



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ
ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

**ΣΧΕΔΙΑΣΗ ΚΑΙ ΑΝΑΠΤΥΞΗ ΑΝΟΙΚΤΗΣ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ
ΔΙΑΜΕΣΟΛΑΒΗΤΩΝ ΚΑΙ ΠΡΟΤΥΠΩΝ ΠΡΟΦΙΛ
ΓΙΑ ΥΠΗΡΕΣΙΕΣ ΜΕ ΕΠΙΓΝΩΣΗ ΤΟΥ ΠΛΑΙΣΙΟΥ ΧΡΗΣΗΣ ΤΟΥΣ**

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Σοφίας Ηλ. Καπελλάκη

Διπλωματούχου Μηχανολόγου & Αεροναυπηγού Μηχανικού Πανεπιστημίου Πατρών

Αθήνα, Μάιος 2007



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ
ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Σοφίας Ηλ. Καπελλάκη

Διπλωματούχου Μηχανολόγου και Αεροναυπηγού Μηχανικού Πανεπιστημίου Πατρών

ΣΧΕΔΙΑΣΗ ΚΑΙ ΑΝΑΠΤΥΞΗ ΑΝΟΙΚΤΗΣ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΔΙΑΜΕΣΟΛΑΒΗΤΩΝ ΚΑΙ ΠΡΟΤΥΠΩΝ ΠΡΟΦΙΛ ΓΙΑ ΥΠΗΡΕΣΙΕΣ ΜΕ ΕΠΙΓΝΩΣΗ ΤΟΥ ΠΛΑΙΣΙΟΥ ΧΡΗΣΗΣ ΤΟΥΣ

Συμβουλευτική Επιτροπή:

Δ.-Θ. Ι. Κακλαμάνη, Αν. Καθηγήτρια ΕΜΠ

Ι. Στ. Βενιέρης, Καθηγητής ΕΜΠ

Γ. Ι. Στασινόπουλος, Καθηγητής ΕΜΠ

Εγκρίθηκε από την επταμελή Εξεταστική Επιτροπή την 3^η Μαΐου 2007,

.....
Δ.-Θ. Ι. Κακλαμάνη
Αν. Καθηγήτρια ΕΜΠ

.....
Ι. Στ. Βενιέρης
Καθηγητής ΕΜΠ

.....
Γ. Ι. Στασινόπουλος
Καθηγητής ΕΜΠ

.....
Ε. Ν. Πρωτονοτάριος
Καθηγητής ΕΜΠ

.....
Ν. Ουζούνογλου
Καθηγητής ΕΜΠ

.....
Σ. Παπαβασιλείου
Επικ. Καθηγητής ΕΜΠ

.....
Χ. Δουληγέρης
Αν. Καθηγητής Παν. Πειραιώς

.....
Σοφία Ηλ. Καπελλάκη

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

Copyright © Σοφία Ηλ. Καπελλάκη, 2007.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

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

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

Περίληψη

Σκοπός της παρούσας διατριβής είναι η σχεδίαση και ανάπτυξη ενός συστήματος υποστήριξης υπηρεσιών βασισμένων στο πλαίσιο χρήσης τους. Συγκεκριμένα, ο στόχος του προτεινόμενου συστήματος είναι να καλύψει τις ανάγκες για ανάκτηση, διαχείριση και παροχή της απαιτούμενης πληροφορίας στις παρεχόμενες προς το χρήστη υπηρεσίες με ομοιόμορφο και προτυποποιημένο τρόπο, με χρήση προτύπων προφίλ και ενός συνόλου διαμεσολαβητών.

Σε πρώτη φάση, γίνεται μία συνοπτική αναδρομή στη σημερινή κατάσταση στον τομέα της παροχής υπηρεσιών βασισμένων στο πλαίσιο χρήσης τους. Καταγράφονται οι έως τώρα προσπάθειες για τον καθορισμό των στοιχείων που αποτελούν το πλαίσιο χρήσης μίας υπηρεσίας, οι προσεγγίσεις μοντελοποίησης του και οι διάφορες αρχιτεκτονικές που υποστηρίζουν την παροχή τέτοιου είδους υπηρεσιών. Στη συνέχεια περιγράφεται μία πρώιμη προσπάθεια υλοποίησης πλατφόρμας παροχής υπηρεσιών αναζήτησης βασισμένων στη θέση και στις προτιμήσεις του χρήστη, καθώς και τα προβλήματα που παρουσιάστηκαν, από τα οποία προέκυψαν καιρικά συμπεράσματα για τον σχεδιασμό και την υλοποίηση του προτεινόμενου συστήματος.

Ακολουθεί εκτενής περιγραφή και ανάλυση του προτεινόμενου συστήματος. Παρουσιάζεται η προσέγγιση που ακολουθήθηκε για την μοντελοποίηση της πληροφορίας πλαισίου χρήσης υπηρεσίας με χρήση προφίλ βασισμένων σε ευρέως αποδεκτά πρότυπα, όπως το 3GPP Generic User Profile. Εν συνεχεία, περιγράφονται οι βασικές αρχές που ακολουθήθηκαν για τον σχεδιασμό του συστήματος, τα βασικά δομικά στοιχεία του, διαμεσολαβητές και υποσυστήματα, καθώς και η συνολική αρχιτεκτονική. Ιδιαίτερη βαρύτητα δίνεται στην περιγραφή της λειτουργικότητας και της υλοποίησης των επιμέρους οντοτήτων και των μηχανισμών επικοινωνίας τους. Τέλος, αναλύονται διεξοδικά οι βασικές λειτουργίες του συστήματος και περιγράφονται συνοπτικά οι υπηρεσίες που υλοποιήθηκαν για την επιβεβαίωση της ορθής λειτουργίας του. Επίσης, παρατίθεται η αποτίμηση της απόδοσης του συστήματος ως προς την ανάκτηση πληροφορίας θέσης χρήστη σε σχέση με την πλατφόρμα παροχής υπηρεσιών αναζήτησης.

Η διατριβή ολοκληρώνεται με τα σημαντικότερα συμπεράσματα, καθώς και με την παρουσίαση των μελλοντικών επεκτάσεων της προτεινόμενης λύσης.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Μοντελοποίηση Πληροφορίας Πλαισίου Χρήσης, 3GPP Generic User Profile, Ανάκτηση & Διαχείριση Πληροφορίας Πλαισίου Χρήσης, Παροχή Υπηρεσιών βασισμένων στο Πλαίσιο Χρήσης τους, Μεσισμικό, MLP (Mobile Location Protocol), Διεπαφή Κινητικότητας OSA/Parlay.

Summary

The scope of this thesis is the design and implementation of a system for the support of context aware service provision. The system shall provide the tasks of retrieval, management and provision of the relative to the user context information to the services with a prototyped and homogenous way.

The first part includes a concise retrospection of the current situation in the field of context aware services regarding the definition and modeling of context, as well as the various architectures and systems that support the provision of this kind of services. In addition, a first initial attempt of implementing such a service platform is described. The platform in discussion offers a number of searching services based on the user location and preferences. The problems and shortcomings of the platform have resulted in very significant conclusions that have constituted the guidelines for the design and implementation of the proposed system.

An extensive description and analysis of the proposed system follows. Initially, the approach for the modeling of context that has been followed is presented. The model was based on the concept of profiles, using widely accepted standards such as the 3GPP Generic User Profile. In the following, the design principles, the basic architectural components and the overall architecture are presented. Special attention is given in the offered functionalities description, the implementation of the individual components and their relations, as well as the communication mechanisms among them. Finally, a short description of the deployed services is cited as a proof of concept and validation of proper operation of the system. In addition, a comparative evaluation between the proposed system and the previously implemented service platform is presented.

The thesis concludes with the evaluation of the more important results of the approach, as well as with the presentation of the potential future evolution and exploitation of the proposed system.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Context Modeling, 3GPP Generic User Profile, Retrieval and Management of Context, Context Aware Service Provision, Brokerage Framework, Middleware, MLP (Mobile Location Protocol), OSA/Parlay Mobility API.

Αντί Προλόγου

Η παρούσα διατριβή αποτελεί το προϊόν της ερευνητικής μου δραστηριότητας στο εργαστήριο ευφών επικοινωνιών και δικτύων ευρείας ζώνης του ΕΜΠ. Έναυσμα της διατριβής αποτέλεσε αδιαμφισβήτητα η άμεση εμπλοκή μου στο ευρωπαϊκό ερευνητικό πρόγραμμα SIMPLICITY. Τα ερεθίσματα που εισέπραξα και οι τεχνικές προκλήσεις με τις οποίες ήρθα σε επαφή καθ' όλη τη διάρκεια εκπόνησης του εν λόγω ερευνητικού προγράμματος λειτούργησαν ως παρακαταθήκη παρέχοντας μου τις βάσεις για την ολοκλήρωση της εργασίας. Κύριο θέμα της διατριβής αποτελεί η παροχή υπηρεσιών με επίγνωση του πλαισίου χρήσης τους, το οποίο και αναλύεται διεξοδικά στα αμιγώς τεχνικά κεφάλαια που ακολουθούν.

Αναγνωρίζοντας ότι η διατριβή μου θα ήταν αδύνατον να πραγματοποιηθεί χωρίς την καθοδήγηση και την αμέριστη συμπαράσταση των καθηγητών μου και των συναδέλφων μου, θα ήθελα στο σημείο αυτό να ευχαριστήσω όλους όσους με βοήθησαν και με στήριξαν στην πορεία μου αυτή. Ξεκινώντας από τη συμβουλευτική μου επιτροπή, θα ήθελα ιδιαίτερα να ευχαριστήσω την επιβλέπουσα καθηγήτρια μου κυρία Δήμητρα Κακλαμάνη, Αναπληρώτρια Καθηγήτρια, η οποία με εμπιστεύτηκε, με καθοδήγησε και μου συμπαραστάθηκε σε κάθε μου βήμα. Πάντοτε πρόθυμη και διαλλακτική, πάντοτε διαθέσιμη για να συζητήσουμε και να επιλύσουμε οποιοδήποτε επιστημονικό –και όχι μόνο– ζήτημα. Την ευχαριστώ από καρδιάς για όσα μου δίδαξε, για τις γνώσεις και την εμπειρία που μου μετέδωσε και για τη φιλία της που απλόχερα μου προσέφερε. Συνεχίζοντας, θα ήθελα να ευχαριστήσω τον Καθηγητή κύριο Ιάκωβο Βενιέρη. Πνεύμα οξυδερκίες, με βαθύ επιστημονικό και τεχνικό υπόβαθρο, αποτέλεσε σημαντικό αρωγό για την ολοκλήρωση της διατριβής. Η οξύτητα της κρίσης του, η διαύγεια στη σκέψη του και η πολύπλευρη γνώση του επί παντός επιστητού με έκανε να αναζητώ συχνά τη συμβουλή του και να τον εμπιστεύομαι όχι μόνο ως καθηγητή μου, αλλά και ως «πατέρα». Επίσης, θερμές ευχαριστίες στους Καθηγητές, κύριο Γεώργιο Στασινόπουλο, πάντοτε ευγενικό και άριστο συμβουλάτορα, κύριο Νικόλαο Ουζούνογλου και κύριο Εμμανουήλ Πρωτονοτάριο, στους οποίους οφείλεται σε μεγάλο βαθμό το κύρος και το όνομα που φέρει η σχολή μας πανευρωπαϊκά αλλά και παγκοσμίως, στον Επίκουρο Καθηγητή κύριο Συμεών Παπαβασιλείου πάντα πρόθυμο να με βοηθήσει σε ό,τι ήθελα και στον Αναπληρωτή Καθηγητή κύριο Χρήστο Δουληγέρη από το Πανεπιστήμιο του Πειραιά, που μου έκανε την τιμή να συμπεριληφθεί στην επταμελή μου εξεταστική επιτροπή.

Αν και νιώθω ότι πιθανόν μακροηγορώ, δεν είναι δυνατόν να μην αφιερώσω λίγα λόγια για ορισμένους από τους συναδέλφους μου με τους οποίους μοιράστηκα αυτήν την τόσο σημαντική περίοδο της ζωής μου. Ξεκινώντας την αναδρομή από τα πρώτα κιόλας χρόνια, δεν μπορώ να μην

αναφερθώ στον Ανδρέα Παπαδάκη, ικανότατο μηχανικό, με σφαιρικές γνώσεις, υπόδειγμα εργατικότητας στο εργαστήριο, στον Προκόπη Γιαννακάκη, πνεύμα οξύ, πρότυπο ηρεμίας και εντιμότητας και στον Σωτήρη Μανιάτη, εξαιρετο μηχανικό και χαρισματικό άνθρωπο. Ήταν οι πρώτοι μου «δάσκαλοι». Δεύτερος στην «ιεραρχία», «δάσκαλος» και αυτός, ο Μανώλης Χανιωτάκης, πάντα άμεσος και αποτελεσματικός. Από τις νεότερες γενιές του εργαστηρίου θα ήθελα να ευχαριστήσω, τόσο για τη συνεργασία όσο και για τις στιγμές που μοιραστήκαμε τόσα χρόνια, τον Πέτρο Σαμπατάκο, την Ευγενία Νικολούζου, τη Λίλα Δημοπούλου, την Χρύσα Παπαγιάννη, το Γιώργο Λεολέη, το Γιάννη Παπάνη, το Νίκο Δέλλα, τους «Βαγγέληδες», Κοσμάτο και Νηια, άλλα και όλα τα νεότερα παιδιά. Ιδιαίτερες ευχαριστίες για τον ξεχωριστό συνεργάτη και φίλο, Γιώργο Λιουδάκη. Επίσης, ευχαριστώ θερμά τις Μαρίες, τη Μαρία Στριμπάκου και τη Μαρία Χατζάρα, τις “φίλες μου στο εργαστήριο”, που τόσα χρόνια μοιραζόμασταν τις ανησυχίες και τους προβληματισμούς μας και ενθαρρύναμε η μία την άλλη, καθώς και τη Ρένα Συγκιούνα και την Αναστασία Καλταμπάνη. Τέλος, δεν μπορώ να μην αναφερθώ στους νεότερους φίλους μου στο «κάτω εργαστήριο». Δεν θα ξεχάσω τα ξένοιαστα και χαρούμενα διαλείμματα από τη δουλειά με το Γιώργο Κουντουράκη, την ψυχή του εργαστηρίου, και την αστείρευτη Άρτεμη Κουτσορόδη, αλλά και με τον Δημήτρη Κυριαζάνο, τον Δημήτρη Αξιώτη, τον Μιχάλη Αργυρόπουλο, τη Λουΐζα Παπαδοπούλου, τον Νίκο Κουτσούρη, τον Διονύση Κεφαλληνό, την Αγγελική Σγόρα...

Ιδιαίτερα θα ήθελα να αναφερθώ στο Γιώργο Πρεζεράκο. Καλός φίλος και συνάδελφος, ταλαντούχος μηχανικός, ευγενής και καλλιεργημένος, με βοήθησε και με συμβούλεψε πολλές φορές. Τον ευχαριστώ θερμά για όλα. Τελευταίος έμεινε, εσκεμμένα, ο πιο στενός μου συνεργάτης και καλός φίλος, Λευτέρης Κουτσολουκάς. Εξαιρετικός μηχανικός, υπόδειγμα θετικού μυαλού και ακέραιου χαρακτήρα, πάντα πρόθυμος, αθόρυβος και αποτελεσματικός, με ιδιαίτερο χιούμορ. Τον ευχαριστώ εκ βάθους, τόσο για τις ατέλειωτες ώρες που μοιραστήκαμε στο εργαστήριο και στα ταξίδια, όσο και για την πολύτιμη και ανιδιοτελή συμβολή του στην ολοκλήρωση της διατριβής μου. Χωρίς τον Λευτέρη δεν θα τα είχα καταφέρει...

Ένα μεγάλο ευχαριστώ οφείλω και στους αγαπημένους φίλους και κουμπάρους μου, τον Κώστα Τσαγκάρη και τη Τζώρτζια Παράσχη, για την συμπαράσταση τους και την κατανόηση που έδειξαν. Οι ευχάριστες ώρες μαζί τους, ιδιαίτερα κατά τη διάρκεια της συγγραφής, με ανανέωναν και με βοηθούσαν να συνεχίσω. Όπως ακριβώς συνέβαινε και με τον Νίκο Πλέρο και τη Μαρία Παπαδοπούλου, τον Γιάννη Στάμο και την Ειρήνη Τζινιέρη, το Μανώλη Πουλή και τη Χρυσούλα Γλωσσίδου, το Γιώργο Πρεζεράκο και την Κατερίνα Νυδριώτη.

Άφησα για το τέλος την οικογένεια μου, τον άντρα μου, Νίκο Τσελίκα, τη μητέρα μου Βούλα και την αδελφή μου Στέλλα. Ο Νίκος, μου στάθηκε και με βοήθησε σαν σύντροφος, σαν φίλος και σαν συνεργάτης. Του οφείλω πολλά και τον ευχαριστώ θερμά για την υπομονή, τη βοήθεια, τη

διαρκή του υποστήριξη και το ενδιαφέρον που έδειξε όλο αυτό το διάστημα. Από τα βάθη της ψυχής μου ευχαριστώ τη μητέρα και την αδερφή μου για την αγάπη και τη φροντίδα που μου χάριζαν και μου χαρίζουν καθημερινά και την πίστη τους σε εμένα σε όλη τη διάρκεια των σπουδών μου. Η ειπώση της διατριβής, αλλά και η γενικότερη έως τώρα πορεία μου, οφείλεται κατά ένα πολύ σημαντικό μέρος στην οικογένειά μου. Τους αγαπώ και τους ευχαριστώ θερμά.

Τέλος, ένα μεγάλο ευχαριστώ σε εκείνους που ενώ έφυγαν νωρίς τους ένιωθα πάντα δίπλα μου να δρουν σαν κινητήρια δύναμη για να συνεχίζω εγώ....

Σοφία Καπελλάκη

Περιεχόμενα

<i>Περίληψη</i>	5
<i>Summary</i>	7
<i>Αντί Προλόγου</i>	9
<i>Περιεχόμενα</i>	13
<i>Σχήματα & Πίνακες</i>	17
1ο Κεφάλαιο	21
1.1 Εισαγωγή.....	23
1.2 Διάθροση της διατριβής	24
1.3 Τι σημαίνει Πλαίσιο Χρήσης Υπηρεσίας (Context)?.....	25
1.3.1 Κατηγορίες Πληροφορίας Πλαισίου Χρήσης Υπηρεσίας.....	27
1.4 Τι σημαίνει Επίγνωση Πλαισίου Χρήσης Υπηρεσίας (Context-Awareness)?	28
1.5 Εφαρμογές και Αρχιτεκτονικές με Επίγνωση Πλαισίου Χρήσης Υπηρεσίας	30
1.5.1 Εφαρμογές με Επίγνωση Πλαισίου Χρήσης τους.....	30
1.5.2 Αρχιτεκτονικές και Συστήματα παροχής υπηρεσιών με Επίγνωση Πλαισίου Χρήσης τους.....	33
1.6 Μία πρώτη προσέγγιση – Πλατφόρμα Παροχής Υπηρεσιών Αναζήτησης βάσει Θέσης και Χαρακτηριστικών του Χρήστη.....	37
1.6.1 Η αρχιτεκτονική της πλατφόρμας.....	37
1.6.1.1 Ο Εξυπηρετητής Εφαρμογών (Application Server) και οι Εφαρμογές	38
1.6.1.2 Η Βάση Δεδομένων – Προφίλ και προτιμήσεις χρηστών	42
1.6.1.3 Η Οντότητα Θέσης (Location Entity).....	43
1.7 Συμπεράσματα και κατευθύνσεις.....	52
1.8 Αναφορές.....	54
2ο Κεφάλαιο	61
2.1 Εισαγωγή	63
2.2 Προσεγγίσεις Μοντελοποίησης της Πληροφορίας Πλαισίου Χρήσης Υπηρεσίας	64
2.3 Τα Μοντέλα Πληροφορίας και Δεδομένων	68
2.3.1 Το Μοντέλο Πληροφορίας.....	68
2.3.1.1 Το μοντέλο πυρήνα (core model)	70
2.3.1.2 Το μοντέλο χρήστη	71
2.3.1.3 Το μοντέλο συσκευών.....	72
2.3.1.4 Το μοντέλο δικτύου	73
2.3.1.5 Το μοντέλο υπηρεσίας.....	74
2.3.1.6 Το μοντέλο πληροφορίας πλαισίου χρήσης υπηρεσίας.....	76
2.3.2 Το Μοντέλο Δεδομένων	76
2.3.2.1 Προφίλ Χρήστη	78
2.3.2.2 Προφίλ Τεραματικού.....	81
2.3.2.3 Προφίλ Υπηρεσίας.....	82

2.3.2.4 Προφίλ Δικτύου	84
2.3.2.5 Το σχήμα Πληροφορίας Πλαισίου Χρήσης Υπηρεσίας	85
2.3.2.6 Αναπαράσταση ενός στιγμιότυπου Πληροφορίας Πλαισίου Χρήσης Υπηρεσίας	86
2.4 Αναφορές	88
3ο Κεφάλαιο	93
3.1 Εισαγωγή	95
3.2 Η Αρχιτεκτονική του συστήματος	95
3.2.1 Η οντότητα του Διαμεσολαβητή: Μεσάζων & Υποσύστημα	95
3.2.2 Μηχανισμός Επικοινωνίας Διαμεσολαβητών.....	96
3.2.2.1 Αρχιτεκτονική Υποσυστήματος Επικοινωνίας Διαμεσολαβητών	98
3.2.3 Ο Διαχειριστής Εξωτερικής Διεπαφής	104
3.2.4 Προσωπική Συσκευή Χρήστη / Υποσύστημα Διαχειριστή Πρόσβασης Προσωπικής Συσκευής Χρήστη (ΥΔΠΠΣΧ).....	105
3.2.5 Προσωπικός Βοηθός Χρήστη/ Υποσύστημα Διαχειριστή Εξωτερικής Διεπαφής.....	108
3.2.6 Απαιτούμενες Λειτουργίες - Υποσυστήματα	109
3.2.6.1 Υποσύστημα Διαχείρισης Προφίλ Χρήστη (ΥΔΠΧ)	109
3.2.6.2 Υποσύστημα Διαχείρισης Δυνατοτήτων Τερματικών Συσκευών (ΥΔΔΤΣ)	110
3.2.6.3 Υποσύστημα Διαχείρισης Υπηρεσιών (ΥΔΥ).....	110
3.2.6.4 Υποσύστημα Διαχείρισης Δικτυακών Συνδέσεων (ΥΔΔΣ).....	111
3.2.6.5 Υποσύστημα Διαχείρισης Πληροφορίας Θέσης Χρήστη (ΥΔΠΘΧ).....	111
3.2.6.6 Υποσύστημα Διαχείρισης Πληροφορίας Πλαισίου (ΥΔΠΠ)	112
3.2.7 Η Συνολική Αρχιτεκτονική	112
3.3 Η Υλοποίηση του συστήματος	114
3.3.1 Η οντότητα του Διαμεσολαβητή: Μεσάζων & Υποσύστημα	114
3.3.1.1 Δημιουργία και εγγραφή υποσυστήματος στον Μεσάζοντα	118
3.3.1.2 Προώθηση γεγονότος μέσω του Μεσάζοντα	119
3.3.2 Υποσύστημα Επικοινωνίας Διαμεσολαβητών	120
3.3.2.1 Πρωτόκολλο Επικοινωνίας Διαμεσολαβητών.....	120
3.3.2.2 Υλοποίηση	121
3.3.3 Υποσύστημα Διαχείρισης Προφίλ Χρήστη.....	130
3.3.4 Υποσύστημα Διαχείρισης Δυνατοτήτων Τερματικών Συσκευών	134
3.3.5 Υποσύστημα Διαχείρισης Δικτυακών Συνδέσεων.....	137
3.3.6 Υποσύστημα Διαχείρισης Υπηρεσιών	138
3.3.7 Υποσύστημα Διαχείρισης Πληροφορίας Θέσης Χρήστη.....	141
3.3.8 Υποσύστημα Διαχείρισης Πληροφορίας Πλαισίου.....	142
3.4 Βασικές Αλληλεπιδράσεις	144
3.4.1 Διαδικασία Εισαγωγής στο σύστημα (Login Procedure).....	144
3.4.2 Διαδικασία πρώτης ανάκτησης πληροφοριών Context από τους Διαμεσολαβητές	146
3.4.3 Διαδικασία Ανάκτησης Διαθέσιμων Υπηρεσιών	148
3.4.4 Μηχανισμοί Ενημέρωσης.....	149
3.4.5 Διαδικασίες Διαχείρισης των Προφίλ	152
3.4.6 Γενικευμένο Παράδειγμα Χρήσης Υπηρεσίας.....	154
3.5 Παρεχόμενες Υπηρεσίες	156
3.6 Αναφορές	158
4ο Κεφάλαιο	161
4.1 Εισαγωγή	163

4.2 Ανάκτηση πληροφορίας θέσης μέσω της προτεινόμενης αρχιτεκτονικής.....	164
4.3 Αποτίμηση και σύγκριση των προσεγγίσεων	167
4.3.1 Μοντέλο αποτίμησης απόδοσης	168
4.3.2 Μετρήσεις – Αποτελέσματα	174
4.4 Αναφορές	179
5ο Κεφάλαιο	181
5.1 Εισαγωγή	183
5.2 Συμπεράσματα	183
5.3 Σενάρια Ανάπτυξης του Συστήματος – Κλιμάκωση και Κατανομή	188
5.4 Επιχειρηματικό Μοντέλο	193
5.5 Μελλοντικές Εξελίξεις-Προτεινόμενες Επεκτάσεις.....	196
5.5.1 Σύνθεση Υπηρεσιών με εικεταλλευση Προτύπων Χρήσης (Usage Patterns).....	196
5.5.1.1 Επιλογή Προτύπου Χρήσης και Δημιουργία Προσαρμοσμένου στο Χρήστη Πρότυπου Χρήσης	197
5.5.1.2 Ενσωμάτωση Προτύπων Χρήσης στην προτεινόμενη αρχιτεκτονική	199
5.5.2 Ασφάλεια και Ιδιωτικότητα – Η Προσέγγιση Discreet.....	200
5.6 Αναφορές	202

Σχήματα & Πίνακες

ΛΙΣΤΑ ΣΧΗΜΑΤΩΝ

Σχήμα 1: Η Αρχιτεκτονική της πλατφόρμας	37
Σχήμα 2: Διαγράμματα κλάσεων της διαδικασίας εγγραφής	39
Σχήμα 3: Διάγραμμα κλάσεων της υπηρεσίας MatchMe	40
Σχήμα 4: Διάγραμμα κλάσεων της υπηρεσίας MatchViaNickname	40
Σχήμα 5: Διάγραμμα κλάσεων της υπηρεσίας MatchViaPreferences	41
Σχήμα 6: Διάγραμμα κλάσεων της υπηρεσίας Infokiosk	42
Σχήμα 7: Στιγμιότυπο της βάσης	43
Σχήμα 8: Διάγραμμα Αλληλεπιδράσεων Υπηρεσία Match-Me	45
Σχήμα 9: Διάγραμμα Αλληλεπιδράσεων Υπηρεσία SearchByNickname	46
Σχήμα 10: Διάγραμμα Αλληλεπιδράσεων Υπηρεσία Infokiosk.Search	47
Σχήμα 11: Η Αρχιτεκτονική της πλατφόρμας με χρήση της Διεπαφής Κινητικότητας του OSA/Parlay	48
Σχήμα 12: Διάγραμμα αλληλεπιδράσεων παροχής υπηρεσίας με χρήση της Διεπαφής OSA/Parlay	50
Σχήμα 13: Διάγραμμα κλάσεων του συστήματος	52
Σχήμα 14: Η δομή του μετα-σχήματος	69
Σχήμα 15: Συμβάσεις σχέσεων	70
Σχήμα 16: Το μοντέλο πυρήνα	70
Σχήμα 17: Το μοντέλο χρήστη	71
Σχήμα 18: Το μοντέλο συσκευών	72
Σχήμα 19: Το μοντέλο δικτύου	73
Σχήμα 20: Το μοντέλο υπηρεσίας	74
Σχήμα 21: Το μοντέλο πληροφορίας πλαισίου χρήσης υπηρεσίας	76
Σχήμα 22: Η δομή του 3GPP GUP	77
Σχήμα 23: Η δομή του CoP	77
Σχήμα 24: Επικοινωνία Υποσυστημάτων μέσω του Μεσάζοντα	96
Σχήμα 25: Τα στρώματα του Μηχανισμού Επικοινωνίας Διαμεσολαβητών	97
Σχήμα 26: Το Υποσύστημα Επικοινωνίας Διαμεσολαβητών	98
Σχήμα 27: Διαδικασία Ανακάλυψης και Εγγραφής Απομακρυσμένων Υποσυστημάτων	99
Σχήμα 28: Μεταφορά γεγονότος σε απομακρυσμένο υποσύστημα	101
Σχήμα 29: Αρχιτεκτονική του Υποσυστήματος Επικοινωνίας Διαμεσολαβητών (ΥΕΔ)	102
Σχήμα 30: Αρχιτεκτονική Υποσυστήματος Επικοινωνίας Διαμεσολαβητών (ΥΕΔ) με μεταφερόμενο Listener	103
Σχήμα 31: Γενική Αρχιτεκτονική Διαμεσολαβητή που αλληλεπιδρά με εξωτερική οντότητα	105
Σχήμα 32: Η αρχιτεκτονική της Προσωπικής Συσκευής Χρήστη και του Υποσυστήματος Διαχείρισης Πρόσβασης Προσωπικής Συσκευής Χρήστη	107
Σχήμα 33: Διαστρωμάτωση της Αρχιτεκτονικής	113
Σχήμα 34: Η προτεινόμενη αρχιτεκτονική: οντότητες και διεπαφές	114
Σχήμα 35: Διάγραμμα Κλάσεων του Διαμεσολαβητή	115
Σχήμα 36: Διάγραμμα αλληλεπιδράσεων δημιουργίας και εγγραφής υποσυστήματος στον Μεσάζοντα	118
Σχήμα 37: Διάγραμμα αλληλεπιδράσεων προώθησης γεγονότος μέσω του Μεσάζοντα	119
Σχήμα 38: Στοίβα Πρωτοκόλλων της Διεπαφής ΥΕΔ	121
Σχήμα 39: Διάγραμμα Κλάσεων των μηνυμάτων του AEP	122
Σχήμα 40: Αλγόριθμος ανάθεσης BCID και OriginatorID στα μηνύματα ΥΕΔ	124
Σχήμα 41: Εγκατάσταση Listener, Μεταφορά Μηνύματος και Απεγκατάσταση Listener	124

Σχήμα 42: Διάγραμμα κλάσεων του Υποσυστήματος Διαχείρισης Προφίλ Χρήστη στον Διαμεσολαβητή Δικτύου.....	131
Σχήμα 43: Διάγραμμα κλάσεων του Υποσυστήματος Διαχείρισης Προφίλ Χρήστη στον Διαμεσολαβητή Τερματικού.....	133
Σχήμα 44: Διάγραμμα κλάσεων του Υποσυστήματος Διαχείρισης Δυνατοτήτων Τερματικών Συσκευών στον Διαμεσολαβητή Δικτύου.....	135
Σχήμα 45: Διάγραμμα κλάσεων του Υποσυστήματος Διαχείρισης Δυνατοτήτων Τερματικών Συσκευών στον Διαμεσολαβητή Τερματικού.....	136
Σχήμα 46: Διάγραμμα κλάσεων του Υποσυστήματος Διαχείρισης Δικτυακών Συνδέσεων στον Διαμεσολαβητή Δικτύου.....	137
Σχήμα 47: Διάγραμμα κλάσεων του Υποσυστήματος Διαχείρισης Δικτυακών Συνδέσεων στον Διαμεσολαβητή Τερματικού.....	138
Σχήμα 48: Διάγραμμα κλάσεων του Υποσυστήματος Διαχείρισης Υπηρεσιών στον Διαμεσολαβητή Δικτύου.....	139
Σχήμα 49: Διάγραμμα κλάσεων του Υποσυστήματος Διαχείρισης Υπηρεσιών στον Διαμεσολαβητή Τερματικού.....	140
Σχήμα 50: Διάγραμμα κλάσεων του Υποσυστήματος Διαχείρισης Πληροφορίας Θέσης Χρήστη στον Διαμεσολαβητή Δικτύου.....	141
Σχήμα 51: Διάγραμμα κλάσεων του Υποσυστήματος Διαχείρισης Πληροφορίας Πλαισίου στον Διαμεσολαβητή Δικτύου.....	143
Σχήμα 52: Διάγραμμα αλληλεπιδράσεων διαδικασίας Login.....	145
Σχήμα 53: Ανάκτηση Προφίλ από τα υποσυστήματα διαχείρισης τους στους Διαμεσολαβητές Δικτύου και Τερματικού.....	146
Σχήμα 54: Ανάκτηση Προφίλ από το υποσύστημα διαχείρισης πληροφορίας πλαισίου στον Διαμεσολαβητή Δικτύου.....	147
Σχήμα 55: Διάγραμμα αλληλεπιδράσεων της διαδικασίας ανάκτησης διαθέσιμων υπηρεσιών.....	148
Σχήμα 56: Διάγραμμα αλληλεπιδράσεων Μηχανισμού Εγγραφής και Ενημέρωσης.....	151
Σχήμα 57: Διάγραμμα αλληλεπιδράσεων διαδικασίας ανανέωσης προφίλ εκκινούμενης από τυχαίο υποσύστημα στον Διαμεσολαβητή Τερματικού.....	152
Σχήμα 58: Διάγραμμα αλληλεπιδράσεων διαδικασίας ανανέωσης προφίλ εκκινούμενης από τυχαίο υποσύστημα στον Διαμεσολαβητή Δικτύου.....	154
Σχήμα 59: Διάγραμμα αλληλεπιδράσεων Γενικευμένου Παραδείγματος Χρήσης Υπηρεσίας.....	155
Σχήμα 60: Αρχιτεκτονική Συστήματος Παροχής Υπηρεσιών Αναζήτησης βάσει Θέσης και Προτιμήσεων Χρήστη με χρήση του προτεινόμενου συστήματος.....	165
Σχήμα 61: Διαδικασία ανάκτησης Θέσης Χρήστη Μέσω του Διαμεσολαβητή Δικτύου.....	166
Σχήμα 62: Απεικόνιση των χρονικών τιμών που συμβάλλουν στην καθυστέρηση α) με χρήση του πρωτοκόλλου MLP πάνω από HTTP β) με χρήση της Πύλης Εντοπισμού Θέσης πάνω από στρώμα μεσοσπικτικού και γ) με χρήση του Διαμεσολαβητή Δικτύου.....	170
Σχήμα 63: Ο μέσος συνολικός χρόνος εξυπηρέτησης (t_{TOTAL}) της υπηρεσίας.....	175
Σχήμα 64: Συνδυασμένη Ανάπτυξη του Συστήματος.....	193
Σχήμα 65: Βασικές Οικονομικές Ροές του Επιχειρηματικού μοντέλου.....	195
Σχήμα 66: Διαδοχικά βήματα για την επιλογή των προς σύνθεση υπηρεσιών βάσει του στόχου του χρήστη.....	196
Σχήμα 67: Διαδοχικά βήματα για την δημιουργία του Προσαρμοσμένου στο Χρήστη Πρότυπου Χρήσης.....	198
Σχήμα 68: Προσαρμοσμένη Αρχιτεκτονική για την υποστήριξη σύνθεσης υπηρεσιών με εκμετάλλευση Προτύπων Χρήσης.....	200

ΛΙΣΤΑ ΠΙΝΑΚΩΝ

Πίνακας 1: Η λειτουργικότητα της Πύλης Θέσης (Αντιστοίχιση μεθόδων OSA/Parlay σε μηνύματα του MLP και αντιστρόφως).....	51
Πίνακας 2: Γεγονότα ενημέρωσης διδύμων υποσυστημάτων.....	150

Σχήματα

Πίνακας 3: Γεγονότα εγγραφής και ενημέρωσης τοπικών υποσυστημάτων	151
Πίνακας 4: Οι μετρούμενες τιμές – Υλοποίηση με τη διεπαφή OSA/Parlay.....	171
Πίνακας 5: Οι μετρούμενες τιμές – Υλοποίηση με χρήση Διαμεσολαβητών.....	172
Πίνακας 6: Οι Χρόνοι t_M και t_N	177
Πίνακας 7: Οι Χρόνοι t_C και t_G	177

1ο Κεφάλαιο

Εισαγωγή – Επισκόπηση υπηρεσιών βασισμένων στο πλαίσιο χρήσης τους

Το εισαγωγικό κεφάλαιο αποτελεί μία συνολική επισκόπηση σχετική με την παροχή υπηρεσιών βασισμένων στο πλαίσιο χρήσης τους. Καταγράφονται οι έως τώρα ερευνητικές προσπάθειες για τον ορισμό και τον καθορισμό των στοιχείων που αποτελούν το πλαίσιο χρήσης μίας υπηρεσίας καθώς και οι διάφορες αρχιτεκτονικές που υποστηρίζουν την παροχή τέτοιου είδους υπηρεσιών. Στη συνέχεια περιγράφεται μία πρώιμη προσπάθεια υλοποίησης πλατφόρμας παροχής υπηρεσιών αναζήτησης βασισμένων στη θέση και στις προτιμήσεις του χρήστη καθώς και τα προβλήματα και οι ελλείψεις που παρουσιάστηκαν. Από την αποτίμηση αυτής της εργασίας προκύπτουν καιρικά συμπεράσματα, τα οποία αποτελούν στη συνέχεια τις κατευθυντήριες γραμμές για τον σχεδιασμό και την υλοποίηση του προτεινόμενου συστήματος. Τέλος, παρουσιάζεται συνοπτικά η δομή των υπολοίπων κεφαλαίων.

1.1 Εισαγωγή

Με την πάροδο του χρόνου, την αύξηση των αναγκών των χρηστών και την εξέλιξη της τεχνολογίας δημιουργήθηκαν οι κατάλληλες προϋποθέσεις για την παροχή πιο προηγμένων και πολύπλοκων υπηρεσιών, όπως οι υπηρεσίες με επίγνωση του πλαισίου χρήσης τους. Ωστόσο τέτοιου είδους υπηρεσίες απαιτούν προκειμένου να εκτελεστούν εξειδικευμένες ρυθμίσεις και πληροφορίες, τις οποίες καλείται να πραγματοποιεί και να παρέχει ο χρήστης, εφόσον το μοντέλο παροχής υπηρεσιών παραμένει με τη μορφή που εμφανίζεται στις περιπτώσεις απλών υπηρεσιών και δεν παρέχει αυτοματοποιημένες διαδικασίες. Το γεγονός αυτό κατά συνέπεια εισάγει ένα μεγάλο όγκο πολύπλοκων διαδικασιών και προϋποθέτει κατάλληλες γνώσεις, διαθέσιμο χρόνο και διάθεση από την πλευρά του χρήστη ώστε να τις πραγματοποιήσει. Η απαίτηση αυτή αυτομάτως μειώνει την γκάμα των υποψηφίων χρηστών και δρα σαν τροχοπέδη για την ανάπτυξή και την διείσδυση νέων προηγμένων υπηρεσιών στην αγορά. Για τον λόγο αυτό δημιουργήθηκε η ανάγκη αυτοματοποίησης των διαδικασιών που απαιτούνται προκειμένου να παρασχεθεί μία τέτοιου είδους υπηρεσία. Έτσι, από τις αρχές ακόμα της δεκαετίας του '90, ξεκίνησαν οι πρώτες ερευνητικές προσπάθειες για τον σχεδιασμό και την υλοποίηση εφαρμογών με επίγνωση του πλαισίου χρήσης τους που δεν απαιτούν την εμπλοκή του χρήστη. Οι πρώτες προσπάθειες ήταν περιορισμένης εμβέλειας και παρείχαν την υποδομή για την παροχή συγκεκριμένων υπηρεσιών (application specific) σε συγκεκριμένα πεδία (domain specific). Τα βήματα που ακολούθησαν οδήγησαν τις ερευνητικές προσπάθειες σε πιο γενικευμένες αρχιτεκτονικές, οι οποίες σαν στόχο είχαν να δημιουργήσουν υποδομές οι οποίες θα μπορούσαν να υποστηρίξουν όλες τις απαιτούμενες διαδικασίες ανεξαρτήτως εφαρμογής. Ωστόσο και οι προσπάθειες αυτές δεν αποτέλεσαν πανάκεια στο υπό συζήτηση πρόβλημα.

Βασική προϋπόθεση για την επιτυχημένη πραγματοποίηση ενός τέτοιου εγχειρήματος, που θέτει ουσιαστικά υψηλούς στόχους διαλειτουργικότητας, είναι η ύπαρξη κοινών σημείων αναφοράς. Στην προκειμένη περίπτωση, τα κοινά σημεία αναφοράς είναι ο κοινός ορισμός και καθορισμός της σημασίας του πλαισίου χρήσης μίας υπηρεσίας, ποια στοιχεία καθορίζουν το πλαίσιο αυτό, τι σημαίνει υπηρεσία με επίγνωση του πλαισίου χρήσης της και ποιες είναι οι διαδικασίες που μία τέτοια υπηρεσία οφείλει να εκτελεί. Εν συνεχεία απαιτείται ο σχεδιασμός και η υλοποίηση ενός συστήματος βασισμένου στα κοινά αυτά σημεία αναφοράς, το οποίο να δύναται να ανακτήσει και να διαχειριστεί την πληροφορία πλαισίου και φυσικά να τροφοδοτεί με αυτή, με ομοιόμορφο και προτυποποιημένο τρόπο, τις υπηρεσίες.

1.2 Διάρθρωση της διατριβής

Στο σημείο αυτό, παρατίθεται η διάρθρωση των ενοτήτων που ακολουθούν. Στην συνέχεια του κεφαλαίου αυτού παρουσιάζεται μία αναδρομή στις ερευνητικές προσπάθειες που πραγματοποιήθηκαν μέχρι σήμερα, όσον αφορά στα προαναφερθέντα ζητήματα. Εν συνεχεία παρατίθεται μία πρώτη προσπάθεια που πραγματοποιήθηκε στο πλαίσιο της παρούσας διατριβής για την υλοποίηση μίας πλατφόρμας παροχής υπηρεσιών αναζήτησης βάσει θέσης και χαρακτηριστικών του χρήστη. Η προσπάθεια αυτή κατατάσσεται στην πρώτη κατηγορία των πρώιμων context aware υπηρεσιών. Τα προβλήματα και οι ελλείψεις που παρουσιάστηκαν αποτέλεσαν στη συνέχεια τις κατευθυντήριες γραμμές για τον σχεδιασμό και την υλοποίηση του προτεινόμενου συστήματος.

Στο δεύτερο κεφάλαιο, παρατίθεται μία επισκόπηση των προσεγγίσεων μοντελοποίησης της πληροφορίας πλαισίου χρήσης υπηρεσίας και περιγράφονται τα μειονεκτήματα και τα πλεονεκτήματα της καθεμιάς. Εν συνεχεία περιγράφεται η προσέγγιση που ακολουθήθηκε στο πλαίσιο της παρούσας διατριβής, η οποία πραγματοποιήθηκε σε δύο στάδια. Το πρώτο ήταν αυτό του καθορισμού του μοντέλου πληροφορίας των εμπλεκόμενων οντοτήτων με χρήση του προτύπου Common Information Model της Distributed Management Task Force και το δεύτερο ο καθορισμός του μοντέλου δεδομένων όπου ακολουθήθηκε markup-scheme προσέγγιση ακολουθώντας το πρότυπο Generic User Profile της 3GPP, καθώς και άλλα τέτοιου είδους πρότυπα ανάλογα με τις κάθε φορά προκύπτουσες ανάγκες.

Στο τρίτο κεφάλαιο, περιγράφεται η προτυποποιημένη αρχιτεκτονική που υλοποιήθηκε για την υποστήριξη παροχής υπηρεσιών προσαρμοσμένων στον εκάστοτε χρήστη και την τρέχουσα κατάστασή του. Στο πρώτο μέρος του κεφαλαίου παρουσιάζονται οι βασικές αρχές που ακολουθήθηκαν για τον σχεδιασμό του συστήματος καθώς και τα βασικά δομικά του στοιχεία, καταλήγοντας τελικά στη συνολική του αρχιτεκτονική. Εν συνεχεία, ιδιαίτερη βαρύτητα δίνεται στην περιγραφή της λειτουργικότητας και της υλοποίησης των επιμέρους εμπλεκόμενων οντοτήτων και των μηχανισμών επικοινωνίας τους. Τέλος, αναλύονται διεξοδικά οι βασικές αλληλεπιδράσεις και λειτουργίες του συστήματος και περιγράφονται συνοπτικά οι υπηρεσίες που υλοποιήθηκαν για την επιβεβαίωση της ορθής λειτουργικότητας και την επαλήθευση των δυνατοτήτων του συστήματος.

Το τέταρτο κεφάλαιο αρχικά αναλύει και περιγράφει την ανάπτυξη μίας ήδη υπάρχουσας εφαρμογής αναζήτησης επάνω από το προτεινόμενο σύστημα, καθώς και τις αλλαγές και προσθήκες που πραγματοποιήθηκαν για αυτό το εγχείρημα. Ακολούθως, παρουσιάζεται η αποτίμηση της απόδοσης του συστήματος κατά την παροχή της συγκεκριμένης υπηρεσίας, η

οποία απαιτεί σαν κύρια πληροφορία πλαισίου χρήσης την πληροφορία της θέσης του χρήστη. Η αποτίμηση πραγματοποιήθηκε σε σχέση με την υλοποίηση της υπηρεσίας με χρήση της διεπαφής Κινητικότητας του OSA/Parlay πάνω από τρεις διαφορετικές τεχνολογίες μεσισμικού καταναεμημένων αντικειμένων. Η απόδοση των προτεινόμενων λύσεων συγκρίνεται και με την αντίστοιχη της τρέχουσας υλοποίησης που προτείνεται από τον οργανισμό προτυποποίησης της 3GPP.

Τέλος, στο πέμπτο κεφάλαιο, παρατίθεται μία επισκόπηση των όσων παρουσιάστηκαν στα κυρίως κεφάλαια της διατριβής. Στόχος της επισκόπησης αυτής είναι η επιβεβαίωση της επίτευξης των στόχων της εργασίας σε σχέση με τις αρχικές απαιτήσεις που είχαν τεθεί. Επίσης παρουσιάζεται ένα πιθανό σενάριο για την ανάπτυξη του συστήματος σε μεγάλη κλίμακα, καθώς και ένα πιθανό επιχειρηματικό μοντέλο, το οποίο θα μπορούσε να ακολουθηθεί για την διεύθυνση της προτεινόμενης αρχιτεκτονικής στο χώρο της αγοράς. Η τελευταία ενότητα του κεφαλαίου αποτελεί τις πιθανές και προτεινόμενες μελλοντικές εξελίξεις από τεχνικής πλευράς για την περαιτέρω ανάπτυξη και επέκταση του παρόντος έργου.

1.3 Τι σημαίνει Πλαίσιο Χρήσης Υπηρεσίας (Context)?

Το «Πλαίσιο Χρήσης Υπηρεσίας» (Context), ενώ σαν έννοια είναι γενικά σιωπηρά αποδεκτή και κατανοητή, παρουσιάζει ιδιαίτερη δυσκολία στην περιγραφή της μέσω ενός ορισμού λίγων γραμμών. Οι περισσότερες προσπάθειες καταλήγουν είτε στην χρήση παραδειγμάτων μέσω των οποίων διαφαίνεται η έννοια του context είτε στην χρήση συνωνύμων λέξεων. Για την δημιουργία λοιπόν ενός πλήρους και ολοκληρωμένου ορισμού παρατίθενται στη συνέχεια οι προσπάθειες που παρουσιάζονται στην βιβλιογραφία προς αυτόν τον στόχο.

Στις πρώτες επιστημονικές προσπάθειες που εμφανίζονται στον συγκεκριμένο ερευνητικό χώρο το context ορίζεται ως *η θέση του χρήστη, οι ταυτότητες των ανθρώπων και των αντικειμένων γύρω από τον χρήστη στον συγκεκριμένο τόπο και οι αλλαγές που πραγματοποιούνται σε σχέση με την κατάστασή τους* [1]. Ένας παρόμοιος ορισμός δίνεται στο [2] εμπλουτίζοντας τον προηγούμενο με στοιχεία όπως *η ώρα, η εποχή, η θερμοκρασία* κλπ. Στο [3], το context ορίζεται ως *η θέση και η ταυτότητα του χρήστη, το περιβάλλον του και η ώρα* ενώ στο [4] ως *η συναισθηματική κατάσταση του χρήστη, το που έχει στραμμένη την προσοχή του, η θέση του και ο προσανατολισμός του, η ημερομηνία και η ώρα, τα αντικείμενα και οι άνθρωποι στον περιβάλλοντα χώρο*.

Κοινό χαρακτηριστικό όλων των παραπάνω ορισμών είναι το γεγονός ότι αποτελούν απαριθμήσεις παραδειγμάτων πιθανών στοιχείων του πλαισίου χρήσης μίας υπηρεσίας και αυτό έχει σαν αποτέλεσμα η εφαρμογή τους να είναι δυνατή κατά περίπτωση. Είναι πιθανό σε κάποιον ορισμό να μη συμπεριλαμβάνεται κάποιο είδος πληροφορίας που σε κάποια δεδομένη κατάσταση να αποτελεί στοιχείο context.

Μία δεύτερη ομάδα ορισμών παρουσιάζεται στη συνέχεια, της οποίας το χαρακτηριστικό στοιχείο είναι ότι αποτελείται από μία σειρά συνωνύμων της έννοιας του context όπως για παράδειγμα *περιβάλλον* ή *κατάσταση*. Σε ορισμένες περιπτώσεις σαν context θεωρείται το περιβάλλον του χρήστη και σε άλλες το περιβάλλον της εφαρμογής. Στην δημοσίευση [5], οι συγγραφείς ορίζουν το πλαίσιο χρήσης μίας υπηρεσίας από έναν χρήστη ως *τα στοιχεία του περιβάλλοντος του χρήστη των οποίων την ύπαρξη είναι σε θέση να γνωρίζει το τερματικό του χρήστη*, π.χ. *ο υπολογιστής του*, ενώ στην [6] context θεωρείται γενικά *η κατάσταση του χρήστη*. Επίσης το context έχει κατά καιρούς οριστεί ως *η κατάσταση των όσων περιβάλλουν την εφαρμογή* [7] ή ως *οι ρυθμίσεις της εφαρμογής* [8]. Τέλος ένας ορισμός που περιλαμβάνει συνολικά όλα τα εμπλεκόμενα κατά την παροχή κάποιας υπηρεσίας περιβάλλοντα είναι ο εξής: *ως context ορίζονται οι διάφορες απόψεις (aspects) της γενικότερης τρέχουσας κατάστασης* [9]. Όπως και στην προηγούμενη περίπτωση ορισμών έτσι και σε αυτή δεν συναντάμε έναν πλήρη και συνολικό ορισμό ο οποίος να είναι δυνατόν εφαρμοσθεί ή να επικαλεστεί στο σύνολο των περιπτώσεων παροχής υπηρεσιών βασισμένων στο πλαίσιο χρήσης τους.

Ωστόσο έχουν υπάρξει και προσπάθειες πιο συνολικών και εφαρμόσιμων ορισμών, οι οποίοι όμως μοιάζουν περισσότερο με προσπάθειες κατηγοριοποίησης των ειδών πληροφορίας που αποτελούν το context. Από τους ορισμούς αυτούς προκύπτει ότι τα σημαντικότερα στοιχεία του context είναι η θέση του χρήστη, η ταυτότητά του καθώς και η ταυτότητα των γύρω προσώπων, καθώς και οι περιβάλλοντες υπολογιστικοί πόροι (resources) [10]. Πιο συνολικά το context αποτελείται από τη φυσική, κοινωνική, συναισθηματική και υπολογιστική κατάσταση του χρήστη [11]. Τέλος σε μία πιο γενική προσέγγιση σαν context θεωρείται το υποσύνολο των φυσικών και μη καταστάσεων (physical & conceptual states) μίας οντότητας το οποίο επηρεάζει και σχετίζεται άμεσα με την εκτέλεση και παροχή μίας υπηρεσίας [12]. Ακόμα όμως και οι τελευταίοι ορισμοί, στην προσπάθειά τους να είναι γενικοί και να περιλάβουν όλα τα πιθανά είδη πληροφορίας, έγιναν υπερπλήρης και υπερβολικά συγκεκριμένοι. Αυτό έχει ως αποτέλεσμα να υπάρχουν και περιπτώσεις που πληροφορία που θεωρείται και χαρακτηρίζεται σαν πληροφορία context βάσει αυτών των ορισμών να μην είναι απαραίτητη για την παροχή κάποιας υπηρεσίας βασισμένης στο πλαίσιο χρήσης

της ή να μην αποτελεί σημαντικό στοιχείο για τον συγκεκριμένο χρήστη της υπηρεσίας και έτσι αυτομάτως να ακυρώνεται ο ορισμός.

Λαμβάνοντας υπόψη και συνδυάζοντας τους προαναφερθέντες ορισμούς προκύπτει ο παρακάτω:

«Ως context δύναται να θεωρηθεί οποιαδήποτε πληροφορία χαρακτηρίζει την κατάσταση μίας οντότητας. Ως οντότητες εννοούνται πρόσωπα, τόποι ή αντικείμενα που σχετίζονται με την αλληλεπίδραση του χρήστη με μία υπηρεσία, συμπεριλαμβανομένων του ίδιου του χρήστη και της υπηρεσίας».

Ο ορισμός αυτός είναι πιο γενικός αλλά και πιο ευέλικτος και διαμορφώνεται κατάλληλα ανάλογα με τις περιστάσεις. Με τη χρήση του ορισμού αυτού δίνεται η δυνατότητα στον πάροχο μίας υπηρεσίας να καθορίζει το context ανάλογα με τις ανάγκες της υπηρεσίας του. Εάν κάποια πληροφορία χαρακτηρίζει την κατάσταση των εμπλεκόμενων οντοτήτων τότε αυτή η πληροφορία αποτελεί μέρος του context.

1.3.1 Κατηγορίες Πληροφορίας Πλαισίου Χρήσης Υπηρεσίας

Έχοντας ολοκληρώσει με τον ορισμό του context, παρατίθενται οι αντίστοιχες προσπάθειες για τον καθορισμό κατηγοριοποίησης των πληροφοριών που το συνθέτουν. Οι Ryan et al. [3] προτείνουν σαν βασικές κατηγορίες context τη θέση, το περιβάλλον, την ταυτότητα και το χρόνο. Οι Schilit et al. [10] θεωρούν πως στην ερώτηση ποιοι είναι οι βασικοί τύποι του context απαντούν οι εξής ερωτήσεις: «πού είσαι, με ποιον είσαι και τι υπολογιστικοί πόροι υπάρχουν στον χώρο». Η τελευταία κατηγοριοποίηση ουσιαστικά περιλαμβάνει μόνο πληροφορίες θέσης και ταυτότητας, ενώ η προηγούμενη περιλαμβάνει τη λέξη “περιβάλλον” η οποία ουσιαστικά αποτελεί συνώνυμο του context. Τροποποιώντας λοιπόν τον πρώτο ορισμό και αντικαθιστώντας τη λέξη “περιβάλλον” με τη λέξη “δραστηριότητα” καταλήγουμε ότι οι βασικές και πιο σημαντικές κατηγορίες πληροφορίας context είναι οι εξής:

- Πληροφορίες Θέσης
- Πληροφορίες Ταυτότητας
- Πληροφορίες Δραστηριοτήτων
- Χρόνος

Αυτές ουσιαστικά είναι οι κατηγορίες που αποτελούν το πρωτεύον context και μπορούν να χαρακτηρίσουν την κατάσταση μίας οντότητας. Από τον συνδυασμό αυτών των πληροφοριών μπορεί να προκύψουν και περαιτέρω δευτερεύουσες πληροφορίες context. Για

παράδειγμα, γνωρίζοντας την ταυτότητα του χρήστη είναι δυνατή η ανακάλυψη και άλλων στοιχείων όπως το τηλέφωνό του από κάποια υπηρεσία τηλεφωνικού καταλόγου, κλπ.

Μία ακόμα προσπάθεια κατηγοριοποίησης είναι η παρακάτω [10] η οποία διαχωρίζει το context σε:

- Πληροφορίες Υπολογιστικού Περιβάλλοντος (computational context): διαθέσιμοι επεξεργαστές, προσβάσιμες από τον χρήστη τερματικές συσκευές, δυνατότητες τερματικών συσκευών, οθόνες, διαθέσιμα δίκτυα, δυνατότητες σύνδεσης στο δίκτυο, κόστη κλπ.
- Πληροφορίες Χρήστη (user context): θέση, ταυτότητα, προσωπικά δεδομένα, προτιμήσεις, ενδιαφέροντα, πρόγραμμα, πρόσωπα στο τρέχον περιβάλλον, κοινωνική κατάσταση κλπ.
- Πληροφορίες Φυσικών Συνθηκών (physical context): φωτισμός, θερμοκρασία, θόρυβος κλπ.

Αντίστοιχα οι Schmidt, Beigl and Gellersen [27] χωρίζουν τα είδη της πληροφορίας context σε «ανθρώπινους παράγοντες» και «φυσικό περιβάλλον», οι οποίοι σε δεύτερο επίπεδο αποτελούνται από «πληροφορίες χρήστη, κοινωνικό περιβάλλον, δραστηριότητα» και «θέση, υποδομή και φυσικές συνθήκες» αντίστοιχα.

1.4 Τι σημαίνει Επίγνωση Πλαισίου Χρήσης Υπηρεσίας (Context-Awareness)?

Επίγνωση Πλαισίου Χρήσης Υπηρεσίας (Context-awareness) θεωρήθηκε αρχικά ότι διαθέτει το λογισμικό το οποίο προσαρμόζεται ανάλογα με την τοποθεσία όπου χρησιμοποιείται, τα πρόσωπα και αντικείμενα που είναι παρόντα και με τις αλλαγές που συμβαίνουν στα αντικείμενα αυτά με την πάροδο του χρόνου [1]. Ο ορισμός αυτός ωστόσο περιορίζεται στις εφαρμογές οι οποίες προσαρμόζονται κατάλληλα ανάλογα με το εκάστοτε context. Έτσι, η έννοια της επίγνωσης του context ταυτίζεται με έννοιες όπως προσαρμοζόμενος (adaptive) [14], αντιδρών (reactive) [15], αποκριτικός (responsive) [16], ευαίσθητος στο context (context-sensitive) [17], κατευθυνόμενος από το περιβάλλον (environment directed) [18], γεγονός που έρχεται σε αντίθεση με την κατηγοριοποίηση των υπηρεσιών με επίγνωση πλαισίου χρήσης σε εκείνες που προσαρμόζονται με βάση το context αλλά και σε εκείνες που απλώς ενημερώνονται γι' αυτό, το παρέχουν ή/και το χρησιμοποιούν χωρίς να μεταβάλλεται η λογική τους.

Μερικοί από του ορισμούς για την επίγνωση πλαισίου χρήσης υπηρεσίας παρατίθενται στη συνέχεια:

Επίγνωση πλαισίου χρήσης υπηρεσίας είναι:

- η δυνατότητα ανίχνευσης, ανάκτησης, μετάφρασης και απόκρισης στα δεδομένα του περιβάλλοντος του χρήστη και των τερματικών του συσκευών [9] [3][12][19].
- οι διαδικασίες που απαιτούνται για την αυτοματοποίηση βάσει του context του χρήστη ενός συστήματος λογισμικού [11].
- η δυνατότητα μέγιστης ευελιξίας και προσαρμοστικότητας μίας υπηρεσίας βάσει πληροφορίας context πραγματικού χρόνου [20].

Επιπροσθέτως, οι παρακάτω ορισμοί περιγράφουν υπηρεσίες με δυνατότητες προσαρμογής ανάλογα με το πλαίσιο εντός του οποίου χρησιμοποιούνται. Υπηρεσίες με επίγνωση πλαισίου χρήσης τους είναι αυτές που:

- δυναμικά αλλάζουν ή προσαρμόζουν την συμπεριφορά/λογική τους βάσει της πληροφορίας context [2][7] [10][21][22][23][24].
- παρακολουθούν και λαμβάνουν δεδομένα από περιβαλλοντικούς αισθητήρες και επιτρέπουν στον χρήστη να επιλέγει ανάλογα με τις τρέχουσες δραστηριότητες και τα ενδιαφέροντά του το πλαίσιο χρήσης της υπηρεσίας [25].
- αυτόματα παρέχουν πληροφορία ή πραγματοποιούν κάποια ενέργεια βασιζόμενες στο τρέχον context του χρήστη όπως αυτό ανακτάται από αισθητήρες [26].
- Παρακολουθούν τις αλλαγές στο περιβάλλον και τροποποιούνται βάσει προκαθορισμένων ή καθορισμένων από το χρήστη κανόνων [18].

Από τους παραπάνω λοιπόν ορισμούς εξάγεται το συμπέρασμα ότι ένα σύστημα μπορεί να χαρακτηριστεί σαν context-aware εφόσον διαθέτει υποχρεωτικά τα παρακάτω χαρακτηριστικά:

- Αισθάνεται και ανακτά την σχετική πληροφορία κυρίως σε πραγματικό χρόνο
- Επεξεργάζεται την σχετική πληροφορία
- Και αναλόγως μεταβάλλει τη λειτουργία του.

Ας θεωρήσουμε όμως μία υπηρεσία αναζήτησης πληροφοριών βασισμένων στη θέση του χρήστη. Η υπηρεσία αυτή έχει σχεδιαστεί και υλοποιηθεί με τέτοιο τρόπο, ώστε λαμβάνοντας σαν είσοδο την τρέχουσα θέση του χρήστη να παρέχει σε αυτόν διάφορες πληροφορίες. Η υπηρεσία ωστόσο δεν υλοποιεί η ίδια τον μηχανισμό ανάκτησης της θέσης, αλλά της τον παρέχει κάποια εξωτερική οντότητα. Επίσης η λογική της υπηρεσίας δεν αλλάζει ανεξαρτήτως της θέσης του χρήστη, απλώς μεταβάλλεται το αποτέλεσμα που

επιστρέφει ανάλογα με αυτή. Η υπηρεσία λοιπόν δεν συμμορφώνεται με τον παραπάνω ορισμό και κατά συνέπεια δεν είναι context-aware, συμπέρασμα το οποίο είναι άτοπο. Συνεπώς υπάρχει ανάγκη για έναν πιο γενικευμένο ορισμό, που να μπορεί να καλύπτει όλο το φάσμα των context-aware συστημάτων και υπηρεσιών:

«Θεωρείται ότι ένα σύστημα, και κατά συνέπεια και μία υπηρεσία ή εφαρμογή, έχουν επίγνωση του πλαισίου εντός του οποίου χρησιμοποιούνται, εφόσον χρησιμοποιούν την πληροφορία που περιγράφει αυτό το πλαίσιο για να παρέχουν σχετικές πληροφορίες ή/και υπηρεσίες στον χρήστη.»

1.5 Εφαρμογές και Αρχιτεκτονικές με Επίγνωση Πλαισίου Χρήσης Υπηρεσίας

Στην ενότητα αυτή παρουσιάζονται εν συντομία προηγούμενες εργασίες στο πεδίο των εφαρμογών και αρχιτεκτονικών με επίγνωση του πλαισίου χρήσης τους. Αρχικά περιγράφονται οι πρώτες και παλαιότερες προσπάθειες σχεδιασμού και υλοποίησης απλών σχετικά εφαρμογών που κάνουν χρήση περιορισμένων στοιχείων του context. Η ενότητα αυτή ολοκληρώνεται με την παρουσίαση πιο ολοκληρωμένων και πρόσφατων έργων που αντιμετωπίζουν συνολικά τη συλλογή και παροχή του context στις υπηρεσίες.

1.5.1 Εφαρμογές με Επίγνωση Πλαισίου Χρήσης τους

Call Forwarding [13]: Η εφαρμογή αυτή παρείχε μέσω του συστήματος Active Badge τη θέση του χρήστη και βάσει αυτής πραγματοποιούνταν δρομολόγηση, αρχικά χειροκίνητη και στη συνέχεια ψηφιοποιημένη, των τηλεφωνημάτων του στο κοντινότερο του τηλέφωνο. Ωστόσο η συγκεκριμένη εφαρμογή δεν διέθετε την ευφυΐα ώστε σε περιπτώσεις που ο χρήστης ήταν απασχολημένος, π.χ. σε κάποια συνάντηση, να μην δρομολογεί καθόλου την κλήση προς αυτόν και να ενημερώνει τον καλούντα προκειμένου να αφήσει κάποιο μήνυμα ή να ξανακαλέσει αργότερα.

Teleporting ή FollowMe Computing [28]: Αποτελεί εργαλείο που βασίζεται στο σύστημα Active Badge και μεταφέρει το υπολογιστικό περιβάλλον του χρήστη στο πιο κοντινό του τερματικό. Νεότερη εκδοχή του συστήματος χρησιμοποιεί για τον εντοπισμό της θέσης του χρήστη το σύστημα Bat3 [29].

Active Map [30][31][32]: Πρόκειται για εντοπισμό προσώπων μέσω του συστήματος PARC Tab και απεικόνιση τους σε χάρτη με βάση τον αριθμό του δωματίου στο οποίο βρίσκονται.

Shopping Assistant [33]: Η συσκευή αυτή, χρησιμοποιώντας πληροφορίες θέσης, καθοδηγεί τους πελάτες ενός καταστήματος μέσα σε αυτό, παρέχει πληροφορίες για τα προϊόντα, τους βοηθάει να εντοπίσουν προϊόντα που αναζητούν, τονίζει τυχόν προσφορές, παρέχει συγκριτικά τιμών κλπ.

Cyberguide [34][35]: Το σύστημα αυτό παρέχει βάσει θέσης (είτε σε εξωτερικό είτε σε εσωτερικό χώρο) υπηρεσίες πληροφοριών σε τουρίστες, όπως κατευθύνσεις, χάρτες, προτάσεις για αξιοθέατα κλπ. Το σύστημα διατηρεί και αρχείο με προηγούμενα ταξίδια του χρήστη και βάσει αυτών πραγματοποιεί τις προτάσεις του. Αντίστοιχο σύστημα είναι και το GUIDE [36][37]. Επίσης υπάρχουν παρόμοια συστήματα μικρότερης κλίμακας για επισκέπτες μουσείων [38][39] ή εκθέσεων [40].

Conference Assistant [41]: Η εφαρμογή αυτή χρησιμοποιώντας στοιχεία context - όπως η θέση και τα ενδιαφέροντα των συμμετεχόντων του συνεδρίου, η ώρα και το πρόγραμμα των παρουσιάσεων - παρέχει στους συμμετέχοντες πληροφορίες και τους προτείνει πιθανές παρουσιάσεις προς παρακολούθηση. Όταν κάποιος συμμετέχων εισέλθει στον χώρο που πραγματοποιείται κάποια παρουσίαση, η εφαρμογή του παρουσιάζει το όνομα του ομιλητή, τον τίτλο της ομιλίας κλπ. Επίσης πραγματοποιείται αυτόματη καταγραφή της ομιλίας προκειμένου οι χρήστες να έχουν την δυνατότητα να την ανακτήσουν εκ των υστέρων.

Adaptive GSM phone and PDA [42]: Στην περίπτωση του κινητού τηλεφώνου πραγματοποιείται αλλαγή του προφίλ προκειμένου το κινητό ανάλογα με τη θέση του (στο χέρι του χρήστη, πάνω στο τραπέζι, στην τσάντα, σε εξωτερικό χώρο) να χτυπάει, να δονείται, να προσαρμόζει κατάλληλα την ένταση του ήχου ή να μπαίνει σε αθόρυβη λειτουργία. Στην περίπτωση του PDA, εφαρμογή κειμενογράφου (notepad application) ρυθμίζει το μέγεθος της γραμματοσειράς ανάλογα με την δραστηριότητα του χρήστη (μεγάλο μέγεθος αν ο χρήστης κινείται, μικρότερο αν όχι) καθώς και ανάλογα με τις φυσικές συνθήκες (π.χ. χαμηλός φωτισμός). Αντίστοιχη προσπάθεια αποτελεί και το SenSay Phone [43].

Office Assistant [44]: Ο Βοηθός Γραφείου είναι ένας πράκτορας (agent) που αλληλεπιδρά με τους επισκέπτες στην είσοδο κάθε γραφείου και διαχειρίζεται το πρόγραμμα του ιδιοκτήτη του γραφείου και επισκεπτόμενου. Ο βοηθός ενεργοποιείται όταν πλησιάζει κάποιος επισκέπτης, ο οποίος ανιχνεύεται μέσω της πίεσης που ασκεί σε ειδικά χαλάκια στρωμένα στις πόρτες των γραφείων. Ο βοηθός ανακτά τα στοιχεία της ταυτότητας του

επισκέπτη ρωτώντας τον και ανάλογα με το πρόγραμμα του επισκεπτόμενου πραγματοποιεί τις κατάλληλες ενέργειες προκειμένου να πραγματοποιηθεί ή όχι η συνάντηση, ενημερώνοντας επισκέπτη και επισκεπτόμενο. Αντίστοιχες εφαρμογές αποτελούν το Active Floor [45] και το Smart Floor [46].

Location-aware Information Delivery [47]: Το ερευνητικό πρόγραμμα ComMotion που πραγματοποιήθηκε στο πανεπιστήμιο του MIT κάνει χρήση της θέσης και του χρόνου προκειμένου να αποστέλλει φωνητικά μηνύματα υπενθυμίσεων. Ο CybreMinder [48] αποτελεί εμπλουτισμένη έκδοση του εργαλείου καθώς λαμβάνει υπόψη πιο πολύπλοκα στοιχεία context, όπως τα πρόσωπα που περιτριγυρίζουν τον χρήστη και τις καιρικές συνθήκες. Επίσης, η συσκευή MemoClip [49] αποτελεί άλλη μία τέτοιου είδους προσπάθεια.

MobileWard [50]: Ο στόχος του MobileWard είναι να βοηθά το προσωπικό μίας πτέρυγας ενός νοσοκομείου όπως π.χ. τις νοσοκόμες, στην πραγματοποίηση των καθημερινών τους καθηκόντων. Το σύστημα ενημερώνεται για τη θέση του χρήστη και με την είσοδο του στο δωμάτιο του ασθενή παρέχει αυτομάτως μέσω ενός PDA όλες τις απαραίτητες πληροφορίες για το ιστορικό του καθώς και τη δυνατότητα για την εισαγωγή νέων πληροφοριών. Όταν το σύστημα εντοπίζει τη νοσοκόμα στον διάδρομο της πτέρυγας της παρέχει συνολική πληροφορία για όλη την πτέρυγα. Περισσότερες εφαρμογές σε νοσοκομειακό περιβάλλον παρουσιάζονται στο [51].

Η παραπάνω λίστα αποτελεί μία επιλεκτική παρουσίαση ορισμένων εφαρμογών με επίγνωση του πλαισίου χρήσης τους. Οι εφαρμογές αυτές αποτελούν τις πρώτες προσπάθειες για υλοποίηση τέτοιου είδους συστημάτων και φέρουν τα εξής χαρακτηριστικά:

- Οι περισσότερες συλλέγουν και εκμεταλλεύονται περιορισμένο αριθμό ειδών πληροφορίας context, συνήθως μόνο τη θέση του χρήστη.
- Έχουν σχεδιαστεί και υλοποιηθεί με σκοπό να καλύψουν τις ανάγκες ενός συγκεκριμένου σεναρίου μέσω της συγκεκριμένης εφαρμογής. Κατά συνέπεια αποτελούν κλειστά, περιορισμένης εμβέλειας και μη επεκτάσιμα συστήματα, των οποίων η υποδομή δεν μπορεί να χρησιμοποιηθεί για να την παροχή νέων υπηρεσιών.

Με την πάροδο όμως του χρόνου γεννήθηκε η ανάγκη για πιο γενικευμένα συστήματα και σταδιακά η ερευνητική κοινότητα στράφηκε στη μελέτη συνολικών αρχιτεκτονικών και

πλαισίων για την υποστήριξη παροχής υπηρεσιών με επίγνωση του πλαισίου χρήσης τους ανεξαρτήτως του είδους και της φύσης τους. Τα συστήματα αυτά παρέχουν επαναχρησιμοποιήσιμες δομικές μονάδες για την ανάκτηση, μετάφραση, σύνθεση και παροχή του context στις εφαρμογές και τις κατάλληλες διεπαφές για την πρόσβαση στην πληροφορία κρύβοντας τις λεπτομέρειες των διαδικασιών που εκτελούν. Στην επόμενη ενότητα παρουσιάζονται οι πιο σημαντικές προσπάθειες.

1.5.2 Αρχιτεκτονικές και Συστήματα παροχής υπηρεσιών με Επίγνωση Πλαισίου Χρήσης τους

Η ποικιλία των πληροφοριών context και των πηγών τους, η πληθώρα των διαφορετικών τεχνολογιών και προσεγγίσεων για την ανάκτηση και τη δόμηση τους, καθώς και η διαφορετικότητα των εφαρμογών που πρέπει να υποστηρίζονται οδήγησαν σε λύσεις μεμονωμένες, περιορισμένης εμβέλειας που αντιμετώπιζαν κάθε φορά ένα συγκεκριμένο είδος εφαρμογής αξιοποιώντας συγκεκριμένα είδη πληροφορίας context και χωρίς προοπτικές εξέλιξης. Η πολυπλοκότητα αυτή λοιπόν και η αδυναμία να δοθεί μία ολοκληρωμένη και γενική λύση αποτέλεσαν τροχοπέδη στην ανάπτυξη και ευρεία διάδοση των context-aware υπηρεσιών.

Ο τρόπος που αντιμετωπίστηκε το πρόβλημα αυτό από την ερευνητική κοινότητα ήταν ο αποσυσχετισμός των διαδικασιών ανάκτησης και δόμησης του context σε κατανοητή από τις εφαρμογές μορφή από την παροχή και κατανάλωσή του από αυτές καθώς και από την προσαρμογή των εφαρμογών βάσει αυτού. Για την πραγματοποίηση του στόχου αυτού είναι απαραίτητη η εισαγωγή ενός στρώματος μεσισμικού (middleware) ανάμεσα στις πηγές του context και στις εφαρμογές το οποίο αφ' ενός θα συλλέγει την απαραίτητη πληροφορία και θα την δομεί κατάλληλα και αφ' ετέρου θα παρέχει τις κατάλληλες προγραμματιστικές διεπαφές προκειμένου οι εφαρμογές ανεξαρτήτως είδους, λογικής και τρόπου υλοποίησης να μπορούν να ανακτούν το context.

Στη συνέχεια παρουσιάζονται συνοπτικά κάποιες από τις σημαντικότερες προσπάθειες προς αυτήν την κατεύθυνση.

Το *Context Managing Framework (CMF)* [52][53] αποτελεί ένα πλαίσιο λογισμικού για κινητά τερματικά, το οποίο παρέχει λειτουργίες ανάκτησης και επεξεργασίας πληροφοριών context και παροχή τους στις εφαρμογές. Η διαχείριση της πληροφορίας βασίζεται σε οντολογία ορισμένη μέσω RDF (Resource Description Framework), η οποία καθορίζει τη δομή και τις ιδιότητες των στοιχείων που αποτελούν το context. Η αρχιτεκτονική

αποτελείται από τις εξής βασικές οντότητες: τον context manager, τον resource server, την context recognition service και την change detection service.

Ο context manager, οι resource servers και οι εφαρμογές εκτελούνται εντός του κινητού τερματικού ενώ οι υπόλοιπες οντότητες μπορούν να είναι είτε τοπικές είτε κατανεμημένες. Ο context manager βασίζεται στην λογική των blackboard συστημάτων [54], όσον αφορά στην επικοινωνία με τις υπόλοιπες οντότητες και οι λειτουργίες που πραγματοποιεί είναι αποθήκευση του context και παροχή του βάσει εγγραφών στις εφαρμογές, αλλά και στις οντότητες του συστήματος. Ο resource server συνδέεται με τις πηγές του context και παρέχει στον context manager την πληροφορία που ανακτά. Η recognition service σε συνεργασία με τον resource server μετατρέπει την πληροφορία context σε κατάλληλη μορφή βάσει της οντολογίας του συστήματος. Η change detection service παρέχει εναλλακτικούς τρόπους για την ανίχνευση του context. Το CMF αποτελεί μία πιο ολοκληρωμένη προσπάθεια, παρόλα αυτά παρουσιάζει τα εξής μειονεκτήματα: περιορίζεται σε κινητές τερματικές συσκευές που εν γένει έχουν χαμηλές επεξεργαστικές ικανότητες και η αρχιτεκτονική είναι συγκεντρωτική (centralized) και κατά συνέπεια με μοναδικό σημείο αστοχίας (single point of failure).

Η αρχιτεκτονική *SOCAM (Service-oriented Context-Aware Middleware)* [55] αποτελεί μία αρχιτεκτονική που υποστηρίζει την παροχή context aware services. Βασίζεται επίσης σε οντολογία αλλά σε σύνταξη OWL [56]. Τα βασικά δομικά της στοιχεία είναι:

- ο context provider, ο οποίος ανακτά την πληροφορία context από τις πηγές της και την μετατρέπει σε αναπαραστάσεις OWL ώστε να μπορούν να την κατανοήσουν και να την επεξεργαστούν οι υπόλοιπες οντότητες του συστήματος.
- ο context interpreter ο οποίος αποτελεί έναν κεντρικό εξυπηρετητή που επεξεργάζεται τις πληροφορίες
- η context database στην οποία αποθηκεύονται οι οντολογίες
- και η service-locating service μέσω της οποίας «διαφημίζονται» (advertising) οι context providers και οι context interpreters

Στο υψηλότερο επίπεδο τοποθετούνται οι υπηρεσίες στις οποίες παρέχεται η δυνατότητα να ανακτούν την εκάστοτε απαραίτητη πληροφορία και να προσαρμόζονται ανάλογα.

Μία ακόμα παρόμοια αρχιτεκτονική βασισμένη στο μοντέλο του κεντρικού εξυπηρετητή (centralized server) για την υποστήριξη υπηρεσιών σε PDA και γενικά σε μικρά κινητά τερματικά είναι η *CASS (Context Awareness Sub-Structure)* [57]. Το μεσισμικό αυτό

αποτελείται από έναν Interpreter, έναν ContextRetriever, μία RuleEngine και έναν SensorListener. Ο SensorListener παρακολουθεί τις αλλαγές από τους αισθητήρες και τα δεδομένα που συλλέγει τα αποθηκεύει σε βάσεις (databases) και ο ContextRetriever ανακτά από τη βάση τα δεδομένα. Και οι δύο οντότητες χρησιμοποιούν τις υπηρεσίες που παρέχει ο Interpreter. Ο ChangeListener παρέχει υπηρεσίες επικοινωνίας και επιτρέπει στα κινητά τερματικά να ενημερώνονται για τις αλλαγές του context.

Η *CoBrA (Context Broker Architecture)* [58] αποτελεί μία αρχιτεκτονική βασισμένη στην τεχνολογία των διαμεσολαβητών (agent-based) για την υποστήριξη παροχής υπηρεσιών βασισμένων στο πλαίσιο χρήσης τους σε έξυπνους χώρους (intelligent spaces). Η κεντρική οντότητα του συστήματος είναι ο Context Broker, ο οποίος ανακτά τις πληροφορίες context από ετερογενείς πηγές και τις προσαρμόζει βάσει κατάλληλης οντολογίας προκειμένου να μπορούν να τις χρησιμοποιήσουν οι υπόλοιπες οντότητες του συστήματος, όπως π.χ. εφαρμογές που φιλοξενούνται στα κινητά τερματικά του χρήστη, υπηρεσίες που παρέχονται από συσκευές εντός του έξυπνου χώρου κλπ. Ο Context Broker αποτελείται από τέσσερις λειτουργικές οντότητες: την Context Knowledge, την Base, την Context Inference Engine, το Context Acquisition Module και το Privacy Management Module.

Το *Context Toolkit* [59][60] είναι άλλο ένα παρόμοιο με τα προηγούμενα πλαίσιο, το οποίο όμως βασίζεται σε Peer-to-Peer αρχιτεκτονική διατηρώντας ταυτόχρονα μία κεντρική οντότητα, τον discoverer, στον οποίο εγγράφονται οι καταναμημένοι αισθητήρες προκειμένου να μπορούν να τους ανακαλύψουν οι εφαρμογές.

Το πρόγραμμα *Hydrogen* [61] παρέχει μηχανισμό ανάκτησης context εξειδικευμένο για κινητά τερματικά. Σε αντίθεση με τις περισσότερες αρχιτεκτονικές, το σύστημα αυτό αποφεύγει το συγκεντρωτικό μοντέλο διαχωρίζοντας το context σε τοπικό και απομακρυσμένο. Τοπικό θεωρείται αυτό που ανακτά το τερματικό του χρήστη και απομακρυσμένο αυτό που ανακτά οποιοδήποτε άλλο τερματικό. Τα τερματικά έχουν την δυνατότητα να ανταλλάσσουν πληροφορία context με P2P τρόπο μέσω WLAN, Bluetooth κλπ. όταν βρίσκονται σε μικρή μεταξύ τους απόσταση. Η αρχιτεκτονική αυτή αποτελείται από τρία επίπεδα, τα οποία τοποθετούνται εντός του κινητού τερματικού. Το επίπεδο προσαρμογής (Adaptor layer) που ανακτά την πληροφορία από τους αισθητήρες, το επίπεδο διαχείρισης (management layer) που ανακτά την πληροφορία από το προηγούμενο και την παρέχει. Στο επίπεδο αυτό βρίσκεται ο Context Server, ο οποίος μέσω σύγχρονων και ασύγχρονων μεθόδων παρέχει την πληροφορία στις εφαρμογές. Το υψηλότερο επίπεδο είναι

αυτό των εφαρμογών. Στο επίπεδο αυτό τοποθετείται κατάλληλος κώδικας ο οποίος δρα ανάλογα με τις αλλαγές του context για τις οποίες ενημερώνεται από των context server. Η επικοινωνία μεταξύ των επιπέδων της αρχιτεκτονικής βασίζεται σε πρωτόκολλο XML.

Άλλο ένα παράδειγμα μεσισμικού context-aware για ad-hoc κινητά περιβάλλοντα είναι το σύστημα CORTEX. Βασίστηκε στο Sentient Object Model [62]. Κάθε sentient object αποτελείται από τρία μέρη: το Sensory capture, την Context hierarchy και την Inference engine. Μέσω κατάλληλων διεπαφών κάθε sentient object επικοινωνεί με τους αισθητήρες που παράγουν γεγονότα (events) και με τους actuators που τα καταναλώνουν.

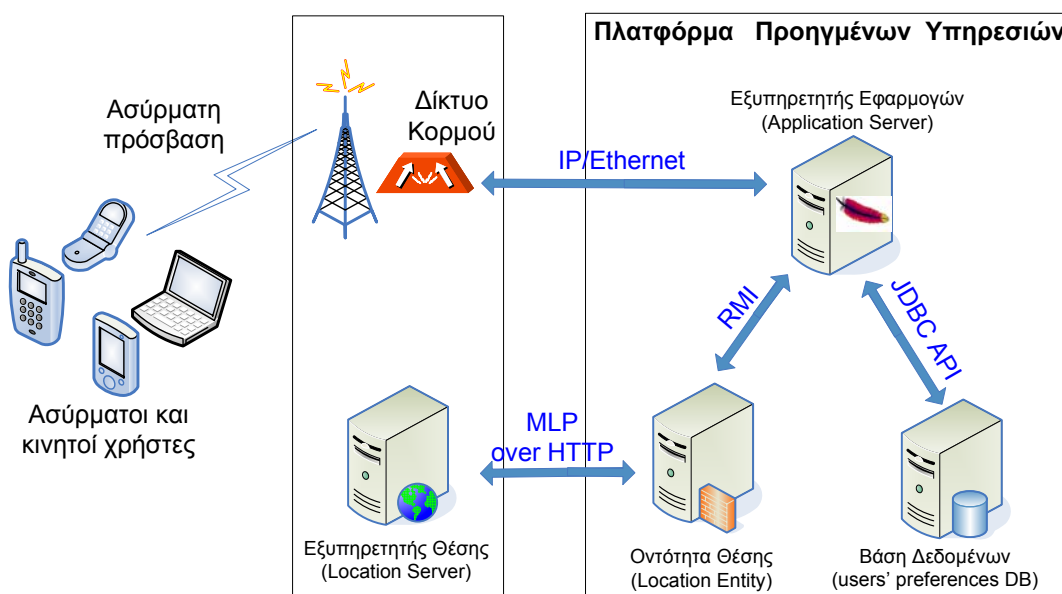
Το σύστημα *Gaia* [63][64] επεκτείνει τις αρχές ενός τυπικού λειτουργικού συστήματος προκειμένου να γίνει context-aware. Υποστηρίζει την ανάπτυξη και εκτέλεση κινητών υπηρεσιών ενεργών χώρων (active spaces). Παρέχει υπηρεσίες αναζήτησης και χρήσης των υπαρκτών πόρων καθώς και πρόσβασης και χρήσης της πληροφορίας context. Οι βασικές οντότητες του συστήματος που σχετίζονται με το context είναι ο Event Manager και η Context Service. Ο Event Manager υλοποιεί τον βασισμένο σε γεγονότα (event-based) μηχανισμό επικοινωνίας, ο οποίος βασίζεται στη λογική των προμηθευτών (suppliers), καταναλωτών (consumers) και καναλιών (channels). Κάθε κανάλι έχει έναν ή περισσότερους προμηθευτές που του παρέχουν πληροφορία μέσω των events και έναν ή περισσότερους καταναλωτές που την καταναλώνουν. Μέσω της Context Service οι εφαρμογές μπορούν να εγγράφονται και να ενημερώνονται για τις πληροφορίες context που ενδιαφέρονται.

Το *Java Context Aware Framework (JCAF)* [65] αποτελεί μία γενική υπηρεσία μεσισμικού διαχείρισης context. Είναι υλοποιημένη ως επεκτάσιμη ιεραρχία κλάσεων. Η επικοινωνία των οντοτήτων του συστήματος είναι P2P και υποστηρίζει μηχανισμό ειδοποιήσεων βασισμένο σε γεγονότα (event based notification mechanism). Παρέχει απλές προγραμματιστικές διεπαφές οι οποίες μπορούν να επεκταθούν για την υλοποίηση εξειδικευμένων κλάσεων ανάλογα με το πεδίο εφαρμογής τους (domain specific). Ωστόσο το σύστημα δεν βασίζεται σε ένα συγκεκριμένο μοντέλο context. Αντίστοιχες προσπάθειες αποτελεί και το σύστημα WildCAT [66].

1.6 Μία πρώτη προσέγγιση – Πλατφόρμα Παροχής Υπηρεσιών Αναζήτησης βάσει Θέσης και Χαρακτηριστικών του Χρήστη

1.6.1 Η αρχιτεκτονική της πλατφόρμας

Η πλατφόρμα παροχής υπηρεσιών αναζήτησης, που περιγράφεται στο σημείο αυτό, αποτελεί μία απόπειρα παροχής ολοκληρωμένων – κατά το δυνατόν – υπηρεσιών, για χρήστες ασύρματων και κινητών δικτύων επικοινωνιών. Γνώμονα της επιτυχίας των προηγμένων υπηρεσιών αποτελεί η ίδια η φύση των υπηρεσιών, η οποία επιβάλλει τη χρήση context που σχετίζεται αφενός με τη γεωγραφική θέση και αφ' ετέρου με τις προσωπικές προτιμήσεις και τα ιδιαίτερα χαρακτηριστικά του κάθε χρήστη.



Σχήμα 1: Η Αρχιτεκτονική της πλατφόρμας

Στο Σχήμα 1 φαίνεται η γενικότερη μορφή της αρχιτεκτονικής της εν λόγω πλατφόρμας καθώς και οι διασυνδέσεις της με το δίκτυο κορμού του Λειτουργού Δικτύου. Συγκεκριμένα, η Πλατφόρμα Υπηρεσιών Αναζήτησης αποτελείται από τις εξής τρεις λειτουργικές οντότητες: α) τον Εξυπηρετητή Εφαρμογών (Application Server), ο οποίος περιέχει τη λογική των προσφερόμενων εφαρμογών β) τη Βάση Δεδομένων (Data Base), στην οποία αποθηκεύονται τα προφίλ και οι προτιμήσεις των χρηστών (users' profiles and preferences) και γ) την Οντότητα Θέσης (Location Entity), η οποία είναι υπεύθυνη για τη διασύνδεση της Πλατφόρμας Υπηρεσιών Αναζήτησης με τον Εξυπηρετητή Θέσης (Location

Server) του Δικτύου Κορμού του Λειτουργού Δικτύου, όταν υπάρχει ανάγκη εντοπισμού της γεωγραφικής θέσης του χρήστη από την υπηρεσία.

Η επικοινωνία των ασύρματων και κινητών τερματικών των χρηστών με την Πλατφόρμα Υπηρεσιών Αναζήτησης γίνεται μέσω του Application Server της Πλατφόρμας, βάσει του πρωτοκόλλου SOAP (Simple Object Access Protocol) [67] πάνω από το “πανταχού παρόν” πρωτόκολλο HTTP (Hyper Text Transfer Protocol). Για τη διασύνδεση με τις διάφορες υπηρεσίες που προσφέρει η Πλατφόρμα έχει αναπτυχθεί ο αντίστοιχος πελάτης (Client) του Application Server, ο οποίος στηρίζεται στην τεχνολογία Java 2 Micro Edition [68] και εφαρμόζεται, συν τοις άλλοις, ακόμα και σε συσκευές κινητών τηλεφώνων – εκτός από φορητούς υπολογιστές και PDA. Ο χρήστης συνδέεται σε μία κεντρική σελίδα με τις προσφερόμενες υπηρεσίες, από την οποία μπορεί να επιλέξει την κάθε υπηρεσία που επιθυμεί να χρησιμοποιήσει. Για κάθε υπηρεσία, όπως αναλύεται διεξοδικά στη συνέχεια, απαιτούνται διαφορετικού είδους δεδομένα εισαγωγής και για το λόγο αυτό υπάρχουν οι αντίστοιχες φόρμες εισαγωγής των δεδομένων για κάθε επί μέρους υπηρεσία.

Ως τελευταία διασύνδεση της Πλατφόρμας Υπηρεσιών Αναζήτησης με εξωτερικές λειτουργικές οντότητες διακρίνουμε την επικοινωνία με τον Εξυπηρετητή Θέσης (Location Server) που διαθέτει ο Λειτουργός του Δικτύου. Η επικοινωνία αυτή επιτυγχάνεται μέσω της Οντότητας Θέσης (Location Entity) της Πλατφόρμας. Σκοπός της διασύνδεσης αυτής είναι η ανάκτηση της γεωγραφικής θέσης ενός ή περισσότερων κινητών χρηστών, με σκοπό τη χρησιμοποίησή της από τη λογική των υποστηριζόμενων υπηρεσιών.

Στη συνέχεια της διατριβής ακολουθεί αναλυτική περιγραφή καθώς και περισσότερες τεχνικές λεπτομέρειες, όσον αφορά τις επί μέρους λειτουργικές οντότητες της Πλατφόρμας. Παράλληλα αναλύεται και ο τρόπος επικοινωνίας και αλληλεπίδρασης μεταξύ των λειτουργικών οντοτήτων εσωτερικά της Πλατφόρμας αλλά και με τις περιφερειακές και εξωτερικές δικτυακές οντότητες που παίρνουν μέρος.

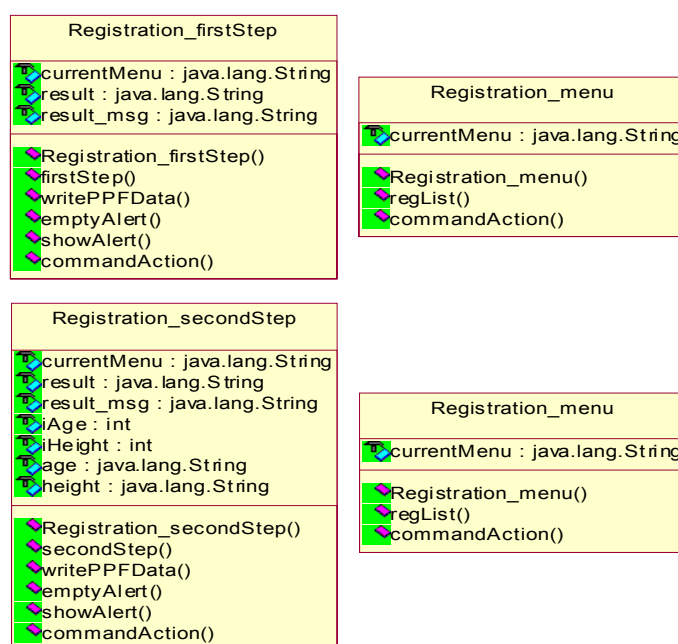
1.6.1.1 Ο Εξυπηρετητής Εφαρμογών (Application Server) και οι Εφαρμογές

Ο Εξυπηρετητής Εφαρμογών (Application Server) περιέχει ουσιαστικά τις εφαρμογές που προσφέρονται στους χρήστες. Στηρίζεται στην τεχνολογία κατανεμημένης επεξεργασίας μεσισμικού Apache AXIS (έκδοση 1.0) [69] που υλοποιεί το πρωτόκολλο SOAP και αποτέλεσε το πλαίσιο για την απομακρυσμένη επίκληση μεθόδων (Remote Method Invocation, RPC) από την πλευρά των χρηστών. Οι υπηρεσίες του Apache AXIS έγιναν προσπελάσιμες μέσω ενός servlet που αναπτύχθηκε στον διαδικτυακό εξυπηρετητή (web server) Apache Tomcat (έκδοση 4.1) [70]. Οι μέθοδοι που μπορεί να εκθέσει ο

Εξυπηρετητής Εφαρμογών προσφέρονται επακριβώς και από έναν συζυγή RMI Εξυπηρετητή (mirror RMI Server) μέσω της αντίστοιχης διεπαφής RMI (RMI Interface). Κατά συνέπεια οι κινητοί χρήστες μπορούν να έχουν πρόσβαση στις προσφερόμενες υπηρεσίες από την Πλατφόρμα με επίκληση απομακρυσμένης μεθόδου είτε με SOAP πάνω από HTTP είτε με RMI πάνω από TCP/IP. Η επικοινωνία του Εξυπηρετητή Εφαρμογών με την Οντότητα Θέσης, στηρίζεται σε διεπαφή RMI, ώστε οι επί μέρους λειτουργικές οντότητες της Πλατφόρμας Υπηρεσιών Αναζήτησης να μην λειτουργούν δικτυοκεντρικά, αλλά να έχουν τη δυνατότητα να είναι ανεξάρτητες η μία από την άλλη καθώς και καταναεμημένες.

Οι υπηρεσίες που προσφέρονται από τον Εξυπηρετητή Εφαρμογών – και κατ' επέκταση από την Πλατφόρμα Υπηρεσιών Αναζήτησης – είναι οι εξής:

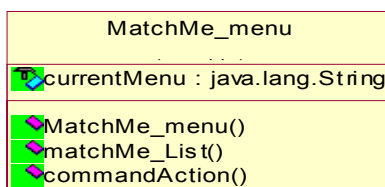
1) Εγγραφή (Registration):



Σχήμα 2: Διαγράμματα κλάσεων της διαδικασίας εγγραφής

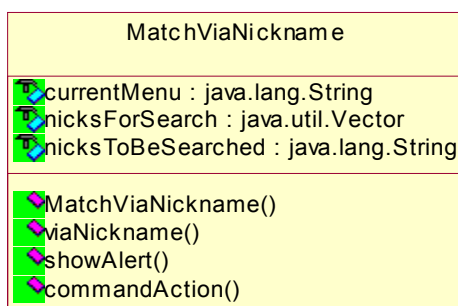
Προφανώς αποτελεί την απαραίτητη προϋπόθεση για να έχει ο χρήστης τη δυνατότητα πρόσβασης στις υπόλοιπες υπηρεσίες. Η εγγραφή πραγματοποιείται σε δύο βήματα. Στο πρώτο βήμα ο χρήστης παρέχει μόνο το ψευδώνυμό του (nickname) και έναν συνθηματικό κωδικό (password), ενώ στο δεύτερο συμπληρώνει στη διαθέσιμη φόρμα κάποια βασικά στοιχεία αναφορικά με το προφίλ και τις προτιμήσεις του. Όλες οι μέθοδοι που διατίθενται στο τερματικό του χρήστη και χρησιμοποιούνται κατά την εγγραφή του φαίνονται αναλυτικά στο Σχήμα 2.

2) Υπηρεσία Match me: Η συγκεκριμένη υπηρεσία δίνει τη δυνατότητα στον χρήστη να αναζητήσει κάποιον άλλον εγγεγραμμένο χρήστη και χωρίζεται σε δύο επί μέρους υπηρεσίες, ανάλογα με τον τρόπο αναζήτησης. Ουσιαστικά προσφέρει τις δυνατότητες αναζήτησης που περιγράφονται παρακάτω, ενώ στο Σχήμα 3 περιγράφεται το αντίστοιχο διάγραμμα κλάσεων του μενού της συγκεκριμένης υπηρεσίας.



Σχήμα 3: Διάγραμμα κλάσεων της υπηρεσίας `MatchMe`

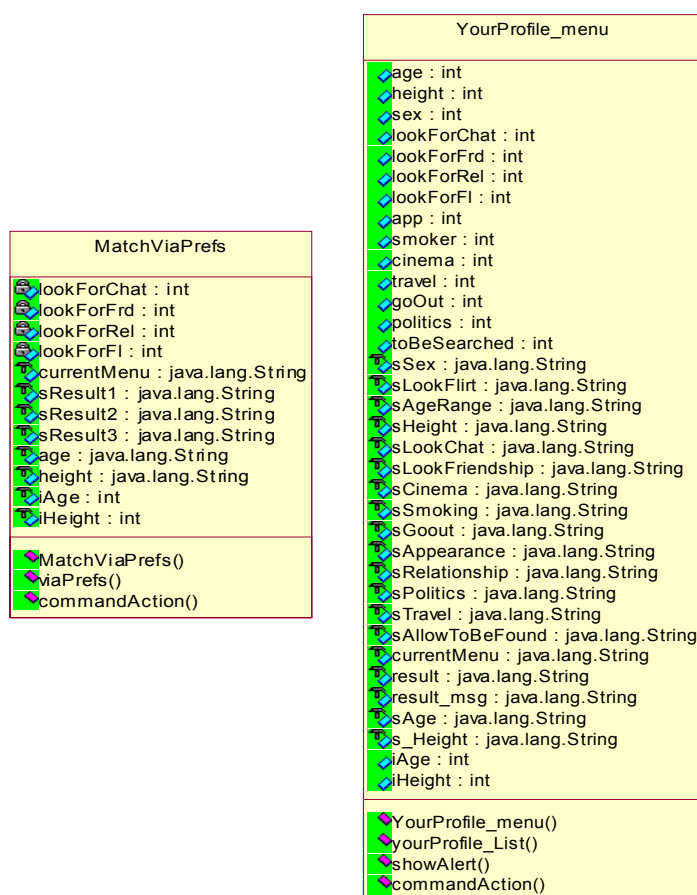
2α) Υπηρεσία MatchViaNickname: Μέσω της υπηρεσίας αυτής, ο χρήστης έχει τη δυνατότητα να αναζητήσει τη γεωγραφική θέση ενός ή περισσότερων χρηστών, για τους οποίους γνωρίζει επιπλέον και εκ των προτέρων τα ψευδώνυμά τους (nicknames), με επίκληση της μεθόδου `matchViaNickname()`.



Σχήμα 4: Διάγραμμα κλάσεων της υπηρεσίας `MatchViaNickname`

Σε πρώτη φάση γίνεται διασταύρωση και επαλήθευση μέσω της Βάσης Δεδομένων των προς αναζήτηση ψευδωνύμων και στην περίπτωση που κατ' αρχάς αυτά είναι υπαρκτά, αλλά και κατά δεύτερο λόγο οι συγκεκριμένοι χρήστες επιτρέπουν έχοντας ενεργοποιημένη τη δυνατότητα αναζήτησης της θέσης τους από τρίτους, γίνεται ουσιαστικά επίκληση της μεθόδου `viaNickname()`, η οποία με τη σειρά της καλεί τη μέθοδο `findDistances()` της Οντότητας Θέσης, όπως θα δούμε και παρακάτω. Τα αποτελέσματα της αναζήτησης εμφανίζονται με τη μορφή λίστας των αντίστοιχων τηλεφωνικών αριθμών των προς αναζήτηση χρηστών. Το διάγραμμα κλάσεων του μενού της υπηρεσίας αυτής φαίνεται στο Σχήμα 4.

2β) Υπηρεσία *MatchViaPreferences*: Μέσω της υπηρεσίας αυτής, ο χρήστης έχει τη δυνατότητα να αναζητήσει τη γεωγραφική θέση ενός ή περισσότερων χρηστών, τους οποίους, αφ’ ενός μεν πιθανόν να μην τους γνωρίζει αφ’ ετέρου παρουσιάζουν παραπλήσιο προφίλ και προτιμήσεις με το αντίστοιχο “ζητούμενο” από τον χρήστη.

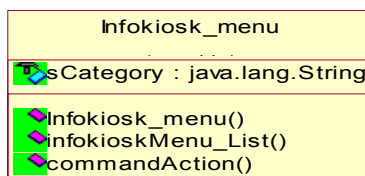


Σχήμα 5: Διάγραμμα κλάσεων της υπηρεσίας *MatchViaPreferences*

Ο χρήστης δηλαδή, επιλέγοντας αυτόν τον τρόπο αναζήτησης, εκτός από τα προσωπικά του στοιχεία (*username*, *password*) συμπληρώνει μία φόρμα με ορισμένα βασικά πεδία προτιμήσεων όπως φύλο, ύψος, ηλικία, σκοπός της αναζήτησης (π.χ. συζήτηση, φλερτ κτλ.) κ.α. Με την αποστολή της συμπληρωμένης φόρμας γίνεται ουσιαστικά επίκληση της μεθόδου *matchViaPrefs()* στον Εξυπηρετητή Εφαρμογών, με την οποία σε πρώτη φάση γίνεται αναζήτηση στην Βάση Δεδομένων των διαθέσιμων χρηστών που πληρούν τις προδιαγραφές των προς αναζήτηση προφίλ και προτιμήσεων. Οι επιστρεφόμενοι χρήστες, αποτελούν ουσιαστικά τα ορίσματα της επίκλησης της μεθόδου *viaPrefs()*, η οποία με τη σειρά της καλεί τη μέθοδο *findDistances()* της Οντότητας Θέσης, η οποία είναι ακριβώς η ίδια (έχει τα ίδια ακριβώς ορίσματα) με τη μέθοδο που καλείται και στην περίπτωση της

υπηρεσίας MatchViaNickname, όπως θα δούμε και παρακάτω. Τα αποτελέσματα της αναζήτησης εμφανίζονται με τη μορφή λίστας των αντίστοιχων τηλεφωνικών αριθμών των προς αναζήτηση χρηστών. Το διάγραμμα κλάσεων του μενού της υπηρεσίας αυτής φαίνεται στο Σχήμα 5.

3) Υπηρεσία Infokiosk: Μέσω της υπηρεσίας αυτής, ο χρήστης έχει τη δυνατότητα να αναζητήσει τη γεωγραφική θέση ενός ή περισσοτέρων σημείων ενδιαφέροντος (π.χ. πρατήρια βενζίνης, φαρμακεία, εστιατόρια, κινηματογράφους κτλ.).



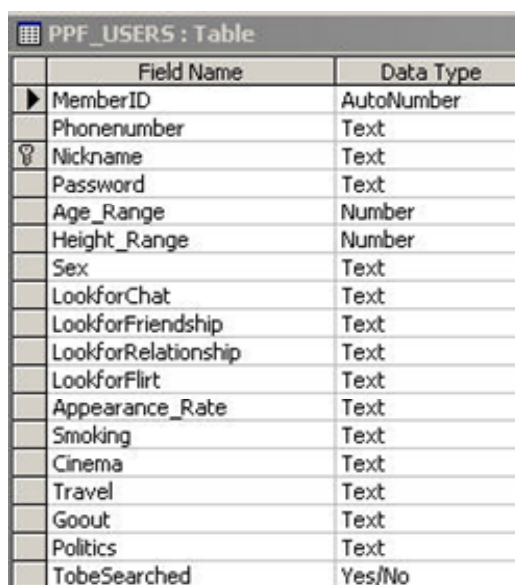
Σχήμα 6: Διάγραμμα κλάσεων της υπηρεσίας Infokiosk

Το νόημα της υπηρεσίας αυτής είναι ότι ως αποτέλεσμα επιστρέφεται μία λίστα με τα πλησιέστερα (γεωγραφικά) σημεία ενδιαφέροντος για τα οποία γίνεται η αναζήτηση. Ο χρήστης, εκτός από τα προσωπικά του στοιχεία (username, password), δηλώνει και την κατηγορία καταστήματος για την οποία ενδιαφέρεται (π.χ. εστιατόρια). Με την αποστολή της συμπληρωμένης φόρμας γίνεται ουσιαστικά επίκληση της μεθόδου *infokiosk_Menu()* στον Εξυπηρετητή Εφαρμογών, και σε πρώτη φάση γίνεται αναζήτηση στην Βάση Δεδομένων των συντεταγμένων όλων των καταστημάτων που ανήκουν στην κατηγορία “εστιατόρια”. Η επιστρεφόμενη λίστα από τη Βάση Δεδομένων καλεί τη μέθοδο *infokiosk_Menu_List()* του Εξυπηρετητή Εφαρμογών, ο οποίος με τη σειρά του καλεί τη μέθοδο *findDistancesInfokiosk()* της Οντότητας Θέσης με ορίσματα τον τηλεφωνικό αριθμό του χρήστη αλλά και όλη τη λίστα των εστιατορίων της Βάσης Δεδομένων, όπως θα δούμε παρακάτω. Το διάγραμμα κλάσεων του μενού της συγκεκριμένης υπηρεσίας φαίνεται στο Σχήμα 6.

1.6.1.2 Η Βάση Δεδομένων – Προφίλ και προτιμήσεις χρηστών

Η επικοινωνία του Εξυπηρετητή Εφαρμογών με τη Βάση Δεδομένων στηρίζεται στην αρχιτεκτονική του JDBC API (Java Data Base Connectivity Application Programming Interface) [71]. Το JDBC είναι μία Java διεπαφή για την εκτέλεση εντολών SQL (Structured Query Language), η οποία υποστηρίζεται από τις περισσότερες σύγχρονες Βάσεις Δεδομένων, όπως MySQL, Oracle, Ingres, Sybase, Informix, MS Access κ.α. Το JDBC

παρέχει όλες τις απαραίτητες κλάσεις και διεπαφές για την επικοινωνία με τη Βάση Δεδομένων και προτιμάται ιδιαίτερα από τους δικτυακούς προγραμματιστές, επειδή για την επικοινωνία απαιτείται μόνο απλή γνώση της γλώσσας προγραμματισμού Java.



PPF_USERS : Table	
Field Name	Data Type
MemberID	AutoNumber
Phonenumber	Text
Nickname	Text
Password	Text
Age_Range	Number
Height_Range	Number
Sex	Text
LookforChat	Text
LookforFriendship	Text
LookforRelationship	Text
LookforFlirt	Text
Appearance_Rate	Text
Smoking	Text
Cinema	Text
Travel	Text
Goout	Text
Politics	Text
TobeSearched	Yes/No

Σχήμα 7: Στιγμιότυπο της βάσης

Η Βάση Δεδομένων της Πλατφόρμας περιέχει όλα τα δεδομένα (προφίλ και προτιμήσεις) των εγγεγραμμένων χρηστών στις υπηρεσίες, καθώς και τις συντεταγμένες των σταθερών σημείων που μπορεί να αναζητηθούν μέσω της υπηρεσίας Infokiosk (εστιατόρια, φαρμακεία κτλ.). Για την Πλατφόρμα Υπηρεσιών Αναζήτησης χρησιμοποιήθηκε Βάση Δεδομένων Microsoft Access SQL Server, αλλά – όπως προαναφέρθηκε – η ίδια ακριβώς υλοποίηση μπορεί να εφαρμοστεί σε μία πλειάδα άλλων σχεσιακών Βάσεων Δεδομένων. Το Σχήμα 7 αποτυπώνει ένα ενδεικτικό μέρος του πίνακα χρηστών της Βάσης.

1.6.1.3 Η Οντότητα Θέσης (Location Entity)

Η Οντότητα Θέσης αποτελεί τον ακρογωνιαίο λίθο για την επικοινωνία της Πλατφόρμας Υπηρεσιών Αναζήτησης με τον Εξυπηρετητή Θέσης (Location Server) του δικτύου.

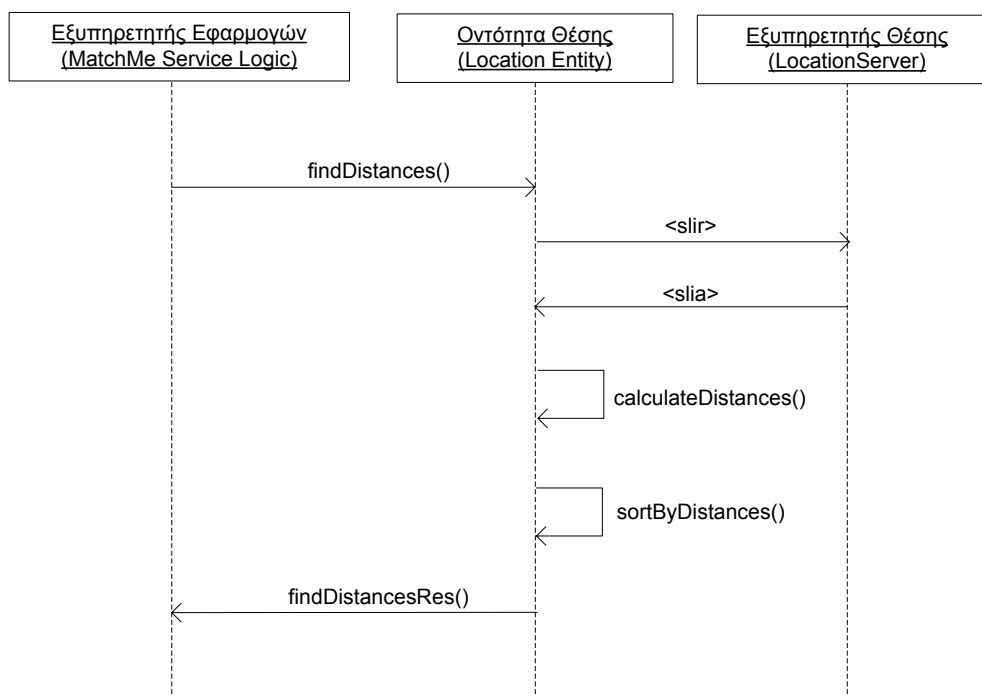
Υλοποίηση με χρήση του πρωτοκόλλου MLP

Η επικοινωνία με τον Εξυπηρετητή Θέσης γίνεται βάσει του πρωτοκόλλου MLP (Mobile Location Protocol) [72], το οποίο αποτελεί πρωτόκολλο επιπέδου εφαρμογής και δημιουργήθηκε από την OMA-LIF (Open Mobile Alliance-Location Interoperability Forum) με πρωταρχικό σκοπό την προσπάθεια καθιέρωσης ενός πρότυπου τρόπου επικοινωνίας των βασισμένων στη θέση του χρήστη υπηρεσιών (Location based Services) με το Δίκτυο Κορμού των Δικτύων Κινητών Επικοινωνιών για τον εντοπισμό θέσης των κινητών

τερματικών [73]. Τα μηνύματα του MLP βασίζονται στην περιγραφική γλώσσα XML (Extensible Mark-up Language) και μπορούν να χρησιμοποιήσουν διάφορα υποκείμενα πρωτόκολλα για τη μεταφορά τους, αν και κατά κύριο λόγο χρησιμοποιείται το HTTP. Το πρωτόκολλο MLP παρέχει τη δυνατότητα ανταλλαγής αιτήσεων και αποκρίσεων μεταξύ της Πλατφόρμας και του Εξυπηρετητή Θέσης για τη γεωγραφική θέση των κινητών τερματικών. Όσον αφορά την εσωτερική επικοινωνία στην Πλατφόρμα μεταξύ του Εξυπηρετητή Εφαρμογών και της Οντότητας Θέσης χρησιμοποιείται η RMI επικοινωνία. Με άλλα λόγια η Οντότητα Θέσης λειτουργεί ως RMI server για τον Εξυπηρετητή Εφαρμογών, ο οποίος και έχει το ρόλο του RMI client κάνοντας απομακρυσμένες επικλήσεις μεθόδων στην Οντότητα Θέσης. Στις επικλήσεις αυτές η Οντότητα Θέσης απ' ενός επικοινωνεί με τον Εξυπηρετητή Θέσης – περνώντας μέσα στα MLP μηνύματα τα κατάλληλα ορίσματα, τα οποία και έχει λάβει από την RMI επίκληση του Εξυπηρετητή Εφαρμογών – και απ' ετέρου είναι σε θέση να διενεργεί τις κατάλληλες διαδικασίες για τον υπολογισμό και τη μετατροπή των γεωγραφικών συντεταγμένων σε αποστάσεις μεταξύ σημείων, καθώς επίσης και να ταξινομεί τις τιμές αυτές κατά αύξουσα σειρά (ανάλογα με τη λογική της κάθε υπηρεσίας). Για την αναλυτικότερη περιγραφή της Οντότητας Θέσης και της λειτουργικότητά της, παρατίθενται και εξηγούνται στη συνέχεια τα διαγράμματα ροής (sequence diagrams) κατά την εκτέλεση της κάθε υπηρεσίας της Πλατφόρμας Υπηρεσιών Αναζήτησης.

Υπηρεσία “Match-Me”: Βάσει της λογικής της υπηρεσίας “Match-me”, ο χρήστης της υπηρεσίας είναι δυνατόν να αναζητήσει τους πλησιέστερους χρήστες που εμφανίζουν παραπλήσιες προτιμήσεις (preferences) με τον ίδιο (π.χ. κοινά ενδιαφέροντα, χόμπυ κτλ.). Έτσι, η λογική της υπηρεσίας από τον Εξυπηρετητή Εφαρμογών προβαίνει σε επίκληση της μεθόδου *findDistances()* της Οντότητας Θέσης, έχοντας ως ορίσματα τους αριθμούς του κινητού συνδρομητή, αλλά και τους αντίστοιχους αριθμούς των υποψήφιων χρηστών, οι οποίοι είναι καταχωρημένοι στην Βάση Δεδομένων με παραπλήσιες προτιμήσεις με αυτές του χρήστη της υπηρεσίας. Η Οντότητα Θέσης – με τη σειρά της, κάνοντας χρήση της μεθόδου *<slir>* (*StandardLocationImmediateRequest*) του MLP στον Εξυπηρετητή Θέσης, ζητά την γεωγραφική θέση όλων των ανωτέρω συνδρομητών. Ο Εξυπηρετητής Θέσης αποκρίνεται με τις συντεταγμένες του χρήστη και των υποψήφιων συνδρομητών καλώντας τη μέθοδο *<slia>* (*StandardLocationImmediateAnswer*) στην Οντότητα Θέσης, η οποία στη συνέχεια – με χρήση των μεθόδων *calculateDistances()* και *sortByDistances()* – υπολογίζει αρχικά όλες τις αποστάσεις μεταξύ των υποψήφιων συνδρομητών και του χρήστη της υπηρεσίας και επιστρέφει σε αυτόν τους πέντε (5) πλησιέστερους γεωγραφικά συνδρομητές

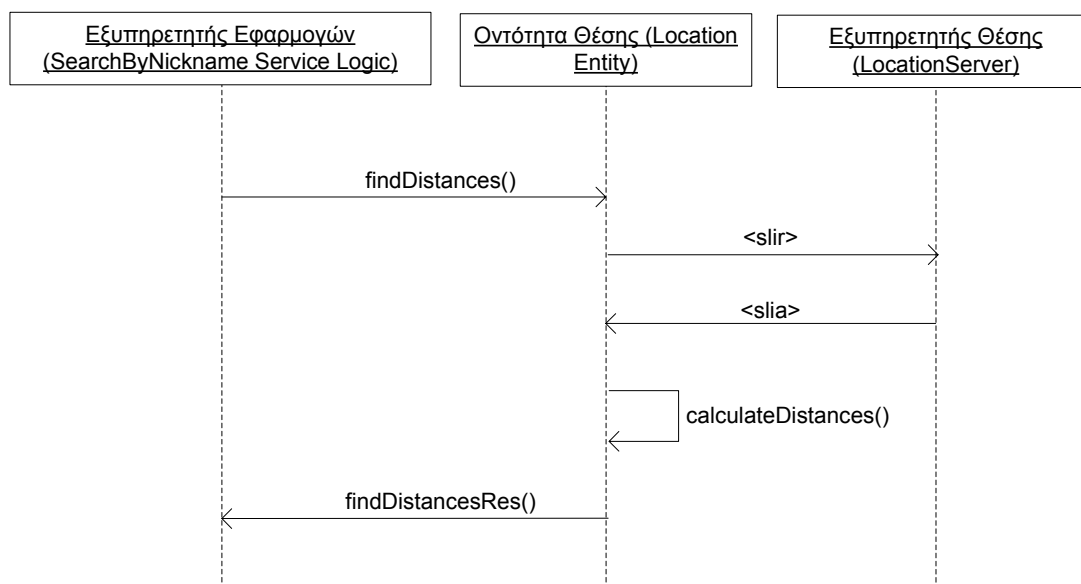
(με χρήση της μεθόδου *findDistancesRes()*), με τους οποίους μπορεί να ξεκινήσει συνομιλία ή ανταλλαγή γραπτών μηνυμάτων (Σχήμα 8).



Σχήμα 8: Διάγραμμα Αλληλεπιδράσεων Υπηρεσία Match-Me

Υπηρεσία “*SearchByNickname*”: Η λογική της υπηρεσίας “*SearchByNickname*” αποτελεί ουσιαστικά υποπερίπτωση της προηγούμενης υπηρεσίας. Η ειδοποιός διαφορά της είναι ότι ο χρήστης της υπηρεσίας αυτής δεν αναζητά την απόσταση μεταξύ του ίδιου και ορισμένων υποψήφιων χρηστών που διακρίνονται για τις συγκεκριμένες προτιμήσεις τους, αλλά ζητά ρητά την απόσταση μεταξύ του ίδιου και ενός συνδρομητή, του οποίου γνωρίζει ήδη το ψευδώνυμο. Η ερώτηση αυτή γίνεται κατά τα γνωστά από τη λογική της υπηρεσίας *SearchByNickname* του Εξυπηρετητή Εφαρμογών με επίκληση της μεθόδου *findDistances()* της Οντότητας Θέσης, με ορίσματα αυτή τη φορά τον αριθμό του χρήστη και το αριθμό του προς αναζήτηση συνδρομητή. Ο αριθμός του τελευταίου έχει ανακτηθεί από το αντίστοιχο ψευδώνυμο της Βάσης Δεδομένων. Εν συνεχεία, η Οντότητα Θέσης κάνοντας χρήση της μεθόδου *<slir>* (*StandardLocationImmediateRequest*) του MLP στον Εξυπηρετητή Θέσης, ζητά την γεωγραφική θέση των χρηστών που αντιστοιχούν στους δύο ανωτέρω αριθμούς συνδρομητών. Ο Εξυπηρετητής Θέσης αποκρίνεται με τις συντεταγμένες του χρήστη και του προς αναζήτηση συνδρομητή με τη μέθοδο *<slia>* (*StandardLocationImmediateAnswer*) που καλείται πίσω στην Οντότητα Θέσης. Ακολουθεί χρήση της μεθόδου *calculateDistances()* για τον υπολογισμό της μεταξύ τους απόστασης και

κοινοποίηση της απόστασης αυτής με χρήση της μεθόδου *findDistancesRes()* στον Εξυπηρετητή Εφαρμογών, όπως φαίνεται στο Σχήμα 9.

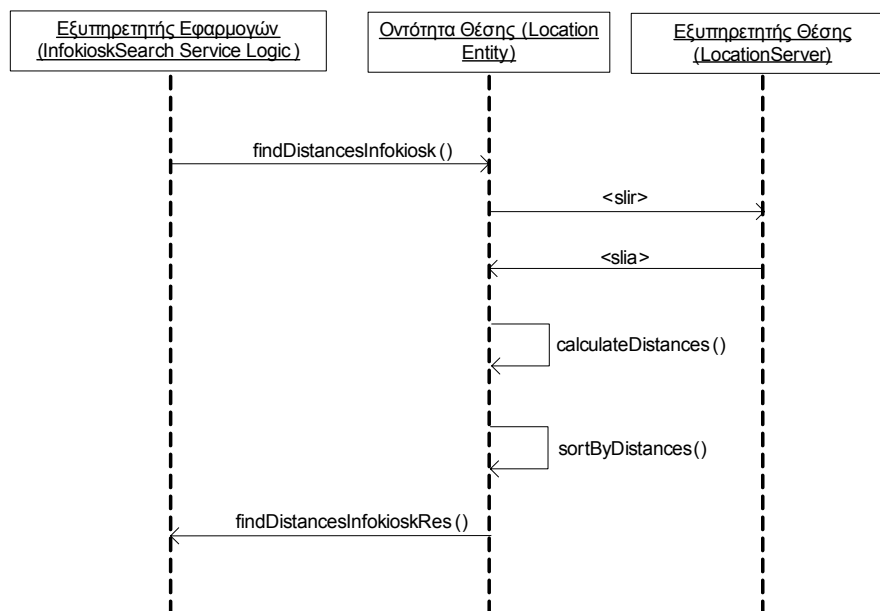


Σχήμα 9: Διάγραμμα Αλληλεπιδράσεων Υπηρεσία SearchByNickname

Υπηρεσία “*InfokioskSearch*”: Η λογική της υπηρεσίας “*InfokioskSearch*” αποτελεί επίσης παραλλαγή της υπηρεσίας “*Match-Me*”. Το ιδιαίτερο χαρακτηριστικό της υπηρεσίας αυτής είναι ότι η αναζήτηση των αποστάσεων γίνεται μεταξύ ενός κινητού χρήστη (του χρήστη της υπηρεσίας) και ορισμένων σταθερών σημείων που πληρούν συγκεκριμένες προτιμήσεις του χρήστη (π.χ. όλα τα ιταλικά εστιατόρια).

Κατά την εκτέλεση της υπηρεσίας αυτής, η λογική της υπηρεσίας, που κατοικοεδρεύει στον Εξυπηρετητή Εφαρμογών, καλεί τη μέθοδο *findDistancesInfokiosk()* της Οντότητας Θέσης με ορίσματα τον αριθμό του κινητού συνδρομητή του χρήστη και τις γεωγραφικές συντεταγμένες των σταθερών σημείων που αναζητά ο χρήστης (π.χ. τις συντεταγμένες όλων των ιταλικών εστιατορίων που είναι καταχωρημένα στην Βάση Δεδομένων). Οι συντεταγμένες αυτές έχουν ανακτηθεί από την Βάση Δεδομένων, στην οποία είναι καταχωρημένες άπαξ, δεδομένου ότι πρόκειται για σταθερά γεωγραφικά σημεία. Στη συνέχεια, η Οντότητα Θέσης κρατά τις συντεταγμένες αυτές και με επίκληση της μεθόδου *<slir>* (*StandardLocationImmediateRequest*) του MLP στον Εξυπηρετητή Θέσης, ζητά την γεωγραφική θέση του χρήστη, ενώ ο Εξυπηρετητής Θέσης αποκρίνεται –κατά τα γνωστά – επιστρέφοντας τις συντεταγμένες του χρήστη με κλήση της μεθόδου *<slia>* (*StandardLocationImmediateAnswer*) της Οντότητας Θέσης. Ακολουθεί ο υπολογισμός των αποστάσεων του χρήστη από το κάθε σταθερό σημείο, με τη βοήθεια της μεθόδου υπολογισμού αποστάσεων *calculateDistances()* και η κατάταξη των αποστάσεων αυτών σε

αύξουσα σειρά, μέσω της μεθόδου *sortByDistances()*. Τέλος, οι πέντε (5) πλησιέστεροι γεωγραφικά προορισμοί, επιστρέφονται στον Εξυπηρετητή Εφαρμογών με επίκληση της μεθόδου *findDistancesInfokioskRes()* για την ενημέρωση των αποτελεσμάτων στον χρήστη. Το διάγραμμα ροής της υπηρεσίας φαίνεται στο Σχήμα 10.



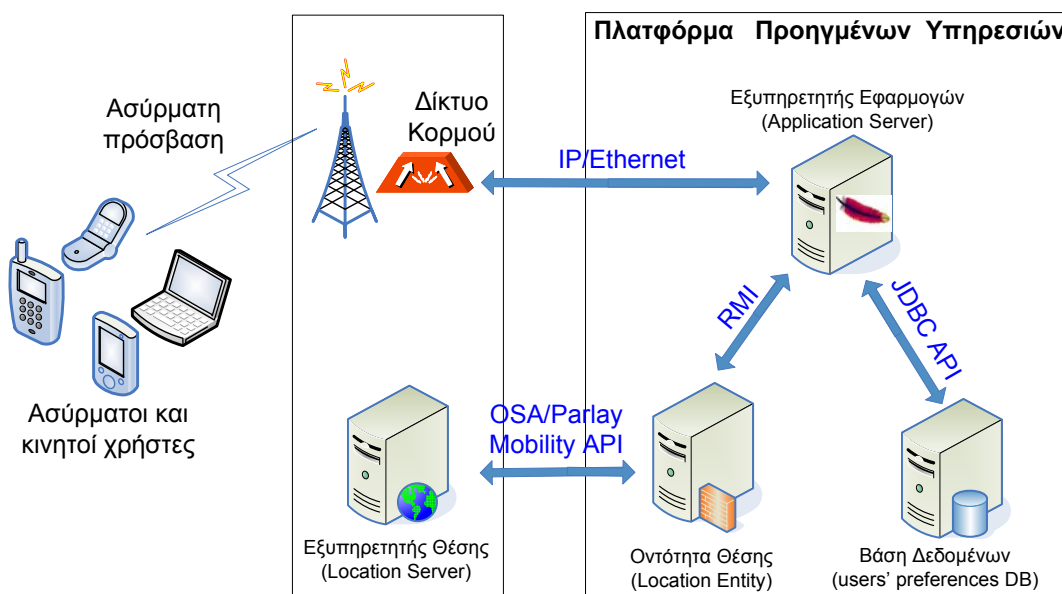
Σχήμα 10: Διάγραμμα Αλληλεπιδράσεων Υπηρεσία InfokioskSearch

Υλοποίηση με χρήση της ανοικτής διεπαφής OSA/Parlay Mobility

Με την απελευθέρωση της αγοράς των τηλεπικοινωνιών έχουν κάνει την εμφάνισή τους στο τηλεπικοινωνιακό προσκήνιο οι προγραμματιστικές διεπαφές (Application Programming Interfaces, APIs). Οι προγραμματιστικές διεπαφές αποτελούν επάξιο αντικαταστάτη των κλασικών τηλεπικοινωνιακών πρωτοκόλλων και προσδίδουν απλοποίηση στον τρόπο και στην κατανόηση της επικοινωνίας των εμπλεκόμενων οντοτήτων. Ο μηχανικός – προγραμματιστής, δεν χρειάζεται απαραίτητα να γνωρίζει σε βάθος κάποιο πρωτόκολλο επικοινωνίας, αλλά απλώς αρκεί να είναι γνώστης των θεμελιωδών αρχών του προγραμματισμού και των βασικότερων εννοιών του [74]. Όσον αφορά την παροχή τηλεπικοινωνιακών υπηρεσιών, οι περισσότερες από τις εταιρίες-μεγαθήρια στον τομέα των τηλεπικοινωνιών έχουν υλοποιήσει τις δικές τους ιδιοταγείς (proprietary) προγραμματιστικές διεπαφές. Στην αντίπερα όχθη, σε μία προσπάθεια ανεξαρτητοποίησης από οποιαδήποτε δέσμευση από τους επιμέρους τηλεπικοινωνιακούς οίκους, βρίσκονται οι ομάδες εργασίας και οι διεθνείς οργανισμοί για τη σύσταση “ανοιχτών προγραμματιστικών διεπαφών” (open Application Programming Interfaces, open APIs). Οι ανοιχτές προγραμματιστικές διεπαφές αποτελούν έναν εναλλακτικό τρόπο παροχής σύγχρονων και

προηγμένων υπηρεσιών από ανεξάρτητους Παρόχους Υπηρεσιών, οι οποίοι δεν έχουν απαραίτητα και τον ρόλο του Λειτουργού Δικτύου (Network Operator). Με χρήση και εφαρμογή των ανοιχτών προγραμματιστικών διεπαφών είναι δυνατή η παροχή υπηρεσιών ανεξάρτητα από την υποκείμενη υποδομή και τα υποκείμενα χρησιμοποιούμενα πρωτόκολλα επικοινωνίας [75].

Σημαντικότερο εκπρόσωπο των τηλεπικοινωνιακών ανοιχτών προγραμματιστικών διεπαφών για την παροχή υπηρεσιών αποτελεί το OSA/Parlay forum [76] που προτείνει τις ομώνυμες διεπαφές, το OSA/Parlay API [77]. Οι διεπαφές OSA/Parlay χωρίζονται σε επί μέρους υποσύνολα, ανάλογα με τη λειτουργικότητα που προσφέρει το καθένα από αυτά. Ανάμεσα στα υποσύνολα των διεπαφών OSA/Parlay ξεχωρίζουμε και τη διεπαφή Κινητικότητας του OSA/Parlay (Mobility API) [78], η οποία ενδείκνυται για την περίπτωση παροχής τηλεπικοινωνιακών υπηρεσιών που απαιτούν γνώση της θέσης των χρηστών (Location Based Services). Η συγκεκριμένη διεπαφή μπορεί κάλλιστα να εφαρμοστεί και στην περίπτωση που εξετάσαμε παραπάνω. Με άλλα λόγια, η διεπαφή Κινητικότητας του OSA/Parlay είναι εφαρμόσιμη και στην Πλατφόρμα Παροχής Υπηρεσιών Αναζήτησης που σχετίζονται με τη θέση του χρήστη και τις προτιμήσεις του. Συγκεκριμένα, οι μοναδικές προαπαιτήσεις για την λειτουργία της πλατφόρμας με χρήση των διεπαφών OSA/Parlay, είναι η υλοποίηση και χρησιμοποίηση, από την Οντότητα Θέσης, της διεπαφής Κινητικότητας του OSA/Parlay (Σχήμα 11).



Σχήμα 11: Η Αρχιτεκτονική της πλατφόρμας με χρήση της Διεπαφής Κινητικότητας του OSA/Parlay

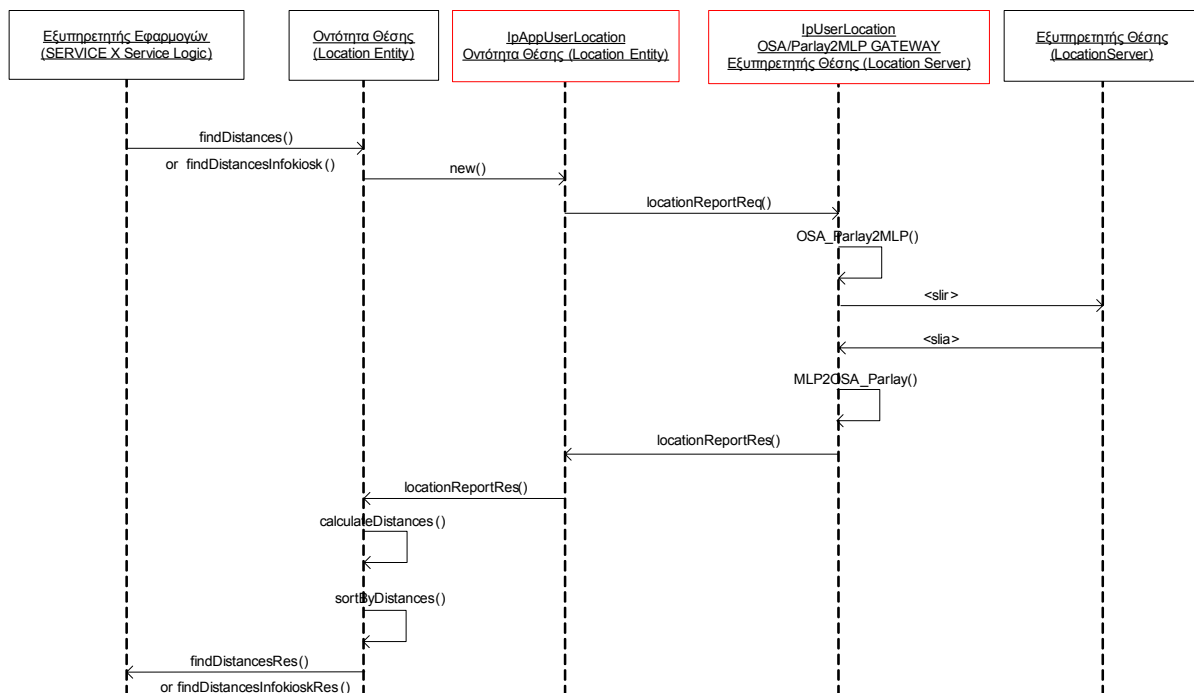
Η διεπαφή Κινητικότητας – όπως και κάθε διεπαφή του OSA/Parlay – αποτελείται από δύο μέρη: την πλευρά των εφαρμογών (application part) και την πλευρά του δικτύου (network

part). Στην περίπτωση μας, η πλευρά των εφαρμογών της διεπαφής Κινητικότητας πρέπει να υποστηρίζεται από την Οντότητα Θέσης, ενώ η πλευρά του δικτύου της διεπαφής Κινητικότητας πρέπει να υλοποιείται στον Εξυπηρετητή Θέσης για την μεταξύ τους επικοινωνία. Ουσιαστικά, ο Εξυπηρετητής Θέσης θα λειτουργεί ταυτόχρονα και σαν “Πύλη διασύνδεσης” του OSA/Parlay με το MLP (OSA/Parlay to MLP Gateway). Η λειτουργικότητα της Πύλης (Gateway) είναι η μετατροπή των μεθόδων – αιτήσεων που λαμβάνει από την πλευρά των εφαρμογών μέσω της διεπαφής Κινητικότητας του OSA/Parlay, στα αντίστοιχα μηνύματα που υποστηρίζει το υποκείμενο δικτυακό πρωτόκολλο MLP, για τον εντοπισμό της θέσης ενός ή περισσότερων χρηστών.

Για την ενσάρκωση των τριών λογικών υπηρεσιών που περιγράφηκαν ανωτέρω (“Match-Me”, “SearchByNickname” και “InfokioskSearch”) απαιτείται η ανάπτυξη του υποσυνόλου `IpUserLocation` της διεπαφής Κινητικότητας του OSA/Parlay πάνω από στρώμα μεσισμικού [79]. Δύο οντότητες – αντικείμενα καθορίζονται και συγκεκριμένα τα αντικείμενα `IpUserLocation` προς την πλευρά του δικτύου και ουσιαστικά στον Εξυπηρετητή Θέσης, καθώς και το συμμετρικό αυτού προς την πλευρά των εφαρμογών, δηλαδή το `IpAppUserLocation`, στην Πλατφόρμα Υπηρεσιών Αναζήτησης και συγκεκριμένα στην Οντότητα Θέσης [80]. Τα πρώτα αντικείμενα εκθέτουν τις μεθόδους μέσω του στρώματος μεσισμικού προς τις εφαρμογές για την ανάθεση και τον καθορισμό των κριτηρίων σκανδαλισμού στους κόμβους του δικτύου, δηλαδή την πρόσβαση στον Εξυπηρετητή Θέσης, ενώ τα τελευταία υλοποιούν τις αντίστοιχες callback μεθόδους. Το ζεύγος αντικειμένων `IpUserLocation` και `IpAppUserLocation` ακολουθεί το μοντέλο αναφοράς άμεσης αίτησης/απόκρισης αναφοράς θέσης (immediate response/request location reporting model) ή περιοδικής αναφοράς θέσης (periodic location reporting model), το οποίο αρκεί για την υλοποίηση και των τριών υπηρεσιών που παρουσιάστηκαν παραπάνω. Και οι τρεις υπηρεσίες που βρίσκονται στον Εξυπηρετητή Εφαρμογών κατέχουν και διατηρούν την αναφορά των αντικειμένων της πλευράς του δικτύου της διεπαφής Κινητικότητας του OSA/Parlay. Οι μέθοδοι των αντικειμένων αυτών, καθώς και η συνολική τους λειτουργικότητα, είναι υλοποιημένη στον Εξυπηρετητή Θέσης και συγκεκριμένα στην Πύλη Θέσης (Location Gateway), η οποία κληρονομεί τα χαρακτηριστικά της διεπαφής `IpUserLocation` του OSA/Parlay.

Η Πύλη Θέσης μετατρέπει τις ληφθείσες – μέσω του μεσισμικού – μεθόδους της διεπαφής Κινητικότητας του OSA/Parlay στα αντίστοιχα μηνύματα του υποκείμενου πρωτοκόλλου MLP, και αντιστρόφως. Συγκεκριμένα, η μέθοδος του OSA/Parlay `locationReportReq()` μετατρέπεται στο

μήνυμα *<slir>* του MLP. Στο ίδιο ακριβώς μήνυμα του MLP μετατρέπεται και η μέθοδος *extendedLocationReportReq()* της διεπαφής *IpUserLocation* του OSA/Parlay. Στην αντίθετη κατεύθυνση, η Πύλη Εντοπισμού Θέσης λαμβάνει τα αποτελέσματα από τον Εξυπηρετητή Εντοπισμού Θέσης (Location Server) υπό τη μορφή μηνυμάτων του MLP, τα αναλύει και τα μετατρέπει στις αντίστοιχες μεθόδους του OSA/Parlay, τις οποίες και επικαλείται στα callback αντικείμενα της πλευράς των εφαρμογών, στην Οντότητα Θέσης. Τα αποτελέσματα που περιέχονται στα μηνύματα *<slia>* or *<slirep>* του MLP μετατρέπονται σε ορίσματα των μεθόδων *locationReportRes()* ή *extendedLocationReportRes()* του OSA/Parlay. Σε περίπτωση που η Πύλη Εντοπισμού Θέσης αναγνωρίσει τυχόν λάθος σε κάποιο από τα παραπάνω μηνύματα του MLP, επικαλείται τη μέθοδο *locationReportErr()* στα αντικείμενα της πλευράς των εφαρμογών. Το διάγραμμα ροής μηνυμάτων στο Σχήμα 12 απεικονίζει τις αλληλεπιδράσεις των εμπλεκόμενων αντικειμένων (μέθοδοι – μηνύματα) κατά την εκτέλεση της υπηρεσίας .



Σχήμα 12: Διάγραμμα αλληλεπιδράσεων παροχής υπηρεσίας με χρήση της Διεπαφής OSA/Parlay

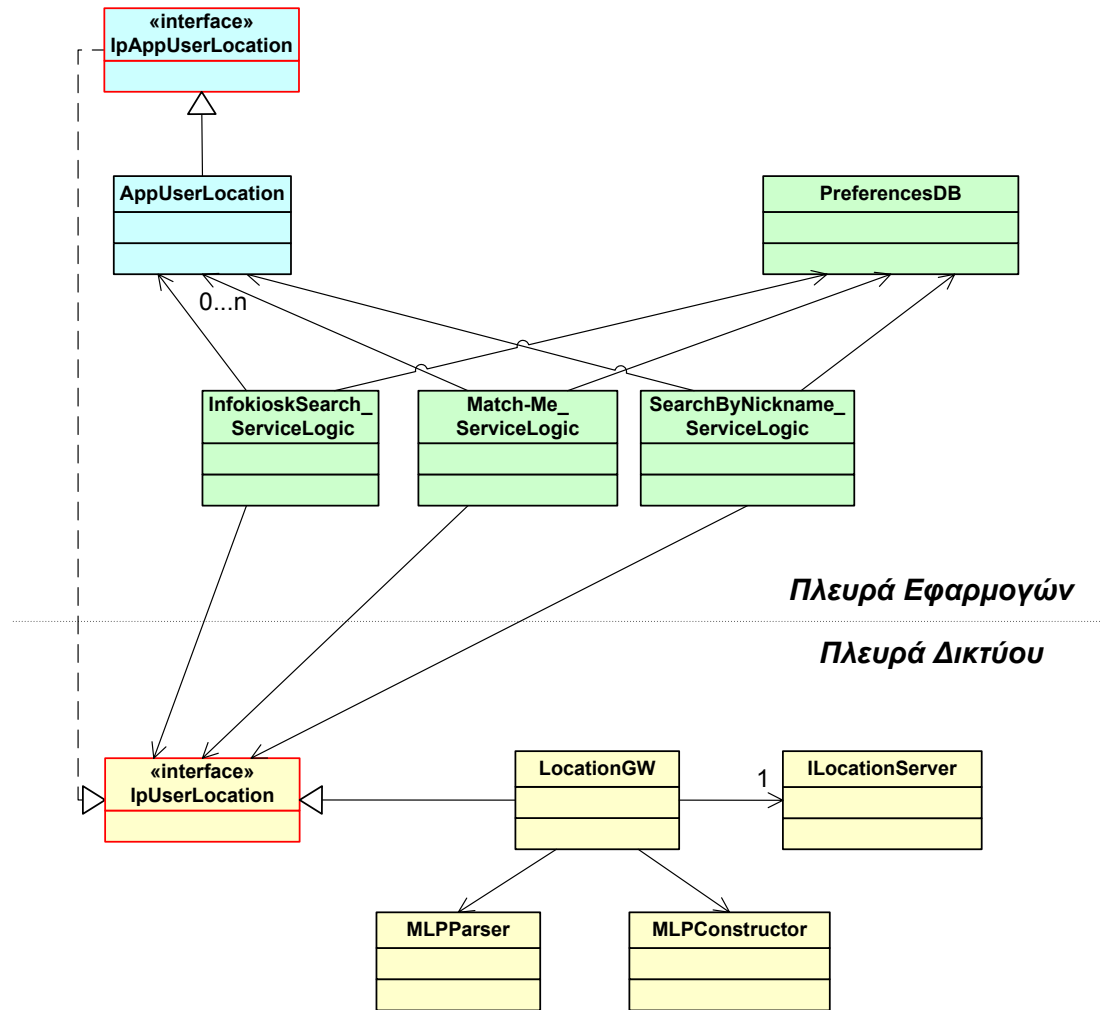
Το σύνολο των αντιστοιχίσεων και μετατροπών που πραγματοποιεί η Πύλη Θέσης (Location Gateway) μεταξύ των μεθόδων της διεπαφής Κινητικότητας του OSA/Parlay και των μηνυμάτων του υποκείμενου πρωτοκόλλου MLP φαίνονται στον Πίνακα 1.

Πίνακας 1: Η λειτουργικότητα της Πύλης Θέσης (Αντιστοίχιση μεθόδων OSA/Parlay σε μηνύματα του MLP και αντιστρόφως)

Application to Network		
Method	Invoked on	Wrapped to MLP Element
locationReportReq	IpUserLocation	<slir>
extendedLocationReportReq	IpUserLocation	<slir>
Network to Application		
MLP Message Element	Reported to	Wrapped to Method
<slia> or <slirep>	IpAppUserLocation	locationReportRes
Error message in <slia>	IpAppUserLocation	locationReportErr

Ο Πίνακας 1 αντικατοπτρίζει τη λειτουργικότητα της Πύλης Θέσης, δηλαδή της κλάσης LocationGW. Βάσει των παραμέτρων και των ορισμάτων των εισερχόμενων στην Πύλη μεθόδων κατασκευάζονται τα μηνύματα του MLP, με χρήση της λειτουργικότητας της κλάσης MLPConstructor (χρήση της μεθόδου OSA_Parlay2MLP()). Στην αντίθετη κατεύθυνση τα λαμβανόμενα στην Πύλη MLP μηνύματα από τον Εξυπηρετητή Θέσης, αναλύονται και μετατρέπονται στις μεθόδους του OSA/Parlay με χρήση της λειτουργικότητας που προσφέρει η κλάση MLPParser (χρήση της μεθόδου MLP2OSA_Parlay()). Στην πλευρά του δικτύου η οντότητα ILocationServer αντιπροσωπεύει την αφηρημένη διεπαφή που παρέχεται στην Πύλη Θέσης από τον Λειτουργό του Δικτύου. Με χρήση αυτής, η Πύλη Θέσης εκμεταλλεύεται την πληροφορία θέσης που προσδίδει ο Λειτουργός Δικτύου και προωθεί τα MLP μηνύματα προς τον Εξυπηρετητή Θέσης του Λειτουργού Δικτύου.

Το σύνολο των προαναφερθέντων εμπλεκόμενων οντοτήτων κατά την εκτέλεση των τριών υπηρεσιών φαίνεται στο διάγραμμα κλάσεων του Σχήματος 13. Από το ίδιο σχήμα, στην πλευρά των εφαρμογών παρατηρούμε ότι η λογική των υπηρεσιών εμπεριέχεται στις αντίστοιχες οντότητες (Match-Me_ServiceLogic, SearchByNickname_ServiceLogic και Infokiosk_ServiceLogic). Και οι τρεις υπηρεσίες χρησιμοποιούν κατά την εκτέλεσή τους περιεχόμενα και πληροφορίες που βρίσκονται αποθηκευμένα στη Βάση Δεδομένων (PreferencesDB). Τα αντικείμενα αυτά, προσφέρουν στο στρώμα μεσοσυστημικού τις κατάλληλες διεπαφές για την ανάκτηση των ζητούμενων δεδομένων μέσω μεθόδων απομακρυσμένων κλήσεων (Remote Procedure Calls, RPC). Για κάθε αίτηση υπηρεσίας, και οι τρεις υπηρεσίες γενούν δυναμικά το callback αντικείμενο του OSA/Parlay (AppUserLocation), ενώ πριν την εκτέλεσή της η κάθε υπηρεσία δίνει υπόσταση μέσω του αντικειμένου AppUserLocation στη διεπαφή IpAppUserLocation για την περαιτέρω επικοινωνία με την πλευρά του δικτύου.



Σχήμα 13: Διάγραμμα κλάσεων του συστήματος

1.7 Συμπεράσματα και κατευθύνσεις

Έχοντας λοιπόν ολοκληρώσει την επισκόπηση των έως τώρα σχετικών ερευνητικών προσπαθειών εξάγονται τα παρακάτω συμπεράσματα. Όπως έχει ήδη προαναφερθεί οι περισσότερες από τις πρώτες απόπειρες για την δημιουργία context aware συστημάτων οδήγησαν στην υλοποίηση συγκεκριμένων εφαρμογών εντός ενός συγκεκριμένου τομέα (π.χ. intelligent spaces) με χρήση περιορισμένων ειδών πληροφορίας context. Στην κατηγορία αυτή ανήκει και η πλατφόρμα παροχής υπηρεσιών αναζήτησης βάσει θέσης και χαρακτηριστικών του χρήστη που υλοποιήθηκε στο πλαίσιο της παρούσας διατριβής και παρουσιάστηκε στην προηγούμενη ενότητα. Τα συστήματα αυτά κατά συνέπεια έπασχαν όσον αφορά την επεκτασιμότητά τους και την διαλειτουργικότητά τους με άλλες εφαρμογές. Έτσι ο κόσμος της έρευνας στράφηκε σε πιο γενικευμένες λύσεις που παρουσιάστηκαν στην αντίστοιχη ενότητα. Ακόμα όμως και αυτές δεν αποτέλεσαν πανάκεια στο πρόβλημα

αναζήτησης μιας γενικευμένης λύσης για την υποστήριξη παροχής context-aware υπηρεσιών. Σχεδόν στο σύνολό τους οι προτεινόμενες λύσεις εντοπίζονται στην περιοχή των κινητών τερματικών. Ακολουθούν συγκεντρωτικό αρχιτεκτονικό μοντέλο με τουλάχιστον μία οντότητά τους, συνήθως την θεμελιώδη, να αποτελεί μοναδικό σημείο αστοχίας (single point of failure).

Παρόλο που παρέχουν την δυνατότητα παροχής περισσότερων της μίας εφαρμογών συνεχίζουν να παραμένουν domain specific. Σχεδόν οι περισσότερες εκμεταλλεύονται πληροφορίες context που προέρχονται από φυσικούς αισθητήρες δίνοντας λιγότερη έμφαση σε πληροφορίες όπως π.χ. το προφίλ του χρήστη, καταλήγοντας τελικά να παρέχουν στην εφαρμογή πλήρη εικόνα για το περιβάλλον του χρήστη αλλά όχι για τον χρήστη αυτόν καθ' εαυτό. Κάθε μία από την προσεγγίσεις εφαρμόζει το δικό της μοντέλο πληροφορίας ορισμένο με διαφορετικό κάθε φορά τρόπο και μέσα. Τέλος, σχεδόν σε όλες τις περιπτώσεις, παρέχονται έξυπνοι μηχανισμοί μετάφρασης και συνδυασμού των αρχικά ανακτώμενων πληροφοριών, προκειμένου να δημιουργηθούν υψηλότερης ποιότητας πληροφορίες. Ωστόσο η διαδικασία αυτή προσθέτει πολυπλοκότητα σε ένα ήδη πολύπλοκο σύστημα. Δεν είναι δυνατόν να προδιαγραφούν όλοι οι πιθανοί συνδυασμοί πρωτευουσών πληροφοριών context που ο συνδυασμός τους να παράγει περαιτέρω αξιοποιήσιμη πληροφορία. Συνεπώς μία προσέγγιση όπου η διαδικασία αυτή θα εκτελούνταν κατά περίπτωση από την ίδια την εφαρμογή θα ήταν προτιμότερη.

Με βάση όλα τα παραπάνω καταλήγουμε στο συμπέρασμα ότι απαιτείται:

1. Συνολικός και καθολικός καθορισμός και μοντελοποίηση της πληροφορίας context με ευρέως αποδεκτές μεθόδους και εργαλεία, προκειμένου τα μέρη που εμπλέκονται στην διαδικασία ανάκτησης και εκμετάλλευσης του context να χρησιμοποιούν κοινές αναφορές και ονοματολογία.
2. Σχεδιασμός και υλοποίηση ενός ανοικτού κατανεμημένου ανθρωποκεντρικού συστήματος με τα παρακάτω χαρακτηριστικά και δυνατότητες:
 - i. Ανάκτηση και δόμηση της πληροφορίας με βάση το μοντέλο.
 - ii. Διατήρηση, ανανέωση και διαχείριση της πληροφορίας.
 - iii. Διαχωρισμός των παραπάνω διαδικασιών από τυχόν διαδικασίες προσαρμογής της υπηρεσίας βάσει της πληροφορίας context.
 - iv. Παροχή της πληροφορίας στις εφαρμογές με ομοιόμορφο και προτυποποιημένο τρόπο.
 - v. Ανοικτή και ευέλικτη αρχιτεκτονική προκειμένου να είναι δυνατή η προσθήκη νέων εγγενών λειτουργιών του συστήματος (π.χ. ανάγκη

εισαγωγής νέων διαδικασιών για την διαχείριση επιπλέον πληροφοριών context), καθώς και η ενσωμάτωση εξωτερικών υπηρεσιών για την εκμετάλλευση της πληροφορίας με ελάχιστο υπολογιστικό, χρονικό και οικονομικό κόστος και με τις λιγότερες δυνατές αλλαγές τόσο στο σύστημα όσο και στην πλευρά της υπηρεσίας.

- vi. Κατανεμημένο σύστημα για λόγους εφεδρείας και κατανομής του φορτίου.
- vii. Έλεγχος και διαχείριση των προσωπικών δεδομένων του χρήστη από τον ίδιο.
- viii. Ανεξάρτητο των εφαρμογών και των υπηρεσιών καθώς και του πεδίου εφαρμογής του (domain)
- ix. Εφαρμόσιμο και σε περιπτώσεις συστημάτων μεγάλης κλίμακας.

Λαμβάνοντας υπ' όψιν όλα τα παραπάνω σχεδιάστηκε και υλοποιήθηκε το σύστημα που παρουσιάζεται στα επόμενα κεφάλαια.

1.8 Αναφορές

- [1] Schilit, B., Theimer, M. "Disseminating Active Map Information to Mobile Hosts". IEEE Network, 8(5) (1994) 22-32
- [2] Brown, P.J., Bovey, J.D. Chen, X. "Context-Aware Applications: From the Laboratory to the Marketplace". IEEE Personal Communications, 4(5) (1997) 58-64
- [3] Ryan, N., Pascoe, J., Morse, D. Enhanced Reality Fieldwork: the Context-Aware Archaeological Assistant. Gaffney, V., van Leusen, M., Exxon, S. (eds.) Computer Applications in Archaeology (1997)
- [4] Dey, A.K. Context-Aware Computing: The CyberDesk Project. AAAI 1998 Spring Symposium on Intelligent Environments, Technical Report SS-98-02 (1998) 51-54
- [5] Brown, P.J. The Stick-e Document: a Framework for Creating Context-Aware Applications. Electronic Publishing '96 (1996) 259-272
- [6] Franklin, D., Flaschbart, J. All Gadget and No Representation Makes Jack a Dull Environment. AAAI 1998 Spring Symposium on Intelligent Environments, Technical Report SS-98-02 (1998) 155-160
- [7] Ward, A., Jones, A., Hopper, A. A New Location Technique for the Active Office. IEEE Personal Communications 4(5) (1997) 42-47
- [8] Rodden, T., Cheverst, K., Davies, K. Dix, A.. Exploiting Context in HCI Design for Mobile Systems. Workshop on Human Computer Interaction with Mobile Devices (1998)
- [9] Hull, R., Neaves, P., Bedford-Roberts, J. Towards Situated Computing. 1st International Symposium on Wearable Computers (1997) 146-153

- [10] Schilit, B., Adams, N. Want, R. Context-Aware Computing Applications. 1st International Workshop on Mobile Computing Systems and Applications. (1994) 85-90
- [11] Dey, A.K., Abowd, G.D., Wood, A. CyberDesk: A Framework for Providing Self-Integrating Context-Aware Services. Knowledge-Based Systems, 11 (1999) 3-13
- [12] Pascoe, J. Adding Generic Contextual Capabilities to Wearable Computers. 2nd International Symposium on Wearable Computers (1998) 92-99
- [13] Want, R., Hopper, A., Falcao, V., Gibbons, J. The Active Badge Location System. ACM Transactions on Information Systems 10(1) (1992) 91-102
- [14] Brown, M. Supporting User Mobility. International Federation for Information Processing (1996)
- [15] Cooperstock, J., Tanikoshi, K., Beirne, G., Narine, T., Buxton, W. Evolution of a Reactive Environment CHI '95 (1995) 170-177
- [16] Elrod, S., Hall, G., Costanza, R., Dixon, M., des Rivieres, J. Responsive Office Environments. CACM 36(7) (1993) 84-85
- [17] Rekimoto, J., Ayatsuka, Y., Hayashi, K. Augment-able Reality: Situated Communication through Physical and Digital Spaces. 2nd International Symposium on Wearable Computers (1998) 68-75
- [18] Fickas, S., Korteum, G., Segall, Z. Software Organization for Dynamic and Adaptable Wearable Systems. 1st International Symposium on Wearable Computers (1997) 56-63
- [19] Pascoe, J., Ryan, N.S., Morse, D.R. Human-Computer-Giraffe Interaction – HCI in the Field. Workshop on Human Computer Interaction with Mobile Devices. (1998)
- [20] Salber, D., Dey, A.K., Abowd, G.D. Ubiquitous Computing: Defining an HCI Research Agenda for an Emerging Interaction Paradigm. Georgia Tech Gvu Technical Report GIT-GVU-98-01 (1998)
- [21] Abowd, G.D., Dey, A.K., Orr, R., Brotherton, J. Context-Awareness in Wearable and Ubiquitous Computing. 1st International Symposium on Wearable Computers (1997) 179-180
- [22] Davies, N., Mitchell, K., Cheverst, K. Blair, G. Developing a Context Sensitive Tourist Guide. 1st Workshop on Human Computer Interaction with Mobile Devices, GIST Technical Report G98-1 (1998)
- [23] Korteum, G., Segall, Z., Bauer, M. Context-Aware, Adaptive Wearable Computers as Remote Interfaces to “Intelligent” Environments. 2nd International Symposium on Wearable Computers (1998) 58-65
- [24] Dey, A.K., Abowd, G.D. CyberDesk: The Use of Perception in Context-Aware Computing. 1st Workshop on Perceptual User Interfaces (1997) 26-27
- [25] Ryan, N. Mobile Computing in a Fieldwork Environment: Metadata Elements. Project working document, version 0.2 (1997)
- [26] Brown, P.J. Triggering Information by Context. Personal Technologies, 2(1) (1998) 1-9

- [27] Schmidt, A., Beigl, M. Gellersen, H-W. There is More to Context than Location: Environment Sensing Technologies for Adaptive Mobile User Interfaces. Workshop on Interactive Applications of Mobile Computing IMC'98 (1998)
- [28] Frazer Bennett, Tristan Richardson, and Andy Harter. Teleporting – making applications mobile. In Proceedings of IEEE Workshop on Mobile Computing Systems and Applications, pages 82-84, Santa Cruz, California, December 1994. IEEE Computer Society Press.
- [29] Andy Harter, Andy Hopper, Pete Steggles, Andy Ward, and Paul Webster. The anatomy of a context-aware application. In Proceedings of the Fifth Annual ACM/IEEE International conference on Mobile Computing and Networking, pages 59-68, Seattle, WA, August 1999. ACM Press.
- [30] Mark Weiser. Some computer science issues in ubiquitous computing. Communications of the ACM, 36(7):75-84, July 1993.
- [31] Roy Want, Bill N. Schilit, Norman I. Adams, Rich Gold, Karin Petersen, David Goldberg, John R. Ellis, and Mark Weiser. An overview of the PARCTAB ubiquitous computing experiment. IEEE Personal Communications, 2(6):28-43, December 1995.
- [32] Roy Want, Bill N. Schilit, Norman I. Adams, Rich Gold, Karin Petersen, David Goldberg, John R. Ellis, and Mark Weiser. The ParcTab Ubiquitous Computing Experiment. In Tomasz Imielinski and Henry F. Korth, editors, Mobile Computing, chapter 2, Kluwer Academic Publishers, 1996.
- [33] Abhaya Asthana, Mark Cravatts, and Paul Krzyzanowski. An indoor wireless system for personalized shopping assistance. In Proceedings of IEEE Workshop on Mobile Computing Systems and Applications, pages 69-74, Santa Cruz, California, December 1994. IEEE Computer Society Press.
- [34] Gregory D. Abowd, Christopher G. Atkeson, Jason Hong, Sue Long, Rob Kooper, and Mike Pinkerton. Cyberguide: A mobile context-aware tour guide. Wireless Networks, 3(5):421-433, October 1997.
- [35] Sue Long, Rob Kooper, Gregory D. Abowd, and Christopher G. Atkeson. Rapid prototyping of mobile context-aware applications: the Cyberguide case study. In Proceedings of the Second Annual International Conference on Mobile Computing and Networking, pages 97-107, White Plains, NY, November 1996. ACM Press.
- [36] Nigel Davies, Keith Cheverst, Keith Mitchell, and Adrian Friday. Caches in the air: Disseminating tourist information in the GUIDE system. In Proceedings of Second IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, Louisiana, February 1999. IEEE Computer Society Press.
- [37] Cheverst K, Davies N, Michell K (2002) A reflective study of the GUIDE system. In: Proceedings of the 1st workshop on mobile tourism support, pp 17–23.
- [38] Benjamin B. Bederson. Audio augmented reality: A prototype automated tour guide. In

Proceedings of Conference on Human Factors and Computing Systems, CHI '95, pages 210-211, Denver, CO, May 1995. ACM Press.

- [39] Reinhard Oppermann and Marcus Specht. Adaptive support for a mobile museum guide. In Proceedings of Workshop on Interactive Applications of Mobile Computing (IMC '98), Rostock, Germany, November 1998. Neuer Hochschulschriftverlag.
- [40] Reinhard Oppermann and Marcus Specht. A context-sensitive nomadic exhibition guide. In Proceedings of Second International Symposium on Handheld and Ubiquitous Computing, HUC 2000, pages 127-142, Bristol, UK, September 2000. Springer Verlag.
- [41] Anind K. Dey, Masayasu Futakawa, Daniel Salber, and Gregory D. Abowd. The Conference Assistant: Combining Context-Awareness with Wearable Computing. In Proceedings of the 3rd International Symposium on Wearable Computers (ISWC '99), pages 21-28, San Francisco, CA, October 1999. IEEE Computer Society Press.
- [42] Albrecht Schmidt, Kofi Asante Aidoo, Antti Takaluoma, Urpo Tuomela, Kristof Van Laerhoven, and Walter Van de Velde. Advanced interaction in context. In Proceedings of First International Symposium on Handheld and Ubiquitous Computing, HUC'99, pages 89-101, Karlsruhe, Germany, September 1999. Springer Verlag.
- [43] Siewiorek, D., A. Smailagic, J. Furukawa, N. Moraveji, K. Reiger, and J. Shaffer. SenSay: A Context-Aware Mobile Phone. In Proceedings of IEEE International Symposium on Wearable Computers (ISWC). 2003.
- [44] Hao Yan and Ted Selker. Context-aware office assistant. In Proceedings of the 2000 International Conference on Intelligent User Interfaces, pages 276- 279, New Orleans, LA, January 2000. ACM Press.
- [45] Michael D. Addlesee, Alan Jones, Finnbar Livesey, and Ferdinando Samaria. The ORL Active Floor. IEEE Personal Communications, 4(5):35-41, October 1997.
- [46] Robert J. Orr and Gregory D. Abowd. The Smart Floor: A mechanism for natural user identification and tracking. In Proceedings of the 2000 Conference on Human Factors in Computing Systems (CHI 2000), The Hague, Netherlands, April 2000. ACM Press.
- [47] Natalia Marmasse and Chris Schmandt. Location-aware information delivery with ComMotion. In Proceedings of Second International Symposium on Handheld and Ubiquitous Computing, HUC 2000, pages 157-171, Bristol, UK, September 2000. Springer Verlag.
- [48] Anind K. Dey and Gregory D. Abowd. CybreMinder: A context-aware system for supporting reminders. In Proceedings of Second International Symposium on Handheld and Ubiquitous Computing, HUC 2000, pages 172-186, Bristol, UK, September 2000. Springer Verlag.
- [49] Michael Beigl. MemoClip: A location-based remembrance appliance. Personal Technologies,4(4):230-233, September 2000.
- [50] Skov, B. and Høegh, T. Supporting information access in a hospital ward by a context-aware mobile electronic patient record. Personal Ubiquitous Comput. 10, 4 (March 2006), 205-214

- [51] Jakob E. Bardram. Applications of ContextAware Computing in Hospital Work – Examples and Design Principles. In proceedings of SAC '04, pp. 1574-1579, March 14-17, 2004, Nicosia, Cyprus
- [52] Korpipaa, P. and Mantyjarvi, J. An ontology for mobile device sensor-based context awareness. In Proceedings of CONTEXT, 2003, volume 2680 of Lecture Notes in Computer science, pages 451–458.
- [53] Panu Korpipaa, Jani Mantyjarvi, Juha Kela, Heikki Keranen, Esko-Juhani Malm, Managing context information in mobile devices, Pervasive Computing, IEEE July-Sept. 2003 Volume: 2, Issue: 3 On page(s): 42- 51, 2003.
- [54] R. Englemore and T. Morgan, eds., Blackboard Systems, Addison-Wesley, 1988.
- [55] T. Gu, H. K. Pung, D. Q. Zhang. A Service-Oriented Middleware for Building Context-Aware Services. Elsevier Journal of Network and Computer Applications (JNCA), Vol. 28, Issue 1, pp. 1-18, January 2005
- [56] OWL Web Ontology Language Overview, <http://www.w3.org/TR/owl-features>
- [57] Fahy, P. and Clarke, S. (2004). CASS – a middleware for mobile context-aware applications. In Workshop on Context Awareness, MobiSys 2004.
- [58] Harry Chen, Tim Finin, Anupam Joshi, "A Context Broker for Building Smart Meeting Rooms", In Proceedings of the Knowledge Representation and Ontology for Autonomous Systems Symposium, 2004 AAAI Spring Symposium, Stanford CA., March 2004, 2004.
- [59] Dey, A. K. and Abowd, G. D. (2001). A conceptual frame work and a toolkit for supporting rapid prototyping of context-aware applications. Human-Computer Interactions (HCI) Journal, 16(2-4):7–166.
- [60] Dey, A. K. and Mankoff, J. 2005. Designing mediation for context-aware applications. ACM Trans. Comput.-Hum. Interact. 12, 1 (Mar. 2005), 53-80
- [61] Hofer, T., Schwinger, W., Pichler, M., Leonhartsberger, G., and Altmann, J. (2002). Context awareness on mobile devices - the hydrogen approach. In Proceedings of the 36th Annual Hawaii International Conference on System Sciences, pages 292–302.
- [62] Biegel, G. and Cahill, V. (2004). A framework for developing mobile, context-aware applications. In Proceedings of the 2nd IEEE Conference on Pervasive Computing and Communication.
- [63] M. Roman, C. K. Hess, R. Cerqueira, R. H. Campbell, and K. Narhstedt, "Gaia: A Middleware Infrastructure to Enable Active Spaces," IEEE Pervasive Computing Magazine, vol. 1, pp. 74-83, 2002.
- [64] Mobile Gaia: A Middleware for Ad-hoc Pervasive Computing. Shiva Chetan, Jalal Al-Muhtadi, Roy Campbell and M.Dennis Mickunas, In IEEE Consumer Communications & Networking Conference (CCNC 2005) , Las Vegas, Jan. 2005.
- [65] J. E. Bardram. The Java Context Awareness Framework (JCAF)—a service

infrastructure and programming framework for context-aware applications. In H. Gellersen, R. Want, and A. Schmidt, editors, Proceedings of the 3rd International Conference on Pervasive Computing (Pervasive 2005), Lecture Notes in Computer Science, Munich, Germany, May 2005. Springer Verlag.

- [66] David, P. and Ledoux, T. 2005. WildCAT: a generic framework for context-aware applications. In Proceedings of the 3rd international Workshop on Middleware For Pervasive and Ad-Hoc Computing (Grenoble, France, November 28 - December 02, 2005). MPAC '05, vol. 115. ACM Press, New York.
- [67] Simple Object Access Protocol 1.2, <http://www.w3.org/TR/2003/REC-soap12-part0-20030624/>
- [68] Java ME, <http://java.sun.com/javame/index.jsp>
- [69] Apache AXIS, <http://ws.apache.org/axis/>
- [70] Apache Tomcat, <http://tomcat.apache.org/>
- [71] JDBC API, <http://java.sun.com/products/jdbc/overview.html>
- [72] Open Mobile Alliance, Mobile Location Protocol Specification, v. 3.1, http://www.openmobilealliance.org/release_program/docs/MLP/OMA-LIF-MLP-V3_1-20040316-C.pdf
- [73] 3GPP, TS 23.271, Functional Stage 2 Description of LCS, release 7, December 2006.
- [74] A. Papadakis, E. Chaniotakis, N. Tselikas, G. Leoleis, I. Venieris “Parlay and Mobile Agents is a Homogenized Service Provision Architecture”, on proceedings of European Conference on Universal Multiservice Networks (ECUMN02), Colmar, France, April 2002
- [75] A. Papadakis, E. Chaniotakis, P. Giannakakis, N. Tselikas, I. Venieris “Parlay-based Service Provision in Circuit- and Packet-switched Telecommunications Networks” International Journal of Communication Systems, IJCS, Vol.17 (1), pp. 63-83, Jan 2004.
- [76] The Parlay Group. <http://www.parlay.org>
- [77] Νικόλαος Δ. Τσελίκας, “Ενοποιημένη και Ευφυής Μεταγωγή Υπηρεσιών και Έλεγχος Κλήσεων για Σταθερά, Κινητά και Διαδικτυακά Περιβάλλοντα”, Διδακτορική διατριβή, Εθνικό Μετσόβιο Πολυτεχνείο, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Η/Υ, Τομέας Συστημάτων Μετάδοσης και Τεχνολογίας Υλικών, Απρίλιος 2004.
- [78] OSA/Parlay Mobility API specifications, http://www.parlay.org/en/specifications/apis_archives.asp
- [79] N. Tselikas, E. Koutsoloukas, S. Kapellaki, E. Chaniotakis and I. Venieris, “An OSA/Parlay based Middleware architecture for Location Based Services”, Journal of Personal Wireless Communications, Volume 30, Issue 2 - 4, p.p. 247 – 265 Sep. 2004, (Kluwer).
- [80] S. Kapellaki, N. Tselikas, E. Chaniotakis, E. Koutsoloukas, G. Prezerakos and I. Venieris, "The “Le” interface: Performance Evaluation of 2-tier and 3-tier 3GPP compliant realizations", on proceedings of IEEE ICC 2005, (IEEE International Conference on Communications), pp. 1423-1427, May 18, 2005 Seoul (Korea).

2ο Κεφάλαιο

Μοντελοποίηση πληροφορίας πλαisiού χρήσης υπηρεσίας

Στο παρόν κεφάλαιο παρατίθεται μία επισκόπηση των προσεγγίσεων μοντελοποίησης της πληροφορίας πλαisiού χρήσης υπηρεσίας και περιγράφονται τα μειονεκτήματα και τα πλεονεκτήματα της καθεμιάς. Εν συνέχεια, περιγράφεται η προσέγγιση που ακολουθήθηκε στο πλαίσιο της παρούσας διατριβής, η οποία πραγματοποιήθηκε σε δύο στάδια. Το πρώτο ήταν αυτό του καθορισμού του μοντέλου πληροφορίας των εμπλεκόμενων οντοτήτων με χρήση του προτύπου Common Information Model της Distributed Management Task Force και το δεύτερο ο καθορισμός του μοντέλου δεδομένων, όπου ακολουθήθηκε markup-scheme προσέγγιση βασισμένη στο πρότυπο Genetic User Profile της 3GPP, καθώς και άλλου τέτοιου είδους πρότυπα ανάλογα με τις κάθε φορά προκύπτουσες ανάγκες.

2.1 Εισαγωγή

Ο βασικός και θεμέλιος λίθος κάθε συστήματος ανάκτησης και παροχής πληροφορίας πλαισίου χρήσης υπηρεσίας είναι το μοντέλο πληροφορίας που ακολουθεί. Το μοντέλο πληροφορίας αποτελεί ουσιαστικά τον ορισμό και τον καθορισμό των πληροφοριών που δύναται να διαχειριστεί το σύστημα και αποτελεί τη βάση πάνω στην οποία χτίζεται η λειτουργικότητα του. Επίσης, το μοντέλο είναι απαραίτητο για λόγους διαλειτουργικότητας, προκειμένου να υπάρχει κοινή ονοματολογία και κοινά σημεία αναφοράς για τις οντότητες που αλληλεπιδρούν για την ανάκτηση και εκμετάλλευση των πληροφοριών.

Βασικά χαρακτηριστικά ενός επιτυχημένου μοντέλου πληροφορίας είναι η λεπτομερής και μεθοδική δομή του, ώστε να διευκολύνεται ο έλεγχος και η εκμετάλλευση του context, η δυνατότητα ανταλλαξιμότητας της πληροφορίας ανάμεσα στις εμπλεκόμενες οντότητες, η δυνατότητα οι πληροφορίες context να συντίθενται, αποσυντίθενται και να επανασυντίθενται με καταναμημένο τρόπο, η ύπαρξη ομοιομορφίας στην αναπαράσταση των διαφορετικών ειδών πληροφορίας, η επεκτασιμότητα και η χρήση προτύπων για τον καθορισμό των μοντέλων.

Οι προσεγγίσεις που έχουν ακολουθηθεί κατά καιρούς είναι πολλές και οι σημαντικότερες είναι τα μοντέλα key-value, τα markup-scheme, τα γραφικά μοντέλα, τα αντικειμενοστραφή, τα λογικά και τα μοντέλα βασισμένα σε οντολογίες. Η επόμενη ενότητα περιλαμβάνει μία παρουσίαση και αποτίμηση των προσεγγίσεων αυτών.

Στο πλαίσιο της παρούσας διατριβής και για τις ανάγκες σχεδιασμού και υλοποίησης του συστήματος, που παρουσιάζεται στο επόμενο κεφάλαιο, καθορίστηκε το αντίστοιχο μοντέλο πληροφορίας σε δύο στάδια, χρησιμοποιώντας δυο διαφορετικές προσεγγίσεις. Στο πρώτο στάδιο καθορίστηκε το μοντέλο πληροφορίας του συστήματος. Αποτελεί γραφικό μοντέλο, βασισμένο στο πρότυπο Common Information Model (CIM) της Distributed Management Task Force (DTMF) [1], το οποίο για την γραφική αναπαράσταση μοντέλων χρησιμοποιεί την Unified Modeling Language (UML) [2]. Σε δεύτερο επίπεδο ακολουθήθηκε markup-scheme προσέγγιση, για τον καθορισμό του μοντέλου δεδομένων του συστήματος, που βασίστηκε στο Generic User Profile της 3GPP [3], αλλά και σε άλλα τέτοιου είδους πρότυπα.

Στην επόμενη ενότητα παρουσιάζεται μια επισκόπηση των μεθόδων μοντελοποίησης πληροφορίας context και ορισμένες εφαρμογές τους και στη συνέχεια παρατίθεται το προτεινόμενο από την διατριβή μοντέλο.

2.2 Προσεγγίσεις Μοντελοποίησης της Πληροφορίας Πλαισίου Χρήσης Υπηρεσίας

Η μοντελοποίηση της πληροφορίας πλαισίου χρήσης μίας υπηρεσίας αποτελεί την βασική προαπαιτήση για την εκμετάλλευση της από τις υπηρεσίες και τις εφαρμογές. Αρχικά, οι πληροφορίες context συλλέγονται από τις πηγές τους. Οι πηγές των πληροφοριών είναι πολλές και διαφορετικές, όπως αισθητήρες που παρέχουν δεδομένα για το φυσικό περιβάλλον, πληροφορίες που παρέχει απευθείας ο χρήστης ή ακόμα πληροφορίες που προέρχονται από λογικούς αισθητήρες, όπως κάποιο λογισμικό ή μία βάση δεδομένων. Συνεπώς η αρχική μορφή των πληροφοριών συμφωνεί και εξαρτάται κάθε φορά από την πηγή που προέρχεται. Προκειμένου η πληροφορία να είναι εκμεταλλεύσιμη απαιτείται κατάλληλη επεξεργασία, έτσι ώστε η μορφή και η περιγραφή της να είναι κατανοητή από τις υπηρεσίες. Εν ολίγοις απαιτείται η μοντελοποίηση της.

Ωστόσο το εγχείρημα της μοντελοποίησης της πληροφορίας context θεωρείται, εκτός από απαραίτητο, και εξαιρετικά πολύπλοκο για πολλούς λόγους. Οι πηγές και τα είδη της πληροφορίας είναι πολλές, εξαιρετικά ανομοιογενείς και με διαφορετικά χαρακτηριστικά. Υπάρχει η δυνατότητα εναλλακτικών τρόπων παρουσίασης της πληροφορίας, καθώς και πολλά επίπεδα διύλισης της, καθώς η πληροφορία, από την στιγμή που συλλέγεται από την πηγή της μέχρι να φτάσει στην υπηρεσία, περνάει από διάφορα στάδια επεξεργασίας, ερμηνείας, μετάφρασης για την εξαγωγή υψηλότερου επιπέδου πληροφορίας. Επίσης, μπορούν να γίνουν συσχετισμοί και συνδυασμοί των πληροφοριών context, εκ των οποίων να εξάγονται χρήσιμα συμπεράσματα, τα οποία να αποτελούν περαιτέρω πληροφορία context.

Προκειμένου λοιπόν να αντιμετωπιστούν και να καλυφθούν οι απαιτήσεις μοντελοποίησης πρέπει το μοντέλο να έχει τα παρακάτω χαρακτηριστικά [4]:

- *Δομή*: Η δομημένη αναπαράσταση δίνει την δυνατότητα οργάνωσης, διύλισης, ελέγχου και ερμηνείας της πληροφορίας με πιο αποτελεσματικό τρόπο.
- *Ανταλλαξιμότητα (Interchangeability)*: Η δυνατότητα σειριοποιημένης αναπαράστασης του context επιτρέπει την μεταφορά και ανταλλαγή του εντός ενός συστήματος.
- *Σύνθεση – Αποσύνθεση – Επανασύνθεση* : Εν γένει τα συστήματα παροχής υπηρεσιών με επίγνωση του πλαισίου χρήσης τους αποτελούν καταναμημένα συστήματα στα οποία δεν υπάρχει κεντρικό σημείο ανάκτησης και διαχείρισης του context. Η δυνατότητα σύνθεσης, αποσύνθεσης και επανασύνθεσης του context

επιτρέπει την διαχείριση, αποθήκευση και εκμετάλλευση του context σε τέτοιου είδους καταναμημένα περιβάλλοντα.

- *Ομοιομορφία:* Η ομοιόμορφη αναπαράσταση των πληροφοριών context, είτε αυτή αφορά το χρήστη, το τερματικό ή το δίκτυο, διευκολύνει την επεξεργασία και τη χρήση τους για προσαρμογή των υπηρεσιών.
- *Επεκτασιμότητα:* Αποτελεί απαραίτητο χαρακτηριστικό προκειμένου το σύστημα να είναι ευέλικτο και να μπορεί να καλύψει μελλοντικές ανάγκες για εμπλουτισμό των πληροφοριών context.
- *Προτυποποίηση:* Η προτυποποίηση και η χρήση προτύπων για την ανάπτυξη ενός μοντέλου context διευκολύνει την υιοθέτηση του από περισσότερα εμπλεκόμενα μέρη στην αλυσίδα παροχής υπηρεσιών με επίγνωση του πλαισίου χρήσης τους. Επίσης, διευκολύνει την εφαρμογή του σε ήδη υπάρχουσες υποδομές όπως για παράδειγμα είναι οι Web Services [5].

Στο [6] παρουσιάζεται και αποτιμάται μία σειρά προσεγγίσεων μοντελοποίησης πληροφορίας context.

Μοντέλα key-value:

Τα μοντέλα *key-value* μοντελοποιούν την πληροφορία context σαν μεταβλητή. Η δομή τους είναι από τις πιο απλές, με αποτέλεσμα να διευκολύνει την διαχείριση της πληροφορίας. Ωστόσο, αυτού του είδους η μοντελοποίηση δεν είναι τόσο αποτελεσματική όταν πρόκειται για πιο σύνθετες μορφές και δομές πληροφορίας. Συχνά αυτή η προσέγγιση χρησιμοποιείται σε καταναμημένα περιβάλλοντα παροχής υπηρεσιών [7][8]. Συνήθως, σε αυτές τις περιπτώσεις οι υπηρεσίες περιγράφονται από ένα σύνολο στοιχείων (attributes) με μορφή ζεύγων key-value. Κατάλληλοι μηχανισμοί ανακάλυψης υπηρεσιών εκτελούν αλγορίθμους ακριβούς αντιστοίχισης των στοιχείων αυτών με τα αντίστοιχα ζεύγη key-value του context. Δεν παρέχουν σχήματα ή ορισμούς της πληροφορίας σε αφηρημένο επίπεδο με αποτέλεσμα να μην είναι δυνατός ο έλεγχος (validation) του μοντέλου. Τέλος, η σύνθεση και αποσύνθεση του context για χρήση του σε καταναμημένα περιβάλλοντα παρέχεται μόνο σε επίπεδο στιγμιότυπου (instance).

Μοντέλα Markup-Scheme:

Μία άλλη προσέγγιση αποτελούν τα μοντέλα *Markup-Scheme*. Ακολουθούν ιεραρχική δομή και συνήθως ορίζουν διάφορους τύπους προφίλ. Στην πλειοψηφία τους οι προσεγγίσεις αυτές καλύπτουν τις ανάγκες του εκάστοτε συστήματος για το οποίο έχουν οριστεί και χρησιμοποιούν μικρό μέρος των πληροφοριών context, χωρίς αυτό να σημαίνει ότι δεν

μπορούν να καλύψουν πιο γενικευμένη και ευρεία γκάμα. Ωστόσο όμως, παρέχουν κατάλληλα σχήματα (schema) για τον έλεγχο (validation) της πληροφορίας και βασίζονται σε τεχνολογίες και εργαλεία ευρέως αποδεκτά και προτυποποιημένα διευκολύνοντας έτσι την υιοθέτησή τους και από υπάρχοντα συστήματα. Όσον αφορά στην σύνθεση – αποσύνθεση ο βαθμός στον οποίο επιτυγχάνεται εξαρτάται κάθε φορά από την υλοποίηση του μοντέλου. Το γεγονός αυτό μπορεί να ερμηνευτεί αρνητικά, θεωρώντας τα μοντέλα προσανατολισμένα στην εκάστοτε εφαρμογή, αλλά και θετικά, ερμηνεύοντας το χαρακτηριστικό αυτό ως χαρακτηριστικό ανοιχτότητας και ευελιξίας που δίνει την δυνατότητα στον προγραμματιστή να προσαρμόσει το μοντέλο ανάλογα με τις ανάγκες που υπάρχουν. Τέτοιες προσεγγίσεις αφορούν κυρίως πληροφορίες προφίλ. Ορισμένα παραδείγματα τέτοιων προσεγγίσεων αποτελούν τα *Composite Capabilities/Preference Profile (CC/PP)* [9], *User Agent Profile (UAProf)* [10], *Comprehensive Structured Context Profiles (CSCP)* [3], *CC/PP Context Extension* [11], *Pervasive Profile Description Language (PPDL)* [12] και *ConteXtML* [13].

Γραφικά Μοντέλα:

Μία μέθοδος μοντελοποίησης σε μορφή πλήρως κατανοητή από τον άνθρωπο είναι τα *γραφικά μοντέλα* (graphical models) τα οποία κυρίως χρησιμοποιούνται για τον καθορισμό της δομής της πληροφορίας και από το οποία εξάγεται κάποιος/μερικός κώδικας ή μοντέλα ER (Entity-Relationship Model) [14]. Το χαρακτηριστικό της σύνθεσης της πληροφορίας παρουσιάζει περιορισμούς σε επίπεδο δομής διότι η ενοποίηση σε αφηρημένο επίπεδο του μοντέλου είναι λιγότερο αποτελεσματική απ' ό τι σε επίπεδο στιγμιότυπου (instance). Έλεγχος της πληροφορίας είναι δυνατός ως ένα βαθμό. Η μέθοδος αυτή επίσης δεν ενδείκνυται για τη μοντελοποίηση πληροφοριών context όπως π.χ. στοιχεία ιστορικού (histories), αβέβαιες ή ημιτελείς πληροφορίες, πληροφορίες με διαφορετικά χαρακτηριστικά ή αλληλεξαρτήσεις μεταξύ πληροφοριών. Σαν γενικό συμπέρασμα, η συγκεκριμένη μέθοδος ενδείκνυται για κατανοητή από τον άνθρωπο δόμηση της πληροφορίας (human readable format) και όχι για την εκμετάλλευση της από μηχανές (υπολογιστές, computer readable format) Παραδείγματα τέτοιου είδους μοντελοποιήσεων αποτελούν η *Unified Modeling Language (UML)* [15], το *Contextual Extended ORM (Object Role Modeling)* [16][17] και η *ContextUML* [18].

Αντικειμενοστρεφή Μοντέλα:

Τα *αντικειμενοστρεφή μοντέλα* (Object Oriented Models) αποτελούν μία άλλη εναλλακτική λύση μοντελοποίησης των πληροφοριών πλαισίου χρήσης υπηρεσίας. Ο στόχος τους είναι να εκμεταλλευτούν τις κλασικές ιδιότητες των αντικειμενοστραφών προσεγγίσεων, όπως η

επαναχρησιμοποίηση (reusability) και η ενσωμάτωση (encapsulation), προκειμένου να αντιμετωπίσουν τα προβλήματα που προκύπτουν από τη δυναμική φύση του context. Οι λεπτομέρειες της επεξεργασίας του context πραγματοποιούνται σε επίπεδο αντικειμένου και η πρόσβαση σε αυτό επιτρέπεται μέσω κατάλληλων διεπαφών. Όσον αφορά στην κατανομημένη σύνθεση του context είναι δυνατή, καθότι τα αντικειμενοστραφή συστήματα δύνανται να διαχειρίζονται νέους τύπους πληροφορίας που ορίζονται με τη μορφή νέων κλάσεων ή/και αντικειμένων. Ο έλεγχος αυτών των μοντέλων πραγματοποιείται με χρήση μεταγλωττιστών (compiler) σε επίπεδο δομής και σε επίπεδο στιγμιότυπου μέσω του τρέχοντος περιβάλλοντος (runtime environment). Τα αντικειμενοστραφή μοντέλα παρέχουν τη δυνατότητα ενσωμάτωσής τους σε υπάρχουσες υποδομές, αλλά οι απαιτήσεις τους σε υπολογιστικούς πόρους πολλές φορές δεν μπορούν να καλυφθούν από τερματικά με χαμηλές επεξεργαστικές ικανότητες, π.χ. κινητά τηλέφωνα και PDA, που εν γένει κυριαρχούν στον τομέα των context aware συστημάτων. Παραδείγματα τέτοιων προσεγγίσεων αποτελούν οι *cues* [19] που υλοποιήθηκαν στο πλαίσιο του προγράμματος *TEA* [20][21] και το *Active Object Model* [22] του προγράμματος *GUIDE*.

Λογικά Μοντέλα:

Άλλο ένα είδος μοντέλων που συναντάται στην βιβλιογραφία είναι τα *Λογικά Μοντέλα* (Logic Based Models). Σε αυτήν την περίπτωση οι πληροφορίες context μοντελοποιούνται ως γεγονότα (facts), εκφράσεις (expressions) και κανόνες (rules). Με χρήση κανόνων λογικής εξάγονται συμπεράσματα με τη μορφή γεγονότων ή εκφράσεων τα οποία προκύπτουν από άλλα γεγονότα ή εκφράσεις υπό συγκεκριμένες προϋποθέσεις. Τα λογικά μοντέλα παρέχουν δυνατότητα κατανομημένης σύνθεσης του context. Ο έλεγχος (validation) του μοντέλου είναι σχετικά δύσκολο να πραγματοποιηθεί και κατά συνέπεια, παρόλο που ο καθορισμός της δομής τους είναι εξαιρετικά λεπτομερής και μεθοδικός, η εμφάνιση λαθών είναι πολλή πιθανή. Το μεγαλύτερο ίσως μειονέκτημα των συγκεκριμένων μοντέλων είναι η εφαρμογή τους σε υπάρχουσες υποδομές, οι οποίες στην πλειοψηφία τους δεν υποστηρίζουν μηχανές εξαγωγής λογικών συμπερασμάτων (logic reasoners). Παραδείγματα τέτοιων μοντέλων αποτελούν τα: *Multicontext Systems* [23], *Extended Situation Theory* [24], *Sensed Context Model* [25].

Μοντέλα βασισμένα σε Οντολογίες:

Τέλος, τα μοντέλα βασισμένα σε οντολογίες (Ontology Based Models) αποτελούν τις πιο πρόσφατες προσπάθειες μοντελοποίησης του context. Μία οντολογία αποτελεί ένα μοντέλο δεδομένων που αναπαριστά ένα σύνολο από έννοιες εντός ενός πεδίου (domain) και τις σχέσεις μεταξύ τους. Στο πεδίο των πληροφοριών context, οι οντολογίες αποτελούν πολλά

υποσχόμενο εργαλείο [26][27] για την αναπαράσταση πληροφοριών που περιγράφουν και χρησιμοποιούνται στην καθημερινή ζωή, με τη μορφή δομών δεδομένων. Μέσω των οντολογιών ορίζονται έννοιες, υπό-έννοιες και συσχετισμοί οι οποίοι αποτελούν κοινά αποδεκτή ονοματολογία από τις εμπλεκόμενες οντότητες διευκολύνοντας έτσι την από κοινού χρήση και επαναχρησιμοποίηση του context [28]. Επιπροσθέτως οι οντολογίες επιτρέπουν και την εξαγωγή συμπερασμάτων (inference). Παραδείγματα τέτοιων προσεγγίσεων αποτελούν οι οντολογίες *CoOL* [29], *CONON* [30], *CoBRA* [31] και *SOUPA* [32].

Από τις παραπάνω περιγραφές των διαφόρων προσεγγίσεων μοντελοποίησης που βρίσκουν εφαρμογή και στο πεδίο των συστημάτων με επίγνωση του πλαισίου χρήσης τους, διαπιστώνεται ότι καμία δεν ικανοποιεί πλήρως όλες τις προϋποθέσεις για ένα πλήρες και ολοκληρωμένο μοντέλο context. Καθεμία χαρακτηρίζεται από πλεονεκτήματα και μειονεκτήματα σε διαφορετικούς τομείς κάθε φορά. Συνεπώς, η επιλογή του τρόπου μοντελοποίησης της πληροφορίας context, παρόλο που το ζητούμενο θεωρητικά είναι να είναι ανεξάρτητη συστήματος και εφαρμογής, δεν είναι τελικά δυνατόν να επιλεγεί χωρίς να ληφθούν, τουλάχιστον έως έναν βαθμό, υπ' όψιν και οι απαιτήσεις που τίθενται από το ίδιο το σύστημα και από τις λειτουργίες που καλείται να παρέχει.

Στην επόμενη ενότητα παρουσιάζεται η προσέγγιση που ακολουθήθηκε στο πλαίσιο της παρούσας διατριβής και τεκμηριώνονται οι λόγοι που οδήγησαν στη συγκεκριμένη επιλογή.

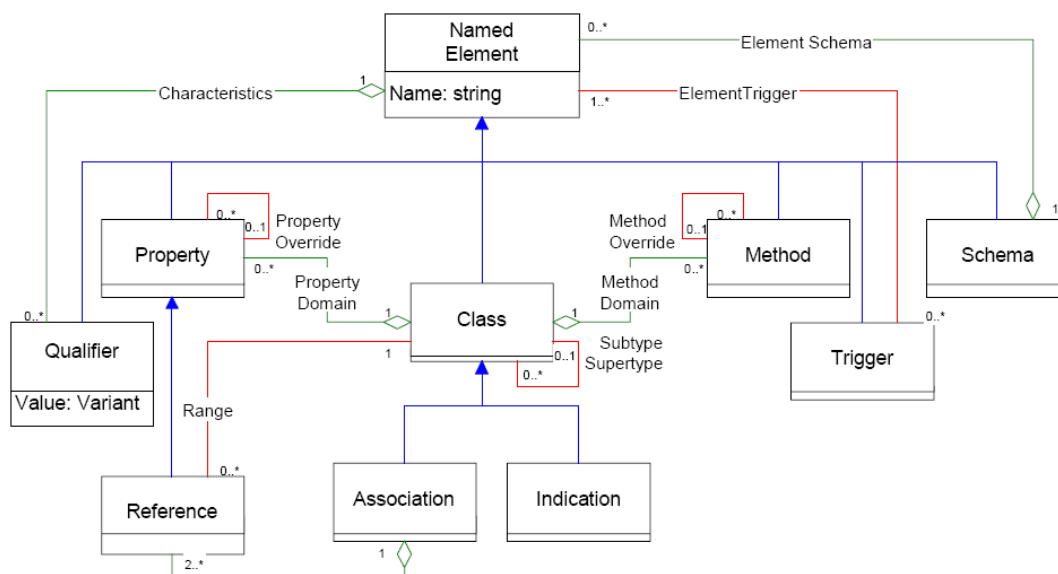
2.3 Τα Μοντέλα Πληροφορίας και Δεδομένων

Η προσέγγιση που ακολουθήθηκε στο πλαίσιο της παρούσας διατριβής ακολούθησε δύο στάδια. Στο πρώτο στάδιο καθορίστηκε το Μοντέλο Πληροφορίας (Information Model) και στο δεύτερο το Μοντέλο Δεδομένων (Data Model). Ως μοντέλο πληροφορίας ορίζουμε την αφηρημένη αναπαράσταση των οντοτήτων εντός ενός περιβάλλοντος, συμπεριλαμβανομένων των ιδιοτήτων, λειτουργιών και σχέσεων μεταξύ τους, η οποία είναι ανεξάρτητη τεχνολογιών, πρωτοκόλλων, εφαρμογών και λειτουργικών συστημάτων. Το μοντέλο δεδομένων από την άλλη αποτελεί την πραγματική αναπαράσταση ενός μοντέλου πληροφορίας βάσει συγκεκριμένων τεχνολογιών και εργαλείων [33].

2.3.1 Το Μοντέλο Πληροφορίας

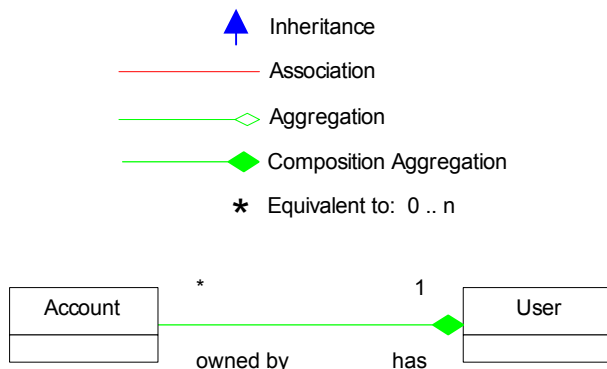
Για τον καθορισμό του μοντέλου πληροφορίας χρησιμοποιήθηκε η μεθοδολογία, η σύνταξη και οι έννοιες (semantics) του προτύπου Common Information Model (CIM) της Distributed Management Task Force (DTMF) [1]. Το CIM είναι ένα σχήμα (schema) που αποτελείται

από ένα σύνολο μοντέλων (core και common) για την περιγραφή των οντοτήτων, των καταστάσεων, της διαμόρφωσης και των σχέσεων τους εντός υπολογιστικών και δικτυακών περιβαλλόντων. Το περιεχόμενο του μοντέλου είναι ανεξάρτητο πεδίου (domain) και υλοποίησης και σκοπός του είναι η διευκόλυνση και διευθέτηση της από-άκρο-σε-άκρο διαχείρισης των οντοτήτων ενός συστήματος. Αποτελεί μία ιεραρχική και αντικειμενοστραφή αρχιτεκτονική με αποτέλεσμα να διευκολύνει τον εντοπισμό και να αναπαριστά επιτυχώς και με ευκολία τις πολύπλοκες αλληλοεξαρτήσεις και σχέσεις μεταξύ των οντοτήτων. Χρησιμοποιεί το Managed Object Format (MOF) [1] που βασίζεται στην Interface Definition Language (IDL) [34] για την περιγραφή των αντικειμένων σε μορφή κειμένου (textual description), ενώ για την γραφική αναπαράσταση των μοντέλων χρησιμοποιεί την Unified Modeling Language (UML) [2]. Επίσης, καθορίζει ένα μετα-σχήμα (Meta-schema) στο οποίο ορίζονται οι όροι που χρησιμοποιούνται για την έκφραση των μοντέλων καθώς και η χρήση και οι έννοιες του. Στο Σχήμα 14 παρουσιάζεται η δομή του μετα-σχήματος.



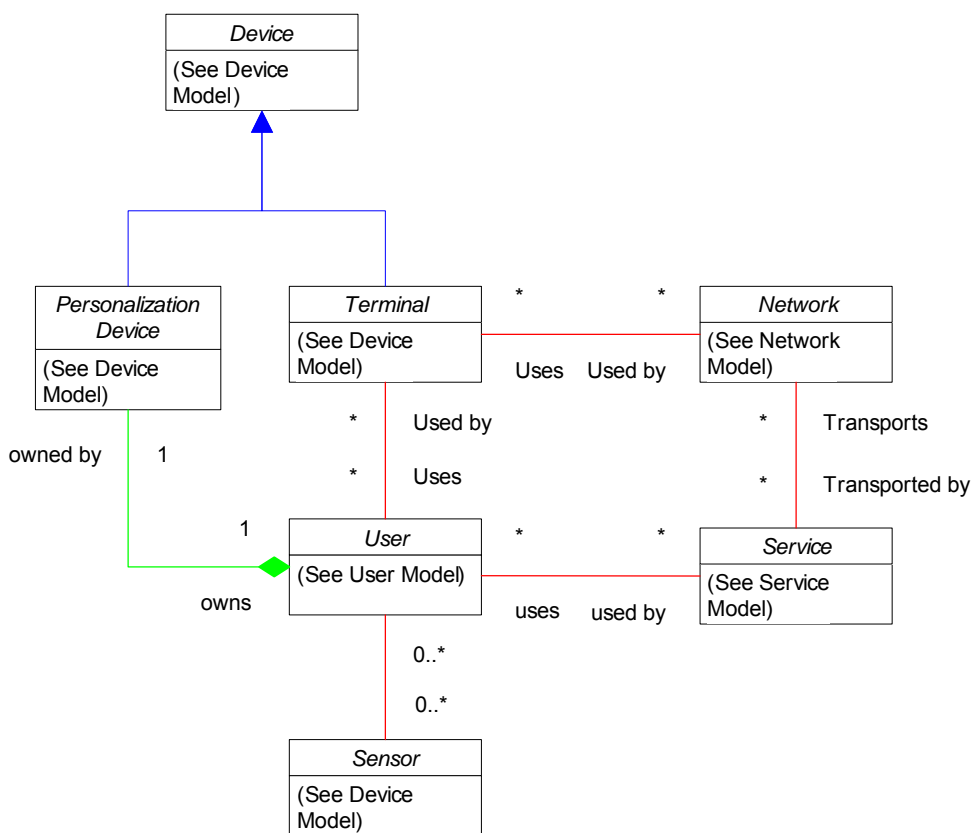
Σχήμα 14: Η δομή του μετα-σχήματος

Ακολουθώντας λοιπόν την σύνταξη και τις συμβάσεις του CIM ορίστηκε το μοντέλο πληροφορίας του συστήματος το οποίο αποτελεί κατ' ουσία την οντολογία του [35]. Η αναπαράσταση της οντολογίας βασίζεται στην UML, όπως και το CIM και οι συμβάσεις που ακολουθήθηκαν παρουσιάζονται στο Σχήμα 15.



Σχήμα 15: Συμβάσεις σχέσεων

2.3.1.1 Το μοντέλο πυρήνα (core model)

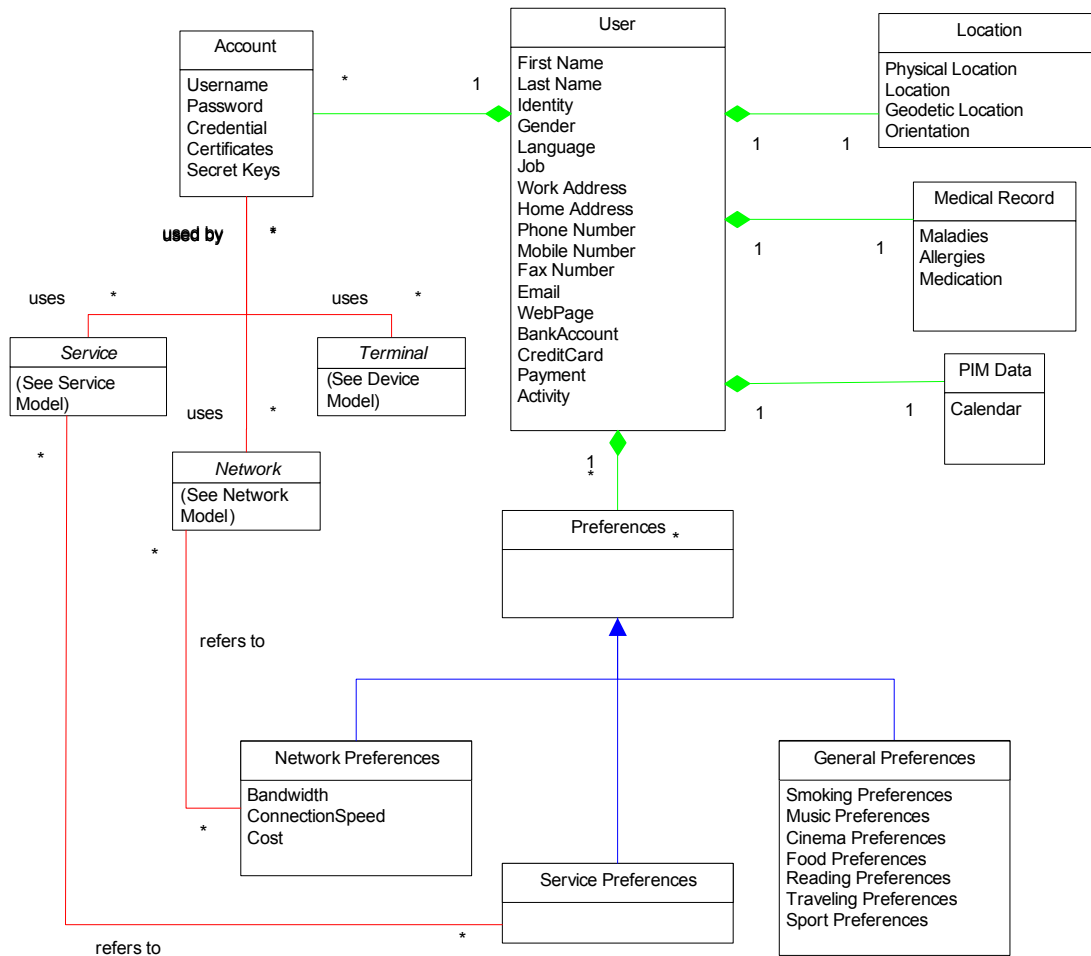


Σχήμα 16: Το μοντέλο πυρήνα

Το μοντέλο πυρήνα αναπαριστά τις βασικές οντότητες του υπό σχεδίαση συστήματος και παρουσιάζεται στο Σχήμα 16. Η βασική οντότητα είναι ο χρήστης ο οποίος μπορεί να χρησιμοποιεί απεριόριστο αριθμό τερματικών μέσω των οποίων συνδέεται στα διαθέσιμα δίκτυα και χρησιμοποιεί τις υπηρεσίες που παρέχονται. Επίσης ο χρήστης έχει στην κατοχή του μία Προσωπική Συσκευή (Personalization Device). Η Προσωπική Συσκευή Χρήστη αποτελεί ένα από τα δομικά στοιχεία της προτεινόμενης αρχιτεκτονικής και θα παρουσιαστεί με περισσότερη λεπτομέρεια στο επόμενο κεφάλαιο. Για λόγους καλύτερης

κατανόησης αναφέρουμε σε αυτό το σημείο ότι η συσκευή αυτή περιέχει προσωπικά δεδομένα του χρήστη και συνεπώς αποτελεί μία σημαντική πηγή πληροφοριών context. Τα τερματικά αλλά και οι προσωπικές συσκευές αποτελούν εξειδικεύσεις συσκευών (Device). Στις επόμενες ενότητες παρουσιάζονται τα μοντέλα πληροφορίας που περιγράφουν τις επιμέρους κλάσεις του μοντέλου πυρήνα.

2.3.1.2 Το μοντέλο χρήστη



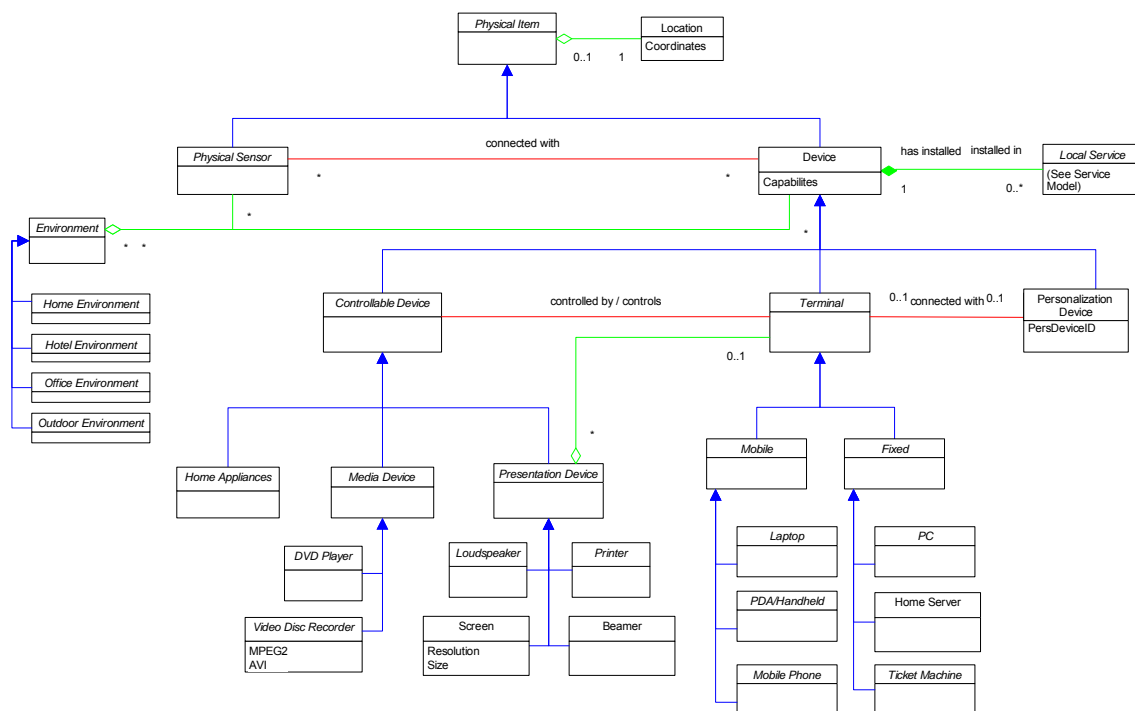
Σχήμα 17: Το μοντέλο χρήστη

Το μοντέλο πληροφορίας χρήστη (Σχήμα 17) περιέχει όλες εκείνες τις πληροφορίες που μοναδικά τον χαρακτηρίζουν. Ωστόσο παραμένει ανοικτό για περαιτέρω προσθήκες επιπλέον πληροφοριών.

Ο χρήστης πρωτίστως χαρακτηρίζεται από δεδομένα όπως αυτά που περιέχονται στην κλάση *User*, στα οποία αναφερόμαστε ως προσωπικά στοιχεία ή στοιχεία ταυτότητας. Από το μοντέλο πυρήνα διαφαίνεται ότι ο χρήστης έχει στην κατοχή του τερματικά και χρησιμοποιεί διαθέσιμα δίκτυα και υπηρεσίες. Συνεπώς ο χρήστης έχει διάφορους λογαριασμούς (*Account*) για την πρόσβασή του σε δίκτυα, υπηρεσίες, που είναι

εγγεγραμμένοι, και τερματικά. Από τα σημαντικότερα στοιχεία της κατάστασης του χρήστη είναι η θέση του (*Location*). Επίσης, οι χρήστες διαφοροποιούνται μεταξύ τους και βάσει των προτιμήσεών τους, είτε αυτές σχετίζονται με τεχνικά ζητήματα, όπως για παράδειγμα οι προτιμήσεις του χρήστη για κόσθη ή εύρος ζώνης, αλλά και προσωπικές προτιμήσεις όσον αφορά τυχόν δραστηριότητες και ενδιαφέροντα, όπως π.χ. μουσική, βιβλία κλπ. Σημαντικά στοιχεία επίσης αποτελούν τα ιατρικά δεδομένα αν λάβουμε υπόψη ότι οι context-aware εφαρμογές στο πεδίο αυτό είναι ευρέως διαδεδομένες. Επιπροσθέτως, το ημερολόγιο του χρήστη αποτελεί σημαντική πηγή πληροφοριών για την τρέχουσα κατάστασή του οι οποίες είναι απαραίτητες στην περίπτωση υπηρεσιών με πεδίο εφαρμογής τον χώρο και τις διαδικασίες εργασίας. Φυσικά δεν είναι δυνατόν σε ένα διάγραμμα UML να συμπεριληφθούν όλες οι σχετικές με τον χρήστη πληροφορίες που σε μία δεδομένη στιγμή θα απαιτηθούν για την εκτέλεση μίας υπηρεσίας. Για τον λόγο αυτό τονίζεται ότι το διάγραμμα αυτό καθώς και τα υπόλοιπα που θα παρουσιαστούν στη συνέχεια θεωρούνται επεκτάσιμα και σε επίπεδο υλοποίησης λαμβάνεται μέριμνα προκειμένου τυχόν επεκτάσεις σε επίπεδο μοντελοποίησης να μπορούν να ενσωματωθούν με ελάχιστο προγραμματιστικό κόστος.

2.3.1.3 Το μοντέλο συσκευών



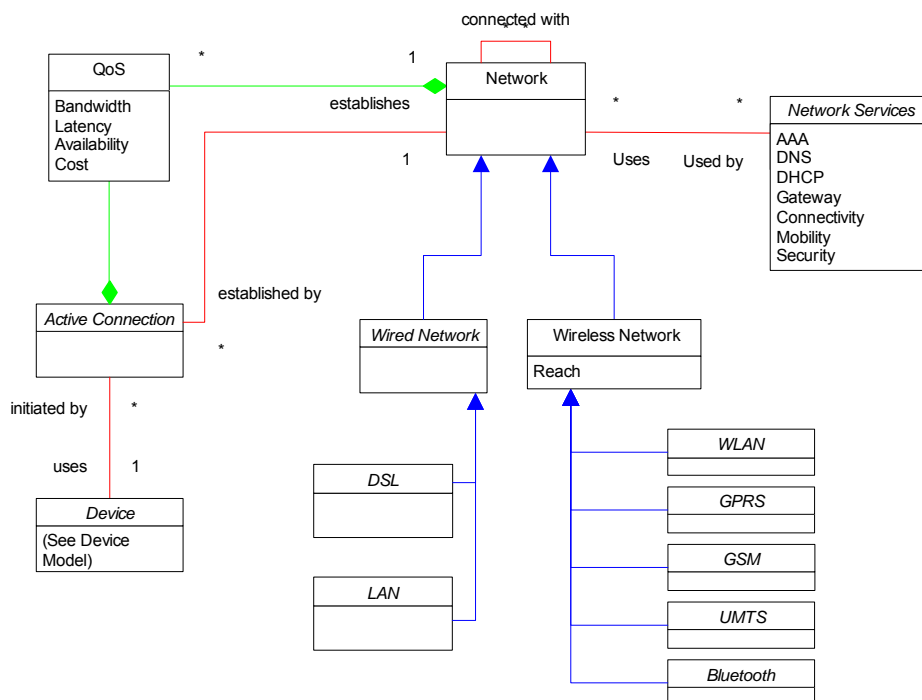
Σχήμα 18: Το μοντέλο συσκευών

Το μοντέλο συσκευών (Σχήμα 18) αποτελεί μία καταγραφή των αντικειμένων εντός ενός χώρου τα οποία μπορούν να λειτουργήσουν σαν πηγές πληροφοριών και που συγχρόνως διαμορφώνουν την υπολογιστική κατάσταση του χρήστη.

Σαν *Physical Item* ορίζονται οι συσκευές (*Device*) και οι αισθητήρες (*Sensor*) που βρίσκονται σε έναν χώρο (*Environment*). Ένας χώρος μπορεί να είναι ένα σπίτι, ένα γραφείο, το δωμάτιο ενός ξενοδοχείου ή οποιοσδήποτε άλλος χώρος. Οι συσκευές έχουν μία θέση εντός του περιβάλλοντος που βρίσκονται. Οι συσκευές διαχωρίζονται σε συσκευές ελεγχόμενες από τον χρήστη όπως οι οικιακές συσκευές, οι συσκευές πολυμέσων (*DVD player* κλπ.) και περιφερειακές συσκευές όπως εκτυπωτές, μικρόφωνα, οθόνες κλπ.

Εξειδίκευση των συσκευών αποτελούν και τα τερματικά του χρήστη, κινητά και σταθερά, όπως το laptop, το PDA, το κινητό τηλέφωνο, αλλά και ο υπολογιστής του, ένας εξυπηρετητής (*server*) εντός του σπιτιού, αλλά και άλλες σταθερές συσκευές όπως για παράδειγμα ένα μηχάνημα έκδοσης εισιτηρίων σε κάποιο σταθμό μέσω μαζικής μεταφοράς. Τέλος, εξειδίκευση συσκευής αποτελεί και η Προσωπική Συσκευή Χρήστη (*Personalization Device*). Πάνω σε κάθε συσκευή υπάρχουν ενσωματωμένοι αισθητήρες, οι οποίοι παρέχουν πληροφορίες για το περιβάλλον.

2.3.1.4 Το μοντέλο δικτύου

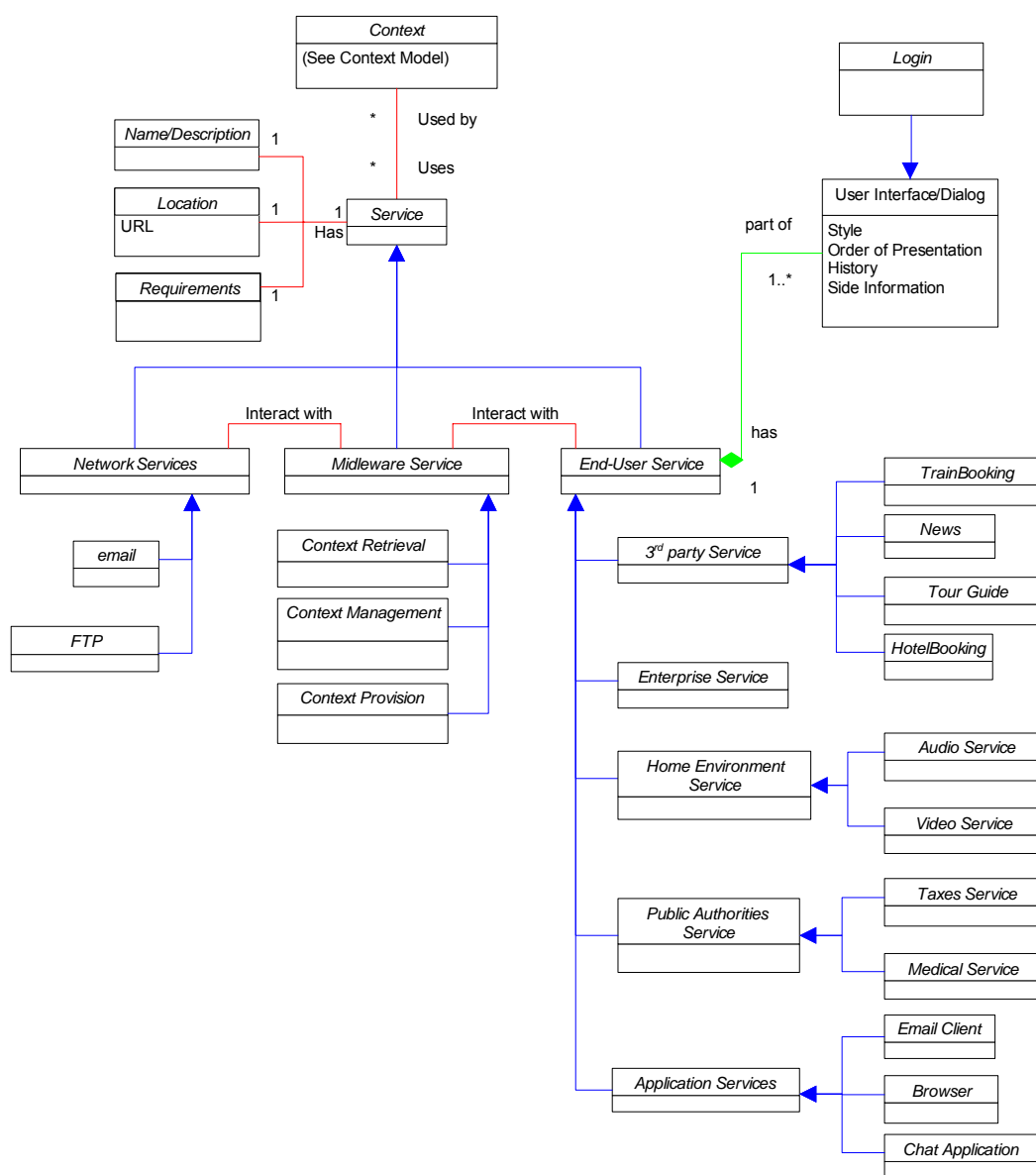


Σχήμα 19: Το μοντέλο δικτύου

Στο Σχήμα 19 απεικονίζονται οι σχετικές με το δίκτυο πληροφορίες. Τα ενσύρματα και ασύρματα δίκτυα αποτελούν εξειδικεύσεις της κλάσης *Network*.

Ένα δίκτυο μπορεί να συνδέεται με άλλα δίκτυα. Τα ενσύρματα δίκτυα διακρίνονται σε DSL και LAN, ενώ τα ασύρματα σε WLAN, GPRS GSM, UMTS και Bluetooth. Κάθε δίκτυο παρέχει διαφορετικό QoS. Η ενεργή σύνδεση εγκαθιδρύεται μέσω ενός δικτύου και κάθε δίκτυο μπορεί να υποστηρίζει πολλές συνδέσεις. Μία δικτυακή σύνδεση εκκινείται μέσω ενός τερματικού το οποίο μπορεί να έχει δυνατότητα σύνδεσης με περισσότερα του ενός δίκτυα.

2.3.1.5 Το μοντέλο υπηρεσίας



Σχήμα 20: Το μοντέλο υπηρεσίας

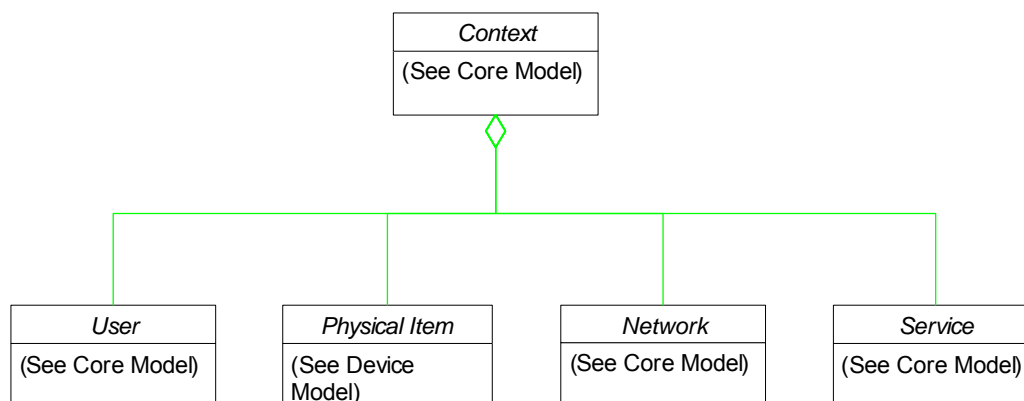
Όσον αφορά στο μοντέλο υπηρεσίας, στο πλαίσιο της παρούσας διατριβής στόχος δεν είναι να δημιουργήσουμε μία οντολογία που να περιγράφει όλες τις διαθέσιμες υπηρεσίες και τα χαρακτηριστικά τους. Στόχος είναι να καταστεί εμφανής ο συσχετισμός ανάμεσα στις υπηρεσίες και στις πληροφορίες πλαισίου χρήσης τους. Μία υπηρεσία με επίγνωση πλαισίου χρήσης για να εκτελεστεί χρησιμοποιεί την πληροφορία context. Συγχρόνως όμως, σχετικές με την υπηρεσία πληροφορίες είναι δυνατόν να αποτελούν μέρος της πληροφορίας context και να παίζουν ρόλο στην διαμόρφωση της τρέχουσας κατάστασης του χρήστη και κατά συνέπεια της υπηρεσίας. Για παράδειγμα, αν μία υπηρεσία έχει συγκεκριμένες, υψηλές απαιτήσεις όσον αφορά τις δυνατότητες ανάλυσης οθόνης, και τα διαθέσιμα τερματικά είναι ένα κινητό τηλέφωνο και ένας σταθερός υπολογιστής, τότε η παρουσίαση της υπηρεσίας θα γίνει μέσω του σταθερού υπολογιστή. Στο επόμενο σχήμα (Σχήμα 20) παρουσιάζεται το μοντέλο πληροφορίας υπηρεσίας από το οποίο διαφαίνεται ο συσχετισμός των υπηρεσιών με το context, οι σχετικές με τις υπηρεσίες πληροφορίες context και συγχρόνως μια ενδεικτική προσπάθεια κατηγοριοποίησης υπηρεσιών.

Οι υπηρεσίες χωρίζονται σε τρεις κατηγορίες:

- Δικτυακές υπηρεσίες (*Network Service*)
- Υπηρεσίες Μεσισμικού (*Middleware Service*)
- Υπηρεσίες Χρήστη (*End-User Service*)

Οι δικτυακές υπηρεσίες παρέχονται από το δίκτυο και δεν υπάρχει απευθείας αλληλεπίδραση με το χρήστη. Τις υπηρεσίες αυτές μπορούν να τις εκμεταλλευτούν οι Υπηρεσίες Μεσισμικού οι οποίες αφορούν στην ανάκτηση, διαχείριση και παροχή πληροφορίας context που παρέχει το σύστημα που παρουσιάζεται στο επόμενο κεφάλαιο. Οι Υπηρεσίες Χρήστη είναι όλες οι υπηρεσίες οι οποίες αλληλεπιδρούν με τον χρήστη μέσω κατάλληλης διεπαφής χρήστη (User Interface) και του παρέχουν περιεχόμενο (content). Οι υπηρεσίες αυτές εκμεταλλεύονται τις υπηρεσίες μεσισμικού που παρέχει το σύστημα προκειμένου να ανακτήσουν το context. Οι υπηρεσίες χρήστη διαχωρίζονται σε κατηγορίες βάσει του Πεδίου Υπηρεσίας (Service Domain) στο οποίο ανήκουν. Υπάρχουν υπηρεσίες που παρέχονται από Ανεξαρτήτους Παρόχους Υπηρεσιών (Independent Service Provider ή 3rd Party Service Provider), από δημόσιες αρχές καθώς και εφαρμογές που εκτελούνται τοπικά όπως π.χ. ένα πελάτης ηλεκτρονικού ταχυδρομείου (email client).

2.3.1.6 Το μοντέλο πληροφορίας πλαισίου χρήσης υπηρεσίας



Σχήμα 21: Το μοντέλο πληροφορίας πλαισίου χρήσης υπηρεσίας

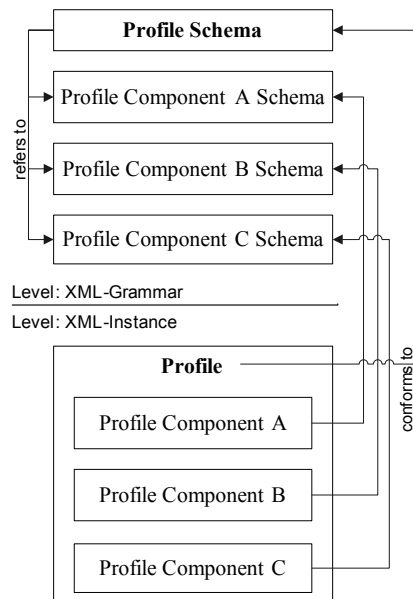
Λαμβάνοντας υπ' όψιν τους διαφορετικούς ορισμούς για την έννοια του context που παρατέθηκαν στο πρώτο κεφάλαιο και έχοντας ολοκληρώσει και μελετήσει το μοντέλο πληροφοριών του συστήματος καταλήγουμε στο συμπέρασμα ότι το σύνολο των πληροφοριών που περιλαμβάνονται στα παραπάνω μοντέλα μπορούν να δώσουν πλήρη εικόνα για την κατάσταση του χρήστη και για το πλαίσιο εντός του οποίου πρόκειται να χρησιμοποιήσει κάποια υπηρεσία. Κατά συνέπεια καταλήγουμε στο γεγονός ότι μπορούμε να μοντελοποιήσουμε το context σαν σύνθεση των μοντέλων του χρήστη, των αντικειμένων στο χώρο, του δικτύου και των υπηρεσιών όπως παρουσιάζεται στο Σχήμα 21.

2.3.2 Το Μοντέλο Δεδομένων

Το στάδιο που ακολούθησε μετά την ολοκλήρωση του μοντέλου πληροφορίας του συστήματος ήταν ο καθορισμός της σύνταξης για την περιγραφή και δόμηση των δεδομένων του context. Στο πλαίσιο της παρούσας διατριβής έμφαση δόθηκε κυρίως σε πληροφορίες προφίλ, και όχι σε πληροφορίες προερχόμενες από αισθητήρες.

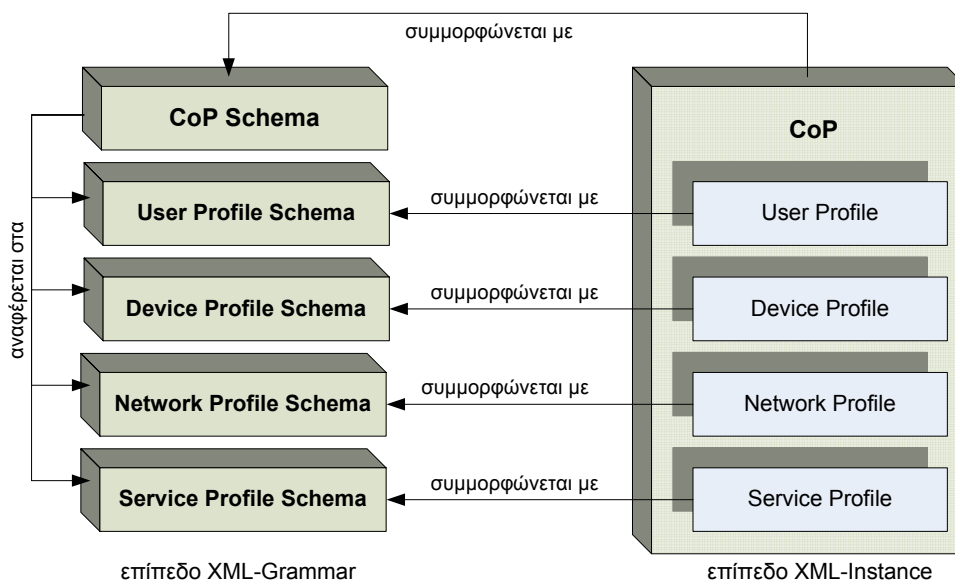
Για το σκοπό αυτό ακολουθήθηκε η αρχιτεκτονική, οι συμβάσεις και η σύνταξη της 3GPP (3rd Generation Partnership Project) για τον καθορισμό του προτύπου Generic User Profile [3]. Το πρότυπο αυτό δεν καθορίζει το περιεχόμενο, την οντολογία του προφίλ, τα στοιχεία που το αποτελούν ή τις σχέσεις μεταξύ αυτών των στοιχείων. Ο στόχος του είναι η δημιουργία ενός συνόλου από κανόνες σύνταξης και δομής για τον ορισμό προφίλ και των στοιχείων του εξασφαλίζοντας έτσι το σημαντικό χαρακτηριστικό της ανοικτότητας και διαλειτουργικότητας για την εκμετάλλευση των δεδομένων του από τις υπηρεσίες. Το σύνολο των κανόνων αυτών ονομάζεται Data Description Method (DDM) [36]. Η DDM ορίζει μέσω σχημάτων XML (XML schema) τον τρόπο καθορισμού του προφίλ και των στοιχείων του. Τα σχήματα XML καθορίζουν για παράδειγμα πώς πρέπει να ορίζονται οι

Χώροι Ονομάτων (namespaces), τα Σχόλια (annotations), και οι Ορισμοί Τύπων Δεδομένων (data type definitions).



Σχήμα 22: Η δομή του 3GPP GUP

Στο Σχήμα 22 παρουσιάζεται η δομή ενός προφίλ. Κάθε προφίλ αποτελείται από στοιχεία (profile components) που μπορούν να περιγράψουν π.χ. γενικές πληροφορίες του χρήστη, πληροφορίες για τις υπηρεσίες που είναι εγγεγραμμένος, πληροφορίες για το δίκτυο, πληροφορίες για το τερματικό κλπ. Κάθε προφίλ και κάθε στοιχείο του, πρέπει να συμμορφώνεται με το αντίστοιχο σχήμα XML.



Σχήμα 23: Η δομή του CoP

Ακολουθώντας την ιεραρχική και βασισμένη σε στοιχεία δομή της 3GPP και το μοντέλο πληροφορίας ορίζουμε το προφίλ πληροφορίας πλαισίου χρήσης υπηρεσίας (Context Profile, CoP). Το CoP αποτελείται από τέσσερα δομικά στοιχεία όπως φαίνεται στο Σχήμα 23. Αυτά είναι το προφίλ χρήστη, συσκευής, δικτύου και υπηρεσίας. Για κάθε ένα από αυτά υλοποιήθηκε το αντίστοιχο XML σχήμα. Το XML αρχείο του CoP αποτελεί κάθε φορά στιγμιότυπο των αντίστοιχων XML σχημάτων και αναπαριστά την τρέχουσα κατάσταση του χρήστη και του περιβάλλοντός του ενσωματώνοντας όλη την σχετική πληροφορία. Στις επόμενες παραγράφους παρουσιάζονται τα επιμέρους σχήματα και το στιγμιότυπο ενός CoP.

2.3.2.1 Προφίλ Χρήστη

Για τον καθορισμό του περιεχομένου του Προφίλ χρήστη ακολουθήθηκε το πρότυπο *Client Side Automated Form Entry* του W3C [37]. Το πρότυπο αυτό καθορίζει τις παρακάτω δομές δεδομένων (data elements): *IdentityInformation*, *ContactInformation*, *PostalInformation*, *PostalInformation (Billing)*, *OrganizationalInformation*. Κάθε μία από τις παραπάνω δομές δεδομένων περιλαμβάνει αντίστοιχα πληροφορίες σχετικά με την ταυτότητα του χρήστη, τα στοιχεία - τηλέφωνα, διευθύνσεις ηλεκτρονικού ταχυδρομείου κλπ. - μέσω των οποίων μπορεί κάποιος να έρθει σε επαφή μαζί του, στοιχεία διευθύνσεων και στοιχεία για τον εργασιακό του χώρο. Προκειμένου το σχήμα XML του προφίλ χρήστη να ανταποκρίνεται στο αντίστοιχο μοντέλο πληροφορίας (ενότητα 2.3.1.3), διατηρήθηκαν οι παραπάνω δομές και προστέθηκαν και οι ακόλουθες: *Schedule*, *Medical Information*, *User General Preferences*, και *User Location*.

Κάθε μία από τις δομές του προφίλ χρήστη περιλαμβάνει τα εξής στοιχεία:

UserIdentity: *FullName*, *FirstName*, *MiddleName*, *LastName*, *CountryOf Residence*, *Nationality*, *BirthDate*, *Gender*, *IDNumber*, *PassportNumber*, *TaxNumber*

ContactInformation: *Telephone-work*, *Telephone-fax*, *Telephone-mobile*, *Email*, *Homepage*, *ChatID*

PostalInformation: *HomeAddress*, *WorkAddress*, τα οποία αποτελούν στοιχεία τύπου *Address*, που περιλαμβάνει τα *Country*, *Shire*, *City*, *Street*, *StreetNumber* και *PostalCode*.

BillingInformation: *CreditCard* που περιλαμβάνει τα στοιχεία *Type*, *Number*, *ExpiryMonth*, *ExpiryYear*, *BankAccount* που περιλαμβάνει τα στοιχεία *BankName*, *BankAccountNumber* και *BillingAddress* που είναι τύπου *Address*.

Organization: *Affiliation*, *Position*

Schedule: *WorkingDays*, *WorkingHours*, *Meetings*. Ο τύπος *Meetings* περιγράφεται από τα στοιχεία *Date*, *Time*, *Duration*, *Location*, *Description*.

MedicalInformation: *Allergies, Maladies, Medication*

UserGeneralPreferences: *Cinema, Reading, Sports, Food, Travelling, Hobbies*

Location: *PhysicalLocation* που είναι τύπου *Address*, *GeodeticLocation* που αποτελείται από τις συντεταγμένες *Longitude*, *Latitude*, *Altitude*, *LogicalLocation*, που μπορεί να είναι μία διεύθυνση IP, και *Orientation*.

Στη συνέχεια παρατίθεται το σχήμα XML του Προφίλ Χρήστη.

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.simplicity.org/UserProfile"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:UP="http://www.simplicity.org/UserProfile"
  xmlns:NP="http://www.simplicity.org/NetworkProfile"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <!-- Definition of Types -->
  <xs:import namespace="http://www.simplicity.org/NetworkProfile" schemaLocation="NetworkProfile.xsd"/>
  <xs:simpleType name="GenderType">
    <xs:restriction base="xs:NMTOKEN">
      <xs:enumeration value="MALE"/>
      <xs:enumeration value="FEMALE"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="TelephoneNumberType">
    <xs:restriction base="xs:string">
      <xs:pattern value="[0-9]+"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="AddressType">
    <xs:element name="Country" type="xs:string"/>
    <xs:element name="Shire" type="xs:string"/>
    <xs:element name="City" type="xs:string"/>
    <xs:element name="Street" type="xs:string"/>
    <xs:element name="StreetNumber" type="xs:positiveInteger"/>
    <xs:element name="PostalCode" type="xs:positiveInteger"/>
  </xs:complexType>
  <xs:simpleType name="StringOfNumbersType">
    <xs:restriction base="xs:string">
      <xs:pattern value="[0-9]+"/>
    </xs:restriction>
  </xs:simpleType>
  <!-- Definition of Elements -->
  <xs:element name="UserIdentity">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="FullName" type="xs:string"/>
        <xs:element name="FirstName" type="xs:string"/>
        <xs:element name="MiddleName" type="xs:string"/>
        <xs:element name="LastName" type="xs:string"/>
        <xs:element name="CountryOfResidence" type="xs:string"/>
        <xs:element name="Nationality" type="xs:string"/>
        <xs:element name="BirthDate" type="xs:date"/>
        <xs:element name="Gender" type="UP:GenderType"/>
        <xs:element name="IDNumber" type="xs:string"/>
        <xs:element name="PassportNumber" type="xs:string"/>
        <xs:element name="TaxNumber" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="ContantInformation">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Telephone-home" type="UP:TelephoneNumberType"/>
        <xs:element name="Telephone-work" type="UP:TelephoneNumberType"/>
        <xs:element name="Telephone-fax" type="UP:TelephoneNumberType"/>
        <xs:element name="Telephone-mobile" type="UP:TelephoneNumberType"/>
        <xs:element name="Email" type="xs:string"/>
        <xs:element name="Homepage" type="xs:string"/>
        <xs:element name="ChatID" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```

```

</xs:complexType>
</xs:element>

<xs:element name="PostalInformation">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="HomeAddress" type="UP:AddressType"/>
      <xs:element name="WorkAddress" type="UP:AddressType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="BillingInformation">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="CreditCard" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Type" type="xs:string"/>
            <xs:element name="Number" type="UP:StringOfNumbersType"/>
            <xs:element name="ExpiryMonth" type="xs:gMonth"/>
            <xs:element name="ExpiryYear" type="xs:gYear"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="BankAccount" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="BankName" type="xs:string"/>
            <xs:element name="BankAccountNumber"
type="UP:StringOfNumbersType"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="BillingAddress" type="UP:AddressType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="Organization">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Affiliation" type="xs:string"/>
      <xs:element name="Position" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="Schedule">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="WorkingDays" type="xs:string"/>
      <xs:element name="WorkingHours" type="xs:string"/>
      <xs:element name="Meetings" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="DateTime" type="xs:dateTime"/>
            <xs:element name="Duration" type="xs:positiveInteger"/>
            <xs:element name="Location" type="xs:string"/>
            <xs:element name="Description" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="MedicalInformation">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Allergies" type="xs:string"/>
      <xs:element name="Malladies" type="xs:string"/>
      <xs:element name="Medication" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="UserGeneralPreferences">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Music" type="xs:anyType"/>
      <xs:element name="Cinema" type="xs:anyType"/>
      <xs:element name="Reading" type="xs:anyType"/>
      <xs:element name="Sports" type="xs:anyType"/>
      <xs:element name="Food" type="xs:anyType"/>
      <xs:element name="Travelling" type="xs:anyType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```



```

        <xs:element name="Hobbys" type="xs:anyType"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Location">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="PhysicalLocation" type="UP:AddressType"/>
        <xs:element name="GeodeticLocation">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Longitude" type="xs:float"/>
              <xs:element name="Latitude" type="xs:float"/>
              <xs:element name="Altitude" type="xs:float"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="LogicalLocation">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:pattern value="[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="Orientation" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
<!-- Definition of the user profile -->
  <xs:element name="UserProfile">
    <xs:sequence>
      <xs:element ref="UP:UserIdentity"/>
      <xs:element ref="UP:ContactInformation"/>
      <xs:element ref="UP:PostalInformation"/>
      <xs:element ref="UP:BillingInformation"/>
      <xs:element ref="UP:OrganizationInformation"/>
      <xs:element ref="UP:Schedule"/>
      <xs:element ref="UP:MedicalInformation"/>
      <xs:element ref="UP:UserGeneralPreferences"/>
      <xs:element ref="UP:UserLocation"/>
      <xs:element name="NetworkPreferences" type="NP:ConnectionParametersType"/>
    </xs:sequence>
  </xs:element>
</xs:schema>

```

2.3.2.2 Προφίλ Τερματικού

Για τον καθορισμό του περιεχομένου του Προφίλ χρήστη ακολουθήθηκε το πρότυπο *UAProf* του WAPFORUM [10]. Το πρότυπο αυτό καθορίζει τις παρακάτω δομές δεδομένων (data elements): *HardwarePlatform*, *SoftwarePlatform*, *BrowserUA*, *NetworkCharacteristics*. Στο πλαίσιο του καθορισμού του Προφίλ Τερματικού διατηρήθηκαν οι δομές *HardwarePlatform*, *SoftwarePlatform* και *NetworkCharacteristics*. Οι δομές αυτές, βάσει του UAProf περιλαμβάνουν τα παρακάτω στοιχεία:

HardwarePlatform: *Vendor*, *CPU*, *Memory*, *StorageSpace*, *ColorCapable*, *ImageCapable*, *SoundCapable*, *Keyboard*, *ScreenSize*, *TextInputCapable*, *VoiceInputCapable*

SoftwarePlatform: *OSName*, *OSVendor*, *OSVersion*, *JVMVersion*

NetworkCharacteristics: *supportedBearers* που μπορεί να παίρνει τιμές όπως *GPRS*, *WLAN*, *Bluetooth* κλπ.

Στη συνέχεια παρατίθεται το σχήμα XML του Προφίλ Τερματικού.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<xs:schema targetNamespace="http://www.simplicity.org/TerminalProfile"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:TP="http://www.simplicity.org/TerminalProfile"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">

  <!-- Definition of Types -->

  <!-- Definition of Elements -->

    <xs:element name="HardwarePlatform">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Model" type="xs:string"/>
          <xs:element name="Vendor" type="xs:string"/>
          <xs:element name="CPU" type="xs:string"/>
          <xs:element name="Memory" type="xs:string"/>
          <xs:element name="StorageSpace" type="xs:boolean"/>
          <xs:element name="ColorCapable" type="xs:boolean"/>
          <xs:element name="ImageCapable" type="xs:boolean"/>
          <xs:element name="SoundCapable" type="xs:boolean"/>
          <xs:element name="Keyboard" type="xs:string"/>
          <xs:element name="ScreenSize" type="xs:string"/>
          <xs:element name="TextInputCapable" type="xs:boolean"/>
          <xs:element name="VoiceInputCapable" type="xs:boolean"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>

    <xs:element name="SoftwarePlatform">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="OSName" type="xs:string"/>
          <xs:element name="OSVendor" type="xs:string"/>
          <xs:element name="OSVersion" type="xs:string"/>
          <xs:element name="JVMVersion" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>

    <xs:element name="NetworkCharacteristics">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="supportedBearers">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:enumeration value="GPRS"/>
                <xs:enumeration value="WLAN"/>
                <xs:enumeration value="Bluetooth"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>

  <!-- Definition of the terminal profile -->

    <xs:element name="TerminalProfile">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="HardwarePlatform" ref="TP:HardwarePlatform"/>
          <xs:element name="SoftwarePlatform" ref="TP:SoftwarePlatform"/>
          <xs:element name="NetworkCharacteristics" ref="TP:NetworkCharacteristics"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>

</xs:schema>

```

2.3.2.3 Προφίλ Υπηρεσίας

Η δημιουργία του Προφίλ Υπηρεσίας δεν βασίστηκε σε κάποιο συγκεκριμένο πρότυπο, αλλά στις ανάγκες που δημιουργήθηκαν κατά το σχεδιασμό και την υλοποίηση της προτεινόμενης αρχιτεκτονικής. Το Προφίλ Υπηρεσίας αποτελείται από μία δομή, την *ServiceProfile*, η οποία περιέχει ένα στοιχείο, το *Service*, το οποίο είναι τύπου *ServiceType*.

Το *ServiceType* αποτελείται από τα στοιχεία *Name*, *Description*, *Location* (η τοποθεσία της υπηρεσίας στο δίκτυο, π.χ. κάποιο URL), *Account* (το username και password του χρήστη), *Cost* (το οποίο μπορεί να είναι μηνιαίο πάγιο, ετήσιο ή κόστος ανά σύνοδο κλπ.), *TerminalRequirements*, το οποίο περιλαμβάνει τις απαιτήσεις της υπηρεσίας ως προς τις δυνατότητες του τερματικού, *NetworkRequirements*, το οποίο περιλαμβάνει τις απαιτήσεις της υπηρεσίας ως προς το δίκτυο, και τέλος *UserPreferences*, που αφορά τις ειδικές προτιμήσεις του χρήστη ως προς την συγκεκριμένη υπηρεσία.

Στη συνέχεια παρατίθεται το σχήμα XML του Προφίλ Υπηρεσίας.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.simplicity.org/ServiceProfile"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:SP="http://www.simplicity.org/ServiceProfile"
  xmlns:NP="http://www.simplicity.org/NetworkProfile"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <xs:import namespace="http://www.simplicity.org/NetworkProfile" schemaLocation="NetworkProfile.xsd"/>
<!-- Definition of Types -->
  <xs:complexType name="PreferenceType">
    <xs:sequence>
      <xs:element name="PreferenceName" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="PreferenceValue" type="xs:string" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="RequirementType">
    <xs:sequence>
      <xs:element name="RequirementName" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="RequirementValue" type="xs:string" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="AccountType">
    <xs:sequence>
      <xs:element name="username" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="password" type="xs:string" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="CostType">
    <xs:choice>
      <xs:element name="MonthlySubscription" type="xsd:string"/>
      <xs:element name="AnnualSubscription" type="xsd:string"/>
      <xs:element name="CostPerSession" type="xsd:string"/>
    </xs:choice>
  </xs:complexType>
  <xs:complexType name="ServiceType">
    <xs:sequence>
      <xs:element name="Name" type="xs:string" maxOccurs="1"/>
      <xs:element name="Description" type="xs:string" maxOccurs="1"/>
      <xs:element name="Location" type="xs:anyURI" maxOccurs="1"/>
      <xs:element name="Account" type="SP:AccountType" maxOccurs="1"/>
      <xs:element name="Cost" type="SP:CostType" maxOccurs="1"/>
      <xs:element name="TerminalRequirements" maxOccurs="1">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Requirement" minOccurs="0" maxOccurs="unbounded"
              type="SP:RequirementType"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="NetworkRequirements" maxOccurs="1">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Requirement" minOccurs="0" maxOccurs="unbounded"
              type="SP:RequirementType"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="UserPreferences" maxOccurs="1">
        <xs:complexType>

```

```

        <xs:sequence>
          <xs:element name="Preference" minOccurs="0" maxOccurs="unbounded"
type="SP:PreferenceType"/>
        </xs:sequence>
      </xs:complexType>
    </xs:sequence>
  </xs:complexType>
<!-- Definition of Elements -->
<!-- Definition of the service profile -->
  <xs:element name="ServiceProfile">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Service" minOccurs="0" maxOccurs="unbounded" type="SP:ServiceType"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

2.3.2.4 Προφίλ Δικτύου

Η δημιουργία του Προφίλ Δικτύου δεν βασίστηκε σε κάποιο συγκεκριμένο πρότυπο, αλλά στις ανάγκες που δημιουργήθηκαν κατά το σχεδιασμό και την υλοποίηση της προτεινόμενης αρχιτεκτονικής. Το Προφίλ Δικτύου αποτελείται από μία δομή, την *NetworkProfile*, η οποία περιέχει δυο στοιχεία, το *Network*, το οποίο είναι τύπου *NetworkType* και το *ActiveConnection*, το οποίο είναι τύπου *NetworkNameType*. Το *NetworkType* αποτελείται από τα στοιχεία *Name*, τύπου *NetworkNameType* (περιλαμβάνει το στοιχείο *isActive*), *Description*, *ConnectionParameters*, οι οποίες είναι τύπου *ConnectionParametersType* (περιλαμβάνει τα στοιχεία *Bandwidth*, *Cost*, *Latency*, *Jitter*), και *NetworkServices*, οι οποίες είναι οι *DNS*, *Gateway*, *DHCP* κλπ., που αποτελούν τύπους *NetworkServiceType* (περιλαμβάνει τα στοιχεία *Description*, *IP*, *Port*).

Στη συνέχεια παρατίθεται το σχήμα XML του Προφίλ Δικτύου.

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.simplicity.org/NetworkProfile"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:NP="http://www.simplicity.org/NetworkProfile"
elementFormDefault="qualified"
attributeFormDefault="unqualified">
<!-- Definition of Types -->
  <xs:complexType name="NetworkServiceType">
    <xs:sequence>
      <xs:element name="Description" type="xs:string"/>
      <xs:element name="IP">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:pattern value="[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Port">
        <xs:simpleType>
          <xs:restriction base="xs:positiveInteger">
            <xs:maxInclusive value="65536"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

```

```

</xs:complexType>

<xs:complexType name="CostType">
  <xs:choice>
    <xs:element name="CostPerKB" type="xs:float"/>
    <xs:element name="MonthlySubscription" type="xs:float"/>
  </xs:choice>
</xs:complexType>

<xs:simpleType name="NetworkNameType">
  <xs:restriction base="xs:string">
    <xs:attribute name="isActive" type="xs:boolean"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="ConnectionParametersType">
  <xs:sequence>
    <xs:element name="Bandwidth" type="xs:string"/>
    <xs:element name="Cost" type="NP:CostType"/>
    <xs:element name="Latency" type="xs:string"/>
    <xs:element name="jitter" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="NetworkType">
  <xs:sequence>
    <xs:element name="Name" type="NP:NetworkNameType"/>
    <xs:element name="Description" type="xs:string"/>
    <xs:element name="ConnectionParameters" type="NP:ConnectionParametersType"/>
    <xs:element name="NetworkServices">
      <xs:element name="DNS" type="NP:NetworkServiceType"/>
      <xs:element name="Gateway" type="NP:NetworkServiceType"/>
      <xs:element name="DHCP" type="NP:NetworkServiceType"/>
    </xs:element>
  </xs:sequence>
</xs:complexType>

<!-- Definition of Elements -->

<!-- Definition of the service profile -->

  <xs:element name="NetworkProfile">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Network" type="NP:NetworkType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="ActiveConnection" type="NP:NetworkNameType" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>

```

2.3.2.5 Το σχήμα Πληροφορίας Πλαισίου Χρήσης Υπηρεσίας

Στη συνέχεια παρατίθεται το Προφίλ Πλαισίου Χρήσης Υπηρεσίας, το οποίο αποτελείται από τέσσερα στοιχεία τα οποία είναι αντίστοιχα τα σχήματα XML των προφίλ χρήστη, τερματικού, υπηρεσίας και δικτύου.

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://www.simplicity.org/ContextProfile"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:SP="http://www.simplicity.org/ServiceProfile"
  xmlns:NP="http://www.simplicity.org/NetworkProfile"
  xmlns:UP="http://www.simplicity.org/UserProfile"
  xmlns:TP="http://www.simplicity.org/TerminalProfile"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">

  <xs:import namespace="http://www.simplicity.org/NetworkProfile" schemaLocation="NetworkProfile.xsd"/>
  <xs:import namespace="http://www.simplicity.org/UserProfile" schemaLocation="UserProfile.xsd"/>
  <xs:import namespace="http://www.simplicity.org/ServiceProfile" schemaLocation="ServiceProfile.xsd"/>
  <xs:import namespace="http://www.simplicity.org/TerminalProfile" schemaLocation="TerminalProfile.xsd"/>

  <!-- Definition of the context profile -->

  <xs:element name="ContextProfile">
    <xs:complexType>

```

```

</xs:sequence>
  <xs:element ref="NP:NetworkProfile"/>
  <xs:element ref="UP:UserProfile"/>
  <xs:element ref="SP:ServiceProfile"/>
  <xs:element ref="TP:TerminalProfile"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

2.3.2.6 Αναπαράσταση ενός στιγμιότυπου Πληροφορίας Πλαισίου Χρήσης Υπηρεσίας

Το αρχείο XML που ακολουθεί αποτελεί στιγμιότυπου του σχήματος XML του Προφίλ Πλαισίου Χρήσης Υπηρεσίας.

```

<?xml version="1.0" encoding="UTF-8"?>
<ContextProfile>
  <NetworkProfile>
    <Network>
      <NetworkName isActive="true">
        GPRS
      </NetworkName>
      <Description>
        Mobile Phone GPRS connection
      </Description>
      <ConnectionParameters>
        <Bandwidth>150Kbps</Bandwidth>
        <Cost>
          <MonthlySubscription>35</MonthlySubscription>
        </Cost>
        <Latency>650ms</Latency>
        <Jitter>450ms</Jitter>
      </ConnectionParameters>
      <NetworkServices>
        <DNS>
          <Description>The DNS server</Description>
          <IP>192.168.5.1</IP>
          <Port>50</Port>
        </DNS>
        <Gateway>
          <Description>Default Gateway</Description>
          <IP>192.168.1.200</IP>
          <Port/>
        </Gateway>
        <DHCP>
          <Description>DHCP server</Description>
          <IP>192.168.5.10</IP>
          <Port>67</Port>
        </DHCP>
      </NetworkServices>
    </Network>
    <ActiveConnection isActive="true">GPRS</ActiveConnection>
  </NetworkProfile>
  <UserProfile>
    <UserIdentity>
      <FullName>James Bond</FullName>
      <FirstName>James</FirstName>
      <MiddleName>??</MiddleName>
      <LastName>Bond</LastName>
      <CountryOfResidence>GB</CountryOfResidence>
      <Nationality>English</Nationality>
      <Birthdate>16-11-1924</Birthdate>
      <Gender>Male</Gender>
      <IDNumber>007</IDNumber>
      <PassportNumber>GB007</PassportNumber>
      <TaxNumber>123456789007</TaxNumber>
    </UserIdentity>
    <ContactInformation>
      <Telephone-home>+44 12 34 567 007</Telephone-home>
      <Telephone-work>+44 76 54 321 007</Telephone-work>
      <Telephone-fax/>
      <Telephone-mobile>+44 55 66 777 007</Telephone-mobile>
      <Email>james.bond@gmail.com</Email>
      <Homepage>http://en.wikipedia.org/wiki/James_Bond</Homepage>
      <ChatID>james.bond@gmail.com</ChatID>
    </ContactInformation>
    <PostalInformation>

```

```

<HomeAddress>
</HomeAddress>
<WorkAddress>
  <Country>England</Country>
  <Shire>London</Country>
  <City>London</City>
  <Street>Vauxhall Cross</Street>
  <StreetNumber>85</StreetNumber>
  <PostalCode>SE1 1BD</PostalCode>
</WorkAddress>
</PostalInformation>
<BillingInformation>
  <CreditCard>
    <Type>VISA</Type>
    <Number>1234 5678 1234 5678</Number>
    <ExpiryMonth>07</ExpiryMonth>
    <ExpiryYear>9999</ExpiryYear>
  </CreditCard>
  <BankAccount>
    <BankName>HSBC</BankName>
    <BankAccountNumber>GB12 ABCD 1234 5612 3456 78</BankAccountNumber>
  </BankAccount>
  <BillingAddress>
    <Country>England</Country>
    <Shire>London</Country>
    <City>London</City>
    <Street>Vauxhall Cross</Street>
    <StreetNumber>85</StreetNumber>
    <PostalCode>SE1 1BD</PostalCode>
  </BillingAddress>
</BillingInformation>
<OrganizationInformation>
  <Affiliation>British Secret Intelligence Service</Affiliation>
  <Position>007 Agent, Licence to kill</Position>
</OrganizationInformation>
<Schedule>
  <WorkingDays>Mon Tue Wed Thu Fri Sat Sun</WorkingDays>
  <WorkingHours>00.00 - 24.00</WorkingHours>
  <Meetings/>
</Schedule>
<MedicalInformation>
  <Allergies/>
  <Malladies/>
  <Medication/>
</MedicalInformation>
<UserGeneralPreferences>
  <Music>Jazz</Music>
  <Cinema>007 films</Cinema>
  <Reading>007 novels</Reading>
  <Sports>Golf</Sports>
  <Food/>
  <Travelling>Middle East, Indonesia, Switzerland</Travelling>
  <Hobbys>Casino</Hobbys>
</UserGeneralPreferences>
<UserLocation>
  <PhysicalLocation>
    <Country>England</Country>
    <Shire>London</Country>
    <City>London</City>
    <Street>Vauxhall Cross</Street>
    <StreetNumber>85</StreetNumber>
    <PostalCode>SE1 1BD</PostalCode>
  </PhysicalLocation>
  <GeodeticLocation>
    <Longitude>-0.124071640905</Longitude>
    <Latitude>51.4872164088</Latitude>
    <Altitude>100</Altitude>
  </GeodeticLocation>
  <LogicalLocation>
    157.158.159.007
  </LogicalLocation>
  <Orientation>
    East
  </Orientation>
</UserLocation>
<NetworkPreferences>
  <Bandwidth>1000Mbit/sec</Bandwidth>
  <Cost>
    <CostPerKB>0,001</CostPerKB>
  </Cost>
  <Latency>200ms</Latency>
  <jitter>20ms</Jitter>
</NetworkPreferences>
</UserProfile>
<ServiceProfile>
  <Service>

```

```

<Name>gMail</Name>
<Description>Access to my gmail</Description>
<Location>http://gmail.com</Location>
<Account>
  <username>james.bond@gmail.com</username>
  <password>007password</password>
</Account>
<Cost/>
<TerminalRequirements>
  <Requirement>
    <RequirementName>MinScreenResolution</RequirementName>
    <RequirementValue>800x600</RequirementValue>
  </Requirement>
  <Requirement>
    <RequirementName>BrowserPresent</RequirementName>
    <RequirementValue>YES</RequirementValue>
  </Requirement>
  <Requirement>
    <RequirementName>javascript enabled</RequirementName>
    <RequirementValue>YES</RequirementValue>
  </Requirement>
</TerminalRequirements>
<NetworkRequirements/>
<UserPreferences>
  <Preference>
    <PreferenceName>Automatic Login</PreferenceName>
    <PreferenceValue>YES</PreferenceValue>
  </Preference>
  <Preference>
    <PreferenceName>Display Name</PreferenceName>
    <PreferenceValue>James Bond</PreferenceValue>
  </Preference>
</UserPreferences>
<ActiveConnection isActive="true">GPRS</ActiveConnection>
</ServiceProfile>
<TerminalProfile>
  <HardwarePlatform>
    <Model>Dimension E520</Model>
    <Vendor>Dell</Vendor>
    <CPU>1.8GHz</CPU>
    <Memory>1024MB</Memory>
    <StorageSpace>160GB</StorageSpace>
    <ColorCapable>true</ColorCapable>
    <ImageCapable>true</ImageCapable>
    <SoundCapable>true</Soundcapable>
    <Keyboard>yes</keyboard>
    <ScreenSize>1280x1024</ScreenSize>
    <TextInputCapable>true</TextInputCapable>
    <VoiceInputCapable>true</VoiceInputCapable>
  </HardwarePlatform>
  <SoftwarePlatform>
    <OSName>Vista</OSName>
    <OSVendor>Microsoft</Microsoft>
    <OSVersion>6.0</OSVersion>
    <JVMVersion>1.5</JVMVersion>
  </SoftwarePlatform>
  <NetworkCharacteristics>
    <supportedBearers>GPRS</supportedBearers>
  </NetworkCharacteristics>
</TerminalProfile>
</ContextProfile>

```

2.4 Αναφορές

- [1] Distributed Management Task Force. Common Information Model. <http://www.dmtf.org/standards/cim/>
- [2] Object Management Group, OMG. <http://www.omg.org/technology/documents/formal/uml.htm>
- [3] 3rd Generation Partnership Project. 3GPP Generic User Profile (GUP) - Architecture. Technical specification Group Services and Systems Aspects, TS 23.240, Version 6.7.0. 2005.
- [4] Held, A., Buchholz, S., and Schill, A. Modeling of context information for pervasive computing applications. In Proceedings of SCI 2002/ISAS 2002 (2002).
- [5] Chen, I.Y.L., Yang, S.J.H., Jia Zhang, Ubiquitous Provision of Context Aware Web Services

- Services Computing, 2006. SCC '06. IEEE International Conference on Sept. 2006. Page(s):60 - 68
- [6] T. Strang and C. Linnhoff-Popien. A context modeling survey. In Proceedings of the Workshop on Advanced Context Modelling, Reasoning and Management, 2004.
- [7] Shilit, B. N., Adams, N. L., and Want, R. Context-aware computing applications. In IEEE Workshop on Mobile Computing Systems and Applications (Santa Cruz, CA, US, 1994).
- [8] Samulowitz, M., Michahelles, F., and Linnhoff-Popien, C. Capeus: An architecture for context-aware selection and execution of services. In New developments in distributed applications and interoperable systems (Krakow, Poland, September 17-19 2001), Kluwer Academic Publishers, pp. 23–39.
- [9] Composite Capabilities/Preference Profile Public Home Page. <http://www.w3.org/Mobile/CCPP>
- [10] WAP Forum. Wireless Application Group, User Agent Profile Specification, WAP Forum Approved Specification WAP-174, 10 November 1999. See: <http://www1.wapforum.org>.
- [11] Indulska, J., Robinson, R., Rakotonirainy, A., and Henriksen, K. Experiences in using cc/pp in context-aware systems. In LNCS 2574: Proceedings of the 4th International Conference on Mobile Data Management (MDM2003) (Melbourne/Australia, January 2003), M.-S. Chen, P. K. Chrysanthis, M. Sloman, and A. Zaslavsky, Eds., Lecture Notes in Computer Science (LNCS), Springer, pp. 247–261.
- [12] Chtcherbina, E., and Franz, M. Peer-to-peer coordination framework (p2pc): Enabler of mobile ad-hoc networking for medicine, business, and entertainment. In Proceedings of International Conference on Advances in Infrastructure for Electronic Business, Education, Science, Medicine, and Mobile Technologies on the Internet (SSGRR2003w) (L'Aquila/Italy, January 2003).
- [13] Ryan, N. ConteXtML: Exchanging Contextual Information between a Mobile Client and the FieldNote Server, August 1999
- [14] Chen, P.-S. The entity-relationship model - toward a unified view of data. ACM Transaction on Database Systems 1, 1 (March 1976), 9–36.
- [15] Bauer, J. Identification and Modeling of Contexts for Different Information Scenarios in Air Traffic, Mar. 2003. Diplomarbeit. Technische Universität Berlin.
- [16] Halpin, T. A. Information Modeling and Relational Databases: From Conceptual Analysis to Logical Design. Morgan Kaufman Publishers, San Francisco, 2001.
- [17] Henriksen, K., Indulska, J., and Rakotonirainy A. Generating Context Management Infrastructure from High-Level Context Models. In Industrial Track Proceedings of the 4th International Conference on Mobile Data Management (MDM2003) (Melbourne/Australia, January 2003), pp. 1–6.
- [18] Quan Z. Sheng, Boualem Benatallah. ContextUML: A UML-Based Modeling Language for Model-Driven Development of Context-Aware Web Services. The 4th International Conference on Mobile Business (ICMB'05), IEEE Computer Society. July 11-13 2005, Sydney, Australia.
- [19] Schmidt, A., Beigl, M., and Gellersen, H.-W. There is more to context than location.

Computers and Graphics 23, 6 (1999), 893–901.

- [20] Schmidt, A., and Laerhoven, K. V. How to Build Smart Appliances. IEEE Personal Communications (August 2001).
- [21] Esprit project 26900: Technology for enabled awareness (TEA), 1998.
- [22] Cheverst, K., Mitchell, K., and Davies, N. Design of an object model for a context sensitive tourist GUIDE. Computers and Graphics 23, 6 (1999), 883–891.
- [23] Giunchiglia, F. Contextual reasoning. Epistemologica - Special Issue on I Linguaggi e le Macchine 16 (1993), 345–364. Also IRST-Technical Report 9211-20, IRST, Trento, Italy.
- [24] Akman, V., and Surav, M. The use of situation theory in context modeling. Computational Intelligence 13, 3 (1997), 427–438.
- [25] Gray, P., and Salber, D. Modelling and Using Sensed Context Information in the design of Interactive Applications. In LNCS 2254: Proceedings of 8th IFIP International Conference on Engineering for Human-Computer Interaction (EHCI 2001) (Toronto/Canada, May 2001), M. R. Little and L. Nigay, Eds., Lecture Notes in Computer Science (LNCS), Springer, p. 317 ff.
- [26] Uschold, M., and Gruninger, M. Ontologies: Principles, methods, and applications. Knowledge Engineering Review 11, 2 (1996), 93–155.
- [27] Gruber, T. G. A translation approach to portable ontologies. Knowledge Acquisition 5, 2 (1993), 199–220.
- [28] De Bruijn, J. Using Ontologies – Enabling Knowledge Sharing and Reuse on the Semantic Web. Tech. Rep. Technical Report DERI-2003-10-29, Digital Enterprise Research Institute (DERI), Austria, October 2003.
- [29] Strang, T., Linnhoff-Popien, C., and Frank, K. CoOL: A Context Ontology Language to enable Contextual Interoperability. In LNCS 2893: Proceedings of 4th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems (DAIS2003) (Paris/France, November 2003), J.-B. Stefani, I. Dameure, and D. Hagimont, Eds., vol. 2893 of Lecture Notes in Computer Science (LNCS), Springer Verlag, pp. 236–247.
- [30] Wang, X. H., Zhang, D. Q., Gu, T., and Pung, H. K. Ontology Based Context Modeling and Reasoning using OWL. In Workshop Proceedings of the 2nd IEEE Conference on Pervasive Computing and Communications (PerCom2004) (Orlando, FL, USA, March 2004), pp. 18–22.
- [31] Chen, H., Finin, T., and Joshi, A. Using OWL in a Pervasive Computing Broker. In Proceedings of Workshop on Ontologies in Open Agent Systems (AAMAS 2003) (2003).
- [32] Chen, H., Perich, F., Finin, T., and Joshi, A., SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications. In Proceedings of International Conference on Mobile and Ubiquitous Systems: Networking and Services, August 2004, pp. 258- 267.
- [33] DMTF Standards and Terminology, Distributed Management Task Force Technical Note, June 2003 http://www.dmtf.org/education/technote_Standards.pdf
- [34] Object Management Group, OMG. <http://www.omg.org/cgi-bin/doc?formal/02-06-39>

- [35] E. Rukzio, G. N. Prezerakos, G. Cortese, E. Koutsoloukas, S. Kapellaki, Context for Simplicity: A Basis for Context-aware Systems Based on the 3GPP Generic User Profile, International Journal of Computational Intelligence IJCI, Volume 1, Number 2, 2004, ISSN:1304-2386, pp. 108-111
- [36] 3rd Generation Partnership Project. 3GPP Generic User Profile (GUP) – Data Description Method (DDM). Technical specification Group Terminals, TS 23.941, Version 6.0.0. 2004.
- [37] Hallam-Baker P. Client Side Automated Form Entry. W3C Working Draft, WD-form-filling-960416. <http://www.w3.org/TR/WD-form-filling.html>

3ο Κεφάλαιο

Ανοικτή Αρχιτεκτονική Παροχής Υπηρεσιών Βασισμένων στο Πλαίσιο Χρήσης τους

Στο κεφάλαιο αυτό, περιγράφεται η προτυποποιημένη αρχιτεκτονική που υλοποιήθηκε για την υποστήριξη παροχής υπηρεσιών προσαρμοσμένων στον εκάστοτε χρήστη και την τρέχουσα κατάσταση του. Στο πρώτο μέρος του κεφαλαίου παρουσιάζονται οι βασικές αρχές που ακολουθήθηκαν για τον σχεδιασμό του συστήματος καθώς και τα βασικά δομικά του στοιχεία, καταλήγοντας τελικά στη συνολική του αρχιτεκτονική. Εν συνέχεια ιδιαίτερη βαρύτητα δίνεται στην περιγραφή της λειτουργικότητας και της υλοποίησης των επιμέρους εμπλεκόμενων οντοτήτων και των μηχανισμών επικοινωνίας τους. Τέλος, αναλύονται διεξοδικά οι βασικές αλληλεπιδράσεις και λειτουργίες του συστήματος και περιγράφονται συνοπτικά οι υπηρεσίες που υλοποιήθηκαν για την επιβεβαίωση της ορθής λειτουργικότητας και την επαλήθευση των δυνατοτήτων του συστήματος.

3.1 Εισαγωγή

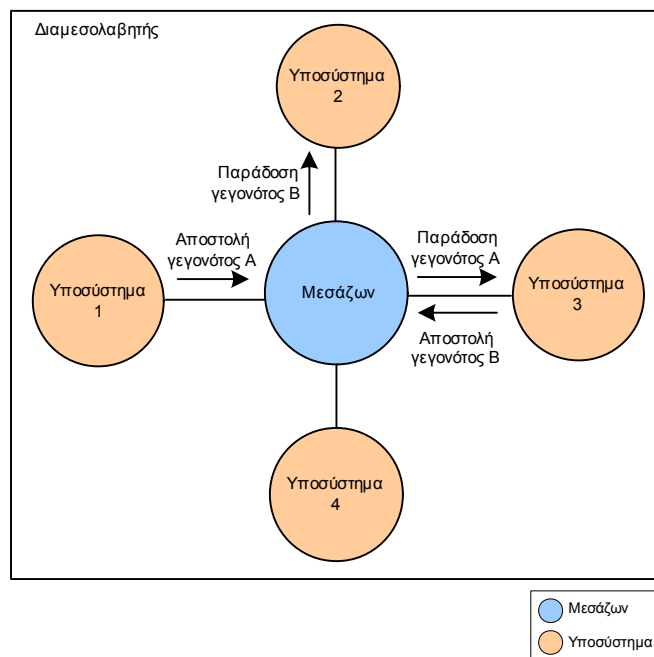
Στο κεφάλαιο αυτό, περιγράφεται η προτυποποιημένη, βασισμένη σε διαμεσολαβητές και πρότυπα προφίλ αρχιτεκτονική που υλοποιήθηκε για την ομογενοποιημένη υποστήριξη παροχής υπηρεσιών προσαρμοσμένων στον εκάστοτε χρήστη και την τρέχουσα κατάσταση του. Ο σχεδιασμός του εν λόγω πλαισίου πραγματοποιήθηκε με δύο βασικά κριτήρια. Το πρώτο ήταν αυτό της ευελιξίας και της ανοικτότητας του συστήματος προκειμένου να παρέχει τη δυνατότητα εύκολης προσθήκης νέων λειτουργιών και υπηρεσιών χωρίς να επηρεάζεται η υπάρχουσα υποδομή και δομή του συστήματος. Ο δεύτερος στόχος ήταν η δημιουργία ενός συστήματος που να παρέχει όλες τις απαραίτητες λειτουργίες για την συλλογή πληροφοριών για την τρέχουσα κατάσταση του χρήστη από τις πηγές τους, την επεξεργασία τους, την διαχείριση τους και τελικά την παροχή τους σε προτυποποιημένη μορφή βάσει των προφίλ, όπως αυτή παρουσιάστηκε στο προηγούμενο κεφάλαιο, σε εξωτερικές οντότητες κατόπιν αιτήσεως.

Στο παρόν κεφάλαιο αρχικά παρατίθενται οι βασικές αρχές που ακολουθήθηκαν για τον σχεδιασμό του συστήματος καθώς και τα βασικά δομικά του στοιχεία, καταλήγοντας τελικά στη συνολική του αρχιτεκτονική. Εν συνεχεία ιδιαίτερη βαρύτητα δίνεται στην περιγραφή της λειτουργικότητας και της υλοποίησης των επιμέρους εμπλεκόμενων οντοτήτων και των μηχανισμών επικοινωνίας τους. Τέλος, αναλύονται διεξοδικά οι βασικές αλληλεπιδράσεις και λειτουργίες του συστήματος και περιγράφονται συνοπτικά οι υπηρεσίες που υλοποιήθηκαν για την επιβεβαίωση της ορθής λειτουργικότητας και την επαλήθευση των δυνατοτήτων του συστήματος.

3.2 Η Αρχιτεκτονική του συστήματος

3.2.1 Η οντότητα του Διαμεσολαβητή: Μεσάζων & Υποσύστημα

Η προτεινόμενη αρχιτεκτονική προκειμένου να καλύψει την απαίτηση της ανοικτότητας και της ευελιξίας και να είναι ανεξάρτητη τερματικού, δικτύων και υπηρεσιών βασίστηκε στην ιδέα ενός καταναμημένου επικαθήμενου (overlay) δικτύου αποτελούμενου από Διαμεσολαβητές (Brokers). Η οντότητα του Διαμεσολαβητή αποτελεί το βασικό δομικό στοιχείο του συστήματος και αποτελείται από μία κεντρική μονάδα, τον Μεσάζοντα (Mediator) και πολλές περιφερειακές, τα Υποσυστήματα (Subsystems) [1][2][3][4].



Σχήμα 24: Επικοινωνία Υποσυστημάτων μέσω του Μεσάζοντα

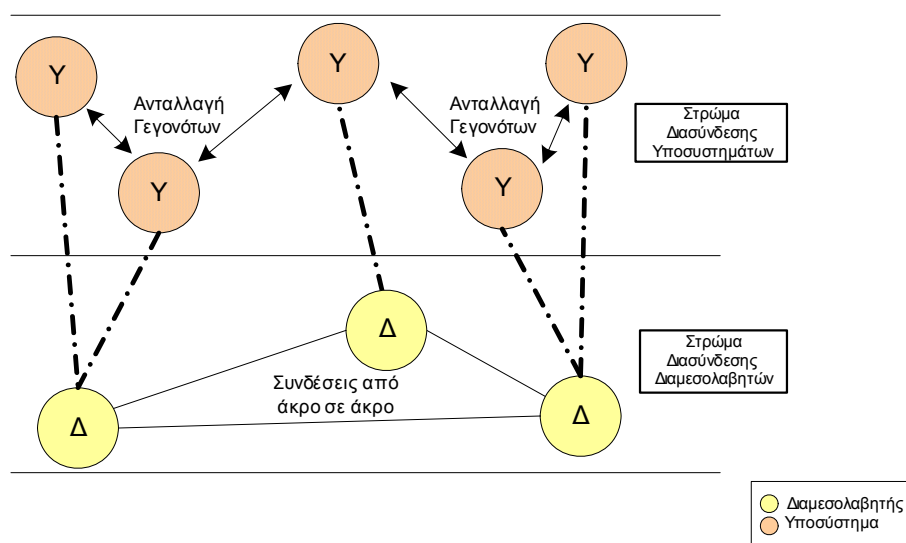
Κάθε υποσύστημα δρα σαν αυτόνομη οντότητα και παρέχει μία συγκεκριμένη λειτουργία ανεξάρτητα από το υπόλοιπο σύστημα, μην έχοντας γνώση γι' αυτό. Η αλληλεπίδραση κάθε υποσυστήματος με τα υπόλοιπα δεν είναι άμεση αλλά πραγματοποιείται με την ανταλλαγή ασύγχρονων μηνυμάτων/γεγονότων (events) μέσω του Μεσάζοντα εφαρμόζοντας την λογική των συστημάτων P2P, και συγκεκριμένα τη λογική των συστημάτων publish-subscribe [5], αλλά σε τοπικό επίπεδο (locally) εσωτερικά στον Διαμεσολαβητή [6][7][8][9]. Κάθε φορά που απαιτείται συνεργασία ανάμεσα σε δύο ή περισσότερα υποσυστήματα ο Μεσάζοντας αναλαμβάνει να συντονίσει την αλληλεπίδραση παραλαμβάνοντας τα μηνύματα από το εκάστοτε υποσύστημα-αποστολέα, πραγματοποιώντας την κατάλληλη επεξεργασία βάσει κατάλληλων κανόνων προκειμένου να συμπεράνει για ποιο ή ποια υποσυστήματα προορίζονται, και τελικά προωθώντας τα στα κατάλληλα υποσυστήματα-παραλήπτες (Σχήμα 24).

Οι κανόνες που επιλέγει να ενεργοποιήσει ο Μεσάζων για την προώθηση των μηνυμάτων εξαρτώνται από τον τύπο του μηνύματος και από τον τύπο του αποστολέα. Ο Μεσάζων είναι σε θέση ακόμα και να απορρίψει κάποιο μήνυμα όταν το απαιτούν οι συνθήκες και το υπαγορεύουν οι κανόνες προώθησης των μηνυμάτων.

3.2.2 Μηχανισμός Επικοινωνίας Διαμεσολαβητών

Για την περίπτωση που το υποσύστημα-αποστολέας βρίσκεται σε κάποιον άλλο Διαμεσολαβητή του κατανεμημένου συστήματος από το υποσύστημα-παραλήπτη, τότε είναι

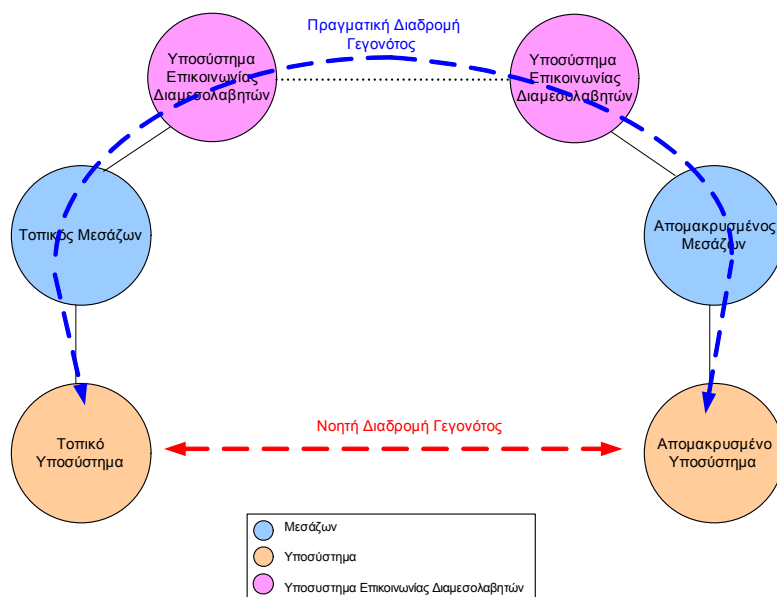
αναγκαία η αλληλεπίδραση ανάμεσα σε δύο απομακρυσμένους Διαμεσολαβητές και απαιτείται κατάλληλος μηχανισμός επικοινωνίας. Ο μηχανισμός αυτός ονομάζεται Μηχανισμός Επικοινωνίας Διαμεσολαβητών (Brokers Communication Mechanism, BCM) [8][9]. Επεκτείνει τον ασύγχρονο μηχανισμό ανταλλαγής γεγονότων που χρησιμοποιείται για την επικοινωνία ανάμεσα στα τοπικά υποσυστήματα ενός Διαμεσολαβητή, προκειμένου τα γεγονότα να μεταφέρονται και δικτυακά σε απομακρυσμένα υποσυστήματα. Βασικό χαρακτηριστικό της δια-διαμεσολαβητικής επικοινωνίας αποτελεί το γεγονός ότι είναι διαφανής στα εμπλεκόμενα υποσυστήματα. Ένα υποσύστημα που αποστέλλει ένα γεγονός δεν αντιλαμβάνεται αν αυτό το γεγονός προωθείται σε τοπικό ή απομακρυσμένο υποσύστημα μέσω του Μεσάζοντα ή του Μηχανισμού Επικοινωνίας Διαμεσολαβητών αντίστοιχα. Το ίδιο ισχύει και για την περίπτωση που ένα υποσύστημα παραλαμβάνει ένα γεγονός, δεν γνωρίζει αν το γεγονός προέρχεται από τοπικό ή απομακρυσμένο υποσύστημα. Ουσιαστικά αυτό το χαρακτηριστικό σε λογικό επίπεδο είναι δυνατόν να αναχθεί στα δύο διαφορετικά στρώματα από τα οποία αποτελείται ο μηχανισμός επικοινωνίας (Σχήμα 25).



Σχήμα 25: Τα στρώματα του Μηχανισμού Επικοινωνίας Διαμεσολαβητών

Στο πρώτο στρώμα βρίσκεται το δίκτυο των διασυνδεδεμένων Διαμεσολαβητών. Στο επίπεδο αυτό κάθε διαμεσολαβητής γνωρίζει την ύπαρξη των υπολοίπων και τη συνολική τοπολογία. Στο υψηλότερο στρώμα συναντάμε το δίκτυο των υποσυστημάτων για το οποίο η τοπολογία του χαμηλότερου στρώματος είναι εντελώς διάφανη. Στο στρώμα αυτό, σε λογικό επίπεδο, δεν υφίσταται η έννοια του τοπικού ή απομακρυσμένου υποσυστήματος. Κάθε ένα μπορεί να ανταλλάξει μηνύματα με οποιοδήποτε άλλο χωρίς να αντιλαμβάνεται αν βρίσκονται στον ίδιο ή σε διαφορετικό Διαμεσολαβητή.

Ο παραπάνω μηχανισμός επικοινωνίας υλοποιείται με τη μορφή υποσυστήματος, το Υποσύστημα Επικοινωνίας Διαμεσολαβητών (Broker Communication Subsystem, BCS), η ύπαρξη του οποίου είναι υποχρεωτική σε οποιονδήποτε Διαμεσολαβητή εκτελεί αλληλεπιδράσεις με άλλους Διαμεσολαβητές. Το υποσύστημα αυτό ουσιαστικά υλοποιεί και τα δύο στρώματα του μηχανισμού επικοινωνίας καθότι είναι υπεύθυνο και για τη διαχείριση των συνδέσεων ανάμεσα στους διαμεσολαβητές αλλά και για τη διάφανη μεταφορά των γεγονότων μεταξύ των υποσυστημάτων.



Σχήμα 26: Το Υποσύστημα Επικοινωνίας Διαμεσολαβητών

Στο Σχήμα 26 παρουσιάζεται ένα παράδειγμα αλληλεπίδρασης απομακρυσμένων υποσυστημάτων. Δημιουργείται ένα γεγονός το οποίο προωθείται στον Μεσάζοντα αλλά δεν μπορεί να εξυπηρετηθεί τοπικά. Ο Μεσάζων προωθεί εν συνεχεία το γεγονός στο Υποσύστημα Επικοινωνίας Διαμεσολαβητών το οποίο εκκινεί διαδικασία εύρεσης κατάλληλου υποσυστήματος-παραλήπτη. Η διαδικασία αυτή πραγματοποιείται σε επίπεδο διαμεσολαβητών. Μόλις ανακαλυφθεί ο Διαμεσολαβητής που φιλοξενεί το υποσύστημα-παραλήπτη, το γεγονός αποστέλλεται σε αυτόν, παραλαμβάνεται από το απομακρυσμένο υποσύστημα επικοινωνίας, προωθείται στον Μεσάζοντα και τελικά στο υποσύστημα-παραλήπτη. Σε λογικό επίπεδο για το στρώμα των υποσυστημάτων η διαδρομή που γίνεται αντιληπτή είναι αυτή που στο Σχήμα 26 εμφανίζεται ως Νοητή Διαδρομή Γεγονότος.

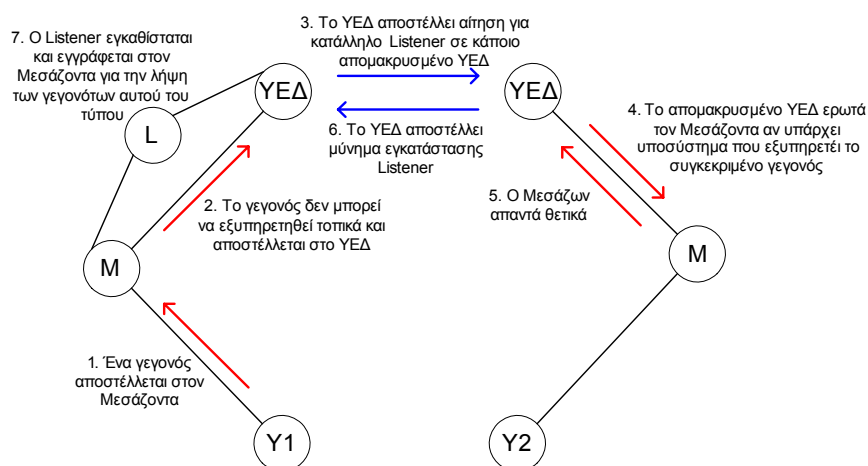
3.2.2.1 Αρχιτεκτονική Υποσυστήματος Επικοινωνίας Διαμεσολαβητών

Το Υποσύστημα Επικοινωνίας Διαμεσολαβητών προκειμένου να εξυπηρετεί την διάφανη αλληλεπίδραση μεταξύ απομακρυσμένων υποσυστημάτων υλοποιεί τις παρακάτω

διαδοχικές διαδικασίες: δύο φάσεις ανεύρεσης, αρχικά διαμεσολαβητών και εν συνεχεία υποσυστημάτων που μπορούν να λαμβάνουν τα γεγονότα που δεν είναι δυνατόν να εξυπηρετηθούν τοπικά στον Διαμεσολαβητή που δημιουργήθηκαν και την επικοινωνία ανάμεσα στα απομακρυσμένα υποσυστήματα για την μεταφορά των γεγονότων [8][9].

Ανακάλυψη Απομακρυσμένων Διαμεσολαβητών: Κατά τη διάρκεια αυτής της φάσης, κάθε ΥΕΔ ανακαλύπτει τα υπόλοιπα ΥΕΔ του συστήματος δημιουργώντας έτσι το πρώτο επίπεδο επικοινωνίας ανάμεσα στους Διαμεσολαβητές του συστήματος οι οποίοι έχουν γνώση της συνολικής τοπολογίας. Αυτή η διαδικασία μπορεί να είναι είτε στατική είτε δυναμική. Στην πρώτη περίπτωση κάθε Διαμεσολαβητής διατηρεί λίστα με τους γειτονικούς του Διαμεσολαβητές η οποία δημιουργείται κατά την εκκίνηση του συστήματος. Στην δεύτερη περίπτωση, απαιτείται η ύπαρξη υπηρεσίας εγγραφής Διαμεσολαβητών (registry service, naming service). Κάθε Διαμεσολαβητής υποχρεούται κατά την ενεργοποίησή του να εγγράφεται στην υπηρεσία και να ανανεώνει την εγγραφή του σε τακτά χρονικά διαστήματα προκειμένου η λίστα να διατηρείται ενήμερη για την τρέχουσα κατάσταση του δικτύου των ενεργών Διαμεσολαβητών. Μετά την ολοκλήρωση της εγγραφής κάθε Διαμεσολαβητής έχει την δυνατότητα να «ρωτήσει» την υπηρεσία εγγραφής και να ενημερωθεί για την λίστα με τους ενεργούς Διαμεσολαβητές.

Ανακάλυψη Απομακρυσμένων Υποσυστημάτων και Εγγραφή τους για την Εξυπηρέτηση Γεγονότων που δεν δύναται να εξυπηρετηθούν τοπικά: Στο Σχήμα 27 παρουσιάζονται τα βήματα που ακολουθούνται κατά την διαδικασία ανακάλυψης και εγγραφής απομακρυσμένων υποσυστημάτων για την εξυπηρέτηση τοπικά δημιουργημένων γεγονότων και εν συνεχεία παρατίθεται η περιγραφή τους.



Σχήμα 27: Διαδικασία Ανακάλυψης και Εγγραφής Απομακρυσμένων Υποσυστημάτων

Η φάση ανακάλυψης και εγγραφής απομακρυσμένων υποσυστημάτων για την εξυπηρέτηση τοπικά δημιουργημένων γεγονότων πραγματοποιείται αφού όλοι οι Διαμεσολαβήτες έχουν λάβει γνώση για την συνολική τοπολογία του δικτύου τους. Κατά τη διάρκεια της, το ΥΕΔ κάθε Διαμεσολαβητή ζητά να ενημερωθεί από τους υπόλοιπους για τα υποσυστήματα που διαθέτουν και τα γεγονότα που αυτά μπορούν να παραλάβουν και να εξυπηρετήσουν. Η διαδικασία αυτή πραγματοποιείται είτε κατά την εκκίνηση του συστήματος, είτε περιοδικά, είτε κατόπιν της αποστολής από τον Μεσάζοντα στο ΥΕΔ κάποιου γεγονότος το οποίο δεν εξυπηρετήθηκε από τα τοπικά υποσυστήματα.

Η διαδικασία που ακολουθείται είναι η εξής: αρχικά το ΥΕΔ είτε λαμβάνει από τον Μεσάζοντα μία λίστα από γεγονότα τα οποία δημιουργούνται από τα τοπικά υποσυστήματα αλλά δεν εξυπηρετούνται από κανένα από αυτά ή λαμβάνει ένα γεγονός το οποίο του παραδόθηκε διότι δεν εξυπηρετήθηκε τοπικά. Το ΥΕΔ επικοινωνεί με τα απομακρυσμένα ΥΕΔ των άλλων διαμεσολαβητών ρωτώντας αν μπορούν να εξυπηρετήσουν τα συγκεκριμένα γεγονότα ή το συγκεκριμένο γεγονός. Οι Διαμεσολαβητές που φιλοξενούν τα κατάλληλα υποσυστήματα απαντούν στον ερωτώντα Διαμεσολαβητή μέσω των ΥΕΔ και δημιουργείται ανάμεσα τους κατάλληλη επικοινωνιακή σύνοδος, η οποία ουσιαστικά καθορίζει ότι το τοπικό γεγονός θα μεταφερθεί στον απομακρυσμένο Διαμεσολαβητή μέσω του ΥΕΔ, το οποίο με τη σειρά του θα το παραδώσει στον Μεσάζοντα, ο οποίος θα φροντίσει να το παραδώσει στο κατάλληλο υποσύστημα.

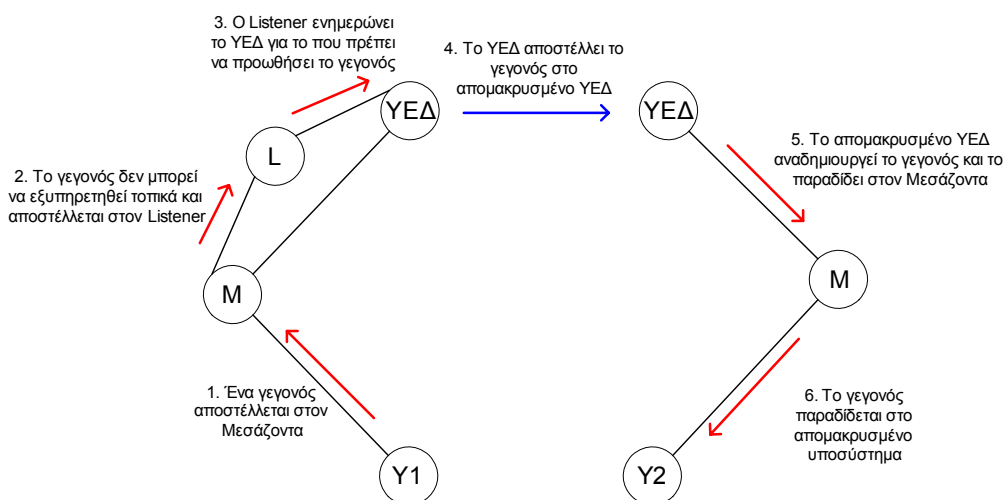
Η σύνοδος που εγκαθίσταται ανάμεσα στους δύο Διαμεσολαβητές ακολουθεί το προγραμματιστικό σχεδιαστικό πρότυπο του αντιπροσώπου που βασίζεται στην λογική της έκθεσης απομακρυσμένων διεπαφών τοπικά. Ουσιαστικά δημιουργείται τοπικά ένας αντιπρόσωπος του απομακρυσμένου αντικείμενου ο οποίος παρέχει την ίδια διεπαφή (interface) και την δυνατότητα αλληλεπίδρασης με το απομακρυσμένο αντικείμενο μέσω του δικτύου αλλά με διαφανή τρόπο. Στο πλαίσιο του συγκεκριμένου συστήματος το αντικείμενο αντιπρόσωπος ονομάζεται Listener. Κάθε φορά που ένα απομακρυσμένο ΥΕΔ απαντά θετικά σε ένα τοπικό ΥΕΔ για την εξυπηρέτηση κάποιου γεγονότος συγχρόνως του δίνει και την εντολή για την δημιουργία κατάλληλου Listener. Το τοπικό ΥΕΔ εν συνεχεία δημιουργεί τον τοπικό Listener για την εξυπηρέτηση του γεγονότος ο οποίος εγγράφεται στον Μεσάζοντα προκειμένου να του αποστέλλονται γεγονότα του συγκεκριμένου τύπου. Στην περίπτωση που το τοπικό ΥΕΔ παραλάβει περισσότερες της μίας θετικές απαντήσεις για την εξυπηρέτηση του γεγονότος προσθέτει στον Listener τους επιπλέον Διαμεσολαβητές-προορισμούς για το συγκεκριμένο γεγονός. Με αυτόν τον τρόπο είναι δυνατόν να πραγματοποιείται αποστολή των γεγονότων σε διαφορετικούς Διαμεσολαβητές

και έτσι να επιτυγχάνεται πιο αποτελεσματική κατανομή φορτίου (load balancing) στο σύστημα.

Το αποτέλεσμα αυτής της φάσης είναι η δημιουργία ενός δεύτερου επιπέδου επικοινωνίας το οποίο αποτελείται από τοπικά και απομακρυσμένα υποσυστήματα τα οποία με την ύπαρξη των Listeners ανταλλάσσουν γεγονότα σαν να ήταν εγκατεστημένα στον ίδιο Μεσάζοντα.

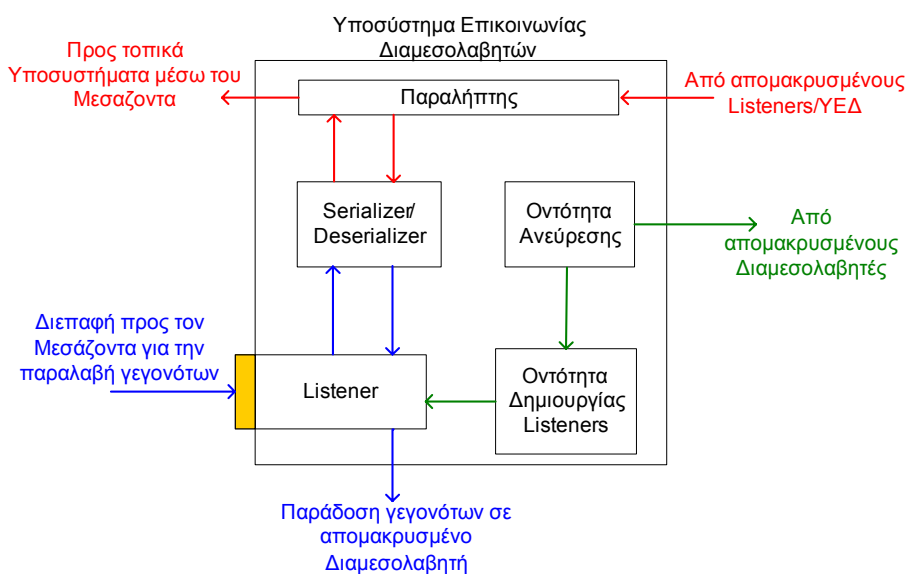
Μεταφορά Γεγονότων σε Απομακρυσμένα Υποσυστήματα: Μετά το πέρας των φάσεων ανακάλυψης απομακρυσμένων Διαμεσολαβητών και Υποσυστημάτων και της εγκατάστασης των Listeners, είναι δυνατή η επικοινωνία και η ανταλλαγή γεγονότων ανάμεσα σε απομακρυσμένα υποσυστήματα. Στο Σχήμα 28 παρουσιάζονται τα βήματα που ακολουθούνται κατά την διαδικασία αποστολής γεγονότος σε απομακρυσμένο υποσύστημα.

Η διαδρομή του γεγονότος ξεκινάει από το Υποσύστημα που το δημιουργεί και το παραδίδει στον Μεσάζοντα. Εν συνεχεία ο Μεσάζοντας προωθεί το γεγονός στον κατάλληλο Listener, μετά παραδίδεται στο ΥΕΔ και αυτό με τη σειρά του το παραδίδει στο απομακρυσμένο ΥΕΔ που του υποδεικνύει ο Listener. Όταν το γεγονός φθάσει στον απομακρυσμένο Διαμεσολαβητή το ΥΕΔ το παραδίδει στον Μεσάζοντα και αυτός στο κατάλληλο υποσύστημα. Κάθε οντότητα από την οποία περνάει το γεγονός κατά την αποστολή του, το σημαδεύει (stamp) έτσι ώστε το γεγονός απάντηση να μπορέσει να ακολουθήσει την ίδια διαδρομή προκειμένου να φτάσει στο υποσύστημα αποστολέα. Κατά την διαδικασία αποστολής το γεγονός πρέπει να διαμορφωθεί ανάλογα με το πρωτόκολλο επικοινωνίας που χρησιμοποιείται από τα ΥΕΔ. Αυτό σημαίνει ότι απαιτείται ο καθορισμός κατάλληλης διαδικασίας σειριοποίησης/αποσειριοποίησης (serialization/de-serialization) των μεταφερόμενων δεδομένων.



Σχήμα 28: Μεταφορά γεγονότος σε απομακρυσμένο υποσύστημα

Στο Σχήμα 29 παρουσιάζεται η εσωτερική αρχιτεκτονική του Υποσυστήματος Επικοινωνίας Διαμεσολαβητών.



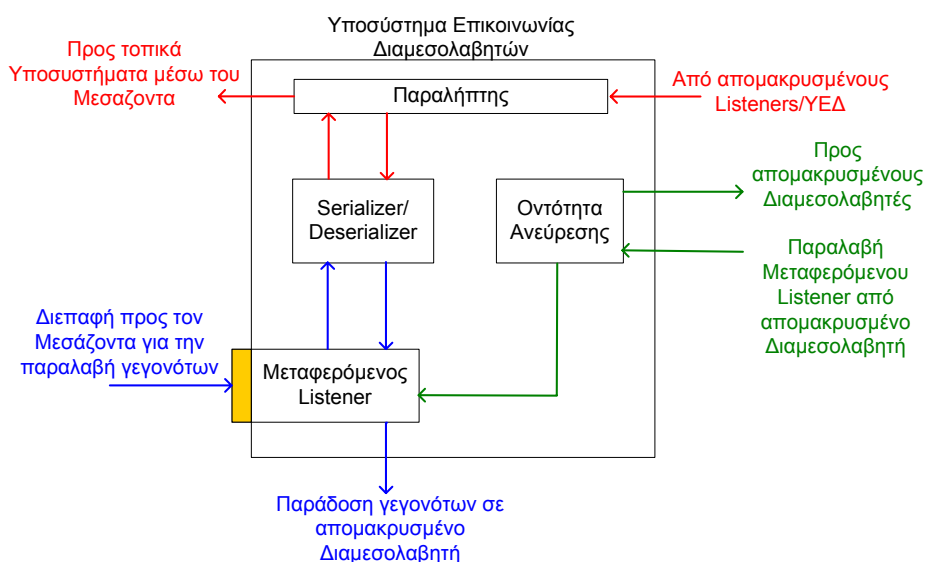
Σχήμα 29: Αρχιτεκτονική του Υποσυστήματος Επικοινωνίας Διαμεσολαβητών (ΥΕΔ)

Το ΥΕΔ αποτελείται από τις παρακάτω λειτουργικές οντότητες [8][9]:

- Ο **Παραλήπτης** συλλέγει τα γεγονότα που προέρχονται από τα απομακρυσμένα ΥΕΔ, τα αναδημιουργεί και τα παραδίδει μέσω του Μεσάζοντα στο τοπικό υποσύστημα. Η διαδικασία αυτή απαιτεί και τις υπηρεσίες που παρέχει η οντότητα Serializer/Deserializer.
- Ο **Serializer/Deserializer** μετατρέπει τα γεγονότα στην μορφή που απαιτείται από το πρωτόκολλο επικοινωνίας που χρησιμοποιούν τα ΥΕΔ. Ο Serializer/Deserializer σειριοποιεί τα γεγονότα χρησιμοποιώντας μία μορφή της δημοφιλούς γλώσσας XML. Έτσι γίνεται εύκολη η απευθείας χρήση του σειριοποιημένου γεγονότος σε πρωτόκολλα ανταλλαγής μηνυμάτων που βασίζονται στην XML (SOAP).
- Η οντότητα **Ανεύρεσης** πραγματοποιεί τις διαδικασίες ανεύρεσης των απομακρυσμένων υποσυστημάτων και των γεγονότων που εξυπηρετούν πραγματοποιώντας τις κατάλληλες ερωτήσεις στους απομακρυσμένους Διαμεσολαβητές. Παρέχει τις σχετικές πληροφορίες στην οντότητα Δημιουργίας Listeners.
- Η οντότητα **Δημιουργίας Listeners** δημιουργεί τους τοπικούς Listeners για τα απομακρυσμένα υποσυστήματα και τους ενσωματώνει στον μηχανισμό αποστολής μηνυμάτων του Μεσάζοντα.

- Ο **Listener** αποτελεί τον τοπικό αντιπρόσωπο του απομακρυσμένου υποσυστήματος. Αποτελείται από δύο μέρη: μία διεπαφή προς τον Μεσάζοντα μέσω της οποίας συλλέγει τα γεγονότα και μία οντότητα που δένει την διεπαφή αυτή με τον μηχανισμό επικοινωνίας του ΥΕΔ και καθιστά δυνατή την μεταφορά των γεγονότων. Η διαδικασία αυτή απαιτεί και τις υπηρεσίες που παρέχει η οντότητα Serializer/Deserializer.

Μία εναλλακτική λύση, όσον αφορά στην διαδικασία δημιουργίας του Listener, θα ήταν αυτή να πραγματοποιείται στον απομακρυσμένο Διαμεσολαβητή με την ενεργοποίηση κάθε νέου υποσυστήματος. Όταν κάποιο γεγονός χρειαστεί να εξυπηρετηθεί από το απομακρυσμένο υποσύστημα η οντότητα Ανεύρεσης εντοπίζει τον ήδη δημιουργημένο Listener στο απομακρυσμένο υποσύστημα και τον μεταφέρει/εγκαθιστά στο τοπικό ΥΕΔ.



Σχήμα 30: Αρχιτεκτονική Υποσυστήματος Επικοινωνίας Διαμεσολαβητών (ΥΕΔ) με μεταφερόμενο Listener

Στο Σχήμα 30 παρουσιάζεται η εσωτερική αρχιτεκτονική για την περίπτωση του μεταφερόμενου Listener. Στο σχήμα αυτό δεν εμφανίζεται η οντότητα Δημιουργίας Listener διότι έχει μεταφερθεί στον απομακρυσμένο Διαμεσολαβητή. Το πλεονέκτημα της προσέγγισης αυτής είναι το γεγονός ότι ο Listener του κάθε υποσυστήματος δημιουργείται μονάχα μία φορά καθ' όλη τη διάρκεια της ζωής του υποσυστήματος και μόνο κατά την ενεργοποίησή του. Αυτό έρχεται σε αντίθεση με την προηγούμενη σχεδιαστική προσέγγιση κατά την οποία ο Listener δημιουργείται κάθε φορά που απαιτείται. Το μειονέκτημα του μεταφερόμενου Listener είναι το γεγονός ότι απαιτείται ο Διαμεσολαβητής που δημιουργεί τον Listener και ο Διαμεσολαβητής που τον παραλαμβάνει να είναι συμβατοί όσον αφορά την πλατφόρμα υλοποίησής τους.

Επιπλέον, η μεταφορά μίας μονάδας λογισμικού δημιουργεί την ανάγκη για πρόσθετα μέτρα ασφάλειας στο άκρο της επικοινωνίας που δέχεται και εγκαθιστά το λογισμικό αυτό. Πρέπει να εξασφαλιστεί ότι το τμήμα λογισμικού που αποστέλλει η μία οντότητα στην άλλη είναι πράγματι ένας Listener και όχι κάποια προσπάθεια κακόβουλης χρήσης των υπηρεσιών του κάθε Διαμεσολαβητή. Για αυτό το σκοπό μπορεί να χρησιμοποιηθεί ένα σύστημα πιστοποίησης με ψηφιακές υπογραφές, που θα υποδεικνύει σε κάθε διαμεσολαβητή ποια τμήματα λογισμικού είναι έγκυρα προς εγκατάσταση και ποια όχι.

Ένα άλλο σημαντικό ζήτημα είναι η διάρκεια ζωής των Listeners. Κάθε φορά που ένας Διαμεσολαβητής ή ένα υποσύστημα αποχωρούν από το σύστημα τότε ακυρώνονται και οι Listeners τους, οι οποίοι βρίσκονται εγκατεστημένοι στα ΥΕΔ άλλων Διαμεσολαβητών. Αν εξαιτίας σφάλματος ή δυσλειτουργίας κάποιος ΥΕΔ δεν ακυρώσει τους απομακρυσμένους Listeners που του αναλογούν τότε είναι ευθύνη του κάθε ΥΕΔ να εφαρμόσει μια πολιτική περιοδικού ελέγχου και να ακυρώσει με δική του πρωτοβουλία τους Listeners που δεν έχουν απομακρυσμένους στόχους. Η διαδικασία ανακάλυψης και δημιουργίας ή αποστολής Listeners επαναλαμβάνεται κάθε φορά που εκκινείται κάποιος διαμεσολαβητής.

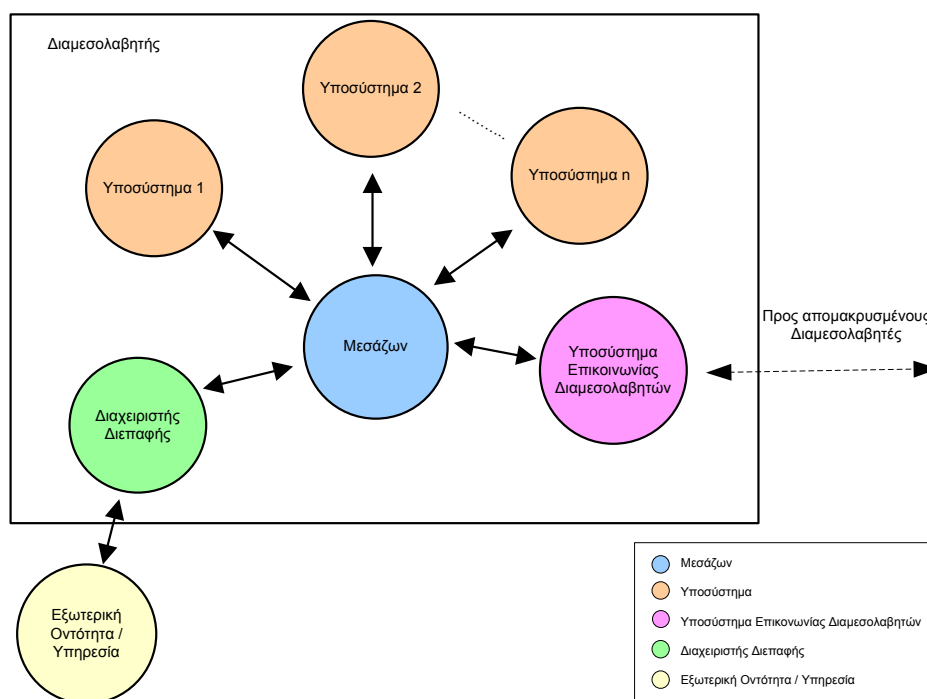
3.2.3 Ο Διαχειριστής Εξωτερικής Διεπαφής

Διατηρώντας λοιπόν την ευφυΐα της αντιστοίχισης ανάμεσα σε αποστολέα, μήνυμα και παραλήπτες στον Μεσάζοντα και στο Υποσύστημα Επικοινωνίας Διαμεσολαβητών, οι διαφορετικές λειτουργίες που καλείται να παρέχει το σύστημα, ακόμα και αν απαιτείται η αλληλεπίδραση τους, διατηρούνται εντελώς ανεξάρτητες η μία από την άλλη και συνεπώς δίνεται η δυνατότητα να προστίθενται στο σύστημα λειτουργίες με τη μορφή νέων υποσυστημάτων χωρίς να χρειάζονται δραστικές αλλαγές στην υπάρχουσα λειτουργικότητα. Κάθε νέο υποσύστημα που προστίθεται στον Διαμεσολαβητή δηλώνει (publish) όλα τα μηνύματα που θα αποστέλλει και εγγράφεται (subscribe) για όλα τα μηνύματα που μπορεί να λαμβάνει στον Μεσάζοντα.

Ο τρόπος ενσωμάτωσης νέων εγγενών λειτουργιών του συστήματος με τη μορφή νέων υποσυστημάτων μπορεί να εφαρμοστεί και στην περίπτωση που απαιτείται η αλληλεπίδραση του συστήματος με κάποια εξωτερική υπάρχουσα οντότητα ή υπηρεσία, όπως για παράδειγμα ο Εξυπηρετητής Θέσης (Location Server) κάποιου παρόχου δικτύου (network provider) για την ανάκτηση της θέσης του χρήστη ή η εφαρμογή κάποιου Ανεξάρτητου Παροχέα Υπηρεσιών (Independent Service Provider, ISP) η οποία αλληλεπιδρά με το σύστημα για την ανάκτηση πληροφορίας απαραίτητης για την εκτέλεση της υπηρεσίας. Σε αυτήν την περίπτωση προστίθεται στον Διαμεσολαβητή ένα υποσύστημα

προσαρμογής, που ονομάζεται Διαχειριστής Εξωτερικής Διεπαφής (Interface Manager, IM). Ο Διαχειριστής αυτός συμπεριφέρεται ακριβώς όπως τα υπόλοιπα υποσυστήματα, αποτελεί την διεπαφή με την εξωτερική οντότητα/υπηρεσία και μέσω αυτού πραγματοποιείται η ροή πληροφορίας ανάμεσα στο σύστημα και την εξωτερική οντότητα. Οποιαδήποτε προσαρμογή απαιτείται στην εξωτερική υπάρχουσα οντότητα/υπηρεσία για την αλληλεπίδρασή της με το σύστημα πραγματοποιείται εντός του Διαχειριστή Εξωτερικής Διεπαφής και η εξωτερική οντότητα παραμένει ανέπαφη. Επίσης, αν η εξωτερική οντότητα απαιτεί διαφορετικής δομής ή επιπέδου πληροφορία από αυτήν που παρέχει το σύστημα, η λειτουργικότητα μετατροπής της υλοποιείται εντός του Διαχειριστή Εξωτερικής Διεπαφής.

Στο Σχήμα 31 παρουσιάζεται η γενική αρχιτεκτονική του Διαμεσολαβητή που αλληλεπιδρά και με εξωτερική οντότητα. Όσον αφορά στην υλοποίηση του συγκεκριμένου υποσυστήματος εξαρτάται κάθε φορά από τους σκοπούς που καλείται να εξυπηρετεί. Σε επόμενες ενότητες θα παρουσιαστούν οι υλοποιήσεις των Διαχειριστών Εξωτερικών Διεπαφών που πραγματοποιήθηκαν στο πλαίσιο της παρούσας διατριβής.



Σχήμα 31: Γενική Αρχιτεκτονική Διαμεσολαβητή που αλληλεπιδρά με εξωτερική οντότητα

3.2.4 Προσωπική Συσκευή Χρήστη / Υποσύστημα Διαχειριστή Πρόσβασης Προσωπικής Συσκευής Χρήστη (ΥΔΠΠΣΧ)

Η Προσωπική Συσκευή Χρήστη (ΠΣΧ) [10] είναι μία μικρή φορητή συσκευή την οποία ο χρήστης έχει πάντα μαζί του και αποτελεί την ταυτότητα του εντός του συγκεκριμένου

πλαίσιο. Εντός της βρίσκονται αποθηκευμένα δεδομένα του χρήστη όπως το προφίλ του, οι προτιμήσεις του, οι πολιτικές του κλπ.. Δεδομένα δηλαδή προσωπικά και ευαίσθητα τα οποία μόνο ο χρήστης πρέπει να διατηρεί και να διαχειρίζεται, καθώς και να τα παρέχει κατά βούληση στο σύστημα. Από την άλλη πλευρά η αποκάλυψη (disclosure) των δεδομένων αυτών είναι απαραίτητη για την παραμετροποιημένη εκτέλεση των υπηρεσιών γεγονός που έρχεται σε αντίθεση με την προσωπική φύση τους. Με την εισαγωγή της συσκευής αυτής στο σύστημα καλύπτονται όλες οι παραπάνω απαιτήσεις. Ο χρήστης διατηρεί και διαχειρίζεται τα δεδομένα του τοπικά εντός της ΠΣΧ, με αποτέλεσμα να αποφεύγεται η αποθήκευση τους στο δίκτυο, στις υποδομές κάποιου μη αξιόπιστου λειτουργού δικτύου ή παρόχου υπηρεσιών, και η παράνομη και μη εξουσιοδοτημένη εκμετάλλευσή τους. Με χρήση πολιτικών/κανόνων ο χρήστης μπορεί να καθορίζει ποια από τα δεδομένα του και υπό ποιες συνθήκες επιτρέπει να παρέχονται προς τις υπηρεσίες διασφαλίζοντας έτσι κατά μία έννοια την ιδιωτικότητά τους. Συγχρόνως όμως, η ΠΣΧ αποτελεί μία νέα πηγή συγκεντρωμένης και ολοκληρωμένης πληροφορίας (context source) από την οποία το υπόλοιπο σύστημα ανακτά ανά πάσα στιγμή τα προσωπικά δεδομένα του χρήστη και κατόπιν κατάλληλης επεξεργασίας τα παραδίδει στις υπηρεσίες για την εκτέλεση τους.

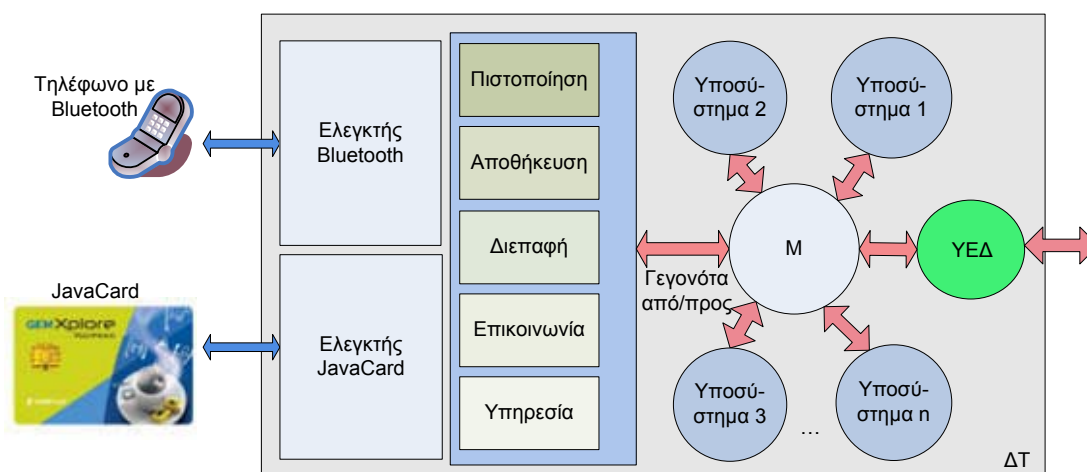
Για την αλληλεπίδραση της ΠΣΧ με το υπόλοιπο σύστημα απαιτείται η σύνδεση της με την τερματική συσκευή του χρήστη και η επικοινωνία της με τον Διαμεσολαβητή Τερματικού μέσω ειδικού υποσυστήματος το οποίο ονομάζεται Υποσύστημα Διαχειριστή Πρόσβασης Προσωπικής Συσκευής Χρήστη (Personal Device Access Manager, PDAM).

Η βασική απαίτηση για τον σχεδιασμό της ΠΣΧ ήταν η μεταφερσιμότητα, να είναι μικρή σε μέγεθος ώστε ο χρήστης να έχει τη δυνατότητα να την έχει πάντοτε μαζί του, όπως παραδείγματος χάριν μία κάρτα Java ή ένα USB stick ή ακόμα Προσωπική Συσκευή σε μορφή λογισμικού. Αυτή η απαίτηση όμως δημιουργεί αυτομάτως περιορισμούς όσον αφορά στις αποθηκευτικές και επεξεργαστικές δυνατότητες της συσκευής. Στην περίπτωση αυτή υπάρχει η δυνατότητα μέρος των λειτουργιών της ΠΣΧ να υλοποιείται στο τερματικό εντός του υποσυστήματος Διαχειριστή Πρόσβασης Προσωπικής Συσκευής Χρήστη, καθώς και να αλληλεπιδρά με τους Διαμεσολαβητές Τερματικού ή Δικτύου οι οποίοι καλούνται να παρέχουν συμπληρωματικές στην ΠΣΧ λειτουργίες.

Στο πλαίσιο της παρούσας προσπάθειας, οι Προσωπικές Συσκευές Χρήστη που χρησιμοποιήθηκαν ήταν οι εξής [10]:

- ΠΣΧ βασισμένη σε JavaCard
- Κάρτα SIM συνδυασμένη με κινητό τηλέφωνο με δυνατότητες Bluetooth.

Και οι δύο υλοποιήσεις καλύπτουν την απαίτηση της μεταφερσιμότητας, μπορούν εύκολα να συνδεθούν με ευρεία γκάμα τερματικών συσκευών (η JavaCard μέσω διεπαφής USB και το κινητό μέσω του ευρέως πλέον διαδεδομένου πρωτοκόλλου Bluetooth) και προσφέρουν, έστω και περιορισμένες αποθηκευτικές και επεξεργαστικές δυνατότητες. Για την αντιμετώπιση των περιορισμών αυτών, καθώς και για την ενσωμάτωση των διαφορετικών ΠΣΧ με τη συνολική αρχιτεκτονική των διαμεσολαβητών υλοποιείται στην πλευρά του τερματικού με τη μορφή υποσυστήματος ο Διαχειριστής Πρόσβασης Προσωπικής Συσκευής Χρήστη (ΔΠΠΣΧ). Εντός του ΔΠΠΣΧ υλοποιούνται όλες εκείνες οι λειτουργίες της ΠΣΧ που είναι ανεξάρτητες της εκάστοτε υλοποίησης ΠΣΧ καθώς και οι Ελεγκτές Επικοινωνίας οι οποίοι υλοποιούν την διεπαφή ανάμεσα στο ΔΠΠΣΧ και την ΠΣΧ ανάλογα με τον μηχανισμό επικοινωνίας που χρησιμοποιεί η τελευταία (π.χ. Bluetooth για την περίπτωση της ΠΣΧ με τη μορφή SIM-κινητού τηλεφώνου). Με αυτόν τον τρόπο παρέχεται ομοιόμορφη διεπαφή για όλους τους τύπους ΠΣΧ προς τον Διαμεσολαβητή Τερματικού και κατά συνέπεια και με τους Διαμεσολαβητές Δικτύου.



Σχήμα 32: Η αρχιτεκτονική της Προσωπικής Συσκευής Χρήστη και του Υποσυστήματος Διαχείρισης Πρόσβασης Προσωπικής Συσκευής Χρήστη

Η κοινή λειτουργικότητα για όλες τις υλοποιήσεις ΠΣΧ χωρίζεται σε πέντε διαχειριστές. Ο Διαχειριστής Πιστοποίησης είναι υπεύθυνος για την πιστοποίηση της ΠΣΧ για θέματα ασφαλείας. Πιστοποιεί τον χρήστη έναντι της ΠΣΧ με την εισαγωγή από την πλευρά του χρήστη μέσω κατάλληλης γραφικής διεπαφής (Graphical User Interface, GUI) Προσωπικού Αναγνωριστικού Αριθμού (Personal Identifier Number, PIN) και πιστοποίηση της ΠΣΧ έναντι του δικτύου ώστε να είναι δυνατή η χρήση των διαθέσιμων υπηρεσιών. Επιπλέον είναι υπεύθυνος για τον έλεγχο δικαιωμάτων πρόσβασης στην ΠΣΧ εξωτερικών οντοτήτων όπως υπηρεσιών ή εφαρμογών. Ο Διαχειριστής Αποθήκευσης είναι υπεύθυνος για την διαχείριση των αποθηκευμένων στην ΠΣΧ δεδομένων. Επιτρέπει την πρόσβαση στα

δεδομένα αυτά μέσω Συστήματος Φακέλων (File System), παρέχοντας δυνατότητες ανάγνωσης (read), εγγραφής (write), διαγραφής (delete) και ανανέωσης (update). Ο Διαχειριστής Διεπαφής αναλαμβάνει να παραλαμβάνει τα γεγονότα που φθάνουν στον ΔΠΠΣΧ από τον Μεσάζοντα και να τα προωθεί στις κατάλληλες οντότητες εντός του υποσυστήματος, όπως και να ενσωματώνει τις απαντήσεις των οντοτήτων του υποσυστήματος σε γεγονότα και να τα προωθεί στον Μεσάζοντα. Ο Διαχειριστής Επικοινωνίας παρέχει τις υπηρεσίες ανίχνευσης παρουσίας και μεταβολής κατάστασης (presence detection & status change) της ΠΣΧ, ανακάλυψης των δυνατοτήτων της, δημιουργίας του κατάλληλου Ελεγκτή Επικοινωνίας και εύρεσης και αλληλεπίδρασης με δεδομένα σχετικά με την ΠΣΧ που βρίσκονται αποθηκευμένα στο δίκτυο. Τέλος, ο Διαχειριστής Υπηρεσίας προσφέρει στις υπόλοιπες οντότητες του υποσυστήματος κάποιες υπηρεσίες όπως προσωρινή αποθήκευση (caching) του προφίλ του από την ΠΣΧ, κωδικοποίηση, αποκωδικοποίηση και κρυπτογράφηση δεδομένων και διατηρεί όλα τα στοιχεία σχετικά με τα δικαιώματα πρόσβασης στα δεδομένα

Στο Σχήμα 32 παρουσιάζεται η αρχιτεκτονική της ΠΣΧ και του ΔΠΠΣΧ. Όσον αφορά περαιτέρω λεπτομέρειες τόσο σε επίπεδο εσωτερικών λειτουργιών όσο και υλοποίησης των ΠΣΧ και του υποσυστήματος ΔΠΠΣΧ, δεν θα επεκταθούμε στο πλαίσιο της συγκεκριμένη διατριβής. Οι εσωτερικές διαδικασίες θεωρούνται πλήρως διαφανείς για το υπόλοιπο σύστημα και η μελέτη των συγκεκριμένων οντοτήτων περιορίζεται σε επίπεδο γεγονότων που λαμβάνουν και αποστέλλουν στο υπόλοιπο σύστημα τα οποία θα αναφερθούν σε επόμενες ενότητες.

3.2.5 Προσωπικός Βοηθός Χρήστη/ Υποσύστημα Διαχειριστή Εξωτερικής Διεπαφής

Η αλληλεπίδραση του χρήστη με τις υπηρεσίες γίνεται μέσω κατάλληλης γραφικής διεπαφής (Graphical User Interface, GUI) η οποία ονομάζεται Προσωπικός Βοηθός Χρήστη (User Personal Assistant, UPA) και υλοποιείται στο τερματικό του χρήστη. Ο Προσωπικός Βοηθός Χρήστη (ΠΒΧ) θεωρείται για το σύστημα εξωτερική εφαρμογή και αλληλεπιδρά με τον Διαμεσολαβητή Τερματικού μέσω κατάλληλου υποσυστήματος τύπου Διαχειριστή Εξωτερικής Διεπαφής που βρίσκεται υλοποιημένο στον Διαμεσολαβητή Τερματικού.

Οι λειτουργίες που παρέχει ο ΠΒΧ είναι οι εξής:

- *Εισαγωγή στο σύστημα:* Με την εισαγωγή της ΠΣΧ στο τερματικό εκκινείται ο ΠΒΧ και παρουσιάζεται στον χρήστη παράθυρο εισαγωγής όνομα χρήστη και κωδικού.

Με την εισαγωγή των κατάλληλων στοιχείων ενεργοποιείται η διαδικασία πιστοποίησης του χρήστη στο σύστημα η οποία παρουσιάζεται αναλυτικά σε επόμενη ενότητα.

- *Παρουσίαση των διαθέσιμων υπηρεσιών:* Μετά την ολοκλήρωση της διαδικασίας πιστοποίησης ο ΠΒΧ ανακτά μέσω του Διαχειριστή Εξωτερικής Διεπαφής από το Υποσύστημα Διαχείρισης Υπηρεσιών τις διαθέσιμες υπηρεσίες και ο χρήστης δύναται να επιλέξει και να χρησιμοποιήσει κάποια από αυτές.
- *Κλήση Υπηρεσιών:* Μέσω κατάλληλης γραφικής διεπαφής ο χρήστης επιλέγει την υπηρεσία την οποία θέλει να χρησιμοποιείται και εκκινείται διαδικασία αυτόματης εισόδου του χρήστη στην υπηρεσία και αυτόματη ενεργοποίηση της εφαρμογής (π.χ. φυλλομετρητής διαδικτύου, web browser).
- *Διαχείριση Προσωπικών Δεδομένων:* Ο ΠΒΧ παρέχει κατάλληλη διεπαφή προκειμένου ο χρήστης να μπορεί να διαχειριστεί τα προσωπικά του δεδομένα εντός της ΠΣΧ. Ο χρήστης δύναται να αλλάξει, ανανεώσει, εμπλουτίσει ή ακόμα και να διαγράψει τα δεδομένα του προφίλ του.

3.2.6 Απαιτούμενες Λειτουργίες - Υποσυστήματα

Για την επίτευξη του δεύτερου στόχου, που είναι η παροχή των απαραίτητων λειτουργιών συλλογής πληροφοριών από τις πηγές τους για την τρέχουσα κατάσταση του χρήστη, την επεξεργασία τους, την διαχείριση τους και τελικά την παροχή τους σε εξωτερικές οντότητες κατόπιν αιτήσεως, εισάγονται στο σύστημα τα παρακάτω υποσυστήματα.

3.2.6.1 Υποσύστημα Διαχείρισης Προφίλ Χρήστη (ΥΔΠΧ)

Το συγκεκριμένο υποσύστημα είναι υπεύθυνο για την ανάκτηση και διατήρηση δεδομένων σχετικών με το προφίλ του χρήστη. Απαιτείται η αλληλεπίδραση του με την Προσωπική Συσκευή Χρήστη για την ανάκτηση των δεδομένων προφίλ χρήστη κάθε φορά που ο χρήστης εισέρχεται στο σύστημα, αλλά και κάθε φορά που ο χρήστης πραγματοποιεί κάποια αλλαγή. Επίσης καλείται να παρέχει την πληροφορία αυτή σε υποσυστήματα τα οποία την χρειάζονται για την διεκπεραίωση άλλων λειτουργιών. Υποστάσεις αυτού του υποσυστήματος εμφανίζονται και στους Διαμεσολαβητές Τερματικού αλλά και στους Διαμεσολαβητές Δικτύου. Τα δύο υποσυστήματα είναι όμοια μεταξύ τους και εκτελούν τις ίδιες λειτουργίες. Η παρουσία του υποσυστήματος στον Διαμεσολαβητή Δικτύου δίνει την δυνατότητα στις δικτυακές οντότητες αλλά και στις υπηρεσίες να ανακτούν πληροφορία προφίλ χρήστη χωρίς άμεση αλληλεπίδραση με το τερματικό αποφεύγοντας κατά συνέπεια

την άσκοπη κατανάλωση των πόρων του τερματικού ή την υπερφόρτωση των καναλιών επικοινωνίας του. Επίσης αντιμετωπίζεται και το πρόβλημα του μοναδικού σημείου αστοχίας. Στην περίπτωση μη ύπαρξης Διαμεσολαβητή Δικτύου όλη η λειτουργικότητα θα ήταν συγκεντρωμένη στο τερματικό. Το ΥΔΠΧ του Διαμεσολαβητή Τερματικού είναι υπεύθυνο για την ανάκτησης της πληροφορίας Προφίλ Χρήστη από την Προσωπική Συσκευή Χρήστη και την αποστολή της στο δίδυμό του υποσύστημα στην πλευρά του δικτύου, καθώς και για την ενημέρωση του σε περιπτώσεις αλλαγών.

3.2.6.2 Υποσύστημα Διαχείρισης Δυνατοτήτων Τερματικών Συσκευών (ΥΔΔΤΣ)

Σε κάθε τερματικό που εγκαθίσταται Διαμεσολαβητής Τερματικού, το συγκεκριμένο υποσύστημα αναλαμβάνει να ανακτήσει και να αποθηκεύσει πληροφορία σχετική με τις δυνατότητες του τερματικού, όπως π.χ. πλατφόρμα υλικού και λογισμικού, ανάλυση οθόνης, λειτουργικό σύστημα, δυνατότητες συνδεσιμότητας κλπ. όπως αυτές ορίζονται στο κεφάλαιο 2.3.2.2, στο προφίλ τερματικού. Κάθε φορά που ο χρήστης συνδέει στο τερματικό την ΠΣΧ του, το συγκεκριμένο υποσύστημα ανανεώνει τα δεδομένα του βάσει της τρέχουσας κατάστασης του τερματικού, όπως επίσης και οποιαδήποτε άλλη στιγμή συμβεί κάποια αλλαγή. Εν συνεχεία, ενημερώνει και το δίδυμό του υποσύστημα στην πλευρά του δικτύου. Ο λόγος ύπαρξης του υποσυστήματος και στην πλευρά του τερματικού αλλά και στην πλευρά του δικτύου είναι ο ίδιος όπως και στην προαναφερθείσα περίπτωση του Υποσυστήματος Διαχείρισης Προφίλ Χρήστη.

3.2.6.3 Υποσύστημα Διαχείρισης Υπηρεσιών (ΥΔΥ)

Υποστάσεις του υποσυστήματος αυτού συναντάμε τόσο στον Διαμεσολαβητή Τερματικού όσο και στον Διαμεσολαβητή Δικτύου. Στον Διαμεσολαβητή Τερματικού το Υποσύστημα Διαχείρισης Υπηρεσιών διατηρεί λίστα με τις υπηρεσίες στις οποίες είναι εγγεγραμμένος ο χρήστης. Οι σχετικές με τις υπηρεσίες πληροφορίες που διατηρούνται στο συνιστούν το προφίλ υπηρεσίας του χρήστη (Κεφάλαιο 2.3.2.3), το οποίο περιέχει το όνομα και τον κωδικό πρόσβασης, το URI της υπηρεσίας, εξειδικευμένες προτιμήσεις του χρήστη που αφορούν την υπηρεσία καθώς και λίστα με τις απαιτήσεις της υπηρεσίας σε χαρακτηριστικά τερματικού και παραμέτρους δικτύου. Κάθε φορά που ο χρήστης εισέρχεται στο σύστημα το υποσύστημα ανακτά από την ΠΣΧ το προφίλ των υπηρεσιών και ανανεώνει τη σχετική λίστα. Η διαδικασία αυτή είναι απαραίτητη γιατί υπάρχει το ενδεχόμενο ο χρήστης να εγγράφηκε/διαγράφηκε σε/από κάποια υπηρεσία όντας συνδεδεμένος από κάποια άλλη

τερματική συσκευή. Μετά τη διαδικασία συγχρονισμού ανάμεσα στην ΠΣΧ και στο ΥΔΥ του Διαμεσολαβητή Τερματικού, ξεκινάει η αλληλεπίδραση με το αντίστοιχο υποσύστημα στην πλευρά του Δικτύου. Το ΥΔΥ στην πλευρά του δικτύου διατηρεί λίστα με όλες τις διαθέσιμες και ενεργές υπηρεσίες, τους εγγεγραμμένους χρήστες σε αυτές αλλά και τους τρέχοντες ενεργούς χρήστες. Κατά την αλληλεπίδραση τους το υποσύστημα του τερματικού ενημερώνει αυτό του δικτύου για την εισαγωγή του νέου χρήστη στο σύστημα και ενημερώνεται για το αν είναι ενεργές οι υπηρεσίες στις οποίες είναι εγγεγραμμένος ο χρήστης προκειμένου να του τις παρουσιάσει. Όταν ο χρήστης επιλέξει κάποια από τις υπηρεσίες, η υπηρεσία ζητάει πληροφορία σχετική με το χρήστη μέσω των Διαχειριστών Εξωτερικής Διεπαφής που έχει εγκατεστημένους στους Διαμεσολαβητές Δικτύου. Οι ΔΕΔ ελέγχουν από το ΥΔΥ αν ο χρήστης για τον οποίο ζητήθηκε η πληροφορία είναι δεμένος στον συγκεκριμένο Διαμεσολαβητή Δικτύου. Μόνο ένας ΔΕΔ θα λάβει θετική απάντηση και τελικά αυτός θα προχωρήσει στην διαδικασία ανάκτησης των πληροφοριών και στην επιστροφή τους στην υπηρεσία.

3.2.6.4 Υποσύστημα Διαχείρισης Δικτυακών Συνδέσεων (ΥΔΔΣ)

Το υποσύστημα Διαχείρισης Δικτυακών Συνδέσεων είναι υπεύθυνο για την ανάκτηση και διατήρηση των σχετικών με το προφίλ Δικτύου Υπηρεσιών πληροφοριών. Ουσιαστικά ανακτά όλη την σχετική πληροφορία για τις δικτυακές συνδέσεις του χρήστη και την δομεί βάσει του προφίλ δικτύου όπως αυτό περιγράφεται στο κεφάλαιο 2.3.2.4. Επίσης, το υποσύστημα αυτό είναι υπεύθυνο για την παροχή των πληροφοριών της τρέχουσας ενεργής δικτυακής σύνδεσης του χρήστη και των χαρακτηριστικών της σε οποιαδήποτε οντότητα τις ζητήσει. Δίδυμο αντίγραφο του υποσυστήματος που υπηρετεί τον ίδιο ακριβώς σκοπό βρίσκεται και στον Διαμεσολαβητή Δικτύου.

3.2.6.5 Υποσύστημα Διαχείρισης Πληροφορίας Θέσης Χρήστη (ΥΔΠΘΧ)

Το υποσύστημα αυτό είναι υπεύθυνο για την ανάκτηση και την αποθήκευση πληροφορίας για την θέση του χρήστη. Για την ανάκτηση της πληροφορίας αυτής το υποσύστημα υποχρεούται να αλληλεπιδράσει με εξωτερική του συστήματος οντότητα. Συνεπώς το σύστημα αυτό παρέχει και λειτουργίες Διαχειριστή Εξωτερικής Διεπαφής. Ανάλογα με τη μέθοδο ανάκτησης της πληροφορίας θέσης είναι δυνατόν να εμφανίζονται Υποσυστήματα Διαχείρισης Πληροφορίας Θέσης Χρήστη και στους δυο Διαμεσολαβητές. Στην περίπτωση χρήσης μεθόδου ανάκτησης πληροφορίας βασισμένης ή υποβοηθούμενης από το δίκτυο το υποσύστημα τοποθετείται στον Διαμεσολαβητή Δικτύου και δεν υπάρχει ανάγκη ύπαρξης

του αντίστοιχου συστήματος στην πλευρά του τερματικού. Στην αντίθετη περίπτωση (η περίπτωση τερματικού GPS για παράδειγμα), το υποσύστημα βρίσκεται στον Διαμεσολαβητή Τερματικού και ενημερώνει αυτό του Διαμεσολαβητή Δικτύου. Στην περίπτωση που χρησιμοποιούνται περισσότερες της μίας μέθοδοι και συνεπώς και τα δύο υποσυστήματα ανακτούν πληροφορία από διαφορετικές πηγές απαιτείται συγχρονισμός και διύλιση της πληροφορίας προκειμένου τελικά να χρησιμοποιείται σαν θέση του χρήστη το πιο ακριβές και αξιόπιστο αποτέλεσμα ή αυτό που κάθε φορά εξυπηρετεί καλύτερα την υπηρεσία που χρησιμοποιεί ο χρήστης.

3.2.6.6 Υποσύστημα Διαχείρισης Πληροφορίας Πλαισίου (ΥΔΠΠ)

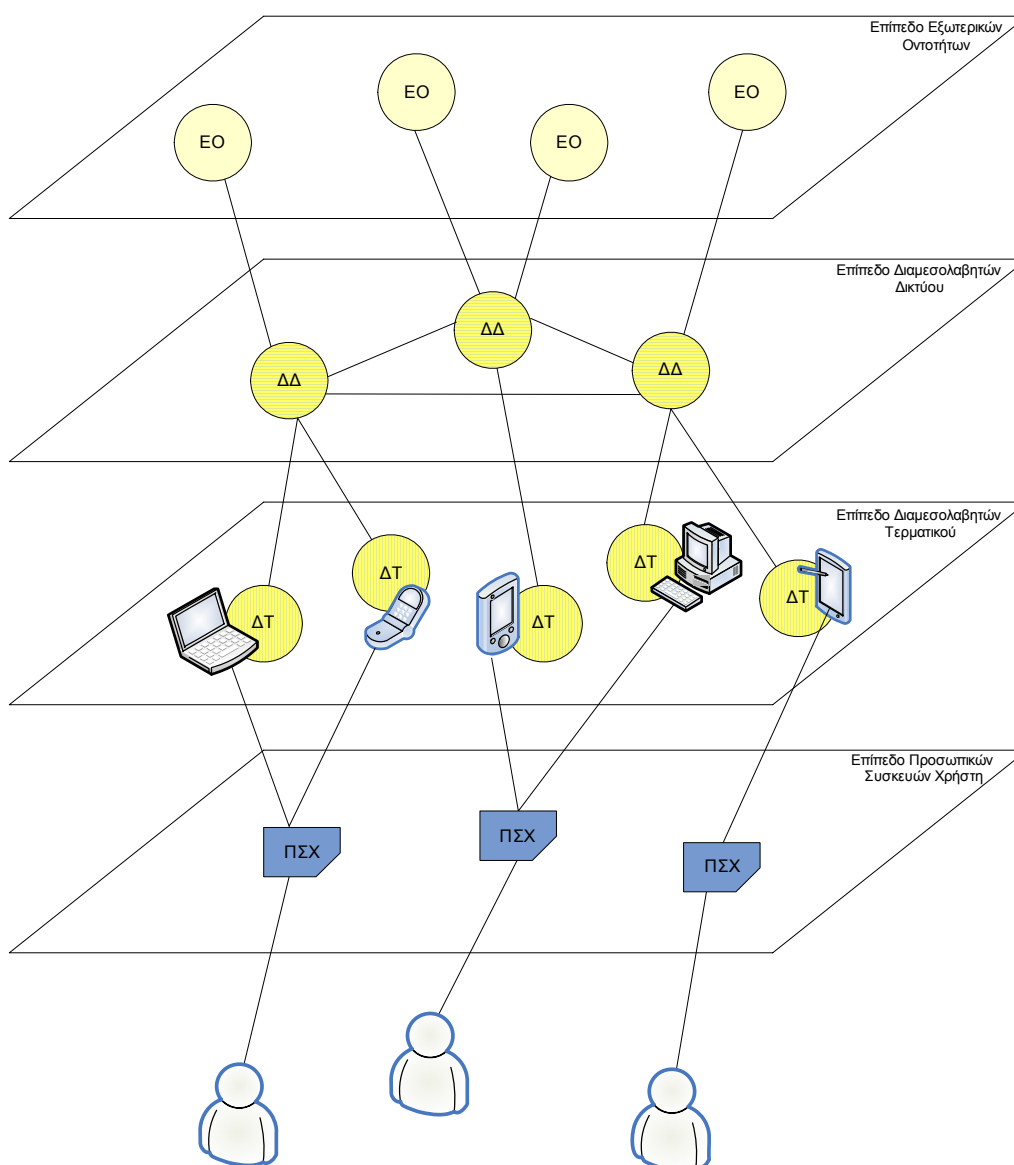
Το υποσύστημα Διαχείρισης Συνολικής Πληροφορίας Πλαισίου είναι ίσως η πιο νευραλγική συνιστώσα του συστήματος μας. Το υποσύστημα αυτό εμφανίζεται μόνο στην πλευρά του Δικτύου και η βασική του λειτουργία είναι να συλλέγει τα επιμέρους κομμάτια της πληροφορίας και να τα συνθέτει στην πρότυπη μορφή του CoP όπως περιγράφεται στο Κεφάλαιο 2.3.2.5. Συνεπώς το υποσύστημα αυτό αλληλεπιδρά με όλα τα προαναφερθέντα υποσυστήματα για την ανάκτηση της αντίστοιχης πληροφορίας από το καθένα, εγγράφεται για να λαμβάνει ειδοποιήσεις για τυχόν αλλαγές και κατόπιν αυτών αναλαμβάνει να ενημερώνει την σχετική πληροφορία προκειμένου να αντικατοπτρίζεται πλήρως η τρέχουσα κατάσταση του χρήστη. Με την οντότητα αυτή αλληλεπιδρούν ουσιαστικά και όλοι οι Διαχειριστές Εξωτερικής Διεπαφής των υπηρεσιών οι οποίες συνεργάζονται με το σύστημα. Όταν μία υπηρεσία μέσω του Διαχειριστή της ζητήσει την πληροφορία κατάστασης χρήστη ή μέρος αυτής το αίτημα της προωθείται στο εν λόγω υποσύστημα. Το ΥΔΠΠ αναλαμβάνει να απομονώσει την πληροφορία που απαιτείται από την υπηρεσία και να την προωθήσει στην κατάλληλη μορφή στον Διαχειριστή εξωτερικής Διεπαφής και κατά συνέπεια στην υπηρεσία για την ολοκλήρωση της εκτέλεσης της.

3.2.7 Η Συνολική Αρχιτεκτονική

Μία πρώτη γενική διαστρωματωμένη αναπαράσταση της αρχιτεκτονικής βασισμένη στην ιδέα των διαμεσολαβητών παρουσιάζεται Σχήμα 33. Στο επίπεδο του δικτύου βρίσκονται οι Διαμεσολαβητές Δικτύου, ενώ στο επίπεδο των τερματικών τοποθετούνται οι Διαμεσολαβητές Τερματικού. Οι αρχές λειτουργίας των διαφορετικών διαμεσολαβητών είναι ακριβώς ίδιες, οι διαφορές τους έγκεινται στις διαφορετικές λειτουργίες που παρέχουν, δηλαδή στα διαφορετικά υποσυστήματα που φιλοξενούν καθώς και στην φυσική τους θέση μέσα στο δίκτυο. Οι Διαμεσολαβητές Τερματικών βρίσκονται εγκατεστημένοι με τη μορφή

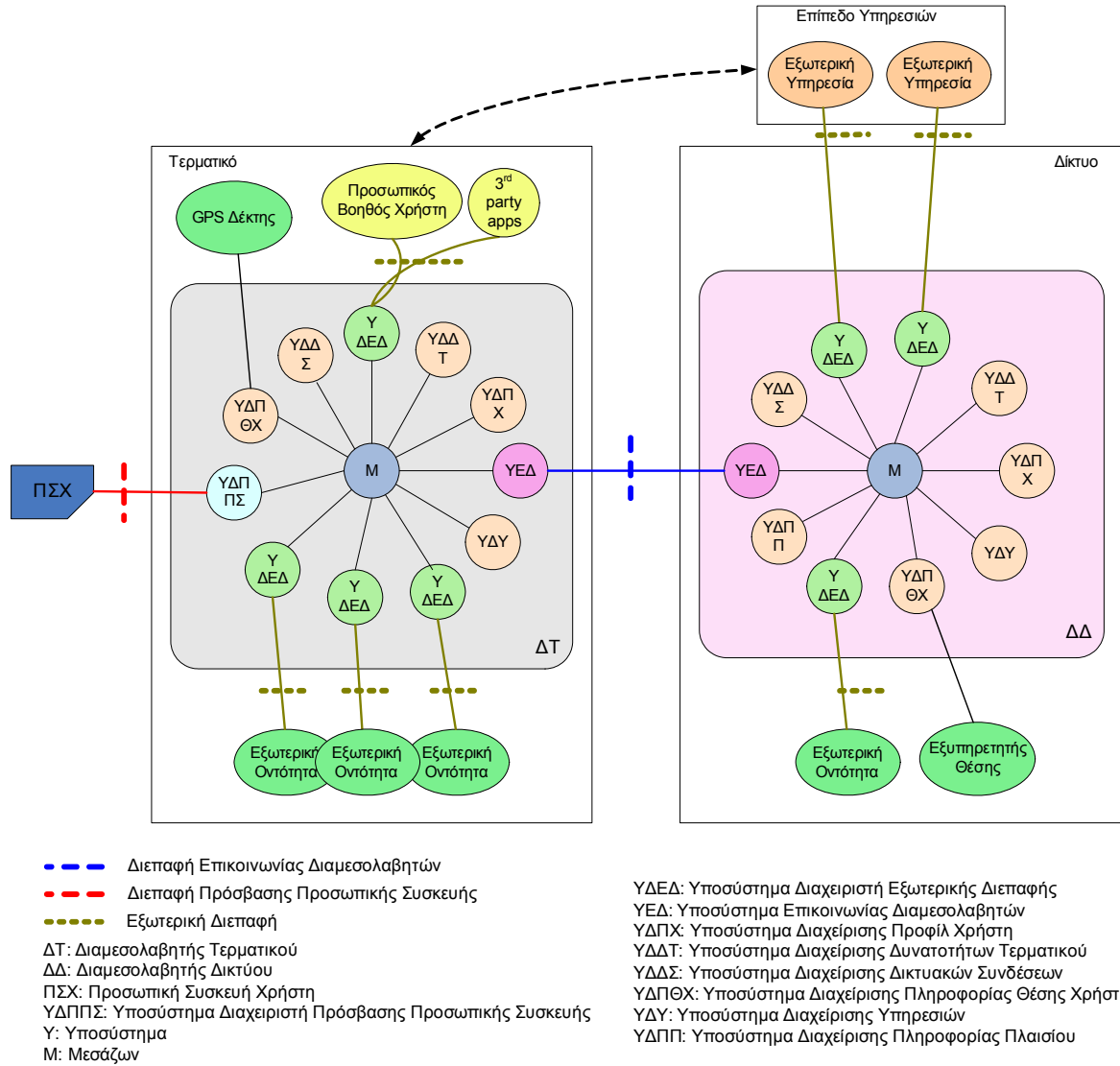
κατάλληλου λογισμικού στα τερματικά των χρηστών, ενώ οι Διαμεσολαβητές Δικτύου αποτελούν κόμβους εγκατεστημένους στο δίκτυο που περιέχουν τις αντίστοιχες λειτουργικές οντότητες.

Το ακριβώς υποκείμενο επίπεδο περιέχει τις Προσωπικές Συσκευές Χρηστών που αποτελούν το δεύτερο δομικό στοιχείο του συστήματος.



Σχήμα 33: Διαστρωμάτωση της Αρχιτεκτονικής

Στο Σχήμα 34 παρατίθεται η συνολική αρχιτεκτονική του συστήματος με όλες τις εμπλεκόμενες οντότητες και τις μεταξύ τους διεπαφές, όπως έχουν περιγραφεί στις προηγούμενες ενότητες [4].

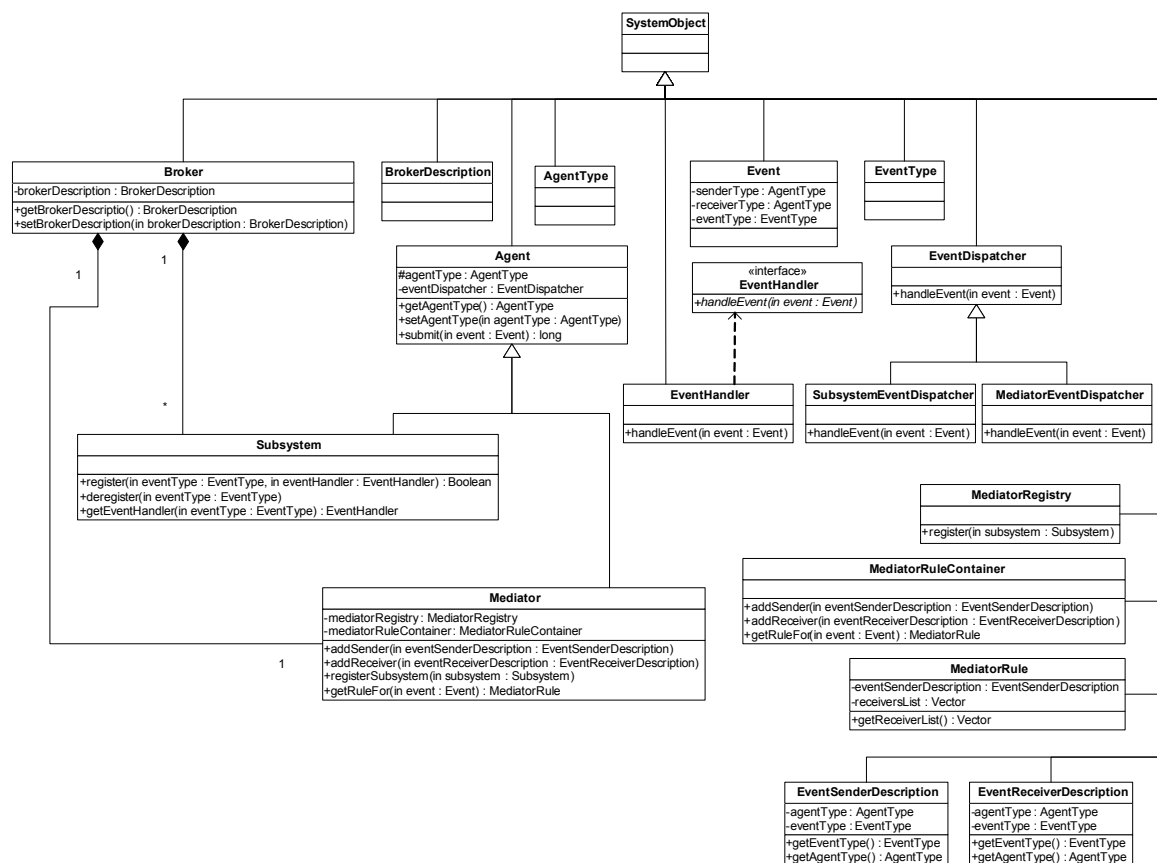


Σχήμα 34: Η προτεινόμενη αρχιτεκτονική: οντότητες και διεπαφές

3.3 Η Υλοποίηση του συστήματος

3.3.1 Η οντότητα του Διαμεσολαβητή: Μεσάζων & Υποσύστημα

Στο Σχήμα 35 παρουσιάζονται αναλυτικά όλες οι κλάσεις που απαιτούνται για την υλοποίηση του Διαμεσολαβητή και του εσωτερικού μηχανισμού επικοινωνίας μεταξύ των υποσυστημάτων.



Σχήμα 35: Διάγραμμα Κλάσεων του Διαμεσολαβητή

Η κλάση *Broker* εκκινεί και διαμορφώνει τον Διαμεσολαβητή. Δίνει υπόσταση στον Μεσάζοντα και στα Υποσυστήματα μέσω των κλάσεων *Mediator* και *Subsystem*. Κάθε υποσύστημα εγγράφεται στον μεσάζοντα και εγγράφει και όλα τα γεγονότα που αποστέλλει και λαμβάνει με τη μορφή αντικειμένων *EventSenderDescription* και *EventReceiverDescription*. Ο μεσάζοντας αποθηκεύει την πληροφορία αυτή σε ένα αντικείμενο της κλάσης *MediatorRuleContainer*. Αμέσως μετά ο *Broker* ενεργοποιεί τον Μεσάζοντα και είναι δυνατή η ανταλλαγή γεγονότων ανάμεσα στα υποσυστήματα. Για την λήψη γεγονότων κάθε υποσύστημα πρέπει να υλοποιεί συγκεκριμένους χειριστές γεγονότων μέσω της κλάσης *EventHandler*, οι οποίοι επικαλούνται με την λήψη από το υποσύστημα του γεγονότος για το οποίο είναι υπεύθυνοι. Για κάθε γεγονός ο Μεσάζοντας πρέπει να αποφασίζει σε ποιο υποσύστημα θα το παραδώσει. Αυτή η απόφαση καθορίζεται από τον τύπο του γεγονότος (*EventType*) και από τον τύπο του αποστολέα του γεγονότος (*AgentType*) βάσει της πληροφορίας που είναι αποθηκευμένη στο *MediatorRuleContainer*. Έτσι δημιουργείται κατάλληλος κανόνας, *MediatorRule*, ο οποίος υποδεικνύει στον Μεσάζοντα τους παραλήπτες για το συγκεκριμένο γεγονός.

Στη συνέχεια παρατίθεται σύντομη περιγραφή όλων των κλάσεων που παρουσιάζονται στο Σχήμα 35.

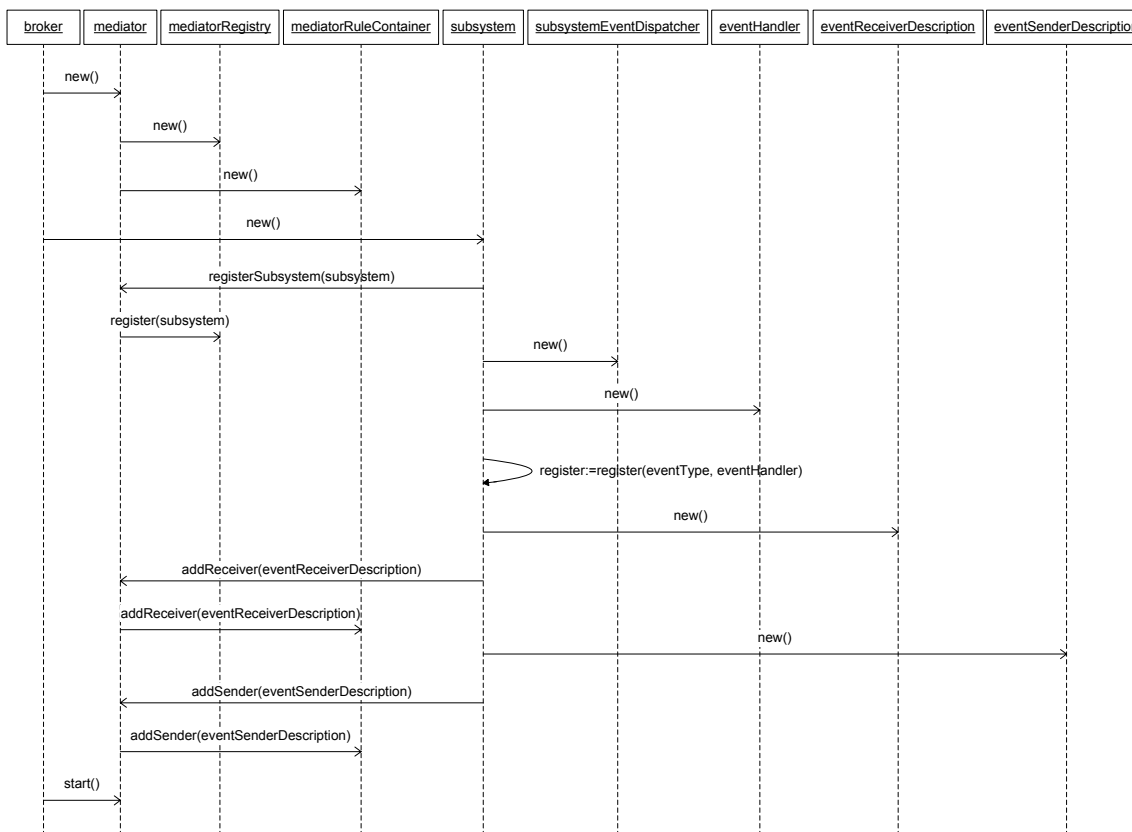
- *Broker*: Η κλάση *Broker* είναι υπεύθυνη για την εκκίνηση του συστήματος. Δημιουργεί όλα τα αντικείμενα των υποσυστημάτων και το αντικείμενο του μεσάζοντα και εν συνεχεία ενεργοποιεί τον τελευταίο προκειμένου να μπορούν να πραγματοποιηθούν ανταλλαγές γεγονότων μεταξύ των υποσυστημάτων.
- *BrokerDescription*: Παρέχει μεθόδους για την ανάκτηση του ρόλου του διαμεσολαβητή και της ταυτότητας του μέσω μοναδικού identifier.
- *Agent*: Η κλάση *Agent* αποτελεί την βασική κλάση (base class) από την οποία κληρονομούν οι κλάσεις *Mediator* και *Subsystem*. Υλοποιεί το Active Object Pattern, μέσω του οποίου αποσυσχετίζεται η κλήση της μεθόδου από την εκτέλεση της.
- *AgentType*: Στην κλάση *AgentType* ορίζονται και δημιουργούνται όλοι οι τύποι *Agent* που υπάρχουν στο σύστημα.
- *Event*: Η κλάση *Event* παρέχει το μέσο επικοινωνίας μεταξύ των υποσυστημάτων. Ένα αντικείμενο της κλάσης *Event* μπορεί να περιέχει πληροφορία για το υποσύστημα αποστολέα (*senderType*), το υποσύστημα παραλήπτη (*receiverType*) και τον τύπο του γεγονότος (*eventType*)
- *EventType*: Στην κλάση *EventType* ορίζονται και δημιουργούνται όλοι οι τύποι *Event* που χρησιμοποιούνται στο σύστημα.
- *EventHandler*: Η κλάση *EventHandler* υλοποιεί την διεπαφή *EventHandler*. Παρέχει τη μέθοδο *handleEvent(event)* για τον χειρισμό των γεγονότων από τα υποσυστήματα. Κάθε υποσύστημα υλοποιεί τόσα αντικείμενα *EventHandler* όσα και τα μηνύματα για τα οποία έχει εγγραφεί στον μεσάζοντα ότι μπορεί να λαμβάνει.
- *EventReceiverDescription*: Η κλάση αυτή υλοποιεί αντικείμενα τα οποία ουσιαστικά αποτελούν δεμένα ζεύγη αντικειμένων *EventType* και *AgentType*. Η σχέση των αντικειμένων αυτών είναι η εξής: το υποσύστημα του συγκεκριμένου *AgentType* λαμβάνει γεγονότα του συγκεκριμένου *EventType*.
- *EventSenderDescription*: Η κλάση αυτή υλοποιεί αντικείμενα τα οποία ουσιαστικά αποτελούν δεμένα ζεύγη αντικειμένων *EventType* και *AgentType*. Η σχέση των αντικειμένων αυτών είναι η εξής: το υποσύστημα του συγκεκριμένου *AgentType* αποστέλλει γεγονότα του συγκεκριμένου *EventType*.
- *EventDispatcher*: Αποτελεί την κλάση από την οποία κληρονομούν οι κλάσεις *MediatorEventDispatcher* και *SubsystemEventDispatcher*. Αναλαμβάνει να

παραλαμβάνει γεγονότα και να τα παραδίδει στα κατάλληλα αντικείμενα με χρήση της μεθόδου *handleEvent(event)*.

- *MediatorEventDispatcher*: Αποτελεί υποκλάση της *EventDispatcher*. Υλοποιεί την μέθοδο *handleEvent(event)*. Παραλαμβάνει τα γεγονότα που έχουν παραδοθεί στον Μεσάζοντα και βάσει κανόνων που ανακτά με την μέθοδο *getRule(event)* του *Mediator* τα παραδίδει στη συνέχεια στο κατάλληλο υποσύστημα.
- *SubsystemEventDispatcher*: Αποτελεί υποκλάση της *EventDispatcher*. Υλοποιεί την μέθοδο *handleEvent(event)*. Παραλαμβάνει τα γεγονότα που έχουν παραδοθεί στο υποσύστημα και τα παραδίδει στη συνέχεια στον κατάλληλο *EventHandler*.
- *Mediator*: Η κλάση *Mediator* αποτελεί έναν έξυπνο μεταγωγέα γεγονότων. Μέσω της μεθόδου της *registerSubsystem(subsystem)* τα υποσυστήματα εγγράφονται στον Μεσάζοντα και εγγράφουν και τα αντικείμενα *EventSenderDescription* και *EventReceiverDescription* με κλήση των μεθόδων *addSender(EventSenderDescription)* και *addReceiver(EventReceiverDescription)* γνωστοποιώντας έτσι στον μεσάζοντα τους τύπους των γεγονότων τα οποία αποστέλλουν και τους τύπους των γεγονότων που λαμβάνουν.
- *Subsystem*: Αποτελεί την κλάση από την οποία δημιουργούνται όλα τα αντικείμενα που αντιπροσωπεύουν τα υποσυστήματα του Διαμεσολαβητή. Με την μέθοδο *register(EventType, EventHandler)* εγγράφει σε λίστα που διατηρεί τα αντικείμενα *EventHandler* για το χειρισμό των *EventType* που λαμβάνει.
- *MediatorRegistry*: Κάθε αντικείμενο *Mediator* είναι δεμένο με ένα αντικείμενο *MediatorRegistry*. Το αντικείμενο *MediatorRegistry* διατηρεί λίστα με όλα τα εγγεγραμμένα στον Μεσάζοντα υποσυστήματα. Ο *Mediator* μέσω της μεθόδου *registerSubsystem(subsystem)* της κλάσης *MediatorRegistry* εγγράφει τα υποσυστήματα στην *MediatorRegistry*.
- *MediatorRuleContainer*: Διατηρεί λίστες με όλα τα αντικείμενα *EventSenderDescription* και *EventReceiverDescription*. Ο *Mediator* προσθέτει τα αντικείμενα αυτά στο *MediatorRuleContainer* με κλήση των μεθόδων *addSender(EventSenderDescription)* και *addReceiver(EventReceiverDescription)* αντίστοιχα. Η κλήση της μεθόδου *getRule(event)* αυτής της κλάσης από τον Μεσάζοντα δημιουργεί ένα νέο αντικείμενο *MediatorRule* για κάθε υπό αποστολή γεγονός.

- *MediatorRule*: Με κλήση της μεθόδου *getReceiverList()* ο *MediatorEventDispatcher* ανακτά από το αντικείμενο *MediatorRule* που έχει δημιουργηθεί για το υπό αποστολή γεγονός τη λίστα με τα υποσυστήματα στα οποία πρέπει να το προωθήσει.

3.3.1.1 Δημιουργία και εγγραφή υποσυστήματος στον Μεσάζοντα



Σχήμα 36: Διάγραμμα αλληλεπιδράσεων δημιουργίας και εγγραφής υποσυστήματος στον Μεσάζοντα

Στο παραπάνω διάγραμμα αλληλεπιδράσεων (sequence diagram) (Σχήμα 36) παρουσιάζονται οι διαδοχικές ενέργειες που πραγματοποιούνται για την δημιουργία και εγγραφή ενός υποσυστήματος στον Μεσάζοντα.

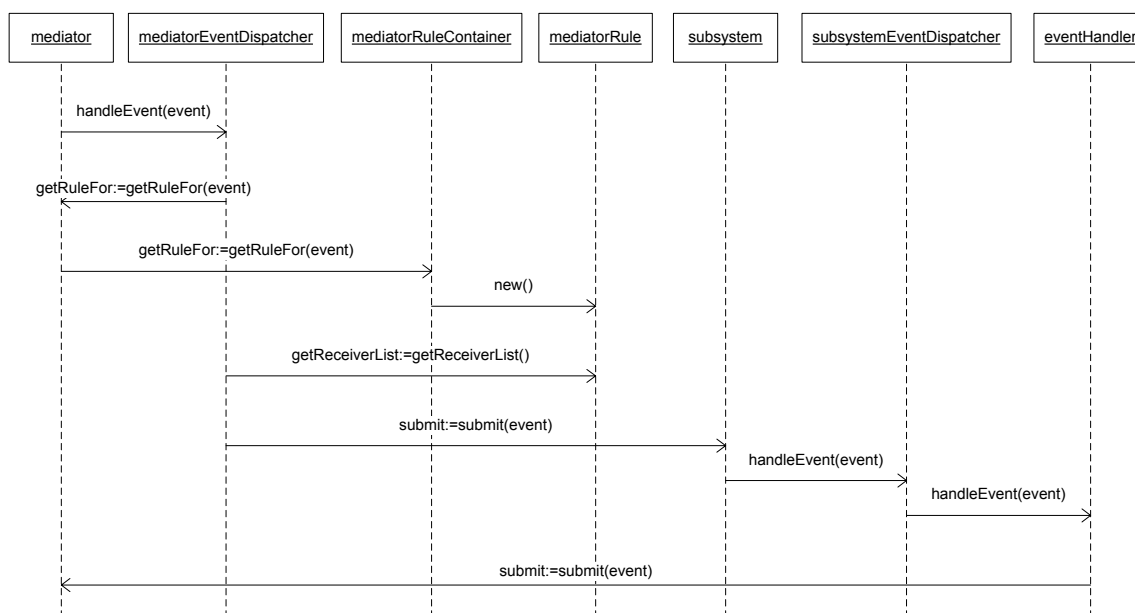
Η δημιουργία και εγγραφή των υποσυστημάτων πραγματοποιείται κατά την εκκίνηση του συστήματος. Υπεύθυνος για την διαδικασία αυτή είναι ο Διαμεσολαβητής. Κατά την εκκίνηση δημιουργείται το αντικείμενο *broker*, το οποίο εν συνεχεία δημιουργεί νέες υποστάσεις των κλάσεων *Mediator*, *MediatorRegistry* και *MediatorRuleContainer*. Επίσης ο *broker* δημιουργεί και όλα τα αντικείμενα *subsystem*. Κάθε *subsystem* εγγράφεται στον *mediator* με κλήση της μεθόδου *registerSubsystem(subsystem)* του *mediator*, ο οποίος προωθεί την εγγραφή στην *mediatorRegistry* με κλήση της μεθόδου της *register(subsystem)*.

Εν συνεχεία κάθε υπόσταση υποσυστήματος πραγματοποιεί της παρακάτω ενέργειες:

- Δημιουργία του αντικειμένου *subsystemEventDispatcher*
 - Δημιουργία των αντικειμένων *eventHandler* για την διαχείριση των γεγονότων που δύναται να λαμβάνει το υποσύστημα και εγγραφή τους εσωτερικά στο υποσύστημα (*register(eventType, eventHandler)*).
 - Δημιουργία των αντικειμένων *eventReceiverDescription* για κάθε γεγονός για το οποίο το υποσύστημα δρα ως παραλήπτης και εγγραφή τους στο *mediatorRule Container* με κλήση της μεθόδου *addReceiver(eventReceiverDescription)* αρχικά από το *subsystem* στον *mediator* και στη συνέχεια από τον *mediator* στο *mediatorRuleContainer*.
 - Δημιουργία των αντικειμένων *eventSenderDescription* για κάθε γεγονός για το οποίο το υποσύστημα δρα ως αποστολέας και εγγραφή τους στο *mediatorRuleContainer* με κλήση της μεθόδου *addSenderDescription (eventSenderDescription)* αρχικά από το *subsystem* στον *mediator* και στη συνέχεια από τον *mediator* στο *mediatorRuleContainer*
- Τέλος ο broker εκκινεί τον mediator (*start()*) προκειμένου να είναι δυνατή η επικοινωνία ανάμεσα στα υποσυστήματα.

3.3.1.2 Προώθηση γεγονότος μέσω του Μεσάζοντα

Στο παρακάτω διάγραμμα αλληλεπιδράσεων (sequence diagram) (Σχήμα 37) παρουσιάζονται οι διαδοχικές ενέργειες που πραγματοποιούνται για την παράδοση ενός γεγονότος από τον Μεσάζοντα στο κατάλληλο υποσύστημα.



Σχήμα 37: Διάγραμμα αλληλεπιδράσεων προώθησης γεγονότος μέσω του Μεσάζοντα

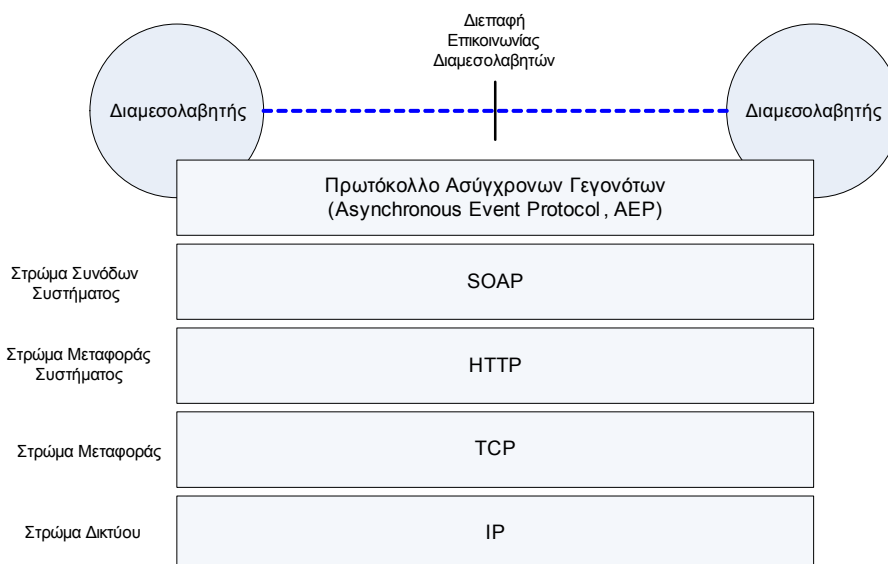
Αρχικά θεωρούμε ότι ο Μεσάζων έχει παραλάβει ένα γεγονός από ένα υποσύστημα. Το γεγονός αυτό δεν φέρει στοιχεία για τον παραλήπτη του. Συνεπώς πρέπει ο Μεσάζων να πραγματοποιήσει όλες τις απαραίτητες ενέργειες προκειμένου να συμπεράνει σε ποιο υποσύστημα πρέπει να παραδώσει το γεγονός. Αρχικά παραδίδει το γεγονός στον *mediatorEventDispatcher* του, μέσω κλήσης της μεθόδου *handleEvent(event)*. Ο *mediatorEventDispatcher* εν συνεχεία ζητά από τον *mediator* τον κανόνα που πρέπει να εφαρμόσει για την παράδοση του συγκεκριμένου γεγονότος καλώντας τη μέθοδο *getRuleFor(event)*. Ο *mediator* καλεί την αντίστοιχη μέθοδο στο αντικείμενο *mediatorRuleContainer*, το οποίο δημιουργεί ένα νέο αντικείμενο *mediatorRule* για το συγκεκριμένο γεγονός. Το αντικείμενο *mediatorRule* διατηρεί λίστα με όλους του υποψήφιους παραλήπτες. Με κλήση της μεθόδου *getReceiverList()* του *mediatorRule*, ο *mediatorEventDispatcher* ανακτά τα υποσυστήματα στα οποία πρέπει να προωθήσει το γεγονός. Καλώντας τη μέθοδο *submit(event)* σε κάθε ένα από αυτά τα υποσυστήματα ο *mediatorEventDispatcher* τους παραδίδει το γεγονός. Το *subsystem* εν συνεχεία καλεί τη μέθοδο *eventHandle(event)* στο αντικείμενο *subsystemEventDispatcher*, το οποίο αναλαμβάνει να καλέσει την μέθοδο *eventHandle(event)* του κατάλληλου *eventHandler*. Τέλος, ο τελευταίος πραγματοποιεί όλες τις απαραίτητες ενέργειες και αποστέλλει μέσω της μεθόδου *submit(event)* το γεγονός-απάντηση στον *mediator* ο οποίος προωθεί το γεγονός στο υποσύστημα αποστολέα.

3.3.2 Υποσύστημα Επικοινωνίας Διαμεσολαβητών

3.3.2.1 Πρωτόκολλο Επικοινωνίας Διαμεσολαβητών

Η αλληλεπίδραση ανάμεσα σε δυο υποσυστήματα ΥΕΔ πραγματοποιείται μέσω απευθείας σύνδεσης μεταξύ τους και η αλληλεπίδραση τους βασίζεται σε ασύγχρονη μονόδρομη ανταλλαγή μηνυμάτων (*asynchronous one-way message-based interaction*). Το πρωτόκολλο που καθορίζει την διεπαφή ανάμεσα στα ΥΕΔ ονομάζεται Πρωτόκολλο Ασύγχρονων Γεγονότων (*Asynchronous Event Protocol, AEP*) [8][9]. Το AEP καθορίζει έναν αριθμό από μηνύματα που ανταλλάσσουν οι ΥΕΔ με σκοπό να στείλουν απομακρυσμένα γεγονότα ή να υλοποιήσουν τις διαδικασίες ανακάλυψης και εγκατάστασης απομακρυσμένων στόχων για τα γεγονότα που δεν μπορούν να χειριστούν τοπικά. Το πρωτόκολλο που χρησιμοποιείται για την επικοινωνία τους και την υλοποίηση των μηνυμάτων του AEP είναι το SOAP, ενώ στο επίπεδο μεταφοράς (*transport layer*) χρησιμοποιείται το HTTP. Η σύνταξη των μηνυμάτων βασίζεται στην XML και αποτελούν μηνύματα SOAP στα οποία το σώμα του

μηνύματος περιέχει τη σειριοποιημένη μορφή των γεγονότων που χειρίζεται ο ΥΕΔ. Η επιλογή του SOAP πραγματοποιήθηκε διότι είναι ένα πρωτόκολλο βασισμένο στην XML παρέχοντας έτσι διαλειτουργικότητα ανάμεσα σε διαφορετικές πλατφόρμες. Επίσης παρέχονται υλοποιήσεις του όπως το kSOAP [11] για J2ME, για τερματικά δηλαδή με μειωμένες δυνατότητες (resource constrained)

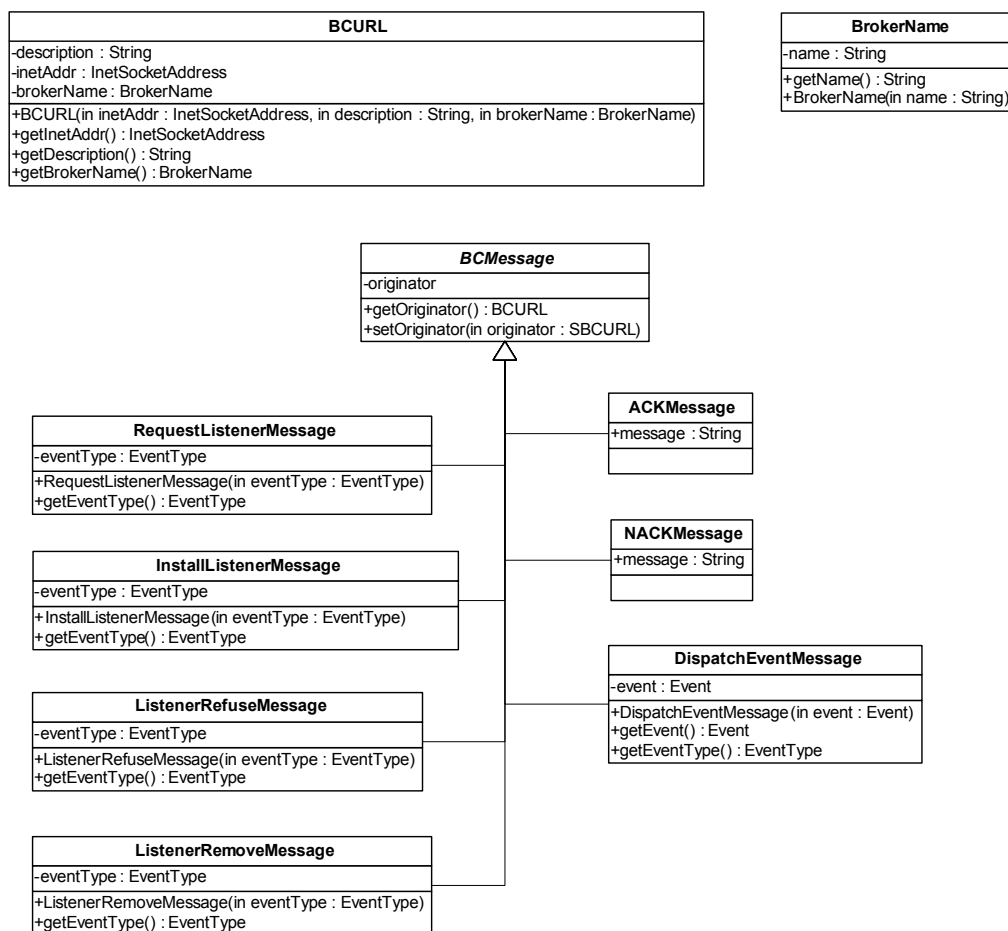


Σχήμα 38: Στοιβά Πρωτοκόλλων της Διεπαφής ΥΕΔ

Και στις δύο περιπτώσεις αλληλεπίδρασης των ΥΕΔ, και κατά την διάρκεια των φάσεων ανεύρεσης και εγκατάστασης των επικοινωνιακών συνόδων αλλά και κατά τη διάρκεια ανταλλαγής γεγονότων, η επικοινωνία πραγματοποιείται με την ανταλλαγή μηνυμάτων υλοποιημένων σαν μηνύματα SOAP. Στην πρώτη περίπτωση τα μηνύματα SOAP που ανταλλάσσονται ενθυλακώνουν ερωτήσεις για την παρουσία υποσυστημάτων που δύνανται να εξυπηρετήσουν συγκεκριμένα γεγονότα. Αν το μήνυμα-απάντηση είναι θετικό πραγματοποιείται η εγκατάσταση της συνόδου, διαφορετικά το ΥΕΔ που πραγματοποιεί την διαδικασία ανεύρεσης αποστέλλει νέα μηνύματα αναζήτησης SOAP προς άλλους διαμεσολαβητές. Κατά την μεταφορά γεγονότων ανταλλάσσονται και πάλι μηνύματα SOAP τα οποία περιέχουν τα γεγονότα σε σειριοποιημένη μορφή βασισμένη στους κανόνες κωδικοποίησης που παρέχει το SOAP και σε αυτούς που παρέχει η υλοποίηση της μηχανής SOAP (SOAP Engine) που χρησιμοποιείται. Στο Σχήμα 38 παρουσιάζεται η στοιβά πρωτοκόλλων της διεπαφής των ΥΕΔ.

3.3.2.2 Υλοποίηση

Στη συνέχεια παρουσιάζονται οι σημαντικότερες κλάσεις και μέθοδοι της υλοποίησης του Υποσυστήματος Επικοινωνίας Διαμεσολαβητών.



Σχήμα 39: Διάγραμμα Κλάσεων των μηνυμάτων του AEP

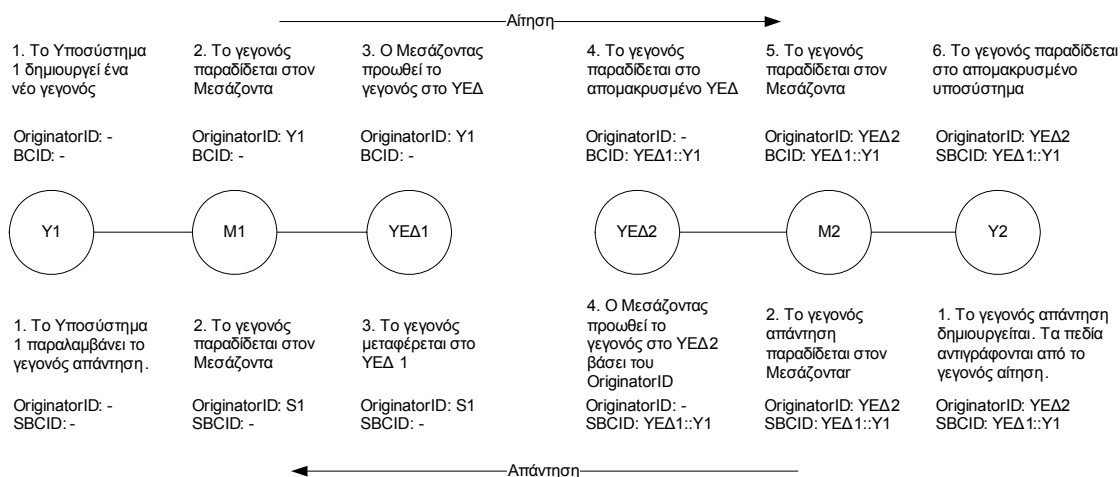
Το YED υλοποιείται με την κλάση *BCSubsystem* η οποία αποτελεί υποκλάση της κλάσης *Subsystem*. Η κλάση *BCSubsystem* είναι υπεύθυνη για την δημιουργία των αντικείμενων των υπολοίπων κλάσεων του YED και για την διαχείριση της μηχανής SOAP (SOAP engine). Επίσης το YED υλοποιεί κατάλληλο *EventHandler*, τον *BCEventHandler*, ο οποίος παραλαμβάνει τα γεγονότα από τον *Mediator* και τα παραδίδει στην κατάλληλη κλάση. Εν συνεχεία υλοποιούνται μία σειρά από διαχειριστές, οι *ListenerManager*, *DispatchManager*, *DiscoveryManager* οι οποίοι παρέχουν μεθόδους *deliver()* για την διαχείριση μηνυμάτων YED προερχόμενων από απομακρυσμένους διαμεσολαβητές και συγκεκριμένες μεθόδους για την διαχείριση των γεγονότων που προέρχονται εσωτερικά από τον διαμεσολαβητή. Η κλάση *BCReceiver* υλοποιεί την διεπαφή ανάμεσα στο YED και την μηχανή SOAP (SOAP engine). Παραλαμβάνει αντικείμενα της κλάσης *BCMessage* και τα παραδίδει στον κατάλληλο διαχειριστή. Η διεπαφή *BCMessageCall* παρέχει μεθόδους στις κλάσεις των διαχειριστών για τη μετατροπή των *BCMessage* σε μηνύματα SOAP.

Η κλάση *BCMessage* αποτελεί την κλάση γονέα για τα μηνύματα που υποστηρίζει το πρωτόκολλο επικοινωνίας των ΥΕΔ (Asynchronous Event Protocol, AEP). Οι κλάσεις που υλοποιούν αυτά τα μηνύματα παρατίθενται στη συνέχεια:

- *ACKMessage/NACKMessage*
- *ListenerRequestMessage/ListenerInstallMessage/ListenerRefuseMessage/ListenerRemoveMessage*
- *BrokerListRequestMessage/BrokerListMessage*
- *BrokerRegisterMessage/BrokerRegisterResponseMessage*.

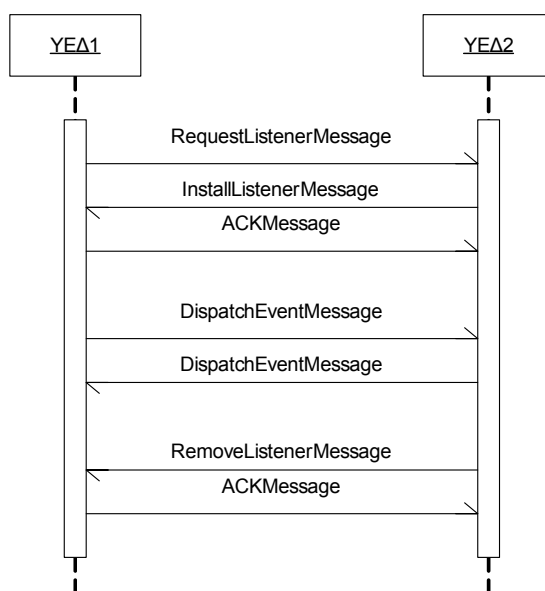
Στο παραπάνω διάγραμμα κλάσεων (Σχήμα 39) παρουσιάζονται τα μηνύματα του πρωτοκόλλου επικοινωνίας των ΥΕΔ, καθώς και κάποιες άμεσα σχετιζόμενες κλάσεις όπως οι *BCURL* και *BrokerName*. Οι κλάσεις *BCURL*, *OriginatorID* και *BrokerName* αποτελούν βασικές κλάσεις του υποσυστήματος και παρέχουν ουσιαστικά την αναπαράσταση της θέσης ενός συστήματος BC, την προέλευση ενός μηνύματος AEP (ή του γεγονότος που μεταφέρει) και το όνομα ενός ΥΕΔ αντίστοιχα. Η *BCURL* ορίζει την θέση του ΥΕΔ στο δίκτυο των διαμεσολαβητών. Η θέση του ΥΕΔ καθορίζεται είτε βάσει συνδυασμού διεύθυνσης IP και πόρτας μέσω της κλάσης της Java *java.net.InetSocketAddress*, είτε μέσω του αντικείμενου *BrokerName* το οποίο περιέχει ένα αντικείμενο *java.lang.String* που καθορίζει το URL του Διαμεσολαβητή. Η κλάση *BCMessage* αποτελεί την υπερκλάση για όλα τα μηνύματα που υποστηρίζει το πρωτόκολλο επικοινωνίας των ΥΕΔ. Περιέχει το πεδίο *originator* μέσω του οποίου ορίζεται ο αποστολέας του *BCMessage*. Το *RequestListenerMessage* αποστέλλεται από ένα ΥΕΔ σε ένα άλλο για την ανακάλυψη υποσυστήματος ικανού να εξυπηρετήσει κάποιο γεγονός συγκεκριμένου τύπου. Η απάντηση σε αυτό το μήνυμα είναι είτε ένα *InstallListenerMessage* είτε ένα *ListenerRefuseMessage*. Ο διαμεσολαβητής στον οποίο πρέπει να εγκατασταθεί ο Listener προκύπτει από το πεδίο *originator* του μηνύματος. Μετά την εγκατάσταση του Listener το τοπικό ΥΕΔ αποστέλλει ένα *ACKMessage* στον απομακρυσμένο διαμεσολαβητή. Η απεγκατάσταση των Listeners πραγματοποιείται με την αποστολή του *ListenerRemoveMessage*, μέσα στο οποίο καθορίζεται και ο τύπος του γεγονότος (*eventType*) το οποίο διαχειρίζεται ο προς απεγκατάσταση Listener. Το *DispatchEventMessage* χρησιμοποιείται για την μεταφορά των γεγονότων από το τοπικό στο απομακρυσμένο ΥΕΔ. Το απομακρυσμένο ΥΕΔ είναι υπεύθυνο για την αναδημιουργία του γεγονότος και την ανάθεση σε αυτό κατάλληλων *BCID* και *OriginatorID* προκειμένου το μήνυμα-απάντηση να μπορέσει να φτάσει στο αντικείμενο

από το οποίο προήλθε το αρχικό μήνυμα. Ο αλγόριθμος χρήσης των πεδίων *BCID* και *OriginatorID* παρουσιάζεται στο Σχήμα 40.



Σχήμα 40: Αλγόριθμος ανάθεσης BCID και OriginatorID στα μηνύματα YEA

Στο επόμενο διάγραμμα αλληλεπιδράσεων (Σχήμα 41) παρουσιάζεται η αλληλουχία μηνυμάτων ανάμεσα σε δύο YEA για την εγκατάσταση Listener, την μεταφορά του μηνύματος και την απεγκατάσταση του Listener.



Σχήμα 41: Εγκατάσταση Listener, Μεταφορά Μηνύματος και Απεγκατάσταση Listener

Στον πίνακα που ακολουθεί παρατίθενται οι κλάσεις που συγκροτούν το Υποσύστημα Επικοινωνίας Διαμεσολαβητών.

<i>Κλάση</i>	<i>Μέθοδος</i>	<i>Τύπος</i>	<i>Παράμετροι</i>
BCSubsystem	BCSubsystem()	constructor	-
	submit()	long	Event e
BCEventHandler	BCEventHandler()	constructor	Subsystem s
	handleEvent()	void	Event e
BCContext	getInstance()	BCContext	-
	getBCURL()	BCURL	-
	getRole()	BrokerRole	-
	getRegistry()	BCURL	-
	getPort()	int	-
BCURL	BCURL()	constructor	-
	getBrokerName()	BrokerName	-
	getURL()	String	-
	setBrokerName()	void	BrokerName name
	setURL()	void	String url
BrokerName	BrokerName()	constructor	-
	getName()	String	-
	getDescription()	String	-
	setName()	void	String name
	setDescription()	void	String description
OriginatorId	OriginatorID	constructor	-
	getOriginatorBC()	BCURL	-
	getOriginatorSubsystem()	AgentType	-
	setOriginatorBC()	void	BCURL bc
	setOriginatorSubsystem()	void	AgentType subsystem
DiscoveryManager	getInstance()	DiscoveryManager	-
	discover()	void	EventType evTp
	updateBrokerList()	void	-
	registerBroker()	void	-
	registerIMS()	void	-
	deliver()	void	BrokerListMessage msg
	getBrokerList()	BCURL[]	-

DispatchManager	getInstance()	DispatchManager	-
	submit()	void	Event e
	submitQueued()	void	EventType e
	deliver()	void	DispatchEventMessage msg
ListenerManager	getInstance()	ListenerManager	-
	deliver()	void	ListenerRequestMessage msg
	deliver()	void	ListenerRefuseMessage msg
	deliver()	void	ListenerInstallMessage msg
	getListenerRegistry()	ListenerRegistry	-
BCMessageCall	doCall()	void	ACKMessage msg, String endpoint
	doCall()	void	NACKMessage msg, String endpoint
	doCall()	void	DispatchEventMessage msg, String endpoint
	doCall()	void	ListenerInstallMessage msg, String endpoint
	doCall()	void	ListenerRequestMessage msg, String endpoint
	doCall()	void	ListenerRefuseMessage msg, String endpoint
	doCall()	void	ListenerRemoveMessage msg, String endpoint
	doCall()	void	BrokerListRequestMessage msg, String endpoint
	doCall()	void	BrokerListMessage msg, String endpoint
	doCall()	void	BrokerRegisterRequestMessage msg, String endpoint
	doCall()	void	BrokerRegisterResponseMessage msg, String endpoint
BCReceiver	receive()	void	ACKMessage msg
	receive()	void	NACKMessage msg
	receive()	void	DispatchEventMessage msg
	receive()	void	ListenerRequestMessage msg
	receive()	void	ListenerInstallMessage msg
	receive()	void	ListenerRefuseMessage msg
	receive()	void	ListenerRemoveMessage msg
	receive()	void	BrokerListRequestMessage msg
	receive()	void	BrokerListMessage msg
	receive()	void	BrokerRegisterRequestMessage msg
	receive()	void	BrokerRegisterResponseMessage msg

Όλα τα μηνύματα που ανταλλάσσουν τα ΥΕΔ μεταφέρονται εντός μηνυμάτων SOAP και συνεπώς απαιτείται καθορισμός συγκεκριμένων κανόνων σειριοποίησης σε XML όλων των αντικειμένων. Για τον λόγο αυτό όλα τα σχετιζόμενα με το μηχανισμό επικοινωνίας διαμεσολαβητών αντικείμενα εκφράζονται σε XML σχήματα (XML Schema). Τα XML σχήματα παρέχουν τη δομή και τη σύνταξη της σειριακής μορφής των μηνυμάτων.

Το `BCMessage.xsd` καθορίζει τη δομή του `BCMessage` και των σχετικών με αυτό αντικειμένων (`BCURL`, `BrokerName`).

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:bc="http://www.ist-simplicity.org/bc"
  targetNamespace="http://www.ist-simplicity.org/bc"
  elementFormDefault="qualified"
  attributeFormDefault="qualified">

  <xsd:element name="BCMessage" type="bc:BCMessageType" />

  <xsd:complexType name="BCMessageType">
    <xsd:element name="originator" type="bc:BCURLType" />
  </xsd:complexType>

  <xsd:complexType name="BCURLType">
    <xsd:sequence>
      <xsd:element name="description" type="xsd:String"/>
      <xsd:element name="inetAddr" type="bc:InetAddress"/>
      <xsd:element name="brokerName" type="bc:BrokerName"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="InetAddress">
    <xsd:sequence>
      <xsd:element name="ip" type="xsd:String"/>
      <xsd:element name="port">
        <xsd:simpleType>
          <xsd:restriction base="xsd:positiveInteger">
            <xsd:maxExclusive value="65536"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="BrokerName">
    <xsd:sequence>
      <xsd:element name="name" type="xsd:String"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

Το `Core.xsd` καθορίζει τη δομή των δύο βασικών αντικειμένων, της υπερκλάσης `Event` και της κλάσης `EventType class`.

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:events="http://www.ist-simplicity.org/core"
  targetNamespace="http://www.ist-simplicity.org/core"
  elementFormDefault="qualified"
  attributeFormDefault="qualified">

  <xsd:complexType name="EventType">
    <xsd:sequence>
```

```

        <xsd:element name="typeName" type="xsd:string"/>
        <xsd:element name="id" type="xsd:int"/>
        <xsd:element name="baseType" type="EventType" nillable="true"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Event">
    <xsd:sequence>
        <xsd:element name="type" type="core:EventType"/>
        <xsd:element name="date" type="xsd:dateTime"/>
    </xsd:sequence>
</xsd:complexType>

</xsd:schema>

```

Τα XML σχήματα που ακολουθούν καθορίζουν τη δομή των μηνυμάτων που ανταλλάσσουν τα ΥΕΔ.

RequestListenerMessage.xsd

```

<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:bc="http://www.ist-simplicity.org/bc"
    xmlns:core="http://www.ist-simplicity.org/core"
    targetNamespace="http://www.ist-simplicity.org/bc"
    elementFormDefault="qualified"
    attributeFormDefault="qualified">

<xsd:include schemaLocation="BCMessage.xsd"/>
<xsd:import namespace="http://www.ist-simplicity.org/core" schemaLocation="Core.xsd"/>

<xsd:element name="RequestListenerMessage" type="bc:RequestListenerMessageType"/>

<xsd:complexType name="RequestListenerMessageType">
    <xsd:sequence>
        <xsd:element name="originator" type="bc:BCURLType" />
        <xsd:element name="eventType" type="core:EventType" />
    </xsd:sequence>
</xsd:complexType>

</xsd:schema>

```

InstallListenerMessage.xsd

```

<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:bc="http://www.ist-simplicity.org/bc"
    xmlns:core="http://www.ist-simplicity.org/core"
    targetNamespace="http://www.ist-simplicity.org/bc"
    elementFormDefault="qualified"
    attributeFormDefault="qualified">

<xsd:include schemaLocation="BCMessage.xsd"/>
<xsd:import namespace="http://www.ist-simplicity.org" schemaLocation="Core.xsd"/>

<xsd:element name="InstallListenerMessage" type="bc:InstallListenerMessageType"/>

<xsd:complexType name="InstallListenerMessageType">
    <xsd:sequence>
        <xsd:element name="originator" type="bc:BCURLType" />
        <xsd:element name="eventType" type="core:EventType" />
    </xsd:sequence>
</xsd:complexType>

</xsd:schema>

```


ListenerRefuseMessage.xsd

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:bc="http://www.ist-simplicity.org/bc"
  xmlns:core="http://www.ist-simplicity.org/core"
  targetNamespace="http://www.ist-simplicity.org/bc"
  elementFormDefault="qualified"
  attributeFormDefault="qualified">

<xsd:include schemaLocation="BCMessage.xsd"/>
<xsd:import namespace="http://www.ist-simplicity.org/core" schemaLocation="Core.xsd"/>

<xsd:element name="ListenerRefuseMessage" type="bc:ListenerRefuseMessageType"/>

<xsd:complexType name="ListenerRefuseMessageType">
  <xsd:sequence>
    <xsd:element name="originator" type="bc:URLType" />
    <xsd:element name="eventType" type="core:EventType" />
  </xsd:sequence>
</xsd:complexType>
</xsd:schema>
```

ListenerRemoveMessage.xsd

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:bc="http://www.ist-simplicity.org/bc"
  xmlns:core="http://www.ist-simplicity.org/core"
  targetNamespace="http://www.ist-simplicity.org/bc"
  elementFormDefault="qualified"
  attributeFormDefault="qualified">

<xsd:include schemaLocation="BCMessage.xsd"/>
<xsd:import namespace="http://www.ist-simplicity.org/core" schemaLocation="Core.xsd"/>

<xsd:element name="ListenerRemoveMessage" type="bc:ListenerRemoveMessageType"/>

<xsd:complexType name="ListenerRemoveMessageType">
  <xsd:sequence>
    <xsd:element name="originator" type="bc:BCURLType" />
    <xsd:element name="eventType" type="core:EventType" />
  </xsd:sequence>
</xsd:complexType>
</xsd:schema>
```

DispatchEventMessage.xsd

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:bc="http://www.ist-simplicity.org/bc"
  xmlns:core="http://www.ist-simplicity.org/core"
  targetNamespace="http://www.ist-simplicity.org/bc"
  elementFormDefault="qualified"
  attributeFormDefault="qualified">

<xsd:include schemaLocation="BCMessage.xsd"/>
<xsd:import namespace="http://www.ist-simplicity.org/core" schemaLocation="Core.xsd"/>

<xsd:element name="DispatchEventMessage" type="bc:DispatchEventMessageType"/>

<xsd:complexType name="DispatchEventMessageType">
  <xsd:sequence>
    <xsd:element name="originator" type="bc:BCURLType" />
    <xsd:element name="event" type="events:Event" />
  </xsd:sequence>
</xsd:complexType>
</xsd:schema>
```

ACKMessage.xsd

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:bc="http://www.ist-simplicity.org/bc"
  targetNamespace="http://www.ist-simplicity.org/bc"
  elementFormDefault="qualified"
  attributeFormDefault="qualified">

<xsd:include schemaLocation="Message.xsd"/>

<xsd:element name="ACKMessage" type="bc:ACKMessageType"/>

<xsd:complexType name="ACKMessageType">
  <xsd:sequence>
    <xsd:element name="message" type="xsd:string" />
  </xsd:sequence>
</xsd:complexType>

</xsd:schema>
```

NACKMessage.xsd

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:bc="http://www.ist-simplicity.org/bc"
  targetNamespace="http://www.ist-simplicity.org/bc"
  elementFormDefault="qualified"
  attributeFormDefault="qualified">

<xsd:include schemaLocation="BCMessage.xsd"/>

<xsd:element name="NACKMessage" type="bc:NACKMessageType"/>

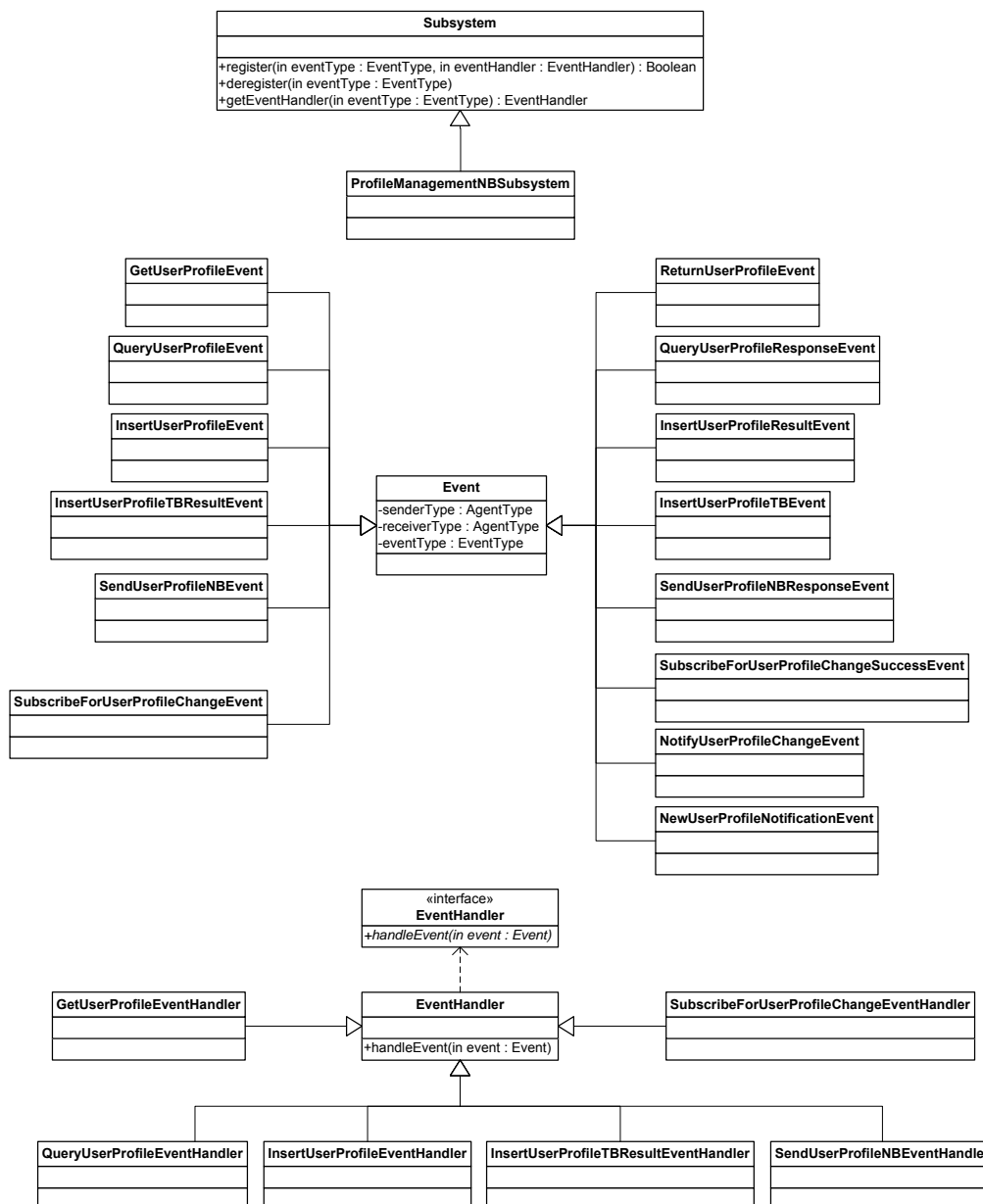
<xsd:complexType name="NACKMessageType">
  <xsd:sequence>
    <xsd:element name="message" type="xsd:string" />
  </xsd:sequence>
</xsd:complexType>

</xsd:schema>
```

3.3.3 Υποσύστημα Διαχείρισης Προφίλ Χρήστη

Στα παρακάτω διαγράμματα (Σχήμα 42, Σχήμα 43) παρουσιάζονται οι σημαντικότερες κλάσεις της υλοποίησης των Υποσυστημάτων Διαχείρισης Προφίλ Χρήστη στους Διαμεσολαβητές Δικτύου και Τερματικού.

Κάθε ένα από τα δύο υποσυστήματα αποτελεί υποκλάση της κλάσης *Subsystem* και λαμβάνει και αποστέλλει γεγονότα (events) που αποτελούν υποκλάσεις της κλάσης *Event*. Τα γεγονότα που παρουσιάζονται στην αριστερή πλευρά κάθε διαγράμματος είναι αυτά τα οποία δύναται να λάβει το υποσύστημα, ενώ στα δεξιά παρουσιάζονται αυτά που αποστέλλει. Για κάθε γεγονός που λαμβάνει το υποσύστημα υλοποιείται και κατάλληλος διαχειριστής γεγονότος (*EventHandler*).



Σχήμα 42: Διάγραμμα κλάσεων του Υποσυστήματος Διαχείρισης Προφίλ Χρήστη στον Διαμεσολαβητή Δικτύου

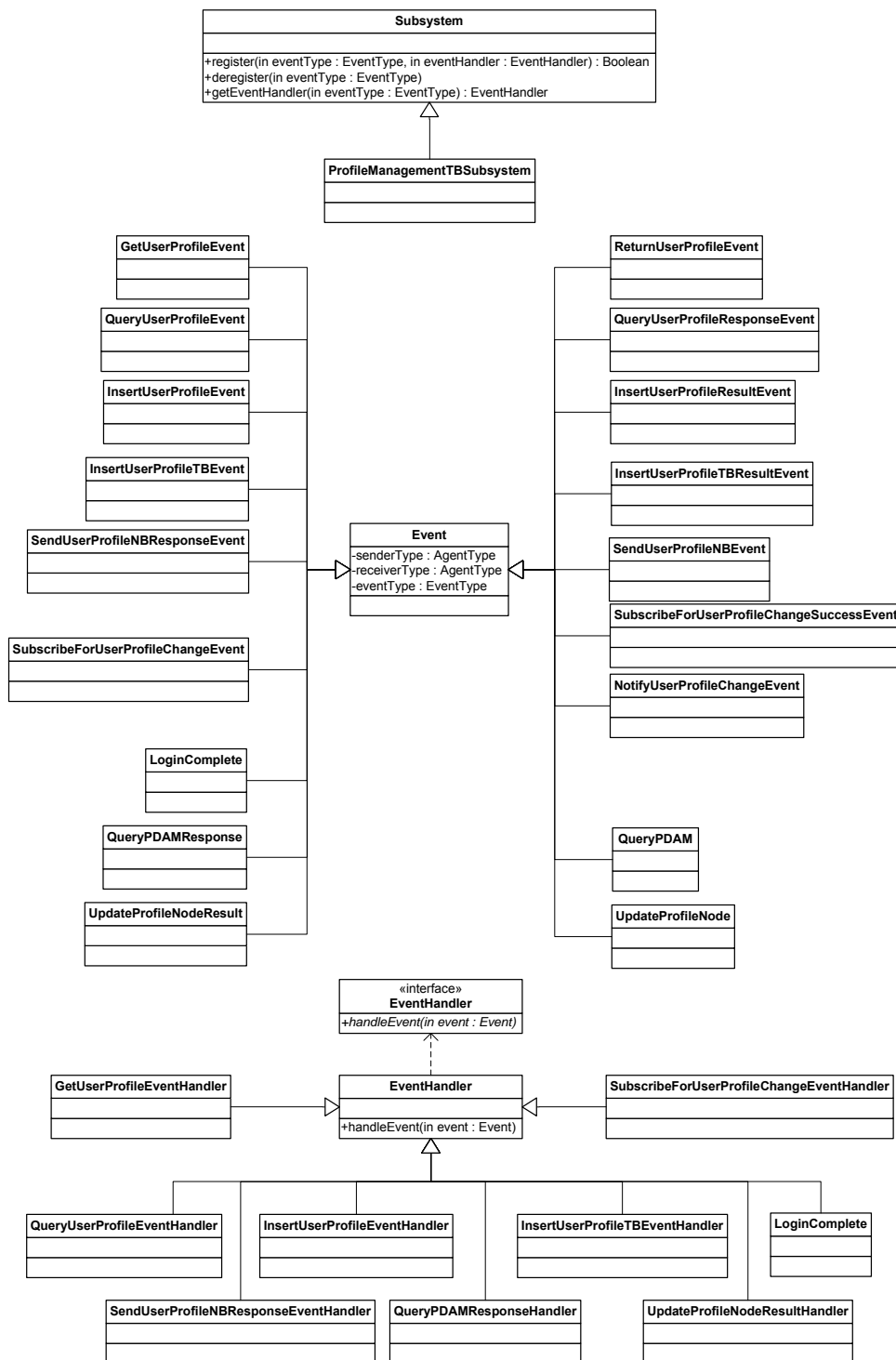
Το γεγονός *GetUserProfileEvent* είναι δυνατόν να αποσταλεί στον Διαχειριστή Προφίλ Χρήστη από οποιοδήποτε υποσύστημα βρίσκεται στον Διαμεσολαβητή Δικτύου προκειμένου να ανακτήσει ολόκληρο το προφίλ του χρήστη. Το γεγονός αυτό φέρει ως παράμετρο τον διακριτικό αριθμό (UserID), που καθορίζει τον χρήστη του οποίου το προφίλ ζητείται. Το υποσύστημα Διαχειριστή Προφίλ Χρήστη σαν απάντηση στο γεγονός αυτό αποστέλλει προς το αιτούμενο υποσύστημα το γεγονός *SendUserProfileEvent* μέσω του οποίου επιστρέφει το προφίλ χρήστη σε μορφή XML αρχείου. Όταν ένα υποσύστημα του Διαμεσολαβητή Δικτύου χρειαστεί ένα μέρος και όχι ολόκληρο το προφίλ του χρήστη, μπορεί να αποστείλει προς το υποσύστημα Προφίλ Χρήστη το γεγονός

QueryUserProfileEvent. Μέσω του γεγονότος αυτού με κατάλληλες παραμέτρους καθορίζεται ποιο μέρος του προφίλ ζητείται και φυσικά για ποιόν χρήστη. Το ερωτώμενο υποσύστημα απαντά με το γεγονός *QueryUserProfileResponseEvent* το οποίο σαν παραμέτρους φέρει το ID του χρήστη και το ζητούμενο μέρος του προφίλ σε μορφή XML. Μέσω των γεγονότων *InsertUserProfileEvent* και *InsertUserProfileTBEvent* παρέχεται η δυνατότητα ανανέωσης ή διαγραφής μερών του προφίλ του χρήστη. Το πρώτο γεγονός αποστέλλεται από ένα τοπικό υποσύστημα του Διαμεσολαβητή Δικτύου στο Υποσύστημα Διαχείρισης Προφίλ Χρήστη και εν συνεχεία το τελευταίο προωθεί το αίτημα στο δίδυμό του υποσύστημα στον Διαμεσολαβητή Τερματικού μέσω του δεύτερου γεγονότος. Οι παράμετροι είναι και πάλι το ID του χρήστη, μία αναφορά στο μέρος του προφίλ που πρέπει να ανανεωθεί και οι νέες τιμές τις οποίες πρέπει να πάρει το μέρος του προφίλ που πρόκειται να ανανεωθεί. Όταν πρόκειται για διαγραφή μερών του προφίλ, η νέα τιμή είναι *null*.

Με την ολοκλήρωση των διαδικασιών ανανέωσης ή διαγραφής μερών του προφίλ τα υποσυστήματα που έστειλαν τα γεγονότα λαμβάνουν και τις αντίστοιχες απαντήσεις μέσω των γεγονότων *InsertUserProfileResultEvent* και *InsertUserProfileTBResultEvent* με παραμέτρο ακέραιο αριθμό 1 ή 0, υποδηλώνοντας έτσι την επιτυχία ή αποτυχία της διαδικασίας.

Στις περιπτώσεις αλλαγών των προφίλ απαιτείται η ενημέρωση των υποσυστημάτων που έχουν εγγραφεί προκειμένου να λαμβάνουν γεγονότα ενημέρωσης αλλαγών. Η εγγραφή ενός υποσυστήματος για ενημέρωση πραγματοποιείται με αποστολή του γεγονότος *SubscribeForUserProfileChangeEvent* με παράμετρο το ID του χρήστη στο Υποσύστημα Διαχείρισης προφίλ του Διαμεσολαβητή. Η επιτυχία ή αποτυχία της διαδικασίας εγγραφής γνωστοποιείται στο υποσύστημα που εγγράφεται με αποστολή σε αυτό του γεγονότος *SubscribeForUserProfileChangeEvent* με την κατάλληλη παράμετρο. Στην περίπτωση που πραγματοποιείται αλλαγή σε κάποιο προφίλ, το Υποσύστημα Διαχείρισης Προφίλ Χρήστη αποστέλλει σε όλα τα εγγεγραμμένα τοπικά υποσυστήματα το γεγονός *NotifyUserProfileChangeEvent* αποστέλλοντας το ανανεωμένο προφίλ με τη μορφή XML αρχείου συνοδευμένο από το ID του χρήστη. Με το γεγονός *SendUserProfileNBEvent* το Υποσύστημα Διαχείρισης Προφίλ Χρήστη του Διαμεσολαβητή Τερματικού αποστέλλει το προφίλ κάποιου χρήστη στο αντίστοιχο υποσύστημα του Διαμεσολαβητή Δικτύου κάθε φορά που πραγματοποιείται κάποια αλλαγή, αλλά και κατά την εισαγωγή ενός νέου χρήστη στο σύστημα. Στην δεύτερη περίπτωση, όταν δηλαδή το *SendUserProfileNBEvent* αποστέλλεται λόγω εισαγωγής νέου χρήστη στο σύστημα, το Υποσύστημα Διαχείρισης Προφίλ Χρήστη του Διαμεσολαβητή Δικτύου αποστέλλει το γεγονός

NewUserProfileNotificationEvent στο Υποσύστημα Διαχείρισης Πληροφορίας Πλαισίου προκειμένου το τελευταίο να εκκινήσει διαδικασίες ανάκτησης των πληροφοριών για τον νέο χρήστη, αλλά και εγγραφής για μελλοντική ενημέρωση σε περιπτώσεις αλλαγών.



Σχήμα 43: Διάγραμμα κλάσεων του Υποσυστήματος Διαχείρισης Προφίλ Χρήστη στον Διαμεσολαβητή Τερματικού

Όσον αφορά στο αντίστοιχο υποσύστημα του Διαμεσολαβητή τερματικού, τα γεγονότα που λαμβάνει και αποστέλλει ταυτίζονται με αυτά που αναφέρθηκαν παραπάνω και

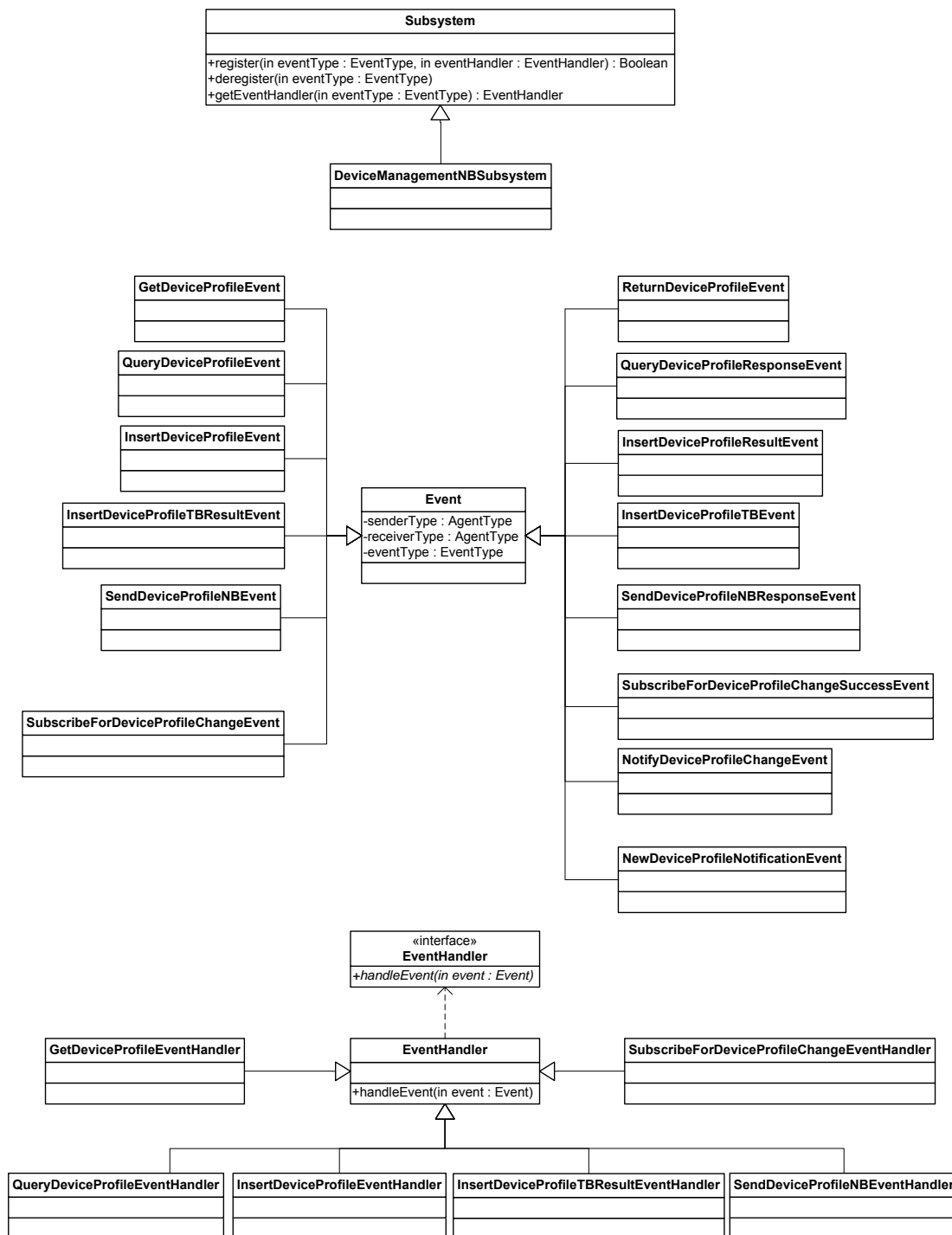
πραγματοποιούν τις ίδιες ακριβώς διαδικασίες. Η μόνη διαφορά ανάμεσα στις δύο υποστάσεις του Υποσυστήματος Διαχείρισης Προφίλ Χρήστη είναι ότι αυτό του Διαμεσολαβητή Τερματικού αλληλεπιδρά με το Υποσύστημα Διαχειριστή Πρόσβασης Προσωπικής Συσκευής Χρήστη, το οποίο λειτουργεί ως ενδιάμεσος για την επικοινωνία ολόκληρου του συστήματος με την Προσωπική Συσκευή Χρήστη.

Τα γεγονότα που εμπλέκονται στις αλληλεπιδράσεις αυτών των υποσυστημάτων είναι τα *LoginComplete*, *QueryPDAM*, *QueryPDAMResponse*, *UpdateProfileNode* και *UpdateProfileNodeResult*. Το *LoginComplete* αποστέλλεται από το Υποσύστημα Διαχειριστή Πρόσβασης Προσωπικής Συσκευής Χρήστη στο υποσύστημα Διαχείρισης Προφίλ Χρήστη του Διαμεσολαβητή Τερματικού προκειμένου να ενημερωθεί το δεύτερο για την ολοκλήρωση της διαδικασίας εισαγωγής στο σύστημα ενός νέου χρήστη. Το γεγονός αυτό μεταφέρει ως παράμετρο το διακριτικό αριθμό χρήστη (UserID). Μετά τη λήψη του συγκεκριμένου γεγονότος το υποσύστημα Διαχείρισης Προφίλ Χρήστη δημιουργεί και στέλνει στο Υποσύστημα Διαχειριστή Πρόσβασης Προσωπικής Συσκευής Χρήστη το γεγονός *QueryPDAM* προκειμένου να ανακτήσει το προφίλ του νεοεισαχθέντα χρήστη. Με το *QueryPDAMResponse* επιστρέφεται στο Υποσύστημα Διαχείρισης Προφίλ ολόκληρο το προφίλ του χρήστη με τη μορφή XML αρχείου.

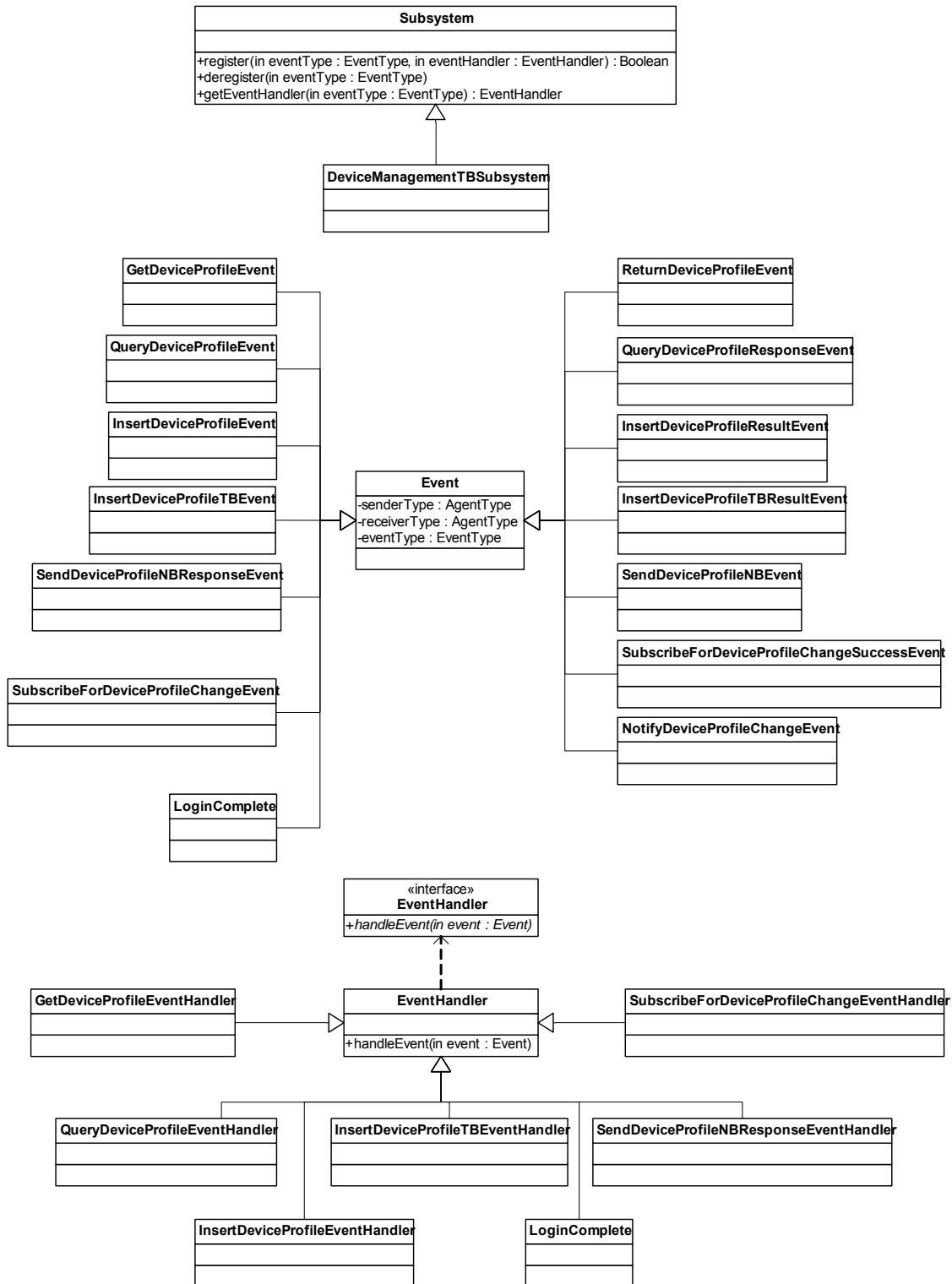
Τέλος, με την αποστολή του γεγονότος *UpdateProfileNode* από το Υποσύστημα Διαχείρισης Προφίλ Χρήστη προς το Υποσύστημα Διαχειριστή Πρόσβασης Προσωπικής Συσκευής Χρήστη πραγματοποιούνται οι διαδικασίες αλλαγής, εισαγωγής νέων στοιχείων ή διαγραφής υπαρχόντων στο προφίλ του χρήστη σε περιπτώσεις αλλαγών. Η διαδικασία που πρέπει να πραγματοποιηθεί καθορίζεται από ειδικό πεδίο του γεγονότος που μπορεί να πάρει τις τιμές REPLACE, ADD ή DELETE. Το γεγονός *UpdateProfileNodeResult* ενημερώνει τον αποστολέα του *UpdateProfileNode* γεγονότος για την επιτυχή ή όχι έκβαση της διαδικασίας.

3.3.4 Υποσύστημα Διαχείρισης Δυνατοτήτων Τερματικών Συσκευών

Στην ενότητα αυτή παρουσιάζονται τα διαγράμματα κλάσεων των Διαχειριστών Δυνατοτήτων Τερματικών Συσκευών των Διαμεσολαβητών Δικτύου και Τερματικού (Σχήμα 44, Σχήμα 45). Η υλοποίηση ταυτίζεται πλήρως με αυτή των Υποσυστημάτων Διαχείρισης Προφίλ Χρήστη και τα γεγονότα που ανταλλάσσονται αποτελούν υποσύνολο των γεγονότων που παρουσιάστηκαν στην προηγούμενη ενότητα. Για αυτόν τον λόγο παρατίθενται μόνο τα διαγράμματα κλάσεων.



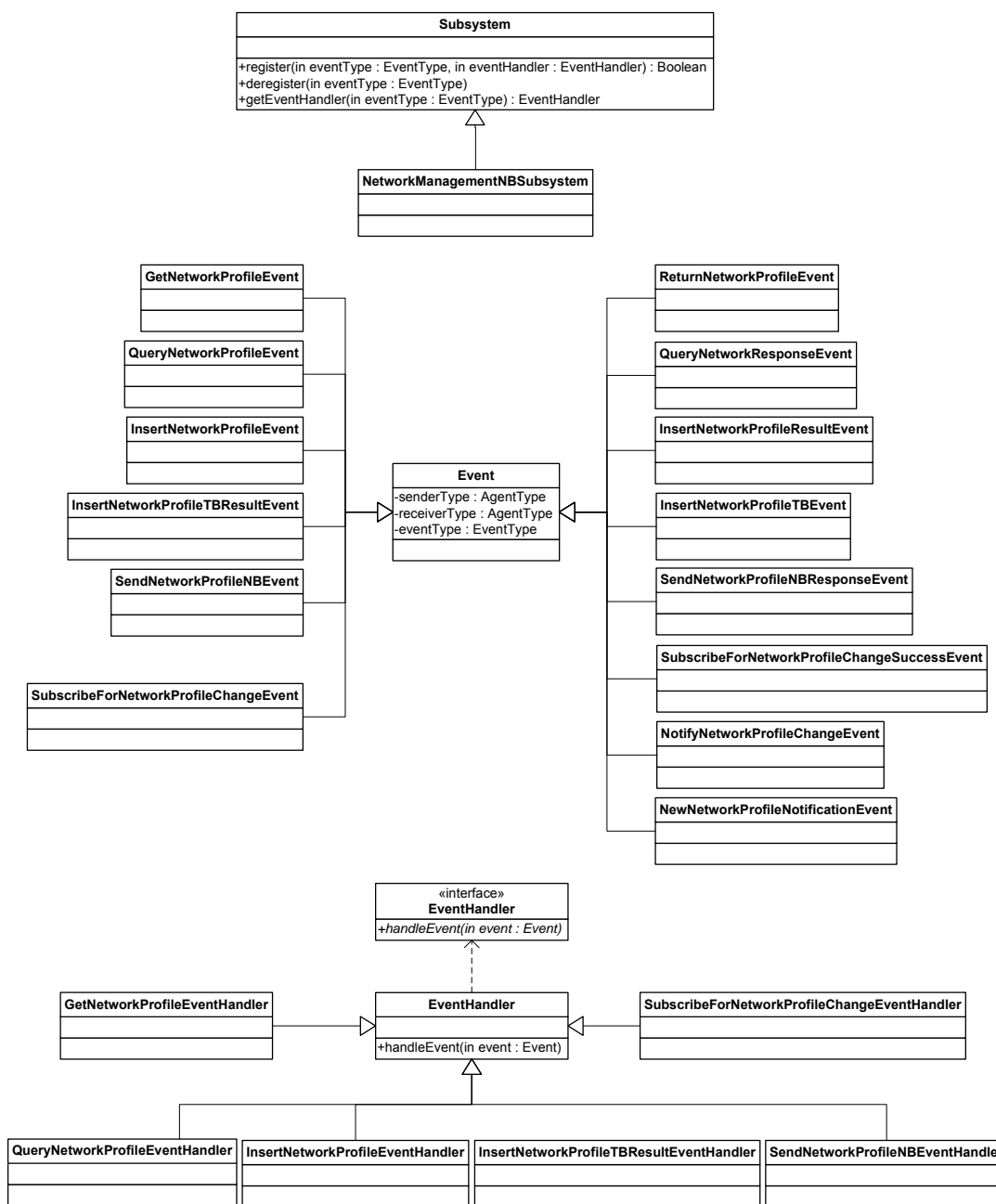
Σχήμα 44: Διάγραμμα κλάσεων του Υποσυστήματος Διαχείρισης Δυνατοτήτων Τερματικών Συσκευών στον Διαμεσολαβητή Δικτύου



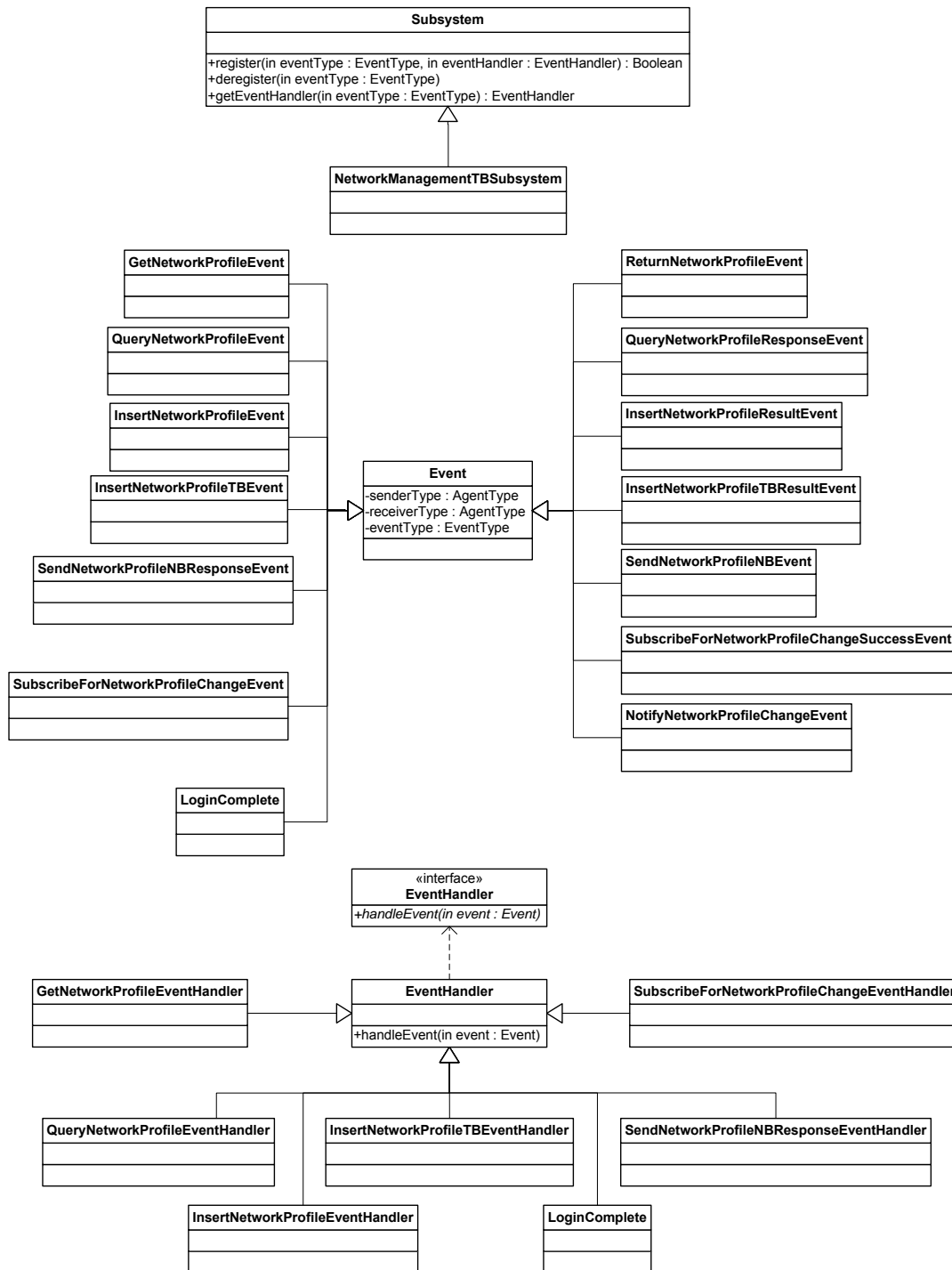
Σχήμα 45: Διάγραμμα κλάσεων του Υποσυστήματος Διαχείρισης Δυνατοτήτων Τερματικών Συσκευών στον Διαμεσολαβητή Τερματικού

3.3.5 Υποσύστημα Διαχείρισης Δικτυακών Συνδέσεων

Στην ενότητα αυτή παρουσιάζονται τα διαγράμματα κλάσεων των Διαχειριστών Δικτυακών Συνδέσεων των Διαμεσολαβητών Δικτύου και Τερματικού (Σχήμα 46, Σχήμα 47). Η υλοποίηση ταυτίζεται πλήρως με αυτή των Υποσυστημάτων Διαχείρισης Προφίλ Χρήστη και τα γεγονότα που ανταλλάσσονται αποτελούν υποσύνολο των γεγονότων που παρουσιάστηκαν στην ενότητα 3.3.3. Για τον λόγο αυτό παρατίθενται μόνο τα διαγράμματα κλάσεων.



Σχήμα 46: Διάγραμμα κλάσεων του Υποσυστήματος Διαχείρισης Δικτυακών Συνδέσεων στον Διαμεσολαβητή Δικτύου

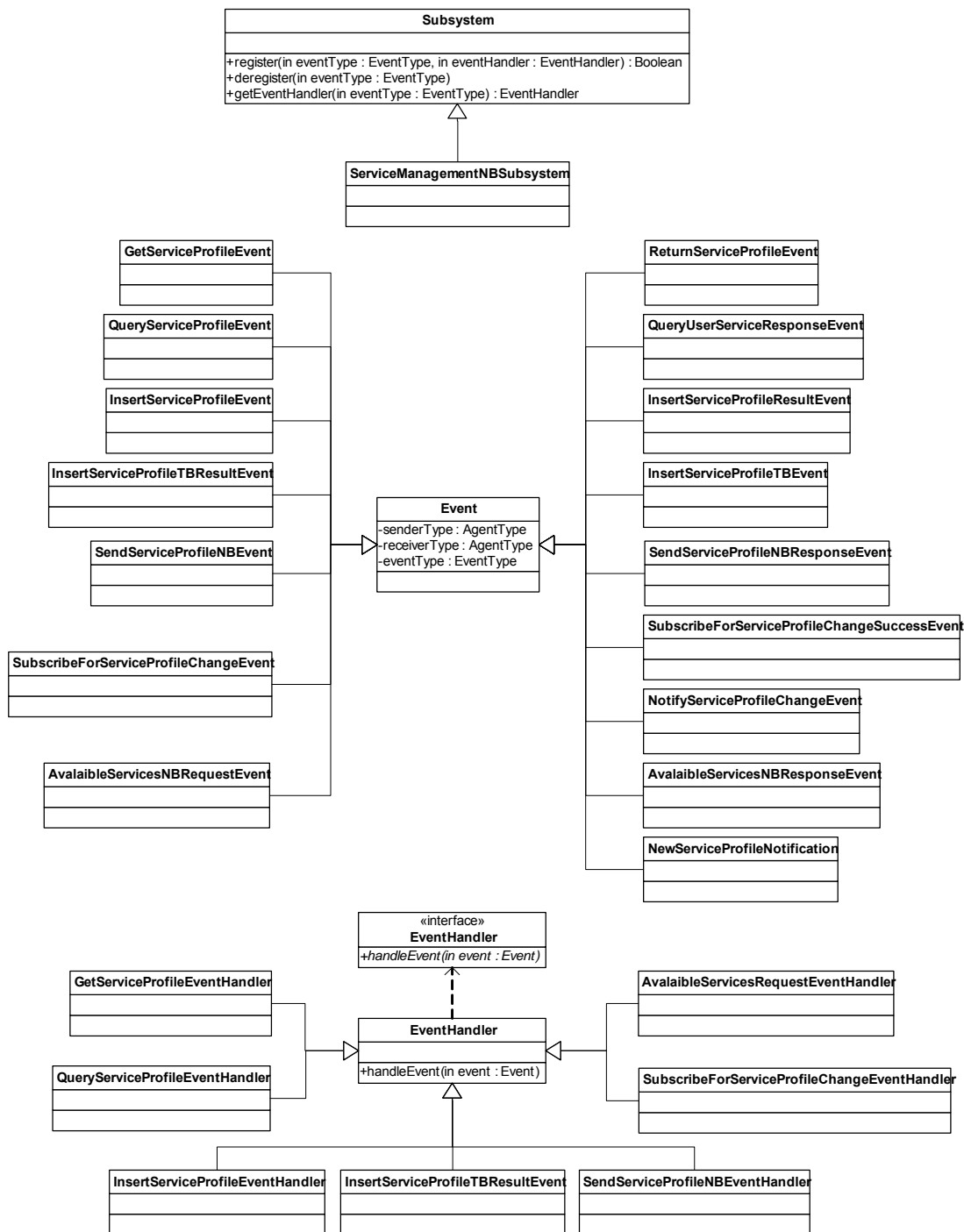


Σχήμα 47: Διάγραμμα κλάσεων του Υποσυστήματος Διαχείρισης Δικτυακών Συνδέσεων στον Διαμεσολαβητή Τερματικού

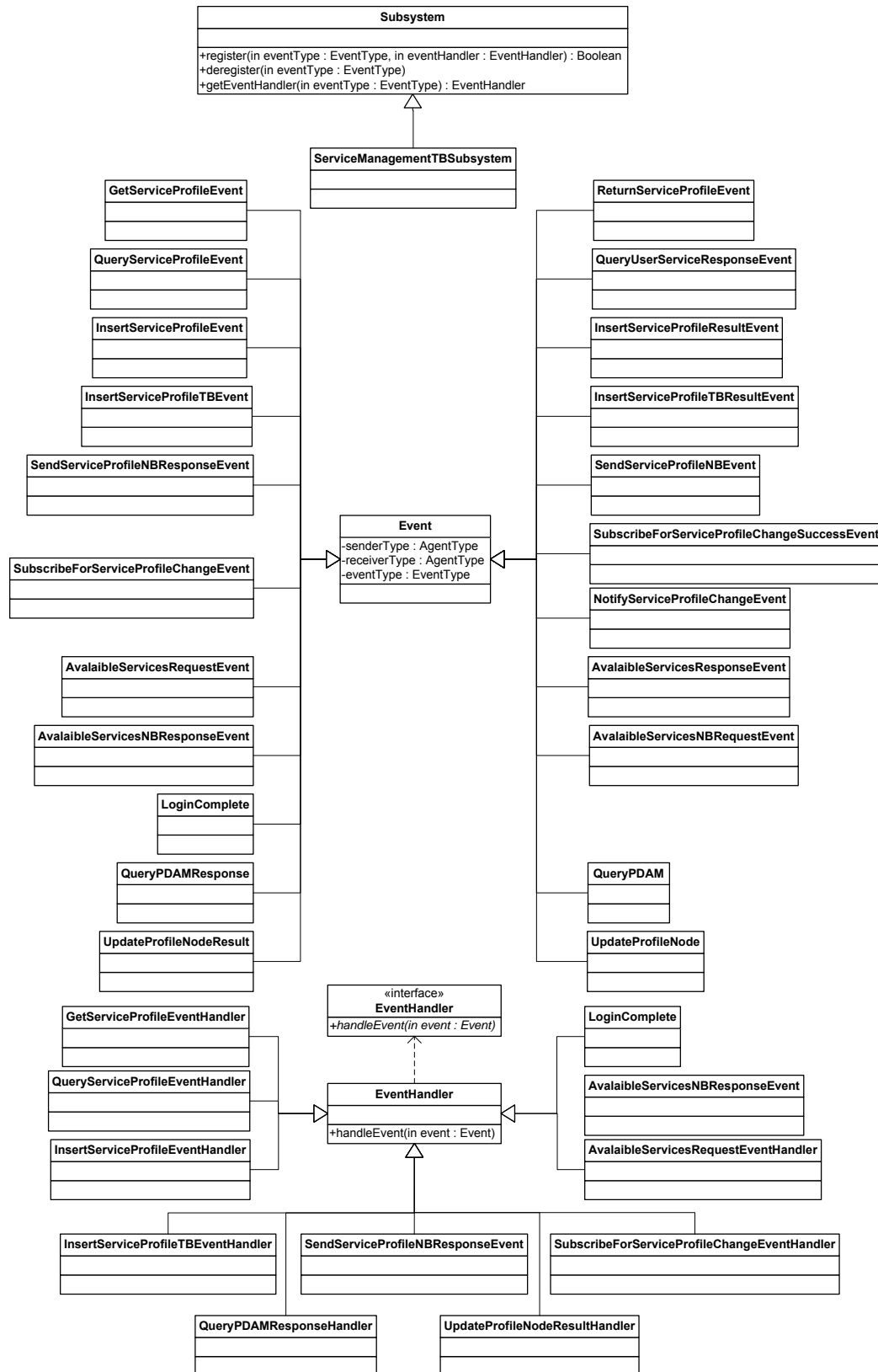
3.3.6 Υποσύστημα Διαχείρισης Υπηρεσιών

Στην ενότητα αυτή παρουσιάζονται τα διαγράμματα κλάσεων των Διαχειριστών Τερματικών Συσκευών των Διαμεσολαβητών Δικτύου και Τερματικού (Σχήμα 48, Σχήμα 49). Η υλοποίηση ταυτίζεται πλήρως με αυτή των Υποσυστημάτων Διαχείρισης Προφίλ Χρήστη. Πέρα από τα κοινά γεγονότα με τα Υποσυστήματα που παρουσιάστηκαν στις

προηγούμενες ενότητες, τα Υποσυστήματα Διαχείρισης Υπηρεσιών αποστέλλουν και λαμβάνουν ορισμένα ακόμα. Όπως παρουσιάζεται στο Σχήμα 48, το υποσύστημα της πλευράς του δικτύου λαμβάνει επιπροσθέτως το γεγονός *AvailableServiceNBRequestEvent* και σαν απάντηση στο συγκεκριμένο γεγονός αποστέλλει το *AvailableServiceNBResponseEvent*.



Σχήμα 48: Διάγραμμα κλάσεων του Υποσυστήματος Διαχείρισης Υπηρεσιών στον Διαμεσολαβητή Δικτύου



Σχήμα 49: Διάγραμμα κλάσεων του Υποσυστήματος Διαχείρισης Υπηρεσιών στον Διαμεσολαβητή Τερματικού

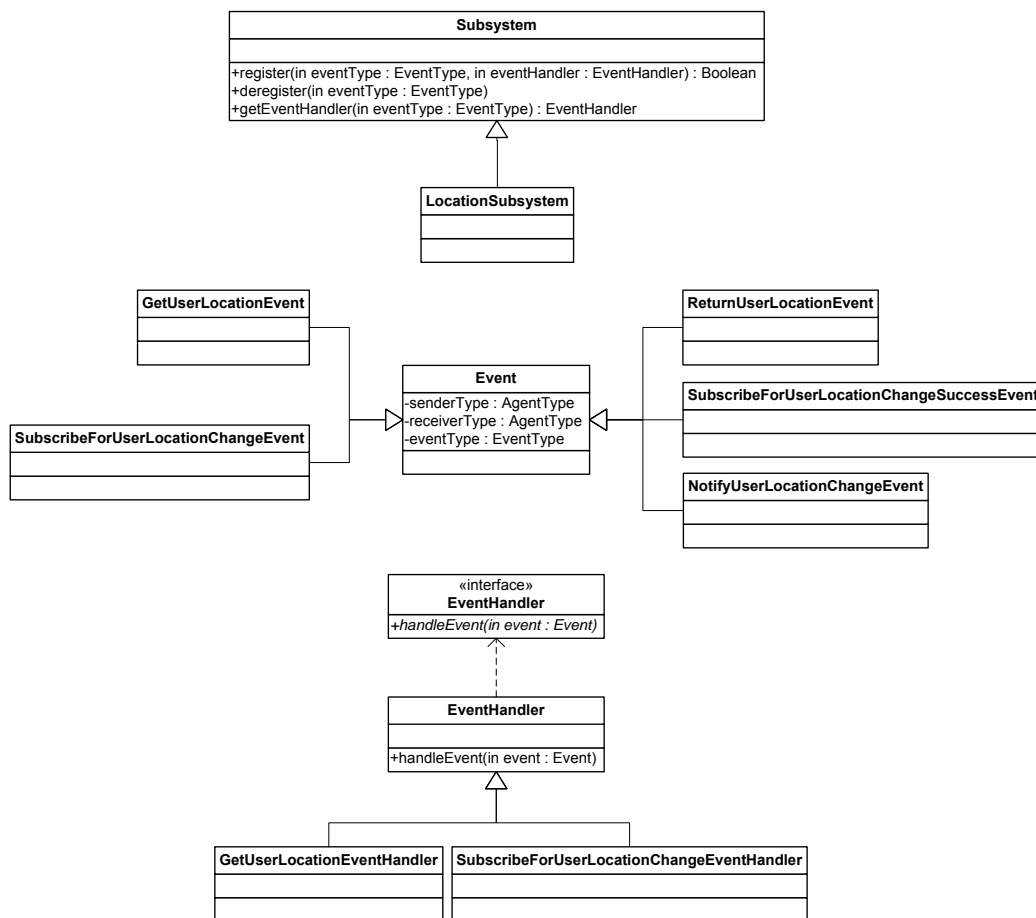
Το γεγονός-αίτηση προέρχεται από το δίδυμο υποσύστημα στην πλευρά του τερματικού και το ζητούμενο είναι η λίστα με τις διαθέσιμες από το σύστημα υπηρεσίες που διατηρεί το

Υποσύστημα Διαχείρισης υπηρεσιών του Διαμεσολαβητή Δικτύου. Με το γεγονός-απάντηση αποστέλλεται στο Υποσύστημα Διαχείρισης Υπηρεσιών του Διαμεσολαβητή Τερματικού η λίστα με τις υπηρεσίες και η θέση τους (π.χ. ένα URL) στο δίκτυο.

Η αποστολή του γεγονότος *AvailableServiceNBRequestEvent* από το υποσύστημα του τερματικού σε αυτό του Διαμεσολαβητή Δικτύου είναι αποτέλεσμα της λήψης από το Υποσύστημα Διαχείρισης Υπηρεσιών του Διαμεσολαβητή Τερματικού του γεγονότος *AvailableServiceRequestEvent*. Αφού το υποσύστημα στην πλευρά του δικτύου λάβει τη λίστα με τις υπηρεσίες μέσω του *AvailableServiceNBResponseEvent*, την προωθεί στο τερματικό μέσω του γεγονότος *AvailableServiceResponseEvent*.

3.3.7 Υποσύστημα Διαχείρισης Πληροφορίας Θέσης Χρήστη

Το Υποσύστημα Διαχείρισης Πληροφορίας Θέσης Χρήστη, όπως και τα υποσυστήματα που έχουν περιγραφεί παραπάνω, αποτελεί υποκλάση της κλάσης *Subsystem* και λαμβάνει και αποστέλλει γεγονότα (events) που αποτελούν υποκλάσεις της κλάσης *Event*.



Σχήμα 50: Διάγραμμα κλάσεων του Υποσυστήματος Διαχείρισης Πληροφορίας Θέσης Χρήστη στον Διαμεσολαβητή Δικτύου

Τα γεγονότα που παρουσιάζονται στην αριστερή πλευρά κάθε διαγράμματος είναι αυτά τα οποία δύναται να λάβει το υποσύστημα, ενώ στα δεξιά παρουσιάζονται αυτά που αποστέλλει. Για κάθε γεγονός που λαμβάνει το υποσύστημα υλοποιείται και κατάλληλος διαχειριστής γεγονότος (*EventHandler*). Η βασική διαφορά του συγκεκριμένου υποσυστήματος από τα υπόλοιπα είναι ότι υπόστασή του υπάρχει μόνο στο Διαμεσολαβητή Δικτύου. Αυτό οφείλεται στο γεγονός ότι στο πλαίσιο της συγκεκριμένης υλοποίησης, η ανάκτηση της θέσης του χρήστη πραγματοποιείται με χρήση μεθόδου βασισμένης στο δίκτυο και όχι στο τερματικό. Τα γεγονότα που δέχεται το υποσύστημα αυτό είναι τα *GetUserLocationEvent* και *SubscribeForUserLocationChangeEvent*. Το γεγονός *GetUserProfileEvent* είναι δυνατόν να αποσταλεί από οποιοδήποτε υποσύστημα βρίσκεται στον Διαμεσολαβητή Δικτύου προκειμένου να ανακτήσει τη θέση του χρήστη. Το γεγονός αυτό φέρει ως παράμετρο τον διακριτικό αριθμό (*UserID*), που καθορίζει τον χρήστη του οποίου η θέση ζητείται. Το υποσύστημα Διαχειριστή Προφίλ Χρήστη σαν απάντηση στο γεγονός αυτό αποστέλλει προς το αιτούμενο υποσύστημα το γεγονός *ReturnUserLocationEvent* μέσω του οποίου επιστρέφει τη θέση του χρήστη με τη μορφή συντεταγμένων.

Στις περιπτώσεις αλλαγών θέσης απαιτείται η ενημέρωση των υποσυστημάτων που έχουν εγγραφεί προκειμένου να λαμβάνουν γεγονότα ενημέρωσης αλλαγών. Η εγγραφή ενός υποσυστήματος για ενημέρωση πραγματοποιείται με αποστολή του γεγονότος *SubscribeForUserLocationChangeEvent* με παράμετρο το ID του χρήστη στο Υποσύστημα Διαχείρισης Πληροφορίας Θέσης Χρήστη του Διαμεσολαβητή. Η επιτυχία ή αποτυχία της διαδικασίας εγγραφής γνωστοποιείται στο υποσύστημα που εγγράφεται με αποστολή σε αυτό του γεγονότος *SubscribeForUserLocationChangeEvent* με την κατάλληλη παράμετρο. Στην περίπτωση που πραγματοποιείται αλλαγή, το Υποσύστημα Διαχείρισης Πληροφορίας Θέσης Χρήστη αποστέλλει σε όλα τα εγγεγραμμένα τοπικά υποσυστήματα το γεγονός *NotifyUserLocationChangeEvent* αποστέλλοντας τη νέα θέση του χρήστη.

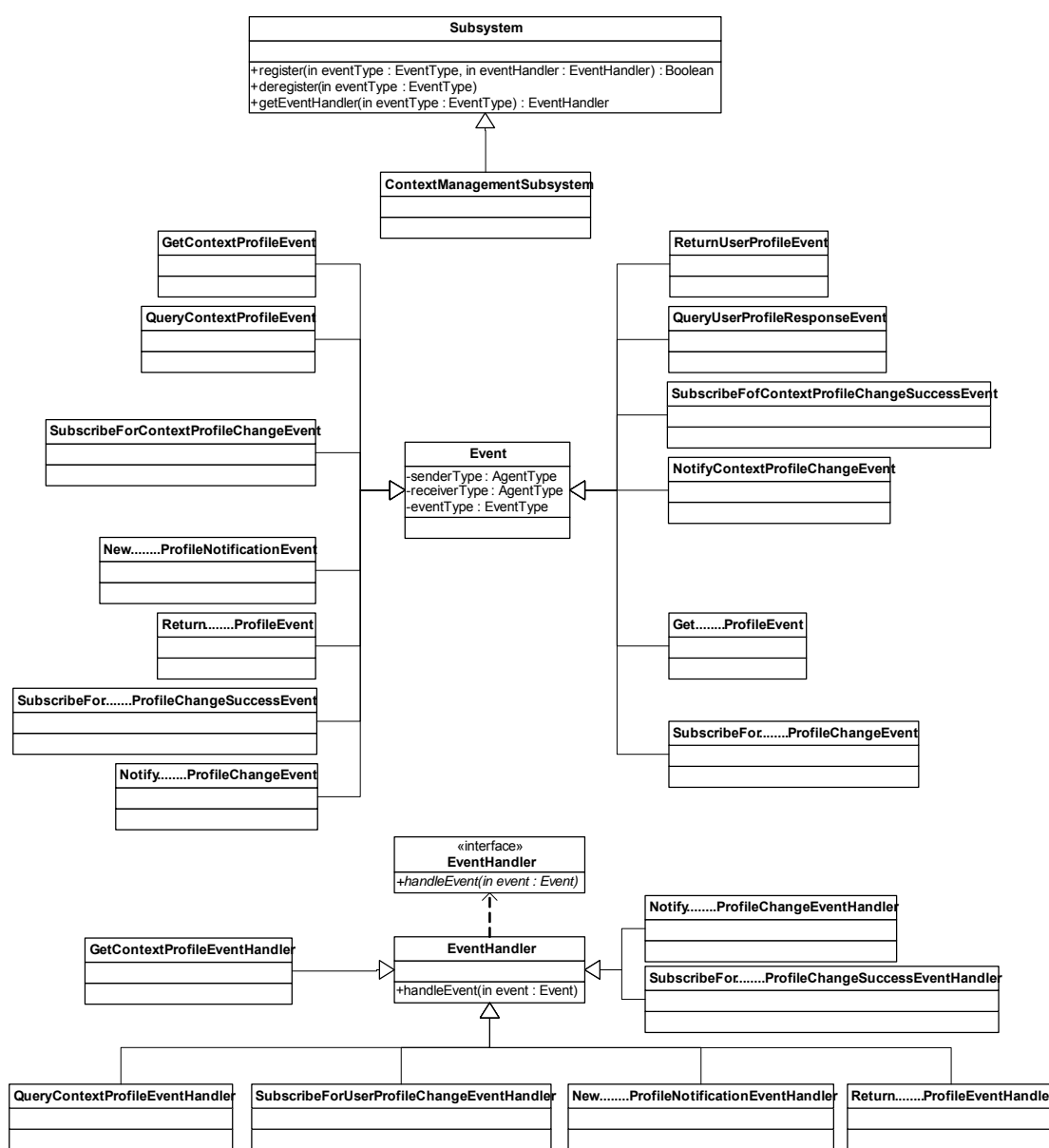
3.3.8 Υποσύστημα Διαχείρισης Πληροφορίας Πλαισίου

Όπως και το υποσύστημα πληροφορίας θέσης χρήστη έτσι και το Υποσύστημα Διαχείρισης Πληροφορίας Πλαισίου έχει υπόσταση μόνο στον Διαμεσολαβητή Δικτύου. Οι λόγοι που οδήγησαν στην επιλογή αυτή ήταν οι εξής:

- Η ύπαρξη ενός ακόμα υποσυστήματος στην πλευρά του τερματικού θα το επιβάρυνε περισσότερο όσον αφορά στις διαδικασίες επεξεργασίας. Αυτό θα είχε σαν συνέπεια

υψηλότερη κατανάλωση επεξεργαστικών πόρων γεγονός που στην περίπτωση των τερματικών με χαμηλές δυνατότητες θα ήταν κρίσιμο.

- Στην περίπτωση του Διαμεσολαβητή Δικτύου, που οι επεξεργαστικοί πόροι δύσκολα δημιουργούν προβλήματα και που υπάρχει πάντα η δυνατότητα κατανεμημένων λύσεων, επιλέχθηκε η λύση του υποσυστήματος Διαχείρισης Πληροφορίας Πλαισίου. Έτσι, το σύνολο των προφίλ ενσωματώνονται σε ένα υποσύστημα και κατά συνέπεια οι εξωτερικές οντότητες για την ανάκτηση των πληροφοριών απαιτείται να γνωρίζουν και να αλληλεπιδρούν με ένα και μοναδικό υποσύστημα ελαχιστοποιώντας έτσι ζητήματα πολυπλοκότητας των αλληλεπιδράσεων.



Σχήμα 51: Διάγραμμα κλάσεων του Υποσυστήματος Διαχείρισης Πληροφορίας Πλαισίου στον Διαμεσολαβητή Δικτύου

Όπως παρουσιάζεται στο Σχήμα 51, το Υποσύστημα Διαχείρισης Πληροφορίας Πλαισίου δέχεται αντίστοιχα γεγονότα με αυτά που δέχονται και τα άλλα υποσυστήματα διαχείρισης των διαφόρων τύπων προφίλ. Εν ολίγοις αυτά είναι τα *GetContextProfileEvent*, *QueryContextProfileEvent*, *SubscribeForContextProfileChangeEvent*, καθώς και τα αντίστοιχα γεγονότα – αποκρίσεις.

Κάθε φορά που ένα υποσύστημα διαχείρισης προφίλ του Διαμεσολαβητή Δικτύου δεχθεί μία νέα εισαγωγή ενημερώνει το Υποσύστημα Διαχείρισης Πληροφορίας Πλαισίου με αποστολή του γεγονότος *NewUserProfileNotification*, *NewDeviceProfileNotification*, *NewNetworkProfileNotification* και *NewServiceProfileNotification* αντίστοιχα. Εν συνεχεία για την ανάκτηση των νέων προφίλ από το εκάστοτε υποσύστημα αποστέλλεται από το υποσύστημα Διαχείρισης Πληροφορίας Πλαισίου γεγονός του τύπου *Get....ProfileEvent* και σαν απόκριση λαμβάνει γεγονός τύπου *Return....ProfileEvent* το οποίο κάθε φορά φέρει το αντίστοιχο προφίλ με τη μορφή XML αρχείου. Επίσης το Υποσύστημα Διαχείρισης Πληροφορίας Πλαισίου εγγράφεται και λαμβάνει τα γεγονότα ενημέρωσης αλλαγών των διαφόρων προφίλ χρησιμοποιώντας τα γεγονότα εγγραφών των υποσυστημάτων διαχείρισης των προφίλ που παρουσιάστηκαν στις προηγούμενες ενότητες.

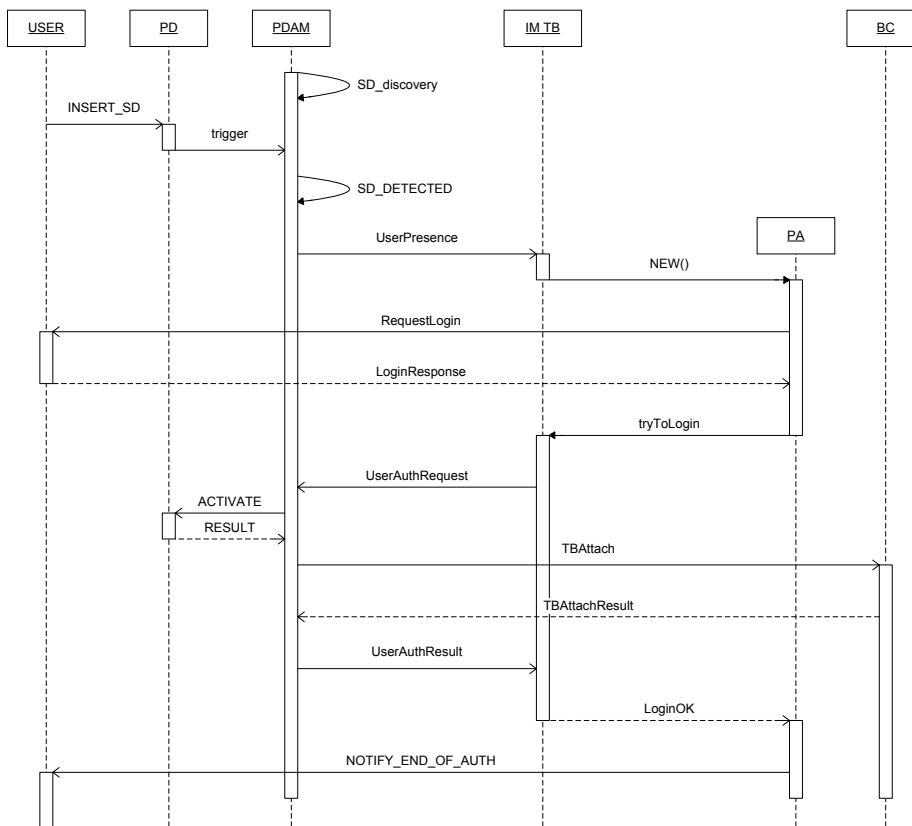
3.4 Βασικές Αλληλεπιδράσεις

Στις ενότητες που ακολουθούν παρουσιάζονται με χρήση διαγραμμάτων αλληλεπιδράσεων UML, αλλά και περιγράφονται, οι βασικότερες αλληλεπιδράσεις ανάμεσα στα υποσυστήματα των Διαμεσολαβητών Δικτύου και τερματικών, προκειμένου να αναδειχθεί η σχέση και η αλληλουχία των γεγονότων. Οι διαδικασίες που παρουσιάζονται είναι η εισαγωγή ενός νέου χρήστη στο σύστημα, η ανάκτηση των πληροφοριών προφίλ και η δημιουργία του context του νεοεισαχθέντα χρήστη, η διαδικασία ανάκτησης της λίστας των διαθέσιμων υπηρεσιών, οι διαδικασίες διαχείρισης των προφίλ και οι μηχανισμοί ενημέρωσης αλλαγών. Τέλος, παρουσιάζεται ένα γενικευμένο παράδειγμα χρήσης υπηρεσίας.

3.4.1 Διαδικασία Εισαγωγής στο σύστημα (Login Procedure)

Η διαδικασία εισαγωγής νέου χρήστη στο σύστημα ξεκινάει με την εισαγωγή της Προσωπικής Συσκευής Χρήστη στο τερματικό. Το Υποσύστημα Διαχειριστή Πρόσβασης Προσωπικής Συσκευής Χρήστη εκτελεί κυκλική διαδικασία εντοπισμού της Προσωπικής Συσκευής Χρήστη και με την εισαγωγή της στο τερματικό την εντοπίζει και στέλνει προς

τον Διαχειριστή Εξωτερικής Διεπαφής το γεγονός *UserPresence*. Με τη λήψη του γεγονότος ο Διαχειριστής Εξωτερικής Διεπαφής δημιουργεί μία νέα υπόσταση του Προσωπικού Βοηθού και στην οθόνη του τερματικού του χρήστη εμφανίζεται «παράθυρο» εισαγωγής των στοιχείων του χρήστη *username* και *password* προκειμένου να πραγματοποιηθούν η διαδικασία πιστοποίησης και εξουσιοδότησης του στο σύστημα.

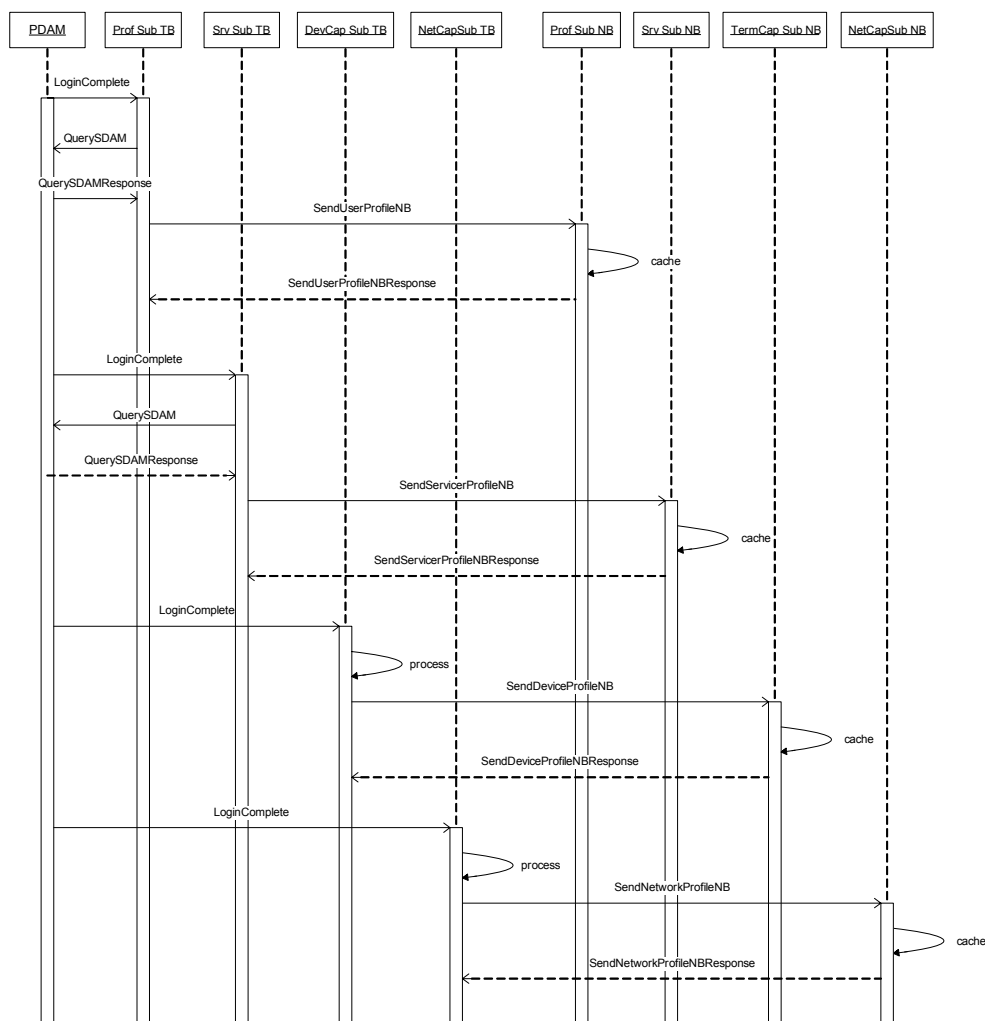


Σχήμα 52: Διάγραμμα αλληλεπιδράσεων διαδικασίας Login

Τα στοιχεία που εισάγει ο χρήστης στον Προσωπικό Βοηθό προωθούνται μέσω των υποσυστημάτων Διαχειριστή Εξωτερικής Διεπαφής και Διαχειριστή Πρόσβασης Προσωπικής Συσκευής Χρήστη στην Προσωπική Συσκευή Χρήστη. Εντός της συσκευής πραγματοποιείται έλεγχος της ορθότητας των στοιχείων που εισήγαγε ο χρήστης. Στην περίπτωση εισαγωγής ορθών στοιχείων, ο χρήστης ενημερώνεται για την επιτυχή έκβαση της διαδικασίας και παράλληλα ο Διαμεσολαβητής Δικτύου ενημερώνεται μέσω των Υποσυστημάτων Επικοινωνίας Διαμεσολαβητών για την ύπαρξη του νέου ενεργού Διαμεσολαβητή Τερματικού (*TBAttach*).

3.4.2 Διαδικασία πρώτης ανάκτησης πληροφοριών Context από τους Διαμεσολαβητές

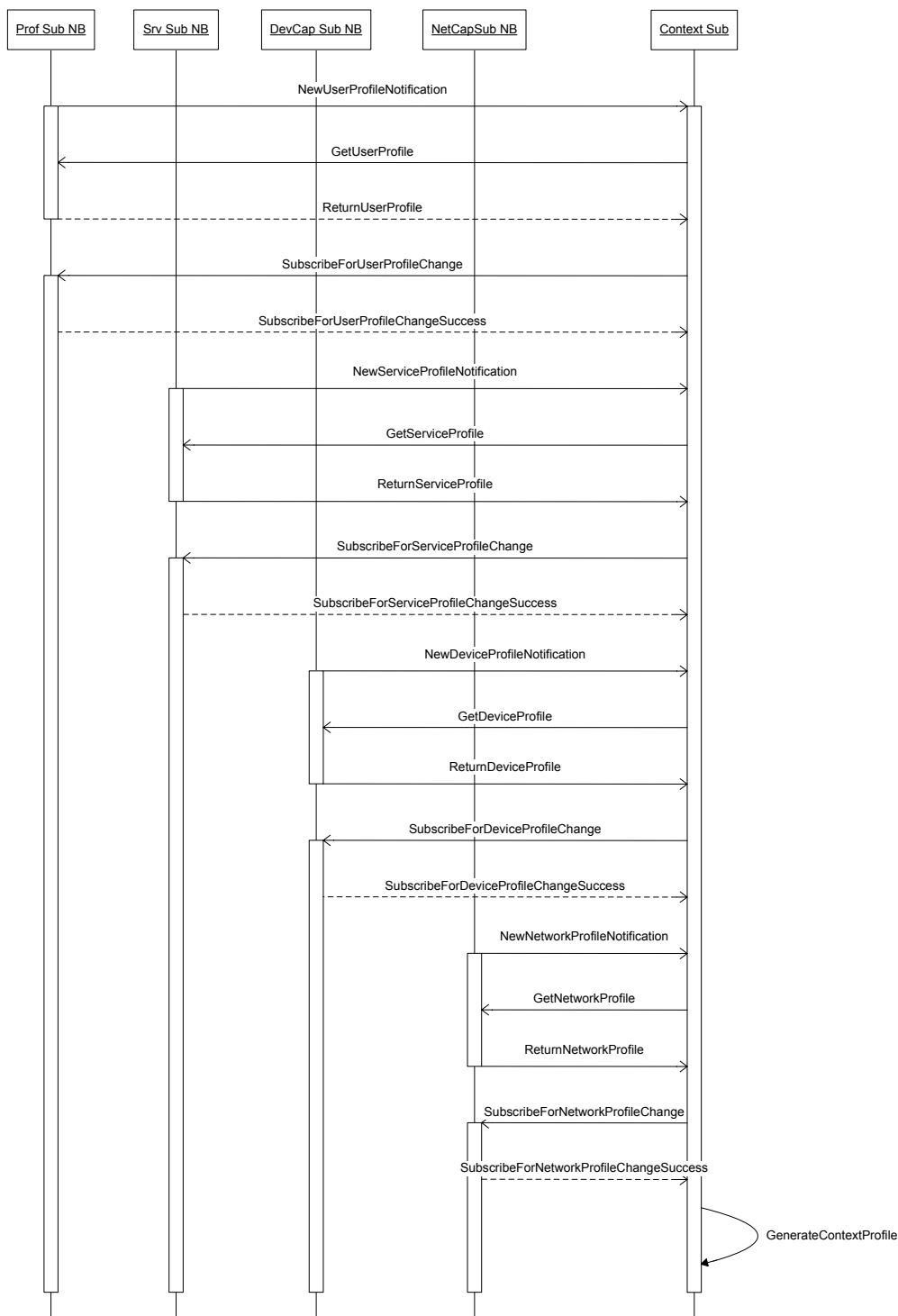
Αφού ολοκληρωθεί η διαδικασία Login, το Υποσύστημα Διαχειριστή Πρόσβασης Προσωπικής Συσκευής Χρήστη ενημερώνει όλα τα υποσυστήματα διαχείρισης προφίλ του Διαμεσολαβητή Δικτύου αποστέλλοντας στο καθένα από αυτά ένα γεγονός *LoginComplete*.



Σχήμα 53: Ανάκτηση Προφίλ από τα υποσυστήματα διαχείρισης τους στους Διαμεσολαβητές Δικτύου και Τερματικού

Εν συνεχεία κάθε ένα από τα ενήμερα πλέον υποσυστήματα αποστέλλει το γεγονός *QuerySDAM* προς το Υποσύστημα Διαχειριστή Πρόσβασης Προσωπικής Συσκευής Χρήστη για την ανάκτηση του προφίλ και το τελευταίο με τη σειρά του τα επιστρέφει μέσω του μηνύματος *QuerySDAMResponse*. Μετά τη λήψη του προφίλ κάθε υποσύστημα το αποστέλλει στο δίδυμό του υποσύστημα στον Διαμεσολαβητή Δικτύου μέσω των γεγονότων *SendUserProfileNBEvent*, *SendServiceProfileNBEvent*, *SendDeviceProfileNBEvent* και

SendNetwork ProfileNBEvent. Οι επιβεβαιώσεις λήψης των προφίλ και από την πλευρά του δικτύου έρχονται μέσω των γεγονότων *SendUserProfileNBResponseEvent*, *SendServiceProfileNBResponseEvent*, *SendDeviceProfileNBResponseEvent* και *SendNetworkProfileNBResponseEvent*.

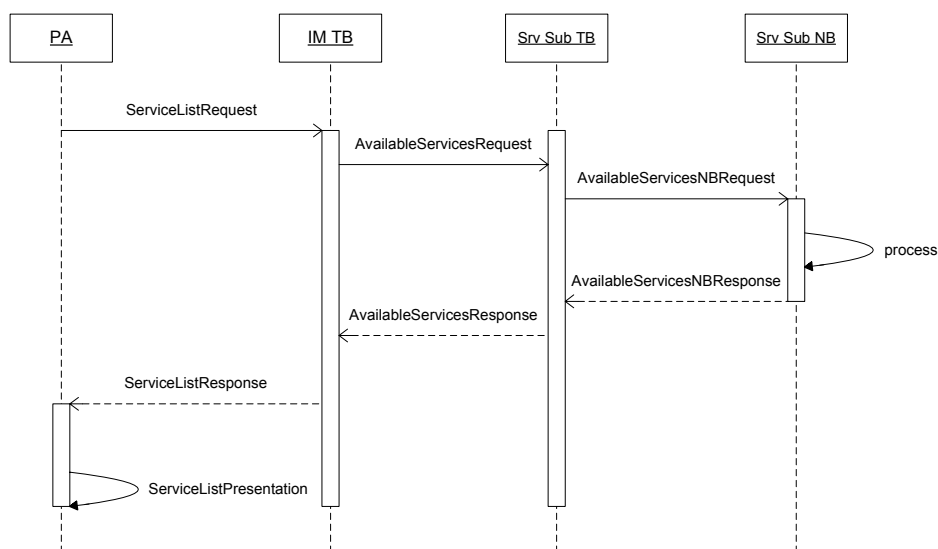


Σχήμα 54: Ανάκτηση Προφίλ από το υποσύστημα διαχείρισης πληροφορίας πλαισίου στον Διαμεσολαβητή Δικτύου

Στη συνέχεια τα υποσυστήματα διαχείρισης προφίλ του Διαμεσολαβητή Δικτύου ενημερώνουν το υποσύστημα Διαχείρισης Πληροφορίας Πλαισίου για την ύπαρξη νέων προφίλ μέσω των γεγονότων *NewUserProfileNotification*, *NewServiceProfileNotification*, *NewDeviceProfileNotification* και *NewNetworkProfileNotification*. Το υποσύστημα μέσω των γεγονότων *GetUserProfile*, *GetServiceProfile*, *GetDeviceProfile* και *GetNetworkProfile* ζητά από τα αντίστοιχα υποσυστήματα τα νέα προφίλ, και αυτά του επιστρέφονται μέσω των γεγονότων *ReturnUserProfile*, *ReturnServiceProfile*, *ReturnDeviceProfile* και *ReturnNetworkProfile*.

Καθώς το πλαίσιο χρήσης υπηρεσιών του χρήστη ανά πάσα στιγμή μπορεί να αλλάξει, το Υποσύστημα Διαχείρισης Πληροφορίας Πλαισίου είναι απαραίτητο να εγγραφεί στα τοπικά υποσυστήματα διαχείρισης προφίλ για να ενημερώνεται για τυχόν αλλαγές. Η εγγραφή πραγματοποιείται με αποστολή στα αντίστοιχα τοπικά υποσυστήματα των γεγονότων *SubscribeForUserProfileChange*, *SubscribeForServiceProfileChange*, *SubscribeForDeviceProfileChange* και *SubscribeForNetworkProfileChange*. Αφού ολοκληρωθούν όλες οι παραπάνω διαδικασίες το υποσύστημα δημιουργεί από τις επιμέρους πληροφορίες που έχει ανακτήσει το συνολικό Προφίλ Πληροφορίας Πλαισίου του χρήστη.

3.4.3 Διαδικασία Ανάκτησης Διαθέσιμων Υπηρεσιών



Σχήμα 55: Διάγραμμα αλληλεπιδράσεων της διαδικασίας ανάκτησης διαθέσιμων υπηρεσιών

Αφού ολοκληρωθεί η διαδικασία ανάκτησης των πληροφοριών πλαισίου από όλα τα υποσυστήματα τόσο στο τερματικό όσο και στο δίκτυο, δίνεται η δυνατότητα στο χρήστη να επιλέξει από τον Προσωπικό Βοηθό να ξεκινήσει η διαδικασία ανάκτησης της λίστας των

διαθέσιμων υπηρεσιών. Ο Προσωπικός Βοηθός αλληλεπιδρά με το Υποσύστημα Διαχειριστή Εξωτερικής Διεπαφής το οποίο προωθεί το αίτημα της ανάκτησης της λίστας των υπηρεσιών προς το Υποσύστημα Διαχείρισης Υπηρεσιών του Διαμεσολαβητή Τερματικού με το γεγονός *AvailableServicesRequest*. Εν συνεχεία δημιουργείται και αποστέλλεται προς το αντίστοιχο υποσύστημα του Δικτύου το γεγονός *AvailableServicesNBRequest*. Μετά από κατάλληλη επεξεργασία επιστρέφεται στον Προσωπικό Βοηθό η λίστα με τις υπηρεσίες μέσω των γεγονότων *AvailableServicesNBResponse* και *AvailableServicesResponse*. Οι παράμετροι που χαρακτηρίζουν τις υπηρεσίες είναι το όνομα και η θέση του εξυπηρετητή που βρίσκονται.

3.4.4 Μηχανισμοί Ενημέρωσης

Στην ενότητα αυτή παρουσιάζονται οι μηχανισμοί ενημέρωσης που χρησιμοποιούνται από το σύστημα προκειμένου να μπορεί να παρακολουθεί και να ελέγχει τις αλλαγές που πραγματοποιούνται στα προφίλ των χρηστών. Οι εναλλακτικές που υπάρχουν προκειμένου τα υποσυστήματα, που διατηρούν και διαχειρίζονται τα προφίλ, να έχουν αποθηκευμένη ανά πάσα στιγμή και να μπορούν να παρέχουν την πιο πρόσφατη και πραγματική έκδοση των προφίλ είναι οι παρακάτω.

- Κάθε φορά που ένα τοπικό υποσύστημα στον Διαμεσολαβητή Δικτύου ζητάει πληροφορία, τότε εκκινείται διαδικασία ανάκτησης της πληροφορίας εκ νέου από τον Διαμεσολαβητή Τερματικού. Σε αυτήν την περίπτωση η πληροφορία δεν είναι απαραίτητο να βρίσκεται αποθηκευμένη στην πλευρά του δικτύου, αλλά κάθε φορά που απαιτείται τα δίδυμα υποσυστήματα επικοινωνούν μέσω κατάλληλων γεγονότων τύπου αίτηση/απόκριση και ανταλλάσσουν τις πληροφορίες προφίλ που απαιτούνται. Η συγκεκριμένη λύση από την μία πλευρά εξασφαλίζει την ανά πάσα στιγμή εγκυρότητα της πληροφορίας, αλλά από την άλλη υπάρχει κίνδυνος επικοινωνιακής υπερφόρτωσης του συστήματος λόγω του μεγάλου αριθμού γεγονότων που θα ανταλλάσσονται ανάμεσα στα υποσυστήματα Τερματικού και Δικτύου κάθε φορά που μία υπηρεσία θα ζητάει πληροφορία από το σύστημα.
- Η δεύτερη εναλλακτική θα ήταν ο ορισμός χρονιστών, με την λήξη των οποίων τα αντίγραφα των προφίλ σε οποιοδήποτε σημείο του συστήματος θα ανανεώνονταν με χρήση κατάλληλων γεγονότων. Ο σχεδιασμός αυτός κρύβει τον παρακάτω κίνδυνο. Έστω ότι έχει πραγματοποιηθεί κάποια αλλαγή στις πληροφορίες προφίλ, ο χρονιστής δεν έχει λήξει, και κατά συνέπεια το προφίλ δεν έχει ανανεωθεί. Σε εκείνο το χρονικό διάστημα, μεταξύ αλλαγής στην κατάσταση του χρήστη και λήξης του

χρονιστή, κάποια υπηρεσία αιτείται πληροφορίας για τον συγκεκριμένο χρήστη. Το αποτέλεσμα θα είναι η υπηρεσία να λάβει σαν απάντηση στο αίτημα της, δεδομένα που δεν ανταποκρίνονται πλήρως στην πραγματική κατάσταση του χρήστη

- Η τελευταία και πιο αποτελεσματική εναλλακτική λύση είναι η χρήση μηχανισμού ανανέωσης όλων των αντιγράφων προφίλ που διατηρούνται στο σύστημα κάθε φορά που πραγματοποιείται κάποια αλλαγή. Για τον σχεδιασμό και την υλοποίηση του μηχανισμού ακολουθήθηκαν οι αρχές των publish-subscribe συστημάτων.

Η ενημέρωση των διδύμων υποσυστημάτων είναι αυτόματη διαδικασία. Κάθε φορά που πραγματοποιείται μία αλλαγή ο Διαμεσολαβητής Τερματικού αποστέλλει στα δίδυμα υποσυστήματα του Διαμεσολαβητή Δικτύου τα ανανεωμένα προφίλ μέσω των γεγονότων που αποστέλλονται και κατά την ολοκλήρωση της διαδικασίας εισαγωγής χρήστη στο σύστημα. Τα γεγονότα αυτά παρουσιάζονται στον παρακάτω πίνακα (Πίνακας 2):

Πίνακας 2: Γεγονότα ενημέρωσης διδύμων υποσυστημάτων

Υποσύστημα	Γεγονός
Υποσ. Διαχ. Προφίλ Χρήστη	SendUserProfileNB
Υποσ. Διαχ. Δυνατοτήτων Τερματικών Συσκευών	SendDeviceProfileNB
Υποσ. Διαχ. Δικτυακών Συνδέσεων	SendNetworkProfileNB
Υποσ. Διαχ. Υπηρεσιών	SendServiceProfileNB

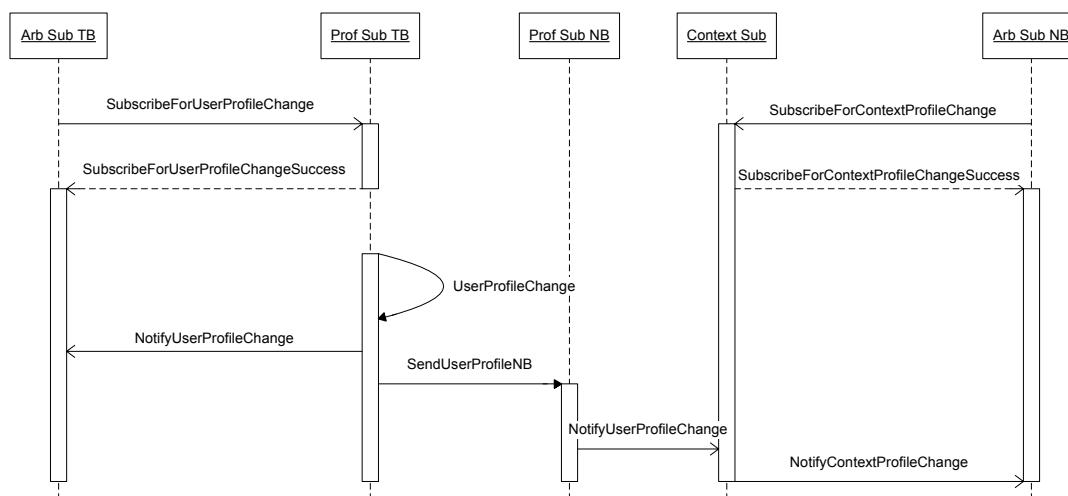
Στο σημείο αυτό πρέπει να αναφερθεί ότι στις περιπτώσεις που απαιτείται αλλαγή κάποιου προφίλ στην πλευρά του δικτύου, το υπεύθυνο υποσύστημα διαχείρισης προφίλ προωθεί το αίτημα για αλλαγή στην πλευρά του τερματικού. Εκεί πραγματοποιούνται όλες οι απαραίτητες διαδικασίες ανανέωσης και στη συνέχεια ενημερώνεται και το δίδυμο υποσύστημα στον Διαμεσολαβητή Δικτύου με αποστολή των προαναφερθέντων γεγονότων. Η διαδικασία αυτή διαφαίνεται στο Σχήμα 58 της επόμενης παραγράφου.

Εκτός των διδύμων υποσυστημάτων, υπάρχει δυνατότητα οποιοδήποτε τοπικό υποσύστημα να εγγραφεί στα υποσυστήματα διαχείρισης των προφίλ προκειμένου να ενημερώνεται για τις αλλαγές σε αυτά. Τα γεγονότα που απαιτούνται για την πραγματοποίηση των διαδικασιών εγγραφής και ενημέρωσης εμφανίζονται στον παρακάτω πίνακα (Πίνακας 3) και είναι ίδια και για τους δύο διαμεσολαβητές.

Πίνακας 3: Γεγονότα εγγραφής και ενημέρωσης τοπικών υποσυστημάτων

Υποσύστημα	Γεγονός Εγγραφής	Γεγονός Επιβεβαίωσης Εγγραφής	Γεγονός Αναφοράς Αλλαγής
Υποσ. Διαχ. Προφίλ Χρήστη	SubscribeForUserProfileChange	SubscribeForUserProfileChangeSuccess	NotifyUserProfileChange
Υποσ. Διαχ. Δυνατοτήτων Τερματικών Συσκευών	SubscribeForDeviceProfileChange	SubscribeForDeviceProfileChangeSuccess	NotifyDeviceProfileChange
Υποσ. Διαχ. Δικτυακών Συνδέσεων	SubscribeForNetworkProfileChange	SubscribeForNetworkProfileChangeSuccess	NotifyNetworkProfileChange
Υποσ. Διαχ. Υπηρεσιών	SubscribeForServiceProfileChange	SubscribeForServiceProfileChangeSuccess	NotifyServiceProfileChange
Υποσ. Διαχ. Πληροφορίας Πλαισίου	SubscribeForContextProfileChange	SubscribeForContextProfileChangeSuccess	NotifyContextProfileChange

Στο παρακάτω διάγραμμα αλληλεπιδράσεων (Σχήμα 56) παρουσιάζεται η διαδικασία εγγραφής δύο υποσυστημάτων, ενός στον Διαμεσολαβητή Δικτύου και ενός στον Διαμεσολαβητή Τερματικού, στο τοπικό υποσύστημα Διαχείρισης Προφίλ Χρήστη.



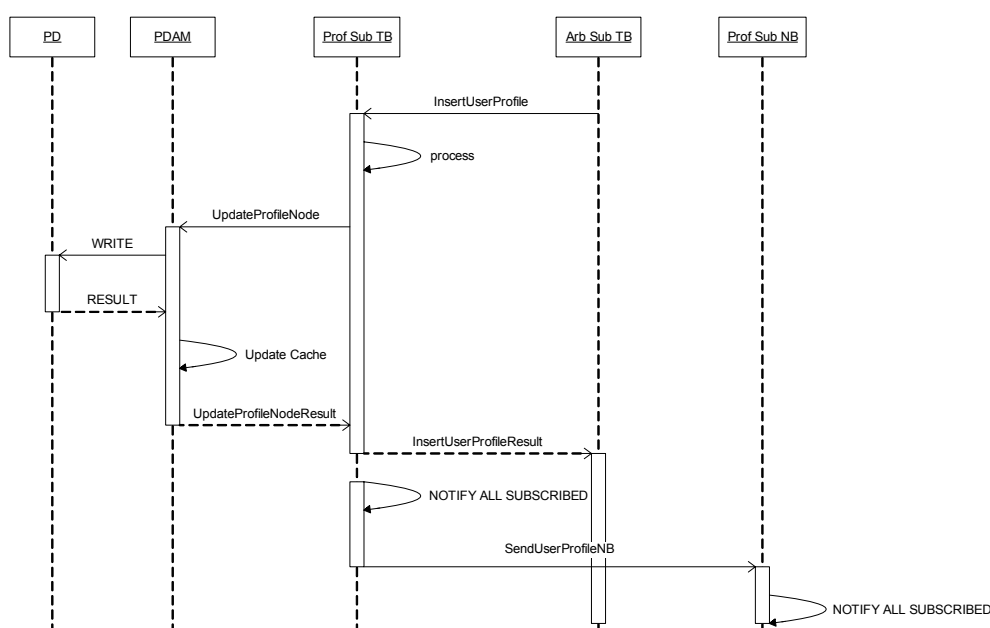
Σχήμα 56: Διάγραμμα αλληλεπιδράσεων Μηχανισμού Εγγραφής και Ενημέρωσης

Η εγγραφή πραγματοποιείται τοπικά μέσω του γεγονότος *SubscribeForUserProfileChangeEvent* και η επιβεβαίωση της εγγραφής έρχεται με τα γεγονότα *SubscribeForUserProfileChangeSuccessEvent*. Το γεγονός της εγγραφής φέρει σαν παραμέτρους τον διακριτικό αριθμό χρήστη και το μέρος του προφίλ για το οποίο το εγγραφόμενο υποσύστημα επιθυμεί να λαμβάνει ενημερωτικά γεγονότα. Όταν πραγματοποιηθεί αλλαγή στο προφίλ, το αντίστοιχο υποσύστημα διαχειριστής ενημερώνει όλα τα τοπικά εγγεγραμμένα υποσυστήματα με το γεγονός *NotifyUserProfileChangeEvent* το οποίο μεταφέρει το ανανεωμένο μέρος του προφίλ. Εν συνεχεία αποστέλλεται στο

αντίστοιχο υποσύστημα στην πλευρά του δικτύου όλο το νέο προφίλ με το γεγονός *SendUserProfileNB*. Αφού λάβει το γεγονός, το Υποσύστημα Διαχείρισης Προφίλ Χρήστη του Διαμεσολαβητή Δικτύου ενημερώνει τα τοπικά εγγεγραμμένα υποσυστήματα αποστέλλοντας το γεγονός *NotifyUserProfileChangeEvent*. Αντίστοιχες είναι και οι διαδικασίες των υπολοίπων υποσυστημάτων διαχείρισης προφίλ.

3.4.5 Διαδικασίες Διαχείρισης των Προφίλ

Στην ενότητα αυτή περιγράφονται οι διαδικασίες ανάκτησης ολόκληρων ή μέρους των προφίλ καθώς και η διαδικασίες αλλαγής και ανανέωσης αυτών.



Σχήμα 57: Διάγραμμα αλληλεπιδράσεων διαδικασίας ανανέωσης προφίλ εκκινούμενης από τυχαίο υποσύστημα στον Διαμεσολαβητή Τερματικού

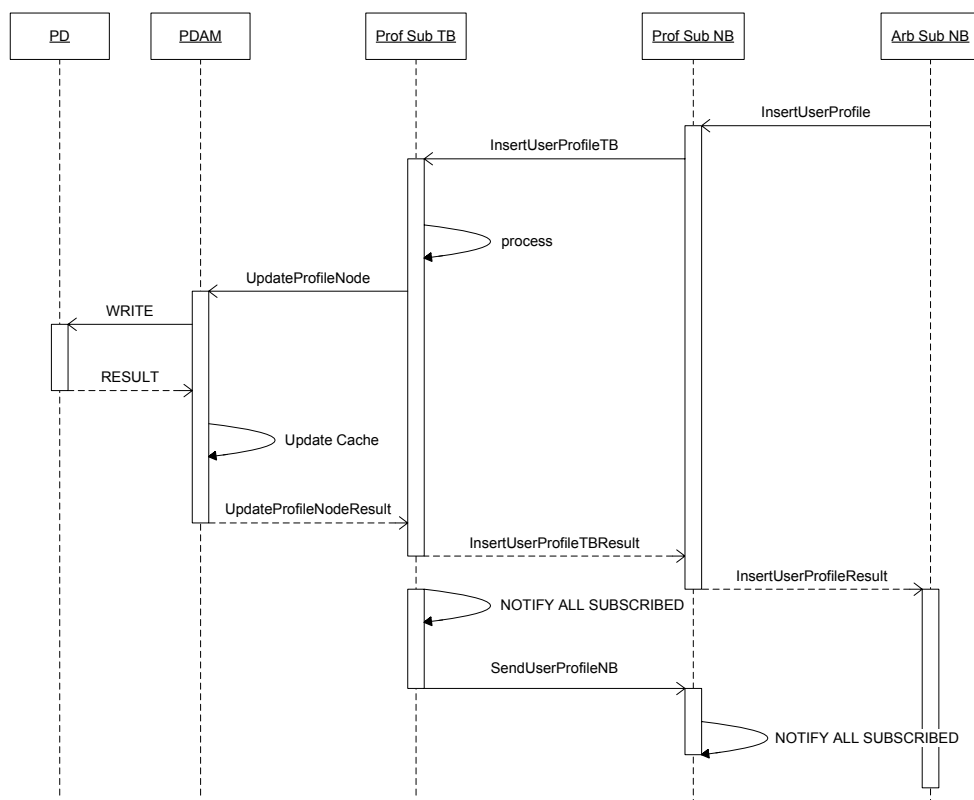
Για την ανάκτηση μέρους ή όλου του προφίλ από ένα τυχαίο τοπικό υποσύστημα από τα υποσυστήματα διαχείρισης προφίλ τόσο στον Διαμεσολαβητή Δικτύου όσο και σε αυτόν του Τερματικού χρησιμοποιούνται αντίστοιχα τα γεγονότα *QueryUserProfileEvent*, *QueryDeviceProfileEvent*, *QueryServiceProfileEvent*, *QueryNetworkProfileEvent* και *QueryContextProfileEvent* (μόνο στον Διαμεσολαβητή Δικτύου). Τα γεγονότα αυτά φέρουν ως παραμέτρους τον διακριτικό αριθμό χρήστη (UserID) και τα στοιχεία του προφίλ τα οποία ζητούνται. Το UserID ορίζεται από το string *_user* και το string *_what* καθορίζει το μέρος του προφίλ. Η μη εισαγωγή της παραμέτρου *_what* σημαίνει ότι το αιτών υποσύστημα επιθυμεί να του επιστραφεί ολόκληρο το προφίλ. Αφού ο κατάλληλος Event Handler λάβει ένα από τα παραπάνω γεγονότα ανακτά τις παραμέτρους και εν συνεχεία δημιουργεί το

κατάλληλο γεγονός απάντηση και επιστρέφει στο αιτόν υποσύστημα το ζητούμενο. Τα γεγονότα απαντήσεις είναι αντίστοιχα: *QueryUserProfileResponseEvent*, *QueryDeviceProfileResponseEvent*, *QueryServiceProfileResponseEvent*, *QueryNetworkProfileResponseEvent* και *QueryContextProfileResponseEvent*.

Για την περίπτωση που ένα υποσύστημα στον Διαμεσολαβητή Δικτύου επιθυμεί να ανανεώσει μέρος κάποιου προφίλ σε ένα από τα τοπικά υποσυστήματα διαχείρισης προφίλ πρέπει να δημιουργήσει και να αποστείλει ένα από τα παρακάτω γεγονότα *InsertUserProfileEvent*, *InsertDeviceProfileEvent*, *InsertServiceProfileEvent* και *InsertNetworkProfileEvent* αντίστοιχα. Οι παράμετροι που φέρουν τα γεγονότα αυτού του τύπου είναι ο διακριτικός αριθμός χρήστη, το μέρος του προφίλ που πρέπει να ανανεωθεί και τα νέα δεδομένα τα οποία πρέπει να εισαχθούν στο προφίλ.

Ο κατάλληλος Event Handler ανακτά τις παραμέτρους, δημιουργεί ένα νέο γεγονός τύπου *Insert...ProfileResultEvent* και καλεί κατάλληλη μέθοδο με τις προαναφερθείσες παραμέτρους. Η μέθοδος δημιουργεί το γεγονός *UpdateProfileNodeEvent*, με παραμέτρους το πεδίο του προφίλ που πρέπει να αλλάξει και την τιμή που πρέπει να πάρει, και το στέλνει στο Υποσύστημα Διαχειριστή Πρόσβασης Προσωπικής Συσκευής Χρήστη. Εν συνεχεία η μέθοδος περιμένει να λάβει την απάντηση. Αφού πραγματοποιηθεί η αλλαγή του προφίλ εντός της Προσωπικής Συσκευής Χρήστη, επιστρέφεται το γεγονός *UpdateProfileNodeResultEvent*. Η παράμετρος που φέρει το συγκεκριμένο γεγονός ανακτάται και ενσωματώνεται στο γεγονός *Insert...ProfileResultEvent* το οποίο επιστρέφεται στο αρχικά αιτόν την αλλαγή υποσύστημα ενημερώνοντας το για την επιτυχή ή όχι έκβαση της διαδικασίας αλλαγής. Εν συνεχεία το υποσύστημα διαχείρισης προφίλ μέσω του οποίου πραγματοποιήθηκε η αλλαγή ενημερώνει τα τοπικά εγγεγραμμένα υποσυστήματα για την αλλαγή μέσω γεγονότων τύπου *Notify...ProfileChange* και ενημερώνει το δίδυμό του υποσύστημα στην πλευρά του Δικτύου με αποστολή γεγονότος τύπου *Send...ProfileNB*. Τέλος, το δίδυμο υποσύστημα ενημερώνει τα τοπικά εγγεγραμμένα υποσυστήματα για την αλλαγή μέσω γεγονότων τύπου *Notify...ProfileChange*.

Η ίδια διαδικασία και τα ίδια γεγονότα χρησιμοποιούνται και στην περίπτωση που πρόκειται για διαγραφή δεδομένων από το προφίλ. Στην περίπτωση αυτή, η παράμετρος του *Insert...ProfileEvent*, που αντιπροσωπεύει τα νέα δεδομένα, έχει κατάλληλη μηδενική τιμή (null).

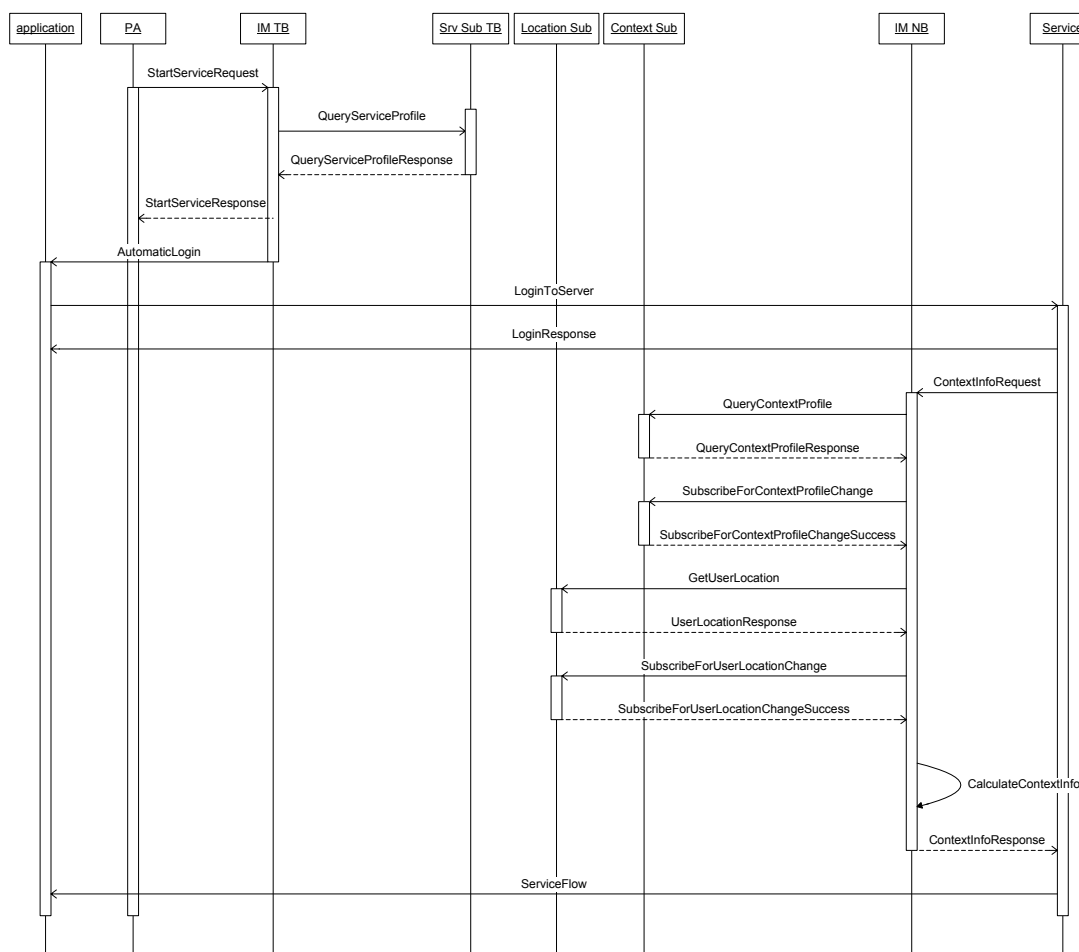


Σχήμα 58: Διάγραμμα αλληλεπιδράσεων διαδικασίας ανανέωσης προφίλ εκκινούμενης από τυχαίο υποσύστημα στον Διαμεσολαβητή Δικτύου

Στην περίπτωση που η αλλαγή σε κάποιο προφίλ αποτελεί πρωτοβουλία τοπικού τυχαίου υποσυστήματος στο Διαμεσολαβητή Δικτύου αποστέλλει και πάλι προς το κατάλληλο υποσύστημα διαχείρισης το γεγονός *Insert...ProfileEvent*. Εν συνεχεία το υποσύστημα διαχείρισης προωθεί το αίτημα στο δίδυμό του υποσύστημα στην πλευρά του τερματικού με το γεγονός *Insert...ProfileTBEvent*. Η διαδικασία που ακολουθείται στη συνέχεια είναι ακριβώς η ίδια.

3.4.6 Γενικευμένο Παράδειγμα Χρήσης Υπηρεσίας

Στο διάγραμμα αλληλεπιδράσεων που παρουσιάζεται στο Σχήμα 59 περιγράφονται οι αλληλεπιδράσεις που λαμβάνουν χώρα για την ανάκτηση πληροφοριών πλαισίου χρήσης από μία υπηρεσία.



Σχήμα 59: Διάγραμμα αλληλεπιδράσεων Γενικευμένου Παραδείγματος Χρήσης Υπηρεσίας

Η υπηρεσία αλληλεπιδρά με το σύστημα μέσω του Διαχειριστή Εξωτερικής Διεπαφής (Interface Manager, IM NB) που έχει υλοποιηθεί και εγκατασταθεί στο σύστημα για το σκοπό αυτό. Η διαδικασία εκκινείται από τον χρήστη μέσω του Προσωπικού Βοηθού (PA) όταν ο χρήστης επιλέξει την υπηρεσία από κατάλληλη γραφική διεπαφή. Η αίτηση για την υπηρεσία προωθείται στον Διαχειριστή Εξωτερικής Διεπαφής του Διαμεσολαβητή Τερματικού (Interface Manager, IM TB), ο οποίος με τη σειρά του αποστέλλει στο Υποσύστημα Διαχείρισης Υπηρεσιών το γεγονός *QueryServiceProfile* ζητώντας τα σχετικά με την υπηρεσία username και password του χρήστη καθώς και τη τοποθεσία του εξυπηρετητή της υπηρεσίας στο δίκτυο. Με κατάλληλο γεγονός *QueryServiceProfileResponse*, επιστρέφονται τα ζητούμενα στοιχεία και το Υποσύστημα Εξωτερικής Διεπαφής εκκινεί την εφαρμογή μέσω της οποίας ο χρήστης θα χρησιμοποιήσει την υπηρεσία (π.χ. φυλλομετρητή διαδικτύου, web browser) και προωθεί προς την υπηρεσία αίτημα για την είσοδο του χρήστη αποστέλλοντας το username και το password (*LoginToServer*). Μόλις ολοκληρωθεί επιτυχώς η διαδικασία εισόδου του χρήστη στην

υπηρεσία, η υπηρεσία μέσω του Διαχειριστή Εξωτερικής Διεπαφής (*ContextInfoRequest*), ξεκινάει την διαδικασία ανάκτησης του context. Ο Διαχειριστής Εξωτερικής Διεπαφής επικοινωνεί μέσω των γεγονότων *QueryContextProfile* και *GetUserLocation* με τα υποσυστήματα διαχείρισης Πληροφορίας Πλαισίου και Πληροφορίας Θέσης Χρήστη αντίστοιχα και ανακτά μέσω των αντίστοιχων γεγονότων-απαντήσεων την πληροφορία που απαιτείται για την εκτέλεση της υπηρεσίας. Συγχρόνως ο Διαχειριστής Εξωτερικής Διεπαφής της υπηρεσίας, ακολουθώντας τις διαδικασίες εγγραφής για ενημέρωση σε περίπτωση αλλαγών που περιγράφησαν σε προηγούμενες ενότητες, εγγράφεται στα δύο υποσυστήματα. Και στη συνέχεια πραγματοποιούνται όλες οι απαραίτητες αλληλεπιδράσεις για την εκτέλεση της υπηρεσίας (service flow).

3.5 Παρεχόμενες Υπηρεσίες

Με την ολοκλήρωση του σχεδιασμού και της υλοποίησης των βασικών λειτουργιών (core functionality) του συστήματος σχεδιάστηκε και υλοποιήθηκε ένα σύνολο από υπηρεσίες προκειμένου να ελεγχθεί και να αποδειχθεί η ορθή και αποτελεσματική λειτουργία του συστήματος [4]. Κάποιες από τις υπηρεσίες προϋπήρχαν της αρχιτεκτονικής και διαμορφώθηκαν κατάλληλα προκειμένου να εκμεταλλευτούν τις παρεχόμενες από το σύστημα λειτουργίες, ενώ οι υπόλοιπες σχεδιάστηκαν και υλοποιήθηκαν εξ' αρχής για το συγκεκριμένο σύστημα.

Στην δεύτερη κατηγορία υπηρεσιών ανήκουν οι εξής:

- **Υπηρεσία Προσαρμοζόμενης Ανταλλαγής Μηνυμάτων Πολυμέσων (Adaptive Multimedia Messaging Service)** [12]. Το σύστημα παρέχει στην υπηρεσία όλη την απαραίτητη πληροφορία προκειμένου, ανάλογα με το είδος της πληροφορίας (κείμενο, στατική εικόνα, αρχεία ήχου και κινούμενης εικόνας κλπ.) που ανταλλάσσεται ανάμεσα σε δύο χρήστες που την χρησιμοποιούν και τις δυνατότητες των τερματικών τους, να προσαρμόζεται κατάλληλα όσον αφορά στην ποιότητα, τα χρώματα και το μέγεθος των αρχείων.
- **Υπηρεσία Ροής Πολυμέσων – Οικιακό Σύστημα Διασκέδασης (Media Streaming – Home Entertainment System)** [13]. Η Υπηρεσία αυτή λειτουργεί επικουρικά προς τους χρήστες στις αλληλεπιδράσεις τους με διαδικτυακή πύλη (web portal) πώλησης πολυμέσων (αρχεία μουσικής, ταινίες κλπ.) αυτοματοποιώντας τις διαδικασίες επιλογής και αγοράς αυτών, βασιζόμενη στις πληροφορίες που ανακτά από το σύστημα. Επίσης παρέχει αυτόματη πληρωμή και αποθήκευση των αποκτηθέντων πολυμέσων σε κατάλληλους προσωπικούς αποθηκευτικούς χώρους

και τέλος, προσαρμόζει τη ροή των πολυμέσων προς το χρήστη ανάλογα με τις δυνατότητες των τερματικών, των δικτύων αλλά και των προτιμήσεων του.

- **Υπηρεσία MyPC** [4]. Η υπηρεσία αυτή επιτρέπει την προσαρμογή του υπολογιστικού περιβάλλοντος κοινόχρηστων τερματικών βάσει του χρήστη που τα χρησιμοποιεί. Έτσι οι χρήστες έχουν την δυνατότητα να δουλεύουν σε περιβάλλον παρόμοιο με αυτό των προσωπικών τους τερματικών και όπου κι αν βρίσκονται ανεξαρτήτως τερματικού να βρίσκουν τις ίδιες ρυθμίσεις και τα ίδια δεδομένα. Για παράδειγμα ο χρήστης σε οποιοδήποτε τερματικό κι αν βρεθεί με την εισαγωγή της Προσωπικής Συσκευής Χρήστη σε αυτό αυτομάτως ανακτά τη λίστα με τις επαφές του (address book), τα αγαπημένα του links (favourites), το wallpaper κλπ. Η υπηρεσία πραγματοποιεί την προσαρμογή των τερματικών βάσει των πληροφοριών προφίλ που ανακτά από το προτεινόμενο σύστημα αλλά και σε συνεργασία με δικτυακή υπηρεσία αποθήκευσης των δεδομένων του χρήστη.
- **Υπηρεσία Αυτόματου Συμπληρώματος Φορμών** [4]. Η υπηρεσία αυτή παρέχει το αυτόματο συμπλήρωμα φορμών που εμφανίζονται σε ιστοσελίδες, βάσει δεδομένων που ανακτά από τα προφίλ που διαχειρίζεται το σύστημα. Ανάλογα με το επίπεδο αβεβαιότητας το κείμενο που τοποθετείται στο πεδίο χρωματίζεται και με διαφορετικό χρώμα. Υλοποιείται σαν plug-in εφαρμογή στην εφαρμογή του φυλλομετρητή διαδικτύου (web browser).
- **Υπηρεσία Παρακολούθησης Πανεπιστημιακού Δικτύου (Campus Network Monitoring)** [14][15]. Η υπηρεσία αυτή εκμεταλλεύεται τις πληροφορίες που παρέχει το σύστημα προκειμένου να παρακολουθεί και να διαχειρίζεται το φορτίο του ασύρματου δικτύου μίας πανεπιστημιούπολης. Ο στόχος είναι Η υπηρεσία πραγματοποιεί έλεγχο πρόσβασης και εξισορρόπηση φόρτου βασισμένα στο είδος των χρηστών. Οι «πρώτης προτεραιότητας» χρήστες έχουν τη δυνατότητα χρήσης όλων των σημείων πρόσβασης, ενώ οι χρήστες χαμηλότερης προτεραιότητας έχουν δικαίωμα να συνδέονται από συγκεκριμένα σημεία πρόσβασης και μόνο εφόσον το φορτίο τους είναι χαμηλό. Επίσης υπάρχουν αντίστοιχοι περιορισμοί και όσον αφορά την πρόσβαση στο διαδίκτυο (web browsing normal/limited mode). Τέλος οι υψηλής προτεραιότητας χρήστες απολαμβάνουν και υψηλότερη ποιότητα υπηρεσιών .

Οι υπηρεσίες που περιγράφονται στη συνέχεια αποτελούν παραδείγματα υπαρχόντων υπηρεσιών, οι οποίες διαμορφώθηκαν κατάλληλα προκειμένου να μπορούν να εκμεταλλευτούν την παρεχόμενη από το σύστημα υποδομή και πληροφορία.

- **Υπηρεσία Καθοδήγησης Τουριστών (Tour Guide System)** [4]. Αποτελεί υπηρεσία καθοδήγησης για τους επισκέπτες της πόλης Λάνκαστερ. Οι επισκέπτες παραλαμβάνουν ένα τερματικό το οποίο έχει εγκατεστημένο το λογισμικό της υπηρεσίας. Μέσω αυτού λαμβάνουν πληροφορίες σχετικά με τα αξιοθέατα, έχουν στη διάθεση τους χάρτη της περιοχής και τους δίνεται η δυνατότητα να δημιουργήσουν το δικό τους πρόγραμμα επίσκεψης στην πόλη βάση των ενδιαφερόντων και προτιμήσεων τους. Η προηγούμενη έκδοση της υπηρεσίας αυτής απαιτούσε από τους χρήστες να εισαγάγουν όλα τα απαιτούμενα στοιχεία προκειμένου οι υπηρεσίες να προσαρμοσθούν στις εκάστοτε ανάγκες του χρήστη. Στην πρόσφατη έκδοση, με την εισαγωγή της Προσωπικής Συσκευής Χρήστη στο τερματικό, όλη η απαραίτητη πληροφορία ανακτάται από το σύστημα.
- **Υπηρεσίες Αναζήτησης βάσει Θέσης και Χαρακτηριστικών του Χρήστη.** Τέλος, ενσωματώθηκε στο σύστημα και μία από τις υπηρεσίες της πλατφόρμας που παρουσιάστηκε στο πρώτο κεφάλαιο. Η υπηρεσία που ενσωματώθηκε ήταν η υπηρεσία αναζήτησης σημείων ενδιαφέροντος του χρήστη βάσει θέσης και προτιμήσεων. Η λειτουργικότητα που παρέχει η υπηρεσία είναι ακριβώς η ίδια που περιγράφεται στο πρώτο κεφάλαιο. Η μόνη διαφορά είναι ότι πλέον όλες οι πληροφορίες που απαιτούνται για την εκτέλεση της ανακτώνται από το προτεινόμενο σύστημα.

Στο επόμενο κεφάλαιο παρατίθεται μία συνολική περιγραφή της υπηρεσίας αυτής και των αλλαγών που πραγματοποιήθηκαν προκειμένου να παρέχεται πάνω από το προτεινόμενο σύστημα. Τέλος πραγματοποιήθηκε μία συγκριτική μελέτη ανάμεσα στις δύο διαφορετικές υλοποιήσεις της υπηρεσίας προκειμένου να εξαχθούν συμπεράσματα για την αποτίμηση του προτεινόμενου συστήματος.

3.6 Αναφορές

- [1] N. Blefari Melazzi, G. Bianchi, G. Cortese, S. Kapellaki, K. Kawamura, C. Noda, S. Salsano, I. S. Venieris, “The Simplicity project: easing the burden of using complex and heterogeneous ICT devices and services: Part I: Overall Architecture”, In Proc. of 13th IST Mobile & Wireless Communications Summit, Volume 1, pp 282-286, Lyon, June 2004.
- [2] R. Seidl, F. Berger, S. Kapellaki, T. Frantti, E. Rukzio, J. Hamard, N. Blefari Melazzi, “The Simplicity Project: Personalized and simplified communication spaces for mobile users”, In Proc. of 13th IST Mobile & Wireless Communications Summit, Volume 2, pp 1117-1121, Lyon, June 2004.
- [3] N. Blefari Melazzi, S. Salsano, G. Bartolomeo, F. Martire, E. Fischer, C. Meyer, C. Niedermeier,

- R. Seidl, E. Rukzio, E. Koutsoloukas, J. Papanis, I. S. Venieris: "The Simplicity System Architecture", 14th IST Summit, 19-23 June 2005, Dresden, Germany.
- [4] G. Bartolomeo, N. Blefari Melazzi, S. Kapellaki, E. Koutsoloukas, F. Martire, G. N. Prezerakos, S. Salsano, N. D. Tselikas, I. S. Venieris, "The Simplicity project and its demonstrator: improving ease of use and personalization of ICT services", In Proc. of the IEEE Globecom 2006 Conference, November 2006, San Francisco, California, USA.
- [5] Eugster, P. T., Felber, P. A., Guerraoui, R., and Kermarrec, A. 2003. The many faces of publish/subscribe. *ACM Comput. Surv.* 35, 2 (Jun. 2003), 114-131. DOI=<http://doi.acm.org/10.1145/857076.857078>
- [6] Nikolaos Dellas, Sofia Kapellaki, Eleftherios Koutsoloukas, George N. Prezerakos, Iakovos S. Venieris, "Performance Evaluation of a P2P overlay brokerage network", In Proc. of 2nd International Workshop on 'Next Generation Networking Middleware', NGNM 2005, co-located with Networking 2005, Waterloo Ontario, Canada, May 2005.
- [7] Nikolaos Dellas, Eleftherios Koutsoloukas, Sofia Kapellaki, Nikolaos Tselikas, John Papanis, George N. Prezerakos and Iakovos S. Venieris, "Location and Interests-aware Overlay Topology Adaptation Mechanism for Unstructured Peer-to-Peer Systems" submitted to *Computer Communication Journal* (Elsevier).
- [8] E. Koutsoloukas, S. Kapellaki, G.N. Prezerakos, N. Tselikas, I.S. Venieris, D. I. Kaklamani, "Asynchronous event-based communication across Service Brokers", In Proc. of the 12th European Wireless Conference, EW2006, Session B8, Paper B8.4, Athens, Greece, April 2006.
- [9] E. A. Koutsoloukas, S. H. Kapellaki, N. D. Tselikas, N. L. Dellas, J. P. Papanis, G. N. Prezerakos, I. S. Venieris, "An event based communication middleware for personalized service provision: design & evaluation", In Proc. of 3rd International Workshop on 'Next Generation Networking Middleware', NGNM 2006, co-located with Networking 2006, Coimbra, Portugal, May 2006.
- [10] J. Papanis, S. Kapellaki, E. Koutsoloukas, N. Dellas, G.N. Prezerakos, I.S. Venieris, "Facilitating context-awareness through hardware personalization devices: The Simplicity Device", In Proc. of 2nd International Workshop on Mobility Aware Technologies and Applications, MATA 2005, LNCS 3744, pp. 252-262, Springer-Verlag Berlin Heidelberg 2005, Montreal, Canada, October 2005.
- [11] kSOAP, www.ksoap.org
- [12] Elisabeth-Anna Fischer, Carsten Meyer, Christoph Niedermeier, Robert Seidl, Sofia Kapellaki, George Prezerakos, "Realizing Simplicity: Ambient Aware Service Adaptation", In Proc. of 14th IST Mobile & Wireless Communications Summit, Dresden, June 2005.
- [13] E. Koutsoloukas, S. Kapellaki, J. Papanis, N. Dellas, N. Tselikas, D. I. Kaklamani, I. Venieris, "Specification of context-aware mobile services over a distributed brokerage framework: The ubiquitous media Streaming example", In Proc. of Workshop on Information and

Communication Technologies, SoftCOM 2004, pp. 26-30, Split, October 2004.

[14] N. Blefari-Melazzi, D. Di Sorte, M. Femminella, G. Reali: "Access Network Control within the Simplicity Brokerage Framework", 14th IST Summit, 19-23 June 2005, Dresden, Germany.

[15] N. Blefari-Melazzi, D. Di Sorte, M. Femminella, G. Reali: "Toward an autonomic control of wireless access networks". IEEE GLOBECOM 2005, 28 November-2 December 2005, St. Louis, USA.

4ο Κεφάλαιο

Αποτίμηση του συστήματος

Το κεφάλαιο αυτό αρχικά αναλύει και περιγράφει την ανάπτυξη της ήδη υπάρχουσας εφαρμογής αναζήτησης Infokiosk επάνω από το προτεινόμενο σύστημα, καθώς και τις αλλαγές και προσθήκες που πραγματοποιήθηκαν για αυτό το εγχείρημα. Εν συνέχεια, παρουσιάζεται μία αποτίμηση απόδοσης του συστήματος κατά την παροχή της συγκεκριμένης υπηρεσίας, η οποία απαιτεί σαν κύρια πληροφορία πλαισίου χρήσης την πληροφορία της θέσης του χρήστη. Η αποτίμηση πραγματοποιήθηκε σε σχέση με την υλοποίηση της υπηρεσίας με χρήση της διεπαφής Κινητικότητας του OSA/Parlay με χρήση τριών διαφορετικών τεχνολογιών μεσισμικού κατανεμημένων αντικειμένων. Η απόδοση των προτεινόμενων λύσεων συγκρίνεται και με την αντίστοιχη της τρέχουσας υλοποίησης που προτείνεται από τον οργανισμό προτυποποίησης της 3GPP.

4.1 Εισαγωγή

Για την αποτίμηση του συστήματος και την απόδειξη της ορθής λειτουργίας του σχεδιάστηκε και υλοποιήθηκε ένα σύνολο από υπηρεσίες, οι οποίες αναπτύχθηκαν επάνω από την προτεινόμενη αρχιτεκτονική. Κάποιες από αυτές σχεδιάστηκαν και υλοποιήθηκαν εξ' αρχής και κάποιες αποτελούν παλαιότερες υπηρεσίες οι οποίες διαμορφώθηκαν κατάλληλα. Όπως ήδη αναφέρθηκε και στο τέλος του προηγούμενου κεφαλαίου, μία από τις ήδη υπάρχουσες υπηρεσίες που διαμορφώθηκαν ήταν και η Υπηρεσία Αναζήτησης Σημείων Ενδιαφέροντος (InfoKioskService), που αναπτύχθηκε στο πλαίσιο της υλοποίησης της Πλατφόρμας Παροχής Υπηρεσιών Αναζήτησης βάσει Θέσης και Χαρακτηριστικών του Χρήστη που περιγράφηκε στο πρώτο κεφάλαιο.

Δύο ήταν οι βασικοί λόγοι που επιλέχθηκε η συγκεκριμένη υπηρεσία. Ο πρώτος λόγος ήταν ότι με την μεταφορά μίας ήδη υπάρχουσας υπηρεσίας πάνω από το προτεινόμενο σύστημα, καθίσταται δυνατή η εξαγωγή συμπερασμάτων σχετικά με το βαθμό ευκολίας που παρέχει το συγκεκριμένο σύστημα ως προς την ενσωμάτωση παλαιότερων υπηρεσιών. Κατά συνέπεια, ελέγχεται κατά πόσο ο αρχικός σχεδιαστικός στόχος της ανοικτότητας και της ευελιξίας επιτεύχθηκε. Ο δεύτερος λόγος, επίσης πολύ βασικός, που σχεδόν επέβαλλε μία τέτοια επιλογή, ήταν ότι αυτομάτως θα υπήρχαν δύο εναλλακτικές υλοποιήσεις της ίδιας υπηρεσίας και αυτό θα καθιστούσε δυνατή την αποτίμηση του προτεινόμενου συστήματος σε σχέση με άλλες προσεγγίσεις. Με λίγα λόγια αυτή η προσέγγιση δημιουργούσε απευθείας ένα πολύ ισχυρό μέτρο σύγκρισης.

Στο παρόν κεφάλαιο παρατίθεται η περιγραφή των προσαρμογών που πραγματοποιήθηκαν στην Υπηρεσία Αναζήτησης Σημείων Ενδιαφέροντος κατά τη μεταφορά της στο προτεινόμενο σύστημα. Εν συνεχεία, ακολουθεί η σύγκριση των διαφορετικών προσεγγίσεων υλοποίησης της υπηρεσίας. Παρουσιάζεται το σύστημα πάνω στο οποίο διενεργήθηκαν οι μετρήσεις και τέλος παρατίθεται ο σχολιασμός των αποτελεσμάτων και τα εξαγόμενα συμπεράσματα.

Οι προσεγγίσεις που συγκρίνονται είναι η ανάκτηση της πληροφορίας θέσης χρήστη, αρχικά με τις μεθόδους που περιγράφονται στο πρώτο κεφάλαιο και τελικά μέσω της προτεινόμενης αρχιτεκτονικής. Η επιλογή, οι μετρήσεις να περιοριστούν μόνο στην ανάκτηση της πληροφορίας θέσης του χρήστη και όχι στην ανάκτηση και άλλου είδους πληροφοριών από το σύστημα οφείλεται στους λόγους που ακολουθούν.

Αρχικά θεωρούμε τη θέση του χρήστη σαν το πιο ευαίσθητο στοιχείο της πληροφορίας πλαισίου χρήσης μίας υπηρεσίας. Αυτό οφείλεται στο γεγονός ότι η θέση του χρήστη, στην

μεγάλη πλειοψηφία των περιπτώσεων, αλλάζει συνεχώς και πάρα πολύ συχνά, ειδικά στην περίπτωση των κινητών χρηστών. Συνεπώς, για την ανάκτηση και την παροχή αυτής της πληροφορίας στην υπηρεσία απαιτείται ταχύτητα. Οποιαδήποτε καθυστέρηση θα συνεπαγόταν λήψη λανθασμένων στοιχείων από την υπηρεσία και κατά συνέπεια το αποτέλεσμα της να μην ανταποκρίνεται στην τρέχουσα θέση του χρήστη άλλα σε κάποια που είχε βρεθεί νωρίτερα.

Άλλος ένα λόγος για την πραγματοποίηση μετρήσεων σχετικά με την ανάκτηση της θέσης είναι το γεγονός ότι η πληροφορία αυτού του τύπου είναι η μοναδική σε όλο το υλοποιημένο σύστημα που δεν ανακτάται από εγγενείς οντότητες του συστήματος αλλά από μία εξωτερική οντότητα, τον Εξυπηρετητή Θέσης ενός δικτύου κινητών επικοινωνιών. Συνεπώς, εύκολα συμπεραίνεται ότι και μόνο η επικοινωνία του Υποσυστήματος Διαχείρισης Πληροφορίας Θέσης Χρήστη με την εξωτερική οντότητα προσθέτει στην όλη διεργασία χρονικό αλλά και επεξεργαστικό κόστος, γεγονός που επιβάλλει την μελέτη και διερεύνηση του. Επίσης, σχεδόν με απόλυτη βεβαιότητα μπορεί να ειπωθεί ότι η ανάκτηση οποιασδήποτε άλλης πληροφορίας από εσωτερικές οντότητες του συστήματος, που τοπολογικά βρίσκονται στο ίδιο σημείο, ή ακόμα και στο ίδιο μηχάνημα, δεν είναι δυνατόν να καθυστερεί περισσότερο από την ανάκτηση πληροφορίας που απαιτεί την αλληλεπίδραση με εξωτερική οντότητα.

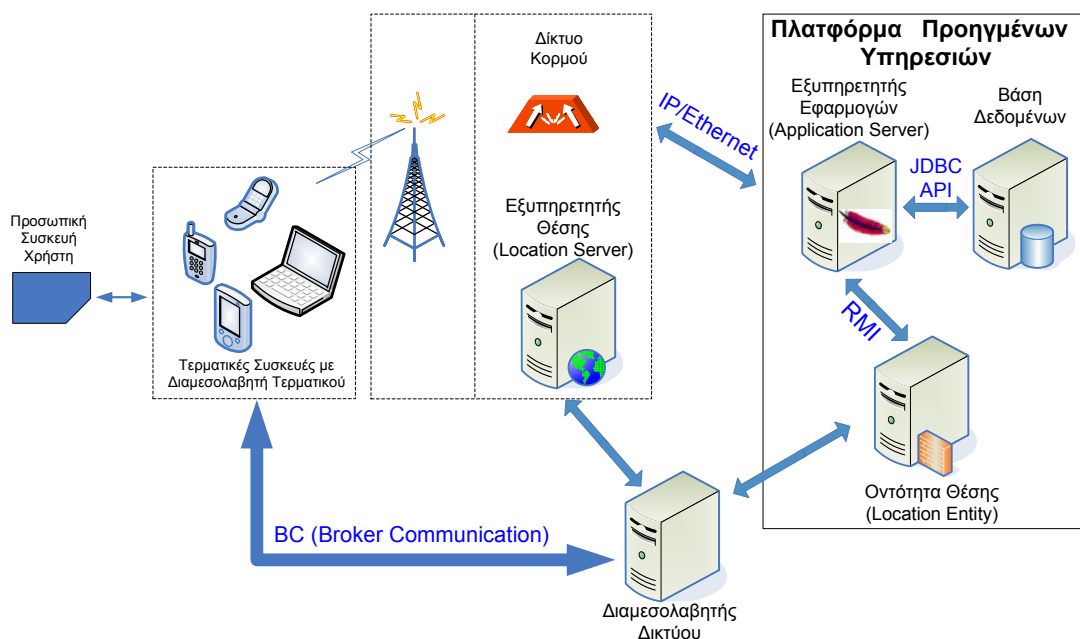
Τέλος, πρέπει να αναφέρουμε ότι ο πλήρως συμμετρικός και αναλογικός σχεδιασμός του Διαμεσολαβητή μας δίνει την δυνατότητα να εξάγουμε με ελάχιστο κίνδυνο γενικότερα συμπεράσματα για την συνολική συμπεριφορά του συστήματος, ακόμα και από μετρήσεις που αφορούν την αλληλεπίδραση μόνο δύο υποσυστημάτων. Αυτό οφείλεται στο γεγονός ότι είτε έχουμε έναν μεγάλο αριθμό υποσυστημάτων που αποστέλλει έναν αριθμό γεγονότων προς τον Μεσάζοντα, είτε δύο μόνο υποσυστήματα που ανταλλάσσουν μεταξύ τους μέσω του Μεσάζοντα τον ίδιο με την προηγούμενη περίπτωση αριθμό γεγονότων, ο επεξεργαστικός φόρτος για τον Διαμεσολαβητή παραμένει ο ίδιος.

4.2 Ανάκτηση πληροφορίας θέσης μέσω της προτεινόμενης αρχιτεκτονικής

Όπως παρουσιάστηκε στο πρώτο κεφάλαιο, οι δύο εναλλακτικές λύσεις που εφαρμόστηκαν για την ανάκτηση της θέσης του χρήστη στο πλαίσιο της υλοποίησης της Πλατφόρμας Παροχής Υπηρεσιών Αναζήτησης ήταν η υλοποίηση του πρωτοκόλλου MLP και της ανοιχτής προγραμματιστικής διεπαφής Κινητικότητας του OSA/Parlay (OSA/Parlay

Mobility API) πάνω από τρεις διαφορετικές τεχνολογίες μεσισμικού (CORBA, RMI, SOAP) [1][2][3]. Παράλληλα για την ανάκτηση των δεδομένων του χρήστη χρησιμοποιήθηκε βάση δεδομένων.

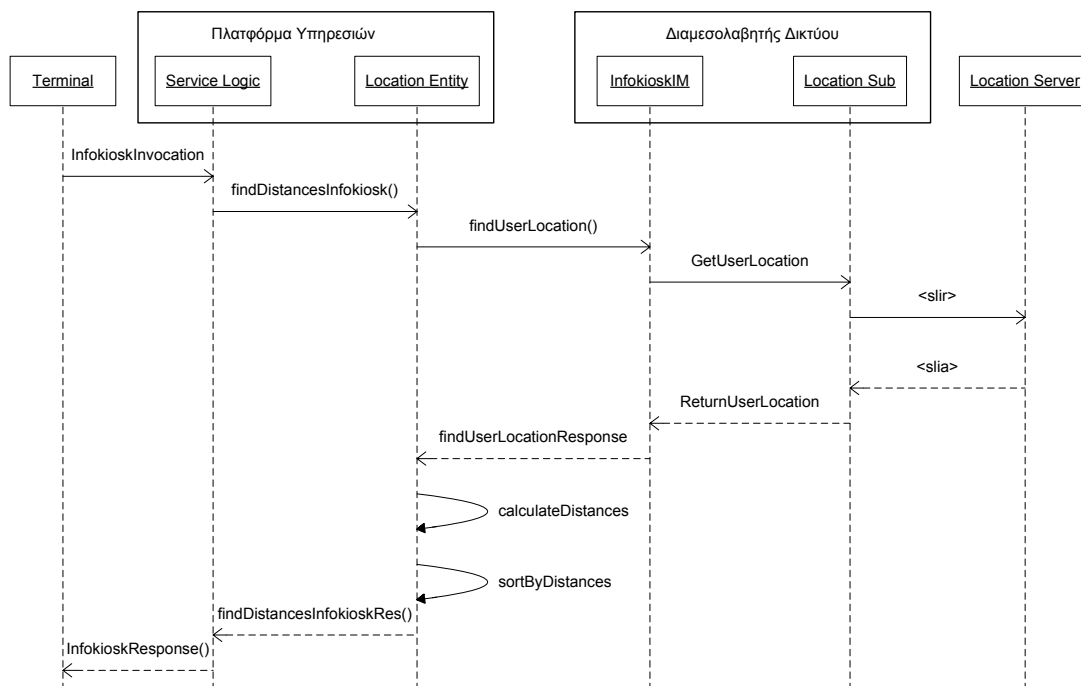
Παρατηρώντας τη δομή της πλατφόρμας, των παρεχόμενων υπηρεσιών και των δεδομένων που απαιτούνται για την εκτέλεση τους, και από την άλλη τον σχεδιασμό του προτεινόμενου συστήματος καθώς και τα είδη της πληροφορίας που δύναται να παρέχει, εύκολα μπορεί να εξαχθεί το συμπέρασμα ότι η λειτουργία της πλατφόρμας με χρήση της υποδομής των διαμεσολαβητών θα ήταν εφικτή και αποτελεσματική. Η προσπάθεια συνδυασμού των δύο αυτών προσεγγίσεων είχε ως αποτέλεσμα το σύστημα που παρουσιάζεται στο Σχήμα 60.



Σχήμα 60: Αρχιτεκτονική Συστήματος Παροχής Υπηρεσιών Αναζήτησης βάσει Θέσης και Προτιμήσεων Χρήστη με χρήση του προτεινόμενου συστήματος

Όπως φαίνεται από το Σχήμα 60, η βάση δεδομένων, όσον αφορά τα στοιχεία και τις προτιμήσεις του χρήστη, έχει καταργηθεί και η ανάκτηση των δεδομένων αυτών να γίνεται από τα κατάλληλα υποσυστήματα διαχείρισης προφίλ του Διαμεσολαβητή Δικτύου. Διατηρεί πλέον μόνο τα δεδομένα που δεν αφορούν τους χρήστες και κατά συνέπεια δεν μπορούν να ανακτηθούν από τους Διαμεσολαβητές, όπως για παράδειγμα οι συντεταγμένες των σταθερών σημείων ενδιαφέροντος που μπορεί να αναζητήσει κάποιος χρήστης μέσω της υπηρεσίας InfoKiosk. Επίσης, ο Διαμεσολαβητής Δικτύου λειτουργεί ως το μεσισμικό ανάμεσα στην εφαρμογή και τον Εξυπηρετητή Θέσης, αναλαμβάνοντας να επιστρέφει τη θέση του χρήστη στην εφαρμογή μέσω του Υποσυστήματος Διαχείρισης Πληροφορίας Θέσης Χρήστη. Επίσης διατηρείται η Οντότητα Θέσης. Στις πρώτες εκδοχές της

πλατφόρμας, η οντότητα θέσης λάμβανε αιτήσεις για εντοπισμό χρηστών και είτε τις μετέφραζε σε αιτήσεις MLP προς τον Εξυπηρετητή Θέσης του δικτύου, είτε σε μεθόδους του υλοποιημένου μέρους εφαρμογής της διεπαφής Κινητικότητας του OSA/Parlay (OSA/Parlay Mobility API Application Part). Αφού λάμβανε τις απαντήσεις, πραγματοποιούσε κατάλληλους υπολογισμούς για τον προσδιορισμό αποστάσεων ανάμεσα σε δύο ή περισσότερους χρήστες ή ανάμεσα σε έναν χρήστη και σταθερά σημεία ενδιαφέροντος.



Σχήμα 61: Διαδικασία ανάκτησης Θέσης Χρήστη Μέσω του Διαμεσολαβητή Δικτύου

Μεταφέροντας την υπηρεσία πάνω από το προτεινόμενο σύστημα, οι αρμοδιότητες της Οντότητας Θέσης του συστήματος αλλάζουν ελαφρώς. Πλέον, η οντότητα θέσης καλείται να αλληλεπιδράσει με το Υποσύστημα Διαχειριστή Διεπαφής που υλοποιήθηκε και τοποθετήθηκε στον Διαμεσολαβητή προκειμένου να διαχειρίζεται τις αιτήσεις για ανάκτηση πληροφοριών που προέρχονται από την εν λόγω πλατφόρμα. Από εδώ και στο εξής θα αναφερόμαστε στο συγκεκριμένο υποσύστημα με το όνομα InfokioskIM (Infokiosk Interface Manager). Συνεπώς, μετά τη λήψη μίας αίτησης για εντοπισμό κάποιου χρήστη, η Οντότητα θέσης καλεί την μέθοδο *findUserLocation()* στο InfokioskIM. Το InfokioskIM με την κλήση της μεθόδου δημιουργεί το γεγονός *GetUserLocationEvent* και το προωθεί στον Μεσάζοντα. Εκείνος με τη σειρά του προωθεί το γεγονός στο Υποσύστημα Διαχείρισης Πληροφορίας Θέσης Χρήστη, το οποίο υλοποιεί το πρωτόκολλο MLP. Με τη λήψη του γεγονότος καλείται στον Εξυπηρετητή Θέσης η μέθοδος *<slir>*. Εν συνεχεία ο

Εξυπηρετητής Θέσης επιστρέφει με την <slia> την ζητούμενη θέση, η οποία ακολουθώντας την ίδια διαδρομή, φτάνει τελικά στην Οντότητα Θέσης της πλατφόρμας. Εν τω μεταξύ, η Οντότητα Θέσης έχει ανακτήσει από τη βάση δεδομένων τις θέσεις των υποψήφιων προς επιστροφή σημείων ενδιαφέροντος του χρήστη και εν συνεχεία πραγματοποιεί όλους τους απαραίτητους υπολογισμούς αποστάσεων προκειμένου τελικά να επιστραφούν στο χρήστη τα κοντινότερα σε αυτόν σημεία ενδιαφέροντος με φθίνουσα σειρά. Οι παραπάνω αλληλεπιδράσεις παρουσιάζονται στο Σχήμα 61.

4.3 Αποτίμηση και σύγκριση των προσεγγίσεων

Έχοντας λοιπόν ολοκληρώσει όλες τις απαραίτητες προσαρμογές στην πλευρά της πλατφόρμας, καθώς και την υλοποίηση του InfokioskIM που αποτελεί τη διεπαφή της πλατφόρμας με τον Διαμεσολαβητή Δικτύου, μας δίνεται η δυνατότητα να μελετήσουμε και να συγκρίνουμε εν παραλλήλω τις τρεις διαφορετικές υλοποιήσεις ανάκτησης θέσης χρήστη για την παροχή της υπηρεσίας Infokiosk. Δηλαδή, την χρήση του πρωτοκόλλου MLP, που αποτελεί και πρόταση της 3GPPP προς τους λειτουργούς δικτύων κινητών επικοινωνιών [4], τη χρήση της διεπαφής Κινητικότητας του OSA/Parlay και τέλος τη χρήση, ως μεσισμικού, της υποδομής των Διαμεσολαβητών.

Για την περίπτωση της υλοποίησης με χρήση της διεπαφής Κινητικότητας του OSA/Parlay πρέπει να αναφερθεί ότι πραγματοποιήθηκε με χρήση τριών διαφορετικών τεχνολογιών μεσισμικού. Οι τεχνολογίες αυτές ήταν η CORBA, το RMI και το SOAP. Η επιλογή τριών διαφορετικών προσεγγίσεων πραγματοποιήθηκε διότι η απόδοση του υλοποιημένου με την διεπαφή Κινητικότητας συστήματος, εξαρτάται κυρίως από την απόδοση της χρησιμοποιούμενης τεχνολογίας μεσισμικού και συνεπώς κρίθηκε απαραίτητη η μελέτη περισσότερων προσεγγίσεων για την εξαγωγή πιο αντικειμενικών και καθολικών συμπερασμάτων [1][2][3].

Οι παράμετροι που επιδρούν στην απόδοση των συστημάτων μεσισμικού, όπως είναι στη προκειμένη περίπτωση η υλοποίηση της διεπαφής κινητικότητας ή οι διαμεσολαβητές της προτεινόμενης αρχιτεκτονικής, ποικίλλουν [1][2][3]. Ο εσωτερικός αρχιτεκτονικός σχεδιασμός, τα αντικείμενα που λαμβάνουν μέρος στην υλοποίηση, καθώς επίσης και τα ποιοτικά χαρακτηριστικά κάθε μεσισμικού έχουν το δικό τους σημαντικό ρόλο στην απόδοση του συνολικού συστήματος. Στην περίπτωση που υποστηρίζονται επιπλέον χαρακτηριστικά στο όλο σύστημα, όπως εφεδρικοί μηχανισμοί (για περιπτώσεις βλαβών ή έκτακτης ανάγκης), μηχανισμοί διαλειτουργικότητας μεταξύ ανομοιογενών πλατφορμών

κ.α., είναι προφανές ότι η απόδοσή του θα εμφανίζεται μειωμένη σε σχέση με απλούστερες υλοποιήσεις [4]. Επιπλέον εξωτερικές του μεσισμικού παράμετροι, όπως η τοπολογία του δικτύου και η υπολογιστική ισχύς των κόμβων που φιλοξενούν τις οντότητες του μεσισμικού μπορούν να επηρεάσουν την απόδοση του συστήματος.

Κατά συνέπεια είναι αρκετά εύλογο στο σημείο αυτό να υποστηρίξει κανείς ότι τα αποτελέσματα της μελέτης αποτίμησης της απόδοσης που θα ακολουθήσουν είναι αρκετά εξειδικευμένα και αντικατοπτρίζουν μόνο μία συγκεκριμένη δικτυακή τοπολογία. Ο ισχυρισμός αυτός δεν είναι απόλυτα σωστός, δεδομένου ότι τα προσδοκώμενα αποτελέσματα θα αποτελέσουν το υπόβαθρο και μία πρώτη ένδειξη αναφορικά με την καλύτερη και αποδοτικότερη λύση που μπορεί να χρησιμοποιηθεί για την ανάκτηση πληροφοριών θέσης χρήστη από την οντότητα του Εξυπηρετητή Θέσης των Δικτύων Κινητών Επικοινωνιών.

Η ανάλυση που θα ακολουθήσει, δίνει έμφαση στην επάρκεια (ή μη) της κυρίως υλοποίησης των διαφορετικών μεσισμικών βάσει της απόδοσης που το κάθε ένα αντίστοιχα παρουσιάζει. Αποφεύγεται να δοθεί ιδιαίτερο βάρος σε άλλους εξωτερικούς παράγοντες, που πιθανώς επηρεάζουν το όλο σύστημα, όπως η τοπολογία του IP δικτύου π.χ. από πόσους δρομολογητές είναι δυνατόν να διέλθουν τα πακέτα μίας αίτησης ή μίας απάντησης κτλ., αλλά βέβαια τηρείται η ίδια πολιτική και στις τρεις διαφορετικές υλοποιήσεις με αποτέλεσμα να είναι δυνατή η σχετική τους σύγκριση. Βάσει των τελικών αποτελεσμάτων που θα ληφθούν, γίνεται αφενός σύγκριση μεταξύ των τριών υλοποιήσεων του OSA/Parlay (πάνω από CORBA, RMI και SOAP αντίστοιχα), της υλοποίησης με χρήση της υποδομής των Διαμεσολαβητών και αφετέρου σύγκριση των υλοποιήσεων αυτών σε σχέση με την ισχύουσα και τρέχουσα υλοποίηση που υποστηρίζουν τα σύγχρονα Δίκτυα Κινητών Επικοινωνιών και στηρίζεται αποκλειστικά στο πρωτόκολλο MLP. Σύμφωνα με τα παραπάνω, γίνεται κατανοητό ότι ο βασικός στόχος της ανάλυσης αυτής είναι η κατανόηση και αποτίμηση του τρόπου συμβολής ενός στρώματος μεσισμικού για την ανάκτηση της θέσης του χρήστη από τα Δίκτυα Κινητών Επικοινωνιών. Επιπλέον στόχος της μελέτης είναι η γέννηση ενός πρώτου μέτρου σύγκρισης, ικανού να κρίνει κατά πόσο αξίζει να θυσιάστουν οι υπάρχουσες αποδεκτές υλοποιήσεις στο βωμό της ανοικτότητας και επεκτασιμότητας που υπηρετεί το προτεινόμενο σύστημα.

4.3.1 Μοντέλο αποτίμησης απόδοσης

Στο σημείο αυτό καθορίζονται και αναλύονται οι σημαντικότερες παράμετροι που λαμβάνονται κατά την αποτίμηση των προτεινόμενων υλοποιήσεων. Εξετάζεται η απόδοση

και των δύο προσεγγίσεων και συγκρίνονται με την υπάρχουσα υλοποίηση που προσφέρει το πρωτόκολλο MLP.

Τα υπό αποτίμηση συστήματα φιλοξενούνται σε ένα Τοπικό Δίκτυο (Local Area Network, LAN), το οποίο προσφέρει δυνατότητες μεταφοράς πάνω από το IP πρωτόκολλο. Η τοπολογία στην οποία πραγματοποιήθηκαν οι μετρήσεις των υλοποιήσεων με χρήση της ανοικτής διεπαφής πάνω από τις τεχνολογίες λογισμικού αποτελείται από δύο σταθμούς εργασίας (workstations). Ο πρώτος με επεξεργαστή PIII στα 733 MHz, μνήμη 256 Mb RAM και λειτουργικό σύστημα Windows XP, ο οποίος φιλοξενεί την πλευρά του δικτύου της υλοποίησης του OSA/Parlay και τον Εξυπηρετητή Θέσης (Location Server). Όσον αφορά στον δεύτερο σταθμό εργασίας, πρόκειται για έναν απομακρυσμένο υπολογιστή που βασίζεται στο λειτουργικό σύστημα Windows NT, με επεξεργαστή PIII στα 450 MHz και μνήμη 256 Mb RAM. Σε αυτόν τον σταθμό εργασίας φιλοξενείται η λογική της υπηρεσίας *InfoKioskSearch* που υποστηρίζει την αντίστοιχη πλευρά των εφαρμογών της διεπαφής Κινητικότητας του OSA/Parlay.

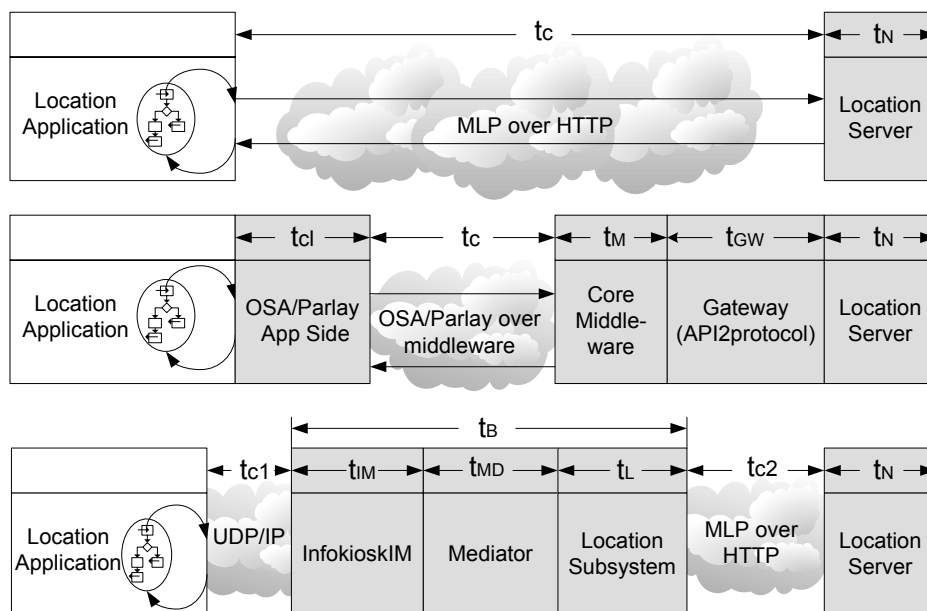
Στην περίπτωση της υλοποίησης του στρώματος μεσισμικού βάσει της CORBA, η διεπαφή Κινητικότητας του OSA/Parlay, καθώς και οι απαραίτητοι τύποι δεδομένων των βοηθητικών κλάσεων, αναπτύχθηκαν βάσει του IDL μεταφραστή που παρέχει η SUN στην έκδοση 1.4.1. του Software Development Kit (SDK) της Java. Η Υπηρεσία Ονοματολογίας της CORBA καθώς και ο πυρήνας της, παρέχονται από το ίδιο SDK. Ανάλογα, και στην περίπτωση της υλοποίησης με RMI οι διεπαφές και οι τύποι δεδομένων αναπτύχθηκαν βάσει του Java Development Kit (JDK) της έκδοσης 1.4.1. της SUN. Η υπηρεσία ονοματολογίας της RMI (rmiregistry) που χρησιμοποιήθηκε ανήκει στο ίδιο JDK. Τέλος, στην περίπτωση του SOAP, χρησιμοποιήθηκε η τεχνολογία κατανεμημένης επεξεργασίας μεσισμικού Apache AXIS (έκδοση 1.0) που υλοποιεί το πρωτόκολλο SOAP και αποτέλεσε το πλαίσιο για την ανάπτυξη των προδιαγραφών του OSA/Parlay βάσει της τεχνολογίας των Υπηρεσιών Διαδικτύου (Web Services). Οι υπηρεσίες του Apache AXIS έγιναν προσπελάσιμες μέσω ενός servlet που αναπτύχθηκε στον διαδικυακό εξυπηρετητή (web server) Apache Tomcat (έκδοση 4.1). Στην περίπτωση του SOAP, και η πλευρά των εφαρμογών αλλά και η πλευρά του δικτύου της διεπαφής Κινητικότητας του OSA/Parlay υλοποιήθηκαν αυτόματα, τροφοδοτώντας την WSDL περιγραφή που διατίθεται από το φόρουμ του OSA/Parlay στο WSDL2Java μετατροπέα που παρέχεται από το AXIS.

Στην περίπτωση των μετρήσεων για την ανάκτηση της πληροφορίας θέσης μέσω του Διαμεσολαβητή Δικτύου η τοπολογία αποτελούνταν από τρεις απομακρυσμένους μεταξύ τους σταθμούς εργασίας. Οι δύο ήταν οι ίδιοι που χρησιμοποιήθηκαν και στην προηγούμενη

περίπτωση και κατά αντιστοιχία φιλοξενούσαν τον Εξυπηρετητή Θέσης και τη λογική της υπηρεσίας. Ο τρίτος σταθμός εργασίας που χρησιμοποιήθηκε είχε επεξεργαστή PIV Celeron στα 2,5GHz μνήμη 512 Mb RAM και λειτουργικό σύστημα Windows XP. Αυτός ο σταθμός εργασίας φιλοξενεί τον Διαμεσολαβητή Δικτύου.

Τέλος, όσον αφορά την υλοποίηση του Εξυπηρετητή Θέσης, σε όλες τις περιπτώσεις, χρησιμοποιήθηκε ένας εξομοιωτής Εξυπηρετητή Θέσης (Location Server). Υποστηρίζει τα μηνύματα του πρωτοκόλλου MLP και η πρόσβαση σε αυτόν γίνεται μέσω του MLP πάνω από το HTTP. Η φύση του Εξυπηρετητή Θέσης δεν επηρεάζει την αποτίμηση της απόδοσης του στρώματος μεσισμικού όπως θα εξηγηθεί εκτενώς στη συνέχεια της διατριβής. Σημειώνεται ότι ο Εξυπηρετητής Θέσης δεν αφορά πραγματική υλοποίηση, αλλά πρόκειται για σύστημα εξομοίωσης, που προσφέρει τυχαίες συντεταγμένες για κάθε ζητούμενο χρήστη. Προκειμένου να κατασκευαστεί το μήνυμα απόκρισης του MLP (που περιέχει τη θέση του ζητούμενου χρήστη), η ανάλυση και ανασύνθεση του MLP μηνύματος ήταν η ελάχιστη δυνατή, όση ακριβώς απαιτείται για την αντιστοίχιση του αναγνωριστικού MSISDN και το συσχετισμό αυτού με μία τυχαία θέση στο χώρο.

Για τη γέννηση των αιτήσεων από πλευράς της υπηρεσίας προς την πλευρά του δικτύου, υλοποιήθηκε μοντέλο που βασίζεται στην κατανομή Poisson, με σκοπό την εξασφάλιση ανεξαρτησίας από την τυχαιότητα στη χρήση των υπηρεσιών [6].



Σχήμα 62: Απεικόνιση των χρονικών τιμών που συμβάλλουν στην καθυστέρηση α) με χρήση του πρωτοκόλλου MLP πάνω από HTTP β) με χρήση της Πύλης Εντοπισμού Θέσης πάνω από στρώμα μεσισμικού και γ) με χρήση του Διαμεσολαβητή Δικτύου

Οι διαφορετικές αιτίες πρόκλησης καθυστέρησης στις υλοποιήσεις απεικονίζονται στο Σχήμα 62. Οι διαφορετικές χρονικές τιμές που διαφαίνονται στο Σχήμα 62 αφορούν τους αντίστοιχους χρόνους επεξεργασίας που απαιτούνται για τις επί μέρους διεργασίες και προς τις δύο κατευθύνσεις (αποστολή αίτησης ή αναμονή απάντησης). Οι ορισμοί των επί μέρους χρόνων του Σχήματος 62, στους οποίους στηρίζεται και η τελική αποτίμηση της απόδοσης του συστήματος, δίνονται στον Πίνακα 4 για την περίπτωση της υλοποίησης με τη διεπαφή OSA/Parlay και στον Πίνακα 5, που παρατίθεται στη συνέχεια, για την βασισμένη στο προτεινόμενο σύστημα υλοποίηση.

Πίνακας 4: Οι μετρούμενες τιμές – Υλοποίηση με τη διεπαφή OSA/Parlay

Συντομογραφία	Ορισμός
t_{CL}	Ο χρόνος που απαιτείται για την επεξεργασία από πλευράς εφαρμογών
t_C	Ο χρόνος που απαιτείται για τη μετάδοση των αιτήσεων/αποκρίσεων
t_M	Ο χρόνος που καταναλώνεται λόγω των μηχανισμών επικοινωνίας του μεσισμικού
t_G	Ο χρόνος που απαιτείται από την Πύλη Εντοπισμού Θέσης του μεσισμικού. Περιλαμβάνει τη μετατροπή από OSA/Parlay σε MLP και αντιστρόφως.
t_N	Ο χρόνος που απαιτείται από τον Εξυπηρετητή Θέσης για να επιστρέψει τα αποτελέσματα στην Πύλη Εντοπισμού Θέσης.

Πιο συγκεκριμένα, αναφορικά με τους επί μέρους χρονικούς παράγοντες που εμπλέκονται στην υλοποίηση του συστήματος, ο t_{CL} σχετίζεται με τον απαιτούμενο χρόνο από την υλοποίηση της διεπαφής Κινητικότητα του OSA/Parlay στην πλευρά των εφαρμογών κατά την κατασκευή των μεθόδων και των δομών δεδομένων του OSA/Parlay με σκοπό την αποστολή των αντίστοιχων αιτήσεων προς την πλευρά του δικτύου. Στην αντίθετη φορά, ο t_{CL} αντιπροσωπεύει τον χρόνο απογύμνωσης των μεθόδων και των δομών δεδομένων του OSA/Parlay από τις LBS για την ανάκτηση της ζητούμενης πληροφορίας.

Η τιμή του t_C εξαρτάται από την τοπολογία του δικτύου που χρησιμοποιείται για την παροχή των υπηρεσιών και τον Εξυπηρετητή Θέσης. Ένας ελαφρύς και γρήγορος διάυλος επικοινωνίας από την πλευρά του μεσισμικού μπορεί να συνεισφέρει θετικά στην συνολική απόδοση της επικοινωνίας σε σχέση με την χρήση του πρωτοκόλλου HTTP για τη μετάδοση των MLP μηνυμάτων.

Η τιμή της παραμέτρου t_M στην περίπτωση της CORBA και του RMI αντιπροσωπεύει τον χρόνο που απαιτούν οι αντίστοιχες κλάσεις της CORBA και του RMI για την πραγματοποίηση των κλήσεων των μεθόδων στα απομακρυσμένα αντικείμενα και την αναγέννηση των κατάλληλων παραμέτρων σε μορφή αντιληπτή στο επίπεδο εφαρμογών από την προηγούμενη μορφή τους (τη σειριοποιημένη στον ΠΟΡ διάυλο της CORBA ή στο

αντίστοιχο κανάλι του RMI). Στην περίπτωση του SOAP, η τιμή της παραμέτρου t_M απεικονίζει τις καθυστερήσεις που προκαλούνται από την ανάλυση των SOAP μηνυμάτων και τη χρήση του συγκεκριμένου διαδικτυακού εξυπηρετητή (Apache Tomcat web server). Η τιμή της παραμέτρου t_M είναι δύσκολο να υπολογιστεί, δεδομένου ότι απαιτεί προγραμματιστική επέμβαση στις κλάσεις κορμού που παρέχονται από κάθε τεχνολογία μεσισμικού, αλλά εναλλακτικά είναι δυνατόν να υπολογιστεί έμμεσα ως η διαφορά των υπολοίπων επί μέρους χρονικών παραμέτρων (t_{CL} , t_C , t_G and t_N) από το συνολικό χρόνο (t_{TOTAL}). Η τιμή της συγκεκριμένης παραμέτρου t_M αναμένεται να προκαλέσει εμφανείς διαφοροποιήσεις μεταξύ των τριών υλοποιήσεων λόγω της διαφορετικότητας στα εσωτερικά μοντέλα απομακρυσμένης επικοινωνίας που χρησιμοποιείται στην κάθε μία (Πίνακας 6).

Η τιμή του χρόνου που ορίζεται ως t_G αντικατοπτρίζει την πολυπλοκότητα των χρησιμοποιούμενων δομών δεδομένων και των επιπρόσθετων επεξεργασιών που απαιτούνται από την Πύλη Εντοπισμού Θέσης για την ανάλυση και την κατασκευή των μηνυμάτων MLP από τις αντίστοιχες μεθόδους του OSA/Parlay και αντιστρόφως.

Τέλος, η τιμή της παραμέτρου t_N εξαρτάται άμεσα από την εσωτερική δικτυακή τοπολογία του Λειτουργού Δικτύου, τη δικτυακή κίνηση και άλλα θέματα μη σχετικά με την εφαρμογή του μεσισμικού. Η θεώρηση ότι ο συγκεκριμένος χρόνος θα είναι παραπλήσιος, είτε με χρήση των τεχνολογιών μεσισμικού είτε χρησιμοποιώντας την υπάρχουσα υλοποίηση που στηρίζεται στο πρωτόκολλο MLP, φαίνεται εκ των αποτελεσμάτων να επαληθεύεται και κατά συνέπεια να είναι σωστή (Πίνακας 6).

Πίνακας 5: Οι μετρούμενες τιμές – Υλοποίηση με χρήση Διαμεσολαβητών.

Συντομογραφία	Ορισμός
t_{C1}	Ο χρόνος που απαιτείται για τη μετάδοση των αιτήσεων/αποκρίσεων ανάμεσα στην εφαρμογή και τον Διαμεσολαβητή
t_{IM}	Ο χρόνος που απαιτείται από το Υποσύστημα Διαχειριστή Εξωτερικής Διεπαφής για την μετατροπή της αίτησης που θα λάβει σε κατάλληλο γεγονός και αντίστροφα
t_{MD}	Ο χρόνος που καταναλώνεται από τη στιγμή που ο Μεσάζοντας θα λάβει το γεγονός ώσπου να το αποστείλει
t_L	Ο χρόνος που απαιτείται από Υποσύστημα Διαχείρισης Πληροφορίας Θέσης Χρήστη. Περικλείει τη μετατροπή του γεγονότος σε MLP και αντίστροφα.
t_{C2}	Ο χρόνος που απαιτείται για τη μετάδοση των αιτήσεων/αποκρίσεων ανάμεσα στο Υποσύστημα Διαχείρισης Πληροφορίας Θέσης Χρήστη και τον Εξυπηρετητή Θέσης
t_N	Ο χρόνος που απαιτείται από τον Εξυπηρετητή Θέσης για να ανακτήσει και να αποστείλει τα αποτελέσματα.

Πιο συγκεκριμένα η τιμή του t_{C1} εξαρτάται από την τοπολογία του δικτύου που χρησιμοποιείται για την παροχή των υπηρεσιών και από την τεχνολογία που επιλέγει ο κάθε

ανεξάρτητος πάροχος υπηρεσιών για να υλοποιήσει την επικοινωνία ανάμεσα στην υπηρεσία και τον Διαμεσολαβητή Δικτύου, ή πιο συγκεκριμένα με το Υποσύστημα Διαχειριστή Εξωτερικής Διεπαφής της υπηρεσίας. Στην συγκεκριμένη περίπτωση επιλέξαμε η επικοινωνία αυτή να πραγματοποιείται με χρήση UDP/IP. Οι λόγοι που οδήγησαν στην επιλογή αυτή ήταν αρχικά η απλότητα της υλοποίησης και το γεγονός ότι ο στόχος της μελέτης αυτής δεν είναι ο υπολογισμός του χρόνου που αναλώνεται κατά την επικοινωνία της υπηρεσίας με τον Διαμεσολαβητή Δικτύου αλλά η απόδοση του υλοποιημένου συστήματος. Σε μία πραγματική υλοποίηση, ο πάροχος υπηρεσιών θα μπορούσε να αντικαταστήσει την υλοποίηση της διεπαφής αυτής από UDP με οποιαδήποτε τεχνολογία μεσισμικού/ανοιχτής διεπαφής όπως το SOAP ή η CORBA. Μία τέτοια επιλογή θα εισήγαγε στο σύστημα επιπλέον καθυστέρηση, αλλά αυτός είναι ένας παράγοντας για τον οποίο καλείται να αναλάβει την ευθύνη και να τον ρυθμίσει κατά περίπτωση ο πάροχος υπηρεσιών.

Ο χρόνος t_{IM} αντικατοπτρίζει το διάστημα που απαιτείται από τη στιγμή που το Υποσύστημα Διαχειριστή Εξωτερικής Διεπαφής της υπηρεσίας θα παραλάβει την αίτηση από την Οντότητα Θέσης της πλατφόρμας, θα ανακτήσει τα ορίσματα που θα παραλάβει με την αίτηση, θα δημιουργήσει κατάλληλο γεγονός και θα το προωθήσει στον Μεσάζοντα. Από την στιγμή που ο Μεσάζοντας θα λάβει το γεγονός αυτό ξεκινάει το χρονικό διάστημα t_{MD} , το οποίο ολοκληρώνεται αφού ο Μεσάζοντας βάσει κατάλληλων κανόνων θα αποστείλει το γεγονός στο Υποσύστημα Διαχείρισης Πληροφορίας Θέσης Χρήστη. Τέλος, η τιμή του χρόνου t_L αναπαριστά το διάστημα που απαιτείται από το Υποσύστημα Διαχείρισης Πληροφορίας Θέσης Χρήστη για την ανάλυση και την κατασκευή των μηνυμάτων MLP από τα γεγονότα που παραλαμβάνει από τον Μεσάζοντα. Στην αντίθετη κατεύθυνση, δηλαδή κατά την αποστολή των απαντήσεων από τον Εξυπηρετητή Θέσης προς την υπηρεσία, οι χρόνοι αυτοί αντικατοπτρίζουν τις ακριβώς αντίστροφες διαδικασίες. Επίσης πρέπει να αναφερθεί ότι το άθροισμα των τριών αυτών χρόνων αποτελεί τον t_{BR} , ο οποίος είναι ο χρόνος επεξεργασίας που καταναλώνεται από την στιγμή που μία αίτηση ή απόκριση εισέλθει στον διαμεσολαβητή μέχρι να φύγει από αυτόν. Αυτή η χρονική καθυστέρηση επηρεάζεται από τα τεχνικά χαρακτηριστικά του εξυπηρετητή που φιλοξενεί τον Διαμεσολαβητή Δικτύου, από τον όγκο των γεγονότων που ο τελευταίος καλείται να εξυπηρετήσει, αλλά και από την τεχνολογία που χρησιμοποιεί ο πάροχος των υπηρεσιών για την επικοινωνία του με τον Διαμεσολαβητή. Από το τελευταίο ουσιαστικά καθορίζεται ο χρόνος που απαιτείται προκειμένου το υποσύστημα InfokioskIM να κατασκευάσει από την

αίτηση που λαμβάνει το κατάλληλο γεγονός αίτηση ή από το γεγονός απόκριση την απόκριση προς την υπηρεσία.

Και ο τελευταίος χρόνος, εκτός από τον t_N ο οποίος ταυτίζεται με αυτόν της προηγούμενης τοπολογίας, είναι ο t_{C2} . Είναι ο χρόνος που απαιτείται προκειμένου να μεταφερθεί η MLP αίτηση πάνω από το HTTP πρωτόκολλο από τον Διαμεσολαβητή Δικτύου στον Εξυπηρετητή Θέσης και η MLP απόκριση από τον Εξυπηρετητή Θέσης στον Διαμεσολαβητή Δικτύου.

Από τους παραπάνω ορισμούς των χρονικών παραμέτρων, για την αποτίμηση της απόδοσης των διαφορετικών υλοποιήσεων μεσισμικού, ορίζεται ο μέσος συνολικός χρόνος ($\overline{t_{TOTAL}}$) που απαιτείται στην περίπτωση χρησιμοποίησης του απλού πρωτοκόλλου MLP, ως:

$$\overline{t_{TOTAL}} = \overline{t_C} + \overline{t_N}$$

στην περίπτωση που γίνεται χρήση της Διεπαφής Κινητικότητας, ως :

$$\overline{t_{TOTAL}} = \overline{t_{CL}} + \overline{t_C} + \overline{t_M} + \overline{t_G} + \overline{t_N}$$

και στην περίπτωση της εκμετάλλευσης του προτεινόμενου συστήματος ως:

$$\overline{t_{TOTAL}} = \overline{t_{C1}} + \overline{t_{IM}} + \overline{t_{MD}} + \overline{t_L} + \overline{t_{C2}} + \overline{t_N}$$

Ο μέσος συνολικός χρόνος ($\overline{t_{TOTAL}}$) αποτελεί προφανώς μία πρώτη ένδειξη για την απόδοση του στρώματος μεσισμικού, εφόσον περιλαμβάνει όλες τις επί μέρους προαναφερθείσες χρονικές παραμέτρους. Οι τιμές των παραπάνω παραμέτρων εξαρτώνται άμεσα από το χρησιμοποιούμενο μεσισμικό και επηρεάζουν την απόδοση του συστήματος. Μοναδική εξαίρεση αποτελεί η χρονική παράμετρος t_N , η οποία όπως εξηγήθηκε προηγουμένως είναι ανεξάρτητη του μεσισμικού.

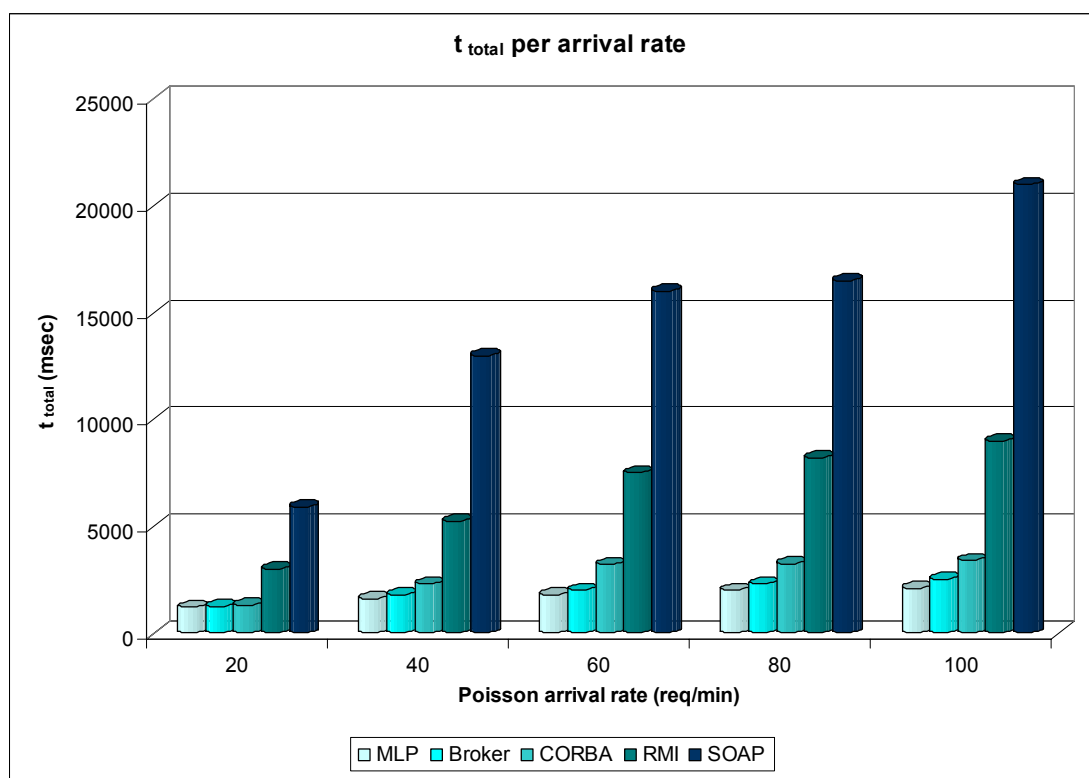
4.3.2 Μετρήσεις – Αποτελέσματα

Όπως προαναφέρθηκε, η μέση τιμή του συνολικού χρόνου ($\overline{t_{TOTAL}}$) υπολογίζεται και για τις εξής πέντε περιπτώσεις: την απλή χρήση του MLP πρωτοκόλλου, την χρήση της διεπαφής OSA/Parlay Mobility με τις τρεις διαφορετικές τεχνολογίες μεσισμικού και την υλοποίηση που σαν μεσισμικό χρησιμοποιείται η υποδομή του συστήματος Διαμεσολαβητών.

Το μοντέλο που χρησιμοποιήθηκε για τη γέννηση και αποστολή αιτήσεων των LBS βασίζεται, όπως έχει ήδη αναφερθεί, στην κατανομή Poisson. Η κατανομή Poisson χρησιμοποιείται για τη μοντελοποίηση ενός αριθμού γεγονότων που λαμβάνουν χώρα μέσα σε ένα δεδομένο χρονικό διάστημα. Όταν μία τυχαία μεταβλητή x ακολουθεί την κατανομή Poisson με παράμετρο λ ($\lambda > 0$), η συνάρτηση μάζας πιθανότητάς της δίνεται από τον τύπο:

$$p(x, \lambda) = \frac{e^{-\lambda} \lambda^x}{x!}, \quad x=0,1,2,\dots$$

Η παράμετρος λ υποδεικνύει τον μέσο ρυθμό των αφίξεων που λαμβάνουν χώρα σε συγκεκριμένη χρονική περίοδο. Κατά τη διάρκεια των πειραμάτων χρησιμοποιήθηκαν για την παράμετρο λ πέντε διαφορετικές τιμές, 20, 40, 60, 80 και 100 αιτήσεων ανά λεπτό αντίστοιχα. Κάθε πείραμα έλαβε χώρα για περίπου τρεις ώρες, με σκοπό να επιτευχθεί όσο το δυνατόν μεγαλύτερη αξιοπιστία στη χρήση της κατανομής Poisson, καθώς και στα αποτελέσματα που θα προκύψουν με τη χρήση αυτής.



Σχήμα 63: Ο μέσος συνολικός χρόνος εξυπηρέτησης ($\overline{t_{TOTAL}}$) της υπηρεσίας

Στο διάγραμμα του Σχήματος 63, απεικονίζεται ο μέσος συνολικός χρόνος του κύκλου ζωής των αιτήσεων της υπηρεσίας, ο χρόνος δηλαδή εξυπηρέτησης (συμπεριλαμβανομένου και του χρόνου απάντησης στις αιτήσεις) για τις πέντε διαφορετικές τιμές φορτίου αιτήσεων στο σύστημα (20, 40, 60, 80 και 100 αιτήσεις ανά λεπτό). Τα πέντε αυτά πειράματα έλαβαν χώρα και για τις πέντε περιπτώσεις, είτε μέσω της διεπαφής OSA/Parlay Mobility πάνω από τεχνολογία μεσισμικού (CORBA, RMI και SOAP), είτε με χρήση του Διαμεσολαβητή Δικτύου, είτε χωρίς μεσισμικό (απλό πρωτόκολλο MLP).

Η πρώτη εμφανής παρατήρηση επί του παραπάνω διαγράμματος, είναι ότι η τιμή του $\overline{t_{TOTAL}}$ στην περίπτωση της χρήσης του πρωτοκόλλου MLP, είναι αρκετά χαμηλότερη από

τις αντίστοιχες κατά την χρήση σχεδόν όλων των τεχνολογιών μεσισμικού (ιδιαίτερα για χαμηλούς ρυθμούς άφιξης αιτήσεων) αλλά και ελάχιστα μικρότερη από την περίπτωση χρήσης του Διαμεσολαβητή Δικτύου. Η εξήγηση είναι σχεδόν προφανής, εφόσον στην περίπτωση απλής χρήσης του πρωτοκόλλου MLP δεν περιέχονται οι επί μέρους μέσοι χρόνοι $\overline{t_{CL}}$, $\overline{t_M}$ και $\overline{t_G}$ ή οι $\overline{t_{IM}}$, $\overline{t_M}$ και $\overline{t_L}$ αντίστοιχα, δεδομένου ότι δεν γίνεται χρήση κάποιας τεχνολογίας μεσισμικού και της διεπαφής Κινητικότητας του OSA/Parlay, ούτε παρεμβάλλεται η οντότητα του Διαμεσολαβητή Δικτύου, αλλά η επικοινωνία εγκαθίσταται πάνω από το πρωτόκολλο μεταφοράς HTTP.

Συγκρίνοντας το $\overline{t_{TOTAL}}$ μεταξύ των διαφορετικών τεχνολογιών μεσισμικού εύκολα κανείς διακρίνει ότι στην περίπτωση χρήσης των Web Services και SOAP η αντίστοιχη τιμή είναι κατά πολύ μεγαλύτερη από αυτή στις περιπτώσεις χρήσης RMI και CORBA. Συγκεκριμένα παρουσιάζεται σχεδόν διπλάσια από αυτή του RMI και σχεδόν τετραπλάσια από την αντίστοιχη της CORBA. Η διακύμανση αυτή μεταξύ των διαφορετικών τεχνολογιών μεσισμικού αντικατοπτρίζει ουσιαστικά τις διαφορές των επί μέρους μέσων τιμών των χρόνων t_{CL} , t_C και t_M , δεδομένου ότι οι τιμές των υπολοίπων παραμέτρων (t_G και t_N) δεν παρουσιάζουν σημαντικές διαφορές (Πίνακας 7).

Η υψηλότερη τιμή για τη χρονική παράμετρο t_{CL} στην περίπτωση του SOAP δικαιολογείται, δεδομένου ότι για την κατασκευή/ανακατασκευή των πολύπλοκων δομών δεδομένων της διεπαφής Κινητικότητας του OSA/Parlay απαιτείται περισσότερος χρόνος επεξεργασίας κατά τη σειριοποίηση και αποσειριοποίησή τους (serialization/deserialization) από το AXIS, σε αντίθεση τους αντίστοιχους χρόνους στην περίπτωση του RMI και της CORBA.

Αναφορικά με την τιμή της παραμέτρου t_M , παρουσιάζεται επίσης υψηλότερη στην υλοποίηση του SOAP συγκριτικά με την αντίστοιχη στις υλοποιήσεις της CORBA και του RMI (Πίνακας 6). Το γεγονός αυτό εξηγείται, διότι ο συγκεκριμένος χρόνος περιέχει εκτός από τον απαραίτητο χρόνο για την εγκατάσταση των μηχανισμών επικοινωνίας του ίδιου του μεσισμικού και τις καθυστερήσεις που συμβαίνουν στον Web Server (Apache Tomcat) καθώς και κατά την ανάλυση και επεξεργασία των μηνυμάτων του SOAP.

Τέλος, και η χρονική παράμετρος t_C επιβαρύνεται στην περίπτωση της υλοποίησης του SOAP έναντι των αντίστοιχων της CORBA και του RMI (Πίνακας 7). Ο χρόνος t_C φωτογραφίζει ξεκάθαρα την απόδοση του υποκείμενου μηχανισμού μεταφοράς που χρησιμοποιεί κάθε μία από τις τεχνολογίες μεσισμικού. Τα αποτελέσματα των μετρήσεων είναι αναμενόμενα, δεδομένου ότι στην περίπτωση του SOAP γίνεται χρήση του HTTP για την μεταφορά των αντίστοιχων μηνυμάτων, ενώ στις περιπτώσεις του RMI και της CORBA

γίνεται χρήση των αντίστοιχων διαύλων JRMP (Java Remote Method Protocol) και ΙΙΟΡ (Internet Inter-ORB Protocol) που στηρίζονται στο TCP/IP και έχουν φανερά καλύτερη απόδοση. Στην καθυστέρηση αυτή που παρουσιάζει το SOAP συμβάλλει βεβαίως και το μέγεθος των μηνυμάτων (τα μηνύματα του SOAP είναι μεγαλύτερα σε σχέση με αυτά του RMI και της CORBA), αλλά και το περιεχόμενο των μηνυμάτων που μεταφέρονται (μεταφορά απλού κειμένου στην περίπτωση του SOAP έναντι δυαδικής μεταφοράς στις περιπτώσεις του RMI και της CORBA).

Πίνακας 6: Οι Χρόνοι t_M και t_N

Poisson Arrival Rate (Req/min)	t_M (msec)			t_N (msec)		
	SOAP	CORBA	RMI	SOAP	CORBA	RMI
20	3526.17	91.30	292.97	1086.43	935.83	1583.21
40	9894.24	194.92	739.68	1629.68	1921.35	1841.54
60	12221.91	283.77	605.01	2472.79	2560.90	2986.37
80	12993.25	387.02	716.95	2554.63	2496.18	3272.08
100	17077.15	321.49	1009.38	2635.42	2823.59	3016.98

Πίνακας 7: Οι Χρόνοι t_C και t_G

Poisson Arrival Rate (Req/min)	t_C (msec)			t_G (msec)		
	SOAP	CORBA	RMI	SOAP	CORBA	RMI
20	65.12	20.01	34.84	84.79	26.21	38.24
40	100.05	31.14	54.55	154.32	35.33	54.22
60	304.45	85.12	103.78	227.31	69.03	102.75
80	316.28	108.23	164.21	198.25	73.32	114.89
100	341.98	100.15	178.09	245.89	52.67	185.47

Η δεύτερη (και αναμφίβολα εντυπωσιακότερη) οφθαλμοφανής παρατήρηση, είναι βέβαια οι ανταγωνιστικές τιμές που παρουσιάζει ο μέσος συνολικός χρόνος εξυπηρέτησης ($\overline{t_{TOTAL}}$) στην υλοποίηση της CORBA, σε σχέση με την κλασική υλοποίηση που στηρίζεται στο MLP, παρά το γεγονός ότι στην τελευταία δεν συνυπολογίζονται στον $\overline{t_{TOTAL}}$ οι χρόνοι t_{CL} , t_M και t_G . Η σύγκλιση των τιμών του $\overline{t_{TOTAL}}$ στις δύο περιπτώσεις οφείλεται στους εξής δύο παράγοντες. Αφενός στην χρήση του HTTP ως πρωτόκολλο μεταφοράς από πλευράς του MLP, έναντι του πολύ ταχύτερου ΙΙΟΡ της CORBA και αφετέρου στη διαφορετική φύση των μηνυμάτων που αποστέλλονται πάνω από τα συγκεκριμένα πρωτόκολλα μεταφοράς (αποστολή απλού κειμένου και δυαδικής μορφής αντίστοιχα). Κατά συνέπεια, οι επιπρόσθετες καθυστερήσεις (t_{CL} , t_M και t_G), που εμφανίζονται στην υλοποίηση της CORBA, ισοσταθμίζονται με τη διόλου ασήμαντη υπεροχή που η ίδια εμφανίζει, όσον

αφορά τη μετάδοση της πληροφορίας, επιτυγχάνοντας πολύ μικρότερο t_C έναντι της υλοποίησης του MLP. Το γεγονός αυτό έχει σαν αποτέλεσμα η υλοποίηση που χρησιμοποιεί την τεχνολογία μεσισμικού της CORBA να επιτυγχάνει συγκρίσιμη απόδοση με αυτή του MLP, ιδιαίτερα για υψηλά φορτία κίνησης (Σχήμα 63).

Αντίστοιχα συγκρίσιμη απόδοση με αυτή του πρωτοκόλλου MLP όσον αφορά στον συνολικό χρόνο απόδοσης t_{TOTAL} παρουσιάζει και η περίπτωση χρήσης του Διαμεσολαβητή Δικτύου. Αυτό οφείλεται κυρίως στο γεγονός ότι στην υλοποίηση αυτή δεν χρησιμοποιούνται για την υλοποίηση της επικοινωνίας των οντοτήτων τεχνολογίες λογισμικού, οι οποίες κατά γενική ομολογία, παρόλα τα θετικά τους στοιχεία, αποτελούν βαριές τεχνολογίες που προσθέτουν επεξεργαστικό φόρτο στην επικοινωνία οντοτήτων [7]. Άλλος ένας παράγοντας στον οποίο οφείλεται η γρήγορη απόκριση του συστήματος με τον Διαμεσολαβητή Δικτύου είναι ο τρόπος σχεδιασμού και υλοποίησης του. Ο διαμεσολαβητής Δικτύου έχει σχεδιαστεί και υλοποιηθεί προκειμένου να καλύπτει υψηλές απαιτήσεις στον αριθμό των υποσυστημάτων που φιλοξενεί, με αποτέλεσμα την αποδοτικότερη χρήση των διαθέσιμων πόρων για την υποστήριξη και εξυπηρέτηση μεγάλου όγκου γεγονότων. Ο λόγος που οι μετρήσεις που παρατίθενται φτάνουν μόνο μέχρι τις 100 αιτήσεις ανά λεπτό είναι για να είναι τα αποτελέσματα συγκρίσιμα με αυτά των άλλων υλοποιήσεων.

Επίσης πρέπει να αναφερθούν και να ληφθούν υπ' όψιν και τα καλύτερα τεχνικά χαρακτηριστικά του εξυπηρετητή που φιλοξενούσε τον Διαμεσολαβητή Δικτύου. Τα εξαιρετικά όμως χαμηλά αποτελέσματα δίνουν την δυνατότητα να συμπεράνει κανείς ότι ακόμα και με χαμηλότερων δυνατοτήτων PC να είχαν διενεργηθεί τα πειράματα, οι χρόνοι της συγκεκριμένης υλοποίησης θα ήταν και πάλι ανταγωνιστικοί.

Όσον αφορά στην σύγκριση των επιμέρους χρόνων, αυτή είναι δυνατή μόνο σε λογικό επίπεδο. Αυτό οφείλεται στο ότι η τοπολογία των πειραμάτων στις περιπτώσεις του MLP και της διεπαφής Κινητικότητας είναι εντελώς διαφορετική από αυτή της υλοποίησης με τον Διαμεσολαβητή Δικτύου. Λογικά μπορεί να γίνει η παρακάτω αντιστοίχιση. Σαν t_{CL} στην περίπτωση του Διαμεσολαβητή Δικτύου μπορούμε να θεωρήσουμε το άθροισμα των χρόνων t_{C1} και t_{IM} . Με το πέρας αυτού του χρόνου, έχει δημιουργηθεί το κατάλληλο γεγονός-αίτηση, όπως αντίστοιχα με το πέρας του χρόνου t_{CL} έχει δημιουργηθεί η κατάλληλη αίτηση Parlay. Εν συνεχεία ο Mediator προωθεί το γεγονός στο Location Subsystem το οποίο ουσιαστικά πραγματοποιεί την ίδια διαδικασία με την Πύλη Εντοπισμού Θέσης (Gateway) η οποία είναι η μετάφραση της αίτησης σε MLP μήνυμα, ή αντίστροφα την MLP απόκριση σε απόκριση

ανάλογη της υλοποίησης. Συνεπώς, μπορεί να ειπωθεί ότι το άθροισμα των χρόνων t_C , t_M και t_{GW} ταυτίζονται λογικά με το άθροισμα των χρόνων t_{MD} και t_L .

4.4 Αναφορές

- [1] N. Tselikas, E. Koutsoloukas, S. Kapellaki, E. Chaniotakis, I. Venieris, “An OSA/Parlay-Based Middleware Architecture for Location-Based Services”, *Wireless Personal Communications*, Volume 30, Issue 2-4, September 2004, Pages 247- 265, Kluwer Academic Publishers.
- [2] S. Kapellaki, E. Koutsoloukas, N. Tselikas, E. Chaniotakis, G. Prezerakos, I. Venieris, “The “Le” interface: Performance Evaluation of 2-tier and 3-tier 3GPP compliant realizations”, In Proc. of IEEE International Conference of Communications, ICC 2005, Seoul, May 2005.
- [3] N. Tselikas, S. Kapellaki, E. Koutsoloukas, S. Maniatis, I. Venieris, “OSA/Parlay based middleware architecture for homogeneous service provision in Next Generation Networks”, accepted to NGNM’04 workshop of Networking 2004 Conference
- [4] 3GPP TS 23.271 V7.8.0 Functional Stage 2 Description of Location Services (LCS), Release 7, March 2007
- [5] S. Vinoski, The Performance Presumption, *IEEE Internet Computing*, vol. 7(2), pp. 89-90, April 2003
- [6] S. Floyd and V. Paxson: Difficulties in Simulating the Internet, *IEEE/ACM Transactions on Networking*, vol. 9, no. 4, pp. 392--403, Aug. 2001.
- [7] Kenneth J. Turner (Editor), Evan H. Magill, David J. Marples, "Service Provision – Technologies for Next Generation Communications", ISBN: 0-470-85066-3, (Wiley), January 2004.

5ο Κεφάλαιο

Συμπεράσματα και μελλοντικές εξελίξεις

Στο τελευταίο αυτό κεφάλαιο, παρατίθεται μία επισκόπηση των όσων παρουσιάστηκαν στα κυρίως κεφάλαια της διατριβής. Στόχος της επισκόπησης αυτής είναι η επιβεβαίωση της επίτευξης των στόχων της εργασίας σε σχέση με τις αρχικές απαιτήσεις που είχαν τεθεί. Επίσης παρουσιάζεται ένα πιθανό σενάριο για την ανάπτυξη του συστήματος σε μεγάλη κλίμακα, καθώς και ένα πιθανό επιχειρηματικό μοντέλο το οποίο θα μπορούσε να ακολουθηθεί για την διείσδυση της προτεινόμενης αρχιτεκτονικής στο χώρο της αγοράς. Τέλος, η τελευταία ενότητα του κεφαλαίου αποτελεί τις πιθανές και προτεινόμενες μελλοντικές εξελίξεις από τεχνικής πλευράς για την περαιτέρω ανάπτυξη και επέκταση του παρόντος έργου.

5.1 Εισαγωγή

Στο παρόν κεφάλαιο παρατίθεται αρχικά μία γενική επισκόπηση των όσων παρουσιάστηκαν στα προηγούμενα κεφάλαια, τόσο για την μοντελοποίηση της πληροφορίας, όσο και για τις επιλογές που ακολουθήθηκαν όσον αφορά στον σχεδιασμό και την υλοποίηση του συστήματος των διαμεσολαβητών, με σκοπό να διαπιστωθεί κατά πόσο επιτεύχθηκαν οι αρχικοί στόχοι και οι απαιτήσεις που είχαν τεθεί.

Εν συνεχεία, παρατίθενται κάποιες πρώτες σκέψεις σχετικά με την ανάπτυξη του συστήματος σε μεγάλη κλίμακα και την εισχώρηση του στο χώρο της αγοράς. Στο πλαίσιο αυτό παρατίθενται πιθανά σενάρια ανάπτυξης του συστήματος τόσο σε τοπική όσο και σε μεγάλη κλίμακα, καθώς επίσης και ένα πιθανό επιχειρηματικό μοντέλο που θα μπορούσε να ακολουθηθεί.

Τέλος, η διατριβή ολοκληρώνεται με δύο προτάσεις επέκτασης του συστήματος. Η μία αφορά στη σύνθεση υπηρεσιών με χρήση προτύπων χρήσης (usage patterns) και πώς η ιδέα αυτή θα μπορούσε να ενσωματωθεί στην προτεινόμενη αρχιτεκτονική. Η δεύτερη αποτελεί μία προσπάθεια αντιμετώπισης του ζητήματος της ιδιωτικότητας (privacy). Το ζήτημα αυτό αποτελεί σήμερα το πιο σημαντικό θέμα στον χώρο των συστημάτων παροχής υπηρεσιών βασισμένων στο πλαίσιο χρήσης τους, το οποίο εγείρει τόσο τεχνικά όσο ηθικά και νομικά θέματα. Στο πλαίσιο αυτό παρουσιάζεται μία εμπλουτισμένη έκδοση της προτεινόμενης αρχιτεκτονικής με στόχο την αντιμετώπιση των ζητημάτων αυτών.

5.2 Συμπεράσματα

Ο βασικός λόγος, ο οποίος λειτούργησε σαν κινητήρια δύναμη, για την πραγματοποίηση της συγκεκριμένης εργασίας ήταν η ανάγκη για μία όσο το δυνατόν πιο ολοκληρωμένη λύση υποστήριξης παροχής υπηρεσιών βασισμένων στο πλαίσιο χρήσης τους. Η ανάγκη αυτή ήταν εύκολο να εντοπιστεί μέσα από τη μελέτη της εκτενούς σχετικής βιβλιογραφίας. Οι περισσότερες από τις πρώτες απόπειρες για την δημιουργία context aware συστημάτων οδήγησαν στην υλοποίηση συγκεκριμένων εφαρμογών εντός ενός συγκεκριμένου τομέα (domain specific, π.χ. intelligent spaces) με χρήση περιορισμένων ειδών πληροφορίας context. Στην κατηγορία αυτή ανήκει φυσικά και η πλατφόρμα παροχής υπηρεσιών αναζήτησης βάσει θέσης και χαρακτηριστικών του χρήστη που υλοποιήθηκε στο πλαίσιο της παρούσας διατριβής και παρουσιάστηκε στο πρώτο κεφάλαιο. Τα συστήματα αυτά κατά συνέπεια έπασχαν όσον αφορά στην επεκτασιμότητά τους και στην διαλειτουργικότητά τους με άλλες εφαρμογές. Έτσι, ο κόσμος της έρευνας στράφηκε σε πιο γενικευμένες λύσεις.

Ακόμα όμως και αυτές δεν αποτέλεσαν πανάκεια στο πρόβλημα αναζήτησης μιας γενικευμένης λύσης για την υποστήριξη παροχής context-aware υπηρεσιών. Σχεδόν στο σύνολό τους οι προτεινόμενες λύσεις εντοπίζονται στην περιοχή των κινητών τερματικών, ακολουθούν συγκεντρωτικό αρχιτεκτονικό μοντέλο, είναι domain specific. Οι περισσότερες εστιάζουν στον τομέα της ανάκτησης δεδομένων από φυσικούς αισθητήρες. Η κάθε μία υλοποιεί και χρησιμοποιεί εκ νέου το δικό της μοντέλο πληροφοριών. Υλοποιούν πολύπλοκους μηχανισμούς για την εξαγωγή συμπερασμάτων και πληροφορίας υψηλότερου επιπέδου και ποιότητας, και πολλές φορές εμπλέκονται και στην διαδικασία προσαρμογής των υπηρεσιών βάσει της πληροφορίας αυτής, καταλήγοντας έτσι σε εξαιρετικά πολύπλοκα, μη επεκτάσιμα, περιορισμένης χρήσης και κλίμακας συστήματα.

Ο στόχος λοιπόν, ήταν αρχικά, ο συνολικός και καθολικός καθορισμός και μοντελοποίηση της πληροφορίας context με ευρέως αποδεκτές μεθόδους και εργαλεία, προκειμένου τα μέρη που εμπλέκονται στην διαδικασία ανάκτησης και εκμετάλλευσης του context να χρησιμοποιούν κοινές αναφορές και ονοματολογία. Και στη συνέχεια, ο σχεδιασμός και υλοποίηση ενός ανοικτού κατανεμημένου επεκτάσιμου ανθρωποκεντρικού συστήματος με χαρακτηριστικά και δυνατότητες που να καλύπτουν τις απαιτήσεις για μία ολοκληρωμένη λύση.

Όσον αφορά στον πρώτο μέρος του στόχου, αφού μελετήθηκαν οι υπάρχουσες λύσεις και διαπιστώθηκε η ετερογένεια που υπάρχει στον συγκεκριμένο τομέα, καθώς και το πλήθος και η διαφορετικότητα των πληροφοριών που απαιτείται να μοντελοποιηθούν, θεωρήσαμε ότι ίσως το σημαντικότερο χαρακτηριστικό ενός τέτοιου μοντέλου θα ήταν η χρήση ευρέως αποδεκτών τεχνολογιών, και κυρίως η χρήση ήδη καθιερωμένων προτύπων, προκειμένου να διευκολυνθεί η αποδοχή και υιοθέτησή του. Λαμβάνοντας βεβαίως υπ'όψιν και όλες τις υπόλοιπες απαιτήσεις που αναφέρονται στο δεύτερο κεφάλαιο και είναι η δομή, η ανταλλαξιμότητα, η δυνατότητα σύνθεσης – αποσύνθεσης – επανασύνθεση καθώς και η ομοιομορφία, η πλέον επικρατούσα προσέγγιση ήταν αυτή ενός mark-up scheme μοντέλου βασισμένου σε ήδη υπάρχοντα πρότυπα. Έτσι, πρωτίστως επιλέχθηκε να ακολουθηθεί το Common Information Model (CIM) της Distributed Management Task Force (DTMF) [1] και οι συμβάσεις του. Έτσι, κατασκευάστηκε το μοντέλο πληροφορίας του συστήματος με χρήση της γλώσσας UML [2], με αποτέλεσμα την πλήρως κατανοητή από τον άνθρωπο γραφική αναπαράσταση των πληροφοριών. Σε δεύτερο επίπεδο, ήταν απαραίτητη η αναπαράσταση των πληροφοριών αυτών και με τρόπο κατανοητό από τον υπολογιστή. Διατηρώντας την αρχή της χρήση προτύπων, για την αναπαράσταση των δεδομένων

ακολουθήθηκαν οι συμβάσεις και η δομή του Generic User Profile (GUP) της 3GPP [3] για τον σχεδιασμό του προφίλ πληροφορίας πλαισίου χρήσης υπηρεσίας (Context Profile, CoP). Το CoP ουσιαστικά αποτελεί τη σύνθεση των προφίλ χρήστη, τερματικού, υπηρεσίας και δικτύου. Για το περιεχόμενο των προφίλ λήφθηκαν υπ' όψιν πρότυπα όπως το UAProf [4], το CC/PP [5] και άλλα. Για την δημιουργία των προφίλ χρησιμοποιήθηκε η XML.

Ακολουθώντας το πρότυπο GUP της 3GPP και ακολουθώντας την ιεραρχική και βασισμένη σε στοιχεία δομή που προδιαγράφει, αυτομάτως επιτεύχθηκε ο στόχος για ομοιόμορφη δομή της πληροφορίας ανεξαρτήτως του είδους της. Επίσης, η αποτελούμενη από στοιχεία δομή, τα οποία καθορίζονται από XML σχήματα, παρέχει τη δυνατότητα της σύνθεσης – αποσύνθεσης – επανασύνθεσης της πληροφορίας. Τέλος, το πρόβλημα της ανταλλαξιμότητας της πληροφορίας ανάμεσα στις εμπλεκόμενες οντότητες, με την επιλογή της συγκεκριμένης λύσης, επιλύεται με τη χρήση των καθιερωμένων μηχανισμών σειριοποίησης και αποσειριοποίησης των XML αρχείων.

Τα μοντέλα πληροφορίας και δεδομένων που αναπτύχθηκαν περιορίστηκαν σε πληροφορίες προφίλ που αφορούσαν τον χρήστη, το τερματικό του, τις παρεχόμενες υπηρεσίες και τα ενεργά δίκτυα. Ωστόσο, ο τρόπος που δομήθηκαν τα προφίλ, και κατά συνέπεια και το συνολικό προφίλ πληροφορίας πλαισίου χρήσης, επιτρέπει ανά πάσα στιγμή την επέκτασή του με νέα υποπροφίλ με τη μορφή νέων στοιχείων που θα περιγράφονται από κατάλληλα XML σχήματα. Ένα παράδειγμα θα μπορούσε να είναι ένα προφίλ με δεδομένα από αισθητήρες του περιβάλλοντος χώρου κλπ.

Μετά την ολοκλήρωση της μοντελοποίησης της πληροφορίας, ακολούθησε ο σχεδιασμός του συστήματος διατήρησης, διαχείρισης και παροχής της. Ο βασικός στόχος προς επίτευξη ήταν ο σχεδιασμός ενός ευέλικτου και επεκτάσιμου συστήματος, ανεξάρτητου υπηρεσιών, τερματικών και δικτύων, το οποίο να αντιμετωπίζει με αποτελεσματικότητα το ζήτημα της πολυπλοκότητας που παρουσιάζουν τέτοιου είδους συστήματα.

Δύο ήταν οι βασικές αποφάσεις που οδήγησαν στην κάλυψη των παραπάνω απαιτήσεων. Πρώτον, το σύστημα δεν θα εξυπηρετούσε διαδικασίες προσαρμογής των υπηρεσιών βάσει του CoP. Αλλά θα περιοριζόταν στην συλλογή και διαχείριση της προκαθορισμένης από το μοντέλο πληροφορίας και της παροχής της στις υπηρεσίες με ομοιόμορφο και προτυποποιημένο τρόπο. Οι διαδικασίες προσαρμογής θεωρήθηκε ότι ανήκουν και πρέπει να υλοποιούνται από τον πάροχο της εκάστοτε υπηρεσίας. Και δεύτερον το σύστημα θα ακολουθούσε την σχεδιαστική αρχή των αποτελούμενων από υποσυστήματα Διαμεσολαβητών και της Προσωπικής Συσκευής Χρήστη.

Η Προσωπική Συσκευή Χρήστη εξυπηρετεί τόσο την φορητότητα όσο και την κατά μία έννοια ιδιωτικότητα των προσωπικών δεδομένων του χρήστη. Με τη χρήση της, ο χρήστης έχει ανά πάσα στιγμή μαζί του τα προσωπικά του στοιχεία που απαιτούνται για την προσαρμογή των υπηρεσιών ανεξαρτήτως τοποθεσίας και τερματικού. Επίσης ο έλεγχος και η διαχείριση των προσωπικών δεδομένων του πραγματοποιείται εξ' ολοκλήρου από τον ίδιο. Ωστόσο τα καίρια ζητήματα της ασφάλειας και της ιδιωτικότητας δεν αντιμετωπίζονται πλήρως και παραμένουν έως ένα βαθμό, από τη στιγμή που τα δεδομένα αυτά μεταναστεύουν από την προσωπική συσκευή τόσο προς τις οντότητες του συστήματος όσο και προς τις υπηρεσίες. Για τον λόγο αυτό σε επόμενη ενότητα παρουσιάζεται μία αρχιτεκτονική λύση η οποία σαν επέκταση του παρόντος συστήματος θα μπορούσε να έδινε λύση και σε αυτό το πρόβλημα.

Η οντότητα του διαμεσολαβητή, από την άλλη, έρχεται τελικά να καλύψει τις περισσότερες από τις απαιτήσεις που είχαν τεθεί αρχικά. Παρέχει επεκτασιμότητα ως προς τις εγγενείς λειτουργίες του συστήματος, ανοικτότητα για την ενσωμάτωση νέων ή και ήδη υπάρχουσών υπηρεσιών, καθιστά το σύστημα ανεξάρτητο εφαρμογών και υπηρεσιών καθώς και πεδίου εφαρμογής (domain), παρέχει ευελιξία, δυνατότητα κατανομής των λειτουργιών του συστήματος, είτε με την ύπαρξη πολλών διαμεσολαβητών ίδιων αρμοδιοτήτων, είτε με την κατανομή των λειτουργιών και την δημιουργία διαμεσολαβητών με διαφορετικές αρμοδιότητες. Όλα τα παραπάνω οφείλονται στη δομή της οντότητας του διαμεσολαβητή, ο οποίος αποτελείται από υποσυστήματα και τον μεσαζόντα.

Κάθε υποσύστημα παρέχει λειτουργικότητα διαχείρισης διαφορετικού είδους πληροφορίας, ανεξάρτητα από τα υπόλοιπα υποσυστήματα. Η επικοινωνία πραγματοποιείται διαφανώς με κατάλληλους ασύγχρονους μηχανισμούς βασισμένους σε γεγονότα, είτε μέσω του Μεσαζόντα, είτε μέσω του Υποσυστήματος Επικοινωνίας Διαμεσολαβητών για την περίπτωση υποσυστημάτων τοποθετημένων σε απομακρυσμένους διαμεσολαβητές. Το μοντέλο που ακολουθείται για την επικοινωνία ανάμεσα στα υποσυστήματα είναι αυτό των συστημάτων publish-subscribe. Η ανεξαρτησία αυτή, η χαλαρή διασύνδεση των υποσυστημάτων, έχει σαν αποτέλεσμα μία αρχιτεκτονική ευέλικτη και επεκτάσιμη, καθώς επιτρέπει την προσθήκη νέων υποσυστημάτων - λειτουργιών χωρίς να επηρεάζεται η υπάρχουσα υποδομή.

Η ιδέα των ανεξάρτητων υποσυστημάτων διαχείρισης θα μπορούσε να ειπωθεί ότι πηγάζει, ή καλύτερα ότι συμβαδίζει, και με τη δομή του μοντέλου δεδομένων. Το Προφίλ Πληροφορίας Πλαισίου χρήσης αποτελείται από υπο-προφίλ. Για τη διατήρηση, ανάκτηση

και διαχείριση κάθε υπο-προφίλ είναι υπεύθυνο και ένα υποσύστημα. Στην περίπτωση λοιπόν, που το προφίλ εμπλουτιστεί με ένα νέο υπο-προφίλ, όπως αναφέρθηκε παραπάνω, στο σύστημα θα πρέπει να εισαχθεί και το κατάλληλο υποσύστημα για τη διαχείριση του. Το γεγονός αυτό δεν επιφέρει καμία δυσκολία λόγω των σχεδιαστικών αρχών που ακολουθήθηκαν. Επίσης, εάν το σύστημα έπρεπε να εγκατασταθεί στο χώρο ενός νοσοκομείου, για να εξυπηρετήσει τόσο ανάγκες ασθενών όσο και ανάγκες του προσωπικού, θα μπορούσε να εμπλουτιστεί τόσο το προφίλ με σχετικά στοιχεία, όσο και ο Διαμεσολαβητής με τα κατάλληλα υποσυστήματα.

Η ιδέα των υποσυστημάτων εξυπηρετεί, εκτός από την ενσωμάτωση νέων εγγενών λειτουργιών στο σύστημα, την περίπτωση που απαιτείται η αλληλεπίδραση του συστήματος με κάποια εξωτερική υπάρχουσα οντότητα ή υπηρεσία, όπως για παράδειγμα ο Εξυπηρετητής Θέσης (Location Server) κάποιου παρόχου δικτύου (network provider) για την ανάκτηση της θέσης του χρήστη ή η εφαρμογή κάποιου Ανεξάρτητου Παροχέα Υπηρεσιών (Independent Service Provider, ISP) η οποία αλληλεπιδρά με το σύστημα για την ανάκτηση πληροφορίας απαραίτητης για την εκτέλεση της υπηρεσίας. Σε αυτήν την περίπτωση προστίθεται στον Διαμεσολαβητή ένα υποσύστημα προσαρμογής, που ονομάζεται Διαχειριστής Εξωτερικής Διεπαφής (Interface Manager, IM), το οποίο συμπεριφέρεται ακριβώς όπως τα υπόλοιπα υποσυστήματα, αποτελεί την διεπαφή με την εξωτερική οντότητα/υπηρεσία και μέσω αυτού πραγματοποιείται η ροή πληροφορίας ανάμεσα στο σύστημα και την εξωτερική οντότητα. Οποιαδήποτε προσαρμογή απαιτείται πραγματοποιείται εντός του Διαχειριστή Εξωτερικής Διεπαφής και η εξωτερική οντότητα παραμένει ανέπαφη. Επίσης, αν η εξωτερική οντότητα απαιτεί διαφορετικής δομής ή επιπέδου πληροφορία από αυτήν που παρέχει το σύστημα, η λειτουργικότητα μετατροπής της υλοποιείται εντός του Διαχειριστή Εξωτερικής Διεπαφής. Αυτό έχει σαν αποτέλεσμα οι λειτουργίες πυρήνα του συστήματος να διατηρούνται ανεξάρτητες από τις εκάστοτε απαιτήσεις προσαρμογής της πληροφορίας ανάλογα με την υπηρεσία, με αποτέλεσμα οι πολύπλοκες διαδικασίες επεξεργασίας των πληροφοριών να μειώνονται στο ελάχιστο και να είναι δυνατόν να πραγματοποιούνται και να ελέγχονται κατά περίπτωση.

Συνεπώς, η σχεδιαστική ιδέα των διαμεσολαβητών με τα ανεξάρτητα υποσυστήματα ικανοποιεί πλήρως τις προαπαιτήσεις για επεκτασιμότητα και ανοικτότητα καθώς παρέχει τη δυνατότητα για προσθήκες στο σύστημα είτε εγγενών είτε εξωτερικών λειτουργιών με ελάχιστο υπολογιστικό, χρονικό και οικονομικό κόστος. Επίσης, με την υλοποίηση των

υποσυστημάτων τύπου Διαχειριστή Εξωτερικής Διεπαφής παρέχεται η πληροφορία πλαισίου στις εφαρμογές με ομοιόμορφο και προτυποποιημένο τρόπο.

Η επιλογή για την ύπαρξη διαφορετικών ειδών διαμεσολαβητών τοποθετημένων και σε διαφορετικά τοπολογικά σημεία (τερματικό και δίκτυο) εξυπηρετεί την αποφυγή μοναδικού σημείου αστοχίας και καλύπτει τις απαιτήσεις για εφεδρεία. Επίσης, η φύση των διαμεσολαβητών παρέχει μεγάλη ευελιξία όσον αφορά τη διαμόρφωση του δικτύου των Διαμεσολαβητών Δικτύου, κυρίως με στόχο τη βέλτιστη κατανομή του φορτίου και την αποτελεσματική λειτουργία του συστήματος. Το χαρακτηριστικό αυτό οδηγεί στο συμπέρασμα ότι το σύστημα θα μπορούσε να εφαρμοστεί και σε περιπτώσεις μεγάλης κλίμακας. Σε επόμενη ενότητα περιγράφονται οι πιθανές κατευθύνσεις ως προς το συγκεκριμένο ζήτημα.

5.3 Σενάρια Ανάπτυξης του Συστήματος – Κλιμάκωση και Κατανομή

Στην ενότητα αυτή επιχειρείται μία περιγραφή των πιθανών σεναρίων ανάπτυξης του συστήματος τόσο σε τοπικό όσο και σε μεγαλύτερης κλίμακας επίπεδο [6]. Η πρώτη περίπτωση αφορά την ανάπτυξη του εντός ενός ιδιωτικού δικτύου με την συμμετοχή περιορισμένου αριθμού ετερογενών συσκευών, ενώ η δεύτερη περιλαμβάνει την εμπλοκή πολλών λειτουργικών δικτύων και παρόχων υπηρεσιών προκειμένου να παρασχεθούν σε μεγάλο αριθμό χρηστών διαφορετικές υπηρεσίες βασισμένες στο πλαίσιο χρήσης τους.

Πριν όμως από την περιγραφή των σεναρίων, κρίνεται σκόπιμο να αναφερθούν οι διαφορετικές μορφές που θα μπορούσε να πάρει ένας Διαμεσολαβητής σε μία πιο διευρυμένη αρχιτεκτονική, η οποία θα πλησίαζε την ανάπτυξη του συστήματος υπό πραγματικές συνθήκες. Πέρα λοιπόν από τους ήδη αναφερθέντες Διαμεσολαβητές, αυτούς του δικτύου και του τερματικού, θα μπορούσαμε να ορίσουμε και τον Διαμεσολαβητή Υπηρεσιών. Αυτό το είδος Διαμεσολαβητή υπάρχει στην πλευρά του κάθε Ανεξάρτητου Παρόχου Υπηρεσιών, ο οποίος τον διατηρεί και τον διαχειρίζεται. Ουσιαστικά αποτελεί την διεπαφή με το υπόλοιπο σύστημα και εκεί βρίσκονται εγκατεστημένα όλα τα Υποσυστήματα Διαχειριστή Εξωτερικής Διεπαφής των υπηρεσιών του παρόχου. Ουσιαστικά, η εισαγωγή ενός τέτοιου είδους διαμεσολαβητή προκύπτει από τον διαχωρισμό του Διαμεσολαβητή Δικτύου σε δύο, διατηρώντας στον τελευταίο τις αυστηρά διαχειριστικές λειτουργίες των πληροφοριών και τοποθετώντας στον Διαμεσολαβητή Υπηρεσιών όλες τις σχετικές με τις υπηρεσίες λειτουργίες.

Επίσης οι Διαμεσολαβητές θα μπορούσαν να διαχωριστούν σε Τοπικούς (Local) και Κοινόχρηστους (Public). Οι Τοπικοί Διαμεσολαβητές βρίσκονται εντός ενός μικρού τοπικού δικτύου και οι αλληλεπιδράσεις πραγματοποιούνται κατά περίπτωση (ad-hoc). Ένα τυπικό σενάριο τοπικών διαμεσολαβητών είναι αυτό ενός χρήστη με ένα PDA μέσω του οποίου ο χρήστης διαχειρίζεται το στερεοφωνικό του ή τη συσκευή του video. Στην περίπτωση αυτή πραγματοποιούνται αλληλεπιδράσεις ανάμεσα στους διαμεσολαβητές των συσκευών χωρίς να απαιτείται η επέμβαση του παρόχου υπηρεσιών, του λειτουργού του δικτύου ή η ύπαρξη ενός Διαμεσολαβητή Δικτύου ή Υπηρεσιών.

Σαν Κοινόχρηστοι Διαμεσολαβητές από την άλλη πλευρά ορίζονται αυτοί που λαμβάνουν μέρος σε ένα καθολικό σενάριο παροχής υπηρεσιών. Στην περίπτωση αυτή ο χρήστης είναι συνδεδεμένος μέσω ενός Δικτύου Πρόσβασης και υπάρχουν πάροχοι υπηρεσιών και λειτουργοί δικτύων που αντίστοιχα διαχειρίζονται Διαμεσολαβητές Υπηρεσιών και Δικτύου. Στο σενάριο αυτό ο Διαμεσολαβητής Τερματικού μπορεί να είναι είτε ένας Τοπικός Διαμεσολαβητής (στον υπολογιστή του χρήστη στο σπίτι του), είτε ένας Κοινόχρηστος Διαμεσολαβητής Τερματικού (σε έναν κοινόχρηστο υπολογιστή).

Σενάριο Τοπικής Ανάπτυξης του συστήματος:

Το σενάριο αυτό επικεντρώνεται στην περίπτωση της ύπαρξης ενός μικρού αριθμού Διαμεσολαβητών εντός ενός τοπικού δικτύου (LAN) μικρής κλίμακας, οι οποίοι αλληλεπιδρούν για την παροχή υπηρεσιών βασισμένων στο πλαίσιο χρήσης τους. Ένα τέτοιο παράδειγμα θα μπορούσε να είναι ένα σπίτι εξοπλισμένο με ασύρματο τοπικό δίκτυο (WLAN), μέσω του οποίου διασυνδέονται όλα τα τερματικά, αλλά και όλες οι άλλες συσκευές, οι οποίες θεωρητικά θα μπορούσαν κατόπιν κατάλληλων προσαρμογών να φιλοξενήσουν έναν διαμεσολαβητή τερματικού. Τέτοιες συσκευές θα μπορούσαν να είναι η συσκευή εγγραφής DVD, η συσκευή εγγραφής Video, η τηλεόραση, οι ηλεκτρικές συσκευές κλπ. Οι διαμεσολαβητές εντός των συσκευών είναι υπεύθυνοι για την διαμόρφωση τους (configuration) και δρουν επικουρικά προς τον χρήστη προκειμένου να φέρει εις πέρας διαδικασίες όπως την εγγραφή ενός DVD και την παρουσίαση του μέσω της κοντινότερης του οθόνης ή τη ρύθμιση της θέρμανσης του σπιτιού με αυτοματοποιημένο και ανάλογα με τις συνθήκες τρόπο.

Αντίστοιχο σενάριο θα μπορούσε να βρει εφαρμογή και εντός ενός γραφείου με υποδομή τοπικού δικτύου που διασυνδέει συσκευές όπως σταθερούς υπολογιστές, εκτυπωτές, φαξ κλπ. Οι διαμεσολαβητές εντός των συσκευών μπορούν να παρέχουν υπηρεσίες μοναδικής και αυτόματης πιστοποίησης των χρηστών για την πρόσβασή τους σε αυτές, ανακάλυψη

συσκευών (π.χ. ανακάλυψη του κοντινότερου εκτυπωτή), διαμόρφωση του υπολογιστικού περιβάλλοντος ανάλογα με το χρήστη κλπ. Για παράδειγμα ένας χρήστης θα μπορούσε να μεταφέρει αυτόματα το αρχείο που χρησιμοποιεί από το ένα PC στο άλλο κλπ.

Και στα δύο παραπάνω σενάρια εκτός από τους διαμεσολαβητές τερματικών θα μπορούσε να έχει ρόλο και ένα Τοπικός Διαμεσολαβητής Υπηρεσιών. Ο ρόλος ενός τέτοιου είδους διαμεσολαβητή θα ήταν να παρέχει ένα σύνολο από προηγμένες τοπικές υπηρεσίες όπως για παράδειγμα Υπηρεσίες Ροής Πολυμέσων, όπως μουσικής ή ταινιών από ένα κεντρικό εξυπηρετητή εντός του σπιτιού ή του γραφείου στα διάφορα τερματικά.

Για την υποστήριξη ωστόσο των παρεχόμενων υπηρεσιών δεν απαιτείται η ύπαρξη ούτε του διαμεσολαβητή δικτύου, ούτε πρόσβαση σε εξωτερικό δίκτυο. Συνεπώς οι απαιτήσεις για τέτοιου είδους συστήματα είναι:

- Αλληλεπίδραση μεταξύ των Διαμεσολαβητών Τερματικού, η οποία δεν υπήρχε στο σύστημα όπως περιγράφηκε στα προηγούμενα κεφάλαια.

- Η αλληλεπίδραση να πραγματοποιείται κατά περίπτωση (ad-hoc approach), με την αυτόματη ανακάλυψη των πόρων (resources discovery) να παίζει σημαντικό ρόλο.

Μία τέτοιου είδους ανάπτυξη θα απαιτούσε την χρήση τεχνολογίας αυτόματης ανακάλυψης πόρων όπως για παράδειγμα το UPnP, δομή και υπηρεσίες συστημάτων P2P ή ακόμα κάποια υπηρεσία ευρείας μετάδοσης (broadcast service). Έτσι, θα ήταν δυνατή η ανακάλυψη μεταξύ των διαμεσολαβητών για την παροχή προς τους χρήστες των διαφόρων υπηρεσιών. Συνεπώς, σε μία τέτοια διαμόρφωση του συστήματος, όλοι οι διαμεσολαβητές είναι διαμεσολαβητές τερματικού και όλοι γνωρίζουν την ύπαρξη και μπορούν να επικοινωνούν με όλους τους άλλους. Έτσι αυτομάτως, σε ένα δίκτυο N διαμεσολαβητών, για να επιτευχθεί κάτι τέτοιο θα χρειαζόντουσαν $N*(N-1)$ συσχετισμοί, πράγμα που σημαίνει ότι ένα τέτοιο σύστημα θα μπορούσε να πραγματοποιηθεί με αποτελεσματικό τρόπο μόνο εφόσον το N θα ήταν πολύ μικρό. Είναι προφανές λοιπόν, ότι μία τέτοια ανάπτυξη δεν θα μπορούσε να εφαρμοστεί σε μεγάλη κλίμακα. Η ένα προς ένα επικοινωνία ανάμεσα στους διαμεσολαβητές καθώς και η έλλειψη οποιουδήποτε μηχανισμού εξισορρόπησης του φορτίου ή κατανομής των λειτουργιών αποτελούν περιοριστικούς παράγοντες

Σενάριο Ανάπτυξης του συστήματος σε Μεγάλη Κλίμακα:

Η περίπτωση της ανάπτυξης του συστήματος σε μεγάλη κλίμακα βασίζεται στην αρχιτεκτονική που παρουσιάστηκε στο 3^ο κεφάλαιο. Αποτελείται από τον Διαμεσολαβητή Δικτύου, που τον διαχειρίζεται ο λειτουργός του συστήματος, από τους Διαμεσολαβητές

Υπηρεσιών, που βρίσκονται στην πλευρά των διαφόρων Παρόχων Υπηρεσιών, και από τους Διαμεσολαβητές Τερματικών των χρηστών. Οι χρήστες συνδέονται στους Διαμεσολαβητές μέσω δικτύου πρόσβασης κάποιου λειτουργού δικτύου και μέσω του Διαδικτύου. Μία τυπική περίπτωση χρήσης του ευρείας κλίμακας συστήματος είναι αυτή ενός χρήστη που μέσω του PDA αποκτά πρόσβαση σε ένα ασύρματο δημόσιο τοπικό δίκτυο (WLAN) και χρησιμοποιεί τις παρεχόμενες από το σύστημα υπηρεσίες που βασίζονται στο πλαίσιο χρήσης του.

Βασικά ζητήματα σε μία τέτοια διαμόρφωση του συστήματος είναι αυτά της κατανομής του φορτίου και της κλιμάκωσης, κυρίως όσον αφορά τον Διαμεσολαβητή Δικτύου. Για την αντιμετώπιση των ζητημάτων αυτών σε επίπεδο αρχιτεκτονικού σχεδιασμού τρεις είναι οι δυνατές επιλογές, μεμονωμένες ή σε συνδυασμό:

A) κατανομή των υποσυστημάτων ανάλογα με τις λειτουργίες που παρέχουν και το φορτίο που καλούνται να εξυπηρετήσουν σε διαφορετικούς διαμεσολαβητές. Έτσι, αυτομάτως δημιουργούνται διαμεσολαβητές που παρέχουν διαφορετικές λειτουργίες και οι προς εξυπηρέτηση αιτήσεις κατανέμονται ανάλογα με το είδος τους.

B) διατήρηση όλων των λειτουργιών σε ένα διαμεσολαβητή αναθέτοντας του όμως προς εξυπηρέτηση συγκεκριμένο κομμάτι του δικτύου ή συγκεκριμένες γεωγραφικές περιοχές

Γ) και τέλος για κάθε ενεργό διαμεσολαβητή η ύπαρξη ενός πανομοιότυπου σε αναμονή (stand-by) έτοιμου να αναλάβει σε περίπτωση ανάγκης. Έτσι, η διαθεσιμότητα του δικτύου διατηρείται σε υψηλά επίπεδα μέσω της χρήσης εφεδρικών διαμεσολαβητών.

Ο σχεδιασμός και η αρχιτεκτονική του προτεινόμενου συστήματος παρέχει τους απαραίτητους μηχανισμούς προκειμένου να είναι δυνατή η εφαρμογή οποιασδήποτε από τις τρεις παραπάνω λύσεις.

Η δρομολόγηση των γεγονότων βάσει του τύπου τους είναι ένας από αυτούς. Με τη χρήση αυτού του μηχανισμού επικοινωνίας επιτυγχάνεται η χαλαρή διασύνδεση (loose coupling) ανάμεσα στα υποσυστήματα και αυτό θεωρείται ο βασικότερος παράγοντας ευελιξίας του συστήματος. Αυτό έχει ως αποτέλεσμα την ύπαρξη διαφάνειας ως προς τη θέση των λειτουργικών οντοτήτων, παρέχοντας έτσι την δυνατότητα ύπαρξης πανομοιότυπων υποσυστημάτων (duplication) σε διάφορους διαμεσολαβητές και σε διάφορα σημεία του δικτύου με αποτέλεσμα την κατανομή του φορτίου.

Επιπροσθέτως, παρόλο που στην Προσωπική Συσκευή Χρήστη υπάρχει καταχωρημένος ο Οικείος Διαμεσολαβητής (Home Broker) του κάθε χρήστη και στον οποίο συνδέεται κατά την είσοδό του στο σύστημα, η αλληλεπίδραση δεν περιορίζεται απαραίτητα μόνο με αυτόν.

Οι επικείμενες αλληλεπιδράσεις του χρήστη μπορούν να προωθηθούν είτε σε κάποιον άλλο πανομοιότυπο διαμεσολαβητή για λόγους κατανομής του φορτίου, είτε στον διαμεσολαβητή που φιλοξενεί τα απαραίτητα για την συγκεκριμένη σύνοδο υποσυστήματα, είτε στον διαμεσολαβητή που εξυπηρετεί το τρέχον domain ή την τρέχουσα γεωγραφική περιοχή που βρίσκεται ο χρήστης. Ωστόσο, χάριν στην ευελιξία και στην διαφάνεια της επικοινωνία που παρέχει το σύστημα, ο Διαμεσολαβητής Τερματικού δεν χρειάζεται να γνωρίζει σε κανένα σημείο του κύκλου ζωής της υπηρεσίας ούτε τον Διαμεσολαβητή Δικτύου που τον εξυπηρετεί, ούτε και τον συνολικό αριθμό των διαμεσολαβητών ή την τοπολογία τους.

Επίσης, η διατήρηση υπηρεσιών εγγραφών (registry services) στους Διαμεσολαβητές δίνει την δυνατότητα για ιεραρχικό σχεδιασμό των domains που εξυπηρετούν οι Διαμεσολαβητές. Έτσι, στις περιπτώσεις αναζήτησης του κατάλληλου υποσυστήματος για την εξυπηρέτηση της αίτησης του χρήστη, πρώτα εξαντλούνται η εντός πεδίου (domain) διαμεσολαβητές και στη συνέχεια η αναζήτηση περνάει σε άλλα πεδία. Κατά συνέπεια αποφεύγονται φαινόμενα πλημμύρας (flooding) που επιβαρύνουν το σύστημα και την απόδοσή του.

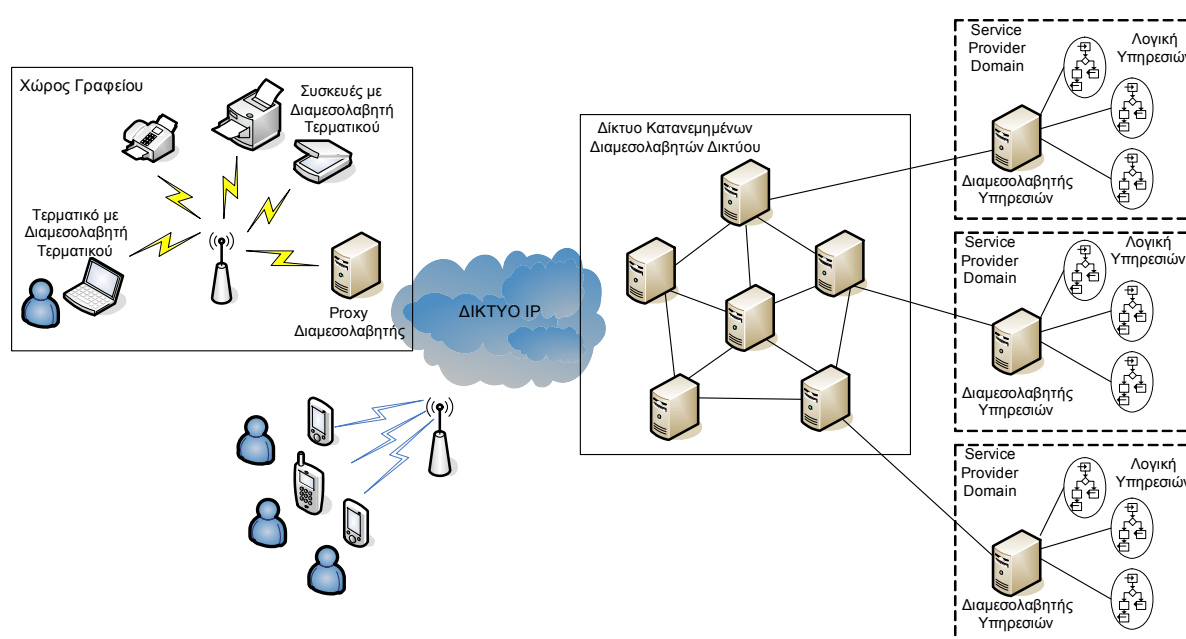
Η ίδια αντιμετώπιση, οι ίδιες λύσεις μπορούν και πρέπει να εφαρμοσθούν και στην περίπτωση των Διαμεσολαβητών Υπηρεσιών για λόγους κατανομής φορτίου αλλά και εφεδρείας. Για παράδειγμα για μία υπηρεσία Ροής Πολυμέσων (media streaming) θα έπρεπε να εξεταστεί το ενδεχόμενο ύπαρξης πανομοιότυπων εξυπηρετητών (mirror servers) για την κάλυψη μίας ευρείας γεωγραφικής περιοχής. Η επιλογή του κατάλληλου διαμεσολαβητή θα μπορούσε να πραγματοποιηθεί με διάφορους τρόπους. Είτε με χρήση καταχωρημένων δεικτών προς αυτούς στα δεδομένα κάποιου προφίλ, είτε μέσω μηχανισμών αναζήτησης και ανακάλυψης, κλπ.

Αλληλεπίδραση ανάμεσα σε συστήματα διαφορετικής ανάπτυξης:

Ένα πιθανό σενάριο χρήσης του συστήματος θα ήταν αυτό ενός χρήστη που βρίσκεται εντός του σπιτιού ή του γραφείου του, συνδεδεμένος στο τοπικό δίκτυο απολαμβάνοντας τις τοπικές υπηρεσίες που παρέχονται μέσω της αλληλεπίδρασης του Διαμεσολαβητή του τερματικού του με τους διαμεσολαβητές των άλλων συσκευών. Ωστόσο, ο χρήστης επιθυμεί να κάνει χρήση και κάποιας υπηρεσίας που παρέχεται από έναν Ανεξάρτητο Πάροχο Υπηρεσιών. Στην προτεινόμενη αρχιτεκτονική η δρομολόγηση των γεγονότων ευσταθεί είτε για την περίπτωση της τοπικής ανάπτυξης, είτε για αυτή της μεγάλης κλίμακας αλλά όχι και για τις δύο συγχρόνως. Συνεπώς, κρίνεται σκόπιμη η ύπαρξη ενός διαμεσολαβητή εντός του τοπικού περιβάλλοντος, ο οποίος θα δρα σαν βοηθητικός εξυπηρετητής (proxy) για την πρόσβαση στις εξωτερικές υπηρεσίες. Ο εξυπηρετητής θα είναι υπεύθυνος για την

δρομολόγηση των γεγονότων στον κατάλληλο Διαμεσολαβητή Δικτύου, αλλά και για τη λήψη των εισερχόμενων από αυτόν γεγονότων και την προώθηση τους στον αιτούντα Διαμεσολαβητή Τερματικού. Για τις διαδικασίες που εκκινούνται από την πλευρά του δικτύου ο proxy πραγματοποιώντας τις τυπικές διαδικασίες δρομολόγησης των γεγονότων, θα τα δρομολογεί προς τον κατάλληλο Διαμεσολαβητή Τερματικού του τοπικού συστήματος. Η συνδυασμένη αυτή προσέγγιση δεν παρουσιάζει περαιτέρω ζητήματα κλιμάκωσης σε σχέση με την περίπτωση της προηγούμενης παραγράφου.

Στο παρακάτω σχήμα (Σχήμα 64) παρουσιάζεται μία γραφική αναπαράσταση του συστήματος όπως αυτό θα διαμορφωνόταν βάσει των προαναφερθέντων αναπτύξεων.



Σχήμα 64: Συνδυασμένη Ανάπτυξη του Συστήματος

5.4 Επιχειρηματικό Μοντέλο

Κάθε νέο προτεινόμενο σύστημα οφείλει να συνοδεύεται και από ένα επιχειρηματικό μοντέλο προκειμένου να προσδιορίζεται κατά πόσο μπορεί να διεισδύσει στην αγορά, να αφομοιωθεί και να επιφέρει κέρδη. Αν και μία τέτοιου είδους μελέτη δεν έχει καθαρά ερευνητικό ενδιαφέρον, αλλά μάλλον εμπορικό, κρίνεται σκόπιμο να παρουσιαστεί μία ποιοτική, και όχι ποσοτική, προσέγγιση στο πλαίσιο της παρούσας διατριβής προκειμένου να ολοκληρωθεί η εικόνα του προτεινόμενου συστήματος.

Η συγκεκριμένη μελέτη επικεντρώθηκε στο να προσδιορίσει ουσιαστικά τα προϊόντα που θα προέκυπταν από την εισαγωγή ενός τέτοιου συστήματος στο χώρο της αγοράς, τους κατασκευαστές/παραγωγούς των προϊόντων, τους πωλητές και μεταπωλητές τους, τους αγοραστές και τους τρόπους προώθησης αυτών. Για την εξαγωγή των συμπερασμάτων

δημιουργήθηκε και αναλύθηκε ένας αριθμός πιθανών σεναρίων [7]. Από την ανάλυση των σεναρίων προέκυψε ότι το σύστημα θα δημιουργούσε προϊόντα τόσο για τους τελικούς χρήστες, όσο και για τους Ανεξάρτητους Παρόχους Υπηρεσιών και τους Λειτουργούς Δικτύων.

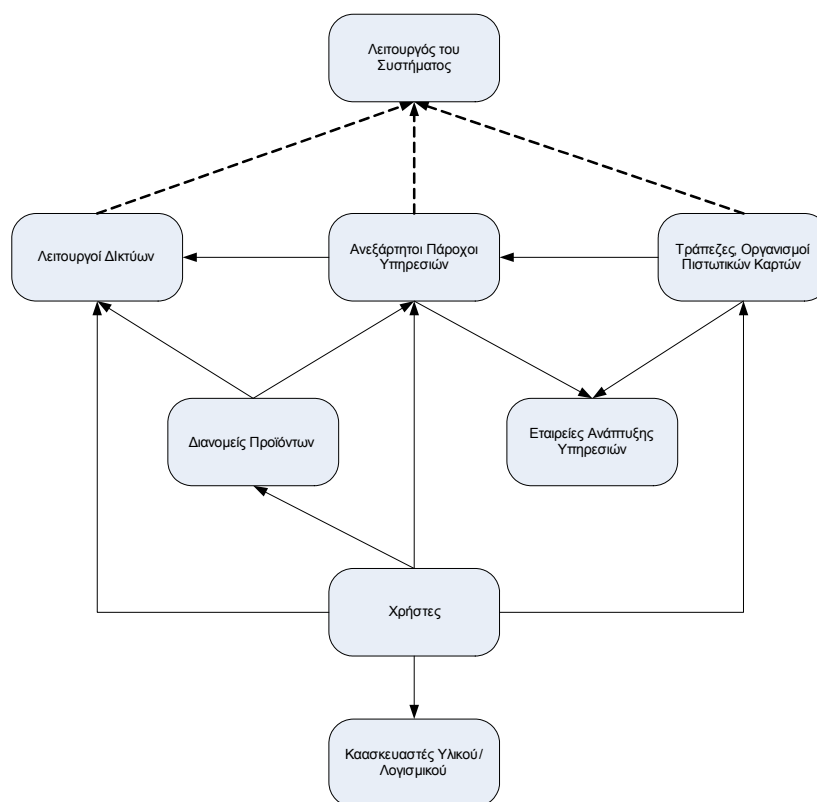
Για την περίπτωση των τελικών χρηστών το προϊόν που θα καλούνταν να αγοράσουν θα μπορούσε να είναι ένα πακέτο αποτελούμενο από μία Προσωπική Συσκευή Χρήστη και ένα σύνολο προπληρωμένων ή όχι υπηρεσιών. Με χρήση της Προσωπικής Συσκευής ο χρήστης θα είχε τη δυνατότητα να εκμεταλλεύεται την υποδομή του λειτουργού του συστήματος προκειμένου να χρησιμοποιεί τις διαθέσιμες υπηρεσίες βάσει του πακέτου που έχει αγοράσει. Ουσιαστικά, το μοντέλο που ακολουθείται ταυτίζεται κατά μία έννοια με αυτό της κινητής τηλεφωνίας, στο οποίο ο χρήστης ουσιαστικά αγοράζει την κάρτα SIM και χρησιμοποιεί τις υπηρεσίες κινητής τηλεφωνίας που του παρέχει ο πάροχος ανάλογα με το συμβόλαιο του.

Επίσης ο χρήστης θα έχει την δυνατότητα να αγοράζει και περαιτέρω υπηρεσίες από αυτές του βασικού πακέτου, είτε παρεχόμενες από τον λειτουργό του συστήματος, είτε από άλλους παρόχους υπηρεσιών. Στην εμπορική αυτή αλυσίδα θα μπορούν να εμπλέκονται και οργανισμοί όπως τράπεζες ή εταιρίες πιστωτικών καρτών για τις διαδικασίες πληρωμής των παρεχόμενων υπηρεσιών.

Όσον αφορά στους Ανεξάρτητους Παρόχους υπηρεσιών, προκειμένου να παρέχουν τις υπηρεσίες τους μέσω του συστήματος, θα πρέπει να έρθουν σε συμφωνία με το λειτουργό του. Επίσης, θα πρέπει να αγοράζουν και το αντίστοιχο λογισμικό, ή ακόμα και να πληρώνουν δικαιώματα χρήσης. Το απαραίτητο λογισμικό επίσης θα πρέπει να αγοράσουν και οι Λειτουργοί των δικτύων, αλλά και τράπεζες και άλλοι οργανισμοί που θα εμπλέκονται στην αλυσίδα παροχής των υπηρεσιών μέσω του συστήματος.

Οι Λειτουργοί των Δικτύων ωστόσο θα μπορούσαν να παίξουν πολύ σημαντικό ρόλο υιοθετώντας οι ίδιοι το ρόλο του λειτουργού του συστήματος και ενσωματώνοντας το στην υποδομή τους. Αυτό το σενάριο, ίσως αποτελεί και την πιο αποτελεσματική λύση αν ληφθεί υπ' όψιν το ισχυρό οικονομικό και εμπορικό προφίλ των λειτουργών δικτύων στο χώρο της αγοράς, τον ευρέως αποδεκτό και καθιερωμένο τρόπο προώθησης των προϊόντων τους (διαφήμιση, εξειδικευμένο δίκτυο καταστημάτων κλπ.) και την ήδη ανεπτυγμένη σχέση εμπιστοσύνης με το σύνολο των πελατών.

Εκτός από τους παραπάνω στην οικονομική αλυσίδα που θα δημιουργούσε η είσοδος του συστήματος στην αγορά μπορούν να προστεθούν και οι εταιρίες ανάπτυξης του κατάλληλου λογισμικού και υλικού.



Σχήμα 65: Βασικές Οικονομικές Ροές του Επιχειρηματικού μοντέλου

Τέλος, θα πρέπει να αναφερθεί και το θέμα της χρέωσης των υπηρεσιών το οποίο αποδεικνύεται εξαιρετικά πολύπλοκο στις περιπτώσεις εμπλοκής πολλών ενδιάμεσων οντοτήτων. Τρεις προτεινόμενες εναλλακτικές είναι οι παρακάτω:

- Η χρέωση να πραγματοποιείται μέσω του λειτουργού δικτύου, διαδικασία που μειώνει την πολυπλοκότητα των οικονομικών συναλλαγών και οι χρήστες την εμπιστεύονται.
- Πώληση προπληρωμένων καρτών που περιλαμβάνουν πρόσβαση σε συγκεκριμένες υπηρεσίες. Μηχανισμός επίσης οικείος προς τους χρήστες αλλά όχι βολικός για την περίπτωση συναλλαγών μεγάλης αξίας.
- Χρέωση των υπηρεσιών από τους παρόχους στις πιστωτικές κάρτες των χρηστών, επιτρέποντας στους τελευταίους να έχουν τον έλεγχο των συναλλαγών τους. Ωστόσο, η προσέγγιση αυτή δεν είναι κατάλληλη για συναλλαγές μικρής αξίας (micro-payment). Επίσης, τίθεται θέμα σχετικά με την διατήρηση της ανωνυμίας των χρηστών και της εμπιστοσύνης τους ως προς την αξιοπιστία των παρόχων για την

χρέωση των πιστωτικών τους καρτών. Το τελευταίο μπορεί να αντιμετωπιστεί τοποθετώντας ανάμεσα στο χρήστη και στον πάροχο την τράπεζα ή κάποια εταιρεία πιστωτικών καρτών. Η τελευταία χρεώνει τον χρήστη και τελικά αποδίδει τα κέρδη στον πάροχο. Έτσι δεν υφίσταται ανάγκη για άμεση αλληλεπίδραση χρήστη – παρόχου σε επίπεδο οικονομικής συναλλαγής.

Η γραφική αναπαράσταση των παραπάνω παρουσιάζεται στο Σχήμα 65.

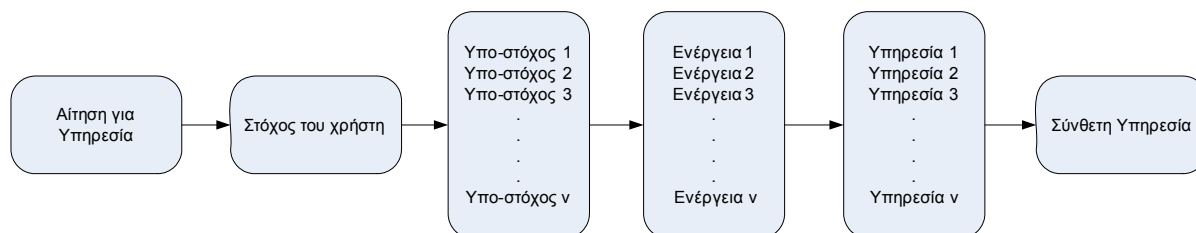
5.5 Μελλοντικές Εξελίξεις-Προτεινόμενες Επεκτάσεις

Η προτεινόμενη αρχιτεκτονική πέρα από την παροχή πληροφορίας πλαισίου προς τις εφαρμογές θα μπορούσε να επεκταθεί και να υποστηρίξει και άλλες λειτουργίες όπως για παράδειγμα την ανακάλυψη ή τη σύνθεση υπηρεσιών [8]. Επίσης, κρίνεται απαραίτητο να ληφθεί μέριμνα όσον αφορά στο θέμα της ιδιωτικότητας και της ασφάλειας των ευαίσθητων προσωπικών δεδομένων [9][10]. Στις υπο-ενότητες που ακολουθούν παρατίθενται δύο πιθανές μελλοντικές εξελίξεις του συστήματος προκειμένου να αντιμετωπιστούν τα παραπάνω ζητήματα.

5.5.1 Σύνθεση Υπηρεσιών με εκμετάλλευση Προτύπων Χρήσης (Usage Patterns)

Για την υποστήριξη της σύνθεσης υπηρεσιών με επίγνωση πλαισίου χρήσης προτείνεται η ενσωμάτωση στο σύστημα της ιδέας των προτύπων χρήσης υπηρεσιών (Usage Pattern).

Ο σημαντικότερος παράγοντας για την επιτυχημένη σύνθεση υπηρεσιών είναι η ανίχνευση του στόχου του χρήστη και ο καθορισμός των επιμέρους ενεργειών που πρέπει να πραγματοποιηθούν, καθώς και η σειρά τους, προκειμένου να επιτευχθεί ο στόχος αυτός [8][11]. Έχοντας λοιπόν προσδιορίσει τις επιμέρους ενέργειες καθίσταται δυνατή η επιλογή υπηρεσιών των οποίων η διαδοχική εκτέλεση θα έχει σαν αποτέλεσμα την επίτευξη των επιμέρους, και κατά συνέπεια, και του συνολικού στόχου.



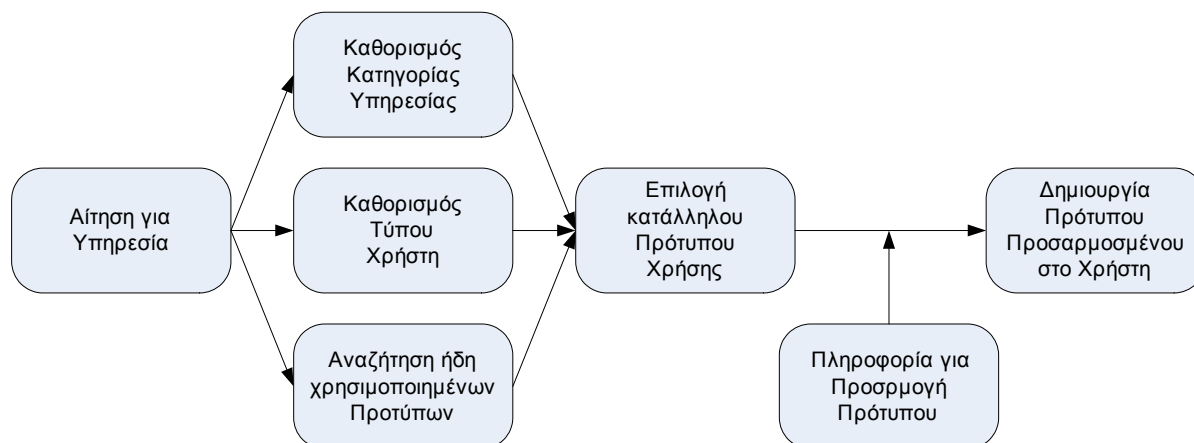
Σχήμα 66: Διαδοχικά βήματα για την επιλογή των προς σύνθεση υπηρεσιών βάσει του στόχου του χρήστη

Σαν παράδειγμα μπορεί να αναφερθεί η οργάνωση ενός επικείμενου ταξιδιού. Ένα ταξίδι απαιτεί την έκδοση εισιτηρίων, την κράτηση δωματίου και, ανάλογα με τον εκάστοτε χρήστη και τις προτιμήσεις του, τον καθορισμό κάποιων άλλων μετακινήσεων ή δραστηριοτήτων (επισκέψεις σε μουσεία, προτάσεις για διασκέδαση κλπ.). Τα διαδοχικά βήματα που απαιτούνται παρουσιάζονται στο Σχήμα 66.

Ωστόσο, η διαδικασία εντοπισμού του στόχου του χρήστη και τελικά η επιλογή των κατάλληλων υπηρεσιών μπορεί να είναι εξαιρετικά πολύπλοκη και να απαιτεί έξυπνους μηχανισμούς εξαγωγής συμπερασμάτων. Αυτό κυρίως οφείλεται στο γεγονός ότι η συμπεριφορά του χρήστη εν γένει είναι απρόβλεπτη. Ο κάθε χρήστης από ένα σύνολο υπηρεσιών μπορεί να επιλέξει διαφορετικούς συνδυασμούς και με διαφορετική σειρά προκειμένου να επιτύχει το στόχο του και να λάβει το επιθυμητό αποτέλεσμα. Για παράδειγμα ένας χρήστης μπορεί να κλείσει πρώτα ξενοδοχείο και στη συνέχεια εισιτήρια. Κάποιος χρήστης θα επιθυμεί τον εντοπισμό του ξενοδοχείου μέσω κάποιας υπηρεσίας χάρτη ενώ κάποιος άλλος όχι κλπ. Για τον λόγο αυτό εισάγεται η έννοια των Προτύπων Χρήσης Υπηρεσιών. Ως Πρότυπο Χρήσης (Usage Pattern) ορίζεται η περιγραφή των διαφορετικών ενεργειών που συνδυασμένες μπορούν να ικανοποιήσουν τον συνολικό στόχο του χρήστη. Ουσιαστικά δηλαδή αποτελούν περιγραφές των πιθανών αναλύσεων του στόχου του χρήστη σε υπο-στόχους. Πηγαίνοντας ένα βήμα παραπέρα και εξειδικεύοντας την έννοια του Πρότυπου Χρήσης, ορίζεται το Προσαρμοσμένο στο Χρήστη Πρότυπο Χρήσης, το οποίο αποτελεί Πρότυπο Χρήσης τροποποιημένο βάσει της τρέχουσας κατάστασης και των προτιμήσεων του χρήστη. Ο περαιτέρω ορισμός του Προσαρμοσμένου στο Χρήστη Πρότυπου Χρήσης οφείλεται στο γεγονός ότι η ίδια αίτηση υπηρεσίας μπορεί να μεταφραστεί σε ένα διαφορετικό σύνολο από υποστόχους, και κατά συνέπεια και υπηρεσίες, ανάλογα με το προφίλ και την κατάσταση του χρήστη από τον οποίο προέρχεται η αίτηση.

5.5.1.1 Επιλογή Πρότυπου Χρήσης και Δημιουργία Προσαρμοσμένου στο Χρήστη Πρότυπου Χρήσης

Για την παροχή μίας σύνθετης υπηρεσίας απαιτείται η αντιστοίχιση της αίτησης του χρήστη με ένα Πρότυπο Χρήσης και στη συνέχεια βάσει του πρότυπου η ανεύρεση και σύνθεση των υπηρεσιών που θα έχουν σαν αποτέλεσμα το επιθυμητό από το χρήστη αποτέλεσμα. Τα βήματα που περιλαμβάνει ο προτεινόμενος μηχανισμός παρουσιάζονται στο Σχήμα 67.



Σχήμα 67: Διαδοχικά βήματα για την δημιουργία του Προσαρμοσμένου στο Χρήστη Πρότυπου Χρήσης

Αρχικά καθορίζεται σε ποια κατηγορία υπηρεσιών ανήκει η αίτηση του χρήστη. Μία κατηγορία υπηρεσίας περιλαμβάνει όλες εκείνες τις υπηρεσίες που αφορούν ενέργειες παρόμοιου είδους. Εν συνεχεία, προσδιορίζεται η ομάδα χρηστών στην οποία ανήκει ο χρήστης. Με τον όρο ομάδα χρηστών ορίζονται οι χρήστες που συνήθως χρησιμοποιούν τα ίδια μέσα για να επιτύχουν παρόμοιους στόχους. Η κατάταξη των υπηρεσιών και των χρηστών σε κατηγορίες μπορεί να πραγματοποιηθεί με πολλούς τρόπους, όπως για παράδειγμα με χρήση ταξινομήσεων βάσει σχετικών επιχειρηματικών μοντέλων. Τέλος, πραγματοποιείται ένας έλεγχος για την περίπτωση που ο χρήστης έχει στο παρελθόν πραγματοποιήσει παρόμοια αίτηση η οποία εξυπηρετήθηκε από κάποιο υπάρχον Πρότυπο Χρήσης. Με βάση τα παραπάνω εξαγόμενα συμπεράσματα είναι δυνατή η επιλογή προς εφαρμογή του κατάλληλου ή ενός αριθμού κατάλληλων Προτύπων Χρήσης. Για την περίπτωση της οργάνωσης κάποιου ταξιδιού η περιγραφή του Πρότυπου Χρήσης θα μπορούσε να είναι “κράτηση ξενοδοχείου – έκδοση εισιτηρίων – πρόγραμμα επισκέψεων σε αξιοθέατα”. Αμέσως μετά ο μηχανισμός τροφοδοτείται με πληροφορία πλαισίου χρήσης (προφίλ χρήστη, προτιμήσεις, περιορισμούς κλπ.) και έτσι επιλέγεται το κατάλληλο πλέον πρότυπο το οποίο τροποποιείται κατάλληλα και δημιουργείται το Προσαρμοσμένο στο Χρήστη Πρότυπο Χρήσης. Το προσαρμοσμένο πρότυπο για την περίπτωση της οργάνωσης του ταξιδιού θα μπορούσε να είναι “κράτηση ξενοδοχείου στην Αθήνα – έκδοση αεροπορικών εισιτηρίων με απευθείας πτήση σε Business Class – Κράτηση εισιτηρίων για παράσταση στο Μέγαρο Μουσικής”. Με είσοδο το τελευταίο σε κατάλληλους μηχανισμούς ανακάλυψης υπηρεσιών αναζητούνται και συντίθενται οι κατάλληλες υπηρεσίες.

Βασική προϋπόθεση για την εφαρμογή των παραπάνω αποτελεί η ύπαρξη των Προτύπων Χρήσης. Ο βασικότερος τρόπος δημιουργίας των Προτύπων Χρήσης είναι μέσω κατάλληλου μηχανισμού παρακολούθησης και εξαγωγής στατιστικών συμπερασμάτων. Για

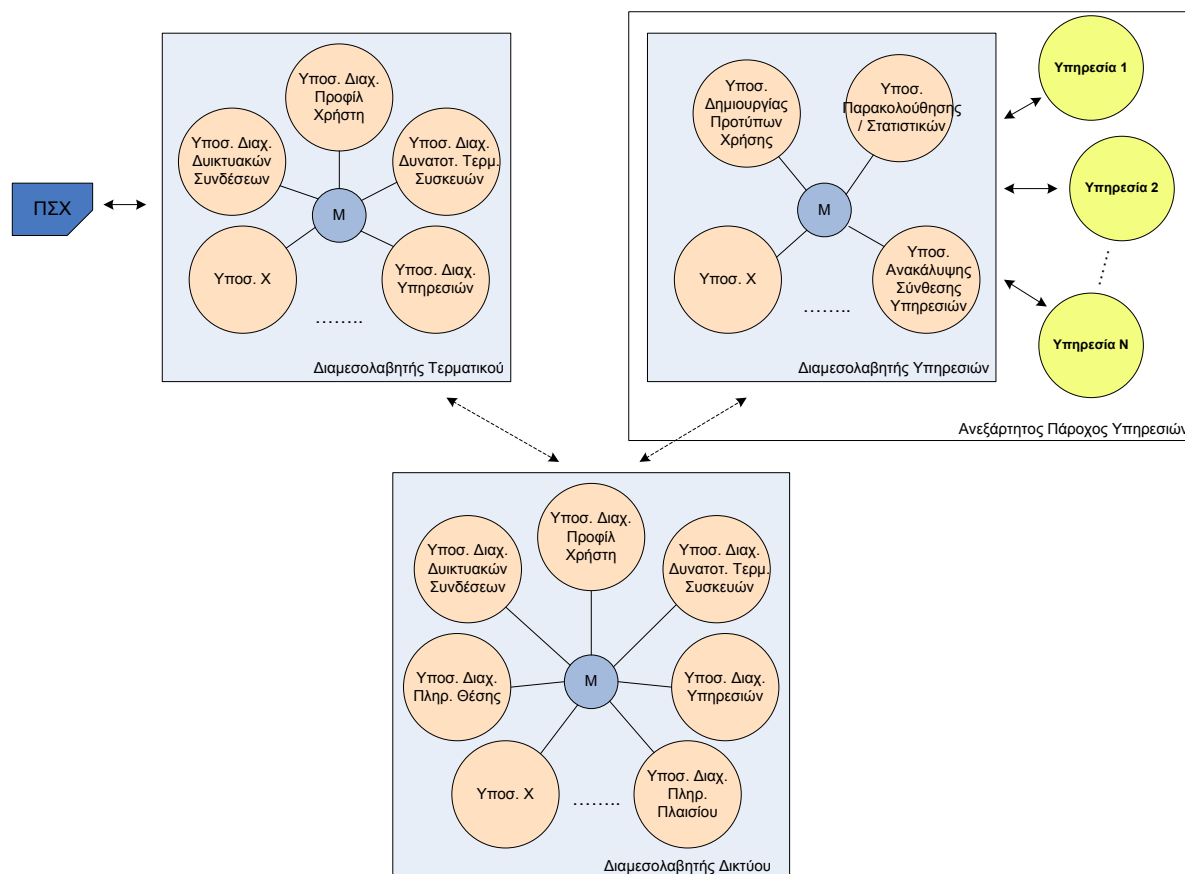
παράδειγμα, ένα συχνά εμφανιζόμενο σενάριο εκτέλεσης διαδοχικών υπηρεσιών μπορεί να αποτελέσει Πρότυπο Χρήσης. Μερικά είδη πληροφορίας που θα μπορούσαν να χρησιμοποιηθούν είναι η συχνότητα χρήσης των υπηρεσιών, τα είδη των υπηρεσιών που συντίθεται, οι παράμετροι που χρησιμοποιούνται για τη σύνθεση των υπηρεσιών, η προέλευση των αιτήσεων για υπηρεσία, η αλληλουχία ανάμεσα στις αλληπάλληλες κλήσεις των υπηρεσιών κλπ. Από τις συλλεγμένες πληροφορίες με χρήση κατάλληλων αλγορίθμων και κατάλληλης επεξεργασίας θα μπορούσε να καταστεί δυνατή η δημιουργία των προτύπων χρήσης, τα οποία να διατηρούνται σε κατάλληλες βάσεις. Η δημιουργία και αποθήκευση των προτύπων χρήσης είναι μία διαδικασία συνεχούς μάθησης από μέρους του συστήματος και παραμένει πάντα δυναμική. Ανάλογα με τα στατιστικά δεδομένα, νέα πρότυπα μπορούν να εισάγονται στο σύστημα ή μη δημοφιλή να διαγράφονται. Επίσης, τα πρότυπα μπορούν να γίνονται πιο γενικά ή πιο ειδικά.

5.5.1.2 Ενσωμάτωση Προτύπων Χρήσης στην προτεινόμενη αρχιτεκτονική

Προκειμένου η προτεινόμενη αρχιτεκτονική να παρέχει μηχανισμούς ανακάλυψης και σύνθεσης υπηρεσιών με χρήση Προτύπων Χρήσης κρίνεται απαραίτητη η ύπαρξη των παρακάτω λειτουργιών:

- Παρακολούθηση Χρήσης Υπηρεσιών και πιο συγκεκριμένα καταγραφή των αλληλεπιδράσεων ανάμεσα σε χρήστες και υπηρεσίες, της αλληλουχίας τους και της συχνότητας τους.
- Στατιστική Επεξεργασία των δεδομένων που ανακτώνται από την παρακολούθηση της χρήσης των υπηρεσιών και
- Μηχανισμοί Δημιουργίας Προτύπων Χρήσης οι οποίοι βασιζόμενοι στα στατιστικά συμπεράσματα θα δημιουργούν συσχετισμούς ανάμεσα στις υπηρεσίες.
- Μηχανισμούς Αναζήτησης και Σύνθεσης Υπηρεσιών

Στο πλαίσιο της προτεινόμενης αρχιτεκτονικής οι λειτουργίες αυτές θα μπορούσαν να ενσωματωθούν στο σύστημα με τη μορφή υποσυστημάτων στον Διαμεσολαβητή Υπηρεσιών που διατηρεί ο εκάστοτε Ανεξάρτητος Πάροχος Υπηρεσιών. Στο σχήμα που ακολουθεί παρουσιάζεται μία πρόμη προσπάθεια προσαρμογής της προτεινόμενης αρχιτεκτονικής προκειμένου να καλυφθούν οι απαιτήσεις για σύνθεση υπηρεσιών με επίγνωση του πλαισίου χρήσης τους με εκμετάλλευση Προτύπων Χρήσης.



Σχήμα 68: Προσαρμοσμένη Αρχιτεκτονική για την υποστήριξη σύνθεσης υπηρεσιών με εκμετάλλευση Προτύπων Χρήσης

Προκειμένου λοιπόν να καλυφθούν οι λειτουργικές ανάγκες για σύνθεση υπηρεσιών μέσω Προτύπων χρήσης εισάγονται στον Διαμεσολαβητή Υπηρεσιών τρία νέα υποσυστήματα. Αυτά είναι το Υποσύστημα Παρακολούθησης και Στατιστικών, το Υποσύστημα Δημιουργίας και Διατήρησης Προτύπων Χρήσης και το Υποσύστημα Ανακάλυψης Σύνθεσης Υπηρεσιών που ουσιαστικά φιλοξενούν τις αντίστοιχες λειτουργίες.

5.5.2 Ασφάλεια και Ιδιωτικότητα – Η Προσέγγιση Discreet

Η ευρεία ανάπτυξη και άνθιση των υπηρεσιών με επίγνωση του πλαισίου χρήσης τους, εκτός από πλεονεκτήματα, παρουσιάζει σαφώς και ένα πολύ σημαντικό μειονέκτημα. Αυτό είναι η επίδραση τους στο δικαίωμα ιδιωτικότητας των χρηστών. Αυτό οφείλεται στο γεγονός ότι για την παροχή τέτοιου είδους υπηρεσιών, απαιτείται η χρήση από την υπηρεσία προσωπικών δεδομένων των χρηστών. Παρόλο, που η πρώτη αναφορά στην βιβλιογραφία για παραβίαση της ιδιωτικότητας εμφανίζεται περισσότερο από έναν αιώνα πριν [12], μόλις τα τελευταία χρόνια οι χρήστες άρχισαν να ανησυχούν ιδιαίτερα για δικαιώματα και τη

διατήρησή τους όσον αφορά στην ιδιωτικότητά τους, σε σχέση με τις ταχύτατα αναπτυσσόμενες τεχνολογίες στο χώρο των επικοινωνιών και των υπηρεσιών [13].

Μέχρι στιγμής ωστόσο, η έννοια της ιδιωτικότητας, φαίνεται να απασχολεί μόνο τους νομοθέτες και τους κοινωνιολόγους. Η προστασία της ιδιωτικότητας καλύπτεται μέσω νόμων και με την παραδοχή ότι οι πάροχοι των υπηρεσιών τους εφαρμόζουν, χωρίς όμως να υπάρχουν σχετικές τεχνικές λύσεις, που η εφαρμογή τους να παρέχει τις κατάλληλες εγγυήσεις. Οι μηχανισμοί ασφάλειας που εφαρμόζονται σήμερα, όπως είναι οι μηχανισμοί κρυπτογραφίας (encryption) ή οι μηχανισμοί ελέγχου πρόσβασης (access control), να μην παρέχουν προστασία στα προσωπικά δεδομένα των χρηστών, για παράδειγμα κατά την μεταφορά τους από το τερματικό του χρήστη στον εξυπηρετητή του παρόχου της υπηρεσίας, αλλά δεν αποτρέπουν τον πάροχο από το να τα χρησιμοποιήσει παραδείγματος χάριν για την εξαγωγή στατιστικών συμπερασμάτων ή ακόμα και να τα προωθήσει σε κάποιον τρίτο. Είναι λοιπόν εμφανής η ανάγκη για την εφαρμογή κατάλληλων τεχνικών λύσεων οι οποίες θα εξαλείψουν ή τουλάχιστον θα ελαχιστοποιήσουν την άσκοπη και κακόβουλη εκμετάλλευση των προσωπικών δεδομένων των χρηστών εις βάρος των τελευταίων και του δικαιώματός τους στην ιδιωτικότητα.

Μία τέτοιου είδους προσέγγιση παρουσιάζεται στα [9] και [10]. Η αρχιτεκτονική μεσισμικού που παρουσιάζεται είναι βασισμένη στην σχετική με την ιδιωτικότητα νομοθεσία και σαν στόχο έχει να την εφαρμόσει με τεχνολογικά μέσα. Η βασική σχεδιαστική αρχή που ακολουθείται είναι αυτή της δημιουργίας μίας οντότητας-εμπιστοσύνης (unit of trust), που ονομάζεται Discreet Box, η οποία θα παίζει το ρόλο μεσολαβητή ανάμεσα στους χρήστες, στους παρόχους υπηρεσιών και στο νόμο. Η οντότητα αυτή θα διατηρεί ξεχωριστά και θα προστατεύει από κακόβουλες επεμβάσεις τα προσωπικά δεδομένα που συλλέγονται για την παροχή υπηρεσιών και θα παρέχει προς αυτές μόνο τα απολύτως απαραίτητα και κάτω από συγκεκριμένες συνθήκες και προϋποθέσεις. Οι αποφάσεις σχετικά με την αποκάλυψη ή όχι των δεδομένων προς τις υπηρεσίες θα βασίζονται σε κατάλληλο πλαίσιο εφαρμογής πολιτικών/κανόνων (policy based decisions) που θα υλοποιεί τόσο την ανάλογη νομοθεσία (σχετικά με όσο και κανόνες υλοποιημένους από την πλευρά των χρηστών).

Η ενσωμάτωση της οντότητας του Discreet Box στην προτεινόμενη από την διατριβή αρχιτεκτονική, θα μπορούσε να αποτελέσει ένα πολύ ενδιαφέρον ερευνητικά και τεχνολογικά εγχείρημα, το οποίο θα είχε ως αποτέλεσμα τόσο την ολοκλήρωση του προτεινόμενου συστήματος, όσο και την επαλήθευση της ορθής λειτουργίας του Discreet

Box εντός μίας ήδη υλοποιημένης αρχιτεκτονικής παροχής υπηρεσιών με επίγνωση του πλαισίου χρήσης τους.

5.6 Αναφορές

- [1] Distributed Management Task Force. Common Information Model. <http://www.dmtf.org/standards/cim/>
- [2] Object Management Group, OMG. <http://www.omg.org/technology/documents/formal/uml.htm>
- [3] 3rd Generation Partnership Project. 3GPP Generic User Profile (GUP) - Architecture. Technical specification Group Services and Systems Aspects, TS 23.240, Version 6.7.0. 2005.
- [4] WAPFORUM. User Agent Profile (UAProf). <http://www.wapforum.org>
- [5] Composite Capabilities/Preference Profile Public Home Page. <http://www.w3.org/Mobile/CCPP>
- [6] D2202: Final System Architecture Specification, IST-SIMPLICITY Project, <http://www.ist-simplicity.org>
- [7] D2102: Enhanced Use cases, requirements and business models, IST-SIMPLICITY Project, <http://www.ist-simplicity.org>
- [8] W.-T. Balke, M. Wagner. Towards Personalized Selection of Web Services. In Proc. of the 12th Int. World Wide Web Conf. (WWW2003) Alternate Track on Web Services, Budapest, Hungary, 2003.
- [9] E. Koutsoloukas, G. V. Lioudakis, S. Kapellaki, N. L. Dellas, C. Katsigiannis, G. N. Prezerakos, D. I. Kaklamani, I. S. Venieris, “A Middleware Architecture for Privacy Protection in Smart Environments”, In Proc. of 15th IST Mobile & Wireless Communications Summit, Myconos, Greece, June 2006
- [10] Georgios V. Lioudakis, Eleftherios Koutsoloukas, Nikolaos L. Dellas, Sofia Kapellaki, Nikos D. Tselikas, George N. Prezerakos, Dimitra I. Kaklamani, Iakovos S. Venieris, “A Middleware Architecture for Privacy Protection”, Accepted for publication in Computer Networks Journal, Elsevier
- [11] Caroline Funk, Elisabeth-Anna Fischer, Carsten Meyer, Christoph Niedermeier, Robert Seidl, George Prezerakos, Nikos Tselikas, Sofia Kapellaki, Lefteris Koutsoloukas, Nikos Dellas and I.S. Venieris, Towards Ambient Aware Service Provisioning, In preparation for submission.
- [12] S. D. Warren and L. D. Brandeis, “The Right to Privacy”, Harvard Law Review, Vol. IV, No. 5, pp. 193–220, December 1890.
- [13] The European Opinion Research Group, “European Union citizens’ views about privacy”, Special Eurobarometer 196, December 2003

