



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Υπηρεσία video με επίγνωση πλαισίου
σε περιβάλλοντα κινητού υπολογισμού**

Γεωργία Ε. Κατσαμάνη

Επιβλέπων: Ευστάθιος Χατζηευθυμιάδης, Επίκουρος Καθηγητής ΕΚΠΑ

ΑΘΗΝΑ

ΜΑΪΟΣ 2009

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Υπηρεσία video με επίγνωση πλαισίου σε περιβάλλοντα κινητού υπολογισμού

Γεωργία Ε. Κατσαμάνη

A.M.: M918

ΕΠΙΒΛΕΠΩΝ:

Ευστάθιος Χατζηευθυμιάδης, Επίκουρος Καθηγητής ΕΚΠΑ

Μάιος 2009

ΠΕΡΙΛΗΨΗ

Το mobile video είναι μια σχετικά καινούργια εφαρμογή στο κομμάτι της κινητής τηλεφωνίας που συνεχώς κερδίζει έδαφος. Εντάσσεται μέσα στη γενικότερη προσπάθεια του κλάδου των τηλεπικοινωνιών να προσεγγίσουν περισσότερο καταναλωτικό κοινό, προσφέροντας συνεχώς νέες εναλλακτικές και επιλογές, οι οποίες μπορεί να είναι άλλοτε χρήσιμες και απαραίτητες για το χρήστη κι άλλοτε να χαρακτηρίζονται καθαρά ως κινήσεις εντυπωσιασμού. Ταυτόχρονα υπάρχει κι ένα άλλο σύνολο υπηρεσιών που κερδίζει έδαφος στην αγορά κυρίως λόγω της διευκόλυνσης που προσφέρει στα άτομα σε πολλούς τομείς της καθημερινότητας. Αυτές οι υπηρεσίες είναι οι λεγόμενες context-aware, των οποίων ο βασικός κορμός της εφαρμογής τους στηρίζεται στο γεγονός ότι ο χρήστης διευκολύνεται σε διάφορους τομείς και ποικίλες δραστηριότητες μέσω της εκμετάλλευσης πληροφοριών που συλλέγει από τον περιβάλλοντα χώρο. Στην εργασία αυτή λοιπόν θα υλοποιήσουμε μια εφαρμογή context-aware υπηρεσίας πάνω σε mobile video. Θα αναζητήσουμε μεθόδους κι εργαλεία ώστε να καταφέρουμε να αποδώσουμε εύχρηστες λειτουργίες video σε ένα κινητό τηλέφωνο. Ουσιαστικά θα υλοποιήσουμε την επεξεργασία video σε Mobile όπως την έχουμε γνωρίσει σε πιο τεχνολογικά ισχυρές και πολύπλοκες συσκευές όπως ο υπολογιστής. Στο κομμάτι της επεξεργασίας εμπεριέχονται λειτουργίες όπως παύση ήχου, εστίαση εικόνας αλλά και χρήση μεταδεδομένων. Η προσπάθεια αυτή θα προσπαθήσει επομένως να αποδώσει μεγαλύτερη χρηστικότητα στο κινητό τηλέφωνο που πλέον έχει καταστεί αναπόσπαστο κομμάτι της καθημερινότητάς μας.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Υπηρεσίες context-aware

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: υπηρεσία με επίγνωση πλαισίου, τεχνολογίες γνώσης, video κινητού, περιεχόμενο , κινητή τηλεφωνία

ABSTRACT

Mobile video it is a relatively new application in the domain of mobile telephony that continuously gains ground. It is included in the more general effort of telecommunications branch to approach more consuming public, offering continuously new alternative and choices that can be previously useful and essential for the user and previously be characterized clearly as movements of impressiveness. Another set of services also exists that gains ground in the market mainly because the facilitation that it offers in the individuals in a lot of sectors of everyday routine. These services are said "context-aware", of which the basic rule is the fact that the user is facilitated in various sectors and various activities via the exploitation of information that collects from environment space in the present. In this paper we are going to implement a context-aware application upon mobile video. We will try to find out methods and tools in order to implement useful video functionalities to a mobile phone. Substantially the scope of our paper is to implement the video processing domain to the mobile devices in the same way that this domain is used today in the more technologically powerful devices such as computers. The sound pausing, the focus of image and the usage of metadata are some of the functionalities that are included in the video processing area. To conclude, this effort will try to attributes the mobile devices with potential that they have at this time.

SUBJECT AREA: Context-aware services

KEYWORDS: context-aware service, knowledge technologies, mobile video, context , mobile telephony

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ.....	13
ΚΕΦΑΛΑΙΟ 1.....	14
ΕΙΣΑΓΩΓΗ	14
1.1 ΓΕΝΙΚΑ	14
1.2 ΟΡΓΑΝΩΣΗ	15
CONTEXT-AWARE ΥΠΗΡΕΣΙΕΣ	16
2.1 ΓΕΝΙΚΑ.....	16
2.2 ΛΟΓΟΙ ΔΗΜΙΟΥΡΓΙΑΣ ΚΑΙ ΠΡΟΒΛΗΜΑΤΑ	17
2.3 ΤΡΟΠΟΣ ΛΕΙΤΟΥΡΓΙΑΣ.....	19
2.3.1 Απευθείας Σύνδεση.....	20
2.3.2 Επικοινωνία μέσω server.....	21
2.4 ΧΡΗΣΗ ΤΕΧΝΟΛΟΓΙΩΝ ΓΝΩΣΗΣ ΓΙΑ ΤΗΝ ΑΝΑΠΤΥΞΗ CONTEXT-AWARE ΥΠΗΡΕΣΙΩΝ.....	23
2.4.1 Πιθανοί τρόποι υλοποίησης.....	23
2.4.2 Περιγραφή παρόμοιων συστημάτων	26
2.4.2.1 ECA –Jess Μοντέλο	26
2.4.2.2 Ανεπτυγμένο Context Management Μοντέλο.....	27
2.4.3 Τρόπος υλοποίησης που ακολουθήθηκε.....	28
2.5 ΠΑΡΑΔΕΙΓΜΑΤΑ CONTEXT-AWARE ΥΠΗΡΕΣΙΩΝ ΚΙ ΕΦΑΡΜΟΓΕΣ.....	30
2.5.1 Συσκευές πλοήγησης με την τεχνολογία GPS.....	30
2.5.2 Ταξιδιωτικός οδηγός αλληλεπιδραστικός με το περιβάλλον (Context-aware tour guide)	32
2.5.3 Αλληλεπιδραστικό κινητό πολυμέσων	33
MOBILE VIDEO.....	35
3.1 ΓΕΝΙΚΑ	35
3.2 MOBILE VIDEO	36
3.3 ΕΦΑΡΜΟΓΕΣ MOBILE VIDEO.....	38
3.3.1 M-learning.....	38
3.3.2 Επαγγελματική κατάρτιση.....	39
3.3.3 Διαφήμιση	40
3.3.4 Video κινητού τηλεφώνου για έκτακτες ανάγκες (Emergency mobile video).....	41
CONTEXT-AWARE MOBILE VIDEO	42
4.1 ΓΕΝΙΚΑ	42
4.2 ΑΠΑΙΤΗΣΕΙΣ.....	42
4.3 ΑΝΤΙΜΕΤΩΠΙΣΗ	43
4.3.1 Sensors.....	44
4.3.2 Εξυπηρετητές Περιεχομένου (Context-server)	48
4.3.3 Ερμηνευτές περιεχομένου (Context-interpreter).....	48
ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ.....	52
5.1 ΓΕΝΙΚΑ	52
5.2 ΣΕΝΑΡΙΑ ΧΡΗΣΗΣ	53
5.2.1 Λειτουργίες.....	53

5.2.2	<i>Αισθητήρες και καταστάσεις χρήστη</i>	54
5.2.3	<i>Μεταδεδομένα (metadata)</i>	57
5.3	ΑΡΧΙΤΕΚΤΟΝΙΚΗ.....	58
ΥΛΟΠΟΙΗΣΗ ΚΑΙ ΠΛΑΤΦΟΡΜΕΣ		62
6.1	JAVA2ME-NETBEANS	62
6.1.1	<i>MMAPI</i>	65
6.1.2	<i>JIP2ME</i>	72
6.1.3	<i>KXML</i>	73
6.2	JIPROLOG IDE.....	77
6.3	POLYSEMA MPEG-7 VIDEO ANNOTATOR.....	78
ΕΠΙΔΟΣΕΙΣ ΚΑΙ ΣΥΜΠΕΡΑΣΜΑΤΑ		79
7.1	ΕΠΙΔΟΣΕΙΣ.....	79
7.1.1	<i>Ποιότητα</i>	80
7.1.2	<i>Σωστή εφαρμογή κανόνων</i>	81
7.1.3	<i>Καθυστέρηση</i>	82
7.1.3.1	<i>Μετρήσεις</i>	83
7.1.3.2	<i>Αξιολόγηση μετρήσεων</i>	86
7.2	ΣΥΜΠΕΡΑΣΜΑΤΑ	88
7.2.1	<i>ΘΕΤΙΚΑ</i>	88
7.2.2	<i>ΠΡΟΒΛΗΜΑΤΑ</i>	89
ΠΑΡΑΡΤΗΜΑ Α΄		91
ΠΑΡΑΡΤΗΜΑ Β΄		93
ΟΡΟΛΟΓΙΑ		99
ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ		103

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 2-1 :Σταθμός αισθητήρων και μετάβαση πληροφοριών σε server[36].	22
Εικόνα 2-2 :Αντιστοίχιση ECA-DL σε Jess[39]	26
Εικόνα 2-3 :Αντιστοίχιση ECA-DL σε Jess[39]	27
Εικόνα 2-4 :Context-aware αρχιτεκτονική[40]	27
Εικόνα 2-5 :Εφαρμογή context-aware πλοήγησης[1]	31
Εικόνα 2-6 :Πειραματική διάταξη context-aware tour guide[2]	33
Εικόνα 2-7 :Εφαρμογή m-Kavaad[3]	34
Εικόνα 2-8 :Χρήση context-aware multimedia mobile [3]	34
Εικόνα 3-1 :Εφαρμογή mobile video[9]	37
Εικόνα 3-2 :Εφαρμογή m-learning [10]	39
Εικόνα 3-3 :Mobile διαφήμιση [11]	40
Εικόνα 3-4 :Emergency mobile video [12].....	41
Εικόνα 4-1 :Είδη ταχύμετρου [21]	47
Εικόνα 4-2 :Αισθητήρας απόστασης και προσανατολισμού [23].....	47
Εικόνα 4-3 :Παράδειγμα δομής context interpreter[17]	49
Εικόνα 4-4 :Κινητικότητα ενός χρήστη mobile video που συνδέεται συνεχώς με νέα δίκτυα και νέες πληροφορίες [14]	50
Εικόνα 4-5 :Συνεχής αλλαγή του περιεχομένου (content) και ανάγκη προσαρμογής του video [14].....	51
Εικόνα 6-1 :MMAPI στη δομή της javaME [25].....	64
Εικόνα 6-2 :States and methods of player [25]	67
Εικόνα 6-3 :Unrealized state [25]	68
Εικόνα 6-4 :Realized state [25]	68
Εικόνα 6-5 :Prefetched state [25].....	69
Εικόνα 6-6 :Started state [25].....	69
Εικόνα 6-7 :Datasource-Manager-Player [25]	70
Εικόνα 6-8 : MMAPi Controls [25].....	67
Εικόνα 6-9 JIProlog IDE [27].....	77
Εικόνα 6-10 Polysema Annotator.....	78
Εικόνα 7-1 Στιγμιότυπο video στο προσομοιωτή τηλεφώνου.....	80
Εικόνα 7-2 Λειτουργία αποκοπής.....	90

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 6-1 Κινητά που υποστηρίζουν MMAPI [25]	65
Πίνακας 6-2 Γεγονότα που αναγνωρίζει ο PlayerListener [25]	70
Πίνακας 7-1 Στατιστικά με sensor timer στα 10 s	83
Πίνακας 7-2 Mean delay με sensor timer στα 10 s	83
Πίνακας 7-3 Στατιστικά με sensor timer στα 5 s	84
Πίνακας 7-4 Mean delay με sensor timer στα 5 s	84
Πίνακας 7-5 Στατιστικά με sensor timer στα 12 s	85
Πίνακας 7-6 Mean delay με sensor timer στα 12 s	85
Πίνακας 7-7 Σύγκριση επί τις εκατό %	87

ΠΡΟΛΟΓΟΣ

Τα τελευταία χρόνια έχει παρατηρηθεί μια σημαντική διείσδυση των κινητών τηλεφώνων στην καθημερινότητά μας τα οποία πλέον χρησιμοποιούνται για ποικίλες λειτουργίες εκτός από την κλασική της τηλεφωνίας .Μεταξύ αυτών των νέων υπηρεσιών που προσφέρει το κινητό τηλέφωνο είναι η αναπαραγωγή κινούμενης εικόνας (video) , περιήγηση στο διαδίκτυο (internet) και παιχνίδια (games) . Η τεχνολογία συνεχώς προχωράει και οι εταιρείες αναζητούν τρόπους να επεκτείνουν τις υπηρεσίες αυτές και να προσφέρουν καινοτόμες προτάσεις ώστε να προσελκύσουν περισσότερο το καταναλωτικό κοινό. Στην παρούσα εργασία θα ασχοληθούμε με μία καινούργια τεχνολογία για την οποία γίνονται προσπάθειες να ενσωματωθεί στην ευρύτερη χρήση του κινητού τηλεφώνου.

Στο σημείο αυτό θα ήθελα να ευχαριστήσω ολόψυχα τον επιβλέποντα καθηγητή Ευστάθιο Χατζηευθυμιάδη που μου έδωσε τη δυνατότητα να έρθω σε επαφή με ένα πολύ ενδιαφέρον πεδίο έρευνας και που χωρίς τη συμβολή και την καθοδήγηση του οποίου αυτή η εργασία δε θα μπορούσε να ολοκληρωθεί καθώς επίσης και το Βασίλειο Τσέτσο για τη σημαντική υποστήριξη και βοήθεια που μου πρόσφερε. Τέλος θα ήθελα να απευθύνω ευχαριστίες στην οικογένειά μου, τους γονείς μου και την αδερφή μου που με στήριξαν στην προσπάθειά μου.

Αθήνα, Μάιος 2009

Γ. Κατσαμάνη

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

1.1 Γενικά

Τα τελευταία χρόνια αντιμετωπίζουμε μια ραγδαία ανάπτυξη των ασύρματων επικοινωνιών μεταξύ των οποίων συγκαταλέγονται και τα κινητά τηλέφωνα. Από τη στιγμή που καθιερώθηκαν στις σύγχρονες κοινωνίες, εξελίσσονται συνεχώς με αποτέλεσμα να αποτελούν έναν από τους πλέον προσοδοφόρους τομείς της παγκόσμιας οικονομίας σήμερα και να γίνονται έντονες προσπάθειες ως προς τη βελτίωση και τον εμπλουτισμό των υπάρχουσών λειτουργιών τους καθώς επίσης και τη δημιουργία καινούριων.

Έχοντας ξεκινήσει ως ένα πολυτελές μέσο επικοινωνίας δια της ομιλίας ,το κινητό τηλέφωνο αποτελεί σήμερα αναπόσπαστο κομμάτι της καθημερινότητάς μας και πολλές φορές καλείται να δώσει τη λύση σε άμεσα προβλήματα μέσω διάφορων λειτουργιών. Μέσα σε αυτές συγκαταλέγονται και οι πρόσφατα αναπτυσσόμενες context-aware υπηρεσίες, δηλαδή υπηρεσίες που προσφέρονται από διάφορα μέσα λαμβάνοντας υπόψη το περιεχόμενο της εκάστοτε εργασίας. Οι υπηρεσίες αυτές παρέχουν ευελιξία στο χρήστη καθώς δεν ακολουθούν μια συγκεκριμένη ακολουθία κάθε φορά αλλά προσαρμόζονται στις συνθήκες. Τυπικό παράδειγμά τους είναι οι υπηρεσίες θέσης (Location Based Services).[34]

Στην παρούσα εργασία θα παρουσιάσουμε μια εφαρμογή που συνδυάζει το κινητό τηλέφωνο με context-aware υπηρεσίες. Πιο συγκεκριμένα θα περιγράψουμε πως είναι δυνατόν το κινητό μας να έχει ευφυΐα ως προς τη λειτουργία του video που υποστηρίζει εδώ και χρόνια. Και με τον όρο ευφυΐα ουσιαστικά αναφερόμαστε στη δυνατότητα του κινητού, μέσω του μηχανισμού που δημιουργήσαμε, να ελέγχει τη ροή της πληροφορίας του video ώστε να είναι πιο χρηστικό στον άνθρωπο χωρίς να χρειάζεται η δική του παρέμβαση. Το κινητό αντιλαμβάνεται συνθήκες του περιβάλλοντος και καταστάσεις του ατόμου μέσω αισθητήρων και λειτουργεί με κάποιους κανόνες με στόχο την αποδοτικότερη παρουσίαση του video στο χρήστη.

1.2 Οργάνωση

Στην ενότητα αυτή θα παρουσιάσουμε συνοπτικά τη δομή της εργασίας μας και τον τρόπο οργάνωσής της σε κεφάλαια.

Στο κεφάλαιο 1, κάνουμε μια πολύ γενική περιγραφή για την ιδέα της εργασίας μας που δεν είναι άλλη από τη χρήση context-aware υπηρεσιών σε mobile video που έχει αρχίσει να αποκτάει έδαφος στη σχετική αγορά.

Στο κεφάλαιο 2, περιγράφουμε την έννοια των context-aware υπηρεσιών, τις αιτίες δημιουργίας τους και τα προβλήματα που αντιμετωπίστηκαν στις απαρχές τους. Ακόμα δίνουμε λεπτομέρειες για τον τρόπο ανάπτυξης σε θεωρητικό αλλά και σε πρακτικό επίπεδο με την καταγραφή παραδειγμάτων.

Στο κεφάλαιο 3, περιγράφουμε την έννοια του mobile video, αναφέρουμε πώς με την πάροδο του χρόνου έχει αποκτήσει μερίδιο στο καταναλωτικό κοινό και καταγράφουμε χαρακτηριστικά παραδείγματα εφαρμογών του.

Στο κεφάλαιο 4, περιγράφουμε τον τρόπο που συνδυάζεται το mobile video με τις context-aware υπηρεσίες. Πιο συγκεκριμένα, καταγράφουμε τις απαιτήσεις ενός τέτοιου συνδυασμός και την αντιμετώπιση αυτών των απαιτήσεων κάποιων αρχιτεκτονικών που προτάθηκαν .

Στο κεφάλαιο 5, περιγράφουμε την αρχιτεκτονική της εφαρμογής που υλοποιήσαμε εμείς. Συγκεκριμένα καταγράφουμε τα σενάρια χρήσης που ορίσαμε, λειτουργίες και συνθήκες περιβάλλοντος και αναφερόμαστε συντόμως στον τρόπο υλοποίησης, στις πλατφόρμες που χρησιμοποιήσαμε και στους λόγους που μας οδήγησαν στην επιλογή αυτών.

Στο κεφάλαιο 6, περιγράφουμε πλέον με αναλυτικά βήματα τόσο την υλοποίηση της εφαρμογής όσο και τα εργαλεία των πλατφορμών που διαθέσαμε. Το κεφάλαιο αυτό είναι περισσότερο τεχνικό.

Στο κεφάλαιο 7, περιγράφουμε τις επιδόσεις της εφαρμογής μας στηριζόμενοι σε πειράματα και συγκεκριμένες μετρήσεις που κάναμε, ενώ καταγράφουμε επίσης και τα συμπεράσματά μας για την υλοποίηση του συστήματος.

Τέλος στο κεφάλαιο 8, καταγράφουμε τις αναφορές και τη βιβλιογραφία στην οποία στηριχθήκαμε για να γίνει εφικτή η συγκεκριμένη εργασία

ΚΕΦΑΛΑΙΟ 2

CONTEXT-AWARE ΥΠΗΡΕΣΙΕΣ

2.1 Γενικά

Ένας από τους στόχους μιας “έξυπνης” (smart) συσκευής είναι να υποστηρίζει και ενισχύει τις δυνατότητες των κατόχων της στην εκτέλεση των στόχων. Αυτοί οι στόχοι κυμαίνονται από την πλοήγηση σε μια άγνωστη περιοχή ως την παροχή των υπενθυμίσεων για τις δραστηριότητες, στην κίνηση των βαριών αντικειμένων για τους ηλικιωμένους ή τα άτομα με ειδικές ανάγκες. Προκειμένου να εξυπηρετηθούν τα άτομα, η έξυπνη συσκευή πρέπει να είναι σε θέση να ανιχνεύει τη τρέχουσα κατάσταση ή το πλαίσιο (context) του περιβάλλοντος και να αποφασίσει ποιες ενέργειες πρέπει να γίνουν με βάση αυτές τις πληροφορίες πλαισίου[13] . Καθορίζουμε ως πλαίσιο (context) οποιαδήποτε πληροφορία μπορεί να χρησιμοποιηθεί για να χαρακτηριστεί η κατάσταση μιας οντότητας, όπου η οντότητα μπορεί να είναι πρόσωπο, τοποθεσία, φυσικό ή υπολογιστικό αντικείμενο. Πιο συγκεκριμένα διακρίνουμε τις εξής κατηγορίες περιεχομένου(context)[35]:

- Υπολογιστικό περιεχόμενο όπως συνδεσιμότητα, επικοινωνιακό κόστος κι εύρος και διαθέσιμες γειτονικές συσκευές πχ εκτυπωτές, οθόνες και servers.
- Περιεχόμενο χρήστη όπως θέση ή προσωπικό profile .
- Φυσικό περιεχόμενο όπως φωτεινότητα, θερμοκρασία, υγρασία ή επίπεδα θορύβου.
- Χρονικό περιεχόμενο όπως ώρα, μέρα, μήνα, χρόνο ή εποχή.

Οι εφαρμογές που χρησιμοποιούν το πλαίσιο αυτό για να παρέχουν σχετικές με τον κάθε σκοπό πληροφορίες ή υπηρεσίες σε έναν χρήστη ονομάζονται context-aware. Η αυξανόμενη διαθεσιμότητα εμπορικών τεχνολογιών έχουν ωθήσει σε μια άνθιση context-aware εφαρμογών σε ποικίλα περιβάλλοντα. Αυτές οι τεχνολογίες μάλιστα επιτρέπουν στις συσκευές να ερμηνεύουν τους χρήστες και τις ανάγκες τους. Επιπλέον, η επικράτηση των ισχυρών υπολογιστικών δικτύων καθιστά εύκολη τη χρησιμοποίηση διαφόρων τεχνολογιών και τη διανομή του context και σε άλλες εφαρμογές. [13]

2.2 Λόγοι Δημιουργίας και Προβλήματα

Όλες οι καινούργιες ιδέες που προωθούνται στην επιστήμη κρύβουν κάποιο κίνητρο που σχεδόν πάντα σχετίζεται με την κάλυψη ή την εκπλήρωση κάποιων ανθρώπινων αναγκών. Και στην περίπτωση λοιπόν των context-aware υπηρεσιών πρέπει να αναζητήσουμε το “κινητήριο μοχλό”. Ποιος ο λόγος που οδηγηθήκαμε στην έννοια του context; Ποια ήταν η αιτία που αναζητήσαμε την κατασκευή έξυπνων συσκευών που θα αντιλαμβάνονταν το περιβάλλοντα χώρο και τους χρήστες; Η απάντηση είναι απλή: Χρηστικότητα, δυναμικότητα, ευελιξία, εξοικονόμηση ενέργειας, εξοικονόμηση χρόνου. Όλα τα παραπάνω είναι και οι λόγοι που ώθησαν τον άνθρωπο να σκεφτεί τις context-aware υπηρεσίες. [29]

Η τεχνολογική πρόοδος έχει διαγράψει ένα μονοπάτι που ολοένα και προχωράει. Οι εξελίξεις πλέον είναι τόσο ραγδαίες που σήμερα δεν μπορούμε να προβλέψουμε τη μορφή του αύριο. Πλέον τα πάντα είναι πιθανά να συμβούν στην επιστήμη, ακόμα και αυτά που δεν έχουμε ακόμα σκεφτεί και το μόνο πράγμα για το οποίο μπορούμε να είμαστε σίγουροι είναι πως κάθε μέρα κάνουμε βήματα μπροστά. Επειδή λοιπόν οι διαδικασίες τρέχουν και είναι πάρα πολλές, η βασική ανάγκη που έχει πλέον ο άνθρωπος ως χρήστης της τεχνολογίας είναι η ευκολία που του παρέχει. Μέσα στη δίνη των επιτευγμάτων θέλει να αποφύγει τη διαδικασία της εκμάθησης σε μια νέα συσκευή, θέλει να εξοικονομήσει χρόνο και ενέργεια για τον εαυτό του και να αναλάβει η τεχνολογία τον έλεγχο των μικροπραγμάτων που ίσως τον αποσπάνε μέσα στην καθημερινότητα και τον επιβαρύνουν. Με λίγα λόγια η ανάγκη αποφόρτισης που κάθε σύγχρονο άτομο έχει εξαιτίας των γρήγορων ρυθμών της καθημερινότητας του οδήγησε στην ιδέα των υπηρεσιών ενημέρωσης πλαισίου (context-aware), δηλαδή υπηρεσίες που θα προσφέρονται αυτόματα στο χρήστη μέσω έξυπνων τεχνολογιών που «κατανοούν» το άτομο, το περιβάλλον και τη σχέση μεταξύ τους.

Όλες όμως οι καινοτομίες αντιμετωπίζουν κάποια προβλήματα στην πρακτική εκτέλεσή τους. Το σημαντικό πρόβλημα που αντιμετώπισε η επιστημονική κοινότητα αρχικά είναι η έλλειψη ομοιόμορφης υποστήριξης για την οικοδόμηση και εκτέλεση των context-aware εφαρμογών. Οι περισσότερες έχουν δομηθεί κατά τέτοιο τρόπο ώστε να επηρεάζονται δραματικά από την τεχνολογία στην οποία στηρίχτηκαν για να αποκτήσουν τις διάφορες πληροφορίες του πλαισίου. Αυτό οδηγεί σε μια έλλειψη γενικότητας που με τη σειρά της εξαναγκάζει κάθε καινούργια εφαρμογή να δρομολογείται εξολοκλήρου από την αρχή καθώς δεν υπάρχει η δυνατότητα αλληλεπίδρασης και συμβατότητας. Πρακτικά αυτό σημαίνει δυσκολία στη μεταφορά των τεχνολογιών σε κάποιο διαφορετικό περιβάλλον από αυτό για το οποίο αρχικά δημιουργήθηκαν καθώς και μη χρηστικότητα των πληροφοριών που συλλέγει μια εφαρμογή μέσα από τις λειτουργίες της σε άλλες εφαρμογές που πιθανώς στηρίζονται σε διαφορετικές τεχνολογικές πλατφόρμες. Θα παρουσιάσουμε κάποιες από αυτές τις προσπάθειες προσέγγισης των context-aware υπηρεσιών στην παράγραφο 2.4 αναλυτικότερα.

Προφανώς, έπρεπε να αντιμετωπιστεί αυτό το πρόβλημα συμβατότητας για να μπορέσουν οι χρήστες να κερδίσουν στο έπακρο τα οφέλη των context-aware υπηρεσιών. Για αυτό το λόγο θα έπρεπε να υπάρξει μια αρχιτεκτονική υποδομή που θα παρέχει μηχανισμούς που θα είναι υποχρεωτικοί για όλες τις context-aware εφαρμογές. Σε αυτήν την υποδομή θα πρέπει να περιλαμβάνεται ουσιαστικά ένα επίπεδο που θα μπορεί να αποκρύπτει πληροφορίες του πλαισίου από ξένες εφαρμογές και ταυτόχρονα να παρέχει αυτές τις πληροφορίες με ευελιξία σε “φιλικές” εφαρμογές που της ζητούν το καταγεγραμμένο περιεχόμενο. Στη δεύτερη περίπτωση μάλιστα θα πρέπει να επιτυγχάνεται η μεταφορά χωρίς να υπάρχει ο κίνδυνος ασυνεννοησίας μεταξύ των εφαρμογών .

2.3 Τρόπος λειτουργίας

Η διαχείριση του περιεχομένου από τις “έξυπνες” συσκευές γίνεται με κάπως αυθαίρετο τρόπο κάθε φορά. Όπως αναφέραμε και στην προηγούμενη παράγραφο (βλ παράγραφο 2.2), η έλλειψη συγκεκριμένων καθολικά αποδεκτών προτύπων έχει οδηγήσει τις context-aware εφαρμογές σε ποικίλους πειραματισμούς ως προς τον τρόπο συλλογής κι εκμετάλλευσης των πληροφοριών του περιβάλλοντος που νοούμε στην περίπτωση μας ως περιεχόμενο. Έπρεπε να ανακαλυφτεί μια μέθοδος από την επιστημονική κοινότητα η οποία θα «δαισιθάνεται» το περιβάλλον γύρω μας, ή αλλιώς θα καταγράφει την πληροφορία του χώρου με μαθηματικοποιημένο τρόπο ώστε να μπορεί στη συνέχεια να χρησιμοποιηθεί αυτή σε τεχνολογικές συσκευές που αντιλαμβάνονται μόνο αριθμούς και υπολογισμούς. Η μέθοδος αυτή που βοήθησε στη ανάπτυξη των context-aware υπηρεσιών είναι η χρήση αισθητήρων που θα αναλύσουμε στη συνέχεια της εργασίας. Υπάρχουν σήμερα διάφορες τεχνικές αξιοποίησης των στοιχείων που προσφέρουν οι αισθητήρες (sensors) στις εφαρμογές. Οι προγραμματιστές των εφαρμογών επιλέγουν συνήθως την τεχνική που είναι πιο εύκολη στην προσέγγισή της και ταυτόχρονα θα έχει το πλεονέκτημα της γενικής χρήσης της σε πολλές διαφορετικές περιπτώσεις. Σε ένα σύνολο τεχνικών που έχουν προταθεί κατά καιρούς, εμείς θα ερευνήσουμε τις δύο πλέον επικρατέστερες:

- Η απευθείας σύνδεση των αισθητήρων (sensors) στις εφαρμογές και
- Η άμεση επικοινωνία μέσω servers που αποθηκεύουν τις πληροφορίες από τους sensors

2.3.1 Απευθείας Σύνδεση

Σε ορισμένες εφαρμογές το λογισμικό των αισθητήρων που χρησιμοποιείται για την “αντίληψη” του περιβάλλοντος συνδέεται άμεσα στα συστήματα των εφαρμογών. Στη συγκεκριμένη περίπτωση οι σχεδιαστές αναγκάζονται να κατασκευάσουν κώδικα που να μπορεί να επικοινωνεί με τους sensors χρησιμοποιώντας το πρωτόκολλο που οι ίδιοι οι αισθητήρες ορίζουνε. Αντιλαμβανόμαστε επομένως ότι υπάρχει μια δέσμευση κι έλλειψη ελευθερίας επιλογών που οδηγούν σε δύο προβλήματα.

Το πρώτο πρόβλημα είναι πως αυτός ο περιορισμός επιβαρύνει αρκετά το “χτίσιμο” της εφαρμογής καθώς οι σχεδιαστές θα πρέπει κάθε φορά να αντιμετωπίζουν εκ νέου την πολυπλοκότητα της ανάκτησης του περιεχομένου. Από τη στιγμή που η εφαρμογή συνδέεται άμεσα με τους αισθητήρες, κάθε νέα αλλαγή ή προσθήκη κάποιου sensor θα οδηγεί στην επίπονη διαδικασία της ενσωμάτωσης νέου κώδικα, που φυσικά θα πρέπει να είναι συμβατός όχι μόνο με τον καινούργιο αισθητήρα (sensor) αλλά και με όλα τα προϋπάρχοντα στοιχεία.

Το δεύτερο πρόβλημα με αυτή την τεχνική είναι ότι δεν υποστηρίζει απαιτούμενες μηχανικές πρακτικές που πρέπει να χαρακτηρίζουν ένα λογισμικό. Αυτό συμβαίνει διότι δεν ενισχύει τον διαχωρισμό των σημασιολογικών εννοιών και των χαμηλού-επιπέδου πληροφοριών που λαμβάνονται από διάφορους αισθητήρες. Με τον τρόπο αυτό χάνεται το πλεονέκτημα της γενικότητας καθώς οι αισθητήρες συνδέονται με μια συγκεκριμένη εφαρμογή και η ταυτόχρονη χρήση τους από άλλες καθίσταται δύσκολη, και πολλές φορές αδύνατη.

2.3.2 Επικοινωνία μέσω server

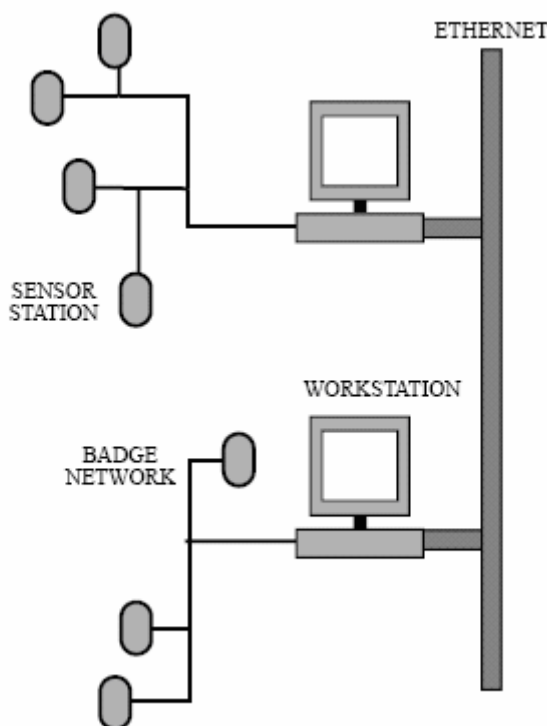
Η δεύτερη διαδομένη τεχνική αναφέρεται στη λειτουργία ενός κεντρικού εξυπηρετητή (server) που σχεδιάζεται για να συσσωρεύει και να αποθηκεύει πληροφορίες του περιβάλλοντος μέσω “ερωτήσεων” στο δίκτυο αισθητήρων. Πιο συγκεκριμένα λοιπόν οι sensors λειτουργούν συνεχώς και συλλέγουν στοιχεία. Σε τακτά χρονικά διαστήματα που ορίζονται από τους σχεδιαστές οι servers κάνουν ερωτήσεις στους αισθητήρες και ως απαντήσεις λαμβάνουν τις συλλεγόμενες πληροφορίες. Για την ακρίβεια οι servers διατηρούν χρήσιμα δεδομένα για λογαριασμό των εφαρμογών. Οι εφαρμογές με τη σειρά τους, όταν το απαιτήσουν οι συνθήκες, απευθύνουν ερωτήσεις στους servers για τα αποθηκευμένα στοιχεία. Με τη μέθοδο αυτή λοιπόν τα δύο προαναφερθέντα προβλήματα της πρώτης τεχνικής βρίσκουν τη λύση τους.

Το πρώτο αντιμετωπίζεται καθώς οι σχεδιαστές ανακουφίζονται από το βάρος της συνεχούς προσθήκης νέου και συμβατού κώδικα που θα πρέπει να ελέγχεται από την αρχή για συμβατότητα με όλα τα προηγούμενα στοιχεία. Όσον αφορά το δεύτερο, οι εξυπηρετητές παρέχουν τη δυνατότητα διαχωρισμού των πληροφοριών με τελικό αποτέλεσμα την ευκολότερη δημιουργία context-aware εφαρμογών. Πλέον, πολλές εφαρμογές θα μπορούν να επικοινωνούν με κοινούς αισθητήρες μέσω της χρησιμοποίησης ενός κοινού server.

Στην εικόνα 2-1 μπορούμε να παρατηρήσουμε διαγραμματικά την αρχιτεκτονική της μεθόδου που μόλις περιγράψαμε.

Παρατηρούμε καταρχήν ότι υπάρχουν πολλοί ξεχωριστοί αισθητήρες που καταγράφονται ως sensor stations. Ο αριθμός τους ποικίλλει ανάλογα με το είδος της πληροφορίας που θέλουμε να συλλέξουμε ή την αξιοπιστία της πληροφορίας. Όταν μας ενδιαφέρουν διαφορετικά δεδομένα του χώρου όπως θερμοκρασία, φωτεινότητα και υγρασία παραδείγματος χάριν χρειαζόμαστε κι ισάριθμους αισθητήρες καθώς ο κάθε ένας είναι υλοποιημένος για διαφορετικό είδος ευαισθησίας στο περιβάλλον. Κάποιες άλλες φορές όμως, μπορεί να μας ενδιαφέρει συγκεκριμένη κατηγορία δεδομένων ,πχ μόνο ο θόρυβος του χώρου. Και στη δεύτερη αυτή περίπτωση θα μπορούσε να

υπάρχει μεγάλος αριθμός από sensors που ο καθένας θα επεξεργαζότανε ένα χρονικό ή χωρικό κομμάτι του περιβάλλοντος κι επομένως θα αυξάναμε σημαντικά την αξιοπιστία των μετρήσεών μας. Στη συνέχεια τα δεδομένα που συλλέγονται από τους εκάστοτε αισθητήρες αποστέλλονται με ασύρματο κυρίως τρόπο σε ένα κεντρικό υπολογιστή που έχουμε ορίσει ως server. Στο σημείο αυτό ο server θα μετατρέψει την πληροφορία που του έρχεται από τα διάφορα σημεία και ίσως με διαφορετικό format, σε μια κοινή μορφή για να την αποστείλει στη συνέχεια μέσω δικτύου Ethernet στην κεντρική πλατφόρμα της εφαρμογής μας (σταθμός εργασίας-workstation). Στο σημείο αυτό μπορεί να προωθηθεί η πληροφορία σε επιμέρους στοιχεία της εφαρμογής μας που δεν εντοπίζονται στο κεντρικό κορμό αλλά είναι συνδεδεμένα με αυτή κι επιθυμούν να πληροφορηθούν. Στην περίπτωση αυτών των επιπλέον τερματικών έχουμε το λεγόμενο δίκτυο διακριτικών (badge network).



Εικόνα 2-1 : Σταθμός αισθητήρων και μετάβαση πληροφοριών σε server[36]

Και η δεύτερη τεχνική, όμως, φέρνει αντιμέτωπους τους μηχανικούς με δύο προβλήματα. Το πρώτο είναι ότι οι εφαρμογές που χρησιμοποιούν τους servers πρέπει να είναι δυναμικές αναγνωρίζοντας πότε είναι η κατάλληλη στιγμή που απαιτείται να απευθύνουν ερωτήσεις στους εξυπηρετητές για τη συλλογή στοιχείων. Αυτό σημαίνει ότι η εφαρμογή πρέπει να αντιλαμβάνεται αν υπάρχει διαφοροποίηση στον περιβάλλον και αν αυτή είναι ενδιαφέρουσα για την ίδια. Το δεύτερο πρόβλημα είναι πως οι servers αναπτύσσονται ανεξάρτητα για κάθε είδος αισθητήρα και παράλληλα διατηρεί ένα ξεχωριστό interface για την κάθε εφαρμογή με την οποία επικοινωνεί. Αυτό με τη σειρά του υποδηλώνει ότι κάθε εφαρμογή αλληλεπιδρά με ένα server μεν, αλλά με διαφορετικό τρόπο κάτι που ουσιαστικά παρομοιάζεται με τον πρόβλημα της πρώτης τεχνικής και της διαχείρισης διαφορετικών αισθητήρων

2.4 Χρήση τεχνολογιών γνώσης για την ανάπτυξη context-aware υπηρεσιών.

Όπως αναφέραμε ήδη οι context-aware υπηρεσίες εστιάζουν στο «περιεχόμενο» και στη διαχείρισή του από έξυπνες συσκευές. Ο τρόπος που μπορεί να επιτευχθεί αυτό ποικίλλει. Στην υπό-ενότητα αυτή θα παρουσιάσουμε τους πιθανούς τρόπους υλοποίησης, θα καταγράψουμε τα πλεονεκτήματα και τα μειονεκτήματα του καθενός, θα περιγράψουμε συνοπτικά κάποια παρόμοια συστήματα και θα εξηγήσουμε τους λόγους για τη δική μας προσέγγιση που ακολουθήσαμε στην εργασία.

2.4.1 Πιθανοί τρόποι υλοποίησης

Για να αντιληφθούμε το «πλαίσιο» γύρω μας, εκμεταλλευόμαστε τις δυνατότητες των αισθητήρων όπως προαναφέραμε. Με ποια υλοποίηση οι αισθητήρες μέσω των server ή και αυτόνομα θα συνδεθούν με την εκάστοτε εφαρμογή; Κάποιοι από τους πιθανούς τρόπους είναι οι εξής:

a. Επεξεργασία σήματος

Στην περίπτωση αυτή, οι αισθητήρες παράγουν αναλογικές τιμές τις οποίες και στέλνουν στην εφαρμογή. Στη συνέχεια η εφαρμογή πρέπει να υλοποιήσει τεχνικές επεξεργασίας σήματος. Ο τρόπος αυτός ανάπτυξης έχει το πλεονέκτημα της απλότητας ως προς τη σύλληψή του και των πολλών διαθέσιμων τεχνικών που υπάρχουν σήμερα. Το βασικό μειονέκτημα όμως είναι ότι διαχειρίζεται αναλογικά σήματα που είναι ευαίσθητα στο θόρυβο και στις παρεμβολές και επομένως το τελικό αποτέλεσμα δεν είναι αρκετά αξιόπιστο. Επιπλέον, αυτός είναι ένας «μονολιθικός» τρόπος χρήσης των τιμών των αισθητήρων αφού δεν είναι εύκολο να αλλάξουν δυναμικά ο σχετικός κώδικας που αναλαμβάνει την επεξεργασία σήματος.

b. Κανόνες (Rules) - Πρώτης τάξης κατηγορήματα (First-order predicates)

Στην περίπτωση αυτή χρησιμοποιούνται κανόνες λογικής για να ορίσουμε το context και αποφασίσουμε τις ενέργειές μας απευθείας. Το περιεχόμενο εδώ καθορίζεται μέσω κατηγορημάτων πρώτης τάξης. Συγκεκριμένα υπάρχει η σύμβαση ότι το όνομα του κάθε κατηγορήματος είναι ο τύπος του περιεχομένου που περιγράφει. Κάποια παραδείγματα είναι τα εξής :

- Location (chris, entering, room 3231)
- Temperature (room 3231 , “=”, 98 F)
- Time (New York , “<”, 12:00 01/01/01)

Η Prolog είναι μια τέτοια γλώσσα που βασίζεται ακριβώς σε αυτή τη δομή των κατηγορημάτων. Το πλεονέκτημα της συγκεκριμένης δομής είναι ότι η παραπάνω σύμβαση μας εξασφαλίζει μια απλή κι ενιαία αναπαράσταση του περιεχομένου. Επίσης παρέχει τη δυνατότητα να περιγραφεί το περιεχόμενο ως οντολογία ενώ χρησιμοποιούμε με ευκολία και τελεστές όπως = και <. Όσον αφορά τα μειονεκτήματα, είναι περίεργο αλλά μπορούμε να εντάξουμε κι εδώ τη σύμβαση ονοματολογίας. Πρέπει να ακολουθούμε πάντα συγκεκριμένες τιμές ακόμα κι αν εμείς θα θέλαμε να ορίσουμε κάπως διαφορετικά το

«περιεχόμενο». Επιπλέον οι μεταβλητές είναι απόλυτα προκαθορισμένες ως προς τον τύπο. Για παράδειγμα αν το είδος του περιεχομένου είναι τοποθεσία (location), τότε η πρώτη μεταβλητή θα πρέπει να είναι πρόσωπο ή αντικείμενο, η δεύτερη θα πρέπει να είναι πρόθεση ή ρήμα και η τρίτη θα πρέπει να είναι τοποθεσία. Καταλαβαίνουμε ότι είναι κάπως δεσμευτικό.[38]

c. Γεγονός –Έλεγχος-Ενέργεια (Event-Control-Action , ECA)

Στην περίπτωση αυτή, χρησιμοποιείται η προτυποποιημένη γλώσσα ECA-DL[39], μια προγραμματιστική γλώσσα αποκλειστικά δημιουργημένη για context-aware εφαρμογές. Η συγκεκριμένη συνδυάζεται με διάφορες άλλες όπως Jess[39] και CLIPS που αποτελούν το συνδυαστικό κρίκο μεταξύ της γλώσσας μηχανής και της κύριας εφαρμογής μας. Το πλεονέκτημα της συγκεκριμένης υλοποίησης είναι ότι το μοντέλο ECA είναι κατασκευασμένο για να διαχειρίζεται κάθε είδους context-aware υπηρεσίας και μπορεί να αποδώσει μια γενικότητα στην εφαρμογή μας. Επίσης παρέχει τη δυνατότητα επικοινωνίας με ποικίλες προγραμματιστικές γλώσσες. Ως μειονέκτημα μπορούμε να θεωρήσουμε το γεγονός πως οι πρέπει να υπάρχει αυτή η μεσολάβηση και η συνεχής αντιστοίχιση της ECA-DL σε γλώσσα κατανοητή για την εφαρμογή όπως η C ή Java.[39]

d. Απευθείας στη γλώσσα προγραμματισμού

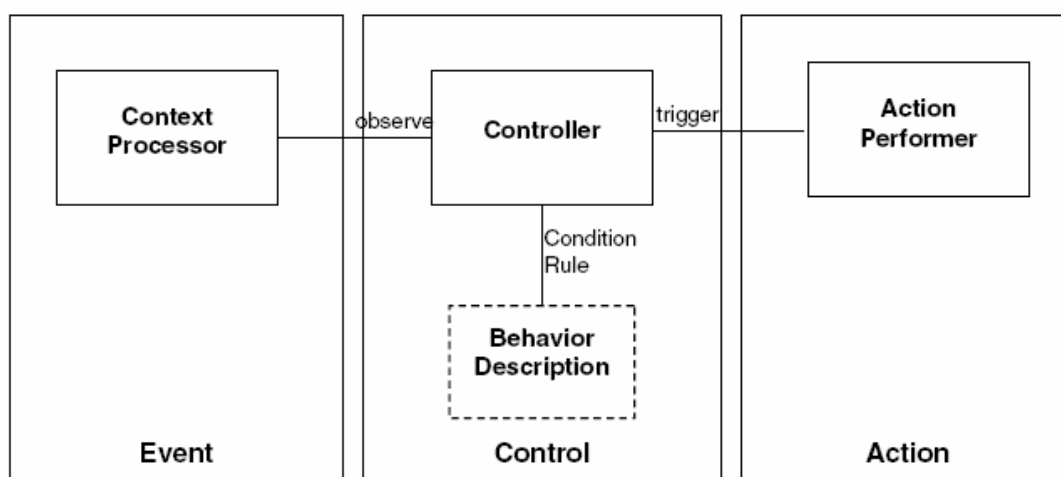
Η υλοποίηση μπορεί να γίνει και μέσω υψηλού επιπέδου (high-level) προγραμματιστική γλώσσα. Αντί να υπάρχει μια μηχανή κανόνων και να φορτώνει τις τιμές των αισθητήρων, θα μπορούσαμε να περνάμε τις μεταβλητές αυτές κατευθείαν σε ένα πρόγραμμα java για παράδειγμα. Το πλεονέκτημα είναι πως δε θα χρειαζότανε να εμπλέξουμε πολλές διαφορετικές τεχνολογίες που ίσως θα αύξανε τη δυσκολία υλοποίησης και την υπολογιστική πολυπλοκότητα της εφαρμογής, όμως το βασικό μειονέκτημα είναι πως χάνεται τελείως η ευελιξία της εκάστοτε εφαρμογής, καθώς θα υπάρχει η ανάγκη της συνεχούς αλλαγής στο κώδικα με κάθε αλλαγή στο χειρισμό των δεδομένων των αισθητήρων.

2.4.2 Περιγραφή παρόμοιων συστημάτων

Στη συνέχεια βλέπουμε συνοπτικά κάποια συστήματα παρόμοια με το αντίστοιχο που θελήσαμε να υλοποιήσουμε εμείς.

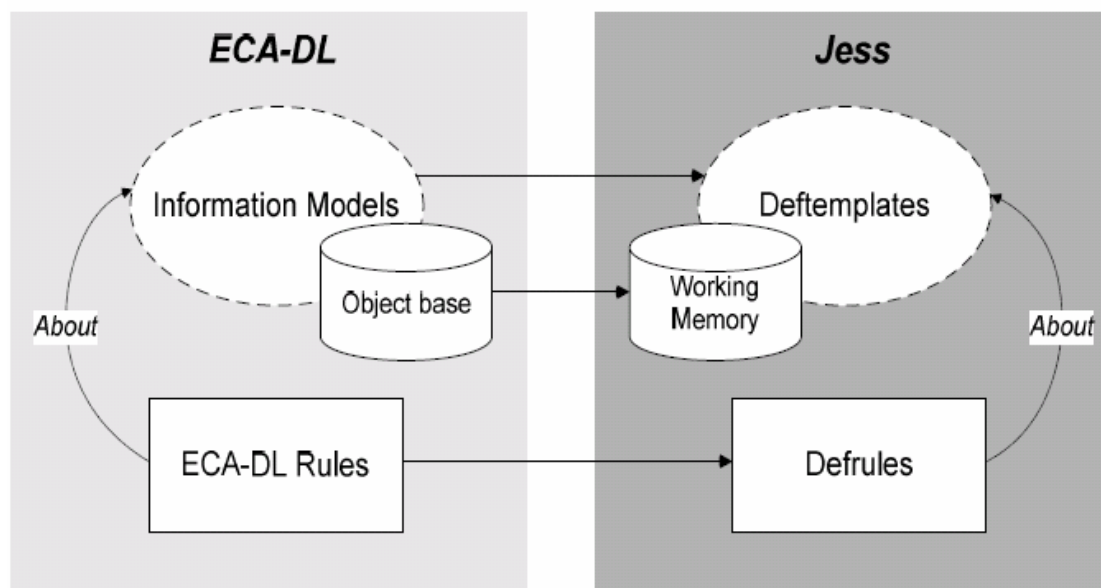
2.4.2.1 ECA –Jess Μοντέλο

Στο Κέντρο Τηλεματικής και Τεχνολογίας της πληροφορικής στο Πανεπιστήμιο της Ολλανδίας, έγινε μια απόπειρα για ανάπτυξη context-aware εφαρμογής. Χρησιμοποίησαν το μοντέλο Event-Control-Action συνδυάζοντας το με τη γλώσσα Jess για να μπορέσουν να το συνδέσουν με τη κύρια πλατφόρμα που αφορούσε ενημέρωση κάποιας ομάδας ατόμων για διάφορες ενέργειες που έπρεπε να ακολουθήσουν. Το μοντέλο που ακολουθήθηκε παρουσιάζεται στην εικόνα 2-2 και 2-3.



Εικόνα 2-2 :Αντιστοίχιση ECA-DL σε Jess[39]

Πιο συγκεκριμένα η εικόνα 2-2 παρουσιάζει τη γενική δομή του μοντέλου. Το τμήμα Event συλλέγει και επεξεργάζεται το περιεχόμενο μέσω του Context Processor. Το τμήμα Control ελέγχει τα δεδομένα που δέχεται από το Event module και με βάση συγκεκριμένες προκαθορισμένες συμπεριφορές (behavior description) που καταγράφονται σε κανόνες ECA (γνωστούς κι ως Condition Rules), στέλνει ερεθίσματα στο μοντέλο Action (trigger). Στο τελευταίο τμήμα αυτό (Action module) χρησιμοποιείται ο λεγόμενος Action Performer ο οποίος αναλαμβάνει και την εκτέλεση των διαφορών ενεργειών.

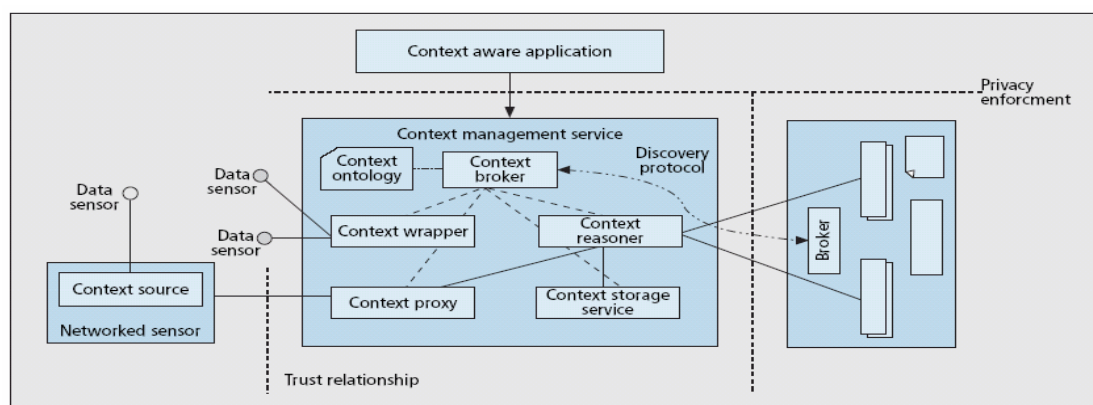


Εικόνα 2-3 :Αντιστοίχιση ECA-DL σε Jess[39]

Στην εικόνα 2-3 παρουσιάζεται απλά διαγραμματικά ο τρόπος με τον οποίο αντιστοιχίζονται οι πληροφορίες από τη context-aware γλώσσα ECA-DL στην γλώσσα κανόνων Jess.

2.4.2.2 Ανεπτυγμένο Context Management Μοντέλο

Στο Πανεπιστήμιο του Twente, πραγματοποιήθηκε άλλη μια προσπάθεια για ανάπτυξη context-aware mobile εφαρμογής στο τομέα της υγείας. Και εδώ χρησιμοποιήθηκε το γενικότερο μοντέλο ECA στο οποίο όμως ενσωματώθηκαν πολλά ανεπτυγμένα context-aware εργαλεία. Στην εικόνα 2-4 παρουσιάζεται συνοπτικά η αρχιτεκτονική που ακολουθήθηκε.



Εικόνα 2-4 :Context-aware αρχιτεκτονική[40]

Όπως παρατηρούμε, από το μοντέλο αυτό υποστηρίζεται ένα ευρύ σύνολο από εργαλεία. Υπάρχει η πηγή περιεχομένου (context source) που είναι υπεύθυνη για τη σύλληψη των δεδομένων από τους αισθητήρες. Στη συνέχεια η αρχική, πρωτογενής πληροφορία περνάει στην κεντρική υπηρεσία διαχείρισης περιεχομένου (Context management service). Μέσω ενός διακομιστή περιεχομένου (context proxy) τα δεδομένα μεταδίδονται προς το αναλυτή περιεχομένου (context reasoner) που αναλύει με βάση συγκεκριμένη λογική τα στοιχεία και το αποδέκτης περιεχομένου (context wrapper) που δέχεται αυτόματα τιμές από άλλους αισθητήρες. Η πληροφορία από το reasoner είτε αποθηκεύεται για μελλοντική χρήση στην υπηρεσία αποθήκευσης περιεχομένου (context storage service) είτε προωθείται στον διασπαστή (broker) και εν συνεχεία στο διασπαστή περιεχομένου (context broker) που έχουν ως εργασία την διάσπαση της πληροφορίας σε διαφορετικά τμήματα έτσι ώστε να αντιστοιχίζεται στις αποθηκευμένες οντολογίες που χρησιμοποιούνται.

2.4.3 Τρόπος υλοποίησης που ακολουθήθηκε

Ο τρόπος υλοποίησης που ακολουθήσαμε εμείς αφορούσε την εκτέλεση κάποιων κανόνων (rules) μέσω μιας απλής και κατανοητής μηχανής. Θέσαμε επομένως τους αισθητήρες να στέλνουν απευθείας πληροφορία στην εφαρμογή μας χωρίς τη μεσολάβηση server ο οποίος θα συνέλλεγε αρχικά τα δεδομένα. Στη συνέχεια χρησιμοποιήσαμε δύο αρχεία γραμμένα σε γλώσσα prolog. Στο ένα είχαμε αποθηκεύσει τους κανόνες που μας ενδιέφεραν για την εφαρμογή μας και στο δεύτερο περνούσαμε κάθε φορά που ενεργοποιούνταν οι αισθητήρες, τα νέα δεδομένα ώστε να είμαστε ενήμεροι με το περιβάλλον. Δεν αναπτύξαμε οντολογίες, ούτε ιεραρχικές δομές περιγραφής context προσπαθώντας να διατηρήσουμε όσο πιο απλή γινότανε την εφαρμογή μας. Διαθέταμε μόνον συγκεκριμένους κανόνες, στους οποίους φορτώνονταν οι τιμές των αισθητήρων για να ληφθούν οι αποφάσεις ενέργειας.

Οι λόγοι που επιλέξαμε την γλώσσα Prolog μεταξύ άλλων ποικίλλουν. Καταρχήν η Prolog είναι ένα προκαθορισμένο πρότυπο κι αυτό είναι ευνοϊκό καθώς είναι αποδεκτή και κατανοητή στο ευρύτερο σύνολο. Παρέχει την εκφραστικότητα που αναζητάμε ώστε να αντιστοιχίσουμε τα δεδομένα του

περιβάλλοντος σε προγραμματιστικούς όρους ενώ η διαχείριση των rules μας βοηθάνε να μεταβάλλουμε εύκολα τους κανόνες χωρίς υπολογιστικό κόστος και χωρίς να επηρεάζεται η εφαρμογή όπως θα γινόταν εάν τους είχαμε υλοποιήσει σε γλώσσα Java ή C. Επίσης είναι διαχειρίσιμη από γλώσσες εφαρμογής όπως η mobile java που χρησιμοποιήσαμε στην εργασία μας ενώ δεν επιβαρύνει σημαντικά το σύνολο της εφαρμογής ως προς την καθυστέρηση και τη μνήμη της εφαρμογής. Ειδικά το τελευταίο σημείο είναι πολύ σημαντικό γιατί όπως θα δούμε στη συνέχεια το Mobile video που επιθυμούμε να αναπτύξουμε χαρακτηρίζεται από πολύ αυστηρούς περιορισμούς ως προς την ισχύ, τη μνήμη και την αποκρισιμότητα λόγω του μεγέθους και της φορητότητας της συσκευής.

2.5 Παραδείγματα context-aware υπηρεσιών κι εφαρμογές

Μέχρι στιγμής αναφερθήκαμε στην γενική έννοια των context-aware υπηρεσιών αλλά δεν έχουμε παρουσιάσει ακόμα πως αυτή η σχετικά καινούργια ιδέα βρίσκει εφαρμογή πρακτικά στην καθημερινή μας ζωή. Στη συνέχεια θα καταγράψουμε διάφορα παραδείγματα τέτοιων υπηρεσιών.

2.5.1 Συσκευές πλοήγησης με την τεχνολογία GPS

Το πιο χαρακτηριστικό παράδειγμα context-aware υπηρεσίας είναι η τεχνολογία GPS που χρησιμοποιείται για την πλοήγηση των οχημάτων στους δρόμους. Η τεχνολογία αυτή σχετίζεται με πληροφορίες που συλλέγονται, με τη βοήθεια δορυφόρων, για το γενικό χάρτη και τις διαδρομές ολόκληρων περιοχών. Το GPS διαθέτει ουσιαστικά δέκτες που επικοινωνούν με δορυφόρους παρέχοντας στο χρήστη τη δυνατότητα να γνωρίζει την ακριβή θέση του και τον περιβάλλοντα χώρο στον οποίο κινείται. Ο χρήστης μπορεί να εκμεταλλευτεί στο έπακρο αυτή την πληροφορία όταν αναζητάει να μεταβεί σε μια συγκεκριμένη περιοχή καθώς η εφαρμογή πλοήγησης πραγματοποιεί όλη τη διαδικασία αναζήτησης. Ο χρήστης εισάγει την επιθυμητή τοποθεσία μετάβασης και το GPS υπολογίζει τη διαδρομή και την παραθέτει στο δέκτη οπτικά (μέσω της οθόνης που διαθέτει) αλλά και ακουστικά (μέσω μαγνητοφωνημένων μηνυμάτων). Η εφαρμογή αυτή αντιλαμβανόμαστε ότι καταλαμβάνει εξέχουσα σημασία στην καθημερινότητά μας πλέον καθώς προσφέρει την ευκολία στο άτομο να κινηθεί χωρίς άγχος ακόμα και σε περιοχές που του είναι εντελώς άγνωστες δίχως να σπαταλήσει άσκοπα χρόνο κι ενέργεια. Πέραν όμως της ευκολίας και της εξοικονόμησης χρόνου, ο χρήστης πλέον έχει τη ευκαιρία να γνωρίσει και το χώρο από τον οποίο διέρχεται καθώς η εφαρμογή δεν καταγράφει μόνο τη διαδρομή αλλά και λεπτομέρειες για αυτόν. Μάλιστα καταγράφονται κτήρια και χώροι που διευκολύνουν τον οδηγό στην διαδρομή του όπως είναι βενζινάδικα, νοσοκομεία, αστυνομικές υπηρεσίες, χώροι στάθμευσης και χώροι αναψυχής. Η εικόνα 2-2 αποτελεί μια χαρακτηριστική παρουσίαση πλοήγησης μέσω της τεχνολογίας GPS. Πιο συγκεκριμένα είναι αναπαράσταση μιας περιοχής της

Αγγλίας. Καταγράφονται δρόμοι με τα ονόματά τους όπως ο Cambridge, διασταυρώσεις και επισημαίνονται κτήρια συγκεκριμένης σημασίας είτε για το ευρύ κοινό είτε για τον εκάστοτε χρήστη. Επί παραδείγματι, έχει επισημανθεί στην εικόνα αυτή το κτήριο Stata Center που βρίσκεται στην οδό Philips και επιπλέον καταγράφονται και πληροφορίες σχετικές με αυτό που μπορούν να ανταλλάγουν και μεταξύ χρηστών που διαθέτουν την ίδια εφαρμογή.

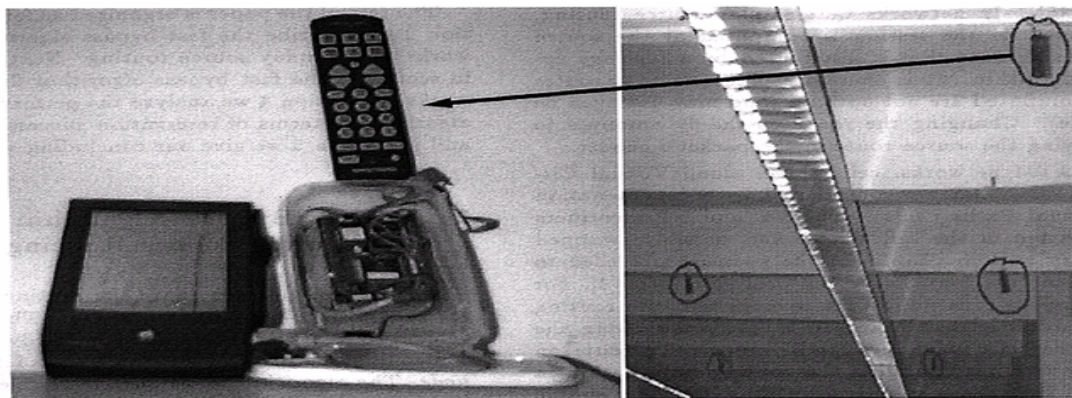


Εικόνα 2-5 :Εφαρμογή context-aware πλοήγησης[1]

2.5.2 Ταξιδιωτικός οδηγός αλληλεπιδραστικός με το περιβάλλον (Context-aware tour guide)

Μια άλλη context-aware εφαρμογή είναι η ιδέα ενός αλληλεπιδραστικού με το περιβάλλον ταξιδιωτικού οδηγού (context-aware tour guide). Η εφαρμογή αυτή μπορεί να αφορά τουρίστες, μπορεί όμως και να αποβεί πολύ σημαντική στο τομέα της εκπαίδευσης. Η γενική ιδέα είναι να μαθαίνουμε καινούργια πράγματα εύκολα και συναρπαστικά μέσα από συνηθισμένες διαδικασίες όπως είναι για παράδειγμα η επίσκεψη σε ένα μουσείο. Μέσα στο μουσείο υπάρχουν ποικίλα εκθέματα που το καθένα κρύβει μια ιστορία. Μέχρι σήμερα υπήρχαν πινακίδες που προσπαθούσαν σε 5 σειρές να περιγράψουν όλα όσα συνδέονταν με τα εκθέματα. Η διαδικασία της εκμάθησης ήταν βαρετή και χωρίς βαθύτερο αποτέλεσμα. Η τεχνολογία έρχεται να το αλλάξει αυτό καθώς μέσω context-aware tour guide μας παρέχει τη δυνατότητα να μπαίνουμε σε ένα ξεχωριστό κόσμο κάθε φορά που θα ζωντανεύει μπροστά μας με ήχο, εικόνα (κινούμενη και στατική) και κείμενο. Κάθε αντικείμενο θα διαθέτει ένα microchip που θα έχει αποθηκευμένες οπτικοακουστικές πληροφορίες και κείμενα. Όταν το άτομο θα πλησιάζει στο κάθε έκθεμα, ο ταξιδιωτικός οδηγός θα λαμβάνει τις αντίστοιχες πληροφορίες και ένας νέος κόσμος γνώσης θα ξεδιπλώνεται μπροστά του σα παιχνίδι. Το άτομο πλέον θα έχει πρόσβαση σε κάθε είδους στοιχείο που τον ενδιαφέρει και η μάθηση θα πάψει να είναι κάτι βαρετό.

Στην εικόνα 2-3 παρουσιάζεται μια πειραματική διάταξη ταξιδιωτικού context-aware οδηγού. Στο δεξιό κομμάτι της φωτογραφίας διακρίνουμε πομπούς ενσωματωμένους στο ταβάνι ενός μουσείου ανά τακτά χωρικά διαστήματα. Οι πομποί αυτοί διαθέτουν λογισμικό που τους παρέχει τη δυνατότητα να περιέχουν πληροφορίες και ταυτόχρονα να τις εκπέμπουν σε συσκευές που πραγματοποιούν ερωτήματα στο χώρο. Τέτοιες συσκευές είναι παρόμοιες με ένα τηλεχειριστήριο όπως βλέπουμε στο αριστερό τμήμα της εικόνας. Όταν κάποιος επισκέπτης εισέρχεται στο μουσείο, μπορεί να επιλέξει να έχει μαζί του κατά την ξενάγηση στο χώρο αυτή τη φορητή συσκευή που αναλαμβάνει το ρόλο του ηλεκτρονικού ταξιδιωτικού οδηγού. Κάθε φορά που ο επισκέπτης πλησιάζει ένα έκθεμα, βρίσκεται αντίστοιχα μέσα στην εμβέλεια ενός από τους προαναφερθέντες πομπούς που διαθέτουν πληροφορίες σχετικές με το αντίστοιχο έκθεμα.



Εικόνα 2-6 :Πειραματική διάταξη context-aware tour guide[2]

2.5.3 Αλληλεπιδραστικό κινητό πολυμέσων

Στην εφαρμογή αυτή στόχος είναι η εκμετάλλευση της τεχνολογίας για τη βελτίωση της επικοινωνίας μεταξύ των ανθρώπων. Στην περίπτωση της εφαρμογής αυτής η ιδέα έχει ως εξής: δεν αναφερόμαστε πλέον σε ένα τυπικό τηλέφωνο με πληροφορίες αλλά σε ένα “κοινό” άλμπουμ από πολυμεσικό περιεχόμενο. Μέσω δικτύου Bluetooth οι χρήστες θα μπορούν να διανέμουν το υλικό τους και να έχουν με τον τρόπο αυτό όλοι πρόσβαση στα ίδια δεδομένα που μπορεί να είναι φωτογραφίες, ήχοι, κείμενα, προσωπικά στοιχεία και ενδιαφέροντα του καθενός. Το κινητό επομένως θα διαθέτει μια εφαρμογή context-aware που μέσω κατάλληλων δεκτών θα συλλέγει τις πληροφορίες αυτές από όλες τις κοντινές συσκευές και θα δίνει τη δυνατότητα στα άτομα να ανακαλύπτουν τις κοινές συνιστάμενες τους. Έτσι τα άτομα θα έρχονται πιο κοντά καθώς θα ανακαλύπτουν τα κοινά τους ενδιαφέροντα και θα μπορούν να μοιραστούν τις ανησυχίες τους και τις σκέψεις τους.

Στην εικόνα 2-4 παρουσιάζεται μια τέτοια εφαρμογή που έχει αναπτυχθεί με την ονομασία m-Kavaad ενώ στην εικόνα 2-5 δείχνουμε πως αυτή χρησιμοποιείται πρακτικά. Πιο συγκεκριμένα στην εικόνα 2-4 παρατηρούμε την μορφή μιας εφαρμογής m-Kavaad. Διαφαίνεται ένα στιγμιότυπο από video στο μεγαλύτερο τμήμα της οθόνης ενώ στο κάτω τμήμα διακρίνουμε τα συνηθισμένα σύμβολα επεξεργασίας όπως το πλήκτρο αναπαραγωγή (play), πίσω(rewind) και μπροστά(forward) καθώς κι επιπλέον επιλογές που είναι tabs για τις ίδιες ή και επιπλέον επιλογές. Για παράδειγμα διακρίνουμε το tab play που αντιστοιχεί στο ανάλογο πλήκτρο αλλά και τα πλήκτρα σίγασης (tabs mute), που καταργεί τελείως τον ήχο και options που παρέχει επιπλέον

δυνατότητες στο χρήστη για την επεξεργασία του video όπως είναι το focus(εστίαση) και το send(αποστολή) για περιπτώσεις ανταλλαγών ροών πληροφορίας. Η εικόνα 2-5 δείχνει ακριβώς την τελευταία αυτή λειτουργία σε χρήση. Ένα σύνολο ατόμων που διαθέτουν την εφαρμογή αυτή μοιράζονται το ίδιο video μέσω της επιλογής αποστολή (send) .Το κεντρικό άτομο είναι ο διαχειριστής που μοιράζει το πολυμεσικό υλικό στους υπόλοιπους και έχει τη δυνατότητα να διοχετεύει συνεχώς πληροφορίες ασύρματα. Επί παραδείγματι μπορεί να επισημάνει μια λεπτομέρεια πάνω στο Video μέσω της εστίασης και αυτή να καταγραφεί και στις υπόλοιπες συσκευές.



Εικόνα 2-7 :Εφαρμογή m-Kavaad[3]



Εικόνα 2-8 :Χρήση context-aware multimedia mobile [3]

Για περισσότερες περιπτώσεις context-aware υπηρεσιών μπορείτε να μεταβείτε στην αναφορά [35].

ΚΕΦΑΛΑΙΟ 3

MOBILE VIDEO

3.1 Γενικά

Το 1908 ο Nuthan B. Stubblefield [4] δημοσίευσε την αμερικάνικη πατέντα με κωδικό 887357 που αρχικά δηλωνόταν με το όνομα “cave radio” ενώ πολύ μετά καθιερώθηκε ο όρος “κινητή τηλεφωνία”. Η ιδέα ήταν να διευκολύνει την επικοινωνία των χρηστών της τηλεφωνίας παρέχοντας τους τη δυνατότητα να χρησιμοποιούν το τηλέφωνο σε μια ευρεία περιοχή χωρίς να περιορίζονται από καλώδια. Το κυψελωτό σύστημα που στηρίζεται σήμερα η κινητή τηλεφωνία ανακαλύφθηκε το 1947 από μηχανικούς της Bell Labs [5] και εξελίχθηκε σημαντικά μέσα στη δεκαετία του '60. Από το σημείο αυτό και έπειτα η κινητή τηλεφωνία έχει διέλθει από πολλά στάδια για να φτάσει σήμερα να είναι αναπόσπαστο κομμάτι της καθημερινότητάς μας .

Ξεκινήσαμε από την zero γενιά (0G) το 1945 [6]. Η γενιά αυτή δεν ακολουθούσε το κυψελωτό σύστημα οργάνωσης, αναφερόταν μόνο σε φωνητική επικοινωνία και δεν υπήρχε η δυνατότητα να γίνονται δρομολογήσεις κλήσεως μεταξύ σταθμών βάσεων ή να προωθούνται οι συνομιλίες σε διαφορετικές συχνότητες. Το κινητό τηλέφωνο δεν έχει διεισδύσει ακόμα στο αγοραστικό κοινό.

Η πρώτη προσπάθεια καταναλωτικού κυψελωτού δικτύου έγινε στην Ιαπωνία το 1979 από τη NTN. Στη δεκαετία του '80 λοιπόν είναι που εισερχόμαστε στη πρώτη γενιά κινητής τηλεφωνίας (1G) [7] κι οποία αναφέρεται σε αναλογική επικοινωνία. Η αναλογική φύση όμως των κινητών τηλεφώνων δημιούργησε πολλά προβλήματα σταθερότητας καθώς το σήμα εξασθενούσε αρκετά γρήγορα, είχαμε έντονα θέματα παρεμβολών και επιπλέον τα ηλεκτρονικά συστήματα καταλάμβαναν αρκετά χώρο με αποτέλεσμα οι συσκευές να μην είναι απόλυτα εύχρηστες.

Τη λύση στα ζητήματα αυτά λάνσαρε για πρώτη φορά η εταιρεία Radiolinja στη Φιλανδία το 1991 όταν και εισήγαγε στην ψηφιακή τεχνολογία μέσω της

δεύτερης γενιάς κινητών (2G) [8]. Είναι η περίοδος που εισάγεται το πρότυπο GSM και το οποίο θα φέρει επανάσταση στην κινητή τηλεφωνία. Πλέον παρέχονται περισσότερες δυνατότητες όπως η ανταλλαγή γραπτών μηνυμάτων SMS αλλά και η υποστήριξη περισσότερων ταυτόχρονων χρηστών χωρίς τον κίνδυνο των παρεμβολών ή υποκλοπών.

Στη συνέχεια ακολούθησε η γενιά 2.5G που χαρακτηρίζεται από τη ζήτηση περισσότερων υπηρεσιών και μεγαλύτερων ρυθμών κυρίως για μεταφορά δεδομένων και παροχή προστιθέμενης αξίας (value-added) υπηρεσιών. Είναι ουσιαστικά η γενιά στην οποία βρισκόμαστε σήμερα καταναλωτικά καθώς η 3G που κυκλοφόρησε το 2001 πάλι από την Ιαπωνία δεν έχει ακόμα καταλάβει το αγοραστικό κοινό που στόχευε εξ αρχής. Στη 3G γενιά αναφερόμαστε πλέον όχι μόνο σε απλή ομιλία και γραπτά μηνύματα αλλά σε μια πληθώρα υπηρεσιών πολυμέσων (ringtones, γραφικά αναπαραγωγή μουσικών κομματιών, ταινιών και video, παιχνίδια), σύνδεση στο διαδίκτυο και ρυθμούς δεδομένων που αγγίζουν τις επιδόσεις ηλεκτρονικών υπολογιστών.

3.2 Mobile video

Όπως αναφέραμε παραπάνω μια από τις υπηρεσίες που προσφέρουν τα σύγχρονα κινητά είναι η αναπαραγωγή video ή αλλιώς αναπαραγωγή κινούμενης εικόνας .

Η εφαρμογή αυτή αναπτύχθηκε μέσα στο γενικότερο πλαίσιο προώθησης νέων υπηρεσιών από το κινητό τηλέφωνο. Κι ο λόγος ήταν απλός. Από τη στιγμή που αντιμετωπίστηκαν τα αρχικά προβλήματα του εύρους συχνοτήτων, των παρεμβολών, της εξασθένησης σήματος, των υποκλοπών και του περιορισμού των χρηστών, οι απαιτήσεις του καταναλωτικού κοινού πέρασαν σε άλλο επίπεδο. Καθώς διέθεταν μια συσκευή που την κουβαλούσαν μαζί τους, επιθυμούσαν να μπορεί να προσφέρει όλο και περισσότερα πέραν της επικοινωνίας. Επιθυμούσαν να τους ενημερώνει, να τους ψυχαγωγεί, να καταγράφει πληροφορίες, να τους υπενθυμίζει καταστάσεις, να τους προειδοποιεί. Στο πλαίσιο της ψυχαγωγίας περιλαμβάνεται το Mobile video. Το κινητό τηλέφωνο βελτιώνει την

υπολογιστική του ισχύ και αποκτάει μεγαλύτερη και υψηλότερης ανάλυσης οθόνη ώστε να είναι δυνατή η αναπαραγωγή video.

Αρχικά το video ήταν απλοϊκές κινούμενες εικόνες κατάλληλες μόνο για το κινητό .Η ευρεία αποδοχή όμως του καταναλωτικού κοινού αναγκάζει την τεχνολογία να προχωρήσει παραπέρα και να οδηγεί το κινητό να γίνεται μια μικρή προσομοίωση της τηλεόρασης και του υπολογιστή. Σήμερα λοιπόν μπορούν οι χρήστες να παρακολουθήσουν ταινίες υψηλής ευκρίνειας με ποιότητα ανάλογη με τον Η/Υ αλλά και τον κινηματογράφο. Οι δυνατότητες είναι άπειρες καθώς υποστηρίζονται πολλές video formats όπως 3GPP,MPEG-4,RTSP και Flash Lite και όλα αυτά με χαμηλό κόστος και κομψότητα στην υλοποίηση των συσκευών.

Η παρακάτω εικόνα (3-1) δείχνει ένα από τα πολλά κινητά τηλέφωνα που διαθέτει δυνατότητα video με δυνατότητα υψηλής ευκρίνειας και πολύ καλής ποιότητας ανάλογης ακόμα και με ηλεκτρονικό υπολογιστή.



Εικόνα 3-1 :Εφαρμογή mobile video[9]

3.3 Εφαρμογές mobile video

Το mobile video ξεκίνησε όπως αναφέραμε και παραπάνω για την κάλυψη της ανάγκης ψυχαγωγίας των χρηστών. Με την πάροδο του χρόνου όμως συνειδητοποιήθηκε πως μπορεί να προσφέρει κάτι παραπάνω από διασκέδαση. Στη σύγχρονη κοινωνία η νέα αυτή εφαρμογή είναι χρήσιμη σε πολλούς και διαφορετικούς τομείς.

3.3.1 M-learning

Καταρχήν, το mobile video μπορεί να βοηθήσει στην ανανέωση της εκπαίδευσης μαθητών και σπουδαστών. Μέχρι τώρα τα κινητά τηλέφωνα απαγορεύονταν στις διαλέξεις των μαθημάτων διότι αποσπούσαν την προσοχή των μαθητών. Έγιναν πολλές προσπάθειες για να καταργηθεί η χρήση τους, όμως όλες απέτυχαν αφού οι εκπαιδευτικοί δε μπορούν να σταματήσουν τη ροή των πραγμάτων και αυτή είναι η καθιέρωση του κινητού στην καθημερινότητα και των μαθητών. Για το λόγο αυτό, γεννήθηκε η ιδέα εκμετάλλευσης των κινητών προς όφελος της εκπαίδευσης με τη χρήση πολυμέσων. Η ιδέα αυτή, γνωστή και ως m-learning, έχει βρει πολλούς υποστηρικτές και θεωρείται το θεμέλιο για το νέο σύγχρονο κι αλληλεπιδραστικό σχολείο του μέλλοντος[10]. Για παράδειγμα στην παρακάτω εικόνα παρατηρούμε εφαρμογή Mobile video που συνδυάζεται και με κείμενο για να βοηθήσει τα παιδιά στην τάξη να μάθουν τα είδη ψαριών που υπάρχουν σε μια συγκεκριμένη περιοχή.

Στην εικόνα 3-2 βλέπουμε ένα χαρακτηριστικό παράδειγμα m-learning. Ο μαθητής έχει τη δυνατότητα να εισέλθει μέσω του κινητού του σε έναν ιστοχώρο που είναι οργανωμένος ως ηλεκτρονική εγκυκλοπαίδεια. Με την είσοδό του στην αρχική σελίδα (αριστερό τμήμα) του δίνεται η δυνατότητα να αναζητήσει την έννοια που τον απασχολεί μέσω του πεδίου search. Στο παράδειγμά μας γίνεται αναζήτηση για τη λέξη «trout» που σημαίνει «πέστροφα». Αμέσως η εφαρμογή φέρνει στο κινητό τηλέφωνο οποιαδήποτε πληροφορία διαθέτει στη βιβλιοθήκη της για το συγκεκριμένο ψάρι κι η οποία όπως βλέπουμε στη συγκεκριμένη περίπτωση είναι κείμενο, εικόνες και υπερσύνδεσμοι (links) που οδηγούν σε άλλα θέματα που ίσως κατά την ανάγνωσή τους θέλουν διευκρίνιση για το μαθητή.



Εικόνα 3-2 :Εφαρμογή m-learning [10]

3.3.2 Επαγγελματική κατάρτιση

Μια άλλη εφαρμογή του mobile video είναι η κατάρτιση εργαζομένων, κάτι ανάλογο με το m-learning για μαθητές και σπουδαστές. Όπως γνωρίζουμε, οι εργαζόμενοι στις εταιρείες, τις δημόσιες υπηρεσίες, τη βιομηχανία κλπ βρίσκονται συνεχώς σε μια διαδικασία εκμάθησης είτε λόγω καινούργιων καθηκόντων που αναλαμβάνουν είτε λόγω αλλαγών που υφίστανται οι θέσεις τους μέσα στην καθημερινότητα. Οι ίδιοι οφείλουν να ανταπεξέλθουν γρήγορα σε όποιες μεταβολές υπάρχουν ώστε να συνεχίσουν να είναι αποδοτικοί κι αποτελεσματικοί. Για να επιτευχθεί αυτό οι εργοδότες διαπίστωσαν πως η χρήση της τεχνολογίας και μάλιστα μιας προσιτής συσκευής όπως είναι το κινητό μπορεί να διευκολύνει κατά πολύ το έργο αυτό.[10]

3.3.3 Διαφήμιση

Η παραδοσιακή τηλεοπτική διαφήμιση έρχεται πλέον στο κινητό μας τηλέφωνο. Η νέα ιδέα του αμερικάνικου τομέα marketing βλέπει ένα μέλλον που το κινητό θα αναπαράγει διαφημίσεις και θα ενισχύει την προσπάθεια τόνωσης της αγοράς μέσω αυτού. Η ιδέα είναι αρκετά προχωρημένη αφού με μικρό κόστος οι έμποροι θα μπορούν να προσεγγίσουν πολλούς πελάτες οι οποίοι θα έχουν την αίσθηση της προσωπικής μεταχείρισης, αφού οι διαφημίσεις θα έρχονται στην προσωπική τους συσκευή κι όχι σε ένα πιο «απρόσωπο» μέσο επικοινωνίας όπως η τηλεόραση. Οι καταναλωτές αναπτύσσουν ένα αίσθημα εμπιστοσύνης και για αυτό είναι πιο αποτελεσματική η προσπάθεια προώθησης των προϊόντων. Μάλιστα οι έμποροι ενδιαφέρονται ιδιαίτερα για την προοπτική να χρησιμοποιήσουν και το σύστημα GPS με το οποίο είναι εφοδιασμένες πολλές κινητές συσκευές, για να στέλνουν τις διαφημίσεις των προϊόντων τους στους καταναλωτές ανάλογα με το μέρος στο οποίο βρίσκονται. Για παράδειγμα θα στέλνονται στα άτομα διαφημίσεις για προϊόντα των οποίων τα καταστήματα θα βρίσκονται στη γύρω περιοχή (proximity-based marketing). [37]

Στην εικόνα 3-3 παρουσιάζεται η καμπάνια που ακολουθεί η εταιρεία παρασκευής ουίσκι Jim Beam, η οποία είναι κι από τις πρώτες εταιρείες που έχουν αρχίσει να δραστηριοποιούνται στο τομέα της διαφήμισης μέσω κινητού. Στο χρήστη του κινητού εμφανίζεται οθόνη με τη μάρκα του προϊόντος, την ονομασία του καθώς και πληροφορίες που έχουν ως στόχο να διεγείρουν το ενδιαφέρον του.



Εικόνα 3-3 :Mobile διαφήμιση [11]

3.3.4 Video κινητού τηλεφώνου για έκτακτες ανάγκες (Emergency mobile video)

Σε αυτή την εφαρμογή φαίνεται ξεκάθαρα πόσο χρήσιμη μπορεί να αποβεί η τεχνολογία κάποιες φορές στον άνθρωπο. Οι καταστάσεις έκτακτης ανάγκης είναι πολλές καθημερινά είτε οφείλονται στον άνθρωπο είτε οφείλονται στη φύση. Ατυχήματα στο δρόμο, στο σπίτι και στη δουλειά, πυρκαγιές, σεισμοί, πλημμύρες ή αιφνίδιες ασθένειες είναι κάποιες από αυτές τις συνθήκες που μας πανικοβάλλουν και αναζητάμε αμέσως βοήθεια από ειδικούς όπως πυροσβέστες, γιατρούς κι αστυνομικούς. Δεν είναι λίγες οι φορές που η καθυστέρηση των αρμοδίων έχει στοιχίσει στους συνανθρώπους μας. Η ιδέα λοιπόν είναι να περιορίσουμε την εξάρτησή μας από άλλα άτομα και να αποκτήσουμε εμείς τη δυνατότητα να αντιμετωπίσουμε τις δύσκολες καταστάσεις που ίσως μας τύχουνε. Τα emergency mobile videos έχουν ακριβώς αυτή την προοπτική. [10]

Η εικόνα 3-4 παρουσιάζει ένα στιγμιότυπο ενός τέτοιου video που διανέμεται στις Ηνωμένες Πολιτείες της Αμερικής κι αναφέρεται σε τρόπους αντιμετώπισης σεισμών από τους πολίτες.



Εικόνα 3-4 :Emergency mobile video [12]

ΚΕΦΑΛΑΙΟ 4

CONTEXT-AWARE MOBILE VIDEO

4.1 Γενικά

Σήμερα, οι λύσεις για context-aware υπηρεσίες μέσω φορητών συσκευών (mobile) παρέχουν τη δυνατότητα να συλληφθούν δυναμικά και να χρησιμοποιηθούν πληροφορίες που θα βελτιώσουν, αλλάξουν ή δημιουργήσουν νέες συνθήκες στο τομέα των επικοινωνιών. Διάφορα ασύρματα δίκτυα πρέπει να χρησιμοποιηθούν στις περιπτώσεις αυτές. Τα πιο διαδεδομένα είναι τα ασύρματα LAN (WLAN), Wi-Fi, WiMAX, κυψελωτά ή GPS με συσκευές, ετικέτες (tags) ή αισθητήρες αντίστοιχα[30]. Τις περισσότερες φορές συσκευές, ετικέτες και αισθητήρες στέλνουν τι συλλεγμένες πληροφορίες μέσω WLAN και το κινητό με τη σειρά του επεξεργάζεται αυτά τα δεδομένα

4.2 Απαιτήσεις

Οι έννοιες των context-aware υπηρεσιών και του mobile video συνδυάζονται και προκύπτει η εφαρμογή context-aware mobile video, η υλοποίηση της οποίας θα μας απασχολήσει στην εν λόγω εργασία. Στην περίπτωση αυτή η αναπαραγωγή του mobile video θα ελέγχεται πλέον και από εξωτερικούς παράγοντες ή αλλιώς από το πλαίσιο του περιβάλλοντος μέσα στο οποίο κινείται το άτομο και όχι μόνο ρητά από τον ίδιο το χρήστη. Η ιδέα είναι να συλλέγονται πληροφορίες για την κατάσταση του χρήστη και τις συνθήκες του περιβάλλοντος και του χώρου ώστε να ρυθμίζεται αυτόματα η αναπαραγωγή του video στο κινητό. Η πρόκληση που αντιμετωπίζουν οι μηχανικοί σε αυτήν την ιδέα είναι η παροχή πραγματικού χρόνου (real-time) δεδομένων και μάλιστα αυτόματα στις κινητές συσκευές χωρίς την παρέμβαση του ατόμου. Μπορούμε να συνοψίσουμε τα θέματα που υπάρχουν[29]:

➤ Διαχείριση ζωνών περιοχής

Πρέπει να τμηματοποιούνται οι περιοχές για τις οποίες μας ενδιαφέρει να συλλέξουμε πληροφορίες σε ζώνες και να καταγράφονται οι εισοδοι και έξοδοι νέων χρηστών σε αυτές τις ζώνες.

➤ Εντοπισμός κινητών

Πρέπει να εντοπίζονται οι κινητές συσκευές σε ένα μεγάλο εύρος περιοχής και να είναι πάντα γνωστή η θέση τους.

➤ Εντοπισμός συνθηκών

Πρέπει να καταγράφουμε τις περιβαλλοντικές συνθήκες του χώρου στο οποίο κινούνται οι χρήστες, όπως θερμοκρασία δωματίου.

➤ Ενημέρωση

Πρέπει να γίνεται αυτόματα ανανέωση της κατάστασης των χρηστών και να παρέχεται πάντα τρόπος πρόσβασης στις διάφορες πληροφορίες που ανακτώνται.

4.3 Αντιμετώπιση

Για να μπορέσουν να ανταπεξέλθουν στις παραπάνω απαιτήσεις, οι μηχανικοί τηλεπικοινωνιών προσπάθησαν να επιβάλλουν κάποια χαρακτηριστικά στοιχεία στην αρχιτεκτονική των συστημάτων ώστε να επιτύχουν και την κινητικότητα (“mobility”) και την ενημερότητα περιεχομένου (context-awareness). Τα στοιχεία αυτά είναι τα εξής[13]:

➤ Αισθητήρες (sensors) για τη συλλογή των πληροφοριών του περιβάλλοντος και της κατάστασης του χρήστη.

➤ Εξυπηρετητές(Servers), όπου απαιτείται, για τη συλλογή και αποθήκευση πληροφοριών από διάφορους αισθητήρες σε ένα σημείο ώστε να γίνεται πιο εύκολη η παράδοσή τους στο κινητό όταν αυτό το απαιτεί.

➤ Ερμηνευτές(Interpreters) για την ερμηνεία των δεδομένων σε γλώσσα που να είναι κατανοητή από τις πλατφόρμες των κινητών τηλεφώνων.

➤ Mobile εφαρμογή. Στην περίπτωση μας αυτή είναι η αναπαραγωγή video.

4.3.1 Sensors

Οι αισθητήρες είναι συσκευές που μετράνε φυσικές ποσότητες και τις μετατρέπουν σε σήματα τα οποία μπορούν να διαβαστούν στη συνέχεια από έναν παρατηρητή ή ένα όργανο (πχ μέτρηση σημάτων από βολτόμετρο).

Οι αισθητήρες δεν είναι μια έννοια άπιαστη για το σύγχρονο άνθρωπο αλλά βρίσκονται μέσα στην καθημερινότητά μας. Ένα χαρακτηριστικό παράδειγμα αισθητήρα είναι ο γνωστός θερμοστάτης του καλοριφέρ που συναντάμε στα σπίτια μας. Στην περίπτωση αυτή ορίζουμε το θερμοστάτη σε μια συγκεκριμένη θερμοκρασία που επιθυμούμε να έχει το σπίτι. Το καλοριφέρ αρχίζει να λειτουργεί και μόλις ο εσωτερικός χώρος φτάσει στην επιθυμητή θερμοκρασία, ο θερμοστάτης το «αντιλαμβάνεται», διακόπτεται η λειτουργία του καλοριφέρ και σταματάει η καύση πετρελαίου. Αν στη συνέχεια η θερμοκρασία πέσει κάτω από την αρχικά ορισμένη, τότε ο θερμοστάτης το «αντιλαμβάνεται» και πάλι και το καλοριφέρ ξεκινάει. Με τη μέθοδο αυτή είναι δυνατό να ρυθμίζουμε ακριβώς τη θερμοκρασία στο σπίτι μας κάνοντας οικονομία στους πόρους πετρελαίου. Υπάρχουν χιλιάδες παραδείγματα αισθητήρων στα αυτοκίνητα, στα μηχανήματα οικιακής χρήσης, στη βιομηχανία, στη φαρμακευτική και σε άλλους κλάδους που ο άνθρωπος δεν αντιλαμβάνεται την ύπαρξή τους όμως τους χρησιμοποιεί συνεχώς.

Υπάρχουν πολλά διαφορετικά είδη αισθητήρων σήμερα και συνεχώς αναπτύσσονται καινούργια. Οι εφαρμογές τους και οι ανάγκες που καλύπτουν μπορεί να είναι πολύ διαφορετικές, όλοι όμως πρέπει να πληρούν τις εξής προδιαγραφές για να είναι λειτουργικοί και αποτελεσματικοί [31]:

- Σημαντική ευαισθησία (sensitivity) , το χαρακτηριστικό που υποδηλώνει πόσο πολύ μεταβάλλεται η έξοδος του αισθητήρα με τη μεταβολή της μετρούμενης ποσότητας. Για παράδειγμα ο υδράργυρος στο θερμόμετρο υψώνεται κατά 1 cm όταν η θερμοκρασία αυξάνεται κατά 1 °C, που σημαίνει ότι χαρακτηρίζεται από ευαισθησία 1 cm/°C. Είναι προφανές πως όσο πιο μικρές μεταβολές καλείται να μετρήσει ο αισθητήρας, τόσο πιο μεγάλη πρέπει να είναι η ευαισθησία του.
- Σημαντική μη ευαισθησία (insensitivity) σε οποιαδήποτε άλλο αίτιο παρουσιάζεται στο περιβάλλον και δεν είναι αρμόδιος ο συγκεκριμένος αισθητήρας για την καταγραφή του.
- Μη επίδραση του αισθητήρα πάνω στη μετρούμενη ποσότητα. Πρέπει να καταγράφει τις μεταβολές χωρίς να αλλοιώνει την πληροφορία.

Οι τύποι αισθητήρων που χρησιμοποιούνται σήμερα είναι οι εξής [32]:

- Θερμικοί
 - Αισθητήρες θερμοκρασίας
 - Αισθητήρες θερμότητας

- Ηλεκτρομαγνητικοί
 - Αισθητήρες ηλεκτρικής αντίστασης πχ ωμόμετρο και πολύμετρο
 - Αισθητήρες ηλεκτρικού ρεύματος πχ γαλβανόμετρο
 - Αισθητήρες ηλεκτρικής τάσης πχ βολτόμετρο
 - Αισθητήρες ηλεκτρικής ισχύος
 - Αισθητήρες μαγνητικού πεδίου πχ πυξίδα
 - Ανιχνευτές μετάλλων
 - Radar

- Μηχανικοί
 - Αισθητήρες πίεσης πχ βαρόμετρο
 - Αισθητήρες ροής υγρών και αερίων πχ ανεμόμετρο
 - Αισθητήρες πυκνότητες υγρών και αερίων
 - Μηχανικοί αισθητήρες πχ αισθητήρας επιτάχυνσης
 - Αισθητήρες υγρασίας πχ υγρόμετρο
 - Αισθητήρες δόνησης

- Χημικοί
 - Αισθητήρες χημικών ποσοτήτων πχ αισθητήρες οξυγόνου και μονοξειδίου του άνθρακα
 - Αισθητήρες οσμών

- Οπτικής ακτινοβολίας
 - Αισθητήρες φωτός πχ φωτοκυψέλες , φωτοдиодοι και φωτοτρανζίστορ
 - Αισθητήρες υπέρυθρης ακτινοβολίας
 - Αισθητήρες οπτικών ινών

- Ακτινοβολίας ιονισμού
 - Αισθητήρες ακτινοβολίας πχ ανίχνευση νετρονίων
 - Αισθητήρες υποατομικών σωματιδίων

- Ακουστικοί
 - Ακουστικοί αισθητήρες υποσυχνοτήτων πχ Sonar
 - Αισθητήρες ήχου πχ μικρόφωνο

- Κίνησης πχ ταχύμετρο



Εικόνα 4-1 :Είδη ταχύμετρο [21]

- Προσανατολισμού πχ γυροσκόπιο
- Απόστασης



Εικόνα 4-2 :Αισθητήρας απόστασης και προσανατολισμού [23]

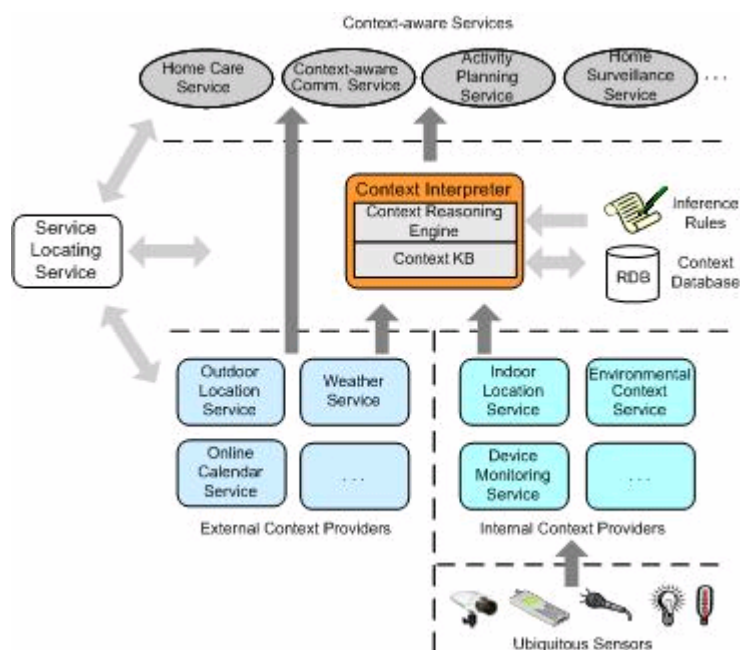
4.3.2 Εξυπηρετητές Περιεχομένου (Context-server)

Σκοπός των servers αυτών είναι ο διαχωρισμός των πληροφοριών. Κατά τη λειτουργία τους συγκεντρώνουν όλο το περιεχόμενο για μια συγκεκριμένη οντότητα που μπορεί να είναι ένα φυσικό πρόσωπο. Ουσιαστικά η θέση τους είναι για να διευκολύνουν τη δουλειά του προγραμματιστή της εφαρμογής. Η εφαρμογή επικοινωνεί μόνο με το server που έχει οριστεί ως ο προσωπικός server περιεχομένου του εκάστοτε ατόμου. Όταν ο μηχανικός δημιουργήσει νέο server, η μόνη του υποχρέωση είναι να καταγράψει τα ονόματα των αισθητήρων στους οποίους θα είναι εγγεγραμμένος και θα επικοινωνεί καθώς επίσης και διάφορες παραμέτρους όπως είναι το χρονικό διάστημα μέσα στο οποίο θα απευθύνει ο server ερώτηση τους αισθητήρες καθώς και το επιτρεπόμενο μέγεθος ή format της πληροφορίας που μπορεί να δεχτεί κάθε φορά[13].

4.3.3 Ερμηνευτές περιεχομένου (Context-interpreter)

Οι ερμηνευτές περιεχομένου είναι υπεύθυνοι για την εφαρμογή του διαχωρισμού εννοιών. Μέχρι τώρα η ερμηνεία του περιεχομένου πραγματοποιούνταν από την ίδια την εφαρμογή. Απαλλάσσοντας τις εφαρμογές από αυτή τη διαδικασία, καταλήγουμε στο να υπάρχει συνεργασία των Interpreters με πολλές εφαρμογές ταυτόχρονα. Όταν λοιπόν λαμβάνει ο ερμηνευτής κάποια πληροφορία, την επεξεργάζεται και είτε τη μετατρέπει σε άλλη μορφή είτε την διευκρινίζει[13].

Ένα απλό παράδειγμα είναι όταν ο context interpreter λαμβάνει πληροφορία για τοποθεσία δωματίου (πχ το άτομο βρίσκεται στο δωμάτιο 35) και τη μετατρέπει σε πληροφορία για την τοποθεσία του κτηρίου (το άτομο βρίσκεται στο κτήριο Α). Ένα άλλο, πιο σύνθετο παράδειγμα για να κατανοήσουμε τη λειτουργία του ερμηνευτή περιεχομένου είναι όταν αυτός λαμβάνει δεδομένα για την τοποθεσία, την ημερομηνία και το θόρυβο του περιβάλλοντος χώρου. Στην περίπτωση αυτή έχει την ικανότητα να συνδυάσει τις διαφορετικές πληροφορίες και να αντιληφθεί μια συγκεκριμένη κατάσταση του χρήστη όπως για παράδειγμα θα είναι η παρουσία του σε μία συνάντηση. Η πολυπλοκότητα ή η απλότητα τους εναπόκειται στις ιδέες του μηχανικού που τους υλοποιεί.

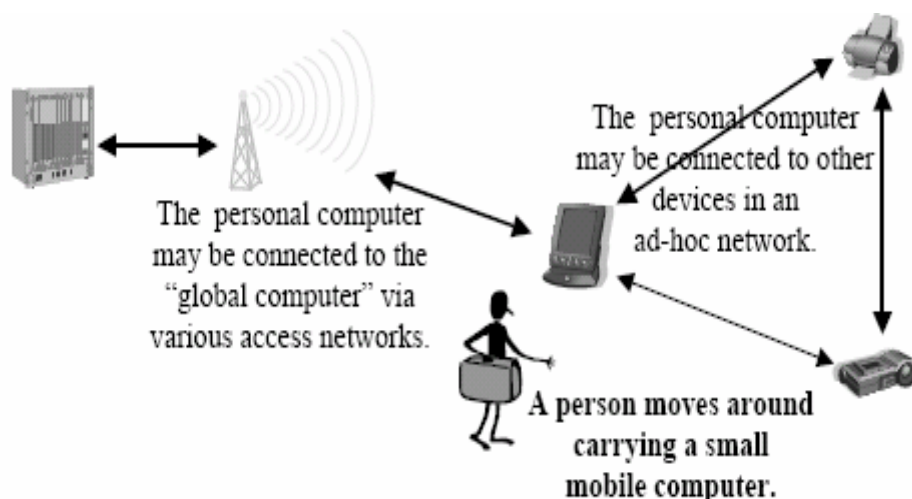


Εικόνα 4-3 :Παράδειγμα δομής context interpreter[17]

Να τονίσουμε βέβαια ότι η αρχιτεκτονική μπορεί να είναι κοινή τις περισσότερες φορές αλλά σε καμία περίπτωση δεν είναι επακριβώς ίδια. Υπάρχει μια πληθώρα από πλατφόρμες, interpreters, sensors και περιεχόμενο. Ανάλογα με τις ανάγκες της εφαρμογής χρησιμοποιούμε και την κατάλληλη αρχιτεκτονική. Όπως αναφέραμε και στην παράγραφο 2.3 ως περιεχόμενο ορίζουμε το σύνολο των πληροφοριών του περιβάλλοντος και για παράδειγμα στο περιεχόμενο αυτό μπορεί να περιλαμβάνεται η θερμοκρασία του περιβάλλοντος, η υγρασία, η φωτεινότητα, η κίνηση του χρήστη, η ώρα της ημέρας και η τοποθεσία. Ξεχωριστοί αισθητήρες θα πρέπει να συνεργάζονται για τη συλλογή των δεδομένων και στη συνέχεια θα πρέπει να απευθύνονται σε συγκεκριμένους interpreters ανάλογα με τον κώδικα που χρησιμοποιεί η συσκευή (πχ java , c++).

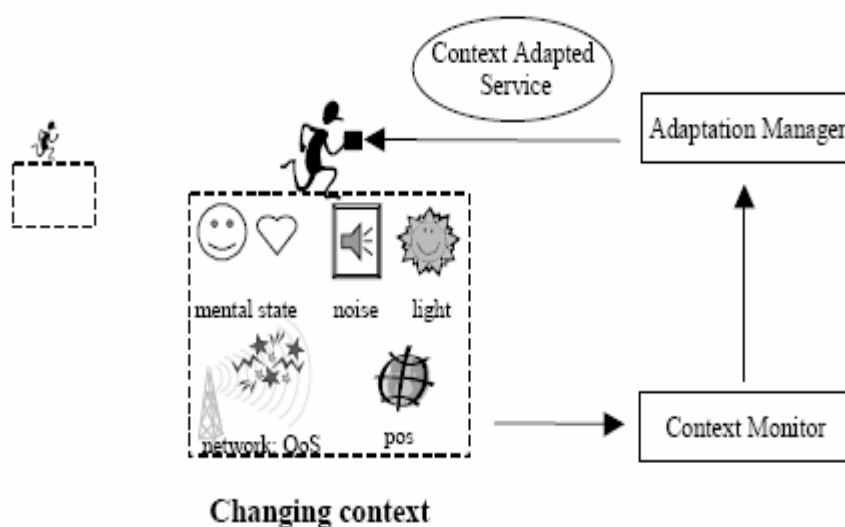
Οι δύο επόμενες εικόνες παρουσιάζουν διαγραμματικά την έννοια της κινητικότητας ενός χρήστη καθώς και την εξυπηρέτηση του κινητού αυτού χρήστη από τη context-aware αρχιτεκτονική που μόλις αναπτύξαμε.

Πιο συγκεκριμένα η εικόνα 4-3 παρουσιάζει ένα χρήστη ο οποίος αναλόγως με το χώρο που βρίσκεται έχει πάντα τη δυνατότητα να συνδέει τον προσωπικό του υπολογιστή με ασύρματους δρομολογητές (router), με κεραίες της κυψελωτής τηλεφωνίας ή και με συσκευές όπως εκτυπωτές ή κινητά τηλέφωνα. Ουσιαστικά έχει σχεδιαστεί η ανάγκη του χρήστη να εντοπίζεται η θέση του και τα υπάρχοντα στοιχεία στο χώρο κι αναλόγως να μπορεί να εκμεταλλευτεί τις δυνατότητες που του παρέχονται κάθε φορά.



Εικόνα 4-4 :Κινητικότητα ενός χρήστη mobile video που συνδέεται συνεχώς με νέα δίκτυα και νέες πληροφορίες [14]

Η εικόνα 4-4 καταγράφει διαγραμματικά την έννοια των context-aware υπηρεσιών που μόλις αναφέραμε. Έχουμε ένα χρήστη ο οποίος διαθέτει μια ένα κινητό τηλέφωνο στο οποίο είναι ενσωματωμένη μια πλατφόρμα για ανάλυση «περιεχομένου». Η συσκευή συνδέεται με ειδικούς αισθητήρες που καταγράφουν τη διανοητική κατάσταση του χρήστη (mental state), το θόρυβο (noise), το φωτισμό (light), τη θέση του στο χώρο (pos) καθώς και την ποιότητα σήματος του τηλεφώνου (QoS). Οι τιμές των μεγεθών αυτών παρουσιάζονται στην οθόνη του κινητού (context-monitor) και στη συνέχεια περνάνε σε μια πλατφόρμα διαχείρισης (Adaptation Manager). Στο συγκεκριμένο σημείο, μέσω κάποιων καθορισμένων αλγορίθμων, αποφασίζεται η γενικότερη κατάσταση του περιβάλλοντος και του χρήστη και δίνεται εντολή στο κινητό τηλέφωνο να προσφέρει μια προσαρμοσμένη υπηρεσία περιεχομένου (Context Adapted Service). Έτσι το κινητό τηλέφωνο φαίνεται να παίρνει αυτόματα πρωτοβουλία να μεταβάλλει τη λειτουργία του για να εξυπηρετήσει τις εκάστοτε συνθήκες.



Εικόνα 4-5 :Συνεχής αλλαγή του περιεχομένου (content) και ανάγκη προσαρμογής του video [14]

ΚΕΦΑΛΑΙΟ 5

ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ

5.1 Γενικά

Στόχος της παρούσας εργασίας είναι η υλοποίηση μιας εφαρμογής mobile video που να υιοθετεί τα χαρακτηριστικά γνωρίσματα του context-awareness. Πιο συγκεκριμένα επιθυμούμε να κάνουμε την αναπαραγωγή του mobile video αλληλεπιδραστική και δυναμική ως προς το χρήστη χωρίς όμως να απαιτείται η προσωπική του παρέμβαση. Το video θα πρέπει να προσαρμόζεται στις επιθυμίες του χρήστη που θα διαμορφώνονται είτε από τον ίδιο είτε από εξωτερικούς παράγοντες που δε θα μπορεί να ελέγξει απόλυτα. Η προσαρμογή της λειτουργίας θα πρέπει να είναι τέτοια ώστε να εξυπηρετεί το χρήστη που έχει επιλέξει να παρακολουθήσει κάτι συγκεκριμένο στο κινητό του τηλέφωνο. Τα πάντα λοιπόν πρέπει να γίνονται αυτόματα αναγνωρίζοντας συγκεκριμένες συνθήκες .

Για να ολοκληρωθεί αυτή η εφαρμογή θα πρέπει να διαθέτουμε κινητό τηλέφωνο που θα έχει τη δυνατότητα αναπαραγωγής video και ένα σύνολο από αισθητήρες που θα συλλέγουν δεδομένα για να αποτυπώσουν τον περιβάλλοντα χώρο. Τα δεδομένα που θα συλλέγονται με μια συγκεκριμένη μορφή καθώς και τα μεταδεδομένα του video θα μετατρέπονται σε μορφή αναγνώσιμη από μηχανή κανόνων και τέλος θα μεταδίδονται στο κινητό του χρήστη. Θα υλοποιήσουμε τη πλατφόρμα αναπαραγωγής video σε mobile java (J2ME) και θα ορίσουμε τους κανόνες που θα αποφασίζουν την κατάσταση του χρήστη και τις ενέργειες που θα λάβουν χώρα με τη βοήθεια γλώσσας Prolog που θα επικοινωνεί με τη Java ενώ θα μας απασχολήσουν και 3 ειδών αισθητήρες: αισθητήρες κίνησης, ήχου και φωτεινότητας.

Στη συνέχεια περιγράψουμε αναλυτικά την αρχιτεκτονική της εφαρμογής που κατασκευάσαμε και τα σενάρια χρήσης της.

5.2 Σενάρια χρήσης

Στην παράγραφο αυτή θα καταγράψουμε τα σενάρια χρήσης για την εφαρμογή μας. Λέγοντας σενάρια χρήσης αναφερόμαστε στις λειτουργίες που επιθυμούμε να εκτελούνται στο video κατά τη διάρκεια αναπαραγωγής του στο κινητό τηλέφωνο, στα στοιχεία που επιθυμούμε να συλλέγονται από τους αισθητήρες ώστε να κατηγοριοποιούνται οι περιβαλλοντικές συνθήκες καθώς και στον ορισμό των καταστάσεων στις οποίες ενδέχεται να βρίσκεται ο χρήστης.

5.2.1 Λειτουργίες

Οι λειτουργίες που θα πραγματοποιούνται στο video είναι οι εξής τέσσερις:

- ✓ Εστίαση (Zoom in & out)
- ✓ Απόκρυψη εικόνας (Hide image)
- ✓ Παύση ήχου (Mute)
- ✓ Αποκοπή εικόνας (Crop)

Οι λειτουργίες αυτές θα πραγματοποιούνται με βάση κάποιες τιμές (values) που θα ορίζουμε μέσα στη java πλατφόρμα και θα αντιστοιχούν στις τιμές που λαμβάνουμε από τους αισθητήρες περιβάλλοντος για την *κίνηση*, την *φωτεινότητα* και τον *ήχο του γύρω χώρου*. Η λήψη των στοιχείων των αισθητήρων θα πραγματοποιείται μέσω μιας rule engine η οποία θα τρέχει συγκεκριμένους κανόνες για να αντιλαμβανόμαστε ανά τακτά χρονικά διαστήματα την κατάσταση του περιβάλλοντος. Η μηχανή κανόνων που θα χρησιμοποιήσουμε είναι η JIProlog και τα εκάστοτε ερωτήματα για τα δεδομένα θα ενεργοποιούνται από τη java.

Όσον αφορά τις λειτουργίες θα κάνουμε μια σύντομη περιγραφή για το πώς θα υλοποιηθούν στο περιβάλλον της Java.

- ✓ Zoom in & out: Αλλάζουν οι διαστάσεις του video δίνοντας τη δυνατότητα στο χρήστη να ζουμάρει και να κινηθεί μέσα στο video ώστε να παρατηρήσει περισσότερες λεπτομέρειες που ίσως τον ενδιαφέρουν.
- ✓ Hide image: Θα αποκρύπτουμε την εικόνα του Video. Μια περίπτωση θα είναι όταν θα επιθυμεί ο χρήστης να διαβάσει για παράδειγμα ένα κείμενο ενώ το Video να συνεχίζει να παίζει κανονικά (για παράδειγμα σε ένα μουσικό video).
- ✓ Mute: Θα αφαιρέσουμε τον ήχο από το Video και ο Player θα είναι αθόρυβος. Αυτό μπορεί να είναι χρήσιμο σε καταστάσεις που ο ήχος είναι ενοχλητικός ή δύσκολο να γίνει αντιληπτός από το χρήστη, οπότε κι αφαιρείται.
- ✓ Crop: Η λειτουργία αυτή είναι κάπως παρόμοια με το zoom in καθώς δίνει έμφαση σε συγκεκριμένη περιοχή του Video. Η διαφορά εδώ είναι ότι τονίζεται μια συγκεκριμένη περιοχή τις οποίας οι συντεταγμένες θα ορίζονται από τα μεταδεδομένα του Video ενώ το μέγεθος του video θα παραμένει ως έχει.

5.2.2 Αισθητήρες και καταστάσεις χρήστη

Όσον αφορά τώρα τα στοιχεία των αισθητήρων, θα θέσουμε ένα εύρος τιμών (0-100) για κάθε ένα από τα τρία προαναφερθέντα στοιχεία του περιβάλλοντος χώρου. Θα παρακολουθούμε τις τιμές καθώς και τις μεταβολές των στοιχείων αυτών και θα ορίσουμε τα κριτήρια (μέσω της plog) με βάση τα οποία θα λαμβάνουμε τις διάφορες αποφάσεις για την κατάσταση του χρήστη κι κατά συνέπεια για τις ενέργειες που θα πραγματοποιούνται στο player (JVM).

Στη συνέχεια παραθέτουμε τα δεδομένα που θα λαμβάνουμε από τους αισθητήρες καθώς και τις καταστάσεις χρήστη που θα ορίσουμε στο μοντέλο μας με βάση τις διάφορες τιμές.

Κίνηση

0-50: Ο χρήστης θεωρείται ακίνητος ή με αργή κίνηση (*slow state*)

50-100: Ο χρήστης θεωρείται ότι κινείται γρήγορα. Θα ορίσουμε σε αυτή την περίπτωση ότι πιθανόν δεν είναι εστιασμένη η προσοχή του στο video και θα αποκρύπτουμε την εικόνα του player (*fast state*)

Φωτεινότητα

0-50: Στην κατάσταση χαμηλής φωτεινότητας θεωρούμε πως ο χρήστης είναι σε εσωτερικό χώρο και επομένως το video πρέπει να είναι διακριτικό. Γίνεται η λειτουργία zoom out, όχι όμως περισσότερο από το $\frac{1}{4}$ του αρχικού πλαισίου του video για να είμαστε σε θέση να μη χάνουμε στοιχεία του περιεχομένου (*low state*)

50-100: Στην κατάσταση υψηλής φωτεινότητας μπορούμε να πραγματοποιήσουμε zoom in ώστε να είναι τα αντικείμενα πιο ευδιάκριτα στο χρήστη (*high state*). Πρέπει βέβαια στο σημείο αυτό να τονίσουμε πως η λειτουργία του zoom in θα πραγματοποιείται μέσα στα πλαίσια της οθόνης του κινητού. Αυτό σημαίνει ότι θα υπάρχει περιορισμός να μην αυξάνεται αλόγιστα ώστε να φτάσουμε στο σημείο να μην αντιλαμβανόμαστε το περιεχόμενο.

Ήχος

0-50: Θεωρείται ότι υπάρχει συνομιλία στο χώρο και το video γίνεται mute. (*quiet state*)

50-100: Θεωρείται ότι υπάρχει έντονος θόρυβος στο χώρο (πιθανώς να βρίσκεται στο εξωτερικό χώρο) κι επομένως η ένταση του ήχου επανέρχεται. (*loud state*)

Συνοπτικά οι κανόνες που θα ορίσουμε θα είναι οι εξής:

✓ Όταν θα βρισκόμαστε σε quiet και slow state ταυτόχρονα θα θεωρούμε ότι βρισκόμαστε σε εσωτερικό χώρο και συμμετέχουμε σε μια συνομιλία οπότε το video θα γίνεται mute.

action (mute):- sound(quiet), motion(slow).

✓ Μόλις η ένταση του ήχου αυξηθεί θα θεωρούμε ότι δε συμμετέχουμε πλέον σε συνομιλία οπότε το video θα μπορεί να ξαναποκτήσει ήχο.

action(restart) :- sound(loud).

✓ Στην περίπτωση που η φωτεινότητα είναι υψηλή, θα θεωρούμε ότι βρισκόμαστε σε εξωτερικό χώρο και επομένως θα πρέπει να ζουμάρουμε στο video ώστε να αντιληφθούμε το περιεχόμενο.

action(zoomIn) :- luminance(high).

✓ Στην περίπτωση που η φωτεινότητα είναι χαμηλή, θα θεωρούμε ότι βρισκόμαστε σε εσωτερικό χώρο και επομένως θα προχωράμε σε zoom out για να είναι διακριτικό το video.

action(zoomOut) :- luminance(low).

✓ Στην περίπτωση που βρισκόμαστε στη fast και loud state ταυτόχρονα, θεωρούμε πως είμαστε σε εξωτερικό χώρο και κινούμαστε πολύ γρήγορα ώστε να μας αφορά το video. Για το λόγο αυτό αποκρύπτουμε την εικόνα.

action (hide):- motion(fast), sound(loud).

✓ Μόλις βρεθούμε στη slow state, θεωρούμε πως πλέον είμαστε σε θέση να παρακολουθήσουμε το video κι άρα προχωράμε στη λειτουργία appear.

action(appear) :- motion(slow).

✓ Τέλος ,όσον αφορά το crop πρέπει να αναφέρουμε εδώ ότι δε θα επηρεάζεται από τις περιβαλλοντικές συνθήκες. Η επιλογή του θα στηρίζεται μόνο στα metadata του video σε σχέση με το περιεχόμενό του σε κάποιο τομέα (frame) όπως θα παρουσιάσουμε αναλυτικότερα στο κεφάλαιο 6.

action(crop,VS) :- event(talking,VS).

5.2.3 Μεταδεδομένα (metadata)

Τα μεταδεδομένα είναι πληροφορίες για τα δεδομένα κάθε πολυμεσικού περιεχομένου (media). Η χρησιμοποίησή τους τα τελευταία χρόνια είναι αρκετά διαδεδομένη καθώς διευκρινίζουν σημαντικά στοιχεία του video, των εικόνων, των ήχων ή των κειμένων που δε θα μπορούσαμε να γνωρίζουμε διαφορετικά. Τα metadata που θα συνοδεύουν το video μας σε μορφή MPEG-7, και θεωρούμε ότι θα εισάγονται στη μηχανή κανόνων μαζί με τις πληροφορίες των αισθητήρων, θα χωρίζονται σε 3 κατηγορίες:

➤ Περιγραφικά ή μεταδεδομένα περιεχομένου (*Descriptive or Content metadata*):

Είναι πληροφορίες για το αντικείμενο που συλλαμβάνεται στην εικόνα όπως το όνομα του αντικειμένου, ο τίτλος, τα υλικά, οι ημερομηνίες και η φυσική περιγραφή.

Τα content metadata είναι πολύ σημαντικά, δεδομένου ότι είναι ο κύριος τρόπος με τον οποίο οι άνθρωποι θα ψάξουν και θα ανακτήσουν τις εικόνες από μια βάση δεδομένων.

➤ Τεχνικά μεταδεδομένα (*technical metadata*):

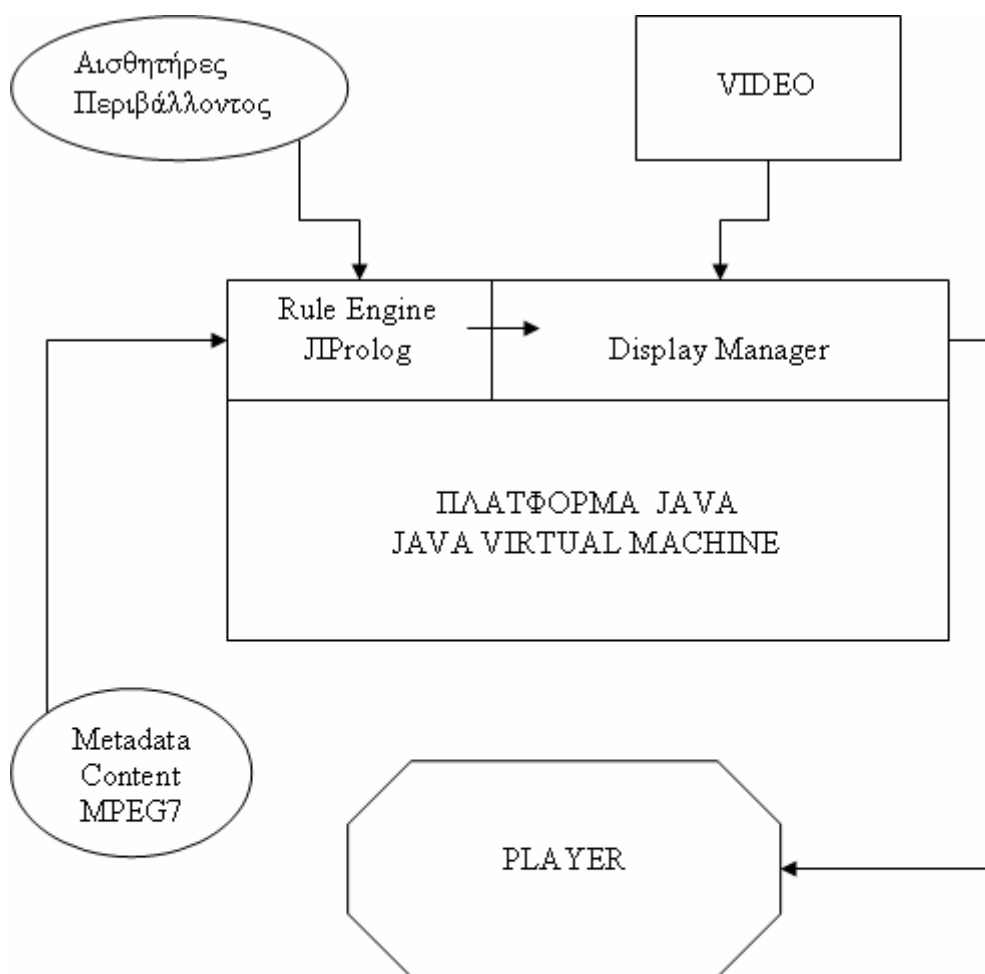
Τα τεχνικά μεταδεδομένα είναι στοιχεία για η ίδια την εικόνα και όχι για το αντικείμενο στην εικόνα. Περιλαμβάνει πληροφορίες για τις τεχνικές διαδικασίες που χρησιμοποιούνται στη σύλληψη και το χειρισμό της εικόνας, το χρώμα, τις μορφές αρχείου κ.λπ.

➤ Μεταδεδομένα διαχείρισης (*administrative metadata*):

Είναι τα μεταδεδομένα που σχετίζονται με τα δικαιώματα διαχείρισης του video.

5.3 Αρχιτεκτονική

Η αρχιτεκτονική της εφαρμογής που υλοποιήσαμε παρουσιάζεται διαγραμματικά στο επόμενο σχήμα και επεξηγείται αναλυτικά στην συνέχεια



Εικόνα 5-1 :Αρχιτεκτονική εφαρμογής.

Όπως παρατηρούμε και στο σχήμα, η εφαρμογή μας στηρίζεται στην αντικειμενοστραφή γλώσσα Java και συγκεκριμένα στην υποκατηγορία J2ME που είναι ένα σύνολο κλάσεων και μεθόδων της Java τα οποία έχουν προσαρμοστεί κατάλληλα ώστε να μπορούν να υποστηρίζονται από κινητές συσκευές όπως κινητά τηλέφωνα και PDA. Σε αυτή τη γλώσσα λοιπόν ξεκινήσαμε να χτίζουμε την εφαρμογή μας, με πρώτη στάση τη δημιουργία των λειτουργιών που απαιτούσαμε να πραγματοποιούνται αυτόματα στο video με βάση τα σενάρια χρήσης που περιγράψαμε στην αμέσως παραπάνω παράγραφο. Κατασκευάσαμε λοιπόν 4 διαφορετικές κλάσεις που θα επιτελούσαν τις 4 διαφορετικές λειτουργίες (Zoom in & out, Hide image, Mute και Crop)

Στη συνέχεια περάσαμε στην εφαρμογή αυτών των κλάσεων για την αναπαραγωγή του video μέσω του κινητού τηλεφώνου. Για να γίνει αυτό έπρεπε να κατευθυνθούμε στο τρόπο διαχείρισης των media μέσω mobile. Γνωρίζουμε πως το κινητό είναι μια πολύπλοκη και πολυσύνθετη πλατφόρμα που οτιδήποτε υποστηρίζει αποτυπώνεται σε εικόνα μέσω της οθόνης (και κάποιες φορές σε ήχο). Η οθόνη είναι η διεπαφή μέσω της οποίας ο χρήστης μπορεί να αντιληφθεί τη σωστή ή λανθασμένη λειτουργία και για αυτό το λόγο έπρεπε να υπάρχει ειδικός τρόπος χειρισμού των media μέσω αυτής. Ο χειρισμός αυτός πραγματοποιείται μέσω του διαχειριστή οθόνης (display manager). Εμείς καθορίσαμε τον τρόπο με τον οποίο ο display manager θα μπορούσε να διαχειριστεί τις κατασκευασμένες λειτουργίες ώστε να έχουμε το επιθυμητό αποτέλεσμα στην οθόνη. Το video που θα υποστεί την επεξεργασία θεωρούμε ότι είναι αποθηκευμένο στη συσκευή πιθανώς ύστερα από λήψη του μέσω δικτύου (Bluetooth από άλλο χρήστη ή κάποιο site στο internet που μπόρεσε να πλοηγηθεί το άτομο χάρη στις ιδιότητες 3G που διαθέτει το κινητό). Στην εφαρμογή μας δεν μας απασχολεί ο τρόπος απόκτησης του video παρά μας ενδιαφέρει μόνο το γεγονός ότι μπορούμε να το ανασύρουμε από το κινητό μας τηλέφωνο ανά πάσα στιγμή. Το video λοιπόν είναι το αντικείμενο που διαχειριζόμαστε με το display manager.

Το επόμενο βήμα προήλθε από την ιδέα των σεναρίων χρήσης. Όπως προαναφέραμε, στόχος μας ήταν η αυτόματη εκτέλεση των λειτουργιών στο video όταν κάποια σενάρια θα ικανοποιούνταν με βάση τις συνθήκες περιβάλλοντος και χρήστη. Έπρεπε επομένως να βρεθεί ένας τρόπος να αποφασίζεται πότε ο display manager θα χειριζότανε ποικιλοτρόπως το video χωρίς φυσικά να απαιτείται η παρέμβαση του χρήστη. Στο σημείο αυτό λοιπόν οδηγηθήκαμε στην υλοποίηση μιας μηχανής κανόνων (rule engine). Ως rule engine ορίζουμε ένα σύστημα λογισμικού που εκτελεί ένα ή και περισσότερους κανόνες σε πραγματικού χρόνου (runtime) περιβάλλον και οι κανόνες αυτοί πρέπει να «υπακούουν» «νόμιμες» μεθόδους. Έχοντας ορίσει κάποια δεδομένα και γεγονότα σε αυτές (facts), μπορούμε να αντιλαμβανόμαστε αυτόματα για τις καταστάσεις που επικρατούν εκτελώντας τους κανόνες. Η δυνατότητα μάλιστα συνδυασμού πολλών κανόνων οδηγεί στην αποτύπωση περισσότερο πολύπλοκων καταστάσεων και συνθηκών. Τελικά η ιδέα ήταν πως η java θα εκτελεί συνεχώς ερωτήματα(queries) στη μηχανή κατά τη διάρκεια αναπαραγωγής του video, οι κανόνες θα εκτελούνται, οι αποφάσεις περί της κατάστασης του χρήστη θα λαμβάνονται και εν τέλει οι λειτουργίες της java θα εφαρμόζονται με τη βοήθεια του display manager.

Αφού λοιπόν αποφασίσαμε ότι η ύπαρξη ενός τέτοιου λογισμικού είναι απαραίτητη, έπρεπε να αποφασίσουμε ποιο θα ταίριαζε στο δικό μας σενάριο. Η μηχανή κανόνων που θα χρησιμοποιούσαμε στην εφαρμογή μας θα έπρεπε να είναι συμβατή με τη java ώστε να είναι δυνατή η επικοινωνία με την κύρια πλατφόρμα που θα χειριζότανε το video. Υπάρχουν πολλές rule engine για java. Μερικές από αυτές που είναι Drools, SweetRules, TermWare, Mandarax, JLisa, OrneRules, Zilonis, Open Lexicon, JEOPS, JRuleEngine, Prova language και JIProlog. Από όλες αυτές τις μηχανές, επιλέξαμε να χρησιμοποιήσουμε τη JIProlog καθώς είναι εύκολη στην εγκατάστασή της, αρκετά σταθερή χωρίς πολλά προβλήματα και επιπλέον είναι συμβατή όχι απλά με τη java αλλά και με τη mobile java που μας απασχολεί.[28]

Τέλος έπρεπε να ορίσουμε την πηγή δεδομένων της JIProlog και τη μορφή των στοιχείων εισόδου έτσι ώστε να είναι κατανοητά κι αναγνώσιμα από τη μηχανή για την εκτέλεση των κανόνων. Τα δεδομένα που θα περνούν θα ήταν φυσικά τα στοιχεία που θα συλλέγονταν από τους αισθητήρες σε τακτά χρονικά διαστήματα καθώς επίσης και τα μεταδεδομένα του video τα οποία ακολουθούν το πρότυπο MPEG-7. Στη φάση αυτή έπρεπε να κατασκευάσουμε και τα metadata του video μας. Υπήρχαν δύο επιλογές :

- ✓ Απευθείας δημιουργία από εμάς τους ίδιους
- ✓ Αυτόματη δημιουργία μέσω ενός λογισμικού.

Επιλέξαμε τη δεύτερη περίπτωση διότι η κατασκευή μεταδεδομένων αφήνεται σε μια εφαρμογή με αποτέλεσμα η πιθανότητα λάθος δομής να είναι ελάχιστη. Πιο συγκεκριμένα υπάρχουν οι video annotators οι οποίοι διευκολύνουν τη δουλειά των μηχανικών στον τομέα των metadata. Πρόκειται για λογισμικό το οποίο δέχεται ως είσοδο ένα video και στηριζόμενο σε συγκεκριμένο format που έχει καθοριστεί με βάση τα γενικά πρότυπα, παράγει αρχείο MPEG7 με πληροφορίες του εν λόγω βίντεο. Υπάρχουν διάφορα εργαλεία που επιτελούν αυτή τη λειτουργία όπως το Munino, SVAT ή το MPEG-7 Video Annotator της IBM. Στην παρούσα εργασία επιλέξαμε το εργαλείο POLYSEMA MPEG-7 Video Annotator που υλοποιήθηκε από το τμήμα Πληροφορικής και Τηλεπικοινωνιών του ΕΚΠΑ. Εισήγαμε το video, ρυθμίσαμε κάποιες παραμέτρους όπως τα στιγμιότυπα που μας ενδιέφεραν και τα αντικείμενα ενδιαφέροντος και στη συνέχεια τρέξαμε το εργαλείο δημιουργώντας το κατάλληλο Mpeg7 αρχείο, το οποίο κι παρουσιάζεται στο Παράρτημα Β.



Εικόνα 5-2 :JIProlog Server[27]

ΚΕΦΑΛΑΙΟ 6

ΥΛΟΠΟΙΗΣΗ ΚΑΙ ΠΛΑΤΦΟΡΜΕΣ

Στην ενότητα αυτή θα αναλύσουμε τα εργαλεία και το λογισμικό που χρησιμοποιήσαμε για να στήσουμε την αρχιτεκτονική της εφαρμογής. Ουσιαστικά δηλαδή στο κεφάλαιο αυτό θα καταγραφούνε όλα τα βήματα της υλοποίησης με τις τεχνικές λεπτομέρειες τους. Τα πρακτικά στοιχεία της προαναφερθείσας αρχιτεκτονικής είναι:

- NETBEANS (java2me)
- JIProlog IDE (jiprolog)
- POLYSEMA MPEG-7 Video Annotator (mpeg7)

6.1 JAVA2ME-NETBEANS

Όπως αναφέραμε παραπάνω ο θεμέλιος λίθος της αρχιτεκτονικής μας ήταν η mobile java, Java Platform, Micro Edition ή Java ME. Η J2ME είναι μια εύχρηστη προγραμματιστική γλώσσα που υιοθετεί τα στοιχεία της κλασικής java (μεθόδους κλάσεις) και ταυτόχρονα ενισχύεται ώστε να μπορεί να υποστηριχθεί σε μικρής υπολογιστικής ισχύος συσκευές όπως τα κινητά. Η ανάγκη για τη νέα αυτή γλώσσα προήλθε από τη ζήτηση της αγοράς για παιχνίδια στο κινητό τηλέφωνο.

Οι συσκευές JavaME υλοποιούν ένα προφίλ (profile) κάθε φορά [24]. Τα πλέον συνηθισμένα είναι το Mobile Information Device Profile που ειδικεύεται στις κινητές συσκευές και δη τα κινητά τηλέφωνα καθώς και το Personal Profile που στοχεύει σε καταναλωτικά προϊόντα όπως PDAs. Τα profiles με τη σειρά τους είναι υποκατηγορίες των διαμορφώσεων (configurations) πάνω στα οποία στηρίζονται οι εφαρμογές και τα οποία είναι[25]:

- ✓ Connected Limited Device Configuration
- ✓ Connected Device Configuration

✓ **Connected Limited Device Configuration**

Η διαμόρφωση αυτή αποτελεί ένα αυστηρό υποσύνολο βιβλιοθηκών από java κλάσεις και είναι περιέχει τα ελάχιστα στοιχεία που απαιτούνται για να λειτουργήσει μια εικονική μηχανή Java. Ουσιαστικά παρέχει τα πιο βασικά σύνολα βιβλιοθηκών και τα χαρακτηριστικά γνωρίσματα της εικονικής μηχανής που πρέπει να είναι πάντα παρόντα σε ένα J2ME περιβάλλον. Τα profiles που υποστηρίζει είναι τα εξής:

❖ **Mobile Information Device Profile(MIDP)**

Σχεδιασμένο για κινητά τηλέφωνα, το profile περιλαμβάνει ένα GUI API. Όλες οι εφαρμογές που είναι σχεδιασμένες για αυτό το profile ονομάζονται MIDlets. Πλέον όλα τα κινητά διαθέτουν MIDP υλοποίηση.

❖ **Information Module Profile(IMP)**

Το profile IMP αναφέρεται σε ενσωματωμένες συσκευές όπως μηχανές πώλησης, βιομηχανικές ενσωματωμένες εφαρμογές, συστήματα ασφαλείας και άλλες παρόμοιες συσκευές που δε διαθέτουν οθόνη και χαρακτηρίζονται από περιορισμένη δικτυακή συνδεσιμότητα.

✓ **Connected Device Configuration**

Το configuration αυτό είναι υποσύνολο της standard έκδοσης java (javaSE), περιλαμβάνει όλες τις βιβλιοθήκες που δε σχετίζονται με GUI και διαθέτει περισσότερα στοιχεία από ότι το CLDC. Τα profile που υποστηρίζει είναι:

❖ **Foundation Profile(FP)**

Το profile αυτό προβλέπεται να χρησιμοποιηθεί από συσκευές που απαιτούν μια πλήρη υλοποίηση εικονικής μηχανής java. Οι τυπικές υλοποιήσεις θα χρησιμοποιήσουνε ένα υποσύνολο αυτού του profile.

❖ **Personal Basic Profile(PBP)**

Αποτελεί επέκταση του foundation profile.

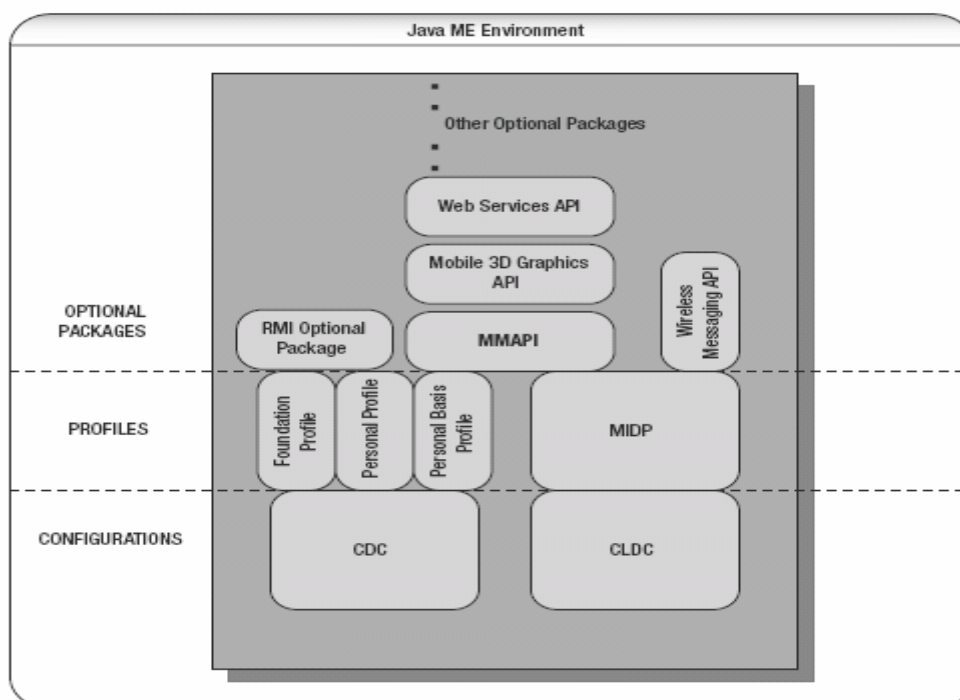
❖ **Personal Profile(PP)**

Αποτελεί επέκταση του personal basic profile.

Στην εφαρμογή μας εμείς χρησιμοποιήσαμε τη πρώτη διαμόρφωση που αναφέρεται σε μικρής υπολογιστικής ισχύς συσκευές και το Mobile Information Device Profile (MIDP 2.0). Ουσιαστικά κατασκευάσαμε ένα MIDlet εμπλουτίζοντάς το συνεχώς με νέες βιβλιοθήκες (JSR) ανάλογα με τις ανάγκες που προέκυπταν από την εφαρμογή. Η κατασκευή του MIDlet πραγματοποιήθηκε στην πλατφόρμα NetBeans 6.1 η οποία υποστηρίζει J2ME που είναι η δεύτερη και η πιο πρόσφατη έκδοση mobile java. Η συγκεκριμένη πλατφόρμα επιλέχτηκε. Στη συνέχεια περιγράφουμε τις βιβλιοθήκες που χρησιμοποιήσαμε στην υλοποίηση.

Στην επόμενη εικόνα (εικόνα 6-1) βλέπουμε και διαγραμματικά αυτή τη δομή της mobile java. Στο κατώτερο στρώμα βρίσκονται τα configurations CDC και CLDC. Στο επόμενο στρώμα εντοπίζουμε τα διάφορα προφίλ που περιγράψαμε:FP, PP και PBP για το CDC και MIDP για το CLDC (η δομή που χρησιμοποιούμε εμείς). Στο τελευταίο στρώμα συναντάμε τις ποικίλες βιβλιοθήκες που ενσωματώνουμε ελεύθερα στην εφαρμογή μας ανάλογα με τις επιθυμίες μας. Κάποιες από αυτές είναι οι:

MMAPI, Wireless Messaging API, RMI Optional Package, Mobile 3D Graphics API και Web Services API.



Εικόνα 6-1 :MMAPI στη δομή της javaME [25]

6.1.1 MMAPI

Η πρώτη βιβλιοθήκη που μας απασχόλησε ήταν η MMAPI (MOBILE MEDIA API - JSR 135) [25]. Είναι το προαιρετικό API (optional package) στο οποίο οι μηχανικοί στηρίζονται για τη διαχείριση των ανεπτυγμένων multimedia ιδιοτήτων μιας συσκευής. Χάρη σε αυτό μπορούμε να κατασκευάσουμε εφαρμογές με ποικίλες ιδιότητες. Μεταξύ αυτών είναι η αναπαραγωγή διαφορετικών format audio και video από το δίκτυο, από ένα αποθηκευμένο δίσκο ή ένα jar αρχείο, η εγγραφή ήχου και η σύλληψη εικόνων (στατικών και κινούμενων), η μετάδοση ήχου μέσω δικτύου (stream audio) κ.α.

Όλες οι συσκευές οι οποίες υποστηρίζουν MIDP 2.0 μπορούν να εκμεταλλευτούν την προσφορά του MMAPI . Κάποιες από αυτές παρουσιάζονται στον παρακάτω πίνακα:

Πίνακας 6-1 :Κινητά που υποστηρίζουν MMAPI [25]

Device Manufacturer	Supported Devices	Wireless Toolkit
Alcatel	One Touch 756	http://www.my-onetouch.com
BenQ	AX75 (MIDP 1.0), C70, C75, CF75/76, CL75, CX70/EMOTY, CX75, M75, S75, SL75, SXG75	http://www.benq.com/developer
Motorola	C975, E1000, A1000, A630, A780, A845, C380, C650, E398, E680, SLVR, T725, V180, V220, V3, V300, V303, V360, V400, V500, V525, V550, V551, V600, V620, V635, V8, V80, V980, i730	http://www.motocoder.com
Nokia	All Series 40, Series 60, and Series 80 based devices	http://forum.nokia.com/main.html
Samsung	E310, E380, E710, D400, P705, D410, I76X192 Series, E810, E310	http://uk.samsungmobile.com/club/developers_club/cl_de_sdk_01.jsp
Sony-Ericsson	W900, Z600, T610, T616, T618, V600, W800, K608, W550, W600, z520, D750, Z800, K600, K750, K300, K500, K700, J300, V800, Z500, S700, Z1010	http://developer.sonyericsson.com

Όπως αναφέραμε και παραπάνω η βιβλιοθήκη αυτή μπορεί να διαχειριστεί υλικό από διάφορα σημεία (δίκτυο, δίσκο) και να το κάνει αναγνώσιμο από την εκάστοτε κινητή συσκευή. Για να επιτευχθεί αυτό η MMAPi χρησιμοποιεί τρία high-level αντικείμενα, Players, Manager και DataSources. Χάρη σε αυτά τα δύο αντικείμενα μπορούμε να προχωρήσουμε σε επεμβάσεις στο Video κατά την αναπαραγωγή του. Ο Player (`javax.microedition.media`) είναι ένα Interface που χειρίζεται την επεξεργασία και την αναπαραγωγή των multimedia δεδομένων, ο DataSource (`javax.microedition.media.protocol`) είναι μια αφηρημένη κλάση που ενσωματώνει τη διαδικασία εύρεσης των δεδομένων ενώ ο Manager(`javax.microedition.media`) αποτελεί τον ενδιάμεσο κρίκο ανάμεσα στα δύο προηγούμενα.

➤ **DataSource**

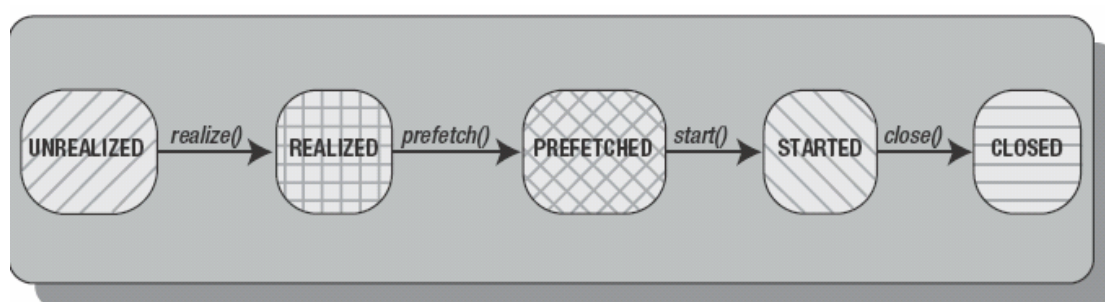
Η κλάση αυτή παρέχει πρόσβαση σε multimedia αρχεία εντοπίζοντας την τοποθεσία αποθήκευσής τους και ανοίγοντας μια σύνδεση με αυτά. Κάθε τέτοια κλάση αποτελείται από μία ή πολλές ροές (streams), τα λεγόμενα SourceStream [25]. Για παράδειγμα αρχεία video μπορούν να διασπαστούν σε 2 ξεχωριστές ροές, μία για τον ήχο και μια για την κινούμενη εικόνα. Με τον τρόπο αυτό μπορούμε να διαχειριζόμαστε ανεξάρτητα τα επιμέρους στοιχεία του video κι αυτό μας παρέχει μεγαλύτερη ελευθερία επιλογών. Συνήθως βέβαια τα DataSource στιγμιότυπα (instances) χαρακτηρίζονται από μια ροή.

➤ **Player**

Με το interface αυτό μπορούμε να χειριζόμαστε την αναπαραγωγή και τη διαχείριση των δεδομένων μας. Ελέγχουμε, συγχρονίζουμε στοιχεία με άλλους Players και αντιλαμβάνομαστε γεγονότα όπως ξεκίνημα, παύση και διακοπή. Το τελευταίο πραγματοποιείται με τη χρήση του PlayerListener interface που λαμβάνει ενημερώσεις για όλα τα παραπάνω κι ακόμα περισσότερα events. (`player.addPlayerListener(this)`).[25]

Ο Player μπορεί να χαρακτηρίζεται από τις εξής καταστάσεις: Μη πραγματοποιήσιμη (Unrealized), πραγματοποιήσιμη (realized), προδημιουργημένη (prefetched), δημιουργημένη (started) και κλειστή (closed). Ένα στιγμιότυπο του player μπορεί να περάσει από όλες τις καταστάσεις αυτές εφόσον έχει ήδη ξεκινήσει. Η μετακίνηση δεν είναι υποχρεωτικά γραμμική και μπορεί να συμβεί είτε αυτόματα είτε ύστερα από παρέμβαση του χρήστη. Η μόνη περίπτωση που δεν είναι δυνατή είναι η μετάβαση από closed σε οποιαδήποτε άλλη κατάσταση.

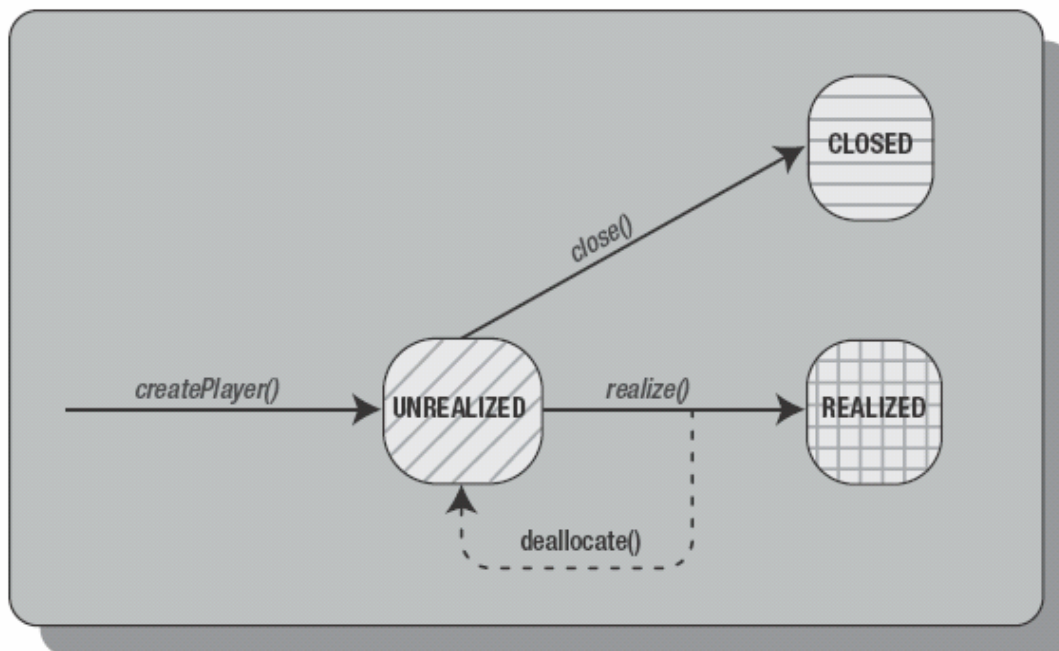
Όταν ξεκινάει ο player βρίσκεται πάντα στην unrealized state. Μεταβαίνει στη realized μέσω της μεθόδου realize(). Στη συνέχεια μπορεί να προχωρήσει τη prefetched κατάσταση με την κλήση της μεθόδου prefetch(). Έπειτα μέσω της start(), οδηγούμαστε στη started state και τέλος μέσω της close(). Αυτή ακριβώς η ακολουθία παρουσιάζεται στην εικόνα 6-2.



Εικόνα 6-2 :States and methods of player [25]

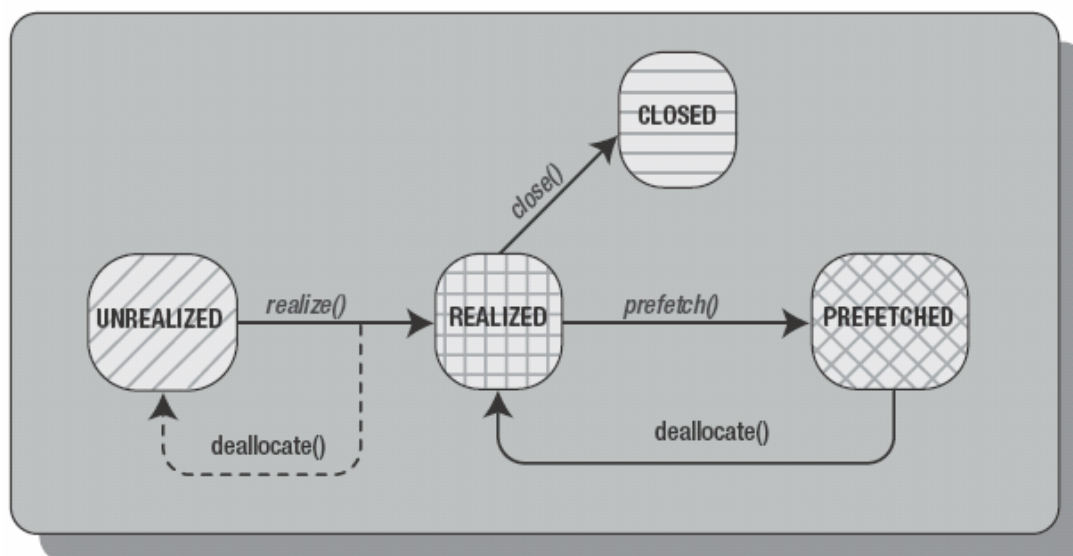
Στις εικόνες που ακολουθούν αμέσως μετά έχουμε αποτυπώσει διαγραμματικά τις εκάστοτε καταστάσεις που μόλις αναφέραμε κι αναλύουμε τις δυνατές μεταβάσεις που μπορούν να πραγματοποιηθούν. Πιο συγκεκριμένα η εικόνα 6-3 αναλύει τη unrealized κατάσταση και τον τρόπο μετάβασης σε realized. Η εικόνα 6-4 περιγράφει τη realized κατάσταση και τη διασύνδεσή της με τις καταστάσεις unrealized, prefetched και closed. Η εικόνα 6-5 παρουσιάζει τη prefetched κατάσταση και τις διασυνδέσεις της με τις καταστάσεις realized, started και closed. Τέλος η εικόνα 6-6 μας δείχνει την βασική κατάσταση started και πως μπορούμε να περάσουμε σε αυτήν ή να φύγουμε από αυτήν.

○ **UNREALIZED**



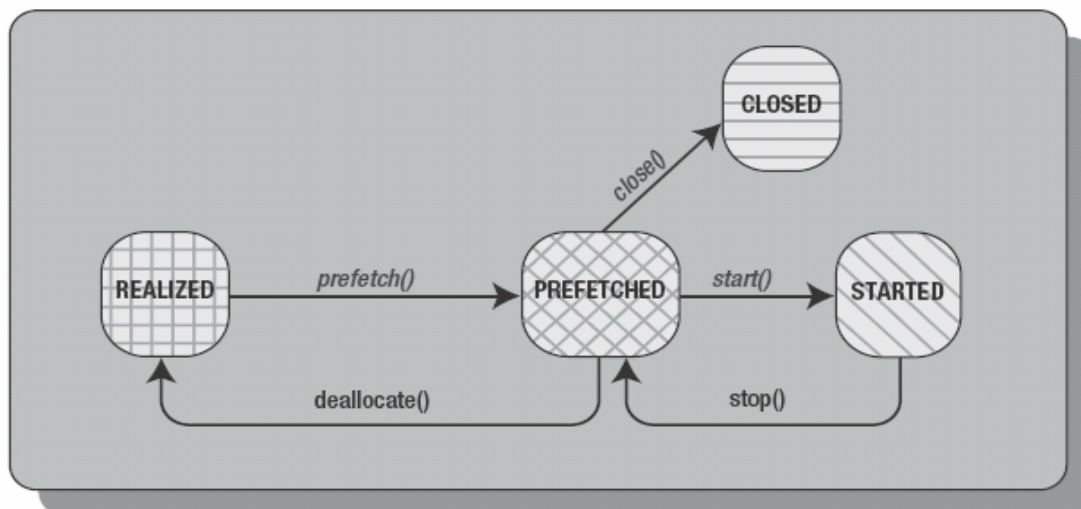
Εικόνα 6-3 :Unrealized state [25]

○ **REALIZED**



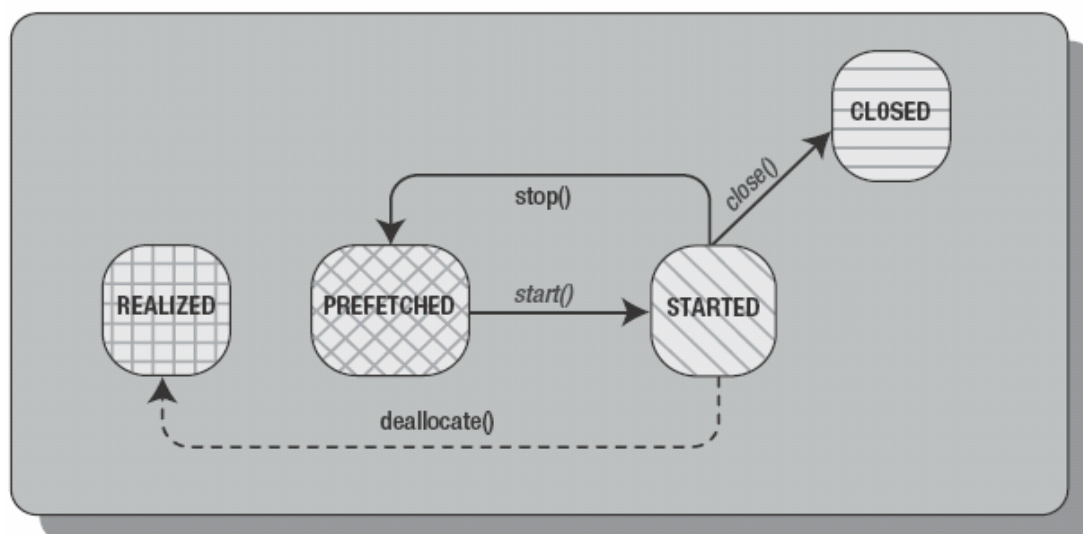
Εικόνα 6-4 :Realized state [25]

- **PREFETCHED**



Εικόνα 6-5 :Prefetched state [25]

- **STARTED**



Εικόνα 6-6 :Started state [25]

- **CLOSED**

Η συγκεκριμένη κατάσταση δεν έχει δυνατότητα μετάβασης σε άλλη. Είναι η κατάσταση στην οποία δε προκαλείται καμία εξαίρεση (exception). Μάλιστα αν κάποιο σφάλμα συμβεί, το video θα καταλήξει αυτόματα σε αυτό το state.

Όσον αφορά το `PlayerListener` έχουμε τις εξής περιπτώσεις γεγονότων που μπορούν να γίνουν αντιληπτά στην παρακάτω εικόνα:

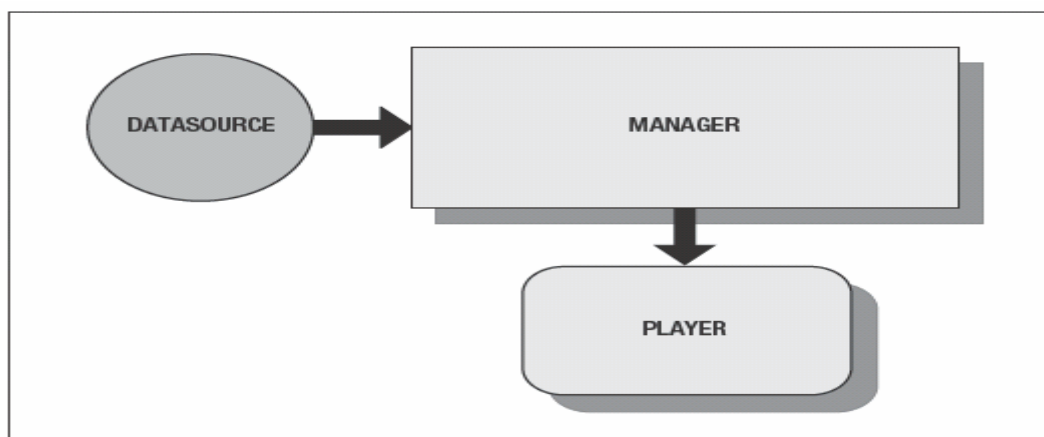
Πίνακας 6-2 :Γεγονότα που αναγνωρίζει ο `PlayerListener` [25]

Player Event Constant	Constant Value	When Fired
<code>BUFFERING_STARTED</code>	<code>bufferingStarted</code>	When an instance has started buffering media data for processing or playback.
<code>BUFFERING_STOPPED</code>	<code>bufferingStopped</code>	When an instance has exited the buffering stage.
<code>CLOSED</code>	<code>closed</code>	When an instance is closed.
<code>DEVICE_AVAILABLE</code>	<code>deviceAvailable</code>	When a system resource required by a <code>Player</code> instance becomes available for use.
<code>DEVICE_UNAVAILABLE</code>	<code>deviceUnavailable</code>	When a system resource required by a <code>Player</code> instance becomes unavailable. This event must precede the previous event.
<code>DURATION_UPDATED</code>	<code>durationUpdated</code>	When the duration of previously unknown media data becomes available.
<code>END_OF_MEDIA</code>	<code>endOfMedia</code>	When an instance has reached the end of the media during the current loop.
<code>ERROR</code>	<code>error</code>	When an error, which is usually fatal, occurs.
<code>RECORD_ERROR</code>	<code>recordError</code>	When an error occurs during recording (audio or video).
<code>RECORD_STARTED</code>	<code>recordStarted</code>	When recording of media data has started.
<code>RECORD_STOPPED</code>	<code>recordStopped</code>	When recording of media data has stopped.
<code>SIZE_CHANGED</code>	<code>sizeChanged</code>	When the size of a video display has changed for whatever reason.
<code>STARTED</code>	<code>started</code>	When the instance has entered the <code>STARTED</code> state.
<code>STOPPED</code>	<code>stopped</code>	When the instance has paused due to the <code>stop()</code> method being called.
<code>STOPPED_AT_TIME</code>	<code>stoppedAtTime</code>	When the instance has paused due to the <code>StopTimeControl</code> 's <code>setStopTime()</code> method.
<code>VOLUME_CHANGED</code>	<code>volumeChanged</code>	When the volume of an audio device is changed.

➤ Manager

Όπως προαναφέραμε, η κλάση αυτή είναι η γέφυρα μεταξύ `DataSource` και `Player`, όπως διαφαίνεται και στην εικόνα 6-7. Στην περίπτωση μας:

```
Manager.createPlayer(getClass().getResourceAsStream("/video/phantom.mpg"), "video/mpeg");
```



Εικόνα 6-7 :`Datasource-Manager-Player` [25]

➤ Controls

Στη βιβλιοθήκη αυτή υπάρχουν επίσης και τα control αντικείμενα μέσω των οποίων μπορούμε να κατέχουμε τον έλεγχο του Player. Υπάρχει ένα μεγάλο φάσμα controls που παρουσιάζονται στην εικόνα 6-8, ώστε να παρέχονται πολλές δυνατότητες στους χρήστες χωρίς να χρειάζεται να προγραμματίζουν οι ίδιοι νέες μεθόδους κάθε φορά.

Control Interface	Description
FramePositioningControl	A control for video data that allows access to individual frames.
GUIControl	A control for data that requires a display, such as video.
MetaDataControl	Used to determine the metadata information stored within a media stream, such as title, copyright, author, and so on.
MIDIControl	A fully functional control that enables access to a device's MIDI player.
PitchControl	Used to control the pitch (frequency) of audio data.
RateControl	Used to control the playback rate of a Player.
RecordControl	Allows you to control the recording of data from a capture device, such as video from a camera or audio from a sound recorder.
StopTimeControl	A control that allows you to set a preset time when you want the Player to stop playing.
TempoControl	Similar to RateControl, this control allows you to change the tempo (speed) of playback for an audio Player, typically, a MIDI Player.
ToneControl	A fully functional control that allows you to play monotonic tone sequences.
VideoControl	Extends GUIControl and controls the display of video.
VolumeControl	The simplest control that allows you to control the volume of audio in a Player.

Εικόνα 6-8 :MMAPI Controls [25]

Στη συγκεκριμένη εφαρμογή χρησιμοποιήθηκαν μόνο δύο:

javax.microedition.media.control.VideoControl

javax.microedition.media.control.VolumeControl

καθώς όλες οι υπόλοιπες λειτουργίες που πραγματοποιήσαμε στο video υλοποιήθηκαν από εμάς.

6.1.2 JIP2ME

Η JIP2ME είναι η βιβλιοθήκη που προσθέσαμε στο πρόγραμμά μας για να μπορέσουμε να κάνουμε τη σύνδεση με τη JIProlog. Όπως έχουμε αναφέρει και παραπάνω, οι λειτουργίες στο video θα πραγματοποιούνταν ύστερα από αποφάσεις που θα λαμβάνονταν σχετικά με κάποιες τιμές του περιβάλλοντος. Υπάρχουν κανόνες (rules) που ορίζουν περιβαλλοντικές καταστάσεις με βάση τις παραμέτρους ήχο, φωτεινότητα και κίνηση. Το κινητό τηλέφωνο κατά τη διάρκεια της αναπαραγωγής του video θα εκτελεί ερωτήματα (queries) πάνω σε αυτούς τους κανόνες κι αναλόγως με τις απαντήσεις, θα λαμβάνονται και οι αντίστοιχες αποφάσεις. Το στάδιο αυτό εκτελέστηκε σε 2 βήματα:

a) Δημιουργία αρχείου κανόνων

Δημιουργήσαμε ένα αρχείο rules.pl στο οποίο καταγράψαμε τη κατάσταση του περιβάλλοντος στηριζόμενοι στις 3 γνωστές παραμέτρους. Με το ξεκίνημα της εφαρμογής, δημιουργείται instance μιας jip engine και το file αυτό φορτώνεται (Πλέον στο κινητό μας τρέχει μια μηχανή κανόνων που ανά πάσα στιγμή μπορεί να δεχτεί ερωτήματα (queries) και να δώσει απαντήσεις (answers).

Το αρχείο αυτό περιλαμβάνεται στο παράρτημα Α .

b) Δημιουργία των δεδομένων (facts)

Με τη βοήθεια μιας μηχανής τυχαίας αναπαραγωγής αριθμών (generator()) καταφέραμε να προσομοιώσουμε τις τυχαίες τιμές των αισθητήρων του περιβάλλοντος.

Πιο συγκεκριμένα χρησιμοποιήσαμε την εν λόγω γεννήτρια generator με σκοπό να παράγει τιμές αισθητήρων σε τακτά χρονικά διαστήματα, τα οποία εμείς ορίσαμε μέσω ενός timer της java. Ρυθμίσαμε λοιπόν κάποιο timer έτσι ώστε να αρχίζουμε να λαμβάνουμε τις πρώτες τιμές αισθητήρων 1 δευτερόλεπτο αφού έχει ξεκινήσει η εφαρμογή μας και από εκεί και ύστερα θα επαναλαμβάνεται η λήψη των τιμών των sensors κάθε 10 δευτερόλεπτα. Στο κεφάλαιο 7, θα δούμε ότι θα αλλάζουμε την αριθμητική αυτή παράμετρο σε ένα σύνολο πειραμάτων ώστε να καταγράψουμε με αξιοπιστία τη συνολική καθυστέρηση της εφαρμογής. Οι χρόνοι αυτοί επιλέχθηκαν ύστερα από μελέτη του συστήματος και δεδομένου ότι

έπρεπε να παρουσιάσουμε τη λειτουργικότητά της εφαρμογής μας σε ένα video 21 δευτερολέπτων. Κάθε φορά που παράγονταν οι τιμές αυτές λοιπόν, τις φορτώναμε από τη java στη μηχανή κανόνων ως facts για τον ήχο, τη φωτεινότητα και την κίνηση.

Καλέσαμε την εντολή `assert` της Jiprlog με την οποία ορίζουμε τα δεδομένα ,στη συνέχεια την εντολή `consult` και στη συνέχεια τρέξαμε ερωτήματα για τις ενέργειες που πρέπει να πραγματοποιηθούν στο video λόγω της περιβαλλοντικής κατάστασης.

Με βάση επομένως τις απαντήσεις που λαμβάναμε από τα queries στη μηχανή κανόνων εκτελούνταν και οι αντίστοιχες λειτουργίες στο video. Αυτό το στάδιο υλοποιήθηκε λοιπόν χάρη στη JIP2ME library.

6.1.3 KXML

Η KXML είναι η βιβλιοθήκη που χρησιμοποιήσαμε για να πραγματοποιηθεί η σύνδεση της εφαρμογής μας με το αρχείο `mpeg7` των μεταδεδομένων του video. (`import org.kxml.parser.*;`). Η διαχείριση των μεταδεδομένων θα μπορούσε ίσως να επιτευχθεί και μέσω του `metadataControl` που ορίζεται στο player της `microedition java`, κάτι τέτοιο όμως θα ήταν εφικτό μόνο στην περίπτωση που αναφερόμασταν σε video με ενσωματωμένα `metadata`. Εμείς όμως δε διαθέτουμε τέτοιο video κι επομένως δεν μπορούμε να διαχειριστούμε το συγκεκριμένο control. Επιπλέον όμως δεν μας απασχολεί και η εύρεση τέτοιων `embedded metadata`,όπως λέγονται, καθώς αυτά δεν είναι υψηλού επιπέδου που χρειαζόμαστε εμείς για την εφαρμογή μας. Για τους λόγους αυτούς στην εργασία μας, οι πληροφορίες για το video σε γενικό και συγκεκριμένο επίπεδο είναι αποθηκευμένες σε ένα εξωτερικό αρχείο,ανεξάρτητο από το video,το οποίο είναι κι αυτό αποθηκευμένο στην εφαρμογή μας . Ως γενικές πληροφορίες μπορούμε να χαρακτηρίσουμε τον κατασκευαστή, την ημερομηνία λήψης, το είδος του video,τον κάτοχο, την τοποθεσία αποθήκευσης, την περιγραφή περιεχομένου κτλ ενώ στο συγκεκριμένο κομμάτι αντιστοιχούν οι λεπτομερείς πληροφορίες σε επίπεδο `frame` ή χρόνου. Κατασκευάσαμε λοιπόν ένα `mpeg7` αρχείο στηριζόμενοι στο περιεχόμενο του video `phantom.mpg`.Το αρχείο `phantom.mp7.xml` είναι διαθέσιμο στο παράρτημα Β.

Η διαδικασία που ακολουθήθηκε ήταν η εξής:

a) Δημιουργία μεθόδου για parsing του metadata αρχείου.

Κατασκευάσαμε μια μέθοδο την οποία θα μπορούσαμε να καλούμε ανά πάσα στιγμή στην εφαρμογή μας με παράμετρο το εκάστοτε αρχείο xml που μας ενδιαφέρει. Στη μέθοδο αυτή ορίζουμε ένα αντικείμενο Parser και ένα αντικείμενο document στο οποίο θα καταγραφεί το περιεχόμενο.

```
XmlParser parser = null;
```

```
Document doc = new Document();
```

```
InputStream in = this.getClass().getResourceAsStream(file);
```

```
InputStreamReader is = new InputStreamReader(in);
```

```
parser = new XmlParser( is );
```

```
doc.parse( parser );
```

Στη συνέχεια ξεκινήσαμε την διέλευση του αρχείου από το root και ελέγχοντας όλα τα παιδιά (childs) και τα χαρακτηριστικά (attributes) ένα προς ένα. Για κάθε ένα item που βρίσκαμε υπολογίζαμε τα παιδιά του, τα οποία στη συνέχεια γινόντουσαν νέα αντικείμενα (items).

b) Αποθήκευση στοιχείων σε δομές δεδομένων

Όταν εντοπίζαμε items τα οποία μας ενδιέφεραν για τη λειτουργικότητα της εφαρμογής μας, αποθηκεύαμε το όνομα και τα χαρακτηριστικά του ώστε να μπορέσουμε να χρησιμοποιήσουμε. Ένα χαρακτηριστικό παράδειγμα ήταν η καταγραφή των χρόνων έναρξης και της διάρκειας κάποιων frames που είχαν επισημανθεί όπως και οι χωρικές συντεταγμένες συγκεκριμένων αντικειμένων μέσα στο video.

Χρονικά σημεία:

```
<VideoSegment id="vs3">  
  <MediaTime>  
    <MediaTimePoint>T00:00:09</MediaTimePoint>  
    <MediaDuration>PT2S</MediaDuration>  
  </MediaTime>
```

Χωρικές συντεταγμένες:

```
<SpatioTemporalDecomposition>  
  <StillRegion>  
    <SpatialLocator>  
      <Polygon>  
        <Coords xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001"  
mpeg7:dim="8">95:90:242:290:242:129:93:129</Coords>  
      </Polygon>  
    </SpatialLocator>
```

Στην εφαρμογή μας χρησιμοποιήσαμε τη δομή hashtable που παρέχει αρκετή ευελιξία στη διαχείρισή της.

Αποθήκευση της χρονικής έναρξης ενός frame:

```
if (itemMedia.getName().equals("MediaTimePoint"))  
{hashtable.put("MediaStart", mediaTimePoint); }
```

Στην παρακάτω εικόνα παρουσιάζεται ο γενικός τρόπος parsing ενός xml αρχείου, τον οποίο περιγράψαμε μόλις τώρα για την εφαρμογή μας

c) Κλήση της μεθόδου από το κυρίως πρόγραμμα

Είναι το αμέσως επόμενο βήμα μετά την δημιουργία της μηχανής κανόνων και του φορτώματος των rules. Καλείται η μέθοδος metadata, η οποία μας επιστρέφει τη δομή hashtable με όλες τις πληροφορίες που θα μας παρουσιαστούν χρήσιμες στη συνέχεια.

```
vsArray=metadata("files/phantom.mp7.xml");
```

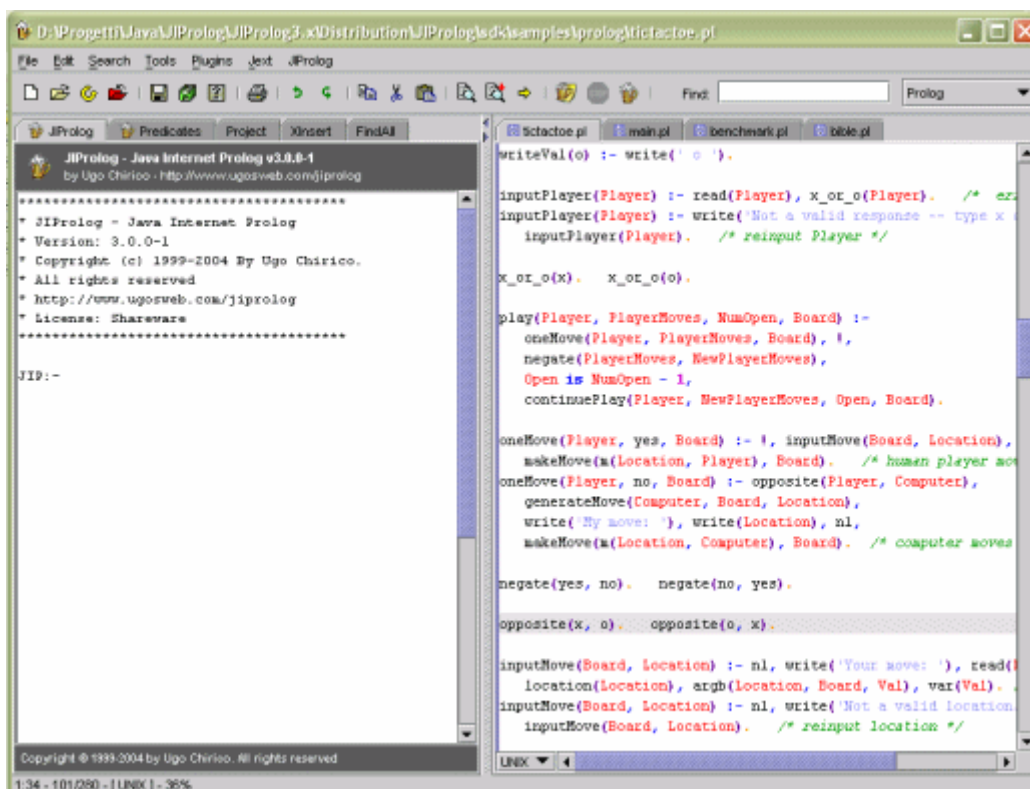
d) Ορισμός crop παραμέτρων

Η λειτουργία crop που καλούμαστε να εφαρμόσουμε στο video έχει μια ιδιαιτερότητα σε σχέση με όλες τις υπόλοιπες. Καλούμαστε να κάνουμε crop σε ένα συγκεκριμένο σημείο όταν υπάρχει «σημασιολογικό ενδιαφέρον». Για παράδειγμα στην περίπτωση μας, θέλουμε να επισημάνουμε στο χρήστη το γεγονός της ομιλίας σε κινητό τηλέφωνο (calling event). Για το λόγο αυτό έχουμε χαρακτηρίσει ως calling event ένα frame που δείχνει έναν άντρα να μιλάει στο κινητό. Μόλις το video φτάσει στο εν λόγω frame, το crop θα γίνει αυτόματα στην περιοχή γύρω από τον άντρα χωρίς να μας ενδιαφέρουν οι περιβαλλοντικοί παράμετροι. Επομένως στη φάση αυτή θα καλέσουμε το πρώτο query στη μηχανή κανόνων, ζητώντας να μας ορίσει αν υπάρχει τέτοιο event στο video μας. Εφόσον υπάρχει θα καταγράψουμε τη χρονική αφετηρία και τη χρονική διάρκεια του εν λόγω frame και θα καλέσουμε τη μέθοδο crop που έχουμε κατασκευάσει με βάση αυτά τα χρονικά κριτήρια και για τις συγκεκριμένες συντεταγμένες που επίσης έχουν οριστεί. Να τονίσουμε βέβαια πως λόγω του γεγονότος ότι το crop στηρίζεται στα metadata και όχι στους αισθητήρες, πρέπει να διακόψουμε τη λειτουργία αυτών ώστε να μην υπάρχει επικάλυψη της λειτουργίας είτε με το zoom in/zoom out είτε με το hide. Αυτό επιτυγχάνεται με τη χρήση timers που είναι υπεύθυνοι για τη διακοπή της γεννήτριας παραγωγής τιμών αισθητήρων και την επαναφορά της μετά το πέρας του crop. Τέλος, να αναφέρουμε ότι στην περίπτωση που βρισκόμαστε στη λειτουργία hideImage, το crop δε θα πραγματοποιηθεί αν και έχει προγραμματισθεί διότι τη χρονική στιγμή εκείνη θεωρούμε με βάση τους κανόνες που καταγράψαμε στην παράγραφο 5.2.2 ότι ο χρήστης δεν παρακολουθεί το video. Επομένως δεν υπάρχει λόγος να προχωρήσουμε σε crop.

Αφού λοιπόν έχουμε εκτελέσει όλα τα παραπάνω με τη βοήθεια των επιπλέον βιβλιοθηκών, επιστρέφουμε στο βασικό κορμό και ξεκινάμε την προσομοίωση των αισθητήρων (μέσω τυχαίας αναπαραγωγής αριθμών).

6.2 JIProlog IDE

Εξηγήσαμε εκτεταμένα παραπάνω ότι χρειαζόταν μια μηχανή κανόνων ώστε να μπορέσουμε να λάβουμε αποφάσεις για την περιβαλλοντική κατάσταση και τη λειτουργία που καλείται να εφαρμοστεί πάνω στο video. Η μηχανή αυτή (rule engine) υλοποιήθηκε σε prolog γλώσσα και το περιβάλλον που χρησιμοποιήθηκε ήταν το JIProlog IDE. Η εικόνα 6-9 παρουσιάζει το εν λόγω interface.[33]



```
writeVal(0) :- write(' 0 ').

inputPlayer(Player) :- read(Player), x_or_o(Player). /* ok
inputPlayer(Player) :- write('Not a valid response -- type x
inputPlayer(Player). /* reinput Player */

x_or_o(x). x_or_o(o).

play(Player, PlayerMoves, NumOpen, Board) :-
  oneMove(Player, PlayerMoves, Board), !,
  negate(PlayerMoves, NewPlayerMoves),
  Open is NumOpen - 1,
  continuePlay(Player, NewPlayerMoves, Open, Board).

oneMove(Player, yes, Board) :- !, inputMove(Board, Location),
  makeMove(m(Location, Player), Board). /* human player move
oneMove(Player, no, Board) :- opposite(Player, Computer),
  generateMove(Computer, Board, Location),
  write(' My move: '), write(Location), nl,
  makeMove(m(Location, Computer), Board). /* computer moves

negate(yes, no). negate(no, yes).

opposite(x, o). opposite(o, x).

inputMove(Board, Location) :- nl, write('Your move: '), read(
  Location(Location), argb(Location, Board, Val), var(Val).
inputMove(Board, Location) :- nl, write('Not a valid location
inputMove(Board, Location). /* reinput location */
```

JIProlog - Java Internet Prolog v3.0.0-1
by Ugo Chiaro - http://www.ugoweb.com/jiprolog

* JIProlog - Java Internet Prolog
* Version: 3.0.0-1
* Copyright (c) 1999-2004 By Ugo Chiaro.
* All rights reserved
* http://www.ugoweb.com/jiprolog
* License: Shareware

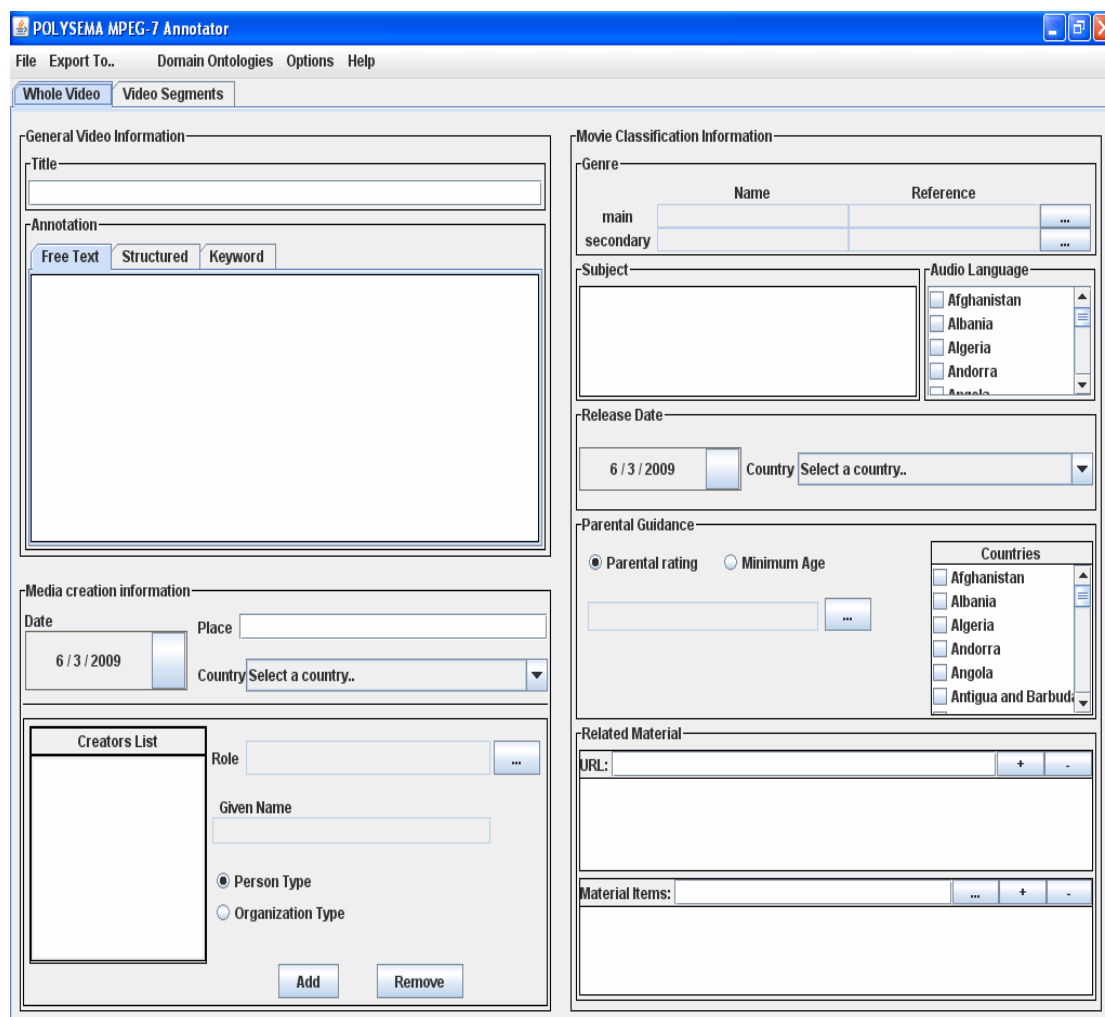
JIP:-

Copyright © 1999-2004 by Ugo Chiaro. All rights reserved
1:34 - 101/280 - [UNIX] - 35%

Εικόνα 6-9 :JIProlog IDE [27]

6.3 POLYSEMA MPEG-7 Video Annotator

Το εργαλείο αυτό χρησιμοποιήθηκε για τη δημιουργία MPEG-7 αρχείου metadata. Όπως προαναφέραμε θα μπορούσαμε να χειριστούμε την εφαρμογή με metadata ήδη ενσωματωμένα στο video, όμως λόγω δυσκολίας εύρεσης και κατασκευής τέτοιων video, αποφασίσαμε να τα δημιουργήσουμε εμείς. Επιλέξαμε λοιπόν ένα λογισμικό να παράγει το αρχείο αυτό με βάση κάποιες προδιαγραφές που του έχουν οριστεί για να αποφύγουμε πολλά πιθανά σφάλματα στην περίπτωση που το κατασκευάζαμε εξ αρχής οι ίδιοι. Η εικόνα 6-10 παρουσιάζει το περιβάλλον αυτού του εργαλείου.



Εικόνα 6-10 :Polysema Annotator

ΚΕΦΑΛΑΙΟ 7

ΕΠΙΔΟΣΕΙΣ ΚΑΙ ΣΥΜΠΕΡΑΣΜΑΤΑ

Στην ενότητα αυτή θα παρουσιάσουμε τις επιδόσεις της πλατφόρμας που υλοποιήσαμε καθώς και τα συμπεράσματα που προέκυψαν μέσα από αυτήν την εργασία. Στο κομμάτι των συμπερασμάτων θα αναλύσουμε τα θετικά στοιχεία της εφαρμογής μας αλλά και τις δυσκολίες που αντιμετωπίσαμε σε όλη τη διάρκεια της υλοποίησής της.

7.1 ΕΠΙΔΟΣΕΙΣ

Όσον αφορά τις επιδόσεις του συστήματος, θα αναλύσουμε θέματα τα οποία είναι κρίσιμα για εφαρμογές κινητού υπολογισμού. Τέτοιου είδους θέματα μπορεί να είναι η ποιότητα του αναπαραγόμενου video ως προς τον ήχο, την εικόνα και το συγχρονισμό αυτών, η σωστή εφαρμογή των κανόνων (rules) για τις τιμές των περιβαλλοντικών αισθητήρων καθώς και η καθυστέρηση της εφαρμογής. Το γεγονός ότι αναφερόμαστε σε εφαρμογή κινητού τηλεφώνου, καθιστά την καθυστέρηση ίσως τον πιο σημαντικό και κρίσιμο παράγοντα επίδοσης. Αυτό συμβαίνει διότι ο χρήστης χρησιμοποιεί το κινητό μέσα στην καθημερινότητά του που ο χρόνος τρέχει και οι καταστάσεις μεταβάλλονται συνεχώς. Έχει πολύ ενδιαφέρον λοιπόν να δούμε τις καθυστερήσεις της συγκεκριμένης πλατφόρμας, η οποία έχει αρκετά πολύπλοκη δομή.

7.1.1 Ποιότητα

Η συνολική ποιότητα του video ήταν άψογη. Στην εφαρμογή μας χρησιμοποιήσαμε το video phantom.mpg το οποίο αναπαριστά στιγμιότυπα από διάφορες ανθρώπινες δραστηριότητες όπως έναν άντρα να μιλάει στο τηλέφωνο και τοπία όπως πόλεις και δρόμους. Είχε μέγεθος 706 kBytes, διάρκεια 21.02 δευτερόλεπτα, 630 frames και συμπίεση mpeg. Τα χαρακτηριστικά αυτά υποδηλώνουν ότι το συγκεκριμένο Video ήταν σύντομο σε διάρκεια αλλά είχε αρκετές εναλλαγές στο περιεχόμενο του και πολλές πληροφορίες. Παρόλη όμως την ταχύτητα των εναλλαγών, το Video ήταν καθαρό ως προς την εικόνα και τον ήχο χάρη στη δυνατότητα υποστήριξης του από την βιβλιοθήκη MMAPi της mobile java που παρουσιάσαμε στην *ενότητα 6.1* . Αυτό αποδεικνύεται στην εικόνα 7-1 όπου παρουσιάζεται ένα στιγμιότυπο από το video που χρησιμοποιήσαμε στην υλοποίηση της πλατφόρμας.



Εικόνα 7-1 :Στιγμιότυπο video στο προσομοιωτή τηλεφώνου

7.1.2 Σωστή εφαρμογή κανόνων

Η επιτυχημένη εφαρμογή των κανόνων είναι κρίσιμης σημασίας για το πρόγραμμά μας καθώς σε αυτήν την επιτυχία στηρίζεται η όλη φιλοσοφία της πλατφόρμας. Εάν υπήρχε κάποιο πρόβλημα στη διεκπεραίωση των ερωτημάτων στην μηχανή κανόνων ή αν δε δίνονταν οι σωστές απαντήσεις με βάση τα κριτήρια που είχαμε αρχικά θέσει, το εγχείρημά μας θα είχε αποτύχει πλήρως. Όσον αφορά λοιπόν το θέμα αυτό, διαχειριστήκαμε κατάλληλα τη μηχανή κανόνων και επιτύχαμε σωστά αποτελέσματα. Για να το καταφέρουμε αυτό υπήρχε ένα κρίσιμο σημείο στην εφαρμογή μας. Το πέρασμα των παραμέτρων στη rule engine (στην περίπτωση μας οι τυχαίες τιμές των αισθητήρων) αλλά και η διαγραφή των δεδομένων με το πέρας του κάθε ερωτήματος. Αυτό είναι πολύ σημαντικό καθώς όταν περνάμε γεγονότα (facts) σε μια μηχανή κανόνων μέσω της εντολής assert (είναι διαθέσιμη στη βιβλιοθήκη JIP2ME), αυτές αποθηκεύονται αυτόματα στην μνήμη εργασίας της. Εάν λοιπόν εμείς θέλουμε να εκτελέσουμε νέο ερώτημα με διαφορετικές τιμές στη συγκεκριμένη μηχανή χωρίς να έχουμε διαγράψει τις προηγούμενες τιμές τότε το αποτέλεσμα θα βγει λανθασμένο. Θα κάναμε assert τα νέα δεδομένα, αλλά η rule engine θα είχε κρατήσει τις προηγούμενες τιμές και δε θα τις αντικαθιστούσε οπότε οι απαντήσεις που θα δίνονταν δε θα ήταν οι σωστές. Θα μπορούσε βέβαια να αποφευχθεί αυτό «κλείνοντας» μετά από κάθε ερώτημα τη rule engine και ξαναδημιουργώντας την στο επόμενο query. Αυτό όμως θα καθυστερούσε πολύ την εφαρμογή μας, όπως θα δούμε και στην επόμενο παράγραφο, και για αυτό αποφασίστηκε να δημιουργείται μία φορά στο ξεκίνημα και να κλείνει μόνο όταν θα ολοκληρώνεται και η συνολική εφαρμογή. Για το λόγο αυτό έπρεπε να διαγράψουμε τις τιμές των αισθητήρων κάθε φορά που λαμβανόντουσαν οι αποφάσεις για τις λειτουργίες στο Video. Αυτό το επιτύχαμε με την εντολή retract που είναι και αυτή διαθέσιμη μέσω της βιβλιοθήκης JIP2ME .

7.1.3 Καθυστέρηση

Η καθυστέρηση είναι πολύ σημαντικός παράγοντας για την εφαρμογή μας και για αυτό πραγματοποιήσαμε δοκιμές για να μετρήσουμε το χρόνο εκτέλεσης των εντολών. Για μεγαλύτερη ακρίβεια πραγματοποιήσαμε 3 σύνολα δοκιμών. Σε κάθε σύνολο αλλάζαμε τη χρονική παράμετρο που είχαμε ορίσει για τον timer. Όπως παρουσιάσαμε και παραπάνω, είχαμε θέσει ένα timer που όριζε κάθε 10 δευτερόλεπτα να δεχόμαστε τιμές από τους αισθητήρες. Στη διαδικασία των δοκιμών αλλάξαμε αυτή την παράμετρο σε 5 και 12 δευτερόλεπτα. Εφόσον δεν υπήρχε άλλη χρονική παράμετρος στο πρόγραμμά μας, δεν αλλάξαμε τίποτα άλλο.

Οι χρόνοι λοιπόν που μελετήσαμε ήταν οι εξής 11:

1. Χρόνος δημιουργίας μηχανής κανόνων.(Rule Eng)
2. Χρόνος για το consult του αρχείου κανόνων rules.pl.(C.Rules)
3. Χρόνος για το parsing των metadata MPEG-7.(P.Metadata)
4. Χρόνος για το parsing του πίνακα δεδομένων όπου αποθηκεύονται σημαντικά στοιχεία από το αρχείο των metadata phantom.mp7.xml.(P.Results)
5. Χρόνος για το πρώτο query στη rule engine για να συγχρονιστεί η λειτουργία του crop εφόσον υπάρχει κάποιο στοιχείο που θέλουμε να επισημάνουμε στο video.(Query.Crop)
6. Χρόνος για το συγχρονισμό του crop με την εφαρμογή. (TimingCrop)
7. Χρόνος για το συγχρονισμό της διακοπής παραγωγής τιμών αισθητήρων και ερωτημάτων στη μηχανή κανόνων. (TimingPause)
8. Χρόνος για το συγχρονισμό της επαναφοράς παραγωγής τιμών αισθητήρων και ερωτημάτων στη μηχανή κανόνων. (TimingRestart)
9. Χρόνος για την παραγωγή τιμών αισθητήρων μέσω τυχαίας γεννήτριας αριθμών. (TimingSensor)
10. Χρόνος για τα queries στη rule engine και τη λήψη των solutions.(QueryRuleEng)
11. Χρόνος για το φιλτράρισμα των solutions από τη java και εκτέλεση των εκάστοτε λειτουργιών στο video, όπως αποκοπή κι εστίαση. (Functions)

7.1.3.1 Μετρήσεις

Στη συνέχεια παρουσιάζουμε τα αποτελέσματα από τις εκτελέσεις της εφαρμογής μας ενώ ο σχολιασμός αυτών των τιμών θα πραγματοποιηθεί στην αμέσως επόμενη παράγραφο 7.1.3.2

Οι πίνακες που θα ακολουθήσουν αποτελούνται από 10 στήλες ,όσες δηλαδή και τα πειράματα που εκτελέσαμε ενώ συναντάμε και 11 γραμμές κάθε μία από τις οποίες αντιστοιχεί σε συγκεκριμένους χρόνους καθυστέρησης. Οι χρόνοι αυτοί έχουν περιγραφεί στην προηγούμενη υπό-ενότητα.

ι. Χρόνοι καθυστέρησης με λήψη τιμών αισθητήρων κάθε 10 s

Πίνακας 7-1 :Στατιστικά με sensor timer στα 10 s

Χρόνοι	Test 1 (ms)	Test 2 (ms)	Test 3 (ms)	Test 4 (ms)	Test 5 (ms)	Test 6 (ms)	Test 7 (ms)	Test 8 (ms)	Test 9 (ms)	Test 10 (ms)
Rule Eng	735	922	687	687	938	671	688	672	672	688
C.Rules	47	47	47	47	78	47	31	47	47	47
P.Metadata	703	906	641	641	797	641	688	657	641	656
P.Results	31	47	31	31	31	31	31	31	31	31
Query.Crop	31	31	31	31	31	32	16	31	31	32
TimingCrop	0	0	0	0	0	0	0	0	0	0
TimingPause	0	0	0	0	0	0	0	0	0	0
TimingRestart	0	0	0	0	0	0	15	0	0	0
TimingSensor	15	0	0	0	0	0	0	0	0	0
QueryRuleEng	31	47	32	15	47	15	15	16	16	16
Functions	0	0	16	0	0	0	0	0	0	0

Πίνακας 7-2 :Mean delay με sensor timer στα 10 s

ΧΡΟΝΟΙ ΚΑΘΥΣΤΕΡΗΣΗΣ	ΜΕΣΕΣ ΤΙΜΕΣ(ms)
Φόρτωση Rule engine	736
Consult rules	48,5
Parsing αρχείου metadata	697,1
Parsing αποτελεσμάτων από metadata	32,6
Query για το crop	29,7
Συγχρονισμός για το crop	0
Συγχρονισμός διακοπής	0
Συγχρονισμός επαναφοράς	1,5
Συγχρονισμός αισθητήρων	1,5
Queries στη rule engine	25
Εκτέλεση λειτουργιών	1,6
ΣΥΝΟΛΙΚΗ ΚΑΘΥΣΤΕΡΗΣΗ(για το video που χρησιμοποιήσαμε)	1573,5

ii.Χρόνοι καθυστέρησης με λήψη τιμών αισθητήρων κάθε 5 s

Πίνακας 7-3 :Στατιστικά με sensor timer στα 5 s

Χρόνοι	Test 1 (ms)	Test 2 (ms)	Test 3 (ms)	Test 4 (ms)	Test 5 (ms)	Test 6 (ms)	Test 7 (ms)	Test 8 (ms)	Test 9 (ms)	Test 10 (ms)
Rule Eng	672	890	984	735	906	687	672	922	688	688
C.Rules	47	63	63	47	78	47	47	62	47	47
P.Metadata	656	875	828	718	1047	687	781	813	656	656
P.Results	31	47	31	32	31	32	32	31	31	31
Query.Crop	31	31	47	47	32	47	15	16	32	32
TimingCrop	0	0	0	0	0	0	0	0	0	0
TimingPause	0	0	0	0	0	0	0	0	0	0
TimingRestart	0	0	16	0	15	0	0	0	0	0
TimingSensor	0	0	0	0	0	0	0	0	0	0
QueryRuleEng	16	32	31	31	31	16	31	31	16	16
Functions	0	16	0	0	0	0	16	0	0	0

Πίνακας 7-4 :Mean delay με sensor timer στα 5 s

ΧΡΟΝΟΙ ΚΑΘΥΣΤΕΡΗΣΗΣ	ΜΕΣΕΣ ΤΙΜΕΣ(ms)
Φόρτωση Rule engine	784,4
Consult rules	54,8
Parsing αρχείου metadata	771,7
Parsing αποτελεσμάτων από metadata	32,9
Query για το crop	33
Συγχρονισμός για το crop	0
Συγχρονισμός διακοπής	0
Συγχρονισμός επαναφοράς	3,1
Συγχρονισμός αισθητήρων	0
Queries στη rule engine	25,1
Εκτέλεση λειτουργιών	3,2
ΣΥΝΟΛΙΚΗ ΚΑΘΥΣΤΕΡΗΣΗ(για το video που χρησιμοποιήσαμε)	1708,2

iii.Χρόνοι καθυστέρησης με λήψη τιμών αισθητήρων κάθε 12 s

Πίνακας 7-5 :Στατιστικά με sensor timer στα 12 s

Χρόνοι	Test 1 (ms)	Test 2 (ms)	Test 3 (ms)	Test 4 (ms)	Test 5 (ms)	Test 6 (ms)	Test 7 (ms)	Test 8 (ms)	Test 9 (ms)	Test 10 (ms)
Rule Eng	687	687	672	656	1109	672	672	672	687	672
C.Rules	47	32	47	47	79	47	47	46	32	46
P.Metadata	672	672	656	640	984	656	657	657	656	688
P.Results	31	31	32	32	31	31	31	31	31	31
Query.Crop	16	31	15	31	31	32	31	16	31	16
TimingCrop	0	0	0	0	0	0	0	0	0	0
TimingPause	0	0	0	0	0	0	0	0	0	0
TimingRestart	0	0	16	0	0	0	16	0	0	0
TimingSensor	0	0	0	0	0	0	0	0	0	0
QueryRuleEng	16	31	15	16	31	16	15	15	15	16
Functions	0	0	0	0	0	0	0	0	0	0

Πίνακας 7-6 :Mean delay με sensor timer στα 12 s

ΧΡΟΝΟΙ ΚΑΘΥΣΤΕΡΗΣΗΣ	ΜΕΣΕΣ ΤΙΜΕΣ (ms)
Φόρτωση Rule engine	718,6
Consult rules	47
Parsing αρχείου metadata	693,8
Parsing αποτελεσμάτων από metadata	31,2
Query για το crop	25
Συγχρονισμός για το crop	0
Συγχρονισμός διακοπής	0
Συγχρονισμός επαναφοράς	3,2
Συγχρονισμός αισθητήρων	0
Queries στη rule engine	18,6
Εκτέλεση λειτουργιών	0
ΣΥΝΟΛΙΚΗ ΚΑΘΥΣΤΕΡΗΣΗ(για το video που χρησιμοποιήσαμε)	1537,4

7.1.3.2 Αξιολόγηση μετρήσεων

Στην παράγραφο αυτή θα αξιολογήσουμε τις μετρήσεις που παρουσιάσαμε στους προηγούμενους πίνακες. Μελετώντας τις θα μπορέσουμε να αντιληφθούμε ποια στοιχεία της εφαρμογής μας προκαλούν καθυστερήσεις, πόσο κρίσιμες είναι αυτές οι καθυστερήσεις καθώς και αν διαδραματίζει τελικά κάποιο ρόλο στην υπόθεση αυτή, η χρονική παράμετρος για την προσομοίωση των αισθητήρων.

Παρατηρώντας επομένως και τα 3 ζεύγη πινάκων για τις 3 διαφορετικές χρονικές παραμέτρους του generator() που έχουμε αναφέρει στην παράγραφο 6.1.2, παρατηρούμε ότι τα στάδια όπου εισάγουν τη σημαντική καθυστέρηση είναι ακριβώς τα ίδια, η δημιουργία της μηχανής κανόνων και το parsing του αρχείου μεταδεδομένων. Για τις περιπτώσεις των 10 ms, 5ms και 12 ms του generator(), έχουμε 736, 784.4 και 718.6 για τη rule engine ενώ 697.1 ms, 771.7 ms και 693.8 ms για τα metadata αντίστοιχα. Οι μεγάλες αυτές τιμές είναι λογικές αν αναλογιστεί κανείς πως και στις δύο περιπτώσεις η java προσπαθεί να επικοινωνήσει με εξωτερικά στοιχεία. Η δημιουργία rule engine είναι μια απαιτητική σε πόρους διαδικασία. Το parsing των μεταδεδομένων επίσης είναι μια χρονοβόρα διαδικασία ανάλογα και με τη λεπτομέρεια που έχουμε θέσει εμείς τις πληροφορίες. Όσο εκτενέστερο το αρχείο, τόσο περισσότερος χρόνος απαιτείται για να διαβαστεί και να αποθηκευτούν τα δεδομένα που μας ενδιαφέρουν στη μνήμη του συστήματος. Οι καθυστερήσεις λοιπόν αυτές είναι δεδομένες και δυστυχώς με την υπάρχουσα δομή δεν μπορούν να περιοριστούν περαιτέρω. Σχετική ταύτιση έχουμε όσον αφορά και τα υπόλοιπα σημεία της εφαρμογής μας αλλά οι καθυστερήσεις είναι πολύ μικρές αναφορικά με την εφαρμογή και δε φαίνεται να επηρεάζουν την επίδοση του συστήματός μας. Πιο συγκεκριμένα η διαδικασία consult rules, το parsing του πίνακα δεδομένων και τα δύο ερωτήματα που πραγματοποιούμε στη rule engine κυμαίνονται σε τιμές 50,30,30 και 25 ms αντίστοιχα. Για το συγχρονισμό που απαιτείται στην εφαρμογή ώστε να επανέλθει η λειτουργία του video ύστερα από την εκτέλεση της αποκοπής, η καθυστέρηση κυμαίνεται από 1.5-3 ms, για την λήψη των αποφάσεων και την κλήση των λειτουργιών 0-3, για την παραγωγή τιμών αισθητήρων 0-1.5 ms ενώ για το συγχρονισμό για το crop και τη διακοπή η

καθυστέρηση είναι πάντα μηδενική. Συγκρίνοντας τώρα τις 3 περιπτώσεις στα συνολικά σημεία επισήμανσης, θα αντιληφθούμε αυτό που θα προστάζει και η απλή λογική. Αν και οι διαφορές όπως επισημίναμε είναι μικρές, παρόλα αυτά υπάρχει ένα μικρό προβάδισμα στη περίπτωση των 5 ms. Στη συνέχεια ακολουθεί η περίπτωση των 10 ms και τελευταία των 12 ms. Αυτό είναι λογικό καθώς όσο πιο μικρή η ορισμένη χρονική παράμετρος, τόσο περισσότερες φορές θα λαμβάνει η εφαρμογή τιμές αισθητήρων και επομένως θα “τρέχει” ερωτήματα στη rule engine με αποτέλεσμα να πραγματοποιούνται και περισσότερες λειτουργίες στο video. Για το λόγο αυτό θα πρέπει να είμαστε προσεκτικοί στον ορισμό του διαστήματος μέσα στο οποίο θα λαμβάνουμε τιμές για το περιβάλλον.

Πίνακας 7-7 :Σύγκριση επί τις εκατό %.

ΧΡΟΝΟΙ ΚΑΘΥΣΤΕΡΗΣΗΣ	ΔΙΑΦΟΡΑ	ΔΙΑΦΟΡΑ	ΔΙΑΦΟΡΑ
	CASE2-CASE1	CASE3-CASE1	CASE3-CASE2
Φόρτωση Rule engine	0,484	-0,174	-0,658
Consult rules	0,063	-0,015	-0,078
Parsing αρχείου metadata	0,746	-0,033	-0,779
Parsing αποτελεσμάτων από metadata	0,003	-0,014	-0,017
Query για το crop	0,033	-0,047	-0,08
Συγχρονισμός για το crop	0	0	0
Συγχρονισμός διακοπής	0	0	0
Συγχρονισμός επαναφοράς	0,016	0,017	0,001
Συγχρονισμός αισθητήρων	-0,015	-0,015	0
Queries στη rule engine	0,001	-0,064	-0,065
Εκτέλεση λειτουργιών	0,016	-0,016	-0,032
ΣΥΝΟΛΙΚΗ ΚΑΘΥΣΤΕΡΗΣΗ	1,347	-0,361	-1,708

Τελικά στον πίνακα 7-7 παρατηρούμε ότι στην περίπτωση των 5 ms έχουμε περισσότερη καθυστέρηση κατά 1.347% σε σχέση με την περίπτωση των 10ms και 1.708 % σε σχέση με τα 12 ms. Επίσης τα 10 ms επιβαρύνουν κατά 0.361% περισσότερο την εφαρμογή από ότι τα 12 ms. Παρόλα αυτά εμείς επιλέξαμε τη χρονικό διάστημα των 10 ms με σκοπούς καθαρά λειτουργικούς καθώς είχαμε ένα video διάρκειας 21 s και θέλαμε να παρουσιάσουμε κάποιες λειτουργίες μέσα σε αυτό.

7.2 ΣΥΜΠΕΡΑΣΜΑΤΑ

Στη συνέχεια της ενότητας αυτής θα συνοψίσουμε τα πλεονεκτήματα αλλά και τα προβλήματα της εφαρμογής μας.

7.2.1 ΘΕΤΙΚΑ

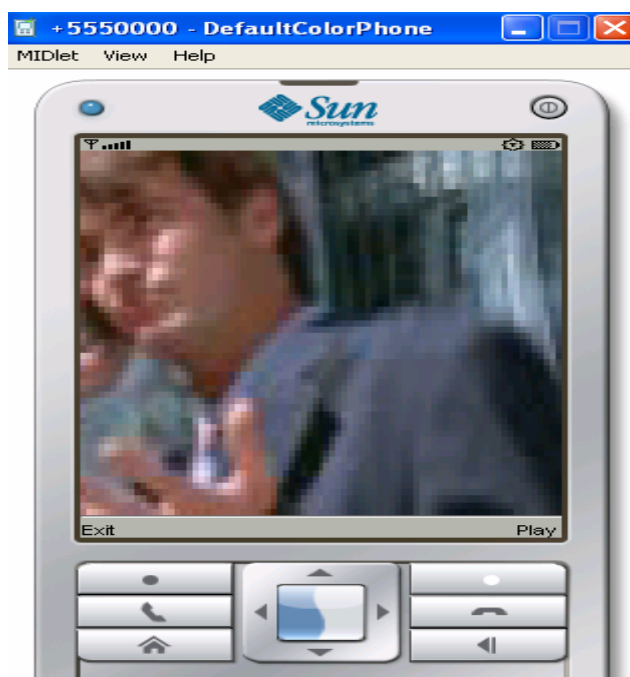
Ο στόχος ήταν να υλοποιήσουμε μια εφαρμογή κινητού τηλεφώνου που θα στηριζόταν σε context-aware πληροφορίες . Αυτό επιτεύχθηκε χάρη στο συνδυασμό mobile java, jprolog και mpeg-7 πρότυπο συμπίεσης. Κατασκευάσαμε μια εφαρμογή που μπορεί να εγκατασταθεί σε όλα τα σύγχρονα κινητά τηλέφωνα που υποστηρίζουν πλατφόρμα java MIDP 2.0 . Η συντριπτική πλειοψηφία πλέον των κινητών συσκευών στηρίζεται στη γλώσσα προγραμματισμού java, οπότε και το αντίκρισμα στην αγορά είναι ευρύ. Επίσης η λειτουργικότητά της μπορεί να εξαπλωθεί σε όλες τις εφαρμογές mobile video που παρουσιάσαμε αναλυτικά στο Κεφάλαιο 3. M-learning , διαφήμιση, επαγγελματική κατάρτιση και επιχειρήσεις έκτακτης ανάγκης μπορούν να επωφεληθούν από video που θα μπορεί να συντονίζεται με βάση τις περιβαλλοντικές συνθήκες και να προχωρεί στις υλοποιημένες λειτουργίες και κυρίως zoom in/zoom out και crop. Όπως είδαμε και παραπάνω η ποιότητα του ήχου, της εικόνας και του συγχρονισμού είναι άψογη ενώ και οι λειτουργίες εκτελούνται άμεσα χωρίς καθυστέρηση όπως είδαμε και στην προηγούμενη ενότητα με τις μετρήσεις. Η ακρίβεια εκτέλεσης είναι πολύ σημαντικός παράγοντας καθώς θα υπήρχε σημαντικό πρόβλημα εάν ο χρόνος αποκατάστασης μιας λειτουργίας ήταν μεγαλύτερος από το ρυθμό λήψης δεδομένων από τους αισθητήρες. Ουσιαστικά στην περίπτωση αυτή δεν θα ήταν εφικτό το πλάνο. Τέλος πρέπει να τονίσουμε και την επιτυχημένη σύνδεση της java με την prolog και την επιτυχημένη εφαρμογή των κανόνων. Χάρη στη διαγραφή των προηγούμενων δεδομένων που επιτύχαμε με την εντολή retract (βλ. Κεφάλαιο 6), καταφέραμε να διατηρήσουμε τη rule engine ανοιχτή στη διάρκεια της εφαρμογής και να λαμβάνουμε σωστά solutions χωρίς να επικαλύπτονται οι τιμές των αισθητήρων. Αυτό είναι πολύ σημαντικό καθώς γλυτώσαμε από ένα σημαντικό παράγοντα καθυστέρησης που είναι η δημιουργία της rule engine όπως είδαμε και στη προηγούμενη ενότητα με τις επιδόσεις.

7.2.2 ΠΡΟΒΛΗΜΑΤΑ

Κατά τη διάρκεια των σταδίων υλοποίησης αντιμετωπίσαμε κάποια προβλήματα όπως ήταν φυσικό. Κάποια από αυτά ήταν στο επίπεδο της υλοποίησης και καταφέραμε να τα ξεπεράσουμε μέσω του προγραμματισμού. Τα πλην επομένως της εφαρμογής είναι τα προβλήματα αυτά που δεν είχαν σχέση με τον κώδικα και επηρεάζουν τη λειτουργικότητα της εφαρμογής μας.

Το πρώτο πρόβλημα που μπορούμε να αναφέρουμε είναι το γεγονός ότι δε διαθέταμε video με ενσωματωμένα μεταδεδομένα όπως έχουμε επισημάνει και στην παράγραφο 6.1.3. Για το λόγο αυτό, έπρεπε να δημιουργήσουμε το δικό μας αρχείο, το οποίο όμως δε θα μπορούσαμε να το ενσωματώσουμε στο video μας έτσι ώστε να έχουμε ένα ενιαίο αρχείο με κινούμενη εικόνα και πληροφορία. Επίσης το αρχείο αυτό έπρεπε να στηρίζεται σε συγκεκριμένη μορφή που καθορίζεται από συγκεκριμένα καθορισμένα πρότυπα. Η απαίτηση λοιπόν ενός τέτοιου αρχείου metadata ήταν αρκετά σημαντική και δυσκόλεψε το έργο μας καθώς χρειάστηκε να προσθέσουμε μια επιπλέον βιβλιοθήκη KXML (βλ .6.1.3) που θα παρείχε τη δυνατότητα στο κινητό τηλέφωνο να διαχειρίζεται αυτά τα αρχεία. Παράλληλα όμως ίσως η αντιμετώπιση αυτού του προβλήματος να αποτελεί κι ένα εμπορικό πλεονέκτημα καθώς θα υπάρχει η δυνατότητα να ενσωματώνουμε κάθε φορά όσα αρχεία επιθυμούμε.

Το δεύτερο πρόβλημα που αντιμετωπίζουμε έχει να κάνει με την απόδοση της εικόνας του video όταν πραγματοποιούμε τη λειτουργία crop. Επειδή στην περίπτωση αυτή επισημαίνουμε μια συγκεκριμένη περιοχή και την φέρνουμε μεγεθυμένη στην οθόνη του κινητού, η ποιότητα του video δεν είναι πολύ καλή όπως μπορείτε να δείτε και στην εικόνα 7-2.



Εικόνα 7-2 :Λειτουργία αποκοπής

Τέλος το βασικότερο ίσως πρόβλημα που αντιμετωπίζουμε είναι η καθυστέρηση της εφαρμογής λόγω της rule engine και του parsing των μεταδεδομένων. Είδαμε ότι εισάγουν σημαντική καθυστέρηση κάτι που είναι πολύ αρνητικό για την περίπτωση ενός κινητού τηλεφώνου του οποίου ο χρήστης θέλει να το χειρίζεται άμεσα κι αποτελεσματικά. Το θετικό στο συγκεκριμένο κομμάτι είναι ότι η καθυστέρηση εμφανίζεται απλά κατά το ξεκίνημα της εφαρμογής και στη συνέχεια αυτά τα δύο στοιχεία δεν επηρεάζουν ούτε την ποιότητα του video ούτε την αλληλεπίδραση των αισθητήρων με τις λειτουργίες .

ΠΑΡΑΡΤΗΜΑ Α´

RULES.PL

% We need rules for the sensors.We define situations according
% the elements of the sensors.The sensors are 3: motion,luminance and
sound.

% We define the sensors with the X,Y,Z letters

% The values X,Y,Z will be loaded from external source

% load_file needed.

%-----

%Facts

%-----

%sound

sound(loud):- currentSound(Z),Z>50.

sound(quiet):- currentSound(Z),Z<50.

%luminance

luminance(high):-currentLuminance(Y),Y>50.

luminance(low):- currentLuminance(Y),Y<50.

%motion

motion(fast):-currentMotion(X),X>50.

motion(slow):-currentMotion(X),X<50.

```
%-----  
%Rules according to functions  
%-----  
  
%conversation  
action(mute) :- sound(quiet), motion(slow).  
%Outer space  
action(zoomIn) :- luminance(high).  
%Inside space  
action(zoomOut) :- luminance(low).  
%Running  
action(hide) :- motion(fast).  
%Normal  
action(appear) :- motion(slow).  
%Definition of a part  
%action(crop) :- sound(loud),luminance(high).  
action(crop,VS) :- event(talking,VS).  
%Hide image  
action(hide) :- sound(loud).
```

ΠΑΡΑΡΤΗΜΑ Β΄

PHANTOM.MP7.XML

```
<?xml version="1.0" encoding="UTF-8" ?>
<Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:mpeg:mpeg7:schema:2001
metadata\archives\MPEG7_Annotator\MPEG7_Annotator\Resources\XMLSch
ema\Mpeg7-2001.xsd">
<Description xsi:type="ContentEntityType">
<DescriptionMetadata>
  <Version>1.0</Version>
  <Comment>
    <FreeTextAnnotation />
  </Comment>
  <Creator>
    <Role href="creatorCS">
      <Name>Creator</Name>
    </Role>
    <Agent xsi:type="PersonType">
      <Name>
        <GivenName>Georgia</GivenName>
        <FamilyName>Katsamani</FamilyName>
      </Name>
      <Affiliation>
        <Organization>
          <Name>DI.UOA</Name>
        </Organization>
      </Affiliation>
      <Address>
        <PostalAddress>
          <AddressLine>Inffeldgasse 21a</AddressLine>
```

```
<AddressLine>8010 Graz</AddressLine>
<AddressLine>AUSTRIA</AddressLine>
</PostalAddress>
</Address>
<ElectronicAddress>
  <Email>grad0918@di.uoa.gr</Email>
</ElectronicAddress>
</Agent>
</Creator>
<CreationTime>2008-10-05T22:38</CreationTime>
<Instrument>
  <Tool>
    <Name>Caliph v0.9.25</Name>
  </Tool>
</Instrument>
</DescriptionMetadata>
```

```
<MultimediaContent xsi:type="VideoType">
<Video id="Test">
  <MediaInformation>
    <MediaProfile master="true">
      <MediaFormat>
        <Content href="video"/>
        <VisualCoding>
          <Pixel bitsPer="24"/>
          <Frame width="160" height="120" />
        </VisualCoding>
      </MediaFormat>
    <MediaInstance>
      <InstanceIdentifier />
    <MediaLocator>
```

```
<MediaUri>file:/D:/netbeans/ContextAwareMobile/src/video/phantom.mpg</MediaUri>
  </MediaLocator>
</MediaInstance>
</MediaProfile>
</MediaInformation>
<CreationInformation>
  <Creation>
    <Title>Creation information</Title>
    <Creator>
      <Role href="creatorCS">
        <Name>Creator</Name>
      </Role>
      <Agent xsi:type="PersonType">
        <Name>
          <GivenName>Georgia</GivenName>
          <FamilyName>Katsamani</FamilyName>
        </Name>
        <Affiliation>
          <Organization>
            <Name>DI-UOA</Name>
          </Organization>
        </Affiliation>
        <ElectronicAddress>
          <Email>grad0918@di.uoa.gr</Email>
        </ElectronicAddress>
      </Agent>
    </Creator>
  </Creation>
</CreationInformation>
<TextAnnotation>
  <FreeTextAnnotation>General Sites</FreeTextAnnotation>
</TextAnnotation>
```

```
<TemporalDecomposition>
<VideoSegment id="vs1">
<TextAnnotation>
<FreeTextAnnotation>Central Street</FreeTextAnnotation>
<StructuredAnnotation>
<WhatObject
href="http://polysema.di.uoa.gr/ont/wikipedia.owl#Discovery_Channel">
<Name>new york</Name>
</WhatObject>
</StructuredAnnotation>
</TextAnnotation>
<MediaTime>
<MediaTimePoint>T00:00:03</MediaTimePoint>
<MediaDuration>PT5S</MediaDuration>
</MediaTime>
</VideoSegment>

<VideoSegment id="vs2">
<TextAnnotation>
<FreeTextAnnotation>Gallery</FreeTextAnnotation>
<StructuredAnnotation>
<WhatObject
href="http://polysema.di.uoa.gr/ont/wikipedia.owl#Discovery_Channel">
<Name>calling</Name>
</WhatObject>
</StructuredAnnotation>
</TextAnnotation>
<MediaTime>
<MediaTimePoint>T00:00:08</MediaTimePoint>
<MediaDuration>PT2S</MediaDuration>
</MediaTime>
</VideoSegment>
```



```
<VideoSegment id="vs3">
  <TextAnnotation>
    <FreeTextAnnotation>This event symbolizes general talking in a
mobile</FreeTextAnnotation>
    <StructuredAnnotation>
      <WhatObject
href="http://polysema.di.uoa.gr/ont/wikipedia.owl#Discovery_Channel">
        <Name>talking</Name>
      </WhatObject>
    </StructuredAnnotation>
  </TextAnnotation>
  <MediaTime>
    <MediaTimePoint>T00:00:09</MediaTimePoint>
    <MediaDuration>PT2S</MediaDuration>
  </MediaTime>

  <SpatioTemporalDecomposition>
    <StillRegion>
      <SpatialLocator>
        <Polygon>
          <Coords xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001"
mpeg7:dim="8">95:90:242:290:242:129:93:129</Coords>
        </Polygon>
      </SpatialLocator>

      <VisualDescriptor xsi:type="DominantColorType">
        <ColorQuantization>
          <Component>R</Component>
          <NumOfBins>8</NumOfBins>
          <Component>G</Component>
          <NumOfBins>8</NumOfBins>
          <Component>B</Component>
          <NumOfBins>8</NumOfBins>
        </ColorQuantization>
```

```
<SpatialCoherency>31</SpatialCoherency>
<Value>
  <Percentage>31</Percentage>
  <Index>19 46 76</Index>
  <ColorVariance>0 0 0</ColorVariance>
</Value>
</VisualDescriptor>
</StillRegion>
</SpatioTemporalDecomposition>
</VideoSegment>

<VideoSegment id="vs4">
<TextAnnotation>
  <FreeTextAnnotation>Aeroplane</FreeTextAnnotation>
  <StructuredAnnotation>
    <WhatObject
      href="http://polysema.di.uoa.gr/ont/wikipedia.owl#Discovery_Channel">
      <Name>aeroplane</Name>
    </WhatObject>
  </StructuredAnnotation>
</TextAnnotation>
<MediaTime>
  <MediaTimePoint>T00:00:10</MediaTimePoint>
  <MediaDuration>PT5S</MediaDuration>
</MediaTime>
</VideoSegment>

</TemporalDecomposition>
</Video>
</MultimediaContent>
</Description>

</Mpeg7>
```

ΟΡΟΛΟΓΙΑ

Υπηρεσίες θέσης	Location Based Services
Ενήμερη υπηρεσία πλαισίου	Context aware service
Πλαίσιο/Περιεχόμενο	Context
Αισθητήρες	Sensors
Εξυπηρετητής	Server
Σταθμοί αισθητήρων	Sensor stations
Μορφή	Format
Σταθμός εργασίας	Workstation
Δίκτυο διακριτικών	Badge Network
Κανόνες	Rules
Πρώτης τάξης κατηγορήματα	First-order predicates
Γεγονός –Έλεγχος- Ενέργεια	Event-Control-Action
Υψηλού επιπέδου	High level
Ερεθίσματα	Trigger
Κεντρική υπηρεσία διαχείρισης περιεχομένου	Context management service
Πηγή περιεχομένου	Context source
Διακομιστής περιεχομένου	Context proxy

Αναλυτής περιεχομένου	Context reasoner
Αποδέκτης περιεχομένου	Context wrapper
Υπηρεσία αποθήκευσης περιεχομένου	Context storage service
Διασπαστής	Broker
Διασπαστής περιεχομένου	Context broker
Πλήκτρο	Tab
Αναπαραγωγή	Play
Πίσω	Rewind
Μπροστά	Forward
Πλήκτρο σίγασης	Tab mute
Εστίαση	Focus
Αποστολή	Send
Κινητή τηλεφωνία	Cave radio
Προστιθέμενης αξίας	Value added
Πέστροφα	Trout
Υπερσύνδεσμοι	Links
Διαφήμιση για προϊόντα των οποίων τα καταστήματα θα βρισκονται στη γύρω περιοχή	Proximity-based marketing
Video κινητού τηλεφώνου για	emergency mobile videos

έκτακτες ανάγκες	
Πραγματικού χρόνου	Real-time
Ενημερότητα περιεχομένου	Context-awareness
Ερμηνευτές	Interpreters
Ευαισθησία	Sensitivity
Μη ευαισθησία	Insensitivity
Εξυπηρετητής περιεχομένου	Context-server
Ερμηνευτής περιεχομένου	Context-Interpreter
Δρομολογητής	Router
Πνευματική κατάσταση	Mental state
Θόρυβος	Noise
Φως	Light
Θέση στο χώρο	POS
Ποιότητα σήματος	QoS
Οθόνη κινητού	Context monitor
Πλατφόρμα διαχείρισης	Adaptation manager
Προσαρμοσμένη υπηρεσία περιεχομένου	Context Adapted Service
Περιγραφικά ή μεταδεδομένα περιεχομένου	Descriptive or Content metadata
Τεχνικά μεταδεδομένα	Technical metadata
Μεταδεδομένα διαχείρισης	Administrative metadata
Διαχειριστή οθόνης	Display manager
Μηχανή κανόνων	Rule engine
Γεγονότα	Facts
Πραγματικού χρόνου	Runtime
Ερωτήματα	Queries
Διαμόρφωση	Configuration
Πολυμέσα	Multimedia

Η μετάδοση ήχου μέσω δικτύου	Stream audio
Στιγμιότυπα	Instances
Ροές	Sources
Μη πραγματοποιήσιμη	Unrealized
Πραγματοποιήσιμη	Realized
Προδημιουργημένη	Prefetched
Δημιουργημένη	Started
Κλειστή	Closed
Εξαίρεση	Exception
Απαντήσεις	Answers
Μηχανή τυχαίας αναπαραγωγής αριθμών	Generator
Τομέα	Frame
Παιδιά	Childs
Χαρακτηριστικά	Attributes
Αντικείμενα	Items
Γεγονός της ομιλίας σε κινητό τηλέφωνο	Calling event
Διαδίκτυο	Internet
Κινούμενη εικόνα	Video
Παιχνίδια	Games

ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ

ECA	Event – Action –Control
ECA-DL	Event – Action –Control Data Language
CLIPS	" C Language Integrated Production System
GPS	Global Positioning System
m-Kavaad	Mobile Kavaad
GSM	Global System for Mobile communications
SMS	Short Message Service
2.5 G	2.5th Generation
3 G	3rd Generation
3GPP	3rd Generation Partnership Project
MPEG-4	Moving Picture Experts Group -4
RTSP	Real Time Streaming Protocol
m-learning	Mobile learning
LAN	Local Area Network
WLAN	Wireless Local Area Network
WiMAX	Worldwide Interoperability for Microwave Access
Wi-Fi	Wireless Fidelity
QoS	Quality of Service
J2ME	Java Platform, Micro Edition
JVM	Java Virtual Machine
PDA	Personal Digital Assistant

CLDC	Connected Limited Device Configuration
CDC	Connected Device Configuration
MIDP	Mobile Information Device Profile
GUI API	Graphical User Interface Application Programming Interface
IMP	Information Module Profile
FP	Foundation Profile
PBP	Personal Basic Profile
PP	Personal Profile
MMAPI	Mobile Media API
JIP2ME	Java Interactive Profiler Micro Edition
JCP	Java Community Process
JIP	Java Interactive Profiler
kXML	k-Extensible Markup Language

ΑΝΑΦΟΡΕΣ

- [1] <http://context.media.mit.edu/press/> (v. January 2009).
- [2] G. D. Abowd ,C. G. Atkeson, J. Hong, S. Long, R. Kooper and Mike Pinkerton, in Wireless Networks ,
Cyberguide, a mobile context-aware tour guide.
- [3] Anxo Cereijo Roibás SCMIS, University of Brighton (UK) & Nina Sabnani National Institute of Design (India) ,
A mobile interactive multimedia system to encourage activist discussion and social participation.
- [4] http://en.wikipedia.org/wiki/Nathan_Stubblefield (v. January 2009).
- [5] http://en.wikipedia.org/wiki/Bell_Labs (v. January 2009).
- [6] <http://en.wikipedia.org/wiki/OG> (v. January 2009).
- [7] <http://en.wikipedia.org/wiki/1G> (v. January 2009).
- [8] <http://en.wikipedia.org/wiki/GSM> (v. January 2009).
- [9] http://www.phonemag.com/images/uploads/others/mobile_video.jpg
(v. January 2009).
- [10] Judy Brown, MASIE Fellow Learning 2007 -October 22, 2007 ,
MOBILE LEARNING: 101.
- [11] <http://mayolounge.blogspot.com/2008/12/jim-beam-launches-integrity-character.html>(v. April 2009).
- [12] <http://www.emergencymobilevideos.com/index.htm> (v. January 2009).
- [13] Anind K. Dey, Gregory D. Abowd and Daniel Salber ,
A Context-Based Infrastructure for Smart Environments.
- [14] Jacqueline Floch, Svein Hallsteinsen, Arne Lie and Hans I. Myrhaug,
A Reference Model for Context-Aware Mobile Services.

- [15] http://developer.apple.com/leopard/overview/images/screenshot_dashcode.jpg (v. January 2009).
- [16] http://edndoc.esri.com/arcobjects/9.2/NET_Server_Doc/developer/ADF/graphics/common_resource_impl.png (v. January 2009).
- [17] http://www1.i2r.a-star.edu.sg/~tgu/gutao/socam_figure2.jpg (v. January 2009).
- [18] http://electronicdesign.com/Files/29/18911/Figure_01.jpg (v. January 2009).
- [19] http://www.laboptic.com/images/detecteurs_accueil_clair.jpg (v. January 2009).
- [20] http://www.sciencescope.co.uk/images/New_Sound.jpg (v. January 2009)
- [21] <http://www.speedometersolutions.com/images/vintage%20speedometer.jpg> (v. January 2009).
- [22] <http://www.skynetsecurity.ie/images/shock1.jpg> (v. January 2009).
- [23] http://www.inition.co.uk/inition/images/product_mocaptrack_logitech_head_tracker.jpg (v. January 2009).
- [24] http://en.wikipedia.org/wiki/Java_Platform,_Micro_Edition (v. January 2009).
- [25] Vikram Goyal , Pro Java ME MMAPi.
- [26] <http://swiki.hfbk-hamburg.de:8888/MusicTechnology/747> (v. March 2009).
- [27] www.ugosweb.com/jiprolog/index.aspx (v. December 2008).
- [28] <http://www.ugosweb.com/jiprolog/index.aspx> (v. December 2008).
- [29] Enabling Context-Aware Mobility. PDF from <http://www.cisco.com/go/contextaware>
- [30] The Technologies behind a Context-Aware.pdf from <http://www.cisco.com/go/contextaware>
- [31] <http://en.wikipedia.org/wiki/Sensor> (v. January 2009).

- [32] http://en.wikipedia.org/wiki/List_of_sensors (v. January 2009).
- [33] Ugo Chirico JIProlog 3.0.3 Reference Manual.pdf.
- [34] <http://www.ibm.com/developerworks/ibm/library/i-lbs/>
- [35] Guanling Chen and David Kotz , A Survey of Context-Aware Mobile Computing Research.
- [36] Roy Want¹, Andy Hopper², Veronica Falcão³ and Jonathan Gibbons⁴ , The Active Badge Location System.
- [37] http://en.wikipedia.org/wiki/Proximity_marketing (v. February 2009).
- [38] <http://robotlab.csie.ntu.edu.tw/people/~bipo/An%20infrastructure%20for%20context-awareness%20based%20on%20first%20order%20logic.pdf> (v. May 2009).
- [39] Laura Daniele, Patrícia Dockhorn Costa, and Luís Ferreira Pires Towards a Rule-Based Approach for Context-Aware Applications*.
- [40] <http://www3.cs.utwente.nl/~wegdam/papers/sinderen-vanHalteren-wegdam-meeuwissen-eertink-Supporting-Context-aware-Mobile-Application-an-Infrastructure-Based-Approach-IEEEComMag-Sept2006.pdf> (v. May 2009).