

Τεχνολογικό Εκπαιδευτικό Ίδρυμα Δυτικής Ελλάδας
Σχολή Διοίκησης και Οικονομίας
Τμήμα Διοίκηση Επιχειρήσεων

Πτυχιακή Εργασία

**Περιρρέουσα Νοημοσύνη και
Διάχυτος Υπολογισμός,
Σύγχρονες Εφαρμογές και Προοπτικές**

Σπουδάστριες

Γύλαρη Στυλιανή
Μόλβαλη Ευριδίκη

Επιβλέπων Καθηγητής

Γιωτόπουλος Κωνσταντίνος

Πάτρα, Αύγουστος 2016

Ευχαριστίες

Θα θέλαμε να ευχαριστήσουμε αρχικά τους γονείς μας Λίτσα,Παρασκευά και Βιργινία,Χαράλαμπο καθώς επίσης και τα αδέρφια μας Κυριάκο και Σταύρο που μας στήριξαν όχι μόνο σε αυτούς τους 6 μήνες της εργασίας, ούτε σε αυτά τα 4 χρόνια των σπουδών μας αλλά κυρίως που μας στηρίζουν σε όλη την διάρκεια της ζωής μας!

Επίσης ευχαριστούμε θερμά τον Κ.Γιωτόπουλο Κώστα για την καθοδήγησή του και την ομαλή επίτευξη της παρούσας εργασίας

Περίληψη

Η διαχείριση γνώσης και σημασιολογικής πληροφορίας έχουν βρει εφαρμογή σε πολλές περιοχές της Πληροφορικής & των Τηλεπικοινωνιών τα τελευταία χρόνια. Ο λόγος είναι ότι μια τέτοια διαχείριση μπορεί να προσδώσει στα σύγχρονα και μελλοντικά

υπολογιστικά περιβάλλοντα ευελιξία, προστιθέμενη αξία και χρηστικότητα.

Το όραμα της περιρρέουσας νοημοσύνης (Ambient Intelligence) αφορά μια νέα μορφή υπολογιστικού περιβάλλοντος εμπλουτισμένου με υπολογιστικές συσκευές ενσωματωμένες στο φυσικό περιβάλλον που μπορούν να συνεργάζονται και να εκτελούν εφαρμογές, οι υπηρεσίες των οποίων «ακολουθούν» το χρήστη και είναι άμεσα προσβάσιμες οποτεδήποτε και από οπουδήποτε. Σε μια από τις πιθανές υλοποιήσεις του, η περιρρέουσα τεχνολογία ενσωματώνεται σε καθημερινά αντικείμενα όπως έπιπλα, ρούχα, οικιακές και ηλεκτρικές συσκευές, οχήματα, δρόμους, και «έξυπνα» υλικά, ενώ οι άνθρωποι εξοπλίζονται με εργαλεία και διεργασίες απαραίτητες για την επίτευξη αλληλεπιδράσεων με αυτά τα περιβάλλοντα. Ένα Ami περιβάλλον μπορεί να φιλοξενεί μεγάλο αριθμό Ami εφαρμογών οι οποίες χρησιμοποιούν τις υπηρεσίες υποδομής που παρέχονται από το περιβάλλον και τις συσκευές/αντικείμενα που το απαρτίζουν.

Βασικό χαρακτηριστικό ενός τέτοιου περιβάλλοντος αποτελεί η συνένωση του φυσικού και του ψηφιακού χώρου, δηλαδή τα απτά φυσικά αντικείμενα και περιβάλλοντα αποκτούν και μια ψηφιακή διάσταση. Καθώς ο υπολογιστής «εξαφανίζεται» στα περιβάλλοντα που φιλοξενούν τις δραστηριότητες των ανθρώπων, τα αντικείμενα που βρίσκονται μέσα σε αυτά επταυζάνονται με στοιχεία πληροφορικής τεχνολογίας και επικοινωνίας (Information and Communication Technology components), όπως αισθητήρες, ενεργοποιητές, επεξεργαστές, μνήμη, μονάδες ασύρματης επικοινωνίας κτλ., και αποκτούν δυνατότητες λήψης, αποθήκευσης, επεξεργασίας και μετάδοσης πληροφοριών και δεδομένων. Τα αντικείμενα αυτά αναφέρονται ως «επταυξημένα» αντικείμενα ή τεχνουργήματα, ενώ τα συστήματα τέτοιων αντικειμένων καλούνται συστήματα διάχυτου υπολογισμού (ubiquitous systems).

Η παρούσα πτυχιακή εργασία αρχικά αναλύει τις έννοιες της Περιρρέουσας νοημοσύνης και του Διάχυτου υπολογισμού. Έπειτα επεκτείνετε στην κατάταξη των συστημάτων διάχυτου υπολογισμού και στην αρχιτεκτονική τους, καθώς επίσης και στα συστήματα περιρρέουσας νοημοσύνης όπως και σε διάφορα τεχνουργήματα αυτής. Στη συνέχεια διερευνά το Διαμεσολαβητικό λογισμικό GAS-OS, αναφερόμενη στην σχεδίασή του, στην αρχιτεκτονική του όπως και στην λειτουργικότητά του. Τέλος γίνεται μια αναφορά στα Ασύρματα Δίκτυα Αισθητήρων σε σχέση με τον Διάχυτο προγραμματισμό.

Περιεχόμενα

ΕΙΣΑΓΩΓΗ	7
Κεφάλαιο 1	8
Εισαγωγή – Βασικές Έννοιες	8
1.1 Διάχυτος Υπολογισμός	8
1.2 Περιρρέουσα Νοημοσύνη (Ambient Intelligence)	9
Κεφάλαιο 2	10
Συστήματα Διάχυτου Υπολογισμού	10
2.1 Κατάταξη Συστημάτων Διάχυτου Υπολογισμού	12
2.1.1 Κατάταξη με βάση σχεδιαστικές (μή λειτουργικές) απαιτήσεις	13
2.1.2 Κατάταξη με βάση λειτουργικές απαιτήσεις.....	22
2.2 Αρχιτεκτονική για Συστήματα Διάχυτου Υπολογισμού	26
Κεφάλαιο 3	42
Συστήματα Περιρρέουσας Νοημοσύνης	42
3.1 Φυσική υπόσταση των τεχνουργημάτων	44
3.2 Δομή των τεχνουργημάτων.....	45
3.3 Πρωτότυπα τεχνουργήματα	49
3.3.1 Τεχνούργημα θρανίο(EDesk)	49
3.3.2 Τεχνούργημα Καρέκλα(EChair).....	50
3.3.3 Τεχνούργημα Βιβλίο (Ebook)	51
3.3.4 Τεχνούργημα κύβος Διάθεσης(EMoodCube).....	52
3.3.5 Τεχνούργημα Χαλί(ECarpet).....	53
3.3.6 Τεχνούργημα Φωτιστικού δαπέδου(EFloorLamb) και πορτατίφ (EDeskLamb).....	54
3.3.7 Τεχνούργημα MP3Player(EMp3player).....	55
3.3.8 Τεχνούργημα Ρολόι.....	57
ΚΕΦΑΛΑΙΟ 4	57
Διαμεσολαβητικό Λογισμικό GAS-OS	57
4.1 Τι είναι Διαμεσολαβητικό Λογισμικό (Middleware)	58
4.2 Ανάλυση και Σχεδίαση του GAS-OS	63

4.3 Λειτουργικές Απαιτήσεις Μοντέλου Συνδέσμων/Συνάψεων	66
4.4 Αρχιτεκτονική του GAS-OS	69
4.5 Υλοποίηση του GAS-OS	71
ΚΕΦΑΛΑΙΟ 5	78
ΑΣΥΡΜΑΤΑ ΔΙΚΤΥΑ ΑΙΣΘΗΤΗΡΩΝ	78
5.1 Δίκτυα Αισθητήρων και Διάχυτος Προγραμματισμός	78
5.2. Συνοπτική Περιγραφή των Ασύρματων Δικτύων Αισθητήρων	79
5.3 Δίκτυα Αισθητήρων (WSNs): Σύνομη Ιστορική Αναδρομή, Εφαρμογές και ερευνητικές περιοχές	82
5.3.1 Η Σύνομη Ιστορία των WSNs	82
5.3.2 Εφαρμογές των WSNs	83
5.4 Αρχιτεκτονική Συστήματος	87
5.5 Βέλτιστη Τοποθέτηση SNs	90
5.6 Ενέργεια - Έλεγχος Δικτύου και Επικοινωνιών	93
5.7 Επικοινωνιακά Πρωτόκολλα Εξοικονόμησης Ενέργειας	95
5.8 Ασφάλεια - Έλεγχος Συμμόρφησης	96
Δρομολόγηση Δεδομένων και Διαχείριση Τοπολογίας	97
Συμπεράσματα	100
Βιβλιογραφία	101

Λίστα Πινάκων

Πίνακας 1:Λειτουργικές απαιτήσεις	23
-----------------------------------	----

Λίστα Εικόνων

Εικόνα 1: Προτεινόμενη αρχιτεκτονική για συστήματα διάχυτου υπολογισμού.....	27
Εικόνα 2: Επίπεδο Επικοινωνίας με το διαμεσολαβητικό λογισμικό.....	32
Εικόνα 3: Δομή ενός Τεχνουργήματος.....	46
Εικόνα 4: Άποψη του πρωτότυπου θρανίου και του απαιτούμενου υλικού.....	49
Εικόνα 5: Πρωτότυπη Καρέκλα και απαιτούμενο υλικό.....	51
Εικόνα 6: Άποψη του Πρωτότυπου βιβλίου.....	52
Εικόνα 7: Άποψη του πρωτότυπου κύβου διάθεσης.....	53
Εικόνα 8: Άποψη του πρωτότυπου χαλιού.....	54
Εικόνα 9: Άποψη του φωτιστικού δαπέδου και του πορτατίφ και αντίστοιχου υλικού...	55
Εικόνα 10: Το GAS-OS ως γέφυρα μεταξύ τεχνολογικών επιπέδων αντικειμένων και χρηστών.....	64
Εικόνα 11: Εφαρμογή του GAS-OS.....	64
Εικόνα 12: Πρωτόκολλα επικοινωνίας τεχνουργημάτων	74
Εικόνα 13: Τα επιμέρους συστήματα ενός SN.....	80

ΕΙΣΑΓΩΓΗ

Η περιρρέουσα νοημοσύνη αποτελεί ένα ραγδαία εξελισσόμενο πεδίο των πληροφοριακών συστημάτων με δυνητικά τεράστια απήχηση στο μέλλον. Ο όρος

«ambient intelligence» έχει οριστεί το 2003 από τη συμβουλευτική ομάδα τον προγράμματος πληροφορικής τεχνολογίας της ευρωπαϊκής ένωσης ως η συνένωση του «διάχυτου υπολογισμού», της «διάχυτης επικοινωνίας» και των προσαρμοζόμενων στους χρήστες διεπαφών.

Το όραμα της περιρρέουσας νοημοσύνης είναι ουσιαστικά εξέλιξη της ιδέας του M. Weiser για «ανεπαίσθητες τεχνολογίες» (calm technologic) ο οποίος το 1991 στο άρθρο του «The computer of the 21st Century» έθεσε λεπτομερώς την έννοια του «διάχυτου υπολογισμού» (ubiquitous computing). Ο Weiser, που πλέον θεωρείται ο «πατέρας» του διάχυτου υπολογισμού, μίλησε για πανταχού παρόντες υπολογιστές που υπηρετούν τους ανθρώπους στην καθημερινή τους ζωή, στο επαγγελματικό και στο προσωπικό χώρο, λειτουργώντας αόρατα και διακριτικά, απελευθερώνοντας σε μεγάλο βαθμό τους ανθρώπους από κουραστικές και μονότονες εργασίες. (Garlan 2002)

Με τη σημερινή έννοια, η περιρρέουσα νοημοσύνη αφορά υπολογιστικές συσκευές ενσωματωμένες στο φυσικό περιβάλλον που μπορούν να συνεργάζονται και να εκτελούν εφαρμογές, οι υπηρεσίες των οποίων «ακολουθούν» το χρήστη και είναι άμεσα προσβάσιμες οποτεδήποτε και από οπουδήποτε. Το ερευνητικό κέντρο Xerox Park, το οποίο ηγήτο ο Weiser, ξεκίνησε έρευνες και πειράματα σχετικά με πώς θα έπρεπε να σχεδιαστούν κατάλληλα συστήματα και συσκευές για να υποστηρίξουν και να πραγματοποιήσουν αυτό το όραμα. Μια τέτοια προσέγγιση εξαπλώνεται σε μεγάλο αριθμό από ερευνητικά πεδία συμπεριλαμβανομένων: τεχνολογιών εύρεσης θέσης και εντοπισμού χρηστών, τεχνολογιών ασύρματη πρόσβασης σε δίκτυα, καινοτόμων ασυρμάτων συσκευών, ανεύρεσης υπηρεσιών, τεχνολογιών αισθητήρων, σύλληψης, αντίληψης και έκφρασης τον συγκεκριμένου πλαισίου, λογισμικών που προσαρμόζονται σε αλλαγές ενός δυναμικού περιβάλλοντος (π.χ. λόγω κινητικότητας συσκευών). αλλά και εφαρμογών εξειδικευμένων επιστημονικών πεδίων (π.χ. ιατρικών, βιολογικών, κτλ). Μια από τις πιθανές υλοποιήσεις τον οράματος της περιρρέουσας νοημοσύνης, οι χρήστες των M1 εφαρμογών περιβάλλονται από ένα χώρο πλούσιο σε υπολογιστικές συσκευές κάποιες από τις οποίες μπορεί να είναι «εξαφανισμένες» πίσω από απλά

καθημερινά αντικείμενα. Η περιρρέουσα τεχνολογία μπορεί να βρίσκεται ενσωματωμένη σε έπιπλα, ρούχα, οικιακές και ηλεκτρικές συσκευές, οχήματα, δρόμους, και έξυπνα υλικά. Βασικό χαρακτηριστικό ενός τέτοιου περιβάλλοντος αποτελεί η συνένωση του φυσικού και του ψηφιακού χώρου, δηλαδή τα απ' τα φυσικά αντικείμενα και περιβάλλοντα αποκτούν και μια ψηφιακή διάσταση. Καθώς ο υπολογιστής «εξαφανίζεται» στα περιβάλλοντα που φιλοξενούν τις δραστηριότητες των ανθρώπων, τα αντικείμενα που βρίσκονται μέσα σε αυτά επαυξάνονται με στοιχεία τεχνολογίας Πληροφορικής και επικοινωνιών, όπως αισθητήρες, ενεργοποιητές, επεξεργαστές, μνήμη, μονάδες ασύρματος επικοινωνίας κτλ , και αποκτούν δυνατότητες λήψης, αποθήκευσης.

Κεφάλαιο 1

Εισαγωγή – Βασικές Έννοιες

1.1 Διάχυτος Υπολογισμός

Αρκετά συνηθισμένο είναι ότι η τεχνολογία εξελίσσεται συνεχώς. Καινούργιες τεχνολογίες και εφαρμογές καλύπτουν την επαγγελματική και καθημερινή μας ζωή, σε σημείο που δεν γίνεται αντιληπτή. Με τον τρόπο αυτό θα πραγματοποιηθεί το όραμα του Mark Weiser για *“ήρεμη τεχνολογία”*. Ο MarkWeiser έγινε γνωστός για την σκιαγράφιση και ανάλυση του Παροχών Παρούσης Υπολογιστική αλλιώς και Διάχυτος Υπολογισμός (Ubiquitous Computing), είναι μια έννοια στην τεχνολογία λογισμικού και στην επιστήμη των υπολογιστών. (Garlan 2002)

Κάποιες από τις αρχές που περιέγραψε ο MarkWeiser για τον Διάχυτο Υπολογισμό ήταν, ένας υπολογιστής χαρακτηρίζεται ως καλύτερος όταν υπάρχουν αρκετοί από αυτούς στον χώρο διασκορπισμένοι-διάχυτοι, και δεν γίνονται αντιληπτοί από τους χρήστες, ένας από τους βασικούς σκοπούς του

είναι να ενδέχεται να επιλύει τα προβλήματα, αλλά να μπορεί να επεκτείνει τις υποσυνείδητες ικανότητες μας.

Ένας ορισμός για το Διάχυτο Υπολογισμό είναι ο εξής: Η σχεδίαση και η δημιουργία υπολογιστικών περιβαλλόντων που περιλαμβάνουν υλικό, επικοινωνίες και λογισμικό και που είναι πλήρως και διαφανώς ενσωματωμένα στο φυσικό περιβάλλον του χρήστη. (Garlan 2002)

Για την υλοποίηση του χρειάζεται να περάσει από κάποια στάδια και διαφορετικές ερευνητικές περιοχές.

- Έξυπνες και αναπροσαρμοζόμενες διεπαφές χρήστη (adaptive user interfaces)
- Επίγνωση πλαισίου και κατάστασης (context and situationawareness)
- Κυβερνοβοσκή (cyberforaging)
- Προδραστικότητα (proactivity)
- Ευφυείς Χώροι (smartspaces)

1.2 Περιρρέουσα Νοημοσύνη (Ambient Intelligence)

Η έννοια της περιρρέουσας νοημοσύνης αναφέρεται στο όραμα της εξέλιξης της κοινωνίας της γνώσης: οι άνθρωποι περιβάλλονται από φυσικές ευφυείς διεπαφές χρήσης που είναι ενσωματωμένες σε όλα τα καθημερινά αντικείμενα. Το περιβάλλον αυτό είναι σε θέση να αναγνωρίζει την παρουσία κάθε ανθρώπου, και να προσαρμόζεται σ' αυτήν. Και το πιο σημαντικό είναι ότι οι άνθρωποι δε συνειδητοποιούν ότι βρίσκονται σε ένα υπολογιστικό περιβάλλον επειδή είναι συνήθως «αόρατη». Η έμφαση δηλαδή είναι στην ευκολία χρήσης, και στην υποστήριξη και ενδυνάμωση της αλληλεπίδρασης των ανθρώπων με το περιβάλλον (Weberetal 2005).

Κεφάλαιο 2

Συστήματα Διάχυτου Υπολογισμού

Τα συστήματα διάχυτου υπολογισμού αποτελούν επέκταση των κατανεμημένων συστημάτων πάνω από ασύρματες δικτυακές υποδομές και υπηρεσίες λογισμικού με στόχο την αδιάλειπτη πρόσβαση σε απομακρυσμένες και διάχυτες πληροφορίες ικανοποιώντας απαιτήσεις ποιότητας υπηρεσιών. Στο κεφάλαιο αυτό θα παρουσιαστούν αρχικά βασικές έννοιες και προκλήσεις των κατανεμημένων συστημάτων, και στη συνέχεια θα αναλυθεί η επέκτασή τους στο χώρο της περιρρέουσας νοημοσύνης, μαζί με ένα σύνολο από νέες απαιτήσεις και νέους στόχους. (Weberetal 2005).

Η εξέλιξη των κατανεμημένων συστημάτων συμβαδίζει και με τη ραγδαία πρόοδο στην ηλεκτρονική και τις ασύρματες τεχνολογίες που έχει ως αποτέλεσμα τη διάδοση ευρείας κλίμακας προϊόντων με υπολογιστικές ικανότητες και δυνατότητες επικοινωνίας μέσω δικτύων, πέραν των προσωπικών υπολογιστών που χρησιμοποιούσαμε μέχρι σήμερα. Συσκευές με μικρό μέγεθος, όπως κινητά τηλέφωνα, προσωπικοί ψηφιακοί βοηθοί, ψηφιακές κάμερες, αισθητήρες μέτρησης περιβαλλοντικών φαινομένων ή ιατρικών παραμέτρων κτλ., εισέρχονται συνεχώς στην καθημερινή μας ζωή προσφέροντας νέες υπηρεσίες στους χρήστες τέτοιων εφαρμογών. Ενώ το μέγεθος των συσκευών γίνεται όλο και πιο μικρό, οι υπολογιστικές τους δυνατότητες γίνονται μεγαλύτερες, όπως και οι δυνατότητες των δικτύων που υποστηρίζουν την επικοινωνία τους. Αυτό δίνει τη δυνατότητα στους προγραμματιστές αυτής της νέας μορφής κατανεμημένων συστημάτων, των συστημάτων διάχυτου υπολογισμού, να δημιουργούν ολοένα και πιο λειτουργικές και ισχυρές εφαρμογές. (Weberetal 2005).

Κατά την ανάπτυξη κατανεμημένων εφαρμογών οι σχεδιαστές δε χρειάζεται να αντιμετωπίσουν ρητά τα προβλήματα που σχετίζονται με την κατανομή, όπως ετερογένεια, κλιμάκωση, διαμοίραση πόρων και ανοχή

σφαλμάτων. Διαμεσολαβητικό λογισμικά (middleware) που αναπτύσσονται πάνω από δικτυακά συστήματα παρέχουν στους σχεδιαστές εφαρμογών μεγαλύτερο επίπεδο αφαιρετικότητας, κρύβοντας την πολυπλοκότητα που υπάρχει λόγω της χωροταξικής διασποράς των κόμβων. Υπάρχοντα διαμεσολαβητικό λογισμικά που είναι προσανατολισμένα σε δοσοληψίες (transaction-oriented), σε μηνύματα (message-oriented) ή σε αντικείμενα (object-oriented) δημιουργήθηκαν για να κρύψουν όσο το δυνατό περισσότερο την κατανομή, έτσι ώστε το σύστημα να εμφανίζεται ως ένα ολοκληρωμένο σύνολο υπολογιστικών υπηρεσιών. Με άλλα λόγια η κατανομή των υπολογιστικών πόρων γίνεται διαφανής (transparent) στον αποδέκτη (χρήστη) των υπηρεσιών του συστήματος. (Garlan 2005)

Αυτές οι τεχνολογίες σχεδιάστηκαν και έχουν χρησιμοποιηθεί επιτυχώς για σταθερά κατανεμημένα συστήματα. Όμως, όπως θα διασαφηνιστεί και στη συνέχεια, μερικές από τις απαιτήσεις που οφείλονται στη δυναμική φύση των συστημάτων διάχυτου υπολογισμού δε μπορούν να αντιμετωπιστούν από τις υπάρχουσες παραδοσιακές αρχιτεκτονικές. Για παράδειγμα, βασικά στοιχεία αλληλεπίδρασης όπως κατανεμημένες δοσοληψίες, αιτήσεις αντικειμένων (objectrequests) ή απομακρυσμένες κλήσεις διαδικασιών (remoteprocedurecalls), υποθέτουν σταθερή, υψηλού εύρους ζώνη και συνεχή σύνδεση μεταξύ των αλληλοεπιδρώντων κόμβων. Επιπλέον η σύγχρονη και από σημείο-σε-σημείο (point-to-point) επικοινωνία που υποστηρίζεται από διαμεσολαβητικά λογισμικά προσανατολισμένα στα αντικείμενα, όπως το CORBA [CORBA, 1998], απαιτούν ένα σημείο συνάντησης (rendez-vous) μεταξύ του πελάτη που ζητά μια υπηρεσία και του εξυπηρετητή που παρέχει την υπηρεσία. Αντιθέτως, σε συστήματα διάχυτου υπολογισμού, η αδυναμία προσιτότητας ενός κόμβου (unreachability) δεν είναι η εξαίρεση ενώ η σύνδεση μπορεί να μην είναι σταθερή. Επίσης, είναι πολύ πιθανό οι πελάτες και οι εξυπηρετητές να μην είναι συνδεδεμένοι στο σύστημα την ίδια χρονική στιγμή, εξαιτίας εκουσίων αποσυνδέσεων (π.χ. για εξοικονόμηση ενέργειας) ή και αναγκαστικών αποσυνδέσεων (π.χ. απώλεια κάλυψης δικτύου). Η αποσύνδεση αντιμετωπίζεται σαν σύνηθες σφάλμα από πολλά παραδοσιακά διαμεσολαβητικά λογισμικά. Παρ' όλα αυτά, τεχνικές διαμοίρασης δεδομένων και δημιουργίας αντιγράφων που έχουν υιοθετηθεί με

επιτυχία από παραδοσιακά συστήματα, μπορεί να μην είναι κατάλληλες και έτσι νέες μεθοδολογίες πρέπει να διερευνηθούν (Weberetal 2005).

2.1 Κατάταξη Συστημάτων Διάχυτου Υπολογισμού

Τα περισσότερα συστήματα διάχυτου υπολογισμού χρησιμοποιούν κάποιο είδος διαμεσολαβητικού λογισμικού για την αποτελεσματική διαχείριση και προγραμματισμό ενός κατανεμημένου συστήματος, χωρίς να αποκλείεται και η χρήση εναλλακτικών αρχιτεκτονικών. Τέτοιες διαμεσολαβητικές πλατφόρμες αποτελούν το μεσάζοντα μεταξύ του χώρου εκτέλεσης μια κατανεμημένης εφαρμογής και του υφιστάμενου δικτύου που χρησιμοποιείται για την αποστολή μηνυμάτων σε απομακρυσμένες οντότητες που συνεργάζονται σε μια κατανεμημένη εφαρμογή. Οι απομακρυσμένες οντότητες μπορεί να είναι άλλες κατανεμημένες εφαρμογές, απομακρυσμένοι εξυπηρετητές, το υλικό ενός υπολογιστή ή και αισθητήρες που συλλέγουν δεδομένα από το περιβάλλον στο οποίο βρίσκονται. Η επικοινωνία μεταξύ απομακρυσμένων οντοτήτων που δε βρίσκονται σε ένα μηχάνημα, αλλά τοποθετημένες σε περιβάλλοντα ετερογενούς δικτύωσης μεγάλης κλίμακας, είναι ένα αρκετά δύσκολο έργο λόγω της πληθώρας διαφορετικών απαιτήσεων που πρέπει να ληφθούν υπόψη από τους προγραμματιστές συστημάτων. Η κλιμάκωση (scalability), η ετερογένεια (heterogeneity), η ανοχή σε σφάλματα (faulttolerance), η ανοικτότητα (openness) και η ασφάλεια (security) είναι μερικά παραδείγματα σχεδιαστικών (μη-λειτουργικών) απαιτήσεων που πρέπει να ικανοποιηθούν κατά την ανάπτυξη μιας πλατφόρμας ώστε αυτή να μπορεί να ικανοποιήσει τις ανάγκες διαφορετικών εφαρμογών περιρρέουσας νοημοσύνης (Johanson 2002).

Από την άλλη, όπως ήδη αναφέρθηκε, ένα διαμεσολαβητικό λογισμικό βρίσκεται μεταξύ μιας κατανεμημένης εφαρμογής και του δικτύου. Αυτό υπονοεί και έναν αριθμό από λειτουργικές απαιτήσεις που πρέπει να ληφθούν υπόψη προκειμένου να καταστεί εφικτή η επικοινωνία μεταξύ της εφαρμογής που δε

γνωρίζει τη δικτυακή υποδομή και του ίδιου του δικτύου. Απαιτήσεις όπως ειδοποίηση με γεγονότα (eventnotification), αναζήτηση περιβάλλοντος (addressing), ανεύρεση υπηρεσιών (servicediscovery), επίγνωση του συγκειμένου πλαισίου (contextawareness) σχετίζονται εγγενώς με τη λειτουργικότητα του συστήματος.

2.1.1 Κατάταξη με βάση σχεδιαστικές (μη λειτουργικές) απαιτήσεις

Όπως αναφέρθηκε και προηγουμένως, κατά την ανάπτυξη ενός διαμεσολαβητικού λογισμικού πρέπει να ληφθεί υπόψη ένας αριθμός απαιτήσεων. Ένα διαμεσολαβητικό λογισμικό είναι και αυτό ένα καταναμημένο σύστημα. Άρα θα πρέπει να ικανοποιεί τις απαιτήσεις κάθε τύπου καταναμημένου συστήματος, δηλαδή ένα σύστημα αποτελούμενο από ψηφίδες (components) λογισμικού που βρίσκονται - εκτελούνται σε διαφορετικές υπολογιστικές συσκευές και σε διαφορετικές τοποθεσίες, και που παρέχουν υπηρεσίες σε εφαρμογές αλληλοεπιδρώντας μεταξύ τους. Σε όλα τα καταναμημένα συστήματα, η αλληλεπίδραση μεταξύ απομακρυσμένων ψηφίδων λογισμικού πραγματοποιείται μέσω ανταλλαγής μηνυμάτων πάνω από μια υποκείμενη δικτυακή υποδομή. Μια λίστα απαιτήσεων για καταναμημένα συστήματα παρέχεται από το [Coulouris, 2001]. Η λίστα αυτή περιλαμβάνει το χειρισμό σφαλμάτων, την ανοικτότητα, την ετερογένεια, την κλιμάκωση, τη διαμοίραση πόρων (resourcesharing), τη διαφάνεια (transparency), τον ταυτοχρονισμό (concurrency) και την ασφάλεια (Coulouris 2001).

Στην έρευνα μας θα χρησιμοποιηθεί μέρος της λίστας των απαιτήσεων που αναφέρεται στο (Coulouris, 2001), εκτός της απαίτησης της διαφάνειας η οποία καλύπτεται εξ' ορισμού από τα διαμεσολαβητικά λογισμικά και της απαίτησης της ταυτόχρονης χρήσης. Η τελευταία αποκτά νέες προεκτάσεις στο χώρο των διάχυτων συστημάτων από ότι στο χώρο των καταναμημένων όπου χρειάζεται να αντιμετωπισθεί όταν εφαρμογές και υπηρεσίες παρέχουν πόρους που μπορούν να διαμοιράζονται από πελάτες σε ένα καταναμημένο σύστημα. Για

να είναι ένα αντικείμενο ασφαλές σε ένα τέτοιο περιβάλλον, οι λειτουργίες του θα πρέπει να συγχρονίζονται με τέτοιο τρόπο ώστε τα δεδομένα να παραμένουν συνεπή. Σε ένα περιβάλλον διάχυτου υπολογισμού οι λειτουργίες είναι εξαιρετικά δύσκολο να συγχρονιστούν, ενώ σημαντική παράμετρος στη διαμοίραση πόρων αποτελεί η προτεραιότητα και η κρισιμότητα της ψηφίδας-πελάτη, που ποικίλει ανάλογα με την εκάστοτε εφαρμογή και σχετίζεται με τη λειτουργία του συστήματος. Για το λόγο αυτό, η απαίτηση της ταυτόχρονης χρήσης έχει αφαιρεθεί από τη λίστα των σχεδιαστικών απαιτήσεων της εργασίας, και έχει ενσωματωθεί στη λειτουργική απαίτηση του σχεδίου δράσης που περιγράφεται στην επόμενη ενότητα.

Η λίστα των σχεδιαστικών (μη-λειτουργικών) απαιτήσεων θα εμπλουτιστεί με επιπλέον απαιτήσεις οι οποίες πρέπει επίσης να διευθετηθούν από τους σχεδιαστές/προγραμματιστές συστημάτων διάχυτου υπολογισμού. Το σύνολο των απαιτήσεων θα χρησιμοποιηθούν για την κατάταξη και την αποτίμηση των συστημάτων, σύμφωνα με το αν και σε ποιο βαθμό ένα συγκεκριμένο σύστημα διάχυτου υπολογισμού ικανοποιεί κάθε μία. Η λίστα των απαιτήσεων που αναφέρονται στην ενότητα αυτή δεν αποτελεί κλειστό σύνολο, δηλαδή ο σχεδιαστής ενός συστήματος έχει τη δυνατότητα να λάβει υπόψη του επιπλέον απαιτήσεις, όπως την διατήρηση της ιδιωτικότητας (privacy), την ύπαρξη διαχειριστικών πεδίων επίβλεψης (administrativesupervision) ή κριτήρια κοστολόγησης (Johanson 2002).

Στη συνέχεια επεξηγούνται οι απαιτήσεις της σχεδιακής (μη λειτουργικές) λαμβάνοντας υπόψη τους συγκεκριμένους παράγοντες που πιθανώς μπορούν να επηρεάσουν τις σχεδιαστικές απαιτήσεις ενός διαμεσολαβητικού λογισμικού που λειτουργεί σε ένα περιβάλλον περιρρέουσας νοημοσύνης (Coulouris 2001).

Αντιμετώπιση ετερογένειας:

Μερικές από τις εκατοντάδες ψηφίδες ενός συστήματος διάχυτου υπολογισμού είναι οικιακές συσκευές, έξυπνα έπιπλα, αυτοκίνητα, ιατρικός εξοπλισμός, κάμερες, κινητά τηλέφωνα, παιχνίδια, αισθητήρες και ενεργοποιητές. Οι ψηφίδες αυτές μπορεί να είναι υλοποιημένες σε διαφορετικές γλώσσες προγραμματισμού, να εκτελούνται σε διαφορετικά λειτουργικά συστήματα, σε διαφορετικές πλατφόρμες υλικού, και να χρησιμοποιούν διαφορετικές τεχνολογίες επικοινωνίας. Ένα διαμεσολαβητικό λογισμικό για τέτοια συστήματα θα πρέπει να μπορεί να αντιμετωπίζει τα παραπάνω είδη ετερογένειας, ενσωματώνοντας κατάλληλες αφαιρετικές έννοιες και μοντέλα που θα καθιστούν τη διαφορετικότητα των ψηφίδων διαφανή προς τους προγραμματιστές διάχυτων συστημάτων, παρέχοντας ευέλικτη πλατφόρμα επικοινωνίας γεφυρώνοντας τα ανομοιογενή πρωτόκολλα που υποστηρίζονται από τις διάφορες συσκευές, χρησιμοποιώντας κινητό κώδικα, δηλαδή κώδικα που μπορεί να αποστέλλεται από μια ψηφίδα σε μια άλλη και να εκτελείται στον προορισμό, κτλ.

Ανοικτότητα:

Η ανοικτότητα ενός διάχυτου συστήματος καθορίζεται πρωταρχικά από το βαθμό που νέες υπηρεσίες μπορούν να προστεθούν και να καταστούν διαθέσιμες για χρήση από ψηφίδες-πελάτες, όπως και από το βαθμό που υπάρχουσες υπηρεσίες μπορούν να αναπρογραμματιστούν για να αντιμετωπίσουν μεταβαλλόμενες λειτουργικές απαιτήσεις. Στα ανοικτά συστήματα οι βασικές τους διεπαφές είναι γνωστές ή μπορούν εύκολα να γνωστοποιηθούν. Οι διεπαφές αυτές μαζί με την παροχή ομοιόμορφων μηχανισμών επικοινωνίας αποτελούν το βασικό τρόπο πρόσβασης σε πόρους και υπηρεσίες. Ένα διαμεσολαβητικό λογισμικό για διάχυτα συστήματα θα πρέπει να έχει τα κατάλληλα χαρακτηριστικά για να καταστήσει ανοικτό το σύστημα που υποστηρίζει. Έτσι θα πρέπει να μπορεί να επεκτείνεται και να τροποποιείται κατά τη διάρκεια ζωής του, να παρέχει κατάλληλα πρωτόκολλα που να επιτρέπουν τη γνωστοποίηση των διεπαφών των ψηφίδων και μηχανισμούς για τη χρήση των

διεπαφών αυτών για την επικοινωνία και διασύνδεσή τους. Επίσης, επειδή τα δεδομένα δεν πρέπει να σταματήσουν να παρέχονται στην εφαρμογή, ειδικά στην περίπτωση εφαρμογών πραγματικού χρόνου, η διαδικασία της ενημέρωσης ή επέκτασης του διαμεσολαβητικού λογισμικού δεν πρέπει να απαιτεί παύση της λειτουργίας του (Jacquet. 2005).

Κλιμάκωση:

Ένα σύστημα χαρακτηρίζεται ως κλιμακούμενο αν παραμένει αποτελεσματικό όταν συμβαίνει σημαντική αύξηση στον αριθμό των πόρων και στον αριθμό των χρηστών του. Σε σύγκριση με παραδοσιακά καταναμημένα συστήματα των οποίων οι κόμβοι επικοινωνούν είτε με σταθερές είτε με κινητές συνδέσεις, ο αριθμός των ψηφίδων σε ένα σύστημα διάχυτου υπολογισμού μπορεί να είναι κατά τάξεις μεγέθους μεγαλύτερος από τους κόμβους ενός παραδοσιακού καταναμημένου συστήματος. Για παράδειγμα σε μια εφαρμογή έξυπνου κτιρίου, είναι πιθανό όλα τα αντικείμενα (π.χ. έπιπλα, στυλό, χρηστικά είδη, ρούχα, κτλ.) να είναι ψηφίδες ενός συστήματος διάχυτου υπολογισμού. Συνεπώς είναι σημαντικό το σύστημα να μπορεί να κλιμακώνει καλά ως προς τον αριθμό των κόμβων και των συνδέσεων μεταξύ τους. Άλλοι παράγοντες που μπορεί να επηρεάζουν την κλιμάκωση είναι ο μέγιστος αριθμός ταυτόχρονων χρηστών, ο αριθμός των δοσοληψιών που εκτελούνται στη μονάδα του χρόνου, κτλ. (Jacquet. 2005)

Χειρισμός σφαλμάτων:

Ο χειρισμός σφαλμάτων αναφέρεται στη δυνατότητα των συστημάτων να ανακάμπτουν από σφάλματα χωρίς παύση ολόκληρου του συστήματος. Σε ένα σύστημα διάχυτου υπολογισμού, μερικοί κόμβοι μπορεί να δυσλειτουργούν ή να παρεμποδίζεται η επικοινωνία τους λόγω έλλειψης ενέργειας ή να πάθουν

κάποια φυσική βλάβη ή να δεχθούν εξωτερικές παρεμβολές. Οι ασύρματοι σύνδεσμοι που χρησιμοποιούνται για την επικοινωνία στο δίκτυο μπορεί επίσης να σταματήσουν να λειτουργούν ή να παρουσιάσουν προβλήματα όπως σφάλματα μετάδοσης λόγω αναξιόπιστου μέσου μετάδοσης και μη-ανιχνεύσιμων συγκρούσεων πακέτων, αυξημένη και απρόβλεπτη καθυστέρηση, υψηλή απώλεια πακέτων κτλ. Τα σφάλματα που μπορεί να συμβούν σε ένα σύστημα διάχυτου υπολογισμού είναι πιο κοινά και πιο συχνά απ' ό τι σε ένα κινητό δίκτυο ή σε ένα καταμεμημένο σύστημα με μόνιμες συνδέσεις και τερματικούς κόμβους. Η δυσλειτουργία μεμονωμένων κόμβων δε θα πρέπει να επηρεάζει τη συνολική εργασία του συστήματος διάχυτου υπολογισμού, κάτι που οδηγεί σε αυξημένες ανάγκες παροχής μηχανισμών διασφάλισης της ανοχής σε σφάλματα. Ένας άλλος παράγοντας που καθιστά απαραίτητη την ανοχή σε σφάλματα είναι ότι μετά την εγκατάσταση μιας εφαρμογής, αλλαγές στην τοπολογία είναι πολύ πιθανό να συμβούν λόγω αλλαγών στις θέσεις των κόμβων, στην δυνατότητα επικοινωνίας μαζί τους (λόγω παρεμβολών, θορύβου, κινητών αντικειμένων κτλ.). Επίσης, πρόσθετοι κόμβοι μπορεί να εισέλθουν ανά πάσα στιγμή στο σύστημα για να αντικαταστήσουν άλλους κόμβους, ενώ κάποιοι κόμβοι μπορεί να σταματήσουν να λειτουργούν λόγω έλλειψης ενέργειας. Έτσι, ακόμα και σε περιπτώσεις συνεχών αλλαγών της τοπολογίας του δικτύου, το διαμεσολαβητικό λογισμικό θα πρέπει να είναι σε θέση να διεκπεραιώνει τις εργασίες του και να παρέχει αξιόπιστες υπηρεσίες στην εφαρμογή (Jacquet. 2005).

Ασφάλεια:

Η ασφάλεια των υπολογιστικών πόρων έχει τρεις συνιστώσες: την εμπιστευτικότητα (confidentiality), δηλαδή την προστασία απέναντι στην αποκάλυψή τους σε μη-εξουσιοδοτημένους χρήστες, την ακεραιότητα (integrity), δηλαδή την προστασία απέναντι στην παραποίηση ή καταστροφή τους και τη διαθεσιμότητα (availability), δηλαδή την προστασία απέναντι σε παρεμβολές που παρεμποδίζουν την πρόσβαση στους πόρους αυτούς. Στο χώρο των

κατανεμημένων και διάχυτων συστημάτων εγείρονται και δύο νέες απαιτήσεις ασφαλείας: η παρεμπόδιση σε επιθέσεις υπηρεσιών και η ασφάλεια κινητού κώδικα. Τέλος, στο χώρο των διάχυτων συστημάτων, η πιθανότητα φυσικής αιχμαλώτισης ψηφίδων του συστήματος θέτει νέα ζητήματα στο χώρο της ασφαλείας. Το διαμεσολαβητικό λογισμικό θα πρέπει να παρέχει κατάλληλους μηχανισμούς ασφαλείας όπως ταυτοποίηση (authentication), εξουσιοδότηση (authorization), και στατιστικά χρήσης (accounting) για έξυπνο έλεγχο της πρόσβασης σε υπολογιστές και δικτυακούς πόρους, για την επιβολή πολιτικών, την τήρηση στατιστικών στοιχείων χρήσης του δικτύου από όπου μπορούν να εξαχθούν συμπεράσματα παραβίασης της ασφαλείας κτλ. (Garlan, 2002).

Επίδοση:

Η επίδοση μπορεί να αποτελεί απαίτηση του συστήματος σε διάφορες περιπτώσεις. Για παράδειγμα ένα διαμεσολαβητικό λογισμικό που χρησιμοποιείται σε μια κατανεμημένη εφαρμογή πραγματικού χρόνου θα πρέπει να εκτελεί κάποιες λειτουργίες στον ελάχιστο δυνατό χρόνο, ενώ μια συσκευή περιορισμένης μνήμης μπορεί να απαιτεί βελτιστοποίηση του συστήματος ως προς τη μνήμη. Παράγοντες όπως η κατανάλωση ενέργειας και η αποδοτικότητα μνήμης πρέπει να λαμβάνονται υπόψη θεωρώντας τους περιορισμούς ενέργειας και μνήμης σε ένα σύστημα διάχυτου υπολογισμού. Για παράδειγμα, καθώς οι κόμβοι ενός συστήματος διάχυτου υπολογισμού λειτουργούν κατά κύριο λόγο με μπαταρίες, θα πρέπει να αναπτυχθούν μηχανισμοί βελτίωσης της κατανάλωσης ενέργειας για την μεγιστοποίηση του χρόνου ζωής και αυτονομίας του συστήματος. Επιπλέον πρωτόκολλα που λαμβάνουν υπόψη την ενέργεια θα πρέπει να επιλεγθούν και να ρυθμιστούν κατάλληλα έτσι ώστε να ελαχιστοποιηθεί ο αριθμός των κόμβων που η ενέργειά τους εξαντλείται ή να μεγιστοποιηθεί ο χρόνος στον οποίο η εφαρμογή μπορεί να λειτουργήσει ώστε να φέρει εις πέρας το έργο που της έχει ανατεθεί. Ένα άλλο θέμα επίδοσης είναι εάν το διαμεσολαβητικό λογισμικό μπορεί να ικανοποιήσει τις απαιτήσεις πραγματικού χρόνου μιας εφαρμογής. Η απαίτηση ενός διαμεσολαβητικού

λογισμικού με δυνατότητες πραγματικού χρόνου είναι πολύ σημαντική σε περιπτώσεις που το σύστημα ή μέρος του συστήματος χρησιμοποιείται από εφαρμογές που πρέπει να λαμβάνουν δεδομένα πραγματικού χρόνου από αισθητήρες (π.χ. θερμοκρασίες, σεισμικές δονήσεις, κτλ.). Τέλος συγκεκριμένα κριτήρια επίδοσης μπορεί να τεθούν από τις εκάστοτε εφαρμογές, κάτι που καθιστά απαραίτητη την παραμετροποίηση του διαμεσολαβητικού λογισμικού και τον καλό σχεδιασμό (Garlan, 2002).

Προσαρμοστικότητα:

Μεταβολή στις λειτουργίες των εφαρμογών και των χρηστών οφειλόμενη σε αλλαγές στις απαιτήσεις των χρηστών ή των χαρακτηριστικών ή των συνθηκών λειτουργίας των κόμβων του δικτύου μπορεί να απαιτούν την παρουσία μηχανισμών προσαρμογής. Η ανάγκη υλοποίησης μηχανισμών προσαρμογής σε ένα διαμεσολαβητικό λογισμικό που λειτουργεί σε ένα περιβάλλον περιρρέουσας νοημοσύνης, οφείλεται κυρίως σε δύο διαφορετικούς τύπους μεταβολών που μπορεί να συμβούν κατά τη λειτουργία του. Από τη μία έχουμε αλλαγές που μπορεί να συμβούν στο δίκτυο. Για παράδειγμα μπορεί να μεταβληθεί η τοπολογία του δικτύου, λόγω της προσθήκης νέων κόμβων ή τη δυσλειτουργία υπαρχόντων. Επίσης, όταν μια κινητή συσκευή εισέρχεται σε δίκτυο που υποστηρίζει διαφορετικό πρωτόκολλο μεταφοράς (transportprotocol) από αυτό που προσφέρεται από το τρέχον δίκτυο που βρίσκεται η συσκευή, τότε θα πρέπει το σύστημα να επιλέξει αυτόματα κατάλληλο πρωτόκολλο που θα επιτρέψει την ομαλή λειτουργία της συσκευής στο νέο της περιβάλλον. Οι μεταβολές αυτές πρέπει να ληφθούν υπόψη από το διαμεσολαβητικό λογισμικό έτσι ώστε οι απαιτήσεις της εφαρμογής να εξακολουθήσουν να ικανοποιούνται. Από την άλλη, εφαρμογές που χρησιμοποιούν το διαμεσολαβητικό λογισμικό μπορεί να μεταβάλλουν τις απαιτήσεις τους σε χρόνο εκτέλεσης, κάτι που απαιτεί προσαρμοστικότητα από το διαμεσολαβητικό λογισμικό για να μπορέσει να

ανταπεξέλθει στις νέες απαιτήσεις. Για παράδειγμα η ακρίβεια των δεδομένων που είναι επιθυμητή από την εφαρμογή μπορεί να αλλάξει κατά τη διάρκεια ζωής της εφαρμογής. Αυτό με τη σειρά του υπονοεί ότι οι μηχανισμοί φιλτραρίσματος στα δεδομένα που λαμβάνονται από τους αισθητήρες ενός τεχνουργήματος θα πρέπει να ενημερωθούν ώστε να παρέχουν στην εφαρμογή τη νέα επιθυμητή ακρίβεια (Garlan, 2002).

Εφικτότητα:

Περιορισμοί υλικού όπως η διαθέσιμη μνήμη ενός κόμβου, η εναπομείνασα ενέργεια και η υπολογιστική ισχύς του είναι παραδείγματα περιορισμένων πόρων που πρέπει να λαμβάνονται υπόψη από το διαμεσολαβητικό λογισμικό. Ένας άλλος περιορισμός είναι το περιορισμένο εύρος ζώνης, η αυξημένη απώλεια πακέτων και η καθυστέρηση που μπορεί να παρατηρηθούν σε ένα σύστημα διάχυτου υπολογισμού. Αυτό σημαίνει ότι δεν είναι δυνατό όλες οι εργασίες να έλθουν εις πέρας εγκαίρως από κάθε κόμβο του συστήματος. Για το λόγο αυτό το διαμεσολαβητικό λογισμικό θα πρέπει να είναι ενημερωμένο για τους περιορισμούς των διαθέσιμων πόρων και να κατανέμει και να διαχειρίζεται τις εργασίες ανάλογα, έτσι ώστε να επιτυγχάνει την επιθυμητή συμπεριφορά (Goumoroulas, 2007).

Συνθεσιμότητα:

Οι ψηφίδες ενός συστήματος διάχυτου υπολογισμού μπορούν να χρησιμοποιηθούν ως δομικά στοιχεία μεγαλύτερων και πιο πολύπλοκων συστημάτων. Αυτό οφείλεται ως ένα βαθμό στη μονάδα επικοινωνίας που ενσωματώνουν, ενώ η πραγματοποίηση της σύνθεσης απαιτεί από τις ψηφίδες που συμμετέχουν, να έχουν καλά ορισμένες διεπαφές, σαφείς εισόδου - εξόδους και κοινές περιγραφές των εργασιών και υπηρεσιών που μπορούν να

προσφέρουν. Η εκδήλωση της συνθεσιμότητας μιας ψηφίδας μπορεί να προϋδεάσει τους χρήστες για τον τρόπο με τον οποίο μπορεί να χρησιμοποιηθεί σε συνδυασμό με άλλες ψηφίδες.

Αυτονομία:

Μια άλλη απαίτηση που συναντάται στα συστήματα διάχυτου υπολογισμού είναι η διατήρηση της αυτονομίας των ψηφίδων, σύμφωνα με την οποία κάθε ψηφίδα πρέπει να λειτουργεί έστω και στοιχειωδώς αυτόνομα από την ύπαρξη άλλων ψηφίδων και υποδομών. Η διατήρηση της αυτονομίας θα πρέπει να ισχύει ακόμα και για συσκευές περιορισμένων πόρων. Λαμβάνοντας υπόψη την μη- προκαθορισμένη (ad hoc) φύση ενός περιβάλλοντος περιρρέουσας νοημοσύνης όπως και το γεγονός ότι η μη-διαθεσιμότητα υπηρεσιών είναι ένα συχνό φαινόμενο και όχι η εξαίρεση, είναι φανερό ότι η αυτονομία αποτελεί πρωταρχικό παράγοντα για τη διατήρηση της ομαλής λειτουργίας μιας εφαρμογής. Επιπλέον, η αυτονομία των ψηφίδων καθιστά τις εφαρμογές πιο ευέλικτες και εύκολα μετακινήσιμες, αφού η λειτουργία τους δεν εξαρτάται από κάποιου είδους υποδομή (Garlan, 2002).

Σημασιολογία:

Οι ψηφίδες ως τα βασικά συστατικά των εφαρμογών περιρρέουσας νοημοσύνης, πρέπει να μπορούν να επικοινωνούν μεταξύ τους και για να είναι αυτό εφικτό πρέπει να

μπορεί να «καταλαβαίνει» η μία την άλλη με συνεπή και μη- διφορούμενο τρόπο.

Οι σχεδιαστικές απαιτήσεις που αναφέρθηκαν στην ενότητα αυτή αναφέρονται σε ένα σύστημα διάχυτου υπολογισμού γενικά, δηλαδή δεν εξαρτώνται από τις συγκεκριμένες εφαρμογές που το χρησιμοποιούν. Από την

άλλη, οι λειτουργικές απαιτήσεις που εκφράζονται ως υπηρεσίες, εργασίες ή λειτουργίες που πρέπει να φέρει εις πέρας το σύστημα, εξαρτώνται σε πολλές περιπτώσεις από τη συγκεκριμένη εφαρμογή που χρησιμοποιεί το σύστημα και από το δικτυακό περιβάλλον πάνω από το οποίο λειτουργεί.

2.1.2 Κατάταξη με βάση λειτουργικές απαιτήσεις

Στην προηγούμενη ενότητα αναφέρθηκαν οι βασικές σχεδιαστικές απαιτήσεις που πρέπει να αντιμετωπίσει μια πλατφόρμα για να καλύψει τις ανάγκες καταμεμημένων εφαρμογών που λειτουργούν σε περιβάλλοντα περιρρέουσας νοημοσύνης. Στην ενότητα αυτή θα γίνει μια επισκόπηση των κύριων λειτουργικών στοιχείων και υπηρεσιών που κατά πάσα πιθανότητα πρέπει να υλοποιούνται από ένα διαμεσολαβητικό λογισμικό. Οι λειτουργικές απαιτήσεις με τις οποίες ασχολείται η παρούσα εργασία παρουσιάζονται στον Πίνακα 1 .

ΠΙΝΑΚΑΣ 1 ΛΕΙΤΟΥΡΓΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ

Απαίτηση	Περιγραφή
-----------------	------------------

<p>Διαχείριση Υλικού (Hardware Management)</p>	<p>Αυτό το λειτουργικό στοιχείο αντιπροσωπεύει τα φυσικά στοιχεία μιας ψηφίδας του καταμεμημένου συστήματος π.χ. CPU, αισθητήρες και ενεργοποιητές, μονάδα δικτύου κτλ. και τον τρόπο με τον οποίο το διαμεσολαβητικό λογισμικό τα προσπελαύνει, τα διαχειρίζεται και τα επηρεάζει.</p>
<p>Τοπικό Λειτουργικό Σύστημα (Local OS)</p>	<p>Αναπαριστά τις λειτουργίες του τοπικού λειτουργικού συστήματος του κόμβου όπως διαχείριση διακοπών, τοπικός προγραμματισμός εργασιών, βασικές λειτουργίες εισόδου/εξόδου, επικοινωνία με αισθητήρες και ενεργοποιητές. Κάποιες από τις λειτουργίες του λειτουργικού συστήματος είναι πιθανό να χρειάζεται να διαχειρίζονται από το διαμεσολαβητικό λογισμικό.</p>
<p>Ειδοποίηση με γεγονότα (EventNotification)</p>	<p>Η δυνατότητα παράδοσης μηνυμάτων σε έναν ή πολλούς παραλήπτες. Τα συστήματα ειδοποίησης με γεγονότα λέγονται επίσης και συστήματα δημοσίευσης/εγγραφής (publish/subscribe). Σε νομαδικά δίκτυα η λειτουργικότητα της ειδοποίησης με γεγονότα φέρεται συνήθως εις πέρας από την σταθερή υποδομή, ενώ οι κινητές συσκευές απλώς λαμβάνουν τα γεγονότα. Σε δίκτυα χωρίς συγκεκριμένη υποδομή (ad hoc) οι λύσεις πρέπει να είναι πιο αποκεντρωμένες καθώς οι κόμβοι έχουν περίπου το ίδιο είδος ρόλων. Κάθε κόμβος έχει ένα λειτουργικό στοιχείο ειδοποίησης γεγονότων. Τυπικά τα συστήματα δημοσίευσης/εγγραφής περιέχουν παροχείς και καταναλωτές πληροφοριών. Οι παροχείς πληροφοριών δημοσιεύουν γεγονότα στους καταναλωτές πληροφοριών και αυτοί εγγράφονται για συγκεκριμένες κατηγορίες γεγονότων. Το διαμεσολαβητικό λογισμικό</p>

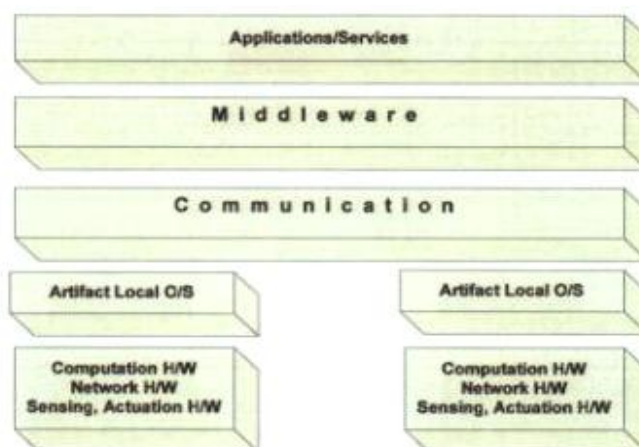
	<p>διασφαλίζει την παράδοση όλων των δημοσιευμένων γεγονότων σε όλους τους ενδιαφερόμενους καταναλωτές.</p>
--	---

<p>Κινητικότητα και επίγνωση θέσης (Mobility and locationawareness)</p>	<p>Η δυνατότητα περιπλάνησης από θέση σε θέση είναι ένα ουσιαστικό στοιχείο τέτοιων συστημάτων. Σε όρους του διαμεσολαβητικού λογισμικού, η κινητικότητα είναι η ικανότητα αναγνώρισης ενός νέου περιβάλλοντος και η προσαρμογή σε αυτό. Η κινητικότητα των κόμβων είναι πολύ συνηθισμένο φαινόμενο στα περιβάλλοντα περιρρέουσας νοημοσύνης, καθώς πολλοί από τους κόμβους που συνιστούν μια εφαρμογή μπορεί να είναι φορητές (portable) ή ακόμα και φορετές (wearable) συσκευές. Αυτό έχει ως αποτέλεσμα τη δυναμική αλλαγή της δομής του κατανεμημένου συστήματος πράγμα που μπορεί συχνά να οδηγήσει και στη διακοπή της διαθεσιμότητας ορισμένων υπηρεσιών όπως για παράδειγμα όταν ο κόμβος που προσφέρει μια συγκεκριμένη υπηρεσία βγει εκτός της εμβέλειας της εφαρμογής.</p>
<p>Αναζήτηση Περιβάλλοντος (Addressing)</p>	<p>Αυτό το λειτουργικό στοιχείο αναγνωρίζει διαφορετικές συσκευές που βρίσκονται στον περιβάλλοντα χώρο προκειμένου να μπορεί να επικοινωνήσει με αυτές.</p>
<p>Ανεύρεση υπηρεσιών (ServiceDiscovery)</p>	<p>Η ικανότητα αναγνώρισης νέων υπηρεσιών και πιθανώς πρωτοκόλλων. Αυτό είναι ένα σημαντικό λειτουργικό στοιχείο του διαμεσολαβητικού λογισμικού που επίσης σχετίζεται με την ικανότητα επανατοποθέτησης.</p>
<p>Ενημέρωση Κώδικα (CodeUpdate)</p>	<p>Η ικανότητα δυναμικής αλλαγής του πρωτοκόλλου και του κώδικα των εκτελούμενων εφαρμογών ή του ίδιου του διαμεσολαβητικού λογισμικού προκειμένου να μπορεί να προσαρμόζεται στο περιβάλλον.</p>

<p>Δρομολόγηση πολλών βημάτων (Multi-hopRouting)</p>	<p>Παρέχει επικοινωνία από σημείο-σε-σημείο και πολλαπλών σημείων με χρήση πολλών βημάτων (hops). Συγκεκριμένα στην περίπτωση επικοινωνίας πολλαπλών σημείων το σύνολο των δεκτών μπορεί να ταυτοποιηθεί από τον συγκεκριμένο θέμα του μηνύματος (multicast ή subject-basedrouting), από το περιεχόμενό του (content-basedrouting) ή από τη φυσική θέση του δέκτη (geocast).</p>
<p>Επίγνωση συγκεκριμένου πλαισίου (Contextawareness)</p>	<p>Η επίγνωση συγκεκριμένου πλαισίου αφορά τη δυνατότητα ενός συστήματος να γνωρίζει το συγκεκριμένο πλαίσιο στο οποίο λειτουργεί, δηλαδή πληροφορίες για το περιβάλλον του και για το νόημα που έχει το περιβάλλον υπό κάποιες συνθήκες.</p>
<p>Σχέδιο Δράσης (Reaction)</p>	<p>Σημαντικό ζήτημα είναι επίσης ο καθορισμός του τρόπου με τον οποίο οι ψηφίδες ενός συστήματος διάχυτου υπολογισμού αντιδρούν στα εξωτερικά ερεθίσματα, π.χ. από αισθητήρες, από άλλες ψηφίδες, δηλαδή η λογική τους.</p>

2.2 Αρχιτεκτονική για Συστήματα Διάχυτου Υπολογισμού

Με βάση τις προκλήσεις, τις σχεδιαστικές και τις λειτουργικές απαιτήσεις των συστημάτων διάχυτου υπολογισμού που παρουσιάστηκαν παραπάνω, προτείνεται η παρακάτω γενική αρχιτεκτονική. Στόχος της αρχιτεκτονικής αυτής είναι να αποτελέσει ένα γενικευμένο αρχιτεκτονικό πλαίσιο ανεξάρτητο από συγκεκριμένες τεχνολογίες και υλοποιήσεις, που να επιτρέπει τη σύνθεση υπηρεσιών και εφαρμογών περιρρέουσας νοημοσύνης, ολοκληρώνοντας ετερογενείς αρχιτεκτονικές υλικού/λογισμικού. Η αρχιτεκτονική φαίνεται στην Εικόνα 1 (Judd, 2003).



Εικόνα 1. Προτεινόμενη αρχιτεκτονική για συστήματα διάχυτου υπολογισμού

Εικόνα 1 Προτεινόμενη αρχιτεκτονική για συστήματα διάχυτου υπολογισμού

Η βασική ιδέα πίσω από την αρχιτεκτονική είναι ότι τα αντικείμενα που απαρτίζουν μια εφαρμογή περιρρέουσας νοημοσύνης είναι επαυξημένα με δυνατότητες υπολογισμού, επικοινωνίας και αίσθησης του περιβάλλοντος και ότι τα επαυξημένα αυτά αντικείμενα (τεχνουργήματα) χρησιμοποιούνται ως ψηφίδες (components) ενός τέτοιου συστήματος. Η αρχιτεκτονική διαχωρίζει τις λειτουργίες των ψηφίδων σε πέντε επίπεδα από το υλικό μέχρι το επίπεδο εφαρμογών. Κάθε επίπεδο αποτελεί μια εικονική μηχανή για το αμέσως ανώτερο του οι υπηρεσίες του οποίου παρέχονται μέσω καλώς ορισμένων API. Το μοντέλο αυτό, που μπορεί να αντιστοιχηθεί με το δικτυακό μοντέλο ISO/OSI, είναι αρκετά γενικό ώστε να μπορεί να συμπεριλάβει τρέχουσες προσεγγίσεις αρχιτεκτονικών κατακεταμμένων συστημάτων, όπως για παράδειγμα J2EE, JINI, CORBA, GAS κτλ. (Judd. 2003).

Επίπεδο 1: Υλικό

Το επίπεδο υλικού περιλαμβάνει έναν αριθμό από ετερογενή στοιχεία υλικού που καλύπτουν τις ανάγκες των επαυξημένων αντικειμένων τα οποία θα πρέπει να ενσωματωθούν σε αυτά φυσικά και λειτουργικά. Τέτοια στοιχεία είναι διάφοροι τύποι αισθητήρων και ενεργοποιητών (π.χ. αισθητήρες πίεσης, θερμοκρασίας, gps, λαμπτήρες, αναπαραγωγείς ήχου κτλ.), υπολογιστικές μονάδες (π.χ. PDAs, πλακέτες ειδικού σκοπού κτλ.) και μονάδες δικτύωσης (π.χ. ελεγκτές ethernet, bluetooth, IrDA,, WiFi κτλ.). Το επίπεδο αυτό μπορεί να περιλαμβάνει επίσης αρχιτεκτονικές ενσωματωμένων δικτύων αισθητήρων (sensornetworks) αποτελούμενων από μεγάλο αριθμό κόμβων, αρχιτεκτονικές υπολογιστικών μονάδων που να ενσωματώνουν δυνατότητες επεξεργασίας, αποθήκευσης και επικοινωνίας σε μικρές γενικού σκοπού πλακέτες, FPGAs, τεχνολογίες σμίκρυνσης υλικού κτλ.

Η επιλογή του υλικού και των αρχιτεκτονικών υλικού που θα χρησιμοποιηθεί εξαρτάται από την τρέχουσα τεχνολογία, από το είδος των εφαρμογών που πρέπει να υποστηριχθούν από το σύστημα και από τις επιλογές των κατασκευαστών των τεχνουργημάτων (Stanford, 2007).

Επίπεδο 2: Τοπικό λειτουργικό σύστημα τεχνουργημάτων

Τα τεχνουργήματα έχοντας μεγάλο αριθμό αισθητήρων και ενεργοποιητών, πρόσβαση σε υπολογιστικές δυνατότητες και ικανότητα επικοινωνίας με άλλα τεχνουργήματα, χρειάζονται ένα τοπικό λειτουργικό σύστημα που να διαχειρίζεται αυτούς τους πόρους και να υποστηρίζει τη σύνθεση και τη λειτουργική τους ολοκλήρωση. Το λειτουργικό σύστημα θα πρέπει να διεκπεραιώνει όλες τις εργασίες που χειρίζεται ένα παραδοσιακό λειτουργικό σύστημα (π.χ. διαχείριση μνήμης, διαχείριση διεργασιών, κτλ.) και

επιπλέον να ολοκληρώνει τα στοιχεία υλικού με διαφανή τρόπο και να μεταχειρίζεται τα περιρρέοντα τεχνουργήματα ως μέρος του συγκειμένου πλαισίου του. Επίσης θα πρέπει να είναι τμηματοποιημένο ώστε να διασφαλίζει ότι μόνο η απαραίτητη σε κάθε χρονική στιγμή λειτουργικότητα θα είναι ενεργή καθώς μπορεί να μην υπάρχει επάρκεια πόρων, να λαμβάνει αποφάσεις με βάση τους περιορισμούς των πόρων, να είναι σε θέση να εντοπίζει και να χρησιμοποιεί πόρους του περιβάλλοντος, και να εφαρμόζει πολιτικές ασφαλείας (Stanford, 2007).

Τα παραπάνω καθιστούν αναγκαία την ανάπτυξη εξειδικευμένων λειτουργικών συστημάτων για τεχνουργήματα μιας και προς το παρόν δεν υπάρχουν καθιερωμένες τεχνολογίες σε αυτό το χώρο. Παρ' όλα αυτά, η προτεινόμενη αρχιτεκτονική δεν απαγορεύει τη χρήση καθιερωμένων και γενικού σκοπού λειτουργικών συστημάτων (π.χ. windows, linux κτλ.). Τέτοιες λύσεις μπορούν να χρησιμοποιηθούν ως τοπικά λειτουργικά συστήματα τεχνουργημάτων αναλαμβάνοντας παραδοσιακές λειτουργίες, ενώ η επιπλέον λειτουργικότητα μπορεί να μεταφερθεί στο επίπεδο του διαμεσολαβητικού λογισμικού. Επιπλέον, ένα καθιερωμένο λειτουργικό σύστημα μπορεί να συνδυαστεί με έτοιμες λύσεις εικονικών μηχανών (π.χ. Java VM) ισοσταθμίζοντας έτσι το φόρτο καθηκόντων που διαμοιράζονται μεταξύ τοπικού λειτουργικού συστήματος και διαμεσολαβητικού λογισμικού (Stanford, 2007).

Επίπεδο 3: Επικοινωνία

Το επίπεδο της επικοινωνίας είναι υπεύθυνο για την επικοινωνία μεταξύ των κόμβων μιας εφαρμογής περιρρέουσας νοημοσύνης. Υλοποιεί αλγορίθμους και πρωτόκολλα για μετάδοση δεδομένων μεταξύ τεχνουργημάτων μέσω καναλιών επικοινωνίας που δημιουργούνται δυναμικά, όπως και μηχανισμούς για διακίνηση των πληροφοριών στο εσωτερικό του τεχνουργήματος. Ένα σύνολο κόμβων που ανταλλάσσουν πληροφορίες μέσω δυναμικών καναλιών συνιστούν ένα p2p δίκτυο του οποίου η τοπολογία αλλάζει δυναμικά (ad-hoc). Σε p2p

επικοινωνίες κάθε κόμβος αποτελεί μια αυτόνομη και «ομότιμη» (peer) με τους άλλους κόμβους δικτυωμένη μονάδα, και υλοποιεί ένα ή περισσότερα πρωτόκολλα δικτύου. Η ad-hoc φύση του δικτύου οφείλεται στη δυναμική αλλαγή της τοπολογίας του λόγω μη προβλέψιμων καταστάσεων που μπορεί να συμβούν λόγω της προσέλευσης ή αποχώρησης των τεχνουργημάτων από το δίκτυο για παράδειγμα λόγω κίνησης, δυσλειτουργίας κτλ. (Goumoroulas, 2007).

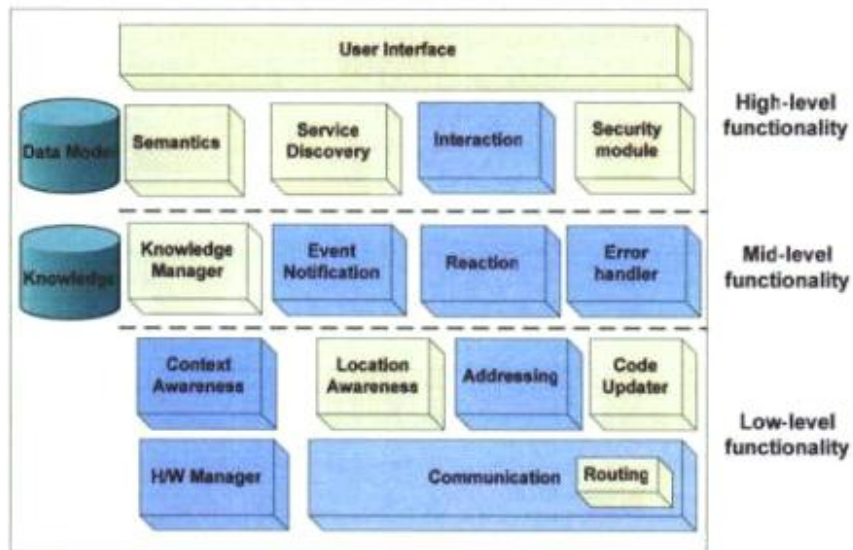
Η επικοινωνία αποτελεί το βασικό μέσο αλληλεπίδρασης μεταξύ τεχνουργημάτων και αν η αλληλεπίδραση είναι το κλειδί της αναδύμενης συμπεριφοράς τότε η υψηλής ποιότητας επικοινωνία είναι ο τρόπος για την επίτευξη αποδοτικών αλληλεπιδράσεων. Η ποιότητα της επικοινωνίας επηρεάζεται από έναν αριθμό από διαφορετικούς παράγοντες κάποιοι από τους οποίους είναι εγγενείς στο χρησιμοποιούμενο μέσο, ενώ άλλοι εξαρτώνται από την εκάστοτε υλοποίηση. Ο καλός σχεδιασμός του επιπέδου επικοινωνίας με γνώμονα την εξασφάλιση ποιότητας, μπορεί να αποτελέσει λύση σε πολλές από τις σχεδιαστικές απαιτήσεις των συστημάτων διάχυτου υπολογισμού. Έτσι ένα επίπεδο επικοινωνίας με δυνατότητα συνεργασίας με πολλές διαφορετικές καθιερωμένες τεχνολογίες (π.χ. Bluetooth, 802.11, IrDA κτλ.) μπορεί να ικανοποιήσει την απαίτηση της ετερογένειας αφού διαφορετικά τεχνουργήματα μπορεί να χρησιμοποιούν διαφορετικό δικτυακό υλικό και δικτυακό πρωτόκολλο. Η παροχή δε τεχνικών δυναμικής προσαρμογής στο εκάστοτε δικτυακό περιβάλλον μπορεί να ικανοποιήσει και την απαίτηση της προσαρμοστικότητας. Για παράδειγμα όταν μια κινητή συσκευή εισέρχεται σε δίκτυο που υποστηρίζει διαφορετικό πρωτόκολλο μεταφοράς (transportprotocol) από αυτό που προσφέρεται από το τρέχον δίκτυο που βρίσκεται η συσκευή, τότε θα πρέπει το σύστημα να επιλέξει αυτόματα κατάλληλο πρωτόκολλο που θα επιτρέψει την ομαλή λειτουργία της συσκευής στο νέο της περιβάλλον (Stanford, 2007).

Από το επίπεδο επικοινωνίας εξαρτάται σε μεγάλο βαθμό και η απαίτηση της κλιμάκωσης αφού μεγάλος αριθμός τεχνουργημάτων συνεπάγεται μεγάλο φόρτο επικοινωνίας και ανταλλαγής δεδομένων. Η χρήση λιτών μηνυμάτων και η έλλειψη σύγχρονης επικοινωνίας μπορεί να είναι δύο σχεδιαστικές αποφάσεις που αυξάνουν τη δυνατότητα κλιμάκωσης του

συστήματος. Ακόμα, το επίπεδο επικοινωνίας θα πρέπει να φροντίζει για την αξιοπιστία, τη σταθερότητα και την επίδοση της επικοινωνίας μεταξύ τεχνουργημάτων αντιμετωπίζοντας τα διάφορα σφάλματα που μπορεί να προκύψουν. Τέλος, πρωτόκολλα ασφαλείας μπορούν να προστεθούν για να ικανοποιήσουν την απαίτηση της ασφάλειας και να διασφαλίσουν την προστασία προσωπικών και ευαίσθητων δεδομένων των χρηστών των εφαρμογών (Goumopoulos, 2007).

Από λειτουργικής άποψης, το επίπεδο επικοινωνίας θα πρέπει να χρησιμοποιεί ειδοποιήσεις με γεγονότα (ασύγχρονη επικοινωνία). Σε δίκτυα χωρίς συγκεκριμένη υποδομή (ad hoc), όπως είναι τα δίκτυα των συστημάτων διάχυτου υπολογισμού, οι λύσεις πρέπει να είναι αποκεντρωμένες καθώς οι κόμβοι έχουν περίπου το ίδιο είδος ρόλων. Έτσι το επίπεδο επικοινωνίας κάθε κόμβου θα πρέπει να έχει και ένα λειτουργικό στοιχείο ειδοποίησης γεγονότων μέσω του οποίου θα ειδοποιεί το διαμεσολαβητικό λογισμικό για εισερχόμενες πληροφορίες από άλλα τεχνουργήματα, αλλά και θα ειδοποιείται από το διαμεσολαβητικό λογισμικό για εξερχόμενες πληροφορίες προς άλλα τεχνουργήματα. Επίσης, η κινητικότητα των κόμβων είναι πολύ συνηθισμένο φαινόμενο στα περιβάλλοντα περιρρέουσας νοημοσύνης, καθώς πολλοί από τους κόμβους που συνιστούν μια εφαρμογή μπορεί να είναι φορητές ή ακόμα και φορετές συσκευές. Αυτό έχει ως αποτέλεσμα τη δυναμική αλλαγή της δομής του κατανεμημένου συστήματος πράγμα που μπορεί συχνά να οδηγήσει και στη διακοπή της διαθεσιμότητας ορισμένων υπηρεσιών όπως για παράδειγμα όταν ο κόμβος που προσφέρει μια συγκεκριμένη υπηρεσία βγει εκτός της εμβέλειας της εφαρμογής. Το επίπεδο επικοινωνίας θα πρέπει να αντιμετωπίσει την κινητικότητα με τη χρήση κατάλληλων ασύρματων τεχνολογιών επικοινωνίας, αλλά και με τη χρήση πρωτοκόλλων δρομολόγησης που παρέχουν επικοινωνία από σημείο-σε-σημείο και πολλαπλών σημείων με χρήση πολλών βημάτων (hops) (Stanford, 2007).

Το επίπεδο επικοινωνίας (communication) σε σχέση με το διαμεσολαβητικό λογισμικό φαίνεται στην Εικόνα 2.



Εικόνα 2 Επίπεδο επικοινωνίας με το διαμεσολαβητικό λογισμικό

Επίπεδο 4: Διαμεσολαβητικό Λογισμικό

Διαμεσολαβητικό λογισμικό είναι το λογισμικό που βοηθάει μια εφαρμογή να αλληλοεπιδρά ή να επικοινωνεί με άλλες εφαρμογές, δίκτυα, υλικό και/ή λειτουργικά συστήματα. Ένα διαμεσολαβητικό λογισμικό μπορεί να χειρίζεται δυναμικά μια συλλογή από ετερογενείς ψηφίδες που εκτελούνται σε ετερογενή περιβάλλοντα. Μπορεί να θεωρηθεί ως ένα ψηφιδοκεντρικό λειτουργικό σύστημα όπου η διαχείριση όλων των ψηφίδων - τεχνουργήματα γίνεται από μια υποδομή που έχει επίγνωση των μεταξύ τους εξαρτήσεων, ενώ οι παρεχόμενοι μηχανισμοί χρησιμοποιούν αυτές τις εξαρτήσεις για να υποστηρίξουν τη δυναμική αναδιαμόρφωσή τους. Μέσω του διαμεσολαβητικού λογισμικού, τα τεχνουργήματα θα πρέπει να διαπραγματεύονται τη συνεργασία τους, να διαμοιράζονται πόρους ή να ανταλλάσσουν πληροφορίες, ενώ παράλληλα θα πρέπει να συμπεριφέρονται και εγωκεντρικά ώστε να διασφαλίζουν την αυτονομία και διατήρησή τους. Το διαμεσολαβητικό λογισμικό θα πρέπει επίσης να παρέχει μηχανισμούς για τη δυναμική διαχείριση των εξαρτήσεων μεταξύ των τεχνουργημάτων προκειμένου να διατηρεί τις εφαρμογές συνεπείς. Οι εξαρτήσεις αυτές μπορεί να είναι είτε στατικές (πόροι υλικού, υπηρεσίες λογισμικού κτλ. που

απαιτούνται από ένα τεχνούργημα για τη σωστή λειτουργία του) είτε δυναμικές για την υποστήριξη προσαρμογής σε χρόνο εκτέλεσης (Stanford, 2007).

Η γνώση σε ένα περιβάλλον περιρρέουσας νοημοσύνης μπορεί να είναι άφθονα διαθέσιμη (π.χ. internet, τοπικοί εξυπηρετητές) ενώ τα τεχνουργήματα θα διαθέτουν περιορισμένους πόρους. Όμως λόγω του ότι το περιβάλλον θα αλλάζει συνεχώς πρέπει να οριστούν κατώφλια μεταξύ τοπικής και ολικής αποθήκευσης της γνώσης. Στην προτεινόμενη αρχιτεκτονική η γνώση αναπαρίσταται με τη χρήση ενός μοντέλου δεδομένων που μπορεί να αναπαριστά σύνολο εννοιών ενός πεδίου, τις σχέσεις μεταξύ τους και να μπορεί να παράγει συλλογισμούς για τα αντικείμενα του πεδίου (π.χ. οντολογία). Για το λόγο αυτό το διαμεσολαβητικό λογισμικό πρέπει να ενσωματώνει μηχανισμούς για διαχείριση της γνώσης με βάση το μοντέλο δεδομένων τέτοιους ώστε να μπορούν να ενσωματωθούν ακόμα και σε τεχνουργήματα με μικρές υπολογιστικές δυνατότητες.

Η Εικόνα 2 παρουσιάζει τα κύρια λειτουργικά στοιχεία (modules) που μπορεί να υπάρχουν σε ένα διαμεσολαβητικό λογισμικό που λειτουργεί σε ένα περιβάλλον περιρρέουσας νοημοσύνης. Τα συστατικά στοιχεία του διαμεσολαβητικού λογισμικού χωρίζονται σε τρεις ζώνες: τη ζώνη χαμηλού επιπέδου λειτουργικότητας, μεσαίου και υψηλού. Η χαμηλού επιπέδου λειτουργικότητα αναφέρεται στα στοιχεία που βρίσκονται πιο κοντά στο υλικό και στις λειτουργίες του τοπικού λειτουργικού συστήματος, όπως για παράδειγμα η διαχείριση υλικού, η ανεύρεση περιβάλλοντος κτλ. Η μεσαίου επιπέδου λειτουργικότητα αναφέρεται σε στοιχεία που αποτελούν κεντρικά στοιχεία του διαμεσολαβητικού λογισμικού, όπως η ειδοποίηση με γεγονότα, ο χειρισμός σφαλμάτων και η λογική αντίδρασης στα γεγονότα. Τέλος η υψηλού επιπέδου λειτουργικότητα αναφέρεται σε στοιχεία που βρίσκονται πιο κοντά στην πλευρά του τελικού χρήστη και στην εφαρμογή, όπως το στοιχείο αλληλεπίδρασης, ασφάλειας, ανεύρεσης υπηρεσιών κτλ. (Stanford, 2007).

Ακολουθεί μια σύντομη περιγραφή κάθε στοιχείου της αρχιτεκτονικής του διαμεσολαβητικού λογισμικού. Τα στοιχεία που απεικονίζονται με γαλάζιο χρώμα θεωρούνται απαραίτητα για τη λειτουργία του διαμεσολαβητικού λογισμικού, ενώ τα άλλα όχι. Τα απαραίτητα στοιχεία θα πρέπει να μπορούν να εκτελούνται

ακόμα και σε συσκευές περιορισμένων πόρων, κάτι που επιβάλλει μια μινιμαλιστική σχεδίαση (π.χ. σχεδίαση μικροπυρήνων).

Λειτουργικότητα χαμηλού επιπέδου

- **Διαχειριστής Υλικού (H/W Manager):**

Διαχειρίζεται την επικοινωνία με το υλικό και κυρίως με τους αισθητήρες και ενεργοποιητές, μιας και η διαχείριση των υπολογιστικών μονάδων και των μονάδων δικτύωσης γίνεται από το τοπικό λειτουργικό σύστημα.

- **Επικοινωνία (communication):**

Είτε αποτελεί την επαφή συνεργασίας του διαμεσολαβητικού λογισμικού με το επίπεδο επικοινωνίας, είτε είναι το ίδιο το επίπεδο επικοινωνίας. Περιλαμβάνει το στοιχείο δρομολόγησης το οποίο αν υπάρχει δίνει στο τεχνούργημα δυνατότητες επικοινωνίας και εκτός της εμβελείας των ασυρμάτων τεχνολογιών.

- **Επίγνωση συγκεκριμένου πλαισίου (contextawareness):**

Το στοιχείο αυτό παρέχει στο σύστημα τη δυνατότητα να γνωρίζει το συγκεκριμένο πλαίσιο στο οποίο λειτουργεί, δηλαδή πληροφορίες για το περιβάλλον του και για το νόημα που έχει το περιβάλλον υπό κάποιες συνθήκες. Τις πληροφορίες αυτές τις λαμβάνει από τους αισθητήρες του τεχνουργήματος μέσω του διαχειριστή υλικού, ενώ για την ερμηνεία των πληροφοριών και την παραγωγή του συγκεκριμένου πλαισίου χρησιμοποιεί το μοντέλο δεδομένων. Η παραγόμενη μετα- γνώση αποθηκεύεται στη βάση

γνώσης και χρησιμοποιείται από το διαμεσολαβητικό λογισμικό για τη λήψη αποφάσεων, για καθορισμό της αντίδρασης του τεχνουργήματος, για αλληλεπίδραση με άλλα τεχνουργήματα κτλ. (Stanford, 2007).

- **Αναζήτηση περιβάλλοντος (Addressing):**

Μέσω αυτού του στοιχείου το διαμεσολαβητικό λογισμικό αναγνωρίζει τα τεχνουργήματα που βρίσκονται στον περιβάλλοντα χώρο προκειμένου να μπορεί να επικοινωνήσει και να συνεργαστεί μαζί τους. Η αναζήτηση περιβάλλοντος χρησιμοποιεί το επίπεδο επικοινωνίας για να αποστείλει το μήνυμα ανεύρεσης και για να λάβει τις απαντήσεις των τεχνουργημάτων που βρίσκονται κοντά. Μέσω του στοιχείου αυτού το διαμεσολαβητικό λογισμικό αναγνωρίζει τον κόσμο γύρω του πληροφορία που μπορεί να χρησιμοποιήσει στη συνέχεια για να διαπραγματευτεί συνεργασίες, να αλληλεπιδράσει, να ανταλλάξει γνώσεις κτλ.

- **Επίγνωση θέσης (Locationawareness):**

Η δυνατότητα περιπλάνησης από θέση σε θέση είναι ένα ουσιαστικό στοιχείο των συστημάτων διάχυτου υπολογισμού, κάτι που σε πολλές εφαρμογές δημιουργεί την ανάγκη επίγνωσης πληροφοριών σχετικά με τη σχετική ή την απόλυτη θέση των τεχνουργημάτων. Οι πληροφορίες αυτές γίνονται διαθέσιμες στο διαμεσολαβητικό λογισμικό μέσω αυτού του στοιχείου. Η απόκτηση τέτοιου είδους πληροφοριών μπορεί να γίνει είτε με χρήση εξειδικευμένων αισθητήρων (π.χ. gps, ultrasonic) οι οποίες περνάν στο στοιχείο επίγνωσης θέσης μέσω του διαχειριστή υλικού, είτε με τη χρήση του επιπέδου επικοινωνίας το οποίο μέσω υπολογισμού της ισχύος του σήματος επικοινωνίας (μόνο για ασύρματη επικοινωνία) μπορεί να υπολογίσει τη σχετική θέση πομπού και δέκτη (Stanford, 2007).

- **Τροποποιητής κώδικα (Codeupdater):**

Το στοιχείο αυτό προσδίδει την ικανότητα δυναμικής αλλαγής του πρωτοκόλλου και του κώδικα των εκτελούμενων εφαρμογών ή του ίδιου του διαμεσολαβητικού λογισμικού προκειμένου να μπορεί να προσαρμόζεται στο περιβάλλον. Το στοιχείο μπορεί να χρησιμοποιηθεί επίσης για μεταφορά κώδικα μεταξύ τεχνουργημάτων η οποία γίνεται μέσω του επιπέδου επικοινωνίας.

Λειτουργικότητα μέσου επιπέδου

- **Ειδοποίηση με γεγονότα (Eventnotification):**

Κάθε κόμβος έχει ένα στοιχείο ειδοποίησης γεγονότων το οποίο φροντίζει για την ασύγχρονη μετάδοση πληροφοριών μεταξύ των στοιχείων του διαμεσολαβητικού λογισμικού. Ανά πάσα στιγμή κάποιο στοιχείο είναι παροχέας πληροφοριών και κάποιο ο αποδέκτης. Οι παροχείς δημοσιεύουν γεγονότα στους αποδέκτες και οι αποδέκτες εγγράφονται για συγκεκριμένες κατηγορίες γεγονότων. Το στοιχείο αυτό διασφαλίζει την παράδοση όλων των δημοσιευμένων γεγονότων σε όλους τους ενδιαφερομένους αποδέκτες (Stanford, 2007).

- **Διαχειριστής γνώσης (KnowledgeManager):**

Το στοιχείο αυτό αποτελεί την πύλη πρόσβασης στη γνώση του τεχνουργήματος. Όποιο άλλο στοιχείο χρειάζεται να αλληλοεπιδράσει με τη γνώση, πρέπει να αλληλοεπιδράσει με το διαχειριστή γνώσης. Στην προτεινόμενη αρχιτεκτονική η γνώση αναπαρίσταται με τη χρήση ενός μοντέλου δεδομένων (Goumopoulos, 2007).

- **Σχέδιο δράσης (Reaction):**

Το σχέδιο δράσης αποτελεί τη λογική των τεχνουργημάτων, τον τρόπο δηλαδή με τον οποίο τα τεχνουργήματα αντιδρούν στα εξωτερικά ερεθίσματα, είτε αυτά προέρχονται από τους αισθητήρες τους, είτε από άλλα τεχνουργήματα. Το στοιχείο αυτό καθορίζει αν η αντίδραση θα είναι από μια απλή αυτοματοποιημένη δράση, ή το αποτέλεσμα μιας διαδικασίας λήψης αποφάσεων με βάση κανόνες. Ανάλογα με την πολυπλοκότητα του σχεδίου δράσης, το στοιχείο αυτό μπορεί να επικοινωνήσει με το διαχειριστή γνώσης και με το στοιχείο σημασιολογίας και να δημιουργήσει από απλούς έως πολύπλοκους κανόνες εκφρασμένους με όρους του μοντέλου δεδομένων.

- **Χειριστής σφαλμάτων (Error-handler):**

Το στοιχείο αυτό ενεργοποιείται κάθε φορά που συμβαίνει μια μη-ομαλή λειτουργία στο διαμεσολαβητικό λογισμικό. Ο ρόλος του είναι ανάλογα με τον τύπο της δυσλειτουργίας να αποφασίσει την κατάλληλη ενέργεια έτσι ώστε το σύστημα συνολικά να παραμείνει σταθερό και να συνεχίσει να παρέχει τις υπηρεσίες που προσέφερε. Για παράδειγμα αν χαθεί η επικοινωνία με ένα τεχνούργημα που παρείχε μια υπηρεσία, τότε ο χειριστής σφαλμάτων θα πρέπει να επικοινωνήσει με το στοιχείο ανεύρεσης υπηρεσιών ώστε να εντοπίσει ένα άλλο τεχνούργημα που να προσφέρει την ίδια ή παρόμοια υπηρεσία και να την αντικαταστήσει. (OZONE,2007)

Λειτουργικότητα υψηλού επιπέδου

- **Σημασιολογία (Semantics):**

Το στοιχείο αυτό παρέχει κοινή γλώσσα επικοινωνίας μεταξύ των τεχνουργημάτων έτσι ώστε να μπορεί να «καταλαβαίνει» το ένα το άλλο με συνεπή και μη-διφορούμενο τρόπο. Αυτό γίνεται με χρήση του μοντέλου δεδομένων, δηλαδή ουσιαστικά το στοιχείο αυτό δεν είναι άλλο παρά μια πύλη προς το μοντέλο δεδομένων.

- **Ανεύρεση υπηρεσιών(ServiceDiscovery):**

Ο ρόλος του στοιχείου αυτού είναι η αναζήτηση στο περιβάλλον τεχνουργημάτων που να παρέχουν μια ζητούμενη υπηρεσία. Η διαδικασία της ανεύρεσης χρησιμοποιεί το στοιχείο της σημασιολογίας για να περιγράψει τη ζητούμενη υπηρεσία στην κοινή γλώσσα και το επίπεδο επικοινωνίας για να μεταδοίσει την αναζήτηση στο περιβάλλον μέσω του δικτύου (Goumoroulas, 2007).

- **Αλληλεπίδραση (Interaction):**

Το στοιχείο αυτό αποτελεί μια πύλη προς τις δυνατότητες και τις υπηρεσίες του τεχνουργήματος για τους τελικούς χρήστες και για τα άλλα τεχνουργήματα. Μέσω του στοιχείου αυτού ένα τεχνουργήματα μπορεί να συνδεθεί λογικά με ένα άλλο και να ανταλλάξει πληροφορίες. Το στοιχείο της αλληλεπίδρασης επικοινωνεί με το στοιχείο σημασιολογίας για να μπορέσει να «καταλάβει» τη γλώσσα του απομακρυσμένου τεχνουργήματος, τροφοδοτεί το στοιχείο σχεδίου δράσης με πληροφορίες του απομακρυσμένου τεχνουργήματος έτσι ώστε να πυροδοτήσει μιας

αντίδραση, και τέλος επικοινωνεί με το επίπεδο επικοινωνίας για να υλοποιήσει φυσικά τη λογική επικοινωνία μεταξύ δύο τεχνουργημάτων.

- **Ασφάλεια (Security):**

Το στοιχείο αυτό φροντίζει για την τήρηση πολιτικών ασφαλείας κατά την επικοινωνία με άλλα τεχνουργήματα. Οι πολιτικές ασφαλείας μπορεί να αφορούν θέματα εξουσιοδότησης, ταυτοποίησης κτλ.

- **Διεπαφή χρήστη (User Interface):**

Το στοιχείο αυτό είναι ο ενδιάμεσος μεταξύ των τελικών χρηστών και του διαμεσολαβητικού λογισμικού. Μέσω της διεπαφής χρήστη, οι χρήστες έχουν πρόσβαση στις υπηρεσίες και τη λειτουργικότητα του τεχνουργήματος με εύκολο και φιλικό τρόπο. Πιθανοί τρόποι αλληλεπίδρασης των χρηστών με το διαμεσολαβητικό λογισμικό μέσω της διεπαφής χρήστη είναι: φωνή, χειρονομίες, κίνηση, οπτική, επαυξημένη πραγματικότητα (augmentedreality) κτλ. (Goumopoulos, 2007).

Επίπεδο 5: Εφαρμογές - Υπηρεσίες

Το τελευταίο επίπεδο της αρχιτεκτονικής αφορά την παροχή μοντέλων αλληλεπίδρασης και κανόνων που θα χρησιμοποιούν οι χρήστες για να συνθέτουν και να χρησιμοποιούν εφαρμογές περιρρέουσας νοημοσύνης. Η εκπλήρωση της προοπτικής του διάχυτου υπολογισμού προϋποθέτει οι χρήστες να μπορούν να τιθασεύσουν τις δυνατότητες που παρέχονται από την τεχνολογία. Αυτό θέτει σχεδιαστικά προβλήματα διαφορετικά από αυτά που παρουσιάζονται στους συμβατικούς υπολογιστές. Επιπλέον ο διάχυτος υπολογισμός διευρύνει τη χρήση των υπολογιστών από ένα σχετικά

περιορισμένο σύνολο συγκειμένων πλαισίων σε ένα σαφώς μεγαλύτερο - σχεδόν οπουδήποτε και οποτεδήποτε. Αυτό καθιστά ακόμα δυσκολότερο το ρόλο των σχεδιαστών συστημάτων, οι οποίοι πρέπει να προβλέπουν τον τρόπο με τον οποίο οι άνθρωποι θα χρησιμοποιήσουν την τεχνολογία.

Για το λόγο αυτό απαιτείται μια νέα προσέγγιση δημιουργίας συστημάτων. Αυτή η προσέγγιση επαναπροσδιορίζει το ρόλο των προγραμματιστών, οι οποίοι θα πρέπει πλέον να δημιουργούν συστήματα λογισμικού που να επιτρέπουν στους χρήστες να προγραμματίζουν σε υψηλό επίπεδο (end-user programming) και να μπορούν να μοντελοποιούν τα προβλήματα για τα οποία αυτοί είναι ειδικοί. Οι σχεδιαστές δε χρειάζεται να μαντεύουν τις επιθυμίες των χρηστών, καθώς οι ίδιοι οι χρήστες μπορούν να διαμορφώσουν τη συμπεριφορά των συστημάτων όπως επιθυμούν. Παρ' όλα αυτά, η παραγωγή ενός κατάλληλου συστήματος προγραμματισμού για τελικούς χρήστες, δεν είναι καθόλου εύκολη υπόθεση στο χώρο της περιρρέουσας νοημοσύνης. Εκτός από τα προβλήματα διεπαφής χρήστη, τα περισσότερα συστήματα διάχυτου υπολογισμού απευθύνονται σε ευρεία κοινωνία χρηστών, κάτι που καθιστά πολύ δύσκολο το σχεδιασμό ενός μοντέλου που να ικανοποιεί τις ανάγκες όλων (OZONE,2007).

Μια αρχιτεκτονική που θα υποστήριζε την ενσωμάτωση πολλών μοντέλων θα επέτρεπε την παροχή προγραμματιστικών ευκολιών για τελικούς χρήστες για την ικανοποίηση των αναγκών διαφορετικών ομάδων χρηστών. Επιπλέον, ένα σύστημα βασισμένο στην αρχιτεκτονική αυτή θα μπορούσε να επιτρέψει τη χρήση διαφορετικών μοντέλων για διαφορετικές εργασίες.

Στόχος του επιπέδου εφαρμογών/υπηρεσιών της προτεινόμενης αρχιτεκτονικής, είναι να είναι αρκετά γενικό ώστε να μπορεί να υποστηρίξει διαφορετικά μοντέλα προγραμματισμού εφαρμογών. Τέτοια μοντέλα είναι το παράδειγμα του παζλ (jigsawparadigm) όπου ο χρήστης καθορίζει τις συνεργασίες μεταξύ των τεχνουργημάτων σα να συνδέει κομμάτια παζλ, το παράδειγμα του προγραμματισμού με παραδείγματα (programming-by-example) όπου ο χρήστης μέσα από τον τρόπο που χρησιμοποιεί το περιβάλλον υποδεικνύει στο σύστημα το πώς πρέπει να προγραμματιστεί (π.χ. έξυπνοι πράκτορες), μοντέλα publish-subscribe, μοντέλα βασισμένα σε φόρμες, πίνακες

και διαγράμματα ροής κτλ. Μια ενδιαφέρουσα περιοχή έρευνας είναι η τεχνική του προγραμματισμού με οπτική μοντελοποίηση της πληροφορίας που εμπλέκει τη χρήση τεχνολογιών Semantic Web. Σύμφωνα με το μοντέλο αυτό, απλοί χρήστες μπορούν να μοντελοποιήσουν πολύπλοκα προβλήματα χωρίς τη χρήση γλωσσών προγραμματισμού, διαχειρίζοντας πληροφορίες σε οπτικοποιημένες δένδρικές δομές με τη βοήθεια οντολογιών, ενώ το αποτέλεσμα της διαχείρισης προβάλλεται στον παγκόσμιο ιστό (web). Ένα σχετικό πεδίο έρευνας είναι αυτό των τεχνικών Semantic Web και Web 2.0 που επιτρέπουν την OnLine αλληλεπίδραση με την οπτική αναπαράσταση της πληροφορίας. Στόχος είναι ο προγραμματισμός από τελικούς χρήστες, επιτρέποντάς τους να βλέπουν πάντα το συγκείμενο πλαίσιο της πληροφορίας που διαχειρίζονται και να λαμβάνουν άμεση ανταπόκριση πάνω σε αλλαγές που εκτέλεσαν (OZONE,2007).

Η υποστήριξη πολλαπλών μοντέλων επιτυγχάνεται από την καθαρά ψηφιοκεντρική αρχιτεκτονική του διαμεσολαβητικού λογισμικού, που επιτρέπει τη χρήση διαφορετικών ψηφίδων για κάθε στοιχείο του και από την πληρότητα λειτουργιών που προδιαγράφει η αρχιτεκτονική. Οι επιλογές που θα κάνει ο σχεδιαστής ενός συστήματος για τα διάφορα λειτουργικά στοιχεία της αρχιτεκτονικής σε συνδυασμό με τον τρόπο που θα παρουσιάσει τις υπηρεσίες του συστήματος στον χρήστη (διεπαφή χρήστη) μπορούν να οδηγήσουν στην υλοποίηση ευρέως συνόλου προγραμματιστικών μοντέλων. Επιλέγοντας για παράδειγμα οντολογίες [Christoroulou, 2005] ως μοντέλο δεδομένων (στοιχείο datamodel), δημιουργώντας μια συγκεκριμένη οντολογία για το επιθυμητό προγραμματιστικό μοντέλο (π.χ. ορισμός συνδέσμων, συνάψεων κτλ.), ορίζοντας τις πολιτικές αλληλεπίδρασης (π.χ. περιορισμούς σύνδεσης δύο συνδέσμων στο στοιχείο Interaction), την Jess [Jess, 2007] ως διαχειριστή γνώσης (στοιχείο KnowledgeManager), και συνδέοντας τα παραπάνω με ένα επίπεδο επικοινωνίας βασισμένο σε sockets και ανταλλαγή μηνυμάτων [Goumopoulos 2003], μπορούμε να δημιουργήσουμε το μοντέλο Συνδέσμων/Συνάψεων του GAS-OS [Kameas, 2003], [Drossos, 2006]. Επιλέγοντας μια βάση δεδομένων ψηφίδων ως μοντέλο δεδομένων (π.χ. αποθήκη ψηφίδων του GAIA), την εμβέλεια ασύρματης

επικοινωνίας και την κλήση απομακρυσμένων μεθόδων ως τρόπο αλληλεπίδρασης (ψηφίδα Interaction), και την ενημέρωση κώδικα για φόρτωση των κατάλληλων ψηφίδων στους κόμβους του συστήματος, μπορούμε να δημιουργήσουμε το μοντέλο των Ενεργών Χώρων του GAIA (Roman, 2002)

Τέλος, στο επίπεδο εφαρμογών/υπηρεσιών μπορούν να ενσωματωθούν μεταβλητές διεπαφές χρήστη (π.χ. φυσική γλώσσα, απόσπαση προσοχής), που να προσαρμόζονται στο συγκεκριμένο πλαίσιο χρήσης (π.χ. πόροι μιας συσκευής χειρός, πολυεκφραστικές δραστηριότητες, ανάδραση, κτλ.). Από την πλευρά της διεπαφής χρήστη ο προγραμματισμός και η διαμόρφωση των τεχνουργημάτων είναι συχνά προβληματικά, ιδιαίτερα για τεχνουργήματα που δεν έχουν συσκευές εισόδου ή εξόδου. Έτσι είναι σημαντική η επινόηση μεθόδων προγραμματισμού και διαμόρφωσης των τεχνουργημάτων με χρήση φυσικών δραστηριοτήτων όπως χειρονομίες και ήχοι, αλλά και η παροχή των κατάλληλων εργαλείων χρήστη (Roman, 2002).

Κεφάλαιο 3

Συστήματα Περιρρέουσας Νοημοσύνης

Ένα περιβάλλον Περιρρέουσας Νοημοσύνης έχει τα εξής βασικά χαρακτηριστικά (Aarts-Harwigetal., 2001) :

- Εμφύτευση (Embedded): Πολλές συσκευές είναι διασκορπισμένες στο περιβάλλον χωρίς να φαίνονται.
- Επίγνωση περιεχομένου(Contextaware): Το περιβάλλον έχει τη δυνατότητα να αναγνωρίζει το εκάστοτε πρόσωπο και την κατάστασή του.

- Διάχυση(pervasive): Η αλληλεπίδραση με το σύστημα δεν γίνεται μόνο από ένα σταθμό εργασίας, αλλά από επιμέρους συσκευές.
- Εξατομίκευση(Personalized): Το περιβάλλον μπορεί να συμπεριφέρεται ανάλογα με το ποιός χρήστης εμφανίζεται κάθε φορά.
- Προσαρμοστικότητα(Adaptive): Το περιβάλλον αλλάζει ανάλογα με τις ανάγκες του κάθε χρήστη.
- Προνοητικότητα(Anticipatory): Το περιβάλλον έχει τη δυνατότητα να προνοεί σχετικά με τις επιθυμίες κάθε χρήστη.

Για να γίνει πραγματικότητα ένα περιβάλλον Περιρρέουσας Νοημοσύνης είναι απαραίτητες οι παρακάτω τεχνολογίες:

- Τεχνολογίες πρόσβασης (κινητά τηλέφωνα, Laptops, PDAs, κ.α)
- Τεχνολογίες πρόσληψης και αντίληψης του φυσικού περιβάλλοντος (αισθητήρες, smartdust,RFIDs, κ.α)
- Τεχνολογίες επικοινωνίας(ασύρματα δίκτυα κ.α)
- Τεχνολογίες ασφάλειας δικτύων (firewalls,ψηφιακές υπογραφές, κ.α)

Τα συστατικά των εφαρμογών περιρρέουσας νοημοσύνης ονομάζονται τεχνουργήματα. Συνδυάζοντας με κατάλληλο τρόπο τα συστατικά αυτά είναι εφικτή η υλοποίηση ενός αρκετά μεγάλου εύρους εφαρμογών περιρρέουσας νοημοσύνης όπως εφαρμογές σπιτιού, γραφείου, εξωτερικών χώρων κτλ. Στην

ενότητα αυτή θα εξεταστεί η φυσική υπόσταση και η δομή των τεχνουργημάτων, θα περιγράψει το υλικό και το λογισμικό που χρησιμοποιήθηκε για την κατασκευή πρωτότυπων τεχνουργημάτων και τέλος θα παρουσιαστούν μερικά πρωτότυπα τεχνουργήματα, καθώς επίσης και διάφορα συστήματα για τον περιβάλλον της Περιρρέουσας Νοημοσύνης (Roman, 2002).

3.1 Φυσική υπόσταση των τεχνουργημάτων

Κάθε αντικείμενο του χώρου που μας περιβάλλει έχει σχεδιαστεί για κάποιες συγκεκριμένες εργασίες. Για παράδειγμα ένα ποτήρι χρησιμοποιείται για να συγκρατεί κάποια ποσότητα υγρού. Οι τρόποι που μπορούμε να χρησιμοποιήσουμε ένα συνηθισμένο αντικείμενο εξαρτώνται από τις ιδιότητες που του έχουν δοθεί από τον σχεδιαστή του. Αυτή η σχέση είναι ουσιαστικά αμφίδρομη: από τη μία τα αντικείμενα έχουν σχεδιαστεί να είναι κατάλληλα για ορισμένες εργασίες, από την άλλη οι φυσικές τους ιδιότητες περιορίζουν τις εργασίες για τις οποίες τα χρησιμοποιούν οι άνθρωποι. Παρόλα αυτά, πολλές φορές οι άνθρωποι μπορεί να χρησιμοποιήσουν αντικείμενα ακόμα και με μη αναμενόμενο τρόπο: ένα ποτήρι για παράδειγμα εκτός από να το γεμίσουμε με κάποια ποσότητα υγρού όπως αναφέρθηκε πιο πάνω, μπορεί να χρησιμοποιηθεί και ως "βαρίδιο" για τον αέρα σε χαρτιά. Γενικά, τα καθημερινά αντικείμενα μπορούν να χρησιμοποιηθούν με πολλούς διαφορετικούς τρόπους, αρκεί να μην παραβιάζονται οι φυσικές τους ιδιότητες και να μην αλλοιώνεται η φυσική τους υπόσταση (Goumoroulas, 2007).

Οι υπολογιστές είναι επίσης ξεχωριστά αντικείμενα που έχουν τις δικές τους φυσικές ιδιότητες και χρήσεις. Όταν ο υπολογιστής ενσωματώνεται στο περιβάλλον, δεν θεωρείται πλέον διακριτό αντικείμενο που μπορεί κάποιος να χρησιμοποιήσει για να φέρει εις πέρας τις εργασίες του. Επιπλέον, καθώς τα καθημερινά αντικείμενα επαυξάνονται με υπολογιστικές και επικοινωνιακές δυνατότητες, οι χρήστες πρέπει να μάθουν τους νέους τρόπους χρήσης τους (που υποδεικνύονται από τη σχεδίαση νέων δυνατοτήτων καταληπτότητας

χρήσης) και τις εργασίες στις οποίες αυτά μπορούν να συμμετέχουν. Συνεπώς ο χώρος της περιρρέουσας νοημοσύνης εισάγει νέες προκλήσεις για την αλληλεπίδραση ανθρώπου-υπολογιστή. Πρωταρχικά οι χρήστες πρέπει να ενημερώσουν τα μοντέλα εργασίας τους, αφού πλέον δε θα αλληλεπιδρούν με απλά αντικείμενα αλλά με επαυξημένα, ενώ θα πρέπει επίσης να αλλάξουν τις συνήθειες τους και να δημιουργήσουν νέα μοντέλα για τα επαυξημένα αντικείμενα που χρησιμοποιούν (Goumououlas, 2007).

Με τον τρόπο αυτό, η αλληλεπίδραση ανθρώπου-υπολογιστή ξεπερνάει τα όρια του υπολογιστή ,αφού πλέον οι υπολογιστικές εφαρμογές είναι καταναμημένες σε αντικείμενα που μας περιβάλλουν, και εισέρχεται στον φυσικό κόσμο. Αυτή η αλληλεπίδραση επηρεάζεται επίσης και από τη σχεδίαση της μορφής των αντικειμένων και των φυσικών του ιδιοτήτων. Στην ουσία, η σχεδίαση των αντικειμένων που αποτελεί και την διεπαφή τους, πρέπει να επαναπροσδιοριστεί έτσι ώστε οι νέες τους δυνατότητες να μπορούν να μεταδοθούν στο χρήστη (μέσω κατάλληλα σχεδιασμένων δυνατοτήτων καταληπτότητας χρήσης). Ένα ακόμη αποτέλεσμα της ενσωμάτωσης του υπολογιστή στα καθημερινά αντικείμενα είναι ότι τα εννοιολογικά μοντέλα που κατέχουν οι άνθρωποι για αυτά τα αντικείμενα θα πρέπει να εξελιχθούν.

3.2 Δομή των τεχνουργημάτων

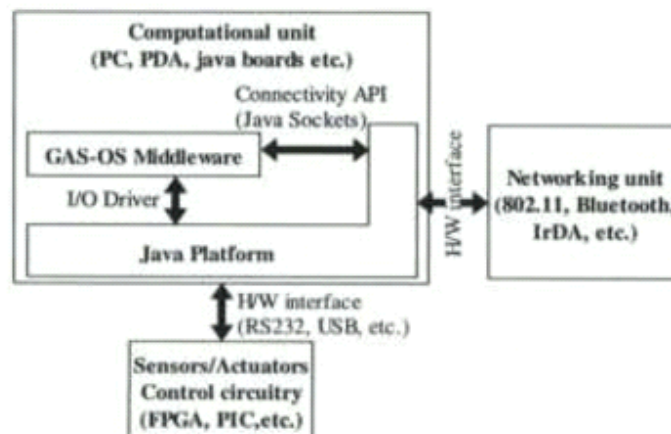
Ένα τεχνούργημα αποτελείται από:

- ένα σύνολο αισθητήρων και ενεργοποιητών μαζί με τα απαραίτητα κυκλώματα ελέγχου (π.χ. FPGA, μικροελεγκτές PIC, κτλ.). Η σύνδεση των αισθητήρων και ενεργοποιητών γίνεται συνήθως με απλά καλώδια που μεταφέρουν τα σήματά τους στα κυκλώματα ελέγχου, τα οποία χρησιμοποιούν κατάλληλα πρωτόκολλα (π.χ. σειριακά, παράλληλα) για την ομαδοποίηση και μετάδοση της πληροφορίας στην υπολογιστική μονάδα. Εναλλακτικά, η μετάδοση των δεδομένων από και προς τους

αισθητήρες/ενεργοποιητές μπορεί να γίνει μέσω Ethernet μέσω κατάλληλων κυκλωμάτων και υποστήριξη του πρωτοκόλλου Microsoft UniversalPlug and Play (UPNP). Το τελευταίο πρέπει να συνεργάζεται με κατάλληλο λογισμικό ελέγχου UPNP το οποίο όπως θα δούμε αργότερα μπορεί να ενσωματώνεται στο μηχανισμό οδηγών του GAS-OS.

- μια υπολογιστική μονάδα η οποία μπορεί να φιλοξενήσει την πλατφόρμα της Java. Ως υπολογιστικές μονάδες μπορούν να χρησιμοποιηθούν είτε προσωπικοί υπολογιστές (PC), είτε υπολογιστές παλάμης (PDAs) είτε κινητά τηλέφωνα, είτε οποιεσδήποτε υπολογιστικές πλακέτες με υποστήριξη Java,
- μια μονάδα ενσύρματης ή ασύρματης δικτύωσης (802.11, bluetooth, IrDA κτλ.)
- σε μερικές περιπτώσεις οθόνη ή πλήκτρα ελέγχου

Στην παρακάτω εικόνα φαίνεται η δομή ενός τεχνουργήματος:



Εικόνα 3 Δομή ενός τεχνουργήματος

Το σύνολο των αισθητήρων και ενεργοποιητών μαζί με τα κυκλώματα ελέγχου είναι υπεύθυνα για τη μετατροπή των δεδομένων του περιβάλλοντος του τεχνουργήματος (π.χ. πίεση, φωτεινότητα κτλ.) σε ψηφιακά και το αντίστροφο. Στην περίπτωση που το τεχνούργημα είναι ηλεκτρονική συσκευή, κάποια από τα απαιτούμενα κυκλώματα είναι συνήθως ενσωματωμένα στη συσκευή. Και στις δύο περιπτώσεις, τα ψηφιακά δεδομένα εξάγονταν την υπολογιστική μονάδα με χρήση κατάλληλων διεπαφών υλικού (π.χ. RS232, USB κτλ.). Η υπολογιστική μονάδα εκτελεί το απαιτούμενο λογισμικό και με βάση τα δεδομένα που λαμβάνει από τους αισθητήρες και από άλλα τεχνουργήματα μέσω της μονάδας δικτύου, εκτελεί έναν αριθμό από ενέργειες, το αποτέλεσμα των οποίων εκδηλώνεται είτε στο περιβάλλον, μέσω των ενεργοποιητών, είτε σε άλλα τεχνουργήματα, μέσω του δικτύου που τα συνδέει.

Τέλος η μονάδα δικτύου είναι μια ενσύρματη ή ασύρματη μονάδα (π.χ. 802.11, Bluetooth, κτλ.) που αναλαμβάνει την φυσική επικοινωνία με άλλα τεχνουργήματα ή και εργαλεία χρηστών.

Στις περισσότερες περιπτώσεις το υλικό του τεχνουργήματος είναι ενσωματωμένο στο τεχνούργημα (π.χ. Εικόνα 8), ενώ του ανατίθεται ένα μοναδικό αναγνωριστικό (ID), π.χ. ένας σειριακός αριθμός, από τον κατασκευαστή του, κάτι που εγγυάται τη μοναδικότητά του. Παρ' όλα αυτά, καθώς η τεχνολογία ωριμάζει ολοένα και περισσότερα τεχνουργήματα θα είναι «εικονικά», υπό την έννοια ότι θα συντίθενται από καταναμημένα αντικείμενα με γνώμονα τις περιγραφές των υπηρεσιών τους.

Η Εικόνα 3 εκτός από το υλικό του τεχνουργήματος, δείχνει και το απαιτούμενο λογισμικό που χρειάζεται για να μπορέσει ένα τεχνούργημα να αποκτήσει ψηφιακή αναπαράσταση. Το διαμεσολαβητικό λογισμικό GAS-OS εκτελείται στην υπολογιστική μονάδα πάνω από την πλατφόρμα της Java. Το GAS-OS αποτελεί το κεντρικό σημείο ελέγχου ολόκληρου του τεχνουργήματος αφού αυτό διαχειρίζεται και συντονίζει όλες τις περιφερειακές μονάδες. Ο πυρήνας του GAS-OS υλοποιεί τις έννοιες του εννοιολογικού μοντέλου Συνδέσμων/Συνάψεων και τους μηχανισμούς για την υποστήριξη της σύνθεσης

εφαρμογών περιρρέουσας νοημοσύνης. Η επικοινωνία του διαμεσολαβητικού λογισμικού με τους αισθητήρες/ενεργοποιητές του τεχνουργήματος, γίνεται με τη χρήση οδηγών εισόδου/εξόδου (I/O drivers). Παρόλο που τα ψηφιακά δεδομένα λαμβάνονται από το υλικό με ομοιόμορφο τρόπο με χρήση της Java πλατφόρμας, συνήθως χρειάζεται να αναλυθούν και να διαχειριστούν με διαφορετικό τρόπο για κάθε τεχνούργημα, πράγμα που καθιστά απαραίτητη την παρουσία ενός οδηγού GAS-OS εισόδου-εξόδου. Ένας οδηγός του GAS-OS είναι ένα σύνολο από ρουτίνες που συνδέονται με τον πυρήνα του, και χρησιμοποιούνται σαν μέρος του μηχανισμού λειτουργίας μιας συγκεκριμένης μονάδας υλικού. Ο διαχωρισμός των αρμοδιοτήτων του οδηγού από το διαμεσολαβητικό λογισμικό διευκολύνει την προσθήκη νέων μονάδων υλικού χωρίς αλλαγές στο σύστημα.

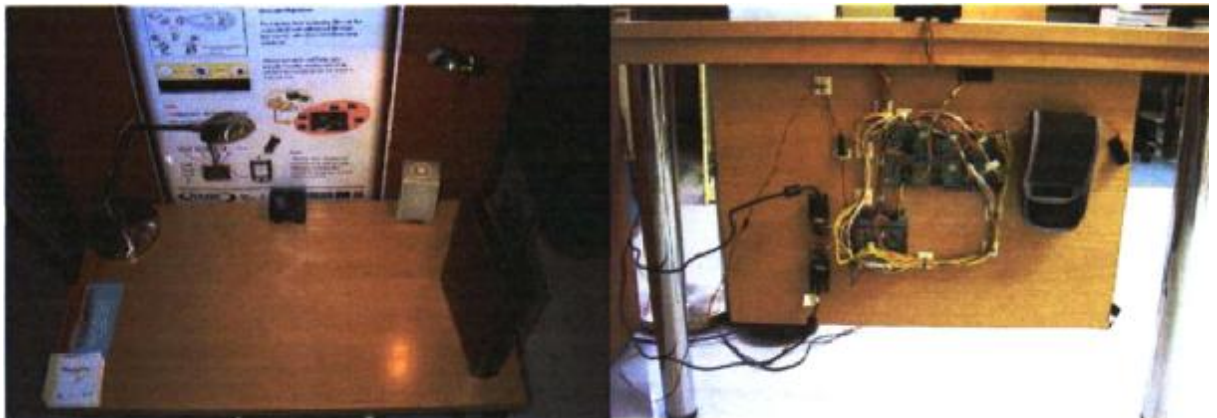
Με παρόμοιο τρόπο, μέσω διεπαφών υλικού και της πλατφόρμας Java, δεδομένα από άλλα τεχνουργήματα γίνονται διαθέσιμα στο GAS-OS μέσω της διεπαφής συνδεσιμότητας (Connectivity API). Η διεπαφή συνδεσιμότητας είναι υπεύθυνη για την ανταλλαγή μηνυμάτων μεταξύ των τεχνουργημάτων και διαχειρίζεται την πολυπλοκότητα της επικοινωνίας (π.χ. πρωτόκολλα επικοινωνίας, δρομολόγησης, ασφάλειας κτλ.).

3.3 Πρωτότυπα τεχνουργήματα

Παρακάτω θα γίνει μία σύντομη αναφορά σε κάποια πρωτότυπα τεχνουργήματα:

3.3.1 Τεχνούργημα θρανίο(EDesk)

Το τεχνούργημα θρανίο στην εικόνα 4 περιέχει 30 αισθητήρες φωτεινότητας (LDR=LightDependentResistor) ισοκατανομημένους στην επιφάνειά του και διατεταγμένους σε ορθογώνιο πλέγμα 6x5.



Εικόνα 4 Άποψη του πρωτότυπου θρανίου και του απαιτούμενου υλικού

Ο αισθητήρας εξάγει ένα ψηφιακό σήμα δύο καταστάσεων (on/off) στην πλακέτα FPGA, όπου το off (λογικό 0) σημαίνει ότι εισέρχεται φως στον αισθητήρα, ενώ το on (λογικό 1) σημαίνει ότι ο συγκεκριμένος αισθητήρας είναι καλυμμένος (π.χ. από ένα αντικείμενο). Το θρανίο διαιρείται σε τρεις τομείς έτσι ώστε να δίνει και πληροφορία για τη θέση των αντικειμένων πάνω του.

Το θρανίο περιέχει επίσης ένα δέκτη υπερήχων ο οποίος μπορεί να χρησιμοποιηθεί για την ανίχνευση της εγγύτητας άλλων αντικειμένων που έχουν πομπό υπερήχων αλλά και ένα δέκτη υπέρυθρων ο οποίος μπορεί να διαβάζει αναγνωριστικά IrDA (tags) που πιθανό να εκπέμπονται από άλλα αντικείμενα.

Οι σύνδεσμοι που ορίζονται από το λογισμικό του θρανίου είναι:

- **ObjectsOnTop:** Σύνδεσμος εξόδου, τύπου Boolean που εκδηλώνει την ύπαρξη αντικειμένων επάνω στο θρανίο.
- **Location:** Σύνδεσμος εξόδου, τύπου Integer που εκδηλώνει τον τομέα του γραφείου που είναι κατειλημμένος.
- **BookOpenOnTop:** Εξειδικευμένος σύνδεσμος εισόδου τύπου Boolean, που επιτρέπει στο θρανίο να συνεργάζεται με τεχνουργήματα βιβλία. Μέσω αυτού του συνδέσμου το θρανίο εισάγει την κατάσταση ενός συνδεδεμένου βιβλίου (ανοιχτό/κλειστό), ενώ μέσω του δέκτη υπέρυθρων μπορεί να ταυτοποιήσει αν το συνδεδεμένο βιβλίο είναι αυτό που βρίσκεται επάνω του.
- **ChairInFront:** Εξειδικευμένος σύνδεσμος εισόδου τύπου Boolean, που επιτρέπει στο θρανίο να συνεργάζεται με τεχνουργήματα καρέκλες. Μέσω αυτού του συνδέσμου το θρανίο εισάγει την κατάσταση μιας συνδεδεμένης καρέκλας (κατειλημμένη ή όχι).
- **O ReadingActivity:** Εξειδικευμένος σύνδεσμος εξόδου τύπου Boolean, που επιτρέπει στο θρανίο, όταν συμμετέχει σε σύνθεση τεχνουργημάτων με βιβλία και καρέκλες, να αποφαινεται αν εκτελείται δραστηριότητα μελέτης, σύμφωνα με κάποια λογική η οποία μπορεί να οριστεί με κανόνες.

3.3.2 Τεχνούργημα Καρέκλα (Echair)

Το τεχνούργημα καρέκλα στην εικόνα 5 έχει μόνο έναν αισθητήρα πίεσης επάνω στο κάθισμα ο οποίος ενεργοποιείται μόνο από σχετικά μεγάλο βάρος όπως για παράδειγμα από έναν άνθρωπο που κάθεται στην καρέκλα.



Εικόνα 5 Πρωτότυπη καρέκλα και απαιτούμενο υλικό

Οι σύνδεσμοι που ορίζονται από το λογισμικό της καρέκλας είναι:

- **Occurancy:** σύνδεσμος εξόδου, τύπου Boolean που εκδηλώνει την κατάσταση της καρέκλας (κατειλημμένη ή όχι).

3.3.3 Τεχνούργημα Βιβλίο (Ebook)

Κάθε βιβλίο με την προσθήκη ενός αναγνωριστικού (Tag) για βιβλία μπορεί να μετατραπεί σε τεχνούργημα βιβλίο στην εικόνα 6. Το αναγνωριστικό αυτό ενσωματώνει έναν αισθητήρα γωνίας που μπορεί να ανιχνεύσει τις καταστάσεις ανοιχτό/κλειστό του βιβλίου, αλλά και ένα πομπό υπέρυθρων που εκπέμπει την ταυτότητά του.



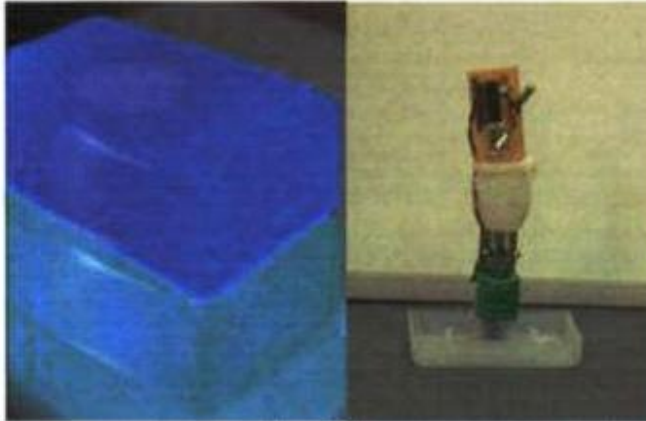
Εικόνα 6 Άποψη του πρωτότυπου βιβλίου

Οι σύνδεσμοι που ορίζονται από το λογισμικό του βιβλίου είναι:

- **Opened.** Σύνδεσμος εξόδου, τύπου Boolean που δηλώνει την κατάσταση του βιβλίου (ανοιχτό/κλειστό).

3.3.4. Τεχνούργημα Κύβος Διάθεσης (EmoodCube)

Το τεχνούργημα κύβος διάθεσης στην εικόνα 7 βασίζεται σε εμπορικό προϊόν της εταιρείας Mathmos [Mathmos, 2007] και έχει το χαρακτηριστικό ότι μπορεί και αλλάζει χρώματα ανάλογα με τον προσανατολισμό του. Για τα σενάρια της εργασίας έχει χαρακτηριστεί ως κύβος διάθεσης (moodcube) και αποτελεί τον τρόπο με τον οποίο ένας χρήστης μπορεί να εισάγει τη διάθεσή του στο σύστημα. Κάθε χρώμα θεωρείται ότι αντιπροσωπεύει και μια διάθεση. Για παράδειγμα όταν ο κύβος είναι γυρισμένος ώστε να ανάβει το κόκκινο χρώμα το σύστημα καταλαβαίνει ότι ο άνθρωπος έχει χαρούμενη διάθεση, ενώ το μωβ μπορεί να παριστάνει μια υποτονική διάθεση.



Εικόνα 7 Άποψη του πρωτότυπου κύβου διάθεσης

Ο κύβος διάθεσης ενσωματώνει τρεις διακόπτες κλίσης που δείχνουν σε τρεις διαγώνιες διευθύνσεις από διαφορετικές γωνίες, πράγμα που επιτρέπει τον υπολογισμό του προσανατολισμού του κύβου μέσα στο FPGA. Επειδή ο κύβος πρέπει να είναι φορητός και άρα να μην εξαρτάται από καλώδια, έχει ενσωματωθεί ένας πομπός ραδιοκυμάτων (RF) μέσω του οποίου μεταδίδεται η κατάσταση των αισθητήρων στο FPGA.

Οι σύνδεσμοι που ορίζονται από το λογισμικό του κύβου διάθεσης είναι:

- **Mood.** Σύνδεσμος εξόδου, τύπου Integer που εκδηλώνει, μέσω κωδικοποίησης ακεραίων, το ενεργό χρώμα του κύβου (κόκκινο, μωβ, μπλε κτλ.).

3.3.5.Τεχνούργημα Χαλί (Ecarpet)

Το τεχνούργημα χαλί στην εικόνα 8 περιέχει αισθητήρες πίεσης χωρισμένους σε 6 ενεργούς τομείς.



Εικόνα 11. Άποψη του πρωτότυπου χαλιού

Εικόνα 8 Άποψη του πρωτότυπου χαλιού

Οι σύνδεσμοι που ορίζονται από το λογισμικό του χαλιού είναι:

- **Weight** σύνδεσμος εξόδου, τύπου Boolean, που εκδηλώνει την ύπαρξη βαριών αντικειμένων πάνω στο χαλί
- **Location** σύνδεσμος εξόδου, τύπου Integer που εκδηλώνει τον τομέα του χαλιού που είναι κατειλημμένος.

3.3.6.Τεχνούργημα Φωτιστικού δαπέδου (EfloorLamb) και πορτατίφ (EdeskLamp)

Τα δύο φωτιστικά που βλέπουμε στην εικόνα 9 είναι πανομοιότυπα από άποψη υλικού ενώ οι διαφορές τους έγκειται στη μηχανική τους κατασκευή. Σε αντίθεση με όλα τα προηγούμενα τεχνουργήματα τα φωτιστικά είναι αντικείμενα εξόδου, έχουν δηλαδή ενεργοποιητές (actuators) και όχι αισθητήρες.



Εικόνα 9 Άποψη του φωτιστικού δαπέδου και του πορτατίφ και του αντίστοιχου υλικού

Οι σύνδεσμοι που ορίζονται από το λογισμικό των φωτιστικών είναι:

- **Light:** Σύνδεσμος εισόδου, τύπου Integer που εισάγει την επιθυμητή φωτεινότητα της λάμπας.
- **Switch:** Σύνδεσμος εισόδου, τύπου Boolean που εισάγει εντολές για άνοιγμα ή κλείσιμο του φωτός.

3.3.7.Τεχνούργημα MP3Player(EMp3player)

Εκτός από τα τεχνουργήματα που είδαμε μέχρι τώρα τα οποία ήταν επαυξημένα φυσικά αντικείμενα, υπάρχουν και αυτά που δεν έχουν φυσική υπόσταση και εκτελούνται μόνο ως διεργασίες. Ένα τέτοιο τεχνούργημα είναι και το MP3 player, ένα πρόγραμμα που μπορεί και αναπαράγει μουσική σε μορφή MP3. Η ύπαρξη μόνο της ψηφιακής υπόστασης του τεχνουργήματος αυτού, καθιστά απαραίτητη τη χρήση υπολογιστικής πλακέτας με οθόνη (π.χ. PDA) έτσι ώστε να είναι εφικτή η αλληλεπίδραση του χρήστη με αυτό. (Jiangi & Manivannan 2004).

Οι δυνατότητές του μοιάζουν με αυτές ενός παραδοσιακού ηχοσυστήματος αφού μπορεί και αναπαράγει μουσική, υποστηρίζει λίστες τραγουδιών (π.χ. Mixed, Pop, Rock, Easy) και πλοήγηση σε αυτές, αυξομείωση της έντασης κτλ.

Οι σύνδεσμοι που ορίζονται από το λογισμικό του MP3 player είναι:

- **PlayStop:** Σύνδεσμος εισόδου/εξόδου, τύπου Boolean που είτε εισάγει εντολή αναπαραγωγής μουσικής, είτε εξάγει την κατάσταση αναπαραγωγής.
- **Pause:** Σύνδεσμος εισόδου/εξόδου, τύπου Boolean που είτε εισάγει εντολή παγώματος της αναπαραγωγής, είτε εξάγει την κατάσταση παγώματος.
- **PreviousTrack:** Σύνδεσμος εισόδου/εξόδου, τύπου Boolean που είτε εισάγει εντολή αναπαραγωγής του προηγούμενου τραγουδιού, είτε εξάγει το γεγονός ότι εκτελέστηκε μεταπήδηση στο προηγούμενο κομμάτι.
- **NextTrack:** Σύνδεσμος εισόδου/εξόδου, τύπου Boolean που είτε εισάγει εντολή αναπαραγωγής του επόμενου τραγουδιού, είτε εξάγει το γεγονός ότι εκτελέστηκε μεταπήδηση στο επόμενο κομμάτι.
- **Gender:** Σύνδεσμος εισόδου/εξόδου, τύπου Integer που είτε εισάγει εντολή επιλογής λίστας αναπαραγωγής (π.χ. Rock, Pop, κτλ.), είτε εξάγει τη λίστα που επιλέχθηκε.
- **Volume:** Σύνδεσμος εισόδου/εξόδου, τύπου Integer που είτε εισάγει εντολή αλλαγής της έντασης της μουσικής, είτε εξάγει την τιμή της έντασης.

3.3.8 Τεχνούργημα Ρολόι

Ένα άλλο τεχνούργημα που υφίσταται μόνο ως διεργασία είναι το ρολόι. Όπως και το MP3 player έτσι και το ρολόι απαιτεί την ύπαρξη οθόνης έτσι ώστε να είναι εφικτή η διεπαφή με το χρήστη. Οι δυνατότητες που υποστηρίζει είναι, η προβολή της τρέχουσας ώρας και η αφύπνιση (Jiangi & Manivannan 2004).

Οι σύνδεσμοι που ορίζονται από το λογισμικό του ρολογιού είναι:

- **Time:** Σύνδεσμος εξόδου, τύπου String που εξάγει την τρέχουσα ώρα.
- **Alarm:** Σύνδεσμος εισόδου εξόδου, τύπου Boolean. Ως εισόδου εισάγει εντολή για να θέσει σε λειτουργία την αφύπνιση, ενώ ως εξόδου εξάγει ειδοποίηση ότι η αφύπνιση τέθηκε σε λειτουργία.

ΚΕΦΑΛΑΙΟ 4

Διαμεσολαβητικό Λογισμικό GAS-OS

Το κεφάλαιο αυτό παρουσιάζει η σχεδίαση και η υλοποίηση του διαμεσολαβητικού λογισμικού GAS-OS για τη σύνθεση εφαρμογών περιρρέουσας νοημοσύνης. Συγκεκριμένα, αρχικά θα δοθούν ορισμοί για τα διαμεσολαβητικό λογισμικό και θα γίνει μια αναφορά στους διαφορετικούς τύπους τέτοιων συστημάτων. Στη συνέχεια θα αναλυθούν οι απαιτήσεις και οι σχεδιαστικές αποφάσεις που εξήχθησαν κατά τη διαδικασία σχεδίασης του GAS-OS ενώ αμέσως μετά θα παρουσιαστεί η αρχιτεκτονική του GAS-OS η οποία αποτελεί ένα στιγμιότυπο της γενικής αρχιτεκτονικής που παρουσιάστηκε σε προηγούμενο κεφάλαιο εμπλουτισμένης με το εννοιολογικό πλαίσιο GAS. Το παρόν κεφάλαιο θα αναλύσει στη συνέχεια την υλοποίηση του διαμεσολαβητικού λογισμικού παρουσιάζοντας τεχνικές, πρωτόκολλα, και αλγόριθμους που

επιστρατεύτηκαν κατά τη φάση υλοποίησης. Οι δύο τελευταίες ενότητες αφορούν τους οδηγούς του GAS-OS δηλαδή τον τρόπο με τον οποίο το GAS-OS επικοινωνεί με περιφερειακές συσκευές (αισθητήρες και ενεργοποιητές) και τέλος παρουσιάζεται μια εκτίμηση της επίδοσης του GAS-OS μέσα από θεωρητικές και πειραματικές διαδικασίες που εξήχθησαν στα πλαίσια της εργασίας.

4.1 Τι είναι Διαμεσολαβητικό Λογισμικό (Middleware)

Υπάρχουν διάφοροι ορισμοί για το τι είναι διαμεσολαβητικό λογισμικό (middleware) στη βιβλιογραφία. Αυτός που φαίνεται να είναι πιο πλήρης και καλύπτει τις περισσότερες πτυχές του όρου είναι ο εξής: «Διαμεσολαβητικό λογισμικό είναι το λογισμικό που βοηθάει μια εφαρμογή να αλληλοεπιδρά ή να επικοινωνεί με άλλες εφαρμογές, δίκτυα, υλικό και/ή λειτουργικά συστήματα. Το λογισμικό αυτό βοηθάει επίσης τους προγραμματιστές απαλλάσσοντάς τους από πολύπλοκες διασυνδέσεις που συνήθως απαιτούνται σε ένα κατακεντρωμένο σύστημα. Παρέχει εργαλεία για τη βελτίωση της ποιότητας υπηρεσιών (QoS), της ασφάλειας, της μετάδοσης μηνυμάτων, υπηρεσιών καταλόγου, υπηρεσιών αρχείων, κτλ. που μπορεί να είναι αόρατες προς το χρήστη» (Bishop. 2003).

Τα διαμεσολαβητικά λογισμικά είναι συνήθως σχεδιασμένα έτσι ώστε να κρύβουν κάποια είδη ετερογένειας που καλούνται να αντιμετωπίσουν οι προγραμματιστές των κατακεντρωμένων συστημάτων. Όλα τα διαμεσολαβητικά λογισμικά κρύβουν την ετερογένεια δικτύων και υλικού ενώ αρκετά κρύβουν επίσης την ετερογένεια λειτουργικών συστημάτων ή των γλωσσών προγραμματισμού ή και των δύο. Ακόμα λιγότερα, όπως το CORBA, είναι αυτά που κρύβουν ετερογένειες που παρουσιάζονται μεταξύ διαφορετικών υλοποιήσεων του ίδιου διαμεσολαβητικού λογισμικού. Τέλος, αφηρημένες προγραμματιστικές έννοιες που προσφέρονται από ένα διαμεσολαβητικό λογισμικό, μπορούν να παρέχουν διαφάνεια ως προς την κατανομή σε μια ή περισσότερες από τις ακόλουθες περιπτώσεις: τοποθεσία (location), ταυτοχρονισμός (concurrency), ύπαρξη αντιγράφου (replication), αποτυχίες (failures), και κινητικότητα (mobility). Όπως σύμφωνα με τον κλασικό ορισμό

ένα λειτουργικό σύστημα είναι ένα λογισμικό που καθιστά χρησιμοποιήσιμο το υλικό, έτσι και το διαμεσολαβητικό λογισμικό μπορεί να θεωρηθεί ως το λογισμικό που καθιστά ένα κατανεμημένο σύστημα ικανό να προγραμματιστεί.

Ο όρος διαμεσολαβητικό λογισμικό είναι αρκετά γενικός και καλύπτει μια μεγάλη ποικιλία λογισμικών. Υπάρχουν διάφοροι τρόποι για την ταξινόμηση τέτοιων συστημάτων. Αν και τέτοιου είδους ταξινόμηση είναι εκτός του σκοπού της παρούσας εργασίας παρολαυτά θα γίνει μια σύντομη απαρίθμηση των πιο σημαντικών κατηγοριών συστημάτων διαμεσολάβησης, έτσι ώστε να γίνει σαφής ο διαχωρισμός της στοχευόμενης έρευνας για το GAS-OS (SOAP 2007).

Το Συναλλακτικό διαμεσολαβητικό λογισμικό (Transactional middleware):

Υποστηρίζει συναλλαγές που αφορούν ψηφίδες που εκτελούνται σε κατανεμημένους ξενιστές (hosts). Τέτοια διαμεσολαβητικά λογισμικά χρησιμοποιούν τα δεσμευτικά πρωτόκολλα δύο φάσεων (two-phase commit protocols) (Bernstein, 1987| για να υλοποιήσουν κατανεμημένες συναλλαγές. Παραδείγματα αυτής της κατηγορίας είναι τα IBM CICS (Hudders, 1994| και BEA Tuxedo (Hall, 1996|. β Διαμεσολαβητικό λογισμικό προσανατολισμένο στα μηνύματα (Message Oriented Middleware - MOM): Υποστηρίζει την επικοινωνία μεταξύ κατανεμημένων ψηφίδων του συστήματος μέσω ουρών μηνυμάτων πάνω από δίκτυα. Χαρακτηριστικά που διαφοροποιούν τα MOMs από άλλους τύπους διαμεσολαβητικών λογισμικών είναι: ασύγχρονη επικοινωνία, επικοινωνία ομάδων, αποθήκευση των μηνυμάτων σε επίμονα (persistent) μέσα αποθήκευσης, κτλ.. Παραδείγματα MOM είναι τα IBM MQSeries [Gilman, 1996J. και Sun Java Message Queue (Harner, 1999) διαδικαστικό σύστημα διαμεσολάβησης (Procedural middleware):

Επεκτείνει τη διεπαφή κλήσης διαδικασιών προσφέροντας επιπλέον την ικανότητα της κλήσης συναρτήσεων των οποίων το σώμα βρίσκεται κάπου στο δίκτυο. Τα συστήματα απομακρυσμένης κλήσης διαδικασιών (Remote Procedure Call RPC) είναι συνήθως σύγχρονα, συνεπώς δεν προσφέρουν δυνατότητες παράλληλης επεξεργασίας χωρίς τη χρήση πολλαπλών νημάτων

(threads), ενώ τυπικά έχουν περιορισμένες ικανότητες διαχείρισης εξαιρέσεων (exceptions). Συστήματα με υποστήριξη RPC πρωτοεμφανίστηκαν από τη Sun Microsystems στις αρχές της δεκαετίας του 1980.

Σύστημα διαμεσολάβησης με αντικείμενα (Object middleware): Καθιστά τις αντικειμενοστραφείς αρχές, όπως την αναγνώριση αντικειμένων μέσω αναφορών, την κληρονομικότητα, τον πολυμορφισμό, κτλ., διαθέσιμες για την ανάπτυξη καταναμημένων συστημάτων. Η επικοινωνία μεταξύ αντικειμένων (αντικείμενο- πελάτης και αντικείμενο-εξυπηρετητής) μπορεί να είναι σύγχρονη, σύγχρονη κατά παραγγελία, και ασύγχρονη με χρήση διαφόρων πολιτικών στα εκτελούμενα νήματα. Ένα στάνταρ καταναμημένου συστήματος με αντικείμενα είναι το Common Object Request Broker Architecture (CORBA) [CORBA, 1998]. Είναι μέρος της αρχιτεκτονικής Object Management Architecture (OMA) της ομάδας Object Management Group (OMG) και αποτελεί το πιο διαδεδομένο διαμεσολαβητικό λογισμικό με καταναμημένα αντικείμενα. Άλλα παραδείγματα είναι τα: Microsoft DCOM (Brown, 1998) και Sun Java RMI (Java RMI).

Σύστημα διαμεσολάβησης με ψηφίδες (Component middleware): Οι ψηφίδες είναι μονάδες λογισμικού προερχόμενες από διάφορους προγραμματιστές οι οποίες μπορούν να συντεθούν με διάφορους τρόπους, που μπορεί να μην έχουν προβλεφθεί από τους δημιουργούς τους, για την επίτευξη της εκάστοτε επιθυμητής συμπεριφοράς. Ένα σύστημα διαμεσολάβησης με ψηφίδες είναι μια διαμόρφωση ψηφίδων που έχουν επιλεγθεί είτε κατά το χρόνο ανάπτυξης είτε κατά το χρόνο εκτέλεσης. Παραδείγματα αποτελούν τα: OMG's COR HA Component Model (CORBAComponents, 2002], Microsoft's .NET [.NET, 2007] και Sun's Enterprise Java Beans.

Το Συνδρομητικό σύστημα διαμεσολάβησης (Publish-subscribe middleware)

Το συνδρομητικό σύστημα διαμεσολάβησης είναι ένας τύπος MOM. Διαφέρει από τα συστήματα επικοινωνίας από σημείο-σε-σημείο (point-to-

point) στο ότι η επικοινωνία μεταξύ τερματικών σημείων είναι ανώνυμη, ασύγχρονη και χαλαρά συζευγμένη (loosely coupled). Σε ένα συνδρομητικό μοντέλο επικοινωνίας οι εφαρμογές χρησιμοποιούν θεματικά ονόματα για τη διανομή των δεδομένων, και όχι διευθύνσεις δικτύου. Οι εκδότες (Publishers) δημιουργούν απλά μια δημοσίευση και της δίνουν ένα θεματικό όνομα στο οποίο μπορούν στη συνέχεια να στέλνουν διάφορες πληροφορίες (δεδομένα). Οι συνδρομητές δημιουργούν απλά μια συνδρομή για ένα θεματικό όνομα ενώ ειδοποιούν το σύστημα διαμεσολάβησης να κάνει συγκεκριμένες ενέργειες κάθε φορά που φτάνει μια νέα πληροφορία. Παραδείγματα συνδρομητικών συστημάτων είναι τα: Elvin (Scgalland, 1997), Tibco Rcvdncvouz (Tibco, 2007) και NODS.

Το Σύστημα διαμεσολάβησης προσανατολισμένο σε υπηρεσίες (Service oriented middleware): Ένα τέτοιο σύστημα είναι ουσιαστικά μια συλλογή υπηρεσιών. Μια υπηρεσία πραγματοποιεί κάποιες λειτουργίες και έχει τις ακόλουθες ιδιότητες:

- Η διεπαφή προς την υπηρεσία είναι ανεξάρτητη πλατφόρμας
- Η υπηρεσία μπορεί να βρεθεί και κληθεί δυναμικά
- Η υπηρεσία είναι αυτό-περιεχόμενη (self-contained)

Η ιδέα αρχιτεκτονικών προσανατολισμένων σε υπηρεσίες (service-oriented architectures) απορρέει ως ένα σημαντικό βαθμό από αυτή του αντικειμενοστραφούς προγραμματισμού, που έχει σα βασική θέση ότι τα δεδομένα και η επεξεργασία τους πρέπει να συνδέονται. Έτσι, η προσανατολισμένη σε υπηρεσίες αρχιτεκτονική έχει ως κύριο στόχο την επίτευξη χαλαρών συζεύξεων μεταξύ των ψηφίδων λογισμικού, που είναι υπηρεσίες. Παραδείγματα τέτοιων αρχιτεκτονικών είναι συστήματα υπηρεσιών διαδικτύου όπως το (SOAP 2007).

Αν και τα υπάρχοντα διαμεσολαβητικά λογισμικά καλύπτουν ευρεία έκταση εφαρμογών, έχουν και κάποιες αδυναμίες, όπως:

- Ακαμψία: Δεν αποκρίνονται καλά σε μεταβαλλόμενες απαιτήσεις

- Αδυναμία κλιμάκωσης: Αν και αποκρίνονται καλά στα τοπικά δίκτυα, δε μπορούν να κλιμακωθούν σε δίκτυα ευρείας περιοχής (WAN) όπως και σε συστήματα περιορισμένων πόρων.

Κατά συνέπεια εμφανίστηκαν νέα συστήματα διαμεσολάβησης που αντιμετωπίζουν αυτές τις απαιτήσεις, όπως:

- Προσαρμοστικά και ανακλαστικά συστήματα διαμεσολάβησης (Adaptive and Reflective middleware): Ένα προσαρμοστικό σύστημα διαμεσολάβησης είναι ένα λογισμικό του οποίου οι λειτουργικές ιδιότητες και οι ιδιότητες που σχετίζονται με την ποιότητα υπηρεσιών (QoS) μπορούν να μεταβληθούν είτε:
 - Στατικά, π.χ. μείωση του ίχνους, αξιοποίηση των δυνατοτήτων που προσφέρονται από συγκεκριμένες πλατφόρμες, υποστήριξη υποσυνόλων λειτουργιών, και ελαχιστοποίηση των εξαρτήσεων από υποδομές υλικού και λογισμικού.
 - Δυναμικά, π.χ. βελτιστοποίηση των αποκρίσεων του συστήματος σε μεταβαλλόμενα περιβάλλοντα ή απαιτήσεις όπως αλλαγή στις συνδέσεις μεταξύ ψηφίδων, στα επίπεδα ισχύος, στην επεξεργαστική ισχύ και στο εύρος του δικτύου.

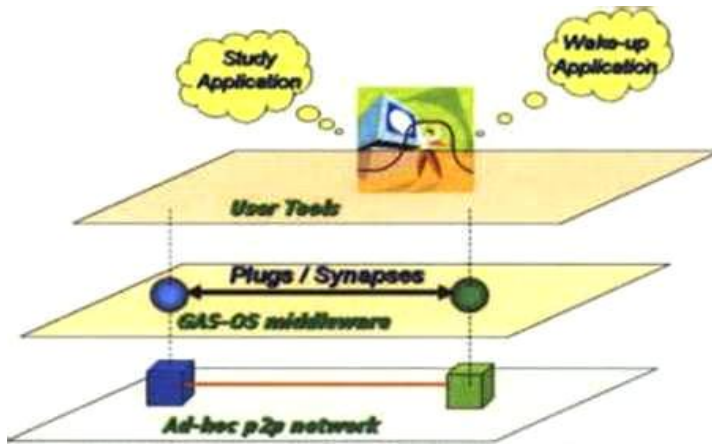
Ένα ανακλαστικό σύστημα διαμεσολάβησης καθιστά τόσο την εσωτερική οργάνωση των συστημάτων όσο και τους μηχανισμούς που χρησιμοποιήθηκαν για την ανάπτυξή τους, ορατά και ρυθμιζόμενα σε χρόνο εκτέλεσης. Επιτρέπει αυτόματο έλεγχο των προφερόμενων δυνατοτήτων και αυτόματη ρύθμιση για την βελτιστοποίησή τους (Schmidt, 2002). Παραδείγματα ανακλαστικών συστημάτων είναι τα OpenCORBA (Ledoux, 1999), FlexiNct (Hayton, 1997) (Singhai, 1998).

- Συστήματα διαμεσολάβησης για κινητούς υπολογιστές (Middleware for Mobile Computing): Αντιμετωπίζουν προβλήματα όπως απώλεια σύνδεσης δικτύου (network unreachability), χαμηλή ταχύτητα σύνδεσης, συνδέσεις p2p. κτλ.
- Συστήματα διαμεσολάβησης πραγματικού χρόνου (Real-time Middleware):

Απευθύνονται σε κατανεμημένες εφαρμογές που χρειάζονται να αντιδρούν σε γεγονότα σε καθορισμένο χρόνο. Τέτοια διαμεσολαβητικά λογισμικά προσφέρουν τα μέσα στις εφαρμογές για έλεγχο χρονικών περιορισμών, προτεραιοτήτων, και δέσμευση πόρων όπως επεξεργαστική ισχύς, μνήμη και εύρος ζώνης. Αυτά απαιτούν μηχανισμούς που να εξασφαλίζουν ότι η εφαρμογή είναι ικανή να τηρεί τις διορίες(SOAP 2007).

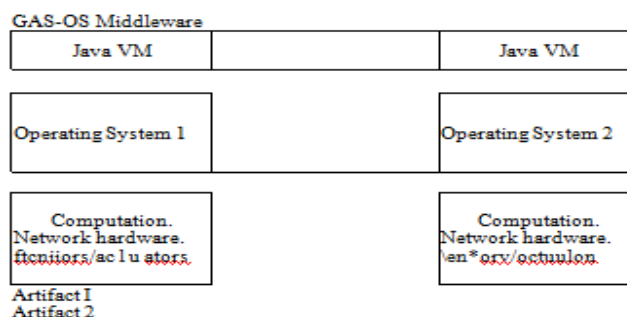
4.2 Ανάλυση και Σχεδίαση του GAS-OS

Το εννοιολογικό πλαίσιο που ορίζεται από το GAS παρουσιάζει μια εξειδικευμένη οπτική του χώρου της περιρρέουσας νοημοσύνης, που βασίζεται στην ενσωμάτωση και εμπλουτισμό των τεχνουργημάτων με δυνατότητες καταληπτότητας χρήσης (affordances). Η υλοποίηση ενός τέτοιου μοντέλου για τη γενίκευση της αλληλεπίδρασης με περιβάλλοντα περιρρέουσας νοημοσύνης υπαγορεύει τη δημιουργία και χρήση ενός διαμεσολαβητικού λογισμικού (middleware) που λειτουργεί ως γέφυρα μεταξύ των βασικών τεχνολογικών επιπέδων (όπως πρωτόκολλα, επικοινωνία, λειτουργικά συστήματα) που υποστηρίζουν την ψηφιακή υπόσταση των αντικειμένων, των ίδιων των αντικειμένων ως φυσικές οντότητες και των ανθρώπων - χρηστών (μέσω κατάλληλων εργαλείων). Ο σχεδιασμός και η ανάπτυξη ενός τέτοιου διαμεσολαβητικού λογισμικού απαιτεί την ισορροπημένη μελέτη τόσο ανθρωποκεντρικών εννοιών, όσο και τεχνολογικών απαιτήσεων και περιορισμών από τη στιγμή της σύλληψής του (SOAP 2007).



Εικόνα 10: Το GAS-OS ως γέφυρα μεταξύ τεχνολογικών επιπέδων, αντικειμένων, και χρηστών.

Η δημιουργία του διαμεσολαβητικού λογισμικού GAS-OS είχε ως κίνητρο την αντιμετώπιση της ετερογένειας και την παροχή μιας ομοιόμορφης και αφαιρετικής οπτικής των υπηρεσιών και των δυνατοτήτων των τεχνουργήματα. Το GAS-OS, ακολουθώντας τη γενική αρχιτεκτονική του Κεφαλαίου 3 και υλοποιώντας το μοντέλο Συνδέσμων/Συνάψεων, αφαιρεί τις λεπτομέρειες της υποκείμενης επικοινωνίας δεδομένων και της πρόσβασης σε αισθητήρες και ενεργοποιητές σε όλη την έκταση ενός κατανεμημένου συστήματος, έτσι ώστε μια εφαρμογή περιρρέουσας υπολογιστικής νοημοσύνης να εμφανίζεται ως μια ολοκληρωμένη υπολογιστική υπηρεσία (SOAP 2007).



Εικόνα 19. Ενοποιώντας ένα κατανεμημένο σύστημα από τεχνουργήματα με το GAS- OS

Εικόνα11: Εφαρμογή του GAS-OS

Πιο συγκεκριμένα ας θεωρήσουμε κάθε κόμβο μιας κατακευμαμένης εφαρμογής περιρρέουσας νοημοσύνης να είναι ένα τεχνούργημα με το δικό του υλικό και λειτουργικό σύστημα. Το GAS-OS μπορεί να εγκατασταθεί σε κάθε κόμβο εφόσον υπάρχει εγκατεστημένη μια Java VM. Η διασύνδεση των διαφόρων τμημάτων του GAS-OS που επιτυγχάνεται μέσω του μοντέλου Συνδέσμων/Συνάψεων έχει ως αποτέλεσμα την εικονική ενοποίηση των κόμβων του κατακευμαμένου συστήματος δίνοντας την εντύποκτη αλληλεπίδρασης με ένα κεντροποιημένο σύστημα. Παράλληλα κρύβει την ετερογένεια των λειτουργικών συστημάτων, δικτύων, υλικού και γλωσσών προγραμματισμού καθιστώντας έτσι το κατακευμαμένο σύστημα ικανό να προγραμματιστεί και μάλιστα από μη-ειδικούς τελικούς χρήστες, μέσω του μοντέλου Συνδέσμων/Συνάψεων. (SOAP 2007)

Έχοντας υπόψη την αρχική ταξινόμηση των διαμεσολαβητικών λογισμικών, μπορούμε να υποστηρίξουμε ότι το GAS-OS συνδυάζει στοιχεία διαμεσολαβητικών λογισμικών προσανατολισμένων σε μηνύματα (Message Oriented Middleware-MOM), συστημάτων διαμεσολάβησης με ψηφίδες (Component middleware) και συστημάτων διαμεσολάβησης προσανατολισμένων σε υπηρεσίες (Service oriented middleware). Επιπλέον όπως και τα συστήματα διαμεσολάβησης για κινητούς υπολογιστές (Middleware for Mobile Computing) αντιμετωπίζει προβλήματα όπως απώλεια σύνδεσης δικτύου (network unreachability), χαμηλή ταχύτητα σύνδεσης, συνδέσεις r2r κτλ.

Ως MOM παρέχει πέρασμα μηνυμάτων χωρίς περιορισμούς σε συνδυασμό με υπηρεσίες αναμονής σε ουρές. Η επικοινωνία μέσω μηνυμάτων όπως και η χρήση ουρών επιτρέπουν στους κόμβους να επικοινωνούν χωρίς να είναι συνδεδεμένοι μέσω κάποιας μόνιμης αφιερωμένης σύνδεσης, ενώ τα στάδια αποστολής και λήψης μηνυμάτων μπορούν να γίνονται σε διαφορετικές χρονικές στιγμές. Κάθε κόμβος επικοινωνεί με έναν άλλο τοποθετώντας μηνύματα σε ουρές και λαμβάνοντας μηνύματα από ουρές.

Ως component middleware επιτρέπει μέσω του μοντέλου Συνδέσμων/Συνάψεων τη σύνθεση των επί μέρους κομματιών του middleware που βρίσκονται σε κάθε τεχνούργημα τα οποία παίζουν το ρόλο των

components. Κάθε τεχνούργημα μπορεί να προέρχεται από διαφορετικούς κατασκευαστές / προγραμματιστές ενώ οι τρόποι σύνθεσής τους μπορεί να μην έχουν προβλεφθεί από τους δημιουργούς τους. Για παράδειγμα ένα τεχνούργημα - δάπεδο μπορεί να έχει κατασκευαστεί για να συνεργάζεται με τα φώτα του δωματίου (π.χ. όταν κάποιος πατάει στο δάπεδο να ανάβουν τα φώτα) αλλά μπορεί κάλλιστα να χρησιμοποιηθεί και ως «τηλεχειριστήριο» για ένα τεχνούργημα - στερεοφωνικό. Η διαμόρφωση των τεχνουργημάτων-components μπορεί να γίνει κατά το χρόνο εκτέλεσης της εφαρμογής μέσω κατάλληλων εργαλείων.

Το GAS-OS δανείζεται στοιχεία και από τα service oriented middleware θεωρώντας τα τεχνουργήματα ως παρόχους και αποδέκτες υπηρεσιών. Με τη χρήση οντολογιών τα τεχνουργήματα μπορούν να «γνωρίζουν» τις υπηρεσίες που παρέχουν και τις υπηρεσίες που χρειάζονται για να υποστηρίξουν μια εφαρμογή. Μέσω κατάλληλων μηχανισμών το GAS-OS μπορεί να αναζητήσει και να χρησιμοποιήσει υπηρεσίες ενώ δυναμικά μπορεί να αντικαταστήσει τη συμμετοχή ενός τεχνουργήματος στην εφαρμογή με κάποιο άλλο που παρέχει την ίδια ή παρόμοια υπηρεσία (π.χ. σε περίπτωση που μια υπηρεσία χάνεται λόγω δυσλειτουργίας ενός τεχνουργήματος, ή σε περίπτωση που η παρεχόμενη υπηρεσία παρέχεται καλύτερα από κάποιο άλλο τεχνούργημα).

4.3 Λειτουργικές Απαιτήσεις Μοντέλου Συνδέσμων/Συνάψεων

Από πλευρά λειτουργικότητας το GAS-OS χρειάζεται να προσφέρει:

- Ανεύρεση συνδέσμων (Plug discovery): Το GAS-OS είναι υπεύθυνο για την ανεύρεση όλων των συνδέσμων, συνεπώς και τεχνουργημάτων, εντός εμβελείας του τεχνουργήματος. Για την παροχή της υπηρεσίας αυτής το GAS-OS χρησιμοποιεί τη μονάδα δικτύωσης. Το GAS-OS δημιουργεί μια πολλαπλή εκπομπή (multicast) με ένα απλό «hello» μήνυμα και όλα τα τεχνουργήματα που βρίσκονται εντός εμβελείας και

ακούν στις IP πολλαπλής εκπομπής, αποκρίνονται με ένα μήνυμα γνωστοποίησης (advertisement). Αυτή η γνωστοποίηση περιέχει όλα τα δεδομένα ενός τεχνουργήματος που χρειάζεται να γνωστοποιούνται σε άλλα τεχνουργήματα, όπως η λίστα των συνδέσμων του. η τρέχουσα διεύθυνσή του στο δίκτυο (π.χ. διεύθυνση IP, αναγνωριστικό Bluetooth κτλ.), η δικτυακή Ούρα (port) στην οποία ακούει κτλ. και που είναι απαραίτητα για να επικοινωνήσει ένα τεχνούργημα με ένα άλλο. Το περιεχόμενο των μηνυμάτων δομείται σύμφωνα με μια προσαρμοσμένη διάταξη βασισμένη σε XML.

- **Δημιουργία και καταστροφή συνάψεων:** Το GAS-OS διαχειρίζεται τη διαδικασία δημιουργίας συνάψεων μεταξύ συνδέσμων. Παρέχει τρόπους για επιτυχείς συνδέσεις-αποσυνδέσεις, εξασφαλίζοντας ότι αυτές οι διαδικασίες εκτελούνται ως ατομικές που είτε πετύχουν ή θα αποτύχουν πριν την αποδέσμευση του συνδέσμου. Επιπλέον, εξασφαλίζει ότι οι σύνδεσμοι δε θα μείνουν κλειδωμένοι για απεριόριστο χρονικό διάστημα, στην περίπτωση αποτυχίας μιας σύναψης. Οι σύνδεσμοι είναι πλήρως λειτουργικοί και δεν παραμένουν κλειδωμένοι κατά τη διάρκεια αυτής της διαδικασίας. Αυτό είναι πολύ σημαντικό γιατί οι καθυστερήσεις στο δίκτυο μπορεί να είναι αρκετά μεγάλες ακόμα και για επιτυχείς δημιουργίες συνάψεων. Η διαδικασία εγκαθίδρυσης της επικοινωνίας (hand shaking) μεταξύ των συνδέσμων είναι ευθύνη του διαμεσολαβητικού λογισμικού. Το GAS-OS πρέπει να διασφαλίσει ότι οι σύνδεσμοι θα συνδεθούν μόνο όταν είναι διαθέσιμοι και εάν είναι «συμβατοί». Επίσης είναι ευθύνη του GAS-OS να διασφαλίσει ότι όταν ένα τεχνούργημα απενεργοποιείται όλοι οι σύνδεσμοι που έχουν σύναψη με τους δικούς του θα ειδοποιηθούν και οι συνάψεις θα καταστραφούν. Τέλος, το GAS-OS θα πρέπει να φροντίζει ώστε κατά την εκκίνησή του θα γίνει προσπάθεια υποκατάστασης όλων των συνάψεων που είχε προτού απενεργοποιηθεί.

Διαχείριση συνάψεων: Τα τεχνουργήματα ειδοποιούνται για τυχόν αλλαγές στην κατάσταση άλλων συνδεδεμένων τεχνουργημάτων μέσω των

συνδέσμων τους. Αυτό σημαίνει ότι το GAS-OS είναι υπεύθυνο για την αποστολή και τη διαχείριση τέτοιων ειδοποιήσεων και την προώθησή τους στην υπολογιστική λογική του τεχνουργήματος. Το GAS-OS λοιπόν λειτουργεί ως μεσάζοντας στη συνεργασία των τεχνουργημάτων.

- Ειδοποίηση για αλλαγές κατάστασης ενός τεχνουργήματος μέσω συνάψεων: Όταν η εσωτερική κατάσταση ενός τεχνουργήματος αλλάζει, τα συνδεδεμένα μέσω συνάψεων τεχνουργήματα μπορούν να αντιληφθούν την αλλαγή. Οι αιτήσεις για ειδοποιήσεις είναι ευθύνη της λογικής του τεχνουργήματος, ενώ η πραγματοποίησή τους γίνεται από το GAS-OS.
- Διαχείριση των ειδοποιήσεων που προέρχονται υπό συνδεδεμένα τεχνουργήματα: Όταν ένα τεχνουργήμα χρειάζεται να ειδοποιηθεί για την αλλαγή της κατάστασης ενός συνδεδεμένου τεχνουργήματος, το GAS-OS είναι αυτό που λαμβάνει την ειδοποίηση και αναλαμβάνει να πληροφορήσει την υπολογιστική λογική του τεχνουργήματος ότι πρέπει να διαχειριστεί το γεγονός αλλαγής κατάστασης και να δημιουργήσει κατάλληλη δράση.
- Διαχείριση συνδέσμων: Το GAS-OS επιβλέπει τη σύναψη για να γνωρίζει ανά πάσα στιγμή εάν είναι ενεργή ή όχι. Επιπλέον διαχειρίζεται όλες τις ειδοποιήσεις για γεγονότα που έρχονται από άλλους συνδεδεμένους συνδέσμους.
- Πρόσβαση ανάγνωσης των χαρακτηριστικών του συνδέσμου: Οποιοσδήποτε απομακρυσμένος σύνδεσμος ή ακόμα και άλλη οντότητα μπορεί να προσπελάσει και να υποβάλει ερωτήσεις "queries" σε έναν σύνδεσμο σχετικά με τα χαρακτηριστικά του.

4.4 Αρχιτεκτονική του GAS-OS

Το GAS-OS αποτελείται από τον πυρήνα (kernel) που συγκεντρώνει τη στοιχειώδη λειτουργικότητα που πρέπει να κατέχει κάθε τεχνούργημα και από έναν αριθμό από πρόσθετες μονάδες (plug-ins), οι οποίες μπορούν και συνεργάζονται με τον πυρήνα μέσω του Plug-in manager για να προσδίδουν περισσότερες δυνατότητες σε ένα τεχνούργημα.

Ο πυρήνας του GAS-OS έχει σχεδιαστεί για να υποστηρίζει αποδοχή και αποστολή μηνυμάτων, διαχείριση των τοπικών πόρων υλικού (αισθητήρες / ενεργοποιητές) και υλοποίηση του διαδραστικού μηχανισμού του μοντέλου συνδέσμων / συνάψεων.

Ο πυρήνας είναι επίσης ικανός να διαχειρίζεται τη διαδικασία ανεύρεσης υπηρεσιών και τεχνουργημάτων προκειμένου να διευκολύνει το σχηματισμό των επιθυμητών συνάψεων (Christopoulou. 2005).

Ο πυρήνας του GAS-OS ενσωματώνει μια μονάδα p2p (peer-to-peer) επικοινωνίας (Communication Module), μια μονάδα διαχείρισης διεργασιών (Process Manager), μια μονάδα διαχείρισης των μεταβλητών κατάστασης (State Variable Manager) και μια μονάδα αποτίμησης ιδιοτήτων (Property Evaluator). Το Communication Module είναι υπεύθυνο για την επικοινωνία σε επίπεδο εφαρμογών μεταξύ των διαφόρων κόμβων του GAS-OS. Η μονάδα αυτή μεταφράζει τις υψηλού επιπέδου αιτήσεις / αποκρίσεις σε μηνύματα και με τη χρήση χαμηλού επιπέδου δικτυακών πρωτοκόλλων τα αποστέλλει στους κατάλληλους ισότιμους κόμβους. Ο Process Manager αποτελεί τη συντονιστική μονάδα του διαμεσολαβητικού λογισμικού. Μερικές από τις σημαντικότερες εργασίες του είναι η διαχείριση των πολιτικών επεξεργασίας του τεχνουργήματος, η αποδοχή και εξυπηρέτηση αιτήσεων που τίθενται από άλλες μονάδες του πυρήνα ή η πυροδότηση αντιδράσεων σε συνεργασία με άλλες μονάδες, και γενικά εργασίες που συνολικά εξυπηρετούν την πραγματοποίηση του μοντέλου Συνδέσμων / Συνάψεων. Επιπλέον, είναι υπεύθυνος για τη διαχείριση της μνήμης του τεχνουργήματος και για την αποθήκευση της πρόσφατης πληροφορίας άλλων τεχνουργημάτων (caching) για τη βελτίωση της απόδοσης

επικοινωνίας στις περιπτώσεις που απαιτείται ανεύρεση υπηρεσιών. Ο State Variable Manager διαχειρίζεται τις μεταβλητές κατάστασης κατά το χρόνο εκτέλεσης, που αντανακλούν τόσο το περιβάλλον υλικού (αισθητήρες / ενεργοποιητές) σε κάθε χρονική στιγμή (πρωτογενείς ιδιότητες), όσο και τις ιδιότητες που εκτιμώνται με βάση δεδομένα αισθητήρων και δεδομένα p2p επικοινωνίας (σύνθετες ιδιότητες). Ο Property Evaluator είναι υπεύθυνος για την αποτίμηση των σύνθετων ιδιοτήτων του τεχνουργήματος σύμφωνα με το Συναρτησιακό του Σχήμα (Functional Schema). Στην τυπική του μορφή ο Property Evaluator βασίζεται σε ένα σύνολο κανόνων που διέπουν τη μετάβαση των τεχνουργημάτων από τη μια κατάσταση στην άλλη. Η διαχείριση κανόνων μπορεί να διαχωριστεί από την λογική αποτίμησης με τη χρήση μιας υψηλού επιπέδου γλώσσας κανόνων και έναν μεταφραστή που μεταφράζει τις υψηλού επιπέδου προδιαγραφές των κανόνων σε XML έτσι ώστε να μπορούν να αξιοποιηθούν από την λογική αποτίμησης (Christopoulou. 2005).

Ακολουθώντας μια αρθρωμένη σε επίπεδα αρχιτεκτονική, είναι δυνατή η αντικατάσταση μια μονάδας χωρίς αυτό να επηρεάζει τη λειτουργικότητα των υπολοίπων, εφόσον βέβαια τα APIs μεταξύ τους παραμένουν συνεπή. Η αρχή αυτή ισχύει για διάφορα επίπεδα της αρχιτεκτονικής όπως και μέσα σε κάθε επίπεδο. Για παράδειγμα, ο αρθρωτός σχεδιασμός του GAS-OS επιτρέπει την ενσωμάτωση ενημερωμένων αλγορίθμων και πρωτοκόλλων με τη μορφή πρόσθετων μονάδων (plugins) (Goumopoulos 2003).

Η επέκταση της λειτουργικότητας του πυρήνα του GAS-OS μπορεί να επιτευχθεί μέσω πρόσθετων μονάδων οι οποίες μπορούν να ενσωματωθούν εύκολα σε ένα τεχνουργήμα μέσω του Plug-in Manager. Χρησιμοποιώντας οντολογίες και το plug-in διαχειριστή οντολογιών τα τεχνουργήματα μπορούν να χρησιμοποιούν κοινά αποδεκτό και κατανοητό λεξιλόγιο υπηρεσιών και δυνατοτήτων προκειμένου να φιλτράρουν την ετερογένεια στο επίπεδο της αντίληψης του περιβάλλοντος και των μοντέλων πραγματικού κόσμου (Christopoulou. 2005).

Με τον τρόπο αυτό είναι δυνατές περιγραφές υψηλού επιπέδου υπηρεσιών και πόρων, ανεξαρτήτως του περιβάλλοντος συγκεκριμένης εφαρμογής, πράγμα που διευκολύνει την ανταλλαγή πληροφοριών μεταξύ ετερογενών τεχνουργημάτων, όπως και την ανεύρεση υπηρεσιών.

Η μονάδα ασφαλείας που εγκαθίσταται και αυτή ως plug-in όταν υλοποιηθεί θα είναι υπεύθυνη για την πραγματοποίηση των στρατηγικών ασφαλείας ενός τεχνουργήματος. Οι πολιτικές αυτές θα κωδικοποιηθούν σαν κανόνες της οντολογίας και θα γίνουν απευθείας διαθέσιμες στον Process Manager. Ο Security Manager θα μεσολαβεί στην ανταλλαγή.

4.5 Υλοποίηση του GAS-OS

Μονάδα Επικοινωνίας (Communication Module)

Η μονάδα επικοινωνίας CM είναι υπεύθυνη για την επικοινωνία μεταξύ των κόμβων του GAS-OS. Υλοποιεί αλγορίθμους και πρωτόκολλα για μετάδοση δεδομένων μέσω καναλιών επικοινωνίας που δημιουργούνται δυναμικά, όπως και μηχανισμούς για διακίνηση των πληροφοριών στο εσωτερικό του πυρήνα. (Goumoroulios 2003).

Η επικοινωνία μεταξύ τεχνουργημάτων διαχωρίζεται σε σύνδεσμο και υποκείμενη. Η επικοινωνία συνδέσμου-συνδέσμου λαμβάνει χώρα όταν δύο τεχνουργήματα ανταλλάσσουν πληροφορίες μέσω μιας σύναψης όπου η κατεύθυνση της ροής και ο τύπος των δεδομένων είναι καθορισμένα από τις ιδιότητες της σύναψης. Από την άλλη η υποκείμενη επικοινωνία λαμβάνει χώρα όταν δύο τεχνουργήματα ανταλλάσσουν πληροφορίες χωρίς την ύπαρξη καθορισμένου καναλιού, πράγμα που συνήθως συμβαίνει όταν τεχνουργήματα αιτούνται ή παραχωρούν πόρους σε άλλα τεχνουργήματα, αλλά και γενικά για εσωτερική επικοινωνία του καταναμημένου συστήματος (π.χ. ανεύρεση υπηρεσιών. ανεύρεση τεχνουργημάτων, επικοινωνία με μονάδα μάθησης).

Ένα σύνολο κόμβων του GAS-OS που ανταλλάσσουν πληροφορίες μέσω δυναμικών καναλιών συνιστούν ένα p2p δίκτυο του οποίου η τοπολογία αλλάζει δυναμικά (ad-hoc) (Goumoroulos 2003).

Σε p2p επικοινωνίες κάθε κόμβος αποτελεί μια αυτόνομη και «ομότιμη» (peer) με τους άλλους κόμβους διωκομένη μονάδα, και υλοποιεί ένα ή περισσότερα πρωτόκολλα δικτύου. Η ad-hoc φύση του δικτύου οφείλεται στη δυναμική αλλαγή της τοπολογίας του λόγω μη προβλέψιμων καταστάσεων που μπορεί να συμβούν λόγω της προσέλευσης ή αποχώρησης των τεχνουργημάτων από το δίκτυο για παράδειγμα λόγω της κινητικότητάς, της δυσλειτουργίας τους κτλ.

Δομή

Η μονάδα επικοινωνίας υλοποιήθηκε υιοθετώντας τις βασικές αρχές του JXTA (JXTA. 2007) και τις ιδιαιτερότητες της αρχιτεκτονικής του GAS-OS. «Peers», «ripes» και «endpoints» συνδυάζονται σε μια αρχιτεκτονική διαδοχικών επιπέδων που παρέχει οπτική κλιμακούμενης λεπτομέρειας καθ' όλη τη διαδικασία επικοινωνίας. Το CM είναι ουσιαστικά ένα μη-κεντροποιημένο σύστημα μηνυμάτων (δεν απαιτεί προκαθορισμένη υποδομή, ούτε την υποστήριξη άλλων βοηθητικών οντοτήτων πλέον των «ομότιμων» κόμβων), που απομονώνει υποκείμενο δίκτυο και πρωτόκολλα επικοινωνίας και παρέχει υπηρεσίες μέσω καλώς ορισμένων διεπαφών. Η επικοινωνία μεταξύ των «ομότιμων» κόμβων αλλά και η υποκείμενη επικοινωνία επιτυγχάνονται μέσω ασύγχρονων XML μηνυμάτων. Στην εικόνα 12 γίνεται μια αναλυτική παρουσίαση όλων των XML μηνυμάτων που χρησιμοποιούνται στο GAS-OS. Το CM διαιρείται σε τρεις επιμέρους υπομονάδες που καθεμία παρέχει διαφορετικό επίπεδο λεπτομέρειας καθ' όλη την διαδικασία της επικοινωνίας (JXTA 2007).

- Peer. Κάθε CM ενός κόμβου κατέχει ακριβώς ένα peer. Ένα peer αντιπροσωπεύει μια δικτυωμένη οντότητα που υλοποιεί ένα ή περισσότερα πρωτόκολλα επικοινωνίας. Συγκεκριμένα

υλοποιεί πρωτόκολλα για την ανεύρεση πόρων και υπηρεσιών, δημοσιοποίηση, δρομολόγηση αλλά και μηχανισμούς ουράς για την υποστήριξη ασύγχρονης ανταλλαγής μηνυμάτων. Για την αποφυγή μεγάλων μηνυμάτων και ως συνέπεια συμφόρηση του δικτύου, τα peers πακετάρουν τις πληροφορίες που απαιτούνται για κάθε πρωτόκολλο σε XML μηνύματα. Κάθε peer λειτουργεί αυτόνομα και ασύγχρονα από όλα τα άλλα και αναγνωρίζεται από ένα μοναδικό Peer ID. Το peer είναι επίσης το μέρος του CM που ορίζει τη διεπαφή με τον υπόλοιπο πυρήνα και συγκεκριμένα με τον PM. Έτσι από τη μια επιτρέπει στον PM να εγγραφεί ως ακροατής συμβάντων, προκειμένου να λαμβάνει ειδοποιήσεις για εισερχόμενα δεδομένα, και από την άλλη δέχεται πληροφορίες μέσω αναφοράς (by reference) από τον PM για αποστολή προς άλλοι peers (Horricks. 2003).

- **Pipe. Tu pipes**

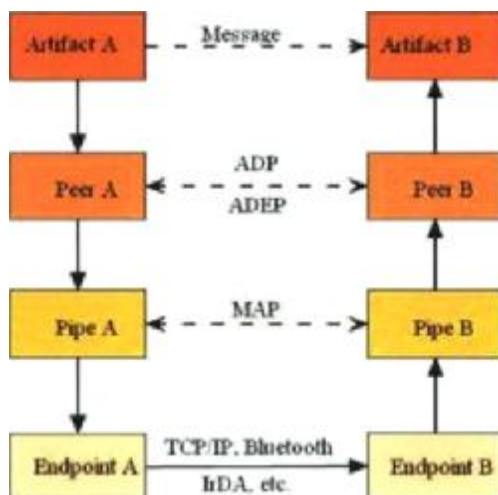
Αντιστοιχούν στα επίπεδα συνόδου (session) και παρουσίασης (presentation) του μοντέλου ISO-OSI, υλοποιώντας πρωτόκολλα για τη δημιουργία σύνδεσης μεταξύ δύο peers, πολλαπλή επικοινωνία (multicast) για την ανεύρεση υπηρεσιών και τεχνουργημάτων, ενώ ταυτόχρονα φροντίζει και για την αξιόπιστη παράδοση των μηνυμάτων. Σε περιπτώσεις δε που χρησιμοποιούνται αξιόπιστα δικτυακά πρωτόκολλα στο επίπεδο μεταφοράς (transport layer), όπως για παράδειγμα το TCP/IP, η λειτουργικότητα των pipes ανάγεται στη βεβαίωση της διαθεσιμότητας των πόρων για το επίπεδο εφαρμογών (application layer). Για παράδειγμα η αποστολή ενός μηνύματος αίτησης σύναψης σε ένα μη-συμβατό σύνδεσμο θα επιστρέφει ένα μήνυμα NACK. Κάθε peer κατέχει ακριβώς δύο pipes: ένα εισόδου που περιμένει για εισερχόμενα μηνύματα και ένα εξόδου που αποστέλλει μηνύματα είτε προς έναν (unicast) είτε προς πολλαπλούς (multicast) παραλήπτες στο δίκτυο.

Κάθε pipe κατέχει ένα ή δύο Endpoints. Ένα pipe εισόδου έχει δύο Endpoints, ένα για την ακρόαση multicast μηνυμάτων και ένα για την ακρόαση

p2p μηνυμάτων, ενώ ένα pipe εξόδου έχει ένα Endpoint το οποίο ρυθμίζεται δυναμικά να στέλνει είτε multicast μηνύματα σε μια προκαθορισμένη multicast ομάδα, είτε p2p μηνύματα. Τα Endpoints θεωρούνται θεμελιώδη δικτυακά στοιχεία της μονάδας επικοινωνίας και συσχετίζονται με συγκεκριμένους δικτυακούς πόρους όπως μια Ούρα TCP, μια διεύθυνση IP, ένα Bluetooth ID κτλ. Ανάλογα με την επιλεγμένη τεχνολογία για το επίπεδο μεταφοράς, μπορούμε να έχουμε πολλά διαφορετικά endpoints όπως για παράδειγμα βασισμένα σε IP, σε Bluetooth, σε IrDA κτλ., πράγμα που καθιστά τα endpoints κατάλληλα να χρησιμοποιηθούν και για τη γεφύρωση διαφορετικών δικτύων. Για παράδειγμα ένα peer που κατέχει ένα IP endpoint και ένα Bluetooth endpoint μπορεί να λαμβάνει δεδομένα από ένα IP δίκτυο και να τα προωθεί σε ένα δίκτυο Bluetooth.

Πρωτόκολλα

Το CM ορίζει ένα σύνολο από XML μηνύματα και πρωτόκολλα που χρησιμοποιούνται για την επικοινωνία των τεχνουργημάτων, και συγκεκριμένα για την ανεύρεση και δημοσιοποίηση τεχνουργημάτων, δικτυακών πόρων, τη δρομολόγηση μηνυμάτων και την ανταλλαγή πληροφοριών. Υπάρχουν τρία βασικά και δύο συμπληρωματικά πρωτόκολλα επικοινωνίας (Goumopoulos 2003]:



Εικόνα12: Πρωτόκολλα επικοινωνίας τεχνουργημάτων

Πρωτόκολλο Ανεύρεσης Τεχνουργημάτων - Artifact Discovery Protocol (ADP). Το πρωτόκολλο αυτό χρησιμοποιείται από τα peers για την ανεύρεση τεχνουργημάτων ή πόρων στο δίκτυο χρησιμοποιώντας κριτήρια. Τα κριτήρια αφορούν συνήθως την ταυτότητα ενός απομακρυσμένου τεχνουργήματος, για παράδειγμα ID, ή τους πόρους του, για παράδειγμα συνδέσμους, αισθητήρες, ενεργοποιητές κτλ. Ένα τεχνούργημα δημιουργεί ένα μήνυμα ανεύρεσης το οποίο και το αποστέλλει προς όλα τα τεχνουργήματα που ακούν μέσω ενός multicast endpoint. Μια ειδική μορφή του μηνύματος επιτρέπει την ανεύρεση όλων των τεχνουργημάτων του περιβάλλοντος. Όταν ένα τεχνούργημα λάβει ένα μήνυμα ανεύρεσης το αναλύει, ελέγχει αν ικανοποιεί τα κριτήρια αναζήτησης και αποκρίνεται με ένα μήνυμα δημοσιοποίησης. Ένα μήνυμα δημοσιοποίησης μπορεί να περιέχει είτε την ταυτότητα του τεχνουργήματος (TPlug) ή χαρακτηριστικά του τεχνουργήματος που ικανοποιούν τα κριτήρια της ανεύρεσης.

Πρωτόκολλο Ανταλλαγής Δεδομένων Τεχνουργημάτων - Artifact Data- Exchange Protocol (ADEP). Το πρωτόκολλο αυτό χρησιμοποιείται από τα peers για την ανταλλαγή δεδομένων μεταξύ τεχνουργημάτων, υπό την προϋπόθεση ότι έχει δημιουργηθεί ήδη η σύναψη. Η επικοινωνία γίνεται με τη χρήση ενός απλού unicast ripe, ενώ κάθε μήνυμα δεδομένων πακετάρετε σε μια φόρμα - «φάκελο» που περιέχει τα αναγνωριστικά του παραλήπτη (Horricks. 2003).

Πρωτόκολλο Αναφοράς Μηνυμάτων - Message Acknowledging Protocol (MAP)

Αυτό το πρωτόκολλο χρησιμοποιείται από τα ripes και είναι υπεύθυνο για την σταθερότητα του συστήματος επικοινωνίας. Κύρια λειτουργία του είναι η παροχή θετικών αναφορών (ACK) στον αποστολέα για κάθε εισερχόμενο μήνυμα και αρνητικών αναφορών (NACK) για κάθε χαμένο μήνυμα. Επιπλέον αν ένα

μήνυμα χαθεί το πρωτόκολλο είναι υπεύθυνο για την επαναποστολή του μηνύματος στον τεχνουργήματος παραλήπτη.

Η Εικόνα 12 συνοψίζει τη διαδικασία επικοινωνίας ανάμεσα στα τεχνουργήματα A και B με τη χρήση των πρωτοκόλλων επικοινωνίας.

Υπάρχουν τρία ακόμα συμπληρωματικά πρωτόκολλα η χρήση των οποίων μπορεί να βελτιστοποιήσει την απόδοση του δικτύου και να ενισχύσει τις δυνατότητες των τεχνουργημάτων.

Πρωτόκολλο Μάθησης - Learning Protocol (LP)

Το πρωτόκολλο μάθησης καθορίζει τον τρόπο με τον οποίο η μονάδα μάθησης παρακολουθεί τον τρόπο χρήσης μιας εφαρμογής περιρρέουσας νοημοσύνης και επηρεάζει τον τρόπο αλληλεπίδρασης των τεχνουργημάτων σύμφωνα με τις προτιμήσεις των χρηστών σε συνεργασία με τον πυρήνα του GAS-OS. Όπως ήδη αναφέρθηκε η παραδοχή που ακολουθείται είναι ότι μόνο τεχνουργήματα που συνδέονται μεταξύ τους με συνάψεις ανήκουν σε ένα σύνολο, δηλαδή ένα σύνολο είναι ένα συνεκτικό δίκτυο από συνδεδεμένα με συνάψεις τεχνουργήματα. Το πρωτόκολλο καθορίζει τη συνεργασία του GAS-OS με τη μονάδα μάθησης για τις παρακάτω περιπτώσεις δημιουργίας νέας σύναψης: Το τεχνουργήμα που ήδη συμμετέχει σε ένα σύνολο και γνωρίζει την ύπαρξη τεχνουργήματος με μονάδα μάθησης μέσα στο σύνολο, δημοσιεύει την ύπαρξη της δυνατότητας αυτής στο τεχνουργήμα με το οποίο συνάπτεται. Κάθε νεοεισαχθέν στο σύνολο τεχνουργήμα που λαμβάνει την πληροφορία της μονάδας μάθησης, αποστέλλει τις μεταβλητές κατάστασής του στο τεχνουργήμα που την κατέχει. Αυτό έχει ως αποτέλεσμα την τροποποίηση της βάσης κανόνων, καθώς οι νέες αυτές μεταβλητές κατάστασης των νέων τεχνουργημάτων, προστίθενται στα ηγούμενα και επακόλουθα μέρη των κανόνων. Οι υπάρχοντες κανόνες δεν τροποποιούνται. Μόνο νέοι κανόνες που προστίθενται από το σημείο αυτό και μετά εκτιμούν τις νέες μεταβλητές ως σημαντικές για την εκτέλεσή τους.

- Καταστροφή υπάρχουσας σύναψης ή αποχώρηση τεχνουργήματος: Το τεχνουργήμα που παραμένει στο σύνολο

μετά την καταστροφή της σύναψης ειδοποιεί το τεχνούργημα που κατέχει τη μονάδα μάθησης για την αλλαγή του δικτύου συνάψεων, το οποίο με τη σειρά του εκτελώ τη φάση αρχικοποίησης για να ανιχνεύσει εκ νέου το δίκτυο μέσω των συνάψεων. Αυτό σημαίνει επαναρχικοποίηση της βάσης κανόνων, που έχει ως αποτέλεσμα την αγνόηση των απορριφθέντων τεχνουργημάτων στην επόμενη εκτέλεση των κανόνων και η θεώρηση των μεταβλητών κατάστασής τους ως μη σημαντικές. Το πρόβλημα που προκύπτει στην περίπτωση αυτή είναι ότι μπορεί να εμφανιστούν πολλοί κανόνες του ίδιου τύπου. Για το λόγο αυτό όταν η εγκυρότητα των δεδομένων ή η εφαρμογή ενός κανόνα είναι αβέβαιη, χρησιμοποιούνται παράγοντες βεβαιότητας ή εμπιστοσύνης ως προσδιοριστές των κανόνων.

- Παρακολούθηση τρόπου χρήσης εφαρμογής: Η μονάδα μάθησης πρέπει να ενημερώνεται για κάθε αλλαγή που συμβαίνει σε μια σύναψη. Όταν ένα τεχνούργημα δέχεται δεδομένα μέσω μιας σύναψης, χρησιμοποιεί την επιλεγμένη λογική του για να αντιδράσει με ευθύνη του Process Manager PM. Εάν η αντίδρασή του υπαγορεύει την αλλαγή της κατάστασης κάποιου ενεργοποιητή τότε ενημερώνεται κατάλληλα η μονάδα διαχείρισης μεταβλητών κατάστασης SVM η οποία και αλλάζει τον ενεργοποιητή Παράλληλα με την SVM. Ο PM αποστέλλει και μια ειδοποίηση στη μονάδα μάθησης για να την ενημερώσει για την αλλαγή. Είναι πιθανό η λογική που εφάρμοσε το τεχνούργημα, επομένως και η αντίδρασή του, να μην ικανοποιήσει τους χρήστες του συστήματος. Σε τέτοια περίπτωση, ένας χρήστης μπορεί να παρέμβει και να αλλάξει το αποτέλεσμα της αντίδρασης, για παράδειγμα επηρεάζοντας με κάποιον τρόπο τον ενεργοποιητή (π.χ. αλλαγή με το χέρι ενός διακόπτη). Αν κάτι τέτοιο συμβεί, η αλλαγή που προκαλεί ο

χρήστης γίνεται αντιληπτή από την SVM η οποία ειδοποιεί τη μονάδα μάθησης μέσω του PM. Η μονάδα μάθησης λαμβάνοντας τέτοιου είδους δεδομένα μπορεί να παράγει κανόνες χρήσης του συστήματος οι οποίοι όταν εφαρμόζονται μπορούν να καθορίζουν τη λογική του τεχνουργήματος.

ΚΕΦΑΛΑΙΟ 5

ΑΣΥΡΜΑΤΑ ΔΙΚΤΥΑ ΑΙΣΘΗΤΗΡΩΝ

5.1 Δίκτυα Αισθητήρων και Διάχυτος Προγραμματισμός

Τα τελευταία χρόνια μια τεράστια επανάσταση βρίσκεται σε εξέλιξη, η οποία έχει μεταβάλλει τον τρόπο με τον οποίο οι άνθρωποι αντιμετωπίζουν την τεχνολογία. Κοιτώντας κανείς γύρω του μπορεί να δει πανίσχυρες ηλεκτρονικές συσκευές, οι οποίες είναι ικανές να εκτελέσουν σύνθετες υπολογιστικές λειτουργίες, αλλάζοντας τον τρόπο που οι άνθρωποι δουλεύουν, σπουδάζουν και ζουν.

Η επανάσταση αυτή στηρίζεται σε τρεις, ανεξάρτητους πυλώνες, τις τηλεπικοινωνίες με την τεράστια πρόοδο που έχουν σημειώσει οι ασύρματες ειδικότερα τα τελευταία χρόνια, τις τεχνικές καταμεμημένου προγραμματισμού, και την ηλεκτρονική με την κατασκευή πανίσχυρων χαμηλού κόστους ηλεκτρονικών υποσυστημάτων για χρήση σε κάθε είδους σύστημα. (Akyildiz, & Melodia 2008)

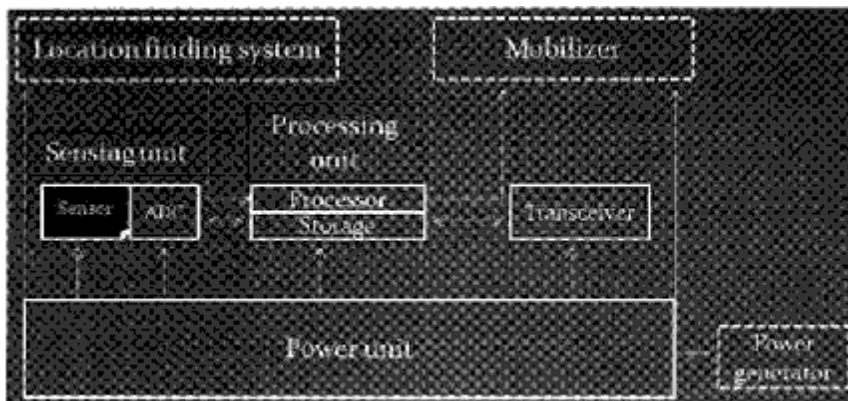
Ο πρώτος ο οποίος μίλησε για την ιδέα του διάχυτου προγραμματισμού ήταν ο Mark Weiser το 1988 όταν ήταν επικεφαλής στο PARC, ένα ερευνητικό κέντρο της Xerox στο Palo Alto των ΗΠΑ. Σε ένα άρθρο της εποχής εκείνης ο Weiser διατύπωσε την άποψη ότι σκοπός των μηχανικών, θα πρέπει να είναι

όχι η κατασκευή υπολογιστικών συστημάτων τόσο ισχυρών ώστε να μην μπορούμε να απομακρυνθούμε από αυτά, αλλά στην ανάπτυξη αόρατων συστημάτων που να κατακλύζουν τον χώρο που διαβιούμε και εργαζόμαστε, και που ενώ θα μπορούμε να τα χρησιμοποιούμε να μην αντιλαμβανόμαστε ότι πράττουμε κάτι τέτοιο.

Είκοσι χρόνια σχεδόν μετά, το όραμα του Weiser αρχίζει να υλοποιείται, και η υλοποίηση αυτή συμβαίνει σε ένα από τα πολλά εργαστήρια και ερευνητικά κέντρα που ασχολούνται με αυτή την ιδέα αλλά στην καθημερινή ζωής μας (π.χ. δίκτυα αισθητήρων αυτοκινήτων). Σε αυτή την διαδικασία της υλοποίησης του οράματος του Weiser τα δίκτυα αισθητήρων παίζουν κεντρικό ρόλο. Η επανάσταση που συντελείται σε αυτά μπορεί να εξαφανίσει τα σημερινά υπολογιστικά συστήματα από το προσκήνιο, και να τα αντικαταστήσει με κάθε λογής «έξυπνα» αντικείμενα, τα οποία θα είναι σε θέση να προγραμματίζονται, να σχηματίζουν δίκτυα μεταξύ τους, και να φροντίζουν για την ικανοποίηση των αναγκών των ιδιοκτητών, χωρίς οι ίδιοι να απαιτείται να προβούν σε καμιά ενέργεια. (Akyildiz, & Melodia 2008).

5.2. Συνοπτική Περιγραφή των Ασύρματων Δικτύων Αισθητήρων

Σύμφωνα με το άρθρο «10 Emerging Technologies That Will Change the World» του MIT Technology Review , το οποίο δημοσιεύτηκε στο διαδίκτυο το 2003, η πρώτη από τις δέκα αναπτυσσόμενες τεχνολογίες οι οποίες πιθανολογείται ότι θα προκαλέσουν τεχνολογική επανάσταση στο μέλλον είναι τα ασύρματα δίκτυα, τα οποία αποτελούνται από κόμβους (nodes) μικρών διαστάσεων και χαμηλού κόστους με ενσωματωμένους αισθητήρες (sensors), για την καταγραφή και παρακολούθηση μετρήσιμων παραμέτρων όπως θερμοκρασία, ήχος, δόνηση, πίεση, κίνηση και ρύπανση. Τα παραπάνω δίκτυα ονομάζονται Ασύρματα Δίκτυα Αισθητήρων (Wireless Sensor Networks. WSNs) (Akyildiz, & Melodia 2008).



Εικόνα 13 Τα επιμέρους υποσυστήματα ενός SN.

Τα WSNs μπορούν να περιλαμβάνουν από μερικές δεκάδες ως και αρκετές χιλιάδες κόμβους αισθητήρων (Sensor Nodes, SNs). Τα SNs είναι συνήθως τυχαία διασπαρμένα στο πεδίο και σχηματίζουν μεταξύ τους ένα αυτό-διαμορφούμενο, αδόμητο (ad-hoc) δίκτυο στο οποίο γίνονται ασύρματες ζεύξεις μεταξύ γειτονικών SNs. Κύριο χαρακτηριστικό αυτών των δικτύων είναι ότι τα SNs που βρίσκονται εκτός εμβέλειας, μεταξύ τους, μπορούν να επικοινωνήσουν ασύρματα χρησιμοποιώντας ενδιάμεσα SNs για την προώθηση των μηνυμάτων (ασύρματη επικοινωνία πολλαπλών βημάτων, multi-hop wireless communication) (Al-Karaki & Ahmed E. Kamal, 2004).

Κάθε SN αποτελείται από τα παρακάτω: υποσύστημα επικοινωνίας (radio unit ή transceiver), υποσύστημα επεξεργασίας (processing unit) το οποίο ενσωματώνει μικροεπεξεργαστή (processor) και μνήμη (storage) επεξεργασίας και αποθήκευσης δεδομένων, υποσύστημα αισθητήρων για συλλογή δεδομένων (sensing unit) και υποσύστημα τροφοδοσίας (power unit) το οποίο συνήθως αποτελείται από μια μικρή μπαταρία, η οποία παρέχει μια περιορισμένη ενεργειακή αυτονομία (Al-Karaki and Ahmed E. Kamal, 2004).

Το υποσύστημα αισθητήρων σε ένα SN εκτός από τους αισθητήρες περιλαμβάνει και ένα κύκλωμα το οποίο μετατρέπει τα δεδομένα που συλλέγονται από τους αισθητήρες από την αναλογική τους μορφή σε ψηφιακή

(Analog to Digital Converter, ADC), έτσι ώστε να είναι δυνατή η επεξεργασία τους από τον επεξεργαστή του SN πριν την αποστολή τους προς μία κεντρική βάση δεδομένων μέσω του ασύρματου δικτύου. Επιπρόσθετα ένα SN μπορεί να είναι εξοπλισμένο με τα παρακάτω: υποσύστημα εύρεσης γεωγραφικού στίγματος (location finding system) το οποίο βοηθάει στην εύρεση της ακριβούς θέσης του SN μέσα στην περιοχή ανάπτυξης του δικτύου, υποσύστημα κίνησης (mobilizer) ώστε να μπορεί να μετακινείται και υποσύστημα αναπλήρωσης ενέργειας (power generator) για την επαναφόρτιση της μπαταρίας του. (Al-Karaki and Ahmed E. Kamal, 2004).

Βασικός στόχος ενός WSN είναι η συλλογή και μεταφορά δεδομένων από τα SNs σε μια κεντρική βάση συγκέντρωσης (sink ή Processing Element, PE) και επεξεργασίας τους. Τα δεδομένα που συλλέγονται από τα SNs μπορούν να προ-επεξεργαστούν από τα ίδια έστω και με τις μειωμένες υπολογιστικές δυνατότητες που αυτά διαθέτουν (λόγω χαμηλής υπολογιστικής ισχύος επεξεργαστή και περιορισμένης μνήμης και ενέργειας) πριν σταλούν προς το PE, έτσι ώστε να μειωθεί η ποσότητα της διακινούμενης πληροφορίας και να μην επιβαρύνεται το VWSN με περιττό δικτυακό φορτίο. Ένα σημαντικό πλεονέκτημα των WSNs αποτελεί το ότι ακόμη και στην περίπτωση που καταστραφούν κάποια SNs ή καταναλωθεί η διαθέσιμη ενέργεια τους, τα υπόλοιπα με τη χρήση κατάλληλων αλγορίθμων δρομολόγησης έχουν τη δυνατότητα να προδιαγράψουν εναλλακτικές διαδρομές και να στείλουν τα συλλεγμένα δεδομένα προς το PE. Τέλος το PE μπορεί να ελέγχεται απ' ευθείας από το χρήστη είτε από ένα τοπικό τερματικό, είτε από ένα απομακρυσμένο τερματικό χρησιμοποιώντας στη δεύτερη περίπτωση το διαδίκτυο, κάποια δορυφορική σύνδεση ή κάποια άλλη τεχνολογία απομακρυσμένης επικοινωνίας. (Jiangi & Manivannan 2004).

5.3. ΔΙΚΤΥΑ ΑΙΣΘΗΤΗΡΩΝ (WSNs): ΣΥΝΤΟΜΗ ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ, ΕΦΑΡΜΟΓΕΣ ΚΑΙ ΕΡΕΥΝΗΤΙΚΕΣ ΠΕΡΙΟΧΕΣ

5.3.1 Η Σύντομη Ιστορία των WSNs

Όπως έχει συμβεί και με πλήθος άλλων τεχνολογιών η «γέννηση» και η εξέλιξη των δικτύων αισθητήρων, πρόγονων των WSNs, προήλθε από στρατιωτικά ερευνητικά προγράμματα. Η πρώτη γνωστή εφαρμογή δικτύων αισθητήρων υπήρξε το SOSUS (Sound Surveillance System, Σύστημα Ηχητικής Παρακολούθησης) . Το παραπάνω δίκτυο χρησιμοποιήθηκε στις αρχές της δεκαετίας του 1950, κατά τη διάρκεια του ψυχρού πολέμου, για την ανίχνευση και τον εντοπισμό Σοβιετικών υποβρυχίων με τη βοήθεια ειδικών ακουστικών αισθητήρων (υδρόφωνα). Το SOSUS παραμένει ακόμη σε λειτουργία, για ειρηνικούς σκοπούς και χρησιμοποιείται κυρίως για την παρακολούθηση διάφορων φαινομένων όπως σεισμικές δραστηριότητες, παρακολούθηση φαλαινών κ.λπ. (Noury 1994).

Το επόμενο δίκτυο αισθητήρων αναπτύχθηκε επίσης για στρατιωτική χρήση. Γύρω στο 1980 το πρόγραμμα Distributed Sensor Networks (DSN. Κατανεμημένο Δίκτυο Αισθητήρων) ξεκίνησε από την Defense Advanced Research Projects Agency (DARPA) [36]. Ο πρόγονος της DARPA ήταν η Advanced Research Projects Agency (ARPA) που είχε δημιουργήσει παλαιότερα (1969) το δίκτυο ARPANET, το οποίο υπήρξε ο πρόγονος του σημερινού Διαδικτύου (Internet). Η δυνατότητα να επεκταθεί το ARPANET στα δίκτυα αισθητήρων απασχόλησε σε μεγάλο βαθμό τον R. Kahn (συν-εφευρέτη του πρωτοκόλλου TCP/IP και με καθοριστικό ρόλο στη δημιουργία και ανάπτυξη του διαδικτύου), ταυτόχρονα με την έρευνα σε εμπειρικές τεχνικές ανάλυσης ακουστικού σήματος. (Noury 1994).

Η ραγδαία ανάπτυξη των ασύρματων επικοινωνιών, η τεχνολογία των μικρομηχανικών συστημάτων (Micro-Electro-Mechanical Systems, MEMS) και η δυνατότητα κατασκευής χαμηλής κατανάλωσης-κόστους μικροεπεξεργαστών επέτρεψε την ανάπτυξη των δικτύων που σήμερα ονομάζονται WSNs. Η πρόοδος της τεχνολογίας συμπιέζει το κόστος παραγωγής των SNs και το μέγεθος τους, με αποτέλεσμα να γίνεται ολοένα και πιο ελκτική η ανάπτυξη

μεγάλης κλίμακας WSNs αποτελούμενων από χιλιάδες SNs. Τέτοιου είδους WSNs θα μπορούν να καλύψουν ολόκληρες πόλεις ή μεγάλες δασικές εκτάσεις και θα έχουν τη δυνατότητα να παρέχουν κάθε είδους πληροφορία στους χρήστες τους.(Matsumoto 1998).

5.3.2 Εφαρμογές των WSNs

Στις επόμενες παραγράφους θα αναφερθούν κάποιες αντιπροσωπευτικές εφαρμογές των WSNs οι οποίες είτε έχουν προταθεί σε ερευνητικό επίπεδο, είτε έχουν ήδη υλοποιηθεί σε σημερινές εμπορικές εφαρμογές. Μεταξύ άλλων οι εφαρμογές αυτές είναι: στρατιωτικές, περιβαλλοντικές, υγείας και άλλες διάφορες εμπορικές εφαρμογές (βιομηχανία, κατασκευές κ.λπ.). Εδώ να τονιστεί ότι η χρήση των WSNs δεν περιορίζεται φυσικά μόνο στους παραπάνω τομείς αλλά επεκτείνεται σε ένα πολύ ευρύτερο φάσμα δραστηριοτήτων της σύγχρονης ζωής. Στρατιωτικές εφαρμογές: Ένας σημαντικός όγκος έρευνας που σχετίζεται με WSNs αφορά τη χρησιμοποίησή τους σε στρατιωτικές εφαρμογές τα WSNs προήλθαν από στρατιωτικά ερευνητικά προγράμματα. Η ταχεία εγκατάσταση, η δυνατότητα αυτό-οργάνωσης και τα μεγάλα ποσοστά ανοχής σε πιθανή βλάβη των SNs καθιστούν τα WSNs ιδανικά για χρησιμοποίησή σε συστήματα διοίκησης, ελέγχου, επικοινωνιών, πληροφορικής, πληροφοριών, επιτήρησης, αναγνώρισεων και σκόπευση (Jiangi & Manivannan 2004).

Εφόσον τα WSNs αποτελούνται από πλήθος χαμηλού κόστους SNs, τα οποία διασκορπίζονται στην περιοχή ενδιαφέροντος με σχετική μεγάλη πυκνότητα, ακόμη και αν καταστραφεί ένα μέρος αυτών από εχθρικές ενέργειες τα υπόλοιπα μπορούν να συνεχίσουν κανονικά το έργο τους χωρίς να επηρεαστεί σημαντικά η συνολική απόδοση του δικτύου. Κάτι τέτοιο αποδεικνύεται ιδιαίτερα χρήσιμο σε ένα πεδίο μάχης, όπου η συνεχής ροή πληροφοριών μπορεί να αποβεί καθοριστική για την έκβαση της. Μερικά παραδείγματα χρήσης των WSNs σε στρατιωτικές εφαρμογές είναι η παρακολούθηση των φιλιών δυνάμεων (έλεγχος ζωτικών λειτουργιών των

στρατιωτών και έλεγχος εξοπλισμού-ποσότητας πυρομαχικών), η παρακολούθηση του πεδίου μάχης και των γεωγραφικών ιδιαιτεροτήτων του, η αναγνώριση εχθρικών δυνάμεων, για την έγκαιρη προειδοποίηση επίθεσης από χημικά, βιολογικά, πυρηνικά όπλα και η εκτίμηση φιλιών και εχθρικών απωλειών (Jiangi & Manivannan 2004).

Περιβάλλον

Μερικές από τις περιβαλλοντικές εφαρμογές που μπορούν να έχουν τα WSNs είναι η παρακολούθηση πτηνών, άγριων ζώων και εντόμων, στον τομέα της γεωργίας για την παρακολούθηση διάφορων παραγόντων που επηρεάζουν τις καλλιέργειες, η παρακολούθηση και η προστασία μιας ευρύτερης γεωγραφικά περιοχής όπως ένα δάσος, η προστασία από χημικά-βιολογικά απόβλητα υδάτινων πόρων, η επιτήρηση εδάφους, υπεδάφους και ατμόσφαιρας μέσω της καταγραφής περιβαλλοντολογικών μετρήσεων και η παρακολούθηση της ατμοσφαιρικής ρύπανσης. Παρακάτω θα αναλυθεί ένα σενάριο ανάπτυξης ενός WSN σε ένα δάσος για εφαρμογές πυροπροστασίας. Τα SNs σε ένα δάσος μπορούν να τοποθετηθούν είτε τυχαία είτε σε συγκεκριμένα σημεία ενδιαφέροντος (π.χ. δυσπρόσιτες περιοχές). Στην περίπτωση που κάποιος αισθητήρας μετρήσει μια απότομη μεταβολή στη θερμοκρασία ή ένα υψηλό επίπεδο φωτός (κατά τη διάρκεια της νύχτας) προωθεί αμέσως ένα μήνυμα συναγερμού προς το PE το οποίο με τη σειρά του ενημερώνει αυτόματα την πυροσβεστική υπηρεσία. Έτσι κερδίζεται πολύτιμος χρόνος για την κινητοποίηση επίγειων και εναέριων μέσων πυρόσβεσης. Σε μια τέτοια εφαρμογή, επειδή τα SNs πρέπει να είναι σε θέση να δουλεύουν αυτόνομα για όσο το δυνατόν μεγαλύτερο χρονικό διάστημα κρίνεται επιτακτική η ανάγκη επαναφόρτισης των μπαταριών τους (π.χ. με χρήση ηλιακής ενέργειας).

Υγεία

Τα WSNs μπορούν να χρησιμοποιηθούν για τη διευκόλυνση ατόμων με ειδικές ανάγκες (π.χ. ενημέρωση τυφλών για πιθανά επικείμενα εμπόδια), παρακολούθηση ασθενών, επίβλεψη ηλικιωμένων ατόμων στο σπίτι, διάγνωση ασθενειών, επίβλεψη φαρμακευτικής αγωγής σε νοσοκομεία, εντοπισμός γιατρών και ασθενών σε χώρους μεγάλων νοσοκομείων. (Batalin 2005) Στην περίπτωση χρήσης ενός WSN σε ένα μεγάλο νοσοκομείο, οι γιατροί και κάθε ασθενής θα φέρουν από ένα μικρό και ελαφρύ SN πάνω τους. Κάθε SN θα έχει συγκεκριμένο έργο αναλόγως το άτομο που θα το φέρει. Για παράδειγμα σε έναν ασθενή θα μπορεί να παρακολουθεί τον καρδιακό ρυθμό ενώ σε έναν άλλον την αρτηριακή πίεση. Ο αισθητήρας που θα φέρει ο κάθε γιατρός θα επιτρέπει στους συνάδελφους τους να τον εντοπίσουν γρήγορα, εντός των εγκαταστάσεων του νοσοκομείου, σε περίπτωση ανάγκης. (Batalin 2005)

«Έξυπνο» Σπίτι

Οι εφαρμογές που μπορούν να έχουν τα WSNs σε περιβάλλοντα σπιτιού είναι πρακτικά απεριόριστες. Ενδεικτικά μερικές εφαρμογές είναι ο αυτοματισμός διάφορων οικιακών συσκευών με την εμφύτευση μικροσκοπικών SNs, τα οποία θα τους δίνουν τη δυνατότητα να αλληλοεπιδρούν μεταξύ τους αλλά και να συνδέονται με εξωτερικά δίκτυα όπως το διαδίκτυο . Έτσι ένα «έξυπνο» ψυγείο, αφού προηγουμένως έχει ενημερωθεί για τις προτιμήσεις του ιδιοκτήτη του, θα μπορεί να παραγγέλνει τα τρόφιμα που εντοπίζει ότι λείπουν, η ηλεκτρική σκούπα θα καθαρίζει αυτόνομα τις ώρες που δεν βρίσκεται κανείς στο σπίτι και ο χρήστης θα μπορεί μέσω του διαδικτύου να δίνει εντολές σε διάφορες συσκευές (π.χ. το σύστημα κλιματισμού) αλλά και να ενημερώνεται για την κατάσταση αυτών, είτε αυτόματα είτε κατά βούληση. Επίσης, τα WSNs σε οικιακά περιβάλλοντα θα μπορούν να προσαρμόζονται στις ανάγκες των χρηστών και να δρουν ανάλογα . Για παράδειγμα, όταν ο χρήστης επιστρέφει το

βράδυ στο σπίτι τα φόντα θα ανάβουν αυτόματα ενώ ταυτόχρονα, αν ο ίδιος το επιθυμεί, τηλεόραση ή οποιαδήποτε άλλη συσκευή ήχου θα ενεργοποιείται στο κανάλι της επιλογής του. Το βράδυ, κατά την είσοδο του σε επιλεγμένα το WSN θα ανάβει αυτόματα το φως και μετέπειτα θα έχει τη δυνατότητα να το σβήνει, όταν διαπιστώνει ότι δεν βρίσκεται πλέον κανείς στο δωμάτιο. Επίσης το WSN θα μπορεί να ελέγχει το επίπεδο φωτισμού και θέρμανσης, για την επίτευξη της μέγιστης δυνατής οικονομίας και άνεσης και επιπλέον θα επιτηρεί τη στάθμη του πετρελαίου της κεντρικής θέρμανσης, έτσι ώστε να ειδοποιεί άμεσα για αναπλήρωση όταν αυτό κριθεί απαραίτητο (Batalin 2005).

Εμπορικές Εφαρμογές των WSNs Τα WSNs μπορούν να χρησιμοποιηθούν σε πλήθος εμπορικών εφαρμογών όπως η καταγραφή και παρακολούθηση της αποχής υλικών, η παρακολούθηση βιομηχανικών μηχανών και δομικών συσκευών, η παρακολούθηση της ποιότητας προϊόντων σε γραμμές βιομηχανικής παραγωγής, στη δημιουργία «έξυπνων» περιβαλλόντων εργασίας, στον κλιματικό έλεγχο κτιρίων, στην καθοδήγηση και έλεγχο μηχανών και ρομπότ σε αυτοματοποιημένα βιομηχανικά περιβάλλοντα, σε «έξυπνα» παιχνίδια που έχουν τη δυνατότητα αλληλεπίδρασης με τους χρήστες, σε αρχαιολογικούς χώρους και μουσεία για την παρακολούθηση των εκθεμάτων και την αλληλεπίδραση με τους επισκέπτες, για παρακολούθηση περιοχών όπου συνέβησαν καταστροφές, στα μέσα μεταφοράς (για την επίτευξη καλύτερου ελέγχου κυκλοφορίας), στην προστασία υλικής περιουσίας, στον εντοπισμό και παρακολούθηση οχημάτων, στην παρακολούθηση αποθηκών και εμπορευμάτων κ.λπ. (Gavalas 2005).

Ερευνητικές Περιοχές των WSNs: Η ανάπτυξη των WSNs στηρίζεται σε ένα ευρύ φάσμα επιμέρους τεχνολογιών όπως υλικό (hardware) των SNs, λογισμικό (software) και τις δικτυακές-ασύρματες επικοινωνίες. Παρακάτω θα εξεταστούν οι περιοχές που παρουσιάζουν μεγάλο ερευνητικό ενδιαφέρον στο χώρο των WSNs και τα σημαντικότερα προβλήματα που απασχολούν σήμερα τους ερευνητές. Η έρευνα στα WSNs θα διαχωριστεί σε δύο μεγάλες κατηγορίες: αυτήν που εστιάζει στην αρχιτεκτονική συστήματος και σε αυτήν που ασχολείται με τον έλεγχο του δικτύου και των επικοινωνιών (Batalin 2005).

5.4 Αρχιτεκτονική Συστήματος

Παρακάτω θα περιγραφούν οι ερευνητικές τάσεις στην αρχιτεκτονική συστήματος των WSNs.

Περιλαμβάνει τους εξής τομείς: λειτουργικά συστήματα και υλικό των SNs, εύρεση θέσης των SNs (localization), χωροχρονική συσχέτιση (spatial temporal correlation), βέλτιστη τοποθέτηση SNs στο πεδίο-χώρο, χρονικό συγχρονισμό (time synchronization) των SNs, ασφάλεια WSNs και μυστικότητα (privacy) δεδομένων που καταγράφονται από αυτά, εντοπισμός και παρακολούθηση στόχων και τέλος γλώσσες προγραμματισμού που χρησιμοποιούνται στα SNs (Batalin 2005).

Λειτουργικά Συστήματα και Υλικό

Πολλές ερευνητικές ομάδες, τόσο στην ακαδημαϊκή κοινότητα όσο και στη βιομηχανία, συμμετέχουν ενεργά στην ανάπτυξη του υλικού των SNs. Σκοπός τους είναι η περαιτέρω μείωση του μεγέθους τους, ταυτόχρονα με τη συμπίεση του κόστους κατασκευής. Ο πιο διαδεδομένος τύπος SN σήμερα είναι οι κόμβοι Berkeley, οι οποίοι και αναπτύχθηκαν αρχικά στο ομώνυμο πανεπιστήμιο. Δυναμική είσοδο στο χώρο κατασκευής SNs έχει πραγματοποιήσει και η εταιρία SUN με τα SunSpots 1199] τα οποία αν και έχουν αρκετά υψηλότερο κόστος από τους κόμβους Berkeley διαθέτουν πολύ μεγαλύτερες δυνατότητες και μεγαλύτερη επεκτασιμότητα (προσθήκη διάφορων τύπων αισθητήρων, μνήμης επέκτασης κ.λπ.). Στον τομέα του λογισμικού ο πιο μεγάλος όγκος των εφαρμογών που «τρέχουν» σήμερα στα δίκτυα αισθητήρων είναι συμβατός με το Tiny-OS ένα λειτουργικό σύστημα που σχεδιάστηκε ειδικά

για δίκτυα του συγκεκριμένου τύπου. Το Tiny-OS προσφέρει σχετικά μεγάλη ευελιξία για την ανάπτυξη εφαρμογών, παρά τις περιορισμένες δυνατότητες των SNs. Γενικά παρέχει στους προγραμματιστές ένα σύνολο από βασικές υπηρεσίες οι οποίες μπορούν να φανούν χρήσιμες για την υλοποίηση καταναμημένων αλγορίθμων. Άλλο ένα σημαντικό πλεονέκτημα του Tiny-Os είναι το ότι αποτελεί ένα δωρεάν και ανοιχτού κώδικα (open source) λειτουργικό σύστημα με πολύ μικρές απαιτήσεις σε μνήμη και επεξεργαστική ισχύ. Βασικά τα μειονεκτήματα αποτελούν η πολυπλοκότητα και η δυσκολία εκμάθησής του, ιδιαίτερα από κάποιον που δεν έχει εμπειρία σε λειτουργικά τύπου Unix-Linux. Το Tiny-Os χρησιμοποιείται σήμερα στο σύνολο των υπαρχόντων SNs, εκτός από τα SunSpots τα οποία νομιμοποιούν μια ανοιχτού κώδικα εικονική μηχανή, την Squawk, η οποία είναι συμβατή με τη γλώσσα προγραμματισμού Java ME (Micro Edition) και «τρέχει» χωρίς την ανάγκη ύπαρξης λειτουργικού συστήματος μιας και η ίδια το υποκαθιστά. Στο μέλλον είναι πολύ πιθανό ότι και άλλες μεγάλες εταιρίες θα το χρησιμοποιήσουν.(Microsoft. IBM) (Gavalas 2005).

Εύρεση Θέσης

Σε πολλές εφαρμογές (π.χ. πυροπροστασίας) όταν λαμβάνεται μια πληροφορία από ένα SN είναι αναγκαία και η ταυτόχρονη πληροφόρηση της ακριβούς θέσης του κόμβου μέσα στο WSN. Επίσης, η γνώση της θέσης του SN αποτελεί προϋπόθεση σε διάφορους αλγόριθμους δρομολόγησης δεδομένων (Batalin 2005).

Παρόλο που η χρήση OPS (Global Positioning System) θα μπορούσε να λύσει το πρόβλημα της εύρεσης θέσης των κόμβων δημιουργεί άλλα προβλήματα. Τα δύο σημαντικότερα είναι η αύξηση του κόστους των SNs και το γεγονός ότι το GPS δεν μπορεί να λειτουργήσει σε εσωτερικά περιβάλλοντα ή σε περιβάλλοντα με έντονο ηλεκτρομαγνητικό θόρυβο ή παρεμβολές. Επίσης ακόμη και σε εξωτερικά περιβάλλοντα, τα οποία όμως έχουν φυσικά εμπόδια,

το σήμα που απαιτείται για τη λειτουργία GPS μπορεί να είναι πολύ ασθενές. Οι παραπάνω λόγοι ώθησαν αρκετές μελέτες να εστιάσουν στην εύρεση της γεωγραφικής θέσης των SNs μέσα σε ένα WSN χωρίς τη χρήση GPS. (Batalin 2005)

Τα WSNs χαρακτηρίζονται από την πυκνή τοποθέτηση των SNs τους, η οποία μπορεί να επιφέρει πλεονασμό στα δεδομένα που συλλέγουν οι αισθητήρες από τον περιβάλλοντα χώρο τους (π.χ. δυο γειτονικοί αισθητήρες μπορεί να συλλέγουν παραπλήσιες αν όχι ίδιες τιμές για τη θερμοκρασία ή το επίπεδο υγρασίας του περιβάλλοντος). Επίσης, ο εντοπισμός διάφορων συμβάντων από ένα WSN μπορεί να απαιτεί από τα SNs την παρατήρηση και αποστολή των συλλεγόμενων δεδομένων ανά τακτά χρονικά διαστήματα, με αποτέλεσμα ένα πλήθος γειτονικών αισθητήρων που εντοπίζει-καταγράφει το συγκεκριμένο συμβάν να στέλνει ταυτόχρονα δεδομένα προς το PH, δημιουργώντας μια άσκοπη σπατάλη ενέργειας και αυξημένη δικτυακή κίνηση. Από τα παραπάνω διαφαίνονται τα σημαντικά πλεονεκτήματα της χρήσης μεθόδων χωρικής και χρονικής συσχέτισης σε περιβάλλοντα WSNs, με κυριότερα τη μείωση της δικτυακής κίνησης και τη μεγάλη οικονομία σε ενέργεια.

Οι Akyildi et al. προτείνουν ένα θεωρητικό πλαίσιο για τη μοντελοποίηση της χωροχρονικής συσχέτισης στα WSNs με απώτερο σκοπό την ανάπτυξη αποδοτικών επικοινωνιακών πρωτοκόλλων, τα οποία θα εκμεταλλεύονται στο έπακρο τη συσχέτιση αυτή και τα πλεονεκτήματα που επιφέρει. Επίσης, ο αλγόριθμος JAJD -MPJ FJ εκμεταλλεύεται τη χωροχρονική συσχέτιση σε εφαρμογές κατανεμημένης συλλογής δεδομένων. (JProbe, 2007)

5.5 Βέλτιστη Τοποθέτηση SNs

Τα SNs μπορούν είτε να διασπαρθούν μαζικά σε μια περιοχή ή χώρο, είτε να τοποθετηθούν ένα προς ένα. Μερικοί μέθοδοι εγκατάστασης των SNs είναι οι παρακάτω:

- Τοποθέτηση ένα προς ένα από άνθρωπο ή robot.
- Διασπορά από αεροπλάνο.
- Διασπορά από ένα βλήμα ή πύραυλο που θα τα μεταφέρει.

Η περιοχή που μπορεί να καλύψει ένα WSN εξαρτάται από τον αριθμό των SNs του, το πρότυπο τοποθέτησής τους αλλά και από τα γεωμορφολογικά χαρακτηριστικά του πεδίου. Για την επαρκή και μέγιστη δυνατή κάλυψη ενός χώρου λαμβάνοντας υπόψη και τη μέγιστη απόσταση επικοινωνίας των SNs, πρέπει να υπάρχει ένα σχέδιο για το πώς πρέπει να τοποθετηθούν τα SNs στο χώρο που πρέπει να καλυφθεί. Έτσι επιτυγχάνεται οικονομία στον αριθμό των χρησιμοποιούμενων SNs και περιορισμό του κόστους, μειώνεται ο ηλεκτρομαγνητικός θόρυβος που επηρεάζει αρνητικά την ασύρματη επικοινωνία μεταξύ των κόμβων αφού λιγοστεύουν οι εκπομπές δεδομένων, μειώνεται η συμφόρηση δεδομένων και αποφεύγονται ανεπιθύμητα φαινόμενα όπως η ύπαρξη πολλών αλληλεπικαλυπτόμενων SNs σε μια περιοχή και η ύπαρξη σχετικά λίγων σε μια άλλη. Οι ερευνητές, όσον αφορά τη βέλτιστη τοποθέτηση των SNs σε ένα γεωγραφικό χώρο, έχουν ασχοληθεί με διάφορες προσεγγίσεις του προβλήματος. Στις παρακάτω εργασίες, τα SNs είναι εφοδιασμένα με μονάδες κίνησης και μπορούν να κινηθούν αυτόνομα για να βρουν τις βέλτιστες θέσεις στο χώρο.

Οι Batalin et al. προτείνουν μια αρχιτεκτονική που συνδυάζει στατικά και κινητά SNs ώστε να επιτυγχάνεται καλύτερη κάλυψη του χώρου, από την άποψη της χωροχρονικής συσχέτισης. Επίσης, τα τελευταία χρόνια γίνεται έρευνα για τη βέλτιστη τοποθέτηση των SNs λαμβάνοντας υπόψη και τις τρεις διαστάσεις του χώρου. (Batalin 2005).

Χρονικός Συγχρονισμός

Στα WSNs ο συντονισμός διαδικασιών είναι απολύτως αναγκαίος, για διεργασίες που πρέπει να ξεκινήσουν σε συγκεκριμένες χρονικές στιγμές στο σύνολο ή μέρος των SNs, για το συγχρονισμό δικτυακών αλλά και άλλων λειτουργιών. Οι λύσεις χρονικού συγχρονισμού που χρησιμοποιούνται σε άλλου τύπου δίκτυα π.χ. το Δικτυακό Χρονικό Πρωτόκολλο, το οποίο χρησιμοποιείται στο Διαδίκτυο, δεν είναι κατάλληλες για περιβάλλοντα WSNs λόγω του μεγάλου πλήθους μηνυμάτων που απαιτούνται για να επιτευχθεί ο συγχρονισμός. Αυτό θα είχε ως αποτέλεσμα την αύξηση της κατανάλωσης της ήδη περιορισμένης ενέργειας των SNs και τη μεγάλη χρήση του εύρους ζώνης (bandwidth) του δικτύου. (JProbe, 2007)

Για τους παραπάνω λόγους, οι ερευνητές ασχολήθηκαν με τη δημιουργία κατάλληλων, για τις ιδιομορφίες των WSNs. πρωτοκόλλων χρονικού συγχρονισμού. Κάποιες εργασίες πρότειναν τη χρήση GPS για το χρονικό συγχρονισμό των SNs. Ένα πολύ σημαντικό στοιχείο στις λύσεις χρονικού συγχρονισμού σε περιβάλλοντα WSNs αποτελεί και ο συνολικός χρόνος στον οποίο επιτυγχάνεται ο συγχρονισμός του συνόλου των SNs. Μερικές εφαρμογές απαιτούν τον ελάχιστο δυνατό χρόνο συγχρονισμού για να λειτουργήσουν σωστά π.χ. μεταπήδηση συχνοτήτων (Frequency Hopping) ταυτόχρονα σε πολλά SNs. Bandwidth είναι το μέγεθος των δεδομένων που μπορεί να μεταφερθεί στη μονάδα του χρόνου. Η ασφάλεια των WSNs είναι ένας τομέας με μεγάλη βαρύτητα, δεδομένου ότι τα δίκτυα αυτά μπορούν να χρησιμοποιηθούν σε «ευαίσθητες» εφαρμογές όπως η προστασία υλικών αγαθών, η επιτήρηση εξουσιοδοτημένων χώρων, στρατιωτικές εφαρμογές κ.λπ. Τα SNs ενός WSN παρουσιάζουν πολλούς περιορισμούς και πολλές φορές, καλούνται να λειτουργήσουν χωρίς επιτήρηση σε εξωτερικά περιβάλλοντα. Αυτό φέρνει στο προσκήνιο το πρόβλημα προστασίας τους από κακόβουλους

χρήστες, οι οποίοι θα προσπαθήσουν να θέσουν ένα μέρος ή ακόμη και ολόκληρο το WSN εκτός λειτουργίας.

Οι Avancha et al. συνοψίζουν τις ερευνητικές κατευθύνσεις που σχετίζονται με την ασφάλεια των WSNs. Επίσης, οι Wood και Staneovic εξετάζουν την αντιμετώπιση των επιθέσεων τύπου άρνησης εξυπηρέτησης (Denial of Service, DoS) σε WSNs και προτείνουν διάφορα αντίμετρα τα οποία όμως δεν μπορούν να προστατεύσουν πλήρως ένα WSN σε όλες τις περιπτώσεις. Σε γενικές γραμμές, οι υπάρχουσες προτάσεις που αφορούν την ασφάλεια των WSNs δεν μπορούν να αντιμετωπίσουν αποτελεσματικά επιθέσεις παρεμβολής των συχνοτήτων επικοινωνίας των SNs (ένας τύπος DoS) με αποτέλεσμα αυτές πολύ συχνά να οδηγούν σε πλήρη κατάρρευση του δικτύου. Αυτός ήταν και ο κύριος λόγος που μας οδήγησε στην έρευνα σε αυτόν τον τομέα και στην πρόταση δύο προσεγγίσεων που δίνουν στα WSNs τη δυνατότητα αποτελεσματικής άμυνας εναντίων τέτοιων επιθέσεων. (Avanacha 2009)

Εκτός από τον τομέα της ασφάλειας, ο τομέας της προστασίας των δεδομένων που συλλέγονται και διακινούνται σε ένα WSN παρουσιάζει επίσης μεγάλο ενδιαφέρον. Τα WSNs είναι σε θέση να συλλέξουν σε μικρό χρονικό διάστημα τεράστιο όγκο πληροφοριών ο οποίος, σε μερικές περιπτώσεις, μπορεί να περιέχει προσωπικά δεδομένα. Φανταστείτε τη λειτουργία ενός WSN σε ένα νοσοκομείο όπου ο κάθε αισθητήρας έχει τη δυνατότητα να δώσει πολλές πληροφορίες για τον κάθε ασθενή. Μπορεί κάτι τέτοιο να εξυπηρετεί βέβαια το ιατρικό προσωπικό αλλά τίθεται θέμα διαχείρισης και προστασίας αυτών των δεδομένων ώστε, για παράδειγμα, να είναι προσβάσιμα μόνο από τους ιατρούς που παρακολουθούν το συγκεκριμένο ασθενή και όχι από όλο το ιατρικό προσωπικό. (Avanacha 2009)

Εντοπισμός και Παρακολούθηση

Μια κοινή εφαρμογή των WSNs είναι ο εντοπισμός και η εύρεση της ακριβούς θέσης ενός στόχου, ο οποίος έχει ανιχνευθεί από δύο ή περισσότερα SNs. Ένα χαρακτηριστικό παράδειγμα του παραπάνω προβλήματος είναι ο υπολογισμός της θέσης ενός οχήματος σύμφωνα με τις μετρήσεις που λαμβάνονται από κάθε SN που βρίσκεται στην περιοχή ενδιαφέροντος.

Οι Bergamo et al. δίνουν μια λύση στο παραπάνω πρόβλημα δυνατότητα να χρησιμοποιήσει ειδικά περιβάλλοντα ανάπτυξης εφαρμογών (Integrated Development Environments, IDEs), π.χ. Netbeans και Eclipse.

5.6 Ενέργεια - Έλεγχος Δικτύου και Επικοινωνιών

Αυτή η ενότητα περιγράφει τις ερευνητικές τάσεις οι οποίες εστιάζουν σε θέματα ελέγχου δικτύου και επικοινωνιών σε WSNs. Οι αρχιτεκτονικές που χρησιμοποιούνται σε άλλους τύπους δικτύων, όπως για παράδειγμα τα ad-hoc δίκτυα που ακολουθούν το πρότυπο 802.11, δεν μπορούν να χρησιμοποιηθούν στα WSNs, κυρίως λόγω των μοναδικών χαρακτηριστικών τους αλλά και των απαιτήσεων των εφαρμογών που «τρέχουν» σε αυτά. Επιπλέον υπάρχουν αρκετές διαφορές μεταξύ των δύο αυτών τύπων δικτύων (WSNs και 802.11 - συμβατών ad-hoc δικτύων) οι οποίες επιτείνουν την «συμβατότητα μεταξύ τους. Περιληπτικά, οι διαφορές αυτές είναι: (Johunson, 2002)

- Ο αριθμός των SNs σε ένα WSN είναι κατά πολύ μεγαλύτερο από τον αριθμό των κόμβων ενός ad-hoc δικτύου.
- Η πυκνότητα ανάπτυξης των SNs ενός WSN στην περιοχή ενδιαφέροντος είναι συχνά μεγάλη.

- Τα « SNs από κατασκευής, έχουν περιορισμένα αποθέματα ενέργειας και μειωμένη υπολογιστική ισχύ και μνήμη.
- Το περιορισμένο εύρος ζώνης ενός WSN δεν επιτρέπει τη χρήση πακέτων επιβεβαίωσης δεδομένων.
- Η πιθανότητα βλάβης κάποιου SN είναι μεγάλη (δεδομένου ότι αυτά τοποθετούνται σε εξωτερικούς χώρους και είναι εκτεθειμένα στις περιβαλλοντολογικές συνθήκες).
- Η μέγιστη δυνατή εμβέλεια επικοινωνίας των SNs είναι περιορισμένη.
- Τα SNs χρησιμοποιούν συνήθως, επικοινωνία «ευρείας εκπομπής» (broadcast) ενώ τα περισσότερα 802.11 ad-hoc δίκτυα βασίζονται στην επικοινωνία σημείου προς σημείο.
- Τα SNs ενός WSN συνήθως, δεν έχουν κάποιο μοναδικό αναγνωριστικό (ID) εξαιτίας του μεγάλου μεγέθους της επικεφαλίδας που απαιτεί μια τέτοια τιμή, καθώς και του μεγάλου αριθμού τους.
- Τα SNs συνήθως, αναπτύσσονται στην περιοχή ενδιαφέροντος χωρίς να μετακινούνται από αυτή, σε αντίθεση με τους κόμβους των ad-hoc δικτύων όπου παρουσιάζεται μεγάλη κινητικότητα.
- Τα SNs στο πλείστο των περιπτώσεων δεν έχουν τη δυνατότητα αναπλήρωσης της ενέργειας τους σε αντίθεση με τους κόμβους των ad-hoc δικτύων που συνήθως, έχουν την παραπάνω δυνατότητα. Επιπλέον η αρχιτεκτονική ενός WSN πρέπει να:
- Χρησιμοποιεί τις ασύρματες επικοινωνίες με τέτοιο τρόπο ώστε να παρέχει τη μέγιστη ενεργειακή οικονομία.

- Να μοιράζει διεργασίες σε γειτονικά SNs.
- Να συνδυάζει τη δρομολόγηση δεδομένων με μεθόδους μεγίστης δυνατής εξοικονόμησης ενέργειας. Όπως συμπεραίνεται από τα παραπάνω, η έρευνα στον έλεγχο δικτύου και επικοινωνιών σε VVSNs ενσωματώνει την έρευνα πάνω σε επικοινωνιακά πρωτόκολλα εξοικονόμησης ενέργειας, ελέγχου συμφόρησης, δρομολόγησης δεδομένων και διαχείρισης τοπολογίας και τέλος μοντελοποίησης. (Akyildiz & Melodia 2007).

5.7 Επικοινωνιακά Πρωτόκολλα Εξοικονόμησης Ενέργειας

Οι κόμβοι ενός WSN πρέπει να είναι σε θέση να λειτουργούν για όσο το δυνατόν μεγαλύτερα χρονικά διαστήματα, σε περιοχές οι οποίες τις πιο πολλές φορές είναι απομακρυσμένες ή δυσπρόσιτες. Για τον παραπάνω λόγο θεωρείται κρίσιμης σημασίας η όσο το δυνατόν μεγαλύτερη εξοικονόμηση ενέργειας, η οποία θα οδηγήσει στην αύξηση της ζωής (lifetime) των SNs και κατά επέκταση του WSN. Η διαδικασία που καταναλώνει τα πιο μεγάλα ποσοστά ενέργειας σε ένα SN είναι η επικοινωνία, δηλαδή η χρήση του πομπού του για την ασύρματη αποστολή δεδομένων [188]. Για να μειωθεί η ενεργειακή κατανάλωση της επικοινωνίας σε ένα SN πρέπει να βελτιστοποιηθούν όλα τα επίπεδα (layers) επικοινωνίας ξεκινώντας από το φυσικό επίπεδο και καταλήγοντας στο επίπεδο εφαρμογής. Το επίπεδο που επηρεάζει περισσότερο την κατανάλωση ενέργειας είναι το επίπεδο διασύνδεσης δεδομένων (Data Link Layer. DLL) και πιο συγκεκριμένα το υποεπίπεδο πρόσβασης στο μέσο (Medium Access Control, MAC), ιδιαίτερα σε δίκτυα με ραδιοπομπούς μικρού κύκλου καθήκοντος (low duty cycle). Έρευνες που έχουν ως στόχο τη μειωμένη

ενεργειακή κατανάλωση των WSNs, είναι οι [48] και [155], οι οποίες και προτείνουν βελτιωμένα πρωτόκολλα MAC. Επίσης οι Reason και Rabaey στο [173] προτείνουν ένα επικοινωνιακά πρωτόκολλο το οποίο βασίζεται στο σύστημα χρονικής πολυπλεξίας (Time Division Medium Access. TDMA), για την επέκταση της ζωής των SNs. (Akyildiz & Melodia 2007).

5.8 Ασφάλεια - Έλεγχος Συμφόρησης

Στα ενσύρματα και ασύρματα δίκτυα προκαλείται συμφόρηση όταν η κίνηση των δεδομένων ξεπερνά τη χωρητικότητα του δικτύου σε οποιοδήποτε σημείο του. Ο έλεγχος της συμφόρησης (congestion control) στα ενσύρματα δίκτυα συνήθως γίνεται χρησιμοποιώντας μηχανισμούς από άκρο σε άκρο (end-to-end), σε συνδυασμό με μηχανισμούς επιπέδου δικτύου. Παρόλα αυτά, οι παραπάνω μηχανισμοί δεν είναι κατάλληλοι για τα ασύρματα δίκτυα διότι οι ταυτόχρονες μεταδόσεις στα διάφορα «κανάλια» επικοινωνίας αλληλοεπιδρούν και επηρεάζουν το ένα το άλλο, ενώ η ποιότητα του κάθε καναλιού παρουσιάζει μεγάλη διακύμανση ανά διαφορετικές χρονικές περιόδους. Επίσης, το μεγάλο πλήθος μηνυμάτων επιβεβαίωσης (Acknowledge Messages) που απαιτούν για τη λειτουργία τους οι μηχανισμοί συμφόρησης των ενσύρματων δικτύων, τη δέσμευση μεγάλου επικοινωνιακού εύρους ζώνης και επιπλέον αυξημένη κατανάλωση ενέργειας των SNs. Για τους παραπάνω λόγους κρίνεται επιτακτική η σχεδίαση αποδοτικών μηχανισμών ελέγχου συμφόρησης για τα WSNs οι οποίοι θα λαμβάνουν υπόψη τις ιδιαιτερότητες τους. Οι Hew et al. στο προτείνουν ένα καταμεμημένο και κλιμακωτό (distributed and scalable) αλγόριθμο ο οποίος εξαλείφει τη συμφόρηση σε ένα WSN και εξασφαλίζει μια ομαλή ροή πακέτων δεδομένων προς το PE (Johunson, 2002).

Δρομολόγηση Δεδομένων και Διαχείριση Τοπολογίας

Ένα WSN που έχει εγκατασταθεί σε ένα δυσπρόσιτο μέρος, μπορεί να χάσει, κάτω από ακραίες συνθήκες, σχετικά εύκολα ένα μεγάλο μέρος από τα SNs του λόγω δυσλειτουργιών των τελευταίων, πράγμα που δυσκολεύει τη διατήρηση της τοπολογίας του δικτύου και τη δρομολόγηση των δεδομένων εντός αυτού. Επίσης, η πυκνότητα σε ένα WSN μπορεί να φτάσει ως και τα 20 SNs ανά κυβικό μέτρο κάτι που δημιουργεί επιπλέον δυσκολίες στη διαχείριση της τοπολογίας. Οι αλλαγές στην τοπολογία ενός WSN οφείλονται στις παρακάτω αλλαγές που μπορούν να συμβούν στα SNs:

- Ποσοστό διαθέσιμης ενέργειας.
- Αλλαγή θέσης.
- Δυνατότητες επικοινωνίας.
- Δυσλειτουργία.

Δυστυχώς, η παρουσίαση δυσλειτουργιών στα SNs ενός WSN είναι ένα σύνηθες φαινόμενο, κυρίως, λόγω της εξάντλησης των ενεργειακών αποθεμάτων τους, καταστροφής από καιρικές συνθήκες ή δολιοφθορά, «μπλοκαρίσματος» των επικοινωνιακών καναλιών κ.λπ. Επίσης, σε WSNs που περιλαμβάνουν και SNs με δυνατότητες εκκίνησης, όπως είναι φυσικό η τυπολογία αλλάζει συνεχώς.

Επιπρόσθετα στα WSNs οι αλγόριθμοι δρομολόγησης θα πρέπει να ικανοποιούν όσο το δυνατόν περισσότερα από τα παρακάτω χαρακτηριστικά:

- Ελάχιστη δυνατή κατανάλωση επικοινωνιακού εύρους ζώνης.
- Ανοχή σε λάθη μετάδοσης (Johunson, 2002).
- Χαμηλή υπολογιστική πολυπλοκότητα.
 - Μικρός χρόνος υπολογισμού δρομολογήσεων.
 - Ταχύτατη ανταπόκριση σε αλλαγές τοπολογίας.

- Αποφυγή κυκλικών διαδρομών.
- Υπολογισμός πολλαπλών διαδρομών.
- Υποστήριξη παροχής υπηρεσιών με χαρακτηριστικά QoS (Quality of Service, Ποιότητα της Υπηρεσίας).

Εκμετάλλευση του τρόπου που τα δεδομένα «ρέουν» στο δίκτυο χρησιμοποιώντας τους πόρους του (network flow, ροή δικτύου).

Πιο αναλυτικά, δυνατότητα δρομολόγησης των δεδομένων με τέτοιο τρόπο ώστε να αυξάνεται ο μέσος όρος ζωής των SNs μέσω της μειωμένης ενεργειακής κατανάλωσης. Οι αλγόριθμοι δρομολόγησης σε WSNs μπορούν να διακριθούν με βάση το αν η δρομολόγηση γίνεται κατακεντρωμένα (distributed) σε κάθε SN ή κεντροποιημένα (centralized) από το SN που αποστέλλει δεδομένα. Στην πρώτη περίπτωση κάθε SN αποφασίζει για το επόμενο SN στο οποίο θα προωθήσει το πακέτο, ενώ στη δεύτερη περίπτωση η διαδρομή που θα ακολουθήσει κάθε πακέτο, καθορίζεται από το SN που έχει αποστείλει αρχικά το πακέτο. Οι διαδρομές αυτές μπορεί να είναι είτε στατικές, είτε να προσαρμόζονται δυναμικά στην εκάστοτε κατάσταση του WSN, κάτι ιδιαίτερα χρήσιμο σε συχνές αλλαγές τοπολογίας. Ένας άλλος τρόπος διαχωρισμού των αλγόριθμων δρομολόγησης είναι σε προσανατολισμένους στα δεδομένα ή δεδομένο-κεντρικούς (data-centric), ιεραρχικούς (hierarchical) και σε αυτούς που βασίζονται στη θέση των SNs (location-based) (Johnson, 2002).

Υπάρχουν ακόμη μερικοί υβριδικοί αλγόριθμοι, οι οποίοι συνδυάζουν τα χαρακτηριστικά δύο κατηγοριών. Οι δεδομένο-κεντρικοί αλγόριθμοι βασίζονται σε ερωτήματα (παρόμοια με τα SQL ερωτήματα σε μια βάση δεδομένων) και εξαρτώνται από την ύπαρξη ονομάτων-ετικετών σε όλα τα είδη δεδομένων που συλλέγουν τα SNs. Οι ιεραρχικοί αλγόριθμοι στοχεύουν στην ομαδοποίηση των SNs με τέτοιο τρόπο ώστε να εξασφαλίζεται η συνδεσιμότητα του WSN, ελαχιστοποιώντας ταυτόχρονα την κατανάλωση ενέργειας. Τέλος, οι αλγόριθμοι που βασίζονται στη θέση των SNs, για να δρομολογήσουν δεδομένα προς αυτά, πρέπει να γνωρίζουν την ακριβή γεωγραφική θέση του κάθε SN στο δίκτυο. Ένας από τους πιο διάσημους ιεραρχικούς αλγόριθμους δρομολόγησης

για WSNs είναι ο LEACH , ο οποίος χωρίζει τα SNs ενός WSNs σε συστάδες (clusters) με σκοπό την καλύτερη δρομολόγηση δεδομένων ώστε να επιτυγχάνεται αύξηση του ορίου ζωής του WSN, μέσω της μειωμένης κατανάλωσης ενέργειας και μειωμένος χρόνος αντίδρασης.

Ο LEACH ενσωματώνει μια κατανεμημένη τεχνική δημιουργίας συστάδων από SNs και ειδικούς αλγόριθμους οι οποίοι προσαρμόζουν τις συστάδες ανάλογα με την τοπολογία και εναλλάσσουν ανά τακτά χρονικά διαστήματα το SN που παίζει το ρόλο του αρχηγού συστάδας (Cluster Head, CH), ώστε να κατανέμεται εξίσου η ενεργειακή κατανάλωση σε όλα τα SNs της συστάδας (το CH καταναλώνει περισσότερη ενέργεια από τα υπόλοιπα SNs λόγω του ότι η επεξεργασία των δεδομένων από όλη τη συστάδα καθώς και η αποστολή τους προς το PE γίνεται μόνο από αυτό).

Επίσης ο LEACH ενσωματώνει τεχνικές που επιτρέπουν τη συνεργασία των SNs μιας συστάδας για τη διεκπεραίωση ορισμένων εργασιών, ώστε να επιτυγχάνεται περαιτέρω οικονομία ενέργειας. Μια από τις εργασίες αυτές είναι ο συγκερασμός δεδομένων με τον οποίο πληροφορίες που συλλέχτηκαν από πολλά SNs, φιλτράρονται βάση κάποιων κανόνων ή κριτηρίων. Παραδείγματα συγκερασμού δεδομένων αποτελούν διάφορες στατιστικές διεργασίες όπως ο μέσος όρος και το συνολικό άθροισμα.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Η ραγδαία εξέλιξη της τεχνολογίας πληροφοριών έχει καταστήσει διαθέσιμο μεγάλο όγκο πληροφοριών και υπηρεσιών. Έτσι, δημιουργήθηκε η ανάγκη για «έξυπνες» εφαρμογές που θα μπορούν να παίρνουν αποφάσεις, λαμβάνοντας υπόψη τις ανάγκες και τις απαιτήσεις του εκάστοτε χρήστη. Για να γίνει δυνατή αυτή η προσθήκη ευφυΐας, χρειάζεται ένας τρόπος αναπαράστασης των προαναφερθέντων στοιχείων, ώστε να υλοποιηθεί μια βάση γνώσης που θα είναι κατανοητή σε μηχανές, και μια μηχανή για την εξαγωγή νέας γνώσης από την ήδη υπάρχουσα. Η πλέον διαδομένη μορφή αναπαράστασης γνώσης για αυτόν τον σκοπό, είναι οι οντολογίες. Ωστόσο, ένα κινητό περιβάλλον χαρακτηρίζεται από περιορισμένους πόρους και έτσι δε μπορεί να υποστηρίξει την υψηλή πολυπλοκότητα των οντολογιών. Συνεπώς, η μετατροπή των οντολογιών σε άλλη αναπαράσταση γνώσης καθίσταται αναγκαία.

Η τεχνολογία έχει εξελιχθεί τόσο πολύ όπου έχει συγχωνευτεί σχεδόν σε όλα τα αντικείμενα της καθημερινότητας χωρίς όμως να γίνεται αντιληπτή. Τα συστήματα αυτά που είναι ενσωματωμένα πάνω σε ασύρματες δικτυακές υποδομές και υπηρεσίες γνωστές και ως συστήματα διάχυτου υπολογισμού επεκτείνονται ώστε να έχουν πρόσβαση σε απομακρυσμένες και διάχυτες πληροφορίες. Η τεχνολογία καθημερινά γίνεται ισχυρότερη και οι υπολογιστές μικρότεροι. Αυτά τα συστήματα έχουν ως κύριο στόχο την επίλυση των απαιτήσεων του κάθε χρήστη, και κατατάσσονται με κάποιους συγκεκριμένους παράγοντες που μπορεί να επηρεαστούν από το περιβάλλον της περιρρέουσας νοημοσύνης. Στην πτυχιακή αναφέρονται διάφορες εφαρμογές της περιρρέουσας νοημοσύνης και του διάχυτου υπολογισμού με την βοήθεια αισθητήρων που όπως φαίνεται θα αλλάξουν κατά πολύ τη ζωή μας στο μέλλον, αν φυσικά τεθούν σε εφαρμογή. Όπως είναι φυσικό για να λειτουργήσει ένα τέτοιο 'σύστημα' χρειάζεται κάποια οικονομική ευχέρεια που στην Ελλάδα του σήμερα είναι δύσκολο να τη βρει κανείς. Παρ' όλα αυτά σιγά σιγά βλέπουμε την τεχνολογία να εισβάλει και στην Ελλάδα με γοργούς αλλά σταθερούς ρυθμούς.

BIBΛΙΟΓΡΑΦΙΑ

Andrea Goldsmith, "Wireless Communications", Cambridge University, 2005.

I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A Survey on Wireless Multimedia Sensor Networks," Computer Networks (Elsevier) Journal vol. 51, no. 4, pp. 921–960, March 2007 [[pdf](#)] [[bibtex](#)]
[ISI Thomson Reuters FAST BREAKING PAPER IN COMPUTER SCIENCE](#)

I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "Wireless Multimedia Sensor Networks: Applications and Testbeds," Proceedings of the IEEE Vol. 96, No. 10. (2008), pp. 1588-1605 [[pdf](#)]

Κωπτής, Χ. Καψάλης, «Κεραίες και Ασύρματες Ζεύξεις», Εκδόσεις Τζιόλα, Αθήνα 2003. Andrea Goldsmith, "Wireless Communications", Cambridge University, 2005.

Omeni , O.C, Eljamaly (2007) , Energy Efficient Medium Access Protocol for Wirelsss Medical Body Area Sensor Networks

Mit technology review "10 emerging technologies that will change the world"

Jamal N. Al-Karaki and Ahmed E. Kamal, (2004) "Routing Techniques in Wireless Sensor Networks: A Survey", The Hashemite University and The Iowa State University.

Jiangi & Manivannan 2004, Routing protocols for sensor networkiking

N. Noury, T. Herve, V. Rialle et al., "Monitoring behaviour in the home using a smart fall sensor and position sensors," in Proceedings of the 1st Annual

International Conference on Microtechnologies in Medicine and Biology, pp. 607–610, 2000.

Gavalas, Mobile network agents for Network monitoring , an application case study 2005

Maxim Batalin, William Kaiser, Richard Pon, Gaurav S. Sukhatme, Gregory Pottie, Yan Yu, Jason Gordon, Mohammad H. Rahimi, and Deborah Estrin, "Task Allocation for Event-Aware Spatiotemporal Sampling of Environmental Variables," In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1846-1853, Edmonton, Canada, Aug 2005

Garlan, (2002) D. Garlan. D, P. Sicwiorck, A. Smailagic and P. Stccknkistic, "Project Aura: Toward Distraction-Free Pervasive Computing.", *IEEE Pcrva-sivc Computing Magazine*, (Vol. I, No.2). pp. 22-31, April-June 2002

Roman, (2002) M.Roman et. al. ."Gaia: A Middleware Infrastructure to Enable Active Spaces". *IEEE. Pervasive Computing*. 2002.

Johanson, (2002) B. Johanson. A. Fox. and T. Winograd. "Experiences with Ubiquitous Computing Rooms". *IEEE Pervasive Computing Magazine*, vol. 1(2). pp. 67-74. 2002

Weber, (1991) Mark Weiser. The computer for the 21st century. *Scientific American*, 265(3):94- -104. September 1991.

Coulouris, 2001 George Coulouris. Jean Dollimore and Tim Kindbcrg. "Distributed Systems: Concepts and Design", 3rd edition, Addison-Wesley, (c) Pearson Education 2001. ISBN: 0201 619-180

Jess, (2007)Jess - the Rule Engine for the Java Platform Available: <http://hcrzberg.ca.sandia.gov/jcss/> (last accessed on 24/4/2007)

Drossos, A. Kamcas. I. Mavrommatti. A. Pounds-Comish, A. Holmes. M. Colley. V. Callaghan, "Medicul eGadgets". Ubicomp 2004. Hie Sixth International Conference on Ubiquitous Computing, Nottingham. England, September 7-10. 2004

Jacquet. (2005) C. Jacquet, Y. Bourda and Y. Bcllik. "An Architecture for Ambient Computing". Proceedings of the 1st IEE International Workshop on intelligent Environments, pp. 47-54. Colchester. UK. June 2005.

Garlan, (2002) D. Garlan. D, P. Sicwiorck, A. Smailagic and P. Stcckkistic, "Project Aura: Toward Distraction-Free Pervasive Computing.", IEEE Pcrva-sivc Computing Magazine, (Vol. I, No.2). pp. 22-31, April-June 2002

Judd. (2003) (J. Judd. P. Stcckkistic. "Providing Contextual Information to Pervasive Computing Applications", PerCom 2003, Dallas. March 23-25,2003.

Stanford, (2007) Stanford Interactive Workspaces Project. Available: <http://iwork.Manford.edu/>

Jini, (1999) K. Arnold. B. O Sullivan. R.W. Schciflcr. J.Waldo. and A. Wollrath. "The Jini Specification" Addison Wesley, 1999.

(OZONE,2007) OZONE Project New Technologies and Services for Emerging Nomadic Societies, Available. [hnp://www.extra.rcsearch.philips.com/euprojects/ozonc/](http://www.extra.rcsearch.philips.com/euprojects/ozonc/)

Ilssamy, (2004) Valeric Issamy. Daniclc Sacchety. Fcrda Tartanoglu, Francoicc Sailhan, Rafik Chibout, Nicole Levy. Angel Talamona. "Developing Ambient

Intelligence Systems: A Solution Based on Web Services". In journal of Automated Software Engineering, 2004.

1999

Oxygen, 2007J MFT Oxygen project. Available: <http://oxygen.lcs.mit.edu/>
(OxygenSoftwareTech, 2007] Oxygen Software Technologies. Available: <http://oxygen.lcs.mit.edu/Software.html> (last accessed on 23/4/2007)

JXTA, 2007) JXTA project website, <http://www.jxta.org>

| Kamcas, 2003) A. Kamcas. et al, "An Architecture that Treats Everyday Objects as Communicating Tangible Components", in Proceedings of the 1st IEEE International Conference on Pervasive Computing and Communications (PervCom03), pp. 115-122. Port Worth. USA. March 2003

Goumopoulos, (2007) Christos Goumopoulos. Achilles Kameas. Brendan O'Hare. "Proactive Agriculture: An Integrated Framework for Developing Distributed Hybrid Systems". In the 4th International Conference on Ubiquitous Intelligence and Computing (UIC-07), Hong Kong, China. July 11-13. 2007

Issamy, (2004) Valeric Issamy. Danicic Sacchety. Ferda Tartanoglu, Francoise Sailhan, Rafik Chibout, Nicole Levy. Angel Talamona. "Developing Ambient Intelligence Systems: A Solution Based on Web Services". In journal of Automated Software Engineering, 2004.

11ST AG, 2007] 1STAG: Working Group I. 1ST Research Content. Final Report. Available: <http://www.cordis.lu/i.st/istng.htm> (last accessed on 23/4/2007)

Jacquet. (2005) C. Jacquet, Y. Bourda and Y. Bcllik. "An Architecture for Ambient Computing". Proceedings of the 1st IEE International Workshop on intelligent Environments, pp. 47-54. Colchester. UK. June 2005.

Java RMI S. Microsystems. Java Remote Method Invocation Distributed Computing for Java. WhitePaper. 1998.

Jess, 2007|Jess - the Rule Engine for the Java Platform Available: <http://hcrzberg.ca.sandia.gov/jcss/> (last accessed on 24/4/2007)

Horricks. (2003) Horricks. I., Patel-Schneider, P., Bolcy. H., Tibet, S., Grosf. B., Dean, M. (2003), "SWRL: A Semantic Web Rule Language Combining OWL and RulcML", version 0.5. 19th November 2003.

Jini, (1999) K. Arnold. B. O Sullivan. R.W. Schciflcr. J.Waldo. and A. Wollrath. "The Jini Specification" Addison Wesley, 1999.

Johanson, (2002) B. Johanson. A. Fox. and T. Winograd. "Experiences with Ubiquitous Computing Rooms". IEEE Pervasive Computing Magazine, vol. 1(2). pp. 67-74. 2002.

Johunson, (2002) Johanson. B.. ct al. PointRight: Experience with Flexible Input Redirection in Interactive Workspaces, in ACM Symposium on User Interface Software and Technology (UIST- 2002). 2002. Paris, France.

JProbe, (2007) JProbc suite website. Available: <http://www.qucst.com/jprobc/indcx.asp> (last accessed on 24/4/2007)

Judd. (2003) (J. Judd. P. Stcenkistc. "Providing Contextual Information to Pervasive Computing Applications", PerCom 2003, Dallas. March 23-25,2003.

JXTA,(2007) JXTA project website, <http://www.jxta.org>