



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΑΦΑΝΗΣ ΠΑΡΑΚΟΛΟΥΘΗΣΗ ΑΛΛΗΛΕΠΙΔΡΑΣΕΩΝ
ΧΡΗΣΤΗ ΜΕ ΣΥΣΤΗΜΑ ΜΕ ΣΤΟΧΟ ΤΗΝ ΠΡΟΒΛΕΨΗ ΤΩΝ
ΕΝΕΡΓΕΙΩΝ ΤΟΥ ΣΕ ΠΕΡΙΒΑΛΛΟΝ ΔΙΕΙΣΔΥΤΙΚΗΣ
ΥΠΟΛΟΓΙΣΤΙΚΗΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Νίκος Κ. Χάματσος

Επιβλέπουσα : Ιωάννα Ρουσσάκη
Λέκτορας Ε.Μ.Π

Αθήνα, Οκτώβριος 2009



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΑΦΑΝΗΣ ΠΑΡΑΚΟΛΟΥΘΗΣΗ ΑΛΛΗΛΕΠΙΔΡΑΣΕΩΝ
ΧΡΗΣΤΗ ΜΕ ΣΥΣΤΗΜΑ ΜΕ ΣΤΟΧΟ ΤΗΝ ΠΡΟΒΛΕΨΗ ΤΩΝ
ΕΝΕΡΓΕΙΩΝ ΤΟΥ ΣΕ ΠΕΡΙΒΑΛΛΟΝ ΔΙΕΙΣΔΥΤΙΚΗΣ
ΥΠΟΛΟΓΙΣΤΙΚΗΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Νίκος Κ. Χάματσος

Επιβλέπουσα : Ιωάννα Ρουσσάκη
Λέκτορας Ε.Μ.Π

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την

.....
Ιωάννα Ρουσσάκη
Λέκτορας Ε.Μ.Π

.....
Μιλτιάδης Αναγνώστου
Καθηγητής ΕΜΠ

.....

Αθήνα, Οκτώβριος 2009

.....
Νίκος Κ. Χάματσος

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

Copyright © Νίκος Κ. Χάματσος, 2009

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

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

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

ΠΕΡΙΛΗΨΗ

Το pervasive computing είναι ένα μοντέλο αλληλεπίδρασης ανθρώπου και υπολογιστή στο οποίο η επεξεργασία της πληροφορίας είναι εξολοκλήρου ενσωματωμένη σε αντικείμενα καθημερινότητας καθώς και σε καθημερινές δραστηριότητες. Ασχολούμενος με συνηθισμένες δραστηριότητες, κάποιος που χρησιμοποιεί pervasive computing απασχολεί πολλές υπολογιστικές συσκευές και συστήματα ταυτόχρονα, και δεν είναι απαραίτητο να γνωρίζει ότι τα χρησιμοποιεί. Αυτό το μοντέλο θεωρείται μια εξέλιξη του επιτραπέζιου υπολογιστή. Βασικό χαρακτηριστικό είναι το context aware καθώς και η ικανότητα για εξαγωγή συμπερασμάτων από τις διάφορες πληροφορίες που συλλέγονται.

Η ιδέα του pervasive computing βασίζεται σε μια απλή ιδέα που με τα πλεονεκτήματα της τεχνολογίας, ο υπολογιστικός εξοπλισμός θα γίνει μικρότερος (συρρικνωθεί) και θα γίνει πιο δυνατός. Αυτό θα επιτρέψει σε μικρές συσκευές να γίνουν ευχρηστες παντού, ενσωματωμένες και σχεδόν αόρατες στην καθημερινή χρήση από τον άνθρωπο και έτσι θα προσφέρουν μια εύκολη και συνεχή πρόσβαση σε υπολογιστικά περιβάλλοντα.

Στην παρούσα διπλωματική εργασία ασχολούμαστε περισσότερο με την καταγραφή πληροφοριών για τις εφαρμογές που παρέχουν γραφικό περιβάλλον στον χρήστη για να αλληλεπιδρά με αυτές. Συγκεκριμένα καταγράφουμε το όνομα του παραθύρου καθώς και την ώρα έναρξης και ώρα τερματισμού του παραθύρου καθώς και της αντίστοιχης διεργασίας. Η καταγραφή των δεδομένων γίνεται σε δύο απλά αρχεία κειμένου. Στο ένα αρχείο φυλάγονται πληροφορίες για τα παράθυρα που τερματιστηκαν, ενώ στο άλλο αρχείο φυλάγονται προσωρινά πληροφορίες για τις ήδη τρέχοντες διεργασίες με τα αντίστοιχα παράθυρα που δημιουργούν.

Για την υλοποίηση της εφαρμογής έγινε χρήση της γλώσσας προγραμματισμού C++ και η εφαρμογή είναι συμβατή με το λειτουργικό σύστημα Microsoft Windows XP και Microsoft Windows Vista.

Τέλος γίνεται μια μικρή στατιστική ανάλυση στα δεδομένα που συλλέκτηκαν με σκοπό να μπορέσουμε να εξάγουμε χρήσιμα συμπεράσματα για τις διάφορες συνήθειες του χρήστη και την αλληλεπίδραση που έχει με το σύστημα.

Λέξεις κλειδιά:

Εξαγωγή συμπερασμάτων, καταγραφή πληροφοριών, pervasive computing

ABSTRACT

Pervasive computing is a post-desktop model of human-computer interaction in which information processing has been thoroughly integrated into everyday objects and activities. In the course of ordinary activities, someone "using" ubiquitous computing engages many computational devices and systems simultaneously, and may not necessarily even be aware that they are doing so. This model is usually considered advancement from the desktop paradigm. Context awareness is the main characteristic so as context inference based on context history management.

The idea behind pervasive computing is based on the technological benefits where the computing equipment will grow smaller and stronger. This will allow to the small devices to become more usefull, embedded and almost invisible to the daily use of the human so that they will offer an easy and continuous access to computing environments.

In this current diploma, we focus on the windows monitoring gathering information about applications with graphical user interface and the interaction of the user. Specifically we gather information about the name of the window, start time and termination time of the window. This information is saved into two text files, one with the terminated processes and their windows, and into the second file we save the information about running processes.

For the development of the application we used the programming language C++ and the application is compatible with Microsoft Windows XP and Microsoft Windows Vista.

Finally a statistical analysis of the gathered information is done so we can predict the user's intentions in the future based on this information.

KeyWords

Pervasive computing, Context awareness, Context inference, Context history managment

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να εκφράσω την ευγνωμοσύνη μου σε όλους όσους στάθηκαν αρωγοί και συνετέλεσαν, άμεσα ή έμμεσα, στην ολοκλήρωση της συγκεκριμένης εργασίας.

Ευχαριστώ την επιβλέπουσα καθηγήτριά μου, κ. Ιωάννα Ρουσσάκη, λέκτορα του Εθνικού Μετσόβιου Πολυτεχνείου, που μου έδωσε την ευκαιρία να ασχοληθώ με το συγκεκριμένο θέμα καθώς επίσης και για την πολύτιμη βοήθεια που μου προσέφερε σε δύσκολες στιγμές κατά τη διάρκεια εκπόνησης της διπλωματικής μου εργασίας καθώς και τον υποψήφιο διδάκτορα κ. Νίκο Καλατζή.

Χωρίς την προσωπική ενασχόληση των δύο προαναφερθέντων προσώπων, η συγκεκριμένη εργασία δε θα μπορούσε να εξελιχθεί και να πάρει τη σημερινή της μορφή.

Τέλος, θα πρέπει να σταθώ στην ηθική και ψυχολογική συμπαράσταση που μου παρείχαν η οικογένεια και οι φίλοι μου, καθώς και τις όποιες σχετικές γνώσεις τους μέχρι την ολοκλήρωση της διπλωματικής μου εργασίας.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ	5
ABSTRACT	6
ΕΥΧΑΡΙΣΤΙΕΣ.....	7
ΚΕΦΑΛΑΙΟ 1	10
ΕΙΣΑΓΩΓΗ	10
1.1 Γενικά	10
1.2 Στόχοι και Προσέγγιση	11
1.3 Οργάνωση εργασίας.....	11
ΚΕΦΑΛΑΙΟ 2	12
PERVASIVE COMPUTING	12
2.1 Γενικά	12
2.2 Context awareness.....	13
2.3 Context Inference	16
2.3 Context History.....	19
2.3.1 Γενικά	19
2.3.2 Σχετική έρευνα με context history	20
2.4 State of Art.....	21
ΚΕΦΑΛΑΙΟ 3	26
Υλοποίηση Εφαρμογής.....	26
3.1 Ψευδοκώδικας.....	26
3.2 Επεξήγηση Υλοποίησης Εφαρμογής	31
3.2.1 Κυρίως Πρόγραμμα.....	31
3.2.2 Η συνάρτηση EnumWindowsProc()	32
3.2.3 Η συνάρτηση CreateLog()	33
3.2.4 Βοηθητική συνάρτηση convertInt ()	35
3.2.5 Βιβλιοθήκες- Επικεφαλίδες (libraries-headers)	36
3.2.6 Προδιαγραφές συστήματος.....	38
3.2.7 Εικόνες Χρήσης της εφαρμογής	39
ΚΕΦΑΛΑΙΟ 4	41
Ανάλυση αποτελεσμάτων	41
4.1 Γενικά	41
4.2 Ανάλυση αποτελεσμάτων.....	41

ΚΕΦΑΛΑΙΟ 5	46
ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ	46
5.1 Συμπεράσματα	46
5.2 Μελλοντική Εργασία	47
Παράρτημα Α	48
Παράρτημα Β	58
ΒΙΒΛΙΟΓΡΑΦΙΑ	64

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

1.1 Γενικά

Έχοντας εισβάλει στην καθημερινότητα μας οι υπολογιστές αποτελούν ένα αναπόσπαστο κομμάτι της ζωής μας. Η ραγδαία εξέλιξη των υπολογιστικών συστημάτων άνοιξε νέους ορίζοντες πέρα από την φαντασία του ανθρώπου. Επίσης η μεγάλη εξέλιξη στις κινητές και ασύρματες συσκευές τις έκαναν μέρος της καθημερινότητας του ανθρώπου. Το όνειρο του Weiser άρχισε να γίνεται πραγματικότητα[1]. Οι συσκευές άρχισαν να εκτελούν έξυπνες εργασίες εκ μέρους του χρήστη. Είναι αυτο που ονομάζουμε context-awareness[2]. Context awareness είναι η γνώση του περιβάλλοντα χώρου συμπεριλαμβανομένου ιδίου του περιβάλλοντος, των ανθρώπων που το περιβάλλει και των εφαρμογών[3]. Έχοντας υπόψιν τα παραπάνω, αναφέρουμε την έννοια των pervasive systems, δηλαδή συστήματα που βρίσκονται παντού.

Το context awareness είναι ο κύριος σκοπός αυτών των συστημάτων, που έχουν σαν κύριο μέλημα να προσφέρουν έξυπνη και αποτελεσματική ανακάλυψη και συνάμα παράδοση των υπηρεσιών τους σε ένα κινούμενο και συχνά αφελή για την ύπαρξη τους χρήστη[4]. Οι χρήστες πλέον δεν είναι περιορισμένοι από την εύρεση διάφορων υπηρεσιών καθώς και από διάφορα προβλήματα ενσωμάτωσης των υπηρεσιών αυτών στις ανάγκες τους. Η κινητικότητα και η φύση του χρήστη που είναι απροσδιόριστη απαιτεί ένα μηχανισμό που όχι μόνο μπορεί αν προβλέψει τις διάφορες προτιμήσεις του χρήστη αλλά ταυτόχρονα να μπορεί να προσαρμοστεί σε διάφορες ανωμαλίες που μπορούν να συμβούν αναπάντεχα.

Ακόμα ένα ζωτικό σημείο των pervasive συστημάτων είναι η φύλαξη παλαιότερων πληροφοριών και η μετέπειτα επεξεργασία τους. Αυτό μπορεί να βοηθήσει στην εύρεση διάφορων συνηθειών του χρήστη καθώς και στην ευκολότερη πρόβλεψη των προτιμήσεων του έχοντας καθορίσει από προηγούμενες πράξεις του ένα συμπεριφοριακό μοτίβο. Επίσης αυτές οι πληροφορίες είναι ιδιωτικές γι' αυτό μέτρα ασφαλείας πρέπει να υπάρχουν για να είναι ασφαλείς.

1.2 Στόχοι και Προσέγγιση

Στόχος αυτής της διπλωματικής εργασίας είναι η υλοποίηση μιας εφαρμογής που θα τρέχει σε λειτουργικό σύστημα Windows XP και Windows Vista και θα καταγράφει τα παράθυρα με τα οποία ο χρήστης αλληλεπιδρά. Αναλυτικότερα, θα καταγράφονται διάφορες πληροφορίες για τις διεργασίες που αλληλεπιδρούν με τον χρήστη μέσω γραφικού περιβάλλοντος. Η καταγραφή γίνεται σε δύο αρχεία κειμένου όπου η δημιουργία τους γίνεται με την έναρξη της εφαρμογής. Η υλοποίηση της εφαρμογής έγινε σε περιβάλλον Microsoft Visual Studio Express Edition σε γλώσσα προγραμματισμού C++. Δεύτερος στόχος της εργασίας ήταν η επεξεργασία των αποτελεσμάτων που θα καταγραφούν από την εφαρμογή με μια απλή στατιστική ανάλυση. Κάνοντας αυτή την ανάλυση σκοπός ήταν η εύρεση κάποιου μοτίβου στις συνήθειες του χρήστη που η χρήση του θα διευκόλυνε στην πρόβλεψη μελλοντικών ενεργειών και διευκόλυνση αυτών από το ίδιο το σύστημα.

1.3 Οργάνωση εργασίας

Το υπόλοιπο της εργασίας είναι οργανωμένο ως εξής: στο Κεφάλαιο 2 γίνεται μια πρώτη γνωριμία με διάφορες ορολογίες και διάφορους σημαντικούς παράγοντες για τα pervasive συστήματα, στο Κεφάλαιο 3 γίνεται μια επεξήγηση της εφαρμογής με μερική ανάλυση του κώδικα, στο Κεφάλαιο 4 παρουσιάζεται μια στατιστική ανάλυση των πληροφοριών που συλλέξαμε με την εφαρμογή και τέλος στο Κεφάλαιο 5 μια σύντομη αναφορά στην εργασία καθώς και σε διάφορες μελλοντικές εργασίες που θα μπορούσαν να γίνουν.

ΚΕΦΑΛΑΙΟ 2

PERVASIVE COMPUTING

2.1 Γενικά

Pervasive computing είναι ένας από τους πολλούς τομείς της επιστήμης της πληροφορικής όπου η ακαδημαϊκή έρευνα (όπως context awareness και αποτελεσματικά πρωτότοκλα τηλεπικοινωνιών) ενσωματώνεται απευθείας στην βιομηχανία προσφέροντας πολλά προτερήματα σε αυτές όπως για παράδειγμα λογισμικό και hardware για διάφορες κινητές συσκευές και για ενσωματωμένα συστήματα. Αυτό το γεγονός όμως έχει σαν αποτέλεσμα όχι μόνο την ευρεία διάθεση υλικών πλατφόρμων (hardware platforms) για έρευνα, αλλά και στην ραγδαία χρησιμοποίηση των ακαδημαϊκών ερευνητικών αποτελεσμάτων από την βιομηχανία[5,6].

Η ιδέα του pervasive computing βασίζεται σε μια απλή ιδέα που με τα πλεονεκτήματα της τεχνολογίας, ο υπολογιστικός εξοπλισμός θα γίνει μικρότερος (συρρικνωθεί) και θα γίνει πιο δυνατός. Αυτό θα επιτρέψει σε μικρές συσκευές να γίνουν ευχρηστες παντού, ενσωματωμένες και σχεδόν αόρατες στην καθημερινή χρήση από τον άνθρωπο και έτσι θα προσφέρουν μια εύκολη και συνεχή πρόβαση σε υπολογιστικά περιβάλλοντα.

Όπως και σε κάθε άλλη εξελισσόμενη ερευνητική περιοχή, έτσι και σε αυτή την περιοχή των pervasive systems υπάρχουν πολλά προκλητικά προβλήματα. Ένα χαρακτηριστικό πρόβλημα συναντάται στο παρακάτω ανοικτό ερώτημα, πώς θα σιγουρευτούμε και θα εξασφαλίσουμε ότι το υπολογιστικό σύστημα είναι ολοκληρωμένο, αόρατο και ενσωματωμένο στο περιβάλλον καθώς και πώς θα ελαχιστοποιήσουμε την πιθανή επιρροή διεισδυτικότητας στην αντίληψη του χρήστη.

Μελετώντας το παραπάνω ερώτημα καταλήγουμε σε δύο κοινές προσεγγίσεις για την επίλυση του προβλήματος της διακριτικότητας σε ένα pervasive system: κάνοντας τις συσκευές μικρές σε μέγεθος και όγκο και ενσωματώνοντας την λογική του συστήματος σε εφαρμόσιμες κινητές συσκευές καθώς και μέσα στο περιβάλλον που τις περικλείει. Έτσι επιτυγχάνεται ως ένα σημείο η εξυπνάδα του συστήματος να αντιληφθεί μελλοντικές ενέργειες του χρήστη μέσω των συμφραζόμενων των παραγόντων του περιβάλλοντος. Σαν αποτέλεσμα, ένα τέτοιο pervasive system θα κρύβεται στο παρασκήνιο και ο χρήστης θα αλληλεπιδρά μαζί του μόνο κατά την διάρκεια του ελεύθερου του χρόνου χωρίς να τον επιβαρύνει και να επιβραδύνει στις διάφορες εργασίες ή δουλειές του.

Εκτός από την προσπάθεια των ειδικών για την υλοποίηση ενός φυσικά αόρατου pervasive computing system, η γενική φιλοσοφία πίσω από την όλη έρευνα υποδηλώνει ότι το σύστημα θα πρέπει να είναι και λογικά αόρατο. Αυτό μπορεί να επιτευχθεί με την προϋπόθεση ότι το σύστημα μπορεί αν προσαρμοστεί από μόνο του στα διάφορα μοτίβα συμπεριφοράς που επιδεικνύονται από τους χρήστες και από τους παράγοντες του περιβάλλοντος, όπως είναι ο χρόνος, η τοποθεσία, διάφορα αντικείμενα και άλλα συστήματα που συνυπάρχουν στον χώρο. Τέτοιες ικανότητες μπορούν να επιτευχθούν με το context awareness που είναι απαραίτητο στοιχείο για να μπορεί το σύστημα να είναι λογικά αόρατο. Έτσι εξασφαλίζεται και η ιδιότητα του συστήματος να είναι και φυσικά αόρατο.

2.2 Context awareness

Είναι ένας ζωτικός παράγοντας για να μπορούν τα pervasive systems να είναι αόρατα στον χρήστη. Μπορούν να προσαρμόσουν τις ιδιότητες και την συμπεριφορά τους σύμφωνα με τις πληροφορίες της συγκεκριμένης κατάστασης του χρήστη, όπως η ψυχολογική κατάσταση, διάφορα μοτίβα συμπεριφοράς, γεωγραφική θέση ή από τις ιδιότητες του περιβάλλοντα χώρου όπως ώρα της ημέρας, κοντινοί χρήστες, αντικείμενα και άλλα συστήματα.

Σαν αποτέλεσμα της ανάπτυξης καλύτερων μηχανισμών για την χρήση context awareness, τα pervasive systems είναι ικανά να επεκτείνουν τις ικανότητες και την έκταση των υπηρεσιών τους σε ένα πιθανώς περιορισμένο αριθμό τοπικών διαθέσιμων πόρων[7]. Για παράδειγμα, βασισμένο στην τοποθεσία και την διαθεσιμότητα, μια context aware

υπηρεσία με περιορισμένους πόρους μπορεί να είναι ικανή για off-load εντατικές λειτουργίες επεξεργαστή ή αποθήκευση σε κοντινά συστήματα με αυξημένους πόρους. Τα σημερινά pervasive συστήματα είναι ικανά για ενσωμάτωση ενός αριθμού πηγών από context πληροφορίες και να προσαρμόσουν την συμπεριφορά τους σε πιθανές απρόβλεπτες καταστάσεις[8]. Ένα σύστημα πλοήγησης αυτοκινήτου που γνωρίζει το απόθεμα βενζίνης του αυτοκινήτου καθώς και την μάρκα προτίμησης και τον τύπο βενζίνης που χρησιμοποιεί το αυτοκίνητο, μπορεί να ειδοποιήσει τον οδηγό όταν είναι σε συγκεκριμένη απόσταση από ένα βενζινάδικο που έχει τον συγκεκριμένο τύπο βενζίνης και σε καλή τιμή.

Το να είναι αόρατη η ύπαρξη των pervasive συστημάτων εξαρτάται από το πόσο επιτυχημένα υλοποιεί ένα συγκεκριμένο επίπεδο νοημοσύνης που θα επιτρέπει του χρήστη να επικεντρώνεται σε άλλα πράγματα παρά να αλληλεπιδρά με το σύστημα[9]. Context aware συστήματα που χρησιμοποιούν αυτού του είδους την νοημοσύνη συχνά υπολογίζουν την πρόθεση του χρήστη βασισμένα σε ένα αριθμό συμφραζόμενων (context). Ως αποτέλεσμα, το context awareness βοηθά στο να μένει συγκεντρωμένος ο άνθρωπος στον σκοπό του προβλέποντας τις αντιδράσεις του χρήστη σε σχετικά γεγονότα.

Περισσότερη έρευνα για τον υπολογισμό και πρόβλεψη των προθέσεων του χρήστη έχει γίνει από πολλούς τομείς. Στα κινητά και στα adhoc δίκτυα, η διακοπή εξυπηρέτησης του χρήστη συχνά προβλέπεται βασιζόμενη σε παλαιότερες συμπεριφορές διαίεταγωγής και τωρινής θέσης. Αυτή η επεξεργασία είναι πρωταρχικά βασισμένη στην τοποθεσία και ελέγχεται από την δομή του δικτύου[10,11]. Pervasive computing συστήματα πρέπει να προβλέψουν τις κινήσεις του χρήστη και να προσαρμοστούν καθώς πρέπει να διατηρήσουν μια προσεχτική ισορροπία μεταξύ της παραγωγικότητας τους και στο να μείνουν αόρατα στον χρήστη. Η διατήρηση ενός βαθμού της ιδιότητας τους αν είναι αόρατα στον χρήστη αποτελεί μια σημαντική προϋπόθεση για ένα Pervasive computing σύστημα. Αν μια εφαρμογή προβλέψει την πρόθεση του χρήστη λάθος, αναποτελεσματικά ή αδέξια, τότε η παρουσία του συστήματος μπορεί να αποκαλυφθεί. Για να γίνουν τα πράγματα χειρότερα, προσαρμόζοντας λάθος συμπεριφορά του συστήματος, ο χρήστης αποσπάται και χρειάζεται να προσαρμόσει το σύστημα αντι να συγκεντρώνεται στον σκοπό του. Όπως και να είναι, αν το σύστημα δεν είναι τόσο αποτελεσματικό στην πρόβλεψη των προθέσεων του χρήστη, τότε το σύστημα μπορεί να συμβιάσει την ιδέα του context-awareness επειδή η συμπεριφορά του δεν θα γίνει αντιληπτή από τον χρήστη. Για παράδειγμα, έστω ένας χρήστης βλέπει μια ροή δεδομένων βίντεο από το διαδίκτυο σε μια κινητη ασύρματη

συσκευή[12]. Καθώς ο χρήστης μετακινείται στον χώρο, αν το διαθέσιμο εύρος ζώνης μειώνεται ή το σήμα εξασθενεί, τότε το pervasive σύστημα πρέπει να έχει την δυνατότητα να προσαρμοστεί δυναμικά σε αυτές τις συνθήκες χωρίς να ειδοποιηθεί ο χρήστης. Αν το εύρος ζώνης μειωθεί τότε μπορεί το σύστημα απλά να ενωθεί σε άλλη διαθέσιμη συνδεση ή να προσφέρει στον χρήστη το βίντεο με λιγότερη ευκρίνεια. Και οι δύο λύσεις είναι καλές, αλλά μια απο αυτές είναι προτιμότερη βασιζόμενη στην πρόθεση του χρήστη.

Η ικανότητα του συστήματος να προσαρμόζεται σε μεταβλητούς παράγοντες στο περιβάλλον του είναι ένας ακόμα σημαντικός παράγοντας της ιδιότητας των pervasive συστημάτων να είναι αόρατα στον χρήστη. Τυπικά, pervasive συστήματα πρέπει να προσαρμόζονται στις αλλαγές της διαθεσιμότητας των πόρων που τους διατίθενται χωρίς να αποκαλύπτουν τις ελλείψεις στην διάθεση των συγκεκριμένων πόρων όπως εύρος ζώνης των δικτύων, διαθέσιμη μνήμη ή επεξεργαστικούς κύκλους. Εξαιτίας της φύσης της ασύρματης επικοινωνίας (καθυστέρηση λόγω ατμοσφαιρικών συνθηκών ή φυσικών διαταραχών) είναι δύσκολο να παρέχεις εγγυημένη συνδεσιμότητα στα pervasive computing περιβάλλοντα[10]. Για αυτό τον λόγο πρέπει να λειφθούν κάποια προληπτικά μέτρα για να διασφαλιστεί η αποτελεσματικότητα και η σωστή ροή δεδομένων. Ένας αριθμός στρατηγικών έχει προταθεί για τον σκοπό αυτό[13]. Η εσωτερική δομή του συστήματος μπορεί δυναμικά να προσαρμόζει την ακρίβεια της μετάδοσης βασιζόμενη στους διαθέσιμους πόρους. Ένα pervasive σύστημα που ανιχνεύει λιγότερους διαθέσιμους πόρους για μια συσκευή, μπορεί να επιλέξει να μειώσει την ακρίβεια της μετάδοσης πχ να προσαρμόσει την ποιότητα της ποιής δεδομένων μορφής βίντεο σε μια κατώτερη ή σε μια πολύ χειρότερη κατάσταση από άποψης εύρους ζώνης να μεταδίδει μόνο τον ήχο.

Εναλλακτικά, ένα σχήμα ποιότητας υπηρεσίας βασισμένο στις κρατήσεις (reservation-based QoS) μπορεί να υλοποιηθεί όπου ένας χρήστης μπορεί να ζητάει κάποιο συγκεκριμένο επίπεδο υπηρεσίας πριν την συμμετοχή του. Έτσι η εσωτερική δομή του δικτύου του pervasive συστήματος θα είναι σε θέση να αποφασίσει αν υπάρχουν αρκετοί διαθέσιμοι πόροι για την ικανοποίηση του αιτήματος του χρήστη και έπειτα αν υπάρχουν θα αποφασίσει στην διάθεση αυτών των πόρων[14]

Τέλος, η εσωτερική δομή των pervasive computing συστημάτων θα μπορούσε δυναμικά να ειδοποιά τον χρήστη για τις αλλαγές στην διαθεσιμότητα των πόρων καθώς και τις διάφορες εναλλακτικές επιλογές που έχει. Για παράδειγμα, όταν μια περιορισμένου εύρους συσκευή ενημερώνετε από το pervasive περιβάλλον για εναλλακτικές κοντινές

περιοχές με λιγότερους χρήστες που καταναλώνουν το εύρος ζώνης, ο χρήστης μπορεί να λαμβάνει μια προτοπή για να μετακινηθεί στην συγκεκριμένη περιοχή για να βελτιώσει την ποιότητα μετάδοσης του[16].

Ένα αντιπροσωπευτικό σχήμα για context aware περιβάλλοντα είναι το παρακάτω[33]

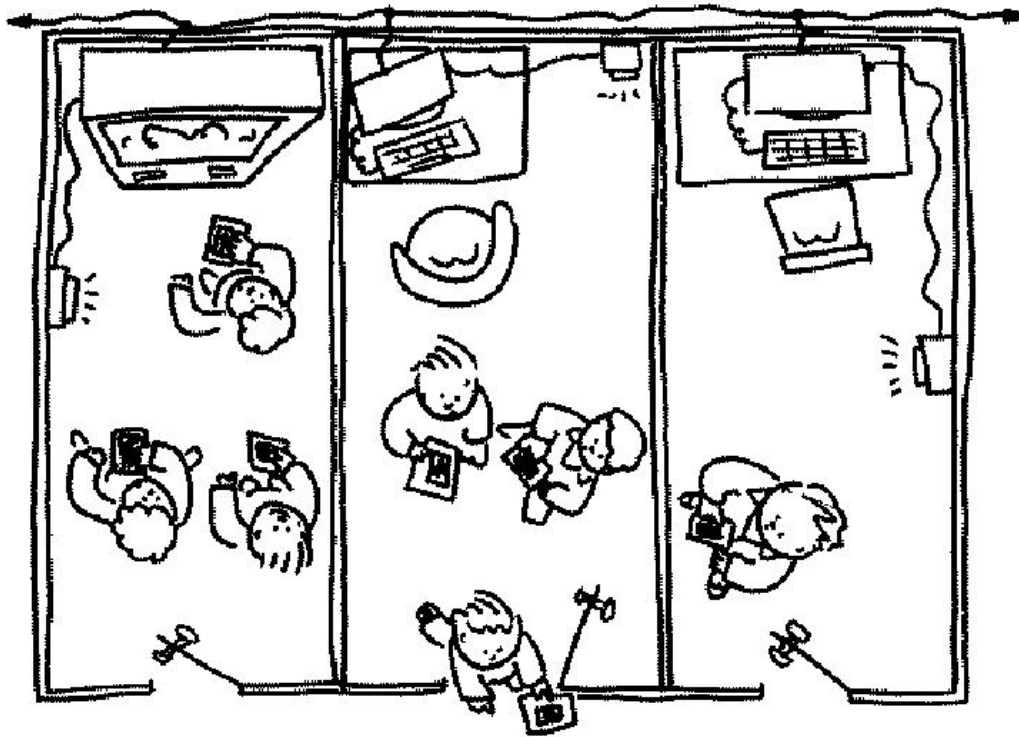


Figure 1: A Context-Aware Computing System (PARCTAB)

2.3 Context Inference

Η εξαγωγή συμπερασμάτων για τους χρήστες από τα συμφραζόμενα είναι ένα πολύ δύσκολο έργο καθώς η πιθανότητες λάθους είναι πολύ μεγάλες. Έχοντας ένα context aware σύστημα πρέπει να εκμεταλλευτούν στο έπακρο οι οποιοσδήποτε πληροφορίες μπορεί να συλλέξει το σύστημα. Η δυσκολία εξαγωγής συμπερασμάτων έγγυται στο γεγονός ότι οι προτιμήσεις των χρηστών διαφέρουν μεταξύ τους. Το περιβάλλον που τους περικλείει, τα ιδιαίτερα χαρακτηριστικά τους όπως φύλο και ηλικία είναι διάφοροι παράγοντες που έχουν σημαντικό ρόλο στην εξαγωγή των συμπερασμάτων. Ιδιαίτερα οι χρήστες που θέλουν να

λαμβάνουν προσωπικές υπηρεσίες βασισμένες στις δικές τους προτιμήσεις είτε στο ηλεκτρονικό εμπόριο είτε σε ανάκτηση πληροφοριών και άλλα. Έτσι καταλαβαίνουμε ότι η πρόβλεψη των προτιμήσεων των χρηστών και η παροχή προσωπικών υπηρεσιών που βασίζονται στις προτιμήσεις των χρηστών είναι ιδιαίτερα σημαντικό και πολύπλοκο στην υλοποίηση της.

Οι περισσότερες έρευνες σε context aware computing έχουν επικεντρωθεί στην συμπερασματική διάγνωση από υψηλού επιπέδου συμφραζόμενα όπως οι τωρινή δραστηριότητα του χρήστη από δεδομένα που προέρχονται από αισθητήρες. Όμως η πρόβλεψη των δραστηριοτήτων των χρηστών που βασίζεται μόνο στα δεδομένα από αισθητήρες είναι ιδιαίτερα περιορισμένη για την παροχή προσωπικών υπηρεσιών και άλλωστε οι προτιμήσεις των χρηστών δεν εξάγονται αυτόματα.

Έτσι αυτές οι υπηρεσίες δεν πρέπει να στηρίζονται μόνο στους πολλαπλούς αισθητήρες που θα καταγράφουν τις προθέσεις του χρήστη αλλά επιπλέον και σε λογικές δυνατότητες του συστήματος να επεξεργάζεται 'ωμές' πληροφορίες από συμφραζόμενα, δηλαδή πληροφορίες που χωρίς την κατάλληλη επεξεργασία δεν έχουν κάποιο φανερό νόημα ύπαρξης, και να τις μετατρέπει σε χρήσιμες πληροφορίες επιτόπου. Πρόσφατα κάποιες μηχανές εξαγωγής συμπερασμάτων όπως η Jena, RacerPto, FacT++ και η Minerva έχουν προταθεί ως ο πυρήνας για την υλοποίηση έξυπνων pervasive συστημάτων. Έχουν γίνει εκτενείς εκτιμήσεις για τις λογικές ικανότητες τους στην ορθότητα, πληρότητα και χρόνο απόκρισης. Αναμένετε ότι αυτές οι μηχανές εξαγωγής συμπερασμάτων θα αντεπεξέλθουν στην ωρίμανση των υπηρεσιών του pervasive computing καλύπτοντας εκτενώς διάφορες περιοχές με διάφορα σύνολα από απαιτήσεις όπως γρήγορα κινούμενους χρήστες και αλλές κινητές υπολογιστικές οντότητες.

Παρακάτω θα δούμε λίγο αναλυτικότερα μερικές μηχανές εξαγωγής συμπερασμάτων

MINERVA

Η Minerva είναι ένα υψηλής απόδοσης αποθηκευτικό, συμπερασματικό και επερωτικό (query) σύστημα υλοποιημένο σε RDBMS. Τα πλεονεκτήματα τους συστήματος μπορούν να κατηγοριοποιηθούν σε δύο σημαντικές απόψεις λογικής: χρόνο απόκρισης και κλιμάκωση. Υπολογίζει όλα τα συμπεράσματα στην σχετική βάση δεδομένων και έτσι γίνεται πιο ευέλικτο[16].

DLDB-OWL/HAWK

Είναι ένα πλαίσιο θεματοφυλακίου (repository framework) και μια γκάμα εργαλείων που υποστηρίζουν OWL. Παρέχει APIs (application programming interface) καθώς και υλοποιήσεις για φρασεολογία εντολών (parsing), αλλαγή και μεταχείριση OWL οντοτήτων. Η αρχιτεκτονική του DLDB-OWL αποτελείται από τρία πακέτα: πυρήνα, owl και χώρο αποθήκευσης. Το πακέτο πυρήνα ορίζει την γενετική διαπροσωπεία των δομών δεδομένων των οντολογιών και τον οντολογικών μοντέλων[16].

PELLET

Η Pellet είναι μια ανοικτού κώδικα βασισμένη σε Java OWL DL μηχανή εξαγωγής συμπερασμάτων. Μπορεί να χρησιμοποιηθεί στην μετάδοση τόσο με το Jena όσο και με OWL API βιβλιοθήκες και επιπλέον προσφέρει και διαπροσωπεία DIG[16].

RacerPro

Το RacerPro είναι μια OWL μηχανή εξαγωγής συμπερασμάτων και εξυπηρετητής συμπερασμάτων για τον σημασιολογικό ιστό. Το RacerPro προέρχεται από την περιοχή της περιγραφικής λογικής. Μπορεί να χρησιμοποιηθεί και σαν σύστημα για διαχείριση οντολογιών του σημασιολογικού ιστού βασισμένες στο OWL[16].

JENA

Το Jena είναι ένα θεματοφυλάκιο (framework) βασισμένο στην Java για την δημιουργία εφαρμογών για τον σημασιολογικό ιστό. Παρέχει προγραμματιστικά περιβάλλοντα για RDF, RDFS και OWL και περιλαμβάνει μια μηχανή εξαγωγής συμπερασμάτων βασισμένη σε κανόνες[16].

Παρακάτω βλέπουμε ένα συγκριτικό πίνακα με τις διάφορες μηχανές εξαγωγής συμπερασμάτων[16].

Table 1. The Comparative Table for Inference Engine

	Import	Inference			Repository		Query	Version	Source
	Ontology Language	Capability	Internal Engine	External Engine	Persistent Storage	Memo-ry	Language		
Minerva	OWL	DL	O	Racer, Pellet	DB2, Derby, HSQLDB	X	SPARQL	Semantics Toolkit-1.1.1	IBM
DLDB-OWL	OWL	DL	X	Racer	MS-Access, PostgreSQL	O	KIF-Like language	HAWK 1.3 beta	Lehigh University SWAT Lab
Pellet	DIG tell document, OWL	DL	O	X	X	O	DIG ask document, RDQL	pellet-1.3	mindswap
Racer	DIG tell document, OWL	DL	O	O	X	O	DIG ask document, Racer Query Language	RacerPro 1.9.0	Racer Systems GmbH & Co. KG
JENA	OWL	DL	O	Racer Pro, Pellet, Fact++ (JenaOntModel)	MySQL, HSQLDB, PostgreSQL, Oracle, MS-SQL Server	O	SPARQL	Jena 2.4	HP Labs

2.3 Context History

2.3.1 Γενικά

Με τον όρο context history εννοούμε απλά την συλλογή παλαιότερων συμφραζομένων πληροφοριών (context) καθώς και παλαιότερες ενέργειες του χρήστη. Με αυτήν την συλλογή πληροφοριών και με την επεξεργασία τους υπάρχουν αυξημένες πιθανότητες στην βελτίωση των υπηρεσιών που προσφέρουν διάφορες εφαρμογές. Αν η συλλογή παλαιότερων συμφραζομένων πληροφοριών μπορεί να χρησιμοποιηθεί, η παροχή διάφορων ‘έξυπνων’ υπηρεσιών προσαρμοσμένες στις διάφορες απαιτήσεις του κάθε χρήστη αφού προηγουμένως έχουν εξαχθεί διάφορα χρήσιμα μοτίβα απο την συλλογή παλαιότερων συμφραζομένων πληροφοριών. Για παράδειγμα, αν υπάρχει η πληροφορία ότι ο χρήστης βλέπει πάντα στην τηλεόραση τις ειδήσεις των 21:00 σε ένα ‘έξυπνο σπίτι’, τότε το σύστημα ενεργοποιεί το περιβάλλον νοημοσύνης που επιτρέπει στον χρήστη να βλέπει στην τηλεόραση τις ειδήσεις σύμφωνα με το μοτίβο του χρήστη. Παρόλο που χρήσιμες πληροφορίες μπορούν να εξαχθούν βασισμένες σε δημογραφικά δεδομένα, η σχετική έρευνα στο θέμα του συνδιασμού του προφίλ του χρήστη και των συμφραζομένων πληροφοριών είναι ελάχιστη ανάμεσα στην όλη έρευνα για συλλογή και χρήση των παλαιότερων συμφραζομένων πληροφοριών.

Γενικά, συμφέρουσες, αυτόματες, έξυπνες και προσωπικές υπηρεσίες μπορούν να προσφερθούν στον χρήστη εξάγοντας χρήσιμες πληροφορίες όπως διάφορες προτιμήσεις του χρήστη, μοτίβα και συνήθειες από την συλλογή παλαιότερων συμφραζομένων πληροφοριών σε context aware υπολογιστικά συστήματα.

2.3.2 Σχετική έρευνα με context history

Τα πρώτα context aware συστήματα όπως το Active Badge [17] Xerox και PARC [18] ήταν βασικά συστήματα που βασίζονταν στην γνώση της τοποθεσίας. Συστήματα τύπου ξεναγού (tour guide) όπως το GUIDE [6] και CyberGuide [19] εξέληξαν την ιδέα του context προσθέτοντας προσωρινές πληροφορίες. Αυτά τα συστήματα είναι κυρίως context aware εφαρμογές που σχεδιάστηκαν για την παροχή καλύτερων και πιο εξειδικευμένων υπηρεσιών στους χρήστες τους. Το Context-Toolkit[20] παρέχει μια προγραμματιστική διαπροσωπεία εφαρμογής (API) για την δημιουργία context aware εφαρμογών αλλά είναι περιορισμένο λόγω χρήσης αισθητήρων. Τα τωρινά context aware συστήματα είναι πλαίσια (frameworks) που παρέχουν context awareness μέσω πλούσιας οντολογίας βασισμένης σε αναπαράσταση συμφραζομένων. Αυτή η πλούσια οντολογία συμφραζομένων λαμβάνει υπόψην της όλες τις παραμέτρους που είναι σχετικές με την αλληλεπίδραση. Το Gaia [21] που είναι ένα καταναμημένο λειτουργικό σύστημα βασισμένο σε CORBA και το CAMUS [22] που είναι μια JINI υπηρεσία βασισμένη σε ένα πλαίσιο παρέχουν context aware υπηρεσίες σε περιορισμένες context aware εφαρμογές. Το COBRA [23] είναι ένα πλαίσιο βασισμένο σε κινητούς πράκτορες (mobile agent framework) που έχει κινητούς πράκτορες για να συλλέξουν συμφραζώμενη πληροφορία από τους αισθητήρες του περιβάλλοντος. Το CAPEUS [24] χρησιμοποιά μια προσέγγιση τύπου εγγράφου που ανταλλάσει πακέτα context aware που περιγράφουν τις αιτήσεις για υπηρεσίες. Η ιστορία των πληροφοριών διατηρείται ως μια συμφραζώμενη βάση δεδομένων (contextual database) στο SOCAM[25], στο Gaia, στο CASS[26] και στο CoBra.

Παρακάτω παρατήθετε ένας πίνακας συγκρίσεως των ανωτέρω συστημάτων

TABLE I. COMPARISON OF CONTEXT AWARE SYSTEMS

	Rich Context	Decentralized	Ontology	Inference Support	History of Information
Active Badge and Xerox PARC (1992)	x	x	x	x	x
Cyberguide and GUIDE (2000)	x	x	x	x	x
Context Toolkit(1999)	✓	x	x	✓	x
CAPEUS	✓	✓	x	✓	x
GAIA OS (2002)	✓	✓	✓	✓	✓
CMF (2003)	✓	✓	x	✓	x
CoBrA (2003)	✓	✓	✓	✓	✓
Hydrogen (2002)	✓	x	x	✓	x
SOCAM (2004)	✓	x	✓	✓	✓

2.4 State of Art

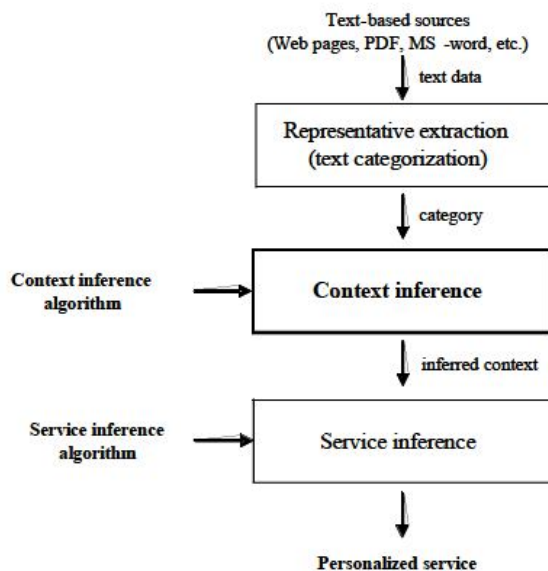
Ο εμπνευστής της ιδέας του pervasive computing ήταν ο Mark Weiser ο οποίος παρουσίασε ένα σύστημα που αποτελείτο από τριών ειδών υπολογιστικών συσκευών, ταμπλετίτσα (tab), μπλοκ (pad) και πίνακα (board). Ο σχεδιασμός της καθεμιάς εξ αυτών των συσκευών εμπνεύστηκε από την αντίστοιχη χρήση των προαναφερόντων εργαλείων ενός τυπικού γραφείου [27]. Οι ταμπλετίτσες ήταν μικρές συσκευές σχεδιασμένες για μικρές σημειώσεις που είχαν αίσθηση του χώρου και μια οθόνη με ευαισθησία στην πίεση. Τα μπλοκ ήταν λίγο μεγαλύτερα σε μέγεθος και ήταν σαν ασύρματα βιβλία για να κρατάς σημειώσεις με στυλό. Τέλος οι πίνακες ήταν σημαντικά μεγαλύτερα, με επιφάνειες αλληλεπίδρασης κοντά στο μέγεθος του τοίχου. Αυτή η έρευνα έδωσε το έναυσμα για περαιτέρω μελέτη χάρις στην σημαντική επίδραση στον τρόπο με τον οποίο ο άνθρωπος αλληλεπιδρά με την υπολογιστική τεχνολογία. Έτσι ο Weiser έθιξε τις βασικές αρχές ενός pervasive συστήματος, που περιλαμβάνει ασύρματη επικοινωνία, ενσωματωμένες και κινητές συσκευές, καταναμημένους υπολογισμούς και τέλος context awareness.

Παρακάτω αναφέρονται σε γενικές γραμμές ορισμένες έρευνες που έχουν γίνει σχετικά με το πεδίο του pervasive computing και συγκεκριμένα της εξαγωγής συμπερασμάτων από τα συμφραζώμενα.

- 1) Οι Evan Welbourne, Jonathan Lester, Anthony LaMarca και Gaetano Borriello ασχολήθηκαν με την κινητή εξαγωγή συμπερασμάτων χρησιμοποιώντας χαμηλού κόστους αισθητήρες[28]. Συγκεκριμένα παρουσίασαν ένα μικρό σύστημα όπου θα ενοποιούσε δεδομένα για την τοποθεσία με δεδομένα από απλους, χαμηλού κόστους αισθητήρες που δεν θα είχαν την αίσθηση της τοποθεσίας, με σκοπό να συμπεράνουν την τοποθεσία του χρήστη καθώς και την κατάσταση του. Το σύστημα βρίσκει την τοποθεσία με ένα κινητό τηλέφωνο που έχει δέκτη GPS και μια συσκευή που διαθέτει ασύρματη πρόσβαση WiFi, τρέχει το PlaceLab και συλλέγει επιπλέον δεδομένα χρησιμοποιώντας ένα κύκλωμα αισθητήρων μεγέθους 2''x1'' με βασικούς αισθητήρες όπως βαρομετρικής πίεσης, για μέτρηση της επιτάχυνσης και άλλους, οι οποίοι αισθητήρες είναι ενσωματωμένοι στην ασύρματη συσκευή. Με αυτή την έρευνα αποδεικνύουν ότι χρησιμοποιώντας απλά βασική επεξεργασία των δεδομένων των αισθητήρων και αλγόριθμους fusing είναι δυνατόν με την βοήθεια αισθητήρων που έχουν αίσθηση της τοποθεσίας και με αισθητήρες που δεν έχουν, να ανοικτούν νέες δυνατότητες στην κινητή εξαγωγή συμπερασμάτων.
- 2) Οι David West, Trent Afted και Aaron Quigley ασχολήθηκαν με την έρευνα για πολυκατάστατη προσέγγιση (multi-modal) για πρόσβαση σε κινητές πληροφορίες και η εξαγωγή συμπερασμάτων [29]. Το context aware computing με πολλές καταστάσεις (multimodal) προσφέρει υποστήριξη του συστήματος που ενσωματώνει την περιβαλλοντική, γνωστική και υπολογιστική κατάσταση κάποιου ατόμου καθώς επιτρέπει την σύνθεση εισαγωγής στοιχείων και λήψης αποτελεσμάτων σε μια ποικιλία διαθέσιμων καταστάσεων επικοινωνίας ανθρώπου – υπολογιστή. Στην έρευνα τους παρουσιάζουν δύο πλάνα υποστήριξης από την αρχιτεκτονική ιδέα του pervasive computing, το Nightingale. Τα πλάνα αποτελούνται από μια πολυκατάστατη εφαρμογή τύπου πλαισίου (multimodal application framework) και ένα διαχειριστή συμφραζωμένων και μια προσέγγιση εξαγωγής συμπερασμάτων βασισμένη σε οντολογίες. Στο Nightingale οι συσκευές που χρησιμοποιούνται σαν ένα είδος προσωπικού χώρου δικτύου (PAN =personal area network) με πρόσβαση σε

τοπικές (PLAN) ή απομακρυσμένες (PWAN) συσκευές ή υπηρεσίες όταν αυτές γίνουν διαθέσιμες.

- 3) Οι Sangjun Kim, Euiho Suh και Keedong Yoo ασχολήθηκαν με την έρευνα σε ένα αλγόριθμο εξαγωγής συμπερασμάτων πάνω σε πληροφοριακό σύστημα που βασίζεται στον παγκόσμιο ιστό (Web-based Information System) [30]. Πιο συγκεκριμένα, ένας αλγόριθμος εξαγωγής συμπερασμάτων από τα συμφραζόμενα έχει σκοπό την αναγνώριση της πρόθεσης του χρήστη στο πληροφοριακό σύστημα. Για να εφαρμοστεί αυτός ο αλγόριθμος, χρησιμοποιούν μια τεχνική κατηγοροποίησης κειμένου για να διαχωρίσει διάφορες πηγές κειμένων που χρησιμοποιούν οι χρήστες όπως ιστοσελίδες, pdf αρχεία, MS-Word αρχεία και άλλα. Έπειτα, η υπηρεσία του αλγορίθμου βασίζεται στις ομοιότητες των μετρήσεων μεταξύ των προτιμήσεων του χρήστη και των υπηρεσιών που είναι ενεργοποιημένες και προσπαθεί να προτείνει μια υπηρεσία που χρειάζεται ο χρήστης σε αυτόν. Ο αλγόριθμος που προτείνεται στην έρευνα αυτή, μπορεί να βοηθήσει τον χρήστη του πληροφοριακού συστήματος σε μια πιο βολική χρήση του πληροφοριακού συστήματος καθώς και να κάνει ένα ήδη υπάρχον πληροφοριακό σύστημα να παρέχει υπηρεσίες τύπου pervasive computing. Παρακάτω έχουμε το διάγραμμα του πλαισίου (framework) εξαγωγής συμπερασμάτων από υπηρεσίες και συμφραζόμενα όπως χρησιμοποιήθηκε στην έρευνα:



<Figure 1> Context and service inference framework

4) Οι Choonsung Shin and Woontack Woo ασχολήθηκαν με την έρευνα για επίλυση επιπλοκών χρησιμοποιώντας την ύπαρξη παλαιότερων πληροφοριών σε context aware εφαρμογές [31]. Σύμφωνα με την έρευνα τους, προτείνουν το Conflict Manager για αναλύσει τις επιπλοκές σε εφαρμογές context aware μέσα σε ένα περιβάλλον έξυπνου σπιτιού. Επιπλοκές παρουσιάζονται όταν πολλοί χρήστες έχουν πρόσβαση σε μια εφαρμογή ή όταν διάφορες εφαρμογές έχουν περιορισμένους πόρους για να παρέχουν τις υπηρεσίες τους. Για να επιλυθούν αυτές οι επιπλοκές ανάμεσα στους χρήστες, το Conflict Manager προσδίδει διάφορες προτεραιότητες ανάμεσα στους χρήστες έτσι ώστε αυτός με την υψηλότερη προτεραιότητα να επιλεγεί έχοντας υπόψιν την προϊστορία επιπλοκών που προκλήθηκαν από τους χρήστες. Επιπλέον το Conflict Manager αναγνωρίζει και επιλύει επιπλοκές ανάμεσα σε εφαρμογές χρησιμοποιώντας τις προτιμήσεις των χρηστών και τις ιδιότητες των υπηρεσιών. Το διάγραμμα του διαχειριστή παλαιότερων επιπλοκών παρουσιάζεται παρακάτω

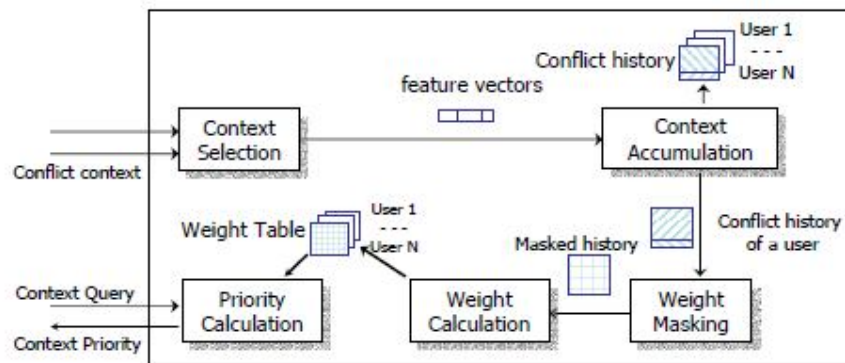
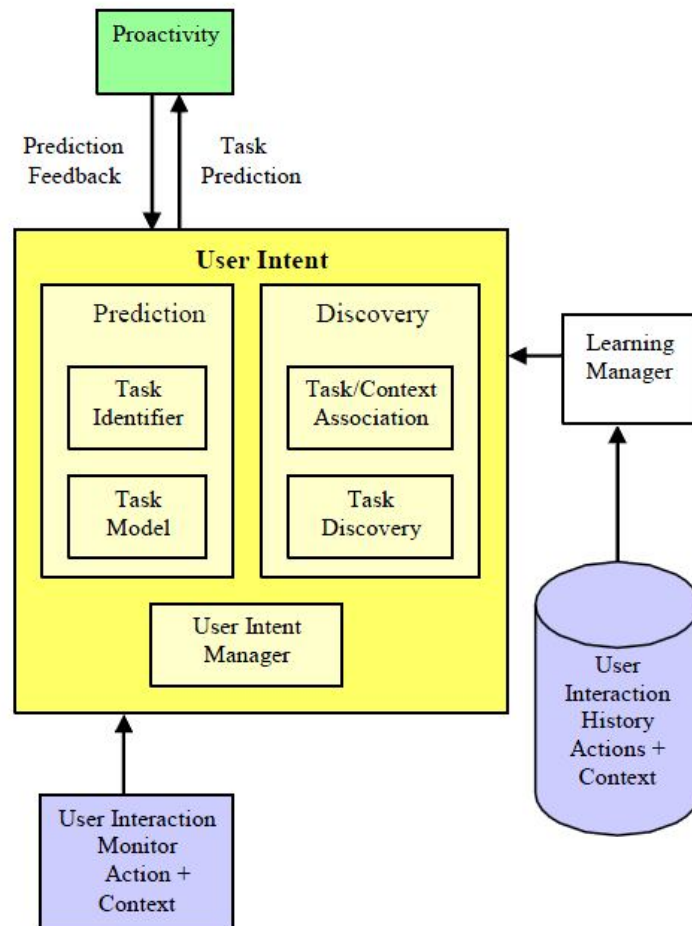


Figure 4. Conflict History Manager

5) Οι Yussuf Abu Shaaban, Sarah McBurney, Nick Taylor, M. Howard Williams, Nikos Kalatzis και Ioanna Roussaki ασχολήθηκαν με την έρευνα για την πρόθεση του χρήστη για υποστήριξη της προνοητικότητας (proactivity) των pervasive συστημάτων [32]. Σε αυτή την έρευνα ασχολήθηκαν με το πρόγραμμα Persist που ξεκίνησε τον Απρίλιο του 2008. Σκοπός του είναι η δημιουργία όχι μιας τόσο διαφορετικής μορφής pervasive συστημάτων αλλά κάνοντας το να επεκτείνει και να προσαρμόσει μερικά χαρακτηριστικά από το σύστημα Daidalos. Συγκεκριμένα, θα ολοκληρώσει την προσωποποίηση και την διαχείριση των προτιμήσεων με ένα σύστημα που καταγράφει τις προθέσεις του χρήστη. Ο σκοπός τους συστήματος καταγραφής των προθέσεων του

χρήστη είναι η ανακάλυψη και η διαχείριση ενός μοντέλου στην συμπεριφορά του χρήστη όσο αφορά τις ενέργειές του και τις δραστηριότητες του. Με τον όρο ενέργεια περιλαμβάνονται οι οποιεσδήποτε αλληλεπιδράσεις του χρήστη με κάποια υπηρεσία και με τον όρο δραστηριότητες οποιαδήποτε αλληλουχία ενεργειών. Έτσι έχοντας και τις προβλέψεις και τις προτιμήσεις του χρήστη, οι μηχανισμοί προνοητικότητας (proactivity) έχουν παροχή δεδομένων μέσα στο πρόγραμμα Persist. Με την προσθήκη των προβλέψεων της πρόθεσης του χρήστη, οι μηχανισμοί προνοητικότητας μπορούν να εκτελούν εργασίες εκ των προτέρων προσφέροντας έτσι ένα περιβάλλον που ελαχιστοποιεί την ενασχόληση του χρήστη με το σύστημα και το κάνει πιο εύχρηστο. Η αρχιτεκτονική του συστήματος εξαγωγής συμπερασμάτων από τις προτιμήσεις του χρήστη που προτείνεται σε αυτή την έρευνα φαίνεται παρακάτω



ΚΕΦΑΛΑΙΟ 3

Υλοποίηση Εφαρμογής

Στόχος της συγκεκριμένης εργασίας ήταν η υλοποίηση ενός προγράμματος που θα τρέχει σε περιβάλλον Windows XP καθώς και σε Windows Vista το οποίο θα πραγματοποιεί μια καταγραφή σε δύο αρχεία ,τα οποία θα εξηγήσω και θα αναλύσω αργότερα, των διαφορών προγραμμάτων που χρησιμοποιά ο χρήστης του υπολογιστή. Συγκεκριμένα καταγράφει μόνο τα προγράμματα που έχουν άμεση διεπαφή με τον χρήστη μέσω γραφικού περιβάλλοντος κρατώντας και μερικές πληροφορίες για αυτά όπως ημερομηνία και ώρα έναρξης και τερματισμού. Τρέχοντας το συγκεκριμένο πρόγραμμα για αρκετές μέρες σε ένα υπολογιστή και μαζεύοντας τις πληροφορίες μπορούμε μετά να τις επεξεργαστούμε και να εξάγουμε διάφορα συμπεράσματα για τον χρήστη καθώς και τις προτιμήσεις του σε διάφορα προγράμματα. Έτσι με μια στατιστική ανάλυση των πληροφοριών μπορούμε να βελτιώσουμε την επαφή του χρήστη με τον υπολογιστή του, φορτώνοντας τα αγαπημένα του προγράμματα με την εκκίνηση του υπολογιστή. Αυτό είναι ένα απλό παράδειγμα της χρήσης του προγράμματος. Παρακάτω παραθέτω ψευδοκώδικα καθώς και επεξήγηση του προγράμματος που υλοποίησα.

3.1 Ψευδοκώδικας

Παρακάτω παρατίθεται ο ψευδοκώδικας της υλοποίησης

/* Διάφορες αρχικοποιήσεις προγράμματος */

```
string convertInt(int number)
{
    Δημιουργία stringstream
    Πρόσθεση αριθμού στο stream
    Επιστροφή του string με τα περιεχόμενα του stream
}
```

```

void CreateLog()
{
    SYSTEMTIME ft;

    if (Η συναρτηση EnumWindows() έχει ξανατρέξει)
    {
        for (ολες τις προηγουμενες διεργασίες)
        {
            for (ολες τις διεργασίες τρέχουν τώρα)

                if (μια προηγούμενη διεργασία ακόμα να τελιώσει)
                    Καθορισμός σημαίας ενεργής διεργασίας

            if (η διεργασία τελειωσε)
            {
                Πέρνω την τοπική ώρα συστήματος

                Μετατροπή της της ώρας συστήματος σε string με την
                συνάρτηση convertInt()
                Φύλαξη της ώρας σε μια μεταβλητή και στην αντιστοιχη
θέση σε σχετικό πικακα
                Ανοίγουμε το αρχείο FinishedProg.txt

                Φυλάγουμε στο τέλος του αρχείου κάθε φορά τα παρακάτω
στοιχεία για τις τερματισμένες διεργασίες process id, το όνομα της διεργασίας, την ώρα
έναρξης καθώς και την ώρα τερματισμού της

                Κλείνουμε το αρχείο
            }

            Απενεργοποίηση σημαίας ενεργής διεργασίας για επόμενο έλεγχο
        }
    }
}

```

```
}
```

Άνοιγμα αρχείου RunningProg.txt

Γράφουμε εκ νέου τα εξής στοιχεία για τις ενεργές διεργασίες process id, το όνομα της διεργασίας και την ώρα έναρξης

Κλείνουμε το αρχείο

```
for (ολες τις διεργασίες τρέχουν τώρα)
```

```
{
```

```
    Φυλάγουμε σε συγκεκριμένους πίνακες τα process id,  
    ονόματα διεργασιων καθώς και ώρα έναρξης
```

```
}
```

```
}
```

```
BOOL CALLBACK EnumWindowsProc(HWND hWnd, LPARAM lParam) {
```

```
    SYSTEMTIME st;
```

```
    if (!hWnd)
```

```
        Not a window
```

```
    if (!::IsWindowVisible(hWnd))
```

```
        Not visible
```

```
    if (!SendMessage(hWnd, WM_GETTEXT, sizeof(String), (LPARAM)String))
```

```
        No window title
```

```
    hInstance = (HINSTANCE)GetWindowLong(hWnd, GWL_HINSTANCE);
```

```
    dwThreadId = GetWindowThreadProcessId(hWnd, &dwProcessId);
```

```
    hProcess = OpenProcess(PROCESS_ALL_ACCESS, FALSE, dwProcessId);
```

```
for(ολες τις προηγουμενες διεργασίες)
{
    if (η διεργασία υπάρχει ήδη)
    {
        Τελική Ώρα έναρξης διεργασιας

        Τελικό Όνομα Διεργασίας

    }
}
```

Φύλαξη αναγνωριστικού διεργασίας σε πίνακα

```
if (έχουμε νέα διεργασία)
{
    Πέρνουμε την ώρα του συστήματος

    Μετατροπή της της ώρας συστήματος σε string με την συνάρτηση
convertInt()

    Αποθήκευση του string στον πίνακα με τις ώρες έναρξης των διεργασιών στο
σωστό index

    Αποθήκευση του ονόματος της διεργασίας στον αντίστοιχο πίνακα με τα
αονόματα
}

Αύξηση μετρητή κατα ένα

Κλείσιμο handle
}
```

```

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR
lpCmdLine, int nShowCmd)
{
    Πέρνω την τοπική ώρα συστήματος

    Μετατροπή της της ώρας συστήματος σε string με την συνάρτηση convertInt()

    Φύλαξη της ώρας σε μια μεταβλητή και στην αντιστοιχη θέση σε σχετικό πικακα

    Ανοίγουμε το αρχείο FinishedProg.txt

    Γράφουμε την ώρα που ξεκίνησε το πρόγραμμα μας

    Κλείνουμε το αρχείο

    while (1)
        {
            EnumWindows(EnumWindowsProc, NULL);

            CreateLog();

            Φυλαξη αριθμού τρέχοντων διεργασιών

            Μηδενισμός μετρητή

            Ενεργοποώ την σημαία για την διεκπέρωση της συνάρτησης
EnumWindows()

            Κοιμίζω το σύστημα για ένα δευτερόλεπτο
        }

    return 0;
}

```

3.2 Επεξήγηση Υλοποίησης Εφαρμογής

Η γενική υλοποίηση του προγράμματος έγινε στην γλώσσα προγραμματισμού C/C++ στο περιβάλλον εργασίας Microsoft Visual Studio 2008 Express Edition το οποίο διατίθεται δωρεάν στο διαδίκτιο και συγκεκριμένα στην ιστοσελίδα της Microsoft. Το πρόγραμμα είναι συμβατό και ελεγμένο με τα Microsoft Windows XP και τα Microsoft Windows Vista.

3.2.1 Κυρίως Πρόγραμμα

Παρατηρώντας τον ψευδοκώδικα βλέπει κανείς ότι το κυρίως πρόγραμμα είναι αρκετά απλό και αποτελείτε βασικά από ένα ατέρμονα βρόχο ο οποίος υλοποιείται με ένα while loop. Μέσα σε αυτό τον βρόχο καλούνται διαδοχικά δύο συναρτήσεις, η EnumWindows()

```
EnumWindows(EnumWindowsProc, NULL);
```

και η CreateLog() όπου θα εξηγήσουμε την λειτουργία τους σε επόμενες παραγράφους.

Έπειτα φυλάγουμε σε μια μεταβλητή τον αριθμό των διεργασιών που επέστρεψε η EnumWindows() και ενεργοποιώ μια σημαία

```
flag3=1;
```

που υποδηλώνει την διεκπέρωση της προηγούμενης συνάρτησης για χρήση της σε βοηθητικές συναρτήσεις που υλοποιήθηκαν.

Τέλος χρησιμοποιούμε την έτοιμη συνάρτηση sleep() που περιλαμβάνεται στην επικεφαλίδα dos.h

```
Sleep(1000);
```

για να κοιμήσουμε το σύστημα για ένα δευτερόλεπτο ούτως ώστε να πέρνουμε ένα δείγμα από τα applications που τρέχει ο χρήστης με ένα αρκετά καλό ρυθμό και ταυτόχρονα να μην επιβαρύνουμε τον επεξεργαστή του συστήματος. Αυτό είναι σημαντικό για το πρόγραμμα αφού η απόδοση του τόσο του προγράμματος τόσο και του συστήματος του χρήστη πρέπει να αποτελεί πρωταρχικό παράγοντα για την υλοποίηση, μιας και τα τεχνικά χαρακτηριστικά του συστήματος χρήστη δεν πρέπει να είναι περιοριστικά για την εφαρμογή.

3.2.2 Η συνάρτηση EnumWindowsProc()

Η συνάρτηση EnumWindows() απαριθμεί όλα τα πρώτου βαθμού (top-level) παράθυρα (windows) που παρουσιάζονται στην οθόνη του υπολογιστή του χρήστη. Η απαρίθμηση επιτυγχάνεται, είτε είναι πρωταρχικό παράθυρο (parent window) είτε είναι παιδί παράθυρο (child window), με το να περνάει τον διαχειριστή (handle) του κάθε παραθύρου με μια σειρά σε μια άλλη application-defined callback συνάρτηση. Η συνάρτηση EnumWindows () συνεχίζει να εκτελείται μέχρι να απαριθμηθούν όλα τα τρέχοντα παράθυρα.

Για όρισμα η συνάρτηση μας πέρνει την callback συνάρτηση EnumWindowsProc() όπου την τροποποιούμε για να πετύχουμε τον σκοπό μας. Συγκεκριμένα ελέγχουμε πρώτα αν ο διαχειριστής που μας επέστρεψε η EnumWindowsProc() αντιστοιχεί όντως σε παράθυρο

```
if (!hWnd)
    return TRUE;
```

και αν αυτό το παράθυρο είναι ορατό στον χρήστη,

```
if (!::IsWindowVisible(hWnd))
    return TRUE;
```

αφού η εφαρμογή έχει σκοπό να καταγράφει τα προγράμματα με τα οποία ο χρήστης αλληλεπιδρά με την βοήθεια κάποιου γραφικού περιβάλλοντος. Τέλος ελέγχουμε αν ο διαχειριστής συνδέεται με ένα παράθυρο που έχει κάποιο τίτλο και τον φυλάμε.

```
if (!SendMessage(hWnd, WM_GETTEXT, sizeof(String), (LPARAM)String))
    return TRUE;
```

Αν οι τρεις παραπάνω προϋποθέσεις ισχύουν τότε ο διαχειριστής που μας επέστρεψε η EnumWindowsProc() αντιστοιχεί σε ένα παράθυρο αλληλεπίδρασης του χρήστη και μπορούμε να προχωρήσουμε την επεξεργασία αλλιώς η συνάρτηση συνεχίζει με τον επόμενο διαχειριστή (handle). Αφού επιτύχουν οι έλεγχοι, χρησιμοποιούνται διάφορες έτοιμες συναρτήσεις που περιέχονται στην βιβλιοθήκη User32.lib για την απόκτηση διάφορων πληροφοριών για τα παράθυρα. Πρώτα καλούμε την GetWindowLong()

```
hInstance = (HINSTANCE)GetWindowLong(hWnd, GWL_HINSTANCE);
```

για την εύρεση μιας 32-bit τιμής που έχει σχέση με την μνήμη και χρησιμοποιείται πιο μετά. Έπειτα καλούμε την GetWindowThreadProcessId() που μας επιστρέφει την χαρακτηριστική τιμή του νήματος που δημιούργησε το παράθυρο, καθώς επίσης και την χαρακτηριστική τιμή της διεργασίας (process id).

```
dwThreadId = GetWindowThreadProcessId(hWnd, &dwProcessId);
```


Έχοντας την χαρακτηριστική τιμή της διεργασίας καλείτε η συνάρτηση `OpenProcess()` για να ανοίξει το αντικείμενο της διεργασίας και να μας επιστρέψει τον διαχειριστή (handle) του παραθύρου.

```
hProcess = OpenProcess(PROCESS_ALL_ACCESS, FALSE, dwProcessId);
```

Αφού γίνουν οι παραπάνω ενέργειες με επιτυχία, τότε γίνεται έλεγχος αν η διεργασία υπήρχε ήδη στην λίστα των διεργασιών και ακόμα να τελειώσει. Σε αυτή την περίπτωση ενεργοποιούμε μια σημαία για αποτυχία εισαγωγής νέας διεργασίας στην λίστα των τρέχοντων διεργασιών

```
flag1=1
```

και φυλάσσεται το όνομα του παραθύρου που δημιούργησε και διατηρείτε ο προηγούμενος χρόνος έναρξης της διεργασίας. Στην αντίθετη περίπτωση όπου έχουμε ανίχνευση νέας διεργασίας, πέρνουμε την τοπική ώρα συστήματος με την συνάρτηση `GetLocalTime()` και την φυλάμε σαν ώρα έναρξης της διεργασίας και έπειτα φυλάμε και τον τίτλο του νέου παραθύρου. Τελειώνοντας ενεργοποιείτε ξανά η σημαία για αποτυχία εισαγωγής νέας διεργασίας με σκοπό διάφορους ελέγχους που γίνονται στο πρόγραμμα και καλώντας την συνάρτηση `CloseHandle()`

```
CloseHandle(hProcess);
```

ελευθερώνω τον διαχειριστή του παραθύρου για να συνεχίσει η `EnumWindowsProc()` τις κλήσεις.

3.2.3 Η συνάρτηση `CreateLog()`

Τελειώνοντας η `EnumWindows()` και έχοντας απαριθμίσει όλα τα τρέχοντα παράθυρα με τα οποία αλληλεπιδρά ο χρήστης και έχοντας φυλάξει στις δομές `vectors` που χρησιμοποιούνται διάφορα στοιχεία όπως τίτλο παραθύρου, ώρα έναρξης και χαρακτηριστική τιμή της διεργασίας που το δημιούργησε, το κυρίως πρόγραμμα καλεί την συνάρτηση `CreateLog()`.

Σκοπός αυτής της συνάρτησης είναι η δημιουργία δύο αρχείων. Το ένα αρχείο που ονομάζεται `RunningProg.txt`

```
ofstream myfile2;
```

```
myfile2.open ("RunningProg.txt",ios::out);
```

έχει σκοπό να φυλάει τα στοιχεία των διεργασιών που είναι ακόμα ενεργές και ο χρήστης εξακολουθεί να βλέπει στην οθόνη του. Το δεύτερο αρχείο που δημιουργείτε ονομάζεται FinishedProg.txt

```
ofstream myfile;  
myfile.open ("FinishedProg.txt",ios::app);
```

και σκοπός του είναι η φύλαξη των στοιχείων των διεργασιών που έχουν τελειώσει με ένα επιπρόσθετο στοιχείο, την ώρα τερματισμού τους.

Καλώντας την συνάρτηση CreateLog() κάνουμε ένα έλεγχο στην σημαία για την διεκπέρωση της συνάρτησης EnumWindows().

```
if (flag3 != 0)  
{  
....  
}
```

Αν είναι η πρώτη φορά που έχει τρέξει η EnumWindows() τότε προφανώς δεν έχουμε κάπου προηγούμενες λίστες με διεργασίες που ήδη μπορούν να τρέξουν και υποθέτουμε χρόνο έναρξης την τωρινή τοπική ώρα του συστήματος όπως ορίστηκε στο σώμα ορισμού της EnumWindows(). Έτσι δημιουργείτε ένα αρχείο RunningProg.txt και φυλάσσονται μέσα τα διάφορα στοιχεία των διεργασιών που λήφθηκαν μέσω της EnumWindows().

```
for (int j=0; j<Counter; j++)  
    myfile2 << tempPID[j] << '\t' << appName[j] << '\t' << startTime[j] << " ----> "  
<< " Running" << endl;
```

Στην αντίθετη περίπτωση που η EnumWindows() έχει ήδη απαριθμήσει ξανά όλα τα παράθυρα αλλάζει η ροή του προγράμματος. Πρώτα γίνεται έλεγχος καθεμίας παλιάς διεργασίας με τις νέες διεργασίες για να ευρεθεί αν οποιαδήποτε παλιά διεργασία έχει τελειώσει.

```
for (int q=0; q<Counter;q++)  
    if (tempPIDprv[k] == tempPID[q])  
        flag4=1;
```

Αν διαπιστωθεί ότι μια παλιά διεργασία δεν περιλαμβάνεται στην νέα λίστα διεργασιών, τότε ανοίγεται το αρχείο FinishedProg.txt και φυλάσσονται τα στοιχεία της διεργασίας για μετέπειτα επεξεργασία.

```
myfile << tempPIDprv[k] << '\t' << prvName[k] << '\t' << prvstartTime[k] << " -----> "  
<< endTime[k] << endl;
```

Η ώρα τερματισμού βρίσκεται με την συνάρτηση `GetLocalTime()` όταν διαπιστωθεί το προηγούμενο.

```
GetLocalTime(&ft);  
string s = convertInt(ft.wHour);  
string s1 = convertInt(ft.wMinute);  
string s2 = convertInt(ft.wDay);  
string s3 = convertInt(ft.wMonth);  
string s4 = convertInt(ft.wYear);  
string temp = s2+"/"+s3+"/"+s4+" @ "+s + ":" + s1;  
endTime[k] = temp;
```

Σε περίπτωση που δεν έχει καμία διεργασία τερματιστεί, η ροή του προγράμματος συνεχίζεται όπως προηγουμένως. Τέλος τα διάφορα στοιχεία που βρίσκονται φυλαγμένα στις δομές `vectors` ενημερώνονται με τις τρέχοντες τιμές.

```
for (int l=0;l<Counter;l++) {  
tempPIDprv[l] = tempPID[l];  
prvName[l] = appName[l];  
prvstartTime[l] = startTime[l];  
}
```

3.2.4 Βοηθητική συνάρτηση `convertInt ()`

Σκοπός της συνάρτησης αυτής η μετατροπή του τύπου ακεραίου (`integer`) σε συμβολοσειρά (`string`). Η χρήση της είναι αναγκαία κάθε φορά που χρησιμοποιούμε την συνάρτηση `GetLocalTime()` για να παίρνουμε την τοπική ώρα συστήματος. Με την `GetLocalTime()` έχουμε πρόσβαση σε δείκτη δομής `SYSTEMTIME`, η οποία δομή περιέχει μέλη όπως χρονολογία, ημερομηνία, ώρα, λεπτά κ.ο.κ. Παίρνοντας τα απαραίτητα μέλη της συγκεκριμένης δομής που χρειάζεται το πρόγραμμα με τύπο ακέραιο, πρέπει να μετατραπούν σε συμβολοσειρές για να φυλαχθούν σε αρχεία. Έτσι η χρήση της κρίνεται απαραίτητη.

```
stringstream ss; //create a stringstream  
ss << number; //add number to the stream  
return ss.str(); //return a string with the contents of the stream
```

3.2.5 Βιβλιοθήκες- Επικεφαλίδες (libraries-headers)

Για την υλοποίηση του προγράμματος έγινε η χρήση ορισμένων απαραίτητων βιβλιοθηκών.

Αυτές είναι όπως χρησιμοποιήθηκαν στο πρόγραμμα οι παρακάτω:

```
#include <windows.h>
#include <dos.h>
#include <vector>
#include <psapi.h>
#include <iostream>
#include <sstream>
#include <fstream>
using namespace std;
```

windows.h

Είναι μια συγκεκριμένη επικεφαλίδα των Windows για την γλώσσα προγραμματισμού C/C++ που περιέχει τις δηλώσεις για όλες τις συναρτήσεις για το Windows API (Application Programming Interface) καθώς και για όλους τους τύπους δεδομένων που χρησιμοποιούν οι συναρτήσεις αυτές. Η χρήση της κρίνεται αναγκαία για οποιοδήποτε πρόγραμμα έχει σχέση με το λειτουργικό σύστημα Windows και με διάφορες εφαρμογές που στηρίζονται σε αυτό. Διάφοροι τύποι δεδομένων που χρησιμοποιήθηκαν για την υλοποίηση και ορίζονται σε αυτήν την επικεφαλίδα είναι:

DWORD

HINSTANCE

HANDLE

dos.h

Η επικεφαλίδα αυτή συμπεριλαμβάνεται για να είναι εφικτή η χρήση της έτοιμης συνάρτησης `sleep()` ούτως ώστε το σύστημα μας να απενεργοποιείτε για όσο χρόνο το ορίζεται στην συνάρτηση.

vector

Τα `vectors` είναι ένας τύπος δυναμικού προσδιορισμού μνήμης που ορίζεται στο `standard template library (STL)` της `C++`. Είναι ικανά να αποθηκεύουν πολλαπλούς στοιχεία οποιουδήποτε τύπου δεδομένων σε συνεχόμενα μπλοκ μνήμης. Αντίθετα με τους πίνακες (`arrays`) η πρόσβαση στα στοιχεία τους είναι πιο ασφαλής όσο χρησιμοποιούνται οι μεθόδοι που συσχετίζονται με αυτά. Επιπλέον η υπέρβαση του τέλους του `vector` θα οδηγήσει απλά σε ένα `exceprtion` παρά τον ανώμαλο τερματισμό του προγράμματος που θα συνέβαινε με τους πίνακες. Για όλους αυτούς τους λόγους προτιμήθηκε η χρήση `vector` για την αποθήκευση των ονομάτων των παραθύρων, των ωρών έναρξης και τερματισμού των παραθύρων και γενικά για αποθήκευση.

psapi.h

Είναι τα αρχικά για το `Process Status Application Programming Interface` δηλαδή κατάσταση των διεργασιών για προγραμματιστικό περιβάλλον μιας εφαρμογής. Είναι μια βοηθητική βιβλιοθήκη που ευκολύνει τον προγραμματιστή να αποκτήσει πρόσβαση σε πληροφορίες για διεργασίες και οδηγούς (`drivers`) του συστήματος. Κατά την διάρκεια της ανάπτυξης της εφαρμογής σε περιβάλλον `Visual Studio 2008 Express Edition` χρειάστηκε να συμπεριλάβουμε στις ήδη υπάρχουσες βιβλιοθήκες του περιβάλλοντος ανάπτυξης και την βιβλιοθήκη `psapi.lib` για να έχει πρόσβαση στις επιθυμητές πληροφορίες η εφαρμογή. Έτσι ο συνδιασμός αυτής της βιβλιοθήκης μαζί με την επικεφαλίδα `psapi.h` και με την επικεφαλίδα `windows.h` κρίνονται απαραίτητες για την λειτουργία της εφαρμογής.

iostream

Είναι μια ένα αρχείο επικεφαλίδα που χρησιμοποιείται για εισαγωγή και εξαγωγή (`input output`) πληροφοριών στην γλώσσα προγραμματισμού `C++`. Είναι μέρος της βασική βιβλιοθήκης της `C++` (`C++ Standard Library`). Το όνομα της αντιστοιχεί στο `Input Output Stream` όπου `stream` είναι η ροή δεδομένων. Παρέχει βασικές υπηρεσίες για την επίτευξη της εισαγωγής και εξαγωγής δεδομένων είτε από την εφαρμογή είτε προς την εφαρμογή. Η χρήση της κρίνεται αναγκαία αφού στην εφαρμογή πρέπει να καταγράψουμε διάφορες πληροφορίες για τα παράθυρα σε δύο αρχεία.

sstream

Το όνομα αυτής της βιβλιοθήκης αντιστοιχεί στο String Steam δηλαδή μια ροή δεδομένων με συμβολοσειρές. Αυτή η βιβλιοθήκη μας επιτρέπει να συμπεριφερόμαστε σε αντικείμενα της μορφής συμβολοσειράς ως ροές δεδομένων. Έτσι μπορούμε να εκτελούμε λειτουργίες πρόσθεσης και αφαίρεσης δεδομένων από και προς τις συμβολοσειρές, που χρησιμεύει στην μετατροπή των συμβολοσειρών σε ακέραιους και το αντίστροφο. Στην εφαρμογή που υλοποιήθηκε, η χρήση της είναι ακριβώς αυτή που ειπώθηκε προηγουμένως. Χωρίς αυτή την βιβλιοθήκη η συνάρτηση `convertInt()` δεν θα μπορούσε να υλοποιηθεί.

fstream

Το όνομα αυτής της βιβλιοθήκης αντιστοιχεί στο File Steam δηλαδή μια ροή δεδομένων σε ένα αρχείο. Είναι μια βασική βιβλιοθήκη της C++ που διαχειρίζεται το διάβασμα αρχείων καθώς και το γράψιμο σε αυτά είτε σαν μορφή κειμένου (text) είτε σε δυαδική μορφή (binary). Η χρήση της κρίνεται αναγκαία αφού στην εφαρμογή πρέπει να καταγράφουμε διάφορες πληροφορίες για τα παράθυρα σε δύο αρχεία.

using namespace std

Τα namespaces επιτρέπουν την ομαδοποίηση κλάσεων, αντικειμένων και συναρτήσεων κάτω από ένα κοινό όνομα. Με αυτή την δήλωση χρησιμοποιούμε το `std` (=standard) για διευκόλυνση στην χρήση των έτοιμων συναρτήσεων.

3.2.6 Προδιαγραφές συστήματος

Ξεκινώντας την δημιουργία της εφαρμογής αυτής, ένας από τους πολλούς παράγοντες που λήφθηκαν υπόψιν ήταν η επίδοση της εφαρμογής τόσο σε πόρους όσο και σε στο υλικό (hardware) του συστήματος που θα έτρεχε. Δεδομένου ότι η εφαρμογή θα χρειάζεται να τρέχει ολόκληρες βδομάδες ή και μήνες μέχρι να μαζευτεί ικανοποιητικός όγκος πληροφοριών για τις συνήθειες του χρήστη, πρέπει να μην επιβαρύνεται σύστημα του χρήστη και ούτε ο χρήστης να αναγκάζεται να τερματίζει την εφαρμογή για να ελευθερώνει διάφορους πόρους που δεσμεύει η εφαρμογή. Λαμβάνοντας όλα τα παραπάνω υπόψιν υπολήφθηκε μια εφαρμογή που έχει ελάχιστες απαιτήσεις για να τρέχει στο παρασκήνιο (background) του συστήματος. Αυτές είναι:

Λειτουργικό Σύστημα: Windows Xp – Windows Vista

Μνήμη (Ram): 256MB και άνω

Επεξεργαστής: 200MHz και άνω

Από αυτές τις απαιτήσεις συμπεραίνει κανείς ότι είναι και οι ελάχιστες για να τρέχει ο υπολογιστής και το λειτουργικό σύστημα.

Ελέγχοντας την εφαρμογή ενώ τρέχει δεσμεύει τους παρακάτω πόρους (σύμφωνα με το Windows Task Manager)

CPU Usage : 0~1%

Memory Usage : 1.092KB ~ 1MB

Βλέποντας τους πόρους που δεσμεύει η εφαρμογή συμπεραίνουμε ότι ο στόχος για χαμηλή δέσμευση πόρων ικανοποιείται αφού μόλις 1 MB μνήμη δεσμεύει και η χρήση του επεξεργαστή είναι σχεδόν πάντα μηδενική. Έτσι ο υπολογιστής δεν επιβαρύνεται και η χρήση του προγράμματος για αρκετό χρονικό διάστημα δεν επηρεάζει την όλη λειτουργικότητα του υπολογιστή.

3.2.7 Εικόνες Χρήσης της εφαρμογής

Παρακάτω προβάλλονται εικόνες από την χρήση της εφαρμογής και συγκεκριμένα από τα αρχεία στα οποία φυλάμε τα στοιχεία από τα παράθυρα που αλληλεπιδρά ο χρήστης :

Αρχείο FinishedProg.txt

```
Program Started @ 24/9/2009 @ 10:46
3352 windows Live Messenger 24/9/2009 @ 10:46 -----> 24/9/2009 @ 10:50
544 Speed Dial - Opera 24/9/2009 @ 10:46 -----> 24/9/2009 @ 11:23
2948 Speed Dial - Opera 24/9/2009 @ 11:23 -----> 24/9/2009 @ 12:12
3352 windows Live Messenger 24/9/2009 @ 12:12 -----> 24/9/2009 @ 12:12
3352 windows Live Messenger 24/9/2009 @ 12:12 -----> 24/9/2009 @ 12:12
3920 µTorrent 1.8.2 24/9/2009 @ 12:12 -----> 24/9/2009 @ 12:12
244 FinishedProg - Notepad 24/9/2009 @ 12:13 -----> 24/9/2009 @ 12:13
2964 EISSE1_08.pdf - Adobe Reader 24/9/2009 @ 12:23 -----> 24/9/2009 @ 12:41
```

Αρχείο RunningProg.txt

```
3560 help - Notepad 24/10/2009 @ 15:25 -----> Running
1224 Speed Dial - Opera 24/10/2009 @ 14:11 -----> Running
564 Program Manager 24/10/2009 @ 11:26 -----> Running
3640 Windows Media Player 24/10/2009 @ 14:30 -----> Running
684 C:\Documents and Settings\Nikos\Desktop\appl.exe 24/10/2009 @ 11:26 -----> Running
1484 AMD:CCC-AEMCapturingwindow 24/10/2009 @ 11:26 -----> Running
3812 Microsoft Word 24/10/2009 @ 14:9 -----> Running
564 Program Manager 24/10/2009 @ 11:26 -----> Running
```

Αυτή την μορφή έχουν τα αρχεία που φυλάσσονται οι πληροφορίες.

Η εφαρμογή δεν έχει γραφικό περιβάλλον μιας και τρέχει στο παρασκήνιο για εξοικονόμηση μνήμης στο σύστημα.

ΚΕΦΑΛΑΙΟ 4

Ανάλυση αποτελεσμάτων

4.1 Γενικά

Ένας από τους στόχους της διπλωματικής εργασίας είναι η ανάλυση των αποτελεσμάτων που συλλέχτηκαν από το τρέξιμο της εφαρμογής. Η εφαρμογή έτρεξε στον προσωπικό μου υπολογιστή για περίπου 14 μέρες με το ξεκίνημα του υπολογιστή κάθε φορά. Το λειτουργικό που τρέχει στον υπολογιστή είναι Microsoft Windows XP Professional Edition με Service Pack 3.

Παρακάτω αναφέρονται μερικά από τα προγράμματα που χρησιμοποιήθηκαν περισσότερο κατά την διάρκεια των 14 αυτών ημερών. Χρησιμοποιήθηκε ο MS Internet Explorer 8 και ο Opera 10 για την πλοήγηση στο διαδίκτιο. Χρησιμοποιήθηκε επίσης το MSN Messenger, το Adobe Reader καθώς και αρκετές άλλες εφαρμογές που φαίνονται αναλυτικότερα στο Παράρτημα Α.

4.2 Ανάλυση αποτελεσμάτων

Η εφαρμογή έτρεξε για συνολικά 14 μέρες αλλά ο υπολογιστής μερικές φορές ξεκινούσε δύο φορές την ημέρα. Γενικά η χρήση του υπολογιστή κυμενόταν μέχρι 6 ώρες την ημέρα, και η εκκίνηση του γινόταν καθόλη την διάρκεια της μέρας.

Συνολικά η εφαρμογή κατέγραψε την εκκίνηση του υπολογιστή 18 διαφορετικές φορές. Κάνοντας μια πρώτη ανάλυση βρίσκουμε ποσες φορές ο χρήστης χρησιμοποίησε έστω και μια φορά κάποια συγκεκριμένη εφαρμογή στο διάστημα αυτών των 18 εκκινήσεων. Συγκεκριμένα η εφαρμογή ξεκίνησε την καταγραφή των εφαρμογών τις ακόλουθες μέρες και ώρες:

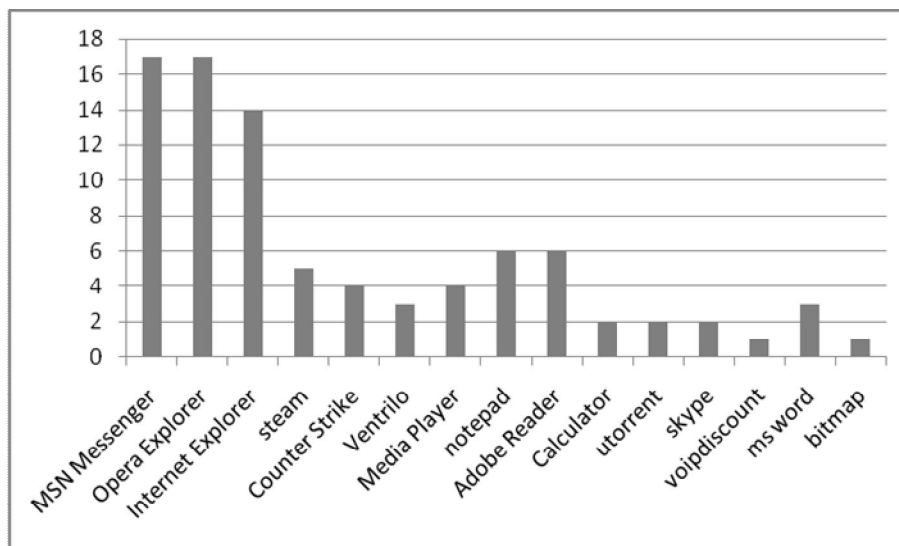
Program Started @ 14/9/2009 @ 10:30

Program Started @ 15/9/2009 @ 11:35

Program Started @ 17/9/2009 @ 11:03

Program Started @ 20/9/2009 @ 13:10
 Program Started @ 21/9/2009 @ 13:15
 Program Started @ 21/9/2009 @ 17:43
 Program Started @ 24/9/2009 @ 10:46
 Program Started @ 5/10/2009 @ 14:37
 Program Started @ 9/10/2009 @ 12:53
 Program Started @ 11/10/2009 @ 17:40
 Program Started @ 21/10/2009 @ 12:35
 Program Started @ 22/10/2009 @ 00:28
 Program Started @ 22/10/2009 @ 11:19
 Program Started @ 22/10/2009 @ 19:10
 Program Started @ 23/10/2009 @ 18:21
 Program Started @ 24/10/2009 @ 11:26
 Program Started @ 25/10/2009 @ 12:16
 Program Started @ 26/10/2009 @ 12:29

Επεξεργάζοντας τα αποτελέσματα καταλήγουμε στο παρακάτω διάγραμμα:

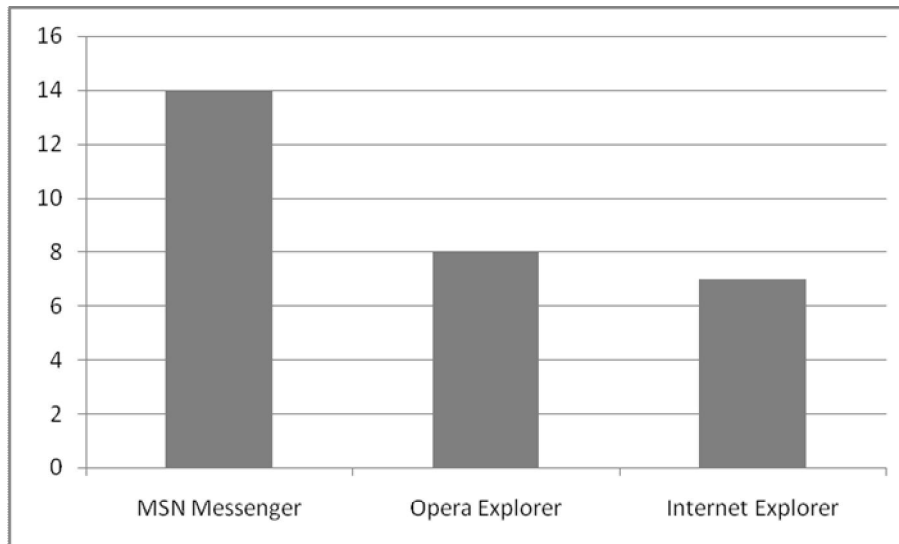


Από το παραπάνω διάγραμμα υπολογίζουμε το ποσοστό χρησιμοποίησης της κάθε εφαρμογής, δηλαδή κατά πόσο ο χρήστης θα έτρεχε γενικά την εφαρμογή και καταλήγουμε ότι:

MSN Messenger:	94.4 %
Opera Explorer:	94.4 %
Internet Explorer:	77.7 %
Steam:	27.7 %
Counter Strike:	22.2 %
Ventrilo:	16.6 %
Media Player:	22.2 %
Notepad:	33.3 %
Adobe Reader:	33.3 %
Calculator:	11.1 %
UTorrent:	11.1%
Skype:	11.1%
VoipDiscount:	5.5%
MS Word:	16.6 %
Bitmap:	5.5%

Παρατηρούμε ότι συγκεκριμένα τρεις εφαρμογές έχουν αυξημένη χρησιμοποίηση, σχεδόν κάθε φορά που ο υπολογιστής ξεκινάει, ο χρήστης θα επιλέξει να τρέξει έστω και μια φορά τις συγκεκριμένες εφαρμογές.

Προχωρώντας στην ανάλυση των καταγραφόμενων πληροφοριών παρατηρείται επίσης ότι κατά την εκκίνηση του υπολογιστή έχουμε κάποιο μοτίβο σχετικά με την εκκίνηση και κάποιων εφαρμογών. Στο παρακάτω διάγραμμα φαίνεται πόσο συχνά τα συγκεκριμένα προγράμματα ο χρήστης τα ξεκινούσε με την εκκίνηση του υπολογιστή. Μερικές φορές δύο εφαρμογές ξεκινούσαν στο ίδιο λεπτό, οπότε λήφθηκαν και οι δύο υπόψιν.



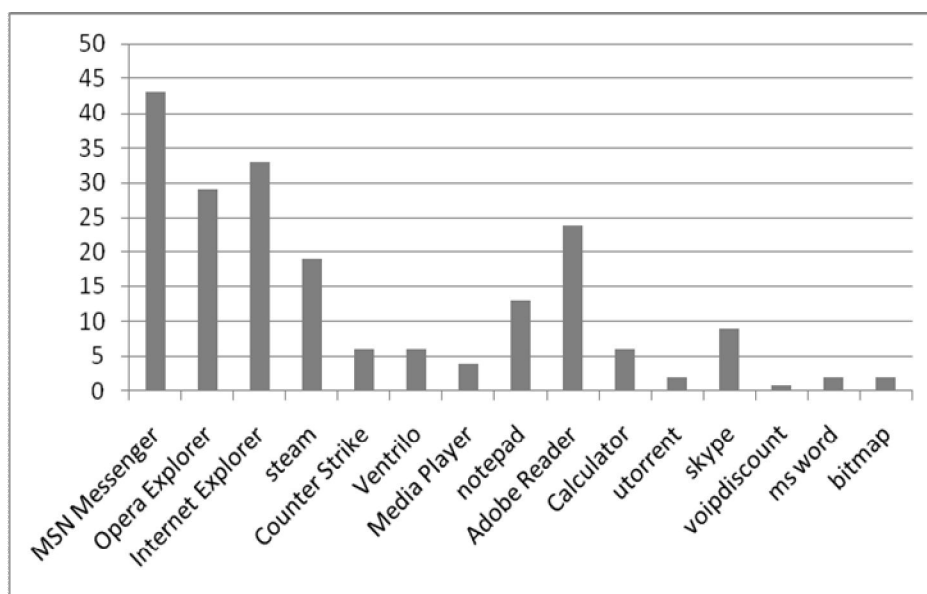
Σύμφωνα με το παραπάνω διάγραμμα παρατηρούμε ότι οι συγκεκριμένες εφαρμογές, κατά το παρακάτω ποσοστό υπήρχε η πιθανότητα να τις τρέξει με την έναρξη του υπολογιστή:

MSN Messenger: 82.3 %

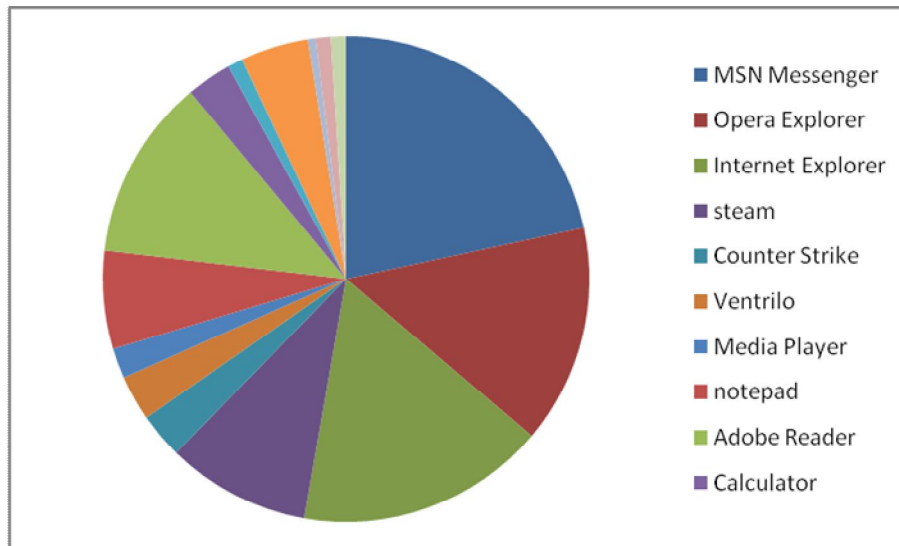
Opera Explorer: 61.5 %

Internet Explorer: 43.7 %

Κάνοντας περαιτέρω ανάλυση σχετικά με το πόσες φορές ο χρήστης αλληλεπιδρά συνολικά αυτές τις 14 μέρες με τις διάφορες εφαρμογές έχουμε το παρακάτω διάγραμμα



Παρατηρούμε ότι ο χρήστης χρησιμοποιεί περισσότερες φορές το MSN Messenger, Opera Explorer, Internet Explorer και Adobe Reader. Συγκεκριμένα στο παραπάνω διάγραμμα φαίνονται χαρακτηριστικά οι εφαρμογές που απασχολούν τον χρήστη περισσότερο. Αυτό φαινεται καλύτερα στο pie-διάγραμμα:



Συνεχίζοντας την ανάλυση μας παρατηρούμε κάποιο μοτίβο στην εκκίνηση των εφαρμογών από τον χρήστη. Κάνοντας κάποιους υπολογισμούς καταλήγουμε στα παρακάτω συμπεράσματα:

- 1) Εκκινώντας το MSN Messenger υπάρχει η πιθανότητα 78.5 % να εκκινήσει και την εφαρμογή Opera Explorer εντός 10 λεπτών.
- 2) Εκκινώντας το MSN Messenger υπάρχει η πιθανότητα 64.3 % να εκκινήσει την εφαρμογή Internet Explorer εντός 10 λεπτών.

Έχοντας κάνει τις παραπάνω αναλύσεις μπορούμε να εξάγουμε κάποια συμπεράσματα σχετικά με τις διάφορες προτιμήσεις του χρήστη. Λαμβάνοντας υπόψιν όμως ότι η εφαρμογή έτρεχε μόνο για 15 μέρες σχεδόν, δεν μπορούμε να εγγυηθούμε την ορθότητα των συμπερασμάτων που καταλήξαμε. Επομένως έχοντας μια επιφύλαξη μπορούμε να υποθέσουμε ότι όντως τα συμπεράσματα που εξάγαμε είναι αντιπροσωπευτικά του μικρού όγκου δεδομένων που είχαμε στην διάθεση μας.

ΚΕΦΑΛΑΙΟ 5

ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ

5.1 Συμπεράσματα

Η συγκεκριμένη διπλωματική εργασία αποτελεί ένα μικρό δείγμα της μελλοντικής προσφοράς των pervasive συστημάτων στην καθημερινή ζωή του ανθρώπου. Κατά την διάρκεια της εκπόνησης της διπλωματικής εργασίας έγινε η ανάπτυξη μιας εφαρμογής, ικανής να τρέχει για όσο χρονικό διάστημα είναι απαραίτητο λόγω μειωμένων απαιτήσεων σε πόρους κάνοντας έτσι δυνατή την συλλογή αρκετών χρήσιμων πληροφοριών για τις προτιμήσεις του χρήστη. Έπειτα η συλλογή των πληροφοριών αυτών αξιοποιήθηκε με μερικές απλές στατιστικές αναλύσεις στο Κεφάλαιο 4.

Η εφαρμογή που αναπτύχθηκε σε αυτή την διπλωματική εργασία είναι μια αρκετά χρήσιμη εφαρμογή αφού με την λειτουργία της δίνει την δυνατότητα παρακολούθησης των προτιμήσεων του χρήστη αθόρυβα, χωρίς ο ίδιος ο χρήστης να χρειάζεται να επεμβαίνει σε κάποιες ρυθμίσεις. Επίσης καταγράφοντας τις προτιμήσεις του χρήστη, δίνεται η δυνατότητα με κάποια επεξεργασία αυτών των καταγραφόμενων πληροφοριών να συμπεράνουμε διάφορα πράγματα για τον χρήστη όπως την κατάσταση του. Επιπλέον μπορεί να διαπιστωθεί κάποιο μοτίβο στις προτιμήσεις του χρήστη, που αναγνωρίζοντας το μια εφαρμογή μπορεί να προσφέρει διάφορες υπηρεσίες στον χρήστη όταν ο χρήστης προτίθεται να τις ζητήσει. Ένα άλλο πλεονέκτημα της εφαρμογής είναι οι μειωμένοι πόροι που απαιτεί για την σωστή λειτουργία. Όπως αναφέρθηκε και στο 3^ο Κεφάλαιο, η εφαρμογή απασχολει μόλις 1MB RAM ελεύθερο για να τρέξει καθώς και η απασχόληση του επεξεργαστή είναι μηδενική σχεδόν την περισσότερη ώρα. Όλα τα παραπάνω καθιστούν την εφαρμογή εξαιρετικά φορητή (portable) και εύχρηστη αφού μπορεί να τρέξει σε σχεδόν όλα τα σημερινά μηχανήματα χωρίς επιπλέον επιβάρυνση για αρκετό μεγάλο διάστημα.

Φυσικά η καταγραφή αυτή των προτιμήσεων του χρήστη μπορεί να θεωρηθεί και σαν καταπάτηση των ιδιωτικών του δικαιωμάτων. Έτσι ο χρήστης πρέπει να είναι ενήμερος και σύμφωνος με την χρήση της εφαρμογής για αν μην υπάρχουν νομικά προβλήματα. Τέλος τίθεται και ένα θέμα ασφάλειας των δεδομένων που καταγράφονται. Επειδή

πρόκειται για προσωπικά δεδομένα που αφορούν τις συνήθειες και τις προτιμήσεις του χρήστη, κάποια μέτρα ασφαλείας πρέπει να ληφθούν για την πρόληψη κακόβουλων ενεργειών από τρίτους.

Τέλος μέσα από αυτή την ανάλυση παρατηρούμε ότι ο χρήστης έχει ένα μοτίβο στην συμπεριφορά του κατά την εκκίνηση του υπολογιστή. Δηλαδή, με βάση απλά μαθηματικά καταλήξαμε σε ένα συμπέρασμα, ότι ο χρήστης έχει κάποιες συγκεκριμένες συνήθειες στην σειρά με την οποία εκκινεί τις διάφορες εφαρμογές που χρησιμοποιά στην καθημερινότητα του. Επίσης παρατηρήθηκε ότι ο χρήστης έχει ιδιαίτερη προτίμηση για καθημερινή χρήση συγκεκριμένων εφαρμογών.

Όλα τα παραπάνω μας παραπέμπουν στο κομμάτι της έρευνας που γίνεται σε θέμα εξαγωγής συμπερασμάτων από τα συμφραζόμενα σε pervasive συστήματα.

5.2 Μελλοντική Εργασία

Έχοντας υπόψιν τα παραπάνω συμπεράσματα και με περαιτερω χρήση και ανάπτυξη της εφαρμογής που δημιουργήθηκε για την συγκεκριμένη διπλωματική εργασία, είναι πιθανή η ανάπτυξη μιας νέας context aware εφαρμογής που θα μπορεί με βάση την προϊστορία στις προτιμήσεις του χρήστη, να μπορεί προβλέψει τις μελλοντικές προτιμήσεις του χωρίς ο χρήστης να γνωρίζει κάτι και να προσφέρει τις υπηρεσίες όταν ο χρήστης τις χρειάζεται.

Παράρτημα Α

Αποτελέσματα Καταγραφής Πληροφοριών απο την εφαρμογή

Program Started @ 14/9/2009 @ 10:30

2684 Windows Live Messenger 14/9/2009 @ 10:30 -----> 14/9/2009 @ 10:30
2412 Speed Dial - Opera 14/9/2009 @ 10:30 -----> 14/9/2009 @ 10:39
3660 Click to play video 14/9/2009 @ 10:40 -----> 14/9/2009 @ 10:40
2684 Ζήνων X. Αντωνίου <zenonantoniou@hotmail.com> 14/9/2009 @ 10:34 -----
> 14/9/2009 @ 10:44
3552 Windows Internet Explorer 14/9/2009 @ 10:39 -----> 14/9/2009 @ 10:45
2684 Ζήνων X. Αντωνίου <zenonantoniou@hotmail.com> 14/9/2009 @ 10:56 -----
> 14/9/2009 @ 10:57
2952 Screen Saver 14/9/2009 @ 11:35 -----> 14/9/2009 @ 11:36
2728 Speed Dial - Opera 14/9/2009 @ 11:36 -----> 14/9/2009 @ 11:37
1248 μTorrent 1.8.2 14/9/2009 @ 11:37 -----> 14/9/2009 @ 11:38
276 Windows Media Player 14/9/2009 @ 11:40 -----> 14/9/2009 @ 12:5
3936 Login - Steam 14/9/2009 @ 12:5 -----> 14/9/2009 @ 12:5
3936 Steam 14/9/2009 @ 12:5 -----> 14/9/2009 @ 12:6
680 OverlayHTML 14/9/2009 @ 12:26 -----> 14/9/2009 @ 12:27
3624 Counter-Strike Source 14/9/2009 @ 12:6 -----> 14/9/2009 @ 13:35
2700 Ventrilo 14/9/2009 @ 12:22 -----> 14/9/2009 @ 13:35
288 Speed Dial - Opera 14/9/2009 @ 13:53 -----> 14/9/2009 @ 13:53
1952 FinishedProg - Notepad 14/9/2009 @ 13:53 -----> 14/9/2009 @ 13:54
808 Windows Internet Explorer 14/9/2009 @ 14:0 -----> 14/9/2009 @ 14:2
2684 Windows Live Messenger 14/9/2009 @ 14:2 -----> 14/9/2009 @ 14:3
1448 Screen Saver 14/9/2009 @ 14:25 -----> 14/9/2009 @ 14:27
2348 Login - Steam 14/9/2009 @ 14:28 -----> 14/9/2009 @ 14:28
2348 Steam 14/9/2009 @ 14:28 -----> 14/9/2009 @ 14:30
896 OpenVPN Connection (ntua-udp) 14/9/2009 @ 14:31 -----> 14/9/2009 @ 14:31
864 Login - Steam 14/9/2009 @ 14:31 -----> 14/9/2009 @ 14:31
1968 Ventrilo 14/9/2009 @ 14:32 -----> 14/9/2009 @ 14:32
864 Steam 14/9/2009 @ 14:31 -----> 14/9/2009 @ 14:49
3152 Counter-Strike Source 14/9/2009 @ 14:32 -----> 14/9/2009 @ 14:54
584 Speed Dial - Opera 14/9/2009 @ 15:30 -----> 14/9/2009 @ 15:31

3244 Counter-Strike Source 14/9/2009 @ 14:55 -----> 14/9/2009 @ 15:38
864 STEAM - nykos_ole 14/9/2009 @ 14:55 -----> 14/9/2009 @ 15:38
3176 Ventrilo 14/9/2009 @ 14:46 -----> 14/9/2009 @ 15:38
896 Exit OpenVPN 14/9/2009 @ 15:38 -----> 14/9/2009 @ 15:38
1576 AMD:CCC-AEMCapturingWindow 14/9/2009 @ 10:30 -----> 14/9/2009 @ 15:46

Program Started @ 15/9/2009 @ 11:35

1308 Windows Internet Explorer 15/9/2009 @ 11:36 -----> 15/9/2009 @ 11:38
416 Windows Live Messenger 15/9/2009 @ 11:36 -----> 15/9/2009 @ 11:45
2240 Speed Dial - Opera 15/9/2009 @ 11:38 -----> 15/9/2009 @ 11:54
2308 Login - Steam 15/9/2009 @ 11:54 -----> 15/9/2009 @ 11:54
252 Ventrilo 15/9/2009 @ 11:56 -----> 15/9/2009 @ 11:56
2308 Steam 15/9/2009 @ 11:55 -----> 15/9/2009 @ 11:57
3816 Counter-Strike Source 15/9/2009 @ 11:57 -----> 15/9/2009 @ 12:6
3352 OverlayHTML 15/9/2009 @ 12:1 -----> 15/9/2009 @ 12:6
2308 STEAM - nykos_ole 15/9/2009 @ 12:6 -----> 15/9/2009 @ 12:6
3448 Counter-Strike Source 15/9/2009 @ 12:6 -----> 15/9/2009 @ 12:31
2308 STEAM - nykos_ole 15/9/2009 @ 12:31 -----> 15/9/2009 @ 12:31
416 Windows Live Messenger 15/9/2009 @ 12:31 -----> 15/9/2009 @ 12:31
3468 Ventrilo 15/9/2009 @ 12:6 -----> 15/9/2009 @ 12:31
416 Windows Live Messenger 15/9/2009 @ 13:1 -----> 15/9/2009 @ 13:1
2936 Screen Saver 15/9/2009 @ 13:47 -----> 15/9/2009 @ 14:1
416 Windows Live Messenger 15/9/2009 @ 14:2 -----> 15/9/2009 @ 14:2
2832 FinishedProg - Notepad 15/9/2009 @ 14:2 -----> 15/9/2009 @ 14:2
612 RunningProg - Notepad 15/9/2009 @ 14:2 -----> 15/9/2009 @ 14:3
3204 Screen Saver 15/9/2009 @ 14:13 -----> 15/9/2009 @ 15:1
3092 Windows Internet Explorer 15/9/2009 @ 15:1 -----> 15/9/2009 @ 15:2
3932 Windows Internet Explorer 15/9/2009 @ 15:2 -----> 15/9/2009 @ 15:36
3596 Windows Media Player 15/9/2009 @ 12:32 -----> 15/9/2009 @ 15:41
3832 Windows Internet Explorer 15/9/2009 @ 15:41 -----> 15/9/2009 @ 15:41
2784 Login - Steam 15/9/2009 @ 15:41 -----> 15/9/2009 @ 15:41
2784 Steam 15/9/2009 @ 15:41 -----> 15/9/2009 @ 15:42

Program Started @ 17/9/2009 @ 11:3

3564 Windows Live Messenger 17/9/2009 @ 11:3 -----> 17/9/2009 @ 11:4
 4032 Connect to acadinfo.central.ntua.gr 17/9/2009 @ 11:19 -----> 17/9/2009 @ 11:19
 3564 Windows Live Messenger 17/9/2009 @ 11:21 -----> 17/9/2009 @ 11:21
 3564 Windows Live Messenger 17/9/2009 @ 11:23 -----> 17/9/2009 @ 11:23
 3168 Speed Dial - Opera 17/9/2009 @ 11:17 -----> 17/9/2009 @ 11:28
 2448 Windows Internet Explorer 17/9/2009 @ 11:4 -----> 17/9/2009 @ 11:29
 436 Login - Steam 17/9/2009 @ 11:46 -----> 17/9/2009 @ 11:47
 3552 Windows Internet Explorer 17/9/2009 @ 12:0 -----> 17/9/2009 @ 12:4
 436 Steam 17/9/2009 @ 11:47 -----> 17/9/2009 @ 12:8
 4092 Windows Internet Explorer 17/9/2009 @ 12:8 -----> 17/9/2009 @ 12:8
 536 FinishedProg - Notepad 17/9/2009 @ 12:9 -----> 17/9/2009 @ 12:10
 3960 Windows Internet Explorer 17/9/2009 @ 12:10 -----> 17/9/2009 @ 12:10
 1124 FinishedProg - Notepad 17/9/2009 @ 12:10 -----> 17/9/2009 @ 12:10
 584 FinishedProg - Notepad 17/9/2009 @ 12:10 -----> 17/9/2009 @ 12:11
 3268 RunningProg - Notepad 17/9/2009 @ 12:10 -----> 17/9/2009 @ 12:11
 3088 RunningProg - Notepad 17/9/2009 @ 12:11 -----> 17/9/2009 @ 12:11
 292 FinishedProg - Notepad 17/9/2009 @ 12:11 -----> 17/9/2009 @ 12:11
 2880 Windows Media Player 17/9/2009 @ 12:12 -----> 17/9/2009 @ 12:21
 436 STEAM - nykos_ole 17/9/2009 @ 12:21 -----> 17/9/2009 @ 12:21
 3944 Counter-Strike Source17/9/2009 @ 12:21 -----> 17/9/2009 @ 13:17
 1244 Speed Dial - Opera 17/9/2009 @ 13:21 -----> 17/9/2009 @ 13:28
 3564 Ζήνων X. Αντωνίου <zenonantoniou@hotmail.com> 17/9/2009 @ 13:9 ----->
 17/9/2009 @ 13:28
 436 STEAM - nykos_ole 17/9/2009 @ 13:31 -----> 17/9/2009 @ 13:31
 3136 Counter-Strike Source17/9/2009 @ 13:31 -----> 17/9/2009 @ 13:32
 3564 Windows Live Messenger 17/9/2009 @ 14:21 -----> 17/9/2009 @ 14:22
 436 STEAM - nykos_ole 17/9/2009 @ 14:22 -----> 17/9/2009 @ 14:23
 1632 Counter-Strike Source17/9/2009 @ 14:23 -----> 17/9/2009 @ 14:33
 3508 Ventrilo 17/9/2009 @ 12:21 -----> 17/9/2009 @ 14:34

Program Started @ 20/9/2009 @ 13:10

2748 Windows Live Messenger 20/9/2009 @ 13:11 -----> 20/9/2009 @ 13:12
 1144 Windows Internet Explorer 20/9/2009 @ 13:11 -----> 20/9/2009 @ 13:15

264 SHE_2007_Xeimerino.pdf - Adobe Reader 20/9/2009 @ 13:34 -----> 20/9/2009 @ 13:35
556 SHE_2007_Xeimerino.pdf - Adobe Reader 20/9/2009 @ 13:36 -----> 20/9/2009 @ 13:36
4076 ΣΗΕ-8η σειρά.pdf (SECURED) - Adobe Reader 20/9/2009 @ 13:38 -----> 20/9/2009 @ 13:38
520 ΣΗΕ-Ασκήσεις 2004.rar - WinRAR (evaluation copy) 20/9/2009 @ 13:39 -----> 20/9/2009 @ 13:39
780 she8hseira.pdf (SECURED) - Adobe Reader 20/9/2009 @ 13:39 -----> 20/9/2009 @ 13:39
3152 she7hseira.pdf (SECURED) - Adobe Reader 20/9/2009 @ 13:39 -----> 20/9/2009 @ 13:39
1992 she6hseira.pdf (SECURED) - Adobe Reader 20/9/2009 @ 13:40 -----> 20/9/2009 @ 13:40
3528 ΣΗΕ-Ασκήσεις 2004.rar - WinRAR (evaluation copy) 20/9/2009 @ 13:39 -----> 20/9/2009 @ 13:40
3660 Speed Dial - Opera 20/9/2009 @ 13:15 -----> 20/9/2009 @ 13:41
3788 Speed Dial - Opera 20/9/2009 @ 13:41 -----> 20/9/2009 @ 13:41
1064 Adobe Updater 20/9/2009 @ 13:41 -----> 20/9/2009 @ 13:41
616 Login - Steam 20/9/2009 @ 13:45 -----> 20/9/2009 @ 13:46
616 Steam 20/9/2009 @ 13:46 -----> 20/9/2009 @ 13:46
2748 NIKOLAS <nikolasmethodiou@hotmail.com> 20/9/2009 @ 13:25 -----> 20/9/2009 @ 13:47

Program Started @ 21/9/2009 @ 13:15

3724 Windows Live Messenger 21/9/2009 @ 13:16 -----> 21/9/2009 @ 13:16
2300 Windows Internet Explorer 21/9/2009 @ 13:16 -----> 21/9/2009 @ 13:16
3516 Speed Dial - Opera 21/9/2009 @ 13:16 -----> 21/9/2009 @ 13:30
1764 137_ΠΡΟΓΡΑΜΜΑ ΕΞΕΤΑΣΕΩΝ_ΣΕΠΤΕΜΒΡΙΟΣ 09_Ορθή Επανάληψη.pdf - Adobe Reader 21/9/2009 @ 13:30 -----> 21/9/2009 @ 13:30

Program Started @ 21/9/2009 @ 17:43

3108 Calculator 21/9/2009 @ 17:44 -----> 21/9/2009 @ 17:46
4000 Windows Internet Explorer 21/9/2009 @ 17:46 -----> 21/9/2009 @ 17:50

3744 LinkClick2.pdf - Adobe Reader 21/9/2009 @ 17:50 -----> 21/9/2009 @ 18:3
1120 LinkClick3.pdf - Adobe Reader 21/9/2009 @ 18:3 -----> 21/9/2009 @ 18:17
3724 LinkClick4.pdf - Adobe Reader 21/9/2009 @ 18:17 -----> 21/9/2009 @ 18:18
3424 LinkClick5.pdf - Adobe Reader 21/9/2009 @ 18:18 -----> 21/9/2009 @ 18:18
1048 Calculator 21/9/2009 @ 18:34 -----> 21/9/2009 @ 18:36
2976 LinkClick6.pdf - Adobe Reader 21/9/2009 @ 18:18 -----> 21/9/2009 @ 18:42
3924 askisi1.pdf - Adobe Reader 21/9/2009 @ 18:43 -----> 21/9/2009 @ 18:44
3224 askisi2.pdf - Adobe Reader 21/9/2009 @ 18:44 -----> 21/9/2009 @ 18:44
2900 LinkClick8.pdf - Adobe Reader 21/9/2009 @ 18:47 -----> 21/9/2009 @ 18:47
1260 LinkClick7.pdf - Adobe Reader 21/9/2009 @ 18:47 -----> 21/9/2009 @ 19:3

Program Started @ 24/9/2009 @ 10:46

3352 Windows Live Messenger 24/9/2009 @ 10:46 -----> 24/9/2009 @ 10:50
544 Speed Dial - Opera 24/9/2009 @ 10:46 -----> 24/9/2009 @ 11:23
2948 Speed Dial - Opera 24/9/2009 @ 11:23 -----> 24/9/2009 @ 12:12
3352 Windows Live Messenger 24/9/2009 @ 12:12 -----> 24/9/2009 @ 12:12
3352 Windows Live Messenger 24/9/2009 @ 12:12 -----> 24/9/2009 @ 12:12
3920 µTorrent 1.8.2 24/9/2009 @ 12:12 -----> 24/9/2009 @ 12:12
244 FinishedProg - Notepad 24/9/2009 @ 12:13 -----> 24/9/2009 @ 12:13
2964 EISSHE1_08.pdf - Adobe Reader 24/9/2009 @ 12:23 -----> 24/9/2009 @ 12:41
2992 Screen Saver 24/9/2009 @ 12:51 -----> 24/9/2009 @ 12:52

Program Started @ 5/10/2009 @ 14:37

3380 Windows Live Messenger 5/10/2009 @ 14:37 -----> 5/10/2009 @ 14:43
1732 Speed Dial - Opera 5/10/2009 @ 14:37 -----> 5/10/2009 @ 14:45
2156 Windows Internet Explorer 5/10/2009 @ 14:45 -----> 5/10/2009 @ 14:48
2440 Speed Dial - Opera 5/10/2009 @ 14:48 -----> 5/10/2009 @ 15:15
632 Windows Internet Explorer 5/10/2009 @ 14:54 -----> 5/10/2009 @ 15:15
3000 Splasher 5/10/2009 @ 15:15 -----> 5/10/2009 @ 15:17
2848 Skype™ - nykos2209 5/10/2009 @ 15:17 -----> 5/10/2009 @ 15:17
3228 SysFader 5/10/2009 @ 15:18 -----> 5/10/2009 @ 15:18
3372 Barkston Gardens Hotel Map 5/10/2009 @ 15:20 -----> 5/10/2009 @ 15:20
2508 Windows Internet Explorer 5/10/2009 @ 15:18 -----> 5/10/2009 @ 15:28
1900 Speed Dial - Opera 5/10/2009 @ 15:17 -----> 5/10/2009 @ 15:28

2820 Screen Saver 5/10/2009 @ 15:38 -----> 5/10/2009 @ 15:40
2848 Skype™ - nykos2209 5/10/2009 @ 15:28 -----> 5/10/2009 @ 15:40
3000 VoipDiscount - nykos22 5/10/2009 @ 15:40 -----> 5/10/2009 @ 15:41

Program Started @ 9/10/2009 @ 12:53

724 http://www.answers.com/topic/context - Opera 9/10/2009 @ 12:54 ----->
9/10/2009 @ 13:4
3696 Windows Live Messenger 9/10/2009 @ 12:53 -----> 9/10/2009 @ 13:4

Program Started @ 11/10/2009 @ 17:40

308 Skype 11/10/2009 @ 17:40 -----> 11/10/2009 @ 17:40
308 Skype™ - nykos2209 11/10/2009 @ 17:40 -----> 11/10/2009 @ 17:44
3640 Windows Live Messenger 11/10/2009 @ 17:40 -----> 11/10/2009 @ 17:44
688 Document1 - Microsoft Word 11/10/2009 @ 17:44 -----> 11/10/2009 @ 17:50
1296 context: West's Encyclopedia of American Law (Full Article) from Answers.com -
Opera 11/10/2009 @ 17:40 -----> 11/10/2009 @ 17:55
3688 DT2006-0230.doc [Compatibility Mode] - Microsoft Word 11/10/2009 @ 18:0 -----
> 11/10/2009 @ 18:1

Program Started @ 21/10/2009 @ 12:35

3556 Skype™ - nykos2209 21/10/2009 @ 12:36 -----> 21/10/2009 @ 12:37
2396 Login - Steam 21/10/2009 @ 12:37 -----> 21/10/2009 @ 12:37
2888 Speed Dial - Opera 21/10/2009 @ 12:35 -----> 21/10/2009 @ 12:38
2396 Steam 21/10/2009 @ 12:37 -----> 21/10/2009 @ 12:38
2068 Counter-Strike Source 21/10/2009 @ 12:38 -----> 21/10/2009 @ 13:2
2752 Windows Live Messenger 21/10/2009 @ 12:35 -----> 21/10/2009 @ 13:3
2272 Splasher 21/10/2009 @ 13:3 -----> 21/10/2009 @ 13:4
2752 Windows Live Messenger 21/10/2009 @ 13:8 -----> 21/10/2009 @ 13:9
700 Speed Dial - Opera 21/10/2009 @ 13:9 -----> 21/10/2009 @ 13:23
2396 STEAM - nykos_ole 21/10/2009 @ 13:23 -----> 21/10/2009 @ 13:23
2204 Counter-Strike Source 21/10/2009 @ 13:23 -----> 21/10/2009 @ 13:23
3356 Speed Dial - Opera 21/10/2009 @ 13:24 -----> 21/10/2009 @ 13:25
4040 Windows Internet Explorer 21/10/2009 @ 13:25 -----> 21/10/2009 @ 13:26
2524 Screen Saver 21/10/2009 @ 14:6 -----> 21/10/2009 @ 14:9

2628 Windows Internet Explorer 21/10/2009 @ 14:10 -----> 21/10/2009 @ 14:10
312 FinishedProg - Notepad 21/10/2009 @ 14:10 -----> 21/10/2009 @ 14:11
2396 STEAM - nykos_ole 21/10/2009 @ 14:11 -----> 21/10/2009 @ 14:11
2396 STEAM - nykos_ole 21/10/2009 @ 14:11 -----> 21/10/2009 @ 14:11
684 OverlayHTML 21/10/2009 @ 14:13 -----> 21/10/2009 @ 14:15
660 Counter-Strike Source21/10/2009 @ 14:11 -----> 21/10/2009 @ 14:28
1412 Screen Saver 21/10/2009 @ 14:38 -----> 21/10/2009 @ 14:49
4012 Screen Saver 21/10/2009 @ 14:59 -----> 21/10/2009 @ 15:0
2752 Windows Live Messenger 21/10/2009 @ 13:51 -----> 21/10/2009 @ 15:0
2396 STEAM - nykos_ole 21/10/2009 @ 15:0 -----> 21/10/2009 @ 15:0
3780 Counter-Strike Source21/10/2009 @ 15:1 -----> 21/10/2009 @ 15:11

Program Started @ 22/10/2009 @ 0:28

3856 Windows Internet Explorer 22/10/2009 @ 0:29 -----> 22/10/2009 @ 0:29
3148 SHE_Π-Ω_Sept2009.pdf - Adobe Reader 22/10/2009 @ 0:31 -----> 22/10/2009
@ 0:33
3644 SHE_Π-Ω_Sept2009 (1).pdf - Adobe Reader 22/10/2009 @ 0:34 ----->
22/10/2009 @ 0:34
3020 Windows Live Messenger 22/10/2009 @ 0:29 -----> 22/10/2009 @ 0:35
2364 sport-fm.gr: NovaΣΠOP FM 94.6 - Opera 22/10/2009 @ 0:28 -----> 22/10/2009
@ 0:39
3020 Windows Live Messenger 22/10/2009 @ 0:28 -----> 22/10/2009 @ 0:40

Program Started @ 22/10/2009 @ 11:19

312 Windows Live Messenger 22/10/2009 @ 11:19 -----> 22/10/2009 @ 11:19
548 Windows Internet Explorer 22/10/2009 @ 11:19 -----> 22/10/2009 @ 11:20
1668 Speed Dial - Opera 22/10/2009 @ 11:19 -----> 22/10/2009 @ 11:26
4064 Screen Saver 22/10/2009 @ 11:36 -----> 22/10/2009 @ 11:49
312 Windows Live Messenger 22/10/2009 @ 11:26 -----> 22/10/2009 @ 11:49

Program Started @ 22/10/2009 @ 19:10

1632 Windows Media Player 22/10/2009 @ 19:11 -----> 22/10/2009 @ 19:27
336 Windows Live Messenger 22/10/2009 @ 19:27 -----> 22/10/2009 @ 19:28
2172 Select message 22/10/2009 @ 19:28 -----> 22/10/2009 @ 19:28

1572 Windows Internet Explorer 22/10/2009 @ 19:28 -----> 22/10/2009 @ 19:29
3912 Speed Dial - Opera 22/10/2009 @ 19:27 -----> 22/10/2009 @ 19:34

Program Started @ 23/10/2009 @ 18:21

1344 Windows Live Messenger 23/10/2009 @ 18:21 -----> 23/10/2009 @ 18:22
1344 Windows Live Messenger 23/10/2009 @ 18:21 -----> 23/10/2009 @ 18:22
3396 Windows Internet Explorer 23/10/2009 @ 18:22 -----> 23/10/2009 @ 18:23
2348 Blank page - Opera 23/10/2009 @ 18:23 -----> 23/10/2009 @ 18:39
1344 Windows Live Messenger 23/10/2009 @ 18:39 -----> 23/10/2009 @ 18:42
2104 Splasher 23/10/2009 @ 18:43 -----> 23/10/2009 @ 18:51
660 Windows Internet Explorer 23/10/2009 @ 19:2 -----> 23/10/2009 @ 19:2
3272 Speed Dial - Opera 23/10/2009 @ 18:52 -----> 23/10/2009 @ 19:5
1344 Ζήνων X. Αντωνίου <zenonantoniou@hotmail.com> 23/10/2009 @ 18:51 ---
--> 23/10/2009 @ 19:5

Program Started @ 24/10/2009 @ 11:26

2756 Windows Live Messenger 24/10/2009 @ 11:27 -----> 24/10/2009 @ 11:29
3024 Speed Dial - Opera 24/10/2009 @ 11:29 -----> 24/10/2009 @ 11:54
2752 Windows Live Messenger 24/10/2009 @ 11:54 -----> 24/10/2009 @ 11:55
3328 Skype 24/10/2009 @ 11:55 -----> 24/10/2009 @ 11:55
584 Untitled - Notepad 24/10/2009 @ 11:56 -----> 24/10/2009 @ 11:56
3972 Windows Internet Explorer 24/10/2009 @ 11:56 -----> 24/10/2009 @ 11:57
3840 Screen Saver 24/10/2009 @ 12:31 -----> 24/10/2009 @ 12:33
3124 Screen Saver 24/10/2009 @ 12:45 -----> 24/10/2009 @ 13:16
3328 Skype™ - nykos2209 24/10/2009 @ 13:25 -----> 24/10/2009 @ 13:25
588 Splasher 24/10/2009 @ 13:23 -----> 24/10/2009 @ 13:48
3540 Windows Live Messenger 24/10/2009 @ 13:48 -----> 24/10/2009 @ 13:49
4036 Windows Internet Explorer 24/10/2009 @ 13:49 -----> 24/10/2009 @ 13:49
2968 μTorrent 1.8.2 24/10/2009 @ 14:8 -----> 24/10/2009 @ 14:8
3472 Speed Dial - Opera 24/10/2009 @ 14:6 -----> 24/10/2009 @ 14:8
1332 appl - Notepad 24/10/2009 @ 14:24 -----> 24/10/2009 @ 14:24
2968 μTorrent 1.8.2 24/10/2009 @ 14:30 -----> 24/10/2009 @ 14:30
2060 appl - WordPad 24/10/2009 @ 14:24 -----> 24/10/2009 @ 15:3
4052 help - Notepad 24/10/2009 @ 15:3 -----> 24/10/2009 @ 15:18

3376 Windows Task Manager 24/10/2009 @ 15:18 -----> 24/10/2009 @ 15:20
3340 FinishedProg - Notepad 24/10/2009 @ 15:27 -----> 24/10/2009 @ 15:33
3472 RunningProg - Notepad 24/10/2009 @ 15:27 -----> 24/10/2009 @ 15:33
3560 help - Notepad24/10/2009 @ 15:25 -----> 24/10/2009 @ 15:33
3636 New Bitmap Image (2) - Paint 24/10/2009 @ 15:29 -----> 24/10/2009 @
15:33
1224 Speed Dial - Opera 24/10/2009 @ 14:11 -----> 24/10/2009 @ 15:33
1712 DT2009-0114.pdf - Adobe Reader 24/10/2009 @ 15:36 -----> 24/10/2009 @
15:36
3756 help - Notepad24/10/2009 @ 15:37 -----> 24/10/2009 @ 15:37
180 Speed Dial - Opera 24/10/2009 @ 15:39 -----> 24/10/2009 @ 15:42
2344 Skype 24/10/2009 @ 15:43 -----> 24/10/2009 @ 15:43
3540 giannis <giannis_chamatsos@hotmail.com> 24/10/2009 @ 15:37 -----> 24/10/2009
@ 15:43
3268 Windows Internet Explorer 24/10/2009 @ 15:47 -----> 24/10/2009 @ 15:47
1728 Windows Internet Explorer 24/10/2009 @ 15:53 -----> 24/10/2009 @ 15:54
2524 Speed Dial - Opera 24/10/2009 @ 15:47 -----> 24/10/2009 @ 15:58
3084 Calculator 24/10/2009 @ 16:5 -----> 24/10/2009 @ 16:5
3540 giannis <giannis_chamatsos@hotmail.com> 24/10/2009 @ 16:6 -----> 24/10/2009
@ 16:6
5124 Calculator 24/10/2009 @ 16:9 -----> 24/10/2009 @ 16:9
5780 Calculator 24/10/2009 @ 16:12 -----> 24/10/2009 @ 16:12
4456 Calculator 24/10/2009 @ 16:14 -----> 24/10/2009 @ 16:14
4048 Windows Internet Explorer 24/10/2009 @ 16:1 -----> 24/10/2009 @ 16:17
2344 Skype™ - nykos2209 24/10/2009 @ 15:43 -----> 24/10/2009 @ 16:21
2760 help - Notepad24/10/2009 @ 16:29 -----> 24/10/2009 @ 16:29
2248 New Text Document - Notepad 24/10/2009 @ 16:29 -----> 24/10/2009 @
16:29
4204 210.pdf - Adobe Reader 24/10/2009 @ 16:30 -----> 24/10/2009 @ 16:31
5872 62043.pdf - Adobe Reader 24/10/2009 @ 16:31 -----> 24/10/2009 @ 16:31
4748 100220061038_340.pdf - Adobe Reader 24/10/2009 @ 16:31 -----> 24/10/2009
@ 16:31
3960 AndreSantos-ContextInferenceMobileApplicationsUPCASEProject.pdf - Adobe
Reader 24/10/2009 @ 16:31 -----> 24/10/2009 @ 16:31

Program Started @ 25/10/2009 @ 12:16

1996 Speed Dial - Opera 25/10/2009 @ 12:18 -----> 25/10/2009 @ 12:36
1500 Windows Internet Explorer 25/10/2009 @ 12:36 -----> 25/10/2009 @ 12:40
3768 Windows Live Messenger 25/10/2009 @ 12:16 -----> 25/10/2009 @ 12:41280

5516 Windows Internet Explorer 25/10/2009 @ 12:47 -----> 25/10/2009 @ 12:48
3768 Windows Live Messenger 25/10/2009 @ 12:47 -----> 25/10/2009 @ 12:48

Program Started @ 26/10/2009 @ 12:29

3988 Windows Live Messenger 26/10/2009 @ 12:29 -----> 26/10/2009 @ 12:29
2848 Speed Dial - Opera 26/10/2009 @ 12:30 -----> 26/10/2009 @ 12:40
3988 Windows Live Messenger 26/10/2009 @ 12:40 -----> 26/10/2009 @ 12:46
2776 anafora [Compatibility Mode] - Microsoft Word 26/10/2009 @ 12:46 ----->
26/10/2009 @ 13:0
1800 Windows Internet Explorer 26/10/2009 @ 12:46 -----> 26/10/2009 @ 13:1

Παράρτημα Β

Ολοκληρωμένος κώδικας εφαρμογής

```
#define STRICT 1
#include <windows.h>
#include <dos.h>
#include <vector>
#include <psapi.h>
#include <iostream>
#include <sstream>
#include <fstream>
#include <dos.h>
using namespace std;
#pragma comment(lib, "psapi")

vector<string> prvName(255);
vector<string> prvstartTime(255);
vector<string> appName(255);
vector<string> startTime(255);
vector<string> endTime(255);
string prvDay;
string currDay;
unsigned int tempPID[255] = {0};
unsigned int tempPIDprv[255] = {0};
int Counter=0;
int prvCounter=0;
int flag1=0;
int flag3=0;
int flag4=0;
DWORD dwThreadId, dwProcessId;
HINSTANCE hInstance;
char String[255];
HANDLE hProcess;
```

```

string convertInt(int number)
{
    stringstream ss;//create a stringstream
    ss << number;//add number to the stream
    return ss.str();//return a string with the contents of the stream
}

void CreateLog()
{
    SYSTEMTIME ft;

    if (flag3 != 0)
    {
        for (int k=0; k<prvCounter; k++)
        {
            for (int q=0; q<Counter;q++)
                if (tempPIDprv[k] == tempPID[q])
                    flag4=1;

            if (flag4==1)
                .
            else
            {
                GetLocalTime(&ft);
                string s = convertInt(ft.wHour);
                string s1 = convertInt(ft.wMinute);
                string s2 = convertInt(ft.wDay);
                string s3 = convertInt(ft.wMonth);
                string s4 = convertInt(ft.wYear);
                string temp = s2+"/"+s3+"/"+s4+" @ "+s + ":" + s1;
                endTime[k] = temp;
                ofstream myfile;
            }
        }
    }
}

```

```

        myfile.open ("FinishedProg.txt",ios::app);
        myfile << tempPIDprv[k] << '\t' << prvName[k] << '\t' <<
prvstartTime[k] << " ----> " << endTime[k] << endl;

        myfile.close();
    }

    flag4=0;

}

}

ofstream myfile2;
myfile2.open ("RunningProg.txt",ios::out);
for (int j=0; j<Counter; j++)
    myfile2 << tempPID[j] << '\t' << appName[j] << '\t' << startTime[j] << " ----
-> " << " Running" << endl;

myfile2.close();

for (int l=0;l<Counter;l++)
{
    tempPIDprv[l] = tempPID[l];
    prvName[l] = appName[l];
    prvstartTime[l] = startTime[l];
}
}

```

```

BOOL CALLBACK EnumWindowsProc(HWND hWnd, LPARAM lParam) {

```

```

    SYSTEMTIME st;

```

```

    if (!hWnd)

```

```

        return TRUE;
    if (!::IsWindowVisible(hWnd))
        return TRUE;

    if (!SendMessage(hWnd, WM_GETTEXT, sizeof(String), (LPARAM)String))
        return TRUE;

    hInstance = (HINSTANCE)GetWindowLong(hWnd, GWL_HINSTANCE);

    dwThreadId = GetWindowThreadProcessId(hWnd, &dwProcessId);

    hProcess = OpenProcess(PROCESS_ALL_ACCESS, FALSE, dwProcessId);

    for(int i=0;i<prvCounter;i++)
    {
        if (tempPIDprv[i] == dwProcessId)
        {
            flag1++;
            startTime[Counter] = prvstartTime[i];
            appName[Counter] = prvName[i];
        }
    }

    tempPID[Counter]=dwProcessId;

    if (flag1 == 0) //neo application
    {
        GetLocalTime(&st);
        string s = convertInt(st.wHour);
        string s1 = convertInt(st.wMinute);
        string s2 = convertInt(st.wDay);
        string s3 = convertInt(st.wMonth);
        string s4 = convertInt(st.wYear);
        string temp = s2+"/"+s3+"/"+s4+" @ "+s + ":" + s1;
    }

```

```

        startTime[Counter] = temp;
        appName[Counter]=String;
    }

    Counter++;
    CloseHandle(hProcess);
    flag1=0;
    return TRUE;
}

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR
lpCmdLine, int nShowCmd)

{
    SYSTEMTIME dt;
    GetLocalTime(&dt);
    string s = convertInt(dt.wHour);
    string s1 = convertInt(dt.wMinute);
    string s2 = convertInt(dt.wDay);
    string s3 = convertInt(dt.wMonth);
    string s4 = convertInt(dt.wYear);
    string temp = s2+"/"+s3+"/"+s4+" @ "+s + ":" + s1;
    ofstream myfile;
    myfile.open ("FinishedProg.txt",ios::app);
    myfile << endl;
    myfile << "Program Started @ " << temp << endl;
    myfile.close();

    while (1)
        {
            EnumWindows(EnumWindowsProc, NULL);
            CreateLog();
            prvCounter=Counter;
            Counter=0;
        }
}

```

```
        flag3=1;
        Sleep(1000);
    }

    return 0;
}
```

BIBΛΙΟΓΡΑΦΙΑ

1. M. Weiser, 'The Computer for the 21st Century'. *Scientific American*. 1991.
2. B. N. Schilit, D. M. Hilbert, and J. Trevor, 'Context-Aware Communication'. *IEEE Wireless Communication*, 2002, pp. 46-54.
3. S. B. Mokhtar, D. Fournier, N. Georgantes, and V. Issarny, 'Context-Aware Service Composition in Pervasive Computing Environments', in *Lecture Notes in Computer Science*, vol. 3943, 2006, pp. 129-140.
4. U. Mahmud, N. Iltaf, A. Rehman, and F. Kamran, 'Context-Aware Paradigm for a Pervasive Computing Environment (CAPP)', in *proceedings of WWW/Internet 2007*, Villa Real, Portugal, vol. 1, pp. 337-346, 5th - 8th October, 2007.
5. R. Ballagas et al, "The smart phone: a ubiquitous input device," *IEEE Pervasive Computing*, vol. 5, Jan-Mar 2006, pp. 70-77.
6. R. Beale, "Supporting social interaction with smart phones," *IEEE Pervasive Computing*, Vol. 4, Jan-Mar 2005, pp. 35-41.
7. M. Ebling, G. Hunt, H. Lei, "Issues for context services for pervasive computing," *Proc. Advanced Topic Workshop Middleware for Mobile Computing, IFIP/ACM Middleware 2001 Conference*, 2001, pp. 55-65.
8. Z. Yu et al, "Supporting context-aware media recommendations for smart phones," *IEEE Pervasive Computing*, Vol. 5, Jul-Sep 2006, pp. 68-75.
9. H. Hagraas, "Embedding computational intelligence in pervasive spaces," *IEEE Pervasive Computing*, Vol. 6, Jul-Sep 2007, pp. 85-89.
10. M. Barbeau, "Mobile, distributed, and pervasive computing," *Handbook of Wireless Networks and Mobile Computing*, I. Stojmenovic, Ed., John Wiley and Sons, 2002.
11. R. Casas et al, "Hidden issues in deploying an indoor location system," *IEEE Pervasive Computing*, Vol. 6, Apr-Jun 2007, pp. 62-69.
12. L. McLaughlin, "Next-generation entertainment: video goes mobile," *IEEE Pervasive Computing*, Vol. 6, Jan-Mar 2007, pp. 7-10.
13. M. Satyanarayanan, "Pervasive computing: vision and challenges," *IEEE Personal Communications*, Vol. 8, Aug 2001, pp. 10-17.
14. H. Chu et al, "QoS-aware resource management for distributed multimedia applications," *Journal on High-Speed Networking*, Vol. 7, Dec 1998, pp. 229 - 257.
15. M. Ebling, G. Hunt, H. Lei, "Issues for context services for pervasive computing," *Proc. Advanced Topic Workshop Middleware for Mobile Computing, IFIP/ACM Middleware 2001 Conference*, 2001, pp. 55-65.

16. Jaemoon Sim, Jihoon Kim, Ohbyung Kwon ‘‘Applying Inference Engine to Context-Aware Computing Services’’
17. R. Want, A. Hopper, V. Falcao and J. Gibbons, ‘The Active Badge Location System’, *ACM Transactions on Information Systems*, 10(1), 1992, pp. 91-102.
18. K. Chaverst, K. Mitchell, and N. Davies, ‘Design of an Object Model for a Context Sensitive Tourist GUIDE’, in *proceedings of the International Workshop on Interactive Applications of Mobile Computing*, 1998.
19. G. D. Abowd, et al., ‘Cyberguide: a mobile Context-Aware Tour Guide’, *ACM Wireless Networks*, 1997.
20. T. Hofer, W. Schwinger, M. Pichler, G. Leonhartsberger and J. Altmann, ‘Context-Awareness on Mobile Devices –The Hydrogen Approach’, in *proceedings of the 36th Annual Hawaii International Conference on System Sciences*, pp. 292-302, 2002.
21. M. Roman, et al., ‘Gaia: A Middleware to Enable Active Spaces’, *IEEE Pervasive Computing*, 2002, pp. 74-83.
22. M. Riaz, S. L. Kiani, S. Lee, S. Han and Y. Lee, ‘Service Delivery in Context Aware Environments: Lookup and Access Control Issues’, in *proceedings of RTSCA’05*, 2005
23. H. Chen, T. Finin and A. Joshi, ‘An Intelligent Broker for Context-Aware Systems’, in *Proceedings of UbiComp 2003*, 2003, pp. 183-184.
24. M. Samulowitz, F. Michahelles and C. Linnhoff-Popien, ‘CAPEUS: An Architecture for Context-Aware Selection and Execution of Services’, in *New developments in distributed applications and interoperable systems*, 2001, pp. 23 – 39.
25. T. Gu, H. K. Pung and D. Q. Zhang, ‘A Middleware for Building Context-Aware Mobile Services’, in *Proceedings of IEEE Vehicular Technology Conference*, LA, USA, 2004, pp. 2656-2660
26. P. Fahy, and S. Clarke, ‘CASS A Middleware for Mobile Context-Aware Applications’, in *Proceedings of MobiSys 2004*, Boston, USA, 2004.
27. M. Weiser, ‘‘Some computer science issues in ubiquitous computing,’’ *Communications of the ACM*, Vol. 36, Jul 1993, pp. 75-84.
28. Evan Welbourne, Jonathan Leste, Anthony LaMarca, and Gaetano Borriello ‘Mobile Context Inference Using Low-Cost Sensors ‘
29. David West, Trent Apter and Aaron Quigley ‘A context inference and multimodal approach to mobile information access’

30. Sangjun Kim, Euiho Suh και Keedong Yoo ‘ A Study of Context inference algorithm on the web-based information systems’
31. Choonsung Shin and Woontack Woo ‘Conflict Resolution Method utilizing Context History for Context-Aware Applications’
32. Yussuf Abu Shaaban, Sarah McBurney, Nick Taylor, M. Howard Williams, Nikos Kalatzis και Ioanna Roussaki ‘User Intent to Support Proactivity in a Pervasive System’
33. Bill N. Schilit, Norman Adams, and Roy Want ‘Context-Aware Computing Applications’