



ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ

«Διδακτικής της Τεχνολογίας & Ψηφιακών Συστημάτων»

Κατεύθυνση Δικτυοκεντρικά Συστήματα

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ :

«Μελέτη, αναλυτική προδιαγραφή και υλοποίηση ενός συστήματος διαχείρισης πληροφορίας περιβάλλοντος για διεισδυτικές εφαρμογές.»

Μαρία Πολύζου –ΑΜ : ΜΕ09072

Επιβλέπων: Επίκουρος καθηγητής Απόστολος Μηλιώνης

[κενή σελίδα]

Ευχαριστίες

Η συγκεκριμένη διπλωματική εργασία εκπονήθηκε από τον Δεκέμβριο του 2012 έως τον Μάιο του 2014 στο τμήμα Δικτυοκεντρικά Συστήματα του μεταπτυχιακού προγράμματος Διδακτικής της Τεχνολογίας & Ψηφιακών Συστημάτων του Πανεπιστημίου Πειραιώς.

Θέλω να ευχαριστήσω ιδιαίτερω τον καθηγητή μου Απόστολο Μηλιώνη για την βοήθεια επιλογής του συγκεκριμένου θέματος, την συνεχή καθοδήγηση και άμεση υποστήριξη σε όλα τα προβλήματα που αντιμετώπισα σχετικά με την κατανόηση του θέματος και την ανάπτυξη του περιεχομένου του. Κυρίως όμως θέλω να τον ευχαριστήσω για την υπομονή και την εμπιστοσύνη που μου έδειξε εξ αρχής.

Επίσης θέλω να ευχαριστήσω θερμά όλους τους αγαπημένους φίλους μου, οι οποίοι με βοήθησαν σημαντικά να αντιμετωπίσω με θάρρος και δύναμη όλες τις δυσκολίες που μου παρουσιάστηκαν τα τελευταία χρόνια στη ζωή μου, δίνοντας μου παράλληλα κουράγιο και ελπίδα να συνεχίσω και να ολοκληρώσω την διπλωματική μου εργασία. Ευχαριστώ ιδιαίτερα τους φίλους μου Κώστα Στρογγυλό και Γιώργο Τόλογλου. Ευχαριστώ του γονείς μου Μιχάλη και Έφη, την αδελφή μου Ευσταθία καθώς και τον Πάρη για την αγάπη, φροντίδα και υποστήριξη που παρέχουν όλα αυτά τα χρόνια. Ευχαριστώ όλους τους συναδέλφους μου για την συμπαράσταση.

Τέλος ευχαριστώ όλους τους φίλους και ανθρώπους που γνώρισα αυτό το τελευταίο διάστημα, καθώς με βοήθησαν να συνειδητοποιήσω σημαντικά πράγματα και για την στήριξη που μου παρείχε ο καθένας με τον δικό του μοναδικό τρόπο!

***Την παρούσα διπλωματική εργασία την
αφιερώνω στην οικογένεια μου και σε
όλους τους φίλους μου, για την απέραντη
αγάπη, την υπομονή, την συμπαράσταση
και κυρίως για την ψυχολογική
υποστήριξη που μου προσφέρουν όλα
αυτά τα χρόνια!***

Μαρία Πολύζου

[κενή σελίδα]

Περίληψη

Τα διάχυτα / διεισδυτικά υπολογιστικά περιβάλλοντα (Ubiquitous / Pervasive computing environments) αποτελούν ένα μεγάλο όραμα για το μέλλον και έρχονται να αντικαταστήσουν τις υπάρχουσες υπολογιστικές τεχνολογίες. Σε αυτά τα περιβάλλοντα θα διατίθεται ενσωματωμένη παντού υπολογιστική τεχνολογία τέτοια ώστε να μπορούν να προσαρμοστούν στις ανάγκες μεμονωμένων χρηστών. Τα δίκτυα, οι υπηρεσίες και οι συσκευές θα πρέπει να είναι πανταχού παρόντα και να εργάζονται από κοινού, ώστε να μπορούν να υποστηρίξουν κάθε άτομο για την εκτέλεση δραστηριοτήτων της καθημερινής ζωής του. Αντίστοιχα οι διάφορες διαδικασίες θα πρέπει να εκτελούνται με διαφανή και διακριτικό τρόπο ώστε να μην είναι αντιληπτά από τον άνθρωπο. Οι διαδικασίες που εκτελούνται από το σύστημα θα πρέπει να απαιτούν την ελάχιστη παρέμβαση του χρήστη και να ανταποκρίνονται στις ανάγκες του. Για να μπορέσουν να ανταποκριθούν σε αυτή την μεγάλη πρόκληση, τα περιβάλλοντα αυτά θα πρέπει να προσαρμόζονται αυτόματα με ευφυή τρόπο προκειμένου να παρέχουν πραγματικό όφελος στον χρήστη, αντί να του δημιουργούν εμπόδια.

Για να μπορέσει ένα τέτοιο περιβάλλον να υποστηρίξει τα παραπάνω θα πρέπει πρώτα από όλα να είναι συνεχώς προσβάσιμο το δίκτυο ώστε να είναι αδιάλειπτη η παροχή υπηρεσιών του. Επομένως είναι αναγκαίο να γίνεται αυτόματα η ανακάλυψη και ο εντοπισμός των διάφορα στοιχείων / οντοτήτων του περιβάλλοντος, αλλά και η διαχείριση και διαμόρφωσή τους ώστε να καλύπτουν τις ανάγκες του χρήστη και να εκτελούν διαδικασίες για λογαριασμό του. Οι διαδικασίες αυτές θα πρέπει να λειτουργούν αδιάλειπτα καθώς ο χρήστης κινείται μέσα στο διάχυτο περιβάλλον.

Επιπλέον ένα τέτοιο σύστημα θα πρέπει διευκολύνει τη διαχείριση και την διάχυση της νοημοσύνης του συστήματος μέσα στο περιβάλλον. Συνεπώς θα πρέπει να εκτελούνται διεργασίες που να παρέχουν πολύτιμες γνώσεις περιβάλλοντος (context awareness) για λήψη αποφάσεων συστήματος. Οπότε το σύστημα θα

πρέπει να συλλέγει, να αξιολογεί και να ταξινομεί πληροφορίες του πλαισίου (context) που αφορούν την κατάσταση όλων των οντοτήτων (φυσικών και ιδεατών) που υπάρχουν σε αυτό. Η διαδικασία αυτή ονομάζεται context management και είναι απαραίτητη προκειμένου το σύστημα να μπορεί να αξιοποιήσει τις πληροφορίες που έχει συλλέξει, ώστε να εξάγει συμπεράσματα πιο υψηλού επιπέδου, με τελικό σκοπό να παρέχει προσαρμοσμένες υπηρεσίες (εξατομίκευση υπηρεσιών) στον κάθε χρήστη του συστήματος για συγκεκριμένο τόπο, χρόνο και γεγονότα. Τέλος, καθώς πολλές από τις πληροφορίες που αποτελούν τμήμα της γνώσης που διαθέτει το σύστημα ενδέχεται να αφορούν προσωπικά και ευαίσθητα δεδομένα, είναι σημαντικό να παρέχονται μηχανισμοί διασφάλισης της ιδιωτικότητας του χρήστη.

Στην παρούσα εργασία γίνεται μελέτη, ανάλυση προδιαγραφών και υλοποίηση ενός συστήματος context management. Το SED (Sensor Event Detector) σύστημα παρέχει μηχανισμούς για την λήψη, επεξεργασία, διαχείριση και διάδοση πληροφοριών του περιβάλλοντος. Ο μηχανισμός του συστήματος καταγράφει και διατηρεί τα δεδομένα προκειμένου να υποστούν επεξεργασία και να παρέχουν συμπεράσματα υψηλότερου επιπέδου για την κατάσταση του περιβάλλοντος (context inference). Το SED σύστημα σχεδιάστηκε έτσι ώστε να επιτρέπει σε ένα έξυπνο περιβάλλον να εστιάζει στις εξατομικευμένες ανάγκες των χρηστών και να παρέχει υπηρεσίες πλήρως προσανατολισμένες και προσαρμοσμένες σε αυτούς, σύμφωνα με τις προτιμήσεις, την τοποθεσία, τις συσκευές, το δίκτυο, τις τρέχουσες συνθήκες (χρόνος και φυσικές συνθήκες) και όλες τις πληροφορίες περιβάλλοντος που σχετίζονται με τον κάθε χρήστη.

Abstract

Ubiquitous and pervasive computing environments form a significant vision for the future of computing and aim to replace the way computing is currently involved in everyday life. These environments will offer computing capabilities integrated in almost everything, in such a manner that they will be able to respond to the needs of each person. Networks, services and devices will cooperate via transparent and unobtrusive processes to form a unified virtual mesh that is available everywhere and can support the activities of people within it. These processes should require minimal user intervention and be able to adapt dynamically to the needs of each person in an intelligent manner, so that they will be actually useful instead of troublesome for the end users.

To support these operations, such an environment must provide continuous and uninterrupted access to the network and its services. It is thus necessary to automatically perform processes such as discovery, location, configuration and management of the various entities within the environment, so that these entities can work effectively towards catering for the user's needs. These operations must not be interrupted as the location of users within the environment changes.

Such an environment should also simplify the management and dispersion of intelligence and acquired knowledge. Therefore, it should provide context awareness as a service to executing processes. Consequently such a system needs to collect, evaluate, classify and sort information regarding the context of all physical and virtual entities within the environment. This process is called context management and is necessary so that the system can leverage the information collected and the knowledge acquired in order to extract higher level information and be able to provide personalized services to each user. Finally, since a significant part of the collected information may involve personal and sensitive data about users, it is very important to provide mechanisms for ensuring the security of such information and the privacy of the users.

This paper presents the requirements analysis, design and implementation of a Context Management system. SED (Sensor Event Detector) provides a robust mechanism for collecting data from various sensors, semantic uplifting and context inference, and managing the context state within a smart environment. SED was designed in such a way so that it can be used in a smart environment in order to allow the provision of personalized services to users according to their preferences, location, as well as temporal and physical conditions and other context information collected.

[κενή σελίδα]

Πίνακας Περιεχομένων

Περίληψη	5
Abstract	7
Πίνακας Περιεχομένων	9
Περιεχόμενα Εικόνων	11
Περιεχόμενα Πινάκων	12
Περιεχόμενα Code Listing	12
1. Εισαγωγή	13
Ορισμός προβλήματος	13
Σκοπός της διπλωματικής εργασίας	15
Δομή της εργασίας	16
2. Pervasive computing (Ambient intelligence).....	19
Τι είναι το Ubiquitous computing	21
Διάφοροι ορισμοί για Ubiquitous computing	22
Βασικές έννοιες του Ubiquitous computing	25
Features of Ubiquitous Computing	29
Context-aware computing	32
Ιστορική αναδρομή	32
Η Πανταχού παρούσα Πληροφορική στη Μάθηση	36
3. Αισθητήρες και εφαρμογές τους.....	43
Αισθητήρες.....	45
Κατηγορίες Αισθητήρων.....	45
Πλέγμα (δίκτυο) Αισθητήρων.....	48
Ασύρματα δίκτυα αισθητήρων.....	49
Ιστορική Αναδρομή.....	51
Εφαρμογές Αισθητήρων.....	53
Πρότυπα και Λειτουργικά.....	59
Έξυπνοι αισθητήρες	62
Smart Devices	64
Industry.....	66
Smart Environments	70
Στο χώρο της υγείας	75
Smart City	78
Mobile computing	81
Φορετά συστήματα (wearable's computing).....	82
4. Context Management σε Έξυπνα περιβάλλοντα	89
Το project M - Zone	91
Τι είναι το context	92
Κατηγορίες Πλαισίου.....	96
Φυσικό και Εικονικό Πλαίσιο	97
Χαρακτηριστικά Πλαισίου.....	98
Προϋποθέσεις Χαρακτηριστικών Context- Aware System	100
Βασικά Στοιχεία του Context -aware Computing	101
Συλλογή Πληροφοριών Πλαισίου	104
Ανάκτηση Πλαισίου	106
Μοντέλα Διαχείρισης Πλαισίου.....	107
Widgets Πλαισίου (Context Widgets).....	107
Μοντέλο Υποδομών (Infrastructure Model)	108
Μοντέλο Μαυροπίνακα (Blackboard).....	109

Κριτήρια Ανταλλαγής Πληροφοριών (Trade-off criteria)	109
Μέθοδοι αφαίρεσης.....	111
Διερμηνείς (Interpreters)	111
Συναθροιστές (Aggregators).....	113
Υπηρεσίες Πλαισίου	114
Υποστήριξη Εφαρμογής Πλαισίου	115
Τα στοιχεία Εξερευνητές του Πλαισίου	116
Προστασία Προσωπικών Δεδομένων	117
Τι μας επιτρέπει να κάνουμε ένα Πλαίσιο	119
Παραδείγματα Context-Aware Συστημάτων	122
5. Συμπεράσματα	127
6. Υλοποίηση ενός πρότυπου Context Engine.....	131
Σχεδιασμός.....	131
Στόχοι / επιθυμητά χαρακτηριστικά.....	131
Αρχιτεκτονική Συστήματος.....	134
7. Υλοποίηση	137
Εισαγωγή.....	137
Υλοποίηση του context engine.....	138
Simulation Support.....	138
Sensor Support.....	140
Context State Management.....	143
Scenario Management	145
Scenario Execution Engine.....	151
Υλοποίηση της γραφικής διεπαφής.....	153
8. Εφαρμογή και Αξιολόγηση Context Engine.....	157
Προσομοίωση εκτέλεσης σεναρίου Έξυπνου Αυτοκινήτου	157
Περιγραφή σεναρίου.....	157
Αισθητήρες και εξοπλισμός σεναρίου smart car	158
Παράμετροι σεναρίου smart car	159
Events σεναρίου smart car	161
Παράμετροι και αποτελέσματα προσομοίωσης	162
9. Future work	173
Βιβλιογραφία	177

Περιεχόμενα Εικόνων

Εικόνα 1 : Τι περιλαμβάνει ένα Ubiquitous System.....	22
Εικόνα 2 : Έξυπνο κινητό	27
Εικόνα 3 : Έξυπνες κάρτες	27
Εικόνα 4 : Apple iPad tablet	27
Εικόνα 5 : Οριζόντια επιφάνεια υπολογιστή	27
Εικόνα 6 : Κάθετος έξυπνος πίνακας.....	27
Εικόνα 7 : Χαρακτηριστικά της πανταχού παρούσα υπολογιστικής.....	31
Εικόνα 8 : Διαχρονική εξέλιξη σχέσης ανθρώπου με υπολογιστές.....	33
Εικόνα 10 : Ταξινόμηση των μαθησιακών περιβαλλόντων (Ogata and Yano, 2004).....	38
Εικόνα 11 : Χαρακτηριστική αρχιτεκτονική δικτύου αισθητήρων	48
Εικόνα 12 : Τυπικό παράδειγμα αρχιτεκτονικής ασύρματου δικτύου αισθητήρων multi-hop.....	50
Εικόνα 13 : Εξέλιξη κόμβων αισθητήρων	53
Εικόνα 14 : Παράδειγμα εφαρμογών αισθητήρων του Τεχνολογικού Πανεπιστημίου Tampere της Φινλανδίας, του τμήματος Υπολογιστικών Συστημάτων	54
Εικόνα 15 : Σύστημα αισθητήρων για έγκαιρη ανίχνευση πυρκαγιών.....	55
Εικόνα 16 : Παγκόσμια αγορά αισθητήρων 2009 – 2016	58
Εικόνα 17 : Έξυπνες συσκευές	65
Εικόνα 18 : Έξυπνη συσκευή επικοινωνίας αυτοκινήτου με κινητό τηλέφωνο.....	66
Εικόνα 19 : Τοποθέτηση ετικέτας θερμοκρασίας κατά την παραγωγή	69
Εικόνα 20 : Ασύρματη τεχνολογία αισθητήρων που χρησιμοποιείται στα διυλιστήρια, με αισθητήρες παρακολούθησης για τις κρίσιμες μεταβλητές σε διάφορες διαδικασίες , καθώς και τα επίπεδα δεξαμενών	70
Εικόνα 21 : Έξυπνο σπίτι.....	72
Εικόνα 22 : Αισθητήρες και ενεργοποιητές μέσα σε ένα δίκτυο.....	73
Εικόνα 23 : Ερευνητικό πρόγραμμα Atraco έξυπνου σπιτιού	75
Εικόνα 24 : Wireless Body Area Network.....	77
Εικόνα 25 : Sesame Ring	86
Εικόνα 26 : υπολογιστές κεφαλής Meta Space Glasses	87
Εικόνα 27: Στοιχεία toolkit πλαισίου, όπου τα αντικείμενα είναι ιεραρχημένα και τα βέλη δείχνουν τις σχέσεις με τις υποκατηγορίες	123
Εικόνα 28 : Τυπική αλληλεπίδραση μεταξύ των εφαρμογών και των στοιχείων του toolkit πλαισίου.....	123
Εικόνα 29 : Πρόοδος της τεχνολογίας.....	130
Εικόνα 30: Η αρχιτεκτονική του Context Engine.....	135
Εικόνα 31: Επισκόπηση της λειτουργίας του εικονικού ρολογιού.....	139
Εικόνα 32: Η ιεραρχική δομή των στοιχείων ενός σεναρίου	146
Εικόνα 33: Επισκόπηση της λειτουργίας του Scenario Execution Engine.....	153
Εικόνα 34: Η γραφική διεπαφή για τη ρύθμιση των παραμέτρων του συστήματος .	154
Εικόνα 35: Η γραφική διεπαφή όπου εμφανίζονται τα μηνύματα από τη λειτουργία του συστήματος.....	155

Περιεχόμενα Πινάκων

Πίνακας 1 : Σύγκριση των χαρακτηριστικών του U-Learning	40
Πίνακας 2 : Σύγκριση των Μαθησιακά Υποδείγματα	42
Πίνακας 3 : Χαρακτηριστικά αισθητήρων.....	47
Πίνακας 4: Παραδείγματα έγκυρων εκφράσεων συνθηκών για συμβάντα του Context Engine	148

Περιεχόμενα Code Listing

Code listing 1: Η βασική προγραμματιστική διεπαφή για ένα αισθητήρα	141
Code listing 2: Η γενικευμένη προγραμματιστική διεπαφή για ένα αισθητήρα.....	142
Code listing 3: Ο ορισμός ενός Context Item	144
Code listing 4: Τμήμα του ορισμού της κλάσης Context	145
Code listing 5: Η αναπαράσταση ενός σεναρίου σε μορφή XML.....	150
Code listing 6: Το σενάριο έξυπνου αυτοκινήτου σε μορφή XML.....	170

1. Εισαγωγή

Ορισμός προβλήματος

Οι εξελίξεις στον τομέα της πληροφορικής, των τηλεπικοινωνιών και της τηλεπληροφορικής είναι ραγδαίες τα τελευταία χρόνια. Η χρήση των υπολογιστών, των κινητών τηλεφώνων και του διαδικτύου έχουν εισέλθει στην καθημερινή ζωή του ανθρώπου, σχεδόν είναι αναπόσπαστα στοιχεία της ζωής του, και επιδρούν με τέτοιο τρόπο που συμβάλλουν δραστικά στην αλλαγή του τρόπου ζωής και των συνθηκών του. Αυτές οι σύγχρονες τεχνολογίες των τηλεπικοινωνιών και των υπολογιστών δημιουργούν σημαντικές αλλαγές στην κοινωνία και στην οικονομία, καθώς χρησιμοποιούνται από όλες τις κοινωνικές και ηλικιακές ομάδες. Με την πάροδο των τεχνολογικών εξελίξεων οι υπολογιστές μικραίνουν σε μέγεθος και αυξάνεται η υπολογιστική τους ισχύ, καθώς οι τιμές τους είναι προσιτές από το κοινό. Η ασύρματη επικοινωνία, είναι σημαντικός παράγοντας για την συνδεσιμότητα, μεταξύ όλης της διασκορπισμένης υπολογιστικής δύναμης. Η εμφάνιση του διαδικτύου επηρέασε σημαντικά τον τρόπο επικοινωνίας και πληροφόρηση των ανθρώπων. Για να υπάρχει επικοινωνία και πρόσβαση στις πληροφορίες του διαδικτύου, δημιουργήθηκαν νέες εφαρμογές. Η υπολογιστική ικανότητα των νέων τεχνολογιών εφαρμόζεται σε γραφεία, σε ερευνητικά εργαστήρια, πανεπιστήμια, σε αυτοκίνητα, σε ανθρώπους εν κινήσει, δηλαδή οπουδήποτε και οποιαδήποτε χρονική στιγμή. Για παράδειγμα, ενώ παλαιότερα οι υπάλληλοι έπρεπε να βρίσκονται με τη φυσική τους παρουσία στην εργασία τους, αυτό έχει ανατραπεί τα τελευταία χρόνια καθώς μπορούν να εργάζονται απομακρυσμένα, να βρίσκονται στο γραφείο τους. Με τον συνδυασμό των τεχνολογιών της κινητής υπολογιστικής και του διαδικτύου, υποστηρίζεται διαρκή σύνδεση των χρηστών μέσω διάφορων φορητών συσκευών τους, ακόμη και όταν μετακινούνται (δηλαδή αλλάζουν γεωγραφική τοποθεσία), ενώ ταυτόχρονα βελτιώνεται η επικοινωνία των χρηστών και η διακίνηση των δεδομένων.

Ο τομέας των υπολογιστικών συστημάτων εξελίχθηκε τόσο πολύ που αναπτύχθηκε μια νέα τεχνολογία, αυτή της διάχυτης υπολογιστικής ή πανταχού παρούσας

υπολογιστικής (Ubiquitous Computing) και διεισδυτικής υπολογιστικής (Pervasive Computing). Πρωτοπόρος αυτής της τεχνολογίας ήταν ο Mark Weiser και το όραμα του είναι η συγκεκριμένη τεχνολογία να προσαρμοστεί με τέτοιο τρόπο στο φυσικό περιβάλλον του ανθρώπου και με τέτοια διακριτικότητα, ώστε να τον διευκολύνει στις καθημερινές του δραστηριότητες. Οι άνθρωποι –τελικοί χρήστες- δεν χρειάζεται να μάθουν τεχνικές λεπτομέρειες λειτουργίας αυτής της τεχνολογίας, παρά μόνο τις δυνατότητες που τους δίνονται για τις διάφορες εφαρμογές που προσφέρουν. Η τεχνολογία υποστηρίζει διάφορες ηλεκτρονικές συσκευές, όπως υπολογιστές, φορητές συσκευές, ηλεκτρικές συσκευές, ρούχα, εργαλεία, σπίτια, κτίρια, και διαθέτουν μεγάλη υπολογιστική ισχύ και διασυνδεσιμότητα μεταξύ σε διάφορα δίκτυα παρά την ετερογένεια τους. Οπότε ο τρόπος αλληλεπίδρασης του ανθρώπου με τον υπολογιστή αλλάζει, καθώς η νέα τεχνολογία επιτρέπει να εφαρμόζεται με αρμονία και ομοιομορφία στο περιβάλλον, καθώς υποστηρίζει τις διάφορες δραστηριότητες του ανθρώπου με διακριτικό, φιλικό και αξιόπιστο τρόπο, δίνοντας την αίσθηση της «ήρεμης» τεχνολογίας (Calm Technology).

Η διάχυτη υπολογιστική ή αλλιώς Περιβάλλουσα Νοημοσύνη (Ambient Intelligence – Ami) έχει ως στόχο να βοηθήσει τους χρήστες σε καθημερινές εργασίες τους με έναν απρόσκοπτο διακριτικό τρόπο. Στο πλαίσιο αυτό, έχουν γίνει διάφορες έρευνες και έχουν ληφθεί ερευνητικές πρωτοβουλίες που στοχεύουν στο σχεδιασμό και την υλοποίηση των έξυπνων χώρων, σε σπίτια, γραφεία, πανεπιστήμια, σχολεία, νοσοκομεία, ξενοδοχεία, μουσεία, σε ιδιωτικούς ή δημόσιους χώρους, όπου διάφορες εγκαταστάσεις αυτοματισμού υποστηρίζουν τους χρήστες.

Η τεχνολογία των υπολογιστών εξελίχθηκε από τα μοντέλα κεντρικού υπολογιστή, στους προσωπικούς υπολογιστές, και πλέον στην διάχυτη υπολογιστική. Η διάχυτη υπολογιστική αποτελεί το «τρίτο παράδειγμα της υπολογιστικής», όπου οι διαδικασίες της υπολογιστικής γίνονται με έναν άορατο τρόπο οπουδήποτε και οποτεδήποτε, βελτιώνοντας έτσι την ποιότητα της ζωής σε όλους τους χώρους και τους τομείς. Στην διάχυτη και διεισδυτική υπολογιστική οι συσκευές, τα δίκτυα και οι εφαρμογές, επικοινωνούν συνέχεια για να καλύψουν τις απαιτήσεις και επιθυμίες των ανθρώπων που ανήκουν στο εκάστοτε περιβάλλον. Οι συσκευές

ενσωματώνονται στο περιβάλλον με τρόπο ώστε να μην είναι πλέον ορατές και αντιληπτές από το χρήστη και τελικά τα συστήματα προσφέρουν υπηρεσίες στους χρήστες οποιαδήποτε ώρα και σε οπουδήποτε σημείο και αν βρίσκονται .

Τα διάχυτα υπολογιστικά συστήματα συλλέγουν χρήσιμες πληροφορίες από το περιβάλλον του χρήστη, και διαθέτουν μηχανισμούς για την επεξεργασία τους. Οπότε είναι αναγκαίο να σχεδιαστούν τέτοιοι μηχανισμοί που θα μπορούν να συλλέγουν, να αποθηκεύουν, να επεξεργάζονται –διαχωρίζουν τις πληροφορίες και να τις ανακτούν όταν είναι απαραίτητο. Συγκεκριμένα θα πρέπει να προβλέπουν την επιθυμία ή την ανάγκη του χρήστη, οπότε να λαμβάνουν αποφάσεις για αυτόν και να ενεργούν αντίστοιχα. Οι πληροφορίες σχετικά με την κατάσταση του χρήστη ονομάζονται «πλαίσιο» (context). Το πεδίο της διαχείρισης πλαισίου (context management) προσπαθεί να αντιμετωπίσει τη δυσκολία της χρήσης του πλαισίου στην πληροφορική, προτείνοντας πληροφορίες του περιβάλλοντος που μπορούν να εξαχθούν από διαφορετικές υπολογιστικές καταστάσεις. Εφαρμόζοντας στο πλαίσιο τις κατάλληλες πληροφορίες ανάλογα και με τις απαιτήσεις, ενισχύεται η εμπειρία του χρήστη και παράγονται νέες εφαρμογές.

Σκοπός της διπλωματικής εργασίας

Στόχος της συγκεκριμένης εργασίας είναι η μελέτη και η κατανόηση των βασικών αρχών των διάχυτων υπολογιστικών συστημάτων. Στην συνέχεια η εργασία επεκτείνεται στην ανάλυση του αναφερόμενου θέματος και διερευνά τη χρήση και τη διαχείριση του πλαισίου σε ένα σύστημα, καθώς εξετάζει τα στοιχεία που απαιτούνται για τη δημιουργία ενός αποτελεσματικού συστήματος διαχείρισης περιβάλλοντος. Στην εργασία παρουσιάζονται οι έννοιες του πλαισίου, τα στοιχεία ενός context-aware συστήματος και η διαχείριση τέτοιων συστημάτων σε διάφορα περιβάλλοντα, πχ Έξυπνο σπίτι, έξυπνη πόλη. Επίσης περιγράφονται τα οφέλη, οι πιθανές χρήσεις του πλαισίου καθώς και μια επισκόπηση των υφιστάμενων context-aware συστημάτων.

Πιο συγκεκριμένα παρουσιάζεται ο σχεδιασμός και η υλοποίηση ενός συστήματος Context Management που πραγματοποιήθηκε μετά από βιβλιογραφική μελέτη αρκετών παρόμοιων συστημάτων, και είχε ως βασικό στόχο την ευελιξία του προκειμένου να μπορεί να προσαρμοστεί σε μια ευρεία γκάμα από περιβάλλοντα εκτέλεσης. Τα επιθυμητά χαρακτηριστικά και οι στόχοι που οδήγησαν στην τελική αρχιτεκτονική περιγράφονται στις επόμενες ενότητες. Τέλος εκτελείται ένα σενάριο ελέγχου, και παρουσιάζονται τα αποτελέσματα της εκτέλεσης στον χρήστη.

Δομή της εργασίας

Η δομή της εργασίας έχει ως εξής:

Κεφάλαιο 1: Σε αυτό το κεφάλαιο γίνεται μια γενική αναφορά του θέματος και ορίζεται ο στόχος της διπλωματική εργασίας.

Κεφάλαιο 2: Στο κεφάλαιο αυτό γίνεται ανάλυση εννοιών και όρων του υπό μελέτη θέματος, συγκεκριμένα για την πανταχού παρούσα υπολογιστική τεχνολογία.

Κεφάλαιο 3: Ομοίως σε αυτό το κεφάλαιο αυτό παρουσιάζονται έννοιες και όροι για τους αισθητήρες και διάφορες εφαρμογές τους σε διάφορους τομείς που αφορούν την συγκεκριμένη τεχνολογία.

Κεφάλαιο 4: Στο κεφάλαιο αυτό παρουσιάζεται αναλυτικά το context management, τι είναι, από τι στοιχεία αποτελείται, ποια είναι τα βασικά χαρακτηριστικά του, πως γίνεται η συλλογή και η ανάκτηση πληροφορία, καθώς παρουσιάζονται και διάφορα μοντέλα.

Κεφάλαιο 5: Συμπεράσματα από την μελέτη.

Κεφάλαιο 6: Σε αυτό το κεφάλαιο παρουσιάζεται ο σχεδιασμός του συστήματος Context Engine, ο στόχος και τα επιθυμητά χαρακτηριστικά και η αρχιτεκτονική του συστήματος.

Κεφάλαιο 7: Σε αυτό το κεφάλαιο παρουσιάζεται ο τρόπος υλοποίησης του Context Engine, περιγράφεται η δημιουργία προγραμματιστικών διεπαφών για αισθητήρες, ο τρόπος που αναπαρίσταται και αποθηκεύεται η πληροφορία. Επιπλέον περιγράφεται ένα σενάριο για μια περίπτωση χρήσης του Context Engine, καθώς η εκτέλεση του, η υλοποίηση της γραφικής διεπαφής..

Κεφάλαιο 8: Σε αυτό το κεφάλαιο παρουσιάζονται η προσομοίωση εκτέλεσης του σεναρίου για ένα Έξυπνο Αυτοκίνητο, καθώς και οι παράμετροι και τα αποτελέσματα εκτέλεσης του σεναρίου

Κεφάλαιο 9: Στο κεφάλαιο αυτό αναφέρονται διάφορες μελλοντικές βλέψεις για τέτοιου είδους συστήματα και πιθανές βελτιώσεις του Context Engine.

[κενή σελίδα]

2. Pervasive computing (Ambient intelligence)



Σε αυτή την ενότητα περιγράφονται και αποσαφηνίζονται διάφοροι ορισμοί για την νέα υπολογιστική τεχνολογία, καθώς αναφέρονται διάφορα στοιχεία και συστατικά του συστήματος. Επίσης γίνεται μια ιστορική αναδρομή για τον κύριο εμπνευστή της 3^{ης} γενιάς υπολογιστικής τεχνολογίας.

[κενή σελίδα]

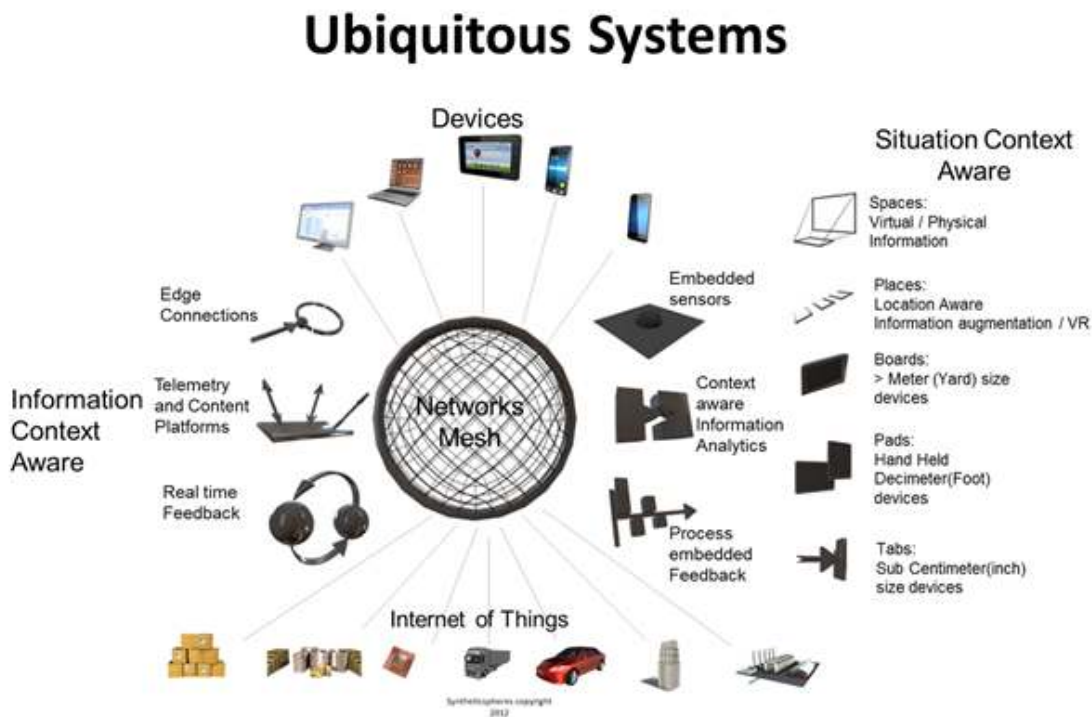
Τι είναι το Ubiquitous computing

Η πανταχού παρούσα υπολογιστική τεχνολογία ή αλλιώς διάχυτη υπολογιστική (Ubiquitous computing) είναι μια προηγμένη έννοια που σημαίνει ότι η χρήση της τεχνολογίας των υπολογιστών εμφανίζονται παντού και οπουδήποτε. Σε αντίθεση με την τεχνολογία των προσωπικών υπολογιστών (desktop computing), η διάχυτη υπολογιστικότητα μπορεί να πραγματοποιηθεί χρησιμοποιώντας οποιαδήποτε συσκευή, σε οποιαδήποτε τοποθεσία και σε οποιαδήποτε μορφή. Ένας χρήστης αλληλεπιδρά με τον υπολογιστή, το οποίο μπορεί να ποικίλει ως συσκευή κάθε φορά, δηλαδή να είναι ένας φορητός υπολογιστής (laptop), ένα tablet, ένα τερματικό καθώς και τα κινητά τηλέφωνα. Οι βασικές τεχνολογίες που πρέπει να υπάρχουν προκειμένου να υποστηρίξουν την διάχυτη υπολογιστικότητα πρέπει να περιλαμβάνουν το Διαδίκτυο, ενδιάμεσα προηγμένα λογισμικά προγράμματα (middleware), το ίδιο το λειτουργικό σύστημα, το λογισμικό του κινητού, αισθητήρες, μικροεπεξεργαστές, νέες διεπαφές (interfaces) για τα στοιχεία εισόδου / εξόδου και των χρηστών, τα δίκτυα, τα πρωτόκολλα των κινητών συσκευών, καθώς και συσκευές που δείχνουν την τοποθεσία και την ακριβή θέση.

Αυτό το νέο πρότυπο της τεχνολογίας περιγράφεται και ως περιβάλλουσα νοημοσύνη (pervasive computing). Είναι γνωστή ως φυσική υπολογιστών (physical computing), το Διαδίκτυο των Πραγμάτων (Internet of Things), απτική υπολογιστών (haptic computing) και τα πράγματα που σκέφτονται. Δεν υπάρχει ενιαίος ορισμός για την πανταχού παρούσα υπολογιστικότητα, γιατί εξαρτάται από ποια πτυχή εξετάζουμε το σύστημα και την εφαρμογή μας, οπότε ανάλογα με την ταξινόμηση των ιδιοτήτων του συστήματος δίνεται και ο αντίστοιχος ορισμός.

Η Πανταχού παρούσα υπολογιστικότητα αγγίζει ένα ευρύ φάσμα ερευνητικών θεμάτων, συμπεριλαμβανομένων των κατανεμημένων συστημάτων πληροφορικής (distributed computing), φορητών υπολογιστών (mobile computing), υπολογιστών εντοπισμού τοποθεσίας (location computing), κινητών δικτύων (mobile networking), γνώσης πλαισίου (context-aware computing), δικτύων αισθητήρων (sensor

networks), της αλληλεπίδρασης ανθρώπου-υπολογιστή (human-computer interaction) και της τεχνητής νοημοσύνης (artificial intelligence). [1]



Εικόνα 1 : Τι περιλαμβάνει ένα Ubiquitous System

Διάφοροι ορισμοί για Ubiquitous computing

Ακολουθούν διάφοροι ορισμοί που περιγράφουν λιγότερο ή περισσότερο την πανταχού παρούσα υπολογιστικότητα:

Post- PC εποχή: περιγράφει την εποχή μετά του προσωπικού υπολογιστή PC. Όμως αυτός όρος αποφεύγεται να χρησιμοποιείται, δεδομένου ότι τα στοιχεία του συστήματος πλέον δεν είναι μόνο υπολογιστές [2]

Pervasive Computing: είναι δύσκολη η διάκριση μεταξύ της λέξης πανταχού παρούσα και διάχυτη υπολογιστικότητα. Θα μπορούσε κανείς να υποστηρίξει ότι ο όρος διάχυτη αναφέρεται περισσότερο στη διαδικασία της διείσδυσης, ενώ η πανταχού παρούσα αναφέρεται περισσότερο στην «τελική κατάσταση» αυτής της

διαδικασίας. Γενικά και οι δύο ορισμοί προτείνονται ως συνώνυμα, μια μικρή διαφοροποίηση τους γίνεται συνήθως όταν η λέξη Διεισδυτική (Pervasive Computing) χρησιμοποιείται στη βιομηχανία (η IBM απέδωσε τον όρο) και η λέξη Διάχυτη (Ubiquitous Computing) χρησιμοποιείται για ακαδημαϊκούς λόγους.

Ubiquitous Computing: ο όρος δόθηκε από τον πρώτο οραματιστή της 3^{ης} γενιάς αυτής της τεχνολογίας τον Mark Weiser μπορεί να ερμηνευθεί ως «οι υπολογιστές παντού», δηλαδή διάφοροι υπολογιστές και συσκευές που να αποτελούν ένα υπολογιστικό περιβάλλον και αυτό να ανταποκρίνεται στην καθημερινή ζωή και ανάγκες του σύγχρονου ανθρώπου με τον πιο απλό, κοινό και ευδιάκριτο τρόπο.

Ambient Intelligence: η περιβάλλουσα νοημοσύνη (AmI) αναφέρεται σε ηλεκτρονικά περιβάλλοντα που είναι ευαίσθητα και να ανταποκρίνονται στην παρουσία των ανθρώπων. Είναι όραμα για το μέλλον των ηλεκτρονικών ειδών ευρείας κατανάλωσης, των τηλεπικοινωνιών και της πληροφορικής, που αρχικά αναπτύχθηκε στα τέλη της δεκαετίας του 1990 για το χρονικό διάστημα 2010-2020. Σε ένα Νοήμον Περιβάλλον, όλες οι συσκευές και τα αντικείμενα λειτουργούν από κοινού για να υποστηρίξουν διάφορες διεργασίες των ανθρώπων της καθημερινής τους ζωής, με έναν εύκολο και φυσικό τρόπο, χρησιμοποιώντας στοιχεία και πληροφορίες που είναι κρυμμένα στο δίκτυο και συνδέει αυτές τις συσκευές (Internet of Things). Όσο αυτές οι συσκευές γίνονται όλο και μικρότερες, τόσο πιο πολύ ενσωματώνονται στο περιβάλλον με τελικό στόχο να είναι σχεδόν «αόρατες» από τον άνθρωπο και τελικά το μόνο που παραμένει αντιληπτό από τους χρήστες είναι η διεπαφή του χρήστη με το σύστημα του περιβάλλοντος. Τελικά το περιβάλλον αντιλαμβάνεται την ανθρώπινη παρουσία και ανάλογα με τις προτιμήσεις του προσαρμόζεται σε αυτήν, οπότε έτσι επιτυγχάνεται η αλληλεπίδραση. Ο Mark Weiser σκόπιμα απέφευγε τον όρο «νοημοσύνη», λόγω ότι αυτή η τεχνολογία δίνει υπερπροσδοκίες. [1]

Disappearing / invisible / calm computing: αυτοί οι όροι χρησιμοποιούνται πολύ λιγότερο για την περιγραφή των ubiquitous computing και pervasive computing. Με λέξη «εξαφάνιση» περιγράφεται μια διαδικασία, ενώ με την λέξη «αόρατο» (D.

Norman 1999) περιγράφει μια «τελική κατάσταση». Η «Ηρεμία» είναι ένα είδος της τεχνολογίας των πληροφοριών, όπου η αλληλεπίδραση μεταξύ της τεχνολογίας και της χρήσης έχει σχεδιαστεί να δρα στην περιφέρειά του χρήστη και όχι συνεχώς στο επίκεντρο της προσοχής. Δηλαδή μπορεί εύκολα να μετατοπιστεί ανάμεσα στο κέντρο της προσοχής και της περιφέρειας, ή ότι το μεγαλύτερο μέρος των πληροφοριών που διαβιβάζονται από την τεχνολογία βρίσκονται στην περιφέρεια αντί στο κέντρο. Κύριοι εμπνευστές αυτής της τεχνολογίας ήταν οι Mark Weiser και John Brown Seely που την περιγράφουν ήρεμη τεχνολογία για «αυτό που ενημερώνει, αλλά δεν απαιτεί την εστίαση ή την προσοχή μας». Ο συνδυασμός της ήρεμης τεχνολογίας με την πανταχού παρούσα υπολογιστική είναι ένας τρόπος για να ελαχιστοποιηθεί η αισθητή εισβολή των ηλεκτρονικών υπολογιστών στην καθημερινή ζωή των ανθρώπων. [1] [3]

Mixed-Mode Systems: είναι ένας όρος που χρησιμοποιείται για να περιγράψει την ετερογένεια μεταξύ των ubiquitous computing κόμβων. Ο συγκεκριμένος όρος αυτός δεν χρησιμοποιείται ευρέως για την περιγραφή των ubiquitous computing καθώς τονίζει μια ιδιαίτερη πτυχή λειτουργίας τέτοιων περιβαλλόντων. [3]

Tangible Bits: ο όρος Ενσώματα Bits χρησιμοποιήθηκε στην Ολλανδία και την Ιαπωνία, αλλά παρέμεινε ασυνήθιστο σε γενικές γραμμές και δεν χρησιμοποιείται. Αναφέρεται κυρίως στο γεγονός ότι οι δικτυωμένοι υπολογιστές γίνονται όλο και πιο πολύ μέρος του φυσικού κόσμου.

Real Time Enterprise: ο όρος δεν θεωρείται ότι ως συνώνυμο για το ubiquitous computing, χρησιμοποιείται για να προσδιορίσει ένα πεδίο εφαρμογής που μπορεί να μειώσει την «καμπύλη εκμάθησης» των πληροφοριών που διαχειρίζονται τέτοια περιβάλλοντα, δηλαδή αφορά οι τιμές του ubiquitous computing των υλικών και των λύσεων που προτείνονται για διάφορες εκτελέσεις διαδικασιών. [3]

Συνοψίζοντας ο ορισμός που επικράτησε είναι του Mark Weiser μέσω του ρητού:

«Οι περισσότεροι επιδραστικές τεχνολογίες είναι αυτές που δεν γίνονται αντιληπτές από τον άνθρωπο...». [4]

Όλοι οι υπολογιστικοί και δικτυακοί πόροι που είναι διασυνδεδεμένοι με κατανεμημένο τρόπο στο φυσικό περιβάλλον, επιτρέπουν στους χρήστες μέσω των φυσικών διεπαφών, να έχουν πρόσβαση σε πληροφορίες και υπολογιστικούς πόρους από οπουδήποτε και για οποιαδήποτε χρονική στιγμή. Ταυτόχρονα παρέχουν υπηρεσίες χωρίς ο άνθρωπος να γνωρίζει ή να αντιλαμβάνεται την ύπαρξη τους στο περιβάλλον. Κύρια χαρακτηριστικά αυτών των υπηρεσιών είναι η δυνατότητα πρόσβασης σε μεγάλο πλήθος δεδομένων καθώς και η ταχύτητα υλοποίησης και προσαρμογής τους στις αντίστοιχες εξατομικευμένες απαιτήσεις, προτιμήσεις και ικανότητες του ανθρώπου και του περιβάλλοντος στο που βρίσκεται κάθε φορά. Το μεγαλύτερο πλεονέκτημα αυτής της τεχνολογίας είναι ότι οι χρήστες δεν χρειάζεται να εκπαιδεύονται για την χρήση των συσκευών και των συστημάτων διότι η τεχνολογία έχει ενσωματώνεται στο «εσωτερικό» του περιβάλλον τόσο ώστε να είναι σχεδόν «αόρατη» και να λειτουργεί στο παρασκήνιο. Στην προκειμένη περίπτωση αυτοί που χρειάζονται εκπαίδευση είναι «το περιβάλλον» των χρηστών, γιατί θα πρέπει να μαθαίνουν και να προσαρμόζονται στα χαρακτηριστικά των εκάστοτε ανθρώπων που αντιλαμβάνονται και να τους υποστηρίζουν στο μέγιστο βαθμό, ώστε τελικά να επιτυγχάνεται με φυσικό τρόπο η αλληλεπίδραση με τα συστήματα και να μη γίνεται αντιληπτή. [5] [6]

Βασικές έννοιες του Ubiquitous computing

Όλα τα μοντέλα της πανταχού παρούσας υπολογιστικής τεχνολογίας (Ubiquitous computing) μοιάζουν στον πυρήνα της σχεδίασης τους καθώς αποσκοπούν σε ένα κοινό όραμα να είναι μικρά σε διαστάσεις, φθηνά, ανθεκτικά στις δικτυωμένες συσκευές επεξεργασίας, να διανέμονται σε όλες τις κλίμακες της καθημερινής ζωής και να μπορούν να μετατραπούν ευδιάκριτα στα κοινά τερματικά σημεία. Για

παράδειγμα ένα σενάριο ενός περιβάλλοντος με πανταχού παρούσα υπολογιστικότητα μπορεί να διασυνδέσει τον φωτισμό και διάφορους περιβαλλοντικούς ελέγχους σύμφωνα με προσωπικές βιομετρικές οθόνες του ανθρώπου που έχουν υφανθεί στα ρούχα, έτσι ώστε σύμφωνα με τις συνθήκες του φωτισμού και της θέρμανσης του χώρου να αλλάζουν συνεχώς και χωρίς αυτές να γίνονται αντιληπτές. Ένα άλλο σενάριο ενός περιβάλλοντος με πανταχού παρούσα υπολογιστικότητα μπορεί να γνωρίζει το περιεχόμενο του ψυγείου ενός σπιτιού, έχοντας διάφορες ετικέτες στα προϊόντα να σχεδιάζει διάφορα μενού και προειδοποιούν τους χρήστες τους για τα χαλασμένα τρόφιμα. [1]

Η πανταχού παρούσα υπολογιστική παρουσιάζει προκλήσεις σε αρκετούς τομείς της επιστήμης των υπολογιστών, όπως στον σχεδιασμό των συστημάτων και της μηχανικής, στη μοντελοποίηση των συστημάτων και σχεδιασμό διεπαφής χρήστη. Τα σύγχρονα μοντέλα αλληλεπίδρασης ανθρώπου με μηχανή είτε αυτά είναι με γραμμή εντολών είτε με μενού GUI –based δεν είναι αρκετά για να καλύψουν την πανταχού παρούσα υπόθεση και αυτό γιατί δεν έχει προκύψει ακόμη μια ισχυρή πανταχού παρούσα υπολογιστική. Αυτό σημαίνει ότι ακόμη δεν έχει προκύψει το παράδειγμα της φυσικής αλληλεπίδρασης για μια πλήρως ισχυρή πανταχού παρούσα υπολογιστική, παρόλο που υπάρχει η αναγνώριση της στον τομέα της περιβάλλουσας τεχνολογίας. Παραδείγματα της τεχνολογίας που υποστηρίζουν αυτήν την ιδέα είναι οι σύγχρονες συσκευές όπως τα κινητά τηλέφωνα, συσκευές αναπαραγωγής ψηφιακού ήχου , αναγνώριση ραδιοσυχνότητας ετικέτες, GPS , και διαδραστικοί πίνακες.

Ο Mark Weiser πρότεινε τρεις βασικές μορφές για την πανταχού παρούσα συσκευές του συστήματος: καρτέλες (tabs), pads και πίνακες (boards).

Tags: μικρές συσκευές που μεταφέρονται ή φοριούνται, όπως smartphones, έξυπνες κάρτες

Pads: συσκευές χειρός, όπως είναι οι φορητοί υπολογιστές.

Εφαρμογές ευφρών συστημάτων

Boards: διαδραστικές συσκευές απεικόνισης, όπως οριζόντιες επιφάνειες υπολογιστών και έξυπνα ταμπλό.



Εικόνα 2 : Έξυπνο κινητό



Εικόνα 3 : Έξυπνες κάρτες



Εικόνα 4 : Apple iPad tablet



Εικόνα 5 : Οριζόντια επιφάνεια υπολογιστή



Εικόνα 6 : Κάθετος έξυπνος πίνακας

Αυτές οι τρεις μορφές που προτείνονται από Weiser χαρακτηρίζονται από την ύπαρξη μακρο-μεγέθους, με επίπεδη μορφή και την ενσωμάτωση οθονών για οπτική έξοδο. Επίσης αυτά προβλέπονται και ως συσκευές πληροφοριών. Αν δούμε πιο χαλαρά τα χαρακτηριστικά αυτών των συσκευών, τότε μπορεί να επεκταθεί το εύρος στις συσκευές Ubiquitous Computing. Για πανταχού συστήματα έχουν προταθεί τρία επιπλέον χαρακτηριστικά: [4]

Dust (σκόνη): μικροσκοπικές συσκευές που μπορεί να μη διαθέτουν οθόνες οπτικής εξόδου, όπως Micro Electro-Mechanical Systems (MEMS), που κυμαίνονται από σε μέγεθος από νανόμετρα ως χιλιοστά του μέτρου.

Δέρμα (skin): υφάσματα που βασίζονται στην εκπομπή φωτός και αγωγή πολυμερή, οργανικά συσκευές του υπολογιστή, οι οποίες μπορούν να διαμορφωθούν σε πιο ευέλικτες επιφάνειες εμφάνισης μη επίπεδες και προϊόντα, όπως ρούχα, κουρτίνες. Οι MEMS συσκευές μπορούν να είναι ζωγραφισμένες πάνω σε διάφορες επιφάνειες, έτσι ώστε μια ποικιλία από δομές του φυσικού κόσμου να μπορεί να λειτουργήσει ως δικτυωμένη στις επιφάνειες των MEMS.

Πηλός (clay): σύνολα των MEMS μπορούν να διαμορφωθούν σε αυθαίρετα τρισδιάστατα σχήματα, όπως αντικείμενα που μοιάζουν με πολλά διαφορετικά είδη του φυσικού αντικειμένου.

Στο βιβλίο του «The Rise of the Network Society», ο Manuel Castells δείχνει τη συνεχή μετατόπιση από τους αποκεντρωμένους stand-alone υπολογιστές προς την αποκλειστική χρήση των pervasive computing. Ο Castells σε ένα μοντέλο του χρησιμοποιεί το παράδειγμα του Διαδικτύου ως την αρχή για την διάχυτη υπολογιστική δύναμη. Δηλαδή τη λογική εξέλιξη από το συγκεκριμένο παράδειγμα είναι ένα σύστημα όπου η λογική δικτύωση αρχίζει να εφαρμόζεται σε κάθε σφαίρα της καθημερινής δραστηριότητας, σε κάθε περιοχή και σε κάθε πλαίσιο. Συγκεκριμένα ο Castells προβλέπει ένα σύστημα όπου τα δισεκατομμύρια

μικροσκοπικών συσκευών ενδοεπικοινωνίας, θα βρίσκονται παντού και θα εξαπλωθούν τόσο πολύ σε όλο τον κόσμο, και τα παρομοιάζει με τέτοιο τρόπο έτσι όπως «η χρωστική ουσία που εισχωρεί στο χρώμα του τοίχου».

Η πανταχού παρούσα υπολογιστική μπορεί να θεωρηθεί ότι αποτελείται από πολλά στρώματα, όπου το καθένα από αυτά έχει το δικό του ρόλο, ενώ ταυτόχρονα όλα μαζί σχηματίζουν ένα ενιαίο σύστημα:

Επίπεδο 1: στρώμα διαχείρισης εργασιών (task management layer)

- Οθόνες για διεργασίες του χρήστη, πλαίσιο και ευρετηρίου του χρήστη
- Αντιστοίχιση των διεργασιών του χρήστη με τις ανάγκες των υπηρεσιών στο περιβάλλον
- Διαχείριση πολύπλοκων εξαρτήσεων

Επίπεδο 2: στρώμα διαχείρισης περιβάλλοντος (environment management layer)

- Παρακολούθηση ενός πόρου και των δυνατοτήτων του
- Αντιστοίχιση των αναγκών υπηρεσίας και το επίπεδο των χρηστών σε ειδικές ικανότητες

Στρώμα 3: στρώμα περιβάλλοντος (environment layer)

- Παρακολούθηση ενός σχετικού πόρου
- Διαχείριση αξιοπιστίας των πόρων. [1]

Features of Ubiquitous Computing

Σύμφωνα με τους ορισμούς που δόθηκαν στην προηγούμενη ενότητα για τα ubiquitous computing τα βασικά χαρακτηριστικά του περιβάλλοντος της πανταχού παρούσας υπολογιστικής είναι τα ακόλουθα:

- Επικεντρώνονται στον ανθρώπινο παράγοντα, οπότε το περιβάλλον δεν λειτουργεί ως «περιβάλλον υπολογιστή», αλλά με ανθρώπινο φυσικό τρόπο.
- Οι συσκευές που είναι διασυνδεδεμένες και ενσωματωμένες στο περιβάλλον είναι σχεδόν «αόρατες».
- Οι συσκευές του δικτύου κάνουν την τεχνολογία πλήρως συνδεδεμένη και συνεχώς διαθέσιμη.
- Τα χαρακτηριστικά των συσκευών συλλέγονται σε πραγματικό χρόνο.
- Οι πληροφορίες που συλλέγονται από το περιβάλλον «δημιουργούν γνώση». Επίσης, οι πληροφορίες διαδίδονται και στα υπόλοιπα περιβάλλοντα (Context Awareness and Adaptation).
- Λειτουργώντας σε πραγματικό χρόνο και έχοντας τη γνώση μπορεί να «προβλέψει» τις προτιμήσεις του χρήστη, και να ανταποκριθεί δυναμικά σε αυτές .
- Οι συσκευές προσαρμόζονται στο εκάστοτε περιβάλλον πχ ο χρήστης μετακινείται από το σπίτι στο γραφείο στο σπίτι είχε τον προσωπικό του υπολογιστή, στη διαδρομή είχε το κινητό του τηλέφωνο και στο γραφείο χρησιμοποιεί το laptop. Η πανταχού παρούσα υπολογιστική λαμβάνει υπόψη την ώρα, την τοποθεσία, τον καιρό, την θερμοκρασία κτλ. Άρα υποστηρίζει την ετερογένεια των συσκευών και του περιβάλλοντος που βασίζονται στην κινητικότητα του χρήστη.
- Η πανταχού παρούσα υπολογιστική επικεντρώνεται σε σχέσεις πολλά - προς-πολλά αντί ενός -προς- ένα, πολλά-προς - ένα ή ένα-προς-πολλά στο περιβάλλον, μαζί με την κεντρική ιδέα της τεχνολογίας που είναι συνεχώς παρούσα.
- Η συγκεκριμένη τεχνολογία βασίζεται στο Internet, στην ασύρματη τεχνολογία, καθώς και σε προηγμένα ηλεκτρονικά συστήματα.

- Η τεχνολογία έχει χαρακτηριστικά τοπικά και παγκόσμια, κοινωνικά και προσωπικά, δημόσια και ιδιωτικά, αόρατα και ορατά.
- Η τεχνολογία έχει κάποιους όταν οι πληροφορίες αφορούν την ιδιωτικότητα του χρήστη, όπως για παράδειγμα οι ψηφιακές συσκευές που φοριούνται (wearable computing) και πρέπει να είναι συνεχώς συνδεδεμένες.
- Η πανταχού παρούσα υπολογιστική χρησιμοποιεί φθηνούς επεξεργαστές, μειώνοντας τις απαιτήσεις για μεγάλη υπολογιστική μνήμη και αποθήκευση δεδομένων (RFID, Sensors, Smart Card, Information Artifacts και Tiny Smart Device/ smart dust). [7]



Εικόνα 7 : Χαρακτηριστικά της πανταχού παρούσα υπολογιστικής

Context-aware computing

Ο όρος Context-aware computing (υπολογίζει) αναφέρεται σε μια γενική κατηγορία των κινητών συστημάτων που μπορεί να «αισθανθεί» το φυσικό τους περιβάλλον και να προσαρμόζουν ανάλογα τη συμπεριφορά τους.

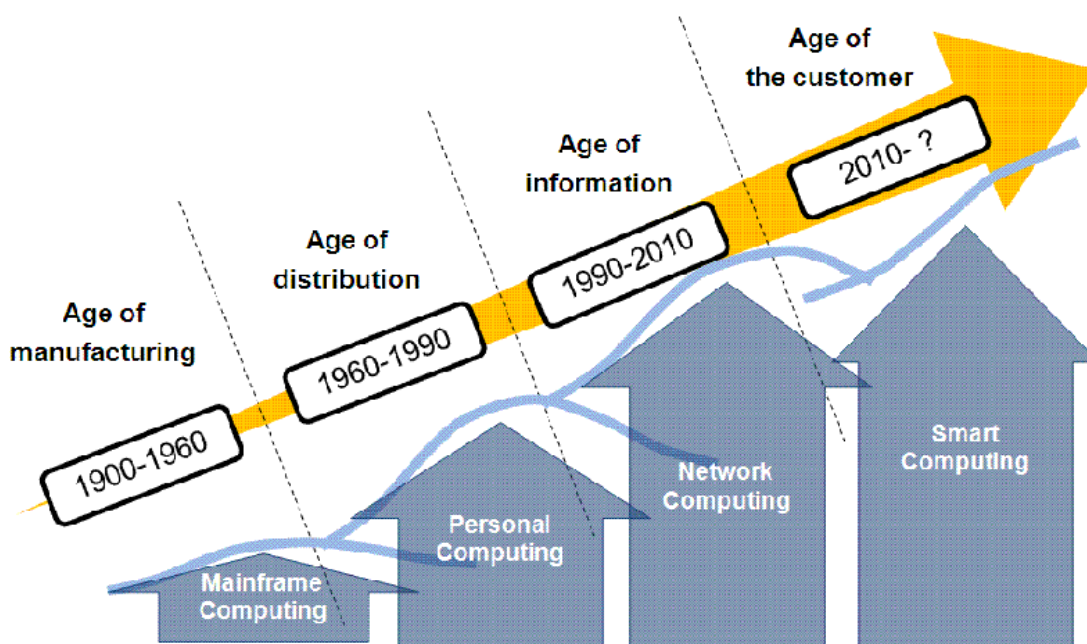
Τα context-aware συστήματα είναι μια συνιστώσα των ubiquitous computing ή αλλιώς του pervasive computing περιβάλλοντος. [8] Ένα context-aware σύστημα πρέπει να περιέχει πληροφορίες σε ερωτήσεις: πού είσαι; ποιος είσαι; και τι πόροι βρίσκονται σε κοντινή απόσταση. Το Context-aware σύστημα χρησιμοποιείται για να συμπεριλάβει πληροφορίες σχετικά με τα άτομα που βρίσκονται στο περιβάλλον, συσκευές, το φωτισμό, το επίπεδο θορύβου, τη διαθεσιμότητα του δικτύου, ακόμη και την κοινωνική κατάσταση, για παράδειγμα, αν είστε με την οικογένειά σας ή έναν φίλο από το σχολείο.

Η ιδέα προέκυψε στις αρχές του 1990 μέσω της έρευνας που γινόταν από το από Xerox PARC σχετικά με την πανταχού παρούσα υπολογιστική. [9] Ο όρος «context-aware» χρησιμοποιήθηκε για πρώτη φορά από τους ερευνητές Schilit και Theimer (1994) που περιγράφουν το πλαίσιο σαν ένα πληροφοριακό μοντέλο στο οποίο οι χρήστες του αλληλεπιδρούν χρησιμοποιώντας κινητές συσκευές και σταθερούς ηλεκτρονικούς υπολογιστές. Το context-aware σύστημα συλλέγει συνέχεια και ανά τακτά χρονικά διαστήματα, πληροφορίες από τους ανθρώπους και τα αντικείμενα και τελικά τροποποιεί την συμπεριφορά του ανάλογα με τις πληροφορίες που συλλέγει. [10]

Ιστορική αναδρομή

Η πληροφορική αποτελεί πλέον σχεδόν αναπόσπαστη σχέση ανθρώπου και υπολογιστή. Ενώ τα πρώτα χρόνια εμφάνισης των υπολογιστών οι άνθρωποι μοιραζόντουσαν τους υπολογιστικούς πόρους μέσα από έναν ενιαίο υπολογιστή (mainframe) για την εκτέλεση εργασιών, στη συνέχεια αυτό εξελίχθηκε με την

χρήση ενός υπολογιστή ανά άνθρωπο χειριστή (personal computer). Τελικά την τελευταία δεκαετία αυτό ανατράπηκε σε μια σχέση «πολλά προς ένα», δηλαδή ένας χρήστης περιβάλλεται από πολλούς υπολογιστές, ή διάφορες φορητές και μη φορητές συσκευές με επεξεργαστικές ικανότητες. Έτσι η νέα εποχή της πληροφορικής αναφέρεται ως «Ubiquitous Computing» και οι ηλεκτρονικοί υπολογιστές ενσωματώνονται σε καθημερινές δραστηριότητες της ζωής μας. Στην παρακάτω εικόνα απεικονίζεται η διαχρονική εξέλιξη αυτών τάσεων. [11]



Εικόνα 8 : Διαχρονική εξέλιξη σχέσης ανθρώπου με υπολογιστές

Σύμφωνα με τον Mark Weiser (1991), πανταχού παρούσα υπολογιστική είναι η μέθοδος για την αύξηση της χρήσης των ηλεκτρονικών υπολογιστών και την καθιστά διαθέσιμη σε όλο το φυσικό περιβάλλον. Ως αποτέλεσμα, οι υπολογιστές θα είναι ουσιαστικά αόρατοι για τον χρήστη στην καθημερινή του ζωή. Ομοίως και ο Zhang (2005) ορίζει την πανταχού παρούσα υπολογιστική περιβάλλοντος ως «well defined περιοχή» δηλαδή περιοχή –περιβάλλον- που περιλαμβάνει μια συλλογή των ενσωματωμένων συστημάτων (υπολογιστές, αισθητήρες, διεπαφές του χρήστη και της υποδομής των υπηρεσιών), η οποία ενισχύεται από την πληροφορική και επικοινωνία τεχνολογιών.

Πρώτος ο Mark Weiser επινόησε τον όρο «πανταχού παρούσα υπολογιστική» κάπου το 1988, κατά τη διάρκεια της θητείας του ως επικεφαλής Τεχνολόγος της Xerox Palo Alto Research Center (PARC). Μεμονωμένα και με PARC Διευθυντής και Chief Scientist John Seely Brown , Ο Weiser έγραψε κάποια έγγραφα μόνος του και κάποια μαζί με τον διευθυντή του PARC και τον υπεύθυνο της επιστημονικής ομάδας John Seely Brown, σχετικά με το θέμα που τον απασχολούσε για την πανταχού παρούσα υπολογιστική. [12]

Ο Weiser γνώριζε ότι η επέκταση της επεξεργαστικής ισχύς σε διάφορα καθημερινά σενάρια θα απαιτούσε κατανόηση των κοινωνικών, πολιτιστικών και ψυχολογικών φαινομένων πέρα από τις σωστές φιλοδοξίες της. Ταυτόχρονα αναγνώρισε ότι θα επηρεαζόντουσαν και από πολλούς τομείς εκτός της επιστήμης των υπολογιστών, συμπεριλαμβανομένης της «φιλοσοφίας, φαινομενολογία, την ανθρωπολογία, την ψυχολογία, τον μετά μοντερνισμό, κοινωνιολογία της επιστήμης και της φεμινιστικής κριτικής». Ήταν σαφής σχετικά με την αναφορά και την ειρωνική δυστοπική λογοτεχνία «την ανθρωπιστική προέλευση του ' αόρατου ιδανικού σε μεταμοντέρνα σκέψη'», στο μυθιστόρημα Ubik του Philip K. Dick.

Ο Andy Hopper από το Πανεπιστήμιο του Cambridge του Ηνωμένου Βασιλείου πρότεινε και επέδειξε την έννοια της «Teleporting» όπου οι εφαρμογές ακολουθούν τον χρήστη όπου αυτός / αυτή κινείται.

Ο ερευνητής και φοιτητής Roy Want εργαζόταν υπό την επίβλεψη του Andy Hopper στο Πανεπιστήμιο του Καίμπριτζ, και εργάστηκε πάνω στο «Active Badge System», το οποίο είναι ένα προηγμένο υπολογιστικό σύστημα τοποθεσίας, όπου η προσωπική κινητικότητα που έχει συγχωνευθεί με την πληροφορική.

Ο Bill Schilit (τώρα εργάζεται στην Google) έκανε επίσης κάποια προηγούμενη εργασία σε αυτό το θέμα, και το 1996 συμμετείχε στο εργαστήριο Mobile Computing που πραγματοποιήθηκε στη Σάντα Κρουζ.

Ο Δρ Ken Sakamura του Πανεπιστημίου του Τόκιο στην Ιαπωνία ηγείται του Εργαστηρίου πανταχού παρούσα Δικτύωσης (Ubiquitous Networking Laboratory - UNL), καθώς των φόρουμ του Τόκιο (Tokyo) και του T-Engine. Ο κοινός στόχος των προδιαγραφών του Ubiquitous Networking Sakamura και του φόρουμ T- Engine, είναι να επιτρέπουν σε κάθε καθημερινή συσκευή να μεταδίδουν και να λαμβάνουν πληροφορίες.

Επίσης το MIT (Massachusetts Institute of Technology) συνέβαλε σημαντική έρευνα σε αυτό τον τομέα, κυρίως σε Πράγματα που Σκέφτονται (Things That Think) σε κοινοπραξία (με κοινή διεύθυνση των Hiroshi Ishii, Joseph A. Paradiso και Rosalind Picard) στο εργαστήριο του MIT Media Lab και αυτή η προσπάθεια CSAIL είναι γνωστή με το έργο «Project Oxygen». Άλλοι σημαντικοί συνεισφέροντες είναι το Πανεπιστήμιο της Ουάσιγκτον Ubicomp Lab (υπό τη διεύθυνση του Shwetak Patel), το Τεχνολογικό κολλέγιο της Γεωργίας (Georgia Tech 's College of Computing), το προσωπικό του εργαστηρίου Aware Computing Lab του Πανεπιστημίου Cornell, το προσωπικό του Διαδραστικού Τηλεπικοινωνιακού Προγράμματος του Πανεπιστημίου της Νέας Υόρκης, το Τμήμα Πληροφορικής της Καλιφόρνιας Irvine, το ερευνητικό κέντρο της Microsoft, το ερευνητικό κέντρο της Intel και τα Πανεπιστήμια του Ισημερινού και της Νότιας Κορέας (Equator, Ajou University UCRi & CUS). [1]

Ένα από τα πρώτα παραδείγματα εφαρμογής τέτοιου συστήματος για την πανταχού παρούσα υπολογιστική, ήταν της καλλιτέχνιδας Natalie Jeremijenko το «Live Wire», επίσης γνωστή και ως «Dangling String», που είχε εγκατασταθεί στο Xerox PARC κατά τη χρονική διάρκεια που εργαζόταν και ο Mark Weiser εκεί.

Αυτό ήταν ένα κομμάτι από νήμα (string) που συνδεόταν με ένα βηματικό κινητήρα και ελεγχόταν από μια LAN σύνδεση. Η δραστηριότητα του δικτύου προκαλούσε συσπάσεις στο νήμα, αποδίδοντας περιφερειακά αξιοσημείωτη ένδειξη της κυκλοφορίας. Ο Weiser ονόματισε αυτό το παράδειγμα ως ηρεμία της τεχνολογίας. Οι περιβάλλουσες συσκευές δημιουργούν μια «σφαίρα», ένα «ταμπλό», και έναν «προγνώστη καιρού»: οι συγκεκριμένες διακοσμητικές συσκευές λαμβάνουν

δεδομένα από ένα ασύρματο δίκτυο και αναφέρουν τα τρέχοντα γεγονότα, όπως οι τιμές των μετοχών και τις καιρικές συνθήκες, όπως η Nabaztag που παράγεται από την Violet Snowden. Ο αυστραλιανός μελλοντολόγος επιστήμονας Mark Pesce έχει δημιουργήσει μια ιδιαίτερα διαμορφώσιμη ενεργοποιημένη λάμπα 52''-LED LAMP, η οποία χρησιμοποιεί το Wi-Fi που ονομάζεται MooresCloud μετά το Νόμο του Moore. Η Ενιαία Εταιρεία Υπολογιστών Ευφυΐας έχει ξεκινήσει μια συσκευή που ονομάζεται «Ubi - πανταχού παρούσα υπολογιστική» που σχεδιάστηκε για να επιτρέπει την αλληλεπίδραση φωνής με το σπίτι και παρέχει συνεχή πρόσβαση σε πληροφορίες. [1]

Η Πανταχού παρούσα Πληροφορική στη Μάθηση

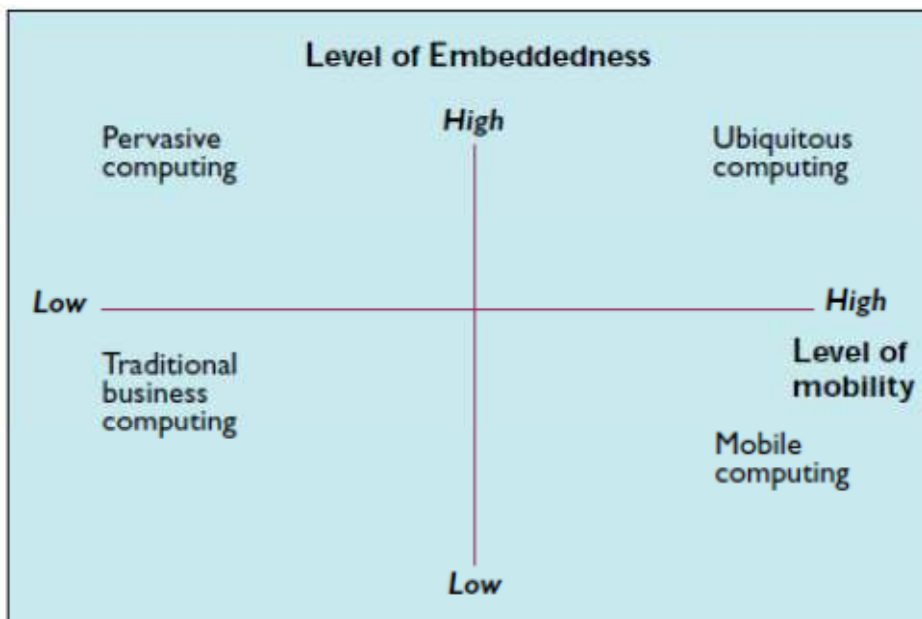
Η τεχνολογία Πανταχού παρούσας μάθησης U-Learning είναι ένα νέο παράδειγμα μάθησης και εφαρμόζεται σε μαθησιακά περιβάλλοντα. Είναι η επέκταση από την ηλεκτρονική μάθηση (e-learning) στην κινητή μάθηση (m-learning) και πλέον γίνεται στροφή προς u-learning. Ο ρόλος της είναι να κατασκευάσει ένα μαθησιακό περιβάλλον, το οποίο επιτρέπει στον κάθε μαθητή να μάθει οπουδήποτε και οποιαδήποτε στιγμή. Παρ' όλα αυτά, ο ορισμός και τα χαρακτηριστικά του U-Learning δεν είναι ακόμα σαφή και απασχολεί ιδιαίτερα τον ερευνητικό τομέα.

Ένα πρόσωπο εκπαιδεύεται όταν είναι στη διαδικασία απόκτησης γνώσεων ή δεξιοτήτων. Η γνώση αποκτάται μέσω της αλληλεπίδρασης μεταξύ του ατόμου και του περιβάλλοντος. Όμως πολλοί ερευνητές και μαθητές πιστεύουν ότι ο καλύτερος τρόπος εκμάθησης είναι η πράξη [13]. Η μάθηση στην πράξη διδάσκεται έμμεσα δηλαδή με βιωματικό τρόπο, με τον κατάλληλο τρόπο και την κατάλληλη στιγμή, ώστε οι μαθητές να είναι σε ένα ωφέλιμο περιβάλλον όπου θα εξυπηρετούνται τα συμφέροντα τους. Ακόμη και αυτός ο τρόπος χρειάζεται μια κατάλληλη μεθοδολογία προκειμένου να ληφθούν οι πληροφορίες από τις πραγματικές καταστάσεις του περιβάλλοντος. Με την εξέλιξη και την ανάπτυξη της τεχνολογίας η διαδικασία της μάθησης από το περιβάλλον γίνεται πιο εύκολη. Αυτό επιτυγχάνεται με την συνεχόμενη επικοινωνία και ανταλλαγή πληροφοριών. Για παράδειγμα, ένας

φοιτητής έχει μια κινητή συσκευή η οποία μπορεί να συνδεθεί με άλλες συσκευές, ενώ ταυτόχρονα έχει πρόσβαση στο δίκτυο μέσω των ασύρματων τεχνολογιών επικοινωνίας [14].

Σύμφωνα με τους Lygتينen & Yoo (2002) u-learning ορίζεται ως «η εξέλιξη της πανταχού παρούσα υπολογιστικής έχει επιταχυνθεί μέσω της βελτίωσης των ασύρματων δυνατοτήτων των τηλεπικοινωνιών, χάρη στα ανοιχτά δίκτυα, τις συνεχόμενες αυξήσεις της υπολογιστικής ισχύος, τη βελτιωμένη τεχνολογία μπαταριών και την εμφάνιση των ευέλικτων αρχιτεκτονικών λογισμικού, συνεπώς επιτρέποντας στην τεχνολογία να ενσωματώνεται στα μαθησιακά περιβάλλοντα για καθημερινές δραστηριότητες». Ο Hwang (2008) αναφέρει ότι δεν υπάρχει σαφής ορισμός της u-learning λόγω ραγδαίων αλλαγών στα μαθησιακά περιβάλλοντα.

Σύμφωνα με τους Ogata και Yano (2004) στην Εικόνα 9 παρουσιάζεται η ταξινόμηση των περιβάλλοντα μάθησης, βάσει των τεσσάρων διαστάσεων της πανταχού παρούσας υπολογιστικής. Οι επιτραπέζιοι υπολογιστές εκμάθησης παρέχουν χαμηλή κινητικότητα και χαμηλή ενσωμάτωση και το μαθησιακό περιβάλλον παραμένει σταθερό. Η εκμάθηση μέσω φορητών συσκευών αυξάνει την ικανότητα των μαθητών να μάθουν οποιαδήποτε στιγμή και οπουδήποτε. Στα Διάχυτα περιβάλλοντα ο μαθητής παίρνει πληροφορίες μέσω της επικοινωνίας των ενσωματωμένων συσκευών και του περιβάλλοντος. Έτσι όμως η μάθηση περιορίζεται καθώς γίνεται τοπικά. Αυτός ο περιορισμός ξεπερνιέται από την Πανταχού παρούσα μάθηση μέσω της υψηλής κινητικότητας (mobiling computing) στο μαθησιακό περιβάλλον. Οπότε οι συσκευές που είναι ενσωματωμένες στο περιβάλλον επιτρέπουν στους μαθητές ενώ κινούνται και ενώ είναι συνδεδεμένοι στο μαθησιακό τους περιβάλλον τους να μαθαίνουν. Σε αυτή την τεχνολογία το επίπεδο της ενσωμάτωσης και της κινητικότητας των συσκευών είναι αρκετά υψηλά και έχουν σημαντικά αποτελέσματα στην εκμάθηση του περιβάλλοντος.



Εικόνα 10 : Ταξινόμηση των μαθησιακών περιβαλλόντων (Ogata and Yano, 2004)

Έτσι πρώτοι οι Ogata και Yano (2004) τον ορισμό u-learning βάσει της παραπάνω ταξινόμησης. Αργότερα ο Dey Casey (2005) υποστήριξε τον ορισμό και τον επαναδιατύπωσε ως εξής «u-learning = e-learning + m-learning».

Επειδή επικρατεί μια σύγχυση μεταξύ των ορισμών του U-learning που έχουν επικρατήσει είναι «οπουδήποτε και οποτεδήποτε μάθηση» και «μάθηση με πανταχού παρούσα τεχνολογία των υπολογιστών» γενικά προτείνεται ο ακόλουθος: «U-learning είναι ένα παράδειγμα μάθησης που λαμβάνει χώρα σε ένα πανταχού παρών υπολογιστικό περιβάλλον που επιτρέπει την μάθηση για το σωστό πράγμα στο σωστό τόπο και χρόνο με το σωστό τρόπο».

Όπως αναφέρθηκε και σε προηγούμενη ενότητα, σύμφωνα με τον Mark Weiser (1991), η πανταχού παρούσα τεχνολογία είναι η μέθοδος για την αύξηση της χρήση των ηλεκτρονικών υπολογιστών ενώ ταυτόχρονα την καθιστά διαθέσιμη σε όλο το φυσικό περιβάλλον. Οι υπολογιστές θα είναι τόσο πολύ ενσωματωμένοι με το περιβάλλον που θα ουσιαστικά αόρατοι για τον χρήστη. Αντίστοιχα ο Zhang (2005) ορίζει το πανταχού παρών υπολογιστικό περιβάλλον ως «πολύ καλά προσδιορισμένη περιοχή που περιλαμβάνει μια συλλογή ενσωματωμένων συστημάτων, όπως υπολογιστές, αισθητήρες, διεπαφές χρηστών και υπηρεσιών, η

οποία ενισχύεται από την εξέλιξη της πληροφορικής και της τεχνολογίας επικοινωνιών». Ο κύριος σκοπός της τεχνολογίας για την U-learning είναι να βοηθήσει τους μαθητές να πάρουν ακριβείς πληροφορίες που ψάχνουν για αυτή, τη σωστή στιγμή, το σωστό χρόνο και με τον σωστό τρόπο . [15]

Τα χαρακτηριστικά του U-Learning

Σύμφωνα με τα παραπάνω γίνεται αντιληπτό το κυριότερο χαρακτηριστικό και στόχος του u-learning: η παροχή των σωστών πληροφοριών στο σωστό τόπο και χρόνο για την υποστήριξη των καθημερινών δραστηριοτήτων των μαθητών. Επειδή δεν έχουν αποσαφηνιστεί όλα τα χαρακτηριστικά και τα κριτήρια υποδομής της u-learning [16]

Οι Chen et al [17] προσδιορίζουν έξι βασικά χαρακτηριστικά της m-learning και στη συνέχεια διάφοροι ερευνητές τα προσάρμοσαν στην u-learning. Τα χαρακτηριστικά είναι: *η ανάγκη για την γρήγορη μάθηση, η πρωτοβουλία της απόκτησης γνώσεων, η κινητικότητα, διαδραστικότητα, η τοποθέτηση από την πλευρά της εκπαιδευτικής δραστηριότητας, και την ενσωμάτωση του εκπαιδευτικού περιεχόμενο.*

Η συγκέντρωση αυτών των χαρακτηριστικών έγινε σταδιακά. Πρώτα από τους Chen et al. (2002) και Curtis et al. (2002) [17], που προσδιόρισαν την μονιμότητα, την προσβασιμότητα, την αμεσότητα, διαδραστικότητα και την τοποθέτηση των διδακτικών δραστηριοτήτων. Στη συνέχεια οι ερευνητές Ogata (2004), Ogata & Yano (2004) και Chiu (2008) προσάρμοσαν τα χαρακτηριστικά: *της μονιμότητας, της προσβασιμότητας και της αμεσότητας.* Οι Hiroaki Ogata και Yoneo Yano (2004) επέκτειναν τα χαρακτηριστικά σχετικά με την κινητικότητα των μαθητών στο ενσωματωμένο περιβάλλοντος και πρόσθεσαν δύο ακόμη: η διαδραστικότητα και το τοποθετήσουμε των διδακτικών δραστηριοτήτων. Ο Hwang et al. (2008) εφάρμοσε τον όρο «contextaware u-learning» όπου σύμφωνα με αυτό τον όρο τα χαρακτηριστικά του u-learning περιλαμβάνουν αδιάλειπτη παροχή υπηρεσιών, context-aware των υπηρεσιών και των προσαρμοσμένων υπηρεσιών. Τελικά, ο Chiu et al. (2008) εξέτασε βαθύτερα τη χρήση του context-aware και του ubiquitous computing στο μαθησιακό περιβάλλον, καταφέροντας να ενθαρρύνει το κίνητρο

των μαθητών για μάθηση, αλλά και να αυξήσει τις επιδόσεις τους. Οπότε τα χαρακτηριστικά του u-learning περιγράφονται ως: η ανάγκη για την γρήγορη μάθηση, η πρωτοβουλία της απόκτησης γνώσεων, διαδραστικότητα στη μαθησιακή διαδικασία, η κατάσταση της εκπαιδευτικής δραστηριότητας, η επίγνωση του περιβάλλοντος (*context awareness*), η παροχή εξατομικευμένων υπηρεσιών, αυτορυθμιζόμενη μάθηση, αδιάλειπτη μάθηση, προσαρμογή του περιεχομένου, και κοινωνικοποίηση της μάθησης.

Συνοψίζοντας τα παραπάνω χαρακτηριστικά του U-Learning παρατηρούνται σημαντικές επικαλύψεις μεταξύ των χαρακτηριστικών τους, οι οποίες απεικονίζονται στον Πίνακα 1.

Chen et al., 2002	Curtis et al., 2002	Ogata, 2004	Hwang, 2008	Chiu et al., 2008	Proposed characteristics
Urgency of learning needs	Permanency	Permanency	Seamless services	Urgency of learning need	Permanency
Initiative of knowledge acquisition	Accessibility	Accessibility	Context-awareness	Initiative of knowledge acquisition	Accessibility
Mobility of learning setting	Immediacy	Immediacy	Adaptive services	Interactivity of learning process	Immediacy
Interactivity of learning process		Interactivity		Situation of instructional activity	Interactivity
Situating of instructional activity		Situating of instructional activities		Context-awareness	Context-awareness
Integration of instructional content				Actively provides personalized services	
				Self-regulated learning	
				Seamless learning	
				Adapt the subject contents	
				Learning community	

Πίνακας 1 : Σύγκριση των χαρακτηριστικών του U-Learning

Από αυτά διατηρούμε πέντε χαρακτηριστικά:

- Μονιμότητα: Οι πληροφορίες που παραμένουν, εκτός και αν αφαιρεθούν σκόπιμα από τους εκπαιδευόμενους.
- Προσβασιμότητα: Οι πληροφορίες είναι πάντα διαθέσιμες όταν οι μαθητές επιθυμούν να τις χρησιμοποιήσουν.
- Αμεσότητα: Οι πληροφορίες μπορούν να ανακτηθούν άμεσα από τους εκπαιδευόμενους.
- Διαδραστικότητα: Οι μαθητές αλληλεπιδρούν με τους συμμαθητές και τους δασκάλους αποτελεσματικά με διάφορα μέσα ενημέρωσης.
- Context-awareness: Το περιβάλλον προσαρμόζεται στις πραγματικές καταστάσεις και ανάγκες των μαθητών, ώστε να τους παρέχει επαρκείς πληροφορίες.

Το δύσκολο σε τέτοια περιβάλλοντα είναι ο καθορισμός της επίγνωσης πλαισίου, δηλαδή «η ικανότητα ενός προγράμματος ή μια συσκευής για την ανίχνευση διάφορων καταστάσεων του περιβάλλοντος ή της ίδιας της». Σύμφωνα με τους Dey & Abowd (2000) είναι δύσκολο να χαρακτηριστεί η κατάσταση μιας οντότητας μέσα στο περιβάλλον βάσει θέσης, ταυτότητας, χρόνου, και περιβάλλον. Πιο συγκεκριμένα σε ένα ubiquitous μαθησιακό περιβάλλον, είναι δύσκολο να αναγνωριστεί αν ένας μαθητής διαθέτει την ίδια γνώση με έναν άλλον ακόμα κι αν είναι στην ίδια θέση. Βάσει των χαρακτηριστικών του Πίνακα 1, η προσαρμοστικότητα, η τοποθέτηση των εκπαιδευτικών δραστηριοτήτων και η αδιάλειπτη μάθηση αποτελούν μέρος της γνώσης προκειμένου να επιτευχθεί ο στόχος του U-Learning.

Στον Πίνακα 2 παρουσιάζεται η σύγκριση των τριών μεγάλων παραδειγμάτων μάθησης.

Criteria	u-learning	m-learning	e-learning
Concept	Learn the right thing at the right place and time in the right way.	Learn at the right place and time.	Learn at the right time.
Permanency	Learners can never lose their work.	Learners may lose their work. Changes in learning devices or learning in moving will interrupt learning activities.	Learners can lose their work.
Accessibility	System access via ubiquitous computing technologies.	System access via wireless networks.	System access via computer network
Immediacy	Learners get information immediately.	Learners get information immediately in fixed environments with specified mobile learning devices.	Learners cannot get information immediately.
Interactivity	Learners' interaction with peers, teachers, and experts effectively through the interfaces of u-learning systems.	Learners can interact with peers, teachers, and experts in specified learning environment.	Learners' interaction is limited.
Context-awareness	The system can understand the learner's environment via database and sensing the learner's location, personal and environmental situations.	The system understands the learner's situation by accessing the database.	The system cannot sense the learner's environment.

Πίνακας 2 : Σύγκριση των Μαθησιακά Υποδείγματα

Αυτοί οι ορισμοί και τα χαρακτηριστικά βοηθάνε τους ερευνητές να κατανοήσουν την έννοια της u-learning και τους προγραμματιστές να σχεδιάσουν και να αναπτύξουν εφαρμογές u-learning. [17]

3. Αισθητήρες και εφαρμογές τους



Οι αισθητήρες (sensors) τα τελευταία είκοσι χρόνια έχουν καθιερωθεί για να δίνουν πολύτιμες και εναλλακτικές λύσεις σε διαδικασίες όπως παρακολούθηση και έλεγχο συστημάτων και παρατήρηση κρίσιμων αλλαγών. Σε αυτή την ενότητα παρουσιάζονται οι αισθητήρες, τα χαρακτηριστικά τους και οι εφαρμογές τους σε διάφορους τομείς όπως τη βιομηχανία (Soft Sensors), σε έξυπνα περιβάλλοντα, στην φορητή υπολογιστική και την υγεία.

[κενή σελίδα]

Αισθητήρες

Ένας αισθητήρας είναι μία συσκευή που ανιχνεύει ένα φυσικό μέγεθος και παράγει από αυτό μία μετρήσιμη έξοδο. Για παράδειγμα, το υδραργυρικό θερμομέτρο μετατρέπει τη μετρούμενη θερμοκρασία σε διαστολή, η οποία μπορεί να αναγνωστεί από ένα βαθμονομημένο σωλήνα. Οι αισθητήρες χρησιμοποιούνται καθημερινά σε διάφορα αντικείμενα, όπως κουμπιά ανελκυστήρων που είναι ευαίσθητα στην αφή και λάμπες φωτισμού που εκπέμπουν πιο έντονα ή απαλότερα αγγίζοντας τη βάση τους. Επίσης, εφαρμογές τους συναντάμε στα αυτοκίνητα, σε μηχανές, στην αεροναυπηγική, την ιατρική, τη βιομηχανία και τη ρομποτική. [18]

Πιο συγκεκριμένα ένας αισθητήρας είναι μια συσκευή, που μετρά μια φυσική ποσότητα (είσοδος) και στη συνέχεια την μετατρέπει σε ένα σήμα το οποίο μπορεί να διαβαστεί (έξοδος) από έναν παρατηρητή ή από κάποιο ηλεκτρονικό μέσο, δηλαδή με τη μορφή ενός ηλεκτρικού ή οπτικού σήματος. Για παράδειγμα, ένα θερμομέτρο υδραργύρου μετατρέπει τη μετρούμενη θερμοκρασία σε διαστολή και συστολή ενός υγρού το οποίο μπορεί να διαβαστεί σε ένα βαθμονομημένο γυάλινο σωλήνα. Ένα θερμοζεύγος μετατρέπει τη θερμοκρασία σε μια τάση εξόδου η οποία μπορεί να διαβαστεί από ένα βολτόμετρο. Για την ακρίβεια, οι περισσότεροι αισθητήρες έχουν βαθμονομηθεί βάσει γνωστών προτύπων. [19]

Κατηγορίες Αισθητήρων

Οι βασικότερες κατηγορίες αισθητήρων είναι:

- Οι θερμικοί, πχ θερμομέτρα και θερμοστάτες.
- Οι ηλεκτρομαγνητικοί, όπως τα ομόμετρα, τα γαλβανόμετρα, τα βολτόμετρα, , οι ανιχνευτές μετάλλων κλπ.
- Οι μηχανικοί, πχ οι αισθητήρες πίεσης όπως το βαρόμετρο και το αλτίμετρο, οι αισθητήρες ροής (ροόμετρα), το ανεμόμετρο και το υδρόμετρο.
- Οι χημικοί που αισθητήρες ανιχνεύουν χημικές ενώσεις.

Εφαρμογές ευφρών συστημάτων

- Οι αισθητήρες φωτός και ακτινοβολίας, πχ φωτοκύτταρα, ανιχνευτές υπέρυθρων ακτινών, ο μετρητής Geiger, κλπ.
- Οι ακουστικοί, εδώ ανήκουν τα μικρόφωνα, τα υδρόφωνα και οι σειсмоγράφοι.
- Άλλα είδη αισθητήρων είναι τα ταχύμετρα και οι αισθητήρες απόστασης.

Στον ακόλουθο πίνακα αναφέρονται τα χαρακτηριστικά αισθητήρων:

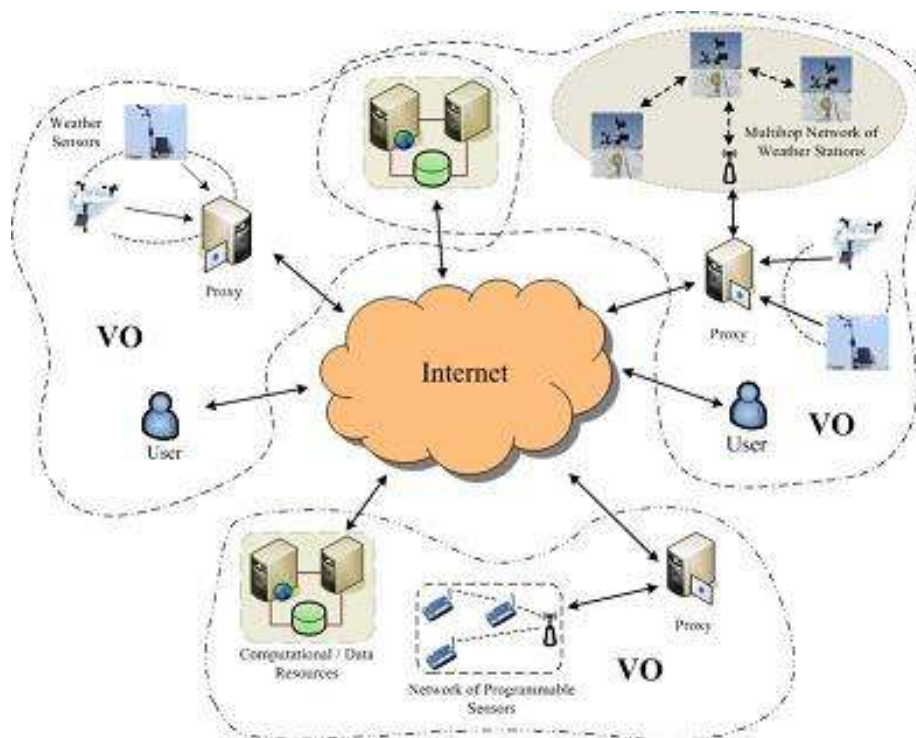
Εύρος	Τα όρια στα οποία η συσκευή λειτουργεί αξιόπιστα.
Ακρίβεια	Η εγγύτητα της τιμής εξόδου προς τη τιμή εισόδου.
Σφάλμα	Η διαφορά ανάμεσα στη μετρούμενη τιμή και τη πραγματική τιμή.
Ανοχή	Το μέγιστο σφάλμα που μπορεί να δημιουργήσει ο αισθητήρας.
Διακριτική ικανότητα	Η μικρότερη αλλαγή τιμής εισόδου που μπορεί να ανιχνεύσει.
Ευαισθησία	Η σχέση της αλλαγής εξόδου προς τη αλλαγή εισόδου, είναι ίση με τη διαφορά των τιμών της εξόδου προς τη διαφορά των αντίστοιχων τιμών εισόδου.
Βαθμονόμηση	Η βαθμολόγηση της κλίμακας σε μονάδες.
Νεκρή ζώνη	Το μέγιστο ποσό αλλαγής της εισόδου που δεν επιφέρει αλλαγή στην έξοδο.

Γραμμικότητα	Ο βαθμός στον οποίο η γραφική παράσταση της εξόδου προσεγγίζει ευθεία ως προς την είσοδο του αισθητήρα.
Απόκριση	Ο χρόνος που απαιτείται για να λάβει τη τελική τιμή η έξοδος.
Καθυστέρηση	Η καθυστέρηση της αλλαγής της εξόδου ως προς την είσοδο.
Ευστάθεια	Η μεταβολή της εξόδου σε μεγάλη χρονική περίοδο, χωρίς μεταβολή της εισόδου και των συνθηκών.
Υστέρηση	Η διαφορά στην έξοδο όταν η κατεύθυνση της μεταβολής της εισόδου αντιστραφεί.
Επαναληψιμότητα	Η παραγωγή του ίδιου αποτελέσματος, σε διαφορετικές χρονικές στιγμές, με την ίδια είσοδο.
Ολίσθηση	Η μεταβολή των χαρακτηριστικών του αισθητήρα με το χρόνο και το περιβάλλον.
Στατικό σφάλμα	Σταθερό σφάλμα σε όλο το εύρος λειτουργίας, το οποίο μπορεί να αντισταθμιστεί.
Χρόνος λειτουργίας	Ο εκτιμώμενος χρόνος λειτουργίας στα πλαίσια των προδιαγραφών του.

Πίνακας 3 : Χαρακτηριστικά αισθητήρων

Πλέγμα (δίκτυο) Αισθητήρων

Ένα πλέγμα αισθητήρων ενσωματώνει με ασύρματη τεχνολογία τα δίκτυα αισθητήρων με ένα δίκτυο ηλεκτρονικών υπολογιστών για να καταστεί δυνατή η συλλογή δεδομένων αισθητήρων σε πραγματικό χρόνο και την ανταλλαγή των υπολογιστικών και αποθηκευτικών πόρων για την επεξεργασία των δεδομένων του αισθητήρα και την διαχείριση. Το δίκτυο αισθητήρων είναι μια τεχνολογία που επιτρέπει την ανάπτυξη υποδομών μεγάλης κλίμακας, την ενοποίηση ετερογενών αισθητήρων, δεδομένων και υπολογιστικών πόρων, που διατίθενται σε μια ευρεία περιοχή, και έχει αναλάβει πολύπλοκα καθήκοντα επιτήρησης, όπως η παρακολούθηση του περιβάλλοντος.



Εικόνα 11 : Χαρακτηριστική αρχιτεκτονική δικτύου αισθητήρων

Το δίκτυο αισθητήρων ορίστηκε για πρώτη φορά από το έργο Discovery Net (pilot project funded by the UK e-Science Programme, 2001–2005) όπου γίνεται διάκριση μεταξύ των όρων «δίκτυα αισθητήρων» και «πλέγμα αισθητήρων». Ο σχεδιασμός ενός δικτύου αισθητήρων αντιμετωπίζει τη λογική και φυσική σύνδεση των αισθητήρων. Ενώ το πλέγμα αισθητήρων αφορά ζητήματα που σχετίζονται με τη

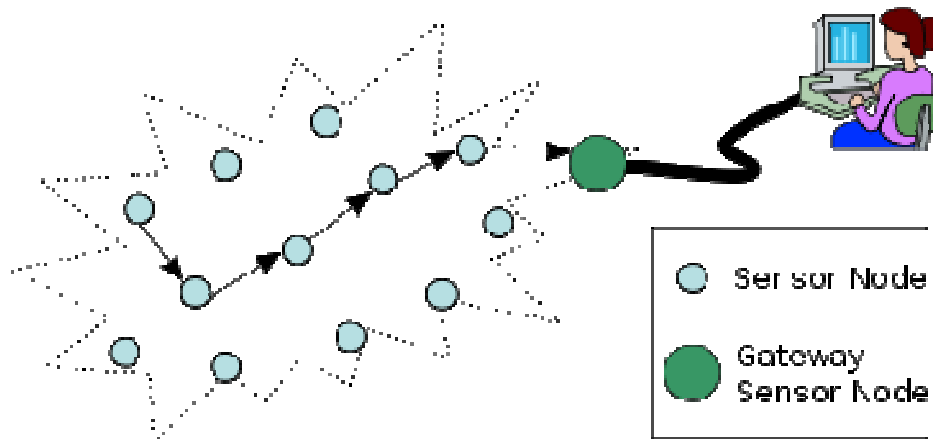
διαχείριση δεδομένων, διαχείριση υπολογισμού, διαχείρισης πληροφοριών και διαχείρισης της γνώσης που σχετίζονται με την ανακάλυψη των αισθητήρων και πώς μπορούν να αντιμετωπιστούν μέσα σε ένα ανοικτό υπολογιστικό περιβάλλον.

Το πλέγμα αισθητήρων επιτρέπει τη συλλογή, την επεξεργασία, τη διανομή, την απεικόνιση, την αρχειοθέτηση και αναζήτηση των μεγάλων ποσοτήτων των δεδομένων του αισθητήρα. Κάθε χρήστης μπορεί να αποκτήσει πρόσβαση σε ένα υποσύνολο των αισθητήρων κατά τη διάρκεια μιας συγκεκριμένης χρονικής περιόδου για να εκτελέσετε μια συγκεκριμένη εφαρμογή, και να εισπράξει το επιθυμητό τύπο των δεδομένων των αισθητήρων. Οι συσκευές αισθητήρων με ενσωματωμένους επεξεργαστές γίνονται πολύ ισχυρά υπολογιστικά εργαλεία. Ένα πλέγμα αισθητήρων παρέχει πρόσβαση σε μια ευρεία ποικιλία πόρων με διάχυτη τρόπο. Εφαρμόζοντας προηγμένες τεχνικές όπως τεχνητή νοημοσύνη, συγχώνευση δεδομένων, εξόρυξη δεδομένων και κατανομημένη επεξεργασία δεδομένων, μπορούμε να βγάλουμε νόημα από τα δεδομένα των αισθητήρων και να δημιουργηθεί νέα γνώση του περιβάλλοντος. Τα αποτελέσματα μπορούν στη συνέχεια να χρησιμοποιηθούν για τη βελτιστοποίηση της λειτουργίας των αισθητήρων, ή να επηρεάσουν την λειτουργία των ενεργοποιητών και να αλλάξει το περιβάλλον. Έτσι, τα δίκτυα αισθητήρων είναι κατάλληλα για pervasive εφαρμογές. Για παράδειγμα η παρακολούθηση της υγείας των ασθενών, η παρακολούθηση και πρόβλεψη του καιρού, των στρατιωτικών και της εσωτερικής ασφάλειας, η παρακολούθηση των προϊόντων και των διαδικασιών παραγωγής, η παρακολούθηση της ασφάλειας των φυσικών δομών και εργοταξίων, έξυπνων σπιτιών και γραφείων. Μπορούν να χρησιμοποιηθούν διάφορες αρχιτεκτονικές για τέτοιες εφαρμογές, καθώς και διάφορα είδη ανάλυσης και εξόρυξης δεδομένων. [20]

Ασύρματα δίκτυα αισθητήρων

Ένα ασύρματο δίκτυο αισθητήρων (Wireless Sensor Network - WSN) αποτελείται από διασκορπισμένους αυτόνομους αισθητήρες για την παρακολούθηση φυσικών ή

περιβαλλοντολογικών συνθηκών, όπως η θερμοκρασία, ο ήχος, η ατμοσφαιρική πίεση, και μεταφέρει τα δεδομένα μέσω του δικτύου σε μια συγκεκριμένη τοποθεσία. Τα πιο μοντέρνα δίκτυα είναι ικανά να δίνουν και να δέχονται και άλλους τύπους πληροφοριών, πράγμα που τους επιτρέπει να ελέγχουν την δραστηριότητα των αισθητήρων. Το κίνητρο για την ανάπτυξη των ασύρματων δικτύων με αισθητήρες ήταν οι στρατιωτικές εφαρμογές, όπως η παρακολούθηση των πεδίων μάχης. Σήμερα τέτοια δίκτυα χρησιμοποιούνται σε πολλές καταναλωτικές και βιομηχανικές εφαρμογές, για την παρακολούθηση και έλεγχο της βιομηχανικής παραγωγής, την παρακολούθηση των μηχανημάτων υγείας και πολλά άλλα.



Εικόνα 12 : Τυπικό παράδειγμα αρχιτεκτονικής ασύρματου δικτύου αισθητήρων multi-hop

Το ασύρματο δίκτυο αισθητήρων αποτελείται από κόμβους, είτε από λίγους, είτε εκατοντάδες ή ακόμα και χιλιάδες, όπου κάθε κόμβος συνδέεται σε έναν ή και πολλούς αισθητήρες. Κάθε τέτοιος κόμβος του δικτύου αισθητήρων έχει χαρακτηριστικά εξαρτήματα:

- ένα ραδιοπομποδέκτη με μια εσωτερική κεραία ή μια σύνδεση με μια εξωτερική κεραία,
- έναν μικροελεγκτή,
- ένα ηλεκτρονικό κύκλωμα για την διασύνδεση με τους αισθητήρες και
- μια πηγή ενέργειας, συνήθως μια μπαταρία η μια ενσωματωμένη μορφή συγκομιδής ενέργειας.

Τα μεγέθη ενός αισθητήριου κόμβου ποικίλουν, από το μέγεθος ενός κουτιού παπουτσιών μέχρι το μέγεθος ενός κόκκου σκόνης, αν και τέτοιου είδους τόσο μικροσκοπικών διαστάσεων δεν έχουν ακόμα δημιουργηθεί. Το κόστος των αισθητήριων κόμβων ποικίλει, ξεκινώντας από μερικά και φτάνοντας σε εκατοντάδες δολάρια, αναλόγως της πολυπλοκότητας των μεμονωμένων αισθητήριων κόμβων. Ανάλογα με το κόστος και το μέγεθος υπάρχουν περιορισμοί σε πόρους όπως ενέργεια, μνήμη, υπολογιστική ταχύτητα και στο εύρος ζώνης των επικοινωνιών. Υπάρχουν διάφορες τοπολογίες αισθητήρων, από τοπολογία αστέρα μέχρι αναπτυγμένο ασύρματο δίκτυο πλέγματος multi-hop.

Στην επιστήμη των υπολογιστών και των τηλεπικοινωνιών, τα ασύρματα δίκτυα αισθητήρων είναι ένας ενεργός τομέας έρευνας με πολυάριθμα εργαστήρια και συνέδρια που διοργανώνονται κάθε χρόνο. [20] [21]

Ιστορική Αναδρομή

Τα δίκτυα αισθητήρων προέκυψαν από ταυτόχρονες έρευνες που έγιναν στους τομείς της τεχνολογίας αισθητήρων, τηλεπικοινωνιών και μικροϋπολογιστικής. Όπως ήδη αναφέρθηκε η έρευνα ξεκίνησε από στρατιωτικές/αμυντικές εφαρμογές. Οι πρώτοι αισθητήρες σε αυτά τα δίκτυα λειτουργούσαν με ιεραρχική δομή και ο άνθρωπος είχε σημαντικό ρόλο στην λειτουργία τους. Κατά τη διάρκεια του ψυχρού πολέμου αναπτύχθηκε το Sound Surveillance System (SOSUS). Το συγκεκριμένο σύστημα διαθέτει ανιχνευτές ήχου που ήταν διασκορπισμένοι στον Ατλαντικό και Ειρηνικό Ωκεανό, για τον εντοπισμό υποβρυχίων. Αυτό χρησιμοποιείται ακόμα και σήμερα από την National Oceanographic and Atmospheric Administration (NOAA) για παρακολούθηση ωκεάνιων οικοσυστημάτων, αλλά και για εντοπισμό σεισμικής δραστηριότητας στον πυθμένα της θάλασσας. Επίσης την ίδια χρονική περίοδο αναπτύχθηκαν εναέρια δίκτυα παρακολούθησης για αμυντικούς σκοπούς που

χρησιμοποιήθηκαν στα αεροπλάνα (Airborne Warning and Control System - AWACS).

Το 1980 ξεκίνησε μια έρευνα στο δίκτυο αισθητήρων για τον εκσυγχρονισμό τους μέσω του προγράμματος Distributed Sensor Networks (DSN) από τη Defense Advanced Research Projects Agency (DARPA). Το arpanet (ο πρόγονος του internet) έδωσε την ιδέα για επέκταση των πρωτοκόλλων του και την εφαρμογή τους στην επικοινωνία στα δίκτυα αισθητήρων. Το 1978 έγινε μια συνολική καταγραφή των δομικών στοιχείων ενός DSN, όπου περιλαμβάνονταν οι αισθητήρες, τα πρωτόκολλα επικοινωνίας, αλγόριθμοι και τεχνικές επιλογής θέσης κόμβων και επεξεργασμένη πληροφορία που έχει συγκεντρωθεί, και κατανεμημένο λογισμικό. Επίσης ξεκίνησε η έρευνα για εφαρμογές τεχνητής νοημοσύνης, επειδή η διαθέσιμη τεχνολογία ήταν ελλιπής, η έρευνα περιορίστηκε σε τεχνικές επίλυσης κατανεμημένων προβλημάτων, υπολογιστικής υποστήριξης και επεξεργασίας σημάτων.

Στη συνέχεια ερευνητική ομάδα από το πανεπιστήμιο Carnegie Mellon (CMU) των ΗΠΑ, ανέπτυξε ένα λειτουργικό δίκτυο, το Accent και ήταν το πρώτο που υποστήριζε transparent δικτύωση και rebinding. Το Accent ήταν το πρώτο σύστημα που λειτούργησε με υπολογιστές συνδεδεμένους σε Ethernet. Ταυτόχρονα η ερευνητική ομάδα στο Massachusetts Institute of Technology (MIT), ανέπτυξε τεχνικές επεξεργασίας ηχητικών σημάτων για εντοπισμό ελικοπτέρων.

Το 1995 σχεδιάστηκε το σύστημα Cooperative Engagement Capability (CEC) για το Αμερικανικό Ναυτικό. Το σύστημα αποτελούνταν πάρα πολλά ραντάρ που συγκέντρωναν δεδομένα εναέριων στόχων, ενώ ταυτόχρονα μπορούσαν να ανταλλάσσουν πληροφορίες μεταξύ τους, μέσω ενός κεντρικού κόμβου και να δίνουν μια συνολική εικόνα για μια περιοχή.

Το 1996, το πρόγραμμα LWIMs (Low Power Wireless Integrated Microsensors), επιδοτήθηκε από τον κρατικό οργανισμό DARPA (Defence Advanced Research Projects Agency), που κατάφερε με την χρήση CMOS κατασκευών, να ενσωματώσει

πολλαπλούς αισθητήρες, ηλεκτρονικές διεπαφές, μονάδες ελέγχου και ασύρματης επικοινωνίας σε μία μόνο συσκευή. Στη συνέχεια το 1998, η ίδια ερευνητική ομάδα κατασκεύασε έναν αισθητήρα τον WINS (Wireless Integrated Network Sensors), που μπορεί να ενσωματώσει διάφορους αισθητήρες, και τους επιτρέπεται η σύνδεση μέσω Internet.

Επειδή οι WINS εξαιτίας του μεγέθους τους και της μεγάλης κατανάλωσης σε ενέργεια δεν μπορούσαν να χρησιμοποιηθούν σε πολλές εφαρμογές, το 1999 μέσω του έργου Smart Dust από το πανεπιστήμιο του Berkeley αναπτύχθηκε ο κόμβος WeC που περιείχε ένα μικροελεγκτή και κατανάλωνε χαμηλή ισχύ. [22]

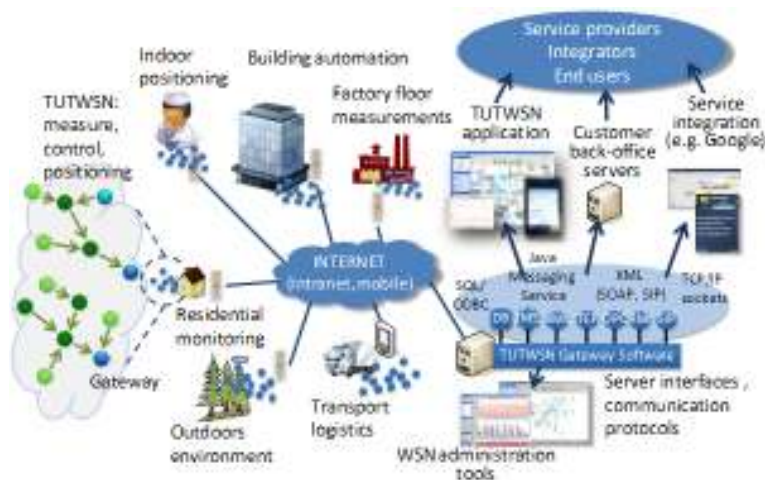
Στην παρακάτω εικόνα φαίνεται η εξέλιξη του μεγέθους των κόμβων με την πρόοδο:



Εικόνα 13 : Εξέλιξη κόμβων αισθητήρων

Εφαρμογές Αισθητήρων

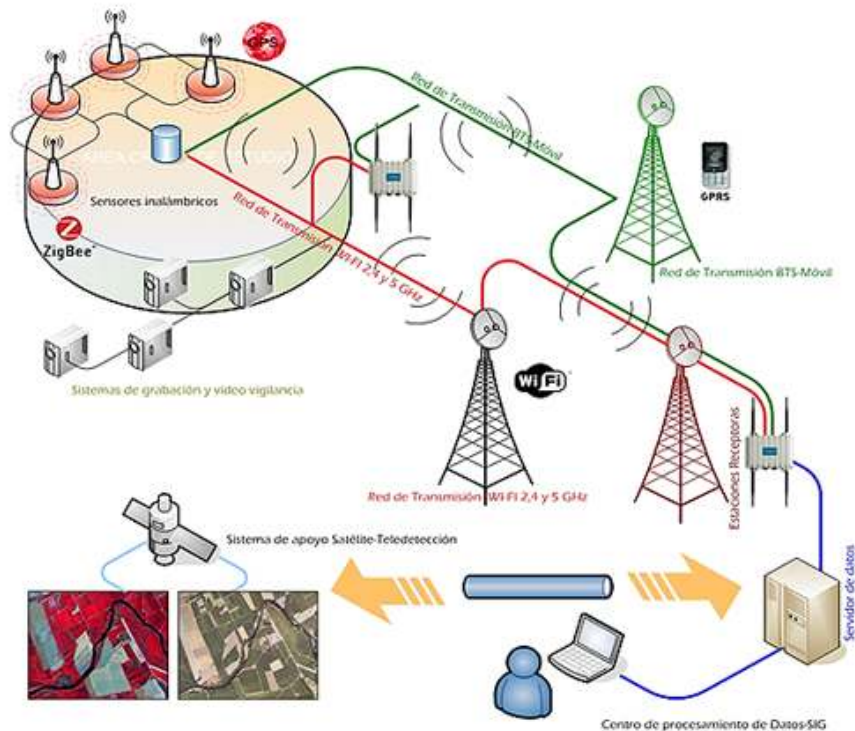
Όπως αναφέρθηκε και σε προηγούμενη ενότητα υπάρχουν αρκετά είδη αισθητήρων και σύμφωνα με αυτά και τα χαρακτηριστικά τους παρουσιάζονται στη συνέχεια διάφορες εφαρμογές τους σε ποικίλους τομείς.



Εικόνα 14 : Παράδειγμα εφαρμογών αισθητήρων του Τεχνολογικού Πανεπιστημίου Tampere της Φινλανδίας, του τμήματος Υπολογιστικών Συστημάτων

A) Παρακολούθηση περιοχής, το ασύρματο δίκτυο αισθητήρων έχει αναπτυχθεί σε μια περιοχή για να παρακολουθεί ένα φαινόμενο, πχ από τον στρατό για να ανιχνευθεί η εχθρική εισβολή.

B) Περιβαλλοντική/ γεωσκόπηση, παρακολούθηση της γεωλογίας και του περιβάλλοντος με αισθητήρες ηφαιστειών, ωκεανών, παγετώνων, δασών. Συγκεκριμένα Παρακολούθηση της ποιότητας του αέρα, όπως ο βαθμός ρύπανσης του αέρα. Ένα δίκτυο αισθητήριων κόμβων μπορεί να εγκατασταθεί σε ένα δάσος για Ανίχνευση δασικών πυρκαγιών. Οι κόμβοι μπορούν να είναι εξοπλισμένοι με αισθητήρες για τη μέτρηση της θερμοκρασίας, την υγρασία και τα αέρια που παράγονται από φωτιά στα δέντρα ή τη βλάστηση. Η έγκαιρη ανίχνευση είναι ζωτικής σημασίας για την επιτυχή δράση των πυροσβεστών, χάρη στα ασύρματα δίκτυα αισθητήρων, η πυροσβεστική θα είναι σε θέση να γνωρίζει πότε μια πυρκαγιά ξεκίνησε και πώς εξαπλώνεται.



Εικόνα 15 : Σύστημα αισθητήρων για έγκαιρη ανίχνευση πυρκαγιών

Ένα σύστημα Ανίχνευσης κατολισθήσεων, ανιχνεύει μικρές κινήσεις του εδάφους και αλλαγές στις διάφορες παραμέτρους που μπορεί να συμβούν πριν ή κατά τη διάρκεια μιας κατολίσθησης. Τα ασύρματα δίκτυα αισθητήρων μπορούν να ενεργήσουν αποτελεσματικά για την Πρόληψη φυσικών καταστροφών, ώστε να αποτρέψουν κακές συνέπειες φυσικών καταστροφών, όπως οι πλημμύρες. Η τοποθέτηση αισθητήρων σε ποτάμια παρακολουθούν σε πραγματικό χρόνο τις μεταβολές της στάθμης του νερού.

Γ) Βιομηχανική παρακολούθηση

Έχουν αναπτυχθεί ασύρματα δίκτυα αισθητήρων για την βασική συντήρηση μηχανημάτων (Condition-based maintenance - CBM), και προσφέρουν σημαντική εξοικονόμηση κόστους και επιτρέπουν νέες λειτουργίες. Σε αντίθεση με τα ενσύρματα συστήματα, όπου η εγκατάσταση τους συχνά περιορίζεται από το κόστος της καλωδίωσης. Επιπλέον απρόσιτες περιοχές, περιστρεφόμενα μηχανήματα, επικίνδυνες ή ζώνες περιορισμένης πρόσβασης, πλέον μπορούν να παρακολουθούνται με ασύρματους αισθητήρες.

Τα ασύρματα δίκτυα αισθητήρων χρησιμοποιούνται επίσης για τη συλλογή δεδομένων για την παρακολούθηση, δηλαδή την Καταγραφή δεδομένων των περιβαλλοντικών πληροφοριών. Αυτό μπορεί να είναι τόσο απλό όσο η παρακολούθηση της θερμοκρασίας σε ένα ψυγείο η περίπλοκο όσο η παρακολούθηση του επιπέδου του νερού σε δεξαμενές υπερχειλίσσης σε πυρηνικούς σταθμούς ηλεκτροπαραγωγής. Το πλεονέκτημα των ασύρματων δικτύων αισθητήρων έναντι των συμβατικών καταγραφών είναι η real time η μετάδοση των δεδομένων. Πλέον πρόσφατες έρευνες έχουν αρχίσει να εξετάζουν ένα ευρύτερο φάσμα θεμάτων, όπως η αξιοπιστία των ασύρματων συνδέσεων, τις δυνατότητες σε πραγματικό χρόνο, ή την ποιότητα της παρεχόμενης υπηρεσίας. Τα νέα αυτά στοιχεία θεωρούνται καταλυτικά για μελλοντικές εφαρμογές σε βιομηχανικές και εφαρμογές ελέγχου σχετικών ασύρματων εννοιών και μερική αντικατάσταση ή την ενίσχυση συμβατικών ενσύρματων δικτύων με τεχνικές WSN. Παραδείγματα Βιομηχανικής λογικής και ελέγχου των αιτήσεων, είναι η Παρακολούθηση νερού/αποβλήτων υδάτων, Παρατήρηση της ποιότητας των υδάτων και Διαχείριση του δικτύου διανομής των υδάτων.

Δ) Γεωργία

Η χρήση ενός ασύρματου δικτύου απελευθερώνει τον αγρότη από τη διατήρηση της καλωδίωσης σε ένα δύσκολο περιβάλλον. Συστήματα νερού, τροφοδοσίας, βαρύτητας, μπορούν να παρακολουθούνται χρησιμοποιώντας πομπούς πίεσης για να παρακολουθούν τα επίπεδα δεξαμενή νερού, αντλίες μπορούν να ελέγχονται με τη χρήση ασύρματων συσκευών εισόδου / εξόδου και η χρήση του νερού μπορεί να μετρηθεί και να μεταδίδεται ασύρματα σε ένα κεντρικό σημείο ελέγχου για τιμολόγηση. Ο αυτοματισμός της άρδευσης επιτρέπει την πιο αποτελεσματική χρήση του νερού και μειώνει τα απόβλητα.

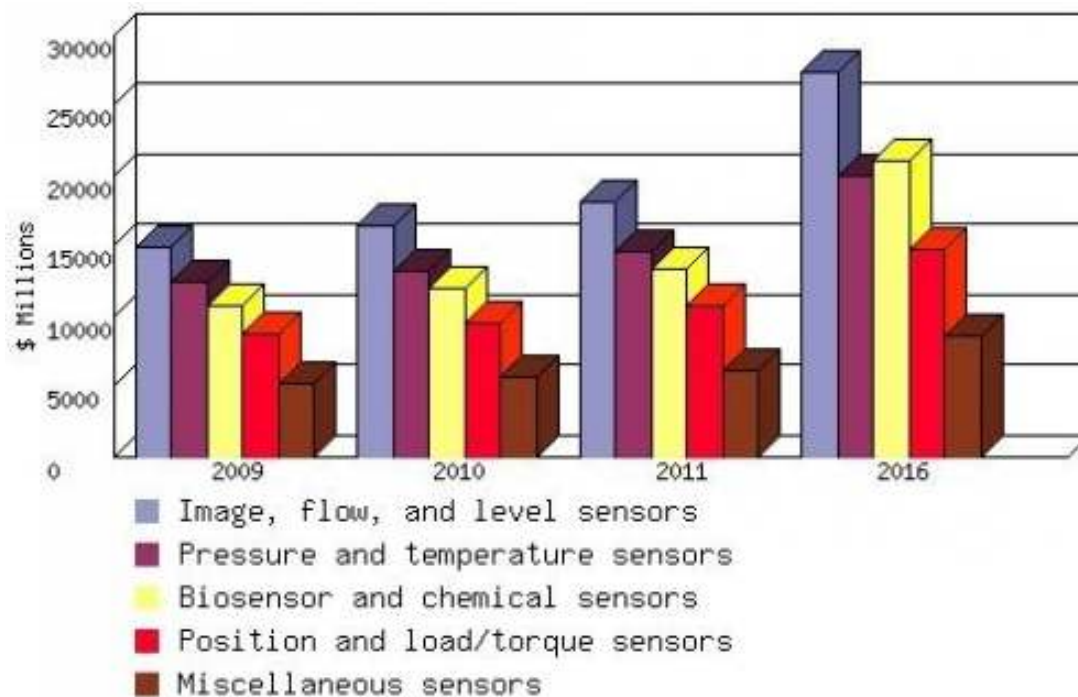
Έτσι τα ασύρματα δίκτυα αισθητήρων επιτρέπουν στους αγρότες να κάνουν ακριβή παρακολούθηση της καλλιέργειας τους κατά το χρόνο της ανάπτυξής της. Οπότε μπορούν να γνωρίζουν άμεσα την κατάσταση της καλλιέργειας σε όλα τα στάδια, κάτι το οποίο θα διευκολύνει τη διαδικασία λήψης απόφασης σχετικά με το χρόνο της συγκομιδής. Ομοίως τα ασύρματα δίκτυα αισθητήρων, να χρησιμοποιηθούν για

να ελέγχουν τα επίπεδα θερμοκρασίας και υγρασίας στο εσωτερικό εμπορικών θερμοκηπίων. Όταν η θερμοκρασία και η υγρασία πέφτει κάτω από συγκεκριμένα επίπεδα, τότε ο διαχειριστής του θερμοκηπίου πρέπει να ειδοποιείται μέσω e-mail ή με μήνυμα κειμένου στο κινητό τηλέφωνο. Ένας άλλος τρόπος ελέγχου από τα συστήματα υποδοχής είναι να πυροδοτήσουν τα συστήματα υδρονέφωσης, ώστε να ανοίξουν τους αεραγωγούς, ενεργοποιώντας τις περσίδες, ή να ελέγξουν μια ευρεία ποικιλία αντιδράσεων του συστήματος.

Άλλες εφαρμογές είναι ο Παθητικός εντοπισμός και παρακολούθηση μη συνεργάσιμων στόχων, δηλαδή άτομα που δεν φορούν οποιοδήποτε ταμπέλα και η Παρακολούθηση έξυπνου σπιτιού, δηλαδή οι δραστηριότητες ανθρώπων μέσα σε ένα έξυπνο σπίτι με τη χρήση ασύρματων αισθητήρων, που είναι ενσωματωμένα σε αντικείμενα καθημερινής χρήσης, και σχηματίζουν ένα WSN. [21]

Η παγκόσμια αγορά για αισθητήρες εκτιμήθηκε σε 56,3 δισεκατομμύρια δολάρια το 2010. Το 2011 αυξήθηκε σε 62,8 δισεκατομμύρια δολάρια, ενώ μέχρι το 2016 θα φτάσει σχεδόν τα 91,5 δισεκατομμύρια δολάρια, σε ένα σύνθετο ρυθμό ετήσιας ανάπτυξης (CAGR) 7,8%. Η αγορά των βιοαισθητήρων και χημικών αισθητήρων αναμένεται να εμφανίσουν τη μεγαλύτερη ανάπτυξη, σε ένα σύνθετο ρυθμό ετήσιας ανάπτυξης (CAGR) 9,6% κατά την περίοδο 5 ετών 2011-2016.

Οι αισθητήρες εικόνες παρουσιάζουν υψηλούς ρυθμούς ανάπτυξης, λόγω της αυξημένης χρήσης τους σε έξυπνα τηλέφωνα και πολλές άλλες φορητές συσκευές. [23]



Εικόνα 16 : Παγκόσμια αγορά αισθητήρων 2009 – 2016

Σύμφωνα με την Global Industry Analysts, Inc, ακόμα κι αν κυριαρχεί η οικονομική κρίση η παγκόσμια αγορά έξυπνων αισθητήρων αναμένεται να φθάσει 7.8 δις δολάρια μέχρι το 2015. Αυτό σημαίνει ότι η παγκόσμια ζήτηση για υψηλά ολοκληρωμένα έξυπνα συστήματα θα αυξηθεί δραματικά για τα επόμενα χρόνια. Σύμφωνα με την έκθεση της Διεθνούς Αγοράς (BizAcumen, Inc), θα αυξηθεί η ζήτηση των δικτύων αισθητήρων και έξυπνων αισθητήρων, που χρησιμοποιούνται ευρέως στις βιομηχανίες, συμπεριλαμβανομένης της αυτοκινητοβιομηχανίας, της ιατρικής, ψυχαγωγία, ασφάλεια και άμυνα, λόγω της αυξημένης χρήσης των ελέγχων σε διάφορους τομείς. Επίσης αυξάνεται η χρήση των έξυπνων αισθητήρων σε αντιτρομοκρατικές εφαρμογές μεταξύ αυτών τον εντοπισμό του φορτίου, η χρήση βιομετρικών στοιχείων. Τα έξυπνα συστήματα γίνονται εύκολα αποδεκτά από την αυτοκινητοβιομηχανία καθώς οι αυτοκινητοβιομηχανίες διερευνούν νέους τρόπους για να ασφαλίζουν τις ζωές των ανθρώπων. Οι αισθητήρες κίνησης παρακολούθησης και αδρανειακών αισθητήρων που βασίζονται σε μικροηλεκτρομηχανικό σύστημα (MEMS) χρησιμοποιούνται ευρέως σε ιατρικές εφαρμογές. Διάφορες εμφυτεύσιμες συσκευές που είναι μικρές σε μέγεθος και

ακόμη διαθέτουν υψηλή αξιοπιστία, χρησιμοποιούνται στη διάγνωση και παρακολούθηση του ασθενή από το σπίτι. [24]

Πρότυπα και Λειτουργικά

Υπάρχουν πολλά πρότυπα για ασύρματα δίκτυα αισθητήρων. Το IEEE επικεντρώνεται στα φυσικά στρώματα MAC. Το Internet Engineering Task Force λειτουργεί από το 3ο επίπεδο δικτύου OSI και άνω. Εκτός από αυτά, οργανισμοί όπως η Διεθνής Εταιρεία Αυτοματισμού προσφέρουν κάθετες λύσεις, που καλύπτουν όλα τα στρώματα πρωτοκόλλου. Επίσης υπάρχουν και πολλοί μη τυποποιημένοι, ιδιόκτητοι μηχανισμοί και προδιαγραφές. Ωστόσο, τα κυρίαρχα πρότυπα που χρησιμοποιούνται συνήθως σε ασύρματα δίκτυα αισθητήρων επικοινωνίας:

- WirelessHART
- IEEE 1451
- ZigBee / 802.15.4
- ZigBee IP
- 6LoWPAN

Μια σημαντική πρόκληση σε ένα WSN είναι η παραγωγή χαμηλού κόστους και μικροσκοπικών αισθητήριων κόμβων. Πολλοί από τους κόμβους είναι ακόμη στο στάδιο της έρευνας και της ανάπτυξης, ιδίως το λογισμικό τους. Επίσης εγγενής στην υιοθέτηση αισθητήρα του δικτύου είναι η χρήση των πολλών μεθόδων χαμηλής ισχύος για απόκτηση δεδομένων. Η ενέργεια είναι ο πιο σημαντικός πόρος στους κόμβους WSN, και καθορίζει τη διάρκεια ζωής των ασύρματων δικτύων αισθητήρων. Τα ασύρματα δίκτυα αισθητήρων προορίζονται για να αναπτύσσονται σε μεγάλους αριθμούς σε διάφορα περιβάλλοντα, συμπεριλαμβανομένων των απομακρυσμένων και εχθρικών περιοχών, όπου οι ad-hoc επικοινωνίες αποτελούν

βασικό συστατικό. Για το λόγο αυτό, οι αλγόριθμοι και τα πρωτόκολλα αυτά θα πρέπει να αντιμετωπίσουν τα ακόλουθα ζητήματα:

- Μεγιστοποίηση της διάρκειας ζωής
- Αντοχή και ανοχή σε σφάλματα
- Αυτο-ρύθμιση

Μερικά από τα σημαντικά θέματα έρευνας λογισμικού στα ασύρματα δίκτυα αισθητήρων (Wireless Sensor Networks) είναι:

- Λειτουργικά συστήματα
- Ασφάλεια
- Κινητικότητα

Τα λειτουργικά συστήματα ασύρματων κόμβων του δικτύου αισθητήρων είναι συνήθως λιγότερο πολύπλοκα από τα γενικής χρήσης λειτουργικά συστήματα. Μοιάζουν με τα ενσωματωμένα συστήματα, για δύο λόγους. Πρώτον, γιατί τα ασύρματα δίκτυα αισθητήρων συνήθως έχουν αναπτυχθεί με μια συγκεκριμένη και όχι ως μια γενική πλατφόρμα. Δεύτερον, υπάρχει η ανάγκη για χαμηλό κόστος και χαμηλή ισχύ των κόμβων.

Το TinyOS είναι ίσως το πρώτο λειτουργικό σύστημα ειδικά σχεδιασμένο για ασύρματα δίκτυα αισθητήρων. Το TinyOS βασίζεται σε ένα event-driven μοντέλο προγραμματισμού αντί multithreading. Τα προγράμματα για το TinyOS αποτελούνται από μηχανισμούς χειρισμού συμβάντων και εργασιών με run-to-completion σημασιολογία. Όταν προκύψει ένα εξωτερικό συμβάν, όπως ένα εισερχόμενο πακέτο δεδομένων ή μια ανάγνωση του αισθητήρα, το TinyOS σηματοδοτεί τον κατάλληλο χειρισμό συμβάντων. Οι χειριστές των συμβάντων μπορούν να αναρτούν εργασίες που έχουν προγραμματιστεί από τον πυρήνα TinyOS για εκτέλεση σε κάποια μελλοντική στιγμή.

Το LiteOS είναι ένα νέο λειτουργικό σύστημα πραγματικού χρόνου για ασύρματα δίκτυα αισθητήρων, το οποίο μοιάζει με το UNIX. Αυτό το λειτουργικό σύστημα επιτρέπει στους χρήστες του να λειτουργούν σε ασύρματα δίκτυα αισθητήρων, όπως λειτουργεί και το Unix. Συγκεκριμένα παρέχει ένα οικείο περιβάλλον προγραμματισμού που βασίζεται στο UNIX και στη γλώσσα προγραμματισμού C.

Το Contiki είναι ένα λειτουργικό σύστημα που χρησιμοποιεί ένα απλό στυλ προγραμματισμού σε C, παρέχοντας παράλληλα δυνατότητες όπως η υποστήριξη 6LoWPAN και Protothreads.

Το RIOT υλοποιεί μια δομή microkernel. Παρέχει multithreading μέσω του API και επιτρέπει την ανάπτυξη σε C/C++. Το RIOT υποστηρίζει κοινά πρωτόκολλα internet, όπως 6LoWPAN, IPv6, RPL, TCP, και UDP.

Οι απευθείας συνδέσεις πλατφορμών διαχείρισης των δεδομένων του αισθητήρα είναι on-line υπηρεσίες βάσεων δεδομένων που επιτρέπουν στους ιδιοκτήτες των αισθητήρων να εγγράφονται και να συνδέουν τις συσκευές τους σε αυτές. Ο στόχος είναι να μεταφέρουν τα δεδομένα σε μια ηλεκτρονική βάση δεδομένων, και να επιτρέπει στους προγραμματιστές να συνδέονται σε αυτή τη βάση δεδομένων ώστε να δημιουργούν τις δικές τους εφαρμογές με βάση τα υπάρχοντα δεδομένα. Παραδείγματα τέτοιων πλατφορμών είναι η Xively και η Wikisensing. Τέτοιες πλατφόρμες απλοποιούν την απευθείας συνεργασία των χρηστών σε διάφορα σύνολα δεδομένων που κυμαίνονται από την ενέργεια και το περιβάλλον δεδομένων που συλλέγονται από υπηρεσίες μεταφορών. Άλλες υπηρεσίες περιλαμβάνουν τη δυνατότητα στους προγραμματιστές να ενσωματώνουν γραφήματα σε πραγματικό χρόνο και widgets σε ιστοσελίδες, την ανάλυση και την επεξεργασία ιστορικών στοιχείων που πάρθηκαν από τις τροφοδοσίες δεδομένων, καθώς στέλνουν ειδοποιήσεις σε πραγματικό χρόνο από κάθε ροή δεδομένων για τον έλεγχο των σεναρίων, συσκευών και περιβαλλόντων. [21]

Έξυπνοι αισθητήρες

Η συσκευή πάνω στην οποία έχει ολοκληρωθεί τουλάχιστον ένα αισθητήριο στοιχείο και ένα κύκλωμα επεξεργασίας σήματος, αποτελεί έναν «έξυπνο αισθητήρα (smart sensor)». Ο όρος έξυπνος δηλώνει μερική ή ακόμη και ολοκληρωτική ενσωμάτωση της κεντρικής μονάδας επεξεργασίας, που διαθέτει ευφυΐα. Οι έξυπνοι αισθητήρες πρέπει να είναι ολοκληρωμένες και ευφυείς συσκευές, ενώ κάθε αισθητήρας που έχει ευφυΐα αλλά δεν είναι πλήρως ενσωματωμένο σύστημα, τότε λέγεται ευφυής αισθητήρας (intelligent sensor). Σύμφωνα με τον ορισμό που προτείνεται από τους Breckenridge και Husson, ένας έξυπνος αισθητήρας πρέπει να διαθέτει τρία βασικά χαρακτηριστικά:

- Να εκτελεί λογικές υπολογίσιμες συναρτήσεις
- Να επικοινωνεί με μία ή περισσότερες συσκευές και
- Να παίρνει μια απόφαση, σύμφωνα με τη λογική ή τα ασαφή δεδομένα του αισθητήρα.

Με το πέρασμα των χρόνων οι αισθητήρες έχουν εξελιχθεί. Οι αισθητήρες 1ης γενιάς συνδέονται με στοιχειώδη ηλεκτρονικά κυκλώματα ενίσχυσης και επεξεργασίας του σήματος τους. Στη συνέχεια οι αισθητήρες 2ης γενιάς αποτελούν τμήμα των αναλογικών ηλεκτρονικών κυκλωμάτων. Στους αισθητήρες 3ης γενιάς ο αισθητήρας και η μονάδα ρύθμισης του σήματος αποτελούνται, είτε από διακριτά στοιχεία, δηλαδή ολοκληρωμένα κυκλώματα και παθητικά στοιχεία, μέσα στο ίδιο άρθρωμα (module), είτε κατασκευάζονται σε υβριδικά ολοκληρωμένα κυκλώματα. Στους αισθητήρες 4ης γενιάς ο αισθητήρας και τα κυκλώματα ρύθμισης του σήματος του κατασκευάζονται στο ίδιο μονολιθικό ή υβριδικό ολοκληρωμένο κύκλωμα, ενώ ο μετατροπέας σήματος και τα ψηφιακά κυκλώματα επεξεργασίας και επικοινωνίας υλοποιούνται με διακριτά στοιχεία και βρίσκονται εκτός της διάταξης του αισθητήρα. Στους αισθητήρες 5ης γενιάς, ανάλογα με τη σχεδίαση τους, μπορούν να παράγουν ψηφιακή έξοδο με δυνατότητα αμφίδρομης επικοινωνίας. [25]

Γενικά αυτό που κάνει έναν αισθητήρα να είναι «έξυπνο» είναι η παρουσία του μικροελεγκτή ή μικροεπεξεργαστή που χρησιμοποιείται για υπολογισμούς, σύμφωνα με προκαθορισμένες εξισώσεις. Οι ευφυείς αισθητήρες διαθέτουν έξυπνες λειτουργίες που σημαίνει αυτοέλεγχος, αυτοεπικύρωση, αυτόδιάγνωση, αυτοπροσαρμογή, αυτοπροσδιορισμού, αυτοβαθμονόμηση κλπ [24]

Συγκριτικά με τους απλούς αισθητήρες, οι έξυπνοι, ακριβώς επειδή βασίζονται στην αυτοβαθμονόμηση και στον ενσωματωμένο αυτοέλεγχο, παρουσιάζουν μεγάλες και δαπανηρές προκλήσεις στον ερευνητικό τομέα. Έτσι ολοένα και περισσότερο, δημιουργούνται ολοκληρωμένοι έξυπνοι αισθητήρες που εκτελούν λογικές λειτουργίες, βασίζονται στην αμφίδρομη επικοινωνία, και μπορούν να προσαρμοστούν στις αλλαγές του περιβάλλοντος τους, καθώς και να λαμβάνουν αποφάσεις.

Η ικανότητά τους να παρέχουν σε πραγματικό χρόνο μέτρησης πολύπλοκα φαινόμενα, αυξάνει την τιμή του αισθητήρα. Στις βιομηχανικές εφαρμογές, οι αλγόριθμοι μπορούν να αναλύσουν δεδομένα από αισθητήρες και να βελτιστοποιήσουν την κατεργασία, την μεταποίηση, να επιταχύνουν την απόδοση, τη μείωση συντήρησης, την ελαχιστοποίηση των απορριμμάτων, και τελικά να βελτιώσουν τη συνολική ποιότητα του προϊόντος ή της διαδικασίας.

Στο μέλλον οι έξυπνοι αισθητήρες θα συνεχίσουν να συρρικνώνονται σε μέγεθος και η σημασία τους θα αναπτύσσεται, δεδομένου ότι εκτελούν μια αύξηση του αριθμού των λειτουργιών τους. Έξυπνα συστήματα αισθητήρων θα προσφέρουν μεγαλύτερη ευελιξία και ευκολία εφαρμογής.

Smart Devices

Όπως έχει ήδη αναφερθεί και σε προηγούμενη ενότητα μια έξυπνη συσκευή είναι μια ηλεκτρονική συσκευή, που γενικά συνδέεται με άλλες συσκευές ή δίκτυα μέσω διαφορετικών πρωτοκόλλων όπως το Bluetooth, NFC, WiFi, 3G κλπ και μπορεί να λειτουργήσει σε κάποιο βαθμό διαδραστικά και αυτόνομα. Υπάρχει ευρέως η γνώμη ότι μελλοντικά αυτά τα είδη των συσκευών θα υπερτερούν αριθμητικά από οποιαδήποτε άλλη μορφή έξυπνης πληροφορικής και επικοινωνίας. Ένας αξιοσημείωτος τύπος έξυπνης συσκευής στην παρούσα χρονική στιγμή είναι τα έξυπνα κινητά τηλέφωνα (smartphones) όπως το iPhone της Apple και οι περισσότερες από τις συσκευές που τρέχουν το λειτουργικό σύστημα Android, όπως το Google Nexus 7. Επίσης, ο όρος μπορεί αναφέρεται σε μια πανταχού παρούσα υπολογιστική συσκευή που παρουσιάζει ορισμένες ιδιότητες της πανταχού παρούσας υπολογιστικής, που μπορεί να συμπεριλαμβάνει και τεχνητή νοημοσύνη. Οι Έξυπνες συσκευές έχουν σχεδιαστεί για να υποστηρίξουν μια ποικιλία εφαρμογών που σχετίζονται με την πανταχού παρούσα υπολογιστική και χρησιμοποιούνται σε τρία βασικά περιβάλλοντα συστήματος: τον φυσικό κόσμο, διάφορους χώρους με επίκεντρο τον άνθρωπο και σε κατανεμημένα υπολογιστικά περιβάλλοντα.

Όπως αναφέρθηκε και σε προηγούμενη ενότητα ο Mark Weiser πρότεινε τρεις βασικές μορφές για συσκευές έξυπνων συσκευών: Tabs (πχ smartphones, έξυπνες κάρτες), Pads (πχ φορητοί υπολογιστές) και πίνακες – Boards (πχ υπολογιστές οριζόντιας επιφάνειας και κάθετους έξυπνους πίνακες. Χαλαρώνοντας τα χαρακτηριστικά τους μπορούμε να επεκτείνουμε το εύρος τους σε διαφορετικές και πιο χρήσιμες συσκευές για το Ubiquitous Computing. Ως εκ τούτου, έχουν προταθεί τρεις επιπλέον μορφές για πανταχού συστήματα:

- Dust (σκόνη): μικροσκοπικές συσκευές χωρίς άμεσες διασυνδέσεις αλληλεπίδρασης ανθρώπου μηχανής, πχ, Micro Electro - Mechanical Systems (MEMS), που κυμαίνονται από νανόμετρα μέχρι σε χιλιοστά.

- Skin (δέρμα): υφάσματα με βάση το φως που εκπέμπουν. Αυτά μπορούν να σχηματίζονται σε πιο ευέλικτες επιφάνειες εμφάνισης μη επίπεδες και σε προϊόντα, όπως ρούχα, κουρτίνες. Οι OLEDs συσκευές (οργανική δίοδος εκπομπής φωτός) είναι δίοδοι εκπομπής φωτός (LED) και χρησιμοποιούνται για να δημιουργήσουν ψηφιακές οθόνες σε συσκευές όπως τηλεοράσεις, οθόνες υπολογιστών, φορητά συστήματα, όπως τα κινητά τηλέφωνα, φορητές κονσόλες παιχνιδιών και PDAs.
- Clay (πηλός): είναι σύνολα των MEMS και μπορούν να σχηματίζονται σε αυθαίρετα τρισδιάστατα σχήματα, όπως αντικείμενα που μοιάζουν με διαφορετικά είδη φυσικού αντικειμένου. [26]



Εικόνα 17 : Έξυπνες συσκευές

Ένα παράδειγμα έξυπνης συσκευής είναι η εφαρμογή Automatic Link που συνδέει το smart phone με το αυτοκίνητο. Η εφαρμογή αρχίζει να λειτουργεί όταν συνδεθεί στο σύστημα του αυτοκινήτου σας (on-board θύρα διάγνωσης), το οποίο βρίσκεται κοντά στο κάθισμα του οδηγού, κάτω από την αριστερή πλευρά του ταμπλό. Η θύρα OBD βρίσκεται σε όλα τα οχήματα που κυκλοφορούν στις ΗΠΑ από το 1996, και ο

κύριος στόχος της είναι να παρέχει πληροφορίες σχετικά με το επίπεδο των καυσίμων, την ταχύτητα και αναφορές σφαλμάτων. Η εφαρμογή αυτή αναλαμβάνει αυτούς τους δείκτες, προσφέροντας σημαντικές πληροφορίες που θα βοηθήσουν τους οδηγούς να ταξιδέψουν με ασφάλεια, μειώνοντας παράλληλα την ενεργειακή κατανάλωση. [27]



Εικόνα 18 : Έξυπνη συσκευή επικοινωνίας αυτοκινήτου με κινητό τηλέφωνο

Industry

Οι Βιομηχανικές εφαρμογές έξυπνου αισθητήρα με κατανεμημένες αρχιτεκτονικές αποτελούν μια νέα ευκαιρία για την υλοποίηση συστημάτων που είναι πιο οικονομικά, αποδοτικά, ευέλικτα, επεκτάσιμα, καθώς και αξιόπιστα. Σε αυτά τα συστήματα υπάρχει άμεση διεπαφή των αισθητήρων και ενεργοποιητών (actuators) με το βιομηχανικό δίκτυο επικοινωνίας, οπότε βελτιώνεται η απόδοση του συστήματος. Τα δεδομένα και η διαγνωστική διαδικασία μπορεί να είναι διαθέσιμη ταυτόχρονα σε πολλά συστήματα, καθώς και από κοινού στον Παγκόσμιο Ιστό. Ωστόσο, οι αισθητήρες - ιδίως εκείνοι με χαμηλό κόστος - δεν μπορούν να χρησιμοποιούν τυποποιημένα πρωτόκολλα επικοινωνίας, κατάλληλα για υπολογιστές και PLC (Programmable Logic Controller). Στα βιομηχανικά συστήματα επικοινωνίας, έχουν σχεδιαστεί διαφορετικές διεπαφές διαύλων για συγκεκριμένες περιοχές εφαρμογών αισθητήρων, λαμβάνοντας υπόψη, το υψηλό επίπεδο βιομηχανικής χρήσης εξοπλισμών ενσύρματης ή ασύρματης Ethernet. Πρόσφατα, οι παραδοσιακές διεπαφές διαύλου αντικαθίσταται από πραγματικού χρόνου

πρωτόκολλα Ethernet, καθώς και από άλλα πρωτόκολλα, που πληρούν τις απαιτήσεις σε πραγματικό χρόνο λειτουργίας.

Ένα παράδειγμα αισθητήρων είναι η εφαρμογή τους στην τεχνολογία τροφίμων. Συγκεκριμένα η χρήση των RFID (Radio Frequency Identification) στη βιομηχανία τροφίμων, ποτών, σημαίνει ότι όλα τα τρόφιμα, τα ποτά και λοιπά προϊόντα, χρησιμοποιούν την συσκευασία για προώθηση, προστασία και διάθεση και χρησιμοποιούν υποστρώματα από χαρτί, χαρτόνι, πλαστικό. Αυτό δίνει την δυνατότητα να έχουμε εκτυπωμένα «ηλεκτρονικά» έντυπα με δυνατότητες εφαρμογής. [25]

Ένα σύστημα RFID αποτελείται από:

- Ετικέτες (tags)
- Αναγνώστες (readers)
- Κεραίες (access point)
- Εκτυπωτές (label printers / tag encoders)
- Λογισμικό εφαρμογών
- Υπολογιστή (host)

Η ανάγκη για την χρήση τέτοιων συστημάτων στη βιομηχανία τροφίμων προέκυψε γιατί ήθελαν να παρακολουθούν τις διαδικασίες παραγωγής και προώθησης των προϊόντων στην αγορά. Για να διατηρείται η ιχνηλασιμότητα των προϊόντων σε όλα τα στάδια χρησιμοποιήθηκε η τεχνολογία RFID, από τα τμήματα ελέγχου ποιότητας των εταιρειών παραγωγής καθώς και από τους προμηθευτές. Η ιχνηλασιμότητα τροφίμων ορίζεται ως η ικανότητα της μοναδικότητας του εντοπισμού ενός προϊόντος , να το ακολουθεί κάποιος σε όλη τη διάρκεια της παραγωγής, μεταποίησης και διανομής του. Οπότε απαιτείται η συλλογή, καταγραφή πληροφοριών καθώς και διασύνδεση των διάφορων συστημάτων επικοινωνίας.

Έχουν αναπτυχθεί διάφορα μοντέλα για την ιχνηλασιμότητα στην εφοδιαστική αλυσίδα τροφίμων. Η ερευνήτρια Monique Jansen Vullers (2003) ανέπτυξε ένα μοντέλο που θέτει τέσσερα βασικά χαρακτηριστικά για την ιχνηλασιμότητα:

Φυσική ακεραιότητα παρτίδας του προϊόντος

- Συλλογή δεδομένων
- Αναγνώριση προϊόντος και σύνδεση διαδικασιών
- Ανάκτηση των δεδομένων όταν ζητηθούν

Οπότε σύμφωνα με αυτό το μοντέλο τα προϊόντα μιας παρτίδας παραγωγής (batch) τοποθετούνται σε κιβώτια παλέτες, οι μονάδες συσκευασίας ονομάζονται TRUs και καθένα από αυτό φέρνει μια ετικέτα RFID. Οι συγκεκριμένες μονάδες φορτώνονται και μεταφέρονται στο επόμενο στάδιο της διαδικασίας. Τα TRUs είναι ανιχνεύσιμες μονάδες και μεταφέρονται σε διάφορα στάδια της αλυσίδας. Τα batches είναι μονάδες που χρησιμοποιούνται εσωτερικά σε ένα στάδιο της διαδικασίας. Ο RFID reader εντοπίζει τα TRUs. Το συγκεκριμένο μοντέλο μας δίνει πληροφορίες σχετικά με την διαδικασία παραγωγής, τις μετατροπές που γίνονται στα προϊόντα και στα batches, καθώς και πληροφορίες για την πορεία ενός TRUs ως προς τον εντοπισμό του ανά πάσα ώρα μέσα στην αλυσίδα, έτσι όπως εντοπίστηκε από το RFID σύστημα. [28]

Ένα άλλο μοντέλο είναι των ερευνητών John Cooney και Barry Winkless που αποτυπώνουν τη νέα τεχνολογία στην εφοδιαστική αλυσίδα και τη συμπεριφορά των τροφίμων. Συγκεκριμένα η έρευνα τους εστιάζει στην αξιοποίηση των μεθοδολογιών TRIZ (Teoriya Resheniya Izobretatelskikh Zadatch) 1τονίζοντας την κρίσιμη σημασία της ποιότητας και ασφάλειας των τροφίμων στην αγορά και τον καταναλωτή. Στο συγκεκριμένο μοντέλο υπάρχουν διάφοροι δείκτες στη συσκευασία τροφίμων, όπως για παράδειγμα μετρητής θερμοκρασίας που μπορεί να διαβάσει με ακρίβεια την θερμοκρασία του προϊόντος. Μπορεί να κάνει σύνθετες μετρήσεις και να μετρήσει την θερμοκρασία ακόμη και αν το πακέτο έχει αλλοιωθεί. Επίσης δίνεται η δυνατότητα για την ανάπτυξη ακόμη πιο σύνθετων

μετρήσεων του πακέτου, που θα μπορεί να μετρήσει τη θερμοκρασία του προϊόντος, την περιεκτικότητα σε βιταμίνες, καθώς και αν έχει υποστεί βλάβη στο περιτύλιγμα της συσκευασίας. [29]

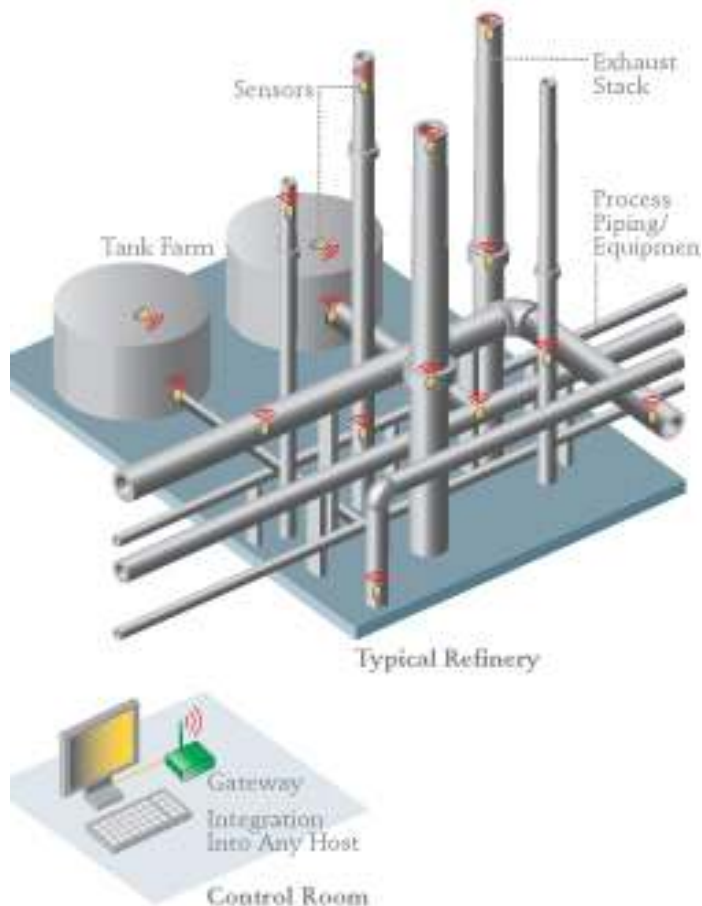
Σημείωση 1: Ρώσικη λέξη. Αγγλικό ακρωνύμιο TIPS, είναι αλγόριθμος επίλυσης προβλημάτων



Εικόνα 19 : Τοποθέτηση ετικέτας θερμοκρασίας κατά την παραγωγή

Ένα άλλο παράδειγμα αισθητήρων στον βιομηχανικό τομέα είναι ο ορθός έλεγχος των συστημάτων κατά τη διάρκεια λειτουργίας τους, που έχει σημαντικό ρόλο τόσο για την επίβλεψή τους, όσο και για την ασφάλεια του προσωπικού. Για παράδειγμα στα διυλιστήρια, το περιβάλλον εργασίας είναι επικίνδυνο για την υγεία του προσωπικού, καθώς είναι κρίσιμες οι παραγωγικές διαδικασίες, λόγω των υψηλών θερμοκρασιών, της τοξικότητας από την ύπαρξη βλαβερών αερίων, ή ακόμη της ύπαρξης περιοχών που δεν προσβάσιμες από τον άνθρωπο. Για τέτοιους είδους περιπτώσεις είναι αναγκαία η χρήση ασύρματων αισθητήρων για την παροχή αυτοματοποιημένων διεργασιών, όπως για παράδειγμα η συντήρηση των μηχανημάτων μέσω της λήψης πληροφοριών για την κατάστασή τους και η λήψη άμεσων αποφάσεων σε περιπτώσεις σφαλμάτων. Στα διυλιστήρια υπάρχουν ασύρματοι αισθητήρες που μετρούν τη θερμοκρασία σε όλα τα στάδια της

επεξεργασίας του πετρελαίου και όταν σημειωθεί υπέρβαση του επιτρεπόμενου ορίου, ειδικά σήματα συναγερμού ενημερώνουν τους τεχνικούς. Επίσης, στα ασύρματα δίκτυα αισθητήρων υπάρχει υπηρεσία απομακρυσμένου ελέγχου που σε συνδυασμό με την ύπαρξη ηλεκτρονικών προγραμμάτων, όπως το LABVIEW που χρησιμοποιείται ευρέως στη βιομηχανία, παρέχουν πλήρη διαχείριση και έλεγχο απαιτητικών εφαρμογών. [30] [31]



Εικόνα 20 : Ασύρματη τεχνολογία αισθητήρων που χρησιμοποιείται στα διυλιστήρια, με αισθητήρες παρακολούθησης για τις κρίσιμες μεταβλητές σε διάφορες διαδικασίες, καθώς και τα επίπεδα δεξαμενών.

Smart Environments

Όπως έχει ήδη αναφερθεί, η έννοια του Έξυπνου Περιβάλλοντος (Smart Space) είναι ένα σύνολο συσκευών και υπηρεσιών που ανήκουν, ελέγχονται, ή χορηγούνται από ένα μόνο χρήστη. Οι συσκευές και οι υπηρεσίες βρίσκονται μέσα σε ένα δυναμικό

χώρο, μπορούν να συνδεθούν και να επικοινωνήσουν μεταξύ τους, ενώ ταυτόχρονα μπορούν να αυτοβελτιώνονται ως προς την συμπεριφορά του συστήματος. Τα έξυπνα περιβάλλοντα είναι ένα «υποπροϊόν» του pervasive computing διαθέτουν σχετικά φτηνή υπολογιστική ισχύ, δημιουργώντας μια ευχάριστη εμπειρία στην αλληλεπίδραση μεταξύ ανθρώπου και συστήματος.

Ένα παράδειγμα έξυπνου περιβάλλοντος είναι το έξυπνο σπίτι, δηλαδή ένα σπίτι με νοημοσύνη που σκέπτεται και ενεργεί για τις καθημερινές ανάγκες και συνήθειες των κατοίκων του. Σε ένα έξυπνο σπίτι είναι εφικτό πολλά πράγματα (π.χ. η ασφάλεια, η θέρμανση, ο φωτισμός, οι ηλεκτρικές συσκευές, τα συστήματα Audio video) να είναι στον απόλυτο έλεγχο των χρηστών του, απλά με το πάτημα ενός κουμπιού, είτε ο χρήστης είναι μέσα είτε εκτός, ακόμη και χιλιόμετρα μακριά. Το έξυπνο σπίτι προσαρμόζεται στον κάθε χρήστη του και μαθαίνει κάθε βήμα βήμα κάθε συνήθεια ολόκληρο το 24ωρο. Το έξυπνο σπίτι μπορεί να είναι εφοδιασμένο με εκατοντάδες ή χιλιάδες ασύρματους αισθητήρες που βοηθούν στην εξασφάλιση της υγείας, της ασφάλειας, και της παραγωγικότητας των κατοίκων του. Αν αυτοί οι αισθητήρες είναι σε συνεχή λειτουργία τότε θα δαπανά πολλή ενέργεια και εύρος ζώνης. Συνεπώς η υποδομή θα χρειάζεται συνεχή συντήρηση. Η εφαρμογή των ασύρματων δικτύων αισθητήρων που διαθέτουν γνωστικές ικανότητες, ενεργούν με έξυπνο τρόπο, και έτσι εξοικονομείται ενέργεια. Σε αυτά τα συστήματα χρησιμοποιούνται οι τεχνικές εκμάθησης για την αναγνώριση των δραστηριοτήτων του χρήστη (περιγράφονται αναλυτικά και σε επόμενη ενότητα) που εκτελούνται μέσα στο έξυπνο περιβάλλον. Οι πληροφορίες που συγκεντρώνονται αποτελούν το πλαίσιο και στη συνέχεια μεταφέρονται στο δίκτυο και αναλόγως το σύστημα αποφασίζει πότε οι κόμβοι θα είναι ανενεργοί, πότε θα ενεργοποιηθούν, και πώς θα γίνεται η δρομολόγηση πληροφοριών και δραστηριοτήτων.

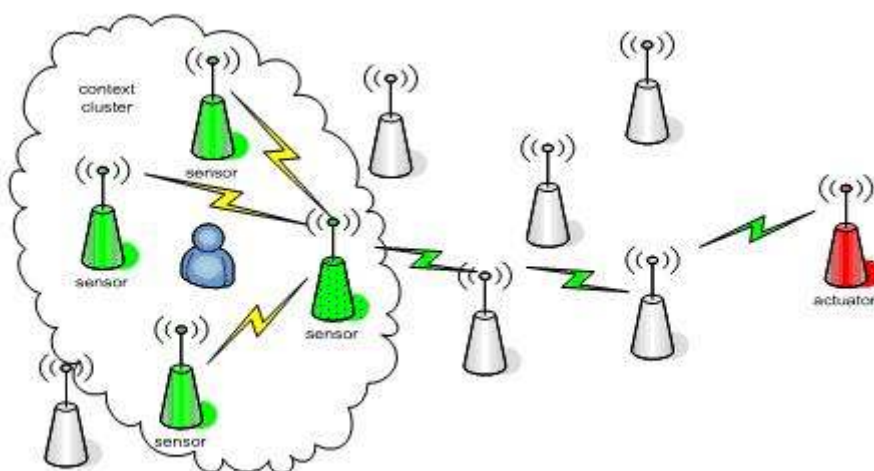


Εικόνα 21 : Έξυπνο σπίτι

Η δημιουργία ενός δικτύου αισθητήρων για ένα έξυπνο σπίτι θέτει νέες προκλήσεις και ευκαιρίες στον τομέα δικτύων. Οι ρυθμίσεις που απαιτούνται για την εγκατάσταση ενός έξυπνου σπιτιού διαφέρει από τις υπόλοιπες εφαρμογές των ασύρματων δικτύων αισθητήρων. Στην περίπτωση του έξυπνου σπιτιού υπάρχει συνεχόμενη ανίχνευση των κινήσεων, που καθοδηγούνται αντίστοιχα από την δραστηριότητα του κατοίκου. Για παράδειγμα αν υπάρχει ένα άτομο με ειδικές ανάγκες ή ασθενής, θα πρέπει να παρακολουθείται συνεχώς δηλαδή 24 ώρες X 7 ημέρες. Για την εγκατάσταση του δικτύου στο σπίτι οι αισθητήρες υποστηρίζονται από μικρού όγκου μπαταρίες για την παροχή ενέργειας, ενώ ταυτόχρονα έχουν χαμηλή κατανάλωση ενέργειας, που έτσι αυξάνεται ο χρόνος ζωής του δικτύου. Για παράδειγμα, αν κανείς δεν είναι στο σπίτι, βάσει αλγορίθμων παρακολούθησης αισθητήρων οι τιμές «πέφτουν» κοντά στο μηδέν, προκειμένου να μειωθεί η κατανάλωση ενέργειας. Αν κάποιος κάτοικος είναι στο σπίτι, η ανίχνευση και η δικτύωση θα πρέπει να παρέχει αξιόπιστη και προβλέψιμη παρακολούθηση.

Όταν το σύστημα είναι σε λειτουργία παρακολούθησης, τότε οι κόμβοι του δικτύου ανιχνεύουν τα αντίστοιχα γεγονότα και εκτελούν την ανάλυση αναγνώρισης της αντίστοιχης δραστηριότητας. Όπως έχει αναφερθεί και σε άλλη ενότητα, οι κόμβοι αυτοί μπορεί να είναι σε θέση να συμπεράνουν πρόσθετες συναφείς πληροφορίες. Λόγω της διαδοχικής σειράς των γεγονότων του αισθητήρα δημιουργούνται

μοντέλα δραστηριοτήτων, και έτσι οι αισθητήρες θα μάθουν τους γειτονικούς κόμβους εντός της χωρικής διαμόρφωσης του περιβάλλοντος που έχει οριστεί. Επίσης, επειδή οι δραστηριότητες των κατοίκων συμβαίνουν με διαφορετικές συχνότητες σε διαφορετικές θέσεις μέσα στο σπίτι, η πληροφορία για το περιεχόμενο της κάθε δραστηριότητας θα ενσωματωθεί στα πρωτόκολλα διαχείρισης τοπολογίας και δρομολόγησης. Η εικόνα που ακολουθεί δείχνει ότι οι αισθητήρες (με πράσινο χρώμα) εντοπίζουν δραστηριότητες των κατοίκων στον κόμβο δημιουργώντας ένα πλαίσιο πληροφοριών. Μέσω ενός κεντρικού αισθητήρα του κόμβου, ο ενεργοποιητής (με κόκκινο χρώμα) διαβάζει τα κατάλληλα μηνύματα ώστε να εκτελέσει μια αυτοματοποιημένη διαδικασία. [32]



Εικόνα 22 : Αισθητήρες και ενεργοποιητές μέσα σε ένα δίκτυο

Ένα από παράδειγμα έξυπνου σπιτιού είναι από την ερευνητική ομάδα του Πανεπιστημίου του Essex, που υλοποίησε το σύστημα iSpace χρησιμοποιώντας ένα δωμάτιο της φοιτητικής εστίας για τους ερευνητικούς σκοπούς των Ευφυή Κτηρίων (Intelligent Building - IB).

Το iSpace εφαρμόστηκε σε ένα δυάρι διαμέρισμα ειδικά κατασκευασμένο για τη μελέτη της πανταχού παρούσας υπολογιστικής. Συγκεκριμένα η υποδομή του διαμερίσματος είναι ειδικά διαμορφωμένη ώστε να υπάρχουν δίκτυα αισθητήρων και ενεργοποιητών, ενώ ψευδοροφές και τοίχοι είναι ειδικά κατασκευασμένοι, έτσι ώστε η ετερογενής δικτύωση να μην είναι εμφανή. Ο εγκατεστημένος εξοπλισμός

αποτελείται από ελεγχόμενο φωτισμό, θέρμανση και κουρτίνες, καθώς και οθόνες αφής και ηχεία. Ο εντοπισμός θέσης του χρήστη γίνεται με ετικέτες ραδιοσυχνότητας. Όλες οι πληροφορίες που συλλέγονται από το δίκτυο αισθητήρων αποστέλλονται σε κεντρική βάση δεδομένων.

Τέτοιου είδους συστήματα δίνουν τη μελλοντική ικανότητα στους ερευνητές να αναπτύξουν εξελιγμένη τεχνολογία σε ευφυή κτήρια και προσαρμοστικά περιβάλλοντα. Στην πανταχού παρούσα υπολογιστική θα επιτρέπεται σε ενσωματωμένους πράκτορες να λειτουργούν αυτόνομα, δηλαδή να παρακολουθούν και να μάθουν από τη συμπεριφορά των χρηστών. Τελικός στόχος των συστημάτων είναι να εξειδικεύουν τη συμπεριφορά τους σύμφωνα με τον χρήστη του κτιρίου με ένα διακριτικό τρόπο, όπου ο χρήστης θα έχει πάντα τον έλεγχο.

Η ομάδα του συστήματος iSpace έχει προβλέψει στον σχεδιασμό του ένα ευρύ φάσμα πιθανών χρηστών, όπως για παράδειγμα άτομα με ειδικές ανάγκες και ηλικιωμένους ανθρώπους. Έτσι το σύστημα στοχεύει στην μεγιστοποίηση του οφέλους της νέας τεχνολογίας για όλα τα τμήματα της κοινωνίας.

Επίσης το iSpace θα επιτρέψει σε όσους ερευνητές ασχολούνται με την κοινωνική έρευνα να αναπτύξουν εικονικές συσκευές και διεπαφές σύμφωνα με τις προτιμήσεις των χρηστών καθώς και την αντίστοιχη προσαρμοστικότητα τους στο εκάστοτε περιβάλλον. Για παράδειγμα η ερευνητική ομάδα έχει αναπτύξει εφαρμογές στο σύστημα που σύμφωνα με τον χρήστη, τις προτιμήσεις του και την θέση του μέσα στο κτήριο μπορεί να ενεργοποιεί τις αντίστοιχες διεπαφές.



Εικόνα 23 : Ερευνητικό πρόγραμμα Atraco έξυπνου σπιτιού

Για παράδειγμα για το έργο Atraco, οι ερευνητές σχεδίασαν τρεις βασικές εφαρμογές διεπαφών. Η διεπαφή για αναπαραγωγής μουσικής προσφέρει τα τυπικά στοιχεία ελέγχου αναπαραγωγής και επιλογής playlist. Η μουσική παρέχεται από ένα μόνο σετ ηχείων στο χώρο, και μπορεί να ελεγχθεί από οποιαδήποτε οθόνη που βρίσκεται κοντά στον χρήστη. Η δεύτερη διεπαφή προσφέρει slideshow φωτογραφιών. Η οθόνη με τις φωτογραφίες συγχρονίζεται και εμφανίζεται σε όλες τις οθόνες που διαθέτουν photo. Τέλος, η διεπαφή ελέγχου φωτισμού όπου ο χρήστης αλληλεπιδρά με τις συσκευές του δικτύου για την αυξομείωση του φωτισμού. [33]

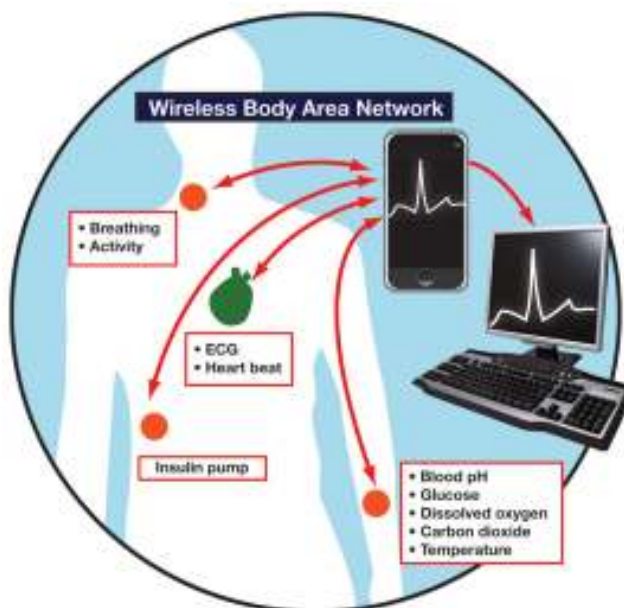
Στο χώρο της υγείας

Με την χρήση της πληροφορικής και των ασύρματων δικτύων, η ιατρική εξελίσσεται στην τηλεϊατρική και επιτρέπεται η επικοινωνία, φροντίδα και υποστήριξη μεταξύ γιατρών, ιατρικών μονάδων και ασθενών ακόμη και σε μεγάλη απόσταση ή όταν δεν είναι εφικτή η πρόσβαση για παροχή υπηρεσιών. Έτσι δημιουργήθηκαν οι όροι κατ' οίκον φροντίδα και τηλεϊατρική, όπου μέσω των εφαρμογών, διασφαλίζεται η άνεση του ασθενή να παρακολουθείται και να παίρνει οδηγίες από το γιατρό μέσω ενός ασύρματου δικτύου.

Στον τομέα της υγείας και της φαρμακευτικής, η χρήση των ασύρματων αισθητήρων είναι ξεχωριστή και δύσκολη αφού σχετίζεται με το ανθρώπινο σώμα. Για αυτό το

λόγο στον τομέα της Ιατρικής δεν χρησιμοποιείται ο όρος WSNs αλλά ο όρος BSN (Body Sensor Area Network), γιατί υπάρχουν κάποιες διαφοροποιήσεις στα συγκεκριμένα δίκτυα. Παρόλο που το ανθρώπινο σώμα είναι μικρής κλίμακας περιβάλλον, λόγω της πολυπλοκότητας του απαιτεί διάφορους τύπους παρακολούθησης συχνοτήτων (συγκεκριμένες συχνότητες που είναι επιτρεπτές για να λειτουργούν αυτές οι συσκευές στο ανθρώπινο σώμα). Υπάρχουν ασύρματες συσκευές που λειτουργούν στο φάσμα MBAN (2360 -2390 MHz) και χρησιμοποιούνται για να παρακολουθούν ενεργά την υγεία του ασθενούς. Για παράδειγμα, να παρακολουθούν τα επίπεδα γλυκόζης του αίματος ή την πίεση. Οπότε ένας αριθμός των ευφυών αισθητήρων ενσωματώνονται σε ένα wearable ασύρματο δίκτυο. Αυτή η περιοχή στηρίζεται σχετικά με τη σκοπιμότητα της εμφύτευσης πολύ μικρών βιοαισθητήρων μέσα στο ανθρώπινο σώμα που είναι άνετα και δεν βλάπτουν τις δραστηριότητες του ανθρώπου. Οι εμφυτευμένοι αισθητήρες στο ανθρώπινο σώμα συλλέγουν διάφορες φυσιολογικές τιμές των οργάνων του σώματος, προκειμένου να παρακολουθούν την κατάσταση της υγείας του ασθενούς. Οι πληροφορίες μεταδίδονται ασύρματα σε μια εξωτερική μονάδα επεξεργασίας. Αυτή η συσκευή μεταδίδει αμέσως όλες τις πληροφορίες σε πραγματικό χρόνο στους υπεύθυνους γιατρούς ανεξάρτητα από την γεωγραφική τους τοποθεσία.

Η ανάπτυξη των προσωπικών δικτύων που αφορούν την τηλεϊατρική παρακολούθηση του ανθρώπινου σώματος αποτελεί μεγάλη μελλοντική πρόκληση. Η χρήση των δικτύων BSN εφαρμόζεται με επιτυχία στην τηλεϊατρική ώστε ο ασθενής να αποδεσμεύεται από τους νοσοκομειακούς περιορισμούς. Ο στόχος είναι ένα καλύτερο επίπεδο ζωής και πιο φθινό κόστος ιατρικής περίθαλψης. [34]
[35]



Εικόνα 24 : Wireless Body Area Network

Άλλες ονομασίες για το Δίκτυο αισθητήρων του σώματος (BSN), είναι Δίκτυο Σώματος (WBAN wireless body area network) ή pPAN (patient Personal Area Network. Ειδικότερα, το δίκτυο αποτελείται από πολλές μικροσκοπικές μονάδες αισθητήρων του σώματος (BSUs) μαζί με μια ενιαία κεντρική μονάδα του σώματος (BCU). Η ανάπτυξη της τεχνολογίας WBAN ξεκίνησε γύρω στο 1995 για τη χρησιμοποίηση προσωπικού δικτύου (WPAN) για επάνω, κοντά και γύρω από το ανθρώπινο σώμα. Ένα σύστημα WBAN μπορεί να χρησιμοποιήσει WPAN ασύρματες τεχνολογίες ως πύλες για να φτάσει μεγαλύτερες αποστάσεις. Οι αρχικές εφαρμογές ήταν κατά κύριο λόγο στον τομέα της υγείας, ιδιαίτερα για τη συνεχή παρακολούθηση και καταγραφή ζωτικών παραμέτρων των ασθενών που πάσχουν από χρόνιες ασθένειες, όπως ο διαβήτης, το άσθμα και καρδιακές προσβολές. Ένα δίκτυο BAN παρακολουθεί έναν ασθενή και μπορεί να ειδοποιεί το νοσοκομείο, πριν προλάβει ο ασθενής να πάθει καρδιακή προσβολή, μετρώντας περιοδικά τις αλλαγές τιμών στα ζωτικά του σημεία. Παράδειγμα δικτύου BAN για έναν διαβητικό ασθενή θα μπορούσε να είναι η αυτόματη έγχυση ινσουλίνης μέσω αντλίας, όταν διαπιστώνεται μείωση του επίπεδου της ινσουλίνης στον οργανισμό του. [36]

Smart City

Η ταχεία ανάπτυξη των νέων τεχνολογιών και των διαδικασιών καινοτομίας οδήγησαν σε ένα νέο μοντέλο της πόλης, «έξυπνη πόλη». Ο τύπος της έξυπνης πόλης χρησιμοποιεί νέες τεχνολογίες για να γίνουν πιο βιώσιμες, λειτουργικές, ανταγωνιστικές και σύγχρονες με τη χρήση των νέων τεχνολογιών, την προώθηση της καινοτομίας και τη διαχείριση γνώσεων, συγκεντρώνοντας 6 βασικούς τομείς παροχής των υπηρεσιών: οικονομία, η κινητικότητα, το περιβάλλον, η ιθαγένεια, η ποιότητα ζωής και τη συμμετοχή των πολιτών στη διαχείριση των πόλεων. Οπότε μια «έξυπνη πόλη» είναι μια περιοχή ενός οικισμού που χρησιμοποιεί με ένα συστηματικό τρόπο (οικολογικό, κοινωνικό και οικονομικό) τα βιώσιμα προϊόντα, υπηρεσίες, τεχνολογίες, διαδικασίες και υποδομές, και υποστηρίζεται από τις τεχνολογίες της πληροφορικής και των τηλεπικοινωνιών. Κατά τον οικονομολόγο Gildo Seisdedos Domínguez, η ιδέα της έξυπνης πόλης σημαίνει να έχει ουσιαστική απόδοση. Βέβαια η απόδοση της θα πρέπει να γίνεται με βάση την έξυπνη διαχείριση και την ολοκληρωμένη χρήση των τηλεπικοινωνιών, καθώς και την ενεργή συμμετοχή των πολιτών της. Από αυτό, συνεπάγεται ένα νέο είδος διακυβέρνησης, με πραγματική συμμετοχή των πολιτών στη δημόσια πολιτική.

Οι βασικοί στόχοι μιας "έξυπνης" πόλης προκειμένου να βελτιώσει τη βιωσιμότητα των οικισμών της, μειώνοντας ή ακόμη και αποφεύγοντας τις αρνητικές συνέπειες της αστικοποίησης είναι:

- Η βελτίωση ποιότητας ζωής όλων των πολιτών και την ικανότητά τους να συμμετέχουν ενεργά στην κοινωνία,
- Η μείωση χρήσης των πεπερασμένων πόρων ώστε να υποστηρίξει την χρήση των ανανεώσιμων πηγών ενέργειας,
- Να εξασφαλίσει και να βελτιστοποιήσει τη μακροπρόθεσμη παροχή δημόσιων υπηρεσιών στους πολίτες της,
- Να ενισχύσει την φυσική κατάσταση για την επιβίωση, την ικανότητα προσαρμογής και γενικότερα την ανθεκτικότητα της ζώνης του οικισμού,

- Να δημιουργήσει μια κουλτούρα της διαφάνειας στη λήψη των αποφάσεων και την κοινωνία της γνώσης,
- Να διατηρήσει και να αυξήσει όσο γίνεται την ανταγωνιστικότητα της τοπικής οικονομίας στο μέλλον.

Η έξυπνη πόλη μπορεί να υποστηριχθεί σε διάφορους τομείς όπως σε Δημόσιες υπηρεσίες, στην ενέργεια, στην προμήθεια και διάθεση διάφορων υλικών (πχ νερό), ασφάλεια - προστασία, κινητικότητα – μεταφορές, παραγωγή - logistics, εμπόριο, εκπαίδευση, υγεία, πολιτισμός, Διακυβέρνηση. Με την ένταξη και τη δικτύωση όλων των αναφερθέντων περιοχών, αυξάνεται η αποδοτικότητα, η αποτελεσματικότητα και η αντοχή του συνολικού συστήματος, αξιοποιώντας έτσι τις δυνατότητες για την πραγματοποίηση κοινωνικών και οικολογικών βελτιώσεων.

Τα ασύρματα δίκτυα αισθητήρων στις έξυπνες πόλεις λειτουργούν μέσω ενός κατανεμημένου δικτύου ευφυών κόμβων αισθητήρων που μπορεί να μετρήσει πολλές παραμέτρους για την αποτελεσματικότερη διαχείριση της πόλης. Τα δεδομένα παραδίδονται ασύρματα και σε πραγματικό χρόνο για τους πολίτες ή τις αρμόδιες αρχές. Στη συνέχεια αναφέρονται χρήσιμες εφαρμογές τέτοιων δικτύων μέσα στην πόλη:

- Να παρακολουθούν οι πολίτες την συγκέντρωση των ρύπων σε κάθε δρόμο της πόλης.
- Να ανιχνεύονται διαρροές νερού
- Να στέλνεται στην αρμόδια υπηρεσία του δήμου μια ειδοποίηση όταν οι κάδοι απορριμμάτων είναι σχεδόν πλήρεις.
- Έλεγχος κυκλοφορίας των οχημάτων, προκειμένου να τροποποιήσει τα φώτα της πόλης με δυναμικό τρόπο.
- Ανίχνευση πλησιέστερου διαθέσιμου χώρου/θέσης στάθμευσης

- Έγκαιρη ειδοποίηση των οδηγών οχημάτων για εντοπισμό δωρεάν parking.

Υπάρχουν πλατφόρμες διαχείρισης δεδομένων που διατηρούν στοιχεία όλων των συνδεδεμένων αισθητήρων του δικτύου και επικοινωνούν on-line με τις υπηρεσίες βάσεων δεδομένων. Σε αυτές έχουν εγγραφεί και καταχωρηθεί οι αντίστοιχοι αισθητήρες των πολιτών – ιδιοκτητών συσκευών, ενώ ταυτόχρονα έχουν πρόσβαση οι προγραμματιστές ώστε να μπορούν να δημιουργούν νέες εφαρμογές βάσει των δεδομένων που έχουν συγκεντρωθεί. Τέτοια παραδείγματα πλατφόρμας είναι η Xively η πλατφόρμα Wikisensing. [37]

Παραδείγματα έξυπνων πόλεων έχουν εφαρμοστεί ήδη σε αρκετές χώρες όπως στην Βιέννη, Άμστερνταμ, Κάιρο, Ντουμπάι. Στην πόλη Σανταντέρ της Ισπανίας υπάρχουν χιλιάδες αισθητήρες που προειδοποιούν τους κατοίκους για το μποτιλιαρίσματα, ποτίζουν τα δημόσια πάρκα και σβήνουν τα φώτα του δρόμου. Το Σανταντέρ έχει περίπου 180.000 κατοίκους και λειτουργεί ως πείραμα έξυπνης πόλης. Ο καθηγητής πληροφορικής Λουίς Μουνιόθ του Πανεπιστημίου της Καντάμπρια, έλαβε περίπου 9 εκατ. ευρώ από την ΕΕ για να αναπτύξει μια έξυπνη πόλη. Το έργο ολοκληρώθηκε το 2013 και εγκατέστησε 10.000 αισθητήρες σε όλο το κέντρο του Σανταντέρ, μια περιοχή 6 τ. χλμ. Οι αισθητήρες είναι κρυμμένοι μέσα σε γκρι κουτιά τοποθετημένα σε στύλους του ηλεκτρικού, κολώνες και τοίχους, κάποιοι είναι θαμμένοι ακόμη και κάτω από την άσφαλτο. Οι συγκεκριμένοι αισθητήρες μετρούν: φως, πίεση, θερμοκρασία, υγρασία, την κίνηση αυτοκινήτων και ανθρώπων. Έτσι κάθε λεωφορείο μεταδίδει τη θέση και την ταχύτητά του καθώς και στοιχεία για το περιβάλλον, όπως τα επίπεδα του όζοντος ή των ρύπων. Ομοίως τα ταξί και τα περιπολικά. Κάποιος που περιμένει το λεωφορείο δεν έχει παρά να στρέψει το κινητό του προς τη στάση και αμέσως θα δει στην οθόνη ποιες γραμμές λεωφορείων την εξυπηρετούν και σε ποιες ώρες. Αν στρέψει το κινητό του προς την αίθουσα συναυλιών, θα δει το πρόγραμμα των επόμενων ημερών, ενώ αν το κατευθύνει προς ένα σουπερμάρκετ θα δει τις προσφορές της ημέρας. Αν εντοπίσει μια λακκούβα στον δρόμο, αρκεί να την φωτογραφίσει με το κινητό του και με ένα

κλικ τη στέλνει στην αρμόδια υπηρεσία του δημαρχείου με τις συντεταγμένες της περασμένες αυτόματα από το GPS. [38]

Mobile computing

Η κινητή υπολογιστική (Mobile computing) είναι η αλληλεπίδραση ανθρώπου υπολογιστή με την οποία ένας υπολογιστής μεταφέρεται κατά τις συνήθειες του χρήστη του. Η κινητή υπολογιστική περιλαμβάνει την επικοινωνία, τον τεχνικό εξοπλισμό (hardware) και το λογισμικό (software). Τα θέματα επικοινωνίας περιλαμβάνουν ad hoc και δίκτυα υποδομών, καθώς και τις ιδιότητες της επικοινωνίας, πρωτοκόλλων, μορφών δεδομένων και συγκεκριμένες τεχνολογίες. Ο τεχνικός εξοπλισμός περιλαμβάνει τις κινητές συσκευές ή εξαρτήματα της συσκευής. Το κινητό λογισμικό ασχολείται με τα χαρακτηριστικά και τις απαιτήσεις των κινητών εφαρμογών.

Μία κινητή υπολογιστική συσκευή είναι οποιαδήποτε συσκευή που έχει δημιουργηθεί με τη χρήση κινητών στοιχείων. Οι κινητές υπολογιστικές συσκευές είναι φορητές συσκευές που μπορούν να λειτουργούν, να εκτελούν και την παρέχουν υπηρεσίες και εφαρμογές, όπως μια τυπική υπολογιστική συσκευή. Επίσης είναι γνωστές ως φορητές υπολογιστικές συσκευές ή φορητές υπολογιστικές συσκευές. Χρησιμοποιούν το Internet ή intranet και τις αντίστοιχες συνδέσεις επικοινωνίας, όπως WAN, LAN, WLAN κλπ. Οι κινητές συσκευές μπορούν να σχηματίζουν ένα ασύρματο προσωπικό δίκτυο ή ένα piconet .

Υπάρχουν τρεις διαφορετικές κατηγορίες των κινητών στοιχείων υπολογιστών :

- Οι φορητοί υπολογιστές, που είναι ελαφριές συμπαγείς μονάδες που περιλαμβάνουν πληκτρολόγιο και χρησιμοποιούνται κυρίως ως hosts για παραμετροποίηση λογισμικό για παράδειγμα laptops, notebooks, notepads.

- Τα κινητά τηλέφωνα που περιλαμβάνουν πιο περιορισμένο πληκτρολόγιο σε σχέση με τους φορητούς υπολογιστές, και επιτρέπει φωνητικές επικοινωνίες, όπως κινητά τηλέφωνα, έξυπνα τηλέφωνα, rhonepads.
- Οι φορετοί (wearable) υπολογιστές, που περιορίζονται αρκετά σε λειτουργικά πλήκτρα και προορίζονται κυρίως για την ενσωμάτωση πρακτόρων λογισμικού, όπως για παράδειγμα ρολόγια, κολιέ, εμφυτεύματα. [39]

Φορετά συστήματα (wearable's computing)

Οι άνθρωποι στην καθημερινότητα τους συνήθως μεταφέρουν προσωπικά αντικείμενα και πολλές φορές επιδιώκουν να φέρουν μαζί τους πληροφορίες που τους είναι απαραίτητες είτε για κάποιες συνήθειες τους είτε και για την εργασία τους. Έτσι σχεδιάζονται συσκευές οι οποίες προσαρτώνται στο ανθρώπινο σώμα με σκοπό να αυξήσουν τις δεξιότητες του ανθρώπου και να βελτιώσουν την ποιότητα της ζωής του. Αυτές οι συσκευές άρχισαν να γνωρίζουν ραγδαία ανάπτυξη από το 1970 από τον Steve Mann. [40] Ο Steve Mann έθεσε ορισμένα χαρακτηριστικά τα οποία θα πρέπει να έχουν οι διεπαφές και οι υπολογιστές που φοριούνται:

- Ανήκουν στον προσωπικό χώρο του χρήστη
- Ελέγχονται εξολοκλήρου από τον χρήστη
- Όταν φοριούνται είναι πάντα ενεργές και προσβάσιμες από το δίκτυο.

Ο άνθρωπος από την φύση του κάνει πολλά πράγματα ταυτόχρονα (multi-task). Για τον λόγο αυτό έχει ανάγκη για διαρκή πληροφόρηση. Ωστόσο τα ανθρώπινα όρια ξεφεύγουν από τα στενά πλαίσια του γραφείου και ο ίδιος εξακολουθεί να έχει την ίδια ανάγκη για πληροφόρηση ενώ βρίσκεται διαρκώς σε κίνηση. Η ίδια ανάγκη

ώθησε την ανάπτυξη του τομέα των διεπαφών ή των υπολογιστών που φοριούνται (Wearable Computer Interfaces).

Ένας υπολογιστής που φοριέται καθορίζεται σαν ένα πλήρως λειτουργικό, αυτοδύναμο, αυτό-μεταφερόμενο υπολογιστή, ο οποίος παρέχει πρόσβαση σε πληροφορίες, με τις οποίες αλληλεπιδρά, οπουδήποτε και οποτεδήποτε [41]. Οι φορητοί υπολογιστές χρησιμοποιούν διάφορους τύπους αισθητήρων. Οι πληροφορίες που καταγράφονται από αυτούς τους αισθητήρες μπορούν να επεξεργαστούν τοπικά από ενσωματωμένους υπολογιστές και να παράγουν υλικό για την πλοήγηση, την επικοινωνία του χρήστη και τον απομακρυσμένο χειρισμό συσκευών. [42]

Ο στόχος της συγχώνευσης της πανταχού παρούσας υπολογιστικής και των φορητών υπολογιστών είναι η παροχή «για τις σωστές πληροφορίες στο σωστό πρόσωπο στο σωστό μέρος τη σωστή στιγμή». Προκειμένου η πανταχού παρούσα υπολογιστική να είναι αρκετά δυναμική, θα πρέπει ο μέσος άνθρωπος να είναι σε θέση να λάβει τα πλεονεκτήματα των πληροφοριών. Η κινητή πρόσβαση είναι η τεχνολογία που επιτρέπει στις πληροφορίες να είναι διαθέσιμες, σε οποιοδήποτε μέρος και οποιαδήποτε στιγμή. Το υπολογιστικό σύστημα πρέπει να γνωρίζει το περιβάλλον του χρήστη, όχι μόνο να είναι σε θέση να απαντήσει με τον κατάλληλο τρόπο σε σχέση με τη γνωστική και κοινωνική κατάσταση του χρήστη, αλλά επίσης να προβλέπει και τις ανάγκες του.

Ο σχεδιασμός των φορητών ηλεκτρονικών υπολογιστών πρέπει να γίνεται με στόχο να συγχωνεύσουν τις πληροφορίες του χώρου του χρήστη με έναν άλλον χώρο πχ με της εργασίας του. Η φορητή συσκευή θα πρέπει να προσφέρει απρόσκοπτη ενσωμάτωση των εργαλείων επεξεργασίας πληροφοριών με το υπάρχον περιβάλλον εργασίας. Συνεπώς το σύστημα που θα φοριέται προσφέρει λειτουργικότητα με ένα φυσικό και διακριτικό τρόπο, επιτρέποντας στο χρήστη να αφιερώσει όλη την προσοχή του στις εργασίες που γίνονται για παράδειγμα στο χέρι του στο χέρι, χωρίς να αποσπάται η προσοχή του από το ίδιο το σύστημα. Όλες οι μέθοδοι αλληλεπίδρασης (συμπεριλαμβανομένων πληκτρολόγιο, ποντίκι,

joystick, και οθόνη,) απαιτούν κάποια σταθερή φυσική σχέση μεταξύ του χρήστη και της συσκευής, η οποία μπορεί να μειώσει σημαντικά την αποτελεσματικότητα του συστήματος που φοριέται.

Η κοινωνία όλα αυτά τα χρόνια έχει εξελίξει διάφορα εργαλεία και προϊόντα της να είναι πλέον φορητές συσκευές, κινητά, και φορέτα. Η φορητή συσκευή συνεπάγεται τη χρήση του ανθρώπινου σώματος ως ένα περιβάλλον στήριξης για το συγκεκριμένο αντικείμενο. Ρολόγια, ραδιόφωνα και τηλέφωνα αποτελούν παραδείγματα αυτής της τάσης. Ενώ και οι υπολογιστές υποβάλλονται σε μια παρόμοια εξέλιξη. Η απλή συρρίκνωση των υπολογιστικών εργαλείων από τους υπολογιστές εργασίας (desktop) σε μια πιο φορητή κλίμακα δεν επωφελούνται εντελώς από ένα νέο πλαίσιο χρήσης. Γιατί για παράδειγμα μπορεί να είναι δυνατόν να μικρογραφηθούν τα πληκτρολόγια, αλλά η εξέλιξη του ανθρώπου σαφώς δεν συμβαδίζει με τη συρρίκνωση των δακτύλων μας. Υπάρχουν τόσο μικρά μεγέθη που είναι δύσκολος ο χειρισμός τους, οπότε η ανθρώπινη ανατομία εισάγει ελάχιστες και μέγιστες διαστάσεις που ορίζουν το σχήμα των αντικειμένων που φοριούνται. Το κινητό πλαίσιο (mobile context) ορίζει επίσης δυναμικές αλληλεπιδράσεις. Για παράδειγμα ο σχεδιασμός του πληκτρολογίου, του ποντικιού, και των joysticks έχουν ένα συγκεκριμένο σχήμα με βάση την ανατομία του ανθρώπινου χεριού. Στα πλαίσια της σωματικής άνεσης ο σχεδιασμός των αντικειμένων γίνονται με βάση κάποιες οδηγίες και κάποια χαρακτηριστικά στα οποία κατέληξαν οι Gemperle, Kasabach, Stivoric, Bauer, και Martin (1998) μελετώντας φυσιολογία, βίο-μηχανική, τις κινήσεις των σύγχρονων χορευτών και τους αθλητές.

Ως Φορετότητα (Wearability) ορίζεται η αλληλεπίδραση μεταξύ του ανθρώπινου σώματος και του αντικειμένου που φοριέται. Η δυναμική Φορετότητα περιλαμβάνει το ανθρώπινο σώμα όταν βρίσκεται σε κίνηση. Ο σχεδιασμός για την φορετότητα λαμβάνει υπόψη του τη φυσική κατάσταση των αντικειμένων και την ενεργή τους σχέση με την ανθρώπινη μορφή. Τα αποτελέσματα από πολλές έρευνες που έχουν γίνει στην ιστορία και σε διάφορους πολιτισμούς σε θέματα που αφορούν τον ρουχισμό, δίνουν πληθώρα επιλογών φορετών υπολογιστών. Οι ερευνητές

κωδικοποίησαν τα αποτελέσματα σε κατευθυντήριες γραμμές για το σχεδιασμό wearable συστήματων ανάλογα με τις απαιτήσεις του αγοραστή. [43]

Οι φορητοί υπολογιστές επικεντρώνεται κυρίως στην ιατρική, στην εργασία και στην καθημερινή ζωή του ανθρώπου. Συγκεκριμένα στην ιατρική και στην υγεία, σχετίζονται με την σταθερότητα των διεπαφών ώστε να εξασφαλίζεται η άνεση και η ασφάλεια του χρήστη. Στην εργασία, η οποία αποτελεί τον τομέα όπου επικεντρώνεται το μεγαλύτερο ενδιαφέρον και επιδιώκεται η ανάπτυξη συσκευών που θα αποδεσμεύουν τα χέρια των χρηστών ώστε να βελτιωθούν οι δεξιότητές τους και να γίνουν περισσότερο αποδοτικοί. Προκειμένου να βελτιωθεί η καθημερινή ζωή του ανθρώπου, πρέπει να κατασκευαστούν συσκευές που προωθούν την ανάπαυση και βελτίωση του τρόπου ζωής του. [42]

Προς το παρόν δεν είναι γνωστά τα αποτελέσματα από την μακροπρόθεσμη χρήση των φορητών υπολογιστών για το ανθρώπινο σώμα. Τα φορητά συστήματα γίνονται ολοένα και πιο χρήσιμα και αν πρόκειται να χρησιμοποιηθούν για μεγάλο χρονικό διάστημα, είναι σημαντικό να δοκιμαστεί η επίδρασή τους στο ανθρώπινο σώμα.

Οι φορητοί υπολογιστές είναι ένας ελκυστικός τρόπος επικοινωνίας διάχυτων υπολογιστικών συστημάτων και ενός χρήστη, ειδικά όταν εφαρμόζονται σε εξωτερικά περιβάλλοντα. Οι μεγαλύτερες προκλήσεις για τη συγχώνευση της πανταχού παρούσας υπολογιστικής και των φορητών συστημάτων, είναι η επίτευξη αρμονίας μεταξύ ανθρώπου και υπολογιστή από την άποψη της διεπαφής του υπολογιστή με τον άνθρωπο, του γνωστικού μοντέλου, της επίγνωσης του context, και της προσαρμογής στις εργασίες που εκτελούνται. [43]

Ταυτόχρονα οι συσκευές εισόδου και εξόδου δεν θα πρέπει να περιορίζουν και να επιβαρύνουν το χρήστη. Αντίθετα θα πρέπει να τον «ενθαρρύνουν» να τις χρησιμοποιεί προκειμένου να αυξήσει τις δεξιότητές του, να τις χρησιμοποιεί απροβλημάτιστα ανεβάζοντας την παραγωγικότητά του και βελτιώνοντας την ζωή του. Τέλος, ο άνθρωπος μπορεί να επηρεαστεί θετικά και να έχει την πρόθεση να

χρησιμοποιήσει μία καινούρια διεπαφή όταν αυτή μπορεί να ολοκληρωθεί και να ενσωματωθεί σε άλλες διεπαφές των υπάρχόντων τεχνολογιών. [42]

Μερικά παραδείγματα Wearable Συσκευών είναι το έξυπνο δαχτυλίδι για την μετακίνηση των ανθρώπων σε μέσα μαζικής μεταφοράς. Συγκεκριμένα το δαχτυλίδι «Sesame Ring» αναπτύσσεται από δύο φοιτητές του MIT και πρόκειται να αντικαταστήσει τις «έξυπνες» κάρτες διαδρομών, που χρησιμοποιούνται στα μέσα μαζικής μεταφοράς σε πολλές χώρες του εξωτερικού. Η έμπνευση τους ήρθε λόγω του γεγονότος ότι ξεχνούσαν ξανά και ξανά τις κάρτες τους οι κάτοικοι της Βοστώνης. Όπως αναφέρεται στην σελίδα, «έχοντας χάσει το τρένο πολλές φορές ενώ ψάχναμε τις Charlie Cards μας (έξυπνες κάρτες που χρησιμοποιούνται για τις μαζικές μεταφορές στη Μασαχουσέτη), αναζητήσαμε μία λύση στην wearable τεχνολογία. Μετά από μήνες σκληρής δουλειάς, δημιουργήσαμε μέσω τρισδιάστατης εκτύπωσης το Sesame Ring, που υποστηρίζεται από την MBTA». [44]



Εικόνα 25 : Sesame Ring

Άλλο παράδειγμα είναι τα Space Glasses που σε πρώτη φάση χρησιμοποιούνται ως εργαλείο για τους developers που θέλουν να ασχοληθούν με την ανάπτυξη εφαρμογών πάνω στο αντικείμενο, παρά ως καταναλωτικό προϊόν (gadget). Η συσκευή δεν διαθέτει ενσωματωμένη μπαταρία ή επεξεργαστές, οπότε απαιτείται σύνδεση με έναν υπολογιστή. Περιλαμβάνει διάφανη οθόνη LCD για κάθε μάτι,

υπέρυθρη depth camera και κανονική έγχρωμη κάμερα, καθώς και επιταχυνσιόμετρο, γυροσκόπιο και πυξίδα. Η δεύτερη έκδοση αναμένεται να είναι ελαφρύτερη και λιγότερο ογκώδης και να περιέχει τόσο μπαταρία όσο και επεξεργαστές (κεντρικό και γραφικών), καθώς και αλλαγές στο λογισμικό.

Τα γυαλιά χαρτογραφούν το χώρο και τα δεδομένα μεταφέρονται στον υπολογιστή, ο οποίος τα επεξεργάζεται δημιουργώντας ένα τρισδιάστατο μοντέλο του χώρου, οπότε δίνει πολλές δυνατότητες, από την προβολή μίας ταινίας σε ένα κομμάτι χαρτί μέχρι τη δημιουργία ενός τρισδιάστατου αντικειμένου που ακολουθεί το χρήστη. Οι δημιουργοί του συστήματος οραματίζονται κάτι που θα αντικαταστήσει τους συμβατικούς υπολογιστές, το οποίο θα μπορούν να χρησιμοποιούν πολλοί άνθρωποι μαζί και ταυτόχρονα, ανεξαρτήτως του αντικειμένου ενασχόλησής τους. Ακόμα πιο μακροπρόθεσμα, εκτιμάται πως κάποια στιγμή η εν λόγω τεχνολογία θα εξελιχθεί ακόμα περισσότερο, ξεπερνώντας τη μορφή των γυαλιών, και αποκτώντας το χαρακτήρα εμφυτεύματος στο οπτικό νεύρο του χρήστη. [45]



Εικόνα 26 : υπολογιστές κεφαλής Meta Space Glasses

[κενή σελίδα]

4. Context Management σε Έξυπνα περιβάλλοντα



Στην παρούσα ενότητα παρουσιάζεται αναλυτικά η έννοια του πλαισίου (context), του στοιχείου ενός context aware συστήματος (element of a context-aware system) και του συστήματος διαχείρισης του πλαισίου (context management system). Διερευνάται η χρήση και η διαχείριση του πλαισίου μέσα από ένα ερευνητικό σύστημα, το M-Zone που παρουσιάστηκε σε ένα παλαιότερο ερευνητικό έργο, στο οποίο εξετάστηκαν όλα τα στοιχεία που απαιτούνται για τη δημιουργία ενός εξελιγμένου συστήματος διαχείρισης πλαισίου. Στις ακόλουθες υποενότητες περιγράφονται τα οφέλη, οι πιθανές χρήσεις του πλαισίου, τα χαρακτηριστικά και συγκρίσεις διάφορων μοντέλων διαχείρισης, γίνονται αναφορές σχετικά με την ιδιωτικότητα των χρηστών. Τελικά αναφέρονται διάφορα παραδείγματα context-aware συστημάτων.

[κενή σελίδα]

To project M - Zone

Το συγκεκριμένο έργο πραγματοποιήθηκε από την ομάδα Τηλεπικοινωνιακών Συστημάτων Λογισμικού, του Τμήματος Πληροφορικής, Μαθηματικών και Φυσικών Επιστημών του Ινστιτούτου Τεχνολογίας στο Waterford της Ιρλανδίας. Ο κύριος στόχος του έργου ήταν να διερευνήσει βασικά στοιχεία σε νέες υποδομές συστημάτων – περιβαλλόντων, δηλαδή συνεργασία και διαχείριση μεταξύ Έξυπνων Περιβαλλόντων. Με τον όρο «Έξυπνο Περιβάλλον» (Smart Space) εννοείται ο χώρος εργασίας που έχει ενσωματωμένους υπολογιστές, διάφορες συσκευές παροχής πληροφοριών και αισθητήρες, που επιτρέπουν στους ανθρώπους να εκτελέσουν αποτελεσματικά τις διεργασίες τους, προσφέροντας πρωτοφανή επίπεδα πρόσβασης στις πληροφορίες και βοήθειας από τους υπολογιστές.

Το σημαντικότερο σημείο της συγκεκριμένης έρευνας ήταν η συνεργασία και διαχείριση των πληροφοριών μεταξύ των Έξυπνων Περιβαλλόντων. Συγκεκριμένα περιλαμβάνει την συλλογή, ερμηνεία, την αποθήκευση και τη διάδοση όλων των πληροφοριών του πλαισίου με δυναμικό τρόπο και σε πραγματικό χρόνο. Συνεπώς θα πρέπει να παρέχεται αδιάλειπτη προσβασιμότητα στις context-aware υπηρεσίες για μια οντότητα, που κινείται μεταξύ διαφορετικών έξυπνων περιβαλλόντων, είτε αυτή η οντότητα είναι άνθρωπος, είτε κάποια συσκευή. Οπότε για τα Έξυπνα Περιβάλλοντα η οργάνωση της πληροφορίας, δηλαδή η διαχείριση (context management) θα πρέπει να παρέχεται συνεχώς στις context-aware υπηρεσίες.

Στο συγκεκριμένο έργο αρχικά εξέτασε το πλαίσιο και την πρόοδο που έχει γίνει μέχρι και εκείνη την χρονική περίοδο (Μάιος 2003) πάνω στον τομέα της διαχείρισης πλαισίου, με σκοπό να εντοπιστούν τα χαρακτηριστικά του, καθώς και τα στοιχεία που θα πρέπει να περιλαμβάνει ένα πλαίσιο. Στη συνέχεια ενσωματώθηκαν στο συνολικό πλαίσιο διαχείρισης του χώρου.

Επίσης το έργο M- Zone εξέτασε τη δυναμική ανταλλαγή πληροφοριών μεταξύ πλαισίου, ιδιωτικού, ημι-ιδιωτικού και δημόσιου Έξυπνου Χώρου έτσι ώστε να καταφέρει να προσδιορίσει τους διαφορετικούς περιορισμούς και τις επιπτώσεις

της κινητικότητας και της ασφάλειας σχετικά με τη διαχείριση των πληροφοριών πλαισίου.

Τι είναι το context

Όπως αναφέρθηκε και σε προηγούμενες ενότητες ο κύριος στόχος του ubiquitous computing είναι να μειώσει την σκληρή προσπάθεια που απαιτείται για την αξιοποίηση των υπολογιστών ως προς την διευκόλυνση των ανθρώπινων δραστηριοτήτων.

«Η διάχυτη και διεισδυτική υπολογιστική έχει ως στόχο την αύξηση της χρήσης του υπολογιστή κάνοντας πολλούς υπολογιστές διαθέσιμους σε όλο το φυσικό περιβάλλον, αλλά τους καθιστά ουσιαστικά άρατους για τον τελικό χρήστη».

Τα συσχετιζόμενα συστήματα και οι υπηρεσίες πρέπει να συνεργάζονται, να ενοποιήσουν τις πληροφορίες τους καθώς και γενικές πληροφορίες του περιβάλλοντος σχετικά με την κατάσταση του χρήστη, όταν αυτός επιθυμεί να εκτελέσει μια συγκεκριμένη διεργασία έτσι ώστε η συγκεκριμένη διεργασία να εξάγει το μέγιστο όφελος από το υπολογιστικό περιβάλλον.

Αυτή η πληροφορία σχετικά με την κατάσταση του χρήστη ονομάζεται Context ή «πλαίσιο». Πάντα οι άνθρωποι χρησιμοποιούσαν διάφορες πληροφορίες καταστάσεων, ή αλλιώς το «πλαίσιο», για να καταφέρουν να κάνουν πλουσιότερες τις διαπροσωπικές τους αλληλεπιδράσεις. Η χρήση του context από τους υπολογιστές και ειδικά όταν χρησιμοποιείται κατά την αλληλεπίδραση με τους ανθρώπους, προσφέρει χρησιμότερες υπηρεσίες και πληροφορίες στον άνθρωπο – χρήστη. Η σημαντικότερη πρόκληση με τους υπολογιστές που χρησιμοποιούν το context κατά την αλληλεπίδραση του με τους ανθρώπους, είναι ότι δεν υπάρχει κάποιο σίγουρο πρότυπο και επαναχρησιμοποιήσιμο μοντέλο, που να μπορεί να καθορίσει την χρήση του context.

Προκειμένου να ληφθούν οι σωστές πληροφορίες βάσει των επιθυμιών των χρηστών, οι υπολογιστές λαμβάνουν τα απαραίτητα στοιχεία εισόδου από το περιβάλλον τους και προσπαθούν να τα διαχωρίσουν από την γύρω πραγματικότητα. Σε αυτή την προσπάθεια του διαχωρισμού τελικά προκύπτουν κάποιες διαφορές ανάμεσα στο τι προσπαθεί ο υπολογιστής να κάνει και στο τι επιθυμεί ο χρήστης να κάνει για αυτόν ο υπολογιστής του. Η αστοχία προκύπτει από την αδυναμία του υπολογιστή να συγκεντρώσει αρκετές πληροφορίες ώστε καταλάβει τι τελικά επιδιώκει ο χρήστης να κάνει. Το πρόβλημα λύνεται δίνοντας στους υπολογιστές περισσότερες πληροφορίες σχετικές με τον πραγματικό κόσμο των ανθρώπων και ειδικότερα πληροφορίες που έχουν να κάνουν με την καθημερινότητα και τις συνήθειές τους. Η αντιμετώπιση και διαχείριση των δυσκολιών χρήσης του πλαισίου γίνεται από το context management, το οποίο προτείνει την χρήση άλλων πληροφοριών που έχουν εξαχθεί από διαφορετικές υπολογιστικές καταστάσεις. Όταν εφαρμόζεται το πλαίσιο με τις κατάλληλες πληροφορίες και πάντα σύμφωνες με τις τρέχουσες απαιτήσεις, τότε προκύπτουν και νέες εφαρμογές. Συνεπώς το πρώτο βήμα για τη διαχείριση του πλαισίου είναι να καθοριστεί ποιές πληροφορίες αποτελούν το πλαίσιο.

Όπως αναφέρθηκε και σε προηγούμενη ενότητα οι πρωτοπόροι του context-aware computing είναι οι Schilit και Theimer, που ορίζουν το πλαίσιο σαν «τοποθεσία, ταυτότητα ατόμων, αντικείμενων, καθώς και τροποποιήσεις των εν λόγω αντικειμένων». Οι Schilit και Theimer για τη συγκέντρωση των πληροφοριών λαμβάνουν υπόψη την τοποθεσία, δηλαδή που βρίσκεται μια οντότητα, ποιες άλλες οντότητες είναι γύρω της, καθώς και τι άλλοι πόροι βρίσκονται σε κοντινή απόσταση από αυτήν που αποτελούν σημαντικές πληροφορίες για τον εμπλουτισμό του πλαισίου. Ο ερευνητής Abowd Dey έκανε μια επέκταση του ορισμού του πλαισίου και το καθορίζουν ως:

«... Κάθε πληροφορία που μπορεί να χρησιμοποιηθεί για να χαρακτηρίσει την κατάσταση μιας οντότητας. Μια οντότητα είναι ένα άτομο, τοποθεσία ή ένα αντικείμενο που θεωρείται σχετικό με την

αλληλεπίδραση μεταξύ ενός χρήστη και μιας εφαρμογής, συμπεριλαμβανομένου του χρήστη και της εφαρμογής».

(Abowd Dey, Orr, και Brotherlon 1997)

Δηλαδή οποιαδήποτε πληροφορία σχετική με την κατάσταση ενός χρήστη μπορεί να θεωρηθεί περιεχόμενο του πλαισίου. Τέτοια παραδείγματα σχετικών πληροφοριών είναι η θερμοκρασία, η παρουσία κάποιου άλλου προσώπου, οι κοντινές συσκευές, οι φορητές συσκευές του χρήστη και ο προσανατολισμός του χρήστη.

Προκειμένου να εξεταστεί καλύτερα το πεδίο εφαρμογής του πλαισίου, καθώς και η σημασία του πλαισίου στα διεισδυτικά συστήματα, εφαρμόζονται οι ερωτήσεις των «Γιατί» (five Why' s):

- Τι είναι το πλαίσιο;
- Ποιος θα μπορούσε να επωφεληθεί από την γνώση του πλαισίου τους, δηλαδή από ποιους για ποιους είναι σημαντικό το πλαίσιο ή τι;
- Πού μπορεί να αξιοποιηθεί η γνώση του πλαισίου;
- Πότε η γνώση του πλαισίου είναι χρήσιμη;
- Γιατί είναι χρήσιμες οι εφαρμογές που έχουν επίγνωσηπλαισίου

(Morse, Armstrong & Dey 2000)

Στον πραγματικό κόσμο των ανθρώπινων επαφών ο ένας μιλάει απευθείας στον άλλον και χρησιμοποιούν άμεσα και απεριόριστα την πληροφορία, δηλαδή ενισχύεται το εύρος της ομιλίας τους. Στην αλληλεπίδραση ανθρώπου - υπολογιστή δεν συμβαίνει το ίδιο, γιατί αυτή η απεριόριστη πληροφορία πρέπει να μεταφερθεί με φυσικό τρόπο ώστε να γίνει διάλογος ανθρώπου – υπολογιστή (Dey 2002). Οπότε προέκυψε η ιδέα η context aware πληροφορική να εκμεταλλευτεί την πρόοδο των αισθητήρων και μηχανισμών, ώστε να παρατηρεί το περιβάλλον και να συλλέγει συστηματικά απαραίτητες πληροφορίες για το πλαίσιο (Abowd et al 2002).

Η πληροφορία του πλαισίου μπορεί να διαμορφωθεί σε ένα αφηρημένο μοντέλο από όλους τους συμμετέχοντες ενός έξυπνου περιβάλλοντος. Το σύστημα χαρακτηρίζεται ως context-aware εφόσον χρησιμοποιεί τις πληροφορίες του αφηρημένου μοντέλου, με σκοπό να παρέχει τις σχετικές πληροφορίες και υπηρεσίες προς τον χρήστη. Οι υπηρεσίες των context-aware συστημάτων αποτελούνται από τη σύνθεση αιτήσεων και αποφάσεων ώστε να εκτελούν ενέργειες για λογαριασμό του χρήστη με φιλικό τρόπο προς αυτόν. Οι συνειδητές αποφάσεις που μπορούν να παίρνουν τα context-aware συστήματα παρέχονται έπειτα από εντολές που έχουν λάβει από τους χρήστες και παρουσιάζονται με τον κατάλληλο τρόπο σε αυτούς. Οι Schilit και Theimer θεωρούν ότι context-aware είναι κάθε σύστημα που «προσαρμόζεται ανάλογα με τη θέση χρήσης του, τη συλλογή πληροφοριών των κοντινών ανθρώπων και των αντικειμένων, καθώς και τις αλλαγές όλων αυτών με την πάροδο του χρόνου» (Schilit, Adams & Want 2002).

Η διατήρηση ιστορικών στοιχείων του πλαισίου είναι ιδιαίτερα σημαντική και λαμβάνεται υπόψη κάθε χρήσιμη πληροφορία που έχει αποθηκευτεί και μπορεί να χρησιμοποιηθεί. Η καταγραφή στοιχείων λαμβάνει πληροφορίες από τους υπολογιστικούς πόρους, τον χρήστη και το φυσικό περιεχόμενο που αποθηκεύονται σε μια χρονική στιγμή. Όταν χρησιμοποιούμε αυτήν την αποθηκευμένη πληροφορία μπορούμε να προσδιορίσουμε τις τάσεις για την έξυπνη χρήση του περιβάλλοντος. Επιπλέον τα ιστορικά στοιχεία του πλαισίου είναι ιδιαίτερα χρήσιμα για mobile-aware εφαρμογές. Τέτοια παραδείγματα εφαρμογών αφορούν την πρόβλεψη και κατανάλωση των πόρων του περιβάλλοντος, που στηρίζονται σε διάφορα πρότυπα που παρατηρούν την κίνηση στον χώρο και τη χρήση του. Ο William Su και άλλοι ερευνητές περιγράφουν μια μέθοδο πρόβλεψης κινητικότητας σχετική με την ιστορικότητα ως εξής:

«Η Ιστορικότητα του πλαισίου θεωρείται χρήσιμη, αλλά σπανίως χρησιμοποιείται στα τρέχοντα context-aware συστήματα. Πρώτα από όλα, το context-aware σύστημα πρέπει να αποφασίσει τι ιστορικές πληροφορίες αξίζει να διατηρήσει και μέχρι ποιο επίπεδο ακρίβειας.»

Στη συνέχεια αφού αξιολογήσει όλες τις ιστορικές πληροφορίες που συνέλεξε από το περιβάλλον, θα πρέπει να διατηρήσει μόνο τις χρήσιμες πληροφορίες, κάτι όμως που θα ήταν πάρα πολύ δαπανηρό καθώς χρειάζονται αποδοτικοί αλγόριθμοι, οι οποίοι πρέπει να υλοποιηθούν επεξεργάζοντας όλο των όγκο των πληροφοριών προκειμένου να εξαχθούν δεδομένα που να έχουν νόημα».

(Su, Lee & Gerla 2002)

Κατηγορίες Πλαισίου

Η αποτελεσματικότητα επίδοσης ενός πλαισίου έχει νόημα και γίνεται πιο ισχυρή μόνο όταν αυτό είναι κοινόχρηστο. (Winograd 2001). Το context-aware σύστημα θα πρέπει πρώτα να καταλάβει τι είναι το πλαίσιο, ώστε να καταφέρει να αναζητήσει και κατηγοριοποιήσει αυτές τις πληροφορίες και αφού συλλέξει και διαχειριστεί τις πληροφορίες, τότε να γίνει κοινόχρηστο. Οι ερευνητές Schilit, Adams και Want (1994) προτείνουν την ακόλουθη ταξινόμηση πληροφοριών πλαισίου:

-
- Υπολογιστικό Context (Computing Context): αφορά τις συνδέσεις του δικτύου, το εύρος ζώνης και τους κοντινούς πόρους, όπως εκτυπωτές, οθόνες, ή σταθμούς εργασίας.
- Context του χρήστη (User Context): αφορά το προφίλ του χρήστη, την τοποθεσία, ότι βρίσκεται κοντά στους ανθρώπους και την τρέχουσα κοινωνική κατάσταση.
- Φυσικό Context (Physical Context): αφορά το επίπεδο φωτισμού, το επίπεδο θορύβου, τις κυκλοφοριακές συνθήκες και τη θερμοκρασία.
- Οι παραπάνω κατηγορίες περιέχουν πλούσιες πληροφορίες σχετικά με το context-aware σύστημα, όμως δεν μπορούν να χρησιμοποιηθούν μεμονωμένα και να παρέχουν πλήρη αποτελέσματα. Συνεπώς όλα τα πλαίσια θα πρέπει να

συνδυαστούν, δηλαδή το Computing Context θα πρέπει να συνδυαστεί με User Context, αντίστοιχα το User Context θα πρέπει να συνδυαστεί με το Physical Context, έτσι ώστε να επιτευχθεί μια πλήρης εικόνα για το context-aware σύστημα.

Φυσικό και Εικονικό Πλαίσιο

Συνοψίζοντας τα παραπάνω το πλαίσιο μπορεί να υποδιαιρεθεί σε δύο βασικές κατηγορίες: Φυσικό και Εικονικό Πλαίσιο. Επειδή το context-aware σύστημα πρέπει να συγκεντρώσει πληροφορίες τόσο του φυσικού όσο και του εικονικού πλαισίου, στη συνέχεια να συγχωνεύσει τα δύο πλαίσια σε ένα, ώστε τελικά να πετύχει μια συνολική εικόνα της υπάρχουσας κατάστασης.

Στο Εικονικό Πλαίσιο περιέχεται η έκδοση του λειτουργικού συστήματος, οι δυνατότητες της διεπαφής, η ασύρματη τεχνολογία που χρησιμοποιείται για την επικοινωνία, τα μηνύματα (εισερχόμενα και εξερχόμενα) του ηλεκτρονικού ταχυδρομείου, καθώς και τα επεξεργασμένα έγγραφα.

Το Φυσικό Πλαίσιο περιλαμβάνει την παρουσία μιας άλλης οντότητας (όπως αναφέρθηκε οντότητα είναι είτε ένας χρήστης είτε μια συσκευή), για παράδειγμα ένας κοντινός εκτυπωτής, ή πληροφορίες σχετικές με την θέση του χρήστη αν στέκεται αν κινείται, ή και για τις υπάρχουσες καιρικές συνθήκες.

Με την ενοποίηση των δύο πλαισίων προκύπτει μια διαφορετική λογική σχετικά με την κατάσταση του χρήστη. Στο παρελθόν λήψη πληροφοριών του context επικεντρωνόταν στο φυσικό πλαίσιο, δίνοντας έμφαση στην τοποθεσία, ενώ τώρα δίνεται περισσότερη σημασία στο εικονικό πλαίσιο, γιατί διατίθεται μεγαλύτερο φάσμα δυνατοτήτων.

Ένα παράδειγμα συστήματος που χρησιμοποιεί εικονικό πλαίσιο είναι το Kimura (Voیدا 2002) που χρησιμοποιεί συστήματα παρακολούθησης των Microsoft Windows, συγκεντρώνει τις πληροφορίες πλαισίου και δημιουργεί το εικονικό πλαίσιο από την επιφάνεια εργασίας του χρήστη. Το Kimura διαχειρίζεται το εικονικό πλαίσιο χρησιμοποιώντας χαρακτηριστικά των hooks μέσα από το Win32 API (Voیدا 2002). Υπάρχουν και τα συστήματα που συλλέγουν δεδομένα βάσει φυσικού πλαισίου, όπου το βασικό στοιχείο είναι η τοποθεσία. Τέτοια παραδείγματα είναι το Active Badge System (Want. 1992), το Cricket Location System (Priyantha, Chakraborty, & Balakrishnan 2000), το Randar (Bahl και Padmanabhan 2000) και το οπτικό σύστημα υπολογιστών EasyLiving (Shafer, Brumitt & Meyers 2000). Εφόσον το πλαίσιο συγκεντρώσει όλα αυτά τα δεδομένα και τα αποθηκεύσει σε ένα προσωρινό χώρο, τελικά ένα πιο εξελιγμένο σύστημα τα συνενώνει ώστε τελικά παρθεί η απόφαση για το τι είναι σχετικό με την παρούσα φάση του χρήστη.

Χαρακτηριστικά Πλαισίου

Τα χαρακτηριστικά που είναι επιθυμητά για το καλύτερο σχεδιασμό ενός context-aware συστήματος, είναι αυτά που διασφαλίζουν ότι το context-aware σύστημα μπορεί να διαχειριστεί το πλαίσιο με τον πιο αποτελεσματικό και αποδοτικό τρόπο. Οι ερευνητές Candolin Catharina και Kari Hannu H. (2002) από το Πανεπιστήμιο του Queensland έκαναν την ακόλουθη ταξινόμηση των χαρακτηριστικών πλαισίου :

1. Η πληροφορία του Context παρουσιάζεται σε ένα φάσμα προσωρινών χαρακτηριστικών

Εδώ η πληροφορία του πλαισίου έχει ήδη υποδιαιρεθεί σε φυσικές και εικονικές πληροφορίες. Υποστηρίζεται η υποδιαίρεση και σε περαιτέρω πληροφορίες όπως σε στατικές και δυναμικές. Οι στατικές πληροφορίες είναι όλες οι πληροφορίες που δεν αλλοιώνονται και σχετίζονται με το περιβάλλον του χρήστη και μπορούν να ανακτηθούν απ ευθείας από αυτόν. Οι δυναμικές πληροφορίες θα πρέπει να συσσωρεύονται αδιάλειπτα, δηλαδή σε πολύ τακτά χρονικά διαστήματα και με

αυτοματοποιημένο τρόπο. Οι παρελθοντικές πληροφορίες του πλαισίου διατηρούνται γιατί μπορεί να χρειαστούν για την καλύτερη κατανόηση της κατάστασης του περιβάλλοντος. Η συντριπτική πλειοψηφία των πληροφοριών πλαισίου είναι δυναμικές.

2. Οι πληροφορίες του πλαισίου είναι ελλιπείς

Για να είναι έγκυρο ένα πλαίσιο θα πρέπει να μην έχει ελλιπείς πληροφορίες και η εγκυρότητα του αφορά τις δυναμικές πληροφορίες του πλαισίου. Οι αμφιβολίες ορθότητας του πλαισίου προκύπτουν κυρίως από δύο λόγους: από την ταχύτητα με την οποία αλλάζουν οι πληροφορίες του πλαισίου και την ανάγκη επεξεργασίας τους σε μετέπειτα φάση προτού αυτές χρησιμοποιηθούν. Οπότε δημιουργείται μια καθυστέρηση μεταξύ της παραγωγής και της χρήσης πληροφοριών που αφορούν το πλαίσιο, (Candolin & Kari 2002) και αυτή η καθυστέρηση δημιουργεί τις αμφιβολίες ανησυχία για την εγκυρότητα του πλαισίου. Ένας ακόμη παράγοντας που σχετίζεται με την εγκυρότητα των πληροφοριών του πλαισίου, είναι η αξιοπιστία των στοιχείων που παράγουν τις πληροφορίες του πλαισίου. Τέτοια στοιχεία είναι για παράδειγμα, οι αισθητήρες και σε περίπτωση μιας «χαλασμένης» διαδρομής επικοινωνίας μεταξύ των στοιχείων παραγωγής του πλαισίου και του σημείου που χρησιμοποιείται από το χρήστη, οδηγεί τον χρήστη να μην μπορεί να χρησιμοποιεί πρόσφατες και ενημερωμένες πληροφορίες.

3. Οι πληροφορίες έχουν πολλές εναλλακτικές αναπαραστάσεις

Τόσο στο φυσικό όσο και στο εικονικό πλαίσιο υπάρχει εμφανής διαφορά μεταξύ των πρωτογενών ακατέργαστων δεδομένων (raw data) που συλλέγονται από το περιβάλλον, και των επεξεργασμένων πληροφοριών που χρησιμοποιούνται από τον τελικό χρήστη. Τα αρχικά ακατέργαστα δεδομένα μπορούν να πάρουν πολλές μορφές, ειδικά όταν συνδυάζονται και με άλλες πληροφορίες του περιβάλλοντος, και μετά την επεξεργασία τους από το context-aware σύστημα αλλάζουν ακόμη πιο πολύ. Η πιθανότητα επιτυχίας για την σωστή καταγραφή των σχέσεων που δημιουργούνται μεταξύ των διαφορετικών αναπαραστάσεων του context-aware συστήματος και των πληροφοριών αγγίζει το 100%).

4. Οι πληροφορίες του πλαισίου είναι ιδιαίτερα αλληλένδετες

Όσες πληροφορίες προέρχονται από συγκεκριμένη προέλευση μπορεί να σχετίζονται στενά με την πηγή τους, τόσο πολύ ώστε τελικά να εξαρτώνται από την αρχική προέλευση. Συνεπώς όταν τα χαρακτηριστικά πληροφοριών που προκύπτουν από τέτοιες αλληλένδετες σχέσεις με τις ιδιότητες των πληροφοριών προέλευσης, τελικά το πλαίσιο μπορεί να μην είναι και τόσο αξιόπιστο.

Προϋποθέσεις Χαρακτηριστικών Context-Aware System

Επειδή αυτά τα συστήματα όπως αναφέραμε εντάσσονται μέσα στο περιβάλλον του χρήστη με τέτοιο τρόπο ώστε να μην γίνονται αντιληπτά, το context-aware σύστημα θα πρέπει να είναι ικανό να μιμείται την ανθρώπινη ικανότητα στο να αναγνωρίσει και να αξιοποιήσει την άρρητη πληροφορία του περιβάλλοντος, προκειμένου να προάγει σε ανώτερο επίπεδο τις εργασίες των λειτουργιών του. Οπότε είναι σημαντικό οι context-aware εφαρμογές να λειτουργούν με τέτοιο τρόπο που να μεταφέρουν τις κατάλληλες πληροφορίες στο σωστό μέρος και τη σωστή στιγμή μέσω της συνολικής πρόθεσης του χρήστη. Οι στόχοι ενός context-aware συστήματος είναι:

1. Να συγκεντρώνει είτε τις πληροφορίες από το περιβάλλον είτε από την κατάσταση του χρήστη.
2. Να μεταφράζονται οι πληροφορίες στην κατάλληλη μορφή.
3. Να μπορεί να συνδυάσει πληροφορίες, οι οποίες δημιουργούν ένα υψηλότερο πλαίσιο περιεχομένου. Ένα τέτοιο πλαίσιο είναι το πλαίσιο πληροφοριών που προκύπτει από την συγχώνευση άλλων πληροφοριών του πλαισίου.
4. Να μπορεί αναλάβει δράση αυτόματα με βάση τις πληροφορίες που έχει ανακτήσει.

5. Να κάνει προσβάσιμες τις πληροφορίες στον χρήστη είτε άμεσα, είτε μελλοντικά, είτε όταν απαιτείται, προκειμένου να ενισχύσουν και να βοηθήσουν στην ολοκλήρωση των εργασιών του χρήστη.

Οι ερευνητές του Πανεπιστημίου Berkeley (Dey Anind 2000) που έφτιαξαν το μοντέλο Context Toolkit προτείνουν την ύπαρξη τριών κατηγοριών που μπορεί να υποστηρίξει ένα context-aware σύστημα:

- Παρουσίαση των πληροφοριών και υπηρεσιών σε ένα χρήστη
- Αυτόματη εκτέλεση μιας υπηρεσίας για ένα χρήστη
- Τοποθέτηση σήμανση (tagging) από το πλαίσιο στις αντίστοιχες πληροφορίες για την μετέπειτα ανάκτηση τους.

Αντίστοιχα οι ερευνητές του συστήματος Kimura το έφτιαξαν ώστε να ενσωματώνει ανεξάρτητα εργαλεία μέσα σε ένα διεισδυτικό υπολογιστικό σύστημα. Συγκεκριμένα έχουν σχεδιάσει με κατανομημένο τρόπο τα συστατικά του. Τα στοιχεία των συστατικών κατατάσσονται σε τρεις κατηγορίες (Voita Steve 2002):

- Απόκτηση πλαισίου: το σύστημα συλλέγει τις πληροφορίες από το περιβάλλον και τις τοποθετεί σε μια αποθήκη
- Ερμηνεία πλαισίου: το σύστημα μετατρέπει τις συγκεντρωμένες πληροφορίες σε ένα πλαίσιο εργασίας
- Αλληλεπίδραση με τον χρήστη: το σύστημα εμφανίζει το πλαίσιο εργασίας στον χρήστη.

Βασικά Στοιχεία του Context -aware Computing

Προκειμένου να έχουμε μια προηγμένης μορφής εφαρμογή, δηλαδή μια context-aware εφαρμογή, πέρα από μια συνηθισμένη απλή και «τυπική» εφαρμογή, θα πρέπει να εφαρμόζονται τα βήματα για την λήψη χαρακτηριστικών των πληροφοριών που περιγράφονται στην προηγούμενη ενότητα. Έτσι προκύπτει η

ανάγκη για την σχεδίαση ενός μοντέλου πλαισίου, ώστε να βελτιστοποιηθούν τα οφέλη των πληροφοριών που αποκτήθηκαν από την προεργασία του πλαισίου στην εφαρμογή. Αυτό το μοντέλο διαχείρισης θα πρέπει να χειριστεί το πλαίσιο με τέτοιο τρόπο ώστε να είναι επαναχρησιμοποιήσιμο και να επιτρέπει σε μια πηγή του πλαισίου να αξιοποιείται από πολλές διαφορετικές εφαρμογές που εκτελούν ποικίλες εργασίες.

Οι εφαρμογές του context-aware συστήματος ανακαλύπτουν και επωφελούνται από τις συναφείς πληροφορίες, όπως για παράδειγμα την τοποθεσία, την ώρα, την ημερομηνία, τους ανθρώπους, τις συσκευές και την δραστηριότητα τους. Αυτό το σύστημα είναι το καταλληλότερο για τους τομείς της κινητής τηλεφωνίας και διεισδυτικών συστημάτων. Σε αντίθεση με άλλα συστήματα όπως τα προσαρμοσμένα συστήματα και εφαρμογές που δεν είναι εμφανής η κινητικότητα χρηστών, τα context-aware συστήματα μπορούν να υποστηρίξουν εφαρμογές για τα mobile – aware.

«Τα Context –aware υπολογιστικά συστήματα χαρακτηρίζονται από τη χρήση των χαρακτηριστικών του περιβάλλοντος, όπως η τοποθεσία του χρήστη, η ώρα, η ταυτότητα και η δραστηριότητα, ώστε να ενημερώνεται η αντίστοιχη υπολογιστική συσκευή, με τέτοιο τρόπο που να μπορούν να παρέχουν πληροφορίες στον χρήστη που σχετίζονται με το πρόσφατο context».

(Jenna Burrell and Geri K. Gay 2001)

Όπως αναφέρθηκε και σε προηγούμενη παράγραφο οι εταιρίες XEROX PARC και Olivetti Research Ltd ήταν πρωτοπόροι σχεδιαστές εφαρμογών για context – aware computing.

Βέβαια αυτό εξελίχθηκε και πλέον το context -aware σύστημα περιλαμβάνει επιπλέον στοιχεία σε σχέση με τον αρχικό σχεδιασμό:

- Η τεχνολογία αισθητήρων (Sensor technology)

Πολλές συσκευές τεχνικού εξοπλισμού (hardware) πρέπει να έχουν την δυνατότητα να συλλέξουν πληροφορίες που θα αποτελέσουν μέρος για το σχεδιασμό του πλαισίου του συστήματος. Αυτές οι συσκευές θα πρέπει να είναι σχετικά φθηνές, εύκολα διαθέσιμες, και ενδεχομένως θα πρέπει να μην απαιτούν πολλές ρυθμίσεις και πολλά βήματα στην διαχείριση τους. Τέλος θα πρέπει να μπορούν να μεταδίδουν τις σχετικές πληροφορίες σε κάποια κεντρική τοποθεσία, ή να επικοινωνούν με άλλες κοντινές συσκευές.

➤ Μοντέλο Πλαισίου (Context model)

Οι πληροφορίες του μοντέλου πλαισίου πρέπει να διαμορφωθούν με τον κατάλληλο τρόπο ώστε να παρέχονται σε έναν πόρο, σε μια πηγή με σκοπό να επωφεληθούν από αυτό διάφορες εφαρμογές. Ερευνητικά έργα όπως το Cameleon (<http://giove.cnuce.cnr.it/cameleon.html>), κατασκευάζουν μοντέλα πληροφοριών πλαισίου σχετικά με τις εφαρμογές. Δηλαδή με τον τρόπο που «οι εκδόσεις τροφοδοτούνται με την ικανότητα να αντιδρούν δυναμικά στις αλλαγές του πλαισίου, όπως τη συνδεσιμότητα του δικτύου, την τοποθεσία του χρήστη και τους ήχους του περιβάλλοντος και τις συνθήκες φωτισμού».

➤ Συστήματα Αποφάσεων (Decision systems)

Εφόσον διαμορφωθεί το μοντέλο πλαίσιο, τα στοιχεία του συστήματος θα πρέπει να παίρνουν αποφάσεις βάσει των διαθέσιμων πληροφοριών. Οι αποφάσεις μπορούν να λαμβάνονται είτε σε επίπεδο εφαρμογής, δηλαδή σύμφωνα με τις πληροφορίες που διαθέτει η εφαρμογή, είτε γίνονται σε κεντρικά σημεία και στη συνέχεια διαδίδονται στα επιμέρους στοιχεία του συστήματος.

➤ Υποστήριξη Εφαρμογής (Application support)

Οι προγραμματιστές των εφαρμογών του πλαισίου θα πρέπει να γνωρίζουν τις πληροφορίες του περιβάλλοντος και όποιων άλλων επιπρόσθετων σχετικών πληροφοριών, οι οποίες θα επηρεάσουν ριζικά την λειτουργία των εφαρμογών.

Ενώ παλαιότερα οι εφαρμογές ενεργούσαν μόνο με ρητή εντολή του χρήστη, πλέον ενεργούν με δική τους «πρωτοβουλία», όμως με τέτοιο τρόπο ώστε να μην προκαλέσουν τυχόν δυσάρεστες εκπλήξεις προς τον χρήστη. Το ιδανικό θα ήταν η αλληλεπίδραση του χρήστη με μια context-aware εφαρμογή, να είναι όσο το δυνατόν απλούστερη και πιο παραγωγική σε σχέση με μια παραδοσιακή εφαρμογή (Keith Cheverst, Nigel Davies, Keith Mitchell, και Christos Efstratiou 2000).

Συλλογή Πληροφοριών Πλαισίου

Κάποιες πληροφορίες του πλαισίου δίνονται με απόλυτη σαφήνεια από την αρχή στο context-aware σύστημα, όπως για παράδειγμα το όνομα ή ηλικία του χρήστη. Κάποιες άλλες πληροφορίες που αφορούν το περιβάλλον μπορούν να συγκεντρωθούν με την χρήση των αισθητήρων. Υπάρχουν γύρω μας κοινοί τύποι αισθητήρων που παρέχουν ακατέργαστες φυσικές πληροφορίες, όπως για παράδειγμα το φως, τη θερμότητα και την ανάγνωση πίεσης. Όμως υπάρχουν και άλλοι τύποι αισθητήρων που υποστηρίζουν πιο εξειδικευμένες πληροφορίες του context, όπως η αναγνώριση προσώπου, που βασίζεται σε κάμερα. Μπορεί αυτός ο αισθητήρας της κάμερας να είναι απλός, αλλά απαιτεί σημαντική επεξεργασία δεδομένων. Δηλαδή για το συγκεκριμένο παράδειγμα απαιτείται η αναγνώριση και επεξεργασία εικόνας, προκειμένου να γίνει χρήση των πληροφοριών που λαμβάνονται.

Οι πληροφορίες που αφορούν την τοποθεσία και την ταυτότητα είναι τα πιο συχνά ανιχνεύσιμα κομμάτια μέσα στο πλαίσιο. Τα Ενεργά Σήματα - Active Badges (Want et al 1992) που παράγονται από την Olivetti και την AT&T εκπέμπουν υπέρυθρα σήματα που δίνουν μια γενική θέση δηλαδή την τοποθεσία και την ταυτότητα του χρήστη. Τα δε Οπτικά συστήματα χρησιμοποιούνται για καλύτερο προσδιορισμό του πλαισίου. Η έρευνα βρίσκεται σε συνεχή εξέλιξη γύρω από τις περιοχές της οπτικής ανίχνευσης, εντοπισμού και κίνησης, της στερεοφωνικής και της 3D

ανακατασκευής, καθώς και της αναγνώρισης αντικειμένων. Η πληροφορία της τοποθεσίας είναι πολύ σημαντικό στοιχείο των πληροφοριών πλαισίου. Κατά καιρούς έχουν γίνει πολλές διαφορετικές προσεγγίσεις, για την τον προσδιορισμό της θέσης με την χρήση διάφορων πρακτόρων (agent) σε ένα context-aware σύστημα. Ιδιαίτερα το GPS, οι υπέρυθρες και τα ραδιοφωνικά σήματα έχουν διερευνηθεί για να χρησιμοποιηθούν στα context-aware συστήματα. Έχουν κατασκευαστεί πολλά συστήματα context-aware γνωρίζοντας μόνο πληροφορίες που αφορούσαν την τοποθεσία. Παρόλο που αυτές οι εφαρμογές είναι πολύ χρήσιμες, η πληροφορία της τοποθεσίας αποτελεί μόνο ένα στοιχείο από το ευρύτερο πλαίσιο και δεν μπορούμε να αρκεστούμε μόνο σε αυτό εφόσον επιθυμούμε μια προηγμένης μορφής εφαρμογή.

«Οι αισθητήρες δεν είναι 100% ακριβείς ή αξιόπιστοι, ιδίως εάν δεν είναι μιας χρήσης. Οι πληροφορίες του Συστήματος Συλλογής πρέπει να έχουν ανοχή σε αισθητηριακές βλάβες, καθώς και όλες οι πληροφορίες που συγκεντρώθηκαν από τους αισθητήρες θα πρέπει να υποβάλλονται σε λογικούς ελέγχους για να βοηθήσουν την εξακρίβωση της ορθότητας. Η συγχώνευση αισθητήρων είναι μια μέθοδος αποφυγής τέτοιων δυσκολιών».

(Schmidt, Beigl & Gellersen 1999)

Με τον όρο συγχώνευση αισθητήρων, εννοούμε την συγκέντρωση και άθροιση των αποτελεσμάτων που συγκεντρώθηκαν από όλους και τους διαφορετικούς αισθητήρες, ώστε να πάρουμε μια πληροφορία με λογική προσέγγιση της κατάστασης του συστήματος. Δηλαδή ακόμη και αν ορισμένοι αισθητήρες αποτύχουν προς την λειτουργία τους ή δώσουν λανθασμένες απαντήσεις, το σύστημα θα εξακολουθεί να είναι σε θέση να προσδιορίσει την πραγματική κατάσταση με τη χρήση ενός μηχανισμού ψηφοφορίας. Για παράδειγμα κατά την εξέταση των αποτελεσμάτων εξόδου των αισθητήρων θερμοκρασίας, θα μπορούσε να είναι διατηρηθεί είτε ο μέσος όρος των αποτελεσμάτων ή, εναλλακτικά, να απορρίψει τις αναφερόμενες τιμές αν αυτές διαφέρουν πάρα πολύ από τις τιμές των άλλων αισθητήρων. Με την χρήση αυτής της μεθόδου τελικά το σύστημα

διαχείρισης πλαισίου μπορεί να διορθώνει ότι θεωρεί ως διακυμάνσεις της θερμοκρασίας, παρόλο που στην πραγματικότητα είναι απλά ένα αποτέλεσμα αποτυχίας του αισθητήρα. Οι ερευνητές Hans-Werner Gellersen, Albrecht Schmidt και Michael Beigl έκαναν ένα ερευνητικό έργο σχετικά με την σύνθεση-αισθητήρων σε context aware συστήματα που αφορούσαν κινητές συσκευές και μικροαντικείμενα (Οκτώβριος 2002).

Η ανακάλυψη των αισθητήρων είναι ένα σημαντικό θέμα για τον συγκεκριμένο τομέα. Για να μπορέσει ένα τέτοιο σύστημα να είναι αποδοτικό στις ανάγκες των χρηστών, απαιτείται η χρήση πολλών και διαφορετικών αισθητήρων, παρόλο που η ρύθμιση και η διαχείριση ενός μεγάλου αριθμού αισθητήρων θα είναι ένα δύσκολο έργο. Υπάρχει μεγάλη πιθανότητα σε ένα τέτοιο σύστημα, το πλήθος των αισθητήρων να ξεπερνά το πλήθος των ανθρώπων, και κατά συνέπεια να προκύψει η ανάγκη για αυτοματοποίηση της ανίχνευσης και ρύθμισης των αισθητήρων. Το πρόγραμμα MUSE των Paul Castro και Richard Muntz χρησιμοποιεί JINI(tm) Connection Technology για την αυτόματη ανίχνευση αισθητήρα (Οκτώβριος 2002).

Ανάκτηση Πλαισίου

Η απόκτηση πληροφοριών σε ένα context-aware σύστημα από τις πηγές πλαισίου, μπορεί να γίνει με δύο μεθόδους: Ώθηση και η Έλξη (Push and Pull). Με την μέθοδο του μοντέλου ώθησης (push) οι πληροφορίες πλαισίου αποστέλλονται περιοδικά στην εφαρμογή. Δηλαδή πριν να είναι απαραίτητες οι πληροφορίες, η πηγή πλαισίου τις συλλέγει, και έτσι οδηγεί την εφαρμογή σε καλύτερη απόδοση. Υπάρχει ένα μειονέκτημα σε αυτή την προσέγγιση σχετικά με την κατανάλωση των πόρων που απαιτούνται για τη συλλογή και τη διάδοση πλαισίου, επειδή τελικά μπορεί αυτές οι πληροφορίες να μην αξιοποιηθούν ποτέ από την υπηρεσία περιβάλλοντος. Οπότε αυτή η μέθοδος συγκεντρώνει μόνο τις πληροφορίες πλαισίου που απαιτείται από την υπηρεσία. Βέβαια η συγκεκριμένη προσέγγιση εκθέτει την υπηρεσία πλαισίου σε προβλήματα στο δίκτυο όπως οι καθυστερήσεις

και η έλλειψη διαθεσιμότητας του δικτύου. Η άλλη μέθοδος του μοντέλου έλξης (pull), υποστηρίζει την ανταλλαγή μεταξύ της ανανέωσης των πληροφοριών πλαισίου και με το κόστος των συχνών ενημερώσεων που πρέπει να μετράται, όταν η προσέγγιση έλξης πρέπει να εξετάζει την ανταλλαγή μεταξύ πόρων που διατηρούνται και την πιθανότητα σφαλμάτων (Ebling Maria, Hunt Guerney και Lei Hui 2002).

Μοντέλα Διαχείρισης Πλαισίου

Εκτός από τα χαρακτηριστικά πληροφοριών που αναφέρθηκαν σε προηγούμενη παράγραφο, σημαντικό βήμα για τη διαχείριση πλαισίου είναι οι context-aware εφαρμογές να παρέχουν κατάλληλα στοιχεία λογισμικού που θα παρέχουν πρόσβαση σε πληροφορίες του περιβάλλοντος. Υπάρχουν τρία μοντέλα με δομημένες προσεγγίσεις για την παροχή πλαισίου στην εφαρμογή:

- Widgets Πλαισίου
- Μοντέλο Υποδομών (Infrastructure)
- Μαυροπίνακα (Blackboard)

Widgets Πλαισίου (Context Widgets)

Τα widgets πλαισίου είναι το πιο διαδεδομένο μοντέλο και βρίσκεται μεταξύ της context-aware εφαρμογής και του περιβάλλοντος, συγκρίσιμο με τον τρόπο του γραφικού widget GUI που υπάρχει μεταξύ της εφαρμογής και του χρήστη. Ο κύριος στόχος του widget είναι ο διαχωρισμός της εφαρμογής από την ανάκτηση πληροφοριών από το πλαίσιο. Ο συγκεκριμένος διαχωρισμός έχει μεγάλη πολυπλοκότητα στη συλλογή και διαχείριση πληροφοριών πλαισίου. Οπότε η προσέγγιση και τα προβλήματα που σχετίζονται με τη συσσώρευση πληροφοριών του πλαισίου είναι ασήμαντα ή ίσως αμελητέα, ώστε να επηρεάζουν τελικά την εφαρμογή. Επιπλέον το context widget χρησιμοποιείται και ως μεσολαβητής για να

περάσει μόνο τις σχετικές πληροφορίες που αφορούν την εφαρμογή. Οποιαδήποτε άλλη πληροφορία που δεν αφορά την εφαρμογή δεν συσσωρεύεται στο πλαίσιο. Τέλος, το context widget διαθέτει μια επαναχρησιμοποιήσιμη δομική μονάδα (building block), η οποία επιτρέπει σε πολλές εφαρμογές να εκμεταλλευτούν την πληροφορία του πλαισίου που ανιχνεύεται από τους αισθητήρες ώστε να ξανά χρησιμοποιηθούν. Το context widget λειτουργεί ανεξάρτητα από τις εφαρμογές ενώ ταυτόχρονα επιτρέπει σε πολλαπλές διαφορετικές εφαρμογές να το χρησιμοποιούν ταυτόχρονα. Επίσης το context widget είναι υπεύθυνο για την διατήρηση πληροφοριών που αποκτήθηκαν για την κατάσταση του χρήστη, ώστε να προσφέρει ένα πλήρες ιστορικό πλαισίου.

Μοντέλο Υποδομών (Infrastructure Model)

Το μοντέλο Υποδομών είναι ισοδύναμο με το μοντέλο πελάτη-εξυπηρετητή (client server) (Joseph J. Martinka 1996) και πρόκειται για ένα πολύ πιο ευέλικτο μοντέλο συγκριτικά με το widget πλαίσιο, δεδομένου ότι προωθεί την ανεξαρτησία των στοιχείων του context-aware συστήματος. Κάθε στοιχείο του μοντέλου έχει την δυνατότητα να εκτελεί την απαιτούμενη λειτουργικότητα, ενώ ταυτόχρονα συμπεριλαμβάνει τη δημιουργία συνδέσεων, την οργάνωση των μηνυμάτων εισόδου / εξόδου, καθώς και τη διαχείριση σφαλμάτων, ώστε τελικά να αυξάνεται σημαντικά η πολυπλοκότητα του μοντέλου. Το μοντέλο Υποδομών μπορεί να υποστηρίξει μια μεγάλη γκάμα συσκευών και εφαρμογών χρησιμοποιώντας τυποποιημένα πρωτόκολλα κωδικοποίησης και δικτύωσης. Επιπλέον, διευκολύνει τις αλλαγές που γίνονται στα διάφορα στοιχεία του εξοπλισμού, με τρόπο ανεξάρτητο και δυναμικό, είτε αυτά είναι οι αισθητήρες, είτε οι υπηρεσίες είτε οι συσκευές. Ένα ακόμη πλεονέκτημα του μοντέλου Υποδομών είναι ότι προσφέρεται για μια απλή προσέγγιση ανάπτυξης και επέκτασης των αισθητήρων, της επεξεργαστικής ισχύς, των δεδομένων και των εφαρμογών (Hong Jason I. και Landay James A. 2001).

Μοντέλο Μαυροπίνακα (Blackboard)

Το μοντέλο Μαυροπίνακα υιοθετεί μια πιο κεντρική άποψη των δεδομένων. Συγκεκριμένα όταν εφαρμόζεται το συγκεκριμένο μοντέλο η εφαρμογή «δημοσιεύει μηνύματα σε ένα κοινόχρηστο πίνακα μηνυμάτων, ενώ ταυτόχρονα καταγράφει τα μηνύματα που λαμβάνει και συνδυάζει όσα ταιριάζουν μεταξύ τους με ένα συγκεκριμένο μοτίβο που έχει αναρτηθεί σε αυτόν τον πίνακα» (Dey Anind K. and Abowd Greogory D. 2000). Όμως το συγκεκριμένο μοντέλο μπορεί να παρουσιάσει πολλά προβλήματα, όπως για παράδειγμα, η ύπαρξη ενός ενιαίου σημείου μιας αποτυχίας, επειδή όλη η επικοινωνία πρέπει να διασχίσει μια κεντρική βάση δεδομένων, και το κοινό μήνυμα μοιράζεται στον πίνακα μηνυμάτων.

Κριτήρια Ανταλλαγής Πληροφοριών (Trade-off criteria)

Επειδή είναι πολύ σημαντική η απόφαση που θα πρέπει να ληφθεί για το πιο μοντέλο είναι ικανότερο για το context-aware σύστημα, θα πρέπει να εξεταστούν όλα τα χαρακτηριστικά που θα ανταλλάσσονται μέσα σε αυτό το σύστημα. Τα διαφορετικά πεδία που πρέπει να μετρηθούν είναι: η αποτελεσματικότητα, η προσπάθεια των ρυθμίσεων, ευρωστία, απλότητα και η επεκτασιμότητα (Winograd Terry 2001). Αναλυτικότερα:

Αποδοτικότητα

Ο στόχος του μοντέλου είναι να επιταχύνει την απόδοση των πληροφοριών και το κάνει πιο αποτελεσματικό. Η αποτελεσματικότητα του μοντέλου εξαρτάται από το εύρος ζώνης και την υστέρηση (latency). Ίσως η αποτελεσματικότητα να μην είναι ένα ζήτημα που αφορά την παρούσα δικτύωση και ταχύτητα επεξεργασίας. Από την άλλη θα πρέπει να εξετάζεται εξαιτίας της εκρηκτικής αύξησης του αριθμού των δικτυακών εφαρμογών και συσκευών, γιατί είναι προφανές ότι η απόδοση δημιουργεί έναν επιπρόσθετο παράγοντα για την επιλογή του καταλληλότερο μοντέλου διαχείρισης πλαισίου.

Προσπάθεια στη Ρύθμιση

Τα context-aware συστήματα περιλαμβάνουν πολλά στοιχεία, οπότε προκύπτει η ανάγκη για εκτέλεση διαδικασιών προσθήκης, αφαίρεσης ή και αλλαγής των ίδιων στοιχείων ή και των πληροφοριών τους, με τον καλύτερο τρόπο ώστε να μην γίνεται επίπονη διαδικασία. Συνεπώς η ρύθμιση των στοιχείων του συστήματος θα πρέπει να γίνεται μεμονωμένα, δηλαδή χωρίς να διακοπεί ή να υποστεί ολική βλάβη το σύστημα, βέβαια όπου αυτό είναι δυνατόν.

Ευρωστία

Σημαντικό κριτήριο σχεδιασμού του context-aware συστήματος είναι η συνεχής του λειτουργία ακόμη και σε περίπτωση κάποιας αποτυχίας, δηλαδή ακόμη και όταν όλα του τα στοιχεία αποτύχουν να λειτουργούν. Στην περίπτωση που το σύστημα παύει να λειτουργεί σωστά, τότε θα πρέπει η διαχείριση των σφαλμάτων αποτυχίας να γίνονται με έναν ελεγχόμενο τρόπο από το σύστημα. Ο βαθμός με τον οποίο το σύστημα μπορεί να αντιμετωπίσει την αποτυχία του έχει ονομαστεί ευρωστία (robustness).

Απλότητα

Το μοντέλο δεν πρέπει να είναι υπερβολικά περίπλοκο, γιατί «αν ένα σύστημα ευρείας εγκατάστασης απαιτεί σύνθετη κατανόηση από τους κατασκευαστές του, τότε σίγουρα θα χρησιμοποιείται μόνο από εκείνους που είχαν την αφοσίωση και το κίνητρο για να το ελέγχουν» (Winograd Terry 2001). Οπότε είναι σημαντικό να έχει κατασκευαστεί με τον πιο απλοϊκό τρόπο για την ευρεία χρήση του.

Επεκτασιμότητα

Τα context-aware συστήματα είναι μια αναδυόμενη τεχνολογία, οπότε είναι σημαντικό «οι υπηρεσίες που υποστηρίζουν μια γενική έννοια του περιβάλλοντος, να είναι εύκολα επεκτάσιμες, με μελλοντικό σκοπό να φιλοξενήσουν νέες και απρόβλεπτες πηγές πληροφοριών του πλαισίου» (Ebling Maria, Hunt Guerney και Lei Hui 2002).

Λαμβάνοντας υπόψη τα παραπάνω κριτήρια ανταλλαγής πληροφοριών χρησιμοποιούνται για την σύγκριση και αντιπαραβολή των διαφορετικών προτύπων μοντέλων πλαισίων που χρησιμοποιούνται στη διαχείριση περιβάλλοντος. Το μοντέλο widget έχει ισχυρή σύζευξη μεταξύ των στοιχείων του συστήματος και αυτό το χαρακτηριστικό το καθιστά πιο αποτελεσματικό μοντέλο σε κάποιες περιπτώσεις. Από την άλλη έχει πολύ σύνθετες ρυθμίσεις και παρουσιάζει προβλήματα αδυναμίας στις περιπτώσεις αποτυχίας. Το μοντέλο Υποδομής όπως αναφέρθηκε προσφέρει ανεξαρτησία στοιχείων που τελικά το καθιστά να είναι εξαιρετικά πολύπλοκο, και τελικά να μην είναι αποδοτικό ούτε για πολλούς προγραμματιστές. Όμως και αυτό έχει θετικά κριτήρια ως προς την απλή ρύθμιση του και την ευρωστία, που αντισταθμίζουν τις αρνητικές επιπτώσεις της πολυπλοκότητας του. Το μοντέλο του Μαυροπίνακα έχει πολύ χαλαρές συνδέσεις των στοιχείων του και για αυτό μπορεί να εμφανίσει προβλήματα στην απόδοση. Ωστόσο παραμένει ένα απλό, εύρωστο και εύκολα ρυθμιζόμενο μοντέλο (Winograd Terry 2001).

Μέθοδοι αφαίρεσης

Εκτός από τα παραπάνω μοντέλα που αναφέρθηκαν και βοηθούν στο σχεδιασμό της διαχείρισης πλαισίου, υπάρχουν και περαιτέρω κάποιες πιο αφηρημένες τεχνικές που επιτρέπουν ουσιαστικά και αποτελεσματικά το χωρισμό των πληροφοριών του πλαισίου. Αυτές οι αφηρημένες τεχνικές είναι οι Διερμηνείς, οι Συναθροιστές και οι Υπηρεσίες πλαισίου που περιγράφονται παρακάτω.

Διερμηνείς (Interpreters)

Εφόσον ολοκληρωθούν οι διαδικασίες ανίχνευσης, απόκτησης και αποθήκευσης του πλαισίου από τις διάφορες πηγές, το επόμενο βήμα για το context aware σύστημα είναι η παραγωγή ενός μηχανισμού για την επίτευξη ερμηνείας του

πλαisiού. Συγκριμένα να δημιουργηθεί ένας μηχανισμός έτσι ώστε οι συγκεντρωμένες πληροφορίες πλαisiού, να χρησιμοποιούνται με έναν προσαρμόσιμο τρόπο μέσα σε αυτό. «*Η Ερμηνεία πλαisiού αναφέρεται στη διαδικασία αύξησης του επιέδου αφαίρεσης από ένα κομμάτι του πλαisiού*» (Dey Anind K. και Abowd Greogory D. 2000). Δηλαδή η ερμηνεία πλαisiού μπορεί να περιλαμβάνει την ενσωμάτωση πολυάριθμων πλαisiών σε ένα, και τελικά να παρέχεται στο σύστημα ένα πλαίσιο υψηλότερου επιπέδου, έτσι ώστε οι διερμηνείς να μπορούν να αλλάζουν τις πληροφορίες του περιβάλλοντος από την αύξηση του επιπέδου αφαίρεσης.

Μια προσέγγιση για την ερμηνεία είναι η χρήση της συγχώνευσης πλαisiού (context aggregation) ώστε να μετατρέψει το «πλαίσιο χαμηλότερου επίπεδου σε υψηλότερο επίπεδο και τελικά να χρησιμοποιηθεί από εφαρμογές» (Mitchell Keith 2002). Η διαδικασία συγχώνευσης πλαisiού μπορεί να παρουσιαστεί με διαφορετικούς τρόπους. Η σύνθεση του πλαisiού μπορεί να γίνει από τον ίδιο τύπο πηγών, προκειμένου να αυξηθεί η ισχύς των πληροφοριών του. Με αυτό τον τρόπο αποφεύγονται οι λήψεις ακατάλληλων αποφάσεων από το σύστημα είτε από τυχόν λανθασμένους αισθητήρες είτε από την λανθασμένη ανάγνωση τους. Επίσης στην σύνθεση πλαisiού γίνεται και η συνάθροιση διαφόρων τύπων πλαisiού, από μια διαφορετική ποικιλία πηγών για την παραγωγή ενός πλαisiού, που είναι εκμεταλλεύσιμη από το σύστημα.

Για την συγχώνευση του πλαisiού το ερευνητικό έργο «Τεχνολογία για την Ενεργοποίηση Γνώσης Πλαisiού» (Technology for Enabling Awareness - TEA) δημιούργησε μια πολυεπίπεδη αρχιτεκτονική. Τα στρώματα της συγκεκριμένης αρχιτεκτονικής περιλαμβάνουν: αισθητήρες (Sensors), Ουρές (Cues), πλαίσιο (Context) και σενάρια (Scripting). Το επίπεδο Αισθητήρων περιλαμβάνει τους αισθητήρες για μετρήσεις φυσικών παραγόντων του περιβάλλοντος, καθώς και λογικούς αισθητήρες. Στο επίπεδο ουράς (Cues) παράγεται μια αφαίρεση από τους φυσικούς και λογικούς αισθητήρες. Στο επίπεδο Πλαisiού προκύπτει μια αποτύπωση της κατάστασης από τις προηγούμενες αφαιρέσεις του στρώματος των Cues. Στο τελευταίο επίπεδο Scripting «*παρέχεται ένας μηχανισμός που*

περιλαμβάνει πληροφορίες του πλαισίου για την εφαρμογή» (Schmidt Albrecht, Beigl Michael και Gellersen Hans-W. 1998). Αναλυτικότερα το στρώμα Σεναρίου έχει τρεις καταστάσεις :

1. Την εισαγωγή σε ένα πλαίσιο : αν μια κατάσταση προσδιορίζεται με μια πιθανότητα που είναι υψηλότερη από ένα όριο που εκτελείται μια ενέργεια μετά από ένα ορισμένο χρονικό διάστημα.
2. Την έξοδο από ένα πλαίσιο : αν μια κατάσταση προσδιορίζεται με μια πιθανότητα που είναι μικρότερη από το κατώτατο όριο που εκτελείται μια ενέργεια μετά από μια ορισμένο χρονικό διάστημα.
3. Για όσο διαρκεί ένα πλαίσιο : αν μια κατάσταση προσδιορίζεται με μια πιθανότητα που είναι μεγαλύτερη από ένα κατώτατο όριο που εκτελείται μια ενέργεια κάθε καθορισμένο χρονικό διάστημα.

(Schmidt Albrecht, Beigl Michael και Gellersen Hans-W. 1998)

«Η διαδικασία συγχώνευσης πλαισίου πρέπει να χειριστεί ομαλά την αίσθηση μεταβίβασης ευθύνης μεταξύ των ορίων των διαφορετικών υπηρεσιών του πλαισίου. Οι στρατηγικές διαπραγμάτευσης, ανταλλαγής και επίλυσης, πρέπει να ενσωματώσουν πληροφορίες από ανταγωνιστικές υπηρεσίες πλαισίου, όταν ταυτόχρονα το ίδιο κομμάτι του πλαισίου παρέχεται από περισσότερες από μία υπηρεσία».

(Abowd Greogory D. και Mynatt Elizabeth D. 1999)

Συναθροιστές (Aggregators)

«Ο Συναθροιστής αναφέρεται στη συλλογή πολλαπλών κομματιών των πληροφοριών πλαισίου, που σχετίζονται λογικά σε ένα κοινό χώρο φύλαξης» (Dey Anind K. και Abowd Greogory D. 2000). Ο Συναθροιστής καταργεί την ανάγκη της context-aware εφαρμογής να συγκεντρώσει το απαιτούμενο πλαίσιο από τις διάφορες πηγές, κάτι που υπό άλλες συνθήκες θα ήταν υποχρεωτικό εξαιτίας της κατακεκομμένης φύσης των context-aware συστημάτων. Οι Συναθροιστές υποστηρίζουν την διανομή πληροφοριών από ένα συγκεκριμένο πλαίσιο σε μια

εφαρμογή, συσσωρεύοντας το σχετικό πλαίσιο που αναζητά μέσα σε μια λογική τοποθεσία. Ένας Συναθροιστής διευκολύνει την ερμηνεία του πλαισίου, συγκεντρώνοντας διάφορες πληροφορίες πλαισίου για τις διάφορες εφαρμογές που αιτούνται. Υπάρχουν μεμονωμένες εφαρμογές που μπορούν να ειδοποιηθούν για τις μεταβολές που έγιναν στο πλαίσιο του συναθροιστή και μπορούν να ωθήσουν (με την χρήση του push μοντέλου) ή να έλξουν (με την χρήση του pull μοντέλου) το πλαίσιο από τους συναθροιστές». «Οι Συναθροιστές παρέχουν ένα πρόσθετο διαχωρισμό ευαισθησίας, για το πώς αποκτάται το πλαίσιο και πώς τελικά χρησιμοποιείται» (Dey Anind K. και Abowd Greogory D. 2000).

Υπηρεσίες Πλαισίου

Τα τρία μοντέλα διαχείρισης Widget, Υποδομής και Μαυροπίνακα, που περιγράφονται παραπάνω, συνδυαστικά με τις δυο αφηρημένες έννοιες Ερμηνεία πλαισίου και Συναθροιστή, αφορούν κυρίως στην απόκτηση γνώσεων και τη διαχείριση πληροφορίας πλαισίου, έτσι ώστε να παρέχουν στην εφαρμογή σχετικές πληροφορίες με έναν εύστοχο τρόπο. Αν εφαρμοστούν αυτά τα στοιχεία για το σχεδιασμό ενός context-aware συστήματος, τότε θα αναπτυχθεί ένα καλό και αποδοτικό σύστημα.

Αυτόματα το context-aware σύστημα αναλαμβάνει δράση βάσει της πληροφορίας που έχει ανακτήσει. «Οι Υπηρεσίες είναι στοιχεία... που εκτελούν ενέργειες για λογαριασμό των εφαρμογών» (Dey Anind K. και Abowd Greogory D. 2000) «ενώ ταυτόχρονα χρησιμοποιούν actuators για μπορέσουν να χειριστούν ή να αλλάξουν την κατάσταση του περιβάλλοντος, δίνοντας έτσι μια απάντηση στην εφαρμογή για μια απόφαση που πάρθηκε με βάση την κεκτημένη γνώση πλαισίου. Η Υπηρεσία του πλαισίου μπορεί να είναι είτε σύγχρονη είτε ασύγχρονη. Μια ασύγχρονη υπηρεσία πλαισίου «απαιτεί ότι η εφαρμογή περιμένει για μια απάντηση» (Ebling Maria, Hunt Guerney και Lei Hui 2001). Όμως δεν υπάρχει καμία εγγύηση ότι η απάντηση θα έχει και σίγουρα αποτέλεσμα, η εφαρμογή ζητάει να ειδοποιηθεί όταν ξεκινάει η υπηρεσία και τα αποτελέσματα των υπηρεσιών θα πρέπει να παραδοθούν όταν θα

είναι σίγουρα διαθέσιμα (Dey Anind K. και Abowd Greogory D. 2000). Μια σύγχρονη υπηρεσία πλαισίου ζητάει πάντα από την εφαρμογή να περιμένει για μια απάντηση.

Υποστήριξη Εφαρμογής Πλαισίου

Οι ερευνητές Guanling Chen και David Kotz (2000) ανάλογα με τις εφαρμογές και την επίγνωση πλαισίου (Context Awareness) που έχουν, χωρίζουν το πλαίσιο σε τύπους:

- **Ενεργό Context Awareness** : μια εφαρμογή αλλάζει συμπεριφορά, στην προσπάθεια της να προσαρμόζεται αυτόματα ώστε να ανακαλύψει το πλαίσιο.
- **Παθητικό Context Awareness** : μια εφαρμογή παρουσιάζει το νέο ή ενημερωμένο πλαίσιο στον ενδιαφερόμενο χρήστη ή στο σύστημα. Εναλλακτικά, η εφαρμογή κάνει το πλαίσιο για το χρήστη ή το σύστημα να ανακτηθεί αργότερα.

Όταν γίνει η λήψη της πληροφορίας που αφορά το πλαίσιο, τότε η χρήση της πληροφορίας του παθητικού context awareness μπορεί να εφαρμοστεί σε μια εφαρμογή. Στην περίπτωση του παθητικού πλαισίου η εφαρμογή εμφανίζει τα δεδομένα ενός αισθητήρα ή της τρέχουσας θέσης ενός ανθρώπου, και τελικά θα χρησιμοποιηθούν από αυτό. Για την περίπτωση του ενεργού context awareness απαιτούνται θεμελιώδεις αλλαγές στον τρόπο λειτουργίας των εφαρμογών και αυτό το καθιστά πιο προκλητικό σε σχέση με το παθητικό πλαίσιο. Ο χρήστης είναι κατά κάποιο τρόπο «υπό τον έλεγχο» της τρέχουσας τεχνολογίας των εφαρμογών, δηλαδή απαιτείται από το χρήστη να κινητοποιήσει τα ενεργά πλαίσια). Οπότε υπάρχει ο κίνδυνος να θεωρήσουν οι χρήστες ότι είναι αβοήθητοι και υπό τον έλεγχο των συστημάτων και των εφαρμογών. Εξαιτίας του γεγονότος ότι αρχίζουν να λαμβάνουν αποφάσεις μόνα τους, καθώς διαθέτουν περισσότερες πληροφορίες από ότι γνωρίζει ή πράττει τελικά ο χρήστης.

Με την χρήση των context-aware συστημάτων οι εφαρμογές που μπορούν να υφίστανται, είναι αυτές που μπορούν να προσαρμοστούν και να χρειάζονται τα λιγότερο σαφή στοιχεία εισόδου από το χρήστη. Για παράδειγμα, πολλές συσκευές θα μπορούσαν να χρησιμοποιήσουν τις πληροφορίες του πλαισίου για να κάνουν διάφορες ενέργειες, όπως ενεργοποίηση και απενεργοποίηση, ανάλογα με το που χρησιμοποιούνται (ή που είναι πιθανό να χρησιμοποιηθούν). Οι συσκευές χρησιμοποιούν μια διακριτική λειτουργία, ανάλογα με την κατάσταση, και μοιράζονται μια ελεγχόμενη ποσότητα πληροφοριών με άλλα συστήματα πάντα με την άδεια του χρήστη. Ταυτόχρονα προσαρμόζουν τις πληροφορίες εξόδου που βασίζονται στους τύπους των τερματικών, κλπ. Όταν οι εφαρμογές υποστηρίζουν αυτές τις δυνατότητες, και τα συστήματα λαμβάνουν αποφάσεις με βάση τις πληροφορίες που έχουν στη διάθεσή τους, τότε ο χρήστης θα λάβει ένα καλύτερο αποτέλεσμα και θα είναι πιο παραγωγικός.

Τα στοιχεία Εξερευνητές του Πλαισίου

Ο Εξερευνητής (Discoverer) είναι εξίσου σημαντικό στοιχείο που αυξάνει την ικανότητα απόδοσης ενός context-aware συστήματος. Ένα Εξερευνητής μπορεί θεωρείται ενεργό στοιχείο της διαχείρισης, γιατί ελέγχει τις δραστηριότητες των άλλων στοιχείων στο context-aware σύστημα. Ο Εξερευνητής διατηρεί ένα μητρώο όλων των στοιχείων και των δυνατοτήτων της συστήματος. Οπότε αν προστεθεί, ή αν αφαιρεθεί ή αν αλλάξει ένα στοιχείο του συστήματος, τότε αντίστοιχα ενημερώνεται το μητρώο του Εξερευνητή. Εάν κάποια στοιχεία του συστήματος αποτύχουν, ο εξερευνητής ενημερώνει το μητρώο του υποδεικνύοντας ότι το συγκεκριμένο στοιχείο δεν είναι πλέον διαθέσιμο προς χρήση. Ο κύριος σκοπός του Εξερευνητή είναι να παρέχει στα άλλα στοιχεία του συστήματος δεδομένα που αφορούν τη σχέση μεταξύ τους. Επομένως είναι ζωτικής σημασίας ότι το μητρώο του εξερευνητή θα ενημερώνεται αμέσως (Dey Anind K. και Abowd Greogory D. 2000).

Προστασία Προσωπικών Δεδομένων

Οι Προσωπικές πληροφορίες των χρηστών πρέπει να λαμβάνονται υπόψη κατά την σχεδίαση και κατασκευή ενός συστήματος που θα προσαρμόζεται σύμφωνα με τις ανάγκες και τους στόχους του χρήστη, ενώ ταυτόχρονα θα λαμβάνει υπόψη του τα ζητήματα της προστασίας ιδιωτικής ζωής των χρηστών. Μέχρι τώρα τα παραδοσιακά συστήματα δεν μπορούσαν να υποστηρίξουν τόσο καλά τα θέματα της ιδιωτικής ζωής του χρήστη, ενώ τώρα με τα context-aware συστήματα αυξάνονται οι προκλήσεις για την βέλτιστη προστασία της ιδιωτικότητας. Οι σχεδιαστές τέτοιων συστημάτων αναφέρουν ότι «Η ιδιωτικότητα είναι άρρηκτα συνδεδεμένη με τον έλεγχο» (Weitzner Daniel J., Ackerman Mark και Darrell Trevor 2001). Συνεπώς, εάν πρόκειται να υλοποιηθεί εξολοκλήρου η προστασία ιδιωτικής ζωής του χρήστη, τότε ο έλεγχος του συστήματος θα πρέπει βρίσκεται στα χέρια του χρήστη.

Παρόλα αυτά, δεν γίνεται κάτι τέτοιο εφικτό στην πράξη, γιατί στις περισσότερες περιπτώσεις είτε δεν επιτρέπεται στον χρήστη να συμμετάσχει στον έλεγχο, είτε γιατί ακόμη και ο ίδιος ο χρήστης δεν επιθυμεί να έχει τον έλεγχο, όταν η απόκτηση και διάδοση της πληροφορία του πλαισίου δεν είναι τόσο απλή. Για τις context-aware εφαρμογές πρέπει να εξετάζονται προσεκτικά όλες οι πληροφορίες που είναι χρήσιμες και για την κάθε περίπτωση που χρησιμοποιούνται, ώστε να αποφύγουμε να δώσουμε στις εφαρμογές πάρα πολλές πληροφορίες που τελικά να θέσουν σε κίνδυνο για την ιδιωτική ζωή των χρηστών. Οπότε είναι αναγκαίο να επιτευχθεί μια ισορροπία μεταξύ των πλεονεκτημάτων της δημοσιοποίησης προσωπικών πληροφοριών, η οποία ενεργοποιεί την εφαρμογή να προσαρμόσει τις υπηρεσίες προς τις ανάγκες των χρηστών, και των μειονεκτημάτων των εν λόγω δημοσιοποιήσεων όταν παύει να υπάρχει η ιδιωτικότητα.

Μια μέθοδος που διασφαλίζει τον διαχωρισμό πληροφοριών πλαισίου για την προστασία της ιδιωτικής ζωής και των δεδομένων του χρήστη, περιγράφεται από το ερευνητικό έργο των Xiaodong Jiang και James A. Landay (2002), καθώς επίσης και

από τις Victoria Bellotti και Abigail J. Sellen (1993). Σύμφωνα με τα παραπάνω ως γενικούς κανόνες προστασίας της ιδιωτικής ζωής αναφέρονται συνοπτικά οι εξής:

- Ένας χρήστης θα πρέπει να ενημερώνεται σχετικά με το είδος των πληροφοριών που συλλέγονται,
- Ένας χρήστης θα πρέπει να έχει την επιλογή να σταματήσει τη συλλογή των προσωπικών πληροφοριών του,
- Ο χρήστης όταν το επιθυμεί θα πρέπει να έχει τη δυνατότητα πρόσβασης στις πληροφορίες που συλλέγονται και να διαγράψει πληροφορίες ,
- Θα πρέπει να διασφαλίζεται στον χρήστη η εξάλειψη της όποιας πιθανότητας η απαγόρευση πρόσβασης μη εξουσιοδοτημένων ατόμων στις πληροφορίες των χρηστών.

Για τις περιπτώσεις που διαφορετικοί άνθρωποι επιθυμούν διαφορετικούς κανόνες προστασίας της ιδιωτικής ζωής, ακόμη δεν έχει αποσαφηνιστεί πώς θα ρυθμίζονται οι κανόνες προστασίας της ιδιωτικότητας. Επίσης παρουσιάζεται το πρόβλημα του τρόπου αλληλεπίδρασης ανθρώπου μηχανής (Human Computer Interface - HCI), καθώς καλούνται να καλυφθούν οι σύνθετες πληροφορίες για μια πολιτική προστασίας ιδιωτικότητας. Κανονικά οι πληροφορίες θα πρέπει να εμφανίζονται στο χρήστη με τέτοιο τρόπο που να είναι εύκολα κατανοητές, έτσι ώστε ο χρήστης να λαμβάνει τα αναγκαία μέτρα για να τις προστατεύσει. Ταυτόχρονα οι ενέργειες που κάνει ο χρήσης στο σύστημα δεν θα δημιουργούν τραγικές αλλαγές, ώστε να υπάρχει μεγάλη απόκλιση στην λειτουργία του.

Η νέα τεχνολογία πληροφορικής Context-aware από τη μια εισάγει επιπλέον νέες ανησυχίες προστασία της ιδιωτικότητας, αλλά από την άλλη μπορεί να εφαρμοστεί για να «βοηθήσει τη διαπραγμάτευση πάνω σε συμφωνίες προστασίας της ιδιωτικής ζωής ... εξετάζοντας τι επιλογή ιδιωτικότητας έκανε ο χρήστης την τελευταία φορά που τέθηκε σε αυτό το συγκεκριμένο πλαίσιο, τι επιλογές ιδιωτικότητας έχει κάνει ο χρήστης με τους υπόλοιπους συνεργάτες του τρέχοντος πλαισίου, και ποιες επιλογές προστασίας της ιδιωτικότητας έκανε ο χρήστης όταν έκανε συσχέτιση των ίδιων δεδομένων μέσα σε ένα διαφορετικό πλαίσιο»

(Weitzner Daniel J., Ackerman Mark και Darrell Trevor 2001). Όπως αναφέρθηκε και σε προηγούμενο κεφάλαιο το έργο Oxygen του MIT αναπτύσσει ένα πρωτότυπο πλαίσιο που λαμβάνει υπόψη του ως κυριότερο κινητήριο παράγοντα την ιδιωτικότητα των χρηστών.

Τι μας επιτρέπει να κάνουμε ένα Πλαίσιο

Τα context-aware συστήματα μπορούν εφαρμοστούν σε πολλές περιπτώσεις, έτσι ώστε στην πραγματικότητα οι εφαρμογές αυτών των συστημάτων περιορίζονται μόνο από τη φαντασία του καθενός. Σύμφωνα με τους Bill Schilit, Norman Adams και Roy Want (1994) οι εφαρμογές των context-aware συστημάτων κατανέμονται στις ακόλουθες κατηγορίες:

- Context-triggered Action – Είναι απλοί κανόνες if-then που καθορίζουν τον τρόπο με τον οποίο τα context-aware συστήματα θα πρέπει να προσαρμοστούν. Ένα παράδειγμα τέτοιας περίπτωσης είναι όταν ένας χρήστης είναι σε σύσκεψη και προκύψει μια τηλεφωνική κλήση για αυτόν, τότε η κλήση θα πρέπει να δρομολογείται στον αυτόματο τηλεφωνητή. Οι triggers μπορούν να τοποθετηθούν σε ορισμένα γεγονότα (events), έτσι ώστε να κάνουν το πλαίσιο να ενεργοποιηθεί ανάλογα, όπως για παράδειγμα να ενημερωθεί ο διαχειριστής του συστήματος για την αύξηση της θερμοκρασίας σε ένα server room.
- Contextual Πληροφορίες και Εντολές – Οι εντολές που δίνονται από τον χρήστη στο σύστημα, μπορούν να παράγουν διαφορετικά αποτελέσματα, γιατί εξαρτάται από ποιο πλαίσιο εκδόθηκαν.
- Επιλογή Εγγύτητας (Proximate selection) – Οι πληροφορίες που προέρχονται από την θέση και την κατάσταση του χρήστη, καθώς και τα αντικείμενα που βρίσκονται σε κοντινή απόσταση από τους χρήστες, μπορούν να τονιστούν

περισσότερο ή να επιλέγονται πιο εύκολα για το περιβάλλον εργασίας του χρήστη. Αυτός ο τρόπος ιεράρχησης των πληροφοριών είναι ιδιαίτερα χρήσιμος στις περιπτώσεις μιας φορητής συσκευής που ενδέχεται να έχει περιορισμένη οθόνη εργασίας. Ένα παράδειγμα τέτοιας περίπτωσης, είναι η πρόταση από το σύστημα ενός εκτυπωτή που είναι ο πιο κοντινός στον χρήστη και μάλιστα να προταθεί πριν καν ο χρήστης δηλώσει την επιθυμία εκτύπωσης κάποιου εγγράφου. Ο συγκεκριμένος τρόπος λειτουργίας του συστήματος ισχύει και για οποιαδήποτε άλλη location-based υπηρεσία χρήστη που επιθυμεί να επωφεληθεί από αυτόν.

- *Automatic Contextual Reconfiguration* – Η αυτόματη επαναρύθμιση της γνώσης πλαισίου αφορά την προσθήκη νέων στοιχείων, διαγραφή στοιχείων, ή ακόμη και την αλλαγή των συνδέσεων μεταξύ των στοιχείων που βασίζονται στο τρέχον πλαίσιο. Τα ζητήματα που προκύπτουν αφορούν τις κοινές ενέργειες με το πεδίο της σύνθεσης των υπηρεσιών (Emre Kiciman, Laurence Melloul, and Armando Fox 2001). Για παράδειγμα, οι ερευνητές Bill N. Schilit, Norman Adams, Rich Gold, Michael M. Tso, και Roy Want (1993) παρήγαγαν ένα σύστημα με το οποίο ο κάθε χρήστης διαθέτει ένα κινητό υπολογιστή PARCTAB και εισέρχεται σε μια αίθουσα, όπου το PARCTAB είναι συνδεδεμένο με ένα εικονικό πίνακα που σχετίζεται με το συγκεκριμένο δωμάτιο. Όταν ο χρήστης τελειώσει και δεν χρησιμοποιεί άλλο τον πίνακα (whiteboard) και αποχωρεί από την αίθουσα, τότε θα διακοπεί η σύνδεση μεταξύ του PARCTAB και του πίνακα. Σε περιπτώσεις σαν και αυτή, οι πληροφορίες του πλαισίου που είναι σχετικές με τη θέση του χρήστη, χρησιμοποιήθηκαν για να ρυθμίσουν τη σύνδεση μεταξύ της κινητής συσκευής και του «στατικού» εικονικού πίνακα.
- *Τοποθέτηση ετικέτας Μεταδεδομένων (Metadata Tagging)* – Όλες οι πληροφορίες του πλαισίου που μπορούν να συνδεθούν με άλλα υπάρχοντα κομμάτια των πληροφοριών, για να δώσουν πιο έμμεσες πληροφορίες για τα αντικείμενα στο σύστημά μας, χρησιμοποιούν την σήμανση με ετικέτες. Αυτό το πλαίσιο λειτουργεί και παράγει μετα-δεδομένα τόσο για τα φυσικά

αντικείμενα, όσο και για τα εικονικά αντικείμενα του συστήματος μας. Ένα παράδειγμα τέτοιας εφαρμογής είναι, όταν ένας χρήστης καταγράφει ένα κλιπ ήχου σε μια φορητή συσκευή, το σύστημα μπορεί να επισυνάψει τις τρέχουσες πληροφορίες πλαισίου (ημερομηνία και ώρα, παρόντες άνθρωποι, τρέχουσα δραστηριότητα) στο κλιπ για εύκολη ανάκτηση και εύρεση. Έτσι μπορούν να δημιουργηθούν αυτόματα ημερολόγια κινήσεων, απλά «καταγράφοντας» το σχετικό πλαίσιο.

- Προσαρμοστικότητα τερματικών (Terminal adaptivity) – Η χρηστικότητα των κινητών συσκευών πάσχει από τις μικρές και τις πολλές διεπαφές με τον χρήστη. Οπότε οι πληροφορίες του πλαισίου μπορούν να βοηθήσουν και να προσαρμοστούν καταλλήλως για να παρουσιάσουν στον χρήστη μόνο τις σχετικές πληροφορίες, παρότι δίνουν συντρυπτικές διαφορετικές και πολλές επιλογές χρήσης.

Το πιο συνηθισμένο σενάριο σε μια context-aware εφαρμογή, είναι να χρησιμοποιεί πληροφορίες σχετικές με την τοποθεσία του χρήστη και να ενεργεί ως ένα είδος εικονικού ξεναγού, όπως το σύστημα Cyberguide (Gregory D. 1997) από το Ινστιτούτο Τεχνολογίας της Γεωργίας στην Ατλάντα, ή το σύστημα Τηλεμεταφοράς (Frazer Bennett, Tristan Richardson και Andy Harter 1994) από το ερευνητικό τμήμα της Olivetti Research Ltd. Τέτοιου είδους συστήματα χρησιμοποιούν είτε GPS είτε υπέρυθρες ακτίνες παρακολούθησης για τον προσδιορισμό θέσης του χρήστη με έναν λογικό βαθμό ακρίβειας. Το σύστημα εμφανίζει τη θέση του χρήστη στην οθόνη μιας φορητής συσκευής και χρησιμοποιεί τη γνώση που έχει από το πλαίσιο, έτσι ώστε όταν ο χρήστης βρίσκεται κοντά σε ένα σημείο του ενδιαφέροντος του, το σύστημα να του εμφανίσει τις αντίστοιχες πληροφορίες στην οθόνη του. Επίσης το πλαίσιο μπορεί να χρησιμοποιηθεί και ως ένα αρχείο καταγραφής. Δηλαδή να καταγράφει τις τοποθεσίες που έχει επισκεφθεί ο χρήστης οι οποίες θα εξεταστούν αργότερα. Αυτές οι εφαρμογές ενός τέτοιου συστήματος εντοπισμού είναι απλές και οι πληροφορίες τους μπορούν να ανακατευθύνονται στην έξοδο του υπολογιστή είτε σε μια κοντινή οθόνη, είτε να διαβιβάζονται αυτόματα οι τηλεφωνικές κλήσεις

του χρήστη στην πλησιέστερη τηλεφωνική συσκευή (Roy Want, Andy Hopper, Veronica Falcao, και Jon Gibbons 1992).

Ένα ακόμη στοιχείο εισόδου χαμηλού κόστους σε ένα context-aware σύστημα είναι τα PDAs που έχουν συνδεδεμένους αισθητήρες. Τοποθετώντας σε ένα context-aware σύστημα PDAs με αισθητήρες, όπως αισθητήρες εγγύτητας, αισθητήρες αφής, αισθητήρες προσανατολισμού, αισθητήρες φωτός και αισθητήρες θερμοκρασίας, η ίδια η συσκευή μπορεί να προσαρμοστεί άμεσα σε αυτό το πλαίσιο πληροφοριών. Για παράδειγμα, δίνεται η επιλογή στον χρήστη αν θα χρησιμοποιήσει τη συσκευή σε λειτουργία πορτρέτου ή τοπίου, κάνοντας μόνο μια κίνηση απλής περιστροφή της οθόνης. Η διαχειριστική δύναμη του πλαισίου μπορεί να γίνει ακόμη πιο έξυπνη ενημερώνοντας τη συσκευή να ενεργοποιείται και εφόσον υπάρχει κίνηση να διατηρείται ενεργή, και να απενεργοποιείται, στην περίπτωση που παραμείνει αδρανής οτιδήποτε από αυτή σε κοντινή απόσταση. Οι συγκεκριμένες context-aware συσκευές λειτουργούν αυτόνομα και γίνονται αρκετά ελκυστικές, επειδή μπορούν να χρησιμοποιήσουν το πλαίσιο χωρίς την ανάγκη των μετρήσεων από τα όργανα του περιβάλλοντα χώρου.

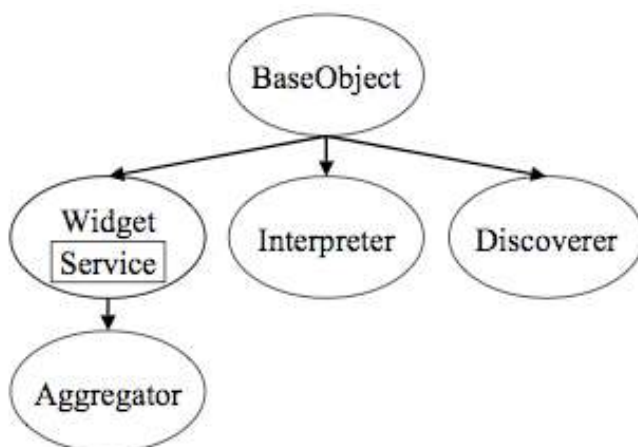
Παραδείγματα Context-Aware Συστημάτων

Υπάρχουν τέσσερα παραδείγματα συστημάτων που χρησιμοποιούν το σύστημα πληροφοριών πλαισίου και δείχνουν διαφορετικές προσεγγίσεις για τη διαχείριση πλαισίου. Ενώ επίσης εμφανίζουν τον τεράστιο όγκο των περιοχών και των ζητημάτων που πρέπει να αντιμετωπισθούν κατά τη χρήση πλαισίου σε ένα σύστημα.

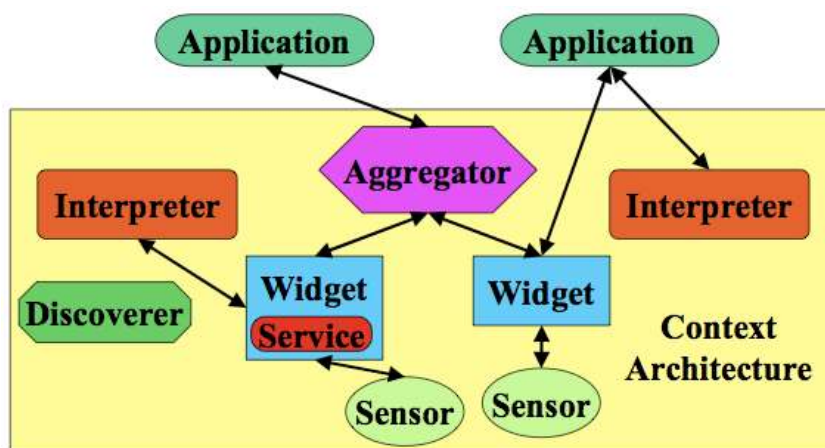
Context Toolkit

Το πλαίσιο toolkit υποστηρίζει context-aware εφαρμογές βοηθώντας μη-context-aware εφαρμογές να χρησιμοποιήσουν το πλαίσιο και να εξελίσσονται σύμφωνα με τις υπάρχουσες context-aware εφαρμογές. Επίσης το συγκεκριμένο πλαίσιο

προστατεύει τις εφαρμογές από τις αλλαγές που γίνονται μέσα σε αυτό, καθώς και τυχόν συνέπειες που μπορεί να επιφέρουν αυτές οι αλλαγές στο σύστημα. Το πλαίσιο toolkit κάνει διαχωρισμό της εφαρμογής από τα θέματα απόκτησης πλαισίου με τη χρήση των widgets. Επίσης αποθηκεύει το πλαίσιο που συγκέντρωσε από το περιβάλλον και κατασκευάζει ένα ιστορικό, το οποίο είναι προσβάσιμο στις εφαρμογές για να πάρουν αποφάσεις σχετικά με την πρόθεση του χρήστη. Επιπλέον το toolkit πλαισίου περιλαμβάνει ένα διερμηνέα πλαισίου για την εκτέλεση λειτουργιών ερμηνείας πλαισίου, όπως περιγράφηκε σε προηγούμενη ενότητα, καθώς και έναν aggregator για να συγκεντρώνει σχετικά πλαίσια που αναζητά η εφαρμογή μέσα σε μία λογική τοποθεσία (Dey Anind K. 2000).



Εικόνα 27: Στοιχεία toolkit πλαισίου, όπου τα αντικείμενα είναι ιεραρχημένα και τα βέλη δείχνουν τις σχέσεις με τις υποκατηγορίες



Εικόνα 28 : Τυπική αλληλεπίδραση μεταξύ των εφαρμογών και των στοιχείων του toolkit πλαισίου

Owl

Η υπηρεσία πλαισίου Owl δημιουργήθηκε από τους Ebling Maria R., Hunt Guerney D. H. και Lei Hui (2001), χρησιμοποιείται σε context-aware συστήματα και «έχει ως στόχο να συγκεντρώνει, να διατηρεί και να παρέχει πληροφορίες στους πελάτες του πλαισίου. Ενώ ταυτόχρονα ασχολείται με διάφορα προηγμένα θέματα, όπως τα δικαιώματα πρόσβασης, το ιστορικό πλαισίου, την ποιότητα του πλαισίου, την επεκτασιμότητα και την ανάπτυξη» (Candolin Catharina και Kari Hannu H. 2002). Επίσης «διαθέτει ένα μοντέλο προγραμματισμού που επιτρέπει εξίσου στα σύγχρονα και ασύγχρονα ερωτήματα (queries) ειδοποιήσεις συμβάντων (events). Προστατεύει την ιδιωτικότητα των ανθρώπων μέσω της χρήσης ενός μηχανισμού ελέγχου πρόσβασης βάσει ρόλου (RBAC)» (Ebling Maria R., Hunt Guerney D. H. και Lei Hui 2001).

Ένα παράδειγμα τέτοιας χρήσης συστήματος θα μπορούσε να ανιχνεύσει ότι ο δέκτης που βρίσκεται στο γραφείο ενός συγκεκριμένου χρήστη την συγκεκριμένη στιγμή, δηλαδή την τρέχουσα, λαμβάνει ένα ενεργό σήμα (active badge). Δηλαδή λαμβάνει τις τρέχουσες πληροφορίες του περιβάλλοντος, ότι η ακριβής τοποθεσία του χρήστη είναι στο γραφείο του. Διατηρώντας ένα μεγάλο ιστορικό από αυτές τις πληροφορίες, το σύστημα μπορεί επίσης να υποθέτει ότι ο χρήστης βρίσκεται συνήθως στο γραφείο του στις 9:00 π.μ. τις εργάσιμες ημέρες. Η υπηρεσία Owl του συστήματος συλλέγει και διατηρεί πληροφορίες για το πλαίσιο από μια ποικιλία πηγών πλαισίου για πολυάριθμα θέματα. Τα τερματικά των χρηστών ζητούν πληροφορίες του πλαισίου για ένα ή περισσότερα τμήματα. Τα τμήματα της υπηρεσίας Owl μπορεί είναι χρήστες ή αντικείμενα. Τερματικά του Owl μπορεί να είναι είτε άλλοι χρήστες του περιβάλλοντος, για το προαναφερόμενο παράδειγμα ένας γραμματέας, ένας συνάδελφος, είτε άλλα προγράμματα δηλαδή εφαρμογές. Τα τερματικά θέτουν ερωτήματα (query) για την τρέχουσα πληροφορία του πλαισίου ή μπορεί να υποβάλουν μια αίτηση για να ενημερωθούν, όταν πρέπει να πληρείται κάποια κατάσταση. Ο ελεγκτής (controller) του πλαισίου μπορεί να είναι το τμήμα στο οποίο αναφέρονται οι πληροφορίες ή μπορεί να είναι ο ιδιοκτήτης του αντικειμένου για το οποίο αναφέρονται οι πληροφορίες. [46]

Kimura

Το κίνητρο του συστήματος Kimura είναι να ενσωματώσει τόσο τις φυσικές όσο τις και εικονικές πληροφορίες του πλαισίου, οι οποίες εμπλουτίζουν τις δραστηριότητες των στοιχείων που επεξεργάζονται τη γνώση. Το σύστημα Kimura χρησιμοποιεί το μοντέλο του μαυροπίνακα που βασίζεται σε πολλαπλούς χώρους (tuple space). Τα τέσσερα στοιχεία που λειτουργούν σε χώρους πλειάδας είναι:

1. Η παρακολούθηση της οθόνης του υπολογιστή (desktop monitoring) και η διαχείριση στοιχείων, τα οποία χρησιμοποιούν χαμηλού επιπέδου Window hooks για να παρατηρήσουν τις δραστηριότητες των χρηστών και το στοιχείο του διερμηνέα,
2. Οι περιφερειακές συσκευές απεικόνισης και τα στοιχεία αλληλεπίδρασης που διαβάζουν και εμφανίζουν τις πληροφορίες πλαισίου, έτσι ώστε ο χρήστης να μπορεί να παρατηρήσει και να αξιοποιήσει το πλαίσιο εντός των εργασιών του,
3. Το στοιχείο ελέγχου πλαισίου, που γράφει σε πλειάδες χαμηλού επιπέδου και ερμηνεύεται αργότερα,
4. Το στοιχείο διερμηνέα, το οποίο μεταφράζεται σε πλειάδες χαμηλού επιπέδου μέσα σε πλειάδες και μπορεί αμέσως να διαβαστεί από την οθόνη του πίνακα και το στοιχείο αλληλεπίδρασης.

(MacIntyre Blair, Mynatt Elizabeth D., Tullio Joe & Volda Steve)

Solar

Οι Guanling Chen, Ming Li, και David Kotz, (2004) προτείνουν τη χρήση των δικτύων Context Fusion (CFNs) για την παροχή υπηρεσιών συγχώνευσης δεδομένων σε context aware εφαρμογές. Οι υπηρεσίες συγχώνευσης δεδομένων αφορούν την συγκέντρωση και την ερμηνεία των δεδομένων που έχουν πάρει από τους αισθητήρες. Τα CFN δίκτυα βασίζονται σε έναν χειριστή γραφικού μοντέλου, στο οποίο η διαδικασία του πλαισίου καθορίζεται από τους προγραμματιστές των εφαρμογών σύμφωνα με τους όρους των πηγών (sources), των καταβόθρων (sinks) και των καναλιών (channels). Στο CFN μοντέλο, οι αισθητήρες αντιπροσωπεύονται από τις πηγές, και εφαρμογές από τις καταβόθρες. Οι χειριστές είναι υπεύθυνοι για

την επεξεργασία των δεδομένων, ενώ μπορούν να λειτουργήσουν τόσο ως πηγές όσο και ως καταβόθρες. Το συγκεκριμένο μοντέλο εφαρμόστηκε μορφή του Solar συστήματος. Το σύστημα αποτελείται από μια κλιμακούμενη πλατφόρμα peer-to-peer (P2P) το οποίο στηρίζει τα γραφήματα του χειριστή κατά τον χρόνο εκτέλεσης για λογαριασμό των context-aware εφαρμογών. Οι σταθμοί εργασίας (hosts Planets) του Solar μπορούν να υποστηρίξουν τις εφαρμογές και την κινητικότητα των αισθητήρων, χρησιμοποιώντας την τεχνική αποθήκευσης των γεγονότων σε μια προσωρινή μνήμη (buffering events), στις περιπτώσεις που υπάρξει περιοδική διακοπή της σύνδεσης. Ωστόσο το Solar σύστημα δεν έχει καταφέρει να αντιμετωπίσει την ετερογένεια, την προστασία της ιδιωτικής ζωής, την παρακολούθηση και τον έλεγχο του συστήματος των χρηστών. [47]

Συνοπτικότερα το Solar σύστημα είναι ένα σύστημα middleware και έχει σχεδιαστεί για να υποστηρίζει context-aware εφαρμογές. Το συγκεκριμένο σύστημα αποτελείται από βασικές πηγές πληροφοριών. Αυτές οι πηγές είναι οι αισθητήρες που συγκεντρώνουν είτε τις φυσικές είτε τις εικονικές πληροφορίες του περιβάλλοντος, τα φίλτρα, οι μετασχηματιστές και συγκεντρωτές (aggregators). Το middleware τροποποιεί το πλαίσιο και τελικά προσφέρει στην εφαρμογή ένα πρακτικό πλαίσιο πληροφοριών. Η ομάδα των ερευνητών του Dartmouth Κολεγίου (Chen G., Li M., και Kotz D., 2004) που σχεδίασαν το Solar, συνεχίζουν να πειραματίζονται με αυτό το σύστημα και φιλοδοξούν να παράγουν ένα σύστημα που να είναι ευέλικτο, επεκτάσιμο και επαναχρησιμοποιήσιμο όσον αφορά την κατανάλωση εύρους ζώνης δικτύου και την επίδρασή της στην επικαιρότητα των δεδομένων. [48]

5. Συμπεράσματα

Ο βασικός στόχος του context aware computing είναι να χρησιμοποιήσει όσο τον δυνατό περισσότερα είδη πληροφοριών με σκοπό να εμπλουτίσει την λειτουργικότητα διάφορων εφαρμογών και να καταργήσει, όσον είναι εφικτό, την προσπάθεια που καταβάλλει ο χρήστης. Τα πλεονεκτήματα των πληροφοριών πλαισίου για τις εφαρμογές είναι ότι όλοι οι τύποι εφαρμογών μπορούν να επωφεληθούν από πλουσιότερες συνολικές πληροφορίες για τη λήψη αποφάσεων.

Η μοντελοποίηση τέτοιων συστημάτων πλαισίου γίνεται σύμφωνα με τους χρήστες του πλαισίου, και συμπεριλαμβάνει την προστασία της ιδιωτικής ζωής αυτών των χρηστών. Επειδή η ιδιωτική ζωή είναι ένα σημαντικό θέμα που πρέπει να ληφθεί υπόψη σε όλα τα στάδια του σχεδιασμού αυτών των συστημάτων.

Επίσης ο σχεδιασμός των context-aware συστημάτων πρέπει να διευκολύνει την πρόληψη των ενδεχομένως ενοχλητικών ή επικίνδυνων καταστάσεων για τον τελικό χρήστη. Τα Context-aware συστήματα πρέπει να είναι έτσι σχεδιασμένα ώστε να καλύπτουν τη «διαχείριση κινδύνου» (risk management) προκειμένου να ληφθεί υπόψη η συνέπεια πιθανών λαθών και η επίδραση στην λήψη των αποφάσεων.

Σημαντική συνιστώσα για τον σχεδιασμό ενός context-aware συστήματος είναι να μπορεί να καθορίσει πόση αξία πλαισίου θα προσθέσει σε κάθε κατάσταση και να μην βασίζονται οι χρήστες στις πληροφορίες ως πανάκεια. Εν ολίγοις οι πληροφορίες πλαισίου θα μπορούσαν να είναι δύσχρηστες και δύσκολα χρησιμοποιούμενες. Είναι ξεκάθαρο όμως, ότι υπάρχει αρκετή αξία στην δυνατότητα του context-value να επικρατήσει.

Τα τεχνολογικά άλματα στους τομείς των αισθητήρων, της αναγνώρισης από υπολογιστές και της ασύρματης δικτύωσης επιτρέπουν την ανάπτυξη νέων κατηγοριών context-aware εφαρμογών. Ωστόσο, παραμένει η ανάγκη για μείωση των φραγμών εισόδου όσον αφορά την αξιοποίηση τέτοιων συστημάτων. Οι τρέχουσες μέθοδοι παροχής πληροφοριών πλαισίου παραμένουν εξαιρετικά

εξειδικευμένες ανά εφαρμογή και δε μπορούν να χρησιμοποιηθούν με γενικό τρόπο. Η ευρεία χρήση context aware συστημάτων θα επιτευχθεί μόνο αφότου (όταν και αν) επικρατήσουν πρότυποι μηχανισμοί για το διαμοιρασμό του context / την κοινόχρηστη πρόσβαση στο context. Σε αυτόν τον τομέα ακόμη παρουσιάζονται πολλά τεχνικά θέματα, και πολλά από αυτά σχετίζονται με τους ανθρώπους. Όταν επιλυθούν αυτά τα θέματα, θα έχουν παραμείνει ακόμη ζητήματα προς εξέταση των δυνατοτήτων για νέα είδη αλληλεπίδρασης, καθώς και για νέες εφαρμογές που κάνουν χρήση του πλαισίου. Επίσης έχουν γίνει μεγάλα βήματα προόδου στους αισθητήρες, στην αναγνώριση κίνησης και στην ασύρματη δικτύωση, που επιτρέπει την δημιουργία νέων κατηγοριών για context-aware εφαρμογές. Για παράδειγμα, οι εταιρείες STMicroelectronics και Movea συμφώνησαν να ενσωματώσουν την τεχνολογία SmartMotion Movea στον μικροελεγκτή STM32F401. Η ενσωμάτωση της υψηλής ακρίβειας μοντέλα κίνησης επεξεργασίας Movea του στην STM32F401 επιτρέπει στους κατασκευαστές κινητών συσκευών και στους προγραμματιστές εφαρμογών να παρέχουν context-aware εφαρμογές και υπηρεσίες, όπως η πλοήγηση πεζών και παρακολούθηση των δραστηριοτήτων τους. Η συγκεκριμένη αρχιτεκτονική STM32F401 μαζί με προηγμένα χαρακτηριστικά κίνησης Movea δίνει μια λύση για Always-On εφαρμογές με εξαιρετική χαμηλή ισχύος. Για την μεγαλύτερη πρόοδο τέτοιων συστημάτων είναι απαραίτητη η μείωση διάφορων φραγμών, ώστε οι εφαρμογές να εισέλθουν στην ευρύτερη αγορά. [49]

Από τα παραπάνω προκύπτει ότι απαιτείται η ύπαρξη ενός πρότυπου για context-aware computing, με στόχο οι προγραμματιστές να μην δημιουργούν για κάθε νέο σύστημα εκ νέου κατηγορίες context aware. Το συγκεκριμένο πρότυπο πρέπει να τεθεί σε λειτουργία ώστε να επιτρέπει την επαναχρησιμοποίηση των context-aware στοιχείων του συστήματος. Οπότε στις περιπτώσεις που δημιουργείται ένα εντελώς νέο σύστημα, να μην χρειάζεται να παράγονται κάθε φορά τα στοιχεία του για κάθε context-aware εφαρμογή. Αν το «πρότυπο πλαισίου» δεν είναι κατασκευασμένο με την επαναχρησιμοποίηση των στοιχείων του, τότε μπορεί να μην καταφέρει ποτέ να επικρατήσει η κυκλοφορία του context aware συστήματος. Δύο επιπλέον ζητήματα που πρέπει να αντιμετωπιστούν στο άμεσο μέλλον, είναι της ασφάλειας και της ιδιωτικής ζωής που προκαλούνται από το context-aware computing, λόγω της

δυναμικής του «φύσης». Οπότε προκειμένου να υπάρχει εγγύηση για την ασφάλεια των προσωπικών στοιχείων των χρηστών, θα πρέπει να μελετηθούν μοντέλα και τεχνικές, για τον σχεδιασμό ενός νέου πρότυπου context aware.

Η μελέτη του M-Zones project [48] επιχειρεί να ρίξει φως στις διαφορετικές προσεγγίσεις για την κατηγοριοποίηση των πληροφοριών που απαρτίζουν το πλαίσιο. Θα μπορούσαμε να πούμε ότι κάθε τύπος πληροφορίας θα μπορούσε να θεωρηθεί ως πλαίσιο. Έτσι είναι σημαντικό να ταξινομηθεί αυτή η πληροφορία σε διαφορετικούς τύπους, προκειμένου να είναι σε θέση να χρησιμοποιηθεί αποτελεσματικά. Ταξινομώντας τις πληροφορίες σε τύπους η διαχείριση των πληροφοριών πλαισίου γίνεται ευκολότερα, ειδικά όταν είναι γνωστά τα ιδιαίτερα χαρακτηριστικά τους. Ωστόσο, δεν μπορεί κάποιος ποτέ να ελπίζει ότι πρακτικά θα καταφέρει να απαριθμήσει όλες τις πληροφορίες που θα μπορούσαν να αποτελέσουν το πλαίσιο.

Στόχος της μοντελοποίησης των χρηστών, των συστημάτων και συσκευών για συστήματα context-aware είναι να δώσει τη δυνατότητα σε διαφορετικά συστήματα πλαισίου να επικοινωνούν μεταξύ τους και να μοιράζονται το πλαίσιο. Στην πραγματικότητα από τη στιγμή που η πληροφορία πλαισίου είναι πανταχού παρούσα, θα καταφέρει να κυκλοφορήσει στο μέγιστο των δυνατοτήτων της. Ωστόσο, είναι δυνατόν να δημιουργηθεί ιεραρχικά, δηλαδή πρώτα από απλές εφαρμογές πλαισίου και στη συνέχεια σε ένα πιο προχωρημένο επίπεδο, το οποίο αξιοποιεί πλήρως τη δύναμη του context-aware computing.

Οι μελέτες και έρευνες σχεδίασης κατάλληλων συστημάτων διαχείρισης πλαισίου συνεχίζονται, ελπίζοντας ότι θα οδηγήσουν σε ένα σύστημα που θα είναι πιο εύκαμπτο και διαισθητικό, τόσο για τον χρήστη όσο και για τον διαχειριστή του συστήματος πλαισίου. Η μέχρι τώρα έρευνα έχει δείξει ότι το μέλλον για το context aware computing είναι πολύ φωτεινό.



Εικόνα 29 : Πρόδος της τεχνολογίας

6. Υλοποίηση ενός πρότυπου Context Engine

Σχεδιασμός

Ο σχεδιασμός του συστήματος πραγματοποιήθηκε μετά από βιβλιογραφική μελέτη αρκετών παρόμοιων συστημάτων, και είχε ως βασικό στόχο την ευελιξία του προκειμένου να μπορεί να προσαρμοστεί σε μια ευρεία γκάμα από περιβάλλοντα εκτέλεσης. Τα επιθυμητά χαρακτηριστικά και οι στόχοι που οδήγησαν στην τελική αρχιτεκτονική περιγράφονται στις επόμενες ενότητες.

Στόχοι / επιθυμητά χαρακτηριστικά

Ένα Context Engine, όπως λέει και το όνομά του, είναι υπεύθυνο για μια σχετικά απλή αλλά συνάμα ιδιαίτερα σημαντική δουλειά στο πλαίσιο ενός έξυπνου συστήματος. Ο ρόλος του είναι να διατηρεί μια σαφή εικόνα της κατάστασης του συστήματος, όπως αυτή προκύπτει από τις τιμές που επιστρέφουν οι διάφοροι αισθητήρες που είναι εγκατεστημένοι σε αυτό, αλλά και από άλλα στοιχεία του περιβάλλοντος εκτέλεσης, όπως πχ η τρέχουσα ημερομηνία και ώρα. Το Context Engine είναι υπεύθυνο να συνθέτει τα δεδομένα και τα στοιχεία αυτά με διαφανή για το υπόλοιπο σύστημα τρόπο, και να εξάγει από αυτά συμβάντα τα οποία έχουν κάποιο νόημα με βάση το εκάστοτε εννοιολογικό πλαίσιο. Πρέπει δηλαδή να πραγματοποιήσει ένα semantic uplifting των raw δεδομένων σε κάποια μορφή που έχει νόημα για το συγκεκριμένο σύστημα και το συγκεκριμένο σενάριο χρήσης.

Οι στόχοι που τέθηκαν κατά το σχεδιασμό του Context Engine και οδήγησαν στην τελική αρχιτεκτονική του ήταν οι εξής:

Υποστήριξη απεριόριστου (θεωρητικά) αριθμού αισθητήρων και συμβάντων

Καθώς κάθε έξυπνο σύστημα έχει διαφορετικές απαιτήσεις και πολυπλοκότητα, είναι σημαντικό να μην υπάρχει περιορισμός στον αριθμό των αισθητήρων που

υποστηρίζονται και τον αριθμό των διαφορετικών συμβάντων τα οποία μπορεί να σημάνει το Context Engine. Παράλληλα, είναι επιθυμητό οι υπολογιστικές απαιτήσεις για την ανάκτηση της τιμής των αισθητήρων και της αξιολόγησης των συμβάντων να αυξάνονται γραμμικά σε σχέση με τον αριθμό τους, έτσι ώστε το Context Engine να μπορεί να ανταποκριθεί ικανοποιητικά σε περιβάλλοντα όπου αλλάζουν πολύ συχνά οι τιμές των αισθητήρων και οι υπολογιστικοί πόροι είναι περιορισμένοι.

Ανεξαρτησία από τον τρόπο λειτουργίας των αισθητήρων

Κάθε αισθητήρας είναι πιθανό να επιστρέφει τιμή διαφορετικού τύπου, την οποία ανακτά ή υπολογίζει με διαφορετικό τρόπο. Το Context Engine πρέπει να μην έχει κάποια εξάρτηση από συγκεκριμένους τύπους τιμών αισθητήρων. Για παράδειγμα, ένας αισθητήρας επιτάχυνσης (accelerometer) μπορεί να επιστρέφει ως τιμή ένα σετ από πολλαπλές ανεξάρτητες τιμές που χρησιμοποιούνται για να περιγράψουν το διάνυσμα της επιτάχυνσης (X, Y, Z, a) όπου οι X, Y, Z περιγράφουν την κατεύθυνση του διανύσματος και το a την τιμή του. Οι (απλές ή σύνθετες) τιμές που επιστρέφει ένας αισθητήρας πρέπει να είναι συγκρίσιμες μεταξύ τους. Η διεπαφή λοιπόν που οφείλει να υλοποιεί κάθε αισθητήρας πρέπει να είναι ίδια, και αφορά την επιστροφή μιας (απλής ή σύνθετης) τιμής.

Υποστήριξη πολλαπλών σεναρίων

Το Context Engine πρέπει να είναι σε θέση να υποστηρίξει την εκτέλεση πολλαπλών σεναρίων – ιδανικά παράλληλα και απομονωμένα το καθένα από τα άλλα, και με διαφορετικές παραμέτρους. Με τον τρόπο αυτό μπορούν να υποστηριχθούν σύνθετες απαιτήσεις, όπως π.χ. η δοκιμή μιας δοκιμαστικής έκδοσης ενός σεναρίου παράλληλα με την λειτουργία της σταθερής έκδοσης.

Ανεξαρτησία από τον τρόπο αναπαράστασης των σεναρίων

Τα σεναρία είναι επιθυμητό να μπορούν να αναπαρασταθούν σε μια μορφή που να επιτρέπει εύκολη δημιουργία και επεξεργασία με χειροκίνητο τρόπο χωρίς να είναι απαραίτητη η χρήση κάποιου εξειδικευμένου εργαλείου επεξεργασίας. Επίσης είναι σημαντικό να αποθηκεύονται σε μια μορφή που να επιτρέπεται η φορητότητά τους σε διαφορετικά λειτουργικά συστήματα. Τέλος, είναι επιθυμητό να υπάρχει δυνατότητα για έλεγχο της συντακτικής εγκυρότητας των σεναρίων.

Ανεξαρτησία από τον τρόπο λειτουργίας των actuators

Σε ένα τυπικό έξυπνο σύστημα, όταν καταγραφεί κάποιο συμβάν ξεκινάει η εκτέλεση μιας ή περισσότερων ενεργειών, προκειμένου το σύστημα να αντιδράσει κατάλληλα με βάση το εκάστοτε σενάριο. Για παράδειγμα, όταν το Context Engine διαπιστώσει πως η θερμοκρασία έχει βγει από το επιθυμητό εύρος τιμών, πρέπει πιθανότατα με κάποιο τρόπο το σύστημα να εκτελέσει κάποιον actuator, που θα ανοίξει / κλείσει το κλιματιστικό, κλπ. Ο τρόπος λειτουργίας αυτών των actuators καθώς και ο μηχανισμός με τον οποίο ενεργοποιείται η εκτέλεσή τους (triggering) δεν αποτελεί αντικείμενο αυτής της εργασίας.

Υποστήριξη καταγραφής όλων των βημάτων της διαδικασίας εκτέλεσης ενός σεναρίου

Είναι επιθυμητό το σύστημα να είναι σε θέση να καταγράψει κάθε βήμα της διαδικασίας που ακολουθείται κατά την εκτέλεση ενός σεναρίου. Η καταγραφή αυτή αποτελεί ένα χρήσιμο διαγνωστικό εργαλείο για την λειτουργία του συστήματος. Παραδείγματα ενεργειών που είναι ενδιαφέρον να καταγραφούν είναι η ανάκτηση τιμών από τους αισθητήρες, η εξέταση των συνθηκών που σχετίζονται με κάποιο συμβάν, και φυσικά η σήμανση κάποιου συμβάντος. Η καταγραφή των βημάτων αυτών πρέπει να μπορεί να ενεργοποιηθεί ή απενεργοποιηθεί κατά βούληση, ενώ είναι χρήσιμο τα γεγονότα που καταγράφονται να σχετίζονται με τη χρονική στιγμή κατά την οποία έλαβαν χώρα, καθώς επίσης και με ένα βαθμό

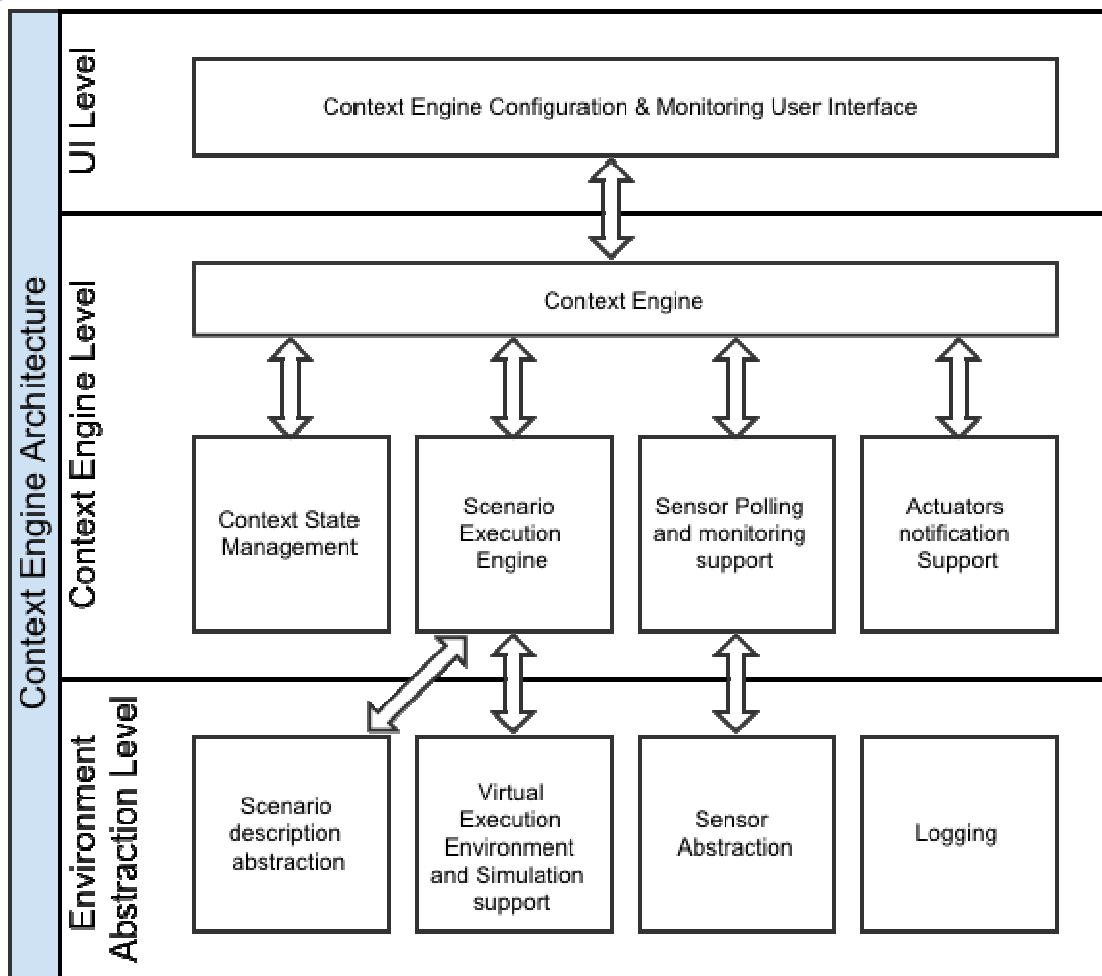
σημαντικότητας, προκειμένου να μπορούν να φιλτραριστούν μόνο οι εγγραφές εκείνες που είναι πιο σημαντικές ή ενδιαφέρουσες.

Υποστήριξη για ρεαλιστική προσομοίωση της εκτέλεσης των σεναρίων

Η προσομοίωση είναι ένα πολύ ισχυρό εργαλείο για την δοκιμή ενός συστήματος με ρεαλιστικό τρόπο. Στην περίπτωση του Context Engine, είναι σαφές πως πολλά συμβάντα είναι πιθανό να εξαρτώνται από συνθήκες που σχετίζονται με την ημερομηνία / ώρα του συστήματος, ενώ είναι επίσης πολύ πιθανό κατά τη δοκιμή ενός συστήματος και την δημιουργία των σεναρίων να μην είναι εύκολη η πρόσβαση σε υλικό αντίστοιχο με αυτό το οποίο χρησιμοποιείται σε ένα παραγωγικό σύστημα (πχ αισθητήρες εγκατεστημένους σε ένα έξυπνο σπίτι). Για το λόγο αυτό η διεπαφή με το εξωτερικό περιβάλλον (τόσο όσον αφορά το υλικό όσο και το λογισμικό) να είναι σαφώς καθορισμένη και όσο πιο απλή γίνεται, προκειμένου να μπορεί να προσομοιωθεί ρεαλιστικά η λειτουργία των στοιχείων του περιβάλλοντος εκτέλεσης.

Αρχιτεκτονική Συστήματος

Η δομή του συστήματος έχει βασιστεί σε μια αρθρωτή αρχιτεκτονική που αποτελείται από τα παρακάτω διακριτά επίπεδα:



Εικόνα 30: Η αρχιτεκτονική του Context Engine

Το χαμηλότερο επίπεδο ουσιαστικά προσφέρει την απαραίτητη αφαίρεση από το περιβάλλον εκτέλεσης του συστήματος. Στο επίπεδο υπάρχουν τα δομικά μπλοκ που απαιτούνται για την αναπαράσταση και αποθήκευση των σεναρίων, την διεπαφή με τους αισθητήρες, την υπηρεσία καταγραφής συμβάντων (logging), και την διευκόλυνση της προσομοίωσης εκτέλεσης των σεναρίων. Η ύπαρξη του επιπέδου αυτού προσφέρει τη δυνατότητα φορητότητας του συστήματος, καθώς τα τμήματα εκείνα που αφορούν αλληλεπίδραση με υλικό (hardware) και το λειτουργικό σύστημα περιέχονται σε αυτό.

Το δεύτερο επίπεδο περιέχει τα δομικά αυτά στοιχεία που είναι απαραίτητα για την υλοποίηση των κύριων λειτουργιών του Context Engine. Αυτά αφορούν την διαχείριση της κατάστασης του συστήματος (Context State Management), τον μηχανισμό που παρακολουθεί τις τιμές των αισθητήρων που είναι συνδεδεμένοι

στο σύστημα, την εξέταση των συνθηκών που πρέπει να ισχύουν για την σήμανση κάποιου συμβάντος, και την ενημέρωση των actuators για τα συμβάντα που έχουν προκύψει. Τα παραπάνω δομικά στοιχεία είναι απαραίτητα για την εκτέλεση ενός σεναρίου, και αξιοποιούν τα δομικά στοιχεία που περιέχονται στο κατώτερο επίπεδο της αρχιτεκτονικής.

Το τρίτο και τελευταίο επίπεδο αφορά την ή τις διεπαφές για την επικοινωνία του context engine με τον εξωτερικό κόσμο. Στο επίπεδο αυτό θα μπορούσαν να υλοποιηθούν client-server αρχιτεκτονικές για απομακρυσμένη παραμετροποίηση και παρακολούθηση της λειτουργίας του Context Engine, εξελιγμένες γραφικές διεπαφές για monitoring & visualization της κατάστασης του συστήματος, κλπ. Καθώς αυτές οι επεκτάσεις θα οδηγούσαν σε σημαντική αύξηση της πολυπλοκότητας και των χρονικών απαιτήσεων για την υλοποίηση του συστήματος, στο επίπεδο αυτό σε πρώτη φάση αρκεί να δημιουργηθεί μια εφαρμογή που να παρέχει την αναγκαία γραφική διεπαφή για την δημιουργία και παραμετροποίηση σεναρίων, τον καθορισμό παραμέτρων προσομοίωσης και εκτέλεσης, και την παρακολούθηση των συμβάντων που καταγράφονται από το Context Engine.

7. Υλοποίηση

Εισαγωγή

Για την υλοποίηση του Context Engine επιλέχθηκε ως πλατφόρμα ανάπτυξης το Microsoft .NET framework, στην τελευταία έκδοση (4.5), ενώ ως γλώσσα προγραμματισμού επιλέχθηκε η C#. Η επιλογή αυτή έγινε με βάση τα χαρακτηριστικά της συγκεκριμένης πλατφόρμας ανάπτυξης, που συνοπτικά είναι τα παρακάτω:

- Εκτεταμένη βιβλιοθήκη ενσωματωμένη στο framework (Framework Class Library)
- Διαθεσιμότητα δωρεάν & open source εργαλείων ανάπτυξης
- Ύπαρξη πληθώρας βιβλιοθηκών ανοιχτού κώδικα πολύ υψηλής ποιότητας
- Μικρό learning curve για Java / C++ developers
- Συμβατότητα με πολλές πλατφόρμες (Windows, Linux, κ.λ.π)
- Υψηλά επίπεδα σταθερότητας (robustness) και απόδοσης
- Μεγάλες δυνατότητες κλιμάκωσης (scalability)
- Υψηλά επίπεδα ασφάλειας απέναντι σε κακόβουλες επιθέσεις καθώς ο κώδικας εκτελείται σε ένα ελεγχόμενο περιβάλλον αντίστοιχο με αυτό της Java Virtual Machine

Η εφαρμογή έχει χωριστεί σε ξεχωριστές βιβλιοθήκες και εφαρμογές που αφορούν στα αντίστοιχα επίπεδα της αρχιτεκτονικής που παρουσιάστηκε στην Εικόνα 30. Συγκεκριμένα έχει χωριστεί στα παρακάτω modules:

- SED.Engine.Base: Μια βιβλιοθήκη κώδικα που περιέχει τις απαραίτητες δομές δεδομένων για την υλοποίηση του πρώτου επιπέδου της αρχιτεκτονικής του συστήματος. Εδώ περιέχονται μεταξύ άλλων οι διάφορες δομές δεδομένων για την αποθήκευση σεναρίων, το επίπεδο αφαίρεσης για την διεπαφή με τους αισθητήρες, και η υλοποίηση της υπηρεσίας καταγραφής (logging). Η βιβλιοθήκη αυτή δεν έχει εξαρτήσεις σε βιβλιοθήκες κώδικα τρίτων (third party libraries).

- SED.Engine: Μια βιβλιοθήκη κώδικα που περιέχει την υλοποίηση της λειτουργικότητας του Context Engine, βασισμένη στις δομές που περιέχονται στην βιβλιοθήκη SED.Engine.Base. Η βιβλιοθήκη αυτή πρακτικά αντιστοιχεί στην υλοποίηση του δεύτερου επιπέδου της αρχιτεκτονικής του συστήματος. Έχει μόνο μια εξάρτηση από εξωτερικές βιβλιοθήκες, και συγκεκριμένα από την βιβλιοθήκη ανοιχτού κώδικα DynamicExpresso. Η βιβλιοθήκη αυτή επιτρέπει την δυναμική εκτέλεση κώδικα γραμμένου σε C#, και χρησιμοποιείται για τον έλεγχο των συνθηκών που πρέπει να ισχύουν προκειμένου να σημανθεί ένα συμβάν από το Context Engine. Η DynamicExpresso δημοσιεύεται υπό τους όρους της άδειας λογισμικού ανοιχτού κώδικα MIT στη διεύθυνση:
<https://github.com/davideicardi/DynamicExpresso> .
- SED.Engine.UI: Μια εφαρμογή με γραφική διεπαφή που υλοποιεί το τρίτο επίπεδο της αρχιτεκτονικής του συστήματος. Επιτρέπει στο χρήστη τον καθορισμό των παραμέτρων της προσομοίωσης και των παραμέτρων για την εκτέλεση των σεναρίων, τη δημιουργία, τροποποίηση και αποθήκευση σεναρίων, την εκτέλεση των σεναρίων και την εμφάνιση των συμβάντων που έχουν καταγραφεί από το Context Engine.

Υλοποίηση του context engine

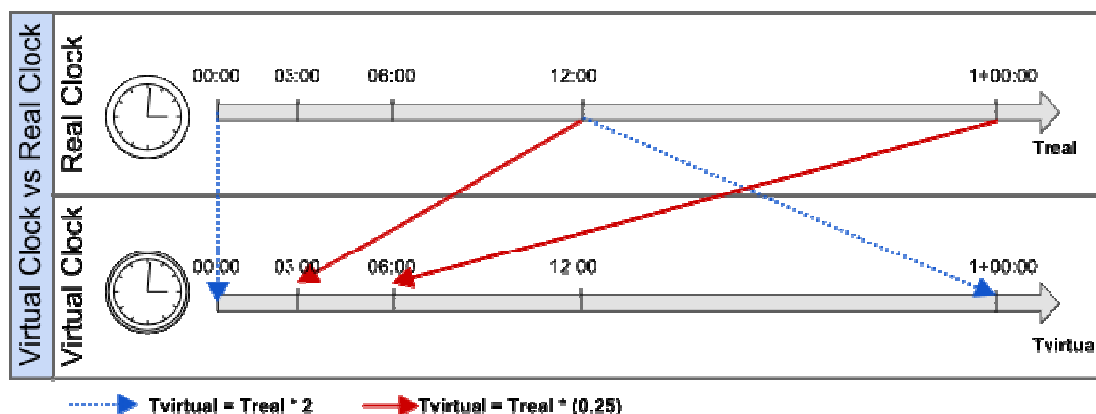
Στην ενότητα αυτή θα παρουσιαστεί πιο αναλυτικά η υλοποίηση των διαφόρων δομικών στοιχείων που απαρτίζουν το Context Engine. Στις επόμενες υποενότητες δεν ακολουθείται πάντα η δομή σε επίπεδα της αρχιτεκτονικής του συστήματος, αλλά η περιγραφή είναι κυρίως βασισμένη στις υπηρεσίες που παρέχονται από το κάθε module που αναφέρθηκε προηγουμένως.

Simulation Support

Όπως αναφέρθηκε στην προηγούμενη ενότητα, μια δυνατότητα που κρίθηκε ως εξαιρετικά υψηλής σημασίας κατά το σχεδιασμό ήταν η υποστήριξη του συστήματος για την ρεαλιστική προσομοίωση της εκτέλεσης σεναρίων. Βασικός παράγοντας για την επίτευξη του στόχου αυτού είναι η δυνατότητα να επιταχύνεται ή επιβραδύνεται κατά βούληση η ταχύτητα εκτέλεσης ενός σεναρίου, προκειμένου

να μπορεί να ελεγχθεί η σωστή λειτουργία του σε σύντομο χρονικό διάστημα. Για να επιτευχθεί κάτι τέτοιο απαιτείται η δημιουργία και χρήση ενός εικονικού ρολογιού (Virtual Clock).

Το εικονικό αυτό ρολόι αρκεί να προσφέρει ένα μηχανισμό γραμμικής διαστολής / συστολής του χρόνου, έτσι ώστε ένα καθορισμένο διάστημα T_{real} στο πραγματικό ρολόι να αντιστοιχεί σε ένα χρονικό διάστημα $T_{virtual} = a * T_{real}$ σε εικονικό χρόνο, όπου a μια σταθερά, για την οποία ισχύει $a > 0$. Αν το a πάρει την τιμή 1 τότε η τιμή που επιστρέφει το εικονικό ρολόι ταυτίζεται με την πραγματική τρέχουσα ώρα, ενώ αν $a = 0.1$ τότε το εικονικό ρολόι θα πηγαίνει 10 φορές πιο «αργά» σε σχέση με το πραγματικό. Η τιμή της σταθεράς a μπορεί να υπολογιστεί εύκολα επιλέγοντας την πραγματική χρονική διάρκεια που αντιστοιχεί σε διάρκεια μιας ημέρας στο εικονικό ρολόι, δηλαδή την περίοδο του εικονικού ρολογιού. Για παράδειγμα, για ένα ρολόι που θα διατρέχει εικονικά το διάστημα μιας ημέρας σε διάστημα 10 λεπτών, ο χρόνος πρέπει να επιταχυνθεί κατά $(24 * 60) / 10 = 144$ φορές. Στην εικόνα γίνεται αντιπαράθεση 2 εικονικών ρολογιών, ένα με περίοδο 12 ωρών (με μπλε) και ένα με περίοδο 4 ημερών (με κόκκινο) σε σχέση με ένα πραγματικό ρολόι.



Εικόνα 31: Επισκόπηση της λειτουργίας του εικονικού ρολογιού

Η λειτουργία του εικονικού ρολογιού παρέχεται από την κλάση VirtualClock που βρίσκεται στη βιβλιοθήκη SED.Engine.Base. Κατά την αρχικοποίηση της απαιτείται να οριστεί το χρονικό διάστημα που αντιστοιχεί στην περίοδο μιας ημέρας, ενώ προαιρετικά το εικονικό ρολόι μπορεί να ξεκινήσει από μια καθορισμένη χρονική στιγμή, διαφορετικά ξεκινά από την τρέχουσα πραγματική ημερομηνία και ώρα.

Από εκεί και έπειτα, η εικονική ημερομηνία και ώρα μπορούν να ανακτηθούν από την στατική ιδιότητά της κλάσης που ονομάζεται `Now`, κατ' αντιστοιχία με την στατική ιδιότητα `DateTime.Now` που παρέχεται από το `.NET Framework Class Library` και παρέχει την πραγματική τρέχουσα ημερομηνία και ώρα.

Sensor Support

Ένας από τους βασικούς στόχους που τέθηκαν κατά το σχεδιασμό του συστήματος αφορούσε την υποστήριξη απεριόριστων σε τύπο και αριθμό αισθητήρων. Ένας αισθητήρας είναι μια συσκευή που ανιχνεύει ή / και μετρά κάποιο φυσικό μέγεθος και το μετατρέπει σε κάποιο σήμα που μπορεί να μετρηθεί από κάποιο μηχανισμό παρατήρησης, πχ ένα όργανο. Για παράδειγμα, μια αναλογική ζυγαριά είναι ένας αισθητήρας βαρύτητας, που μετρά τη μάζα ενός αντικειμένου και τη μετατρέπει σε κάποιο μηχανικό σήμα που κινεί μια βελόνα (δείκτη) πάνω σε μια προσαρμοσμένη κλίμακα με ενδείξεις, η οποία χρησιμεύει ως μηχανισμός παρατήρησης.

Ένας αισθητήρας πρέπει να είναι ευαίσθητος μόνο στις μεταβολές του μεγέθους που μετρά, και να μην επηρεάζει την τιμή του κατά τη μέτρηση. Σε διαφορετική περίπτωση προκύπτουν αποκλίσεις ανάμεσα στην πραγματική και την εκτιμώμενη τιμή του μετρούμενου μεγέθους. Τυπικά η έξοδος ενός αισθητήρα αποτελεί γραμμική ή λογαριθμική συνάρτηση του μετρούμενου μεγέθους. Για παράδειγμα, ένας αισθητήρας μπορεί να μετρά θερμοκρασία και να βγάζει ως έξοδο μια ηλεκτρική τάση γραμμικά ανάλογη με την τιμή εισόδου. Στις περιπτώσεις που η τιμή εξόδου ενός αισθητήρα είναι αναλογική και πρέπει να υποστεί επεξεργασία από μια ψηφιακή συσκευή, είναι απαραίτητο να μετατραπεί σε μια μορφή ψηφιακού σήματος με τη βοήθεια ενός `Analog to Digital Converter`.

Κάθε αισθητήρας είναι πιθανό να επιστρέφει τιμή διαφορετικού τύπου, την οποία ανακτά ή υπολογίζει με διαφορετικό τρόπο. Το `Context Engine` πρέπει να μην έχει κάποια εξάρτηση από συγκεκριμένους τύπους τιμών αισθητήρων. Για παράδειγμα, ένας αισθητήρας θερμοκρασίας μπορεί να επιστρέφει ως τιμή έναν αριθμό κινητής υποδιαστολής, ενώ ένας αισθητήρας επιτάχυνσης (`accelerometer`) μπορεί να

επιστρέφει ως τιμή ένα σετ από πολλαπλές ανεξάρτητες τιμές που χρησιμοποιούνται για να περιγράψουν το διάνυσμα της επιτάχυνσης (X, Y, Z, a) όπου οι X, Y, Z περιγράφουν την κατεύθυνση του διανύσματος και το a την τιμή του. Οι (απλές ή σύνθετες) τιμές που επιστρέφει ένας αισθητήρας πρέπει να είναι συγκρίσιμες μεταξύ τους, αφού στην περίπτωση του Context Engine δεν μας ενδιαφέρει ο συγκεκριμένος τύπος της τιμής, αλλά η δυνατότητα σύγκρισης της τρέχουσας τιμής με τις τιμές που παρατηρήθηκαν στις προηγούμενες μετρήσεις, προκειμένου να είναι εύκολος ο εντοπισμός μιας αλλαγής της.

Η διεπαφή λοιπόν που οφείλει να υλοποιεί κάθε αισθητήρας πρέπει να είναι ίδια, και αφορά την ανάκτηση μιας (απλής ή σύνθετης) τιμής. Δεν μας ενδιαφέρει ο τρόπος με τον οποίο ανακτάται η τιμή αυτή, η διαδικασία της μετατροπής του φυσικού μεγέθους σε ψηφιακό σήμα, κλπ. Υπό αυτές τις προϋποθέσεις, η προγραμματιστική διεπαφή για έναν αισθητήρα αρκεί να υλοποιεί μόνο μια μέθοδο ή ιδιότητα, η οποία θα επιστρέφει την τρέχουσα τιμή που μετρά ο αισθητήρας. Για την πρακτική διαχείριση των αισθητήρων από την εκάστοτε εφαρμογή, στη διεπαφή αυτή έχουν προστεθεί κάποιες επιπλέον ιδιότητες, όπως όνομα, περιγραφή και τοποθεσία του αισθητήρα. Αυτή τη στιγμή η τοποθεσία δεν χρησιμοποιείται, αλλά σε μια άλλη υλοποίηση θα μπορούσε να υπάρχει υποστήριξη για φορητούς αισθητήρες, πχ αισθητήρες πάνω σε ρούχα, οπότε ο αισθητήρας θα μπορούσε να αναφέρει και την τρέχουσα τοποθεσία του μέσα σε ένα καθορισμένο χώρο.

```
interface ISensor
{
    // The current sensor value
    IComparable Value { get; }

    // The name of the sensor. Sensors are uniquely identified by name.
    string Name { get; set; }

    // The sensor description
    string Description { get; set; }

    // The sensor location
    string Location { get; set; }
}
```

Code listing 1: Η βασική προγραμματιστική διεπαφή για ένα αισθητήρα

Προκειμένου να είναι πιο απλή η δημιουργία προγραμματιστικών διεπαφών για αισθητήρες με γνωστό τύπο τιμής, έχει αξιοποιηθεί η υποστήριξη για generics του .NET Framework. Έτσι, έχει δημιουργηθεί μια generic μορφή της διεπαφής των αισθητήρων, που εκτός από συγκεκριμένου τύπου τιμή (με περιορισμό στους τύπους που μπορούν να χρησιμοποιηθούν ώστε να εξασφαλίζεται πως οι τιμές είναι συγκρίσιμες) περιέχει και μια ιδιότητα για την επιστροφή του εύρους τιμών (range) που θεωρούνται έγκυρες για το συγκεκριμένο αισθητήρα, η οποία μπορεί να χρησιμοποιηθεί για σφάλματα κατά την ανάκτηση της τιμής από κάποιο αισθητήρα. Η γενικευμένη διεπαφή λοιπόν για τους αισθητήρες έχει ως εξής (εμφανίζονται μόνο τα πιο σημαντικά μέλη):

```
interface ISensor<T> : ISensor where T : IComparable<T>
{
    // The current sensor value
    new T Value { get; }

    // The default sensor value
    new T Default { get; }

    // The valid / allowed sensor value range
    Range<T> ValueRange { get; }
}
```

Code listing 2: Η γενικευμένη προγραμματιστική διεπαφή για ένα αισθητήρα

Στη διεπαφή αυτή έχει επαναοριστεί η ιδιότητα με όνομα Value που επιστρέφει την τρέχουσα τιμή του αισθητήρα ώστε η τιμή να επιστρέφεται με τον συγκεκριμένο τύπο τιμής του αισθητήρα, και όχι απλά ως IComparable, έτσι ώστε να μπορεί να χρησιμοποιηθεί χωρίς χρειάζεται αλλαγή τύπου (casting). Για παράδειγμα, ένας αισθητήρας θερμοκρασίας θα μπορούσε να κατασκευαστεί υλοποιώντας τη διεπαφή αυτή ως ISensor<double>, υποδηλώνοντας με αυτόν τον τρόπο πως ο τύπος της τιμής που επιστρέφει είναι αριθμός κινητής υποδιαστολής διπλής ακρίβειας (double). Ο τύπος Range<T> αντιπροσωπεύει ένα εύρος τιμών τύπου T, και έχει υλοποιηθεί στη βιβλιοθήκη SED.Engine.Base, αλλά αυτή τη στιγμή δε χρησιμοποιείται για τον έλεγχο των επιτρεπόμενων τιμών που μπορούν να επιστρέψει κάποιος αισθητήρας κατά την εκτέλεση των σεναρίων.

Για τη διευκόλυνση της διαδικασίας υλοποίησης νέων τύπων αισθητήρων, έχει φτιαχτεί μια αφηρημένη κλάση με όνομα `SensorBase<T>` που υλοποιεί τη διεπαφή `ISensor<T>` όσον αφορά τις βασικές ιδιότητες. Έτσι, για την υποστήριξη ενός νέου τύπου αισθητήρα, αρκεί κανείς να δημιουργήσει μια υποκλάση της `SensorBase<T>` η οποία θα χρειαστεί να υλοποιεί μόνο τη διαδικασία με την οποία γίνεται η ανάκτηση της τιμής από τον πραγματικό αισθητήρα. Η τεχνική αυτή έχει χρησιμοποιηθεί για την υλοποίηση της προσομοίωσης των αισθητήρων που απαιτούνται για την εκτέλεση του σεναρίου που παρουσιάζεται στο επόμενο κεφάλαιο.

Context State Management

Όπως αναφέρθηκε ήδη, ένα Context Engine πρέπει να έχει κάποιο μηχανισμό για να αποθηκεύει όλη την εικόνα της κατάστασης του έξυπνου συστήματος. Αυτό σημαίνει πως πρέπει να παρακολουθεί τις συνθήκες λειτουργίας του, τους χρήστες του, τους αισθητήρες και την κατάστασή τους, κλπ. Ένας απλός τρόπος για να επιτευχθεί η αναπαράσταση και αποθήκευση αυτής της πληροφορίας είναι η δημιουργία ενός ευρετηρίου που θα αποθηκεύει αυτά τα στοιχεία του context ως μεταβλητές με μοναδικό αναγνωριστικό. Για το σκοπό αυτό έχουν δημιουργηθεί δύο κλάσεις, οι `ContextItem` και `Context`.

Ένα `ContextItem` αντιπροσωπεύει ένα στοιχείο του Context του συστήματος – για παράδειγμα μπορεί να είναι το αναγνωριστικό του RFID Tag του χρήστη του συστήματος με όνομα 'Alice', η τρέχουσα θερμοκρασία του περιβάλλοντος, η τρέχουσα ημερομηνία και ώρα, ο αριθμός των ατόμων που βρίσκονται σε ένα συγκεκριμένο δωμάτιο ενός έξυπνου σπιτιού, κλπ. Ένα `ContextItem` μπορεί να χρησιμοποιείται ως σταθερά (read-only variable), προκειμένου να χρησιμοποιηθεί ως παράμετρος ενός σεναρίου, ενώ κάποιο άλλο μπορεί να χρησιμοποιείται ως απλή μεταβλητή για να αποθηκεύσει τιμές που ενδέχεται να αλλάζουν κατά την εκτέλεση του σεναρίου, όπως πχ η θερμοκρασία στο εσωτερικό ενός δωματίου ενός έξυπνου σπιτιού.

Στην πιο απλή περίπτωση, ένα ContextItem μπορεί να θεωρηθεί ως ένα ζευγάρι ονόματος - τιμής (key-value pair). Στη συγκεκριμένη υλοποίηση, οι τιμές είναι τύπου συμβολοσειράς (string), οπότε αν απαιτείται ειδική επεξεργασία τους (οτιδήποτε άλλο εκτός από συγκρίσεις και αναθέσεις τιμών) από κάποιο σενάριο είναι απαραίτητο να γίνονται πρώτα οι κατάλληλες μετατροπές στον τύπο δεδομένων. Τα Context Items πρέπει να υποστηρίζουν συγκρίσεις και έλεγχο ισότητας μεταξύ τους. Επίσης ένα ContextItem πρέπει να φροντίζει να «καθαρίζει» το αναγνωριστικό του από μη έγκυρους χαρακτήρες, προκειμένου το τελευταίο να μπορεί να χρησιμοποιηθεί σε συνθήκες χωρίς να προκαλεί συντακτικά σφάλματα.

```
class ContextItem : IComparable<ContextItem>, IEquatable<ContextItem>
{
    /// Gets or sets the name of the context item
    public string Name
    {
        get; set;
    }

    /// Gets or sets the value of the context item
    public string Value
    {
        get; set;
    }
}
```

Code listing 3: Ο ορισμός ενός Context Item

Η κλάση Context είναι υπεύθυνη για την αποθήκευση των Context Items και πρέπει να υποστηρίζει την ανάκτησή τους με βάση το μοναδικό όνομα του καθενός, χωρίς να έχει σημασία ο τύπος της τιμής του καθενός. Εκτός της ανάκτησης των Context Items το Context υποστηρίζει την απαρίθμησή τους (enumeration) ώστε να μπορεί κανείς να διατρέξει εύκολα τη λίστα με τα Context Items. Για την ανάκτηση των Context Items με τη χρήση του ονόματος έχει υλοποιηθεί ένας Indexer, οπότε κατά τη χρήση η ανάκτηση ενός Context Item μπορεί να γίνει με τη χρήση των τελεστών [], δηλαδή κατά αντιστοιχία με την ανάκτηση του στοιχείου ενός πίνακα (array).

Είναι σημαντικό η διαχείριση των Context Items να γίνεται από το Context με τρόπο ασφαλή για πρόσβαση από πολλαπλά νήματα εκτέλεσης (δηλαδή να υποστηρίζει multithreading), καθώς ενδέχεται πολλά σενάρια ταυτόχρονα να εκτελούν ενέργειες

πάνω στο Context προσπαθώντας να αλλάξουν / ανακτήσουν την τιμή του ίδιου Context Item. Για το λόγο αυτό ως δομή δεδομένων για την αποθήκευση των Context Items χρησιμοποιήθηκε ένα εξειδικευμένο ευρετήριο τύπου ConcurrentDictionary<string, ContextItem> (υποστηρίζεται εγγενώς στο Microsoft .NET 4.0 και έπειτα, και βρίσκεται στο namespace System.Collections.Concurrent). Η δομή δεδομένων αυτή επιτρέπει ταυτόχρονη πρόσβαση σε πολλά νήματα εκτέλεσης, ενώ παράλληλα η υλοποίηση του δεν χρησιμοποιεί κλειδώματα (lock-free) με αποτέλεσμα να μην οδηγεί στη δημιουργία αδιεξόδων (deadlocks) κατά την εκτέλεση. Ένα τμήμα της υλοποίησης της κλάσης Context παρουσιάζεται στο Code listing 4, όπου εμφανίζονται οι λεπτομέρειες που αναφέρθηκαν εδώ.

```
class Context : IEnumerable<ContextItem>
{
    private ConcurrentDictionary<string, ContextItem> items;

    /// Indexer for items contained in the collection.
    /// <param name="key">The name of the item</param>
    /// <returns>The item with the given name, null if not found</returns>
    public ContextItemDescriptor this[string key]
    {
        get
        {
            if (string.IsNullOrEmpty(key))
                throw new ArgumentException();

            ContextItemDescriptor result = null;
            items.TryGetValue(key, out result);
            return result;
        }
    }

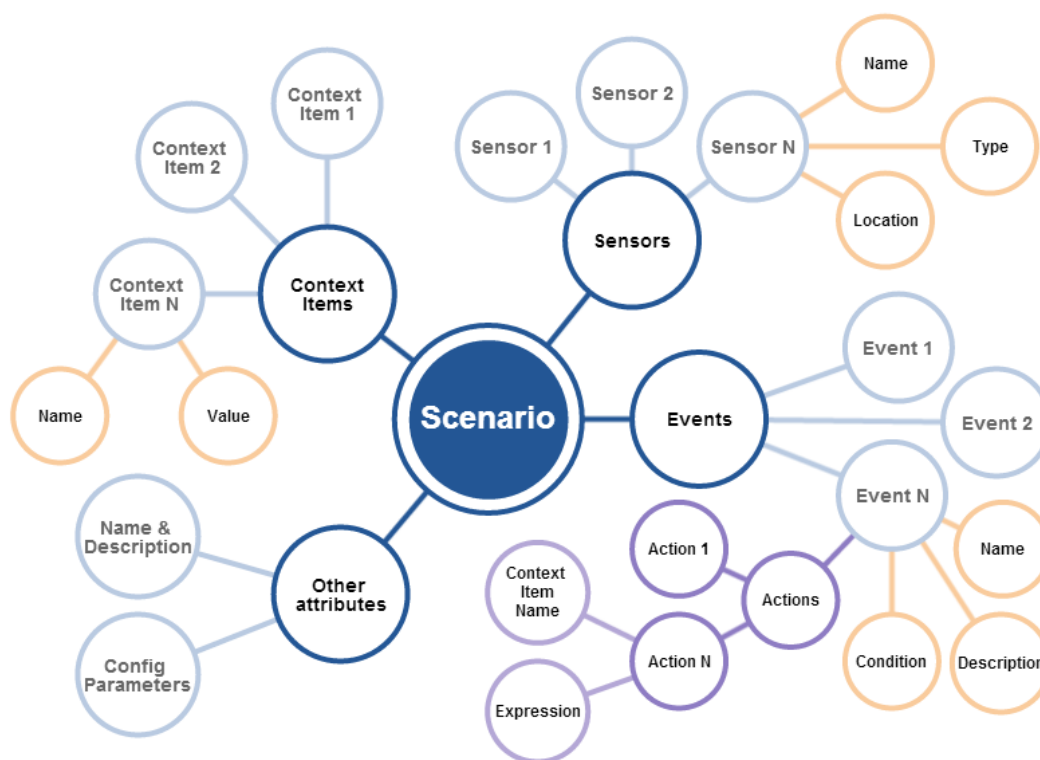
    // Code omitted for brevity
}
```

Code listing 4: Τμήμα του ορισμού της κλάσης Context

Scenario Management

Ένα σενάριο αποτελεί περιγραφή μιας πλήρους περίπτωσης χρήσης του Context Engine. Με άλλα λόγια ένα σενάριο περιγράφει όλη τη λογική την οποία πρέπει να εκτελέσει το έξυπνο σύστημα υπό δεδομένες συνθήκες. Σε ένα σενάριο μπορεί να περιέχονται μόνο εκείνες οι ενέργειες που αφορούν ένα μικρό υποσύστημα ενός έξυπνου περιβάλλοντος, ή ακόμη και όλη η συμπεριφορά που πρέπει αυτό να

υλοποιεί. Για παράδειγμα, ένα σενάριο μπορεί να περιγράφει τον τρόπο λειτουργίας ενός συστήματος για τον αυτοματισμό του φωτισμού σε ένα δωμάτιο ενός έξυπνου σπιτιού, ή να περιέχει όλη τη λογική για όλα τα υποσυστήματα ενός έξυπνου αυτοκινήτου. Το σενάριο περιλαμβάνει επίσης τους ορισμούς των αισθητήρων που είναι συνδεδεμένοι στο έξυπνο περιβάλλον, καθώς και τα στοιχεία του Context που είναι απαραίτητα για τις επιμέρους ενέργειες τις οποίες υλοποιεί, και εν τέλει καθορίζει τις συσχετίσεις μεταξύ των δομικών στοιχείων αυτών. Τα στοιχεία που περιγράφει ένα σενάριο και οι σχέσεις ιεραρχίας μεταξύ τους παρουσιάζονται στην Εικόνα 32.



Εικόνα 32: Η ιεραρχική δομή των στοιχείων ενός σεναρίου

Δεδομένου ότι το Context Engine είναι επιθυμητό να μπορεί να εκτελέσει πολλά αυτόνομα σενάρια ταυτόχρονα, κατά την εκτέλεση ενός σεναρίου πρέπει να διατηρεί ανεξάρτητο state για το καθένα. Έτσι, σε κάθε σενάριο περιλαμβάνεται η περιγραφή όλων των Context Items που απαρτίζουν την κατάσταση του Context Engine για το συγκεκριμένο σενάριο. Επίσης, καθώς ένας αισθητήρας ενδέχεται να χρησιμοποιείται από περισσότερα από ένα σενάρια την ίδια στιγμή, κάθε σενάριο ορίζει τους αισθητήρες που είναι απαραίτητοι για τη λειτουργία του.

Η εκτέλεση ενός σεναρίου ουσιαστικά συνίσταται στην εξέταση των συνθηκών που πρέπει να ισχύουν προκειμένου να σημειωθεί ένα συμβάν (Event). Έτσι, ένα σενάριο περιγράφει επίσης τους ορισμούς των συμβάντων τα οποία πρέπει να σημειωθούν. Κάθε συμβάν χαρακτηρίζεται από ένα όνομα, μια περιγραφή, και μια (απλή ή σύνθετη) συνθήκη. Η συνθήκη ενός συμβάντος δεν είναι παρά μια έκφραση (expression) που όταν εκτελεστεί πρέπει να επιστρέφει μια λογική τιμή. Όταν η τιμή αυτή είναι αληθής (true) αυτό σημαίνει πως πρέπει να σημειωθεί το αντίστοιχο συμβάν. Η συνθήκη αυτή δεν είναι παρά μια συμβολοσειρά που μπορεί να περιέχει αναφορές (references) σε ονόματα αισθητήρων, στοιχεία του Context τα οποία χρησιμοποιούνται ως μεταβλητές, και στοιχεία του περιβάλλοντος εκτέλεσης, όπως η τρέχουσα ημερομηνία/ώρα του συστήματος. Η έκφραση αυτή πρέπει να είναι συντακτικά σωστή με βάση το συντακτικό του εκάστοτε διερμηνευτή (Parser) που αναλαμβάνει να την εκτελέσει, ενώ ο parser αναλαμβάνει να υπολογίσει την λογική τιμή που προκύπτει.

Στη συγκεκριμένη υλοποίηση, όπως αναφέρθηκε ήδη, έχει χρησιμοποιηθεί ως parser και condition evaluator το DynamicExpresso. Ως εκ τούτου, η συνθήκη για ένα Event είναι ουσιαστικά μια έγκυρη έκφραση τύπου bool στη γλώσσα C#, και η οποία τυπικά περιλαμβάνει συγκρίσεις ανάμεσα σε τιμές αισθητήρων και σταθερές ή στοιχεία του Context. Αυτός είναι και ο λόγος που είναι σημαντικό οι τιμές που επιστρέφονται από τους αισθητήρες να είναι συγκρίσιμες μεταξύ τους, καθώς πολλές φορές δεν έχει σημασία η ακριβής τιμή τους αλλά η σχέση τους με κάποια άλλη τιμή ή σταθερά.

Προκειμένου μια συνθήκη να χρησιμοποιήσει την τιμή ενός στοιχείου του Context για να πραγματοποιήσει κάποιο έλεγχο, έχει οριστεί μια μεταβλητή με όνομα context που περιέχει το Context για το συγκεκριμένο σενάριο. Η μεταβλητή αυτή αναγνωρίζεται από τον parser και με τη χρήση των τελεστών [] και όρισμα το όνομα ενός context item επιστρέφει το item αυτό, χάρη στον Indexer που έχει υλοποιηθεί στην κλάση Context. Επιπλέον, ειδικά για την τρέχουσα ημερομηνία και ώρα του συστήματος, προκειμένου να υποστηρίζεται σωστά η προσομοίωση της εκτέλεσης

των σεναρίων, έχει δημιουργηθεί μια μέθοδος με όνομα `CurrentDateTime()` η οποία επιστρέφει μια τιμή τύπου `DateTime` που περιέχει την τρέχουσα ημερομηνία και ώρα του συστήματος, όπως αυτή προκύπτει από το εικονικό ρολόι του `Context Engine`. Τέλος, σε κάθε συνθήκη μπορούν να χρησιμοποιηθούν στατικές κλάσεις και στατικά μέλη κλάσεων του `.NET Framework Class Library`, χωρίς να χρειάζεται να δημιουργηθούν αντίστοιχες μεταβλητές. Ο Πίνακας 4: παρουσιάζει παραδείγματα έγκυρων συνθηκών για τη συγκεκριμένη υλοποίηση, μαζί με την επεξήγησή τους.

Συνθήκη	Επεξήγηση
<code>temperatureSensor.Value > 10</code>	Η τιμή του αισθητήρα με όνομα <code>temperatureSensor</code> είναι μεγαλύτερη από 10 βαθμούς
<code>context["current_user"].Value == "Mary"</code>	Η τιμή του <code>Context Item</code> με όνομα <code>current_user</code> είναι "Mary"
<code>CurrentDateTime().Hour < 9</code>	Η ιδιότητα με όνομα <code>Hour</code> της τρέχουσας ημερομηνίας & ώρας (δηλαδή η ώρα της τρέχουσας στιγμής) είναι μικρότερη του 9. Δηλαδή, η τρέχουσα ώρα είναι πριν τις 9 το πρωί
<code>temperatureSensor.Value >= 15.5 && context["current_user"].Value == "Mary" && CurrentDateTime().DayOfWeek != DayOfWeek.Sunday</code>	Ο αισθητήρας θερμοκρασίας με όνομα <code>temperatureSensor</code> αναφέρει τιμή μεγαλύτερη των 15.5 βαθμών και ο τρέχων χρήστης του συστήματος είναι η "Mary" και δεν είναι ημέρα Κυριακή

Πίνακας 4: Παραδείγματα έγκυρων εκφράσεων συνθηκών για συμβάντα του `Context Engine`

Σε ένα έξυπνο σύστημα εκτός από τους αισθητήρες και το `Context Engine` τυπικά υπάρχει κάποιος μηχανισμός που να δίνει τη δυνατότητα στο σύστημα να αντιδράσει όταν προκύπτουν κάποια συμβάντα. Για παράδειγμα, όταν ένας αισθητήρας της ταχύτητας του αέρα ανιχνεύσει επικίνδυνες ταχύτητες αέρα πρέπει να μαζευτούν οι τέντες, ενώ όταν ένας αισθητήρας έντασης του φωτός εντοπίσει πως έχει ξημερώσει πρέπει να σβήσουν τα φώτα σε κάποιο χώρο. Οι μηχανισμοί που υλοποιούν αυτές τις ενέργειες ονομάζονται `Actuators`, και ο τρόπος λειτουργίας τους δεν περιλαμβάνεται στο πλαίσιο της εργασίας αυτής. Ωστόσο, για να μπορούν να εκτελεστούν σύνθετα σενάρια από την υλοποίηση του `Context Engine` ήταν απαραίτητο να υποστηριχθεί έστω σε βασικό επίπεδο μια υποτυπώδης υλοποίηση `Actuators`.

Για το σκοπό αυτό, δημιουργήθηκε η έννοια των Event Actions. Τα Actions αυτά περιγράφουν ενέργειες που πρέπει να εκτελεστούν από το Context Engine όταν ικανοποιηθούν οι συνθήκες για τη σήμανση ενός γεγονότος, δηλαδή ένα Event συσχετίζεται με μία ή περισσότερες ενέργειες που εκτελούνται ακολουθιακά κάθε φορά που το Context Engine σημαίνει το συμβάν αυτό. Σε μια κανονική υλοποίηση ενός Context Engine η διαχείριση των Actuators θα γινόταν από κάποιο άλλο module, αλλά στην συγκεκριμένη περίπτωση η εκτέλεσή τους γίνεται από την μηχανή εκτέλεσης των σεναρίων (Scenario Execution Engine), η οποία περιγράφεται στην επόμενη ενότητα.

Ένα Event Action έχει δύο ιδιότητες, το όνομα ενός Context Item και ένα expression. Όταν εκτελείται ένα Action ανατίθεται στο Context Item με το συγκεκριμένο όνομα μια τιμή η οποία προκύπτει από το evaluation του expression, κατά αντίστοιχο τρόπο με αυτόν που χρησιμοποιείται για το evaluation της συνθήκης ενός συμβάντος. Ο τύπος της έκφρασης δεν έχει σημασία όπως στην περίπτωση της συνθήκης του Event – το αποτέλεσμα του expression evaluation μετατρέπεται πάντα σε string.

Όπως αναφέρθηκε νωρίτερα, βασικοί στόχοι του σχεδιασμού του συστήματος ήταν ότι το Context Engine πρέπει να υποστηρίζει απεριόριστο αριθμό σεναρίων, τα οποία ενδεχομένως να εκτελούνται ταυτόχρονα, και τα σενάρια αυτά πρέπει να μπορούν να αποθηκευτούν σε μια μορφή που να είναι φορητή ανάμεσα σε λειτουργικά συστήματα / περιβάλλοντα εκτέλεσης, και να είναι δυνατή η χειροκίνητη δημιουργία και επεξεργασία τους. Για τους παραπάνω λόγους αποφασίστηκε τα σενάρια να αποθηκεύονται σε μορφή XML εγγράφων (eXtensible Markup Language).

Για την αποτελεσματική αποθήκευση και φόρτωση των σεναρίων από την αναπαράσταση σε XML αξιοποιήθηκε η υποστήριξη του .NET Framework Class Library για σειριακοποίηση δομών δεδομένων σε μορφή XML (XML Serialization). Ένα παράδειγμα ενός απλού σεναρίου με 1 Event που περιέχει 1 Action, 2 Context Items, και 1 Sensor εμφανίζεται στο Code listing 5. Είναι σαφές ότι κάποιος που

γνωρίζει τον τρόπο με τον οποίο μπορεί να συνταχθεί ένα σενάριο μπορεί να δημιουργήσει ένα σενάριο ή να τροποποιήσει ένα υπάρχον χρησιμοποιώντας ένα απλό επεξεργαστή κειμένου.

```
<?xml version="1.0" encoding="utf-8"?>
<ScenarioDescriptor>
  <Name>Test</Name>
  <Description>Test Scenario</Description>
  <CaseSensitiveExpressionEvaluation>>false</CaseSensitiveExpressionEvaluation>
  <ContextItems>
    <ContextItemDescriptor>
      <Name>temperature_threshold</Name>
      <Value>10</Value>
    </ContextItemDescriptor>
    <ContextItemDescriptor>
      <Name>temperature_ishigh</Name>
      <Value>0</Value>
    </ContextItemDescriptor>
  </ContextItems>
  <Events>
    <ContextEventDescriptor>
      <Name>TemperatureThresholdReachedEvent</Name>
      <Description>Occurs when the temperature rises above 10 C</Description>
      <Condition>tempsensor.Value > 10.0 & &
context["temperature_ishigh"].Value == "0"</Condition>
      <Actions>
        <ContextEventAction>
          <Expression>"1"</Expression>
          <ContextItemName>temperature_ishigh</ContextItemName>
        </ContextEventAction>
      </Actions>
      <Sensors>
        <SensorNameDescriptor>
          <Name>tempsensor</Name>
        </SensorNameDescriptor>
      </Sensors>
    </ContextEventDescriptor>
  </Events>
  <Sensors>
    <SensorDescriptor>
      <Name>tempsensor</Name>
      <Description>Temperature sensor</Description>
      <Location>home</Location>
      <Type>TemperatureSensor</Type>
    </SensorDescriptor>
  </Sensors>
  <SensorPollingInterval />
</ScenarioDescriptor>
```

Code listing 5: Η αναπαράσταση ενός σεναρίου σε μορφή XML

Στο σενάριο που φαίνεται πιο πάνω εμφανίζεται η δομή και η ιεραρχία που χρησιμοποιείται για όλα τα στοιχεία που αναφέρονται στο διάγραμμα στην Εικόνα 32. Εκτός των άλλων στοιχείων που περιγράφηκαν στις προηγούμενες παραγράφους, εμφανίζονται και οι ιδιότητες που αφορούν το configuration της εκτέλεσης του σεναρίου, οι οποίες δεν αναφέρθηκαν ως τώρα. Αυτές είναι οι SensorPollingInterval, που αφορά την περίοδο (με βάση το χρόνο του εικονικού ρολογιού) μεταξύ διαδοχικών ανακτήσεων της τιμής των αισθητήρων, και

CaseSensitiveExpressionEvaluation που καθορίζει το αν τα αναγνωριστικά που χρησιμοποιούνται στις εκφράσεις για τις συνθήκες και τις ενέργειες συμβάντων θα είναι Case Sensitive ή όχι.

Scenario Execution Engine

Η εκτέλεση ενός σεναρίου από το Context Engine είναι μια διαδικασία που αποτελείται από τα παρακάτω βήματα:

1. Φόρτωση του σεναρίου από την μορφή XML
2. Για όσο δεν χρειάζεται να διακοπεί η εκτέλεση του σεναρίου
 - 2.1. Ανακτώνται οι τιμές των αισθητήρων
 - 2.2. Επιλέγονται τα συμβάντα των οποίων οι συνθήκες σχετίζονται με τους αισθητήρες των οποίων η τιμή έχει αλλάξει ή εξαρτώνται από την ώρα του συστήματος
 - 2.3. Για κάθε ένα από τα επιλεγμένα υποψήφια συμβάντα
 - 2.3.1. Γίνεται έλεγχος της συνθήκης του συμβάντος
 - 2.3.2. Αν η συνθήκη είναι αληθής
 - 2.3.2.1. Σημαίνεται το συμβάν
 - 2.3.2.2. Για κάθε ενέργεια που σχετίζεται με το συμβάν
 - 2.3.2.2.1. Εκτελείται η αντίστοιχη ενέργεια
 - 2.3.3. Αν η συνθήκη δεν είναι αληθής
 - 2.3.3.1. Καταγράφεται ο έλεγχος της συνθήκης του συμβάντος

Πολλές από τις επιμέρους υπο-διαδικασίες που αναφέρονται στον παραπάνω αλγόριθμο μπορούν να εκτελεστούν ανεξάρτητα από τις υπόλοιπες. Για παράδειγμα, η διαδικασία της ανάκτησης των τιμών των αισθητήρων μπορεί να πραγματοποιηθεί σε ένα ανεξάρτητο νήμα εκτέλεσης, ώστε οι τιμές να είναι διαθέσιμες όταν τις χρειαστεί κάποια άλλη διαδικασία. Με τον τρόπο αυτό μπορούν να καταγραφούν και να σημειθούν αλλαγές στις τιμές των αισθητήρων, χωρίς να καθυστερεί η διαδικασία ελέγχου των συνθηκών των συμβάντων, η οποία μπορεί να εκτελεστεί παράλληλα με ασύγχρονο τρόπο για κάθε συμβάν. Ωστόσο κάτι τέτοιο δεν είναι επιτρεπτό να συμβεί κατά τη διάρκεια του ελέγχου του προηγούμενου σετ των υποψήφιων συμβάντων. Με βάση το σκεπτικό αυτό, η

μηχανή εκτέλεσης των σεναρίων χρησιμοποιεί διαφορετικά νήματα εκτέλεσης (threads) και μηχανισμούς συγχρονισμού για κάθε σενάριο που εκτελείται.

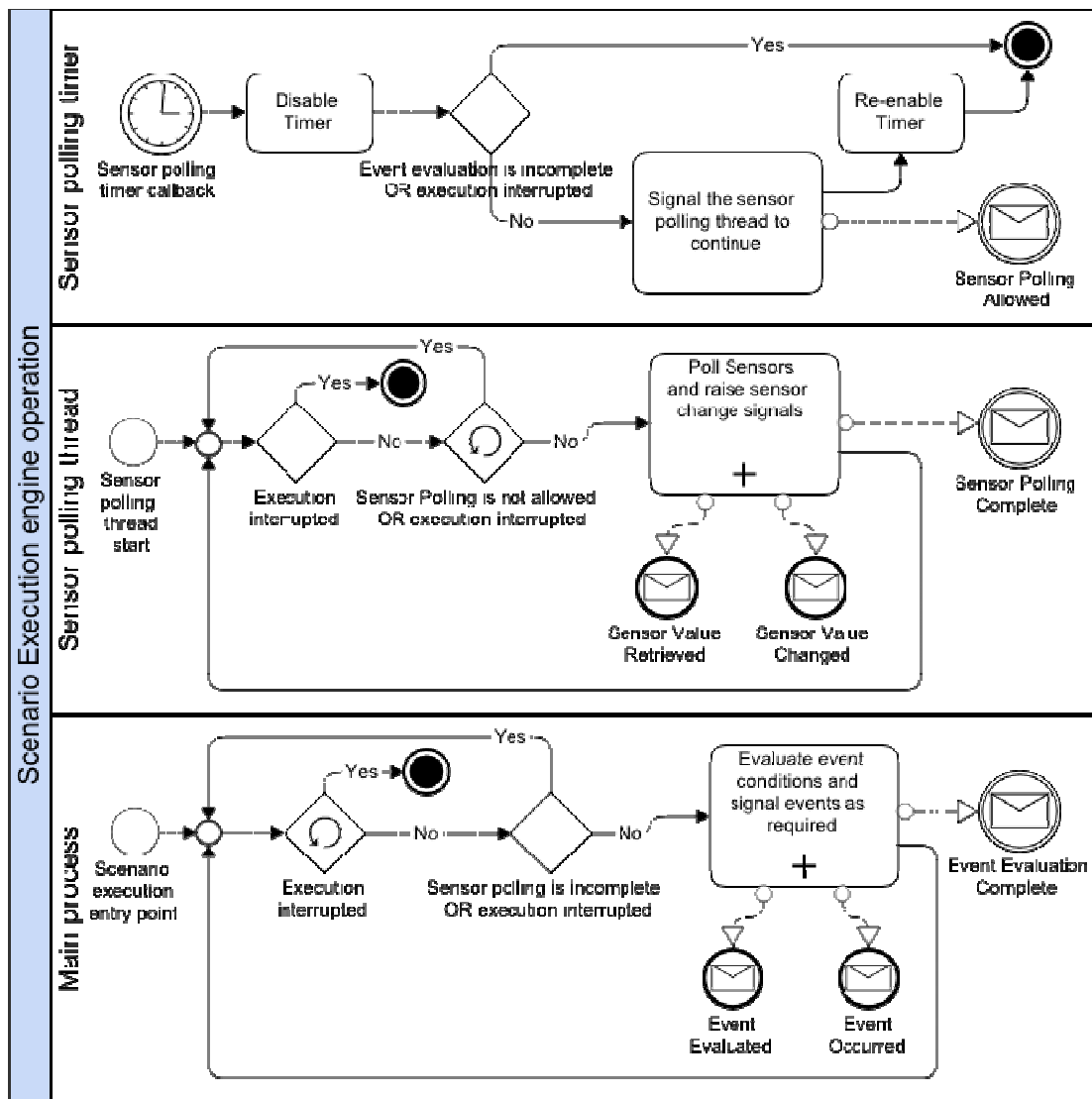
Κάθε σενάριο μπορεί θεωρητικά να εκτελείται επ' άπειρον, αφού τυπικά δεν υπάρχει κάποιος μηχανισμός αυτόματου τερματισμού της εκτέλεσής του, δεδομένου ότι πρέπει να εξυπηρετεί τις ανάγκες ενός smart environment που λειτουργεί συνέχεια. Ωστόσο, είναι απαραίτητο να υπάρχει κάποιος τρόπος να τερματιστεί άμεσα η εκτέλεσή του. Για το λόγο αυτό στην συγκεκριμένη υλοποίηση χρησιμοποιείται η σχετική υποδομή που προσφέρεται από το .NET Framework class library, και συγκεκριμένα οι μηχανισμοί CancellationTokenSource και CancellationToken, ενώ όλοι οι μηχανισμοί συγχρονισμού που χρησιμοποιούνται (ManualResetEventSlim) έχουν επιλεγεί ώστε να υποστηρίζουν τη συγκεκριμένη λειτουργικότητα.

Για την εκτέλεση ενός σεναρίου χρησιμοποιούνται τρία σημεία (σήματα) συγχρονισμού:

- **Sensor polling allowed:** Υποδηλώνει πως η επεξεργασία των προηγούμενων τιμών των αισθητήρων έχει ολοκληρωθεί και μπορεί να γίνει ανάκτηση των τρεχουσών τιμών
- **Sensor polling complete:** Υποδηλώνει πως η ανάκτηση των τιμών των αισθητήρων έχει ολοκληρωθεί και μπορεί να ξεκινήσει η εξέταση των συνθηκών των υποψήφιων συμβάντων
- **Event Evaluation Complete:** Υποδηλώνει πως η εξέταση των συνθηκών των υποψήφιων συμβάντων έχει ολοκληρωθεί και οποιαδήποτε συμβάντα έχουν προκύψει έχουν σημειωθεί, οπότε μπορεί να ξεκινήσει η διαδικασία της ανάκτησης των ενημερωμένων τιμών των αισθητήρων

Με βάση τα σήματα αυτά ξεκινά άμεσα το επόμενο βήμα στη διαδικασία της εκτέλεσης του σεναρίου, αφού κατά τη σήμανση ενός σήματος ξεμπλοκάρει άμεσα η διαδικασία που περιμένει το σήμα αυτό. Έτσι δεν χρειάζεται να χρησιμοποιούνται τεχνικές όπως ατέρμονες βρόχοι με ενδιάμεση καθυστέρηση του νήματος εκτέλεσης και αποκλείονται deadlocks αφού το scenario execution engine λειτουργεί κατά τρόπο αντίστοιχο με ένα state machine με σαφείς και

προκαθορισμένες μεταβάσεις από τη μια κατάσταση στην άλλη. Η χρήση και σήμανση των σημάτων συγχρονισμού για την εκτέλεση ενός σεναρίου παρουσιάζονται με τη μορφή διαγράμματος BPMN πρώτου επιπέδου στην Εικόνα 33.



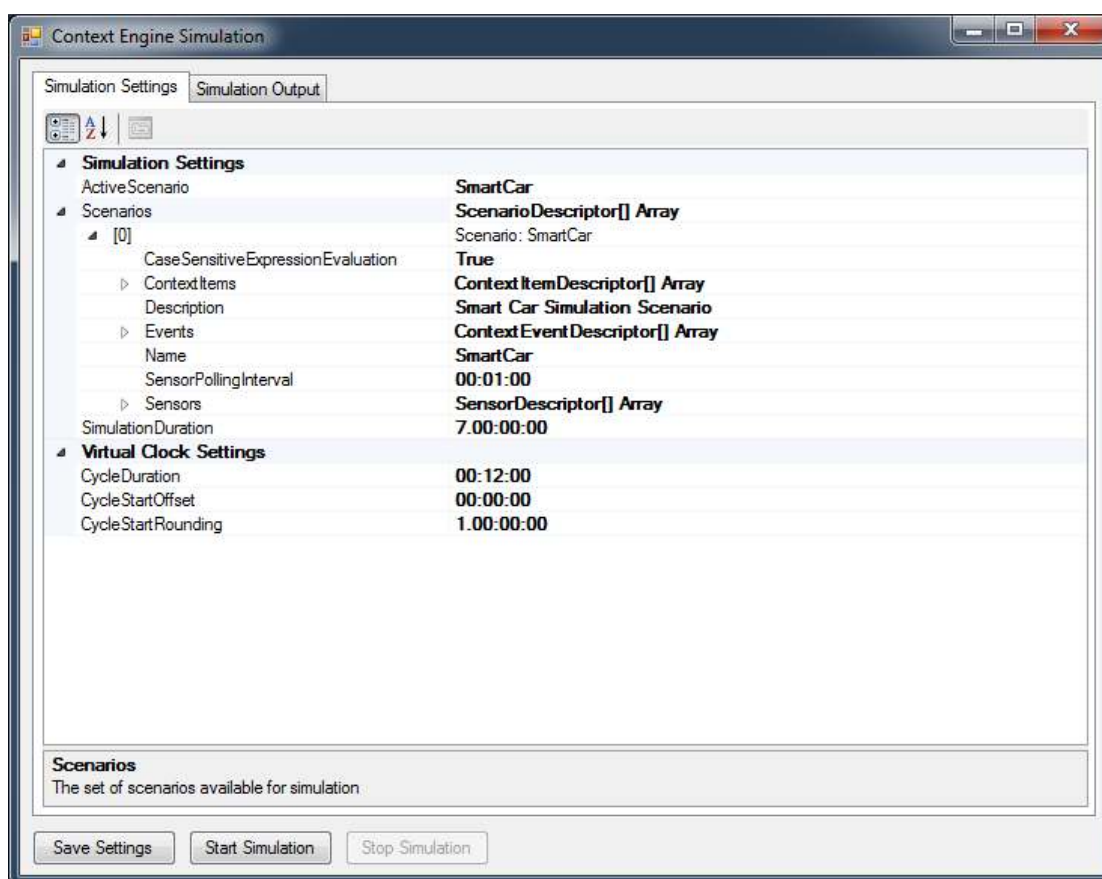
Εικόνα 33: Επισκόπηση της λειτουργίας του Scenario Execution Engine

Υλοποίηση της γραφικής διεπαφής

Η αρχιτεκτονική του συστήματος είναι σχεδιασμένη με τέτοιο τρόπο που το επίπεδο παρουσίασης και αλληλεπίδρασης με το χρήστη να είναι όσο πιο ανεξάρτητο γίνεται από το Context Engine. Αυτό επιτρέπει τη δημιουργία πολλών ειδών γραφικών διεπαφών, που μπορούν να χρησιμοποιηθούν για ρύθμιση των παραμέτρων του συστήματος, παρακολούθηση της λειτουργίας του, κλπ. Στο

πλαίσιο της εργασίας αυτής δόθηκε μεγαλύτερη έμφαση στη σωστή και αποδοτική λειτουργία του Scenario Execution Engine και δημιουργήθηκε μια απλή γραφική διεπαφή. Η διεπαφή αυτή παρέχει τη δυνατότητα ρύθμισης των παραμέτρων του συστήματος και την καταγραφή των συμβάντων που προκύπτουν κατά την προσομοίωση. Παρέχει επίσης τη δυνατότητα της αλληλεπίδρασης με το Context Engine σε βασικό επίπεδο, δηλαδή επιτρέπει στο χρήστη να αποθηκεύσει τις αλλαγές που πραγματοποιεί στις ρυθμίσεις και στα σενάρια, και να ξεκινήσει / σταματήσει την εκτέλεση του Context Engine.

Ο σχεδιασμός των σεναρίων στην έκδοση αυτή γίνεται με τη χρήση μιας δενδροειδούς αναπαράστασης που παρέχεται μέσω ενός Property Grid, το οποίο επιτρέπει τον ορισμό των ιδιοτήτων των σεναρίων, όπως φαίνεται στην Εικόνα 34.



Εικόνα 34: Η γραφική διεπαφή για τη ρύθμιση των παραμέτρων του συστήματος

Όσον αφορά την παρακολούθηση της λειτουργίας του συστήματος, οι διάφορες ενέργειες που εκτελούνται (πχ η ανάκτηση των τιμών αισθητήρων, η αξιολόγηση

των συνθηκών των συμβάντων, η σήμανση συμβάντων, η ενημέρωση για σφάλματα κατά τη λειτουργία), καταγράφονται σε ένα παράθυρο που χρησιμοποιείται για logging, όπως φαίνεται στην Εικόνα 35.

Όπως μπορεί κανείς να παρατηρήσει στην εικόνα αυτή, για κάθε συμβάν που καταγράφεται καταγράφονται διάφορες πληροφορίες που μπορούν να χρησιμοποιηθούν για τον έλεγχο και την αποσφαλμάτωση της λειτουργίας του Context Engine. Συγκεκριμένα εμφανίζονται ο πραγματικός και εικονικός χρόνος κάθε εγγραφής, η σοβαρότητά της (αν αφορά πληροφοριακή εγγραφή, καταγραφή σφάλματος, κλπ.), το νήμα εκτέλεσης το οποίο οδήγησε στην καταγραφή της εγγραφής αυτής και το σχετικό μήνυμα. Ρυθμίζοντας κατάλληλα τον επιθυμητό βαθμό καταγραφής των εγγραφών αυτών, ο χρήστης μπορεί να επιλέξει να ενημερώνεται για περισσότερο ή λιγότερο σημαντικές εγγραφές.

```
[2014-05-11 23:53:16.400][Info][Thread Sensor Polling Thread] ContextEngine: @2014-05-11
11:00:00.585: Sensor 'driver_rfid_sensor' evaluated, Old Value: , New Value: 1234567890abcdef
[2014-05-11 23:53:16.400][Info][Thread Sensor Polling Thread] ContextEngine: @2014-05-11
11:00:00.585: Sensor 'temperature_sensor' evaluated, Old Value: 14,1, New Value: 14
[2014-05-11 23:53:16.400][Info][Thread #9] ContextEngine: @2014-05-11 11:00:00.585: Event
'Driver_Out_Of_Car' evaluated, Success: False
[2014-05-11 23:53:16.400][Info][Thread #9] ContextEngine: @2014-05-11 11:00:00.585: Event
'Temperature_Within_Range' evaluated, Success: False
[2014-05-11 23:53:16.400][Info][Thread #9] ContextEngine: @2014-05-11 11:00:00.585: Event
'Temperature_Not_Within_Range' evaluated, Success: False
[2014-05-11 23:53:16.416][Notice][Thread #9] ContextEngine: @2014-05-11 11:00:02.505: Event
'Driver_In_Car' occurred
[2014-05-11 23:53:16.417][Info][Thread #9] ContextEngine: @2014-05-11 11:00:02.625: Event
'Passenger_Mirror_Needs_Adjustment' evaluated, Success: False
[2014-05-11 23:53:16.417][Info][Thread #9] ContextEngine: @2014-05-11 11:00:02.625: Event
'Rearview_Mirror_Needs_Adjustment' evaluated, Success: False
[2014-05-11 23:53:16.417][Info][Thread #9] ContextEngine: @2014-05-11 11:00:02.625: Event
'Driver_Mirror_Needs_Adjustment' evaluated, Success: False
[2014-05-11 23:53:16.900][Info][Thread Sensor Polling Thread] ContextEngine: @2014-05-11
11:01:00.588: Sensor 'passenger_rfid_sensor' evaluated, Old Value: , New Value: fedcba0987654321
[2014-05-11 23:53:16.903][Notice][Thread #9] ContextEngine: @2014-05-11 11:01:00.948: Event
'Passenger_In_Car' occurred
[2014-05-11 23:53:16.903][Info][Thread #9] ContextEngine: @2014-05-11 11:01:00.948: Event
'Passenger_Out_Of_Car' evaluated, Success: False
[2014-05-11 23:53:18.428][Info][Thread Sensor Polling Thread] ContextEngine: @2014-05-11
11:04:03.959: Sensor 'temperature_sensor' evaluated, Old Value: 14, New Value: 13,9
[2014-05-11 23:53:18.428][Info][Thread #9] ContextEngine: @2014-05-11 11:04:03.959: Event
'Temperature_Within_Range' evaluated, Success: False
[2014-05-11 23:53:18.428][Notice][Thread #9] ContextEngine: @2014-05-11 11:04:03.959: Event
'Temperature_Not_Within_Range' occurred
[2014-05-11 23:53:19.956][Info][Thread Sensor Polling Thread] ContextEngine: @2014-05-11
```

Εικόνα 35: Η γραφική διεπαφή όπου εμφανίζονται τα μηνύματα από τη λειτουργία του συστήματος

[κενή σελίδα]

8. Εφαρμογή και Αξιολόγηση Context Engine

Προσομοίωση εκτέλεσης σεναρίου Έξυπνου Αυτοκινήτου

Για την δοκιμή της υλοποίησης του Context Engine, κατασκευάστηκε ένα αρκετά σύνθετο σενάριο για την προσομοίωση της λειτουργίας ενός έξυπνου αυτοκινήτου (smart car). Στις επόμενες ενότητες περιγράφονται τα χαρακτηριστικά του σεναρίου και τα αποτελέσματα από την εκτέλεσή του.

Περιγραφή σεναρίου

Έστω ότι το Context Engine είναι εγκατεστημένο σε ένα έξυπνο αυτοκίνητο. Το αυτοκίνητο αυτό ανήκει σε ένα ζευγάρι χρηστών, τον Bob και την Alice, που το οδηγούν από κοινού εναλλάξ για τις καθημερινές τους μετακινήσεις. Το αυτοκίνητο έχει εγκατεστημένους διάφορους αισθητήρες και το Context Engine χρησιμοποιείται για να προσαρμόσει αυτόματα το αυτοκίνητο με βάση τις προτιμήσεις του κάθε χρήστη, είτε αυτός βρίσκεται στη θέση του οδηγού, είτε στη θέση του συνοδηγού. Τα χαρακτηριστικά των χρηστών αυτών και το καθημερινό τους πρόγραμμα έχει ως εξής:

Bob: παίρνει το αυτοκίνητο για να πάει στη δουλειά του Δευτέρα έως Παρασκευή 9:00πμ - 18:00μμ. Η διαδρομή από το σπίτι στο γραφείο του απαιτεί 30 λεπτά, δηλαδή οδηγεί το αυτοκίνητο από τις 08:30 ως τις 09:00 και από τις 18:00 ως τις 18:30 περίπου. Η επιθυμητή θερμοκρασία για τον Bob κυμαίνεται μεταξύ 18 και 25 βαθμών Κελσίου, οπότε όταν η θερμοκρασία δεν βρίσκεται μέσα σε αυτά τα όρια χρησιμοποιεί το κλιματιστικό ρυθμισμένο στη θερμοκρασία των 22 βαθμών Κελσίου.

Η επιθυμητή θέση του καθίσματος για τον Bob είναι στο μήκος 70cm από τα πετάλια και η πλάτη του καθίσματος να είναι 100° , ο αριστερός εξωτερικός καθρέφτης είναι στην θέση 100, 50, ο δεξιός εξωτερικός καθρέφτης είναι στην θέση 110, 40, ο εσωτερικός καθρέφτης είναι στην θέση 90, 50.

Alice: Χρησιμοποιεί το αυτοκίνητο τις καθημερινές για τις απογευματινές της δουλειές (ψώνια, pilates, beaute) τις ώρες 19:30μμ – 22:00μμ. Η διαδρομή προς το γυμναστήριο και την αγορά διαρκεί 30 λεπτά. Η επιθυμητή θερμοκρασία για την Alice κυμαίνεται μεταξύ 20 και 29 βαθμών Κελσίου, και ρυθμίζει το κλιματιστικό στους 24 βαθμούς Κελσίου. Η επιθυμητή θέση του καθίσματος είναι στο μήκος 50cm από τα πετάλια και η πλάτη του καθίσματος να είναι 90°, ο αριστερός εξωτερικός καθρέφτης είναι στην θέση 90,60, ο δεξιός εξωτερικός καθρέφτης είναι στην θέση 100,50, ο εσωτερικός καθρέφτης είναι στην θέση 80,40.

Τα σαββατοκύριακα ο Bob και η Alice πηγαίνουν για καφέ/φαγητό/διασκέδαση. Βγαίνουν στις 11:00 το πρωί και οδηγεί η Alice, ενώ στην επιστροφή που γίνεται στις 19:00 οδηγεί ο Bob, και η διάρκεια της διαδρομής για το αγαπημένο τους μέρος είναι 1 ώρα.

Το αυτοκίνητο πρέπει αυτόματα να προσαρμόζεται με βάση το ποιος είναι οδηγός ή / και συνοδηγός σε κάθε χρονική στιγμή. Αυτό σημαίνει πως πρέπει να ρυθμίζει ανάλογα τους καθρέφτες, το κάθισμα και τον κλιματισμό για τον κάθε επιβάτη.

Αισθητήρες και εξοπλισμός σεναρίου smart car

Για την υποστήριξη του παραπάνω σεναρίου στο αυτοκίνητο έχουν εγκατασταθεί οι παρακάτω αισθητήρες:

- Δύο αισθητήρες RFID με ονόματα driver_rfid_sensor και passenger_rfid_sensor που είναι τοποθετημένοι στα καθίσματα του οδηγού και του συνοδηγού αντίστοιχα.
- Ένας αισθητήρας θερμοκρασίας με όνομα temperature_sensor που μετρά την εξωτερική θερμοκρασία
- Τρεις αισθητήρες θέσης για τους καθρέφτες του αυτοκινήτου με ονόματα driver_mirror_sensor, passenger_mirror_sensor, rearview_mirror_sensor για τον αριστερό, δεξί και εσωτερικό καθρέφτη αντίστοιχα

- Δύο αισθητήρες θέσης για τα καθίσματα του αυτοκινήτου με ονόματα `driver_seat_sensor` και `passenger_seat_sensor` για το κάθισμα του οδηγού και του συνοδηγού αντίστοιχα.

Οι αισθητήρες RFID χρησιμοποιούνται για να διαβάσουν RFID Tags τα οποία είναι τοποθετημένα πάνω στα κινητά τηλέφωνα ή τα ρούχα των χρηστών, και επιστρέφουν μια τιμή τύπου `string` που αντιστοιχεί στο αναγνωριστικό ενός Tag, προκειμένου το σύστημα να γνωρίζει ποιος χρήστης βρίσκεται σε κάθε κάθισμα.

Ο αισθητήρας θερμοκρασίας επιστρέφει μια τιμή τύπου αριθμού κινητής υποδιαστολής διπλής ακρίβειας (`double`). Οι αισθητήρες θέσης επιστρέφουν ως τιμή ένα ζευγάρι ακεραίων αριθμών. Στην περίπτωση των αισθητήρων για καθρέφτες αυτοί οι αριθμοί αντιστοιχούν στις γωνίες σε σχέση με τον άξονα της κατεύθυνσης του αυτοκινήτου και του εδάφους, ενώ στους αισθητήρες θέσης για τα καθίσματα, ο πρώτος αριθμός αφορά την απόσταση από τα πετάλια και ο δεύτερος την κλίση του καθίσματος.

Επίσης υποθέτουμε πως το αυτοκίνητο έχει εγκατεστημένο κλιματιστικό, καθώς επίσης και μηχανισμούς ρύθμισης της θέσης των καθρεφτών και των καθισμάτων.

Παράμετροι σεναρίου smart car

Για τη λειτουργία του σεναρίου που παρουσιάστηκε, είναι απαραίτητο να οριστούν κάποια Context items. Κάποια από αυτά θα χρησιμοποιηθούν για την συγκρότηση του προφίλ των χρηστών (Alice & Bob), ενώ άλλα θα χρησιμοποιηθούν ως μεταβλητές κατά την εκτέλεση του σεναρίου. Η λίστα των Context items που χρησιμοποιεί το σενάριο παρουσιάζεται στον επόμενο πίνακα.

Context Item Name	Αρχική τιμή	Επεξήγηση
<code>alice_temp_min</code>	20	Η ελάχιστη θερμοκρασία στην οποία νιώθει άνετα η Alice
<code>alice_temp_max</code>	29	Η μέγιστη θερμοκρασία στην οποία νιώθει άνετα η Alice
<code>alice_temp_ideal</code>	24	Η θερμοκρασία στην οποία η Alice ρυθμίζει τον κλιματισμό
<code>alice_rfid_tag</code>	<code>fedcba0987654321</code>	Το αναγνωριστικό του RFID Tag της Alice

alice_driver_mirror_pos_x		90	Η επιθυμητή τιμή της γωνίας x για τον καθρέφτη οδηγού της Alice
alice_driver_mirror_pos_y		60	Η επιθυμητή τιμή της γωνίας y για τον καθρέφτη οδηγού της Alice
alice_passenger_mirror_pos_x		100	Η επιθυμητή τιμή της γωνίας x για τον καθρέφτη συνοδηγού της Alice
alice_passenger_mirror_pos_y		50	Η επιθυμητή τιμή της γωνίας y για τον καθρέφτη συνοδηγού της Alice
alice_rearview_mirror_pos_x		80	Η επιθυμητή τιμή της γωνίας x για τον εσωτερικό καθρέφτη της Alice
alice_rearview_mirror_pos_y		40	Η επιθυμητή τιμή της γωνίας y για τον εσωτερικό καθρέφτη της Alice
alice_seat_pos_length		50	Η επιθυμητή τιμή της απόστασης του καθίσματος από τα πετάλια για την Alice
alice_seat_pos_angle		90	Η επιθυμητή τιμή της γωνίας του καθίσματος για την Alice
bob_temp_min		18	Η ελάχιστη θερμοκρασία στην οποία νιώθει άνετα ο Bob
bob_temp_max		25	Η μέγιστη θερμοκρασία στην οποία νιώθει άνετα ο Bob
bob_temp_ideal		22	Η θερμοκρασία στην οποία ο Bob ρυθμίζει τον κλιματισμό
bob_rfid_tag	1234567890abcdef		Το αναγνωριστικό του RFID Tag του Bob
bob_driver_mirror_pos_x		100	Η επιθυμητή τιμή της γωνίας x για τον καθρέφτη οδηγού του Bob
bob_driver_mirror_pos_y		50	Η επιθυμητή τιμή της γωνίας y για τον καθρέφτη οδηγού του Bob
bob_passenger_mirror_pos_x		110	Η επιθυμητή τιμή της γωνίας x για τον καθρέφτη συνοδηγού του Bob
bob_passenger_mirror_pos_y		40	Η επιθυμητή τιμή της γωνίας y για τον καθρέφτη συνοδηγού του Bob
bob_rearview_mirror_pos_x		90	Η επιθυμητή τιμή της γωνίας x για τον εσωτερικό καθρέφτη του Bob
bob_rearview_mirror_pos_y		50	Η επιθυμητή τιμή της γωνίας y για τον εσωτερικό καθρέφτη του Bob
bob_seat_pos_length		70	Η επιθυμητή τιμή της απόστασης του καθίσματος από τα πετάλια για τον Bob
bob_seat_pos_angle		100	Η επιθυμητή τιμή της γωνίας του καθίσματος για τον Bob
driver_in_car		0	0 ή 1, ανάλογα με το αν υπάρχει οδηγός στο αυτοκίνητο
passenger_in_car		0	0 ή 1, ανάλογα με το αν υπάρχει συνοδηγός στο αυτοκίνητο
current_driver			Ο τρέχων οδηγός (όνομα)
current_passenger			Ο τρέχων συνοδηγός (όνομα)
air_condition_status		0	0 ή 1 ανάλογα με το αν το κλιματιστικό είναι αναμμένο
driver_mirror_pos_x		0	Η τρέχουσα τιμή της γωνίας x του καθρέφτη του οδηγού

driver_mirror_pos_y	0	Η τρέχουσα τιμή της γωνίας y του καθρέφτη του οδηγού
passenger_mirror_pos_x	0	Η τρέχουσα τιμή της γωνίας x του καθρέφτη του συνοδηγού
passenger_mirror_pos_y	0	Η τρέχουσα τιμή της γωνίας y του καθρέφτη του συνοδηγού
rearview_mirror_pos_x	0	Η τρέχουσα τιμή της γωνίας x του εσωτερικού καθρέφτη
rearview_mirror_pos_y	0	Η τρέχουσα τιμή της γωνίας y του εσωτερικού καθρέφτη
driver_seat_pos_length	60	Η τρέχουσα απόσταση του καθίσματος του οδηγού από τα πετάλια
driver_seat_pos_angle	80	Η τρέχουσα γωνία του καθίσματος του οδηγού
passenger_seat_pos_length	60	Η τρέχουσα απόσταση του καθίσματος του συνοδηγού
passenger_seat_pos_angle	80	Η τρέχουσα γωνία του καθίσματος του συνοδηγού
current_temp_range_min	-100	Η τρέχουσα ελάχιστη επιθυμητή τιμή θερμοκρασίας για όλους τους επιβάτες
current_temp_range_max	-100	Η τρέχουσα μέγιστη επιθυμητή τιμή θερμοκρασίας για όλους τους επιβάτες
current_temp_range_ideal	-100	Η τρέχουσα θερμοκρασία στην οποία θα έπρεπε να ρυθμιστεί ο κλιματισμός για όλους τους επιβάτες

Events σεναρίου smart car

Στον παρακάτω πίνακα παρουσιάζονται τα Events τα οποία πρέπει να εντοπίζει το Context Engine στο πλαίσιο του συγκεκριμένου σεναρίου και περιγράφονται οι συνθήκες που πρέπει να ισχύουν για το καθένα.

Event	Συνθήκη
Driver_In_Car	Ο αισθητήρας driver_rfid_sensor εντοπίζει κάποιο tag ενώ μέχρι τη στιγμή εκείνη δεν καθόταν κάποιος στο κάθισμα του οδηγού, δηλαδή το context item current_driver είναι κενό.
Passenger_In_Car	Ο αισθητήρας passenger_rfid_sensor εντοπίζει κάποιο tag ενώ μέχρι τη στιγμή εκείνη δεν καθόταν κάποιος στο κάθισμα του συνοδηγού, δηλαδή το context item current_passenger είναι κενό.
Driver_Out_Of_Car	Ο αισθητήρας driver_rfid_sensor δεν εντοπίζει κάποιο tag ενώ μέχρι τη στιγμή εκείνη καθόταν κάποιος στο κάθισμα του οδηγού, δηλαδή το context item current_driver δεν είναι κενό.
Passenger_Out_Of_Car	Ο αισθητήρας passenger_rfid_sensor δεν εντοπίζει κάποιο tag ενώ μέχρι τη στιγμή εκείνη καθόταν

	κάποιος στο κάθισμα του συνοδηγού, δηλαδή το context item current_passenger δεν είναι κενό.
Temperature_Within_Range	Η θερμοκρασία που μετρά ο αισθητήρας temperature_sensor είναι εντός των επιθυμητών ορίων για τον τρέχοντα επιβάτη ή επιβάτες.
Temperature_Not_Within_Range	Η θερμοκρασία που μετρά ο αισθητήρας temperature_sensor είναι εκτός των επιθυμητών ορίων για τον τρέχοντα επιβάτη ή επιβάτες.
Driver_Seat_Needs_Adjustment	Η θέση του καθίσματος του οδηγού δεν ταυτίζεται με την επιθυμητή θέση για τον τρέχοντα οδηγό την ώρα που αυτός μπαίνει στο αυτοκίνητο.
Passenger_Seat_Needs_Adjustment	Η θέση του καθίσματος του συνοδηγού δεν ταυτίζεται με την επιθυμητή θέση για τον τρέχοντα συνοδηγό την ώρα που αυτός μπαίνει στο αυτοκίνητο.
Driver_Mirror_Needs_Adjustment	Η θέση του καθρέφτη του οδηγού δεν ταυτίζεται με την επιθυμητή θέση για τον τρέχοντα οδηγό την ώρα που αυτός μπαίνει στο αυτοκίνητο.
Passenger_Mirror_Needs_Adjustment	Η θέση του καθρέφτη του συνοδηγού δεν ταυτίζεται με την επιθυμητή θέση για τον τρέχοντα οδηγό την ώρα που αυτός μπαίνει στο αυτοκίνητο.
Rearview_Mirror_Needs_Adjustment	Η θέση του εσωτερικού καθρέφτη δεν ταυτίζεται με την επιθυμητή θέση για τον τρέχοντα οδηγό την ώρα που αυτός μπαίνει στο αυτοκίνητο.

Προκειμένου να μπορέσει να υλοποιηθεί το παραπάνω σενάριο, τα περισσότερα από αυτά τα Events προκαλούν την εκτέλεση ορισμένων ενεργειών (Actions). Οι ενέργειες αυτές χρησιμοποιούνται για την ανάθεση τιμών σε στοιχεία του Context τα οποία χρησιμεύουν ως μεταβλητές κατά την εκτέλεση του σεναρίου. Με τον τρόπο αυτό μπορούν να απλοποιηθούν πολλές από τις συνθήκες που πρέπει να οριστούν για το κάθε Event και να προσομοιωθεί η ενεργοποίηση Actuators που θα εκτελούσαν κάποια λειτουργία ως αντίδραση σε κάποιο Event.

Παράμετροι και αποτελέσματα προσομοίωσης

Για τη δοκιμή του σεναρίου που παρουσιάστηκε παραπάνω ορίστηκαν κάποιες τιμές στις παραμέτρους της προσομοίωσης προκειμένου η εκτέλεση όλου του σεναρίου να γίνεται σε μικρό χρονικό διάστημα. Το συνολικό διάστημα της προσομοίωσης ορίστηκε σε 7 ημέρες, καθώς αυτή είναι η περίοδος που καλύπτει όλες τις πιθανές εναλλαγές στο state του Context Engine, δηλαδή το πρόγραμμα των χρηστών του έξυπνου αυτοκινήτου έχει κύκλο διάρκειας μιας εβδομάδας. Αυτό

το διάστημα αντιστοιχεί σε εικονικό χρόνο, καθώς το εικονικό ρολόι ρυθμίστηκε με τέτοιο τρόπο ώστε να έχει πραγματική περίοδο 12 λεπτά. Επομένως μια διάρκεια ενός λεπτού εικονικού χρόνου αντιστοιχεί σε μισό δευτερόλεπτο πραγματικού χρόνου. Η ανάκτηση των τιμών των αισθητήρων ρυθμίστηκε να γίνεται κάθε ένα εικονικό λεπτό, δηλαδή κάθε μισό δευτερόλεπτο. Ένας σύγχρονος υπολογιστής μπορεί να υποστηρίξει πολύ γρήγορη προσομοίωση, οπότε οι τιμές των παραμέτρων αυτών θα μπορούσαν να είναι ακόμη μικρότερες προκειμένου να διαρκεί λιγότερο χρόνο η προσομοίωση της εκτέλεσης του σεναρίου. Επίσης χρησιμοποιήθηκε case sensitive expression evaluation για την αναγνώριση των ονομάτων των διαφόρων μεταβλητών και context items.

```
<?xml version="1.0" encoding="utf-8"?>
<ScenarioDescriptor>
  <Name>SmartCar</Name>
  <Description>Smart Car Simulation Scenario</Description>
  <CaseSensitiveExpressionEvaluation>true</CaseSensitiveExpressionEvaluation>
  <ContextItems>
    <ContextItemDescriptor>
      <Name>alice_temp_min</Name>
      <Value>20</Value>
    </ContextItemDescriptor>
    <ContextItemDescriptor>
      <Name>alice_temp_max</Name>
      <Value>29</Value>
    </ContextItemDescriptor>
    <ContextItemDescriptor>
      <Name>alice_temp_ideal</Name>
      <Value>24</Value>
    </ContextItemDescriptor>
    <ContextItemDescriptor>
      <Name>alice_rfid_tag</Name>
      <Value>fedcba0987654321</Value>
    </ContextItemDescriptor>
    <ContextItemDescriptor>
      <Name>alice_driver_mirror_pos_x</Name>
      <Value>90</Value>
    </ContextItemDescriptor>
    <ContextItemDescriptor>
      <Name>alice_driver_mirror_pos_y</Name>
      <Value>60</Value>
    </ContextItemDescriptor>
    <ContextItemDescriptor>
      <Name>alice_passenger_mirror_pos_x</Name>
      <Value>100</Value>
    </ContextItemDescriptor>
    <ContextItemDescriptor>
      <Name>alice_passenger_mirror_pos_y</Name>
      <Value>50</Value>
    </ContextItemDescriptor>
    <ContextItemDescriptor>
      <Name>alice_rearview_mirror_pos_x</Name>
      <Value>80</Value>
    </ContextItemDescriptor>
    <ContextItemDescriptor>
      <Name>alice_rearview_mirror_pos_y</Name>
      <Value>40</Value>
    </ContextItemDescriptor>
    <ContextItemDescriptor>
      <Name>alice_seat_pos_length</Name>

```

```

    <Value>50</Value>
  </ContextItemDescriptor>
</ContextItemDescriptor>
<ContextItemDescriptor>
  <Name>alice_seat_pos_angle</Name>
  <Value>90</Value>
</ContextItemDescriptor>
<ContextItemDescriptor>
  <Name>bob_temp_min</Name>
  <Value>18</Value>
</ContextItemDescriptor>
<ContextItemDescriptor>
  <Name>bob_temp_max</Name>
  <Value>25</Value>
</ContextItemDescriptor>
<ContextItemDescriptor>
  <Name>bob_temp_ideal</Name>
  <Value>22</Value>
</ContextItemDescriptor>
<ContextItemDescriptor>
  <Name>bob_rfid_tag</Name>
  <Value>1234567890abcdef</Value>
</ContextItemDescriptor>
<ContextItemDescriptor>
  <Name>bob_driver_mirror_pos_x</Name>
  <Value>100</Value>
</ContextItemDescriptor>
<ContextItemDescriptor>
  <Name>bob_driver_mirror_pos_y</Name>
  <Value>50</Value>
</ContextItemDescriptor>
<ContextItemDescriptor>
  <Name>bob_passenger_mirror_pos_x</Name>
  <Value>110</Value>
</ContextItemDescriptor>
<ContextItemDescriptor>
  <Name>bob_passenger_mirror_pos_y</Name>
  <Value>40</Value>
</ContextItemDescriptor>
<ContextItemDescriptor>
  <Name>bob_rearview_mirror_pos_x</Name>
  <Value>90</Value>
</ContextItemDescriptor>
<ContextItemDescriptor>
  <Name>bob_rearview_mirror_pos_y</Name>
  <Value>50</Value>
</ContextItemDescriptor>
<ContextItemDescriptor>
  <Name>bob_seat_pos_length</Name>
  <Value>70</Value>
</ContextItemDescriptor>
<ContextItemDescriptor>
  <Name>bob_seat_pos_angle</Name>
  <Value>100</Value>
</ContextItemDescriptor>
<ContextItemDescriptor>
  <Name>driver_in_car</Name>
  <Value>0</Value>
</ContextItemDescriptor>
<ContextItemDescriptor>
  <Name>passenger_in_car</Name>
  <Value>0</Value>
</ContextItemDescriptor>
<ContextItemDescriptor>
  <Name>current_driver</Name>
  <Value />
</ContextItemDescriptor>
<ContextItemDescriptor>
  <Name>current_passenger</Name>
  <Value />
</ContextItemDescriptor>
<ContextItemDescriptor>
  <Name>air_condition_status</Name>
  <Value>0</Value>
</ContextItemDescriptor>

```

```

<ContextItemDescriptor>
  <Name>driver_mirror_pos_x</Name>
  <Value>100</Value>
</ContextItemDescriptor>
<ContextItemDescriptor>
  <Name>driver_mirror_pos_y</Name>
  <Value>50</Value>
</ContextItemDescriptor>
<ContextItemDescriptor>
  <Name>passenger_mirror_pos_x</Name>
  <Value>110</Value>
</ContextItemDescriptor>
<ContextItemDescriptor>
  <Name>passenger_mirror_pos_y</Name>
  <Value>40</Value>
</ContextItemDescriptor>
<ContextItemDescriptor>
  <Name>rearview_mirror_pos_x</Name>
  <Value>90</Value>
</ContextItemDescriptor>
<ContextItemDescriptor>
  <Name>rearview_mirror_pos_y</Name>
  <Value>50</Value>
</ContextItemDescriptor>
<ContextItemDescriptor>
  <Name>driver_seat_pos_length</Name>
  <Value>60</Value>
</ContextItemDescriptor>
<ContextItemDescriptor>
  <Name>driver_seat_pos_angle</Name>
  <Value>80</Value>
</ContextItemDescriptor>
<ContextItemDescriptor>
  <Name>passenger_seat_pos_length</Name>
  <Value>60</Value>
</ContextItemDescriptor>
<ContextItemDescriptor>
  <Name>passenger_seat_pos_angle</Name>
  <Value>80</Value>
</ContextItemDescriptor>
<ContextItemDescriptor>
  <Name>current_temp_range_min</Name>
  <Value>18</Value>
</ContextItemDescriptor>
<ContextItemDescriptor>
  <Name>current_temp_range_max</Name>
  <Value>25</Value>
</ContextItemDescriptor>
<ContextItemDescriptor>
  <Name>current_temp_range_ideal</Name>
  <Value>-100</Value>
</ContextItemDescriptor>
</ContextItems>
<Events>
  <ContextEventDescriptor>
    <Name>Driver_In_Car</Name>
    <Description>Occurs when driver enters the car</Description>
    <Condition>context["current_driver"].Value == "" & &
driver_rfid_sensor.Value != ""</Condition>
    <Order>0</Order>
    <Actions>
      <ContextEventAction>
        <Expression>"1"</Expression>
        <ContextItemName>driver_in_car</ContextItemName>
      </ContextEventAction>
      <ContextEventAction>
        <Expression>(driver_rfid_sensor.Value == context["alice_rfid_tag"].Value)
? "alice" : "bob"</Expression>
        <ContextItemName>current_driver</ContextItemName>
      </ContextEventAction>
      <ContextEventAction>
        <Expression>context["passenger_in_car"].Value == "1" ?
Math.Max(int.Parse(context["alice_temp_min"].Value),
int.Parse(context["bob_temp_min"].Value)).ToString() : context[context["current_driver"].Value

```

```

+ "_temp_min"].Value</Expression>
    <ContextItemName>current_temp_range_min</ContextItemName>
  </ContextEventAction>
</ContextEventAction>
  <Expression>context["passenger_in_car"].Value == "1" ?
Math.Min(int.Parse(context["alice_temp_max"].Value),
int.Parse(context["bob_temp_max"].Value)).ToString() : context[context["current_driver"].Value
+ "_temp_max"].Value</Expression>
    <ContextItemName>current_temp_range_max</ContextItemName>
  </ContextEventAction>
</Actions>
<Sensors>
  <SensorNameDescriptor>
    <Name>driver_rfid_sensor</Name>
  </SensorNameDescriptor>
</Sensors>
</ContextEventDescriptor>
<ContextEventDescriptor>
  <Name>Driver_Out_Of_Car</Name>
  <Description>Occurs when driver exits the car</Description>
  <Condition>context["current_driver"].Value != "" &amp;&amp;
driver_rfid_sensor.Value == ""</Condition>
  <Order>0</Order>
  <Actions>
    <ContextEventAction>
      <Expression>"0"</Expression>
      <ContextItemName>driver_in_car</ContextItemName>
    </ContextEventAction>
    <ContextEventAction>
      <Expression>string.Empty</Expression>
      <ContextItemName>current_driver</ContextItemName>
    </ContextEventAction>
    <ContextEventAction>
      <Expression>context["passenger_in_car"].Value == "1" ?
context[context["current_passenger"].Value + "_temp_min"].Value : "-100"</Expression>
      <ContextItemName>current_temp_range_min</ContextItemName>
    </ContextEventAction>
    <ContextEventAction>
      <Expression>context["passenger_in_car"].Value == "1" ?
context[context["current_passenger"].Value + "_temp_max"].Value : "-100"</Expression>
      <ContextItemName>current_temp_range_max</ContextItemName>
    </ContextEventAction>
  </Actions>
<Sensors>
  <SensorNameDescriptor>
    <Name>driver_rfid_sensor</Name>
  </SensorNameDescriptor>
</Sensors>
</ContextEventDescriptor>
<ContextEventDescriptor>
  <Name>Passenger_In_Car</Name>
  <Description>Occurs when passenger enters the car</Description>
  <Condition>context["current_passenger"].Value == "" &amp;&amp;
passenger_rfid_sensor.Value != ""</Condition>
  <Order>0</Order>
  <Actions>
    <ContextEventAction>
      <Expression>"1"</Expression>
      <ContextItemName>passenger_in_car</ContextItemName>
    </ContextEventAction>
    <ContextEventAction>
      <Expression>(passenger_rfid_sensor.Value ==
context["alice_rfid_tag"].Value) ? "alice" : "bob"</Expression>
      <ContextItemName>current_passenger</ContextItemName>
    </ContextEventAction>
    <ContextEventAction>
      <Expression>context["driver_in_car"].Value == "1" ?
Math.Max(int.Parse(context["alice_temp_min"].Value), int.Parse(context["bob_temp_min"].Value)
).ToString() : context[context["current_passenger"].Value + "_temp_min"].Value</Expression>
      <ContextItemName>current_temp_range_min</ContextItemName>
    </ContextEventAction>
    <ContextEventAction>
      <Expression>context["driver_in_car"].Value == "1" ?
Math.Min(int.Parse(context["alice_temp_max"].Value), int.Parse(context["bob_temp_max"].Value)

```

```

).ToString() : context[context["current_passenger"].Value + "_temp_max"].Value</Expression>
    <ContextItemName>current_temp_range_max</ContextItemName>
</ContextEventAction>
</Actions>
<Sensors>
    <SensorNameDescriptor>
        <Name>passenger_rfid_sensor</Name>
    </SensorNameDescriptor>
</Sensors>
</ContextEventDescriptor>
<ContextEventDescriptor>
    <Name>Passenger_Out_Of_Car</Name>
    <Description>Occurs when passenger exits the car</Description>
    <Condition>context["current_passenger"].Value != "" & &
passenger_rfid_sensor.Value == ""</Condition>
    <Order>0</Order>
    <Actions>
        <ContextEventAction>
            <Expression>"0"</Expression>
            <ContextItemName>passenger_in_car</ContextItemName>
        </ContextEventAction>
        <ContextEventAction>
            <Expression>string.Empty</Expression>
            <ContextItemName>current_passenger</ContextItemName>
        </ContextEventAction>
        <ContextEventAction>
            <Expression>context["driver_in_car"].Value == "1" ?
context[context["current_driver"].Value + "_temp_min"].Value : "-100"</Expression>
            <ContextItemName>current_temp_range_min</ContextItemName>
        </ContextEventAction>
        <ContextEventAction>
            <Expression>context["driver_in_car"].Value == "1" ?
context[context["current_driver"].Value + "_temp_max"].Value : "-100"</Expression>
            <ContextItemName>current_temp_range_max</ContextItemName>
        </ContextEventAction>
    </Actions>
<Sensors>
    <SensorNameDescriptor>
        <Name>passenger_rfid_sensor</Name>
    </SensorNameDescriptor>
</Sensors>
</ContextEventDescriptor>
<ContextEventDescriptor>
    <Name>Temperature_Within_Range</Name>
    <Description>Occurs when the temperature measured by the temperature sensor is
within the current temperature range</Description>
    <Condition>(context["driver_in_car"].Value == "1" ||
context["passenger_in_car"].Value == "1") & &
temperature_sensor.Value.CompareTo(double.Parse(context["current_temp_range_min"].Value))
&gt;= 0 & &
temperature_sensor.Value.CompareTo(double.Parse(context["current_temp_range_max"].Value))
&lt;= 0</Condition>
    <Order>0</Order>
    <Actions>
        <ContextEventAction>
            <Expression>context["air_condition_status"].Value == "1" ? "0" :
"1"</Expression>
            <ContextItemName>air_condition_status</ContextItemName>
        </ContextEventAction>
    </Actions>
<Sensors>
    <SensorNameDescriptor>
        <Name>temperature_sensor</Name>
    </SensorNameDescriptor>
</Sensors>
</ContextEventDescriptor>
<ContextEventDescriptor>
    <Name>Temperature_Not_Within_Range</Name>
    <Description>Occurs when the temperature measured by the temperature sensor is not
within the current temperature range</Description>
    <Condition>(context["driver_in_car"].Value == "1" ||
context["passenger_in_car"].Value == "1") & &
(temperature_sensor.Value.CompareTo(double.Parse(context["current_temp_range_min"].Value))
&lt;= 0 ||

```

```

temperature_sensor.Value.CompareTo(double.Parse(context["current_temp_range_max"].Value)) &gt;
0)</Condition>
    <Order>0</Order>
    <Actions>
        <ContextEventAction>
            <Expression>context["air_condition_status"].Value == "0" ? "1" :
"0"</Expression>
            <ContextItemName>air_condition_status</ContextItemName>
        </ContextEventAction>
    </Actions>
</Sensors>
    <SensorNameDescriptor>
        <Name>temperature_sensor</Name>
    </SensorNameDescriptor>
</Sensors>
</ContextEventDescriptor>
<ContextEventDescriptor>
    <Name>Driver_Seat_Needs_Adjustment</Name>
    <Description>Occurs when the driver's seat needs to be adjusted</Description>
    <Condition>context["driver_in_car"].Value == "1" &amp;&amp;
driver_seat_sensor.Value.CompareTo( new SeatPosition(context[ context["current_driver"].Value
+ "_seat_pos_length"].Value, context[ context["current_driver"].Value +
"_seat_pos_angle"].Value)) !=0</Condition>
    <Order>1</Order>
    <Actions>
        <ContextEventAction>
            <Expression>context[ context["current_driver"].Value +
"_seat_pos_length"].Value</Expression>
            <ContextItemName>driver_seat_pos_length</ContextItemName>
        </ContextEventAction>
        <ContextEventAction>
            <Expression>context[ context["current_driver"].Value +
"_seat_pos_angle"].Value</Expression>
            <ContextItemName>driver_seat_pos_angle</ContextItemName>
        </ContextEventAction>
    </Actions>
</Sensors>
    <SensorNameDescriptor>
        <Name>driver_seat_sensor</Name>
    </SensorNameDescriptor>
</Sensors>
</ContextEventDescriptor>
<ContextEventDescriptor>
    <Name>Passenger_Seat_Needs_Adjustment</Name>
    <Description>Occurs when the passenger's seat needs to be adjusted</Description>
    <Condition>context["passenger_in_car"].Value == "1" &amp;&amp;
passenger_seat_sensor.Value.CompareTo( new SeatPosition(context[
context["current_passenger"].Value + "_seat_pos_length"].Value, context[
context["current_passenger"].Value + "_seat_pos_angle"].Value)) !=0</Condition>
    <Order>0</Order>
    <Actions>
        <ContextEventAction>
            <Expression>context[ context["current_passenger"].Value +
"_seat_pos_length"].Value</Expression>
            <ContextItemName>passenger_seat_pos_length</ContextItemName>
        </ContextEventAction>
        <ContextEventAction>
            <Expression>context[ context["current_passenger"].Value +
"_seat_pos_angle"].Value</Expression>
            <ContextItemName>passenger_seat_pos_angle</ContextItemName>
        </ContextEventAction>
    </Actions>
</Sensors>
    <SensorNameDescriptor>
        <Name>passenger_seat_sensor</Name>
    </SensorNameDescriptor>
</Sensors>
</ContextEventDescriptor>
<ContextEventDescriptor>
    <Name>Driver_Mirror_Needs_Adjustment</Name>
    <Description>Occurs when the driver's mirror needs to be adjusted. Mirrors are
adjusted if necessary when a driver enters the car.</Description>
    <Condition>context["driver_in_car"].Value == "1" &amp;&amp;
driver_mirror_sensor.Value.CompareTo(new

```



```

MirrorPosition(context[context["current_driver"].Value + "_driver_mirror_pos_x"].Value,
context[context["current_driver"].Value + "_driver_mirror_pos_y"].Value)) !=0</Condition>
  <Order>1</Order>
  <Actions>
    <ContextEventAction>
      <Expression>context[ context["current_driver"].Value +
"_driver_mirror_pos_x"].Value</Expression>
      <ContextItemName>driver_mirror_pos_x</ContextItemName>
    </ContextEventAction>
    <ContextEventAction>
      <Expression>context[ context["current_driver"].Value +
"_driver_mirror_pos_y"].Value</Expression>
      <ContextItemName>driver_mirror_pos_y</ContextItemName>
    </ContextEventAction>
  </Actions>
  <Sensors>
    <SensorNameDescriptor>
      <Name>driver_rfid_sensor</Name>
    </SensorNameDescriptor>
    <SensorNameDescriptor>
      <Name>driver_mirror_sensor</Name>
    </SensorNameDescriptor>
  </Sensors>
</ContextEventDescriptor>
<ContextEventDescriptor>
  <Name>Passenger_Mirror_Needs_Adjustment</Name>
  <Description>Occurs when the passenger's mirror needs to be adjusted. Mirrors are
adjusted if necessary when a driver enters the car.</Description>
  <Condition>context["driver_in_car"].Value == "1" &amp;&amp;
passenger_mirror_sensor.Value.CompareTo( new MirrorPosition(context[
context["current_driver"].Value + "_passenger_mirror_pos_x"].Value, context[
context["current_driver"].Value + "_passenger_mirror_pos_y"].Value)) !=0</Condition>
  <Order>1</Order>
  <Actions>
    <ContextEventAction>
      <Expression>context[ context["current_driver"].Value +
"_passenger_mirror_pos_x"].Value</Expression>
      <ContextItemName>passenger_mirror_pos_x</ContextItemName>
    </ContextEventAction>
    <ContextEventAction>
      <Expression>context[ context["current_driver"].Value +
"_passenger_mirror_pos_y"].Value</Expression>
      <ContextItemName>passenger_mirror_pos_y</ContextItemName>
    </ContextEventAction>
  </Actions>
  <Sensors>
    <SensorNameDescriptor>
      <Name>driver_rfid_sensor</Name>
    </SensorNameDescriptor>
    <SensorNameDescriptor>
      <Name>passenger_mirror_sensor</Name>
    </SensorNameDescriptor>
  </Sensors>
</ContextEventDescriptor>
<ContextEventDescriptor>
  <Name>Rearview_Mirror_Needs_Adjustment</Name>
  <Description>Occurs when the rearview mirror needs to be adjusted. Mirrors are
adjusted if necessary when a driver enters the car.</Description>
  <Condition>context["driver_in_car"].Value == "1" &amp;&amp;
rearview_mirror_sensor.Value.CompareTo( new MirrorPosition(context[
context["current_driver"].Value + "_rearview_mirror_pos_x"].Value, context[
context["current_driver"].Value + "_rearview_mirror_pos_y"].Value)) !=0</Condition>
  <Order>1</Order>
  <Actions>
    <ContextEventAction>
      <Expression>context[ context["current_driver"].Value +
"_rearview_mirror_pos_x"].Value</Expression>
      <ContextItemName>rearview_mirror_pos_x</ContextItemName>
    </ContextEventAction>
    <ContextEventAction>
      <Expression>context[ context["current_driver"].Value +
"_rearview_mirror_pos_y"].Value</Expression>
      <ContextItemName>rearview_mirror_pos_y</ContextItemName>
    </ContextEventAction>
  </Actions>
  <Sensors>
    <SensorNameDescriptor>
      <Name>driver_rfid_sensor</Name>
    </SensorNameDescriptor>
    <SensorNameDescriptor>
      <Name>passenger_mirror_sensor</Name>
    </SensorNameDescriptor>
    <SensorNameDescriptor>
      <Name>rearview_mirror_sensor</Name>
    </SensorNameDescriptor>
  </Sensors>
</ContextEventDescriptor>
</ContextEventDescriptor>

```

```

</Actions>
<Sensors>
  <SensorNameDescriptor>
    <Name>driver_rfid_sensor</Name>
  </SensorNameDescriptor>
  <SensorNameDescriptor>
    <Name>rearview_mirror_sensor</Name>
  </SensorNameDescriptor>
</Sensors>
</ContextEventDescriptor>
</Events>
<Sensors>
  <SensorDescriptor>
    <Name>temperature_sensor</Name>
    <Description>Temperature sensor</Description>
    <Location>Smart Car</Location>
    <Type>TemperatureSensor</Type>
  </SensorDescriptor>
  <SensorDescriptor>
    <Name>driver_rfid_sensor</Name>
    <Description>RFID sensor</Description>
    <Location>Driver's seat</Location>
    <Type>RFIDSensor</Type>
  </SensorDescriptor>
  <SensorDescriptor>
    <Name>passenger_rfid_sensor</Name>
    <Description>RFID Sensor</Description>
    <Location>Passenger's seat</Location>
    <Type>RFIDSensor</Type>
  </SensorDescriptor>
  <SensorDescriptor>
    <Name>driver_mirror_sensor</Name>
    <Description>Driver's side mirror position sensor</Description>
    <Location>Driver side</Location>
    <Type>MirrorPositionSensor</Type>
  </SensorDescriptor>
  <SensorDescriptor>
    <Name>passenger_mirror_sensor</Name>
    <Description>Passenger's side mirror position sensor</Description>
    <Location>Passenger side</Location>
    <Type>MirrorPositionSensor</Type>
  </SensorDescriptor>
  <SensorDescriptor>
    <Name>rearview_mirror_sensor</Name>
    <Description>Rear view mirror sensor</Description>
    <Location>Smart Car cabin</Location>
    <Type>MirrorPositionSensor</Type>
  </SensorDescriptor>
  <SensorDescriptor>
    <Name>driver_seat_sensor</Name>
    <Description>Driver's seat position sensor</Description>
    <Location>Driver's seat</Location>
    <Type>SeatPositionSensor</Type>
  </SensorDescriptor>
  <SensorDescriptor>
    <Name>passenger_seat_sensor</Name>
    <Description>Passenger's seat position sensor</Description>
    <Location>Passenger's seat</Location>
    <Type>SeatPositionSensor</Type>
  </SensorDescriptor>
</Sensors>
<SensorPollingInterval>00:01:00</SensorPollingInterval>
</ScenarioDescriptor>

```

Code listing 6: Το σενάριο έξυπνου αυτοκινήτου σε μορφή XML

Κατά την εκτέλεση του σεναρίου τα συμβάντα που σημαίνει το Context Engine καταγράφονται στο παράθυρο καταγραφής της εξόδου. Όπως αναφέρθηκε στο προηγούμενο κεφάλαιο, στη συγκεκριμένη υλοποίηση δεν γίνεται κάποιο άλλο visualization των αποτελεσμάτων της εξόδου, αλλά κάτι τέτοιο αποτελεί εύφορο πεδίο για βελτιώσεις στην τρέχουσα έκδοση.

[κενή σελίδα]

9. Future work

Παρόλο που η υλοποίηση του Context Engine με βάση τους στόχους που τέθηκαν κατά τη φάση του σχεδιασμού ολοκληρώθηκε επιτυχώς, υπάρχουν πολλά περιθώρια για βελτίωση σε όλους τους επιμέρους τομείς.

Το σημαντικότερο σημείο το οποίο μπορεί να επιδεχθεί σημαντική βελτίωση είναι ο ίδιος ο μηχανισμός που χρησιμεύει για την αποθήκευση και ανάκτηση των πληροφοριών του πλαισίου. Αυτή τη στιγμή χρησιμοποιείται μια δομή δεδομένων που είναι ασφαλής για ταυτόχρονη πρόσβαση από πολλά νήματα εκτέλεσης τα οποία εκτελούνται παράλληλα χωρίς να υπάρχει κίνδυνος για deadlocks και δημιουργία ασυνεπειών (inconsistencies) στην αποθηκευμένη πληροφορία. Ωστόσο η τρέχουσα υλοποίηση έχει δύο μειονεκτήματα. Αφενός η πληροφορία αυτή αποθηκεύεται μόνο στη μνήμη για όσο λειτουργεί το Context Engine, και καταστρέφεται όταν η λειτουργία του τελευταίου σταματά. Αφετέρου, το μέγιστο μέγεθος αυτής της συλλογής πληροφοριών πλαισίου περιορίζεται από το μέγεθος της φυσικής και εικονικής μνήμης που διαθέτει το σύστημα στο οποίο λειτουργεί το Context Engine.

Τα μειονεκτήματα του μηχανισμού αυτού μπορούν να αρθούν με τη χρήση κάποιου μηχανισμού που αποθηκεύει την πληροφορία του πλαισίου σε κάποια αποθήκη δεδομένων, όπως π.χ. μια βάση δεδομένων. Για συστήματα μικρής κλίμακας θα μπορούσε να χρησιμοποιηθεί μια βάση δεδομένων περιορισμένης κλίμακας, όπως η SQLite. Για πιο μεγάλα συστήματα όπου η ακρίβεια της πληροφορίας δεν είναι απαραίτητο να είναι 100% ακριβής σε κάθε επερώτηση αλλά είναι σημαντικό να είναι διαθέσιμη η πληροφορία σε μεγάλο αριθμό κόμβων, π.χ. σε κάποιο έξυπνο δίκτυο ubiquitous computing, όπου υπάρχουν απαιτήσεις υψηλής διαθεσιμότητας και οριζόντιας επεκτασιμότητας, οι τεχνολογίες NoSQL μπορούν να πετύχουν πολύ καλά αποτελέσματα. Δεδομένου ότι οι πληροφορίες πλαισίου είναι στην πλειοψηφία τους πληροφορίες τύπου Key-Value pair (αντιστοίχιση ενός κλειδιού με μια τιμή) ή γράφοι, η εξειδικευμένη υποστήριξη για τέτοιους τύπους δεδομένων

που παρέχουν οι NoSQL databases τις καθιστά ιδανική επιλογή για τη συγκεκριμένη χρήση.

Το επόμενο σημείο το οποίο χρήζει βελτίωσης στην τρέχουσα υλοποίηση δεν είναι άλλο από αυτό της γραφικής διεπαφής. Η τρέχουσα έκδοση παρέχει πολύ περιορισμένες δυνατότητες για τον σχεδιασμό των σεναρίων αλλά και για την παρακολούθηση της εκτέλεσής τους. Κάθε ένας από αυτούς τους άξονες μπορεί να δεχθεί πολλές βελτιώσεις, οδηγώντας έτσι σε μια σημαντικά καλύτερη εμπειρία του τελικού χρήστη του συστήματος.

Όσον αφορά τον άξονα του σχεδιασμού σεναρίων, είναι ιδιαίτερα θετικό το γεγονός ότι στην τρέχουσα έκδοση υποστηρίζεται αποθήκευση και φόρτωση σεναρίων από μια μορφή XML. Η δυνατότητα αυτή παρέχει ανεξαρτησία από λειτουργικά συστήματα και πλατφόρμες εκτέλεσης, αφού τα έγγραφα XML είναι ευρέως διαδεδομένος τρόπος για τον διαμοιρασμό τέτοιων δεδομένων. Ωστόσο, η υποστήριξη για τον σχεδιασμό των σεναρίων που παρέχεται αυτή τη στιγμή είναι υποτυπώδης. Για το σκοπό αυτό θα μπορούσε να υλοποιηθεί κάποιος γραφικός editor που να επιτρέπει τον καθορισμό των παραμέτρων του συστήματος, την απεικόνιση του έξυπνου περιβάλλοντος και της τοποθεσίας των αισθητήρων κλπ. Ο editor αυτός ιδανικά θα μπορούσε να επιτρέπει με γραφικό τρόπο τον καθορισμό των συνθηκών που θα οδηγούν στη σήμανση ενός συμβάντος, και να δημιουργεί την αντίστοιχη αναπαράσταση του σεναρίου σε μορφή XML.

Στον άξονα της παρακολούθησης της λειτουργίας του Context Engine, μπορούν επίσης να προκύψουν σημαντικές βελτιώσεις. Παρόλο που αυτή τη στιγμή καταγράφονται όλα τα συμβάντα που προκύπτουν, θα ήταν ενδιαφέρον να παρουσιάζεται σε γραφικό περιβάλλον μια απεικόνιση του έξυπνου συστήματος το οποίο περιλαμβάνει το Context Engine, και να παρουσιάζονται δυναμικά πάνω σε αυτή την απεικόνιση οι αλλαγές που προκύπτουν σε τιμές αισθητήρων ή άλλες παραμέτρους του συστήματος. Στο σημείο αυτό αξίζει να αναφερθεί πως η λειτουργικότητα της παρακολούθησης της εκτέλεσης των σεναρίων θα μπορούσε να παρασχεθεί μέσω ενός μοντέλου client-server εφαρμογών. Δηλαδή, εφαρμογές

τρίτων ή άλλα συστήματα που επιθυμούν να ενημερώνονται για τις αλλαγές στην κατάσταση του Context Engine θα μπορούσαν να συνδέονται στο τελευταίο μέσω ενός καναλιού δικτυακής επικοινωνίας από οπουδήποτε και να αντλήσουν τις πληροφορίες που θέλουν ή ακόμη και να ελέγξουν τη λειτουργία του.

Τέλος, σε επίπεδο υλοποίησης, υπάρχουν μερικές λεπτομέρειες που δεν έχουν ολοκληρωθεί στην τρέχουσα έκδοση, αλλά μπορούν να υλοποιηθούν με σχετικά μικρό κόστος. Αυτές είναι οι παρακάτω:

- Όσον αφορά την παράλληλη εκτέλεση πολλαπλών σεναρίων, παρόλο που το Context Engine το υποστηρίζει, η γραφική διεπαφή που έχει υλοποιηθεί επιτρέπει την ενεργοποίηση μόνο ενός σεναρίου ανά πάσα χρονική στιγμή. Με μικρές αλλαγές η τρέχουσα γραφική διεπαφή μπορεί να βελτιωθεί ώστε να υποστηρίζει την εκτέλεση πολλαπλών σεναρίων ταυτόχρονα.
- Η προγραμματιστική διεπαφή των αισθητήρων περιλαμβάνει μια ιδιότητα (Property) που επιτρέπει σε ένα αισθητήρα να ορίσει το εύρος των επιτρεπτών τιμών του. Παρόλα αυτά, αυτή τη στιγμή δεν εξετάζεται αν η τιμή που επιστρέφει ένας αισθητήρας όταν γίνεται ανάκτηση της τιμής του εμπίπτει σε αυτό το εύρος τιμών. Κάτι τέτοιο θα μπορούσε να πραγματοποιηθεί με μια μικρή αλλαγή στον κώδικα του συστήματος, προκειμένου οι μη έγκυρες τιμές που μπορεί να επιστρέψει ένας αισθητήρας λόγω σφάλματος στη μέτρηση ή τη μετάδοση δεδομένων να αγνοηθούν από το Context Engine.

Όλα τα παραπάνω μπορούν να αποτελέσουν στόχους μελλοντικής προσπάθειας για βελτίωση του συστήματος.

[κενή σελίδα]

Βιβλιογραφία

- [1] «Ubiquitous_computing,» Wikipedia, [Ηλεκτρονικό]. Available: http://en.wikipedia.org/wiki/Ubiquitous_computing.
- [2] D. S. Milojevic, A. Messer, P. Bernadat, I. Greenberg, O. Spinczyk, D. Beuche και W. S. Preikschat, «Psi-Pervasive Services Infrastructure,» *TES '01 Proceedings of the Second International Workshop on Technologies for E-Services*, pp. 187 - 200, 2001.
- [3] M. Mühlhäuser και I. Gurevych, «“Handbook of Research: Ubiquitous Computing Technology for Real Time Enterprises”,» IGI Global, Copyright December 2007.
- [4] M. Weiser και J. S. Brown, «Designing Calm Technology,» Xerox PARC, 1995.
- [5] M. Weiser, «The computer for the 21st century,» *Scientific American*, 1991.
- [6] Μ. Αναστάση, «Μελέτη προσεγγίσεων προστασίας της ιδιωτικότητας σε συστήματα με επίγνωση του πλαισίου χρήσης,» Τμήμα Πληροφορικής του Πανεπιστημίου Κύπρου, Κύπρος, Μάιος 2012.
- [7] C. Janssen, «Ubiquitous Computing,» Techopedia, [Ηλεκτρονικό]. Available: <http://www.techopedia.com/definition/22702/ubiquitous-computing>.
- [8] G. Thyagaraju και U. P. Kulkarn, «Design and Implementation of User Context aware Recommendation Engine for Mobile using Bayesian Network, Fuzzy Logic and Rule Base,» *International Journal of Computer Applications (0975 – 8887)*, τόμ. Volume 40 , αρ. No.3, p. 47–63, February 2012.
- [9] T. Gu, H. K. Pung και D. Zhang, «A Service-Oriented Middleware for Building Context-Aware Services,» *Elsevier Journal of Network and Computer Applications (JNCA)*, τόμ. Vol. 28, αρ. Issue 1, p. pp. 1–18, January 2005.
- [10] «Context-aware pervasive_systems,» Wikipedia, [Ηλεκτρονικό]. Available: http://en.wikipedia.org/wiki/Context-aware_pervasive_systems.
- [11] Κ. Σαρμουρλής, «Ubiquitous and Pervasive Networks,» Πανεπιστήμιο Μακεδονίας Διατμηματικό Πρόγραμμα Μεταπτυχιακών Σπουδών στα Πληροφοριακά Συστήματα Τεχνολογίες Τηλεπικοινωνιών και Δικτύων, Θεσσαλονίκη, 2009.
- [12] M. Weiser, R. Gold και J. S. Brown, «The origins of ubiquitous computing research at PARC in the late 1980s,» *IBM SYSTEMS JOURNAL*, τόμ. VOL 38, αρ. NO 4, pp. pp 693-696, 1999.
- [13] R. C. Schank, *Tell Me a Story: Narrative and Intelligence*, Northwestern University Press, 1995.
- [14] T. Uemukai, Y. Arase, T. Hara και S. Nishio, «OPA Browser: A Web Browser for Cellular Phone Users.,» *Proceedings of the 20th Annual ACM Symposium on User interface Software and Technology*, 2007.
- [15] R. Fisher, «Values for Thinking,» *Nash Pollock, Oxford*, 2001.
- [16] G. J. Hwang, «A conceptual map model for developing intelligent tutoring systems,» *Educational Technology & Society*, τόμ. 16 (3), p. 167–178, 2006.
- [17] S. Yahya, E. A. Ahmad και K. A. Jalil, «The definition and characteristics of ubiquitous learning: A discussion,» τόμ. Vol. 6, αρ. Issue 1, pp. pp. 117-127, 2010.

- [18] P. Elgar, Αισθητήρες Μέτρησης και Ελέγχου, Εκδόσεις TZIOΛΑ, 2003.
- [19] «Sensor,» Wikipedia, [Ηλεκτρονικό]. Available: <http://en.wikipedia.org/wiki/Sensor>.
- [20] «Sensor grid,» Wikipedia, [Ηλεκτρονικό]. Available: http://en.wikipedia.org/wiki/Sensor_grid.
- [21] D. K. Sohraby, D. Minoli και T. Znati, Wireless sensor networks: technology, protocols, and applications, John Wiley & Sons, 2007, p. pp. 203–209.
- [22] Ν. Τσιμπούκη, «Μικροκεραίες σε δίκτυα αισθητήρων για εφαρμογές υγείας,» Διαπανεπιστημιακό Διατμηματικό Πρόγραμμα Μεταπτυχιακών Σπουδών Προηγμένα Συστήματα Υπολογιστών και Επικοινωνιών, Θεσσαλονίκη, Ιούνιος 2009.
- [23] «Sensors: Technologies and Global Markets,» BCC Research LLC, USA, March 2011. [Ηλεκτρονικό]. Available: <http://www.bccresearch.com/market-research/instrumentation-and-sensors/sensors-technologies-markets-ias006d.html>. [Πρόσβαση Report Code: IAS006D].
- [24] Y. S. Prof. Yurish, «Sensors: Smart vs. Intelligent,» *Sensors & Transducers Journal (ISSN 1726- 5479)*, τόμ. Vol.114, pp. pp. I-VI, March 2010.
- [25] Μ. Βερυκοκίδης, «Εξυπνοι αισθητήρες,» Ανώτατο Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης, Σχολή Τεχνολογικών Εφαρμογών, Τμήμα Ηλεκτρολογίας, Ηράκλειο, 2010.
- [26] «Smart device,» Wikipedia, [Ηλεκτρονικό]. Available: http://en.wikipedia.org/wiki/Smart_device.
- [27] «Turn your car into a smart device with Apple iPhone,» TechCrates, 14 Jul 2013. [Ηλεκτρονικό]. Available: <http://www.techcrates.com/vehicle-car-smart-device-iphone/>.
- [28] M. H. Jansen - Vullers, C. A. van Dorp και A. J. Beul, «Managing traceability information manufacture,» *International Journal of Information Management*, pp. 395-413, 2003.
- [29] J. Dr Cooney και B. MSc Winkless, «Utilising TRIZ methodologies to evolve and develop next generation food packaging concepts,» *TheTRIZ-journal*, July 2003.
- [30] Γ. Χαρτουμπέκης, «Ασύρματα δίκτυα αισθητήρων : ανάπτυξη γραφικού περιβάλλοντος με το Matlab για τη λήψη μετρήσεων με χρήση του MTS400/420 board της Crossbow,» Πανεπιστήμιο Πατρών Τμήμα Φυσικής, Οκτώβριος 2010.
- [31] «Flow Control Network,» EBSCO Industries, 2014 . [Ηλεκτρονικό]. Available: <http://www.flowcontrolnetwork.com/articles/q-a-a-world-without-wires>.
- [32] «Environments, Activity-Aware Sensor Network for Smart,» Sensor web Research Laboratory - Georgia State University, 2010. [Ηλεκτρονικό]. Available: <http://sensorweb.cs.gsu.edu/?q=ActiSen>.
- [33] J. Dooley, C. Wagner, H. Hagraς και G. Pruvost, «FollowMe: The Persistent GUI,» σε *Sixth International Symposium on Parallel Computing in Electrical Engineering*, Luton, UK, April 2011.
- [34] «FCC Dedicates Spectrum Enabling Medical Body Area Networks,» Federal Communications Commission, 24 May 2012. [Ηλεκτρονικό]. Available: <http://www.fcc.gov/document/fcc-dedicates-spectrum-enabling-medical-body-area-networks>.

- [35] X. A. Μακρή, «Εφαρμογές των ασύρματων δικτύων στην υγεία,» Μελέτη Συνύπαρξης Ασύρματων Δικτύων Αισθητήρων και Δικτύων Wi-Fi σε Πραγματικό Περιβάλλον, [Ηλεκτρονικό]. Available: http://asirmata-diktua.blogspot.gr/p/blog-page_22.html.
- [36] «Body area network,» Wikipedia, [Ηλεκτρονικό]. Available: http://en.wikipedia.org/wiki/Body_area_network.
- [37] «Smart_city,» Wikipedia, [Ηλεκτρονικό]. Available: http://en.wikipedia.org/wiki/Smart_city.
- [38] «Κόσμος: Στην Ισπανία η έξυπνη πόλη,» Το Βήμα, 15 03 2013. [Ηλεκτρονικό]. Available: <http://www.tovima.gr/world/article/?aid=503113>.
- [39] «Mobile_computing,» Wikipedia, [Ηλεκτρονικό]. Available: http://en.wikipedia.org/wiki/Mobile_computing.
- [40] S. Mann, «WEARABLE COMPUTING as means for PERSONAL EMPOWERMENT,» σε *International Conference on Wearable Computing ICWC-98*, Fairfax, Virginia, May 1998.
- [41] W. Barfield και T. Caudell, «Fundamentals of Wearable Computers and Augmented Reality,» σε *Fundamentals of Wearable Computers and Augmented Reality*, CRC Press, 2001, p. 6.
- [42] «Wearable,» Wikispaces, [Ηλεκτρονικό]. Available: <http://ahci.wikispaces.com/Wearable+Interface>.
- [43] D. P. Siewiorek, «New Frontiers of Application Design,» *No. 12 COMMUNICATIONS OF THE ACM*, τόμ. Vol. 45, pp. p. 79 -82, December 2002.
- [44] «Κάρτα διαδρομών σε δαχτυλίδι,» Ναυτεμπορική, 28 Αυγούστου 2013. [Ηλεκτρονικό]. Available: <http://www.naftemporiki.gr/story/690865>.
- [45] «Meta Spaces Glasses - Μία «εναλλακτική» προσέγγιση στους υπολογιστές κεφαλής,» Ναυτεμπορική, 27 Αυγούστου 2013. [Ηλεκτρονικό]. Available: <http://www.naftemporiki.gr/story/690440>.
- [46] M. R. Ebling, G. D. H. Hunt και H. Lei, «Issues for Context Services for Pervasive Computing,» IBM T. J. Watson Research Center, RC 22104 (99084), June 29 2001.
- [47] K. Henriksen, J. Indulska, T. McFadden και S. Balasubramaniam, «Middleware for Distributed Context-Aware Systems,» *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE*, τόμ. Volume 3760, αρ. pp 846-863, October 31 - November 4, 2005.
- [48] K. Barrett και R. Power, «State of the Art: Context Management,» M-Zones Deliverable 1, Release 2: May 2003.
- [49] «STMicroelectronics and Movea collaborate to bring ultra-low-power, context-aware motion-recognition platform,» [Ηλεκτρονικό]. Available: <http://electroiq.com/blog/2013/09/stmicroelectronics-and-movea-collaborate-to-bring-ultra-low-power-context-aware-motion-recognition-platform/>.
- [50] Y. A. Shaaban, S. McBurney, N. Taylor, H. M. Williams, N. Kalatzis και I. Roussaki, «User Intent to Support Proactivity in a Pervasive System,» σε «*PERSIST Workshop on Intelligent Pervasive Environments*», Edinburgh, Scotland Symposium, 6-9 April 2009.

[κενή σελίδα]



[ΤΕΛΟΣ]