε το Xcode 3.1 με το οποίο γράφουμε κώδικα σε γλώσσα προγραμματισμού Objective-C. Επίσης μπορούμε να βρούμε εφαρμογές από το AppStore της Apple επί πληρωμή.

3.4 Έκδοση της Java2 για Κινητές Συσκευές: Cross-Platform (Java)

Η έκδοση J2ME της java αποτελεί την απάντηση της Sun Microsystems στην ανάπτυξη λογισμικού για κινητές συσκευές. Το J2ME είναι ακρωνύμιο το οποίο αναλύεται ως: έκδοση για κινητές συσκευές της Java 2 (Java 2 Mobile Edition). Βασιζόμενη στην java η J2ME κληρονομεί όλα τα βασικά χαρακτηριστικά της , όπως τη δυνατότητα εκτέλεσης σε πολλαπλές πλατφόρμες (multiplatform) και την αρχή ‘compile once rune everywhere’. Ωστόσο η J2ME δεν περιλαμβάνει/προσφέρει όλη τη λειτουργικότητα της standard έκδοσης της java για εφαρμογές Desktop δεδομένων των κατ’ αντιστοιχία υποδεέστερων σε υπολογιστικές δυνατότητες των κινητών συσκευών. Συγκεκριμένα η πολύ δημοφιλής και πανίσχυρη γραφική εργαλειοθήκη (GUI toolkit) Swing, δεν ενσωματώνεται στον κορμό της J2ME. Υπό το πρίσμα της ποικιλίας όσον αφορά την δυνητικώς παρουσιαζόμενη ετερομορφία μεταξύ των συνδυασμών που μπορεί να προκύψουν σε επίπεδο τόσο h/w (π.χ. διαφορετικοί κατασκευαστές-αρχιτεκτονικές, υποστήριξη για gps, κοκ.) όσο και s/w, η J2ME κάνει τους ακόλουθους διαχωρισμούς: ανάλογα με τις υπολογιστικές δυνατότητες της εκάστοτε συσκευής και των εγγενώς παρεχόμενων/υποστηριζόμενων βιβλιοθηκών λογισμικού. Συγκεκριμένα ανάλογα με τις υπολογιστικές δυνατότητες της εκάστοτε συσκευής, ο προγραμματιστής αρχικά καλείται να επιλέξει σε κατώτερο επίπεδο μεταξύ του configuration για limited (Connected for Limited Device Configuration: CLDC) ή enhanced ‘full featured’ devices (Connected for Device Configuration: CDC). Στην **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.** συνοψίζεται η αρχιτεκτονική της εν λόγο έκδοσης που υποστηρίζει πολλές κινητές συσκευές διαφορετικών κατασκευαστών.



Εικόνα 14 : Αρχιτεκτονική Java2 Mobile Edition

Στο επόμενο επίπεδο έχουμε τα προφίλ (profiles) τα οποία παρέχουν τα βασικά APIs που εστιάζουν σε μια συγκεκριμένη ομάδα κινητών συσκευών. Οι συσκευές αυτές έχουν συνήθως ίδιες ή παρόμοιες διεπαφές (GUI), σύνδεση στο δίκτυο, αποθήκευση δεδομένων κλπ. Ακόμα το όνομα τους δηλώνει το όνομα της ομάδας των συσκευών ή της κοινής λειτουργικότητας με αυτή την ομάδα συσκευών. Με βάση της δυο διαμορφώσεις υπάρχουν δυο οικογένειες προφίλ:

Στην διαμόρφωση CLDC έχουμε :

- **MIDP – Mobile Information Profile:** Χρησιμοποιείτε κυρίως για κινητές συσκευές και είναι ένα από τα βασικότερα προφίλ. Προσφέρει διεπαφή χρήστη, υποστηρίζει πολυμέσα και παιχνίδια, ασφάλεια καθώς και συνδεσιμότητα.
- **IMP – Information Module Profile:** Χρησιμοποιείτε από μικρότερες συσκευές από το MIDP όπως συστήματα ασφαλείας.
- **Digital set-top box profile:** Αφορά σε πολύ περιορισμένων υπολογιστικών δυνατοτήτων set top boxes.

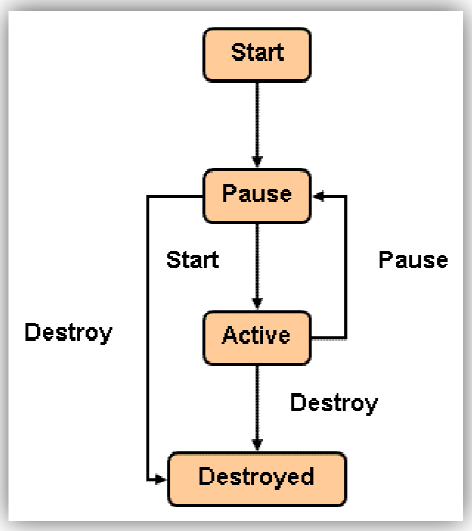
Στην διαμόρφωση CDC έχουμε:

- **Foundation Profile:** Είναι ένα σύνολο από Java APIs που υποστηρίζουν συσκευές χωρίς να παρέχεται κάποιο πρότυπο για την κατασκευή διεπαφών (GUI)
- **Personal Basis Profile:** Παρέχει εφαρμογές βασισμένες στο AWT με lightweight αντικείμενα.
- **Personal Profile:** Παρέχει ένα περιβάλλον ανάπτυξης εφαρμογών με βαρύ τύπου αντικείμενα του AWT. Το συγκεκριμένο προφίλ είναι η πλατφόρμα για τα web applets και είναι ένα μεταβατικό στάδιο πριν της προσωπικές εφαρμογές JAVA.

Το Mobile Information Device Profile (MIDP) είναι ένα προφίλ το οποίο υποστηρίζεται από τη CLDC διαμόρφωση και παρέχει το περιβάλλον στο οποίο μπορούμε να δημιουργήσουμε εφαρμογές. Υπάρχουν τα εξής δυο μοντέλα, το MIDlet σύμφωνα με το οποίο κατασκευάζουμε απλές εφαρμογές και μας παρέχει την δυνατότητα δημιουργίας γραφικής διεπαφής (GUI). Επίσης ένα άλλο μοντέλο είναι τα MIDlet Suites τα οποία έχουν περισσότερη ασφάλεια στα δεδομένα που μοιράζονται. Το MIDlet είναι εφαρμογή που χρησιμοποίει το MIDP προφίλ. Οι εφαρμογές αυτές μπορούν να τρέξουν παιχνίδια σε κινητές συσκευές που έχουν μικρές οθόνες, ένα απλό πληκτρολόγιο (αριθμητικό) καθώς και περιορισμένη σύνδεση στο δίκτυο μέσω HTTP.

Τα MIDlets τα διαχειρίζεται το λογισμικό διαχείρισης το οποίο είναι ενσωματωμένο στην συσκευή, επειδή όταν μια εφαρμογή τρέχει μπορεί να διακοπεί από κάποιο εξωτερικό γεγονός όπως μια εισερχόμενη κλήση. Ο διαχειριστής ελέγχει τις δραστηριότητες των πολλαπλών MIDlets στο περιβάλλον Java ME σε μια συσκευή ώστε να επιλέξει ποιες δραστηριοποιούνται σε συγκεκριμένο χρόνο (όταν ξεκινάνε ή όταν βρίσκονται σε παύση). Υπάρχουν τρείς πιθανές καταστάσεις στον χρόνο ζωής ενός MIDlet (βλέπε Εικόνα 15).

- Παύση (paused) – Το MIDlet έχει κατασκευαστή και είναι ανενεργό.
- Ενεργή (active) – Το MIDlet είναι ενεργό δηλαδή χρησιμοποιείται από τον χρήστη.
- Καταστράφηκε (destroyed) – Το MIDlet τερματίστηκε και είναι έτοιμο για ανάκτηση από τον garbage collector.



Εικόνα 15 : Καταστάσεις χρόνου ζωής ενός MIDlet

Ο διαχειριστής εφαρμογής με αίτηση του χρήστη δημιουργεί το MIDlet, το οποίο περνάει σε κατάσταση παύσης. Στην συνέχεια ενεργοποιείται και έτσι μεταβαίνει από την παύση στην ενεργή κατάσταση. Τέλος όταν λήξει μεταβαίνει στην καταστήσει καταστροφής είτε από την κατάσταση παύσης είτε από την ενεργή κατάσταση.

Για να δημιουργήσουμε ένα MIDlet χρησιμοποιούμε το Netbeans IDE το οποίο παρέχεται δωρεάν. Στην Εικόνα 16 φαίνεται ποιες μεθόδους πρέπει να εφαρμόσουμε προκειμένου να μπορέσει ο διαχειριστής της εφαρμογής να ελέγχει το MIDlet.

```
public class Application extends MIDlet {
    public Application() { }

    // Called when the MIDlet is created or re-started
    public void startApp() { }

    // Called to pause the MIDlet
    public void pauseApp() { }

    // Called to terminate the MIDlet
    public void destroyApp(boolean unconditional) { }
}
```

Εικόνα 16 : Κώδικας Κατασκευής MIDlet

Στο πρώτο επίπεδο της αρχιτεκτονικής της J2ME έχουμε τα προαιρετικά πακέτα τα οποία αποτελούν επιπρόσθετα APIs τα οποία χρησιμοποιούνται για να καλύψουν συγκεκριμένες λειτουργιές. Τέτοιου είδους λειτουργιά μπορεί να θεωρηθεί το GPS. Μερικά τέτοια πακέτα είναι τα έξεις:

- **Wireless Messaging API:** Χρησιμοποιεί τα χαρακτηριστικά του CLDC 1.0 και παρέχει την δυνατότητα συνδεσιμότητας.
- **File Connection:** Παρέχει πρόσβαση στο σύστημα διαχείρισης αρχείων των συσκευών η στην κάρτα μνήμης.
- **JDBC Connectivity:** Χρησιμοποιείτε για να επιτευχτεί η συνδεσιμότητα με βάσης δεδομένων και προσδιορίζεται από τις συσκευές που υποστηρίζουν την διαμόρφωση CDC καθώς και το Foundation Profile.
- **Mobile Media API:** είναι ένα API με υψηλό βαθμό αφαιρετικότητας έτσι ώστε να υποστηρίξει πληθώρα συσκευών με διαφορετικές πολυμεσικές δυνατότητες.
- **Web Services:** Παρέχει δυνατότητες πρόσβασης σε απομακρυσμένες υπηρεσίες βασισμένες σε SOAP και XML προσφέροντας τρόπους προσπέλασης των XML δεδομένων.
- **Security and Send services API:** Προσδιορίζει προαιρετικά πακέτα με σκοπό την παροχή ασφαλών και έμπιστων υπηρεσιών με την χρήση κρυπτογραφικών μεθόδων.
- **Location API:** Επιτρέπει λειτουργιές εντοπισμού θέσης από την συσκευή με την χρήση του GPS.
- **Event Tracking API:** έχει σαν στόχο να προσδιορίζει τον τρόπο με τον οποίο παράγονται γεγονότα από την συσκευή (explorer) σε ένα σύστημα συλλογής, ο τρόπος με τον οποίο θα

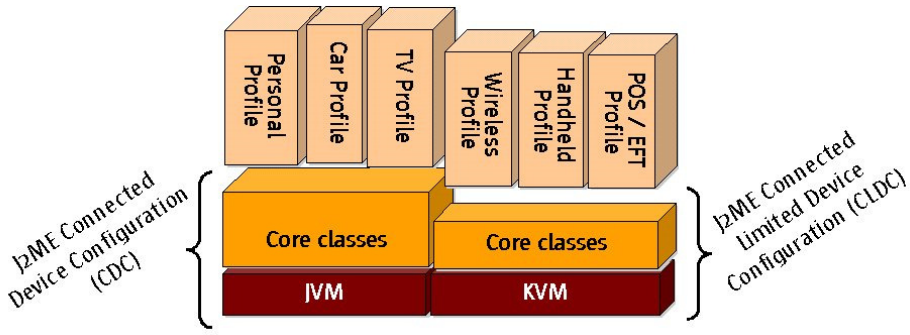
προσδιορίζονται οι διαμορφώσεις, το πρωτόκολλο που θα είναι υπεύθυνο μεταξύ της συσκευής και του συστήματος γεγονότων.

- **Content Handler API:** περιγράφει τον χειρισμό περιεχομένου καθώς και το μοντέλο εκτέλεσης. Το API και το μοντέλο επιτρέπουν σε μια εφαρμογή J2ME να εμπλέξει μια άλλη μη κτισμένη σε Java για περιεχόμενο με χρήση URL, ή τύπο περιεχομένου ή ID χειριστή περιεχομένου. Απαιτεί MIDP ή Personal Basis Profile.
- **Scalable 2D Vector Graphics API:** έχει σαν στόχο φορητές συσκευές χαμηλών δυνατοτήτων με περιορισμένη μνήμη, μέγεθος οθόνης και υπολογιστική ισχύ. Προσδιορίζει τον τρόπο δημιουργίας κλιμακωτών γραφικών δύο διαστάσεων, συμπεριλαμβανομένου εξωτερικές εικόνες τύπου SVG, με σκοπό την χρήση τους για γραφικές απεικονίσεις (visualizations), κλιμακωτά εικονίδια, και άλλες εφαρμογές που απαιτούν κλιμακωτά γραφικά.
- **DataSync API:** επιτρέπει σε εφαρμογές τον συγχρονισμό δεδομένων που βρίσκονται αποθηκευμένα στο τερματικό με τα αντίστοιχα δεδομένα που βρίσκονται αποθηκευμένα σε διακομιστεί έτσι ώστε να παρουσιάζονται οποιεσδήποτε αλλαγές σε οποιαδήποτε στιγμιότυπο των δεδομένων. Προϋποθέτει διαμόρφωση CLDC ή CDC.
- **Mobile Customization User Interface API:** επιτρέπει στους προγραμματιστές να μεταχειρίζονται την εμφάνιση της διεπαφής σε μία κινητή συσκευή.
- **Service Connection API:** παρέχει ομοιόμορφη πρόσβαση σε διαφορετικές υπηρεσίες οι οποίες είναι κτισμένες σε τεχνολογίες όπως AtomPub, Rest καθώς και σε απλούστερες όπως SOAP αλλά και εσωτερικές υπηρεσίες.
- **XML API:** Αφορά διαχείριση xml αρχείων
- **Telematics API:** παρέχει τηλεματικές υπηρεσίες από συσκευές που υποστηρίζουν την J2ME.

Η εκτέλεση προγραμμάτων γραμμένων σε java απαιτεί, όπως στις εκδόσεις για σταθερούς υπολογιστές, την ύπαρξη μιας εικονικής μηχανής προκειμένου να μεταφράζει το Java Byte Code σε γλωσσά που θα καταλαβαίνει εκάστοτε συσκευή (machine language). Οι πιο δημοφιλείς εικονικές μηχανές είναι οι:

- **KVM – Kilo Virtual Machine:** αποτελεί εξέλιξη της Java εικονικής μηχανής έχει γραφτή εξολοκλήρου σε C ώστε να έχει μικρό μέγεθος και βέλτιστη απόδοση, επίσης να μεταφέρετε εύκολα από πλατφόρμα σε πλατφόρμα. Την συγκειμένη πλατφόρμα χρησιμοποιεί η διαμόρφωση CLDC και απευθύνεται σε συσκευές με επεξεργαστή 16/32bit και συνολική μνήμη περίπου στα 256KB.
- **Hotspot Virtual Machine:** αποτελεί εξέλιξη της KVM με καλύτερη απόδοση και λιγότερες απαιτήσεις.

- **CVM:** Είναι η εικονική μηχανή που χρησιμοποιείται από την διαμόρφωση CDC. Περιλαμβάνει πλήρης εικονική μηχανή Java και μεγαλύτερο σύνολο από πηγαίες κλάσεις. Απαιτεί περισσότερη μνήμη από την KVM καθώς και ταχύτερο επεξεργαστή. Η CVM είναι πλήρης JVM σχεδιασμένη για συσκευές υψηλών επιδόσεων με 32bit επεξεργαστή και συνολική μνήμη μεγαλύτερη των 2MB.

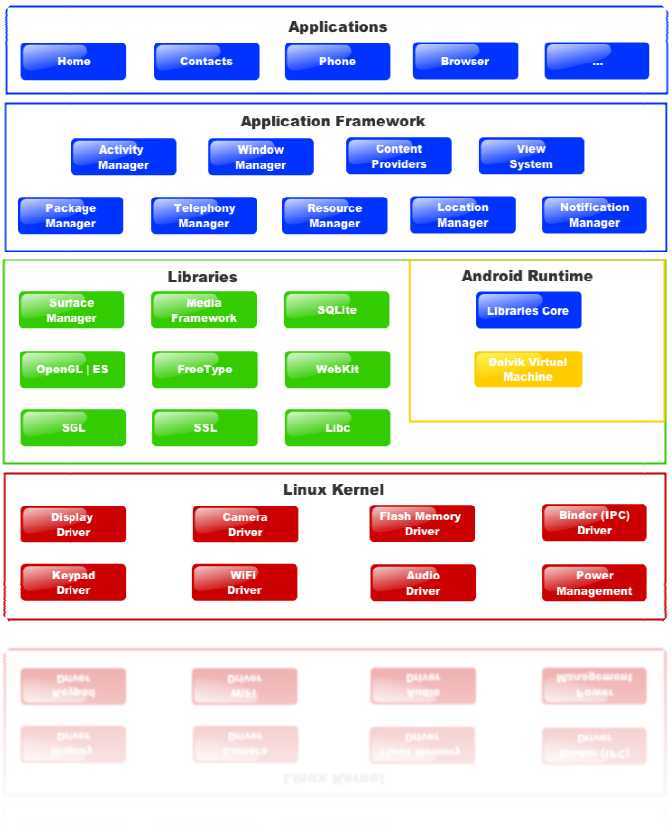


Εικόνα 17 : Κόσμος της J2ME

Στην Εικόνα 17 απεικονίζονται με γραφικό τρόπο τα επίπεδα αρχιτεκτονικής όπως αυτά αναφέρθηκαν παραπάνω.

3.5 Android OS (Java)

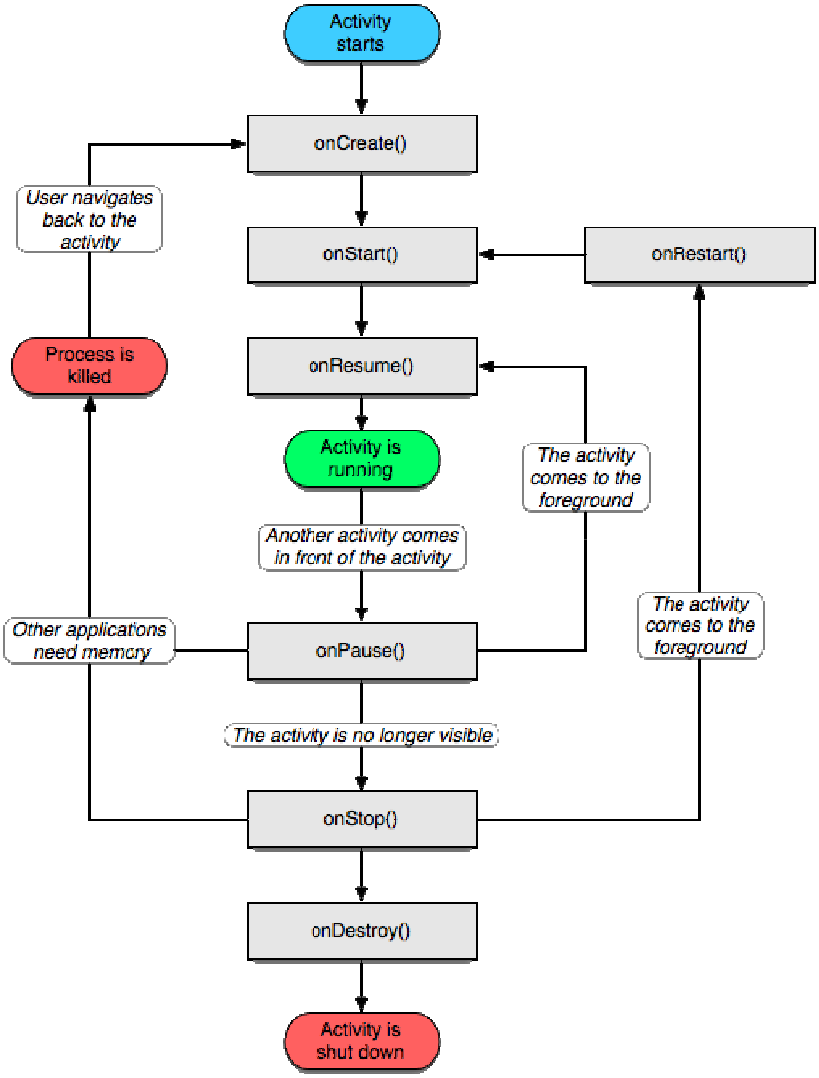
Το Android είναι το πιο δημοφιλές λειτουργικό σύστημα για έξυπνα κινητά τηλέφωνα, αναπτύσσεται από την Google, είναι βασισμένο σε πυρήνα linux, ενώ αξίζει να σημειωθεί ότι είναι ανοιχτού λογισμικού (open source). Το εν λόγω λειτουργικό επιτρέπει στους κατασκευαστές λογισμικού να γράφουν κώδικα ορίζοντας ως κύρια, γλώσσα προγραμματισμού τη java. Ωστόσο όπως είναι φυσικό υπάρχουν αρκετές διαφορές σε σχέση με τη standard edition (SE) έκδοση της java με χαρακτηριστικότερη την αντικατάσταση ολόκληρου του μηχανισμού ανάπτυξης διεπαφών (βλ.: AWT και Swing) με άλλο γηγενώς παρεχόμενο. Στην Εικόνα 18 βλέπουμε την αρχιτεκτονική του Android.



Εικόνα 18 : Αρχιτεκτονική του λειτουργικού συστήματος Android

Το android χρησιμοποιεί ένα εξειδικευμένο τρόπο προκειμένου να διαχειριστεί όσον τον δυνατόν καλύτερα τους πόρους του συστήματος. Συγκεκριμένα εισάγει και αναδεικνύει την έννοια της δραστηριότητας (Activity) ώστε να περιορίσει – χειριστεί με το βέλτιστο δυνατό τρόπο των αριθμό των ταυτόχρονα ενεργών διεργασιών και συνήθως αντίστοιχων συσχετιζόμενων με αυτές διεπαφών. Κάθε activity έχει το δικό του User Interface (UI) καθώς και κατάσταση όσον αφορά τον ενιαία προκαθορισμένο κύκλο ζωής των. Συγκεκριμένα οι καταστάσεις στις οποίες μπορεί να περιέλθει, με αυτοματοποιημένο τρόπο, ανάλογα της περιπτώσεως μια δραστηριότητα είναι οι εξής:

- Αν μια δραστηριότητα είναι σε πρώτο πλάνο (δηλαδή είναι ενεργή ή τρέχει)
- Αν μια δραστηριότητα έχει χάσει την εστίαση της αλλά εξακολουθεί να είναι ορατή (δηλαδή μια νέα δραστηριότητα όχι πλήρους μεγέθους ή διαφανές έχει επικεντρωθεί στην κορυφή) , είναι σε παύση.
- Αν μια δραστηριότητα επισκιάζεται εντελώς από μια άλλη δήλωση έχει σταματήσει. Διατηρεί ακόμα πληροφορίες κατάστασης αλλά δεν είναι πλέον ορατή στον χρήστη.
- Αν μια δραστηριότητα έχει διακοπεί ή να σταματήσει.



Εικόνα 19 : Κύκλος ζωής ενός Activity

Η **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.** δείχνει τα σημαντικά μονοπάτια κατάστασης μιας δραστηριότητας. Τα ορθογώνια τετράγωνα αντιπροσωπεύουν τις μεθόδους επανάκτησης που μπορούμε να εφαρμόσουμε για την εκτέλεση λειτουργιών, στην περίπτωση που η δραστηριότητα αλλάζει καταστάσεις. Τα χρωματιστά τετράγωνα είναι οι καταστάσεις μιας δραστηριότητας.

Η ανάπτυξη διεπαφών μπορεί να γίνει με δύο διαφορετικούς τρόπους:

- Δηλώνοντας τα εμπλεκόμενα διαδραστικά αντικείμενα κάνοντας χρήση της XML γλώσσας. Το android παρέχει ένα απλό λεξιλόγιο (βασισμένο σε XML) που κάνει την αντιστοίχιση με τα κατάλληλα UI Elements (π.χ. ένα κουμπί). Ένα παράδειγμα ενός τέτοιου XML εμφανίζεται στην Εικόνα 20:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
              android:layout_width="fill_parent"
              android:layout_height="fill_parent"
              android:orientation="vertical" >
    <TextView android:id="@+id/text"
              android:layout_width="wrap_content"
              android:layout_height="wrap_content"
              android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Hello, I am a Button" />
</LinearLayout>
```

Εικόνα 20: Παράδειγμα ορισμού διεπαφής με χρήση XML

- Δημιουργία κατά την εκτέλεση. Δίνετε η δυνατότητα κατασκευής ενός UI κάνοντας χρήση της κλασικής προγραμματιστικής διαδικασίας.

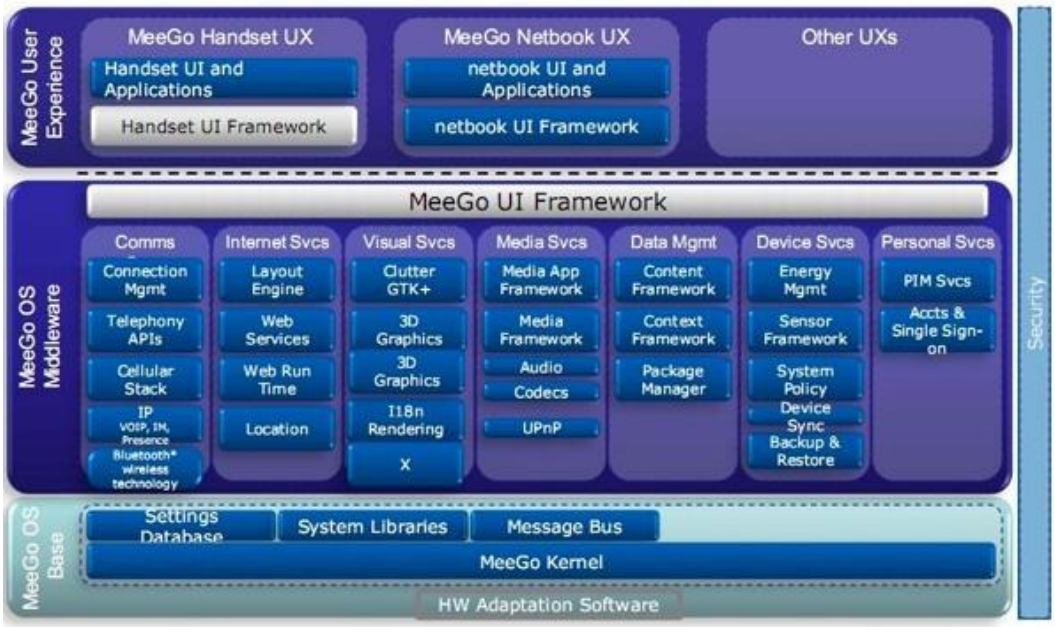
Για να μπορέσουμε να δημιουργήσουμε μια εφαρμογή Android (ενδεικτικά βλέπε Εικόνα 21) μπορούμε να χρησιμοποιήσουμε το Eclipse IDE όπως επίσης το Netbeans IDE εγκαθιστώντας το κατάλληλο πρόσθετο (plugin) κάθε φορά, ενώ αξίζει να σημειωθεί ότι και τα δύο εργαλεία παρέχονται δωρεάν.

Εικόνα 21 : Γραφική διεπαφή Android

Ένα από τα βασικά στοιχεία του android είναι το γεγονός ότι οι εφαρμογές του τρέχουν ανεξάρτητα της έκδοσης του λειτουργικού που είναι εγκατεστημένο σε κάθε συσκευή. Η τελευταία έκδοση είναι το Android 2.3 (έχοντας κωδικό όνομα: 'Gingerbread') στην οποία ενσωματώθηκαν επιπλέον κατάλληλα APIs προκειμένου να υποστηριχθεί η ανερχόμενη τεχνολογία με όνομα 'Near Field Communication' (NFC) [5)].

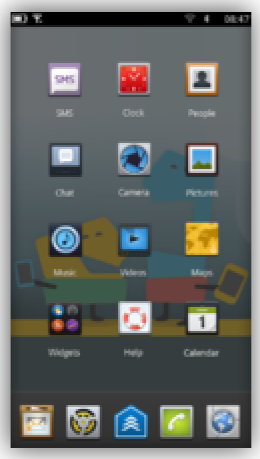
3.6 MeeGo – Linux

Το MeeGo είναι ένα ανοιχτού κώδικα λειτουργικό σύστημα το οποίο χρησιμοποιεί τα Linux. Δημιουργήθηκε με στόχο αρχικά να υποστηρίξει της κινητές συσκευές και συσκευές πληροφοριών στην αγορά ηλεκτρονικών ειδών ευρείας κατανάλωσης. Έχει σχεδιαστεί ως λειτουργικό σύστημα για πλατφόρμες υλικού όπως notebooks, tablets, έξυπνα τηλεφωνά και αλλά ενσωματωμένα συστήματα. Σήμερα η MeeGo φιλοξενείται από την Linux επίσης αποτέλεσε την συγχώνευση της Mobin της Intel και της Maemo της Nokia. Επίσης υποστηρίζει ARM και Intel x86 επεξεργαστές με B-tree σύστημα αρχείων και χρησιμοποιεί τα RPM repositories. Στην Εικόνα 22 βλέπουμε την αρχιτεκτονική του MeeGo.



Εικόνα 22 : Αρχιτεκτονική MeeGo

Το MeeGo παρέχει διαφορετικές γραφικές διεπαφής (GUI) ανάλογα με την πλατφόρμα τις όποιες της ονομάζει έμπειρες χρηστών (User Experiences - UX). Για να σχεδιάσουμε γραφικές διεπαφής σε μια φορητή συσκευή (βλέπε Εικόνα 23) χρησιμοποιούμε το Handset UX το οποίο είναι βασισμένο στο Qt επίσης μπορούμε να χρησιμοποιήσουμε το GTK+ και το Clutter.



Εικόνα 23 : MeeGo GUI

Διατίθενται εφαρμογές στους χρηστές της οποίες μπορούν να κατεβάσουν τόσο από το AppUp της Intel όσο και από το Ovi της Nokia, οι οποίες για να εγκατασταθούν χρειάζονται μια βιβλιοθήκη που ονομάζεται Hildon. Ο επίσημος τρόπος για την ανάπτυξη εφαρμογών είναι να χρησιμοποιήσουμε το Qt framework και το Qt Creator ως περιβάλλον ανάπτυξης αλλά μπορούμε να χρησιμοποιήσουμε και το GTK το οποίο υποστηρίζεται εξίσου.

4. Ανάπτυξη Διεπαφών για Κινητές Συσκευές

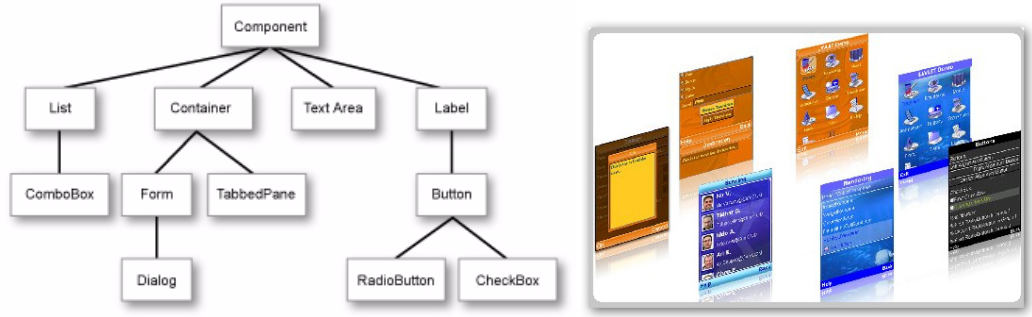
Σε αυτό το κεφάλαιο εστιάζουμε στην ανάπτυξη διεπαφών για κινητές συσκευές. Συγκεκριμένα πέρα των γηγενώς υποστηριζόμενων από κάθε λειτουργικό σύστημα ή γλώσσα προγραμματισμού εργαλείων για την ανάπτυξη λογισμικού, εστιαζόμαστε σε πρόσθετες γραφικές βιβλιοθήκες που μπορεί να χρησιμοποιηθούν ανά περίπτωση. Επίσης αναπτύσσουμε και εμμέσως αντιπαραβάλουμε αυτές που είναι βασισμένες στην κυρίαρχη μέθοδο ανάπτυξης διεπαφών δηλαδή τον προγραμματισμό εργαλειοθήκης (Toolkit-based programming) με αυτές που υπακούουν στην αναδυόμενη προσέγγιση της ανάπτυξης βάσει μοντέλων (model-based programming).

4.1 Toolkit-Based Programming Προσέγγιση

Οι περισσότερες πλατφόρμες που περιγράφηκαν στο προηγούμενο κεφάλαιο δίνουν απευθείας τη δυνατότητα ανάπτυξης γραφικών διεπαφών χωρίς την ανάγκη εισαγωγής πρόσθετων. Πιο συγκεκριμένα παρέχουν ενσωματωμένες εργαλειοθήκες (toolkits) που υλοποιούν κατάλληλες αφαιρετικές προγραμματιστικές δομές καθιστώντας την ανάπτυξη διεπαφών εύκολη και τυπική (κοινό API, κοινή φιλοσοφία). Ωστόσο πέρα της προσέγγισης που βασίζεται στον προγραμματισμό εργαλειοθήκης με καθιερωμένα διαδρστικά αντικείμενα υπάρχουν και εργαλεία που αναπτύσσονται από τρίτους με στόχο είτε την απόδοση νέων ενδεχομένως εξειδικευμένων διαδραστικών αντικειμένων, είτε βελτιωμένων εκδόσεων ήδη υποστηριζόμενων. Παρακάτω συνοψίζονται μερικά από τα αντιπροσωπευτικά εργαλεία της κατηγορίας αυτής.

4.1.1 LWUIT Toolkit

Το LWUIT είναι μια εργαλειοθήκη που αφορά την ανάπτυξη διεπαφών για κινητές συσκευές που υποστηρίζουν είτε τη CLDC είτε τη CDC διαμόρφωση. Είναι επομένως εμφανές ότι αφορά J2ME εφαρμογές με κύριο στόχο την παροχή API μέσω συνόλου προσφερόμενων αντικειμένων που προσιδιάζουν αυτά του Swing. Στην Εικόνα 24 (αριστερή πλευρά) φαίνεται η βασική ιεραρχία των υποστηριζόμενων γραφικών αντικειμένων.



Εικόνα 24 : Ιεραρχία των τμημάτων και γραφικές διεπαφές του LWUIT

Επίσης υποστηρίζει τη δυνατότητα εφαρμογής πλήθους υποστηριζόμενων θεμάτων (themes) όπως φαίνεται στην δεξία πλευρά της παραπάνω εικόνας. Ο κεντρικός υποδοχέας κάθε διεπαφής είναι η φόρμα πάνω στην οποία τοποθετούνται τα αντικείμενα η οποία με τη σειρά της συνδέεται με την κλάση

Display που είναι υπεύθυνη για την αλληλεπίδραση με το χρήστη. Επίσης η διαχείριση γεγονότων είναι υλοποιημένη με φιλοσοφία που να είναι όσο το δυνατόν πι οικεία και εύκολη για προγραμματιστές που είναι ήδη εξοικειωμένοι με την εργαλειοθήκη java/swing. Τέλος το LWUIT υλοποιεί τις κλάσεις προδιαγραφής του προφίλ MIDP (που ήδη έχει περιγραφεί).

4.1.2 Apime

Το Apime είναι μια εργαλειοθήκη μέσω της οποία μπορούμε να δημιουργήσουμε γραφικές διεπαφές. Αφορά κινητές συσκευές που υποστηρίζουν Java (J2ME) και είναι συμβατό με την έκδοση MIDP 1.0 καθώς και το MIDP 2.0.

Εικόνα 25 : Διεπαφές με χρήστη του Apime

Οι γραφικές διεπαφές δημιουργούνται με τα γηγενή της J2ME αντικείμενα ενώ επίσης περιλαμβάνει και κλάσεις για την διαχείριση αρχείων. Τέλος, με την χρήση του μπορούμε να δημιουργήσουμε διαφορετικά είδη εφαρμογών εύκολα και γρήγορα. Στην Εικόνα 25 βλέπουμε πως εμφανίζεται μια γραφική διεπαφή που δημιουργήθηκε με τη χρήση της εργαλειοθήκης Apime.

4.1.3 jMobileCore

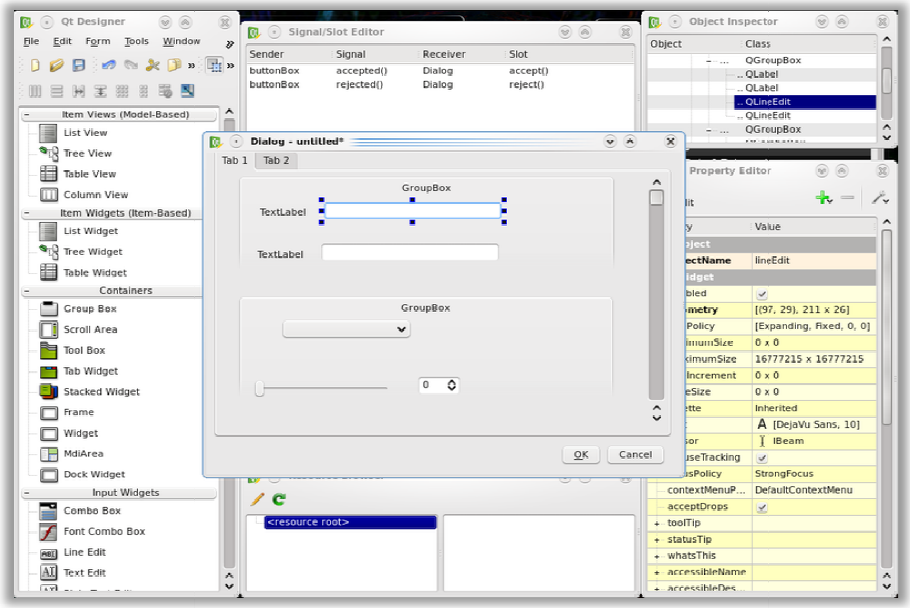
Το jMobileCore είναι ένα ισχυρό εργαλείο για την δημιουργία εφαρμογών J2ME. Παρέχει τη δυνατότητα δημιουργίας γραφικών διεπαφών, γρήγορη πρόσβαση στα δεδομένα, αξιόπιστες επικοινωνίες και απλοποιημένες διαδικασίες για την κατασκευή (πολυνηματικών) multithreading εφαρμογών MIDlet.

Εικόνα 26 : Παράδειγμα διεπαφής jMobileCore

Η συγκεκριμένη εργαλειοθήκη λειτούργει σε οποιαδήποτε κινητή ή PDA συσκευή που υποστηρίζει J2ME με MIDP 1.0 και CLDC 1.0. Ένα παράδειγμα διεπαφής υποστηριζόμενη από τη συγκεκριμένη εργαλειοθήκη εμφανίζεται στην **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.**.

4.1.4 Qt Toolkit

Η Qt είναι ένα cross-platform framework που χρησιμοποιείται ευρέως για την ανάπτυξη τόσο γραφικών όσο και μη-γραφικών διεπαφών. Κατασκευάστηκε από την Nokia και χρησιμοποιείται από μεγάλες εταιρίες σε διάφορα προϊόντα όπως Google Earth, Adobe Photoshop Album, Skype κ.α. Στην Εικόνα 27 απεικονίζεται ένα στιγμιότυπο του ολοκληρωμένου περιβάλλοντος για την σχεδίαση γραφικών διεπαφών που βασίζονται στο Qt.



Εικόνα 27 : Περιβάλλον Qt

Το Qt είναι βασισμένο στην γλωσσά προγραμματισμού C++. Αξιοποιεί μια ειδική γεννήτρια κώδικα (code generator) που ονομάζεται Meta Object Compiler ή MOC που μαζί με μακροεντολές προσφέρουν τη δυνατότητα επιπρόσθετου εμπλουτισμού της γλώσσας. Τρέχει σε όλες τις σημαντικές πλατφόρμες ενώ πέρα των γραφικών δυνατοτήτων της (βλ.: gui toolkit), παρέχει τη δυνατότητα πρόσβασης σε βάσεις δεδομένων (database connectivity), XML parsing, υποστήριξη δικτυακής επικοινωνίας (networking) καθώς και ένα cross-platform API για των χειρισμό αρχείων. Το περιβάλλον είναι ανοιχτού κώδικα και υποστηρίζεται από πολλούς compilers όπως των GCC C++ και του Visual Studio Suite. Στης μέρες μας χρησιμοποιείται από την Nokia για την κατασκευή γραφικών διεπαφών που χρησιμοποιούνται στα λειτουργικά συστήματα Symbian και MeeGo.

4.2 Model-Based UI Engineering Προσέγγιση

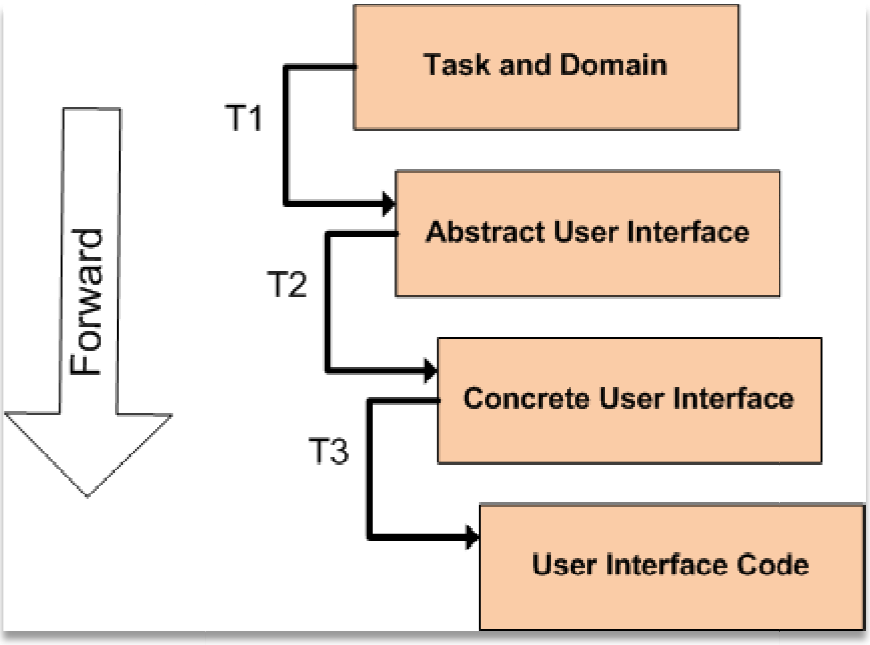
Η μηχανική της ανάπτυξης διεπαφών βασιζόμενων σε μοντέλα συνιστά μια εναλλακτική προσέγγιση κατασκευής διεπαφών που επιδιώκει αφενός να διευκολύνει την ανάπτυξη και επαναπροσαρμογή ενός συστήματος σε μελλοντικές απαιτήσεις και αφετέρου στοχεύει στην δημιουργία διεπαφών για πολλαπλά περιβάλλοντα, π.χ. mobile, web, desktop, κοκ. και πλατφόρμες όπως π.χ.: java/swing, windows/mfc, κοκ.. Το τελευταίο ωστόσο προβάλλει σημαντικούς περιορισμούς που επιτρέπουν τη δυνατότητα περιγραφής και υποστήριξης απλοϊκών διεπαφών που αξιοποιούν ένα πολύ περιορισμένο αριθμό διαδραστικών αντικειμένων που υποστηρίζονται αμιγώς (native) από την εκάστοτε πλατφόρμα αναφοράς (π.χ.: κουμπιά, radio buttons, και λίγα ακόμη).

Η ανάπτυξη γίνεται με χρήση υψηλού επιπέδου μοντέλων τα οποία δρουν ως επίπεδα αφαίρεσης επιτρέποντας στους σχεδιαστές να στοχεύσουν, αναλύσουν, δώσουν έμφαση και να προσδιορίσουν διαφορετικές κατασκευαστικές συνιστώσες στο κύκλο ζωής ενός συστήματος. Με τον τρόπο αυτό

επιχειρείται ο διαχωρισμός του εύρους των θεμάτων που αφορούν την κατασκευή και υλοποίηση διαδραστικών συστημάτων (separation of concerns).

Ειδικότερα ανάλογα με το τρέχων επίπεδο αφαίρεσης δεν ασχολούμαστε με χαμηλού επιπέδου λεπτομέρειες της εφαρμογής αλλά με υψηλού επιπέδου θέματα[1], όπως:

- **Μοντέλο διεργασίας και αντικείμενου** (Task and Object model): εδώ περιγράφονται οι αλληλεπιδραστικές απαιτήσεις ενός συστήματος από την οπτική γωνία του χρήστη, υπό τη μορφή καθηκόντων που πρέπει να εκτελεστούν από τον εκάστοτε χρήστη προκειμένου να επιτευχθεί ο εκάστοτε στόχος του.
- **Μοντέλο αφηρημένης γραφικής διεπαφής** (Abstract User Interface): σε αυτό το επίπεδο αναλύεται η γραφική διεπαφή με τρόπο ανεξάρτητο μέσου (modality - independent).
- **Μοντέλο συγκεκριμένης γραφικής διεπαφής** (Concrete User Interface): στο συγκεκριμένο επίπεδο αφαίρεσης ορίζονται με τρόπο ανεξαρτήτου πλατφόρμας (platform independent) συγκεκριμένοι τύποι διαδραστικών αντικειμένων που είναι να χρησιμοποιηθούν, όπως π.χ. κουμπιά, κοκ. .
- **Τελική γραφική διεπαφή** (Final User Interface): το συγκεκριμένο επίπεδο δεν περιγράφεται με μοντέλα και αποτελεί είτε άμεσα εκτελούμενο είτε διερμηνευόμενο κώδικα.

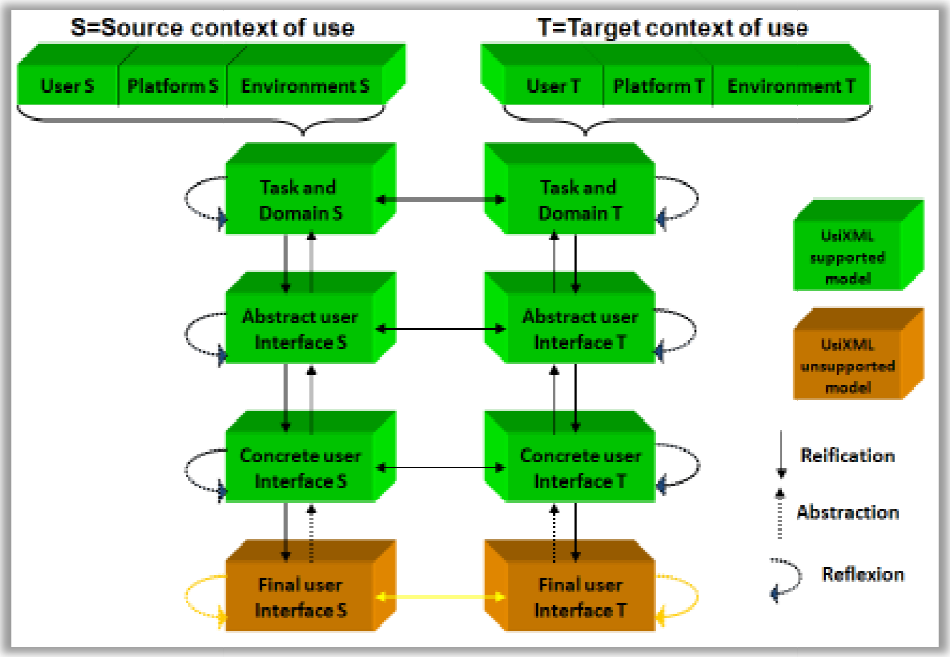


Εικόνα 28 : Επίπεδα αφαίρεσης Model-based UI

[1] Όπως αυτά ορίζονται από το Cameleon Reference Framework, βλ.: [6)]

Μια τυπική διασύνδεση των επιπέδων αφαίρεσης και των βασικών μοντέλων που τα υποστηρίζουν συνοψίζεται στην Εικόνα 28. Οι επιμέρους λεπτομέρειες κάθε μοντέλου και ο τρόπος που αξιοποιείται για τη σχεδίαση διεπαφών εξαρτάται από το εκάστοτε σύστημα ή/και προσέγγιση. Ακολούθως περιγράφεται μια συγκεκριμένη προσέγγιση που βασίζεται στην UsiXML και η οποία αξιοποιήθηκε στα πλαίσια της παρούσας εργασίας.

Η UsiXML (USer Interface eXtensible Markup Language) είναι μια από της πιο δημοφιλής γλώσσες περιγραφής διεπαφών (UIDL). Αποτελεί μια προσπάθεια που στοχεύει σε μια υλοποίηση των οδηγιών του Cameleon Framework το οποίο στοχεύει στην ανάδειξη του plasticity μέσω της προτεινόμενης δομημένης μεθοδολογίας. Επίσης συνοδεύεται από πλήθος εργαλείων για να διευθετεί διαφορετικά ζητήματα κατά την φάση ανάπτυξης μιας διεπαφής. Η μεθοδολογία που υιοθετείται είναι συμβατή με την γενικότερη προσέγγιση ανάπτυξης διεπαφών βάσει μοντέλων (MDE compliant) ενώ αξιοποιεί διαφορετικά αφαιρετικά επίπεδα για να καταστήσει εφικτή την περιγραφή ενός συστήματος με τρόπο ανεξάρτητο της υπολογιστικής υλοποίησης (Computation Independent Model), ανεξάρτητο (Independent) αλλά και εξαρτημένο (Dependent) από την πλατφόρμα (Platform Model). Στην **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.** βλέπουμε τα μοντέλα που υλοποιούν τις αρχές του MDE, τις οποίες χρησιμοποιεί η UsiXML ως υλοποίηση του Cameleon Reference Framework.

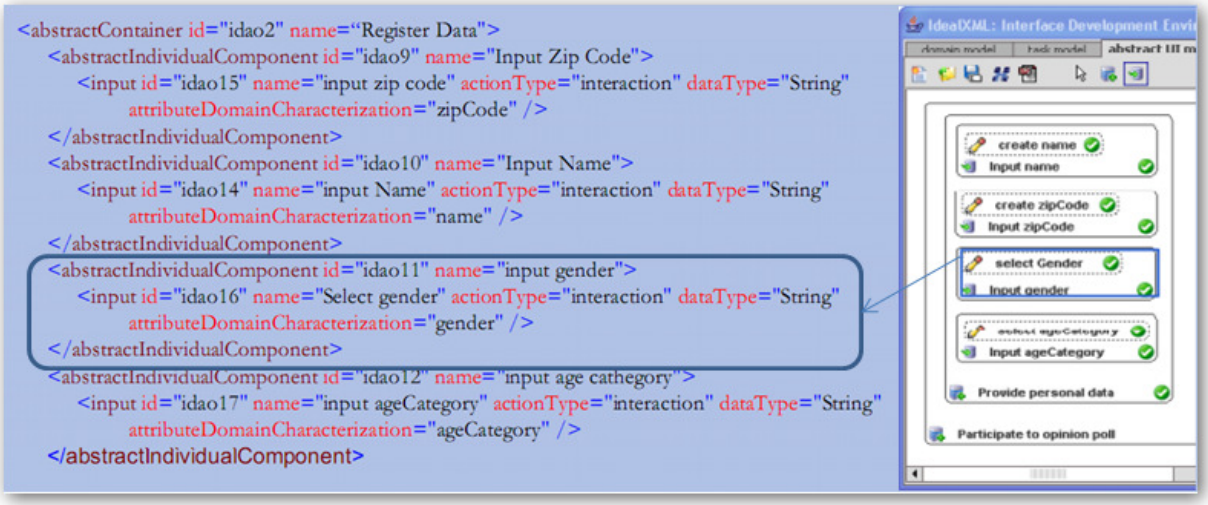


Εικόνα 29 : MDE Cameleon reference framework

Ένα πλεονέκτημα της υλοποίησης διεπαφών σε ξεχωριστές φάσης ανάπτυξης είναι αυτό που ονομάζεται 'separation of concerns'. Αυτό είναι πολύ σημαντικό στην περίπτωση που έχουμε διεπαφές που επιδεικνύουν πλαστικότητα στη χρήση, αφού πλήθος διαφορετικών απαιτήσεων πρέπει να περιγραφτούν κάθε φόρα για κάθε ένα από τα υποστηριζόμενα περιβάλλοντα. Το περιβάλλον ορίζεται από τρία πράγματα, πλατφόρμα, περιβάλλον και στερεότυπου του χρήστη (User Stereotype). Μέσω

μιας διαγραμματικής τεχνικής που αποδίδει ιεραρχίες ανάλυσης καθηκόντων η τεχνική ονομάζεται (Concur Task Trees), δίνεται η δυνατότητα προσδιορισμού του συνόλου τον απαιτήσεων του συστήματος από πλευράς χρηστών που εκτελούνται σε διαφορετικά περιβάλλοντα. Στην συνεχεία το μοντέλο μετασχηματίζεται με τρόπο κατάλληλο έτσι ώστε να προσδιορίσουμε πιο συγκεκριμένα την διεπαφή. Σε αυτό το στάδιο η προηγούμενη περιγραφή καθηκόντων μετατρέπετε σε μορφή αφηρημένων υποδοχέων (Containers) και διαδραστικών αντικειμένων με τρόπο ανεξάρτητο από το κανάλι (modality independent) και την πλατφόρμα (platform independent). Λόγο της πλαστικότητας που μας παρέχει η UsiXML, δίνεται η δυνατότητα στον χρήστη μέσω της τελικής διεπαφής να αλληλεπιδράσει δυνητικώς χρησιμοποιώντας παράλληλα πολλαπλά κανάλια επικοινωνίας (multimodal interaction). Στην Εικόνα 30 βλέπουμε ένα παράδειγμα περιγραφής μιας τέτοιας διεπαφής.

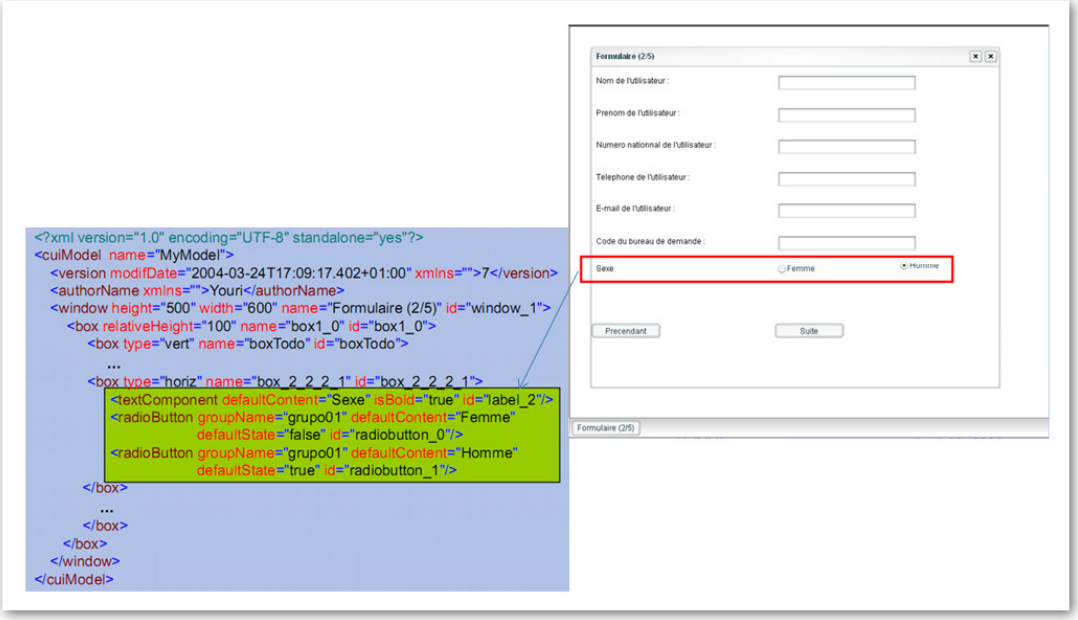
Προκειμένου να μεταβαίνουμε μεταξύ τον διαφορετικών μοντέλων χρησιμοποιούμε την βοήθεια εξειδικευμένων μετασχηματισμών (transformations) και μηχανισμών συσχετίσεων (mappings). Υπάρχουν πολλά είδη μετασχηματισμών που βασίζονται σε μαθηματικά μοντέλα. Η UsiXML χρησιμοποιεί μια εξειδικευμένη μορφή αυτών για να μετασχηματίζει γράφους (graph transformations). Οι μετασχηματισμοί επιτρέπουν την αυτοματοποιημένη ή ημι-αυτοματοποιημένη, ανάλογα με την περίπτωση, μετάβαση των απαιτήσεων από το ένα επίπεδο αφαίρεσης στο άλλο. Αξίζει να σημειωθεί στο σημείο αυτό ότι οι μετασχηματισμοί είναι ένα κύριο χαρακτηριστικό γνώρισμα των MDE προσεγγίσεων και αποτελούν πλεονέκτημα στην ανάπτυξη διεπαφών που βασίζονται σε μοντέλα (Model-based UI Engineering).



Εικόνα 30 : Αφηρημένο μοντέλο (AUI) σε XML και η γραφική αναπαράσταση

Η UsiXML υποστηρίζει τη λεγόμενη 'ανάπτυξη πολλαπλής διαδρομής' (Multipath development) βάση της οποίας δίνεται η δυνατότητα στο χρήστη να ξεκινήσει την ανάπτυξη μιας διεπαφής από το οποιοδήποτε υποστηριζόμενο επίπεδο αφαίρεσης επιθυμεί. Αυτό επιτυγχάνεται μέσω περιγραφής της διεπαφής με τρόπο ανεξάρτητο από την πλατφόρμα αξιοποιώντας μια γκάμα βασικών δημοφιλών

διαδραστικών αντικειμένων που είναι κοινά σε πολλές πλατφόρμες όπως κουμπιά, λίστες κ.ο.κ. Παράδειγμα περιγραφής μιας διεπαφής στο επίπεδο αυτό απεικονίζεται στην Εικόνα 31.



Εικόνα 31 : Το CUI μοντέλο σε XML και σε διεπαφή

Παρόλο που η UsiXML αποτελεί μια από τις πλέον δημοφιλής προσεγγίσεις ένα από τα σημαντικότερα προβλήματα της, αλλά και όλων των γλωσσών αυτής της κατηγορίας, είναι η αδυναμία τους να υποστηρίξουν μη-συμβατικές (non-native) διεπαφές. Αυτό καθιστά αδύνατη ή στην καλύτερη των περιπτώσεων επίπονη τη διαδικασία χρησιμοποίησης εξειδικευμένων διαδραστικών αντικειμένων πέρα από τα δημοφιλή που παρέχονται από τις πιο δημοφιλής πλατφόρμες, εξαιτίας κυρίως των αποκλίσεων και διαφορετικών αφαιρετικών υποθέσεων κάθε περιβάλλοντος. Για την αντιμετώπιση της αδυναμίας αυτής η UsiXML κάνει χρήση επιπλέον μοντέλων ώστε να δώσει τη δυνατότητα μοντελοποίησης των υποστηριζόμενων περιβαλλόντων, όπως επίσης και της μοντελοποίησης οντοτήτων πεδίου (domain object modeling). Όσο αφορά το τελευταίο χρησιμοποιούνται διαγράμματα κλάσεων τα οποία συσχετίζονται με τη διεπαφή αλληλεπιδρώντας με το back-end της εφαρμογής ώστε να προσδώσουν ουσιαστική λειτουργική αξία στη διεπαφή.

5. Υλοποίηση

Σε αυτό το κεφάλαιο θα αναλύσουμε την υλοποίηση της πτυχιακής καθώς και τα σενάρια χρήσης που υποστηρίζονται. Δεδομένης της ανάγκης για λογισμικό ικανό να προσαρμόζεται και να εκτελείται διαφανώς ανεξαρτήτως περιβάλλοντος θεωρούμε ότι η προσέγγισή μας μπορεί να ωφεληθεί υιοθετώντας μια βασισμένη σε μοντέλα προσέγγιση (model-based) και συγκεκριμένα από την πολύ δημοφιλή γλώσσα περιγραφής διεπαφών UsiXML, που ήδη έχει περιγραφεί. Για να ικανοποιήσουμε τις απαιτήσεις για ικανότητα εκτέλεσης σε περιβάλλοντα έξυπνων συσκευών επιλέξαμε να υποστηρίξουμε το πιο δημοφιλές λειτουργικό σύστημα ενδεδειγμένο για αυτές, το Google Android.

Για τη διαχείριση συνόδων και δεδομένων σχετιζόμενων με αυτές, όπως οι τρέχοντες ανά σύνοδο χρήστες καθώς και οι αφηρημένες περιγραφές των πολυχρηστικών διεπαφών αναπτύχθηκαν κατάλληλα τμήματα λογισμικού από πλευράς server (server-side). Ο server που χρησιμοποιήσαμε είναι ο apache Tomcat ενώ οι προσφερόμενες υπηρεσίες διατίθενται υπό τη μορφή υπηρεσιών (services) εκτελούμενων στα πλαίσια του axis2 πλαισίου εργασίας (framework). Ορισμένες από τις υπηρεσίες που αναπτύχθηκαν είναι οι: Εξακρίβωσης στοιχείων (AuthenticationService) που ελέγχει το αν ένας χρήσης έχει δικαίωμα να εισέλθει στο σύστημα, να εισέλθει να δημιουργήσει μια σύνοδο ή να συσχετίσει διεπαφές στα πλαίσια της εκάστοτε, είναι επίσης η υπηρεσία τοποθεσίας (LocationSevice) η οποία είναι παρέχει μέσω ενός κατάλληλου api τη δυνατότητα συσχέτισης των χρηστών με την εκάστοτε, δυνητικώς δυναμικά μεταβαλλόμενη, τοποθεσία των ώστε να υποστηρίξει την λεγόμενη τοποθεσιακή ενημερότητα (Location Awareness). Τέλος, η υπηρεσία διαχείρισης συνεργατικών συνόδων (Collaborative Session management) είναι υπεύθυνη για την διαχείριση των εκάστοτε συνοδών.

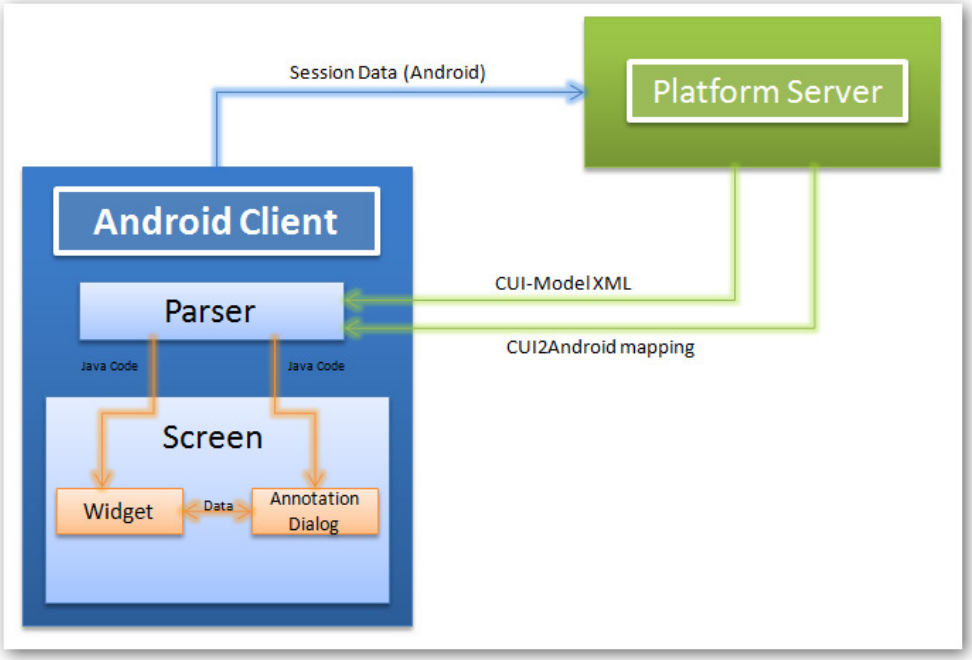
Από πλευράς client δημιουργήθηκαν εκείνες οι δομές ώστε να δύναται ο εκάστοτε χρήστης/client να αλληλεπιδράσει με τις server-side υπηρεσίες. Κύριο τμήμα των τελευταίων αποτελεί η δυνατότητα της client-side αρχιτεκτονικής να διερμηνεύσει αφηρημένες (υπό την έννοια της ικανότητας περιγραφής διεπαφών ανεξαρτήτων πλατφόρμας) περιγραφές και να τις αντιστοιχήσει σε platform-specific διαδραστικά στοιχεία. Κύριο τμήμα και πυρήνας του τελευταίου είναι ο 'Android Platform Server' που περιγράφεται στην επόμενη ενότητα.

Τέλος μελετήσαμε τα παραπάνω υπό το πρίσμα ενός εξειδικευμένου σεναρίου για τις ανάγκες του οποίου χρειάστηκε να δημιουργήσουμε δύο μη γηγενώς παρεχόμενα από το android UI toolkit σύνθετα διαδραστικά αντικείμενα.

5.1 Αρχιτεκτονική

Όπως φαίνεται με τη βοήθεια του παρακάτω σχήματος, η φορητή συσκευή που τρέχει την Android συμβατή (android-compatible) εφαρμογή που αναπτύχθηκε (Android Client), κάνει μια κλήση στον Platform Server, ο όποιος αποτελεί ένα ενσωματωμένο τμήμα λογισμικού της client-side πλατφόρμας, ζητώντας στοιχεία που αφορούν τις διαθέσιμες - ενεργές συνεργατικές συνόδους. Ο Platform Server με τη σειρά του εκτελεί κλήση στην αντίστοιχη υπηρεσία του server-side

επιστρέφοντας τις ανάλογες πληροφορίες βάσει των οποίων ο εκάστοτε χρήστης επιλέγει τη συνεργατική συνεδρία στην οποία επιθυμεί να συμμετέχει. Ως αποτέλεσμα της τελευταίας ενέργειας, δηλαδή της επιλογής συμμετοχής σε μια συγκεκριμένη συνεργατική σύνοδο, ο server απαντάει αποστέλλοντας τόσο την αφηρημένη περιγραφή της διεπαφής όσο και μια σειρά από βοηθητικά μοντέλα ενσωματωμένα βέβαια στα πλαίσια μιας ενιαίας περιγραφής (ui-model). Ο Platform Server από την πλευρά του διατρέχει την αφηρημένη περιγραφή της διεπαφής (cui-model), ενώ ακολουθώντας πλήθος παράλληλων διεργασιών συνθέτει το τελικό platform - executable UI το οποίο και ενσωματώνει στον ανάλογο window container. Αξίζει να σημειωθεί ότι τμήμα της λειτουργικότητας που μόλις περιγράφηκε ενσωματώνει τη δυνατότητα κατανεμημένης μεταφόρτωσης μη εγγενώς υποστηριζόμενων κλάσεων στην περίπτωση χρησιμοποίησης εξειδικευμένων διαδραστικών αντικειμένων.



Εικόνα 32 : Αρχιτεκτονική Client-side Δομής

Παρακάτω δίνουμε στιγμιότυπα (instances) των μοντέλων (models) που χρησιμοποιούνται κατά την εκτέλεση των παραπάνω διαδικασιών.

Συγκεκριμένα, στην παρακάτω εικόνα δίνεται μια ενδεικτική μορφή ενός αρχείου τύπου CUI που περιγράφει με ανεξάρτητου πλατφόρμας τρόπο μια υποθετική διεπαφή. Αξίζει να σημειωθεί ότι για την υποστήριξη μη γηγενώς υποστηριζόμενων αντικειμένων έχουν χρησιμοποιηθεί επιπλέον των ήδη υποστηριζόμενων από το αντίστοιχο μοντέλο της UsiXML ετικέτες (xml-tags), όπως π.χ. το ‘advancedActivitiesDisplay’το οποίο αντιστοιχεί σε ένα εξειδικευμένο γραφικό αντικείμενο που δημιουργήσαμε για τις ανάγκες της τρέχουσας εφαρμογής.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<uiModel id="Test1" name="Test1"
    creationDate="2010-05-15T19:25:34.562+03:00" schemaVersion="1.6.4" xmlns="http://www.usixml.org">
    <head>
        <version modifDate="2010-05-15T19:25:34.562+03:00"/>
        <authorName>Miltos</authorName>
        <comment>Test</comment>
    </head>
    <cuiModel name="Test1-cui" target="web">
        <version modifDate="2010-10-10T17:09:17.402+01:00" xmlns="">7</version>
        <authorName xmlns="">Miltos</authorName>
        <window height="500" width="600" name="Formulaire(2/5)" id="window_1">
            <box type="vertical" relativeHeight="100" name="box1_0" id="2346757">
                <box type="horizontal" name="boxTodo" id="boxTodo">
                    <!-- Text Component-->
                    <outputText defaultContent="Sexe" isBold="true" id="2"/>
                    <comboBox id="434" name="label_3" isDropDown="true">
                        <item id="radiobutton_0" name="radiobutton_0" defaultContent="Femme"></item>
                        <item id="radiobutton_1" name="radiobutton_1" defaultContent="Homme"></item>
                    </comboBox>
                </box>
                <box id="box_123" type="vertical">
                    <button id="0" name="button_0" defaultContent="click me"/>
                    <!-- input-->
                    <inputText id="34" width="100"/>
                </box>
                <box id="box_1234" type="vertical">
                    <advancedActivitiesDisplay id="activities_display_0">
                        <advancedActivityButton id="activity_button_0"></advancedActivityButton>
                        <advancedActivityButton id="activity_button_1"></advancedActivityButton>
                    </advancedActivitiesDisplay>
                </box>
            </box>
        </window>
    </cuiModel>
    <mappingModel id="Test1-mappingModel_0" name="Test1-mappingModel">
        <adheresToModel id="IWF1">
            <source sourceId="activities_display_0"/>
            <target targetId="activities_display_data_0"/>
        </adheresToModel>
    </mappingModel>
    <widgetResourceModel id="wrm_0">
        <widgetResource id="activities_display_data_0">
            <contentItem id="activities_display_data_0" type="advancedActivitiesDisplay" name="activities_display_data" defaultContent="">
                <property id="property_0" type="boolean" name="editable" defaultContent="true"/>
                <property id="property_1" type="string" name="defaultView" defaultContent="'agendaWeek'"/>
                <property id="property_2" type="boolean" name="header" defaultContent="false"/>
                <property id="property_3" type="boolean" name="allDaySlot" defaultContent="false"/>
                <property id="property_4" type="date" name="start" defaultContent="2011-02-02T00:00:00.000+03:00"/>
                <property id="property_5" type="number" name="duration" defaultContent="6"/>
                <property id="property_5" type="number" name="width" defaultContent="200"/>
            </contentItem>
        </widgetResource>
    </widgetResourceModel>
</uiModel>
```

Εικόνα 33 : CUI-Model XML

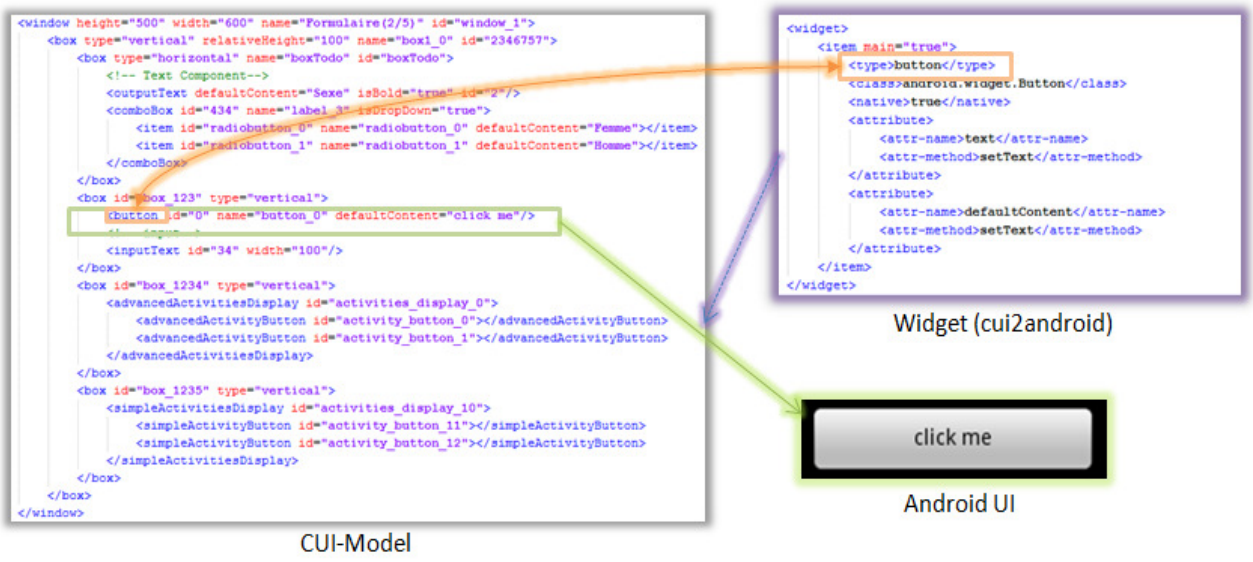
Αξίζει επίσης να σημειωθεί στην παραπάνω εικόνα το ‘mappingModel’ όπως και το ‘widgetresourceModel’, τα οποία αντιστοιχούν μη γηγενώς υποστηριζόμενα αντικείμενα με τρέχουσες τιμές του εκάστοτε instance.

Στην παρακάτω εικόνα παρατηρούμε πως υλοποιούνται οι αντιστοιχίσεις αφηρημένων περιγραφών (μοντέλο CUI2Android), γηγενών και μη υποστηριζόμενων διαδραστικών αντικειμένων, με platform-specific αντικείμενα. Στα πλαίσια αυτά αναφέρουμε ότι ανάλογα με την πλατφόρμα στόχο (target-platform) οι περιγραφές αυτές είναι προφανές ότι θα αντιστοίχιζαν πάλι τις ίδιες αφηρημένες περιγραφές των εκάστοτε ετικετών με διαφορετικά ανά περίπτωση διαδραστικά αντικείμενα στόχους (target - elements) ώστε να καταστεί δυνατή η ανεξαρτήτου πλατφόρμας περιγραφή και εκτέλεση των διεπαφών του CUI.

```
<widget>
    <item main="true">
        <type>advancedActivitiesDisplay</type>
        <class>gr.istl.aaw.AdvanceActivityWidget</class>
        <native urls="syncui_context/libs-android/AdvanceActivityWidget.apk">false</native>
        <container>true</container>
        <cont-method>addActivityView</cont-method>
        <property>
            <prop-name>duration</prop-name>
            <prop-method>setDuration</prop-method>
        </property>
         <property>
            <prop-name>start</prop-name>
            <prop-method>setStartDate</prop-method>
        </property>
    </item>
</widget>
<widget>
    <item main="true">
        <type>box</type>
        <class>android.widget.LinearLayout</class>
        <native>true</native>
        <container>true</container>
        <cont-method>addView</cont-method>
        <attribute>
            <attr-name>id</attr-name>
            <method-name>setId</method-name>
        </attribute>
        <attribute>
            <attr-name>type</attr-name>
            <attr-method>setOrientation</attr-method>
        </attribute>
    </item>
</widget>
<widget>
    <item main="true">
        <type>button</type>
        <class>android.widget.Button</class>
        <native>true</native>
        <attribute>
            <attr-name>text</attr-name>
            <attr-method>setText</attr-method>
        </attribute>
        <attribute>
            <attr-name>defaultContent</attr-name>
            <attr-method>setText</attr-method>
        </attribute>
    </item>
</widget>
```

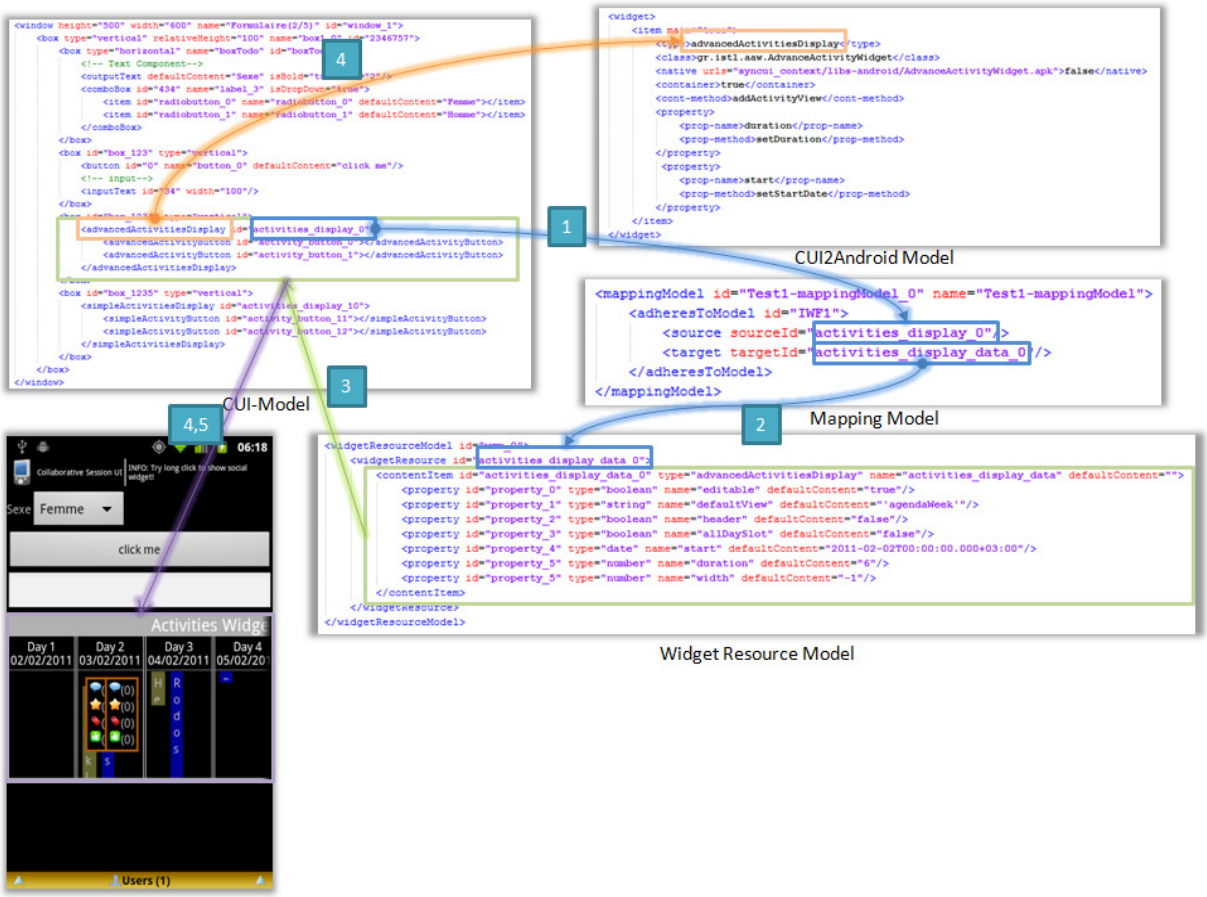
Εικόνα 34 : Παράδειγμα τμήματος στιγμιοτύπου του 'CUI2Android' XML

Ο τύπος μοντέλου αυτός (CUI2Android) είναι νέος και δεν υποστηρίζεται από τη UsiXML, η οποία στα πλαίσια του συγκεκριμένου και προαποφασιμένου αριθμού διαδραστικών αντικειμένων που υποστηρίζει ενσωματώνει τα παραπάνω mappings σε επίπεδο (hard-coded) κώδικα. Σε μια προσπάθεια να δούμε πιο αναλυτικά την παραπάνω διαδικασία, στα πλαίσια ενός γηγενούς υποστηριζόμενου από την πλατφόρμα στόχο αντικειμένου (κουμπί), παραθέτουμε το παρακάτω σχήμα.



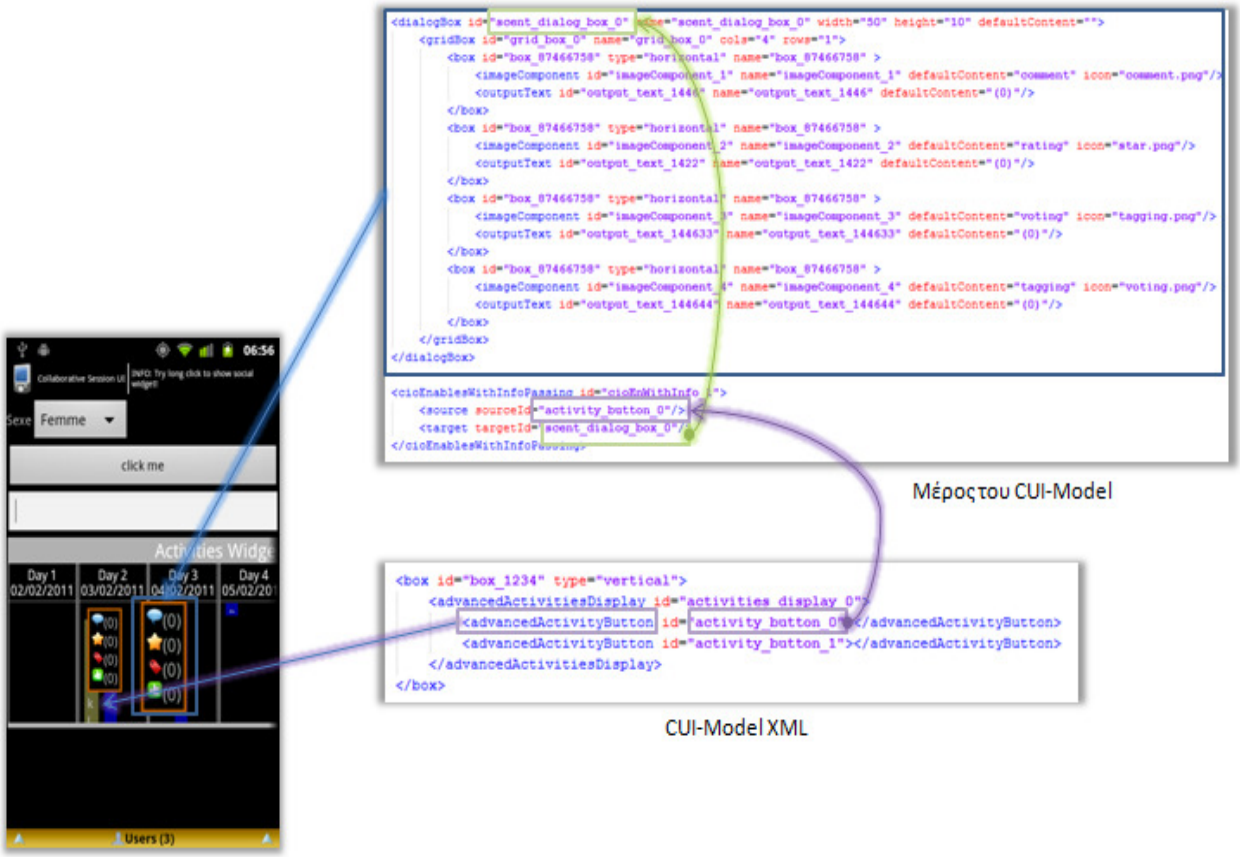
Εικόνα 35 : Διαδικασία δημιουργίας γηγενών αντικειμένων

Όταν κάποιο διαδραστικό αντικείμενο δεν περιέχετε στο mapping model τότε πρόκειται για την περίπτωση των μη-γηγενών διαδραστικών αντικειμένων. Ο parser (που απεικονίζεται στην Εικόνα 32) με της κατάλληλες ενέργειες (αντιστοιχίσεις μέσα στο ίδιο το μοντέλο) βρίσκει και δημιουργεί το διαδραστικό αντικείμενο το οποίο περιγράφεται σε ένα κομμάτι του CUI-Model XML.



Εικόνα 36 : Διαδικασία δημιουργίας μη-γηγενών διαδραστικών αντικειμένων

Μέσω των εικόνων 36 και 37, φαίνεται ο τρόπος με των οποίο δημιουργούνται τα μη-γηγενή διαδραστικά αντικείμενα με στόχο τη δημιουργία της παρακάτω διεπαφής και της απόδοσης κοινωνικών χαρακτηριστικών σε αυτή.



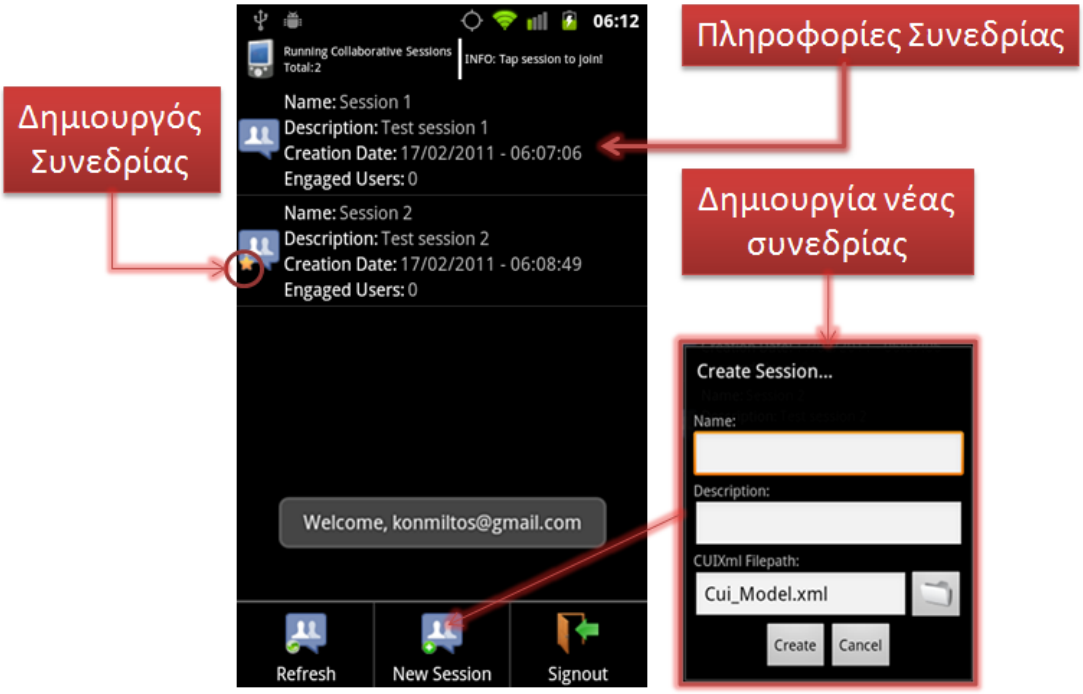
Εικόνα 37 : Συσχέτιση κοινωνικών χαρακτηριστικών με το διαδραστικό αντικείμενο

Ο συγχρονισμός μεταξύ των κατανεμημένων αντιγράφων γίνεται μέσω ενός τύπου μοντέλου που ονομάζεται abstraction model (που περιλαμβάνει ορισμούς ‘αφαιρετικών’ κλάσεων), βασιζόμενοι στο οποίο οι κατάλληλοι μηχανισμοί δημιουργούν τις κατάλληλες κλάσεις καθώς και στιγμιότυπα αυτών κοινά ανεξαρτήτως γραφικής μεταφοράς εξασφαλίζοντας χαλαρή ζεύξη ονομαζόμενο και ως relaxed-WYSINWIS (What You See Is Not What I See). Η σύνδεση των στιγμιότυπων των αφαιρετικών κλάσεων με τη γραφική διεπαφή επιτυγχάνεται μέσω ενός μοντέλου το οποίο ονομάζεται ‘consistency model’ που περιλαμβάνεται στο ενιαίο XML specification που λαμβάνει ο platform server κάθε client.

5.2 Παρουσίαση Εφαρμογής

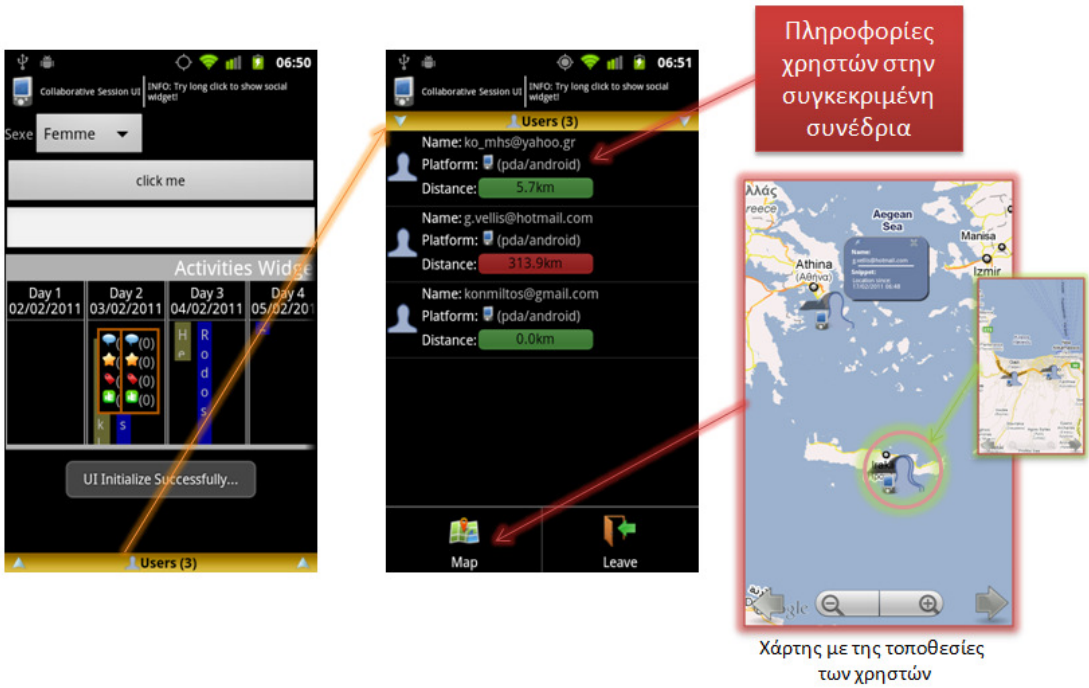
Στην τρέχουσα ενότητα αναφέρονται παραθέτοντας γραφικές απεικονίσεις των βημάτων που ακολουθούνται ή μπορεί να ακολουθηθούν στα πλαίσια της εφαρμογής που αναπτύχθηκε ώστε να μπορεί ένας χρήστης να δει τις διαθέσιμες συνόδου, να δημιουργήσει μια νέα, να συμμετέχει σε μια, κοκ.

Αρχικά, αφού γίνεται η διαδικασία της πιστοποίησης του χρήστη (Authentication) εμφανίζονται σε μορφή λίστας όλες οι διαθέσιμες ενεργές συνεδρίες καθώς και πληροφορίες που αφορούν την κάθε μια ξεχωριστά. Τέτοιες πληροφορίες αφορούν το όνομα, την περιγραφή και το πότε δημιουργήθηκε κάθε συνεδρία καθώς επίσης και τον αριθμό των χρηστών που ήδη αλληλεπιδρούν εκείνη τη χρονική στιγμή στα πλαίσια μιας συγκεκριμένης συνεδρίας. Σε περίπτωση που επιθυμούμε να δημιουργήσουμε μια νέα σύνοδο πρέπει να εισάγουμε τα στοιχεία αυτής όπως περιγράφονται γραφικά μέσω της δεξιάς πλευράς της παραπάνω εικόνας. Επίσης με τη βοήθεια κατάλληλου εικονιδίου μπορούμε να ξεχωρίσουμε τις συνεδρίες, τις οποίες έχουμε δημιουργήσει σε σχέση με αυτές που έχουν δημιουργηθεί από άλλους χρήστες.



Εικόνα 38: Διαθέσιμες συνεδρίες

Στην συνέχεια αφού επιλέξουμε τη συνεδρία στην οποία θέλουμε να εισέλθουμε, κάνοντας 'κλικ' (touch event) πάνω σε αυτή, μεταβαίνουμε στο κυρίως μέρος του εργαλείου βλέποντας την εκάστοτε γραφική διεπαφή που έχει οριστεί – συσχετιστεί με τη συγκεκριμένη συνεδρία. Παράλληλα στα πλαίσια του συγκεκριμένου σημείου δίνεται η επιλογή εμφάνισης στοιχείων που αφορούν τους χρήστες που αλληλεπιδρούν την εκάστοτε χρονική στιγμή, όπως το πόσοι και ποιοι είναι καθώς και τη σχετική απόσταση του καθενός αναφορικά με εμάς (καθώς το ακριβές στίγμα τους με τη βοήθεια χάρτη) σε συνδυασμό με το είδος της εκάστοτε πλατφόρμας στα πλαίσια της οποίας έχουν συνδεθεί και αλληλεπιδρούν. Ας δούμε την παρακάτω εικόνα προκειμένου να κατανοήσουμε καλυτέρα τη ακριβώς προσφέρεται στον χρήστη από την χρήση του συγκεκριμένου εργαλείου.



Εικόνα 39 : Στοιχειά διεπαφής σε μια συνεδρία

Στην παραπάνω εικόνα φαίνονται δύο στιγμιότυπα της εφαρμογής στα πλαίσια των σεναρίων χρήσης που περιγράφησαν. Αξίζει να σημειωθεί επίσης η χρήση δυο διαλόγων ενδεδειγμένων για τη μεταφορά κοινωνικά ευαίσθητων πληροφοριών. Συγκεκριμένα υποστηρίχθηκαν τέσσερις βασικές υπηρεσίες στις οποίες συμπεριλαμβάνονται, ο σχολιασμός (Commenting), η βαθμολόγηση (Rating), η απόδοση ετικετών (Tagging) καθώς και η διεξαγωγή ψηφοφορίας (Voting). Πρόσβαση σε εξειδικευμένους διαλόγους διαχείρισης των υπηρεσιών δίνεται κάνοντας 'κλικ' πάνω σε ένα αντικείμενο μεταβαίνοντας με αυτό τον τρόπο στον ενιαίο διάλογο (Annotation dialog) διαχείρισης, όπως ενδεικτικά φαίνεται μέσω της παρακάτω εικόνας.

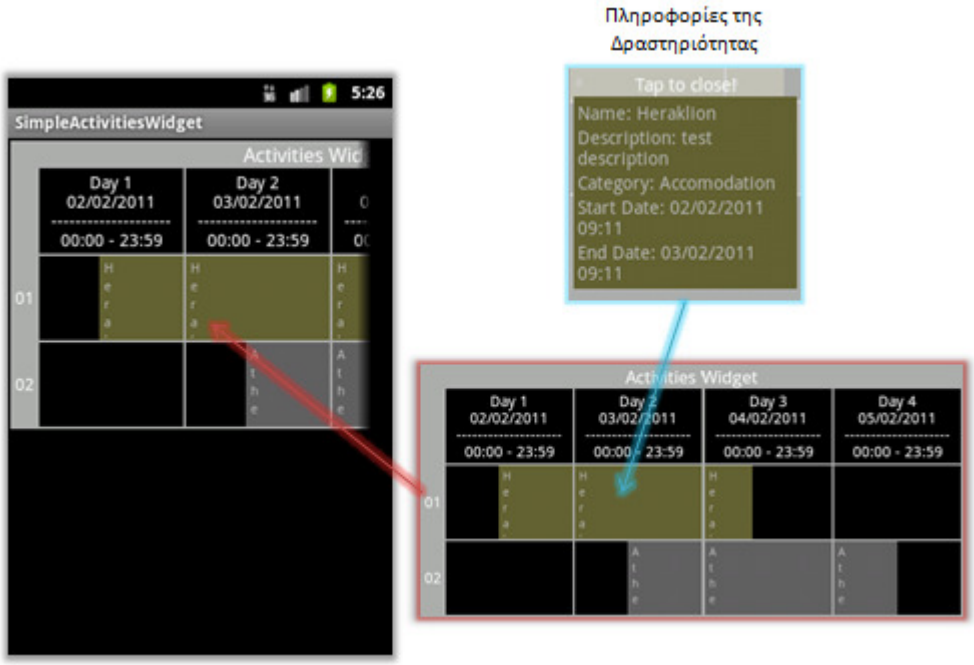
Εικόνα 40 : Εμφάνιση κοινωνικών χαρακτηριστικών σε ένα διαδραστικό αντικείμενο

6. Σύνοψη - Αποτελέσματα

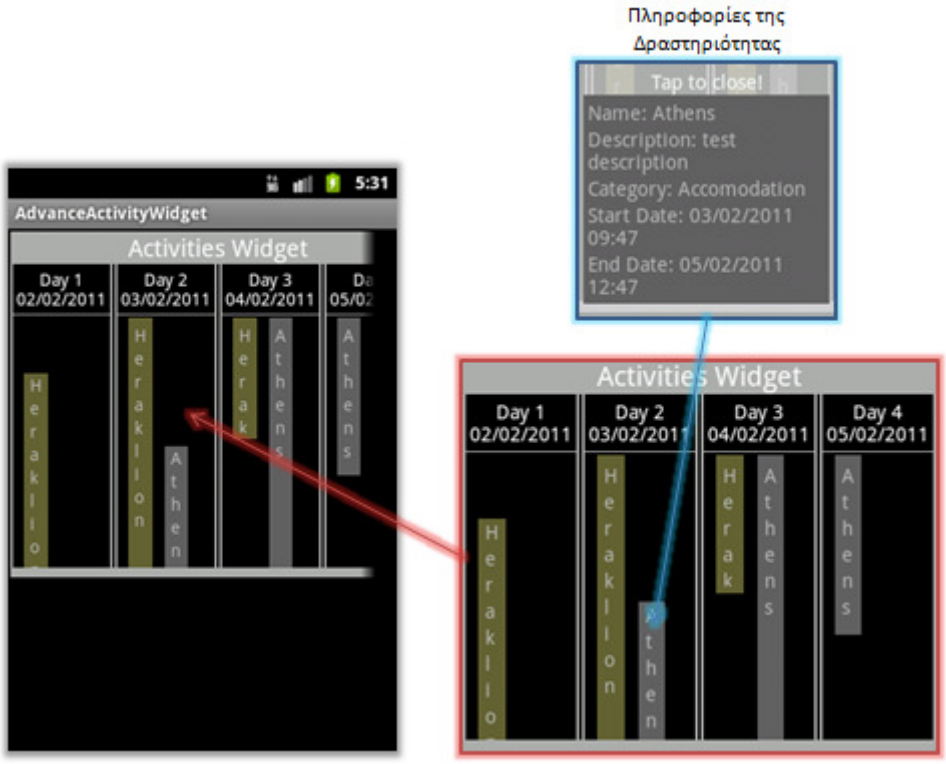
Η πτυχιακή αυτή μελέτησε εξειδικευμένες πτυχές του γνωστικού αντικειμένου της συνεργασίας υποστηριζόμενης από υπολογιστή στα πλαίσια του οποίου ανέπτυξε μεθόδους υποστήριξης συνεργασίας μέσω έξυπνων κινητών τηλεφώνων. Συγκεκριμένα αναπτύχθηκαν τόσο από πλευράς client όσο και server δομές που να μπορούν να υποστηρίξουν εξειδικευμένες έννοιες όπως της διαχείρισης συνόδων. Επίσης υπό τις αναδυόμενες απαιτήσεις που εγείρονται στα πλαίσια της υποστήριξης των χρηστών όσον αφορά τη δυνατότητα πρόσβασης σε υπηρεσίες ανεξαρτήτως περιβάλλοντος χρήσης εξετάσαμε πως θα μπορούσε να υποστηριχθεί, διευκολυνθεί και αυτοματοποιηθεί η ανάπτυξη συνεργατικού λογισμικού με τη χρήση μοντέλων (Model-based UI Engineering). Στα πλαίσια αυτά χρησιμοποιήσαμε μια από τις πιο δημοφιλείς βασιζόμενες σε μοντέλα γλώσσες περιγραφής διεπαφών, τη UsiXML και δείξαμε πως θα μπορούσε να υποστηρίξει την ανάπτυξη πολυχρηστικών διεπαφών.

Επίσης αναδείξαμε τη δυνατότητα περιγραφής πιο σύνθετων διαδραστικών αντικειμένων από αυτά που υποστηρίζονται σήμερα από τις περισσότερες βασιζόμενες σε μοντέλα γλώσσες και επιλέξαμε ένα σενάριο χρήσης που αφορά την ενσωμάτωση μη γηγενών διαδραστικών αντικειμένων τα οποία χρειάστηκε να δημιουργήσουμε.

7. Παράρτημα



Εικόνα 41 : Simple Activity Widget



Εικόνα 42 : Advanced Activity Widget

8. Βιβλιογραφία

1) Wikipedia: http://wikipedia.org

2) UsiXML: http://www.usixml.org

3) Pierre Dillenbourg, (2002), The impact of Awareness Tools on Mutual Modelling in a Collaborative Game, University of Geneva.

4) Carl Gutwin and Saul Greenberg. 2002. A Descriptive Framework of Workspace Awareness for Real-Time Groupware. Computer Supported Cooperative Work 11, 3 (November 2002), 411-446.

5) Near Filed Communication (NFC): http://en.wikipedia.org/wiki/Near_field_communication

6) Cameleon Reference Framework (CRF): http://www.w3.org/2005/Incubator/model-based-ui/wiki/Cameleon_reference_framework