



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Τμήμα Διδακτικής της Τεχνολογίας και Ψηφιακών Συστημάτων

Σχεδιασμός και Υλοποίηση Πλατφόρμας Διαχείρισης Υπηρεσιών σε Ευφυή Περιβάλλοντα

Διπλωματική Εργασία

του

Διακρούση Κωνσταντίνου

Επιβλέπουσα

Σταυρουλάκη Βέρα-Αλεξάνδρα

ΙΟΥΝΙΟΣ 2009

ΠΕΡΙΛΗΨΗ

Η τεχνολογία των έξυπνων χώρων έχει να προσφέρει πολλά στη βελτίωση της καθημερινότητας και καθώς διαθέτει ένα πλήθος εφαρμογών, γίνεται ολοένα και περισσότερο αντικείμενο έρευνας και μελέτης. Αν και για την υλοποίηση ενός έξυπνου χώρου είναι διαθέσιμη μια μεγάλη ποικιλία από συσκευές, τεχνολογίες δικτύωσης και υποσυστήματα έξυπνων χώρων, υπάρχει ωστόσο έλλειψη μιας πλατφόρμας που να ενοποιεί με έναν ενιαίο τρόπο και ανεξαρτήτως κατασκευαστή τα διάφορα συστατικά της. Σε αυτό το πλαίσιο σχεδιάστηκε μια υπηρεσιοστρεφή αρχιτεκτονική, βάσει της οποίας υλοποιήθηκε η πλατφόρμα IESAP (Intelligent Environment's Services Administration Platform) με τη χρήση των τεχνολογιών OSGi και UPnP. Αυτός ο συνδυασμός τεχνολογιών και η υπηρεσιοστρεφή σχεδίαση της αρχιτεκτονικής, δίνουν τη δυνατότητα στην προτεινόμενη πλατφόρμα να καλύψει μερικές από τις βασικές αρχές σχεδίασης συστημάτων έξυπνων χώρων, όπως είναι η επεκτασιμότητα, η διαλειτουργικότητα, η χαλαρή σύζευξη και η ανεξαρτησία πλατφόρμας εκτέλεσης.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ	3
ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ	7
ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ.....	8
ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ	9
1 ΕΙΣΑΓΩΓΗ	11
1.1 Εισαγωγή.....	11
1.2 Δομή της εργασίας	13
2 ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΕΠΙΣΚΟΠΗΣΗ.....	15
2.1 Διεπισδυτική Πληροφορική	15
2.1.1 Ανασκόπηση	15
2.1.2 Σχετικές τεχνολογίες	16
2.2 Έξυπνοι χώροι.....	18
2.2.1 Ορισμοί	18
2.2.2 Ιστορική αναδρομή ερευνητικών έργων	19
2.2.3 Τα συστατικά των έξυπνων χώρων και η κατηγοριοποίηση τους	20
2.2.4 Τεχνολογίες υποδομής έξυπνων χώρων.....	22
2.2.4.1 Συσκευές	22
2.2.4.2 Αντιληπτικές διεπαφές	25
2.3 Παρουσίαση εφαρμογών έξυπνων χώρων	27
2.3.1 Το έξυπνο μουσείο	27
2.3.2 Διαδραστικός χώρος εργασίας	29
2.3.3 Η έξυπνη τάξη	30
2.4 Αρχιτεκτονική έξυπνων χώρων.....	32
2.4.1 Αρχές σχεδίασης αρχιτεκτονικής έξυπνων χώρων	33
2.5 Υλοποιήσεις αρχιτεκτονικής έξυπνων χώρων	35

2.5.1 Το έξυπνο σπίτι Gator Tech	35
2.5.1.1 Τεχνολογίες έξυπνου σπιτιού	35
2.5.1.2 Αρχιτεκτονική ενδιάμεσου λογισμικού	36
2.5.2 Το έξυπνο δωμάτιο του ΑΙΤ	39
2.5.2.1 Αρχιτεκτονική ενδιάμεσου λογισμικού	39
2.5.3 Πλατφόρμα Έξυπνων Χώρων	42
2.5.3.1 Αρχιτεκτονική ενδιάμεσου λογισμικού	42
3 ΤΕΧΝΟΛΟΓΙΕΣ ΥΛΟΠΟΙΗΣΗΣ ΕΞΥΠΝΩΝ ΧΩΡΩΝ	47
3.1 OSGi.....	47
3.2 UPnP	51
3.3 Jini.....	54
3.4 Web Services.....	56
3.5 JADE.....	59
3.6 Γλώσσες σήμανσης διεπαφής (User interface markup languages)	60
4 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΠΡΟΤΕΙΝΟΜΕΝΗΣ ΠΛΑΤΦΟΡΜΑΣ	62
4.1 Εισαγωγή.....	62
4.2 Προδιαγραφές συστήματος.....	63
4.3 Αρχιτεκτονική πλατφόρμας IESAP	64
4.3.1 Διαχείριση δεδομένων αισθητήρων (context management)	66
4.4 Επιλογή τεχνολογιών	70
5 ΥΛΟΠΟΙΗΣΗ ΠΛΑΤΦΟΡΜΑΣ	73
5.1 Προσομοιωτής περιβάλλοντος.....	74
5.1.1 Τεχνικά χαρακτηριστικά προσομοιωτή	75
5.2 Εξομοιωτές συσκευών UPnP	76
5.3 Ανάλυση OSGi bundle.....	78
5.4 Δυναμικές και δηλωτικές υπηρεσίες.....	80

5.5 Παρουσίαση των bundles της πλατφόρμας IESAP	83
6 ΠΕΡΙΓΡΑΦΗ ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑΣ	96
6.1 Εγκατάσταση και εκτέλεση πλατφόρμας.....	96
6.2 Εκκίνηση Προσομοιωτή Περιβάλλοντος και Εφαρμογής PMA	98
6.3 Λειτουργία Ρύθμισης Θερμοκρασίας και Φωτεινότητας.....	99
6.4 Λειτουργία Ενεργοποίησης Φωτεινότητας με την Ανίχνευση Κίνησης...	102
6.5 Λίστα Συσκευών Συστήματος.....	104
6.6 Κονσόλα Μηνυμάτων Συστήματος.....	105
6.7 Δημιουργία και εισαγωγή στο σύστημα νέας συσκευής.....	105
7 ΣΥΜΠΕΡΑΣΜΑΤΑ	108
7.1 Ανασκόπηση	108
7.2 Συμπεράσματα	108
7.3 Προτάσεις για περαιτέρω ανάπτυξη	109
ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ	112

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1. Ταξινόμηση των περιοχών έρευνας της διεπισδυτικής πληροφορικής ..	18
Εικόνα 2. Οι διάφορες τεχνολογίες που ενσωματώνουν οι έξυπνοι χώροι	27
Εικόνα 3. Επισκόπηση του συστήματος του οδηγού	29
Εικόνα 4. Άποψη του iRoom	30
Εικόνα 5. Άποψη της έξυπνης τάξης	32
Εικόνα 6. Ενδιάμεσο λογισμικό έξυπνου χώρου	36
Εικόνα 7. Δομή περιβάλλοντος εκτέλεσης Έξυπνης Πλατφόρμας.....	43
Εικόνα 8. Διαστρωμάτωση του Πλαισίου OSGi.....	48
Εικόνα 9. Αλληλεπίδραση μεταξύ των στρωμάτων του Πλαισίου της OSGi	50
Εικόνα 10. Ανάκτηση περιγραφής Συσκευής και Υπηρεσίας	52
Εικόνα 11. Η UPnP ιεραρχία Συσκευής και Υπηρεσίας.....	53
Εικόνα 12. Αλλαγές Κατάστασης και Ειδοποιήσεις.....	54
Εικόνα 13. Η γενική διαδικασία χρήσης ενός Web service	59
Εικόνα 14. Διαστρωμάτωση πλατφόρμας IESAP	66
Εικόνα 15. Η αλληλεπίδραση μεταξύ Αισθητήρα και Εκκινητήρα.....	68
Εικόνα 16. Διάγραμμα Ακολουθίας λειτουργίας <i>Απόφαση</i>	87
Εικόνα 17. Διάγραμμα Ακολουθίας λειτουργίας <i>Ενεργοποίηση Εκκινητήρων</i>	89
Εικόνα 18. Διάγραμμα Ακολουθίας λειτουργίας <i>Ρύθμιση Φωτεινότητας</i>	94
Εικόνα 19. Διάγραμμα Ακολουθίας λειτουργίας <i>Ενεργοποίηση Φωτισμού με την Ανίχνευση Κίνησης</i>	95
Εικόνα 20. Προσομοιωτής Περιβάλλοντος	98
Εικόνα 21. Η Εφαρμογή PMA	99
Εικόνα 22. Εφαρμογή PMA - Ρύθμιση Θερμοκρασίας και Φωτεινότητας	100
Εικόνα 23. Ο Προσομοιωτής Περιβάλλοντος μετά την εκτέλεση του σεναρίου	101
Εικόνα 24. Εφαρμογή PMA – Ενεργοποίηση Φωτεινότητας με Ανίχνευση Κίνησης.....	102
Εικόνα 25. Ο Προσομοιωτής Περιβάλλοντος μετά την εκτέλεση του σεναρίου	103
Εικόνα 26. Εφαρμογή PMA - Λίστα Συσκευών Συστήματος	104
Εικόνα 27. Εφαρμογή PMA – Μηνύματα Συστήματος.....	105

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1. Σχέση Λειτουργίας, Παρατήρησης, Ενέργειας, Υπηρεσίας.....	69
Πίνακας 2. Κατανομή αισθητήρων και εκκινητήρων.....	73
Πίνακας 3. Χαρακτηριστικά περιβάλλοντος προσομοίωσης.....	74

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

ACL	Access Control List
AIT	Athens Information Technology
API	Application Programming Interface
DCM	Distributed Component Model
DHCP	Dynamic Host Configuration Protocol
FIPA	Foundation for Intelligent Physical Agents
FIPA	Foundation of Intelligent Physical Agents
GENA	General Event Notification Architecture
GSM	Global System for Mobile communications
HTTP	Hypertext Transfer Protocol
IESAP	Intelligent Environment's Services Administration Platform
IIOP	Internet Inter-Orb Protocol
IP	Internet Protocol
JADE	Java Agent Development
LUS	LookUp Service
MAS	Multi-Agent System
MIT	Massachusetts Institute of Technology
MTP	Media Transfer Protocol
OSGi	Open Service Gateway initiative
PDA	Personal Digital Assistant
PMA	Platform Management Application

RFID	Radio-Frequency Identification
RMI	Remote Method Invocation
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
UDP	User Datagram Protocol
UIML	User Interface Markup Language
UMTS	Universal Mobile Telecommunications System
UPnP	Universal Plug and Play
URL	Uniform Resource Locator
WLAN	Wireless Local Area Network
WSD	Web Service Description
WSDL	Web Service Description Language
XAML	Extensible Application Markup Language
XML	Extensible Mark-up Language
XUL	XML User Interface Language

1 ΕΙΣΑΓΩΓΗ

1.1 Εισαγωγή

Σκοπός της παρούσας διπλωματικής εργασίας ήταν ο σχεδιασμός και η υλοποίηση μιας πλατφόρμας διαχείρισης υπηρεσιών σε ευφυή περιβάλλοντα με τη χρήση τεχνολογιών ενδιάμεσου λογισμικού προσανατολισμένων σε υπηρεσίες.

Με τον όρο ευφυή περιβάλλοντα ο Mark Weiser περιέγραψε ένα φυσικό κόσμο που είναι στενά συνυφασμένος με αισθητήρες, εκκινητήρες, οθόνες και υπολογιστικά στοιχεία τα οποία βρίσκονται ενσωματωμένα στα αντικείμενα της καθημερινότητας και είναι διασυνδεδεμένα μέσω δικτύου (Weiser, 1999). Με βάση αυτό το όραμα δόθηκαν διάφοροι ορισμοί για τους χώρους αυτούς και αναπτύχθηκαν διάφορες ερευνητικές προσπάθειες τόσο από εταιρίες όσο και από ερευνητικά ιδρύματα. Σε κάθε περίπτωση πάντως οι χώροι αυτοί διαθέτουν κάποια κοινά χαρακτηριστικά που συνοψίζονται στα εξής :

- Συγχώνευση φυσικών και ψηφιακών χώρων (Ou, Karupiah, Fagg, & Riseman, 2004)
- Επίγνωση της δραστηριότητας του χρήστη και του περιβάλλοντος του (context-awareness) (Dey, 2001)
- Υποστήριξη των καθημερινών δραστηριοτήτων του πραγματικού κόσμου μέσω της ένωσης του φυσικού χώρου και της νοημοσύνης (Ma, και συν., 2005)

Για τη δημιουργία και την υποστήριξη έξυπνων χώρων χρησιμοποιούνται τεχνολογίες οι οποίες περιλαμβάνουν *ασύρματες συσκευές χειρός, έξυπνες συσκευές και αντιληπτικές διεπαφές*. Οι συσκευές χειρός, όπως κινητά τηλέφωνα και PDAs, δίνουν τη δυνατότητα για προσωποποίηση του κατόχου τους, προσφέρουν υπολογιστικούς πόρους και αποθήκευση δεδομένων, λειτουργούν σαν διεπαφές χρήστη, χρησιμοποιούν ποικίλα πρωτόκολλα ασύρματης δικτύωσης και καθώς είναι εύκολες στη μεταφορά, έχουν συνεχή παρουσία μέσα στην εμβέλεια ενός έξυπνου χώρου (Siegemund, Floerkemeier, & Vogt, 2005). Στις αντιληπτικές διεπαφές περιλαμβάνονται τεχνολογίες όπως, αναγνώριση φωνής

και κίνησης, συγχώνευση δεδομένων αισθητήρων, παρακολούθηση βλέμματος, κινητικότητα και δικτύωση.

Ένας έξυπνος χώρος ενσωματώνει πολλά ξεχωριστά στοιχεία επεξεργασίας τα οποία για να συνεργαστούν με ένα συνεπή και ενιαίο τρόπο χρειάζονται και την ανάλογη υποδομή λογισμικού. Μια τέτοια υποδομή πρέπει να πληροί κάποιες αρχές σχεδίασης όπως είναι, η χαλαρή σύζευξη, η επεκτασιμότητα, η υποστήριξη επικοινωνίας ένα προς πολλά, τόσο για μηνύματα όσο και για ροή δεδομένων καθώς και η υποστήριξη διαφορετικών πλατφόρμων εκτέλεσης (Xie, Shi, Xu, & Mao, 2002). Άλλα χαρακτηριστικά που έχουν επισημανθεί είναι η ενσωμάτωση και αλληλεπίδραση συσκευών, η διαχείριση και επεξεργασία πληροφοριών που αφορούν το χρήστη και το περιβάλλον του (context management), η διαλειτουργικότητα ετερογενών οντοτήτων, η εργασιοκεντρική και ανθρωποκεντρική ανάπτυξη διεπαφών χρήστη (Zhang, Zhu, Cheng, Koh, & Mokhtari, 2006) και ο συγχρονισμός και η συνεργασία των κατανεμημένων στοιχείων του συστήματος (Singh, Bhargava, & Kain, 2006).

Η υλοποίηση αρχιτεκτονικής έξυπνων χώρων προϋποθέτει τη χρήση τεχνολογιών λογισμικού που να καλύπτουν τις ιδιαίτερες ανάγκες τέτοιων συστημάτων. Μια πολύ διαδεδομένη τεχνολογία είναι το πλαίσιο OSGi (Open Service Gateway initiative) το οποίο συνεργάζεται πολύ καλά με το πρωτόκολλο UPnP (Universal Plug and Play). Η αρχιτεκτονική δικτύου Jini είναι μια ακόμα ενδεδειγμένη τεχνολογία για την υλοποίηση υποδομών έξυπνων χώρων. Δύο άλλες σχετικές τεχνολογίες είναι τα Web Services και το πλαίσιο JADE (Java Agent Development). Για την υλοποίηση διεπαφών χρήστη χρησιμοποιούνται γλώσσες σήμανσης όπως η XUL (XML User Interface Language), η XAML (Extensible Application Markup Language) και η UIML (User Interface Markup Language).

Με βάση τα παραπάνω, σχεδιάστηκε και υλοποιήθηκε μια πλατφόρμα έξυπνων χώρων (η IESAP – Intelligent Environment's Services Administration Platform) προσανατολισμένη σε υπηρεσίες. Η πλατφόρμα αυτή ικανοποιεί τις σημαντικότερες αρχές σχεδίασης ενός συστήματος έξυπνου χώρου και επιπλέον προτείνει ένα δυναμικό και επεκτάσιμο σύστημα αλληλεπίδρασης μεταξύ αισθητήρων και εκκινητήρων για την καλύτερη διαχείριση και επεξεργασία των

δεδομένων των αισθητήρων. Σαν περιβάλλον εφαρμογής της πλατφόρμας επιλέχθηκε ο χώρος του σπιτιού καθώς είναι αρκετά προφανείς οι ανάγκες του και διαθέτει ένα μεγάλο πεδίο εφαρμογών. Αυτοματοποιήθηκαν λειτουργίες όπως η ρύθμιση θερμοκρασίας, η ρύθμιση φωτεινότητας και η μεγιστοποίηση φωτεινότητας με την ανίχνευση κίνησης. Στην προτεινόμενη αρχιτεκτονική, διακρίνονται πέντε στρώματα. Το Φυσικό Στρώμα που αποτελείται από τις συσκευές, το Στρώμα Υπηρεσιών Αισθητήρων/Εκκινητήρων που περιλαμβάνει τις υπηρεσίες των συστατικών του φυσικού κόσμου, το Στρώμα Αντίληψης που ανιχνεύει τις συνθήκες του περιβάλλοντος και αποφασίζει για τις ενέργειες που πρέπει να γίνουν, το Στρώμα Διαχείρισης Υπηρεσιών που είναι υπεύθυνο για τη διαχείριση και το συγχρονισμό των υπηρεσιών και το Στρώμα Εφαρμογής που περιλαμβάνει τη διεπαφή χρήστη για τη διαχείριση των λειτουργιών της πλατφόρμας.

Για την υλοποίηση της προτεινόμενης αρχιτεκτονικής επιλέχθηκαν οι τεχνολογίες OSGi και UPnP. Πρόκειται για δύο πλήρως συμπληρωματικές τεχνολογίες, ο συνδυασμός των οποίων θεωρείται πολύ ισχυρός για την εκπλήρωση των στόχων μιας πλατφόρμας έξυπνων χώρων (Stavroulaki, Demestichas, Adamopoulou, & Demestichas, 2006).

Για να είναι εφικτή η δοκιμή λειτουργίας της πλατφόρμας, αναπτύχθηκε ένας προσομοιωτής περιβάλλοντος και εξομοιωτές διαφόρων τύπων UPnP συσκευών όπως λαμπτήρες, κλιματιστικά, ανιχνευτές κίνησης κ.α..

1.2 Δομή της εργασίας

Στο Κεφάλαιο 2 γίνεται μια ανασκόπηση της υπάρχουσας βιβλιογραφίας σε ερευνητικά θέματα των έξυπνων χώρων. Ειδικότερα ασχολούμαστε με τις τεχνολογίες που υλοποιούν αυτούς τους χώρους και τις αρχές σχεδίασης της αρχιτεκτονικής τους. Επιπλέον δίνονται κάποια παραδείγματα υλοποίησης και αρχιτεκτονικής σχεδίασης.

Στο Κεφάλαιο 3 παρουσιάζονται οι βασικές τεχνολογίες ανάπτυξης (OSGi, UPnP, Jini, Web Services, Jade).

Στο Κεφάλαιο 4 παρουσιάζεται η προτεινόμενη πλατφόρμα έξυπνων χώρων IESAP. Αρχικά γίνεται αναφορά στις προδιαγραφές και την αρχιτεκτονική της πλατφόρμας. Κατόπιν αναλύονται οι τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της και οι λόγοι για την επιλογή τους.

Στο κεφάλαιο 5 γίνεται μια αναλυτική περιγραφή της υλοποίησης τόσο της πλατφόρμας όσο και των περιφερειακών συστημάτων που αναπτύχθηκαν (προσομοιωτής περιβάλλοντος, εξομοιωτές συσκευών) για τη δοκιμή λειτουργίας της.

Στο Κεφάλαιο 6 παραθέτονται στιγμιότυπα (screenshots) από σενάρια εκτέλεσης των λειτουργιών της πλατφόρμας.

Στο Κεφάλαιο 7 γίνεται παρουσίαση των συμπερασμάτων και των προτάσεων για περαιτέρω ανάπτυξη της πλατφόρμας.

2 ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΕΠΙΣΚΟΠΗΣΗ

2.1 Διεσδυτική Πληροφορική

Η *διεσδυτική πληροφορική* (pervasive computing) αναφέρεται στο μοντέλο αλληλεπίδρασης ανθρώπου – μηχανής στο οποίο η επεξεργασία της πληροφορίας έχει ενσωματωθεί πλήρως στα αντικείμενα και τις δραστηριότητες της καθημερινότητας (Wikipedia, Ubiquitous Computing, 2009). Αντιθέτως με αυτό που συμβαίνει στο παραδοσιακό μοντέλο επικοινωνίας (desktop model), στο οποίο ένας χρήστης συνειδητά απασχολεί μια συσκευή για ένα συγκεκριμένο σκοπό, κάποιος που χρησιμοποιεί τη διεσδυτική πληροφορική, απασχολεί ταυτόχρονα πολλές υπολογιστικές συσκευές και συστήματα, σαν μέρος των καθημερινών του δραστηριοτήτων, χωρίς πιθανώς να αντιλαμβάνεται τη λειτουργία τους.

2.1.1 Ανασκόπηση

Ο Mark Weiser το 1991 περιέγραψε το όραμα του για αυτό που σήμερα ονομάζεται διεσδυτική πληροφορική. Σύμφωνα με αυτό σκοπός είναι η δημιουργία ενός περιβάλλοντος που κατακλύζεται από πληροφοριακές και επικοινωνιακές δυνατότητες, οι οποίες όμως ενοποιούνται αρμονικά με το περιβάλλον του χρήστη. Την εποχή εκείνη τα συστατικά για την πραγματοποίηση αυτού του οράματος έμοιαζαν να μην είναι διαθέσιμα. Σήμερα όμως με την ανάπτυξη νέων τεχνολογιών και την εμφάνιση εμπορικών συσκευών, όπως οι φορητές υπολογιστικές μηχανές και τα δίκτυα αισθητήρων, η επίτευξη του στόχου μοιάζει περισσότερο εφικτή (Satyanarayanan, 2001).

Πριν την εμφάνιση της διεσδυτικής πληροφορικής, η ανάπτυξη των *κατανεμημένων συστημάτων* (distributed systems) και της *κινητής υπολογιστικής* (mobile computing) άνοιξε τον δρόμο για την εξέλιξη που θα ακολουθούσε. Παρόλα αυτά, τα προβλήματα που υπήρχαν σε αυτές τις τεχνολογίες κληρονομήθηκαν αυτομάτως και στη νέα πρακτική. Σε μερικές περιπτώσεις οι λύσεις που είχαν δοθεί χρησιμοποιήθηκαν αυτούσιες και στη διεσδυτική

πληροφορική, σε άλλες όμως χρειάστηκε να αναπτυχθούν νέες καθώς οι απαιτήσεις ήταν διαφορετικές.

2.1.2 Σχετικές τεχνολογίες

Τα καταναμημένα συστήματα αποτελούνται από συστήματα υλικού (hardware) και λογισμικού (software) που περιέχουν περισσότερους από έναν επεξεργαστές, μονάδες αποθήκευσης, ταυτόχρονες διεργασίες ή πολλαπλά προγράμματα τα οποία εκτελούνται κάτω από ένα χαλαρό ή πολύ στενό έλεγχο. Η ανάπτυξη αυτών των συστημάτων γεφυρώνει πολλές περιοχές έρευνας οι οποίες είναι πολύ σημαντικές για τη διεισδυτική πληροφορική. Μερικές είναι οι ακόλουθες :

- Απομακρυσμένη επικοινωνία (Remote communication)
- Ανοχή λαθών (Fault tolerance)
- Υψηλή διαθεσιμότητα (High availability)
- Απομακρυσμένη πρόσβαση πληροφοριών (Remote information access)
- Ασφάλεια (Security)

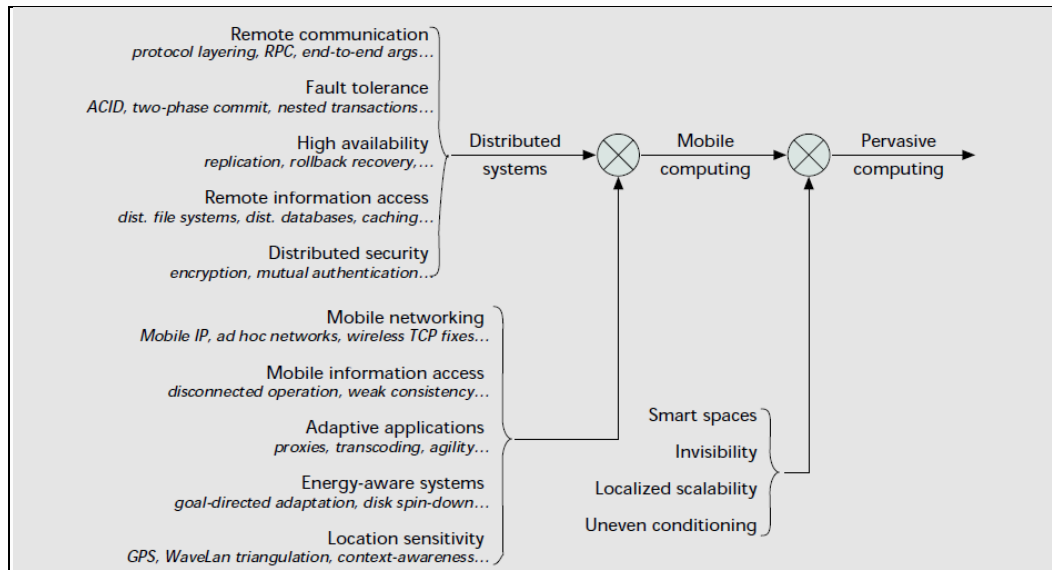
Ο όρος mobile computing περιγράφει τη δυνατότητα της εν' κινήσει χρήση της τεχνολογίας. Αν και οι βασικές αρχές των καταναμημένων συστημάτων έχουν εφαρμογή στο mobile computing, εντούτοις υπάρχουν τέσσερις περιοριστικοί παράγοντες στην ανάπτυξη συγκεκριμένων τεχνικών : απρόβλεπτη διακύμανση της ποιότητας του δικτύου, μειωμένη εμπιστοσύνη και αξιοπιστία των συστατικών του δικτύου, περιορισμοί μεγέθους και βάρους ως προς τους διαθέσιμους πόρους και ως προς την κατανάλωση ενέργειας. Παρόλα αυτά πρόκειται για μια περιοχή έρευνας που έχει δώσει πολύ σημαντικά αποτελέσματα όπως :

- Κινητή δικτύωση (Mobile networking)
- Κινητή πρόσβαση πληροφοριών (Mobile information access)
- Υποστήριξη προσαρμοστικών εφαρμογών (Support for adaptive applications)
- Τεχνικές εξοικονόμησης ενέργειας (System-level energy saving techniques)
- Ευαισθησία θέσης (Location sensitivity)

Εκτός των παραπάνω τεχνολογιών, η διεισδυτική πληροφορική ενσωματώνει και τέσσερις ακόμα περιοχές έρευνας :

- Αποτελεσματική χρήση των *έξυπνων χώρων* (Effective use of smart spaces). Με τον όρο *χώρο* εννοούμε μια κλειστή περιοχή όπως είναι ένα δωμάτιο, ή μια καλά ορισμένη περιοχή όπως μια αυλή. Η ενσωμάτωση της τεχνολογίας σε μια κατασκευή (π.χ. ένα κτίριο), φέρνει τη συγχώνευση αυτών των δύο κόσμων και ενεργοποιεί την παρακολούθηση και τον έλεγχο του ενός (κτίριο) από τον άλλο (τεχνολογία).
- Αορατότητα (Invisibility). Πρόκειται για την όσο το δυνατόν μικρότερη ενόχληση του χρήστη από την παρουσία των συστατικών της διεισδυτικής πληροφορικής.
- Τοπική κλιμάκωση (Localized scalability). Στις περιπτώσεις των έξυπνων χώρων, ο βαθμός της αλληλεπίδρασης του χρήστη με το περιβάλλον του είναι πολύ μεγάλος. Καθώς μάλιστα σε τέτοια περιβάλλοντα μπορεί να βρίσκονται περισσότεροι χρήστες, οι οποίοι μάλιστα μετακινούνται συνεχώς, η κλιμάκωση αυτής της αλληλεπίδρασης έχει πολύ μεγάλη εξάρτηση από την απόσταση των δύο μερών.
- Κάλυψη ανομοιογενούς συμπεριφοράς (Masking uneven condition). Ο βαθμός διείσδυσης των τεχνολογιών της διεισδυτικής πληροφορικής ποικίλει και εξαρτάται από διάφορους μη τεχνικούς παράγοντες, όπως η οργανωτική δομή, τα οικονομικά και επιχειρηματικά μοντέλα. Ο εξοπλισμός ενός έξυπνου χώρου είναι διαφορετικός αν πρόκειται για αίθουσα συναντήσεων, γραφείο ή αίθουσα διδασκαλίας. Αυτή η διαφοροποίηση των συστατικών δεν πρέπει να είναι ενοχλητική για τον χρήστη και δεν πρέπει να εμποδίζει αυτό που τελικά είναι ο σκοπός της διεισδυτικής πληροφορικής, δηλαδή την «εξαφάνιση» της τεχνολογίας.

Οι προαναφερθείσες περιοχές έρευνας που σχετίζονται με την διεισδυτική πληροφορική και η σχέση που τις συνδέει, φαίνονται στην Εικόνα 1.



Εικόνα 1. Ταξινόμηση των περιοχών έρευνας της διεισδυτικής πληροφορικής (Satyanarayanan, 2001)

2.2 Έξυπνοι χώροι

Όπως αναφέρθηκε και παραπάνω (2.1.2), μια από τις περιοχές έρευνας που ενσωματώνει η διεισδυτική πληροφορική, στην προσπάθεια της να καταστήσει την πληροφορία διαθέσιμη παντού και πάντα, είναι η χρήση των έξυπνων χώρων.

2.2.1 Ορισμοί

Εξαιτίας της ποικιλίας των πραγματικών χώρων και περιβαλλόντων, καθώς και των διαφορετικών ερευνητικών υποβάθρων και στόχων, υπάρχουν πολλοί ορισμοί των έξυπνων χώρων. Παρακάτω παρατίθενται μερικοί από αυτούς.

1. Με τον όρο έξυπνοι χώροι περιγράφονται συνηθισμένοι χώροι οι οποίοι είναι εξοπλισμένοι με συστήματα οπτικών και ηχητικών αισθητήρων και μπορούν να αντιλαμβάνονται και να αντιδρούν με τους ανθρώπους χωρίς αυτοί να χρειάζεται να φέρουν ειδικό εξοπλισμό. Διεισδυτικές συσκευές, αισθητήρες, και δίκτυα, παρέχουν την κατάλληλη υποδομή για έξυπνους χώρους που αντιλαμβάνονται τις ανθρώπινες δραστηριότητες και αντιδρούν με αυτές (Singh, Bhargava, & Kain, 2006).

2. Πρόκειται για περιβάλλοντα που έχουν επίγνωση των χρηστών με τους οποίους αλληλεπιδρούν και είναι ικανά για απαρακόλυτη και ευφυή αλληλεπίδραση (Essa, 2000).
3. Οι ενεργοί (έξυπνοι) χώροι είναι προεκτάσεις των φυσικών χώρων οι οποίοι έχουν τη δυνατότητα να αντιλαμβάνονται τις ενέργειες των χρηστών και είναι εξοπλισμένοι με μεγάλη ποικιλία συσκευών που βοηθούν τους χρήστες στις διάφορες δραστηριότητες τους (Roman, Hess, Cerqueira, Ranganat, Campbell, & Nahrstedt, 2002).
4. Τα έξυπνα περιβάλλοντα συνδυάζουν αντιληπτικές και συλλογιστικές ικανότητες με άλλα στοιχεία της διεισδυτικής πληροφορικής σε μια προσπάθεια να δημιουργηθεί ένα ανθρωποκεντρικό σύστημα το οποίο να εμπεριέχεται σε φυσικούς χώρους (Shrobe, 2005).
5. Οι έξυπνοι χώροι είναι περιβάλλοντα εργασίας με ενσωματωμένους υπολογιστές, πληροφοριακές συσκευές και αισθητήρες, που επιτρέπουν στους ανθρώπους να εκτελούν τις εργασίες τους αποτελεσματικά προσφέροντας τους πρωτοφανή επίπεδα πρόσβασης στην πληροφορία και βοήθεια από υπολογιστές (Rosenthal & Stanford, 2000).

2.2.2 Ιστορική αναδρομή ερευνητικών έργων

Η έρευνα σχετικά με τους έξυπνους χώρους ξεκινά το 1993 με ερευνητικές προσπάθειες όπως (Ma, και συν., 2005):

- Boulder's Adaptive House
- Buxton's Reactive Environment

Στο τέλος της δεκαετίας του 1990, τα πιο γνωστά ερευνητικά έργα ήταν τα :

- Georgia Tech's Aware Home
- Inria's Smart Office
- Stanford's iRoom

- Cisco's Internet Home
- Essex's Intelligent Inhabited Environments
- H-P's Cool Town

Μετά το 2000, έχουν ξεκινήσει πολλά έργα σε πανεπιστήμια και μεγάλες εταιρείες για την ανάπτυξη διαφόρων έξυπνων χώρων και περιβαλλόντων, συμπεριλαμβανομένων των :

- ATR's Creative Space
- CMU's Aura
- Xerox's Smart Media Spaces
- IBM's DreamSpace
- KTH's comHOME
- Microsoft's EasyLiving
- MIT's Oxygen
- Philips's Home of the Future
- UW CSE's Portolano
- Intel's Proactive Health
- UF's Assistive Smart House
- Keio's SSLab

2.2.3 Τα συστατικά των έξυπνων χώρων και η κατηγοριοποίηση τους

Ανεξαρτήτως των διαφορετικών ορισμών, υπάρχουν κάποια βασικά και ουσιώδη χαρακτηριστικά τα οποία είναι κοινά στους έξυπνους χώρους. Πρώτα, είναι το φυσικό περιβάλλον το οποίο είναι εξοπλισμένο με ηλεκτρονικές συσκευές και ενσωματωμένα συστήματα, τα οποία μπορεί να είναι διαφορετικού σχήματος, μεγέθους, μορφής και λειτουργικότητας. Αυτές οι συσκευές και τα συστήματα είναι διασυνδεδεμένα με καλώδια ή/και με ασύρματα δίκτυα δημιουργώντας έτσι ένα εικονικό ψηφιακό περιβάλλον, το οποίο πρέπει να είναι σε μεγάλο βαθμό ενοποιημένο με το φυσικό περιβάλλον. Πρόκειται δηλαδή για μια συγχώνευση φυσικών και ψηφιακών χώρων (Ou, Karupiah, Fagg, & Riseman, 2004). Έπειτα, οι χώροι αυτοί πρέπει να διαθέτουν μια σχετική ικανότητα αντίληψης, επίγνωσης,

ανάλυσης, συλλογιστικής και πρόβλεψης για την ύπαρξη ενός χρήστη και του περιβάλλοντος του, πάνω στο οποίο μπορούν κατά περίπτωση να επέμβουν. Αυτές οι ικανότητες μπορούν να θεωρηθούν σαν ένα είδος *νοημοσύνης*, όπως άλλωστε φανερώνει και το προσδιοριστικό των χώρων αυτών δηλ. «έξυπνοι». Το πιο βασικό χαρακτηριστικό αυτής της νοημοσύνης είναι η *context επίγνωση* (*context-awareness*) του πραγματικού χώρου (Dey, 2001). Λέγοντας *context* εννοούμε κάθε πληροφορία που αφορά το χρήστη και το περιβάλλον του, όπως για παράδειγμα η θέση του στο χώρο, η δραστηριότητα του, η θερμοκρασία και ο φωτισμός του περιβάλλοντος (Zhang, Zhu, Cheng, Koh, & Mokhtari, 2006). Τρίτο χαρακτηριστικό των χώρων αυτών είναι η προσπάθεια τους να προσαρμοστούν περισσότερο στην ανθρώπινη δραστηριότητα παρά το αντίστροφο, προσφέροντας καλύτερες υπηρεσίες στους χρήστες στο καθημερινό τους περιβάλλον χωρίς να τους αναγκάζουν στη χρήση προσωπικών και φορητών υπολογιστών. Έτσι τους παρέχουν ικανοποιητική υποστήριξη στις διάφορες καθημερινές δραστηριότητες του πραγματικού κόσμου, μέσω της ένωσης του χώρου και της «έξυπνάδας» (Ma, και συν., 2005).

Αν θελήσουμε να ταξινομήσουμε τους έξυπνους χώρους, τότε αυτό μπορεί να γίνει με διάφορους τρόπους, όπως βάση της λειτουργίας τους, των χωρικών χαρακτηριστικών τους και των λειτουργιών και τεχνολογιών τους.

Ταξινόμηση βάση της λειτουργίας :

- Δωμάτιο, σπίτι, γραφείο, εργαστήριο, αίθουσα διδασκαλίας
- Κτίριο, βιβλιοθήκη, σχολείο, πανεπιστήμιο, εργοστάσιο
- Κατάστημα, εστιατόριο, ξενοδοχείο, κλινική, νοσοκομείο
- Δρόμος, αυλή, πάρκο, αθλητικός χώρος, πόλη
- Όχημα, δρόμος, σιδηρόδρομος, σταθμός, αεροδρόμιο
- Εξοχή, βουνό, πισίνα, λίμνη, ποτάμι

Ταξινόμηση βάση χωρικών χαρακτηριστικών :

- Μικρό ή μεγάλο
- Κλειστό ή ανοικτό
- Σταθερό ή κινητό

- Σχήμα και διάσταση
- Χώρισμα και διάταξη
- Θέση και σχέση με τα αντικείμενα

Ταξινόμηση βάση λειτουργιών και τεχνολογιών.

- Ιδιωτικό ή δημόσιο
- Συγκεκριμένη ή γενική υπηρεσία
- Συγκεκριμένοι χρήστες (παιδιά, ηλικιωμένοι, μαθητές, ασθενείς, κ.α)
- Πλήθος και δραστηριότητα χρηστών
- Έξυπνες συσκευές και υπολογιστές που χρησιμοποιούνται
- Δίκτυα και μέσα που χρησιμοποιούνται
- Είδος context, πλήθος και χρήση

Για παράδειγμα, ένα συνηθισμένο σπίτι είναι ένας σταθερός, κλειστός και ιδιωτικός χώρος με χωρίσματα και πολλά μέλη μιας οικογένειας. Ένα γραφείο και ένα εργαστήριο είναι παρόμοια με το σπίτι άλλα με διαφορετικές λειτουργίες και σκοπούς. Ένα αυτοκίνητο είναι ένας κινητός, ιδιωτικός και μικρός χώρος. Ένα πολυκατάστημα είναι ένας σταθερός, δημόσιος και μεγάλος χώρος συχνά με πολλούς χρήστες. Αυτή η μεγάλη ανομοιομορφία και η ετερογένεια των πραγματικών χώρων είναι μια από τις μεγάλες δυσκολίες για την ανάπτυξη ενός γενικού ερευνητικού συστήματος έξυπνων χώρων (Ma, και συν., 2005).

2.2.4 Τεχνολογίες υποδομής έξυπνων χώρων

Η υλοποίηση των έξυπνων χώρων απαιτεί την εφαρμογή και τη συνεργασία πολλών διαφορετικών τεχνολογιών. Μερικές από αυτές περιγράφονται παρακάτω.

2.2.4.1 Συσκευές

Διεισδυτικές Συσκευές. Σε ένα έξυπνο περιβάλλον είναι πιθανό να υπάρχουν πολλών ειδών διεισδυτικές συσκευές. Οι δύο βασικές κατηγορίες στις οποίες χωρίζονται οι συσκευές αυτές είναι, οι *ασύρματες συσκευές χειρός* (handheld devices) και οι *έξυπνες συσκευές*.

Στην πρώτη κατηγορία ανήκουν τα κινητά τηλέφωνα, οι υπολογιστές τσέπης (palmtop), τα PDAs κ.α. Ο ρόλος των συσκευών αυτών στα ευφυή περιβάλλοντα είναι πολύ σημαντικός, αφού είναι δυνατόν να αποτελούν τη βασική διεπαφή του χρήστη, αλλά και να λειτουργούν για το σύστημα σαν ένα κινητό σημείο πρόσβασης. Επίσης είναι δυνατόν να παρέχουν ενέργεια, υπολογιστικούς πόρους και κινητή αποθήκευση δεδομένων, να διενεργούν αυθεντικοποίηση χρηστών και να διαθέτουν δυνατότητες εντοπισμού. Κάποια από τα βασικά χαρακτηριστικά αυτών των συσκευών που είναι πολύ σημαντικά για τους έξυπνους χώρους είναι τα ακόλουθα (Siegemund, Floerkemeier, & Vogt, 2005) :

Συνεχής παρουσία (habitual presence). Συσκευές όπως κινητά τηλέφωνα και PDAs καθώς μεταφέρονται από τους ιδιοκτήτες τους, βρίσκονται σχεδόν πάντα μέσα στην εμβέλεια ενός έξυπνου χώρου. Μπορούν έτσι να αποτελέσουν τον ενδιάμεσο μεταξύ, των έξυπνων συσκευών του συστήματος, των χρηστών και των υπηρεσιών υποδομής.

Ποικιλομορφία ασύρματων δικτύων. Τα κινητά τηλέφωνα και τα PDAs υποστηρίζουν τεχνολογίες επικοινωνίας τόσο για δίκτυα ευρείας εμβέλειας όσο και για περιορισμένης, π.χ. GSM, UMTS, Bluetooth, WLAN. Το χαρακτηριστικό αυτό, τους δίνει τη δυνατότητα να επικοινωνούν και να ανταλλάσσουν δεδομένα τόσο με συσκευές που βρίσκονται σε κοντινή απόσταση, όσο και με απομακρυσμένους ισχυρούς υπολογιστές, λειτουργώντας έτσι σαν κινητά σημεία πρόσβασης.

Διεπαφή χρήστη και δυνατότητες εισαγωγής δεδομένων. Οι χρήστες είναι εξοικειωμένοι με τη λειτουργικότητα των συσκευών χειρός και για αυτό μπορούν οι συσκευές αυτές να αποτελέσουν τη διεπαφή μεταξύ των χρηστών και των υπολοίπων συστατικών ενός έξυπνου χώρου. Επίσης μπορούν να λειτουργήσουν και σαν συσκευές εισαγωγής δεδομένων.

Αντίληψη. Οι συσκευές χειρός, αναλόγως και της λειτουργικότητας τους, μπορούν να παίξουν το ρόλο του απομακρυσμένου αισθητήρα. Για παράδειγμα, ένα κινητό τηλέφωνο έχοντας τη δυνατότητα κάθε στιγμή να γνωρίζει σε ποια κυψέλη ανήκει, μπορεί να αποτελέσει ένα είδος απομακρυσμένου αισθητήρα θέσης.

Υπολογιστικοί πόροι και αποθήκευση δεδομένων. Σε συσκευές όπως κινητά τηλέφωνα και PDAs, παρόλο που η κατανάλωση ενέργειας πρέπει να είναι μικρή, οι χρήστες τους συνηθίζουν να τις φορτίζουν πολύ τακτικά. Το γεγονός αυτό δίνει τη δυνατότητα σε άλλες έξυπνες συσκευές του περιβάλλοντος, που βρίσκονται μέσα στην εμβέλεια τους και δεν διαθέτουν επαρκείς πόρους, να τους ζητούν την εκτέλεση πολύπλοκων και ενεργειακά δαπανηρών υπολογισμών.

Προσωπικοποίηση. Τα κινητά τηλέφωνα και τα PDAs ανήκουν σε ένα συγκεκριμένο πρόσωπο, το οποίο χρησιμοποιεί τη συσκευή αποκλειστικά. Έτσι οι άλλες έξυπνες συσκευές του περιβάλλοντος μπορούν να προσαρμόσουν τη συμπεριφορά τους ανάλογα με τις συσκευές χειρός που βρίσκονται μέσα στην εμβέλεια τους και να προσφέρουν υπηρεσίες σε συγκεκριμένους ανθρώπους.

Η δεύτερη κατηγορία των διεισδυτικών συσκευών περιλαμβάνει τις λεγόμενες έξυπνες ή ευφυείς συσκευές. Πρόκειται για συσκευές που έχουν επίγνωση για το περιβάλλον τους και το context του (Schmidt & Laerhoven, 2001). Στο έξυπνο σπίτι του Helal και συν. περιγράφονται μερικές τέτοιες συσκευές και οι λειτουργίες τους (Helal, Mann, El-Zabadani, King, Kaddoura, & Jansen, 2005).

Έξυπνο γραμματοκιβώτιο. Γραμματοκιβώτιο που αντιλαμβάνεται την ύπαρξη φακέλου και ειδοποιεί τους ενοίκους του σπιτιού.

Έξυπνη εξωτερική πόρτα. Πόρτα εξοπλισμένη με RFID ετικέτα που αντικαθιστά τα κλειδιά και επιτρέπει μόνο σε εξουσιοδοτημένα άτομα την είσοδο. Εφοδιασμένη με μικρόφωνο, κάμερα, αυτόματο ενεργοποιητή (για το άνοιγμα) και ηχεία, επιτρέπει στους ένοικους του σπιτιού να επικοινωνούν με τους επισκέπτες τους.

Έξυπνο ρολό παραθύρου. Ρολό που μπορεί να προγραμματιστεί ή να προσαρμοστεί μέσω τηλεχειρισμού για να ελέγχει το φώς και να παρέχει ιδιωτικότητα.

Έξυπνος καθρέπτης. Καθρέπτης που προβάλλει μηνύματα και υπενθυμίσεις όπως για παράδειγμα τη λήψη κάποιου φαρμάκου.

Εξυπνος φούρνος. Φούρνος μικροκυμάτων που ρυθμίζει αυτόματα το χρόνο και την ένταση ψησίματος για κάθε κατεψυγμένο φαγητό και προβάλλει στους χρήστες οδηγίες για την προετοιμασία και το μαγείρεμα του.

Εξυπνο ψυγείο. Ψυγείο που παρακολουθεί τη διαθεσιμότητα και την κατανάλωση τροφίμων, εντοπίζει τα ληγμένα τρόφιμα, δημιουργεί λίστες αγορών και παρέχει συμβουλές για την προετοιμασία του γεύματος αναλόγως της επάρκειας των υλικών.

Εξυπνο πάτωμα. Αισθητήρες στο πάτωμα αναγνωρίζουν και παρακολουθούν την τοποθεσία των ενοίκων.

Εξυπνο τηλέφωνο. Τηλέφωνο που εκτός των συμβατικών λειτουργιών, παρέχει υπηρεσίες τηλεχειριστηρίου για όλες τις συσκευές και τους αναπαραγωγείς πολυμέσων. Μπορεί επίσης να μεταδώσει υπενθυμίσεις και σημαντικές πληροφορίες στους ένοικους, ενώ αυτοί απουσιάζουν.

Εξυπνη πρίζα. Αισθητήρας πίσω από την πρίζα αναγνωρίζει την παρουσία μιας ηλεκτρικής συσκευής ή μιας λάμπας και τη συνδέει σε μια απομακρυσμένη εφαρμογή παρακολούθησης και ελέγχου.

Εξυπνος θερμοστάτης. Θερμοστάτης που εξατομικεύει τις ρυθμίσεις του air-condition και της θέρμανσης, ανάλογα με τις καθημερινές δραστηριότητες των ενοίκων.

Εξυπνο σύστημα παρακολούθησης. Σύστημα που παρακολουθεί όλα τα παράθυρα και τις πόρτες ενός σπιτιού και κατόπιν αιτήματος ειδοποιεί τους ενοίκους αν είναι ανοικτές ή ξεκλείδωτες.

2.2.4.2 Αντιληπτικές διεπαφές

Αναγνώριση φωνής (Speech recognition). Αναγνώριση φωνής ή λόγου είναι η ικανότητα μιας μηχανής ή ενός προγράμματος να εκτελεί φωνητικές εντολές ή να δέχεται υπαγόρευση. Για να επιτευχθεί κάτι τέτοιο γίνεται αντιστοίχιση του ηχητικού υποδείγματος με ένα υπάρχον λεξιλόγιο. Συνήθως το λεξιλόγιο αυτό

είναι αρκετά περιορισμένο, αλλά υπάρχει η δυνατότητα ο χρήστης να καταχωρήσει επιπρόσθετες λέξεις.

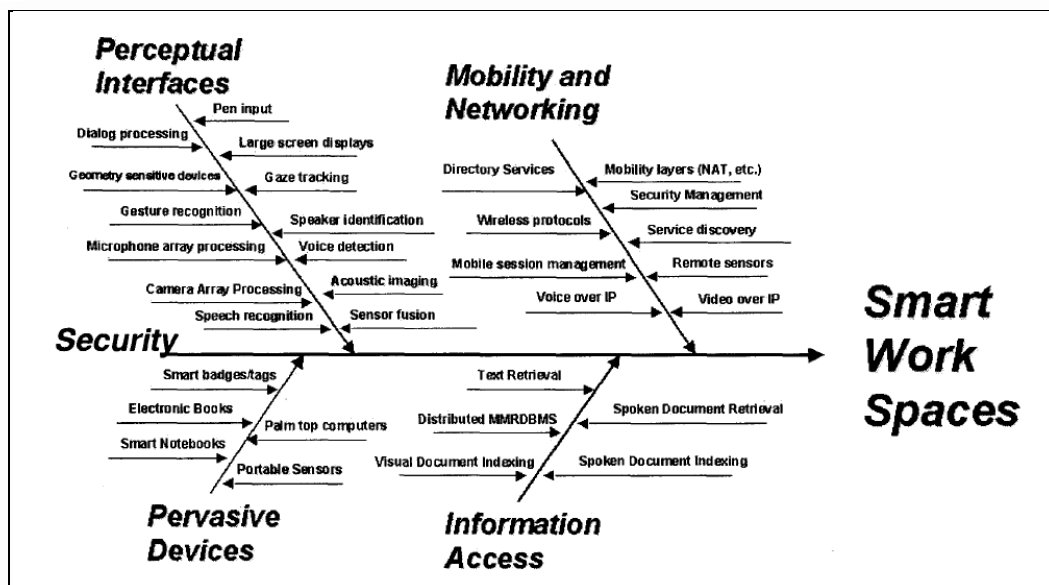
Αναγνώριση ομιλητή (Speaker identification). Πρόκειται για την υπολογιστική εργασία επαλήθευσης της ισχυριζόμενης ταυτότητας ενός χρήστη, χρησιμοποιώντας χαρακτηριστικά της φωνή του.

Αναγνώριση κίνησης (Gesture recognition). Πρόκειται για την αλληλεπίδραση ανθρώπου μηχανής κατά την οποία οι ανθρώπινες κινήσεις αναγνωρίζονται από τον υπολογιστή με τη βοήθεια μαθηματικών αλγορίθμων. Οι κινήσεις μπορεί να προέρχονται από οποιαδήποτε μέρος του σώματος, αλλά συνήθως αφορούν το πρόσωπο ή τα χέρια.

Συγχώνευση δεδομένων αισθητήρων (Sensor fusion). Η μετατροπή ακατέργαστων δεδομένων σε πληροφορία context υψηλού επιπέδου, όπως η τρέχουσα δραστηριότητα του χρήστη, απαιτεί εφαρμογές που προ-επεξεργάζονται τα δεδομένα. Αυτό σημαίνει φιλτράρισμα, μετατροπή ακόμα και συγχώνευση δεδομένων προερχόμενων από ίδιου ή διαφορετικού τύπου αισθητήρες, έτσι ώστε να βελτιωθεί η ποιότητα των παρεχομένων δεδομένων.

Παρακολούθηση βλέμματος (Gaze tracking). Ένα σύστημα που πρέπει να παρακολουθεί ένα χρήστη, δεν αρκεί μόνο να τον παρατηρεί, αλλά και να προσμένει τις αντιδράσεις του. Οι κάμερες ανίχνευσης βλέμματος παρακολουθούν την κίνηση των ματιών του χρήστη προσπαθώντας να κατανοήσουν τις προθέσεις του.

Κινητικότητα και δικτύωση. (Mobility and networking). Σε ένα ιδανικό έξυπνο περιβάλλον, χρειάζεται η ανάπτυξη ενός μεγάλου πλήθους ετερογενών, διασυνδεδεμένων, έξυπνων συσκευών οι οποίες σε συνεργασία παρέχουν απρόσκοπτες υπηρεσίες στους χρήστες. Συνεπώς οι έξυπνοι χώροι χρειάζεται να διαθέτουν διαλειτουργικότητα, κλιμάκωση, εξυπνάδα και αορατότητα για να διασφαλίσουν ότι οι χρήστες θα έχουν απρόσκοπτη πρόσβαση σε υπολογιστική ισχύ οποτεδήποτε τη χρειαστούν.



Εικόνα 2. Οι διάφορες τεχνολογίες που ενσωματώνουν οι έξυπνοι χώροι (Rosenthal & Stanford, 2000)

2.3 Παρουσίαση εφαρμογών έξυπνων χώρων

Παρακάτω παρατίθενται μερικά παραδείγματα υλοποίησης έξυπνων χώρων. Πιο συγκεκριμένα πρόκειται για εφαρμογές της συγκεκριμένης τεχνολογίας σε χώρους όπως είναι ένα μουσείο, μια αίθουσα εργασίας και μια αίθουσα διδασκαλίας.

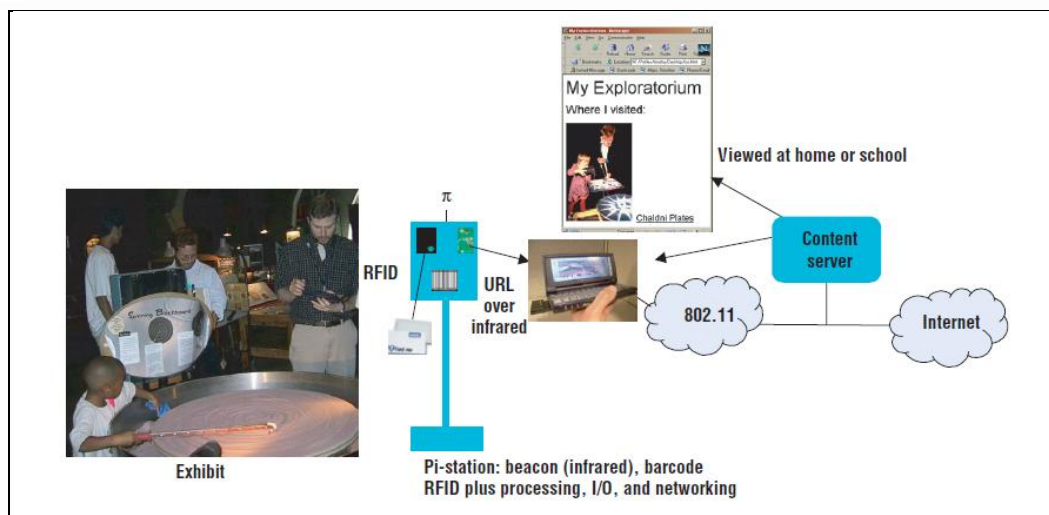
2.3.1 Το έξυπνο μουσείο

Σε αυτή την ερευνητική προσπάθεια (Fleck, Frid, Kindberg, O'Brien-Strain, Rajani, & Spasojevic, 2002) αναπτύχθηκαν κάποια εργαλεία για ένα μουσείο έρευνας και τεχνολογίας (The Exploratorium). Συνοπτικά τα εργαλεία αυτά είναι :

- *Informer*. Τροφοδοτεί το χρήστη με πληροφορίες για το έκθεμα που επισκέπτεται
- *Suggester*. Προτείνει ιδέες για το πώς να επεξεργαστεί ο χρήστης το έκθεμα που επισκέπτεται
- *Guider*. Προτείνει στο χρήστη τη σειρά επίσκεψης των διαφόρων εκθεμάτων
- *Communicator*. Επιτρέπει στους χρήστες να επικοινωνούν μεταξύ τους και να ανταλλάσσουν απόψεις σχετικές με τα εκθέματα

- *Rememberer*. Δίνει τη δυνατότητα στο χρήστη να δημιουργήσει αρχείο με τις εμπειρίες του από το μουσείο, το οποίο μπορεί να συμβουλευτεί κατά τη διάρκεια ή μετά την επίσκεψη του.

Ένας πρότυπος ηλεκτρονικός οδηγός συνδυάζει τις τρεις λειτουργίες, πληροφόρηση (informing), πρόταση (suggesting) και απομνημόνευση (remembering). Οι λειτουργίες αυτές παρέχονται στο χρήστη με τη μορφή ιστοσελίδων, τις οποίες μπορεί να προσπελάσει με τη χρήση PDA ενώ βρίσκεται στο μουσείο ή με υπολογιστή πριν και μετά την επίσκεψη του. Ο οδηγός τροφοδοτεί το χρήστη με περιεχόμενο σχετικό με τα εκθέματα ενώ εκείνος τα επισκέπτεται, καθώς και με ένα αρχείο σε μορφή δικτυακού άλμπουμ (scrapbook) για να καταγράψει τις εντυπώσεις του και να τις ανακτήσει οποτεδήποτε το θελήσει μετά την επίσκεψη του. Η τροφοδότηση γίνεται με αποστολή στατικών ιστοσελίδων στο PDA του χρήστη. Κάθε έκθεμα διαθέτει το δικό του ιστοχώρο ο οποίος περιέχει συνδέσμους σε άλλες ιστοσελίδες με πληροφορίες σχετικές με το έκθεμα αλλά και προτάσεις για το πώς να το περιεργαστεί. Ο μηχανισμός ανάκτησης του ιστοχώρου του εκθέματος περιλαμβάνει την εξής διαδικασία. Ο χρήστης ανακτά το μοναδικό αναγνωριστικό του εκθέματος πλησιάζοντας τη φορητή συσκευή η οποία είναι εφοδιασμένη με αισθητήρα, κοντά στο έκθεμα. Κατόπιν μέσω ειδικής υπηρεσίας το αναγνωριστικό μεταφράζεται σε URL το οποίο και αντιστοιχεί στον ιστοχώρο του εκθέματος. Γνωρίζοντας το URL ο χρήστης μπορεί πλέον να ανακτήσει στο PDA του τον ιστοχώρο του εκθέματος.



Εικόνα 3. Επισκόπηση του συστήματος του οδηγού (Fleck, Frid, Kindberg, O'Brien-Strain, Rajani, & Spasojevic, 2002)

2.3.2 Διαδραστικός χώρος εργασίας

Ένας πρότυπος διαδραστικός χώρος εργασίας με το όνομα iRoom αναπτύχθηκε από ερευνητική ομάδα του πανεπιστημίου του Stanford (Johanson, Fox, & Winograd, 2002). Πρόκειται για ένα δωμάτιο το οποίο περιέχει τρεις οθόνες αφής οι οποίες είναι τοποθετημένες κατά μήκος του ενός τοίχου και μια τέταρτη οθόνη ειδικής κατασκευής που αποκαλείται «διαδραστική τοιχογραφία» (interactive mural). Επιπλέον, υπάρχει ένα τραπέζι με ενσωματωμένη οθόνη, το οποίο είναι σχεδιασμένο να μοιάζει με κανονικό τραπέζι συνεδριάσεων. Το δωμάτιο είναι ακόμα εξοπλισμένο με κάμερες, μικρόφωνα, υποστήριξη για ασύρματο τοπικό δίκτυο, πολλά ασύρματα κουμπιά και άλλες διαδραστικές συσκευές. Η υποδομή του iRoom περιλαμβάνει ακόμα PDAs, σταθμούς εργασίας και φορητούς υπολογιστές τα οποία χρησιμοποιούνται ταυτοχρόνως και εκτελούν ετερογενείς εφαρμογές. Όλα τα προηγούμενα πρέπει είναι προσβάσιμα το ένα από το άλλο με έναν καθορισμένο τρόπο, έτσι ώστε ο χρήστης να τα χρησιμοποιεί σαν μια ομοιόμορφη συλλογή συσκευών.

Το iRoom παρέχει τρεις βασικές διευκολύνσεις στο χρήστη :

- Μετακίνηση δεδομένων από μια εφαρμογή που εκτελείται σε ένα PDA σε ένα άλλο
- Έλεγχο κάθε συσκευής ή εφαρμογής από τη θέση στην οποία βρίσκεται

- Υποστήριξη συνεργασίας των διαφόρων εφαρμογών

Στους χρήστες παρέχεται η δυνατότητα να αναγνωρίζουν τις διάφορες συσκευές του δωματίου, μέσω μιας εφαρμογής κεντρικής διαχείρισης η οποία χρησιμοποιεί ένα μικρό χάρτη του χώρου και υποδεικνύει που βρίσκονται τα φώτα, οι προβολείς και οι οθόνες. Με τη βοήθεια διακοπών και μενού επιλογών, τα οποία σχετίζονται με τις διάφορες συσκευές του χάρτη, υπάρχει η δυνατότητα να ενεργοποιηθούν ή να απενεργοποιηθούν οι προβολείς, οι οθόνες και τα φώτα. Αργότερα η εφαρμογή επεκτάθηκε έτσι ώστε να διαχειρίζεται και ένα πιο ευρύ φάσμα συσκευών με τη χρήση ιστοσελίδων και java applets.



Εικόνα 4. Άποψη του iRoom (Johanson, Fox, & Winograd, 2002)

2.3.3 Η έξυπνη τάξη

Η έξυπνη τάξη (Shi, και συν., 2003) που αναπτύχθηκε από ερευνητική ομάδα του πανεπιστημίου Tsinghua, περιλαμβάνει μια αίθουσα διδασκαλίας η οποία είναι εξοπλισμένη με δύο μεγάλες οθόνες προβολής. Η μια είναι τοποθετημένη στο μπροστινό τοίχο της αίθουσας και προβάλλει το εκπαιδευτικό υλικό του μαθήματος, ενώ η άλλη βρίσκεται στον πλαϊνό τοίχο και προβάλλει τις

απομακρυσμένες εικόνες των μαθητών καθώς και έναν εικονικό βοηθό καθηγητή. Οι απομακρυσμένοι μαθητές που έχουν πρόσβαση στην τάξη μέσω των υπολογιστών τους, μπορούν να δουν το περιεχόμενο του πίνακα με τη χρήση ενός ειδικού προγράμματος. Η αίθουσα είναι εξοπλισμένη με πολλές κάμερες, οι οποίες καταγράφουν τις κινήσεις του διδάσκοντα και τις μετατρέπουν σε εντολές για το ειδικό τηλε-εκπαιδευτικό πρόγραμμα που εκτελείται. Η εικόνα από τις κάμερες, καθώς και ο ήχος που καταγράφεται από τα ασύρματα μικρόφωνα που φέρει ο διδάσκοντας, μεταδίδονται σε πραγματικό χρόνο στους απομακρυσμένους μαθητές.

Για οθόνη προβολής του εκπαιδευτικού υλικού χρησιμοποιήθηκε μια οθόνη αφής, η οποία δίνει τη δυνατότητα στο διδάσκοντα να χρησιμοποιεί ψηφιακά στυλό και σβηστήρες για να γράψει ή να διαγράψει σημειώσεις από τον πίνακα. Τόσο οι απομακρυσμένοι όσο και οι τοπικοί μαθητές έχουν την ίδια εικόνα σε ότι αφορά το υλικό της παρουσίασης και τα περιεχόμενα του πίνακα. Ένα ειδικό στυλό laser παίζει το ρόλο του ποντικιού και δίνει τη δυνατότητα στο διδάσκοντα να επισημαίνει σημεία (με κόκκινο χρώμα) στον πίνακα, τα οποία είναι ορατά και από τους τοπικούς αλλά και από τους απομακρυσμένους μαθητές. Συστήματα αναγνώρισης ομιλίας επιτρέπουν στο διδάσκοντα να δίνει φωνητικές εντολές στις διάφορες συσκευές (όπως στον προβολέα για να προχωρήσει στην επόμενη διαφάνεια), καθώς και να τον ενημερώνουν με ηχητικά μηνύματα όταν συμβαίνουν συγκεκριμένα γεγονότα. Η χρήση της έξυπνης τάξης επιτρέπεται μόνο μετά από αυθεντικοποίηση του διδάσκοντα και αυτό γίνεται με τεχνικές όπως αναγνώριση προσώπου και φωνής.



Εικόνα 5. Άποψη της έξυπνης τάξης (Shi, και συν., 2003)

2.4 Αρχιτεκτονική έξυπνων χώρων

Ένας έξυπνος χώρος είναι ένα πολυσύνθετο σύστημα, το οποίο ενσωματώνει πολλά στοιχεία κατανομής επεξεργασίας που δεν έχουν αναπτυχθεί για να εκτελούνται ταυτόχρονα. Προηγουμένως αναφέρθηκαν μερικά από αυτά όπως η *αναγνώριση φωνής*, η *αναγνώριση κίνησης* κ.α. Για να μπορέσουν αυτά τα ξεχωριστά στοιχεία να συνδεθούν, να επικοινωνήσουν και να συνεργαστούν μεταξύ τους με ένα συνεπή και δομημένο τρόπο, χρειάζεται να υπάρχει μια ανάλογη υποδομή λογισμικού (Xie, Shi, Xu, & Mao, 2002).

Κατά καιρούς έχουν γίνει διάφορες προσπάθειες ανάπτυξης υποδομών λογισμικού για έξυπνους χώρους από ερευνητικά ιδρύματα, όπως το MIT με το *Metaglu* (Coen, Phillips, Warshawsky, Weisman, Peters, & Finin, 1999), το Stanford με το *Event Heap* (Fox, Johanson, Hanrahan, & Winograd, 2000) και το iROS (Ponnekanti, Johanson, Kiciman, & Fox, 2003) και το CMU με το *Aura* (Garlan, Siewiorek, Smailagic, & Steenkiste, 2002). Όλες αυτές οι προσπάθειες δείχνουν πόσο μεγάλη σημασία έχει η υποδομή και η αρχιτεκτονική των συστημάτων που υλοποιούν τους έξυπνους χώρους. Ταυτόχρονα προκύπτει ότι

δεν υπάρχει ένα γενικό και ενιαίο πλαίσιο που να καλύπτει όλες τις ανάγκες, αλλά κάθε προσπάθεια επικεντρώνεται στην αντιμετώπιση συγκεκριμένων προκλήσεων, αναλόγως και του περιβάλλοντος (έξυπνος χώρος) που αφορά (Soldatos, Dimakis, Stamatias, & Polymenakos, 2007).

2.4.1 Αρχές σχεδίασης αρχιτεκτονικής έξυπνων χώρων

Τα ιδιαίτερα χαρακτηριστικά των έξυπνων χώρων και η διαφοροποίησή τους από τα υπόλοιπα κατανεμημένα συστήματα, διαμορφώνουν κάποιες βασικές αρχές σχεδίασης της αρχιτεκτονικής τους.

Σύμφωνα με τον Xie και συν. ένα σύστημα έξυπνου χώρου πρέπει να πληροί τις παρακάτω αρχές (Xie, Shi, Xu, & Mao, 2002).

Χαλαρή σύζευξη (Loose coupling). Ένα σύστημα έξυπνου χώρου είναι ένα πολύ δυναμικό σύστημα, αφού οι διάφορες συνιστώσες (modules) του χρειάζεται να επανεκκινούν ή να μετακινούνται από υπολογιστή σε υπολογιστή. Ένα άλλο χαρακτηριστικό που καθιστά αυτά τα συστήματα ιδιαίτερα, είναι η ανάγκη για συνεχή αλλαγή της παραμετροποίησης τους ανάλογα με τις συνθήκες περιβάλλοντος που διαμορφώνονται.

Επεκτασιμότητα (Extensible). Ένα σύστημα έξυπνου χώρου συνεχώς εμπλουτίζεται με καινούργιες λειτουργικότητες κάτι που σημαίνει και καινούργια modules που υλοποιούν αυτές τις λειτουργίες. Για το λόγο αυτό, είναι απαραίτητο να υποστηρίζεται η εισαγωγή νέων στοιχείων ενώ το σύστημα βρίσκεται σε λειτουργία, χωρίς να επηρεάζεται η λειτουργία των υπολοίπων συστατικών του.

Υποστήριξη επικοινωνίας μηνυμάτων και ροής δεδομένων. Η επικοινωνία στα υπό εξέταση συστήματα χωρίζεται σε δύο κατηγορίες, στην επικοινωνία που βασίζεται σε μηνύματα (message-oriented communication) και σε αυτή που βασίζεται σε ροή δεδομένων (stream-oriented communication). Η πρώτη περίπτωση (επικοινωνία με μηνύματα), είναι και αυτή που χρησιμοποιείται συχνότερα και έχει συνήθως μεγάλη σημασιολογική πληροφορία, για παράδειγμα η εντολή σε ένα module να ενεργοποιήσει τη φωτεινή πηγή που ελέγχει. Τέτοιου είδους επικοινωνία είναι πολύ ευαίσθητη στην απώλεια μηνυμάτων, ενώ από την

άλλη οι απαιτήσεις για το χρόνο παράδοσης του μηνύματος είναι μικρές καθώς μια καθυστέρηση μέχρι 50 ms, για το προηγούμενο παράδειγμα, είναι ανεκτή. Η δεύτερη περίπτωση επικοινωνίας (ροή δεδομένων) είναι αυτή που συμβαίνει συνεχώς. Η σημασιολογική της πληροφορία είναι σχετικώς μικρή και η απώλεια μερικών μονάδων δεδομένων είναι συνήθως αποδεκτή. Παρόλα αυτά είναι ευαίσθητη στη διακύμανση του χρόνου παράδοσης του μηνύματος, παρόλο που τις περισσότερες φορές είναι ανεκτική ως προς αυτό τον παράγοντα. Ένα παράδειγμα είναι η αποστολή βίντεο σε πολλαπλούς υπολογιστές ενός δικτύου.

Υποστήριξη επικοινωνίας ένα προς πολλά. Στα συστήματα έξυπνων χώρων ένα μήνυμα πρέπει να αποστέλλεται ταυτόχρονα σε πολλά από τα στοιχεία του. Για παράδειγμα στην επικοινωνία με ροή δεδομένων που αναφέρθηκε προηγουμένως, οι αποδέκτες της μοναδική ροής δεδομένων του βίντεο μπορεί να είναι περισσότεροι από ένας.

Υποστήριξη σε διαφορετικές πλατφόρμες και γλώσσες ανάπτυξης. Τα διάφορα modules ενός συστήματος έξυπνου χώρου συνήθως έχουν διαφορετικές απαιτήσεις σε ότι αφορά το υλικό και το λειτουργικό σύστημα, αλλά έχουν αναπτυχθεί και με διαφορετικές γλώσσες προγραμματισμού. Ένα σύστημα λοιπόν χρειάζεται όλες αυτές τις παραμέτρους να τις λαμβάνει υπόψη του.

Στην προσέγγιση του Zhang και συν. οι προκλήσεις για την ανάπτυξη τέτοιων συστημάτων περιλαμβάνουν (Zhang, Zhu, Cheng, Koh, & Mokhtari, 2006) :

- Ενσωμάτωση και αλληλεπίδραση συσκευών
- Διαχείριση και επεξεργασία context
- Διαλειτουργικότητα των ετερογενών οντοτήτων
- Εργασιο-κεντρικές και Ανθρωπο-κεντρικές διεπαφές χρήστη

Σύμφωνα με τον Singh και συν. η υποδομή λογισμικού των έξυπνων χώρων πρέπει να είναι διάφανη στους προγραμματιστές με ένα τρόπο που να μην τους απασχολεί η μετάπτωση (migration) των προγραμμάτων και των δεδομένων ή ο συγχρονισμός και η συνεργασία των κατανεμημένων στοιχείων του συστήματος. Ακόμα πρέπει να υποστηρίζεται το context awareness με μεθόδους όπως :

- Η συλλογή πληροφοριών από πηγές όπως είναι οι αισθητήρες
- Η διενέργεια ερμηνείας των δεδομένων
- Η διάδοση contextual πληροφορίας στα ενδιαφερόμενα μέρη με έναν κλιμακούμενο και έγκαιρο τρόπο
- Η παροχή μοντέλων για προγραμματισμό context-aware εφαρμογών

Η υποδομή χρειάζεται να ενσωματώνει συνιστώσες (component) λογισμικού, τα οποία είναι πιθανό να ανήκουν σε τελείως διαφορετικά περιβάλλοντα, ακόμα και σε διαφορετικές συσκευές οι οποίες σε συνεργασία μπορούν να εκτελέσουν πολλές εργασίες. Τέλος πρέπει να είναι προσαρμόσιμη, ανεκτική στα λάθη και να υποστηρίζει καταναεμημένη επεξεργασία.

2.5 Υλοποιήσεις αρχιτεκτονικής έξυπνων χώρων

Κατά καιρούς έχουν προταθεί διάφορες αρχιτεκτονικές βάση των οποίων έχουν σχεδιαστεί και υλοποιηθεί συστήματα έξυπνων χώρων. Παρακάτω γίνεται μια συνοπτική παρουσίαση αυτών των προσπαθειών.

2.5.1 Το έξυπνο σπίτι Gator Tech

Το πανεπιστήμιο της Φλόριντα αναπτύσσει *προγραμματιζόμενους διεισδυτικούς χώρους*, στους οποίους ο έξυπνος χώρος υπάρχει τόσο ως περιβάλλον εκτέλεσης, όσο και ως βιβλιοθήκη λογισμικού (Helal, Mann, El-Zabadani, King, Kaddoura, & Jansen, 2005). Ανακάλυψη υπηρεσίας και πρωτοκόλλα δρομολόγησης ενσωματώνουν αυτόματα τα διάφορα στοιχεία του συστήματος, χρησιμοποιώντας ενδιάμεσο λογισμικό (middleware) το οποίο συντηρεί μια περιγραφή υπηρεσίας για κάθε αισθητήρα και κάθε εκκινητήρα (actuator). Οι προγραμματιστές χρησιμοποιώντας υπηρεσίες συνθέτουν εφαρμογές οι οποίες μπορούν να αναπτυχθούν ή να επεκταθούν και από τρίτα μέρη.

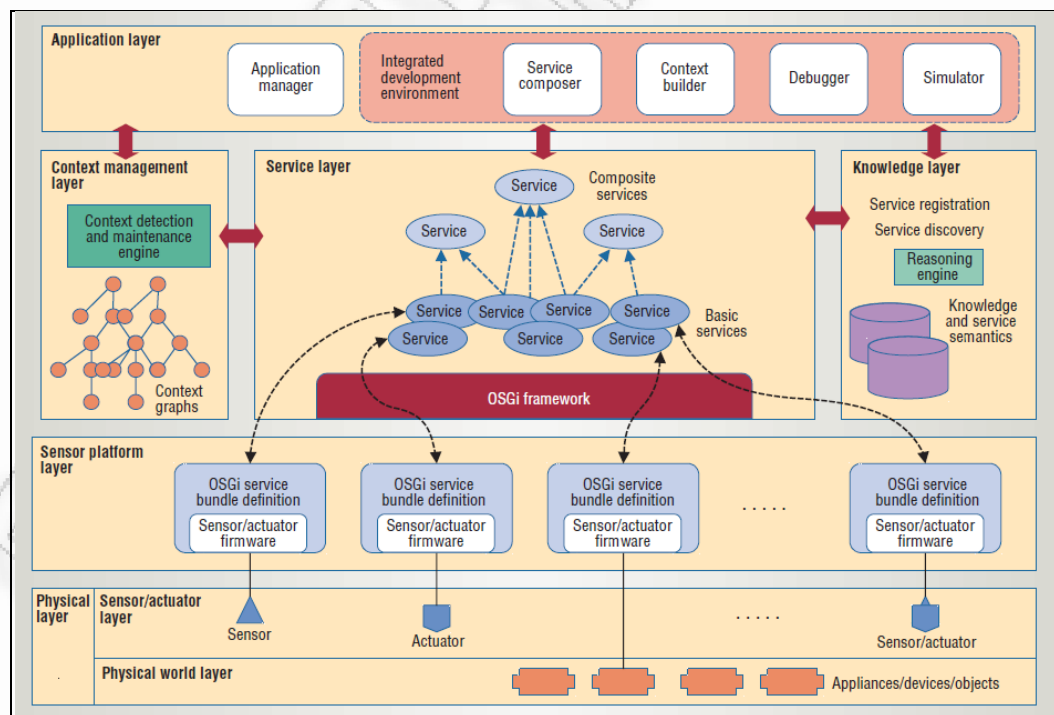
2.5.1.1 Τεχνολογίες έξυπνου σπιτιού

Το πειραματικό σπίτι είναι εξοπλισμένο με διάφορες ευφυείς συσκευές όπως αυτές που περιγράφηκαν στο κεφάλαιο 2.2.4.1. Συνοπτικά αναφέρουμε ότι, υπάρχουν ρολά στα παράθυρα τα οποία προγραμματίζονται και ρυθμίζουν τον

εσωτερικό φωτισμό του σπιτιού, γραμματοκιβώτιο το οποίο αντιλαμβάνεται την άφιξη ενός νέου γράμματος και ειδοποιεί τους ενοίκους, οθόνες τοποθετημένες σε όλο το σπίτι που εξυπηρετούν τόσο την ψυχαγωγία όσο και την ενημέρωση των ενοίκων. Ακόμα υπάρχουν θερμοστάτες που ρυθμίζουν τις συνθήκες θερμοκρασίας του σπιτιού ανάλογα με τις προσωπικές προτιμήσεις των ενοίκων, ειδικό τηλέφωνο το οποίο εκτός των συμβατικών λειτουργιών προσφέρει τη δυνατότητα εξ' αποστάσεως ελέγχου όλων των συσκευών, αισθητήρες που ανιχνεύουν την κίνηση τη θέση και τον προσανατολισμό των ενοίκων καθώς και άλλες τέτοιου είδους ευφυείς συσκευές.

2.5.1.2 Αρχιτεκτονική ενδιάμεσου λογισμικού

Οι δημιουργοί του Gator Tech ανέπτυξαν μια γενική αρχιτεκτονική που μπορεί να έχει εφαρμογές σε κάθε περιβάλλον διεισδυτικής πληροφορικής. Όπως φαίνεται και στην Εικόνα 6 το ενδιάμεσο λογισμικό περιλαμβάνει 7 διαφορετικά στρώματα, *Φυσικό, Πλατφόρμας Αισθητήρα, Υπηρεσίας, Γνωσιακό, Διαχείρισης Context και Εφαρμογής.*



Εικόνα 6. Ενδιάμεσο λογισμικό έξυπνου χώρου (Helal, Mann, El-Zabadani, King, Kaddoura, & Jansen, 2005)

Φυσικό στρώμα. Το στρώμα αυτό αποτελείται από τις διάφορες συσκευές που χρησιμοποιούν οι ένοικοι. Οι περισσότερες από αυτές είναι συσκευές σαν και αυτές που υπάρχουν σε κάθε τυπικό σπίτι όπως, φωτιστικά, τηλεόραση, ραδιόφωνο, κουδούνι και άλλα. Υπάρχουν όμως και συσκευές που χρησιμοποιούν καινοτόμες τεχνολογίες όπως είναι το έξυπνο γραμματοκιβώτιο και το σύστημα εισόδου στο σπίτι χωρίς κλειδί.

Στο ίδιο στρώμα υπάρχουν επίσης οι *αισθητήρες* και οι *εκκινητήρες* όπως είναι οι ανιχνευτές καπνού, οι θερμοστάτες της θέρμανσης και του κλιματισμού καθώς και οι ανιχνευτές κίνησης ως μέρος του συστήματος ασφαλείας. Επιπλέον σε αυτό το στρώμα μπορούν να συμπεριληφθούν αντικείμενα που επιτελούν σημαντικό ρόλο στο χώρο, όπως είναι οι καρέκλες και τα τραπέζια.

Πλατφόρμα αισθητήρα. Κάθε πλατφόρμα αισθητήρα προσδιορίζει κάποια όρια του διεισδυτικού χώρου μέσα στο έξυπνο σπίτι και παρακολουθεί τα αντικείμενα που βρίσκονται μέσα σε αυτά. Για παράδειγμα μπορεί να είναι επιθυμητό να παρακολουθείτε μια τοστιέρα η οποία μπορεί να πάρει φωτιά αν ξεχαστεί ανοικτή, αλλά να μην συμβαίνει το ίδιο με ένα μπλέντερ. Μια πλατφόρμα αισθητήρα επικοινωνεί με μια μεγάλη ποικιλία συσκευών, αισθητήρων, εκκινητήρων και αντιπροσωπεύει όλα αυτά τα αντικείμενα στο υπόλοιπο ενδιάμεσο λογισμικό με έναν ομοιόμορφο τρόπο.

Μια πλατφόρμα αισθητήρα μετατρέπει αποτελεσματικά κάθε αισθητήρα ή εκκινητήρα του φυσικού επιπέδου σε μια υπηρεσία λογισμικού που μπορεί να προγραμματιστεί ή να αποτελέσει μέρος άλλων υπηρεσιών. Έτσι οι προγραμματιστές μπορούν να προσδιορίζουν υπηρεσίες χωρίς να χρειάζεται να κατανοήσουν το φυσικό κόσμο. Αποδεσμεύοντας τους αισθητήρες και τους εκκινητήρες από τις πλατφόρμες των αισθητήρων, διασφαλίζεται η δυναμικότητα του συστήματος και γίνεται πιο εύκολη η εισαγωγή νέας τεχνολογίας μόλις αυτή γίνει διαθέσιμη.

Στρώμα υπηρεσίας. Αυτό το στρώμα περιλαμβάνει το πλαίσιο OSGi, το οποίο και διαχειρίζεται τα μισθωτήρια (leases) των ενεργών υπηρεσιών.

Οι βασικές υπηρεσίες αναπαριστούν το φυσικό κόσμο μέσω των πλατφόρμων αισθητήρων, οι οποίες αποθηκεύουν τους ορισμούς των δεσμών υπηρεσιών (service bundle definitions) για κάθε αισθητήρα και εκκινητήρα που υπάρχει στο πλαίσιο του OSGi. Μόλις ενεργοποιηθεί ένας αισθητήρας ή ένας εκκινητήρας, η αντίστοιχη πλατφόρμα αισθητήρα καταγράφεται στο στρώμα υπηρεσίας αποστέλλοντας τον OSGi ορισμό του service bundle που διαθέτει.

Οι προγραμματιστές δημιουργούν *σύνθετες* υπηρεσίες χρησιμοποιώντας ένα πρωτόκολλο ανακάλυψης υπηρεσίας για να αναζητούν υπάρχοντες υπηρεσίες και μαζί με άλλα services bundle να συνθέτουν νέα OSGi bundles. Οι σύνθετες υπηρεσίες αποτελούν ουσιαστικά τις διαθέσιμες εφαρμογές του διεισδυτικού χώρου.

Για να αυξηθεί η παραγωγικότητα των δημιουργών εφαρμογών, υπάρχει εκ' των προτέρων διαθέσιμο ένα σύνολο από καθορισμένες υπηρεσίες. Τέτοιες υπηρεσίες είναι μεταξύ άλλων η αναγνώριση φωνής, η μετατροπή κειμένου σε ομιλία (text-to-speech recognition), ο χρονοπρογραμματισμός και η μετάδοση πολυμεσικού υλικού.

Γνωσιακό στρώμα. Αυτό το στρώμα περιλαμβάνει μια οντολογία των υπηρεσιών που είναι διαθέσιμες και των συσκευών που είναι συνδεδεμένες στο σύστημα. Έτσι μπορούν να προκύψουν διάφορα συμπεράσματα για τις υπηρεσίες, όπως για παράδειγμα ότι το σύστημα πρέπει να μετατρέψει το αποτέλεσμα της μέτρησης ενός αισθητήρα θερμοκρασίας από βαθμούς Κελσίου σε βαθμούς Φαρενάιτ πριν τροφοδοτήσει μια άλλη υπηρεσία.

Τα πρωτόκολλα διαφήμισης και ανακάλυψης υπηρεσιών χρησιμοποιούν service definition και σημασιολογική λογική για να καταγράψουν ή να ανακαλύψουν μια υπηρεσία. Η *συλλογιστική μηχανή (reasoning engine)* αποφασίζει αν συγκεκριμένες σύνθετες υπηρεσίες είναι διαθέσιμες.

Στρώμα διαχείρισης context. Αυτό το επίπεδο επιτρέπει στους σχεδιαστές των εφαρμογών να δημιουργούν και να καταχωρούν context που τους ενδιαφέρει. Κάθε context είναι ένας γράφος ο οποίος υλοποιείται σαν OSGi υπηρεσία συνδέοντας πολλούς αισθητήρες μαζί. Ένα context μπορεί να προσδιορίζει ή να

απορρίπτει την ενεργοποίηση μιας υπηρεσίας για τις διάφορες εφαρμογές. Μπορεί επίσης να προσδιορίζει τις καταστάσεις στις οποίες ένας διεισδυτικός χώρος δεν μπορεί να βρίσκεται. Η *μηχανή context (context engine)* είναι υπεύθυνη για την ανίχνευση και την πιθανή επαναφορά από τέτοιες καταστάσεις.

Στρώμα εφαρμογής. Το στρώμα αυτό αποτελείται από ένα διαχειριστή εφαρμογής που ενεργοποιεί και απενεργοποιεί τις υπηρεσίες και ένα ολοκληρωμένο γραφικό περιβάλλον με διάφορα εργαλεία για τη δημιουργία έξυπνων χώρων. Πιο συγκεκριμένα, με τον *κατασκευαστή context (context builder)* ένας προγραμματιστής μπορεί να κατασκευάσει με γραφικό τρόπο έναν γράφο ο οποίος συσχετίζει συμπεριφορά με context. Μπορεί επίσης να το χρησιμοποιήσει για να προσδιορίσει μη επιτρεπτά contexts και υπηρεσίες επαναφοράς. Σε αυτό το στρώμα υπάρχει ακόμα ο *συνθέτης υπηρεσιών (service composer)* που χρησιμοποιείται για την αναζήτηση και την ανακάλυψη υπηρεσιών καθώς και για τη σύνθεση και την καταγραφή νέων. Δύο ακόμα διαθέσιμα εργαλεία, είναι ο debugger και ο εξομοιωτής (simulator).

2.5.2 Το έξυπνο δωμάτιο του ΑΙΤ

Το ΑΙΤ (Athens Information Technology) ανέπτυξε ένα έξυπνο δωμάτιο για να προωθήσει την κατανόηση, την εκπαίδευση και την έρευνα στην περιοχή των Αυτόνομων και Ευφών Συστημάτων (Autonomic and Intelligent Systems) (ΑΙΤ, 2007).

Η υποδομή του δωματίου περιλαμβάνει κάμερες, μικρόφωνα, σταθμούς εργασίας καθώς και μια υποδομή ενδιάμεσου λογισμικού που επιτρέπει τη συνεχή παρακολούθηση του δωματίου και την επεξεργασία των δεδομένων των αισθητήρων.

2.5.2.1 Αρχιτεκτονική ενδιάμεσου λογισμικού

Η βασική αρχιτεκτονική του συστήματος περιλαμβάνει 3 στρώματα (Dimakis, Mylonakis, Soldatos, & Polymenakos, 2006), *Αισθητήρα και Εκκινήτηρα, Ανάκτησης Context (Context-Acquisition)* και *Υπηρεσίας*.

Στρώμα αισθητήρα και εκκινητήρα. Η υποδομή αισθητήρων είναι η βάση του συστήματος διεισδυτικής πληροφορικής. Περιλαμβάνει αισθητήρες ήχου και εικόνες οι οποίοι παρακολουθούν το εσωτερικό περιβάλλον εκτιμώντας τις ανθρώπινες δραστηριότητες. Εκτός από μεμονωμένες συσκευές, υπάρχουν και πιο σύνθετες *συστάδες* αισθητήρων οι οποίες με την κατάλληλη επεξεργασία μπορούν να προσφέρουν πιο λεπτομερή εξαγωγή contextual πληροφορίας. Οι αισθητήρες χρησιμοποιούνται για τη συλλογή πληροφορίας από το περιβάλλον. Η πληροφορία αυτή προωθείται σε ένα ανώτερο εννοιολογικό στρώμα το οποίο έχει τη δυνατότητα να εξάγει context εφαρμόζοντας εξελιγμένες τεχνικές επεξεργασίας ψηφιακού σήματος.

Για να γίνει εφικτή η αλληλεπίδραση ανθρώπου – μηχανής, οι εκκινητήρες του έξυπνου χώρου χρειάζεται να συνεργάζονται απρόσκοπτα. Τέτοιες συσκευές είναι τα monitors, τα ηχεία, οι προβολείς κ.α., τα οποία ελέγχονται από εξειδικευμένο λογισμικό που αποτελεί μέρος της γενικότερης διασυνδεδεμένης αρχιτεκτονικής λογισμικού. Όπως και στην περίπτωση των αισθητήρων, έχει χρησιμοποιηθεί εξελιγμένος εξοπλισμός ο οποίος περιλαμβάνει συσκευές παραγωγής ιδιαίτερα στοχευμένων δεσμών ήχου σε συγκεκριμένη τοποθεσία καθώς και προβολείς με δυνατότητα επιλογής της γωνίας προβολής.

Στρώμα ανάκτησης context (context-acquisition). Το δεύτερο στρώμα της αρχιτεκτονικής περιλαμβάνει τις ευφυείς διεπαφές. Σε αυτό το στρώμα τα δεδομένα των αισθητήρων επεξεργάζονται για να εξαχθεί το βασικό context. Για αυτή τη διεργασία έχουν αναπτυχθεί διάφορα αντιληπτικά (perceptual) στοιχεία λογισμικού όπως 2-διάστατα και 3-διάστατα οπτικοακουστικά στοιχεία, πολυτροπικά στοιχεία αντίληψης, στοιχεία σύνθεσης ομιλίας και στοχευμένου ήχου. Τα παραπάνω εφαρμόζουν εξελιγμένους αλγορίθμους επεξεργασίας σήματος στα δεδομένα που παράγονται από τους αισθητήρες και εξάγουν απλό context όπως :

- 2-Διάστατη και 3-διάστατη θέση προσώπου
- Αναγνώριση προσώπου
- 2-Διάστατη και 3-διάστατη θέση ομιλητή
- Ανίχνευση λεκτικής δραστηριότητας

- Αναγνώριση ομιλητή

Στρώμα υπηρεσίας. Το ανώτερο στρώμα της αρχιτεκτονικής αξιοποιεί τα πλεονεκτήματα μιας δομής πολλαπλών πρακτόρων (multi-agent). Αναπτύχθηκε με τη χρήση του πλαισίου JADE (Java Agent Development) το οποίο χρησιμοποιείται για τη δημιουργία εφαρμογών συμμορφωμένων με τις προδιαγραφές FIPA (Foundation of Intelligent Physical Agents) για διαλειτουργικά ευφυή συστήματα. Σκοπός ήταν η απλοποίηση της ανάπτυξης, αλλά και η διασφάλιση της συμμόρφωσης διαμέσου ενός εκτενούς σετ υπηρεσιών συστήματος και πρακτόρων (agents).

Οι υπηρεσίες υλοποιούνται σαν υπηρεσίες πρακτόρων (agent-services) της JADE κοινότητας που αποτελείται από agents καταναμημένους σε διάφορους υπολογιστές. Αυτοί οι agents διαθέτουν εξειδικευμένες λειτουργίες όπως, εξατομίκευση συμμετέχοντος, συντονισμός γεγονότων, ανάκτηση ελέγχου, αντιληπτικά στοιχεία, έλεγχος αισθητήρα κ.α.. Οι agents επικοινωνούν μεταξύ τους χρησιμοποιώντας την υπάρχουσα JADE αρχιτεκτονική επικοινωνίας, δημιουργώντας έτσι μια πλήρως καταναμημένη υπηρεσία.

Στην παρούσα υλοποίηση υπάρχουν 3 ειδών agents :

- *Οι Κεντρικοί Agents*, που είναι υπεύθυνοι για την παροχή επικοινωνίας και την υπηρεσία καταλόγου agents.
- *Οι Agents Βασικής Υπηρεσίας*, οι οποίοι εστιάζουν στη μοντελοποίηση της κατάστασης των πανταχού παρών (ubiquitous) υπηρεσιών και της διεπαφής επικοινωνίας της γνωσιακής βάσης και της βάσης δεδομένων.
- *Οι Agents της Πανταχού Παρών Υπηρεσίας*, οι οποίοι εφαρμόζουν την υπηρεσιοστρεφή λογική της πανταχού παρών υπηρεσίας.

Το ανώτερο στρώμα της αρχιτεκτονικής αξιοποιεί τα 2 κατώτερα στρώματα για να προσφέρει την άνθρωπο-κεντρική και μη ενοχλητική πανταχού παρών υπηρεσία. Αυτό επιτυγχάνεται με τη χρήση ειδικού λογισμικού το οποίο μετατρέπει το βασικό context που ανακτήθηκε από το στρώμα ανάκτησης context σε πιο εκλεπτυσμένο context, ικανό να παρακολουθεί τις ενέργειες ανθρώπων που συμμετέχουν σε μια συνάντηση ή μια διάλεξη. Το παραγόμενο context

προωθείται σε όλα τα μέρη του επιπέδου με τη χρήση του multi-agent πλαισίου επικοινωνίας. Επιπλέον το πλαίσιο αυτό παρέχει ολοκληρωμένα εργαλεία διαχείρισης τα οποία διευκολύνουν την επικοινωνία, την αναζήτηση, την εγγραφή και την διαγραφή των agents του επιπέδου υπηρεσίας.

2.5.3 Πλατφόρμα Έξυπνων Χώρων

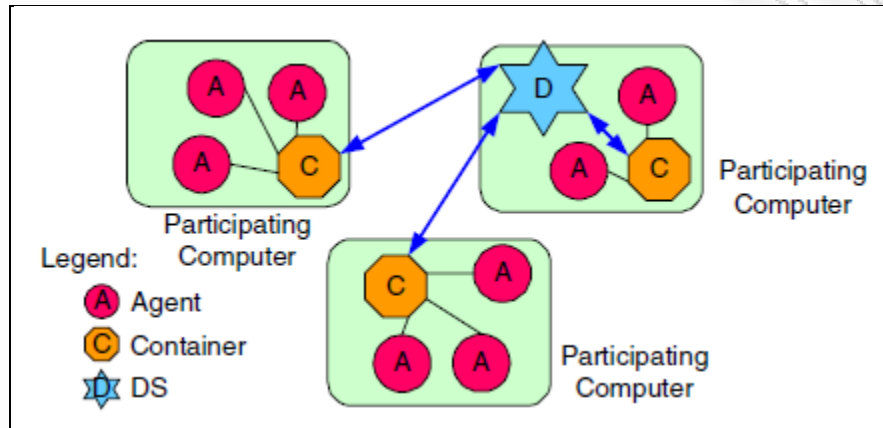
Το πανεπιστήμιο Tsinghua του Πεκίνου ανέπτυξε μια πλατφόρμα έξυπνων χώρων (Xie, Shi, Xu, & Mao, 2002), δίνοντας ιδιαίτερη έμφαση στην ανάγκη για ανταλλαγή μηνυμάτων σε πραγματικό χρόνο μεταξύ των διαφόρων συστατικών του συστήματος και στη φιλικότητα ως προς το χρήστη. Στην πλατφόρμα αυτή υλοποιήθηκε και δοκιμάστηκε η Έξυπνη Τάξη (παρουσιάστηκε στο κεφάλαιο 2.3.3), ένα έργο το οποίο στοχεύει στην κατασκευή ενός έξυπνου χώρου τηλεκπαίδευσης.

2.5.3.1 Αρχιτεκτονική ενδιάμεσου λογισμικού

Δύο πιθανά αφαιρετικά μοντέλα κατανεμημένων συστημάτων είναι το Μοντέλο Κατανεμημένων Στοιχείων (Distributed Component Model - DCM) και το Multi-Agent Σύστημα (Multi-Agent System - MAS). Στην περίπτωση του DCM απαραίτητη προϋπόθεση είναι η ύπαρξη μιας κεντρικής διεργασίας η οποία υλοποιεί όλη τη λογική της εφαρμογής και διαχειρίζεται τα κατανεμημένα αντικείμενα του συστήματος. Σε περιβάλλον έξυπνων χώρων όμως, είναι δύσκολο να υπάρξει μια τέτοια μοναδική διεργασία και αντί αυτής υπάρχουν πάντα πολλές παράλληλες που εκτελούνται την ίδια χρονική στιγμή. Για παράδειγμα ένα υποσύστημα παρακολούθησης χρήστη συνεχώς παρακολουθεί τη θέση του στο χώρο, ενώ ταυτόχρονα ένα άλλο κάνει αναγνώριση της φωνής του. Αντιθέτως στην περίπτωση του μοντέλου MAS, κάθε υποσύστημα έχει τη δική του διαδικασία εκτέλεσης, κάτι το οποίο ταιριάζει καλύτερα στη φύση των έξυπνων χώρων. Για το λόγο αυτό η ομάδα ανάπτυξης υιοθέτησε το δεύτερο μοντέλο για την Πλατφόρμα Έξυπνων Χώρων.

Στην Πλατφόρμα Έξυπνων Χώρων, ο agent είναι η βασική συστατική μονάδα η οποία μπορεί να αφορά οποιοδήποτε στοιχείο λογισμικού, όπως έναν οδηγό

συσκευής, έναν αντιληπτικό αλγόριθμο ή μια υποεφαρμογή, εφόσον όμως μπορεί να εκτελεστεί αυτόνομα και σύμφωνα με τις προδιαγραφές που έχουν προσδιοριστεί για ένα agent.



Εικόνα 7. Δομή περιβάλλοντος εκτέλεσης Έξυπνης Πλατφόρμας (Xie, Shi, Xu, & Mao, 2002)

Δομή περιβάλλοντος εκτέλεσης. Όπως φαίνεται και από την Εικόνα 7, εκτός των agents, υπάρχουν ακόμα δύο δομικά στοιχεία του συστήματος τα, DS και Container.

DS. Σε όλο το περιβάλλον εκτέλεσης της Έξυπνης Πλατφόρμας, πρέπει να εκτελείται ένα στιγμιότυπο του στοιχείου DS κάθε χρονική στιγμή. Πρόκειται για το στοιχείο εκείνο του συστήματος που βρίσκεται στο κέντρο της Έξυπνης Πλατφόρμας και παρέχει πολλές λειτουργίες κλειδιά, όπως η εγγραφή των agents, η υπηρεσία καταλόγου και η αποστολή μηνυμάτων.

Container. Κάθε υπολογιστής που συμμετέχει στην Πλατφόρμα Έξυπνων Χώρων πρέπει να εκτελεί ένα στιγμιότυπο του Container σαν διεργασία δαίμονα. Ένας agent όταν θέλει να αλληλεπιδράσει με άλλα μέρη του συστήματος, επικοινωνεί με τον Container που βρίσκεται στον ίδιο υπολογιστή και αυτός αναλαμβάνει να επικοινωνήσει με τον DS ή με άλλους Containers εκ μέρους του agent. Αυτή η αρχιτεκτονική έχει δύο βασικά πλεονεκτήματα. Πρώτον, περιορίζει τη διανομή προηγούμενης πληροφορίας σχετικής με τη δομή του συστήματος σε ένα μόνο σημείο (τον Container) ανά υπολογιστή, πράγμα το οποίο ικανοποιεί την απαίτηση της χαλαρής σύζευξης (loose coupling). Δεύτερον, παρέχει ένα κεντρικό σημείο διαχείρισης για όλους τους agents του υπολογιστή. Για

παράδειγμα ένας Container μπορεί να παρέχει τη διεπαφή χρήστη για την εκκίνηση ή τον τερματισμό των agents του υπολογιστή στον οποίο βρίσκεται.

Στρώμα επικοινωνίας. Σε ένα σύστημα MAS η επικοινωνία μεταξύ των agents συνήθως πραγματοποιείται με δύο διαφορετικά σχήματα. Το ένα ονομάζεται άμεσο σχήμα και κάθε ζευγάρι agents που χρειάζεται να επικοινωνήσει πρέπει να δημιουργήσει μια απευθείας σύνδεση. Το άλλο ονομάζεται ενδιάμεσο σχήμα και κάθε agent που θέλει να επικοινωνήσει δημιουργεί μια σύνδεση με ένα κεντρικό συστατικό του συστήματος που ονομάζεται *μεσάζων (broker)*, το οποίο αναλαμβάνει να διευθετεί όλες τις επικοινωνίες.

Το δεύτερο σχήμα έχει πολλά πλεονεκτήματα σε σχέση με το πρώτο. Πρώτον, αποσυνδέει τις δύο άκρες της γραμμής επικοινωνίας, δηλαδή την πηγή και τον προορισμό. Δεύτερον, είναι πιο εύκολο με αυτό τον τρόπο να επιτευχθεί η ένα-προς-πολλά επικοινωνία. Τρίτον, ο αριθμός των συνδέσεων που πρέπει να συντηρούνται είναι γραμμικός ως προς το πλήθος των agents, δηλαδή πολύ μικρότερος σε σχέση με το πρώτο σχήμα. Παρόλα αυτά υπάρχει ένα σοβαρό μειονέκτημα σε αυτόν τον τρόπο επικοινωνίας. Καθώς όλες οι επικοινωνίες πρέπει να διαχειρίζονται από ένα κεντρικό συστατικό του συστήματος (τον μεσάζοντα), είναι δύσκολο να εγγυηθεί η καθυστέρηση παράδοσης των μηνυμάτων κάθε μοναδικής επικοινωνίας.

Για αυτό το λόγο, η Πλατφόρμα Έξυπνων Χώρων υιοθετεί ένα υβριδικό σχήμα, έτσι ώστε να προσφέρει τη βέλτιστη υπηρεσία επικοινωνίας σε κάθε περίπτωση. Όλοι οι agents διατηρούν μια επικοινωνία με τον DS (μέσω του Container σύμφωνα με όσα αναφέρθηκαν προηγουμένως) και την ίδια στιγμή μπορούν να δέχονται απευθείας επικοινωνίες διαμέσου UDP πορτών. Για επικοινωνία που βασίζεται σε μηνύματα (message-oriented), τα μηνύματα στέλνονται στον DS και από εκεί προωθούνται στον κατάλληλο παραλήπτη. Για επικοινωνία που βασίζεται σε ροή δεδομένων (stream-oriented), τα μηνύματα στέλνονται στον agent του προορισμού μέσω UDP και εφόσον δεν μεσολαβεί ενδιάμεσο σημείο επικοινωνίας η καθυστέρηση παράδοσης ελαχιστοποιείται.

Στην Πλατφόρμα Έξυπνων Χώρων όποια μορφή επικοινωνίας και αν χρησιμοποιείται, το μήνυμα είναι η βασική μονάδα δεδομένων. Τα μηνύματα στην Πλατφόρμα Έξυπνων Χώρων υλοποιούνται σε XML και αυτό προσφέρει τρία σημαντικά πλεονεκτήματα :

- Υπάρχει μεγάλη επεκτασιμότητα καθώς οι προγραμματιστές μπορούν να δημιουργήσουν μια αρχική δομή ενός μηνύματος και κατόπιν να προσθέσουν επιπλέον πεδία χωρίς να επηρεαστεί η δουλειά που έχει γίνει.
- Η XML είναι φιλική ως προς το χρήστη, καθώς είναι αναγνώσιμη από τον άνθρωπο. Επιπλέον είναι αυτό-περιγραφική.
- Η XML αποτελεί στάνταρτ για το διαδίκτυο και έτσι θα είναι πιο εύκολη η διασύνδεση με άλλα ετερογενή συστήματα έξυπνων χώρων στο μέλλον.

Στρώμα συνεργασίας. Το στρώμα συνεργασίας βρίσκεται σε υψηλότερο και πιο γενικό επίπεδο σε σχέση με το στρώμα επικοινωνίας σε μια πλατφόρμα έξυπνων χώρων. Το δεύτερο ασχολείται περισσότερο με το πώς θα επιτευχθεί επικοινωνία μεταξύ δύο οποιονδήποτε agents, ενώ το πρώτο με το πώς θα συγχρονιστεί το όλο σύστημα.

Υπάρχουν διάφορα πιθανά μοντέλα συνεργασίας σε ένα σύστημα MAS, τα οποία μπορούν να κατηγοριοποιηθούν σε 4 γενικές κατηγορίες : *Direct*, *Blackboard*, *Tuple-Space* και *Publish-and-Subscribe*. Η Πλατφόρμα Έξυπνων Χώρων υιοθετεί το μοντέλο Publish-and-Subscribe για τους ακόλουθους λόγους :

1. Στο μοντέλο αυτό, οι agents δεν χρειάζεται να διατηρούν προηγούμενη γνώση για το άλλο μέρος, όπως συμβαίνει στα υπόλοιπα μοντέλα, κάτι το οποίο διευκολύνει την επίτευξη της χαλαρή σύζευξη του συστήματος.
2. Στο μοντέλο αυτό, η υπηρεσία επικοινωνίας είναι εκ φύσεως χωρίς φραγή (non-blocking), ενώ τα άλλα μοντέλα τείνουν σε λογική φραγής (blocking). Η non-blocking επικοινωνία είναι πιο κατάλληλη για τη χαλαρή σύζευξη, αφού ένας agent δεν σταματά τη λειτουργία του εξαιτίας αποτυχιών των υπολοίπων.
3. Στο μοντέλο αυτό, ο agent ενημερώνεται για την άφιξη μηνυμάτων από κάποιου είδους μηχανισμό ανάκλησης. Ο μηχανισμός αυτός επιτρέπει στο

προγραμματιστή να υλοποιεί έναν agent πιο εύκολα και γρήγορα σε σχέση με τους μηχανισμούς των άλλων μοντέλων.

Η ομάδα ανάπτυξης της Πλατφόρμα Έξυπνων Χώρων εισήγαγε μια νέα έννοια σε ότι αφορά το μοντέλο Publish-and-Subscribe, το Message Group. Οι agents αποστέλλουν και λαμβάνουν μηνύματα με βάση τα Message Groups, τα οποία δημιουργούνται δυναμικά από τους agents κατά τη λειτουργία του συστήματος. Με τη χρήση αυτού του επιπέδου αφαίρεσης, δίνεται η ευελιξία στους προγραμματιστές να ορίσουν οι ίδιοι τη διαστρωμάτωση του κοινού παραλαβής των μηνυμάτων.

Εκτός του ονόματος του, ένα Message Group έχει και μια ακόμα σημαντική ιδιότητα, η οποία προσδιορίζει τον τρόπο επικοινωνίας που θα χρησιμοποιήσει η Πλατφόρμα Έξυπνων Χώρων για την παράδοση μηνυμάτων και η οποία παίρνει τις τιμές *ροή δεδομένων* ή *μηνύματα δεδομένων*. Στην περίπτωση των μηνυμάτων δεδομένων, χρησιμοποιείται το ενδιάμεσο σχήμα. Πιο συγκεκριμένα, ο DS διαχειρίζεται κεντρικά όλες τις εγγραφές (subscriptions) και τα μηνύματα μεταφέρονται διαμέσου της εγκατεστημένη σύνδεση μεταξύ των agents και του DS. Στην περίπτωση των μηνυμάτων ροής, χρησιμοποιείται το άμεσο σχήμα επικοινωνίας. Πρώτα ο DS αναλαμβάνει να αναθέσει μια διεύθυνση πολυεκπομπής (multicast) σε κάθε πιθανή ομάδα (group) ροής δεδομένων του συστήματος. Έπειτα όταν ένας agent εγγράφεται σε ένα τέτοιο group, πρέπει πρώτα να αποκτήσει την multicast διεύθυνση του group από τον DS και μετά να αρχίσει να τη χρησιμοποιεί. Απ' την άλλη, αν ένας agent θέλει να αποστείλει ένα μήνυμα σε ένα τέτοιο group, πρέπει πρώτα να αναζητήσει στον DS την ανατεθείσα multicast διεύθυνση και κατόπιν να στείλει το μήνυμά του σε αυτή. Για λόγους απόδοσης, οι agents διατηρούν τη διεύθυνση αυτή τοπικά.

3 ΤΕΧΝΟΛΟΓΙΕΣ ΥΛΟΠΟΙΗΣΗΣ ΕΞΥΠΝΩΝ ΧΩΡΩΝ

Η υλοποίηση των διαφορών αρχιτεκτονικών για συστήματα έξυπνων χώρων, προϋποθέτει τη χρήση τεχνολογιών λογισμικού όπως αυτές που περιγράφονται παρακάτω.

3.1 OSGi

Η Συμμαχία OSGi ιδρύθηκε το Μάρτιο του 1999 με σκοπό να δημιουργήσει ανοικτές προδιαγραφές για τη δικτυακή παράδοση ελεγχόμενων υπηρεσιών σε τοπικά δίκτυα και συσκευές. Ο οργανισμός OSGi είναι ο πρωτοπόρος φορέας τυποποίησης της επόμενης γενιάς υπηρεσιών διαδικτύου για κατοικίες, αυτοκίνητα, κινητά τηλέφωνα, υπολογιστές, μικρά γραφεία και άλλα περιβάλλοντα (OSGi, 2007).

Η προδιαγραφή της Πλατφόρμα Υπηρεσιών OSGi προσφέρει μια ανοικτή, κοινή αρχιτεκτονική για παρόχους υπηρεσιών, προγραμματιστές, πωλητές λογισμικού, gateway operators και πωλητές εξοπλισμού έτσι ώστε να αναπτύξουν και να διαχειριστούν υπηρεσίες με έναν συντεταγμένο τρόπο. Ενεργοποιεί μια εντελώς νέα κατηγορία έξυπνων συσκευών λόγω της ευέλικτης και ελεγχόμενης ανάπτυξης των υπηρεσιών της. Οι προδιαγραφές της OSGi αφορούν αποκωδικοποιητές, πύλες υπηρεσιών (service gateways), καλωδιακούς διαποδιαμορφωτές (modems), ηλεκτρονικές συσκευές, υπολογιστές, βιομηχανικούς υπολογιστές, αυτοκίνητα, κινητά τηλέφωνα και άλλα. Οι συσκευές οι οποίες υλοποιούν τις προδιαγραφές OSGi δίνουν τη δυνατότητα σε παρόχους υπηρεσιών όπως εταιρείες τηλεπικοινωνιών, παρόχους καλωδιακής τηλεόρασης, εφαρμογές και άλλους να προσφέρουν διαφοροποιημένες και χρήσιμες υπηρεσίες μέσα στα δίκτυα τους.

Επισκόπηση του Πλαισίου OSGi. Το Πλαίσιο αποτελεί τον πυρήνα των Προδιαγραφών της Πλατφόρμας Υπηρεσιών OSGi. Προσφέρει ένα γενικό, ασφαλές και ελεγχόμενο πλαίσιο Java το οποίο υποστηρίζει την ανάπτυξη επεκτάσιμων και καταφορτώσιμων (downloadable) εφαρμογών γνωστών ως *δέσμες (bundles)*.

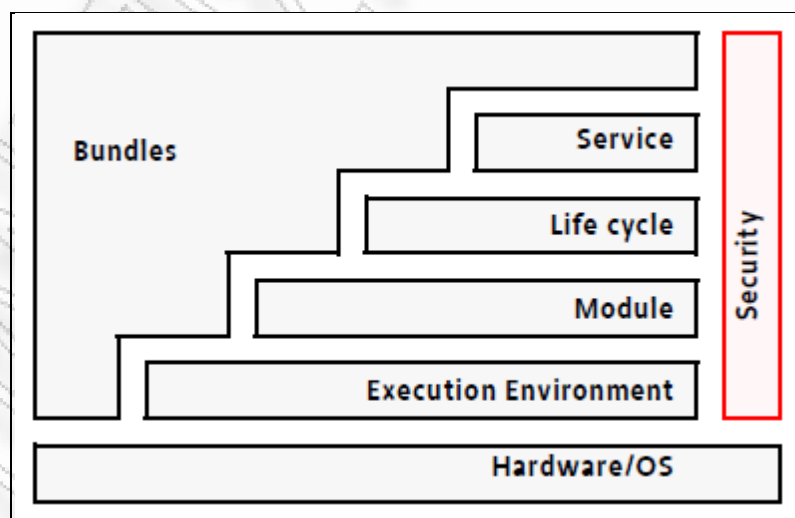
Οι συσκευές που είναι συμβατές με την OSGi μπορούν να καταφορτώσουν (download) και να εγκαταστήσουν OSGi bundles και κατόπιν να τα αφαιρέσουν όταν πλέον δεν τα χρειάζονται. Το πλαίσιο διαχειρίζεται την εγκατάσταση και την ενημέρωση των bundles σε περιβάλλον OSGi με δυναμικό και κλιμακωτό τρόπο. Για να το πετύχει αυτό, διαχειρίζεται λεπτομερώς τις εξαρτήσεις ανάμεσα στα bundles και τις υπηρεσίες.

Επίσης προσφέρει στον δημιουργό των bundles τους κατάλληλους πόρους για να εκμεταλλευτεί την αυτονομία της πλατφόρμας Java και τη δυνατότητα για δυναμικό φόρτωμα του κώδικα, με σκοπό την εύκολη ανάπτυξη υπηρεσιών για συσκευές με μικρή μνήμη οι οποίες μπορούν να αναπτυχθούν σε μεγάλη κλίμακα.

Η λειτουργικότητα του Πλαισίου χωρίζεται στα ακόλουθα στρώματα :

- Στρώμα Ασφαλείας
- Στρώμα Δομοστοιχείου (Module)
- Στρώμα Κύκλου Ζωής
- Στρώμα Υπηρεσίας
- Πραγματικές Υπηρεσίες

Η παραπάνω διαστρωμάτωση παρουσιάζεται στην Εικόνα 8.



Εικόνα 8. Διαστρωμάτωση του Πλαισίου OSGi (OSGi, 2007)

Το **Στρώμα Ασφαλείας** βασίζεται στην ασφάλεια της Java 2 αλλά προσθέτει ακόμα ένα πλήθος περιορισμών και καλύπτει μερικά από τα κενά τα οποία η Java αφήνει ανοικτά. Ορίζει ένα ασφαλές μορφότυπο πακετοποίησης καθώς και αλληλεπίδραση με το στρώμα ασφαλείας της Java 2 κατά τη διάρκεια της εκτέλεσης.

Το **Στρώμα Δομοστοιχείου** ορίζει ένα μοντέλο δομοστοιχειωτοποίησης (modularization model) της Java. Το στρώμα αυτό έχει αυστηρούς κανόνες για το διαμοιρασμό πακέτων Java ανάμεσα στα bundles ή για την απόκρυψη πακέτων από άλλα bundles. Το Στρώμα Δομοστοιχείου μπορεί να χρησιμοποιηθεί χωρίς το Στρώμα Κύκλου Ζωής και το Στρώμα Υπηρεσίας. Το Στρώμα Κύκλου Ζωής παρέχει ένα API για τη διαχείριση των bundles στο Στρώμα Δομοστοιχείου, ενώ το Στρώμα Υπηρεσίας παρέχει ένα μοντέλο επικοινωνίας για την επικοινωνία των bundles.

Το **Στρώμα Κύκλου Ζωής** παρέχει στα bundles ένα API κύκλου ζωής. Αυτό το API παρέχει ένα μοντέλο χρόνου εκτέλεσης (runtime model) για τα bundles. Ορίζει πώς ξεκινούν και σταματάνε τα bundles καθώς και πώς εγκαθίστανται, ενημερώνονται και απεγκαθίστανται. Επιπλέον, παρέχει ένα εκτενή API συμβάντων για να επιτρέπει σε ένα bundle διαχείρισης να ελέγχει τις λειτουργίες της πλατφόρμας υπηρεσίας. Το Στρώμα Κύκλου Ζωής χρειάζεται το Στρώμα Δομοστοιχείου αλλά το Στρώμα Ασφαλείας είναι προαιρετικό.

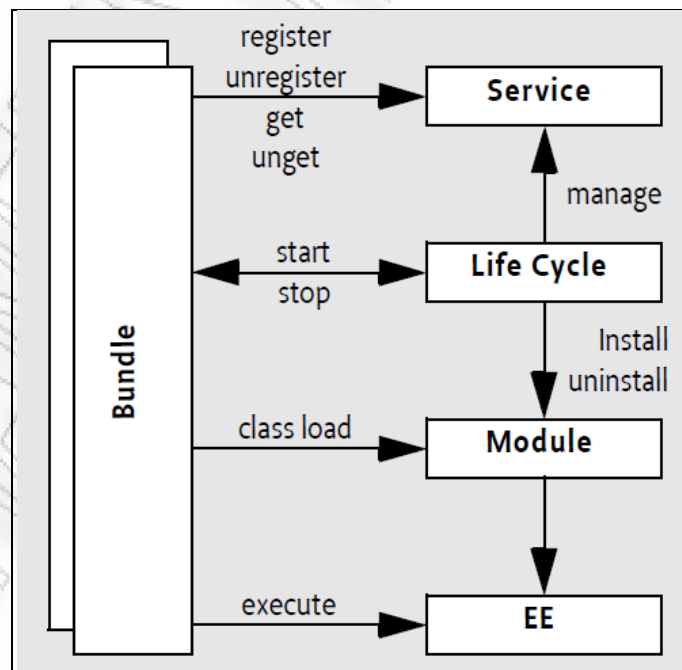
Το **Στρώμα Υπηρεσίας** παρέχει ένα δυναμικό, συνοπτικό και συνεπή μοντέλο προγραμματισμού για προγραμματιστές Java bundles, απλοποιώντας την ανάπτυξη των bundles υπηρεσίας με το να αποσυνδέει τις προδιαγραφές της υπηρεσίας από τις υλοποιήσεις της. Αυτό το μοντέλο επιτρέπει στους δημιουργούς των bundles να συνδέονται με τις υπηρεσίες χρησιμοποιώντας μόνο τις προδιαγραφές διεπαφής. Η επιλογή μιας συγκεκριμένης υλοποίησης, βελτιστοποιημένης για μια συγκεκριμένη ανάγκη ή από ένα συγκεκριμένο προμηθευτή, μπορεί με αυτόν τον τρόπο να μετατεθεί για το χρόνο της εκτέλεσης.

Ένα συνεπές μοντέλο προγραμματισμού βοηθά τους δημιουργούς bundles να ανταπεξέρθουν σε θέματα κλιμακοθετησιμότητας (scalability) σε πολλές

διαφορετικές διαστάσεις. Πρόκειται για μια σημαντική παράμετρο επειδή το Πλαίσιο αποσκοπεί στο να εκτελείται σε διάφορες συσκευές των οποίων τα διαφορετικά χαρακτηριστικά του υλικού μπορεί να επηρεάζουν πολλές πλευρές της υλοποίησης μιας υπηρεσίας. Συνεπείς διεπαφές διασφαλίζουν ότι οι συνιστώσες του λογισμικού μπορούν να αναμειχθούν, να προσαρμοστούν και να εξακολουθήσουν να είναι αποτελεσματικές σε σταθερά συστήματα.

Το Πλαίσιο δίνει τη δυνατότητα στα bundles να επιλέγουν μια διαθέσιμη υλοποίηση κατά την εκτέλεση, μέσω της υπηρεσίας μητρώου. Τα bundles εγγράφουν τις νέες υπηρεσίες, δέχονται ειδοποιήσεις σχετικά με την κατάσταση τους, ή αναζητούν υπάρχοντες υπηρεσίες για να τις εντάξουν στις τρέχουσες δυνατότητες της συσκευής. Αυτή η δυνατότητα του Πλαισίου καθιστά ένα εγκατεστημένο bundle επεκτάσιμο, αφού νέα bundles μπορούν να εγκατασταθούν για να προσφέρουν νέα χαρακτηριστικά ή υπάρχοντα bundles μπορούν να τροποποιηθούν και να ενημερωθούν χωρίς να χρειάζεται η επανεκκίνηση του συστήματος.

Η αλληλεπίδραση μεταξύ των διαφόρων στρωμάτων παρουσιάζεται στην Εικόνα 9.



Εικόνα 9. Αλληλεπίδραση μεταξύ των στρωμάτων του Πλαισίου της OSGi (OSGi, 2007)

3.2 UPnP

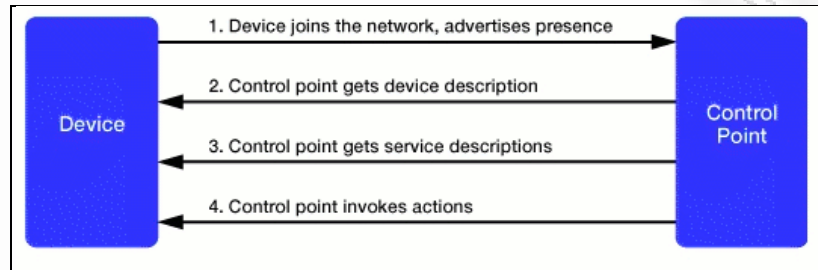
Το UPnP είναι ένα σύνολο πρωτοκόλλων ανακάλυψης υπηρεσίας το οποίο βρίσκεται υπό ανάπτυξη από το Universal Plug and Play Forum, μια κοινοπραξία εταιρειών της οποίας ηγείται η Microsoft. Το UPnP ασχολείται με την τυποποίηση των πρωτοκόλλων που χρησιμοποιούνται μεταξύ των πελατών (clients) και των υπηρεσιών. Οι περιγραφές των συσκευών και των υπηρεσιών είναι κωδικοποιημένες σε XML και ένας αριθμός πρωτοκόλλων για τοπική αυτό-διαμόρφωση, ανακάλυψη, διαφήμιση, αλληλεπίδραση πελάτη/υπηρεσίας και ενημέρωση συμβάντων περιλαμβάνονται στις προδιαγραφές. Τα πρωτόκολλα αυτά τείνουν να βασίζονται σε υπάρχοντα πρότυπα όπως το HTTP (Allard, Chinta, Gundala, & Richard, 2003).

Η αρχιτεκτονική ορίζει έξι φάσεις αλληλεπίδρασης (Miller, Nixon, Tai, & Wood, 2001) :

Διευθυνσιοδότηση. Το θεμέλιο για την UPnP δικτύωση είναι η IP διευθυνσιοδότηση. Κάθε συσκευή πρέπει να διαθέτει ένα DHCP (Dynamic Host Configuration Protocol) πελάτη και να αναζητά έναν DHCP εξυπηρετητή όταν η συσκευή συνδέεται πρώτη φορά στο δίκτυο. Αν δεν υπάρχει διαθέσιμος DHCP εξυπηρετητής, η συσκευή πρέπει να αναθέτει στον εαυτό της μια διεύθυνση. Αν κατά τη διάρκεια της DHCP συναλλαγής, η συσκευή αποκτήσει ένα όνομα τομέα τότε πρέπει να το χρησιμοποιεί και στις ακόλουθες δικτυακές λειτουργίες, αλλιώς η συσκευή πρέπει να χρησιμοποιήσει την IP διεύθυνση της.

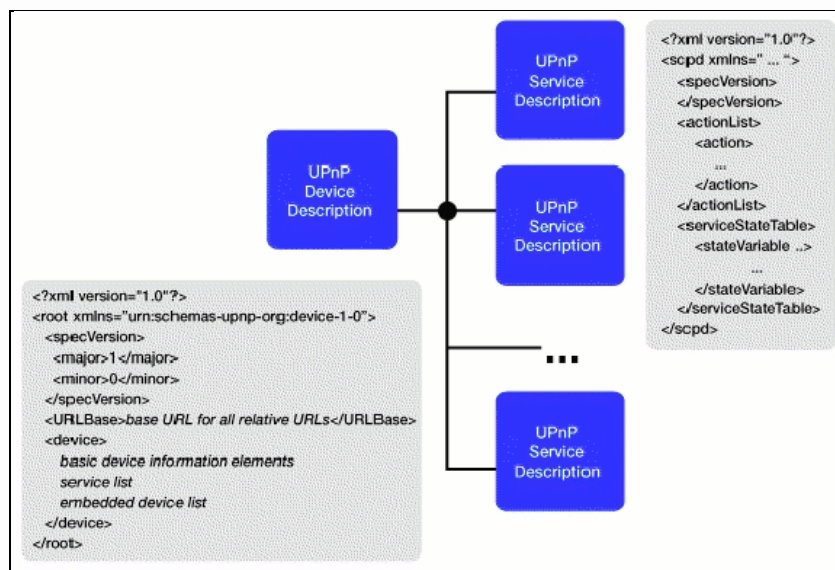
Ανακάλυψη. Έχοντας μια IP διεύθυνση, το πρώτο βήμα σε μια δικτύωση UPnP είναι η Ανακάλυψη. Όταν μια συσκευή προστίθεται στο δίκτυο, το UPnP πρωτόκολλο ανακάλυψης της επιτρέπει να διαφημίσει τις υπηρεσίες της στα σημεία έλεγχου του δικτύου. Ομοίως, όταν ένα σημείο ελέγχου προστίθεται στο δίκτυο, το πρωτόκολλο ανακάλυψης του επιτρέπει να αναζητήσει στο δίκτυο συσκευές του ενδιαφέροντος του. Η βασική ανταλλαγή και στις δύο περιπτώσεις αφορά ένα μήνυμα ανακάλυψης το οποίο περιέχει μερικές ουσιαστικές λεπτομέρειες σχετικές με τη συσκευή ή με μια από τις υπηρεσίες της, για παράδειγμα τον τύπο της, το αναγνωριστικό της και ένα δείκτη σε περισσότερο

λεπτομερείς πληροφορίες. Στην Εικόνα 10 παρουσιάζονται τα βήματα που πρέπει να εκτελεστούν από ένα UPnP σημείο ελέγχου για την ανακάλυψη των δυνατοτήτων μιας συσκευής.



Εικόνα 10. Ανάκτηση περιγραφής Συσκευής και Υπηρεσίας (Jerónimo, 2004)

Περιγραφή. Μετά την ανακάλυψη μιας συσκευής από ένα σημείο ελέγχου, το τελευταίο ακόμα γνωρίζει πολύ λίγα σχετικά με τη συσκευή. Το σημείο ελέγχου για να μάθει περισσότερα για τη συσκευή και τις δυνατότητες της, ή για να αλληλεπιδράσει μαζί της, πρέπει να ανακτήσει την περιγραφή της από το URL που παρέχεται από τη συσκευή στο μήνυμα ανακάλυψης. Η UPnP περιγραφή για μια συσκευή εκφράζεται σε XML και περιλαμβάνει πληροφορίες του κατασκευαστή όπως το όνομα και τον αριθμό του μοντέλου, το σειριακό αριθμό, το όνομα του κατασκευαστή, τα URLs του πωλητή κ.α.. Η περιγραφή περιλαμβάνει ακόμα μια λίστα από όλες τις ενσωματωμένες συσκευές και υπηρεσίες, καθώς και URLs για έλεγχο, ειδοποίηση συμβάντων και παρουσίαση. Κάθε περιγραφή υπηρεσίας περιλαμβάνει μια λίστα με τις εντολές ή τις ενέργειες, στις οποίες η υπηρεσία αποκρίνεται καθώς και παραμέτρους ή ορίσματα για κάθε ενέργεια. Η περιγραφή μιας υπηρεσίας περιλαμβάνει ακόμα μια λίστα από μεταβλητές οι οποίες μοντελοποιούν την κατάσταση της υπηρεσίας στη φάση της εκτέλεσης και περιγράφονται με όρους που έχουν να κάνουν με τον τύπο δεδομένων, το εύρος τιμών και τα χαρακτηριστικά της ειδοποίησης συμβάντων. Η ιεραρχία περιγραφής της συσκευής και της υπηρεσίας παρουσιάζεται στην Εικόνα 11.

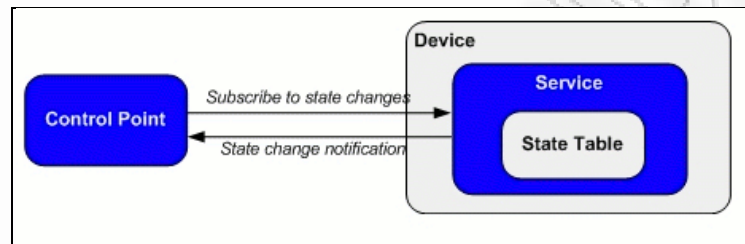


Εικόνα 11. Η UPnP ιεραρχία Συσκευής και Υπηρεσίας (Jeronimo, 2004)

Έλεγχος. Αφού έχει ανακτήσει την περιγραφή της συσκευής, το σημείο ελέγχου μπορεί να αποστείλει εντολές σε μια από τις υπηρεσίες της. Αυτό επιτυγχάνεται στέλνοντας το κατάλληλο μήνυμα ελέγχου στο URL της υπηρεσίας. Τα μηνύματα ελέγχου εκφράζονται επίσης σε XML και χρησιμοποιούν το πρωτόκολλο SOAP. Όπως συμβαίνει και με τις κλήσεις συνάρτησης, η υπηρεσία επιστρέφει τις σχετικές τιμές σε απάντηση στο μήνυμα ελέγχου. Τα αποτελέσματα της ενέργειας, αν υπάρχουν, διαμορφώνονται από τις αλλαγές στις μεταβλητές που περιγράφουν την κατάσταση της υπηρεσία κατά την εκτέλεση της.

Ειδοποίηση Συμβάντος. Το επόμενο βήμα της UPnP δικτύωσης είναι η ειδοποίηση συμβάντος. Μια UPnP περιγραφή μιας υπηρεσία περιλαμβάνει μια λίστα με ενέργειες στις οποίες απαντά η υπηρεσία και μια λίστα με μεταβλητές που διαμορφώνουν την κατάσταση της υπηρεσίας στο χρόνο εκτέλεσης. Η υπηρεσία δημοσιεύει ενημερώσεις όταν αυτές οι μεταβλητές αλλάζουν και ένα σημείο ελέγχου μπορεί να εγγραφεί για να λαμβάνει αυτή την πληροφορία. Η υπηρεσία δημοσιεύει ενημερώσεις στέλνοντας μηνύματα συμβάντων. Τα μηνύματα αυτά εμπεριέχουν το όνομα μιας ή περισσότερων μεταβλητών κατάστασης καθώς και την τρέχουσα τιμή αυτών των μεταβλητών. Και αυτά τα μηνύματα εκφράζονται σε XML και είναι μορφοποιημένα σύμφωνα με την αρχιτεκτονική GENA. Για την υποστήριξη σεναρίων με πολλαπλά σημεία ελέγχου, η ειδοποίηση συμβάντων είναι σχεδιασμένη να κρατά όλα τα σημεία

ελέγχου το ίδιο ενημερωμένα σχετικά με τα αποτελέσματα κάθε ενέργειας. Για αυτό, όλοι οι συνδρομητές λαμβάνουν όλα τα μηνύματα συμβάντων για κάθε μεταβλητή κατάστασης που έχει αλλάξει, ασχέτως του λόγου για τον οποίο άλλαξε η τιμή της. Η Εικόνα 12 δείχνει πώς ένα σημείο ελέγχου επικοινωνεί με μια υπηρεσία για να εγγραφεί σε αυτή και να λαμβάνει ειδοποιήσεις σχετικές με αλλαγές της μεταβλητής κατάστασης.



Εικόνα 12. Αλλαγές Κατάστασης και Ειδοποιήσεις (Jerónimo, 2004)

Παρουσίαση. Το τελικό βήμα της UPnP δικτύωσης είναι η παρουσίαση. Αν μια συσκευή διαθέτει URL για παρουσίαση, τότε το σημείο ελέγχου μπορεί να ανακτήσει τη σελίδα από αυτό το URL, να τη φορτώσει σε έναν φυλλομετρητή ιστού και αναλόγως των δυνατοτήτων της σελίδας, να επιτρέψει σε ένα χρήστη να ελέγξει τη συσκευή και να προβάλει την κατάσταση της. Ο βαθμός στον οποίο κάθε ένα από αυτά μπορούν να πραγματοποιηθούν, εξαρτάται από τις συγκεκριμένες δυνατότητες της σελίδας παρουσίασης και της συσκευής.

3.3 Jini

Πρόκειται για μια αρχιτεκτονική δικτύου για τη δημιουργία καταναμημένων συστημάτων, η οποία βασίζεται στις υπηρεσίες (Wikipedia, Jini, 2009) και η οποία αναπτύχθηκε από τη Sun. Το Jini προσφέρει τα μέσα για την αντιμετώπιση ζητημάτων που έχουν να κάνουν με την καταναμημένη επεξεργασία, την εξέλιξη του συστήματος, την προσαρμοστικότητα, την ασφάλεια και τη δυναμική διαχείριση των υπηρεσιών. Η δυνατότητα μετακίνησης κώδικα είναι ένα από τα βασικά χαρακτηριστικά της πλατφόρμας και παρέχει πολλά πλεονεκτήματα συμπεριλαμβανομένης της ανεξαρτησίας πρωτοκόλλων.

Ένας από τους σκοπούς του Jini είναι να μετατοπίσει την έμφαση της υπολογιστικής επεξεργασίας από την παραδοσιακή προσέγγιση του οδηγού-

δίσκου, σε μια περισσότερο δικτυοκεντρική. Με αυτό τον τρόπο οι πόροι μπορούν να χρησιμοποιούνται μέσω ενός δικτύου σαν να ήταν διαθέσιμοι τοπικά. Το Jini βασίζεται στην Java και μοιάζει με τη μέθοδο RMI, αλλά είναι πιο εξελιγμένο. Το Jini επιτρέπει πιο προχωρημένη αναζήτηση υπηρεσιών, μέσα από μια διαδικασία ανακάλυψης δημοσιευμένων υπηρεσιών.

Υπάρχουν τρία κύρια μέρη σε ένα σενάριο Jini, ο πελάτης, ο εξυπηρετητής και η υπηρεσία αναζήτησης (LookUp Service - LUS). Η υπηρεσία είναι ο πόρος που είναι διαθέσιμος στο κατακευματισμένο περιβάλλον. Αυτό μπορεί να περιλαμβάνει φυσικές συσκευές (όπως εκτυπωτές και οδηγούς δίσκων) και υπηρεσίες λογισμικού (όπως ένα ερώτημα στη βάση δεδομένων). Ο πελάτης είναι η οντότητα η οποία χρησιμοποιεί την υπηρεσία.

Δημιουργία και χρήση υπηρεσίας. Το πρώτο βήμα στη δημιουργία μιας υπηρεσίας Jini, είναι η διαδικασία ανακάλυψης (discovery). Σε αυτό το στάδιο η νέα υπηρεσία εντοπίζει την LUS. Μόλις βρεθεί η LUS επιστρέφει στην υπηρεσία ένα αντικείμενο Service Registrar, το οποίο χρησιμοποιείται για την εγγραφή της στο σχετικό κατάλογο υπηρεσιών. Η διαδικασία αυτή περιλαμβάνει την παροχή πληροφοριών σχετικές με την υπηρεσία που δημοσιεύεται, όπως είναι το ID της, το αντικείμενο που την υλοποιεί καθώς και άλλα χαρακτηριστικά.

Όταν ένας πελάτης θέλει να χρησιμοποιήσει μια υπηρεσία, πρέπει να χρησιμοποιήσει πρώτα την υπηρεσία ανακάλυψης για να βρει την LUS. Μόλις επιτευχθεί αυτό, ο πελάτης λαμβάνει ένα αντικείμενο Service Registrar, το οποίο το χρησιμοποιεί για να αναζητήσει την υπηρεσία στον κατάλογο της LUS με βάση τον τύπο, το όνομα ή την περιγραφή της. Η LUS επιστρέφει ένα Java Proxy, όπου προσδιορίζεται πώς θα γίνει η απευθείας σύνδεση με την υπηρεσία. Αυτό είναι ένα από τα χαρακτηριστικά του Jini το οποίο το καθιστά πιο ισχυρό σε σχέση με τη μέθοδο RMI, στην οποία ο πελάτης χρειάζεται να γνωρίζει εκ' των προτέρων την τοποθεσία της απομακρυσμένης υπηρεσίας. Με τη χρήση του Proxy, ο πελάτης μπορεί να συνδεθεί απευθείας με την υλοποίηση της υπηρεσίας και να τη χρησιμοποιήσει σαν να ήταν τοπική.

Οι υπηρεσίες στο περιβάλλον Jini δεν είναι αναγκαστικά για πάντα διαθέσιμες, γεγονός το οποίο οδηγεί στην έννοια της χρονομίσθωσης (leasing). Όταν μια υπηρεσία εγγράφεται στην LUS, της χορηγείται ένα μισθωτήριο (lease) με συγκεκριμένη διάρκεια, η οποία μπορεί είτε να οριστεί χειροκίνητα, είτε να πάρει μια προκαθορισμένη τιμή π.χ. «για πάντα». Τα μισθωτήρια πρέπει να ανανεώνονται περιοδικά, για να ελέγχεται αν μια υπηρεσία είναι ακόμα ζωντανή, κάτι το οποίο σημαίνει ότι αν αποτύχει ή γίνει μη προσπελάσιμη, τότε μπορεί να παυθεί.

3.4 Web Services

Web service είναι ένα σύστημα λογισμικού σχεδιασμένο να υποστηρίζει διαλειτουργική αλληλεπίδραση μηχανής-προς-μηχανή πάνω από ένα δίκτυο (W3C, 2004). Διαθέτει διεπαφή περιγραφόμενη σε μορφή επεξεργάσιμη από μηχανή (το WSDL). Τα άλλα συστήματα αλληλεπιδρούν με το Web service με έναν τρόπο που προκαθορίζεται από την περιγραφή του, χρησιμοποιώντας μηνύματα SOAP, τα οποία συνήθως μεταφέρονται μέσω HTTP με XML σειριακοποίηση σε συνδυασμό με άλλα πρότυπα σχετικά με το Web.

Agents και Υπηρεσίες. Ένα Web service είναι μια αφηρημένη ιδέα η οποία πρέπει να υλοποιείται από ένα συγκεκριμένο agent (Εικόνα 13). Ο agent είναι το συγκεκριμένο κομμάτι του λογισμικού ή του υλικού που στέλνει και δέχεται μηνύματα, ενώ υπηρεσία είναι ο πόρος που χαρακτηρίζεται από το αφηρημένο σύνολο της λειτουργικότητας που παρέχεται. Ο διαχωρισμός αυτός γίνεται πιο κατανοητός αν παραθέσουμε ένα παράδειγμα. Υλοποιούμε ένα συγκεκριμένο Web service χρησιμοποιώντας έναν agent τη μια μέρα (γραμμένο σε μια γλώσσα προγραμματισμού) και έναν άλλο agent την άλλη μέρα (γραμμένο σε άλλη γλώσσα προγραμματισμού) με την ίδια λειτουργικότητα. Παρόλο που ο agent έχει αλλάξει, το Web service παραμένει το ίδιο.

Αιτών και πάροχος. Ο σκοπός ενός Web service είναι να παρέχει μια λειτουργικότητα εκ μέρους του ιδιοκτήτη του (ένα πρόσωπο ή ένας οργανισμός). Η *οντότητα του παρόχου* είναι το πρόσωπο ή ο οργανισμός που παρέχει έναν κατάλληλο agent για την υλοποίηση μιας συγκεκριμένης υπηρεσίας. Μια

οντότητα αιτούντα είναι ένα πρόσωπο ή ένας οργανισμός ο οποίος ελπίζει να χρησιμοποιήσει το Web service της οντότητας του παρόχου. Θα χρησιμοποιήσει έναν αιτούντα agent για να ανταλλάξει μηνύματα με τον πάροχο agent της οντότητας του παρόχου. Για να μπορέσει η ανταλλαγή μηνύματος να είναι επιτυχημένη, πρέπει πρώτα η οντότητα του αιτούντα και η οντότητα του παρόχου να συμφωνήσουν τόσο στα σημασιολογικά χαρακτηριστικά όσο και στη μηχανική του ανταλλασσόμενου μηνύματος.

Περιγραφή της υπηρεσίας. Η μηχανική του ανταλλασσόμενου μηνύματος είναι τεκμηριωμένη σε μια περιγραφή του Web service (WSD). Το WSD είναι μια προδιαγραφή της διεπαφής του Web service η οποία είναι επεξεργάσιμη από τη μηχανή και είναι γραμμένη σε WSDL. Ορίζει τα μορφότυπα του μηνύματος, τους τύπους δεδομένων και τα πρωτόκολλα μεταφοράς τα οποία πρέπει να χρησιμοποιηθούν ανάμεσα στον αιτούντα agent και τον πάροχο agent. Ακόμα ορίζει μια ή περισσότερες τοποθεσίες δικτύου στις οποίες μπορεί να γίνει επίκληση ενός agent παρόχου και είναι δυνατόν ακόμα να παρέχει μερικές πληροφορίες σχετικές με την αναμενόμενη σχηματομορφή (pattern) ανταλλαγής μηνύματος. Στην ουσία, η περιγραφή της υπηρεσίας αναπαριστά μια συμφωνία η οποία διέπει τη μηχανική της αλληλεπίδρασης με την υπηρεσία.

Σημασιολογία (semantics). Η σημασιολογία ενός Web service είναι η αναμενόμενη προσδοκία σχετικά με τη συμπεριφορά της υπηρεσίας, ειδικότερα σε ότι αφορά την απάντηση στα μηνύματα που αποστέλλονται σε αυτή. Στην ουσία, πρόκειται για το *συμβόλαιο* ανάμεσα στην οντότητα του αιτούντα και την οντότητα του παρόχου αναφορικά με το σκοπό και τις συνέπειες της αλληλεπίδρασης. Αν και αυτό το συμβόλαιο αντιπροσωπεύει τη γενική συμφωνία ανάμεσα στην οντότητα του αιτούντα και την οντότητα του παρόχου σχετικά με το πώς και γιατί οι αντίστοιχοι agents θα αλληλεπιδράσουν, δεν είναι απαραίτητα γραμμένο ή σαφώς διαπραγματευμένο. Μπορεί να είναι προφανές ή εννοούμενο, προφορικό ή γραπτό, επεξεργάσιμο από μηχανή ή ανθρωποκεντρικό, μπορεί να είναι μια νομική συμφωνία ή μια ανεπίσημη (μη νομική) συμφωνία.

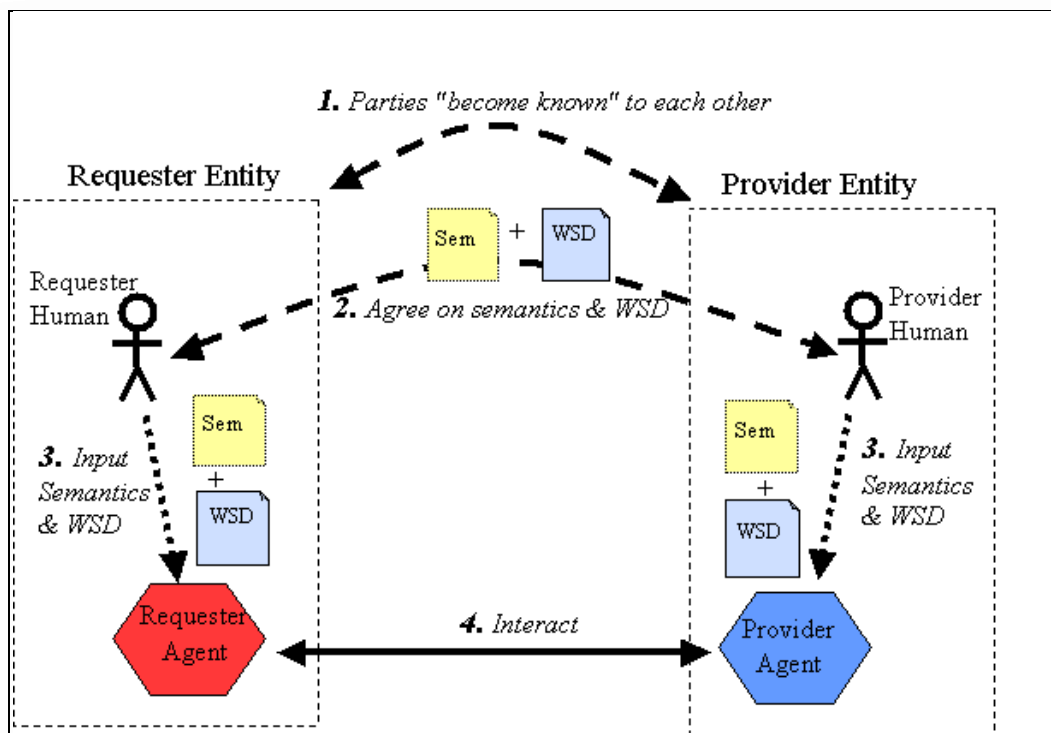
Ενώ η περιγραφή της υπηρεσίας αντιπροσωπεύει ένα συμβόλαιο το οποίο διέπει τη μηχανική της αλληλεπίδρασης με μια συγκεκριμένη υπηρεσία, η σημασιολογία

αντιπροσωπεύει ένα συμβόλαιο το οποίο διέπει τη σημασία και το σκοπό αυτής της αλληλεπίδρασης. Η διαχωριστική γραμμή ανάμεσα σε αυτά τα δύο δεν είναι απαραίτητα ξεκάθαρη. Όσο περισσότερο πλούσια σημασιολογικά γλώσσες χρησιμοποιούνται για να περιγραφεί η μηχανική της αλληλεπίδρασης, τόσο περισσότερη ουσιαστική πληροφορία μπορεί να μεταπηδήσει από την ανεπίσημη σημασιολογία στην περιγραφή της υπηρεσίας. Αν κάτι τέτοιο συμβεί, το μεγαλύτερο μέρος της δουλειάς που απαιτείται για μια επιτυχημένη αλληλεπίδραση μπορεί να αυτοματοποιηθεί.

Επισκόπηση της χρήσης ενός Web service. Υπάρχουν πολλοί τρόποι με τους οποίους μια οντότητα αιτούντα μπορεί να χρησιμοποιήσει ένα Web service. Σε γενικές γραμμές τα βήματα που απαιτούνται παρουσιάζονται στην Εικόνα 13. Πιο συγκεκριμένα :

1. Οι οντότητες του αιτούντα και του παρόχου γίνονται γνωστές η μια στην άλλη.
2. Οι οντότητες του αιτούντα και του παρόχου με κάποιο τρόπο συμφωνούν την περιγραφή της υπηρεσίας και τη σημασιολογία που θα διέπει την αλληλεπίδραση ανάμεσα στους agents τους.
3. Η περιγραφή της υπηρεσίας και η σημασιολογία γίνονται αντιληπτά από τους agents του αιτούντα και του παρόχου.
4. Οι agents του αιτούντα και του παρόχου ανταλλάσσουν μηνύματα, εκτελώντας κατά αυτό τον τρόπο μια εργασία εκ' μέρους των οντοτήτων του αιτούντα και του παρόχου.

Κάποια από αυτά τα βήματα μπορούν να αυτοματοποιηθούν, ενώ άλλα μπορούν να εκτελεστούν χειροκίνητα.



Εικόνα 13. Η γενική διαδικασία χρήσης ενός Web service (W3C, 2004)

3.5 JADE

Πρόκειται για μια υποδομή λογισμικού πλήρως υλοποιημένη σε γλώσσα Java, η οποία αναπτύχθηκε από την Telecom Italia Lab. Η JADE απλοποιεί την υλοποίηση συστημάτων multi-agent μέσω ενός μεσιμικού το οποίο είναι πλήρως συμμορφωμένο με τις προδιαγραφές FIPA και μέσω ενός συνόλου γραφικών εργαλείων τα οποία υποστηρίζουν τις φάσεις της ανάπτυξης και της εκφαλμάτωσης (debugging) (Jade, 2001). Η πλατφόρμα του agent μπορεί να κατανεμηθεί σε πολλές μηχανές (οι οποίες δεν χρειάζεται καν να εκτελούν το ίδιο λειτουργικό σύστημα) και η διαμόρφωση (configuration) της μπορεί να ελεγχθεί μέσω ενός απομακρυσμένου γραφικού περιβάλλοντος. Η διαμόρφωση μπορεί επίσης να αλλάξει ακόμα και σε κατάσταση εκτέλεσης, με τη μεταφορά agents από τη μια μηχανή στην άλλη όταν και όπως χρειαστεί.

Η αρχιτεκτονική επικοινωνίας προσφέρει ευελιξία και αποτελεσματική μηνυματοδοσία (messaging) με την JADE να δημιουργεί και να διαχειρίζεται μια ουρά από εισερχόμενα ACL μηνύματα, ιδιωτικά για κάθε agent. Οι agents μπορούν να προσπελάσουν τις ουρές τους μέσω ενός συνδυασμού πολλών

λειτουργιών : φραγή, σταθμοσκόπηση (polling), εξωχρονισμός (timeout) και ταίριασμα προτύπων (pattern matching). Έχει υλοποιηθεί το πλήρες μοντέλο επικοινωνίας FIPA και τα στοιχεία του έχουν διαχωριστεί και έχουν πλήρως ενοποιηθεί : πρωτόκολλα αλληλεπίδρασης, ACL, γλώσσες περιεχομένου, σχήματα εγκωδίκευσης (encoding schemes), οντολογίες και πρωτόκολλα μεταφοράς. Ο μηχανισμός μεταφοράς μοιάζει με χαμαιλέοντα που προσαρμόζεται σε κάθε κατάσταση, αφού μπορεί και επιλέγει το καλύτερο διαθέσιμο πρωτόκολλο. Java RMI, ειδοποίηση συμβάντων, HTTP και ΠΟΡ είναι μερικά από τα πρωτόκολλα που χρησιμοποιούνται, αλλά υπάρχει η δυνατότητα να προστεθούν εύκολα και άλλα μέσω των JADE διεπαφών MTP και IMTP.

3.6 Γλώσσες σήμανσης διεπαφής (User interface markup languages)

Πρόκειται για γλώσσες σήμανσης (markup languages), οι οποίες αποδίδουν και περιγράφουν γραφικές διεπαφές χρήστη και στοιχεία ελέγχου. Πολλές από αυτές είναι διάλεκτοι της XML και είναι εξαρτώμενες από προ-υπάρχουσες μηχανές σεναριογλωσσών (scripting languages), όπως είναι η μηχανή της JavaScript (Wikipedia, User interface markup language, 2008).

Η ιδέα αυτών των γλωσσών βασίζεται στην προσπάθεια να αποτραπεί η επανάληψη της ανακάλυψη του τροχού σε ότι αφορά τη σχεδίαση, την ανάπτυξη και τη λειτουργικότητα της διεπαφής χρήστη. Μια τυπική γλώσσα σήμανσης διεπαφής χρήστη, στερεοποιεί τα συχνά επαναχρησιμοποιήσιμα προγράμματα ή το σεναριοκώδικα, σε μορφή σήμανσης. Με αυτό τον τρόπο η εστίαση της προσπάθειας μεταφέρεται περισσότερο στο σχεδιασμό της διεπαφής και λιγότερο στη λειτουργικότητα της.

Οι γλώσσες σήμανσης διεπαφής, όπως και οι περισσότερες γλώσσες σήμανσης και προγραμματισμού, βασίζονται σε υπο-εφαρμογές εκτέλεσης για να διερμηνεύσουν και να αποδώσουν το δηλωτικό κώδικα σε κώδικα προγράμματος, ο οποίος μπορεί να επεξεργαστεί και να εξαχθεί στην επιθυμητή μορφή. Στην περίπτωση των γλωσσών σήμανσης διεπαφής που βασίζονται στην XML, την υπο-εφαρμογή εκτέλεσης αποτελεί ένας φυλλομετρητής ιστού (web browser). Το μεγαλύτερο μέρος του κώδικα επανα-διερμηνεύεται στον web browser σε

JavaScript κώδικα. Αυτό σημαίνει ότι η JavaScript μπορεί να χρησιμοποιηθεί για την επέκταση των δυνατοτήτων της γλώσσας σήμανσης.

Ακολουθεί μια συνοπτική αναφορά σε μερικές από τις πιο διαδεδομένες γλώσσες σήμανσης διεπαφής.

XUL (XML User Interface Language). Πρόκειται για τη βασισμένη σε XML γλώσσα του Mozilla, η οποία επιτρέπει την ανάπτυξη εφαρμογών πλούσιων σε χαρακτηριστικά και ανεξαρτήτων πλατφόρμας, οι οποίες μπορούν να εκτελεστούν είτε συνδεδεμένες με το διαδίκτυο είτε όχι (Mozilla, 2008).

XAML (Extensible Application Markup Language). Δημιουργήθηκε από τη Microsoft, βασίζεται στην XML και δίνει τη δυνατότητα στους προγραμματιστές να ορίσουν μια ιεραρχία από αντικείμενα τα οποία περιλαμβάνουν σεντ ιδιοτήτων καθώς και λογική (MSDN, 2009).

UIML (User Interface Markup Language). Σε αντίθεση με άλλες γλώσσες που βασίζονται στη XML και οι οποίες χρησιμοποιούνται για να καθορίσουν έγγραφα (δηλ. λέξεις, εικόνες κ.α.), η UIML χρησιμοποιείται για τον καθορισμό των πραγματικών στοιχείων της διεπαφής. Αυτά περιλαμβάνουν κουμπιά, μενού, λίστες και άλλα στοιχεία ελέγχου που δίνουν τη δυνατότητα σε ένα πρόγραμμα να λειτουργεί σε ένα γραφικό περιβάλλον όπως τα Windows. Η UIML χρησιμοποιείται για να προσδιορίσει τη θέση και το σχεδιασμό των στοιχείων ελέγχου, καθώς και τις ενέργειες που γίνονται όταν συμβαίνουν συγκεκριμένα γεγονότα (UIML.org, 2009).

4 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΠΡΟΤΕΙΝΟΜΕΝΗΣ ΠΛΑΤΦΟΡΜΑΣ

4.1 Εισαγωγή

Στα πλαίσια της παρούσας διπλωματικής εργασίας, σχεδιάστηκε και υλοποιήθηκε μια πλατφόρμα διαχείρισης υπηρεσιών σε ευφυή περιβάλλοντα. Σκοπός ήταν η πλατφόρμα αυτή να χρησιμοποιεί τεχνολογίες ενδιάμεσου λογισμικού προσανατολισμένες σε υπηρεσίες.

Η προτεινόμενη πλατφόρμα στηρίζεται σε μια αρχιτεκτονική που της επιτρέπει να ικανοποιεί τις πιο σημαντικές απαιτήσεις ενός συστήματος έξυπνου χώρου. Η υπηρεσιοστρεφής προσέγγιση και ο συνδυασμός των τεχνολογιών UPnP και OSGi προσφέρουν μεγάλες δυνατότητες επεκτασιμότητας, διαλειτουργικότητας και αλληλεπίδρασης. Παράλληλα ικανοποιείται η ανάγκη της χαλαρής σύζευξης, της ανεξαρτησίας πλατφόρμας εκτέλεσης και της εύκολης συντήρησης. Επιπλέον ενσωματώνει έναν δυναμικό και επεκτάσιμο μηχανισμό αλληλεπίδρασης μεταξύ αισθητήρων και εκκινητήρων για την καλύτερη διαχείριση και επεξεργασία των πληροφοριών που αφορούν το χρήστη και το περιβάλλον του.

Όπως αναφέρθηκε και παραπάνω (κεφάλαιο 2.4), σε ότι αφορά την αρχιτεκτονική των έξυπνων χώρων, δεν είναι εύκολο να υπάρξει ένα γενικό και ενιαίο πλαίσιο που να καλύπτει όλες τις ανάγκες. Αντιθέτως κάθε προσπάθεια επικεντρώνεται στην αντιμετώπιση συγκεκριμένων προκλήσεων, αναλόγως του περιβάλλοντος (έξυπνου χώρου) στο οποίο θέλουμε να εφαρμόσουμε την όποια σχεδίαση (Soldatos, Dimakis, Stamatis, & Polymenakos, 2007).

Για αυτό το λόγο επιλέξαμε η ανάπτυξη της πλατφόρμας να αφορά ένα συγκεκριμένο χώρο, έτσι ώστε να επικεντρωθούμε στην ικανοποίηση των ιδιαίτερων αναγκών του και να προκύψουν πιο αξιόπιστα συμπεράσματα. Ο χώρος που επιλέγει ήταν το *έξυπνο σπίτι*. Η επιλογή αυτή έγινε επειδή είναι αρκετά προφανείς οι ανάγκες που υπάρχουν και οι τρόποι ικανοποίησής τους, αλλά και επειδή υπάρχει ένα μεγάλο πεδίο εφαρμογών των τεχνολογιών της διεισδυτικής πληροφορικής σε τέτοια περιβάλλοντα. Οι λειτουργίες που

εκτελούνται καθημερινά σε ένα σπίτι είναι πολλές και τα αποτελέσματα τους έχουν επίδραση στις δραστηριότητες και τη διάθεση των ενοίκων του. Επίσης, ακριβώς επειδή πρόκειται για έναν ιδιωτικό χώρο, ο βαθμός ενόχλησης των ενοίκων από την ύπαρξη των συστατικών της πληροφορικής (συσκευές, εφαρμογές) πρέπει να είναι όσο το δυνατόν μικρότερος. Όλες αυτές οι διαπιστώσεις καθιστούν την προσπάθεια σχεδίασης και υλοποίησης μιας πλατφόρμας διαχείρισης υπηρεσιών για το συγκεκριμένο χώρο, πολύ ενδιαφέροντα.

4.2 Προδιαγραφές συστήματος

Η αρχιτεκτονική που σχεδιάστηκε και η υλοποίηση που πραγματοποιήθηκε, βασίστηκαν σε κάποιους περιορισμούς ως προς τον χώρο και τις συνθήκες που επικρατούν σε αυτόν. Αυτό έγινε επειδή κατά την ανάλυση απαιτήσεων του συστήματος, προέκυψαν πολλές λειτουργίες και πολλές διαφορετικές παράμετροι, οι οποίες δεν ήταν δυνατόν να ληφθούν όλες υπόψη στα πλαίσια μιας διπλωματικής εργασίας. Οι περιορισμοί αυτοί αφορούν τις λειτουργίες που αυτοματοποιήθηκαν, τα συστατικά της διεισδυτικής πληροφορικής που χρησιμοποιήθηκαν καθώς και την εκτέλεση πιο σύνθετων λειτουργιών από το σύστημα.

Λειτουργίες. Οι λειτουργίες που αυτοματοποιήθηκαν είναι η *ρύθμιση της θερμοκρασίας* και η *ρύθμιση του φωτός* σε επιθυμητά επίπεδα. Πρόκειται για δύο πολύ σημαντικές παραμέτρους διαβίωσης που επηρεάζουν την υγεία και τη διάθεση των ενοίκων σε πολύ μεγάλο βαθμό. Μια ακόμα λειτουργία που αυτοματοποιήθηκε είναι η *ανίχνευση κίνησης*, η οποία (σε συνδυασμό με άλλες λειτουργίες) έχει να κάνει με την ασφάλεια του σπιτιού και την εξοικονόμηση ενέργειας.

Συστατικά. Για την εκτέλεση των αυτοματοποιημένων λειτουργιών χρειάζεται η ενεργοποίηση των συστατικών της διεισδυτικής πληροφορικής. Τα συστατικά που επιλέξαμε χωρίζονται σε τρεις κατηγορίες :

- Εκκινητήρες (Actuators)
- Αισθητήρες (Sensors)

- Υπολογιστικές Μονάδες

Εκκινητήρες είναι οι συσκευές οι οποίες θέτουν σε λειτουργία κάποιο μηχανισμό. Στο συγκεκριμένο σύστημα τέτοιες συσκευές είναι οι *λαμπτήρες* και το *σύστημα κλιματισμού* (air-conditioning). Οι λειτουργίες του λαμπτήρα περιλαμβάνουν την *αύξηση* και την *μείωση* της έντασης του φωτός, ενώ του συστήματος κλιματισμού την *αύξηση* και τη *μείωση* της θερμοκρασίας του περιβάλλοντος.

Αισθητήρες είναι οι συσκευές οι οποίες μετρούν μια φυσική ποσότητα και την μετατρέπουν σε σήμα το οποίο μπορεί να αναγνωστεί από έναν παρατηρητή ή ένα όργανο. Οι αισθητήρες που χρησιμοποιήθηκαν είναι αισθητήρες *φωτός*, *θερμοκρασίας* και *κίνησης*.

Υπολογιστική Μονάδα είναι μια συσκευή η οποία μπορεί να χειρίζεται δεδομένα σύμφωνα με μια λίστα οδηγιών. Στο σύστημα μας το ρόλο αυτό μπορεί να επιτελέσει είτε ένας προσωπικός υπολογιστής (PC), είτε μια συσκευή χειρός (PDA), είτε οποιαδήποτε άλλη υπολογιστική μηχανή αρκεί να έχει εγκατεστημένη το Java Runtime Environment.

Σύνθετες Λειτουργίες. Υπάρχουν και λειτουργίες, οι οποίες προκύπτουν από την σύνθεση άλλων απλών λειτουργιών. Μια τέτοια σύνθετη λειτουργία που αυτοματοποιήθηκε είναι η *μεγιστοποίηση της φωτεινότητας με την ανίχνευση κίνησης*. Ένας αισθητήρας κίνησης ελέγχει συνεχώς το χώρο και μόλις αντιληφθεί κάποια κινητικότητα ενεργοποιεί το φωτισμό.

Οι περιορισμοί που αναφέρθηκαν θεωρούμε ότι δεν στερούν από την προτεινόμενη αρχιτεκτονική τη δυνατότητα επέκτασης ώστε να καλύψει περισσότερες λειτουργικότητες σε ένα περιβάλλον έξυπνου σπιτιού.

4.3 Αρχιτεκτονική πλατφόρμας IESAP

Με βάση τις προδιαγραφές που περιγράφηκαν, σχεδιάστηκε μια πλατφόρμα (IESAP) διαχείρισης υπηρεσιών σε περιβάλλον έξυπνου σπιτιού. Η αρχιτεκτονική της πλατφόρμας IESAP αποτελείται από πέντε στρώματα : *Φυσικό Στρώμα*, *Στρώμα Υπηρεσιών Αισθητήρων/Εκκινητήρων*, *Στρώμα Αντίληψης*, *Στρώμα Διαχείρισης Υπηρεσιών*, *Στρώμα Εφαρμογής*.

Φυσικό Στρώμα. Το στρώμα αυτό αποτελείται από τις συσκευές του συστήματος. Πρόκειται δηλαδή για τους αισθητήρες (θερμοκρασίας, φωτός, κίνησης), τους εκκινήτες (λαμπτήρες, κλιματιστικό) και τις υπολογιστικές μονάδες (PC). Σε μια επέκταση της πλατφόρμας, το στρώμα αυτό θα μπορούσε να περιλαμβάνει οποιαδήποτε ηλεκτρική συσκευή όπως τηλεόραση, ραδιόφωνο, σύστημα ασφαλείας κ.α..

Στρώμα Υπηρεσιών Αισθητήρων/Εκκινήτηρων. Το στρώμα αυτό αποτελεί τον ενδιάμεσο μεταξύ του φυσικού κόσμου και του συστήματος. Εδώ κάθε αντικείμενο του φυσικού κόσμου (αισθητήρας, εκκινήτης, συσκευή) μετατρέπεται σε υπηρεσία λογισμικού, η οποία μπορεί να προγραμματιστεί για να επιτελέσει μια συγκεκριμένη εργασία ή μαζί με άλλες υπηρεσίες να συνθέσουν μια πιο σύνθετη λειτουργία.

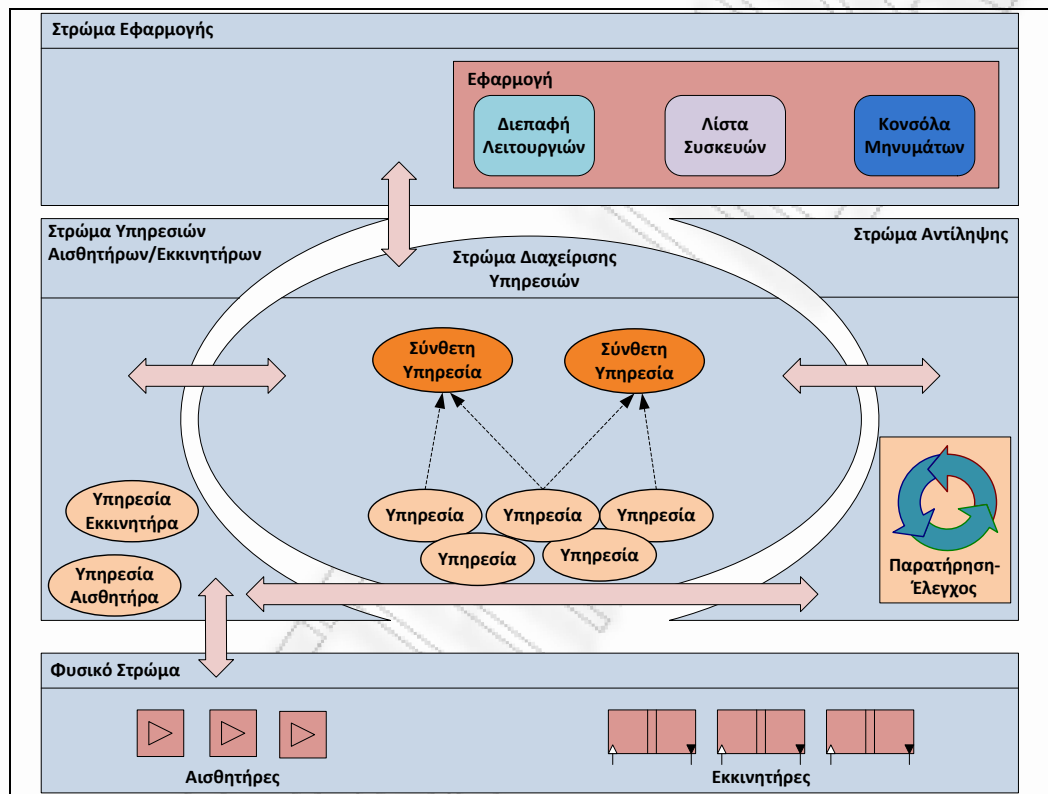
Στρώμα Αντίληψης. Το στρώμα αυτό είναι υπεύθυνο για την ανίχνευση των συνθηκών του περιβάλλοντος και της δραστηριότητας των παρευρισκομένων. Συνεργάζεται με το *Στρώμα Υπηρεσιών Αισθητήρων/Εκκινήτηρων* για να πάρει πληροφορίες και μετρήσεις σχετικές με το περιβάλλον, αποφασίζει για τις ενέργειες που πρέπει να γίνουν και κατόπιν του ζητά να τις εκτελέσει.

Στρώμα Διαχείρισης Υπηρεσιών. Το στρώμα αυτό είναι υπεύθυνο για τη διαχείριση και το συγχρονισμό των υπηρεσιών και συνεργάζεται με όλα τα υπόλοιπα στρώματα του συστήματος (πλην του *Φυσικού Στρώματος*). Διαχειρίζεται τόσο τις απλές όσο και τις σύνθετες υπηρεσίες. Με την ενεργοποίηση μιας συσκευής, αναλαμβάνει την εγκατάστασης της στο σύστημα, διαδικασία η οποία περιλαμβάνει την ενημέρωση ενός κατάλογου τοπολογίας έτσι ώστε η υπηρεσία της να είναι διαθέσιμη σε όλο το σύστημα.

Στρώμα Εφαρμογής. Το στρώμα αυτό αποτελεί τον ενδιάμεσο μεταξύ του συστήματος και του χρήστη. Επικοινωνεί με το *Στρώμα Διαχείρισης Υπηρεσιών* και χρησιμοποιεί τόσο τις απλές όσο και τις σύνθετες υπηρεσίες. Περιλαμβάνει την υποεφαρμογή διαχείρισης λειτουργιών η οποία αποτελείται από ένα γραφικό παραθυρικό περιβάλλον εξοπλισμένο με στοιχεία ελέγχου για το χειρισμό και την εκτέλεση των λειτουργιών του συστήματος. Ακόμα διαθέτει μια υποεφαρμογή

αποσφαλμάτωσης η οποία προβάλλει μηνύματα του συστήματος, καθώς και μια υποεφαρμογή προβολής της λίστας των συσκευών που είναι εγκατεστημένες, μαζί με τα χαρακτηριστικά τους.

Η συνολική αρχιτεκτονική της πλατφόρμας IESAP παρουσιάζεται σχηματικά στην Εικόνα 14.



Εικόνα 14. Διαστρωμάτωση πλατφόρμας IESAP

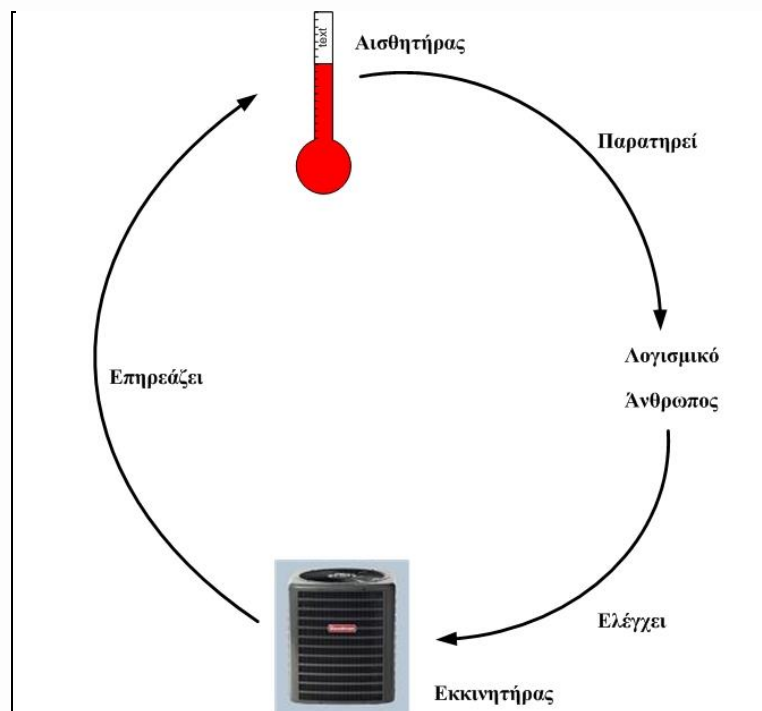
4.3.1 Διαχείριση δεδομένων αισθητήρων (context management)

Μια πολύ σημαντική διαδικασία σε ένα περιβάλλον έξυπνου σπιτιού είναι η διαχείριση των δεδομένων που λαμβάνονται από τους διάφορους αισθητήρες. Στην πλατφόρμα IESAP τα δεδομένα των αισθητήρων χρησιμοποιούνται για την πραγματοποίηση από το σύστημα συγκεκριμένων ενεργειών μέσω των εκκινήτριων, οι οποίες επηρεάζουν το περιβάλλον και δημιουργούν νέες συνθήκες. Εκτός από τους εκκινήτριες, τις συνθήκες του περιβάλλοντος μεταβάλλουν και οι ανθρώπινες δραστηριότητες.

Για παράδειγμα η λειτουργία της *ρύθμισης θερμοκρασίας* περιλαμβάνει τις εξής διαδικασίες :

1. Ο χρήστης ορίζει την επιθυμητή θερμοκρασία
2. Ο αισθητήρας θερμοκρασίας στέλνει στο σύστημα την μέτρηση για την τρέχουσα θερμοκρασία
3. Το σύστημα συγκρίνει τη μέτρηση με την επιθυμητή από το χρήστη τιμή
4. Αν η τρέχουσα τιμή είναι μεγαλύτερη, τότε δίνει εντολή στο κλιματιστικό να μειώσει τη θερμοκρασία
5. Αν η τρέχουσα τιμή είναι μικρότερη, τότε δίνει εντολή στο κλιματιστικό να αυξήσει τη θερμοκρασία
6. Αν η τρέχουσα τιμή είναι ίση με την επιθυμητή τότε δεν δίνει καμία εντολή
7. Η διαδικασία αυτή επαναλαμβάνεται συνεχώς

Ο εκκινητήρας του συστήματος, δηλαδή το κλιματιστικό, επηρεάζει τις συνθήκες του περιβάλλοντος αυξομειώνοντας τη θερμοκρασία. Ο έλεγχος του κλιματιστικού επηρεάζεται από δύο παράγοντες. Ο ένας παράγοντας είναι το λογισμικό, το οποίο ανάλογα με τις μετρήσεις του αισθητήρα δίνει εντολή για αύξηση ή μείωση της θερμοκρασίας. Ο άλλος παράγοντας είναι ο ένοικος του σπιτιού, ο οποίος μπορεί να το ενεργοποιήσει/απενεργοποιήσει ή να αυξομειώσει την ένταση του. Η επαναλαμβανόμενη διαδικασία της *παρατήρησης-ελέγχου* και η αλληλεπίδραση μεταξύ των αισθητήρων και των εκκινητήρων, φαίνεται στην Εικόνα 15.



Εικόνα 15. Η αλληλεπίδραση μεταξύ Αισθητήρα και Εκκινητήρα

Εκτός της λειτουργίας της ρύθμισης θερμοκρασίας, η ίδια διαδικασία της παρατήρησης και του ελέγχου πραγματοποιείται και για τις υπόλοιπες λειτουργίες του συστήματος (ρύθμιση φωτεινότητας, ανίχνευση κίνησης).

Σε ένα ολοκληρωμένο περιβάλλον έξυπνου σπιτιού, η μεταβολή μιας συνιστώσας του περιβάλλοντος είναι πιθανό να προκαλείται από πολλές διαφορετικές ενέργειες. Για παράδειγμα, η ρύθμιση της φωτεινότητας ενός χώρου μπορεί να επιτευχθεί είτε με τη χρήση ενός λαμπτήρα, είτε με το ανοιγόκλεισμα των παραθυρόφυλλων. Για αυτό το λόγο, στη πλατφόρμα IESAP γίνεται διαχωρισμός των εννοιών *ενέργεια* και *υπηρεσία*. Η πρώτη έχει πιο γενική και αφηρημένη σημασία, αφορά την αντίδραση που προκαλείται από την παρατήρηση ενός αισθητήρα και μπορεί να υλοποιείται από πολλές υπηρεσίες. Η δεύτερη αποτελεί την υπηρεσία της *Υπηρεσιοστρεφής Αρχιτεκτονικής* και υλοποιείται από τη λειτουργικότητα μιας συσκευής. Στο παράδειγμα της ρύθμισης της φωτεινότητας, η *αύξηση* και η *μείωση* της φωτεινότητας αποτελούν *ενέργειες* οι οποίες μπορούν να υλοποιούνται από υπηρεσίες όπως «αύξηση έντασης λαμπτήρα» ή «άνοιγμα παραθυρόφυλλων» και «μείωση έντασης λαμπτήρα» ή «κλείσιμο παραθυρόφυλλων» αντίστοιχα.

Στον Πίνακα 1 παρουσιάζονται οι σχέσεις που συνδέουν λειτουργίες, παρατηρήσεις, ενέργειες και υπηρεσίες.

Πίνακας 1. Σχέση Λειτουργίας, Παρατήρησης, Ενέργειας, Υπηρεσίας

Λειτουργία	Παρατήρηση	Ενέργεια	Υπηρεσία
Ρύθμιση Θερμοκρασίας	> επιθυμητό	Μείωση Θερμοκρασίας	Εξαγωγή Κρύου Αέρα ...
	< επιθυμητό	Αύξηση Θερμοκρασίας	Εξαγωγή Ζεστού Αέρα ...
	= επιθυμητό	--	--
Ρύθμιση Φωτεινότητας	> επιθυμητό	Μείωση Φωτεινότητας	Μείωση Έντασης Λαμπτήρα Κλείσιμο Παραθυρόφυλλων ...
	< επιθυμητό	Αύξηση Φωτεινότητας	Αύξηση Έντασης Λαμπτήρα Άνοιγμα Παραθυρόφυλλων ...
	= επιθυμητό	--	--
Ενεργοποίηση Φωτισμού με Ανίχνευση Κίνησης	ΝΑΙ	Μεγιστοποίηση Φωτεινότητας	Μέγιστη Ένταση Λαμπτήρα ...
	ΟΧΙ	Ελαχιστοποίηση Φωτεινότητας	Ελάχιστη Ένταση Λαμπτήρα ...

4.4 Επιλογή τεχνολογιών

Οι έξυπνοι χώροι που αναπτύσσονται σήμερα, στηρίζονται κυρίως σε εταιρικές τεχνολογίες και για αυτό το λόγο στερούνται διαλειτουργικότητας και μακροπρόθεσμης προοπτικής εξέλιξης. Στο μέλλον οι χώροι αυτοί αναμένεται να περιλαμβάνουν μια μεγάλη ποικιλία συσκευών και υπηρεσιών, οι οποίες θα προέρχονται από διαφορετικούς κατασκευαστές και δημιουργούς αντίστοιχα.

Η ανάγκη της διαλειτουργικότητας σε ένα περιβάλλον που συνεχώς μεταβάλλεται, οδηγεί στη σχεδίαση μιας αρχιτεκτονικής η οποία θα είναι προσανατολισμένη στις υπηρεσίες (service oriented architecture – SOA). Υπηρεσιοστρεφής είναι η αρχιτεκτονική που σκοπό έχει να επιτύχει τη χαλαρή σύνδεση μεταξύ αλληλεπιδρώντων agents λογισμικού. Υπηρεσία είναι η εργασία που εκτελείται από έναν πάροχο υπηρεσίας για να επιτευχθεί το επιθυμητό τελικό αποτέλεσμα για έναν καταναλωτή υπηρεσίας. Τόσο ο πάροχος όσο και ο καταναλωτής είναι ρόλοι που υποδύονται από agents λογισμικού εκ' μέρους των ιδιοκτητών τους. Με δεδομένη την αναγκαιότητα μιας υπηρεσιοστρεφούς αρχιτεκτονικής και την απαίτηση για διαλειτουργικότητα, η βασική τεχνολογία που επιλέξαμε για την ανάπτυξη της πλατφόρμας IESAP είναι η Open Service Gateway Initiative (OSGi).

Η OSGi προσπαθεί να ικανοποιήσει τις απαιτήσεις των έξυπνων χώρων προσφέροντας ένα ελεγχόμενο, επεκτάσιμο πλαίσιο για τη σύνδεση διαφόρων συσκευών σε ένα τοπικό δίκτυο ενός σπιτιού, ενός γραφείου ή ενός οχήματος. Ορίζοντας ένα στάνταρντ περιβάλλον εκτέλεσης και διεπαφές υπηρεσίας, η OSGi προάγει τη δυναμική ανακάλυψη και συνεργασία συσκευών και υπηρεσιών από διαφορετικές πηγές. Επιπλέον το πλαίσιο OSGi είναι έτσι σχεδιασμένο ώστε να διασφαλίζει την ομαλή εξέλιξη του χώρου στο χρόνο και να υποστηρίζει τη συνδεσιμότητα με τον έξω κόσμο, προσφέροντας απομακρυσμένο έλεγχο, διάγνωση και διαχείριση (Lee, Nordstedt, & Helal, 2003).

Η OSGi υιοθετεί τη χρήση του μοντέλου SOA σε τοπικές εφαρμογές επιτρέποντας την ανάπτυξη εφαρμογών με τη μορφή δομοστοιχείων (modules) τα οποία ονομάζονται δέσμες (bundles). Έτσι εφόσον οι εφαρμογές χωρίζονται σε

bundles, είναι εφικτός ο διαμοιρασμός κώδικα ή υπηρεσιών. Επιπρόσθετα, με τη χρήση κάποιων μηχανισμών είναι εφικτή η ενημέρωση ενός bundle χωρίς να χρειάζεται η επανεκκίνηση της εφαρμογής (Gouin-Vallerand & Giroux, 2007).

Επιπλέον η τεχνολογία OSGi έχει τη δυνατότητα να συμμορφώνεται με πολυάριθμα στάνταρντ. Καθώς διαθέτει ένα καλά καθορισμένο εξωτερικό πρωτόκολλο διεπαφής υπηρεσιών, μπορεί και παρέχει συμβατές υπηρεσίες για τον έλεγχο συσκευών διαφορετικών πρωτοκόλλων (Bai & Hsu, 2007).

Όπως προκύπτει από τα παραπάνω, η χρήση της τεχνολογίας OSGi σε μια πλατφόρμα έξυπνου χώρου εξυπηρετεί την ανάγκη για διαχείριση των υπηρεσιών του ευφυούς περιβάλλοντος, ειδικότερα σε ότι αφορά τη δυναμική ανάπτυξη, την εξάπλωση, την ανακάλυψη και την εντοπιστική τους. Υπάρχει όμως και η ανάγκη για συνεργασία μεταξύ των συστατικών της υποδομής του περιβάλλοντος (αισθητήρες, εκκινητήρες, συσκευές) καθώς και η ένταξη τους στο σύστημα με έναν ενιαίο τρόπο. Η τεχνολογία που μπορεί να επιτύχει κάτι τέτοιο είναι η UPnP (Universal Plug and Play).

Το UPnP είναι ένα σετ από πρωτόκολλα δικτύου το οποίο επιτρέπει σε συσκευές όπως οι υπολογιστές, οι ευφυείς συσκευές και οι ασύρματες συσκευές να συνδέονται άρρηκτα και να απλοποιούν την πραγματοποίηση των περιφερειακών δικτύων (ambient networks) σε σπίτια και εταιρικά περιβάλλοντα.

Οι τεχνολογίες UPnP και OSGi είναι πλήρως συμπληρωματικές. Είναι ένας συνδυασμός που θεωρείται πολύ ισχυρός και μπορεί να βοηθήσει στην εκπλήρωση των στόχων μιας πλατφόρμας έξυπνων χώρων (Stavroulaki, Demestichas, Adamopoulou, & Demestichas, 2006). Για αυτό το λόγο η υλοποίηση της πλατφόρμας IESAP βασίστηκε σε αυτές τις δύο τεχνολογίες. Σε ότι αφορά τη διαχείριση των υπηρεσιών και εφαρμογών του ευφυούς περιβάλλοντος, αυτή γίνεται από την OSGi. Οι υπηρεσίες και οι εφαρμογές αναπτύσσονται στην πλατφόρμα σαν δέσμες OSGi, επιτρέποντας έτσι τη δυναμική εγγραφή, ενεργοποίηση/απενεργοποίηση και ανακάλυψη των υπηρεσιών καθώς και την αλληλεπίδραση μεταξύ υπηρεσιών που παρέχονται από διαφορετικούς πωλητές. Όμως οι υπηρεσίες και οι εφαρμογές του ευφυούς

περιβάλλοντος εκτός από την μεταξύ τους αλληλεπίδραση χρειάζεται να αλληλεπιδρούν και με τα άλλα στοιχεία του ευφυούς περιβάλλοντος, όπως την υποδομή του δικτύου. Αυτό επιτυγχάνεται με τη χρήση της URnP τεχνολογίας, μέσω της μετατροπής κάθε λειτουργίας των στοιχείων του φυσικού στρώματος, όπως αισθητήρες, εκκινητήρες και συσκευές, σε URnP υπηρεσία.

5 ΥΛΟΠΟΙΗΣΗ ΠΛΑΤΦΟΡΜΑΣ

Η πλατφόρμα IESAP υλοποιήθηκε με τη χρήση της τεχνολογίας OSGi (Bartlett, 2007) (Hall, 2004) στο περιβάλλον ανάπτυξης Eclipse SDK. Πιο συγκεκριμένα χρησιμοποιήθηκε η έκδοση «Eclipse for RCP/Plug-in Developers» η οποία ενσωματώνει την Equinox υλοποίηση της πλατφόρμας OSGi (Equinox, 2004) (Watson & Kaegi, 2007). Κάθε bundle της OSGi αποτελεί και ένα project του Eclipse. Υπάρχουν περίπου 30 bundles κάθε ένα από το οποίο ανήκει και σε ένα από τα στρώματα της αρχιτεκτονικής της πλατφόρμας.

Το σενάριο για την εφαρμογή της λειτουργίας της πλατφόρμας περιλαμβάνει ένα σπίτι με δύο δωμάτια, μια κουζίνα και ένα υπνοδωμάτιο. Υπάρχουν επίσης τρία είδη αισθητήρων, (θερμοκρασίας, φωτός, κίνησης) και δύο είδη εκκινήτων (λαμπτήρας, κλιματιστικό). Το πλήθος των αισθητήρων και των εκκινήτων ανά δωμάτιο παρουσιάζονται στον Πίνακα 2.

Πίνακας 2. Κατανομή αισθητήρων και εκκινήτων

Δωμάτιο	Συσκευή	Πλήθος
Κουζίνα	Αισθητήρας Θερμοκρασίας	1
	Αισθητήρας Φωτός	1
	Αισθητήρας Κίνησης	1
	Κλιματιστικό	1
	Λαμπτήρας	2
Υπνοδωμάτιο	Αισθητήρας Θερμοκρασίας	1
	Αισθητήρας Φωτός	1
	Αισθητήρας Κίνησης	1
	Κλιματιστικό	1
	Λαμπτήρας	2

Με βάση τις προδιαγραφές του συστήματος αυτοματοποιήθηκαν οι λειτουργίες :

- Ρύθμιση θερμοκρασίας
- Ρύθμιση φωτεινότητας
- Ενεργοποίηση φωτισμού με ανίχνευση κίνησης

οι οποίες είναι διαθέσιμες και στα δύο δωμάτια του σπιτιού.

5.1 Προσομοιωτής περιβάλλοντος

Για να είναι εφικτή η δοκιμή της πλατφόρμας IESAP χρειάστηκε να αναπτυχθεί μια εφαρμογή προσομοίωσης περιβάλλοντος. Ο ρόλος αυτής της εφαρμογής είναι να προσομοιώνει τις συνθήκες περιβάλλοντος σε κάθε δωμάτιο, ειδικότερα σε ότι αφορά τη θερμοκρασία, τη φωτεινότητα και την κίνηση. Για κάθε ένα από αυτά τα χαρακτηριστικά ορίστηκε μια μεταβλητή η τιμή της οποίας προσδιορίζει την αντίστοιχη τρέχουσα τιμή του χαρακτηριστικού του κάθε δωματίου. Στον Πίνακα 3 παρουσιάζονται τα χαρακτηριστικά της προσομοίωσης ανά δωμάτιο, η μονάδα μέτρησης και το εύρος τιμών τους.

Πίνακας 3. Χαρακτηριστικά περιβάλλοντος προσομοίωσης

Δωμάτιο	Χαρακτηριστικό	Μονάδα Μέτρησης	Εύρος Τιμών
Κουζίνα	Θερμοκρασία	Κελσίου (C)	-10...50
	Φωτεινότητα	Λουξ (lux)	0...200
	Κίνηση	---	NAI/OXI
Υπνοδωμάτιο	Θερμοκρασία	Κελσίου (C)	-10...50
	Φωτεινότητα	Λουξ (lux)	0...200
	Κίνηση	---	NAI/OXI

Σε πραγματικές συνθήκες τα χαρακτηριστικά του περιβάλλοντος *θερμοκρασία* και *φωτεινότητα* μπορούν να επηρεαστούν είτε από τις ενέργειες των εκκινήτηρων (αυξομείωση έντασης λαμπτήρα, κλιματιστικού) είτε από εξωτερικούς του συστήματος παράγοντες (άνοιγμα παραθύρου, πτώση της εξωτερικής θερμοκρασίας, δύση του ηλίου, κ.α.). Αντίστοιχη είναι και η λειτουργία του προσομοιωτή, επιτρέποντας τη μεταβολή της τιμής των χαρακτηριστικών είτε από την πλατφόρμα IESAP, είτε από το χρήστη (προσομοίωση εξωτερικών παραγόντων).

5.1.1 Τεχνικά χαρακτηριστικά προσομοιωτή

Η εφαρμογή προσομοίωσης διαθέτει ένα παραθυρικό περιβάλλον για την παρακολούθηση και τον έλεγχο των τιμών των μεταβλητών κάθε δωματίου από το χρήστη. Αναπτύχθηκε σε Java, αποτελείται από 3 κλάσεις και η επικοινωνία με την πλατφόρμα IESAP γίνεται μέσω sockets. Οι αρχικές τιμές των χαρακτηριστικών (θερμοκρασία, φωτεινότητα, κίνηση) του περιβάλλοντος για κάθε δωμάτιο, βρίσκονται αποθηκευμένες σε ένα xml αρχείο το οποίο διαβάζει η εφαρμογή με την εκκίνηση της. Ακολουθεί μια συνοπτική περιγραφή των κλάσεων.

Eclipse Project : *EnvironmentServer*

Classes :

EnvironmentTable. Η κλάση αυτή αποτελείται από 3 ιδιότητες που αφορούν, τον κωδικό του δωματίου, το χαρακτηριστικό του περιβάλλοντος (θερμοκρασία, φωτεινότητα, κίνηση) και την αντίστοιχη τιμή του χαρακτηριστικού. Χρησιμοποιείται σαν πίνακας αποθήκευσης των τρεχουσών τιμών των χαρακτηριστικών ανά δωμάτιο.

StartServer. Η κλάση αυτή δημιουργεί και διαχειρίζεται το παραθυρικό περιβάλλον της εφαρμογής, διαβάζει το αρχείο xml με τις αρχικές τιμές των μεταβλητών και δημιουργεί τη σύνδεση μέσω ServerSocket με την πλατφόρμα IESAP.

MainServer. Η κλάση αυτή αναλαμβάνει να εξυπηρετεί τα αιτήματα της πλατφόρμας IESAP για ανάκτηση της τιμής κάποιου χαρακτηριστικού του περιβάλλοντος ή για μεταβολή της τιμής του. Το μήνυμα που λαμβάνει από την πλατφόρμα αποτελείται από τα πεδία :

- Κωδικός πρωτοκόλλου επικοινωνίας
- Κωδικός δωματίου
- Χαρακτηριστικό περιβάλλοντος (θερμοκρασία, φωτεινότητα, κίνηση)
- Ενέργεια (αύξηση, μείωση)

- Βήμα μεταβολής (βήμα αύξησης/μείωσης)

5.2 Εξομοιωτές συσκευών UPnP

Εκτός της προσομοίωσης του περιβάλλοντος, η λειτουργία της πλατφόρμας IESAP προϋποθέτει και την ύπαρξη συσκευών όπως είναι οι αισθητήρες και οι εκκινητήρες, οι οποίες θα πρέπει να είναι συμβατές με το πρωτόκολλο δικτύωσης UPnP. Για το λόγο αυτό αναπτύχθηκαν *εικονικές* συσκευές οι οποίες εξομοιώνουν τη λειτουργία των αντίστοιχων πραγματικών και επιπλέον προσφέρουν στην πλατφόρμα IESAP υπηρεσίες UPnP.

Η ανάπτυξη αυτών των συσκευών βασίστηκε στα παραδείγματα UPnP συσκευών του Apache Felix, μιας υλοποίησης ανοικτού λογισμικού της πλατφόρμας OSGi (Apache, 2006). Με βάση αυτά τα παραδείγματα, δημιουργήθηκαν 5 java packages, ένα για κάθε τύπο συσκευής (αισθητήρας θερμοκρασίας, αισθητήρας φωτεινότητας, αισθητήρας κίνησης, κλιματιστικό, λαμπτήρας). Κάθε package περιλαμβάνει ένα σύνολο από κλάσεις που υλοποιούν τις διεπαφές του UPnP πρωτοκόλλου. Για παράδειγμα στην περίπτωση του αισθητήρα θερμοκρασίας έχουμε :

Eclipse Project : *devices_sensors_temperature*

Classes :

GetTemperatureAction. Η κλάση αυτή υλοποιεί τη διεπαφή *UPnPAction* η οποία αφορά τις ενέργειες που μπορούν να εκτελεστούν από μια υπηρεσία και οι οποίες μπορούν να επηρεάσουν τις μεταβλητές κατάστασης.

TemperatureResultStateVariable. Η κλάση αυτή υλοποιεί τη διεπαφή *UPnPStateVariable* η οποία αφορά τη μεταβλητή κατάσταση που συνδέεται με μια UPnP υπηρεσία.

TemperatureSensor. Η κλάση αυτή πραγματοποιεί την επικοινωνία με τον προσομοιωτή περιβάλλοντος αποστέλλοντας το μήνυμα που περιγράφηκε στην κλάση *MainServer* και λαμβάνοντας το μήνυμα που περιέχει την τιμή της μέτρησης, στην προκειμένη περίπτωση την τιμή της θερμοκρασίας.

TemperatureSensorDevice. Η κλάση αυτή υλοποιεί τις διεπαφές *UPnPDevice* και *UPnPDeviceExInf*. Η πρώτη αφορά την περιγραφή μιας UPnP συσκευής και περιλαμβάνει τον προσδιορισμό των χαρακτηριστικών της όπως :

- Την κατηγορία
- Το «φιλικό» όνομα
- Το αναγνωριστικό
- Τον κατασκευαστή
- Το URL του κατασκευαστή
- Την περιγραφή του μοντέλου
- Το όνομα του μοντέλου
- Τον αριθμό του μοντέλου
- Τον σειριακό αριθμό
- Τον τύπο

Η διεπαφή *UPnPDeviceExInf* δεν περιλαμβάνεται σε αυτές του πρωτοκόλλου UPnP και υλοποιήθηκε εξ' ολοκλήρου για να καλύψει τις ανάγκες της παρούσας πλατφόρμας. Είναι συμπληρωματική της *UPnPDevice* και προσδιορίζει 3 επιπλέον χαρακτηριστικά μιας συσκευής :

- Κωδικός δωματίου
- Κατηγορία (αισθητήρας, εκκινήτριας)
- Χαρακτηριστικό περιβάλλοντος που αφορά (θερμοκρασία, φωτεινότητα, κίνηση)

Στην κλάση *TemperatureSensorDevice* δεν γίνεται απόδοση τιμής στα διάφορα χαρακτηριστικά της συσκευής, αλλά υπάρχει η υποδομή έτσι ώστε ένα OSGi bundle να δημιουργεί αντικείμενα της συγκεκριμένης κλάσης με τις αντίστοιχες τιμές.

TemperatureSensorService. Η κλάση αυτή υλοποιεί τη διεπαφή *UPnPService* η οποία αφορά την υπηρεσία που προσφέρει η συσκευή και διαθέτει τις μεταβλητές κατάστασης οι οποίες μπορούν να ανακτηθούν και να τροποποιηθούν με τις ενέργειες της υπηρεσίας.

TemperatureStateVariable. Η κλάση αυτή υλοποιεί τη διεπαφή *UPnPLocalStateVariable* η οποία δίνει πρόσβαση στην τρέχουσα τιμή μιας μεταβλητής κατάστασης, στην προκειμένη περίπτωση της θερμοκρασίας.

Αντίστοιχα java packages αναπτύχθηκαν και για τα υπόλοιπα είδη συσκευών. Κατόπιν έγινε εξαγωγή αυτών των packages σε jar αρχεία και έτσι δημιουργήθηκε μια βιβλιοθήκη από υλοποιήσεις 5 τύπων UPnP συσκευών.

5.3 Ανάλυση OSGi bundle

Η πλατφόρμα IESAP αποτελείται από OSGi bundles κάθε ένα το οποίο ανήκει και σε ένα από τα στρώματα της προτεινόμενης αρχιτεκτονικής. Στο περιβάλλον ανάπτυξης του Eclipse, κάθε bundle αποτελεί και ένα project το οποίο συνήθως περιλαμβάνει, μια κλάση *Bundle Activator*, ένα αρχείο *MANIFEST.MF*, ένα *xml* αρχείο περιγραφής υπηρεσιών και όσες κλάσεις *Java* χρειάζονται για την υλοποίηση της συγκεκριμένης υπηρεσίας.

Κλάση Activator. Κάθε bundle συνήθως διαθέτει μια κλάση που υλοποιεί τη διεπαφή *BundleActivator*. Όταν ένα bundle ξεκινά, το πλαίσιο OSGi καλεί την μέθοδο *start()* της κλάσης που υλοποιεί τη διεπαφή *BundleActivator* και εκτελεί τον κώδικα που περιέχει. Ομοίως συμβαίνει όταν σταματά ένα bundle με τη μέθοδο *stop()*.

Αρχείο MANIFEST. Το αρχείο αυτό βρίσκεται κάτω από το φάκελο META-INF και περιέχει τις δηλωτικές επικεφαλίδες που έχουν οριστεί από το πλαίσιο OSGi και τις οποίες χρησιμοποιεί ένας δημιουργός bundle για να παράσχει περιγραφικές πληροφορίες για το bundle.

- *Bundle-ManifestVersion* : Η επικεφαλίδα αυτή προσδιορίζει ποιός προδιαγραφής τους κανόνες ακολουθεί το bundle. Έχει την τιμή 1 για την Έκδοση 3 και την τιμή 2 για την Έκδοση 4.
- *Bundle-Name* : Η επικεφαλίδα αυτή ορίζει ένα αναγνώσιμο όνομα για το bundle.

- *Bundle-SymbolicName* : Η επικεφαλίδα αυτή ορίζει ένα μοναδικό όνομα, το οποίο σε συνδυασμό με την έκδοση του bundle, του επιτρέπουν να προσδιορίζεται μοναδικά μέσα στο πλαίσιο της OSGi.
- *Bundle-Version* : Η επικεφαλίδα αυτή ορίζει την έκδοση του bundle.
- *Bundle-Activator* : Η επικεφαλίδα αυτή προσδιορίζει το όνομα της κλάσης που υλοποιεί τη διεπαφή *BundleActivator* και εκτελείται κατά την εκκίνηση και το σταμάτημα του bundle.
- *Import-Package* : Η επικεφαλίδα αυτή προσδιορίζει τα εισαχθέντα (imported) packages για αυτό το bundle.
- *Exported-Package* : Η επικεφαλίδα αυτή προσδιορίζει τα εξαχθέντα (exported) packages του bundle.
- *Service-Component* : Η επικεφαλίδα αυτή ορίζει το XML αρχείο που περιέχει την περιγραφή του service component (αφορά τις δηλωτικές υπηρεσίες).

XML αρχείο περιγραφής υπηρεσιών. Στην περίπτωση που χρησιμοποιούνται δηλωτικές υπηρεσίες (declaratives services), στο αρχείο αυτό υπάρχει η περιγραφή της υπηρεσίας. Πρόκειται για xml αρχείο το οποίο βρίσκεται κάτω από το φάκελο META-INF και αποτελείται από τα εξής στοιχεία :

- **<component>** : Περιλαμβάνει το χαρακτηριστικό *name* το οποίο προσδιορίζει το όνομα του service component και πρέπει να είναι μοναδικό σε όλο το πλαίσιο.
- **<implementation>** : Περιλαμβάνει το χαρακτηριστικό *class* το οποίο προσδιορίζει την κλάση που υλοποιεί την υπηρεσία.
- **<service>** : Περιλαμβάνει το στοιχείο **<provide>**
- **<provide>** : Περιλαμβάνει το χαρακτηριστικό *interface* το οποίο προσδιορίζει τη διεπαφή της υπηρεσίας.
- **<reference>** : Το στοιχείο αυτό δηλώνει την εξάρτηση που έχει το service component από ένα σύνολο άλλων υπηρεσιών. Αποτελείται από τα χαρακτηριστικά :
 - *name* : Το όνομα του στοιχείου **<reference>**.
 - *interface* : Το όνομα της κλάσης που χρησιμοποιείται από το service component για να αποκτήσει πρόσβαση στην υπηρεσία.

- *bind* : Το όνομα της μεθόδου (της κλάσης υλοποίησης) που χρησιμοποιείται για να γνωστοποιήσει ότι μια υπηρεσία είναι διαθέσιμη.
- *unbind* : Το όνομα της μεθόδου (της κλάσης υλοποίησης) που χρησιμοποιείται για να γνωστοποιήσει ότι μια υπηρεσία δεν είναι πλέον διαθέσιμη.
- *cardinality* : Προσδιορίζει αν η εξάρτηση είναι προαιρετική ή υποχρεωτική και αν είναι μοναδική ή πολλαπλή.
- *policy* : Παίρνει τις τιμές *dynamic* και *static* και αφορά τη δυναμικότητα του service component.

Java κλάσεις. Κάθε bundle μπορεί να διαθέτει όσες κλάσεις ή διεπαφές Java χρειάζονται για την υλοποίηση της υπηρεσίας του.

5.4 Δυναμικές και δηλωτικές υπηρεσίες

Το πλαίσιο OSGi παρέχει τη δυνατότητα στο προγραμματιστή να επιλέξει μεταξύ δύο τρόπων υλοποίησης μιας υπηρεσίας. Ο ένας τρόπος είναι δυναμικός και οι υπηρεσίες που υλοποιούνται ονομάζονται *Dynamic Services*, ενώ ο άλλος είναι στατικός και οι υπηρεσίες ονομάζονται *Declarative Services*.

Dynamic Services. Σε αυτή την περίπτωση, μια υπηρεσία είναι ένα απλό Java αντικείμενο το οποίο δημοσιεύεται στον Κατάλογο Υπηρεσιών της OSGi με το όνομα μιας ή περισσότερων διεπαφών Java. Οι καταναλωτές που επιθυμούν να το χρησιμοποιήσουν μπορούν να το αναζητήσουν χρησιμοποιώντας ένα από τα ονόματα των διεπαφών του. Οι υπηρεσίες μπορούν να καταναλώνουν άλλες υπηρεσίες, χωρίς όμως να είναι στενά συνδεδεμένες μεταξύ τους έχοντας τη μορφή ενός σταθερού γράφου. Αντιθέτως οι υπηρεσίες μπορούν να εγγράφονται και να διαγράφονται δυναμικά οποιαδήποτε ώρα, πράγμα που υποδηλώνει μια προσωρινή συσχέτιση μεταξύ τους.

Το πρώτο βήμα για τη δημιουργία μιας δυναμικής υπηρεσίας είναι η εγγραφή της. Ένα παράδειγμα δήλωσης υπηρεσίας της πλατφόρμας IESAP είναι το ακόλουθο :

```
serviceRegistration =
context.registerService(UPnPDevice.class.getName(), lampDevice, dict);
```


Η δήλωση αυτή εγγράφει το αντικείμενο υπηρεσίας `lampDevice` με τις ιδιότητες `dict` και με το όνομα `UPnPDevice.class.getName()` στο πλαίσιο `OSGi`.

Αντίστοιχα η διαγραφή μιας υπηρεσίας πραγματοποιείται με την ακόλουθη δήλωση :

```
serviceRegistration.unregister();
```

Μια πολύ χρήσιμη δυνατότητα του πλαισίου `OSGi` σε σχέση με τις *Dynamic Services* είναι ο μηχανισμός παρακολούθησης υπηρεσιών, γνωστός ως *ServiceTracker*. Ο μηχανισμός αυτός παρακολουθεί το πλαίσιο και αντιλαμβάνεται νέες εγγραφές, μεταβολές και διαγραφές υπηρεσιών με έναν ενιαίο τρόπο. Ένα παράδειγμα κλάσης *ServiceTracker* της πλατφόρμας `IESAP` είναι αυτό που ακολουθεί.

```
public class UpnpTracker extends ServiceTracker {
    private int deviceCount = 0;
    private boolean registering = false;
    private ServiceRegistration registration = null;
    private final DeviceInstallation deviceInst=
        new DeviceInstallation();

    public UpnpTracker(BundleContext context) {
        super(context, UPnPDevice.class.getName(), null);
    }

    public Object addingService(ServiceReference reference) {
        UPnPDevice finder =
            (UPnPDevice) context.getService(reference);
        deviceInst.bindUPnPDevice(finder);
        synchronized(this) {
            deviceCount++;
            if (registering)
                return finder;
            registering = (deviceCount == 1);
            if (!registering)
                return finder;
        }
        synchronized(this) {
            registering = false;
        }
        return finder;
    }
}
```

```

    }

    public void removedService(ServiceReference reference,
    Object service) {
        UPnPDevice finder = (UPnPDevice) service;
        deviceInst.unbindUPnPDevice(finder);
        context.ungetService(reference);
        ServiceRegistration needsUnregistration = null;
        synchronized(this) {
            deviceCount --;
            if (deviceCount == 0) {
                needsUnregistration = registration;
                registration = null;
            }
        }
        if(needsUnregistration != null) {
            needsUnregistration.unregister();
        }
    }
}

```

Η κλάση αυτή παρακολουθεί το πλαίσιο για εγγραφή ή διαγραφή υπηρεσίας τύπου *UPnPDevice*. Μόλις μια συσκευή ενεργοποιηθεί, εκτελείται η μέθοδος *addingService()* η οποία αναλαμβάνει να ειδοποιήσει την πλατφόρμα IESAP ότι μια νέα συσκευή (δηλ. η αντίστοιχη υπηρεσία της) είναι διαθέσιμη. Αντίστοιχα μόλις μια συσκευή απενεργοποιηθεί εκτελείται η μέθοδος *removedService()* η οποία ενημερώνει την πλατφόρμα ότι η συσκευή (δηλ. η αντίστοιχη υπηρεσία της) δεν είναι πλέον διαθέσιμη.

Declarative Services. Η χρήση δυναμικών υπηρεσιών αν και προσφέρει μεγάλη ευελιξία απαιτεί την ανάπτυξη ενός μεγάλου όγκου πανομοιότυπου κώδικα αυξάνοντας έτσι την πολυπλοκότητα και δυσκολεύοντας την συντήρηση του συστήματος. Μια πιο εύκολη και συντηρήσιμη προσέγγιση είναι η χρήση δηλωτικών υπηρεσιών ή *Declaratives Services*. Σε αυτή την περίπτωση, η δήλωση μιας υπηρεσίας στο πλαίσιο, αλλά και η δήλωση εξάρτησης μιας υπηρεσίας από μια άλλη γίνεται μέσω του *xml* αρχείου περιγραφής υπηρεσιών που περιγράφηκε παραπάνω (κεφάλαιο 5.3). Ένα παράδειγμα δήλωσης υπηρεσίας της πλατφόρμας IESAP μέσω του *xml* αρχείου είναι αυτό που ακολουθεί :

```

<?xml version="1.0"?>
<component name="AdjustmentAction">
    <implementation class="admin.action.module.AdjustmentActionImpl"/>
</service>

```

```
        <provide interface="admin.action.module.AdjustmentAction"/>
    </service>
</component>
```

Σε αυτό το παράδειγμα δηλώνεται ότι η κλάση υλοποίησης της υπηρεσίας είναι η `admin.action.module.AdjustmentActionImpl` και η διεπαφή της υπηρεσίας είναι η `admin.action.module.AdjustmentAction`.

Στο ακόλουθο παράδειγμα γίνεται δήλωση εξάρτησης της υπηρεσίας `admin.action.module.AdjustmentActionImpl` από την `admin.get.services.GetDeviceServices`.

```
<?xml version="1.0"?>
<component name="AdjustmentAction">
    <implementation class="admin.action.module.AdjustmentActionImpl"/>
    <reference name="GetDeviceServices"
        interface="admin.get.services.GetDeviceServices"
        bind="bindGetDeviceServices"
        unbind="unbindGetDeviceServices"
        cardinality="0..n"
        policy="dynamic"/>
    <reference name="DeviceCategory"
        interface="admin.location.table.LocationTable"
        bind="bindLocationTable"
        unbind="unbindLocationTable"
        cardinality="0..n"
        policy="dynamic"/>
</component>
```

5.5 Παρουσίαση των bundles της πλατφόρμας IESAP

Στη συνέχεια γίνεται μια αναλυτική περιγραφή των bundles της πλατφόρμας IESAP. Συγκεκριμένα για κάθε bundle παρουσιάζονται, το όνομα του, το στρώμα της πλατφόρμας στο οποίο ανήκει, το όνομα του eclipse project που εμπεριέχει τα σχετικά αρχεία και γίνεται μια συνοπτική περιγραφή της λογικής που υλοποιούν οι κλάσεις του. Τέλος για τις πιο σύνθετες λειτουργίες υπάρχουν Διαγράμματα Ακολουθίας.

Bundle : *lamp_1*

Στρώμα πλατφόρμας : *Στρώμα Υπηρεσιών Αισθητήρων/Εκκινητήρων*

Eclipse Project : *lamp_1*

Κλάσεις :

Activator. Στην κλάση αυτή γίνεται η απόδοση των τιμών των χαρακτηριστικών της συσκευής και η εγγραφή/διαγραφή της υπηρεσίας που προσφέρει.

Όμοια bundles υπάρχουν για όλες τις συσκευές της πλατφόρμας.

Bundle : *admin_location_table*

Στρώμα πλατφόρμας : *Στρώμα Διαχείρισης Υπηρεσιών*

Eclipse Project : *admin_location_table*

Διεπαφές :

LocationTable. Πρόκειται για τη διεπαφή της υπηρεσίας του πίνακα τοπολογίας.

Κλάσεις :

LocationTableImpl. Η κλάση αυτή υλοποιεί την υπηρεσία που διαχειρίζεται τον πίνακα τοπολογίας συσκευών της πλατφόρμας. Οι πληροφορίες του πίνακα αφορούν κάθε συσκευή που είναι ενεργοποιημένη και μεταξύ άλλων περιλαμβάνουν :

- Τον κωδικό της τοποθεσίας της συσκευής
- Το «φιλικό» όνομα της συσκευής
- Το αναγνωριστικό της συσκευής
- Την κατηγορία της συσκευής
- Το χαρακτηριστικό του περιβάλλοντος που επηρεάζει η συσκευή
- Το αντικείμενο που υλοποιεί την υπηρεσία της συσκευής

Η υπηρεσία διαθέτει μεθόδους για την εισαγωγή, διαγραφή και αναζήτηση συσκευών στον πίνακα τοπολογίας.

Bundle : *device_installation*

Στρώμα πλατφόρμας : *Στρώμα Διαχείρισης Υπηρεσιών*

Eclipse Project : *device_installation*

Κλάσεις :

Activator. Στην κλάση αυτή γίνεται η εκκίνηση του μηχανισμού παρακολούθησης υπηρεσιών *UPnPDevices* μέσω της δημιουργίας ενός αντικειμένου (*UrnpTracker*) τύπου *ServiceTracker*.

UrnpTracker. Στην κλάση αυτή γίνεται η παρακολούθηση του πλαισίου για εγγραφή/διαγραφή υπηρεσιών *UPnPDevices* και η ενημέρωση της πλατφόρμας μέσω του αντικειμένου *DeviceInstallation*.

DeviceInstallation. Η κλάση αυτή αναλαμβάνει την εγκατάσταση/απεγκατάσταση μιας συσκευής (δηλ. της υπηρεσίας που προσφέρει) από την πλατφόρμα. Χρησιμοποιεί τη δυναμική υπηρεσία *UPnPDevice* μέσω του tracker *UrnpTracker* ο οποίος αναλαμβάνει να την ενημερώσει για την ενεργοποίηση ή την απενεργοποίηση μιας συσκευής. Κατόπιν και σε κάθε περίπτωση γίνεται κλήση της υπηρεσίας *LocationTable* για την σχετική ενημέρωση (εισαγωγή, διαγραφή) του πίνακα τοπολογίας.

Bundle : <i>admin_get_services</i>

Στρώμα πλατφόρμας : *Στρώμα Αντίληψης*

Eclipse Project : *admin_get_services*

Διεπαφές :

GetDeviceServices. Πρόκειται για τη διεπαφή της υπηρεσία ανάκτησης υπηρεσιών συσκευών.

Κλάσεις :

GetDeviceServicesImpl. Η κλάση αυτή υλοποιεί την υπηρεσία που ανακτά όλες τις υπηρεσίες συσκευών που σχετίζονται με μια ενέργεια (πρόκειται για τη σχέση *ενέργειας* και *υπηρεσίας* που περιγράφηκε στο κεφάλαιο 4.3.1). Η ανάκτηση των υπηρεσιών γίνεται από το προκαθορισμένο xml αρχείο *services.xml* το οποίο για κάθε υπηρεσία συσκευής ορίζει :

- Την ενέργεια που υλοποιεί (<abstractService>)
- Το Id της UPnP υπηρεσίας (<deviceService>)
- Την ενέργεια της UPnP υπηρεσίας (<deviceAction>)
- Τη μεταβλητή κατάστασης (<resultVariable>)

Ένα παράδειγμα δήλωσης μιας υπηρεσίας συσκευής είναι το ακόλουθο :

```
<serviceDictionary>
  <services>
    <abstractService>A_GetLight</abstractService>
    <deviceService>urn:schemas-upnp-org:serviceId:light
    sensor:1</deviceService>
    <deviceAction>GetLight</deviceAction>
    <resultVariable>CurrentLight</resultVariable>
  </services>
</serviceDictionary>
```

Bundle : *admin_decision_module*

Στρώμα πλατφόρμας : *Στρώμα Αντίληψης*

Eclipse Project : *admin_decision_module*

Διεπαφές :

AdjustmentDecision. Πρόκειται για τη διεπαφή της υπηρεσίας απόφασης.

Κλάσεις :

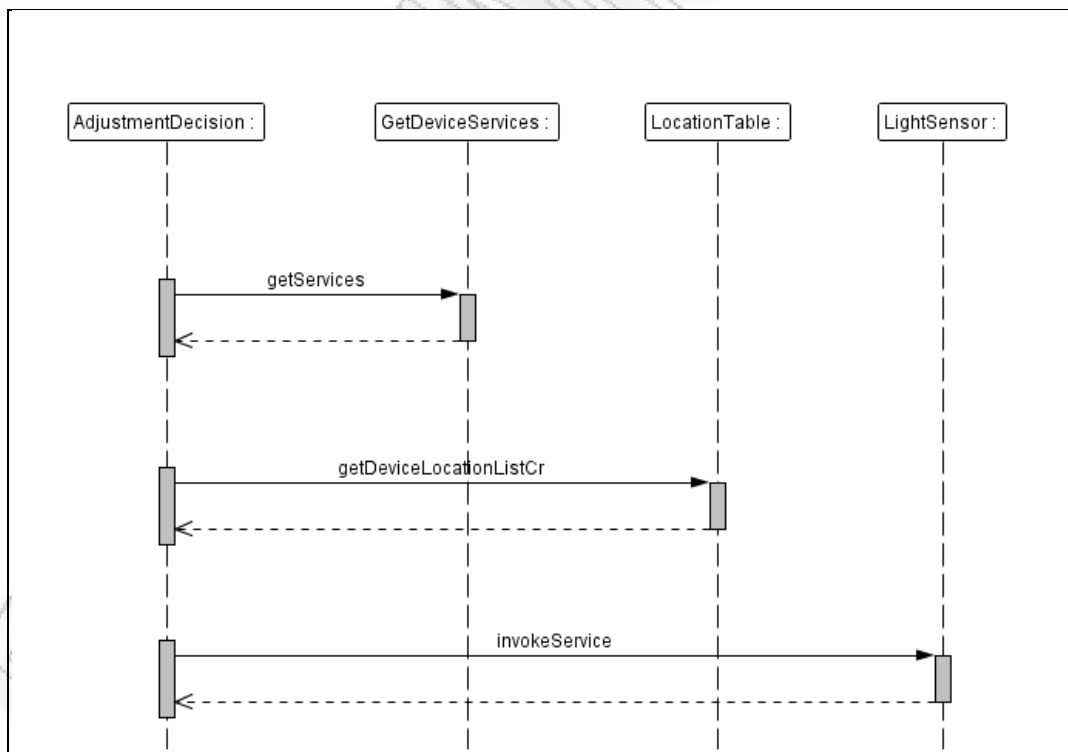
AdjustmentDecisionImpl. Η κλάση αυτή υλοποιεί την υπηρεσία που αποφασίζει αν η τρέχουσα τιμή ενός χαρακτηριστικού του περιβάλλοντος για ένα συγκεκριμένο δωμάτιο είναι μεγαλύτερη, μικρότερη ή ίση με τη ζητούμενη. Για να το επιτύχει αυτό ανακτά την τρέχουσα τιμή του χαρακτηριστικού, τη συγκρίνει με την επιθυμητή και ανάλογα αποφασίζει. Η ανάκτηση της τρέχουσας τιμής του χαρακτηριστικού γίνεται ως εξής :

1. Η υπηρεσία *AdjustmentDecision* ζητά από την υπηρεσία *GetDeviceServices* να τις επιστρέψει όλες τις υπηρεσίες των συσκευών που έχουν να κάνουν με την ενέργεια ανάκτησης τιμής για το συγκεκριμένο χαρακτηριστικό. Για

παράδειγμα, αν το χαρακτηριστικό είναι η θερμοκρασία, η υπηρεσία *AdjustmentDecision* ζητά από την *GetDeviceServices* όλες τις υπηρεσίες συσκευών που σχετίζονται με την ενέργεια *A_GetTemperature*.

2. Η υπηρεσία *AdjustmentDecision* ζητά από την υπηρεσία *LocationTable* να της επιστρέψει όλες τις συσκευές κατηγορίας «αισθητήρα» που βρίσκονται στο συγκεκριμένο δωμάτιο και αφορούν το συγκεκριμένο χαρακτηριστικό.
3. Για κάθε μια συσκευή του δωματίου γίνεται έλεγχος αν η υπηρεσίας της είναι μια από τις υπηρεσίες της ενέργειας ανάκτησης τιμής του χαρακτηριστικού. Αν είναι τότε γίνεται κλήση της υπηρεσίας και ανακτάται η τρέχουσα ένδειξη του συγκεκριμένου αισθητήρα.
4. Λαμβάνονται υπόψη όλες οι ενδείξεις των αισθητήρων του συγκεκριμένου δωματίου και προκύπτει μια μέση τιμή για το συγκεκριμένο χαρακτηριστικό.

Διάγραμμα ακολουθίας :



Εικόνα 16. Διάγραμμα Ακολουθίας λειτουργίας *Απόφαση*

Bundle : *admin_action_module*

Στρώμα πλατφόρμας : *Στρώμα Αντίληψης*

Eclipse Project : *admin_action_module*

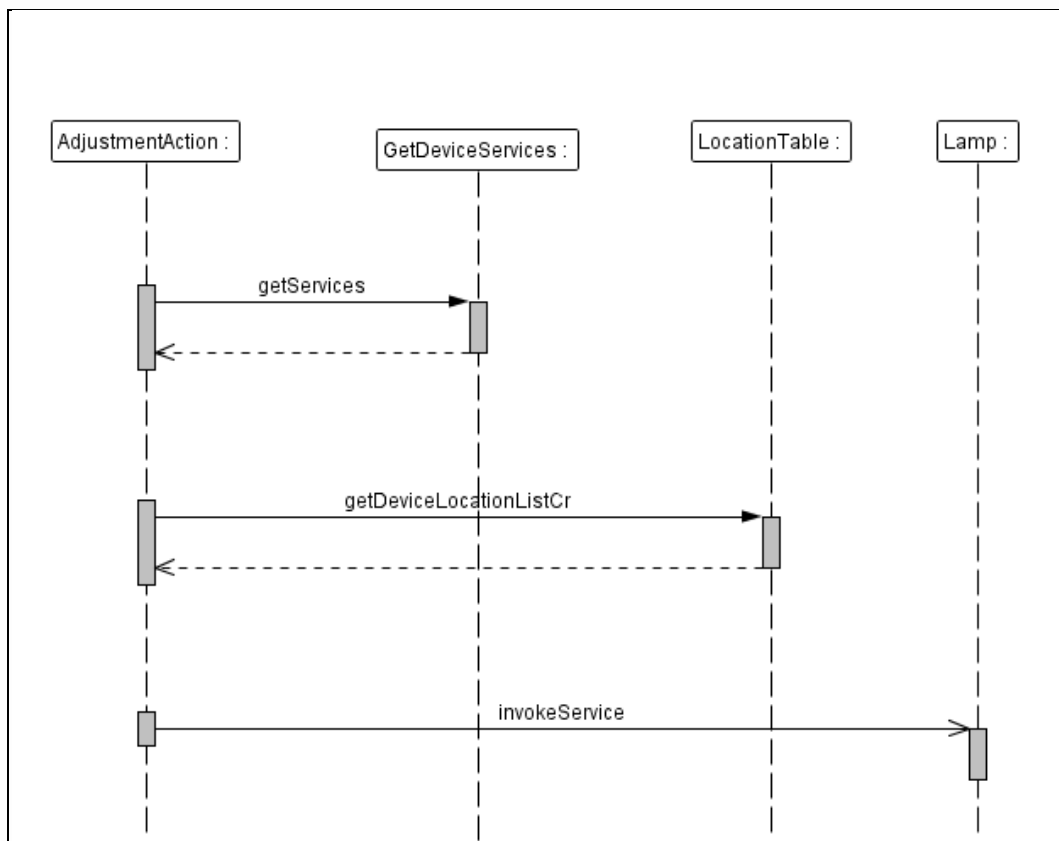
Διεπαφές :

AdjustmentAction. Πρόκειται για τη διεπαφή της υπηρεσίας ενεργοποίησης εκκινήτων.

Κλάσεις :

AdjustmentActionImpl. Η κλάση αυτή υλοποιεί την υπηρεσία που αποφασίζει για τις ενέργειες των εκκινήτων της πλατφόρμας, αναλόγως της ένδειξης της υπηρεσίας *AdjustmentDecision*. Για παράδειγμα, αν η ένδειξη της υπηρεσίας *AdjustmentDecision* είναι ότι η θερμοκρασία στο υπνοδωμάτιο είναι μεγαλύτερη της ζητούμενης, τότε δίνεται εντολή στον αντίστοιχο εκκινήτηρα ή εκκινήτηρες να μειώσουν τη θερμοκρασία. Η ανάκτηση των εκκινήτων και των υπηρεσιών για συγκεκριμένο δωμάτιο και χαρακτηριστικό, γίνεται με τον ίδιο τρόπο όπως και στην περίπτωση της *AdjustmentDecision*. Επιπλέον η υπηρεσία αυτή δέχεται ως όρισμα το βήμα με το οποίο θα δώσει να εντολή στους εκκινήτηρες να μεταβάλουν (αύξηση ή μείωση) την τιμή του αντίστοιχου χαρακτηριστικού.

Διάγραμμα ακολουθίας :



Εικόνα 17. Διάγραμμα Ακολουθίας λειτουργίας *Ενεργοποίηση Εκκινήτηρων*

Bundle : *admin_service_adjustLight*

Στρώμα πλατφόρμας : *Στρώμα Διαχείρισης Υπηρεσιών*

Eclipse Project : *admin_service_adjustLight*

Διεπαφές :

LightAdjustment. Πρόκειται για τη διεπαφή της υπηρεσίας ρύθμισης φωτεινότητας.

Κλάσεις :

LightAdjustmentImpl. Η κλάση αυτή υλοποιεί την υπηρεσία που αναλαμβάνει να διατηρήσει τη φωτεινότητα ενός δωματίου στην επιθυμητή από το χρήστη τιμή. Αρχικά καλεί την υπηρεσία *AdjustmentDecision* η οποία της επιστρέφει μια ένδειξη που προσδιορίζει αν πρέπει να αυξηθεί, να μειωθεί ή να παραμείνει ίδια η

ένταση της φωτεινότητας του δωματίου. Με βάση αυτή την ένδειξη, γίνεται κλήση της υπηρεσίας *AdjustmentAction* η οποία αναλαμβάνει να αυξήσει ή να μειώσει την ένταση της φωτεινότητας.

Bundle : *admin_service_adjustTemp*

Στρώμα πλατφόρμας : *Στρώμα Διαχείρισης Υπηρεσιών*

Eclipse Project : *admin_service_adjustTemp*

Διεπαφές :

TemperatureAdjustment. Πρόκειται για τη διεπαφή της υπηρεσίας ρύθμισης θερμοκρασίας.

Κλάσεις :

TemperatureAdjustmentImpl. Η κλάση αυτή υλοποιεί την υπηρεσία που αναλαμβάνει να διατηρήσει τη θερμοκρασία ενός δωματίου στην επιθυμητή από το χρήστη τιμή. Αρχικά καλεί την υπηρεσία *AdjustmentDecision* η οποία της επιστρέφει μια ένδειξη που προσδιορίζει αν πρέπει να αυξηθεί, να μειωθεί ή να παραμείνει ίδια η θερμοκρασία του δωματίου. Με βάση αυτή την ένδειξη, γίνεται κλήση της υπηρεσίας *AdjustmentAction* η οποία αναλαμβάνει να αυξήσει ή να μειώσει τη θερμοκρασία.

Bundle : *admin_service_setLightMinMax*

Στρώμα πλατφόρμας : *Στρώμα Διαχείρισης Υπηρεσιών*

Eclipse Project : *admin_service_setLightMinMax*

Διεπαφές :

SetLightMinMax. Πρόκειται για τη διεπαφή της υπηρεσίας ρύθμισης φωτεινότητας στη μέγιστη και ελάχιστη τιμή.

Κλάσεις :

SetLightMinMaxImpl. Η κλάση αυτή υλοποιεί την υπηρεσία που αναλαμβάνει να ρυθμίσει τη φωτεινότητα ενός δωματίου στην ελάχιστη ή στη μέγιστη τιμή της. Αρχικά καλεί την υπηρεσία *AdjustmentDecision* η οποία της επιστρέφει μια ένδειξη που προσδιορίζει αν η τρέχουσα τιμή της φωτεινότητας είναι διαφορετική από την ελάχιστη ή τη μέγιστη (αναλόγως την κλήση που έχει γίνει). Αν η τρέχουσα τιμή είναι διαφορετική, τότε γίνεται κλήση της υπηρεσίας *AdjustmentAction* με βήμα τέτοιο ώστε η ένταση της φωτεινότητας να πάρει αμέσως την ελάχιστη ή τη μέγιστη τιμή της. Για τον υπολογισμό του βήματος σε αυτήν την περίπτωση γίνεται ανάκτηση του πλήθους των σχετικών με τη φωτεινότητα εκκινήτων του δωματίου με κλήση της υπηρεσίας *LocationTable*, με σκοπό να υπολογιστεί πόσο πρέπει να είναι το βήμα του κάθε εκκινήτηρα, ώστε να προκύψει άμεσα η ελάχιστη ή η μέγιστη τιμή.

Bundle : <i>admin_service_adjustLightMotion</i>
--

Στρώμα πλατφόρμας : *Στρώμα Διαχείρισης Υπηρεσιών*

Eclipse Project : *admin_service_adjustLightMotion*

Διεπαφές :

LightMotionAdjustment. Πρόκειται για τη διεπαφή της υπηρεσίας ενεργοποίησης φωτισμού με την ανίχνευση κίνησης.

Κλάσεις :

LightMotionAdjustmentImpl. Η κλάση αυτή υλοποιεί την υπηρεσία που αναλαμβάνει να αυξήσει τη φωτεινότητα ενός δωματίου στο μέγιστο, μόλις ανιχνευθεί κίνηση μέσα σε αυτό. Αρχικά καλείται η υπηρεσία *AdjustmentDecision* η οποία επιστρέφει μια ένδειξη για το αν υπάρχει ή όχι κίνηση μέσα στο δωμάτιο. Αν υπάρχει κίνηση, γίνεται κλήση της υπηρεσίας *SetLightMinMax* η οποία θέτει τη φωτεινότητα του δωματίου στη μέγιστη τιμή της.

Bundle : *system_administration*

Στρώμα πλατφόρμας : *Στρώμα Εφαρμογής*

Eclipse Project : *system_administration*

Το bundle αυτό υλοποιεί την *Εφαρμογή Διαχείρισης της Πλατφόρμας IESAP* (PMA – Platform Management Application) και αποτελείται από 3 υποεφαρμογές, τη *Διεπαφή Λειτουργιών*, τη *Λίστα Συσκευών* και την *Κονσόλα Μηνυμάτων*.

Η *Διεπαφή Λειτουργιών* αποτελείται από ένα γραφικό παραθυρικό περιβάλλον, το οποίο δίνει τη δυνατότητα στο χρήστη να διαχειριστεί τις λειτουργίες της πλατφόρμας IESAP. Πιο συγκεκριμένα :

- *Ρύθμιση Θερμοκρασίας – Φωτεινότητας.* Ο χρήστης επιλέγει το δωμάτιο, το χαρακτηριστικό (Θερμοκρασία, Φωτεινότητα) και την τιμή του χαρακτηριστικού που επιθυμεί και θέτει σε εκκίνηση τη λειτουργία. Επιπλέον έχει τη δυνατότητα να επιλέξει το χρόνο που μεσολαβεί μεταξύ των κλήσεων των υπηρεσιών των αισθητήρων για την ανάκτηση της τρέχουσας τιμής του χαρακτηριστικού, αλλά και να ορίσει το βήμα με το οποίο οι εκκινήσεις θα επηρεάζουν το χαρακτηριστικό. Τέλος, μπορεί να επιλέξει η μετάβαση στην επιθυμητή τιμή να γίνεται απ' ευθείας με ένα βήμα.
- *Ενεργοποίηση Φωτισμού με Ανίχνευση Κίνησης.* Ο χρήστης επιλέγει το δωμάτιο που επιθυμεί και θέτει σε εκκίνηση τη λειτουργία. Επιπλέον έχει τη δυνατότητα να επιλέξει το χρόνο που μεσολαβεί μεταξύ των κλήσεων των υπηρεσιών των αισθητήρων για την ανάκτηση της τρέχουσας τιμής του χαρακτηριστικού της κίνησης.

Η *Λίστα Συσκευών* παρουσιάζει τις UPnP συσκευές που είναι συνδεδεμένες τη δεδομένη χρονική στιγμή στην πλατφόρμα IESAP, καθώς και μερικά από τα χαρακτηριστικά τους.

Η *Κονσόλα Μηνυμάτων* λειτουργεί σαν αρχείο καταγραφής (log file) και παρουσιάζει τα μηνύματα της εφαρμογής.

Κλάσεις :

Panels.Panel1. Υλοποιεί τη διεπαφή χρήστη των λειτουργιών *Ρύθμιση Θερμοκρασίας – Φωτεινότητας*, η οποία εμπεριέχει τα κατάλληλα στοιχεία ελέγχου. Επιπλέον δημιουργεί αντικείμενο της κλάσης *Program1*.

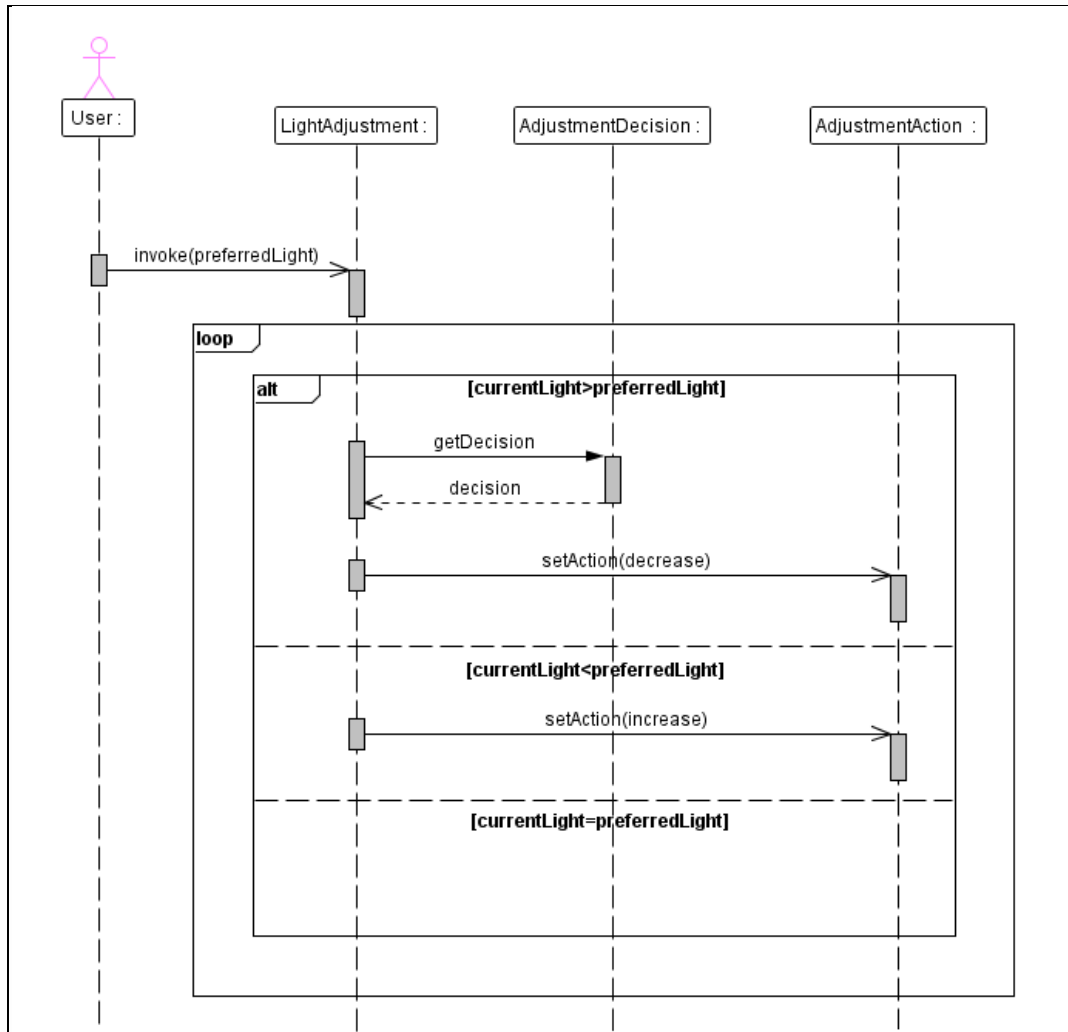
Panels.Panel2. Υλοποιεί τη διεπαφή χρήστη της λίστας συσκευών. Επιπλέον δημιουργεί αντικείμενο της κλάσης *Program2*.

Panels.Panel3. Υλοποιεί τη διεπαφή χρήστη της λειτουργίας *Ενεργοποίηση Φωτισμού με Ανίχνευση Κίνησης*. Επιπλέον δημιουργεί αντικείμενο της κλάσης *Program3*.

Panels.Panel4. Υλοποιεί τη διεπαφή χρήστη της κονσόλας μηνυμάτων.

Programs.Program1. Η κλάση αυτή υλοποιεί τις λειτουργίες *Ρύθμιση Θερμοκρασίας – Φωτεινότητας*. Χρησιμοποιεί τις δυναμικές υπηρεσίες *TemperatureAdjustment* και *LightAdjustment*, μέσω των αντίστοιχων trackers (*TemperatureAdjustmentTracker* και *LightAdjustmentTracker*), τις οποίες καλεί συνεχώς μέσα σε έναν ατέρμονα βρόγχο (endless loop), με καθυστέρηση μεταξύ των κλήσεων αυτή που έχει επιλέξει ο χρήστης.

Διάγραμμα Ακολουθίας :

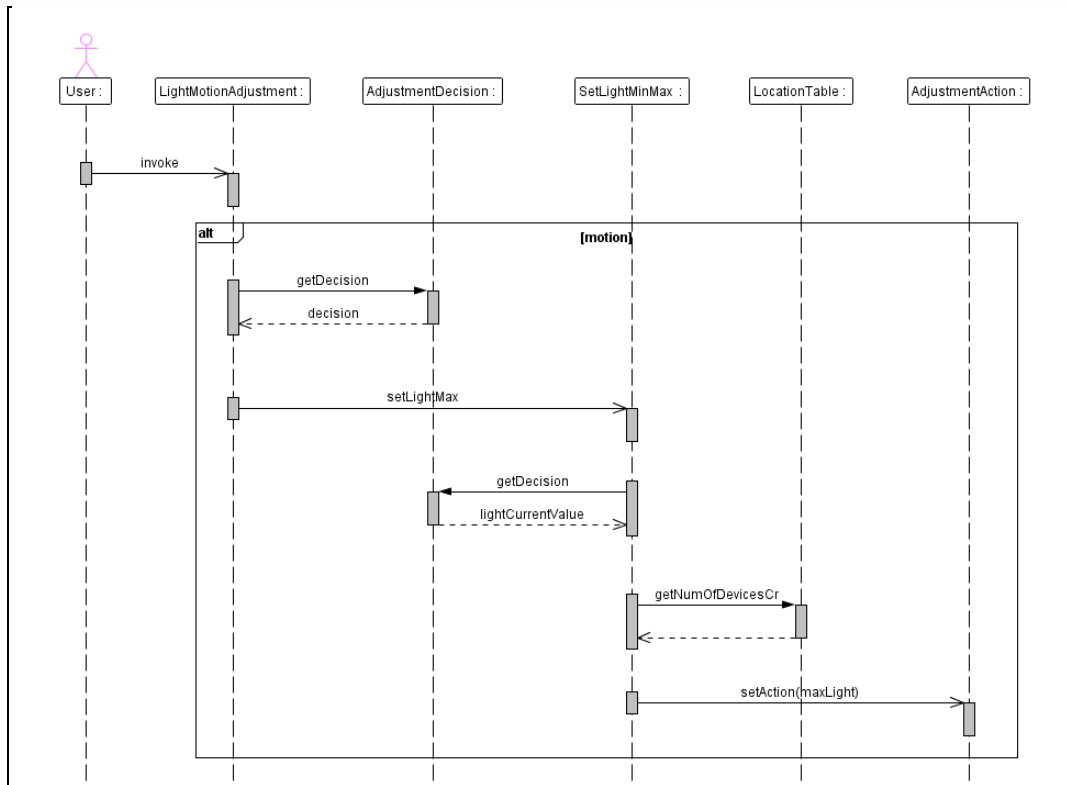


Εικόνα 18. Διάγραμμα Ακολουθίας λειτουργίας Ρύθμιση Φωτεινότητας

Programs.Program2. Η κλάση αυτή υλοποιεί τη λειτουργία προβολής των UPnP συσκευών. Χρησιμοποιεί τη δυναμική υπηρεσία *UPnPDevice* μέσω του tracker *UrpTracker* ο οποίος αναλαμβάνει να την ενημερώσει για την ενεργοποίηση ή απενεργοποίηση μιας συσκευής και κατόπιν ενημερώνεται η σχετική λίστα συσκευών.

Programs.Program3. Η κλάση αυτή υλοποιεί τη λειτουργία *Ενεργοποίηση Φωτισμού με Ανίχνευση Κίνησης*. Χρησιμοποιεί τη δυναμική υπηρεσία *LightMotionAdjustment*, μέσω του tracker *LightMotionAdjustmentTracker*, την οποία καλεί συνεχώς μέσα σε έναν ατέρμονα βρόγχο (endless loop), με καθυστέρηση μεταξύ των κλήσεων αυτή που έχει επιλέξει ο χρήστης.

Διάγραμμα Ακολουθίας :



Εικόνα 19. Διάγραμμα Ακολουθίας λειτουργίας *Ενεργοποίηση Φωτισμού με την Ανίχνευση Κίνησης*

6 ΠΕΡΙΓΡΑΦΗ ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑΣ

6.1 Εγκατάσταση και εκτέλεση πλατφόρμας

Ο πηγαίος κώδικας της πλατφόρμας βρίσκεται στο workspace του Eclipse χωρισμένος σε folders ανά bundle. Η εκτέλεση της πλατφόρμας στο περιβάλλον του Eclipse γίνεται με την επιλογή *Run*.

Υπάρχει όμως η δυνατότητα να γίνει και stand-alone εκτέλεση της εφαρμογής. Σε αυτή την περίπτωση πρέπει να έχει προηγηθεί η εξής προεργασία :

1. Δημιουργούμε ένα folder π.χ. IESAP
2. Στο folder IESAP δημιουργούμε το folder plugins
3. Στο folder plugins δημιουργούμε το folder configuration
4. Στο folder configuration δημιουργούμε το αρχείο config.ini το οποίο περιέχει τις εξής δηλώσεις :

```
osgi.bundles=  
org.eclipse.equinox.ds_1.0.0.v20060828  
@start,org.eclipse.equinox.common_3.3.0.v20070426  
@start,org.eclipse.core.jobs_3.3.1.R33x_v20070709@start,org.eclipse.core.  
contenttype_3.2.100.v20070319  
@start,org.eclipse.equinox.app_1.0.1.R33x_v20070828  
@start,org.eclipse.osgi.services_3.1.200.v20070605  
@start,org.eclipse.equinox.registry_3.3.1.R33x_v20070802@start,org.eclips  
e.equinox.preferences_3.2.101.R33x_v20080117  
@start,javax.servlet_2.4.0.v200706111738  
@start,classes_1.0.0@start,admin_location_table_1.0.0  
@start,... (δήλωση όλων των bundles της πλατφόρμας)  
  
eclipse.ignoreApp=false  
org.osgi.framework.bootdelegation=*
```

5. Στο folder plugins αντιγράφουμε τα εξής jars :

- org.eclipse.osgi_3.4.0.v20080605-1900.jar (υλοποίηση του πλαισίου OSGi)
- org.eclipse.osgi.services_3.1.200.v20070605.jar
- org.eclipse.equinox.registry_3.3.1.R33x_v20070802.jar
- org.eclipse.equinox.preferences_3.2.101.R33x_v20080117.jar
- org.eclipse.equinox.ds_1.0.0.v20060828.jar
- org.eclipse.equinox.common_3.3.0.v20070426.jar

- org.eclipse.equinox.app_1.0.1.R33x_v20070828.jar
- org.eclipse.core.jobs_3.3.1.R33x_v20070709.jar
- org.eclipse.core.contenttype_3.2.100.v20070319.jar
- javax.servlet.jsp_2.0.0.v200706191603.jar

6. Στο folder IESAP δημιουργούμε ένα bat αρχείο π.χ. IESAP.bat για την εκκίνηση του πλαισίου OSGi με την εξής δήλωση :

```
java -jar plugins/org.eclipse.osgi_3.4.0.v20080605-1900.jar -console
```

7. Μέσα από το Eclipse κάνουμε export όλων των bundles στο folder plugins.

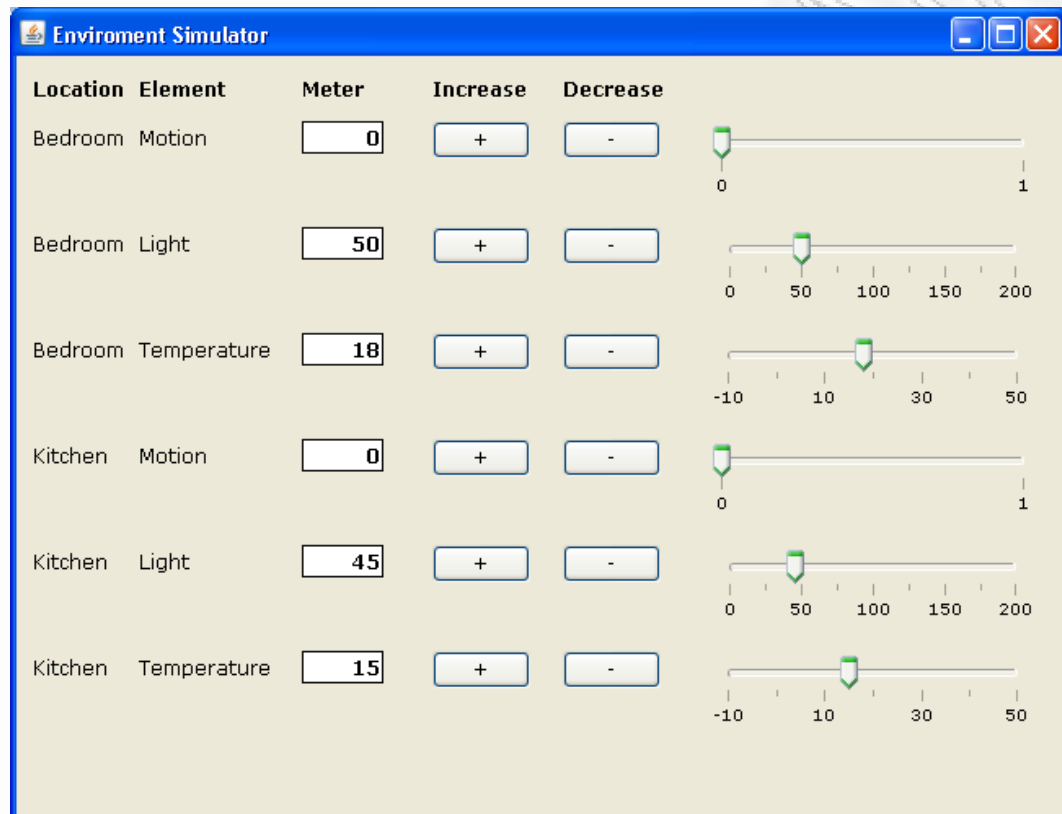
Αυτό γίνεται ως εξής :

- Από το menu επιλέγουμε *File > Export > Deployment plug-ins and fragments*
- Επιλέγουμε τα bundles της πλατφόρμας OSGi
- Ορίζουμε το path του folder plugins στην επιλογή *Directory*
- Επιλέγουμε *Finish*

8. Για να εκκινήσουμε την πλατφόρμα (και την εφαρμογή PMA) αρκεί να εκτελέσουμε το αρχείο IESAP.bat

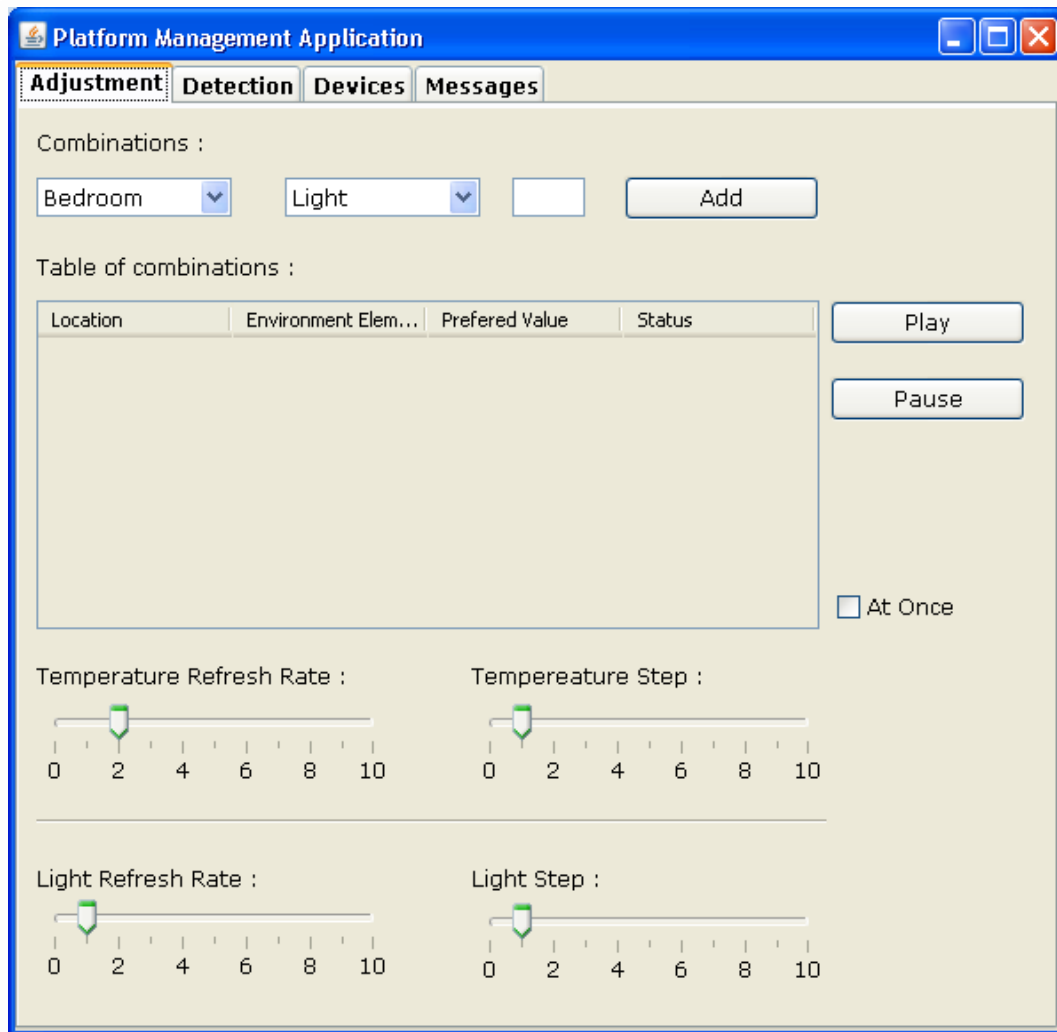
6.2 Εκκίνηση Προσομοιωτή Περιβάλλοντος και Εφαρμογής PMA

Πρώτα πρέπει να γίνει η εκκίνηση του *Προσομοιωτή Περιβάλλοντος* (εκτέλεση αρχείου EnvironmentSimulator.jar) και μετά της *Εφαρμογής PMA*.



Εικόνα 20. Προσομοιωτής Περιβάλλοντος

Ο *Προσομοιωτής Περιβάλλοντος* (Εικόνα 20) παρουσιάζει για κάθε δωμάτιο, την τρέχουσα ένδειξη κάθε χαρακτηριστικού του περιβάλλοντος του. Ο χρήστης μπορεί οποιαδήποτε στιγμή να μεταβάλει αυτή την ένδειξη, είτε με τα κουμπιά «+» και «-», είτε με τον αντίστοιχο slider.



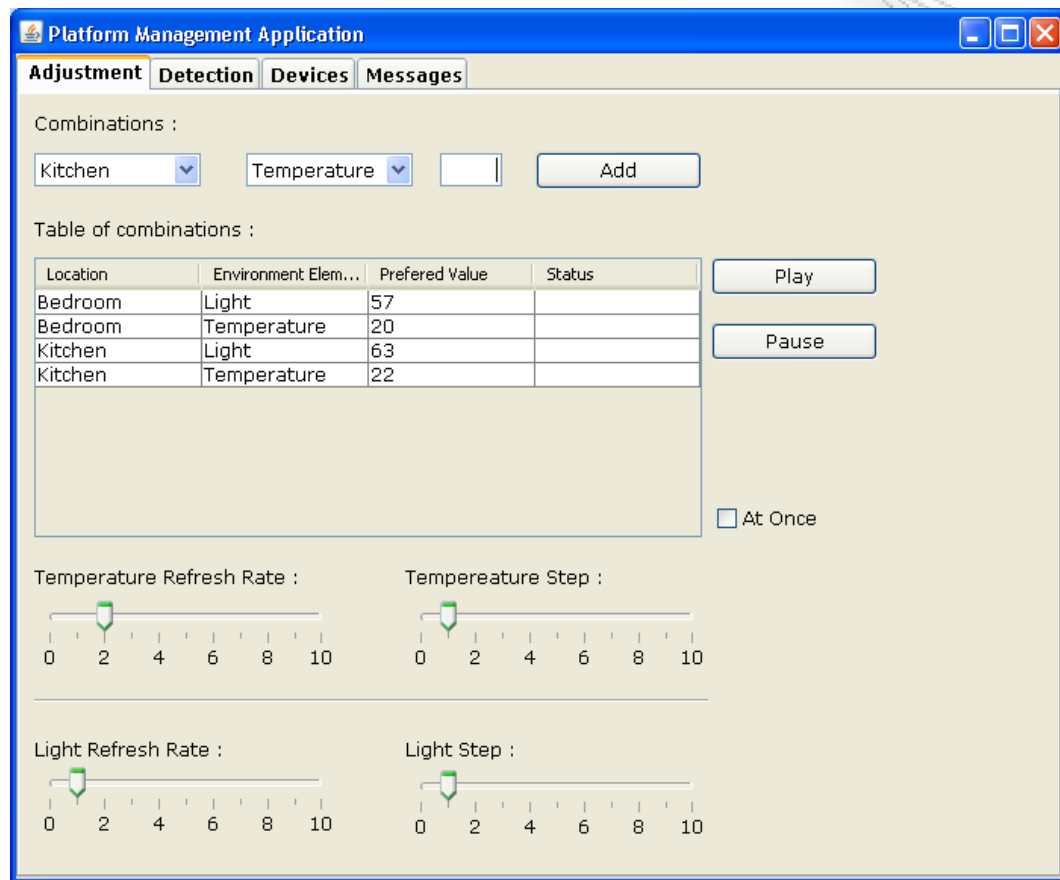
Εικόνα 21. Η Εφαρμογή PMA

Με την εκκίνηση της, η εφαρμογή PMA έχει την μορφή που παρουσιάζεται στην Εικόνα 21. Η πλοήγηση στις λειτουργίες της εφαρμογής, γίνεται μέσω των τεσσάρων tabs.

6.3 Λειτουργία Ρύθμισης Θερμοκρασίας και Φωτεινότητας

Για την εκτέλεση αυτής της λειτουργίας ο χρήστης επιλέγει το tab *Adjustment*. Κατόπιν επιλέγει το δωμάτιο που επιθυμεί να εφαρμόσει τη λειτουργία και το χαρακτηριστικό του περιβάλλοντος (θερμοκρασία, φωτεινότητα), μέσω των σχετικών comboboxs και καταχωρεί την επιθυμητή τιμή. Στη συνέχεια επιλέγει το πλήκτρο «Add» και η ρύθμιση που δημιούργησε προστίθεται στο σχετικό πίνακα. Μπορεί να κάνει και άλλους συνδυασμούς για δωμάτια, χαρακτηριστικά και

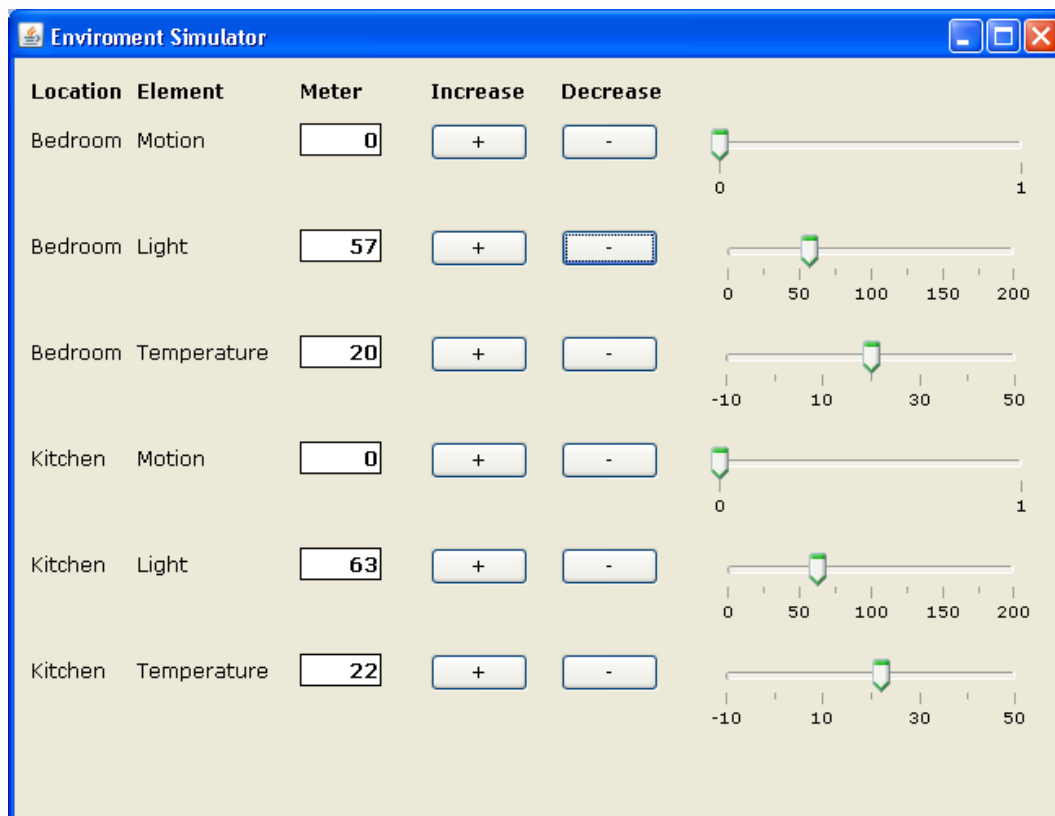
επιθυμητές τιμές. Μετά από διάφορες επιλογές, ο πίνακας συνδυασμών μπορεί να μοιάζει με αυτόν της Εικόνα 22.



Εικόνα 22. Εφαρμογή PMA - Ρύθμιση Θερμοκρασίας και Φωτεινότητας

Ο χρήστης μπορεί ακόμα να προσδιορίσει το ρυθμό ανάκτησης τιμής από τους αισθητήρες για κάθε χαρακτηριστικό, μέσω των σχετικών sliders. Επίσης μπορεί να προσδιορίσει το βήμα με το οποίο οι εκκινητήρες θα μεταβάλουν το χαρακτηριστικό χρησιμοποιώντας τους αντίστοιχους sliders. Τέλος υπάρχει η δυνατότητα για μεταβολή του χαρακτηριστικού με ένα μόνο βήμα, τσεκάροντας το σχετικό checkbox.

Αφού δημιουργηθούν οι επιθυμητοί συνδυασμοί, ο χρήστης μπορεί να εκκινήσει τις λειτουργίες, επιλέγοντας από τον πίνακα τη γραμμή που επιθυμεί και κατόπιν το πλήκτρο «Play». Μια λειτουργία μπορεί να διακοπεί με το πλήκτρο «Pause», αφού πρώτα έχει επιλεγθεί από τον πίνακα.



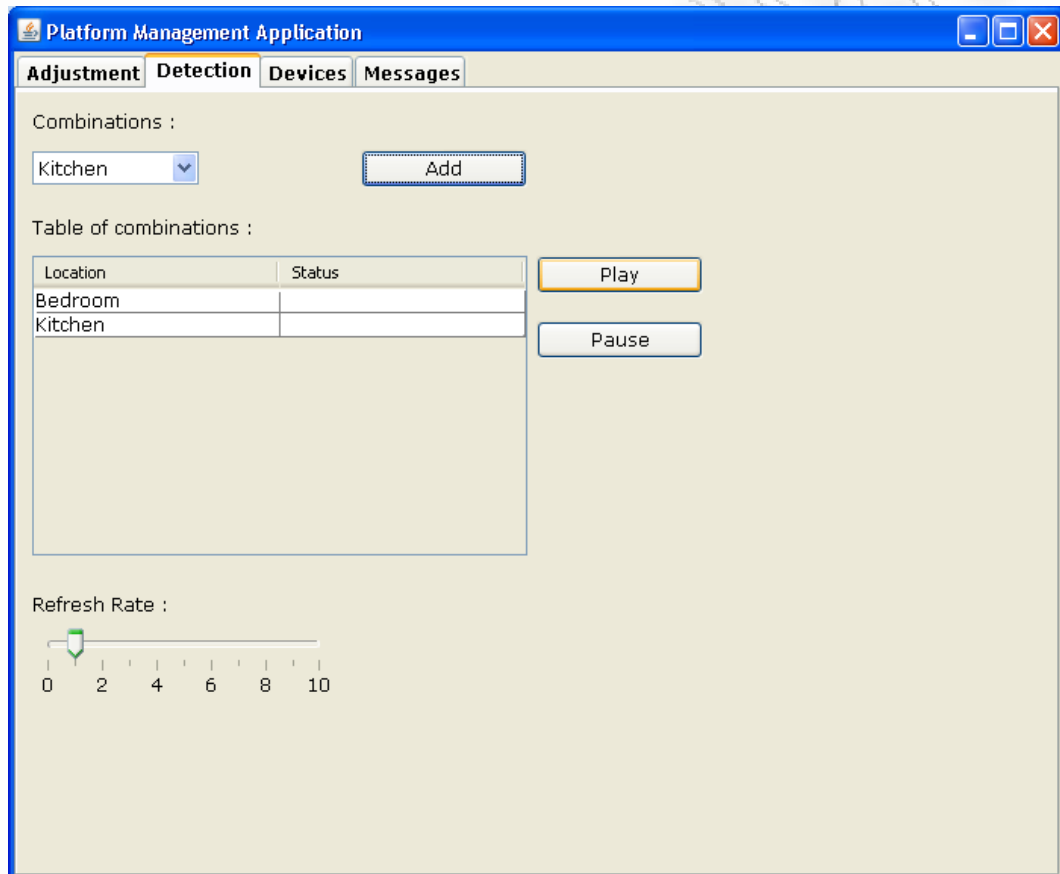
Εικόνα 23. Ο Προσομοιωτής Περιβάλλοντος μετά την εκτέλεση του σεναρίου

Μετά την εκτέλεση του παραπάνω σεναρίου, οι ενδείξεις του *Εξομοιωτή Περιβάλλοντος* θα είναι αυτές της Εικόνα 23.

Στη διάρκεια εκτέλεσης της λειτουργίας μπορεί να γίνει απενεργοποίηση μιας συσκευής χωρίς αυτό να επηρεάσει το σύστημα. Αν για παράδειγμα απενεργοποιήσουμε το κλιματιστικό (*AirCondition 2*) που βρίσκεται στο υπνοδωμάτιο, τότε η λειτουργία της ρύθμισης θερμοκρασίας θα εξακολουθήσει να εκτελείται αλλά η τιμή της θερμοκρασίας θα παραμένει σταθερή. Μόλις ενεργοποιηθεί η συσκευή, η λειτουργία θα αρχίσει και πάλι να εκτελείται και θα προσπαθήσει να ρυθμίσει τη θερμοκρασία μέσω της συσκευής στην επιθυμητή τιμή.

6.4 Λειτουργία Ενεργοποίησης Φωτεινότητας με την Ανίχνευση Κίνησης

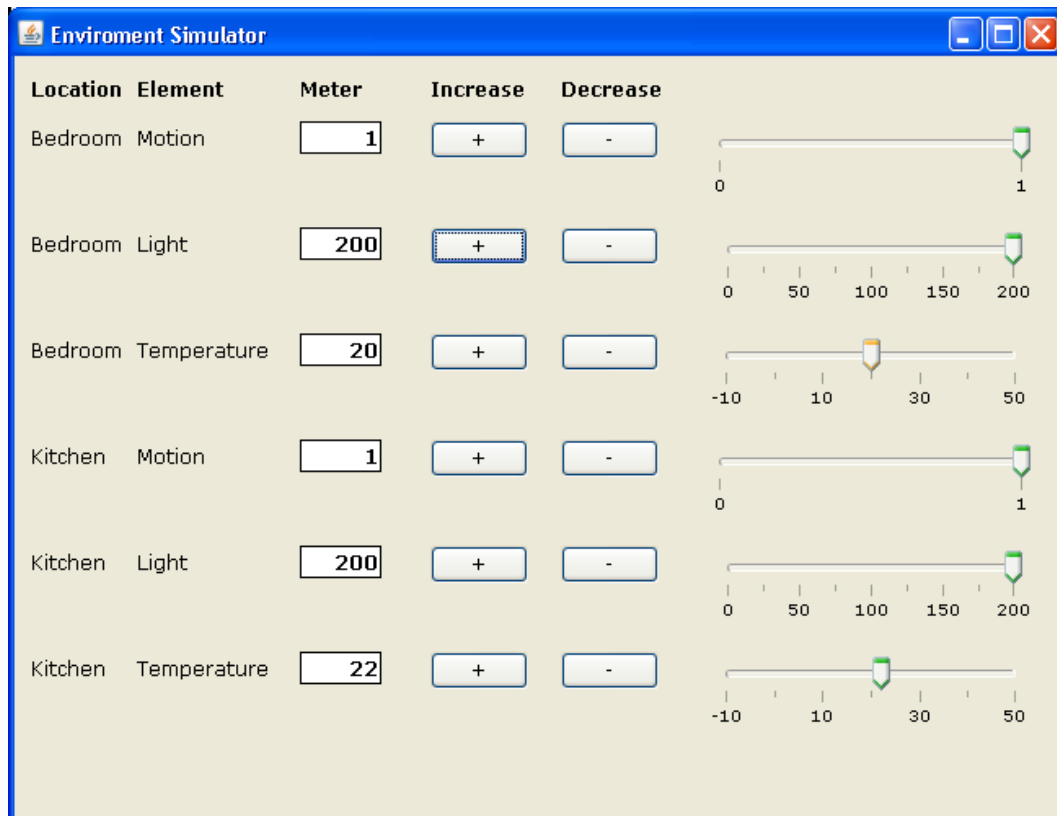
Η μετάβαση σε αυτή τη λειτουργία γίνεται μέσω του tab *Detection*. Με παρόμοιες ενέργειες ο χρήστης επιλέγει τα δωμάτια που επιθυμεί να εφαρμόσει τη λειτουργία και δημιουργεί το σχετικό πίνακα έτσι όπως φαίνεται στην Εικόνα 24.



Εικόνα 24. Εφαρμογή PMA – Ενεργοποίηση Φωτεινότητας με Ανίχνευση Κίνησης

Υπάρχει και εδώ η δυνατότητα ρύθμισης του ρυθμού ανάκτησης τιμής από τους αισθητήρες, μέσω του αντίστοιχου slider.

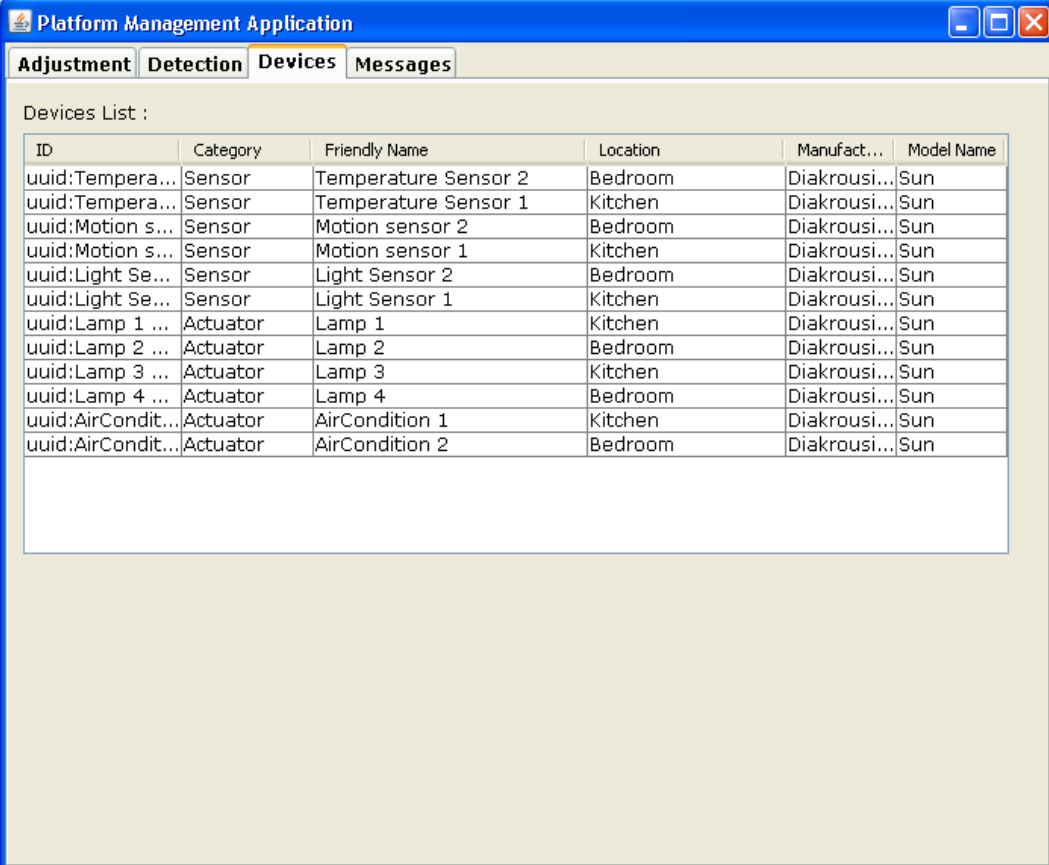
Μετά την εκτέλεση του παραπάνω σεναρίου, οι ενδείξεις του *Εξομοιωτή Περιβάλλοντος* θα είναι αυτές της Εικόνα 25.



Εικόνα 25. Ο Προσομοιωτής Περιβάλλοντος μετά την εκτέλεση του σεναρίου

6.5 Λίστα Συσκευών Συστήματος

Η μετάβαση σε αυτή τη λειτουργία γίνεται μέσω του tab *Devices*. Στο χρήστη προβάλλεται ένας πίνακας με πληροφορίες για όλες τις συσκευές που έχουν ενεργοποιηθεί, όπως φαίνεται στην Εικόνα 26.

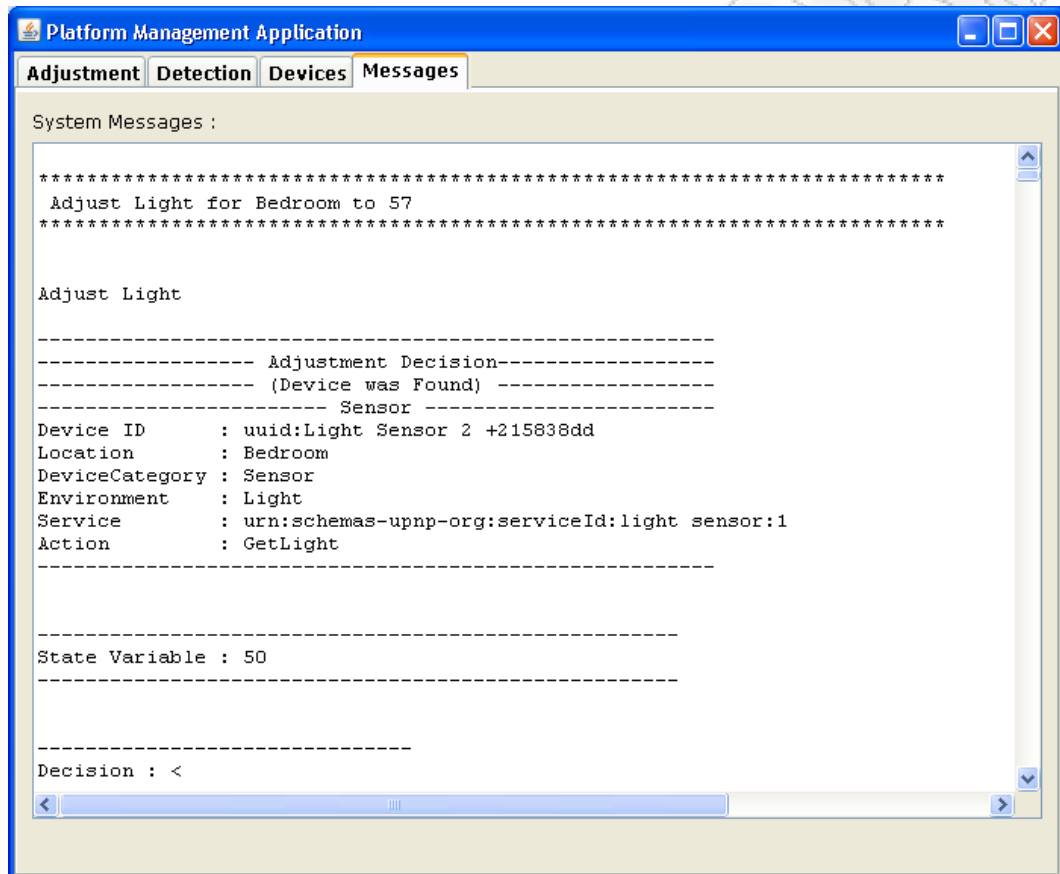


ID	Category	Friendly Name	Location	Manufact...	Model Name
uuid:Tempera...	Sensor	Temperature Sensor 2	Bedroom	Diakrousi...	Sun
uuid:Tempera...	Sensor	Temperature Sensor 1	Kitchen	Diakrousi...	Sun
uuid:Motion s...	Sensor	Motion sensor 2	Bedroom	Diakrousi...	Sun
uuid:Motion s...	Sensor	Motion sensor 1	Kitchen	Diakrousi...	Sun
uuid:Light Se...	Sensor	Light Sensor 2	Bedroom	Diakrousi...	Sun
uuid:Light Se...	Sensor	Light Sensor 1	Kitchen	Diakrousi...	Sun
uuid:Lamp 1 ...	Actuator	Lamp 1	Kitchen	Diakrousi...	Sun
uuid:Lamp 2 ...	Actuator	Lamp 2	Bedroom	Diakrousi...	Sun
uuid:Lamp 3 ...	Actuator	Lamp 3	Kitchen	Diakrousi...	Sun
uuid:Lamp 4 ...	Actuator	Lamp 4	Bedroom	Diakrousi...	Sun
uuid:AirCondit...	Actuator	AirCondition 1	Kitchen	Diakrousi...	Sun
uuid:AirCondit...	Actuator	AirCondition 2	Bedroom	Diakrousi...	Sun

Εικόνα 26. Εφαρμογή PMA - Λίστα Συσκευών Συστήματος

6.6 Κονσόλα Μηνυμάτων Συστήματος

Η λειτουργία αυτή ενεργοποιείται μέσω του tab *Messages* και προβάλλει στο χρήστη μια περιοχή όπου εμφανίζονται όλα τα μηνύματα του συστήματος, έτσι όπως φαίνεται στην Εικόνα 27.



Εικόνα 27. Εφαρμογή PMA – Μηνύματα Συστήματος

6.7 Δημιουργία και εισαγωγή στο σύστημα νέας συσκευής

Έστω ότι θέλουμε να επεκτείνουμε την πλατφόρμα IESAP εμπλουτίζοντας την με μια συσκευή νέου τύπου, όπως για παράδειγμα έναν εκκινητήρα ο οποίος θα ανοιγοκλείνει το παράθυρο του υπνοδωματίου.

Αυτό που πρέπει πρώτα να γίνει είναι να αναπτυχθεί ένας νέος προσομοιωτής συσκευής (εφόσον πρόκειται για νέου τύπου συσκευή), ο οποίος θα προσομοιώνει τη λειτουργία της. Δημιουργείται λοιπόν ένα java package το οποίο περιέχει τις αντίστοιχες κλάσεις που περιγράφηκαν και για τις υπόλοιπες συσκευές.

Επόμενο βήμα είναι η δήλωση της υπηρεσίας συσκευής στο xml αρχείο αντιστοίχισης υπηρεσιών – ενεργειών. Συγκεκριμένα θα γίνει η ακόλουθη προσθήκη :

```
<services>
  <abstractService>A_IncLight</abstractService>
  <deviceService>urn:schemas-upnp-
org:serviceId:window:1</deviceService>
  <deviceAction>OpenWin</deviceAction>
  <resultVariable>NewLight</resultVariable>
</services>
<services>
  <abstractService>A_DecLight</abstractService>
  <deviceService>urn:schemas-upnp-
org:serviceId:window:1</deviceService>
  <deviceAction>CloseWin</deviceAction>
  <resultVariable>NewLight</resultVariable>
</services>
```

Κατόπιν δημιουργείται το OSGi bundle το οποίο αποτελεί τη διεπαφή μιας συγκεκριμένης συσκευής με την πλατφόρμα IESAP (αντίστοιχο του bundle *lamp_1*) και προσδιορίζει τα χαρακτηριστικά της. Ειδικότερα ορίζονται τα εξής :

Κωδικός δωματίου = Υπνοδωμάτιο

Κατηγορία = Εκκινητήρας

Χαρακτηριστικό περιβάλλοντος = Φωτεινότητα

Πλέον έχει δημιουργηθεί η συσκευή και μένει να γίνει η εγκατάσταση της στην πλατφόρμα (αφού πρώτα παραχθεί το σχετικό jar). Η εγκατάσταση γίνεται μέσω του command prompt της OSGi με την ακόλουθη εντολή :

```
osgi> install file:///c:/window_1_1.0.0.jar
```

Η ενεργοποίηση της συσκευής γίνεται με την εντολή :

```
osgi> start 43
```

Το 43 είναι το αποδοθέν από την OSGi id του νέου bundle που μόλις εγκαταστάθηκε.

Με την ενεργοποίηση της, η συσκευή συμμετέχει στις λειτουργίες της πλατφόρμας IESAP και επηρεάζει το χαρακτηριστικό *Φωτεινότητα*, χωρίς να χρειάζεται καμιά νέα ρύθμιση ή παραμετροποίηση.

Για να εμπλουτιστεί η πλατφόρμα και με νέες συσκευές τύπου *εκκινήτρα παραθύρου*, για οποιοδήποτε δωμάτιο, αρκεί να αναπτυχθούν τα σχετικά bundles διεπαφής, να προσδιοριστούν τα χαρακτηριστικά τους και να γίνει η εγκατάσταση και η ενεργοποίηση τους όπως αναφέρθηκε και προηγουμένως. Το σημαντικό είναι ότι σε κάθε περίπτωση δε χρειάζεται επανεκκίνηση της πλατφόρμας.

7 ΣΥΜΠΕΡΑΣΜΑΤΑ

7.1 Ανασκόπηση

Σε αυτή την εργασία ασχοληθήκαμε με το σχεδιασμό και την υλοποίηση μιας πλατφόρμας διαχείρισης υπηρεσιών σε ευφυή περιβάλλοντα, με τη χρήση τεχνολογιών ενδιάμεσου λογισμικού προσανατολισμένων σε υπηρεσίες.

Αρχικά δώσαμε τον ορισμό των έξυπνων χώρων και κάναμε μια ιστορική αναδρομή σε σχετικά ερευνητικά έργα. Κατόπιν έγινε μια κατηγοριοποίηση τους και παρουσιάστηκαν οι βασικές τεχνολογίες υποδομής τους. Για να γίνουν αντιληπτές οι εφαρμογές και οι δυνατότητες αυτής της τεχνολογίας, έγινε αναφορά σε τρεις υλοποιήσεις έξυπνων χώρων οι οποίες περιλαμβάνουν, ένα έξυπνο μουσείο, ένα διαδραστικό χώρο εργασίας και μια έξυπνη τάξη. Στη συνέχεια έγινε αναφορά στην αρχιτεκτονική των πλατφόρμων έξυπνων χώρων και των αρχών που πρέπει να τη διέπουν και έγινε η παρουσίαση τριών σχετικών υλοποιήσεων. Σαν πιο διαδεδομένες τεχνολογίες για την ανάπτυξη αυτών των συστημάτων, αναγνωρίστηκαν το πλαίσιο OSGi, το πρωτόκολλο UPnP, η αρχιτεκτονική δικτύου Jini, το σύστημα των Web Services και η υποδομή λογισμικού Jade.

Το υπόλοιπο μέρος της εργασίας αφορούσε το σχεδιασμό, την υλοποίηση και τη λειτουργία της πλατφόρμας IESAP. Αρχικά προσδιορίστηκαν οι προδιαγραφές του συστήματος και κατόπιν παρουσιάστηκε η διαστρωμάτωση της αρχιτεκτονικής του καθώς και ο τρόπος διαχείρισης των δεδομένων των αισθητήρων. Επιπλέον έγινε μια αναλυτική περιγραφή των δομοστοιχείων της πλατφόρμας και των περιφερειακών συστημάτων της (προσομοιωτής περιβάλλοντος). Τέλος δόθηκαν μερικά παραδείγματα (screenshots) από σενάρια εκτέλεσης των λειτουργιών της πλατφόρμας.

7.2 Συμπεράσματα

Το ευρύ φάσμα εφαρμογών και η συμβολή στη βελτίωση της καθημερινότητας καθιστούν την τεχνολογία των έξυπνων χώρων πολύ σημαντική. Καθώς πρόκειται

για πολυσύνθετα συστήματα με πολλά ξεχωριστά στοιχεία τα οποία πρέπει να συνεργάζονται και να επικοινωνούν, χρειάζεται να υπάρχει και η ανάλογη υποδομή λογισμικού. Σε ότι αφορά αυτή την τελευταία, προκύπτει ότι δεν υπάρχει ένα γενικό και ενιαίο πλαίσιο που να καλύπτει όλες τις ανάγκες, αλλά κάθε προσπάθεια επικεντρώνεται στην αντιμετώπιση συγκεκριμένων προκλήσεων αναλόγως και του περιβάλλοντος που αφορά.

Σε αυτό το πλαίσιο, η προτεινόμενη πλατφόρμα μέσω της υπηρεσιοστρεφής της αρχιτεκτονικής και του συνδυασμού των τεχνολογιών OSGi και UPnP, καταφέρνει να ικανοποιήσει μερικές από τις βασικές αρχές σχεδίασης συστημάτων έξυπνων χώρων. Σε αυτές περιλαμβάνονται η χαλαρή σύζευξη, η επεκτασιμότητα, η ανεξαρτησία πλατφόρμας εκτέλεσης, η ενσωμάτωση και αλληλεπίδραση συσκευών ανεξαρτήτως κατασκευαστή, η διαλειτουργικότητα κ.α..

7.3 Προτάσεις για περαιτέρω ανάπτυξη

Η πλατφόρμα IESAP αναπτύχθηκε στα πλαίσια μιας διπλωματικής εργασίας και το γεγονός αυτό θέτει μερικούς περιορισμούς ως προς την έκταση και το περιεχόμενο της ερευνητικής προσπάθειας. Ακολουθεί μια αναφορά των επεκτάσεων και των μετατροπών που μπορούν να γίνουν, έτσι ώστε η πλατφόρμα να αποκτήσει πιο ολοκληρωμένη μορφή.

Επέκταση αρχιτεκτονικής. Στην πλατφόρμα IESAP, πολύ σημαντικό ρόλο παίζει ο μηχανισμός *Παρατήρησης-Ελέγχου του Στρώματος Αντίληψης*, ο οποίος όμως μπορεί να βελτιωθεί και να γίνει πιο αποτελεσματικός κάτω από πραγματικές συνθήκες. Για παράδειγμα το ανοιγοκλείσιμο των παραθυρόφυλλων για την αύξηση της φωτεινότητας έχει νόημα μόνο την ημέρα. Η υπηρεσία αυτή μπορεί επίσης να επηρεάσει τη θερμοκρασία ενός δωματίου συναρτήσει της ώρας και της εποχής του χρόνου. Έτσι η σχέση *Ενέργεια - Υπηρεσία* όπως περιγράφηκε στον Πίνακα 1 εξαρτάται και από άλλους παράγοντες (όπως η *ώρα* και η *εποχή*). Μια βελτίωση της πλατφόρμας IESAP θα περιελάμβανε την παραμετροποίηση αυτής της σχέσης, έτσι ώστε να λαμβάνει υπόψη της τέτοιους μεταβλητούς παράγοντες.

Επέκταση λειτουργιών. Οι λειτουργίες που υποστηρίζει η πλατφόρμα IESAP είναι μερικές μόνο από αυτές που απαιτούνται για την υλοποίηση ενός ολοκληρωμένου περιβάλλοντος έξυπνου σπιτιού. Στις ήδη υπάρχουσες θα μπορέσουν να προστεθούν :

Δημιουργία και εκτέλεση σεναρίων. Ο χρήστης να έχει τη δυνατότητα να δημιουργεί σενάρια διαβίωσης (π.χ. «Σενάριο Χαλάρωσης», «Σενάριο Ύπνου», «Σενάριο Μελέτης», «Σενάριο Οικονομίας Ενέργειας», «Σενάριο Follow Me») ρυθμίζοντας κατάλληλα τις συνθήκες του περιβάλλοντος και κατόπιν να επιλέγει να εκτελέσει κάποιο από αυτά, είτε σε κάθε δωμάτιο ξεχωριστά, είτε σε όλο το σπίτι.

Προγραμματισμός ενεργειών. Ο χρήστης να έχει τη δυνατότητα να προγραμματίσει την εκτέλεση συγκεκριμένων ενεργειών (π.χ. σβήσιμο όλων των φώτων) καθορίζοντας τόσο τη συχνότητα (π.χ. κάθε μέρα) όσο και τις προϋποθέσεις εκτέλεσης (π.χ. ανάλογα με την ύπαρξη ή όχι κίνησης μέσα στο δωμάτιο).

Εκτέλεση φωνητικών εντολών. Ο χρήστης να έχει τη δυνατότητα μέσω καθορισμένων φωνητικών εντολών, να διαχειρίζεται τους εκκινήτες και τις συσκευές του δωματίου στο οποίο βρίσκεται (π.χ. άνοιγμα των λαμπτήρων, ενεργοποίηση του κλιματισμού, κ.τ.λ.).

Απομακρυσμένη Διαχείριση. Μια χρήσιμη επέκταση της πλατφόρμας IESAP, θα ήταν η δυνατότητα απομακρυσμένης διαχείρισης των λειτουργιών της. Σε μια τέτοια προσέγγιση η εφαρμογή PMA (Platform Management Application) θα μπορούσε να εκτελείται σε ένα Web Server και να είναι διαθέσιμη μέσω HTTP πρωτοκόλλου από οποιοδήποτε σημείο διαθέτει πρόσβαση σε δίκτυο (τοπικό ή διαδίκτυο).

Κατανεμημένη Επεξεργασία. Η δυνατότητα κατανεμημένης επεξεργασίας στην περίπτωση της πλατφόρμας IESAP θα προσέφερε μεγάλη ευελιξία στην ανάπτυξη των έξυπνων χώρων. Σε ένα τέτοιο σενάριο οι διάφορες OSGi υπηρεσίες θα μπορούσαν να εκτελούνται σε διαφορετικές υπολογιστικές μονάδες και να συνεργάζονται με έναν ενιαίο και διάφανο τρόπο. Η υλοποίηση μιας τέτοιας

προσέγγισης περιλαμβάνει προηγουμένως την υποστήριξη της κατανεμημένης επεξεργασίας από την πλατφόρμα OSGi, κάτι το οποίο θα γίνει πραγματικότητα στην επόμενη έκδοση προδιαγραφών της.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΑΙΑ

ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ

AIT. (2007). *Autonomic and Grid Computing*. Ανάκτηση 01 03, 2009, από Athens Information Technology: <http://www.ait.edu.gr/research/RG1/lab.asp#1>

Allard, J., Chinta, V., Gundala, S., & Richard, G. G. (2003). Jini Meets UPnP: An Architecture for Jini/UPnP Interoperability. *Symposium on Applications and the Internet* (σσ. 268- 275). IEEE.

Apache. (2006, 07). *Apache Felix - UPnP Examples*. Ανάκτηση 06 2008, από Apache Felix: <http://felix.apache.org/site/upnp-examples.html>

Bai, Y.-W., & Hsu, J.-P. (2007). Design and implementation of an embedded home-gateway for remote monitoring based on OSGi technology. *European Conference on Proceedings of the IASTED European Conference: internet and multimedia systems and applications* (σσ. 63 - 68). Chamonix: ACTA Press.

Bartlett, N. (2007, 02). *Getting Started with OSGi*. Ανάκτηση 05 2008, από Neil's point-free blog: <http://neilbartlett.name/blog/osgi-articles/>

Coen, M. H., Phillips, B., Warshawsky, N., Weisman, L., Peters, S., & Finin, P. (1999). Meeting the Computational Needs of Intelligent Environments: The Metaglu System. *Proceedings of MANSE '99*. Dublin.

Dey, A. K. (2001). Understanding and Using Context. *Personal and Ubiquitous Computing* , 5.

Dimakis, N., Mylonakis, V., Soldatos, J., & Polymenakos, L. (2006). Reaching Outside the Smart Space: The Memory Jog Gateway. *15th International Conference on Computing* (σσ. 412-420). IEEE.

Equinox. (2004). *Equinox*. Ανάκτηση 05 2008, από Eclipse: <http://www.eclipse.org/equinox/>

Essa, I. A. (2000). Ubiquitous Sensing for Smart and Aware Environments. *IEEE Personal Communications* .

Fleck, M., Frid, M., Kindberg, T., O'Brien-Strain, E., Rajani, R., & Spasojevic, M. (2002). From Informing to Remembering: Ubiquitous Systems in Interactive Museums. *Pervasive Computing*, 1 (2), 13-21.

Fox, A., Johanson, B., Hanrahan, P., & Winograd, T. (2000). Integrating Information Appliances into an Interactive Workspace. *Computer Graphics and Applications*, 20 (3), 54-65.

Garlan, D., Siewiorek, D. P., Smailagic, A., & Steenkiste, P. (2002). Project Aura: Toward Distraction-Free Pervasive Computing. *Pervasive Computing*, 1 (2), 22-31.

Gouin-Vallerand, C., & Giroux, S. (2007). Managing and Deployment of Applications with OSGi in the Context of Smart Home. *Wireless and Mobile Computing, Networking and Communications* (σσ. 70-70). New York: IEEE.

Hall, R. S. (2004). *OSGi and Gravity Service Binder Tutorial*. Ανάκτηση 06 2008, από Oscar Bundle Repository: <http://oscar-osgi.sourceforge.net/tutorial/>

Helal, S., Mann, W., El-Zabadani, H., King, J., Kaddoura, Y., & Jansen, E. (2005). The Gator Tech Smart House: A Programmable Pervasive Space. *Computer*, 38 (3), 50-60.

Jade. (2001). *Technical Description*. Ανάκτηση 02 17, 2009, από Jade - Java Agent Development Framework: <http://jade.tilab.com/>

Jeronimo, M. (2004, 10 05). *It Just Works: UPnP in the Digital Home*. Ανάκτηση 03 02, 2009, από The Artima Developer Community: http://www.artima.com/spontaneous/upnp_digihome3.html

Johanson, B., Fox, A., & Winograd, T. (2002). The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms. *Pervasive Computing*, 1 (2), 67-74.

Lee, C., Nordstedt, D., & Helal, S. (2003). Enabling smart spaces with OSGi. *Pervasive Computing*, 2 (3), 89 - 94.

Ma, J., Yang, L. T., Apduhan, B. O., Huang, R., Barolli, L., Takizawa, M., και συν. (2005). A Walkthrough from Smart Spaces to Smart Hyperspaces towards a Smart World with Ubiquitous Intelligence. *International Journal of Pervasive Computing and Communications* , 1 (1), 53-68.

Miller, B. A., Nixon, T., Tai, C., & Wood, M. D. (2001). Home networking with Universal Plug and Play. *Communications Magazine* , 39 (12), 104-109.

Mozilla. (2008, 09 03). XUL. Ανάκτηση 02 10, 2009, από XUL - MDC: <https://developer.mozilla.org/en/XUL>

MSDN. (2009). XAML. Ανάκτηση 02 10, 2009, από .Net Framework Developer Center : <http://msdn.microsoft.com/en-us/library/ms747122.aspx>

OSGi, A. (2007, 05). *OSGi Service Platform Core Specification Release 4*. Ανάκτηση 03 01, 2009, από OSGi Alliance: <http://www.osgi.org/Specifications/HomePage>

Ou, S., Karuppiah, D. R., Fagg, A. H., & Riseman, E. (2004). An Augmented Virtual Reality Interface for Assistive Monitoring of Smart Spaces. *Pervasive Computing and Communications, 2004. PerCom 2004*, (σσ. 33 - 42). Florida.

Ponnekanti, S. R., Johanson, B., Kiciman, E., & Fox, A. (2003). Portability, Extensibility and Robustness in iROS. *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications* (σσ. 11-19). IEEE.

Roman, M., Hess, C., Cerqueira, R., Ranganat, A., Campbell, R. H., & Nahrstedt, K. (2002). Gaia: A Middleware Infrastructure to Enable Active Spaces. (IEEE, Επιμ.) *IEE Pervasive Computing* .

Rosenthal, L., & Stanford, V. (2000). NIST Smart Space: Pervasive Computing Initiative. *IEEE 9th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises* (σ. 6). WETICE.

Satyanarayanan, M. (2001). Pervasive Computing: Vision and Challenges. *IEEE Personal Communications* .

- Schmidt, A., & Laerhoven, K. V. (2001). How to build smart appliances? *Personal Communications* , 8 (4), 66-71.
- Shi, Y., Xie, W., Xu, G., Shi, R., Chen, E., Mao, Y., και συν. (2003). The Smart Classroom: Merging Technologies for Seamless Tele-Education. *Pervasive Computing* , 2 (2), 47-55.
- Shrobe, H. E. (2005). *Foreword of Smart Environments: Technologies, Protocols and Applications*. (D. J. Cook, & S. K. Das, Επιμ.) Wiley-Interscience.
- Siegemund, F., Floerkemeier, C., & Vogt, H. (2005). The value of handhelds in smart environments. *Personal and Ubiquitous Computing* , 9, 69-80.
- Singh, R., Bhargava, P., & Kain, S. (2006). State of the art smart spaces: application models and software infrastructure. *Ubiquity* , 7 (37).
- Soldatos, J., Dimakis, N., Stamatis, K., & Polymenakos, L. (2007). A breadboard architecture for pervasive context-aware services in smart spaces: middleware components and prototype applications. *Personal and Ubiquitous Computing* , 11 (3), 193-212.
- Stavroulaki, V., Demestichas, K., Adamopoulou, E., & Demestichas, P. (2006). Distributed Web-based Management Framework for Ambient Reconfigurable Services in the Intelligent Environment. *Mobile Networks and Applications* , 11 (6), 889-900.
- UIML.org. (2009). *UIML.org: Intro*. Ανάκτηση 02 15, 2009, από UIML.org: <http://www.uiml.org/intro/index.htm>
- W3C. (2004, 02 11). *Web Services Architecture*. Ανάκτηση 02 22, 2009, από W3C World Wide Web Consortium: <http://www.w3.org/TR/ws-arch/#whatis>
- Watson, T., & Kaegi, S. (2007, 07). *Introduction to Eclipse Equinox and OSGi*. Ανάκτηση 05 2008, από Eclipse Live: <http://live.eclipse.org/node/387>
- Weiser, M. (1999). The computer for the 21st century. *ACM SIGMOBILE Mobile Computing and Communications Review* , 3 (3), 3-11.

Wikipedia. (2009, 01 30). *Jini*. Ανάκτηση 02 09, 2009, από Wikipedia, The Free Encyclopedia: <http://en.wikipedia.org/w/index.php?title=Jini&oldid=267501091>

Wikipedia. (2009, 02 09). *OSGi*. Ανάκτηση 02 09, 2009, από Wikipedia, The Free Encyclopedia: <http://en.wikipedia.org/w/index.php?title=OSGi&oldid=269494001>

Wikipedia. (2009, 02 02). *Ubiquitous Computing*. Ανάκτηση 02 09, 2009, από Wikipedia, The Free Encyclopedia: http://en.wikipedia.org/w/index.php?title=Ubiquitous_computing&oldid=267994956

Wikipedia. (2009, 02 09). *UPnP*. Ανάκτηση 02 09, 2009, από Wikipedia, The Free Encyclopedia: http://en.wikipedia.org/w/index.php?title=Universal_Plug_and_Play&oldid=269519600

Wikipedia. (2008, 12 09). *User interface markup language*. Ανάκτηση 02 10, 2009, από Wikipedia, The Free Encyclopedia: http://en.wikipedia.org/w/index.php?title=User_interface_markup_language&oldid=256891910

Xie, W., Shi, Y., Xu, G., & Mao, Y. (2002). Smart Platform - A Software Infrastructure for Smart Space (SISS). *Fourth IEEE International Conference on Multimodal Interfaces* (σσ. 429-434). IEEE.

Zhang, D., Zhu, M., Cheng, H., Koh, Y., & Mokhtari, M. (2006). Handling Heterogeneous Device Interaction in Smart Spaces. Στο *Ubiquitous Intelligence and Computing* (Τόμ. 4159). Springer Berlin / Heidelberg.