Τεχνολογικό εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών

Τμήμα Μηχανικών Πληροφορικής

Πτυχιακή Εργασία

Τίτλος: 'Σύγχρονη συνεργασία υποστηριζόμενη από κινητές συσκευές: Η περίπτωση χρήσης της συνεργατικής διαμόρφωσης πλάνου για την οργάνωση της παραγωγής βιολογικών προϊόντων'.

Κωνσταντάκης Εμμανουήλ (Α.Μ. : 1925)

Εισηγητής: Νικόλαος Βιδάκης

ΗΡΑΚΛΕΙΟ

2016

Ευχαριστίες

Ένα μεγάλο ευχαριστώ στον εισηγητή μου κ. Νικόλαο Βιδάκη, όπως επίσης και στον καθηγητή μου κ. Γιώργο Βελλή, για την υπομονή του όλο αυτό το καιρό αλλά και τη βοήθεια και σωστή καθοδήγηση που έλαβα. Τέλος ευχαριστώ την οικογένεια μου που ήταν δίπλα μου όλα αυτά τα χρόνια και με την πολύτιμη βοήθεια τους κατάφερα να ολοκληρώσω τις σπουδές μου.

Πίνακας περιεχομένων

1. Εισαγωγή

Τα τελευταία χρόνια η απομακρυσμένη συνεργασία μεταξύ δυο χρηστών γίνεται ολοένα πιο εύκολα και πιο αποτελεσματικά. Ο λόγος είναι τα συστήματα και συσκευές τρίτης γενιάς (tablets,smartphones κ.ά.) που έχουν αντικαταστήσει τον ηλεκτρονικό υπολογιστή. Βασικός στόχος είναι η υποστήριξη των χρηστών να γίνεται όσο το δυνατών πιο αποτελεσματικά.
Ένα τμήμα του κλάδου των συστημάτων υποστήριξης συνεργασίας ασχολείται με συστήματα για την υποστήριξη της εργασίας ομάδων χρηστών που ονομάζονται Συστήματα Υποστήριξης Συνεργατικής Εργασίας με Υπολογιστή (Computer Supported Cooperative Work, CSCW). Τα συστήματα αυτά παρέχουν εργαλεία για την επικοινωνία(γραπτή ή οπτική), εργαλεία πρόσβασης και ανταλλαγής αρχείων και πληροφοριών καθώς και εργαλεία για την από κοινού χρήση χώρων εργασίας και υλοποίηση δραστηριοτήτων. Γι' αυτό το λόγο γίνεται χρήση πολλαπλών μορφών αναπαράστασης και πολλών διαύλων επικοινωνίας, ώστε να παρέχεται αλληλεπίδραση και να υπάρχει η επικοινωνία.**[2]**

2. Συνεργασία υποστηριζόμενη από υπολογιστή

Η υπολογιστική υποστήριξη της ομαδικής εργασίας είναι γνωστή με τους όρους Groupware ή Computer Supported Cooperative Work (CSCW). Από τη μια πλευρά, πολλοί συγγραφείς θεωρούν ότι τα CSCW και groupware είναι συνώνυμα. Από την άλλη πλευρά, διαφορετικοί συγγραφείς υποστηρίζουν ότι ενώ το groupware αναφέρεται σε πραγματικά συστήματα που βασίζονται στη πληροφορική, ενώ το CSCW εστιάζει τη μελέτη των εργαλείων και τεχνικών του groupware.

Το CSCW εξετάζει πως οι άνθρωποι μπορούν να επωφεληθούν και να χρησιμοποιούν τις υπολογιστικές τεχνολογίες. Σκοπός είναι να τους εξυπηρετεί όσο το δυνατό περισσότερο στις συνεργασίες τους με άλλους ανθρώπους, κάτι με το οποίο δεν μπορεί να γίνει με τις περισσότερες ήδη υπάρχουσες εφαρμογές.

Κεντρικά θέματα που εξετάζει το CSCW είναι η αντίληψη ομάδας (groupawareness), τα πολυχρηστικά περιβάλλοντα αλληλεπίδρασης (multi-userinterfaces), ο έλεγχος ταυτόχρονης προσπέλασης (concurrency control), η επικοινωνία και συνεργασία μέσα στην ομάδα, ο διαμοιραζόμενος χώρος πληροφοριών και η υποστήριξη ενός ετερογενούς, ανοιχτού περιβάλλοντος, που ενσωματώνει όλες τις εφαρμογές απλού χρήστη.**[11]**

2.1. Διαχείριση Συνόδων (Session Management)

Η διαχείριση συνόδων είναι υπεύθυνη για τον έλεγχο των χρηστών που θα συμμετέχουν σε μια επικοινωνία αλλά και για την ομαδοποίηση των στιγμιότυπων που θα μοιράζονται. Στη συνέχεια αναφέρονται κάποιες προτεινόμενες πολιτικές για τον έλεγχο συνεδριάσεων:

- Καθορισμός των ορίων σχετικά με το ποιος μπορεί να λάβει μέρος στη συνεδρίαση. Χρειάζεται να υπάρχουν περιορισμοί στο συνολικό πλήθος των μελών της ομάδας καθώς και στο είδος των ατόμων που επιτρέπεται να αποτελούν μέλη της ομάδας ανάλογα με το αν διαθέτουν κάποια βασικά προσόντα.

- Καθορισμός του κατά πόσο επιτρέπεται σε κάποιο μέλος της ομάδας να συμμετέχει και να εγκαταλείπει οποτεδήποτε τη συνεδρία (καθορισμός το κατάλληλου πρωτοκόλλου). Στις περισσότερες εφαρμογές επιτρέπεται στους χρήστες να λαμβάνουν μέρος στις συζητήσεις ή να τις εγκαταλείπουν, χωρίς την ύπαρξη δεσμεύσεων.
- Προστασία από εισβολή ανεπιθύμητων χρηστών στον προσωπικό χώρο. Η τηλεφωνική κλήση είναι ένα παράδειγμα ενοχλητικού μηχανισμού ελέγχου συνεδρίασης καθώς, αφού χωρίς την ύπαρξη αυτόματου τηλεφωνητή ή αναγνώριση κλήσης, δεν μπορεί κάποιος να καθορίσει αν το τηλεφώνημα είναι επιθυμητό ή όχι.
- Παροχή μηχανισμού πρόληψης διακοπών.**[4]**

2.2. Έλεγχος Συγχρονισμού (Concurrency Control)

Η λειτουργία της αναίρεσης ενεργειών σε περιβάλλον πολλών χρηστών είναι ένα δύσκολο πρόβλημα. Όταν οι χρήστες αλληλεπιδρούν στέλνοντας μηνύματα ο ένας στον άλλο ή διακινώντας δεδομένα ίσως η συνέπεια κάποιου χρήστη να παραβιάζεται από κάποιον άλλο. Ο έλεγχος ταυτοχρονισμού παρέχει κανόνες , μεθόδους και σχεδιαστικές μεθοδολογίες οι οποίες διατηρούν την συνέπεια των χρηστών οι οποίοι λειτουργούν ταυτόχρονα όταν αλληλεπιδρούν και επομένως διατηρεί την συνέπεια ολόκληρου του συστήματος. Οι κανόνες της συνέπειας και της ορθότητας είναι ανάλογα με την αποδοτικότητα του συστήματος.

2.3. Έλεγχος Δαπέδου (Floor Control ή Access Control)

Από τη στιγμή που οι χρήστες είναι σε μία συνδιάσκεψη, πρέπει να καθορίζεται το είδος της πρόσβασης που θα έχει ο κάθε ένας στα δεδομένα και στα διάφορα εργαλεία. Μπορεί να υπάρχει ταυτόχρονη πρόσβαση, μπορεί μόνο ένας να έχει πρόσβαση κάθε φορά ή να υπάρχει μεσολαβητής που να ελέγχει την πρόσβαση, ή να υπάρχει χρονικό όριο για κάθε άτομο.

Συχνά η ταυτόχρονη πρόσβαση από οποιονδήποτε-σε-οτιδήποτε μπορεί να δημιουργήσει προβλήματα στην περίπτωση όπου κάποιος χρήστης δεν συνεργάζεται, όμως προτιμάται διότι βοηθάει τη ροή της εργασίας. Η εφαρμογή ενός μοντέλου ενδιάμεσης πρόσβασης έχει το πλεονέκτημα της πρόληψης των λαθών, της αποφυγής αντικρουόμενων αλλαγών στα δεδομένα από τους χρήστες αλλά και της πρόληψης μη εξουσιοδοτημένης πρόσβασης.

Όποτε χρησιμοποιείται τηλεσυνεργασία, ορισμένη πληροφορία χρειάζεται να είναι διαμοιραζόμενη και ορατή, ενώ πρέπει να διασφαλίζεται ότι η υπόλοιπη πληροφορία πρέπει να παραμένει ιδιωτική (private), και ότι η ιδιαίτερα σημαντική πληροφορία προστατεύεται από επιθέσεις ατόμων που θα προσπαθούν να τη μάθουν. Σε πολλές περιπτώσεις, οι χρήστες επιλέγουν να χρησιμοποιούν ψευδώνυμο ή να είναι ανώνυμοι

Έλεγχος και αμοιβαιότητα: Για την επίλυση αυτών των αντικρουόμενων αναγκών, είναι σημαντικό οι χρήστες να έχουν όσο το δυνατόν περισσότερο έλεγχο πάνω σε τι είδος πληροφορία διαμοιράζεται και σε τι παραμένει ιδιωτική. Πρέπει οι χρήστες να αποφασίζουν το πόση πληροφορία θα διαμοιραστούν, και με βάση αυτό να καθορίζουν σε ποιο είδος

πληροφορίας θα έχουν πρόσβαση. Ένα παράδειγμα ιδιωτικής πολιτικής είναι η αρχή της αμοιβαιότητας (reciprocity): αν ένας χρήστης θέλει πληροφορία για κάποιον άλλο χρήστη, τότε πρέπει να παράσχει την αντίστοιχη πληροφορία για τον εαυτό του.**[4]**

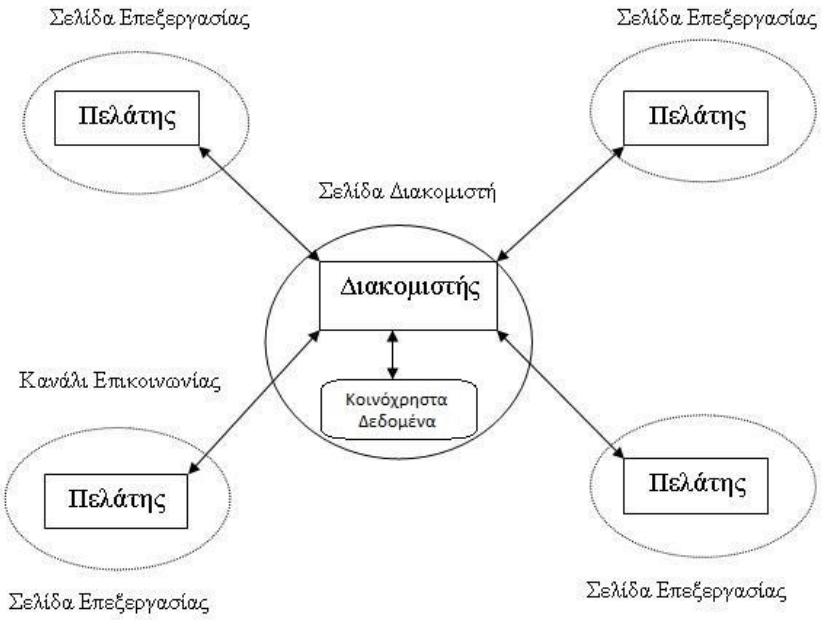
2.4. Συνεργατικές Αρχιτεκτονικές (Collaborative Architectures)

Το κεντρικό ζήτημα που αφορά την αρχιτεκτονική των συστημάτων με διαμοιραζόμενα δεδομένα είναι το πού αποθηκεύονται τα κοινόχρηστα δεδομένα και που διατηρούνται. Η αποθήκευση κοινόχρηστων δεδομένων μπορεί να γίνεται με κεντρικοποιημένη αρχιτεκτονική (centralized), αρχιτεκτονική αντιγραφών(replicated) ή ένα υβρίδιο των δύο προσεγγίσεων.

2.4.1. Κεντρικοποιημένη Αρχιτεκτονική (Centralized architecture)

Με αυτή την αρχιτεκτονική, υπάρχει μια διάκριση ανάμεσα στην τοποθεσία του εξυπηρετητή (server) και στις τοποθεσίες επεξεργασίας. Μπορεί να υπάρχουν πολλαπλές τοποθεσίες επεξεργασίας, καθεμιά «τρέχοντας» μια «client» διεργασία η οποία είναι υπεύθυνη για τη δημιουργία λειτουργιών από τα δεδομένα εισαγωγής των χρηστών που θα ενημερώνουν τα κοινόχρηστα δεδομένα και θα εμφανίζουν το περιεχόμενο του κοινόχρηστου εγγράφου στο τοπικό περιβάλλον εργασίας του χρήστη.

Υπάρχει μόνο μία τοποθεσία για το διακομιστή όπου αποθηκεύονται τα κοινόχρηστα δεδομένα. Η λειτουργία του διακομιστή είναι υπεύθυνη για την ενημέρωση των κοινόχρηστων δεδομένων και την αποστολή του περιεχομένου του περιβάλλοντος εργασίας του χρήστη στον «client» για εμφάνιση. Δεδομένου ότι μόνο με τη λειτουργία του διακομιστή μπορεί να υπάρχει πρόσβαση στα κοινόχρηστα δεδομένα, δεν γίνεται να έχουμε ταυτόχρονη πρόσβαση στα κοινόχρηστα δεδομένα και ως εκ τούτου δεν υπάρχει πρόβλημα ασυνέπειας. Ωστόσο, το να μην υπάρχει ταυτόχρονη πρόσβαση είναι επίσης ένα μειονέκτημα επειδή περιορίζει την απόδοση του συστήματος δεδομένου ότι εξ ορισμού, μόνο ένας χρήστης μπορεί να έχει πρόσβαση στα δεδομένα κάθε φορά. Εάν ένας χρήστης πραγματοποιεί μια μεγάλης διάρκειας λειτουργία (π.χ. μετακίνηση ενός αντικειμένου), κανένας άλλος χρήστης δεν έχει πρόσβαση στα κοινόχρηστα δεδομένα. Τα συστήματα χρησιμοποιούν αυτή την αρχιτεκτονική για να παρέχουν την ψευδαίσθηση της ταυτόχρονης πρόσβασης υποστηρίζοντας μόνο λειτουργίες που μπορούν να εκτελεστούν σε πολύ μικρή διάρκεια (π.χ. το χρώμα ενός pixel σε bitmap έγγραφο).



Εικόνα 1 : Κεντρικοποιημένη Αρχιτεκτονική

Το κύριο μειονέκτημα για την κεντρικοποιημένη αρχιτεκτονική είναι ότι ο τοπικός χρόνος απόκρισης μπορεί να είναι μεγάλος. Αυτό οφείλεται στο γεγονός ότι όταν μια λειτουργία παράγεται, η εκτέλεσή της εμφανίζεται στο τοπικό περιβάλλον εργασίας του χρήστη μόνο μετά από τα παρακάτω βήματα:

- Η λειτουργία αποστέλλεται στο διακομιστή
- Ο διακομιστής εκτελεί τη λειτουργία, έπειτα στέλνει ένα μήνυμα (περιέχει πληροφορίες για το περιβάλλον εργασίας του χρήστη) για να ενημερώσει όλους τους «clients» για το update, και
- Ο «client» δέχεται το μήνυμα απάντησης από το διακομιστή και ενημερώνει το τοπικό περιβάλλον εργασίας του χρήστη.

Η ταχύτητα ολοκλήρωσης αυτών των τριών βημάτων είναι ιδιαίτερα εξαρτημένη από το διάστημα του χρόνου καθυστέρησης του δικτύου. Σε ένα δίκτυο όπου η καθυστέρηση είναι μεγάλη ο χρόνος απόκρισης είναι μεγάλος. Δεδομένου ότι τα «collaborative editing systems» με κεντρικοποιημένη αρχιτεκτονική δεν μπορούν να εγγυηθούν γρήγορο χρόνο απόκρισης, αυτή η αρχιτεκτονική δεν ικανοποιεί τις απαιτήσεις των συνεργατικών συστημάτων επεξεργασίας γραφικών. Επισυνάπτοντας πλεονεκτήματα στης κεντρικοποιημένης αρχιτεκτονικής είναι :

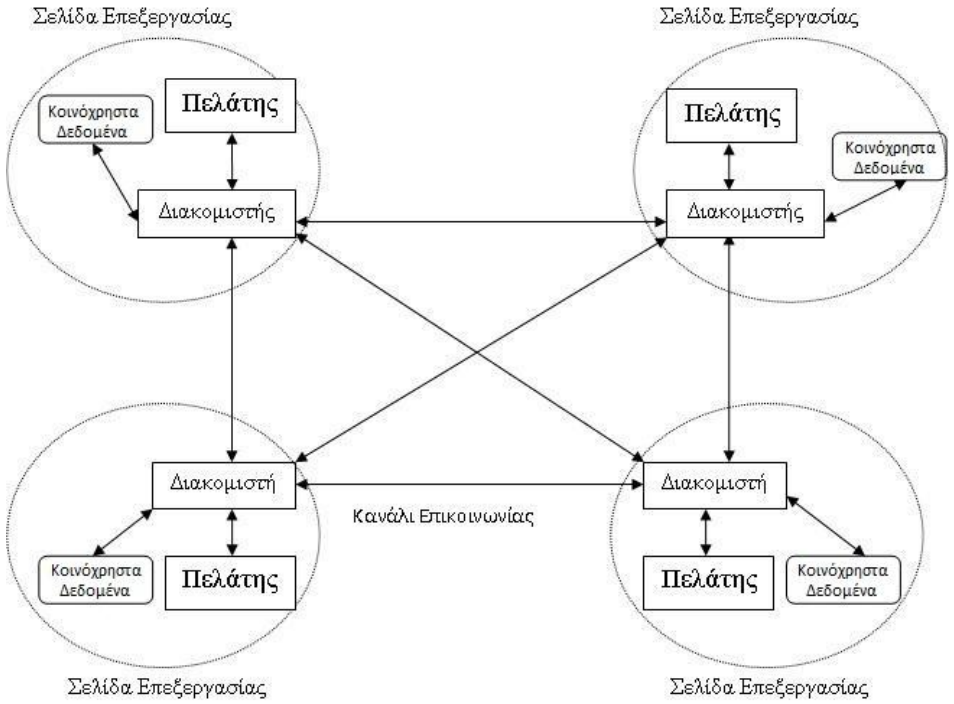
1. Ποιο σίγουρη υποστήριξη και ανταλλαγή πληροφορίας επειδή υπάρχει ένας διακομιστής.
2. Πιο εύκολη εγκατάσταση λογισμικού και αναβαθμίσεων.
3. Εύκολος χειρισμός και βελτιωμένες υπηρεσίες στους τελικούς χρήστες.

Αντίθετα στα μειονεκτήματα έχουμε:

1. Υψηλές απαιτήσεις σε εύρος ζώνης αλλιώς έχουμε μεγάλο χρόνο απόκρισης
2. Περιορισμένη απόδοση συστήματος
3. Σε περίπτωση προβλήματος του διακομιστή ή προσωρινής κατάρρευσης, όλο το σύστημα καταρρέει.

2.4.2. Αρχιτεκτονική Αντιγραφών (Replicated architecture)

Με την αρχιτεκτονική αντιγραφών, δεν υπάρχει τοποθεσία διακομιστή. Ο εξυπηρετητής και τα κοινόχρηστα δεδομένα αναπαράγονται σε όλες τις τοποθεσίες επεξεργασίας. Έτσι, κάθε περιοχή επεξεργασίας περιέχει ένα «client», έναν διακομιστή, και ένα αντίγραφο των κοινόχρηστων εγγράφων. Ο εξυπηρετητής σε κάθε περιοχή είναι υπεύθυνος για τη διατήρηση της σταθερότητας του αντίγραφου των κοινόχρηστων δεδομένων. Οι διακομιστές επικοινωνούν απευθείας ο ένας με τον άλλο για να διασφαλίζουν ότι όλες οι ενημερώσεις είναι διαθέσιμες σε όλες τις τοποθεσίες για να εκτελεστούν. (Αυτή η αρχιτεκτονική φαίνεται στην Εικόνα 2 **: Αρχιτεκτονική Αντιγραφών**).



Εικόνα 2 : Αρχιτεκτονική Αντιγραφών

Τα «collaborative editing systems» με αρχιτεκτονική αντιγραφών μπορούν να προσφέρουν γρήγορο χρόνο απόκρισης με άμεση εκτέλεση. Όταν μια τοπική λειτουργία παράγεται, αυτή εκτελείται άμεσα από τον τοπικό διακομιστή και το αποτέλεσμα ενημερώνεται στον «client» άμεσα (χωρίς απαραίτητο δίκτυο επικοινωνίας). Έτσι ο χρόνος απόκρισης είναι ανεξάρτητος από την αστάθεια του δικτύου. Όμως, το μειονέκτημα αυτής της προσέγγισης είναι ότι μπορεί να

συμβούν ασυνέπειες στα αναπαραγόμενα αντίγραφα, εξαιτίας της ταυτόχρονης ενημέρωσης στα κοινόχρηστα δεδομένα, που δημιουργείται από τους πολλαπλούς διακομιστές.

2.4.3. Υβριδική Αρχιτεκτονική (Hybrid architecture)

Επιπλέον της κεντρικοποιημένης και την αρχιτεκτονικής αντιγραφών, είναι επίσης πιθανό ένα υβρίδιο αυτών των δύο. Με την υβριδική αρχιτεκτονική, κάθε περιοχή επεξεργασίας περιέχει ένα «client», ένα διακομιστή και ένα αντίγραφο των κοινόχρηστων δεδομένων. Υπάρχει επίσης μια τοποθεσία διακομιστή που περιέχει τα κοινόχρηστα δεδομένα.

Λειτουργίες οι οποίες δεν προκαλούν αστάθεια εκτελούνται ταυτόχρονα. Πρώτα τοπικά και μετά αποστέλλονται στις απομακρυσμένες περιοχές(συμπεριλαμβάνοντας και την τοποθεσία του εξυπηρετητή). Έτσι έχουμε έναν καλό χρόνο απόκρισης. Αντίθετα άλλα είδη λειτουργιών, όπως λειτουργίες ενημέρωσης, εκτελούνται μέσω του διακομιστή για να αποφευχθούν αστάθειες.**[8]**

2.5. Τεχνολογίες Κατανεμημένης Επικοινωνίας

Για την υποστήριξη της κατανεμημένης επικοινωνίας πληροφορίας υπάρχει πλήθος τεχνολογικών εργαλείων τα οποία αναλύονται στη συνέχεια.

- **Απομακρυσμένη Κλήση Μεθόδου (RMI)**

Η εν λόγω τεχνολογία είναι βασισμένη στη java και μπορεί να υποστηρίξει μέσω κατάλληλων υποδομών τη διαφανή εκτέλεση κατανεμημένα υλοποιημένων μεθόδων σαν να πρόκειτο για τοπικές. Βασικό μειονέκτημα αυτής η υποστήριξη κατανεμημένης επικοινωνίας αποκλειστικά και μόνο μεταξύ προγραμμάτων υλοποιημένων σε java.

- **Κοινή Αρχιτεκτονική Αιτήσεων για τη Διάθεση Αντικειμένων (CORBA)**

‘Το πρότυπο CORBA (Common Object Request Broker Architecture), αποτελεί μία Κοινή Αρχιτεκτονική Διαχείρισης Αιτήσεων για τη Διάθεση Αντικειμένων λογισμικού. Το ενδιάμεσο επίπεδο του CORBA είναι η Διαχείριση Αιτήσεων Διάθεσης Αντικειμένων – Object Request Broker (ORB) που καθιερώνει τις σχέσεις μεταξύ των Τερματικών Οντοτήτων(ΤΟ) και των εξυπηρετητών μέσω ειδικών αντικειμένων λογισμικού (ΑΛ). Επειδή το CORBA δεν σχεδιάστηκε για τη υποστήριξη τηλεσυνεργατικών συστημάτων, έχουν προταθεί διάφορες αλλαγές για να προσαρμοστεί κατάλληλα και να μπορεί να εξυπηρετεί συστήματα CSCW πραγματικού χρόνου. Μερικά τα χαρακτηριστικά του CORBA που πρέπει να επανασχεδιαστούν στην περίπτωση του CSCW είναι τα παρακάτω:

- Το CORBA χρειάζεται να διαμορφωθεί κατάλληλα έτσι ώστε να εξαλειφθούν αποτελεσματικά οι ιδιότητες που δεν απαιτούνται από τις τηλεσυνεργατικές εφαρμογές ΤΟ πραγματικού χρόνου. Για παράδειγμα η δυναμική επίκληση (dynamic invocation) δέχεται σημαντική λειτουργική επιβάρυνση(overhead) κατά τη χρήση της από ένα σύστημα CSCW πραγματικού χρόνου.

- Η διαφάνεια στο βασικό πρότυπο CORBA επιτρέπει την επίκληση ενός αντικειμένου από έναν απομακρυσμένο κόμβο χωρίς ανώτατο όριο στον λανθάνοντα χρόνο όταν δεν υπάρχουν έλεγχοι για το φορτίο του δικτύου. Αυτό μπορεί να έχει σοβαρές επιπτώσεις στην απόδοση ενός συστήματος CSCW λόγο καθυστερήσεων.
- Η υπηρεσία γεγονότων του CORBA καθορίζει τα κανάλια γεγονότων. Στη περίπτωση του CSCW οι ΤΟ χρησιμοποιούν ένα υποσύνολο των γεγονότων από τον Παροχέα υπηρεσιών, και επομένως πρέπει να εφαρμόσουν επιπρόσθετη διεργασία φιλτραρίσματος για να απορρίψουν τα μη χρήσιμα υποσύνολα γεγονότων.'**[4]**

- **Μοντέλο Αντικειμένων Κατανεμημένων Στοιχείων (DCOM)**

'Το πρότυπο Μοντέλο Αντικειμένων Κατανεμημένων Στοιχείων (Distributed Component Object Model DCOM έχει προταθεί από τη Microsoft και έχει μερικά ενδιαφέροντα χαρακτηριστικά γνωρίσματα που μπορούν να βοηθήσουν τη συνεργασία πραγματικού χρόνου. Κάποια από τα γνωρίσματα αυτά είναι:

- Το DCOM μπορεί να προσδιορίσει τα αντικείμενα στα οποία οι ΤΟ στέλνουν ένα αίτημα επανειλημμένα έτσι ώστε να δημιουργήσει τα ανάλογα αντικείμενα και να ενημερώσει τον κεντρικό εξυπηρετητή. Κατόπιν αιτήσεως από τη ΤΟ για ένα συγκεκριμένο αντικείμενο, το DCOM ενεργοποιεί το αντικείμενο από την πλευρά του κεντρικού εξυπηρετητή. Μετά από την έναρξη της αλληλεπίδρασης το DCOM στέλνει μόνο την ταυτότητα του συνόλου από το οποίο υποβλήθηκε το αίτημα.
- Το DCOM παρέχει τη δυνατότητα διαβιβάσεων των "δέλτα-απαιτήσεων". Αυτό είναι, ένα χαρακτηριστικό γνώρισμα που μπορεί να χρησιμοποιηθεί αποτελεσματικά σε συνεργασία πραγματικού χρόνου όσον αφορά τη μετάδοση των αλλαγών των γεγονότων.

Εντούτοις μερικά χαρακτηριστικά του DCOM δημιουργούν σημαντικά εμπόδια στην πραγματοποίηση τηλεσυνεργασίας, όπως για παράδειγμα το ότι δεν παρέχει υποστήριξη πολύ-πλατφόρμων. Επίσης, παρότι η υποστήριξη της δυναμικής επίκλησης (dynamicinvocation) είναι ένα από τα ουσιαστικά πλεονεκτήματα του DCOM, εντούτοις δεν υπάρχει κανένας ενδογενής μηχανισμός που να εγγυάται το συγχρονισμό των γεγονότων.'**[4]**

2.6. Ενημερότητα (Awareness)

Η ενημερότητα έχει να κάνει με τη διαθεσιμότητα και δυνατότητα μετάδοσης πληροφοριών, που φαίνεται σε τι κατάσταση είναι ο κάθε χρήστης οποιαδήποτε χρονική στιγμή (status). Σκοπός είναι ο καλύτερος συντονισμός των χρηστών με αποτέλεσμα μια πιο αποτελεσματική συνεργασία.

Όσον αφορά τη συνεργασία βασιζόμενη σε εικονικούς διαμοιραζόμενους χώρους, το είδος των πληροφοριών που φαίνεται πως είναι καταλληλότερο να μεταφέρεται μεταξύ των συνεργαζόμενων χρηστών - συμμετεχόντων και που κερδίζει συνεχώς έδαφος είναι αυτό που

περιγράφεται από τους Gutwin & Greenberg. Αναφέρει πως τα βασικά ερωτήματα στα οποία πρέπει να μπορεί να απαντάει ένας χρήστης είναι:

1. Ποιος χρήστης (ιδιότητα)
2. Τι κάνει(δηλαδή ποια ενέργεια εκτελεί)
3. Που(Δηλαδή ποιο αντικείμενο επεξεργάζεται ο χρήστης)**[9]**

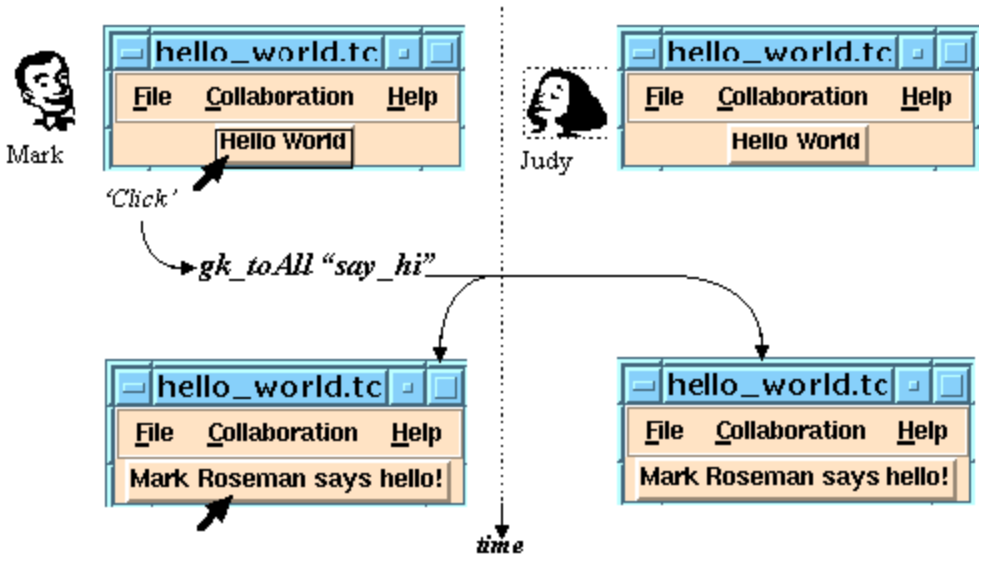
Στις παρακάτω εικόνες και πίνακες φαίνονται πιο αναλυτικά τα ερωτήματα.

Κατηγορία	**Στοιχείο**	**Συγκεκριμένη Ερώτηση**
Πως	Ιστορικό πράξεων	Πως έγινε αυτή η πράξη;
	Ιστορικό τεχνουργημάτων	Πως η εφαρμογή ήρθε σε αυτή την κατάσταση;
Πότε	Ιστορικό Συμβάντων	Πότε έγινε αυτή η ενέργεια;
Ποιος	Ιστορικό χρηστών	Ποιος ήταν ο χρήστης;
Που	Ιστορικό τοποθεσίας	Που ήταν κάποιος χρήστης;
Τι έκανε	Ιστορικό κινήσεων	Τι έχει κάνει κάποιος χρήστης;

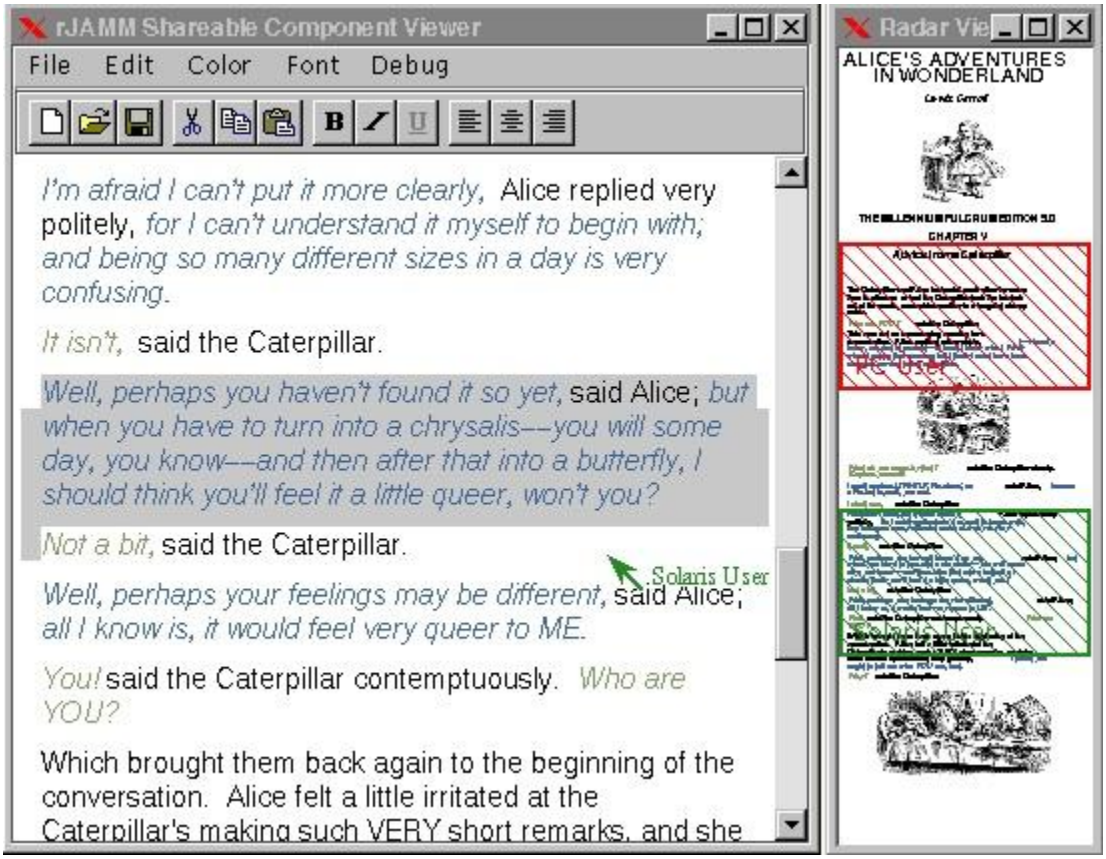
Πίνακας 1 : Στοιχεία του χώρου εργασίας με το παρελθόν

Κατηγορία	**Στοιχείο**	**Συγκεκριμένη ερώτηση**
Ποιος	Ποιος είναι	Είναι κανείς στο χώρο εργασίας;
	Ταυτότητα	Ποιος χρήστης συμμετέχει;
Τι	Σύνταξη	Ποιος κάνει αλλαγές;
	Πράξη	Τι κάνουν οι χρήστες
	Πρόθεση	Τι σκοπό έχουν οι πράξεις;
	Τεχνούργημα	Σε ποιο αντικείμενο εργάζονται;
Που	Τοποθεσία	Που είναι ο κάθε χρήστης;

Πίνακας 2 : Στοιχεία του χώρου εργασίας με την παρούσα κατάσταση



Εικόνα 3 : Groupkit toolkit



Εικόνα 4 : Παραδείγματα μηχανισμών ενημερότητας

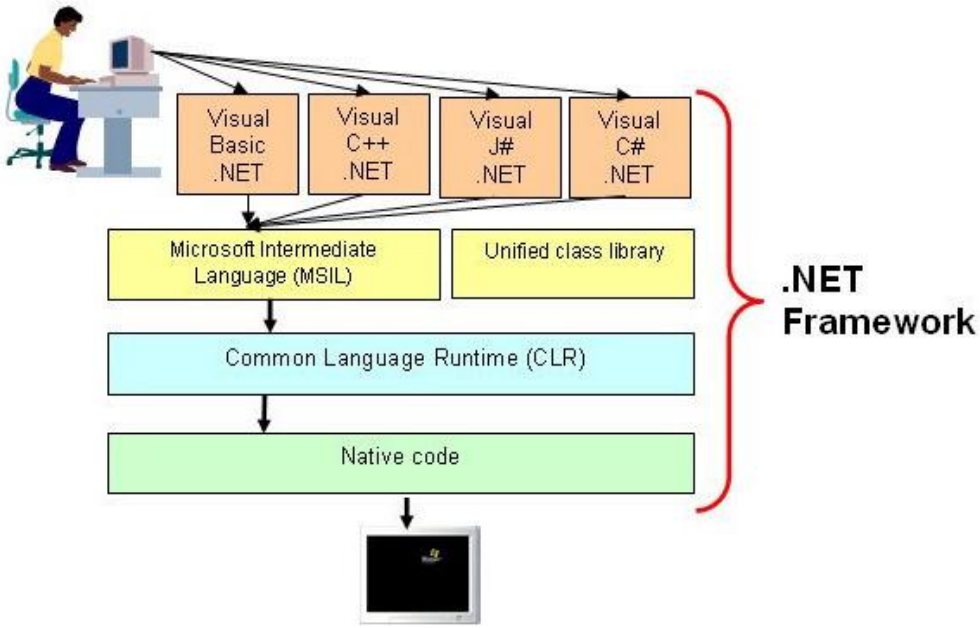
3.Ανάπτυξη λογισμικού για κινητές συσκευές

Σε αυτό το κεφάλαιο περιγράφονται εκείνες οι τεχνολογίες με τις οποίες μπορούμε να αναπτύξουμε λογισμικό για έξυπνες κινητές συσκευές, καθώς επίσης αναφέρουμε πλεονεκτήματα και περιορισμούς που τίθενται από την εκάστοτε αρχιτεκτονική της κάθε πλατφόρμας.

3.1. Windows phone OS

Τα Windows phone δημιουργήθηκε από την Microsoft και είναι λειτουργικό σύστημα για έξυπνα κινητά τηλέφωνα(smartphones). Πριν από το Windows phone η Microsoft χρησιμοποιούσε το Windows mobile. Τα δυο λειτουργικά δεν είναι συμβατά μεταξύ τους.

Όπως με το iOS της Apple, το Android της Google και το BlackBerry OS, οι εφαρμογές τρίτων έχουν τη δυνατότητα να αναπτυχθούν για τα Windows Phone της Microsoft και είναι διαθέσιμα από το εικονικό κατάστημα της Microsoft όπου ο χρήστης έχει τη δυνατότητα να «κατεβάσει» ή να αγοράσει εφαρμογές, μουσική, ταινίες, παιχνίδια, και podcast για το τηλέφωνό του. Για δημιουργία εφαρμογών στο συγκεκριμένο λειτουργικό σύστημα οι προγραμματιστές πρέπει να χρησιμοποιήσουν μια ή περισσότερες γλώσσες προγραμματισμού από την .Net πλατφόρμα. Η.Net(ή .Netframework) είναι ένα πλαίσιο εργασίας αποκλειστικά για windows λειτουργικά. Γλώσσες που υποστηρίζει το .Netframework είναι η c++,VisualBasic .NET, Script.NET, JScript .NET κ.ά. Η Microsoft δεν επιτρέπει σε smartphones - που έχουν τις εκδόσεις των Windows Mobile - να αναβαθμιστούν σε Windows Phone. Επομένως, δεν υπάρχει συμβατότητα στις εφαρμογές ανάμεσα σε smartphone που υποστηρίζουν Windows Mobile και Windows Phone.



Εικόνα 5 : .Netframework

Ίσως το σημαντικότερο αντικείμενο το οποίο οφείλει ο κάθε σχεδιαστής λογισμικού να ξέρει πριν ξεκινήσει την υλοποίηση της εφαρμογής του, είναι ο κύκλος ζωής της εφαρμογής (Εικόνα 7 **: Κύκλος ζωής μιας εφαρμογής**). Ο κύκλος ζωής παρέχει τη γνώση για τη σωστή λειτουργία μιας εφαρμογής και τον τρόπο με τον οποίο αυτή αντιδρά στις επιλογές του χρήστη.

Στα Windows Phone υπάρχουν τέσσερα διαφορετικά συμβάντα, όπου το καθένα επιτελεί διαφορετικές λειτουργίες:

- **Εκκίνηση.** Ξεκινώντας μια εφαρμογή που θέλει ο χρήστης να χειριστεί, «εκτελείται» αυτόματα το συμβάν της εκκίνησης, στο οποίο ο προγραμματιστής μπορεί να προσαρμόσει τον κώδικά του, ώστε να προβεί σε κάποιες χρήσιμες λειτουργίες με το άνοιγμα της εφαρμογής. Ωστόσο, για να γίνει αποδοτικότερη η εκκίνηση χωρίς προβλήματα, μια καλή πρακτική είναι να μη χρησιμοποιούνται «βαριές» δραστηριότητες στο συγκεκριμένο συμβάν.
- **Απενεργοποίηση**. Κατά την απομάκρυνση από την εφαρμογή, «εκτελείται» το συμβάν της απενεργοποίησης. Τότε αποθηκεύεται η κατάσταση της προηγούμενης εφαρμογής και γίνεται εκκίνηση της επόμενης. Σε αυτήν την περίπτωση, δίνεται η δυνατότητα στο χρήστη να επιστρέψει στην προηγούμενη εφαρμογή χωρίς να υπάρξουν μεταβολές στα δεδομένα, όπως θα συνέβαινε αν η εφαρμογή είχε κλείσει οριστικά.
- **Ενεργοποίηση.** Όταν ο χρήστης επιθυμεί να επιστρέψει στην προηγούμενη αδρανή εφαρμογή, τότε «εκτελείται» το συμβάν της ενεργοποίησης. Δηλαδή επαναφέρει την εφαρμογή στο προσκήνιο αφού προηγουμένως την είχε αποθηκεύσει μέσω του συμβάντος της απενεργοποίησης. Ο χρήστης μπορεί να συνεχίσει από το σημείο που είχε σταματήσει πριν ξεκινήσει η απενεργοποίηση της εφαρμογής.
- **Κλείσιμο**. Εφόσον ο χρήστης επιλέξει να τερματίσει την εφαρμογή, τότε «εκτελείται» το συμβάν του κλεισίματος. Σ' αυτό το σημείο ο προγραμματιστής μπορεί να γράψει τον κώδικα που θεωρεί απαραίτητο, ώστε να αποθηκεύσει χρήσιμα στοιχεία πριν το κλείσιμο. Καλό είναι να μην χρησιμοποιούνται από τους προγραμματιστές πλήκτρα εξόδου από την εφαρμογή, για να διατηρηθεί η συνοχή της ροής των Windows Metro.

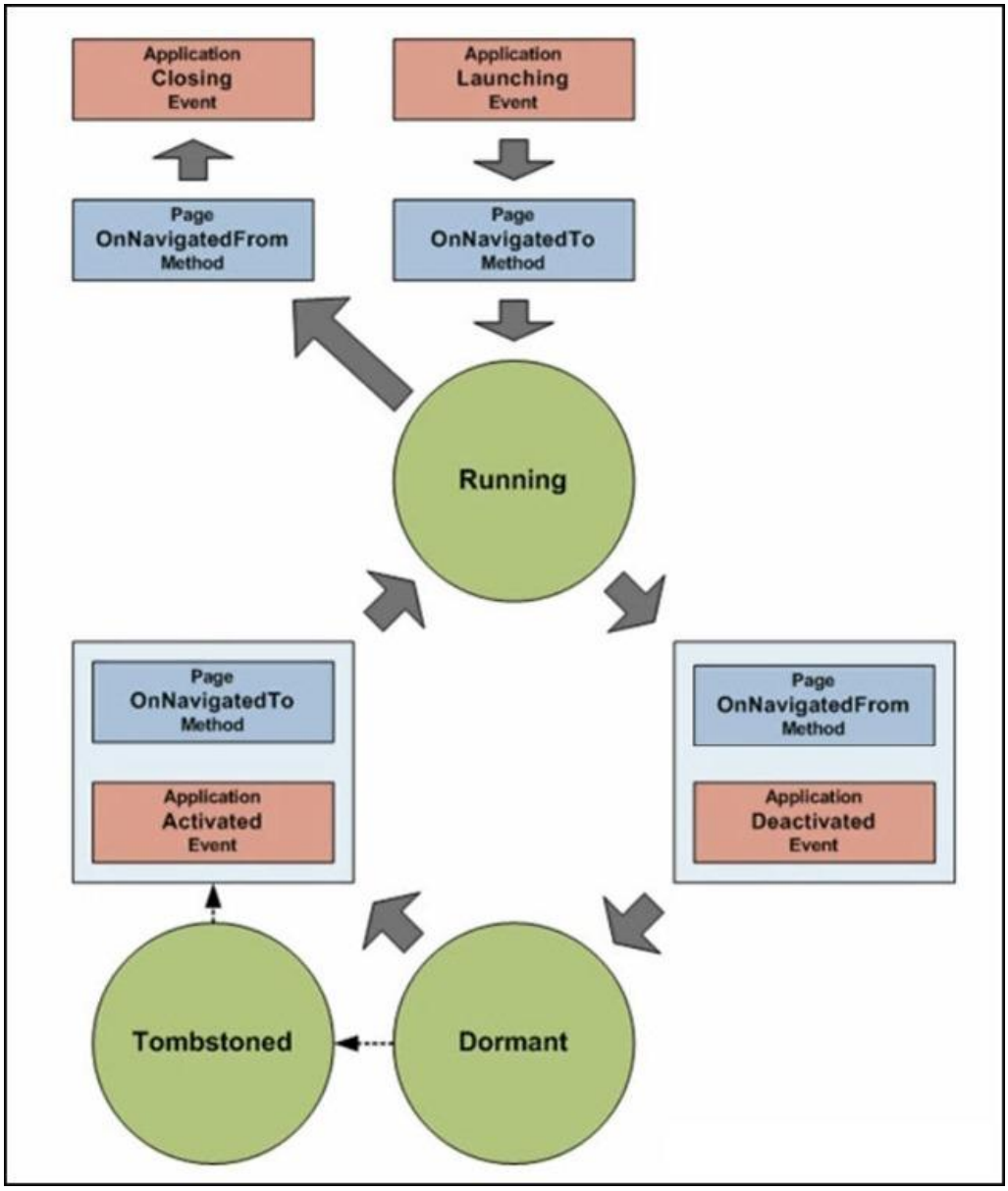
Εκτός από τα συμβάντα που «εκτελούνται» σε μια εφαρμογή, υπάρχουν και οι καταστάσεις στις οποίες μια εφαρμογή μπορεί να βρεθεί. Συγκεκριμένα, υπάρχουν τρεις διαφορετικές καταστάσεις οι οποίες αναλύονται ακολούθως:

- **Κατάσταση λειτουργίας:** Είναι η κατάσταση στην οποία τίθεται όταν η εφαρμογή είναι ανοικτή στην οθόνη της συσκευής και ο χρήστης αλληλεπιδρά με αυτή. Παραμένει μέχρι ο χρήστης να επιλέξει να απομακρυνθεί από την εφαρμογή ή η συσκευή να εισέλθει σε κατάσταση κλειδώματος.
- **Κατάσταση αδράνειας:** Είναι η κατάσταση στην οποία εισέρχεται μια εφαρμογή που είναι αποθηκευμένη στο παρασκήνιο και αναμένει την επιλογή του χρήστη σε περίπτωση που θελήσει να την επαναφέρει. Όταν η συσκευή έχει ελάχιστη μνήμη (και υποχρεωτικά για να λειτουργήσει θα χρειαστεί επιπλέον μνήμη), τότε το λειτουργικό

σύστημα επιλέγει να τερματίσει μερικές εφαρμογές που βρίσκονται σε αυτήν την κατάσταση.

- **<u>Κατάσταση απενεργοποίησης:</u>** Είναι η κατάσταση στην οποία βρίσκεται μια εφαρμογή που προορίζεται για οριστικό τερματισμό. Το σύστημα μπορεί να διαχειριστεί μέχρι πέντε διαφορετικές εφαρμογές σε αυτήν την κατάσταση για αποθήκευση δεδομένων ή πληροφοριών.**[5]**

Εικόνα 6 : Οθόνη του windowsphoneOS

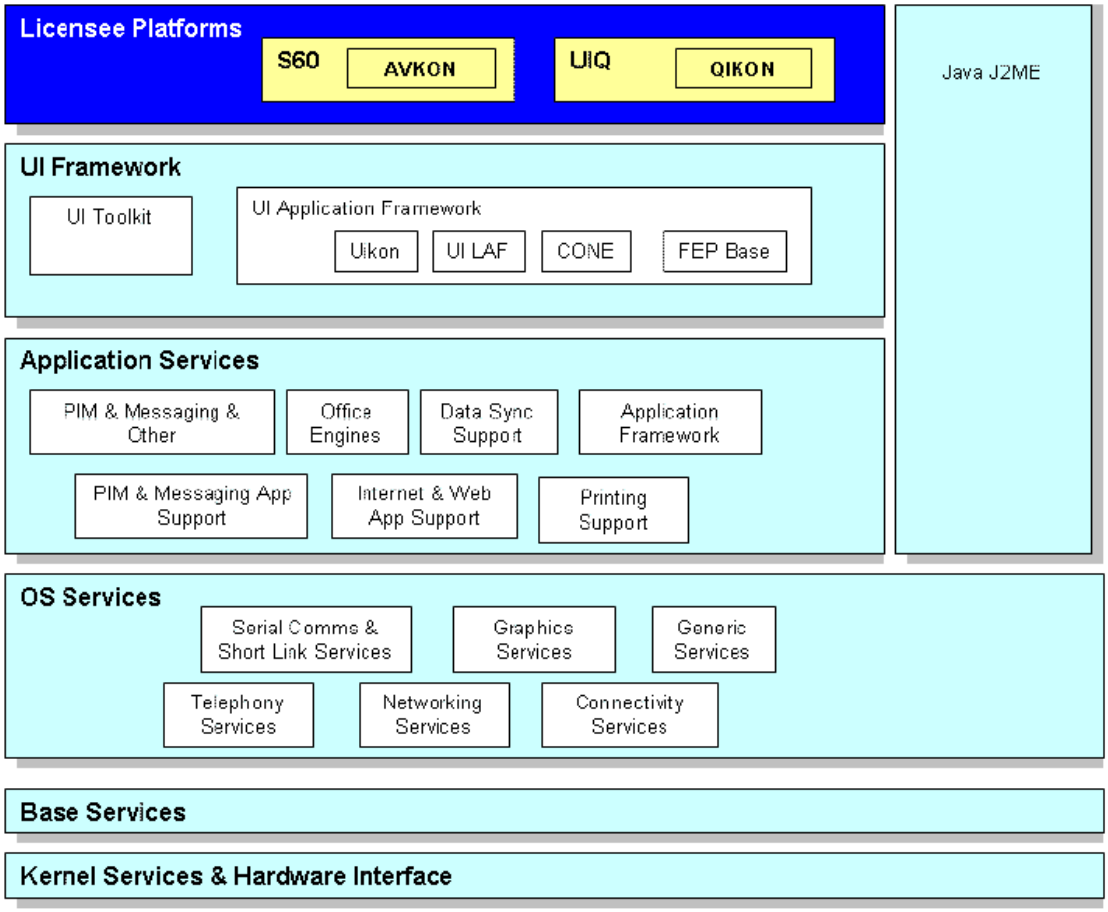


Εικόνα 7 : Κύκλος ζωής μιας εφαρμογής

3.2. SymbianOS

Το Symbian OS είναι λειτουργικό σύστημα για φορητές συσκευές. Αποτελεί εξέλιξη του λειτουργικού συστήματος EPOC από την Psion. Το Symbian OS δημιουργήθηκε με τη γλώσσα προγραμματισμού C++ από τη SymbianLtd. Πριν το 2009 το Symbian OS υποστήριζε διαφορετικά περιβάλλοντα χρήστη. Όμως με την δημιουργία του Symbian Platform, το ίδιο έτος, τα 3 βασικά περιβάλλοντα χρήστη ενώθηκαν σε ένα, το οποίο εξαγοράστηκε από την Nokia και στην συνεχεία μετατράπηκε σε λογισμικό ανοικτού κώδικα. Τα τελευταία χρόνια το μερίδιο του λειτουργικού αυτού συστήματος στην αγορά μειώνεται, αν και οι συσκευές με λογισμικό

Symbian εξακολουθούν να πωλούνται σε μεγάλους αριθμούς στην αγορά. Για την ανάπτυξη εφαρμογών στο περιβάλλον του λειτουργικού υπάρχει το Symbian SDK που χρησιμοποιεί ως γλώσσα προγραμματισμού την C++ σε συνδυασμό με το Qt (Framework εφαρμογών που χρησιμοποιείται από πολλές πλατφόρμες). Για την δοκιμή των εφαρμογών χρησιμοποιείται ένας εξομοιωτής, που τρέχει τον κώδικα απευθείας αντί να προσομοιώνει την λειτουργία του κινητού τηλεφώνου. Το λειτουργικό αυτό λειτουργεί αποκλειστικά με επεξεργαστές ARM. Στην Εικόνα 8 **:Αρχιτεκτονική του SymbianOS** φαίνεται η αρχιτεκτονική του SymbianOS.**[3]**



Εικόνα 8 :Αρχιτεκτονική του SymbianOS

Εικόνα 9 : Η κεντρική οθόνη του SymbianOS

3.3. AppleiOS

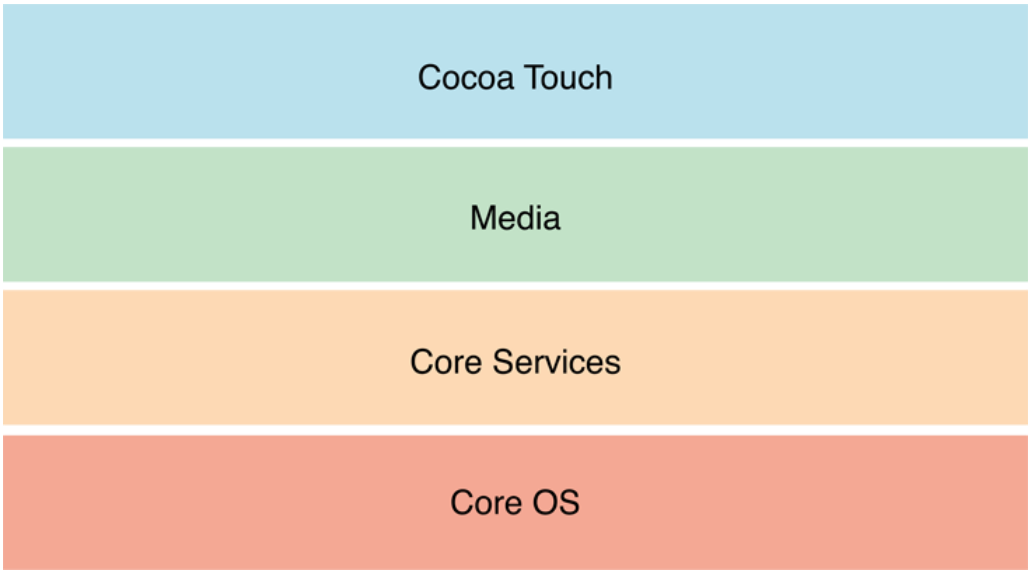
Το iOS είναι το λειτουργικό σύστημα (Operating System, OS) που τρέχει στις φορητές συσκευές της Apple (iPhone, iPod iPad). Η συσκευή πωλείται έχοντας προ εγκατεστημένες στο λειτουργικό της σύστημα κάποιες βασικές εφαρμογές, όπως το Τηλέφωνο, το Mail (εφαρμογή ηλεκτρονικού ταχυδρομείου) και το Safari (περιηγητής ιστού)οι οποίες παρέχουν τις τυπικές υπηρεσίες του συστήματος στο χρήστη. Το λειτουργικό σύστημα διαχειρίζεται το υλικό (hardware) της συσκευής και προσφέρει τις τεχνολογίες που είναι απαραίτητες για τη λειτουργία των εγκατεστημένων εφαρμογών. Το iOS Software Development Kit (SDK) περιέχει διεπαφές και εργαλεία που χρειάζονται για την ανάπτυξη, εγκατάσταση, δοκιμή και τη λειτουργία των εφαρμογών που εμφανίζονται στην αρχική οθόνη μιας συσκευής. Όλες οι εφαρμογές δημιουργούνται χρησιμοποιώντας τα πλαίσια (frameworks) του συστήματος του iOS και την αντικειμενοστραφή γλώσσα προγραμματισμού Objective-C. Η κατανόηση των εργαλείων και των τεχνολογιών που συνθέτουν το iOS SDK είναι αναγκαία για να σχεδιασθεί και να υλοποιηθεί σωστά μία εφαρμογή.

Στο υψηλότερο επίπεδο, το iOS ενεργεί ως μεσάζων μεταξύ του υποκείμενου hardware και των εφαρμογών που εμφανίζονται στην οθόνη. Οι εφαρμογές σχεδόν ποτέ δεν επικοινωνούν άμεσα με το hardware της συσκευής. Αντίθετα, η επικοινωνία γίνεται μέσω μιας σειράς καθορισμένων διεπαφών του συστήματος, γεγονός που προστατεύει την εφαρμογή από αλλαγές στο hardware.

Αυτό σημαίνει ότι οι εφαρμογές μπορούν να λειτουργούν σε συσκευές με διαφορετικό hardware. Οι τεχνολογίες του iOS μπορούν να συγκεντρωθούν σε τέσσερα στρώματα:

a) Το στρώμα Cocoa Touch
b) Το στρώμα Media
c) Το στρώμα Core Services
d) Το στρώμα Core OS

Στα κατώτερα στρώματα του συστήματος βρίσκονται οι θεμελιώδεις υπηρεσίες και τεχνολογίες, που βασίζονται όλες οι εφαρμογές, ενώ τα στρώματα υψηλότερου επιπέδου περιέχουν πιο εξελιγμένες υπηρεσίες και τεχνολογίες.



Εικόνα 10 : Η Διαστρωματωμένη Αρχιτεκτονική του iOS.

Η Apple προσφέρει τις περισσότερες από τις διεπαφές του συστήματος, οργανωμένες σε πακέτα που ονομάζονται πλαίσια (frameworks). Πλαίσιο είναι ένας κατάλογος ο οποίος περιέχει μια δυναμική κοινή βιβλιοθήκη και τους πόρους (όπως αρχεία επικεφαλίδας, εικόνες) που χρειάζονται για την υποστήριξη της. Για να χρησιμοποιηθεί κάποιο πλαίσιο απαιτείται η σύνδεση της εφαρμογής με αυτό, όπως γίνεται με όλες τις κοινές βιβλιοθήκες. Η σύνδεση αυτή δίνει την απαραίτητη πρόσβαση στα χαρακτηριστικά του πλαισίου. Πολλές από τις τεχνολογίες του iOS είναι κοινές με αυτές του OS X(λειτουργικό συστήμα των υπολογιστών της Apple). Οι μεγαλύτερες διαφορές βρίσκονται στο επίπεδο της διεπαφής χρήστη, αλλά ακόμη και εκεί υπάρχουν αρκετές ομοιότητες.

Το στρώμα Core OS

Το στρώμα Core OS περιέχει όλα τα χαρακτηριστικά στα οποία βασίζονται οι περισσότερες άλλες τεχνολογίες. Ακόμα κι αν δε χρησιμοποιούνται άμεσα από την εφαρμογή, χρησιμοποιούνται από άλλα πλαίσια. Στην περίπτωση που χρειάζεται ασφάλεια ή επικοινωνία με κάποιο εξωτερικό hardware, χρησιμοποιούνται τα πλαίσια αυτού του στρώματος. Τα βασικά πλαίσια που υπάρχουν σε αυτό το στρώμα είναι τα:

- Core Bluetooth Framework
- Security Framework
- Accelerate Framework
- Generic Security Framework
- System

Το στρώμα Core Services

Το στρώμα Core Services περιέχει τις θεμελιώδεις υπηρεσίες συστήματος τις οποίες χρησιμοποιούν όλες οι εφαρμογές. Ακόμα και αν δε χρησιμοποιούνται άμεσα από την εφαρμογή, πολλά μέρη του συστήματος είναι φτιαγμένα «πάνω» σε αυτές τις υπηρεσίες. Μερικές από τις βασικές τεχνολογίες είναι:

- Η υπηρεσία iCloud. Η υπηρεσία αυτή δίνει τη δυνατότητα στο χρήστη να αποθηκεύει τα δεδομένα του σε μία κεντρική τοποθεσία, στην οποία μπορεί να έχει πρόσβαση από όλες τις iOS συσκευές του και από τον υπολογιστή του.
- Το Grand Central Dispatch που επιτρέπει τη διαχείριση της εκτέλεσης των εργασιών της εφαρμογής.
- Η προστασία ευαίσθητων δεδομένων του χρήστη.
- Η υποστήριξη block αντικειμένων.
- Η υποστήριξη της βιβλιοθήκης SQLite.

Τα βασικά πλαίσια που βρίσκονται στο στρώμα αυτό είναι τα:

- Ad Support Framework
- Core Foundation Framework
- Accounts Framework
- Core Location Framework
- Social Framework
- Core Media Framework
- Foundation Framework
- Core Telephony Framework
- Core Data Framework
- Core Motion Framework

Το στρώμα Media

Το στρώμα Media περιέχει τις τεχνολογίες γραφικών, βίντεο και ήχου και προσανατολίζεται στη δημιουργία εφαρμογών πολυμέσων με πολύ καλή εμπειρία. Οι τεχνολογίες αυτές είναι πολύ εύκολες στη χρήση. Τα βασικά πλαίσια που περιέχονται στο στρώμα αυτό είναι τα:

- Core Image Framework
- Core Video Framework
- Core Text Framework
- Core Graphics Framework
- Quartz Core Framework
- Core Audio Framework
- GLKit Framework
- OpenGL ES Framework
- Media Player Framework

Το στρώμα Cocoa Touch

Το στρώμα Cocoa Touch περιέχει τα βασικά πλαίσια που είναι απαραίτητα για τη δημιουργία εφαρμογών στο iOS. Το στρώμα αυτό καθορίζει τη βασική υποδομή της εφαρμογής και την υποστήριξη των βασικών τεχνολογιών (multitasking, η τεχνολογία της οθόνης αφής, οι γνωστοποιήσεις τύπου push και άλλες). Κατά το σχεδιασμό της εφαρμογής, θα πρέπει να διερευνηθεί αν οι τεχνολογίες του στρώματος αυτού ανταποκρίνονται στις ανάγκες της εφαρμογής.

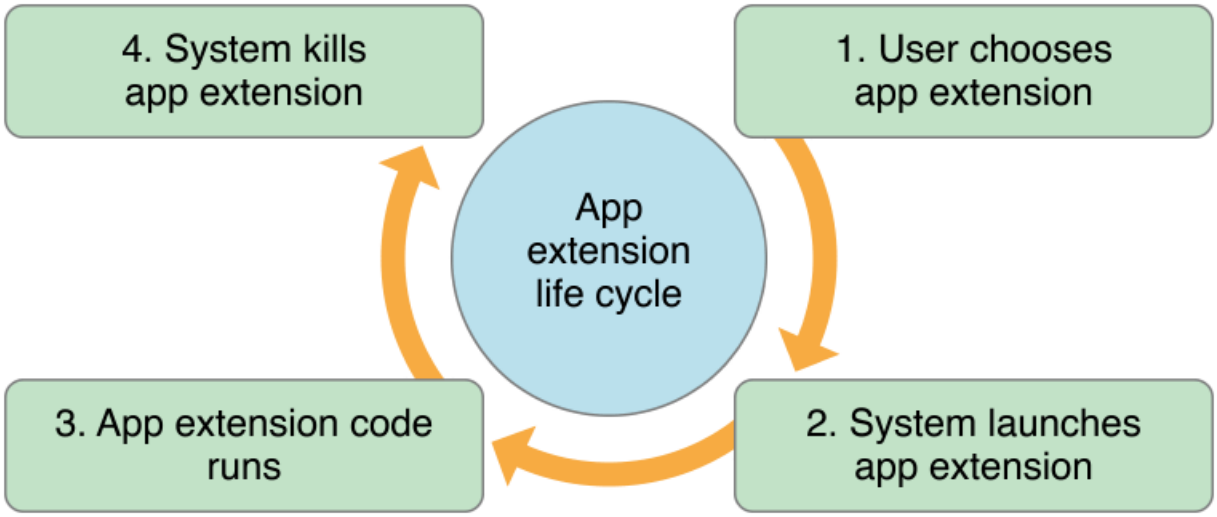
Μερικές από τις βασικές τεχνολογίες που είναι διαθέσιμες στο συγκεκριμένο στρώμα είναι:

- Η υποστήριξη των Storyboards, που βοηθούν στη σχεδίαση ολόκληρης της διεπαφής χρήστη σε ένα μέρος όπου μπορούμε να ελέγξουμε όλες τις οθόνες και τις συνδέσεις μεταξύ τους.
- Η υπηρεσία γνωστοποιήσεων.
- Η πολυδιεργασία (multitasking).
- Η εκτύπωση, που επιτρέπει την αποστολή περιεχομένου για εκτύπωση σε κοντινούς εκτυπωτές.
- Η αναγνώριση χειρονομιών.
- Η υποστήριξη εγγράφων.
- Η υποστήριξη πρότυπων οθονών του συστήματος, όπως η οθόνη σύνθεση μηνύματος ή η οθόνη προβολής των πληροφοριών μιας επαφής.
- Η αυτόματη διάταξη των στοιχείων της διεπαφής χρήστη.
- Οι υπηρεσίες μεταξύ ομότιμων χρηστών (PeertoPeer).

Τα πλαίσια που περιέχονται στο στρώμα αυτό είναι τα:

- Message UI Framework
- Event Kit UI Framework
- Game Kit Framework
- iAd Framework
- Address Book UI Framework
- Map Kit Framework

- UI Kit Framework
- Twitter Framework **[1]**



Εικόνα 11 : Κύκλος ζωής μιας εφαρμογής

Εικόνα 12 : Κεντρική οθόνη του AppleiOS

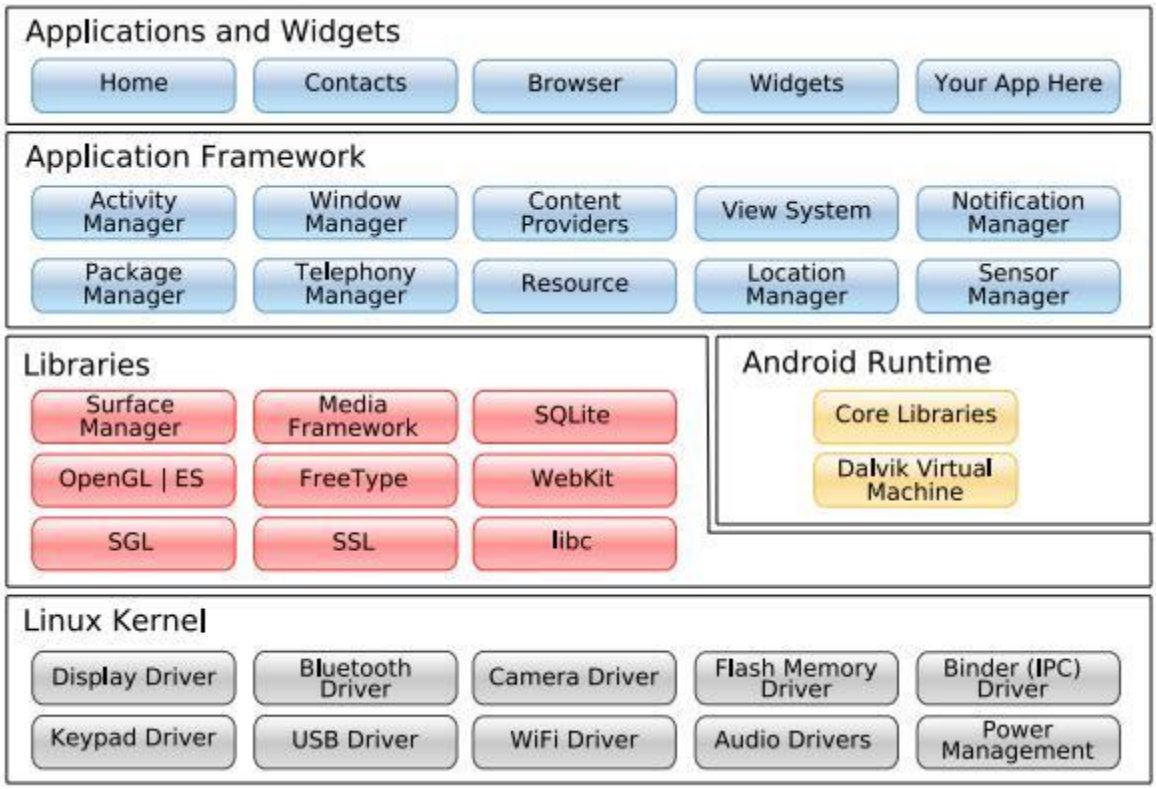
3.4. Android

Το Android είναι λογισμικό για κινητές συσκευές. Αποτελείται από λειτουργικό σύστημα, ενδιάμεσο λογισμικό και βασικές εφαρμογές. Αρχικά αναπτύχθηκε από την Google όμως αργότερα συνεχίστηκε σε συνεργασία με την Open Handset Alliance (OHA).Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού OHA(τηλεπικοινωνιακές εταιρίες, εταιρίες λογισμικού και εταιρίες κατασκευής υλικού, οι οποίες ασχολούνται με την ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις συσκευές ανοιχτής τηλεφωνίας). Παρακάτω αναφέρουμε μερικά μέλη του οργανισμού αυτού για να δείξουμε την τεράστια προοπτική που δημιουργείται:

- Motorola
- Samsung
- Google
- Vodafone

- Htc
- Sprint Nextel
- Sony Ericsson
- T-Mobile
- Texas Instruments
- Verizon

Το Android είναι ένα προϊόν ελεύθερου λογισμικού. Παρόλα αυτά ένα κομμάτι της ανάπτυξης του λογισμικού συνεχίζεται σε ιδιωτικό παρακλάδι. Για να έρθει αυτό το λογισμικό σε κοινή θέαση δημιουργήθηκε ένα παρακλάδι του, το οποίο έδινε τη δυνατότητα μόνο για ανάγνωση(Cupcake). Το Cupcake συνήθως συγχέεται με τον τίτλο μιας ενημέρωσης, σε αντίθεση με όσα δηλώνει η ίδια η Google στην ιστοσελίδα ανάπτυξης του Android: 'το Cupcake αποτελεί ακόμη ένα έργο σε εξέλιξη, όχι μια επίσημη έκδοση'. Αξιοσημείωτες αλλαγές στο λειτουργικό Android θα παρουσιαστούν στο cupcake που περιλαμβάνουν αλλαγές στο σύστημα διαχείρισης των μεταφορτώσεων (download manager), το κυρίως σύστημα, το framework, το σύστημα τηλεφωνίας, το λογισμικό συστήματος, το ραδιόφωνο, το Bluetooth, τα εργαλεία προγραμματισμού και διάφορες εφαρμογές, καθώς και πληθώρες διορθώσεις σφαλμάτων.**[11]**



Εικόνα 13 : Αρχιτεκτονική λειτουργικού συστήματος Android

Το Android αποτελείται από τα ακόλουθα στρώματα:

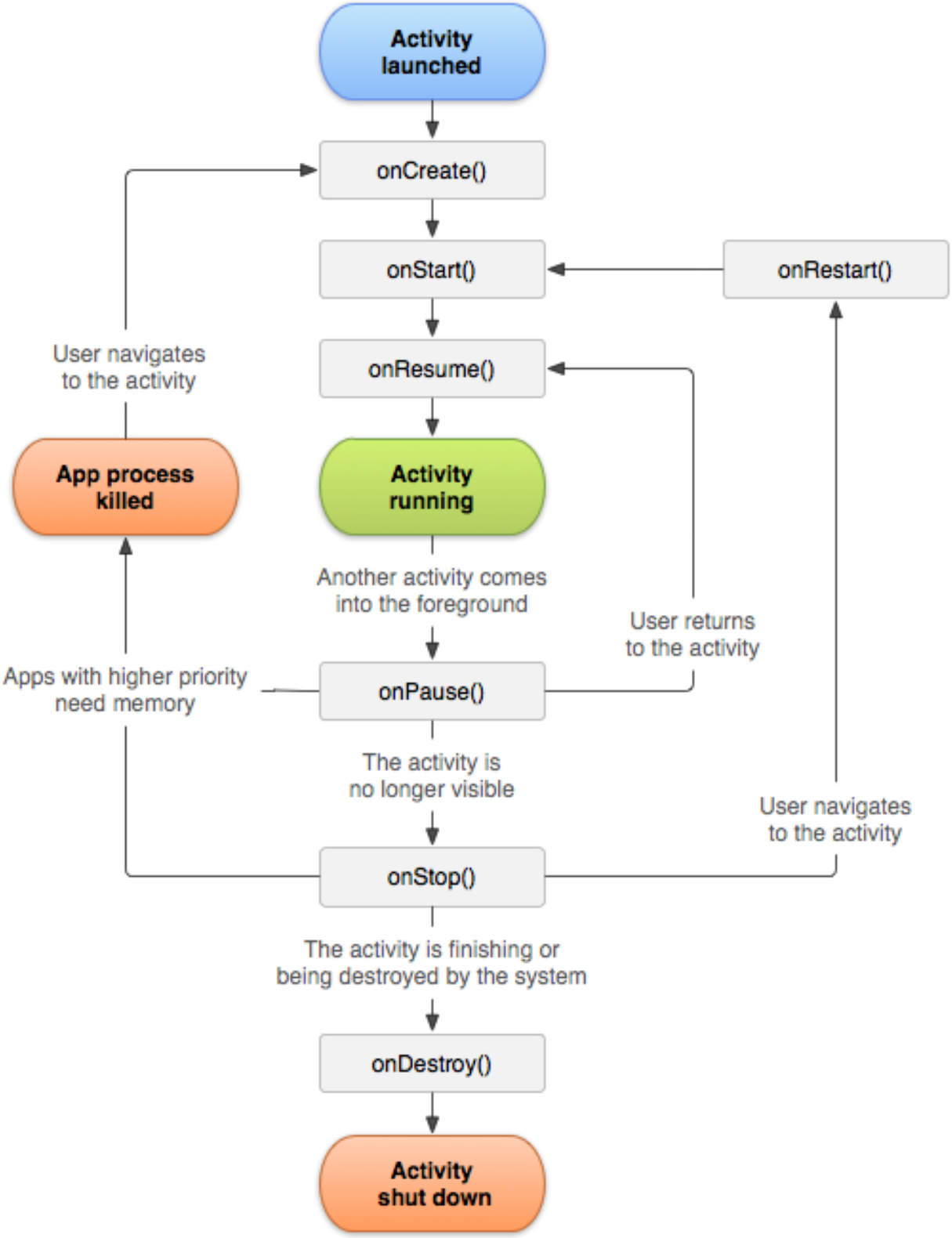
- το πυρήνας του Linux, ο οποίος περιλαμβάνει:
    - τους οδηγούς για το υλικό, τη δικτύωση, την πρόσβαση στο σύστημα αρχείων και στο εσωτερικό των διαδικασιών επικοινωνίας
- εφαρμογές (γραμμένες σε Java, σε εκτέλεση Dalvik)
- βιβλιοθήκες και υπηρεσίες (γραμμένες σε C ή C + +)
- πλαίσιο των υπηρεσιών και βιβλιοθήκες (γραμμένα κυρίως σε java)
    - οι εφαρμογές και οι περισσότεροι κωδικοί πλαισίου εκτελούνται σε μια εικονική μηχανή

Η αρχιτεκτονική του Android περιλαμβάνει τα εξής επίπεδα, ξεκινώντας από το χαμηλότερο:

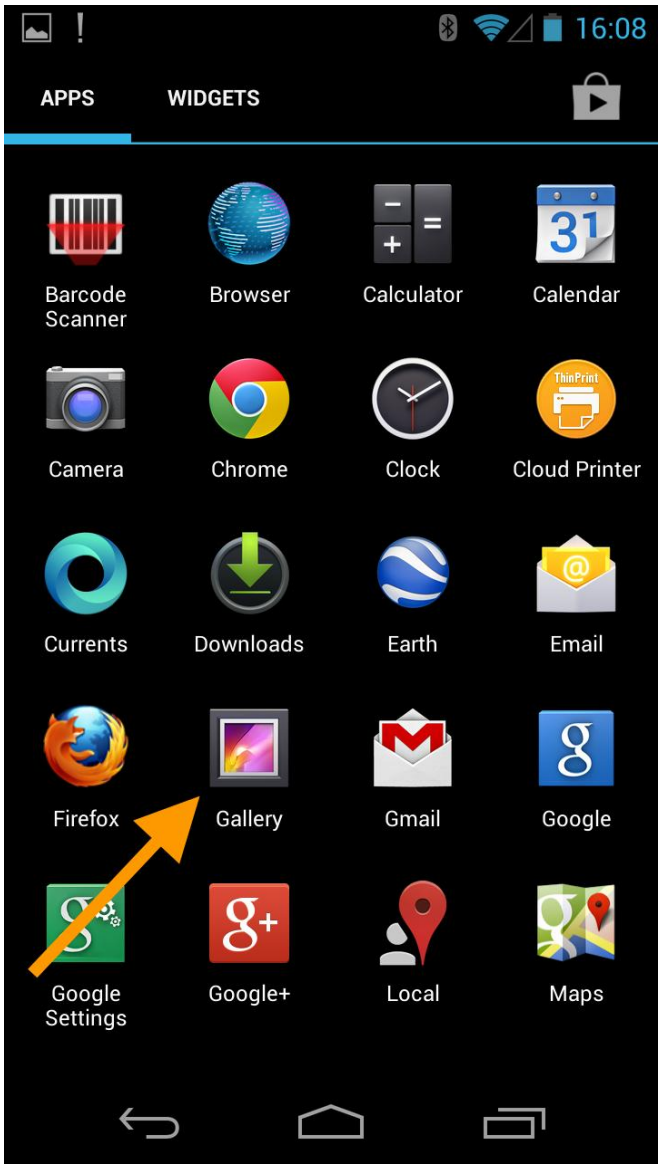
- **Πυρήνας του Linux:** Το Android βασίζεται στον πυρήνα Linux έκδοση 2.6 για βασικές υπηρεσίες συστήματος όπως ασφάλεια, διαχείριση διεργασιών, οδηγούς συσκευών, διαχείριση μνήμης και στοίβα δικτύου. Επίσης ο πυρήνας λειτουργεί ως ένα ενδιάμεσο επίπεδο αφαίρεσης μεταξύ του υλικού και της στοίβας λογισμικού.
- **Επίπεδο Εκτέλεσης (Android Runtime):** Το οποίο αποτελείται από ένα σύνολο από βασικές βιβλιοθήκες και την Dalvik Virtual Machine.
- **Επίπεδο Βιβλιοθηκών (Libraries**):Περιλαμβάνει ένα σύνολο από βιβλιοθήκες γραμμένες σε C/C++ που χρησιμοποιούνται από διάφορα στοιχεία του συστήματος. Οι δυνατότητες που προσφέρουν αυτές οι βιβλιοθήκες είναι η προσβασιμότητα στους προγραμματιστές μέσου του επιπέδου πλαισίου εφαρμογής.
- **Επίπεδο Πλαισίου Εφαρμογών (Applications Framework):**Το Android προσφέρει στους προγραμματιστές τη δυνατότητα να κατασκευάσουν πλούσιες και καινοτόμες εφαρμογές, ακολουθώντας μια ανοικτή πλατφόρμα ανάπτυξης. Οι προγραμματιστές είναι ελεύθεροι να αξιοποιήσουν πλήρως το hardware της συσκευής, να έχουν πρόσβαση σε υπηρεσίες εντοπισμού θέσης, να θέσουν χρονοδιακόπτες για εμφάνιση ειδοποιήσεων, να τρέξουν υπηρεσίες στο background και πολλά άλλα. Επίσης, έχουν πλήρη πρόσβαση στο ίδιο πλαίσιο από APIs που έχουν οι βασικές εφαρμογές του Android. Η αρχιτεκτονική έχει διαμορφωθεί με τέτοιο τρόπο ώστε κάθε εφαρμογή να μπορεί να χρησιμοποιήσει τις δυνατότητες μιας άλλης και επίσης με τέτοιο τρόπο ώστε να δίνει τη δυνατότητα στον χρήστη να αλλάξει τα συστατικά κάθε εφαρμογής. Κάτω από το πλαίσιο των εφαρμογών υπάρχει ένα σύστημα από υπηρεσίες και συστήματα τα οποία περιλαμβάνουν:
    - Ένα διαχειριστή πόρων (ResourceManager) για την πρόσβαση στους πόρους όπως layoutfiles, εικόνες,strings.
    - Ένα σύνολο από γραφικά στοιχεία (Views) για τη δημιουργία γραφικού περιβάλλοντος συμπεριλαμβανομένων πλεγμάτων (grids), λιστών (lists), κουμπιών (buttons) κουτιών κειμένου (textboxes), κειμένου(textView) και άλλων.
    - Έναν διαχειριστή δραστηριοτήτων (Activity Manager),ο οποίος διαχειρίζεται τον κύκλο ζωής των εφαρμογών.

- Έναν διαχειριστή ειδοποιήσεων (Notification Manager),ο οποίος επιτρέπει την προβολή ειδοποιήσεων στην μπάρα κατάστασης (status bar).
- Ένα διαχειριστή περιεχομένου (Content Manager),ο οποίος επιτρέπει το διαμοιρασμό των δικών τους δεδομένων με άλλες εφαρμογές ή την πρόσβαση σε δεδομένα άλλων εφαρμογών .

- **Επίπεδο Εφαρμογών (Applications):** Το Android εξαρχής περιέχει ένα σύνολο από βασικές εφαρμογές που περιλαμβάνουν ένα πρόγραμμα για SMS μηνύματα, περιηγητή ιστού, email client, πρόγραμμα για δομημένη αποθήκευση των επαφών, χάρτες (Google Maps), ημερολόγιο και άλλα. Όλες οι εφαρμογές είναι γραμμένες στην γλώσσα προγραμματισμού Java.**[7]**



Εικόνα 14 : Κύκλος ζωής ενός activity



Εικόνα 15 : Μενού συσκευής με λειτουργικό Android

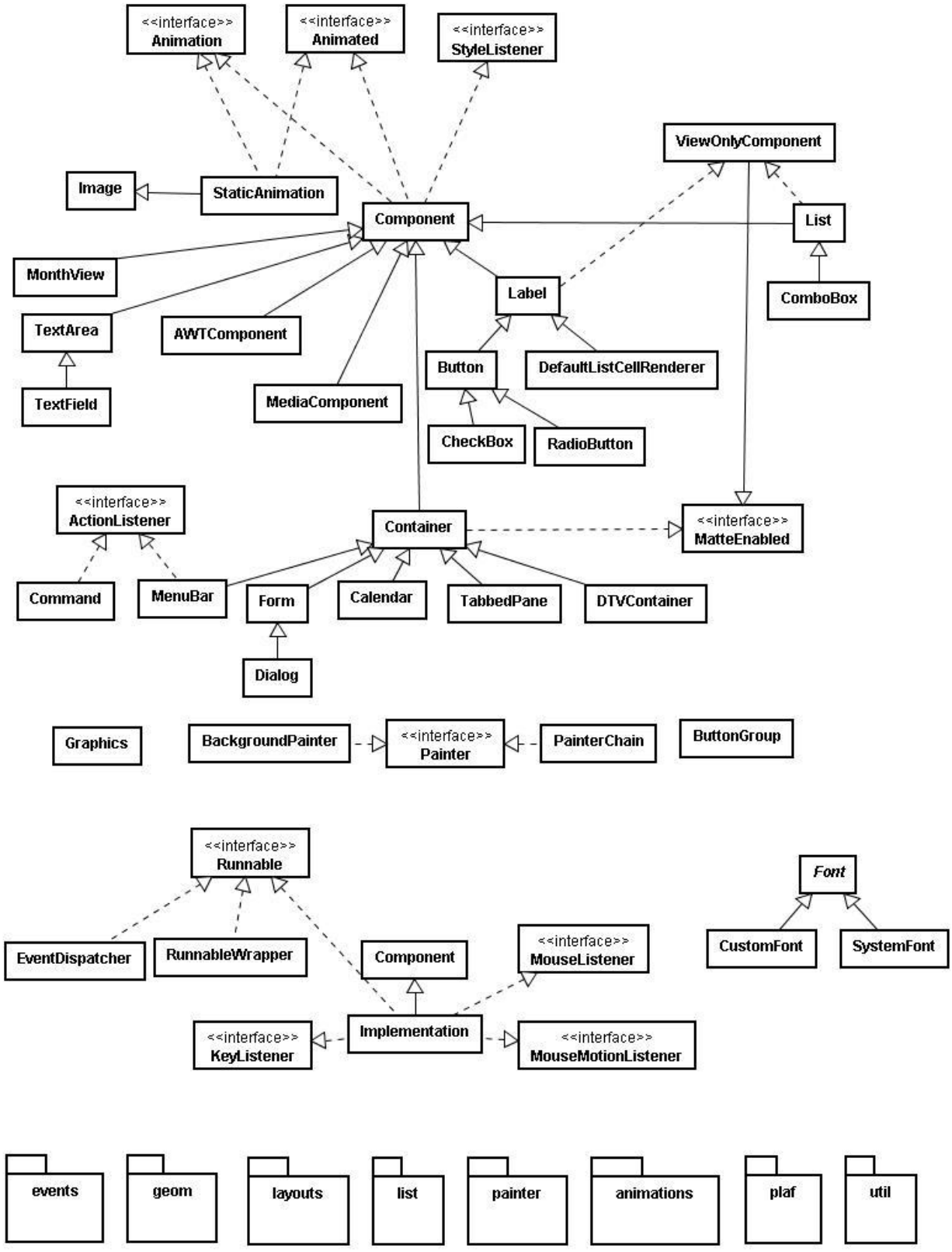
4. Ανάπτυξη Διεπαφών για Κινητές Συσκευές

Στο προηγούμενο κεφάλαιο μελετήσαμε τα λειτουργικά συστήματα. Σε αυτό το κεφάλαιο εστιάζουμε στην ανάπτυξη διεπαφών για κινητές συσκευές, για κάθε λειτουργικό σύστημα. 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-basedprogramming).

4.1. Εργαλειοθηκοκεντική προσέγγιση προγραμματισμού (Toolkit-based programming)

Οι περισσότερες πλατφόρμες που περιγράφηκαν στο προηγούμενο κεφάλαιο δίνουν απευθείας τη δυνατότητα ανάπτυξης γραφικών διεπαφών χωρίς την ανάγκη εισαγωγής πρόσθετων. Εκτός όμως της προσέγγισης που βασίζεται στον προγραμματισμό εργαλειοθήκης με καθιερωμένα διαδραστικά αντικείμενα υπάρχουν και εργαλεία που αναπτύσσονται από τρίτους με στόχο είτε την απόδοση νέων ενδεχομένως εξειδικευμένων διαδραστικών αντικειμένων, είτε βελτιωμένων εκδόσεων ήδη υποστηριζόμενων. Μερικά από τα αντιπροσωπευτικά εργαλεία της κατηγορίας αυτής είναι:

1. **LWUIT Toolkit :** Το LWUIT είναι μια εργαλειοθήκη που αφορά την ανάπτυξη διεπαφών για κινητές συσκευές που υποστηρίζουν είτε τη CDC είτε τη CLDC διαμόρφωση. Επίσης υποστηρίζει τη δυνατότητα εφαρμογής πλήθους υποστηριζόμενων θεμάτων (themes).



Εικόνα 16 : LWUIT toolkit class diagram

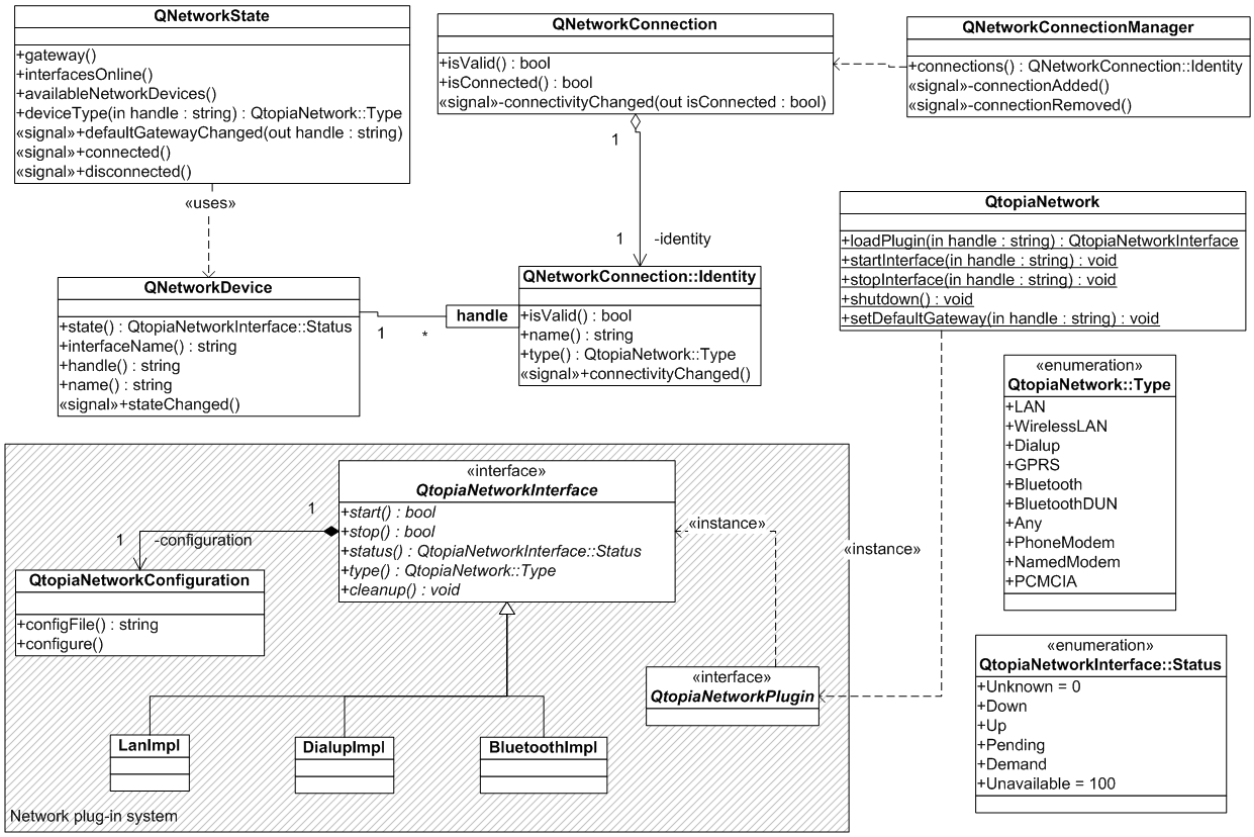
2. **Apime:** Το Apime είναι μια εργαλειοθήκη μέσω της οποία μπορούμε να δημιουργήσουμε γραφικές διεπαφές. Αφορά κινητές συσκευές που υποστηρίζουν Java (J2ME) και είναι συμβατό με την έκδοση MIDP 1.0 καθώς και το MIDP 2.0.Στην Εικόνα 17 **: Διεπαφές με χρήση του Apime** βλέπουμε πως εμφανίζεται μια γραφική διεπαφή που δημιουργήθηκε με τη χρήση της εργαλειοθήκης Apime.

Εικόνα 17 : Διεπαφές με χρήση του Apime

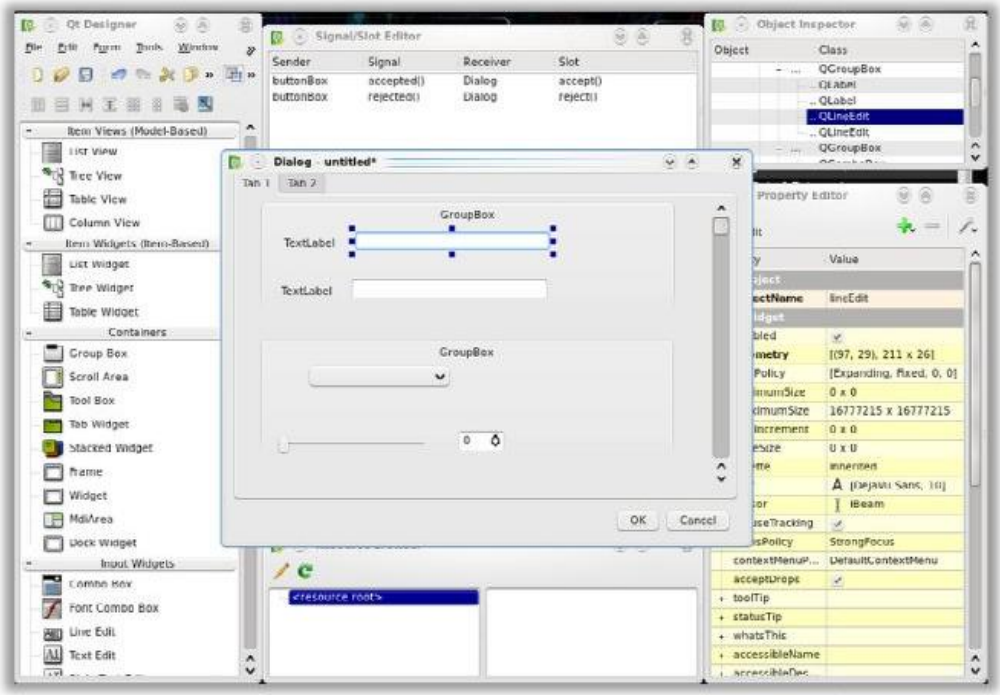
3. **jMobileCore :** Το jMobileCore είναι ένα ισχυρό εργαλείο για την δημιουργία εφαρμογών J2ME.Η συγκεκριμένη εργαλειοθήκη λειτούργει σε οποιαδήποτε κινητή ή PDA συσκευή που υποστηρίζει J2ME με MIDP 1.0 και CLDC 1.0.

Εικόνα 18 : Παράδειγμα διεπαφής jMobileCore

4. **QT Toolkit:** Κατασκευάστηκε από την Nokia και χρησιμοποιείται από μεγάλες εταιρίες σε διάφορα προϊόντα όπως Google Earth, Skype, Adobe Photoshop Album κ.α.Το Qt είναι βασισμένο στην γλωσσά προγραμματισμού C++. Τρέχει σε όλες τις σημαντικές πλατφόρμες ενώ εκτός των γραφικών δυνατοτήτων της, παρέχει και δυνατότητα πρόσβασης σε βάσεις δεδομένων XML parsing, υποστήριξη δικτυακής επικοινωνίας καθώς και ένα cross-platform API για των χειρισμό αρχείων.**[9]**

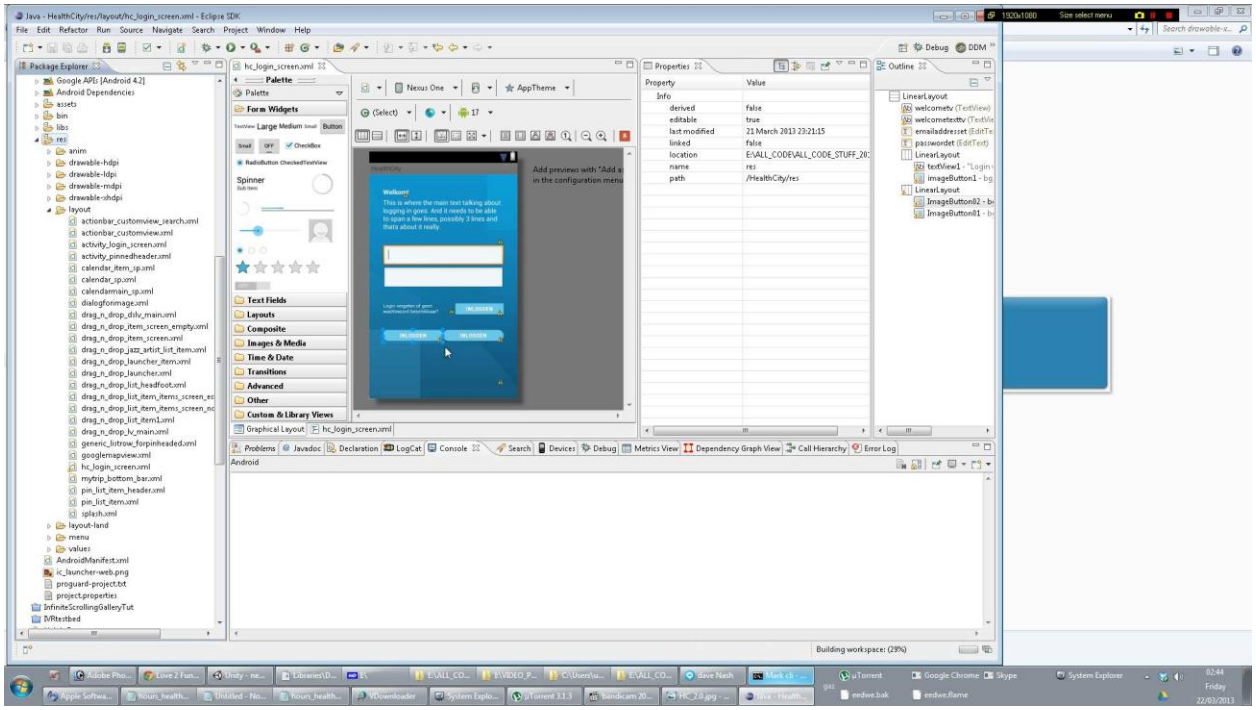


Εικόνα 19 : QT toolkit class diagram



Εικόνα 20 : ΠεριβάλλονQT

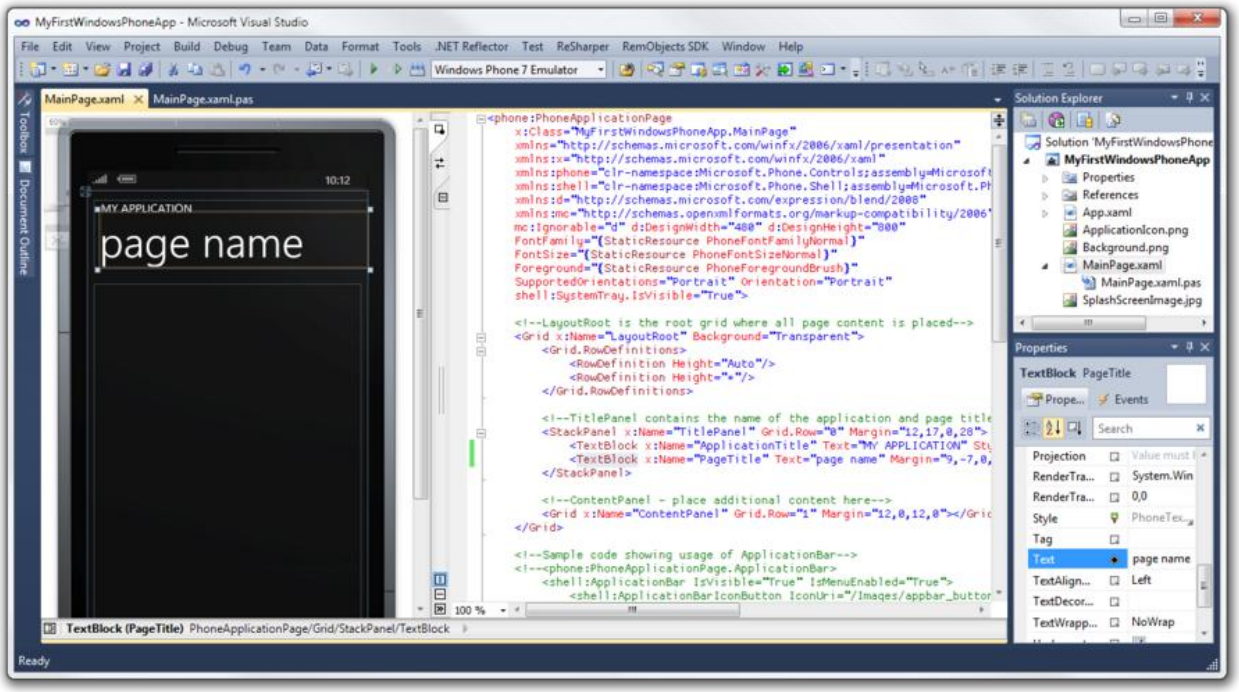
5. **Android user interface:** Όλα τα στοιχεία μιας διεπαφής χρήστη σε μία εφαρμογή για Android φτιάχνονται χρησιμοποιώντας τα αντικείμενα View και ViewGroup. Ένα View είναι ένα αντικείμενο το οποίο σχεδιάζει κάτι στην οθόνη με το οποίο μπορεί να αλληλοεπιδράσει ο χρήστης. Ένα ViewGroup είναι ένα αντικείμενο το οποίο κρατάει άλλα View (και ViewGroup) αντικείμενα προκειμένου να ορίσει την διάταξη μιας διεπαφής χρήστη. Το Android παρέχει μία συλλογή από υποκλάσεις των κλάσεων View και ViewGroup οι οποίες προσφέρουν συνηθισμένα στοιχεία ελέγχου εισόδου (input controls), όπως κουμπιά και πεδία κειμένου, και διάφορους τύπους διατάξεων, όπως γραμμικές (linearLayouts) ή σχετικές διατάξεις (relative Layouts).



Εικόνα 21 : Δημιουργία android user interface

6. **Metro user interface:** Η λέξη Metro χρησιμοποιείται ως κωδική ονομασία για την τεχνική σχεδίασης που χρησιμοποιείται στα Windows Phone. Το στυλ αυτό είναι εμπνευσμένο από τις πινακίδες και τις οδηγίες που βλέπουμε στην καθημερινή μας ζωή, που έχουν στόχο να μας δώσουν ξεκάθαρα την πληροφορία, χωρίς περιττά γραφικά, πλαίσια κ.ά. Μία από τις βασικές αρχές του Metro η τυπογραφία, εκτός από την ομορφιά που δίνει στις εφαρμογές, μπορεί να συμβάλλει σημαντικά και στην χρηστικότητά τους. Το σωστό μέγεθος της γραμματοσειράς και η σωστή τοποθέτηση των γραμμάτων με τις αποστάσεις μπορεί να δημιουργήσει οπτικά μια ιεραρχία στο κείμενο που βλέπουμε, και να βοηθήσει τον χρήστη να περιηγηθεί. Η κίνηση, άλλη μια βασική αρχή, είναι αυτή που κάνει πιο «ζωντανό» το user interface. Το Metro είναι σχεδιασμένο για να

ανταποκρίνεται στις κινήσεις του χεριού (gesture-oriented) και σε συνδυασμό με την πίεση ή το «swipe», προσφέρουν μια μορφή φυσικής κίνησης στο user interface. Αυτό έχει ως στόχο να δώσει στον χρήστη την αίσθηση ότι το user interface ανταποκρίνεται πλήρως στις κινήσεις του, και έχει ένα «βάθος». Το περιεχόμενο και όχι το περίγραμμα είναι μια από τις σημαντικότερες αρχές στο Metro. Σε αντίθεση με τα user interfaces που περιέχουν πολύπλοκα γραφικά, το metro βασίζεται στο πραγματικό περιεχόμενο, την πληροφορία, η οποία είναι πλέον το βασικό user interface. Βλέπουμε μεγάλα hubs αντί για μικρά κουμπιά. Οι τίτλοι στις σελίδες είναι συνήθως μεγάλοι. Αν αφαιρέσουμε όλα τα επιπλέον στοιχεία στο user interface (παράθυρα, πλαίσια, πολύπλοκα μενού κτλ) το περιεχόμενο συγκεντρώνει όλη την προσοχή του χρήστη. Αυτό είναι ιδιαίτερα χρήσιμο στα κινητά, λόγω του μικού μεγέθους της οθόνης. Θα πρέπει η εφαρμογή να είναι σχεδιασμένη αυθεντικά για την συσκευή, με βάση τις δυνατότητές της, την touch οθόνη, και τους τρόπους αλληλεπίδρασης με τον χρήστη που παρέχει. Δεν θα πρέπει να είναι μία απλή μεταφορά ενός desktop UI σε UI για κινητά. Να είναι προσαρμοσμένο το UI στην συσκευή, και όχι η συσκευή στο UI.



Εικόνα 22 : Δημιουργία metro user interface

4.2. Model-Based UI Engineering

Η μηχανική ανάπτυξης διεπαφών που βασίζονται σε μοντέλα μια εναλλακτική προσέγγιση κατασκευής διεπαφών που επιδιώκει:

- να διευκολύνει την ανάπτυξη ενός συστήματος σε τυχών μελλοντικές απαιτήσεις και
- να στοχεύει στη δημιουργία διεπαφών για πολλά περιβάλλοντα και λειτουργικά συστήματα.

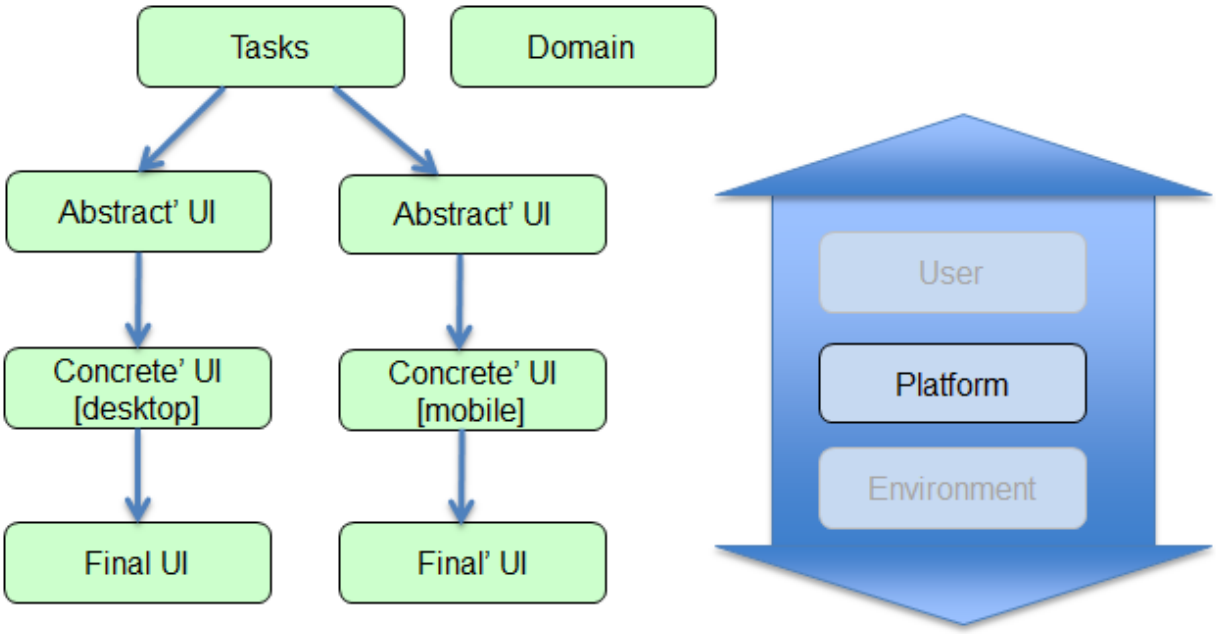
Η ανάπτυξη γίνεται με χρήση υψηλού επιπέδου μοντέλα τα οποία δρουν ως επίπεδα αφαίρεσης και επιτρέπουν στους σχεδιαστές να προσδιορίζουν διαφορετικές κατασκευαστικές συνιστώσες στο κύκλο ζωής ενός συστήματος. Έτσι επιχειρείται διαχωρισμός του εύρους των θεμάτων που αφορούν την κατασκευή και την υλοποίηση διαδραστικών συστημάτων. Ανάλογα με το τρέχον επίπεδο αφαίρεσης ασχολούμαστε με υψηλού επιπέδου θέματα και όχι με χαμηλού επιπέδου λεπτομέρειες της εφαρμογής.**[9]**Τα θέματα αυτά είναι :

•**Μοντέλο διεργασίας και αντικειμένου(Task and object model):** Σε αυτό το επίπεδο περιγράφονται οι λογικές δραστηριότητες εκτελούνται από τον χρήστη για να πετύχει τον στόχο του. Συνήθως αναπαρίστανται από ιεραρχικές ενδείξεις, που εμφανίζονται με ποια σειρά ο χρήστης θα εκτελέσει τις διεργασίες για να φτάσει στον στόχο του.

•**Μοντέλο αφηρημένης γραφικής διεπαφής(Abstrac User Interface):**Στο συγκεκριμένο μοντέλο εστιάζουμε στην υποστήριξη με γραφική διεπαφή της διεργασίας η οποία πρέπει να εκτελεστεί από τον χρήστη. Μόνο η λογική δομή καθορίζεται σε αυτό το επίπεδο, και με αυτό τον τρόπο, είναι ανεξάρτητος ο τρόπος διάδρασης του χρήστη από την πλατφόρμα.

•**Μοντέλο συγκεκριμένης γραφικής διεπαφής(Concrete User Interface):**Η αφηρημένη γραφική διεπαφή, αλλάζει σε συγκεκριμένη γραφική διεπαφή. Η αλλαγή γίνετε με βάση την πλατφόρμα, όπου θα δημιουργηθεί, και με συγκεκριμένες ιδιότητες όπου θα προσδιορίσουν πώς θα γίνει αντιληπτή από τον χρήστη.

• **Μοντέλο τελικής γραφικής διεπαφής(Final User Interface):** Η συγκεκριμένη γραφική διεπαφή μεταφράζεται σε μια γραφική διεπαφή που υλοποιείται από μια συγκεκριμένη γλώσσα προγραμματισμού.

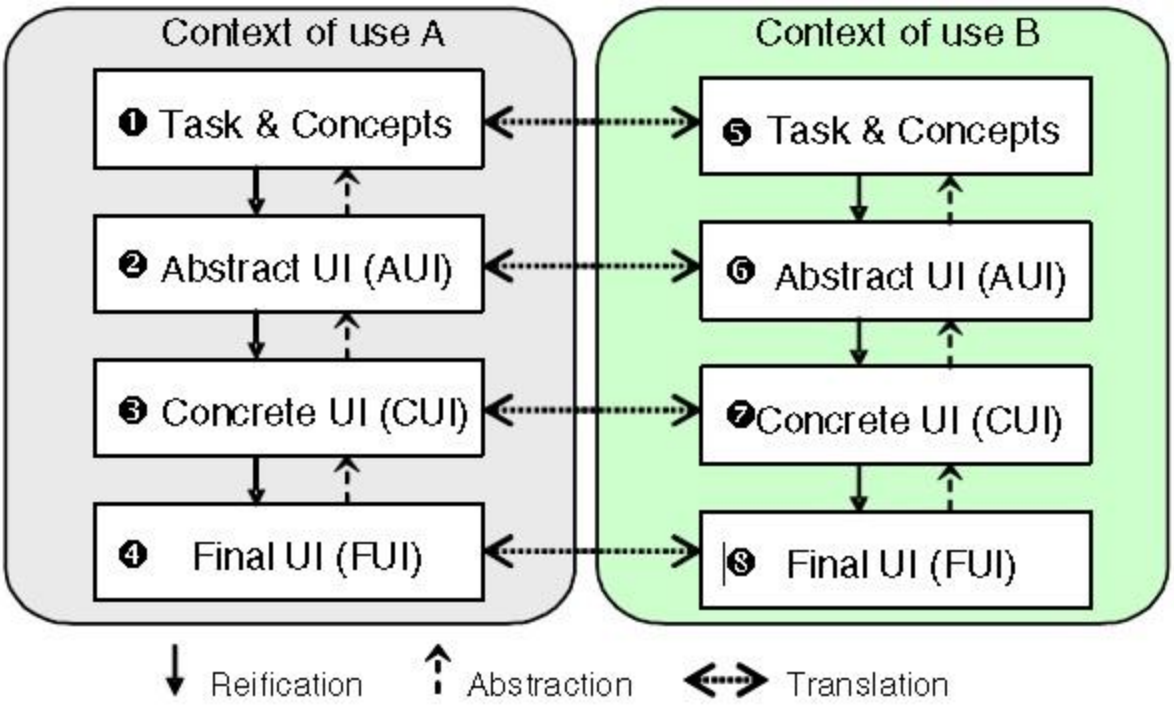


Εικόνα 23 : Επίπεδα αφαίρεσης

Οι επιμέρους λεπτομέρειες κάθε μοντέλου και ο τρόπος που αξιοποιείται για τη σχεδίαση διεπαφών εξαρτάται από το εκάστοτε σύστημα ή/και προσέγγιση.

1. **UIML :** 'Είναι μια γλώσσα βασισμένη σε XML για την περιγραφή, διεπαφών σε πολλαπλές πλατφόρμες. Αναπτύχθηκε από την Harmonia and Virginia Tech. Δεν υποστηρίζει την έννοια του καθήκοντος αλλά στοχεύει κυρίως στην αφηρημένη δομή των διεπαφών.
2. **TERESA :** Η TERESA είναι μια γλώσσα βασισμένη στο XML.Υποστηρίζει μετασχηματισμούς για την μετατροπή του ενός μοντέλου, σε ένα άλλο. Τα μοντέλα είναι το μοντέλο αφηρημένης διεπαφής, το μοντέλο καθηκόντων και το μοντέλο συγκεκριμένης διεπαφής. Για κάθε πλατφόρμα δημιουργεί ξεχωριστό μοντέλο αφηρημένης διεπαφής, μοντέλο καθηκόντων και συγκεκριμένης διεπαφής αντίστοιχα. Η διαφορά της είναι με την UsiXML ότι δεν λαμβάνει υπόψη τα στερεότυπα του χρήστη, το περιβάλλον και δημιουργεί ένα ξεχωριστό taskmodel για κάθε πλατφόρμα ξεχωριστά.
3. **UsiXML :** HUsiXML είναι ίσως από τις δημοφιλέστερες αν όχι η πιο δημοφιλής γλώσσα περιγραφής διεπαφών (UIDL). Η γλώσσα συνοδεύεται από πλήθος εργαλείων πολλαπλών που αποσκοπούν στη διευθέτηση διαφορετικών ζητημάτων κατά τη φάση ανάπτυξης μιας διεπαφής. Η προτεινόμενη μεθοδολογία είναι συνεπής με τις αρχές MDE (compliant), μεταφραζόμενο στα ανάλογα αφαιρετικά επίπεδα υπό τη μορφή μοντέλων. Ένα από τα πλεονεκτήματα της απομόνωσης, προσδιορισμού και υλοποίησης συγκεκριμένων ζητημάτων σε ξεχωριστές φάσεις της ανάπτυξης εστιάζεται στην προώθηση του επονομαζόμενου 'separation of concerns'. Ως περιβάλλον ορίζουμε την τριπλέτα πλατφόρμας, περιβάλλοντος και στερεότυπου του χρήστη(user stereotype).

Έτσι έχουμε τη δυνατότητα, μέσω της διαγραμματικής τεχνικής ιεραρχικής ανάλυσης καθηκόντων (Concur Task Trees), του προσδιορισμού του συνόλου απαιτήσεων του συστήματος από μεριάς χρηστών που είναι να εκτελεστούν ανεξαρτήτως περιβάλλοντος. Το συγκεκριμένο μοντέλο στη συνέχεια μπορεί να μετασχηματιστεί με τρόπο κατάλληλο έτσι ώστε να προσδιορίσουμε με πιο συγκεκριμένο τρόπο τη διεπαφή. Συγκεκριμένα στο επόμενο στάδιο συγκεκριμενοποίησης (reification) δίνεται η δυνατότητα περιγραφής των καθηκόντων, που περιγράφηκαν στο προηγούμενο στάδιο, υπό τη μορφή ωστόσο αφηρημένων υποδοχέων(containers) και διαδραστικών αντικειμένων με τρόπο ωστόσο που είναι ανεξαρτήτου καναλιού (modality independent) και πλατφόρμας(platform independent).Η UsiXML προτείνει ως μεθοδολογία ανάπτυξης διεπαφών αυτή που προτείνεται από το Camaleon Reference Framework, ενώ παράλληλα υποστηρίζει τη λεγόμενη 'πολυμονοπατική ανάπτυξη'(Multipath development). Αυτό σημαίνει ότι δίνεται η δυνατότητα στο χρήστη να ξεκινήσει την ανάπτυξη/περιγραφή/μηχανική μιας διεπαφής από οποιοδήποτε υποστηριζόμενο επίπεδο αφαίρεσης επιθυμεί. Αυτό δίνει ιδιαίτερη ευελιξία στη γλώσσα και στους μηχανικούς ανάπτυξης της διεπαφής.
Τέλος αξίζει να σημειωθεί ότι η UsiXML κάνει και χρήση επιπλέον μοντέλων, συμπληρωματικών, ώστε να δώσει τη δυνατότητα μοντελοποίησης-περιγραφής των υποστηριζόμενων περιβαλλόντων, όπως επίσης και της μοντελοποίησης οντοτήτων πεδίου (domain object modeling). Όσον αφορά το τελευταίο χρησιμοποιούνται διαγράμματα κλάσεων τα οποία συσχετίζονται με τη διεπαφή αλληλεπιδρώντας με το backend της εφαρμογής ώστε να προσδώσουν ουσιαστική λειτουργική αξία στη διεπαφή.
Στο παρακάτω σχήμα απεικονίζονται τα προτεινόμενα μοντέλα, που προσδιορίζουν-υλοποιούν τις αρχές του MDE, που χρησιμοποιεί η UsiXML ως υλοποίηση του Camelon Reference Framework.’ **[10]**



Εικόνα 24 : Cameleonreferenceframework

5. Προσέγγιση

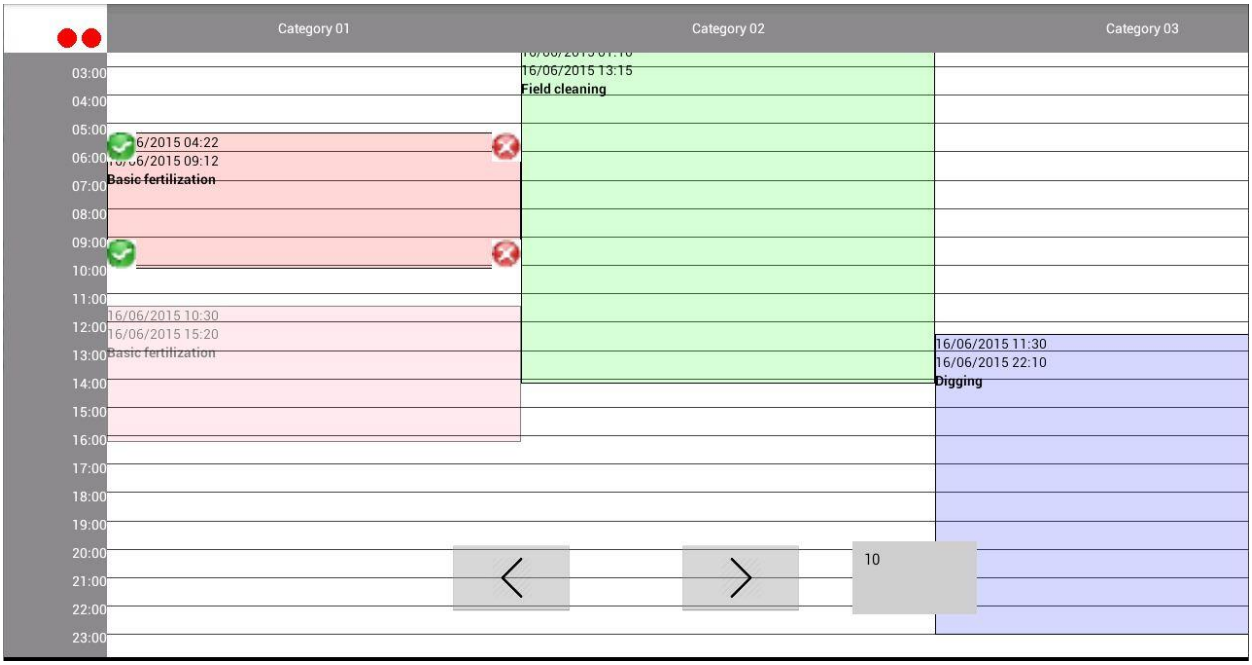
Το πεδίο αναφοράς της εν λόγω εργασίας αφορά τη συνεργατική διαμόρφωση πλάνου καλλιεργητικών εργασιών (collaborative crop profile compilation), για την οργάνωση της παραγωγής βιολογικών προϊόντων. Αυτό υλοποιείται, από πλευράς πρακτικής, διαμέσου της χωροχρονικής αποτύπωσης και συνεργατικής διαπραγμάτευσης των τύπων και της χρονικής έκτασης των καλλιεργητικών δραστηριοτήτων που είναι να λάβουν χώρα. Οι ρόλοι που προβλέπονται είναι αυτός του παραγωγού και του γεωπόνου ο οποίος επεμβατικά προτείνει διαφοροποιήσεις του αρχικού πλάνου που μπορεί να έχει συντάξει ο γεωργός είτε και ο ίδιος. Τέτοιου είδους αλλαγές μπορεί να προταθούν και υλοποιηθούν ακόμη και κατά τη διάρκεια εφαρμογής ενός συγκεκριμένου πλάνου εργασίας λόγω μη προβλεπόμενων γεγονότων, όπως καιρικών συνθηκών, κάποιου τύπου ασθένειας, κοκ.

5.1. Υπολογιστική Υλοποίηση Γλωσσών πεδίου Αντικειμένων

Για την υπολογιστική υποστήριξη αυτών των καθηκόντων χρειάστηκε να αναπτυχθούν εξειδικευμένες γλώσσες πεδίου. Δεδομένου ότι συνιστούν εξιδεικευμένα διαδραστικά αντικείμενα, δεν υποστηρίζονται εγγενώς από τις ευρέως διαθέσιμες διαδραστικές εργαλειοθήκες των γλωσσών προγραμματισμού και ως εκ τούτου για την ανάπτυξή τους απαιτείται χαμηλού επιπέδου εντατικός προγραμματισμός.

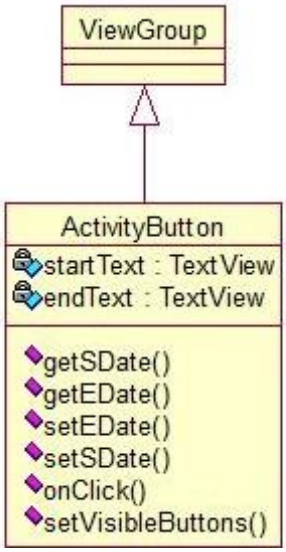
Διεπαφή Γεωπόνου

Για τη διεπαφή του γεωπόνου υλοποιήθηκε η αναπαράσταση της Εικόνα 25.



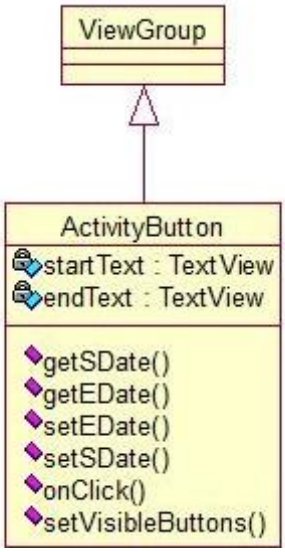
Εικόνα 25: Στιγμιότυπο διεπαφής Γεωπόνου

Η εν λόγω αναπαράσταση διαχωρίζει - ταξιθετεί τις δραστηριότητες με βάση τον τύπο (κατηγορία) τους. Αυτό είναι βολικό για το γεωπόνο μιας και μπορεί να έχει ευκολότερη κατανόηση της ροής δραστηριοτήτων βάση της φύσης τους. Για τη υπολογιστική του υλοποίηση επεκτάθηκε η διαδραστική εργαλειοθήκη του Android όπως φαίνεται παρακάτω.



Εικόνα 26: ActivityButton classDiagram

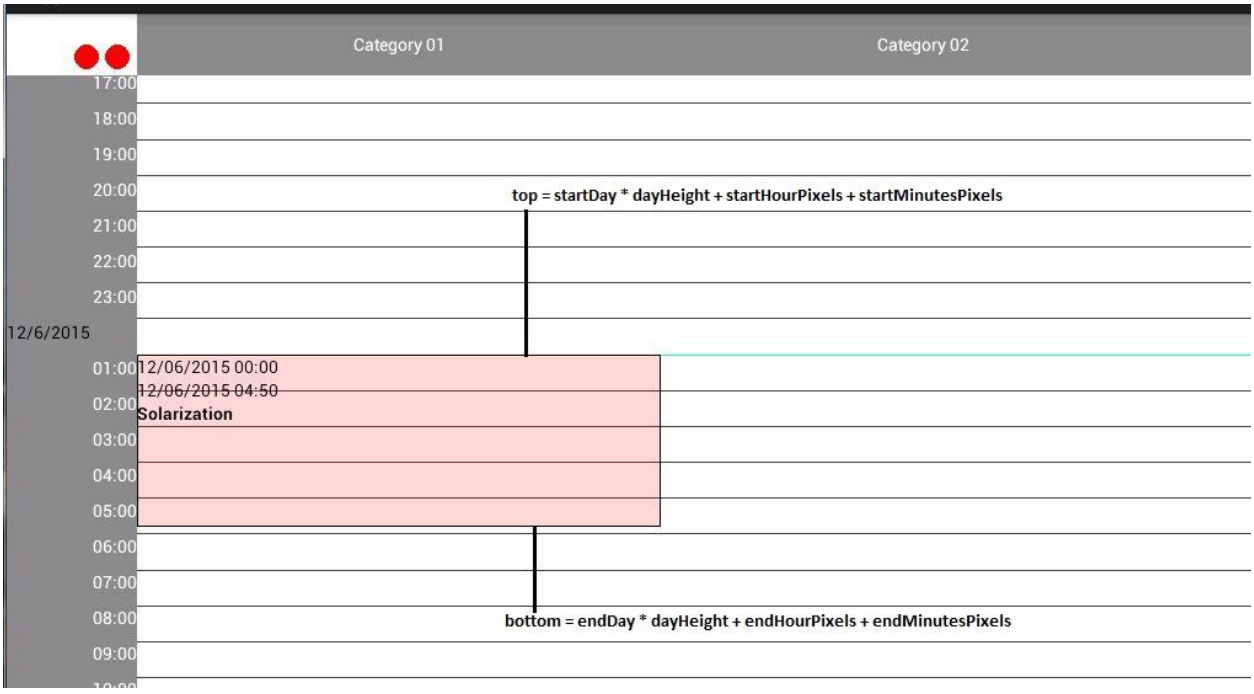
Στην



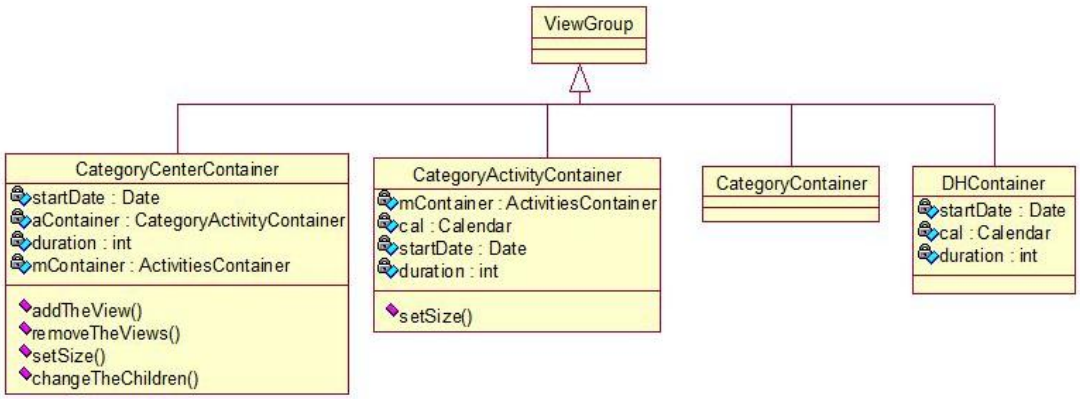
Εικόνα 26 φαίνεται η επέκταση της εργαλειοθήκης για την υποστήριξη του διαδραστικού αντικειμένου της δραστηριότητας. Δεδομένου ότι αυτό μπορεί να φέρει – υποστηρίξει διαβούλευση έπρεπε να δημιουργηθεί ως υποκλάση της ViewGroup, δηλαδή να είναι τύπου container. Συγκεκριμένα για τη συνεργατική διαπραγμάτευση της θέσης μιας δραστηριότητας το αντικείμενο πρέπει να φέρει συγκεκριμένους ‘δείκτες’ (controls) για την αποδοχή ή απόρριψή

της από το εταίρο συνεργαζόμενο μέλος. Αυτό υλοποιήθηκε μέσω των διαδραστικών στοιχείων αποδοχής και απόρριψης όπως φαίνονται στην Εικόνα 25. Αν το εν λόγω αντικείμενο δεν είχε υλοποιηθεί ως τύπου υποδοχέα (ViewGroup), το προηγούμενο δεν θα ήταν δυνατόν.

Για την υλοποίηση του υποδοχέα των δραστηριοτήτων (ActivityButtons), χρειάστηκε να επεκταθεί η εργαλειοθήκη του Android όπως φαίνεται στην Εικόνα 28 **: Category view class DiagramΣφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.**. Συγκεκριμένα, δημιουργήθηκαν τέσσερεις υποκλάσεις της ViewGroup. Η κλάση που φιλοξενεί τα διαδραστικά αντικείμενα είναι η CategoryActivityContainer και ο τρόπος ο οποίος τοποθετείται η κάθε δραστηριότητα (ActivityButton) φαίνεται στην παρακάτω Εικόνα 27 **:** . Το dayHeight είναι ο αριθμός των pixels που χρησιμοποιούμε για μια ημέρα (24 ώρες). Έπειτα κάθε λεπτό της ώρας αντιστοιχεί σε ένα pixel. Έχουμε μια μέρα σταθερή και την χρησιμοποιούμε σαν δείκτη. Οπότε στο startDay και endDay υπολογίζουμε πόσες μέρες πριν ή μετά αρχίζει και τελειώνει αντίστοιχα η εργασία μας. Στις μεταβλητές startHourPixels και endHourPixels λογαριάζουμε τις ώρες επί 60 (ένα pixel για κάθε λεπτό). Τέλος οι startMinutesPixels και endMinutesPixels περιέχουν τα λεπτά που αρχίζει και τελειώνει η εργασία μας , συνεπώς pixels. Δηλαδή το σημείο που τοποθετείται ένα στοιχείο πάνω σε ένα CategoryActivityContainer εξαρτάται από το πότε ξεκινάει και πότε τελειώνει η συγκεκριμένη εργασία.



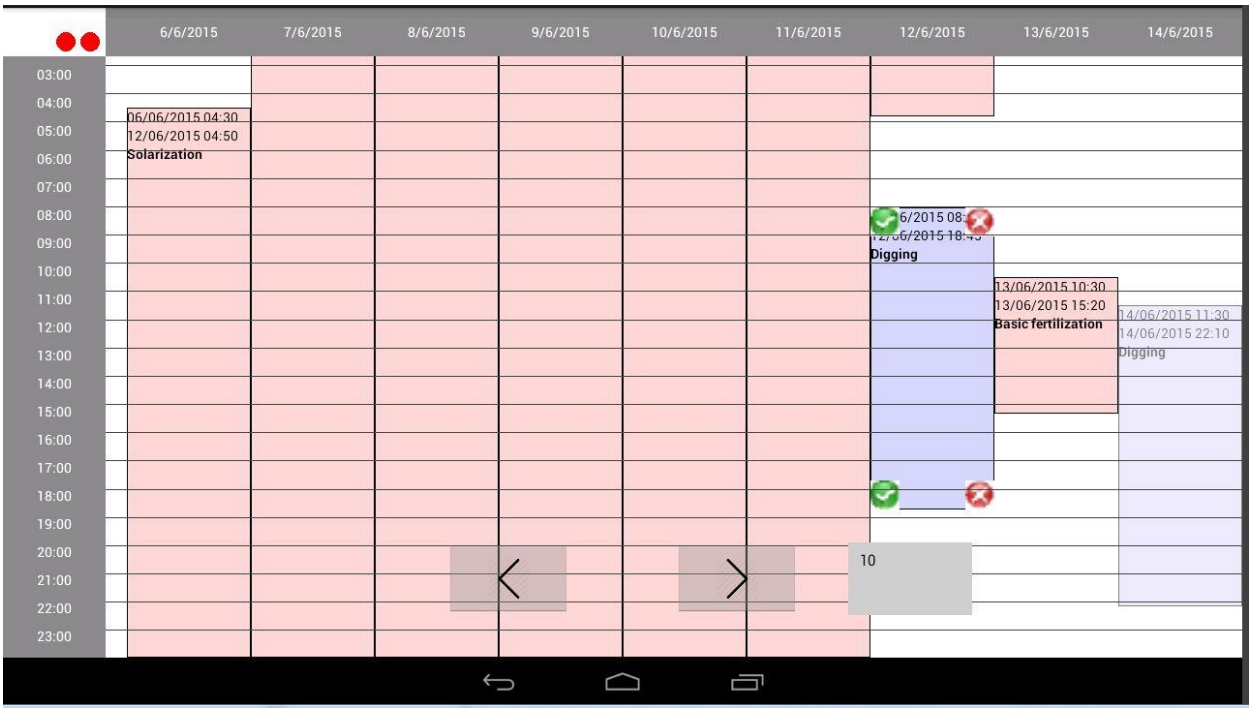
Εικόνα 27 : Διαχειριστής διάταξης (πως ταξιθετείται χρονικά με βάση το χρόνο).



Εικόνα 28 : Category view class Diagram

Διεπαφή Παραγωγού

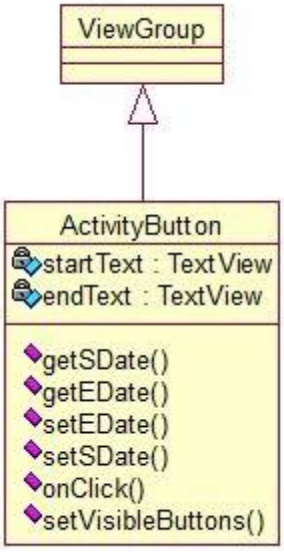
Για τη διεπαφή του παραγωγού υλοποιήθηκε η αναπαράσταση της Εικόνα 29**: Στιγμιότυπο διεπαφής Παραγωγού**. Η συγκεκριμένη αναπαράσταση τοποθετεί τις δραστηριότητες δίνοντας κύρια έμφαση στη χρονική διάσταση των εμπεριεχόμενων δραστηριοτήτων.



Εικόνα 29: Στιγμιότυπο διεπαφής Παραγωγού

Αυτό είναι πρακτικό για τον παραγωγό γιατί μπορεί να δει συγκεντρωτικά τις εργασίες που είναι να εκτελεστούν σειριακά σε εύρος ημερών. Και σ' αυτή την αναπαράσταση διακρίνεται η κατηγορία κάθε δραστηριότητας βάση της χρωματικής εναλλαγής των στιγμιοτύπων της

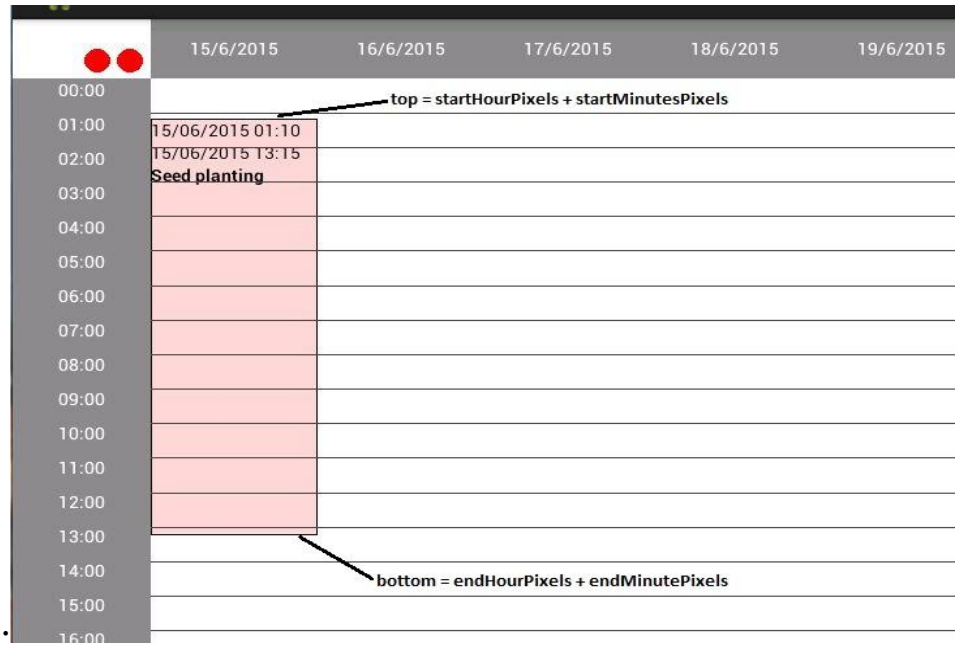
ActivityButton. Και εδώ επεκτάθηκε η διαδραστική εργαλειοθήκη του Android όπως φαίνεται παρακάτω.



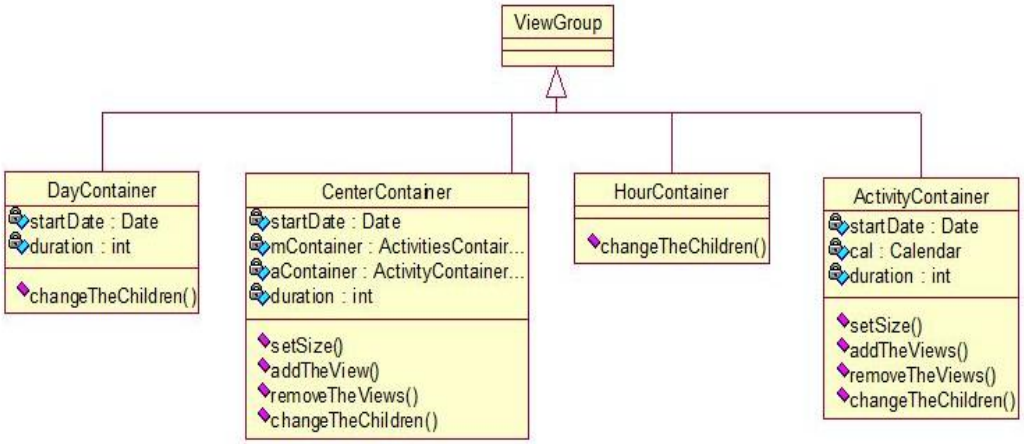
Εικόνα 30 : ActivityButton classDiagram

Όπως και στο Category view έτσι κι εδώ, για τους ίδιους λόγους, έγινε η επέκταση της εργαλειοθήκης για την υποστήριξη του διαδραστικού αντικειμένου της δραστηριότητας (Εικόνα 30 **: ActivityButton classDiagram**).

Για την υλοποίηση του υποδοχέα των δραστηριοτήτων, χρειάστηκε να επεκταθεί η εργαλειοθήκη του Android όπως φαίνεται στην Εικόνα 32 **: Day view class Diagram**. Η κλάση που τοποθετούνται τα διαδραστικά αντικείμενα είναι η ActivityContainer. Κι εδώ ο τρόπος τοποθέτησης των δραστηριοτήτων είναι ανάλογος με τον προηγούμενο (Εικόνα 31 **: Διαχειριστής διάταξης (πως ταξιθετείται χρονικά με βάση το χρόνο).**). Η πάνω μεριά υπολογίζεται βάση της ώρας εκκίνησης της εργασίας (κάθε ώρα = 60 pixels και κάθε λεπτό = 1 pixel) και η κάτω βάση της ώρας λήξης. Δηλαδή το σημείο που τοποθετείται ένα στοιχείο πάνω σ' ένα ActivityContainer εξαρτάται από το πότε ξεκινάει και πότε τελειώνει η κάθε εργασία. Η μόνη διαφορά είναι ότι εδώ βάζουμε για κάθε ημέρα και διαφορετικό ActivityContainer ενώ στην διεπαφή του Γεωπόνου βάζουμε διαφορετικό container σε κάθε κατηγορία.



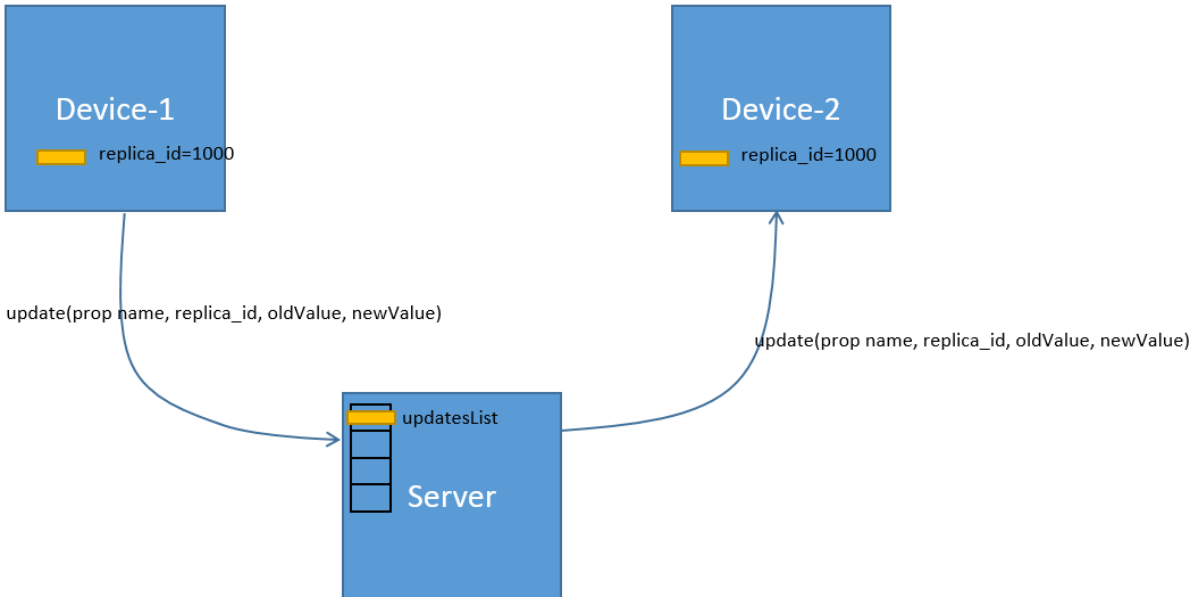
Εικόνα 31 : Διαχειριστής διάταξης (πως ταξιθετείται χρονικά με βάση το χρόνο).



Εικόνα 32 : Day view class Diagram

5.2. Υποστήριξη Κατανεμημένου συγχρονισμού

Η υποστήριξη συγχρονισμού γίνεται διαμέσου του διαμοιρασμού του μοντέλου των παραπάνω αντικειμένων. Συγκεκριμένα, κάθε φορά που καταχωρείται μια αλλαγή σε κάποια από τις κοινές ιδιότητες των μοντέλων, αυτή προκαλεί στα πλαίσια κλήσης της σχετικής μεθόδου τροποποίησης της σχετικής ιδιότητας, την προώθηση ενός κωδικοποιημένου μηνύματος σε ένα οικείο διακομιστή ο οποίος συλλέγει όλα τα γεγονότα, τα οποία προωθεί στους υπόλοιπους συνδεδεμένους χρήστες. Με τη λήψη μιας ειδοποίησης, ως αποτέλεσμα της αλληλεπίδρασης ενός χρήστη με την τοπική γλώσσα πεδίου, αυτή προωθείται σε ένα ενδεδειγμένο αντικείμενο που αναλαμβάνει την ενημέρωση του σχετικού τοπικού αντιγράφου με αποτέλεσμα και την ενημέρωση της διεπαφής και τη διατήρηση συγχρονισμού (feedthrough). Η σχετική ροή φαίνεται



Εικόνα 33: Υψηλού επιπέδου αρχιτεκτονική

στην Εικόνα 33. Στην εικόνα αυτή φαίνεται πως κάθε ιδιότητα συσχετίζεται με ένα αναγνωριστικό αντιγράφου που χαρακτηρίζει κάθε στιγμιότυπο ιδιότητας στην μνήμη μοναδικά. Με το που αλλάζει αυτό στη συσκευή-1, η αλλαγή προωθείται μέσω του java networking API στον οικείο διακομιστή ο οποίος την κρατάει σε μία λίστα, για την υποστήριξη latecomers, και στη συνέχεια την προωθεί σε όλους τους συνδεδεμένους χρήστες (βλ. Device-2).

Με την παραπάνω αρχιτεκτονική μπορούν και συσχετίζονται οι εναλλακτικές αλληλεπιδραστικές αναπαραστάσεις.

6. Βιβλιογραφία

1. Εμμανουήλ Γ. Καραμανής, Πέτρος Φλώριος Ν. Μπάκαλος (2013) Διπλωματική Εργασία: Σχεδίαση και ανάπτυξη εφαρμογής προσωπικού προγραμματιστή δραστηριοτήτων σε πλατφόρμα iOS.
2. Βασίλης Κόμης, Νικόλαος Αβούρης, Χρήστος Κατσάνος Πανεπιστήμιο Πατρών: Συστήματα και Εργαλεία Υποστήριξης Συνεργασίας.
3. Λουμάνης Βασίλειος (2011) Πτυχιακή Εργασία: Υλοποίηση εφαρμογής informationaggregator για πληροφορίες σχετικές με το τμήμα ΤΠΤ σε πλατφόρμα Apple iOS.
4. Φίλιππος Σωτηριάδης τ. Απόστολου (2005) Διδακτορική Διατριβή: ΣΧΕΔΙΑΣΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΠΡΩΤΟΚΟΛΛΟΥ ΓΙΑ ΤΗΝ ΥΠΟΣΤΗΡΙΞΗ ΕΞΕΛΙΓΜΕΝΩΝ ΟΜΑΔΙΚΩΝ ΥΠΗΡΕΣΙΩΝ ΤΗΛΕΜΑΤΙΚΗΣ.
5. Δημήτρης Κωνσταντάκος (2013) Πτυχιακή Εργασία:ΑΣΦΑΛΕΙΣ ΣΥΝΑΛΛΑΓΕΣ ΣΕ ΠΕΡΙΒΑΛΛΟΝ WINDOWSPHONE.
6. Στυλιανός Βούκατα (2011) Πτυχιακή Εργασία: Το λειτουργικό σύστημα Android και η υλοποίηση εφαρμογής με το GoogleSDK.
7. Συμεών Χ. Κωνσταντινίδης (2012) Διπλωματική Εργασία: Ανάπτυξη και σχεδιασμός εφαρμογής σε λογισμικό Android.
8. Στάθης Σιδηρόπουλος Μεταπτυχιακή Εργασία : Συνεργατικά Συστήματα Εργασίας με τη βοήθεια του Υπολογιστή.
9. ΜιλτιάδηςΚονσολάκης (2011) Πτυχιακή Εργασία : Σύγχρονη Συγχρονισμένη Συνεργασία Υποστηριζόμενη από κινητές συσκευές.
10. Κωνσταντίνος Χαβιαράς Πτυχιακή Εργασία : Απόδοση κοινωνικών χαρακτηριστικών σε διαδραστικά αντικείμενα του διαδικτύου.
11. https://el.wikipedia.org/wiki/Android