



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ  
ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Ποιότητα υπηρεσίας και προστασία ιδιωτικότητας  
σε εξατομικευμένες προσαρμοστικές κινητές υπηρεσίες**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αθανάσιος Σ. Βουλόδημος

Αθήνα, Αύγουστος 2007



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ  
ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Ποιότητα υπηρεσίας και προστασία ιδιωτικότητας  
σε εξατομικευμένες προσαρμοστικές κινητές υπηρεσίες**

**Quality of service and privacy protection  
in personalized context-aware mobile services**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αθανάσιος Σ. Βουλόδημος

**Επιβλέπων :** Ε. Ν. Πρωτονοτάριος  
Καθηγητής Ε.Μ.Π.

Αθήνα, Αύγουστος 2007



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ  
ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Ποιότητα υπηρεσίας και προστασία ιδιωτικότητας  
σε εξατομικευμένες προσαρμοστικές κινητές υπηρεσίες**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αθανάσιος Σ. Βουλόδημος

**Επιβλέπων :** Ε. Ν. Πρωτονοτάριος  
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 28<sup>η</sup> Αυγούστου 2007.

.....  
Ε. Ν. Πρωτονοτάριος  
Καθηγητής Ε.Μ.Π.

.....  
Μ. Ε. Θεολόγου  
Καθηγητής Ε.Μ.Π.

.....  
Θ. Βαρβαρίγου  
Αν. Καθηγήτρια Ε.Μ.Π.

Αθήνα, Αύγουστος 2007

.....  
Αθανάσιος Σ. Βουλόδημος

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

Copyright © Αθανάσιος Σ. Βουλόδημος, 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 Μετσόβιου Πολυτεχνείου.

## Περίληψη

Αντικείμενο της παρούσας διπλωματικής εργασίας είναι η μελέτη των εξατομικευμένων (personalized), προσαρμοστικών (context-aware) υπηρεσιών και, ειδικότερα, εκείνων που βασίζονται στη θέση του χρήστη (location-based), με έμφαση στην προστασία της ιδιωτικότητας (privacy) των προσωπικών του δεδομένων. Η παροχή εξατομικευμένων υπηρεσιών συνίσταται στην προσαρμογή της προσφερόμενης υπηρεσίας με βάση τα ατομικά χαρακτηριστικά, τις ανάγκες, καθώς και τις προσωπικές προτιμήσεις και ρυθμίσεις του κάθε χρήστη. Οι προσωπικές αυτές ρυθμίσεις, οι οποίες μπορούν να καλύπτουν ένα ευρύ φάσμα δραστηριοτήτων, συγκροτούν, μαζί με ένα σύνολο προσωπικών πληροφοριών, το προφίλ χρήστη (user profile), για το οποίο προτείνεται μια αναλυτική δομή περιγραφής. Η χρήση μιας εξατομικευμένης υπηρεσίας βασισμένης στη θέση προϋποθέτει την κοινοποίηση στον πάροχο πληροφοριών όπως η γεωγραφική θέση και οι προτιμήσεις του χρήστη, ώστε η υπηρεσία να παρασχεθεί με ακρίβεια (και, επομένως, εξασφαλίζοντας ποιότητα). Το γεγονός αυτό εγείρει το ζήτημα της ασφάλειας των προσωπικών δεδομένων και δημιουργεί την ανάγκη προστασίας της ιδιωτικότητας των προσωπικών πληροφοριών. Στα πλαίσια της διπλωματικής, εξετάζονται μέθοδοι που έχουν προταθεί για το σκοπό αυτό και, στη συνέχεια, προτείνεται ένα μοντέλο για την ποσοτικοποίηση της ιδιωτικότητας των προσωπικών παραμέτρων (γεωγραφικής θέσης και προσωπικών προτιμήσεων). Προτείνεται μια μεθοδολογία οργάνωσης των παραμέτρων αυτών σε επίπεδα διαφορετικής ακρίβειας / αφαίρεσης. Για τη γεωγραφική θέση μπορούν να χρησιμοποιηθούν γνωστά συστήματα οργάνωσης γεωγραφικής πληροφορίας, τα οποία είναι οργανωμένα ιεραρχικά σε διάφορα επίπεδα λεπτομέρειας. Για την αντίστοιχη οργάνωση σε διαφορετικά επίπεδα ακρίβειας των προσωπικών προτιμήσεων μελετώνται και αξιολογούνται αλγόριθμοι συσταδοποίησης (clustering). Σκοπός είναι η εύρεση του εκάστοτε κατάλληλου συνδυασμού λιγότερο ή περισσότερο συγκεχυμένων πληροφοριών, ο οποίος οδηγεί στην προστασία της ιδιωτικότητας, εξασφαλίζοντας ταυτόχρονα ποιότητα (ακρίβεια) της υπηρεσίας. Το προτεινόμενο μοντέλο πρόκειται να εφαρμοστεί πρακτικά στην πλατφόρμα παροχής εξατομικευμένων υπηρεσιών PLASMA, η ανάπτυξη της οποίας βρίσκεται σε εξέλιξη.

## Λέξεις – κλειδιά

Κινητές υπηρεσίες, εξατομίκευση (personalization), context awareness, υπηρεσίες βασισμένες στη θέση (location-based services), προφίλ χρήστη (user profile), ιδιωτικότητα (privacy), εντροπία, clustering, Virtual Earth, πλατφόρμα PLASMA

## **Abstract**

The subject of this diploma thesis is the study of personalized, context-aware (and mainly location-aware) services, with emphasis on the user's personal data privacy protection. The provision of personalized services is based on the customization of the supplied service, according to the user's particular characteristics, needs, personal preferences and settings. These personal settings, which may cover a wide range of activities, together with other personal information, form a user profile. An analytical decomposition of user profiles is presented. The use of a personalized, location-based service requires reporting information such as location and personal preferences on a specific point of interest (related to the service) to the service provider, so that the service can be supplied accurately (corresponding to quality). This fact raises issues regarding personal data security and brings about the need for protecting privacy of personal information. Following the evaluation of various methods towards this goal, a model for the quantification of privacy of personal parameters (location and preferences) is proposed. A methodology to organize these parameters in different levels of accuracy / abstraction is given. Regarding location, well known mapping systems that are hierarchically organized in different levels of detail can be used. As for personal preferences grouping in levels of different accuracy, three clustering algorithms are tested and evaluated. The ulterior goal lies in determining the correct mixture of blurring information about personal information, while trying not to compromise quality due to inaccurate results. The model is used in practice for the provision of personalized, context aware mobile services through the PLASMA platform.

## **Keywords**

mobile services, personalization, context awareness, location-based services (LBS), user profile, privacy, quality of service, entropy, clustering, Virtual Earth, PLASMA platform

Θα ήθελα να ευχαριστήσω τον Καθηγητή κ. Ε. Ν. Πρωτονοτάριο για την καθοδήγηση και τις συμβουλές του, και τους Καθηγητές κ. Μ. Ε. Θεολόγου και κα Θ. Α. Βαρβαρίγου για την υποστήριξή τους. Τέλος, θα ήθελα ιδιαίτερα να ευχαριστήσω τον Επιστημονικό Συνεργάτη του Τομέα Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής Δρα Χ. Πατρικάκη για την πολύτιμη βοήθεια, τις χρήσιμες υποδείξεις και τη συνεργασία του.

# ΠΕΡΙΕΧΟΜΕΝΑ

1.	ΕΙΣΑΓΩΓΙΚΕΣ ΈΝΝΟΙΕΣ.....	1
1.1.	Αειφανής υπολογισμός (Ubiquitous computing).....	1
1.2.	Περιβάλλον (Context).....	1
1.3.	Επίγνωση περιβάλλοντος (Context awareness).....	4
1.4.	Επίγνωση θέσης (Location awareness).....	6
1.5.	Υπηρεσίες βασισμένες στη θέση (Location-Based Services).....	6
1.5.1.	Αρχιτεκτονική LBS.....	8
1.5.2.	Μέθοδοι εντοπισμού και ακρίβεια.....	11
2.	ΟΡΙΣΜΟΣ ΠΡΟΦΙΛ ΧΡΗΣΤΗ.....	14
2.1.	Εισαγωγή.....	14
2.1.1.	Πρότυπα για τον ορισμό προφίλ.....	14
2.1.2.	Υπάρχουσα έρευνα στον ορισμό προφίλ.....	15
2.2.	Σχεδιασμός ορισμού προφίλ κατά τις προδιαγραφές του ETSI.....	16
2.3.	Αποσύνθεση προφίλ.....	18
2.3.1.	Βασικές πληροφορίες χρήστη.....	18
2.3.1.1.	Προσωπικές πληροφορίες χρήστη.....	18
2.3.1.2.	Επαγγελματικές πληροφορίες χρήστη.....	21
2.3.1.3.	Πληροφορίες σχέσης χρήστη – συσκευής.....	22
2.3.2.	Προχωρημένες πληροφορίες χρήστη.....	23
2.3.3.	Πληροφορίες σχετικές με συσκευή και υπηρεσίες.....	25
2.3.3.1.	Πληροφορίες σχετικές με τη συσκευή.....	25
2.3.3.2.	Πληροφορίες σχετικές με υπηρεσίες.....	26
2.4.	Διαχείριση προφίλ.....	27
2.4.1.	Η οντότητα διαχείρισης προφίλ.....	27
2.4.2.	Η οντότητα εφαρμογής προφίλ.....	27
2.4.3.	Εφαρμογή ενός προφίλ χρήστη.....	28
3.	ΙΔΙΩΤΙΚΟΤΗΤΑ ΠΡΟΣΩΠΙΚΩΝ ΔΕΔΟΜΕΝΩΝ ΣΕ ΕΞΑΤΟΜΙΚΕΥΜΕΝΕΣ ΥΠΗΡΕΣΙΕΣ... 30	
3.1.	Γενικά.....	30
3.2.	Μοντέλα που έχουν προταθεί για την προστασία της ιδιωτικότητας.....	31
3.2.1.	Εισαγωγή «θορύβου».....	31
3.2.2.	k-ανωνυμία (k-anonymity).....	32
3.2.2.1.	Περιγραφή συγκεκριμένης υλοποίησης: CASPER.....	33
3.2.3.	Άλλες μέθοδοι.....	37



4.	ΜΑΘΗΜΑΤΙΚΗ ΠΕΡΙΓΡΑΦΗ ΠΟΣΟΤΙΚΟΠΟΙΗΣΗΣ ΤΗΣ ΙΔΙΩΤΙΚΟΤΗΤΑΣ ΜΕ ΒΑΣΗ ΤΗΝ ΕΝΤΡΟΠΙΑ.....	38
4.1.	Εισαγωγή.....	38
4.2.	Το προτεινόμενο μοντέλο για την ποσοτικοποίηση της ιδιωτικότητας.....	39
4.2.1.	Χρήση της εντροπίας για τη μέτρηση της ιδιωτικότητας.....	39
4.2.2.	Εντροπία εντοπισμού θέσης.....	41
4.2.3.	Εντροπία προσωπικών προτιμήσεων.....	43
4.2.4.	Ενοποίηση μετρήσεων εντροπίας και χρήση σε ερωτήματα LBS.....	44
4.3.	Ένα πρακτικό παράδειγμα.....	45
4.4.	Συμπεράσματα.....	54
5.	ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗ ΧΡΗΣΤΩΝ ΒΑΣΕΙ ΠΡΟΣΩΠΙΚΩΝ ΠΡΟΤΙΜΗΣΕΩΝ ΜΕ ΧΡΗΣΗ ΜΕΘΟΔΩΝ CLUSTERING.....	56
5.1.	Εισαγωγή.....	56
5.2.	Μοντελοποίηση προφίλ.....	57
5.3.	Μετρικές κοινωνικής δικτύωσης.....	58
5.4.	Μέθοδοι clustering για δημιουργία κοινοτήτων χρηστών.....	59
5.4.1.	K-means clustering.....	59
5.4.2.	Hierarchical clustering.....	60
5.4.3.	Spectral clustering.....	60
5.5.	Αξιολόγηση αλγορίθμων.....	61
5.6.	Συμπεράσματα.....	72
6.	VIRTUAL EARTH TILE SYSTEM.....	74
6.1.	Προβολή χάρτη (Map projection).....	74
6.2.	Εδαφική ανάλυση (Ground resolution) και Κλίμακα χάρτη (Map scale).....	75
6.3.	Συντεταγμένες Pixel.....	78
6.4.	Συντεταγμένες Tile και Quadkeys.....	79
6.5.	Virtual Earth Tile Servers.....	81
7.	Η ΠΛΑΤΦΟΡΜΑ PLASMA.....	83
8.	ΒΑΣΕΙΣ ΓΙΑ ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ.....	87
	ΠΑΡΑΡΤΗΜΑ : Κώδικας.....	89
	ΒΙΒΛΙΟΓΡΑΦΙΑ.....	120

## **1. ΕΙΣΑΓΩΓΙΚΕΣ ΕΝΝΟΙΕΣ**

### **1.1. Αειφανής Υπολογισμός (Ubiquitous Computing)**

Ο αειφανής υπολογισμός (ubiquitous computing ή ubicomp) είναι το μοντέλο επικοινωνίας ανθρώπου-υπολογιστή (human-computer interaction) στο οποίο η επεξεργασία των πληροφοριών έχει ενσωματωθεί πλήρως στα καθημερινά αντικείμενα και δραστηριότητες [1]. Σε αντίθεση με το μοντέλο των επιτραπέζιων υπολογιστών (desktop computers), όπου ο χρήστης δεσμεύει συνειδητά μία απλή συσκευή για έναν ειδικό σκοπό, στο μοντέλο του ubiquitous computing ο χρήστης δεσμεύει ταυτόχρονα πολλές υπολογιστικές συσκευές και συστήματα κατά την εξέλιξη καθημερινών δραστηριοτήτων, χωρίς απαραίτητα να το συνειδητοποιεί πάντοτε. Ο κύριος στόχος τού ubiquitous computing είναι να υπάρχουν μικρές, φθηνές δικτυωμένες υπολογιστικές συσκευές καταναλωμένες σε όλες τα επίπεδα της καθημερινής ζωής. Το ubiquitous computing, λοιπόν, προσβλέπει προς μία εποχή όπου ο μέσος καταναλωτής θα έχει στην κατοχή του εκατοντάδες διασυνδεδεμένες υπολογιστικές συσκευές. Είναι σαφές ότι η διαχείριση των πολυάριθμων αυτών συσκευών δε θα είναι εύκολη και γι' αυτό αναπτύσσονται μέθοδοι για την αυτοματοποίηση των περισσότερων εργασιών και γενικότερα για τη βελτίωση της επικοινωνίας ανθρώπου – υπολογιστή (human-computer interaction). Οι μέθοδοι αυτές θα έχουν ως αποτέλεσμα την μείωση στο ελάχιστο του νοητικού φόρτου που θα επωμίζονται οι χρήστες αφού πολλές εργασίες θα προβλέπονται και θα εκτελούνται αυτόματα. Για να επιτευχθούν αυτοί οι στόχοι, οι υπολογιστές πρέπει να συλλέγουν και να εφαρμόζουν γνώση σχετικά με το εννοιολογικό πλαίσιο του χρήστη στον πραγματικό κόσμο.

### **1.2. Περιβάλλον (Context)**

Η έννοια του context (πλαίσιο, περιβάλλον) έχει ιδιαίτερη σημασία στην πληροφορική και στις επικοινωνίες. Παρόλα αυτά, είναι εξαιρετικά δύσκολο για τους επιστήμονες της πληροφορικής να δώσουν έναν επακριβή και πλήρη ορισμό για τον όρο αυτό. Σύμφωνα με τους Dey και Abowd [2], [3], ως context ορίζουμε οποιαδήποτε πληροφορία που χρησιμοποιείται για να χαρακτηρίσει την κατάσταση μιας οντότητας. Ως οντότητα

ορίζεται ένα πρόσωπο, μια τοποθεσία ή ένα αντικείμενο που θεωρείται σχετικό με την αλληλεπίδραση μεταξύ του χρήστη και μιας εφαρμογής. Οντότητα μπορεί να θεωρηθεί και ο ίδιος ο χρήστης ή η εφαρμογή. Στα [2], [3] προτείνονται τέσσερις κύριοι τύποι context για να χαρακτηρίσουν μία συγκεκριμένη κατάσταση μία οντότητας:

- *Ταυτότητα (Identity)*: η ταυτότητα των σχετικών οντοτήτων
- *Τοποθεσία (Location)*: η γεωγραφική τοποθεσία των σχετικών οντοτήτων
- *Δραστηριότητα (Activity)*: η δραστηριότητα ή οι δραστηριότητες που εκτελούνται
- *Χρόνος (Time)*: η χρονική περίοδος στην οποία οι οντότητες εκτελούν την δραστηριότητα

Όλοι οι άλλοι τύποι πληροφορίας πλαισίου ορίζονται ως δευτερεύοντες γιατί συχνά για να αποκτήσουν νόημα απαιτούν συσχέτιση με ένα ή περισσότερα μέρη του κύριου εννοιολογικού πλαισίου.

Στη συνέχεια παρουσιάζεται μία ομαδοποίηση διαφορετικών τύπων πλαισίου με ειδική αναφορά στις κινητές υπηρεσίες που στηρίζονται σε χάρτες:

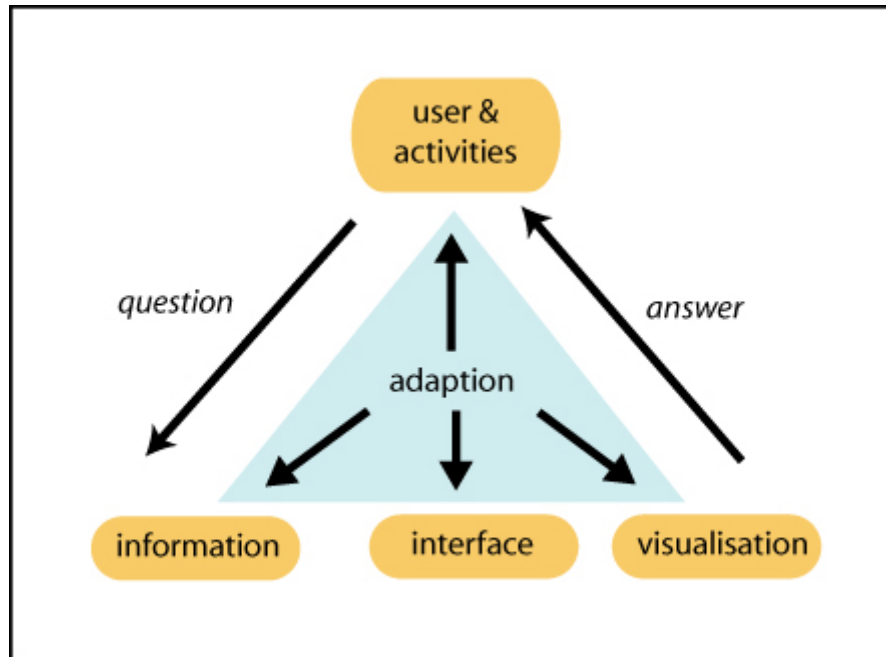
- *Κινητός Χρήστης (Mobile map user)*: Η ταυτότητα του χρήστη είναι σημαντική ώστε να επιτρέψει στην υπηρεσία να έχει πληροφορίες σχετικές με θέματα όπως η ηλικία και το φύλο του, οι προσωπικές του προτιμήσεις, οι φίλοι και οι συνάδελφοι του.
- *Θέση (Location)*: Η τοποθεσία είναι το στοιχείο του context που λαμβάνεται πιο συχνά υπόψη. Επιτρέπει σε πληροφορίες και υπηρεσίες να προσδιοριστούν σε συγκεκριμένη θέση (localization).
- *Χρόνος (Time)*: Ο χρόνος μπορεί να αναφέρεται σε συγκεκριμένες στιγμές της ημέρας ή σε μεγαλύτερα χρονικά διαστήματα όπως πρωί, απόγευμα ή βράδυ, μέρα της εβδομάδας, του μήνα ή εποχή του χρόνου.
- *Προσανατολισμός (Orientation)*: Ο προσανατολισμός του χρήστη είναι σημαντικός ώστε να καθοριστεί η κατεύθυνση του χρήστη και έτσι τι βρίσκεται μπροστά του και τι πίσω.
- *Ιστορικό πλοήγησης (Navigation history)*: Το ιστορικό πλοήγησης επιτρέπει στους χρήστες να βλέπουν πού έχουν πάει, τι έχουν κάνει και τι έχουν δει. Είναι

χρήσιμο στην πλοήγηση ώστε να κατευθύνει το χρήστη καθώς κινείται και να του επιτρέψει να επιστρέψει πίσω αν χαθεί.

- *Σκοπός χρήσης (Purpose of use)*: Ο σκοπός χρήσης ορίζεται από ενέργειες, στόχους, εργασίες και ρόλους των χρηστών. Διαφορετικοί τύποι χρήσεως απαιτούν διαφορετικούς τύπους πληροφορίας, τύπους παρουσίασης (π.χ. χάρτη, κείμενο ή ήχο), τύπους αλληλεπίδρασης.
- *Κοινωνική κατάσταση (Social and cultural situation)*: Η κοινωνική κατάσταση ενός χρήστη χαρακτηρίζεται από τη γειτνίαση με τους άλλους, τις κοινωνικές σχέσεις, τις συλλογικές εργασίες.
- *Φυσικό περιβάλλον (Physical Surroundings)*: Το φυσικό περιβάλλον περιλαμβάνει στοιχεία όπως το επίπεδο φωτισμού, ο περιβάλλον θόρυβος, κ.λπ.
- *Ιδιότητες συστήματος (System Properties)*: Αυτό σχετίζεται με το τι σύστημα χρησιμοποιεί ο χρήστης. Τι τύπο συσκευής και ποιες οι δυνατότητές της (π.χ. touch screen, έγχρωμη ή ασπρόμαυρη οθόνη κλπ.), αν έχουν πρόσβαση συνεχώς στο Internet ή είναι διακοπτόμενη, τι εύρος ζώνης (bandwidth) έχει η σύνδεση, τι ποιότητα έχουν οι πληροφορίες εντοπισμού (positioning information).

Τα συστήματα που αλλάζουν δυναμικά τη συμπεριφορά τους εξαιτίας του context χαρακτηρίζονται με διάφορες ονομασίες: reactive, responsive, situated, context-sensitive και environment directed [4], ωστόσο ο όρος adaptive έχει επικρατήσει στην κινητή χαρτογραφία [5], [6]. Η προσαρμογή μπορεί να λάβει χώρα σε τέσσερα επίπεδα [5]:

1. *Επίπεδο πληροφορίας (Information level)* – Προσαρμόζεται το περιεχόμενο της πληροφορίας. Για παράδειγμα, φιλτράρισμα πληροφοριών σύμφωνα με αυτές που βρίσκονται πιο κοντά στη θέση του χρήστη
2. *Επίπεδο τεχνολογίας (Technology level)* – Η πληροφορία κωδικοποιείται ώστε να ταιριάζει σε συσκευές με διαφορετικά τεχνικά χαρακτηριστικά.
3. *Επίπεδο διεπαφής (User interface level)* – Προσαρμόζεται η διεπαφή, π.χ. αυτόματη προσαρμογή του χάρτη καθώς ο χρήστης κινείται
4. *Επίπεδο παρουσίασης (Presentation level)* – Προσαρμόζεται η εμφάνιση της πληροφορίας. Για παράδειγμα, όποιο αποτέλεσμα είναι πιο κοντά στις επιλογές του χρήστη τονίζεται με πιο έντονα χρώματα ενώ τα υπόλοιπα με πιο αχνά.



Σχήμα 1.1: Επίπεδα προσαρμοστικότητας κατά Reichenbacher [6]

### 1.3. Επίγνωση περιβάλλοντος (Context awareness)

Παραδοσιακά οι υπολογιστές ήταν προσανατολισμένοι σε σενάρια χρήσης στο σπίτι και στο γραφείο. Σήμερα όμως η κινητή υπολογιστική (mobile computing) και η διεισδυτική υπολογιστική (pervasive computing) μάς επιτρέπουν να κινηθούμε πέρα από αυτά τα δεσμευτικά πεδία [7]. Η ενσωμάτωση των υπολογιστών στην καθημερινή μας ζωή συχνά αναφέρεται ως pervasive computing. Η κύρια διαφορά του pervasive computing με το ubiquitous computing, εστιάζεται στο γεγονός ότι το pervasive computing έχει πιο στατική δομή ενώ το ubiquitous computing υποστηρίζει κινητούς hosts (mobile hosts) και κινητό κώδικα (mobile code) [8]. Στις αρχές της δεκαετίας του 1990 ο Weiser [9] παρατήρησε τις νέες αυτές τάσεις και τις χρησιμοποίησε για να κάνει ορισμένες προβλέψεις σχετικά με την επικοινωνία ανθρώπου-υπολογιστή τον 21<sup>ο</sup> αιώνα. Εφηύρε τον όρο ubiquitous computing [9] για να περιγράψει έναν κόσμο όπου οι υπολογιστές εμπλέκονται στη δομή της καθημερινής ζωής ώσπου να μην μπορούν να διακριθούν από αυτή [7].

Οι άνθρωποι χρησιμοποιούν τις πέντε αισθήσεις τους για να αλληλεπιδρούν με το περιβάλλον, μοιράζονται μια κοινή πλούσια γλώσσα και έχουν μια γενική αντίληψη για

τον κόσμο. Αντιθέτως, τα παραδοσιακά υπολογιστικά συστήματα έχουν ένα πολύ περιορισμένο σύνολο επικοινωνιακών διαύλων (όπως η οθόνη, το ποντίκι και το πληκτρολόγιο) [7]. Η υπολογιστική με επίγνωση πλαισίου ή περιβάλλοντος (context-aware computing) [10] προσθέτει στους υπολογιστές αισθητήρες (sensors) και εκκινητήρες (actuators) ώστε να κατανοήσει καλύτερα το φυσικό περιβάλλον και την αλληλεπίδραση με αυτό. Τέτοια συστήματα συλλέγουν δεδομένα από τους αισθητήρες, χτίζουν ένα μοντέλο από το περιβάλλον και χρησιμοποιούν το μοντέλο για να παρέχουν πιο χρήσιμες και διαισθητικές (intuitive) υπηρεσίες στους χρήστες. Συνεπώς, τα αποτελεσματικά context-aware συστήματα θα πρέπει να είναι πιο εύχρηστα από τα παραδοσιακά πληροφοριακά συστήματα. Με αυτόν τον τρόπο, το context-aware computing προσφέρει ένα βήμα προς τον τελικό στόχο που είναι το ubiquitous computing. Η αυτοματοποίηση της συλλογής και ερμηνείας δεδομένων πλαισίου (contextual data) σχετικά με μια συγκεκριμένη υπηρεσία είναι απαραίτητη. Χωρίς ένα τέτοιο σύστημα ο χρήστης θα πρέπει να παρέχει τις πληροφορίες συνεχώς, κάτι που θέλουμε να αποφύγουμε, αφού ο συνδυασμός mobile και pervasive computing συχνά έχει ως αποτέλεσμα ένα συνεχώς μεταβαλλόμενο πλαίσιο για το χρήστη (user context) [7].

Πολλές έρευνες σχετικά με το ubiquitous computing εστιάζουν στην βελτίωση της επικοινωνίας ανθρώπου – υπολογιστή. Έτσι ο «προληπτικός υπολογισμός» (proactive computing) [11] επιδιώκει να περιορίσει την επικοινωνία ανθρώπου – υπολογιστή σε ένα πιο εποπτικό ρόλο. Το proactive computing προσπαθεί να επεκτείνει το πεδίο της εφαρμογής (application domain) και να μειώσει την άμεση ανθρώπινη ανάμιξη συνδέοντας τους υπολογιστές άμεσα με τον φυσικό κόσμο με sensors και actuators. Τα δεδομένα πρέπει να υφίστανται επεξεργασία σε πραγματικό χρόνο, μειώνοντας την καθυστέρηση που προκύπτει από την αναμονή επιβεβαίωσης από το χρήστη. Οι ανάγκες του χρήστη πρέπει να προβλέπονται χρησιμοποιώντας στατιστικά μοντέλα ώστε να αντιμετωπιστεί η αβεβαιότητα.

#### **1.4. Επίγνωση θέσης (Location awareness)**

Μέχρι πρότινος, οι υπολογιστές διέθεταν μεθόδους για τον προσδιορισμό και την κοινή χρήση δύο εκ των τεσσάρων κύριων context types: identity και time. Η επίγνωση θέσης (location awareness) είναι κάτι που έγινε απαραίτητο πρόσφατα. Η γνώση της ταυτότητας και της ώρας είναι καλά ορισμένες εφαρμογές. Η ύπαρξη υπολογιστών με γνώση του εννοιολογικού πλαισίου του χρήστη συνεπάγεται ότι οι υπολογιστές θα συλλέγουν, θα ταξινομούν και θα διανέμουν πολύ περισσότερες προσωπικές πληροφορίες για τους χρήστες. Εξ αυτού το γεγονός προκύπτει άλλωστε το δύσκολο ζήτημα του συνδυασμού των αναγκών για context-aware computing με την επιθυμία των χρηστών να διατηρήσουν τον έλεγχο στην διανομή και την διάδοση προσωπικών πληροφοριών.

Στον τομέα του ubiquitous computing, η πληροφορία σχετικά με την τοποθεσία (location based information) είναι ένα τα πιο σημαντικά δεδομένα εννοιολογικού πλαισίου και χρησιμοποιείται σε ένα μεγάλο εύρος εφαρμογών. Επιπρόσθετα, όταν τα location systems ανιχνεύουν αυτόματα και συνεχώς τη θέση του χρήστη, παράγεται μία μεγάλη ποσότητα εν δυνάμει ευαίσθητων πληροφοριών. Οι χρήστες δεν επιθυμούν απαραίτητα να αποτρέψουν κάθε πρόσβαση σε πληροφορίες που αφορούν την τοποθεσία τους αφού μερικές εφαρμογές μπορούν να χρησιμοποιήσουν αυτές τις πληροφορίες για την παροχή χρήσιμων πληροφοριών. Ωστόσο επιθυμούν να έχουν τον έλεγχο. Για το λόγο αυτό, αρκετή από την τρέχουσα έρευνα στην προστασία της ιδιωτικότητας της θέσης (location privacy) για το ubiquitous computing έχει συγκεντρωθεί στον ορισμό μηχανισμών που θα επιτρέπουν στους χρήστες να ελέγχουν την πρόσβαση στις πληροφορίες που αφορούν τη θέση τους.

#### **1.5. Υπηρεσίες βασισμένες στη θέση (Location-Based Services)**

Οι υπηρεσίες που βασίζονται στη θέση (Location-Based Services – LBS) είναι υπηρεσίες που αναπτύσσονται και διανέμονται από ασύρματους φορείς και παρέχουν πληροφορίες που σχετίζονται ή εξαρτώνται από την θέση του χρήστη. Πλέον οι χρήστες που χρησιμοποιούν κινητές συσκευές δεν χρειάζεται να παρέχουν οι ίδιοι πληροφορίες σχετικά με την θέση τους (π.χ. τον ταχυδρομικό τους κώδικα) ώστε να χρησιμοποιήσουν

location-based services. Οποιαδήποτε υπηρεσία λοιπόν χρησιμοποιεί την θέση του χρήστη μπορεί να χαρακτηριστεί ως location-based service.

Όπως οι περισσότερες ηλεκτρονικές υπηρεσίες, τα LBS μπορούν να κατηγοριοποιηθούν σε τέσσερις μεγάλες κατηγορίες. Οι κατηγορίες αυτές είναι:

- *Επιχείρηση προς Επιχείρηση (Business to Business)* – παράδειγμα υπηρεσιών αυτής της κατηγορίας είναι υπηρεσίες Fleet Management δηλαδή υπηρεσίες διαχείρισης εταιρικού στόλου οχημάτων (παρακολούθηση σε πραγματικό χρόνο, δρομολόγηση κλπ.)
- *Επιχείρηση προς Πελάτη (Business to Consumer)* – παράδειγμα υπηρεσιών αυτής της κατηγορίας είναι υπηρεσίες προώθησης διαφημίσεων που βασίζονται στη θέση όπου ένας χρήστης λαμβάνει διαφημίσεις σχετικές με την θέση του και της επιλογές του.
- *Πελάτης προς Επιχείρηση (Consumer to Business)* – παράδειγμα υπηρεσιών αυτής της κατηγορίας είναι υπηρεσίες Finder & Router (εύρεση κοντινότερου σημείου ενδιαφέροντος π.χ. εστιατορίου ή βενζινάδικου και πρόταση βέλτιστης διαδρομής προς αυτό
- *Πελάτης προς Πελάτη (Consumer to Consumer)* – παράδειγμα υπηρεσιών αυτής της κατηγορίας είναι υπηρεσίες εύρεσης φίλων όπου ο χρήστης ειδοποιείται αν κάποιος φίλος του βρίσκεται σε ακτίνα κάποιων μέτρων (εφόσον ο φίλος του βέβαια έχει επιτρέψει την ανίχνευση του από το χρήστη).

Οι LBS είναι υπηρεσίες που είναι προσβάσιμες από κινητές συσκευές μέσω κινητού δικτύου (mobile network) και εκμεταλλεύονται την ικανότητα να χρησιμοποιούν τη θέση της κινητής συσκευής.

Τα Geographic Information Systems (GIS) έχουν κάποιες ομοιότητες με τα LBS. Τέτοια κοινά χαρακτηριστικά είναι η διαχείριση δεδομένων με αναφορά σε θέση και μεθόδους ανάλυσης του χώρου (spatial analysis functions) που δίνουν απαντήσεις σε ερωτήματα όπως:

- Πού βρίσκομαι;
- Τι βρίσκεται κοντά;
- Πώς μπορώ να φτάσω;



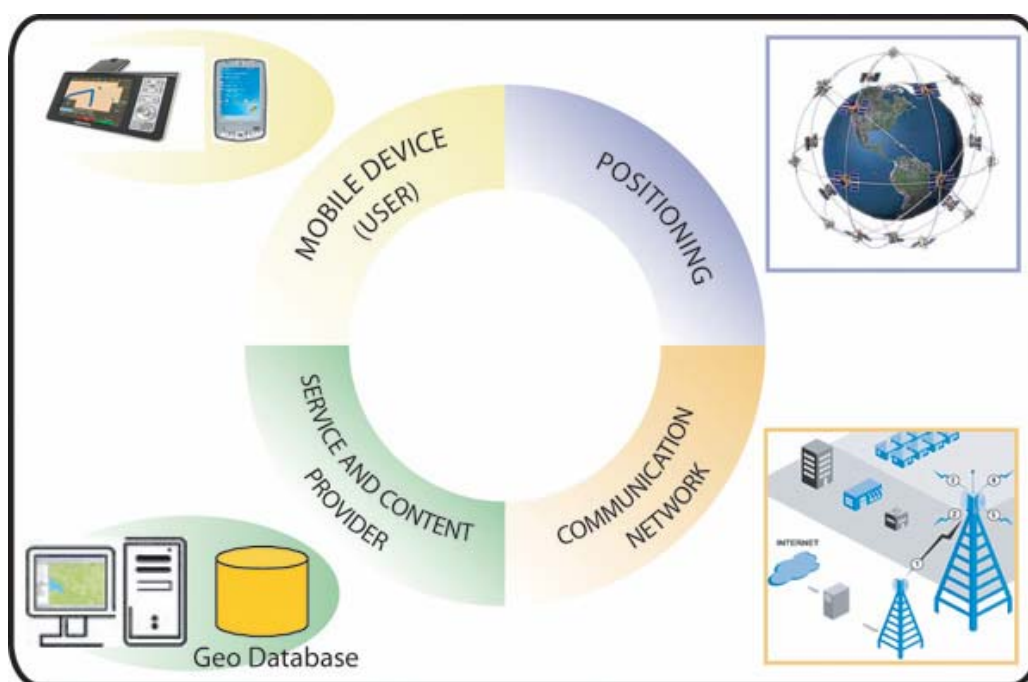
Ωστόσο οι LBS και τα GIS έχουν διαφορετική προέλευση και απευθύνονται σε διαφορετικές ομάδες χρηστών [12]. Τα GIS αναπτύχθηκαν κατά την διάρκεια προηγούμενων δεκαετιών στη βάση επαγγελματικών γεωγραφικών εφαρμογών. Αντίθετα, τα LBS αναπτύχθηκαν αρκετά πρόσφατα με την εξέλιξη των κινητών υπηρεσιών. Όσο αφορά στους χρήστες στους οποίους απευθύνονται, τα GIS μπορούν να θεωρηθούν πιο επαγγελματικά συστήματα που απευθύνονται σε έμπειρους χρήστες και γι' αυτό προσφέρουν μεγάλο εύρος λειτουργιών. Αντιθέτως οι LBS αναπτύχθηκαν σαν συγκεκριμένες υπηρεσίες για μη εξειδικευμένους χρήστες. Εξάλλου, οι LBS έχουν να αντιμετωπίσουν τους περιορισμούς των κινητών υπολογιστικών συσκευών, όπως χαμηλή υπολογιστική ισχύ, μικρές οθόνες και μικρή διάρκεια ζωής μπαταρίας της κινητής συσκευής [6].

### 1.5.1. Αρχιτεκτονική LBS

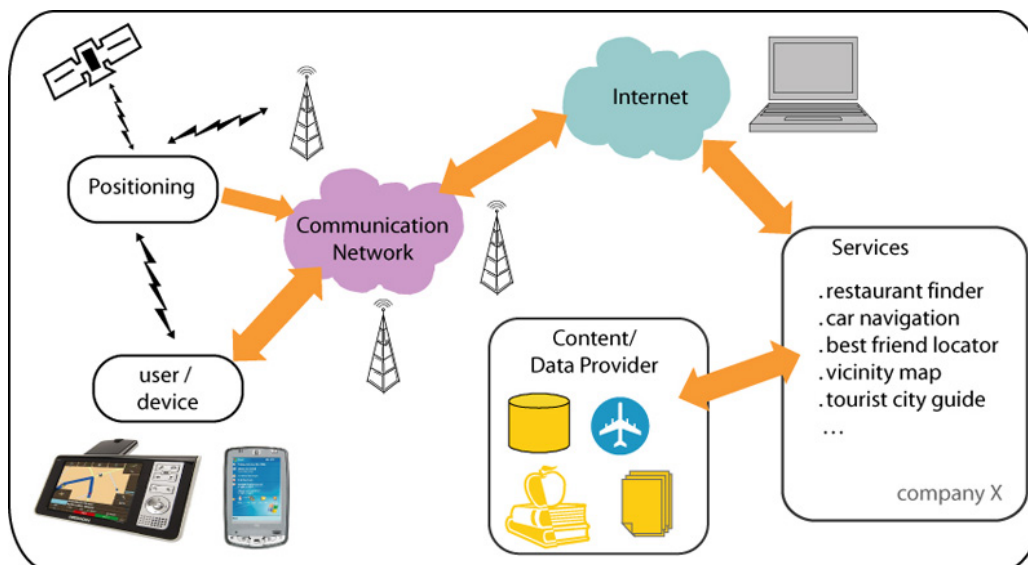
Τα βασικά μέρη μιας LBS είναι τα ακόλουθα:

- *Κινητές συσκευές (Mobile Devices)*: Ένα εργαλείο για το χρήστη ώστε να ζητάει τις πληροφορίες που επιθυμεί. Τα αποτελέσματα μπορούν να δοθούν ηχητικά, με κείμενο ή εικόνες. Τέτοιες συσκευές είναι φορητοί υπολογιστές, PDAs, smartphones, κινητά τηλέφωνα ακόμα και μία μονάδα πλοήγησης ενός αυτοκινήτου.
- *Δίκτυο επικοινωνίας (Communication Network)*: Το δίκτυο που μεταφέρει τα δεδομένα του χρήστη και τα αιτήματα υπηρεσίας από το κινητό τερματικό προς τον πάροχο της υπηρεσίας (service provider) και εν συνεχεία τις ζητούμενες πληροφορίες πίσω στον χρήστη.
- *Τμήμα Προσδιορισμού Θέσης (Positioning Component)*: Η θέση του χρήστη μπορεί να υπολογιστεί είτε κάνοντας χρήση κυρίως του κινητού δικτύου επικοινωνίας είτε χρησιμοποιώντας το Παγκόσμιο Σύστημα Προσδιορισμού Θέσης (Global Positioning System – GPS). Υπάρχουν και άλλες μέθοδοι, κυρίως για την ανίχνευση της θέσης του χρήστη μέσα σε εσωτερικού χώρους. Σε περιπτώσεις που η θέση δεν προσδιορίζεται αυτόματα μπορεί να προσδιοριστεί από τον χρήστη.

- *Πάροχος υπηρεσίας (Service Provider)*: Ο πάροχος της υπηρεσίας προσφέρει έναν αριθμό διαφορετικών υπηρεσιών στο χρήστη και είναι υπεύθυνος για την επεξεργασία του αιτήματος υπηρεσίας. Τέτοιες υπηρεσίες προσφέρουν για παράδειγμα τον υπολογισμό της θέσης, την εύρεση μιας διαδρομής, την αναζήτηση συγκεκριμένων πληροφοριών για αντικείμενα που ενδιαφέρουν το χρήστη κ.λπ.
- *Πάροχος δεδομένων και περιεχομένου (Data and Content Provider)*: Οι πάροχοι υπηρεσιών συνήθως δεν αποθηκεύουν και δεν διατηρούν όλες τις πληροφορίες που μπορούν να ζητηθούν από τους χρήστες. Γι' αυτό τα βασικά γεωγραφικά δεδομένα και πληροφορίες για τοποθεσίες μπορούν συνήθως να ζητηθούν από έναν τρίτο πάροχο (π.χ. εταιρείες παροχής χαρτών, εταιρείες πληροφόρησης για την κίνηση στους δρόμους κλπ.) [6].



**Σχήμα 1.2:** Τα βασικά μέρη μιας LBS: Χρήστης, Δίκτυο Επικοινωνίας, Πάροχος Υπηρεσίας Εντοπισμού Θέσης, Πάροχος Περιεχομένου [6]



**Σχήμα 1.3:** Βασικά τμήματα μιας LBS και ροή πληροφορίας [6]

Σε γενικές γραμμές, οι location based υπηρεσίες μπορούν να διαχωριστούν σε δύο είδη ανάλογα με το αν η πληροφορία μεταφέρεται μετά από αλληλεπίδραση με το χρήστη ή όχι. Έτσι έχουμε:

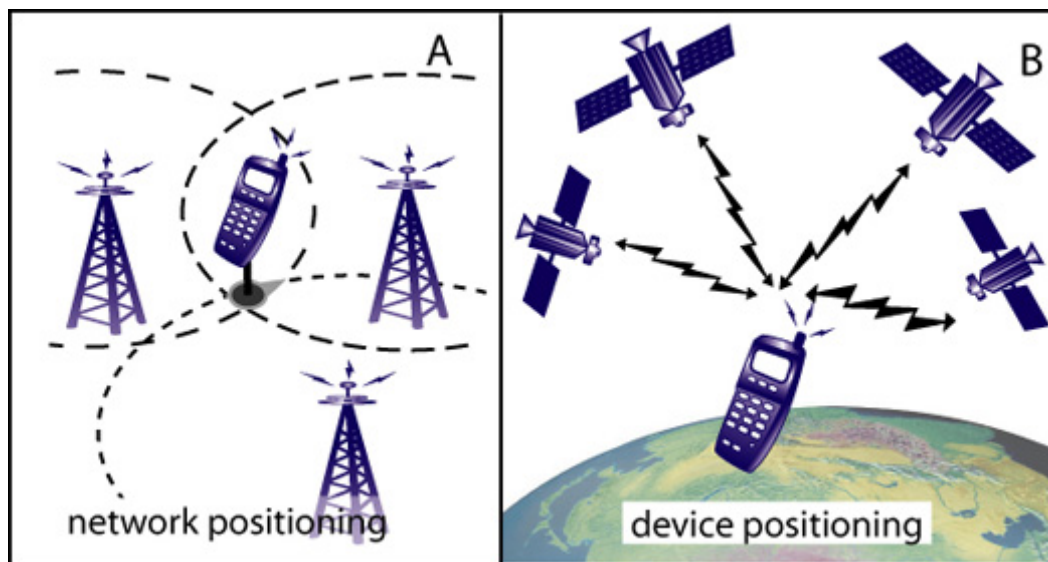
- Push Services, που μεταφέρουν την πληροφορία που ζητείται άμεσα από τον χρήστη.
- Pull Services, που μεταφέρουν πληροφορία, χωρίς να ζητηθεί ή με το να ζητηθεί έμμεσα από το χρήστη. Τέτοιες υπηρεσίες ενεργοποιούνται από ένα γεγονός που μπορεί να συμβεί όταν ο χρήστης βρεθεί σε μία περιοχή ή όταν ένα χρονικό διάστημα έχει παρέλθει [6].

Η βασική ιδέα πίσω από τις LBS είναι να απαντήσει σε ερωτήσεις όπως: «Πού βρίσκομαι;», «Πού είναι οι φίλοι μου;», «Τι βρίσκεται κοντά μου;». Όταν οι χρήστες βρεθούν σε ένα περιβάλλον σχετικά άγνωστο, η συμπεριφορά τους και οι ανάγκες τους είναι αρκετά προβλέψιμες είτε βρίσκονται στην χώρα τους είτε στο εξωτερικό, είτε περπατούν, είτε οδηγούν ένα όχημα. Οι χρήστες έχουν ανάγκη να βρουν κάτι να φάνε, ίσως να βρουν ένα φαρμακείο, ένα ATM ανάληψης χρημάτων, μία αφετηρία ταξί, κ.λπ. Όταν κάποιος βρεθεί στο εξωτερικό υπάρχουν επιπλέον ανάγκες όπως το να βρει κανείς τοπικά τουριστικά αξιοθέατα, να βρει ένα ξενοδοχείο ή ένα ανταλλακτήριο. Καθώς κάποιος οδηγεί, ενδέχεται να υπάρχουν άλλες ανάγκες όπως βοήθεια στην εύρεση μιας διαδρομής σε μία άγνωστη πόλη ή λεπτομέρειες για οδική βοήθεια.

### 1.5.2. Μέθοδοι εντοπισμού και ακρίβεια

Μια γενική ομαδοποίηση των μεθόδων εντοπισμού μπορεί να γίνει ως εξής:

Η πρώτη ομάδα ονομάζεται *network-based positioning*. Η παρακολούθηση και η επαλήθευση της θέσης ενός χρήστη γίνεται χρησιμοποιώντας δίκτυο σταθμών βάσης (*base station network*). Προς τούτο, η κινητή συσκευή στέλνει ένα σήμα προς το δίκτυο ή δέχεται από αυτό. Η δεύτερη ομάδα ονομάζεται *terminal-based positioning*. Εδώ η θέση υπολογίζεται από την ίδια τη συσκευή του χρήστη από σήματα που δέχεται από τους σταθμούς βάσης. Το πιο γνωστό παράδειγμα για τα *terminal-based positioning systems* είναι το *Global Positioning System (GPS)*. Οι σταθμοί βάσης για το σύστημα GPS είναι δορυφόροι GPS. Θα μπορούσε να διακριθεί και μια τρίτη ομάδα τεχνικών εντοπισμού, η οποία προκύπτει από το συνδυασμό *network-based positioning* και *terminal-based positioning* συστημάτων.



**Σχήμα 1.4:** Network-based positioning device-based positioning μέθοδοι εντοπισμού [6]

Οι βασικές αρχές για τον υπολογισμό της θέσης του χρήστη είναι:

- Οι σταθμοί βάσης έχουν γνωστή θέση
- Οι πληροφορίες από ένα σήμα μετασχηματίζονται σε αποστάσεις
- Η θέση του χρήστη υπολογίζεται με τη χρησιμοποίηση των αποστάσεων που υπολογίσθηκαν στους σταθμούς βάσεων.

Στη συνέχεια παρουσιάζονται οι βασικές τεχνικές που χρησιμοποιούνται πιο συχνά για τον εντοπισμό θέσης:

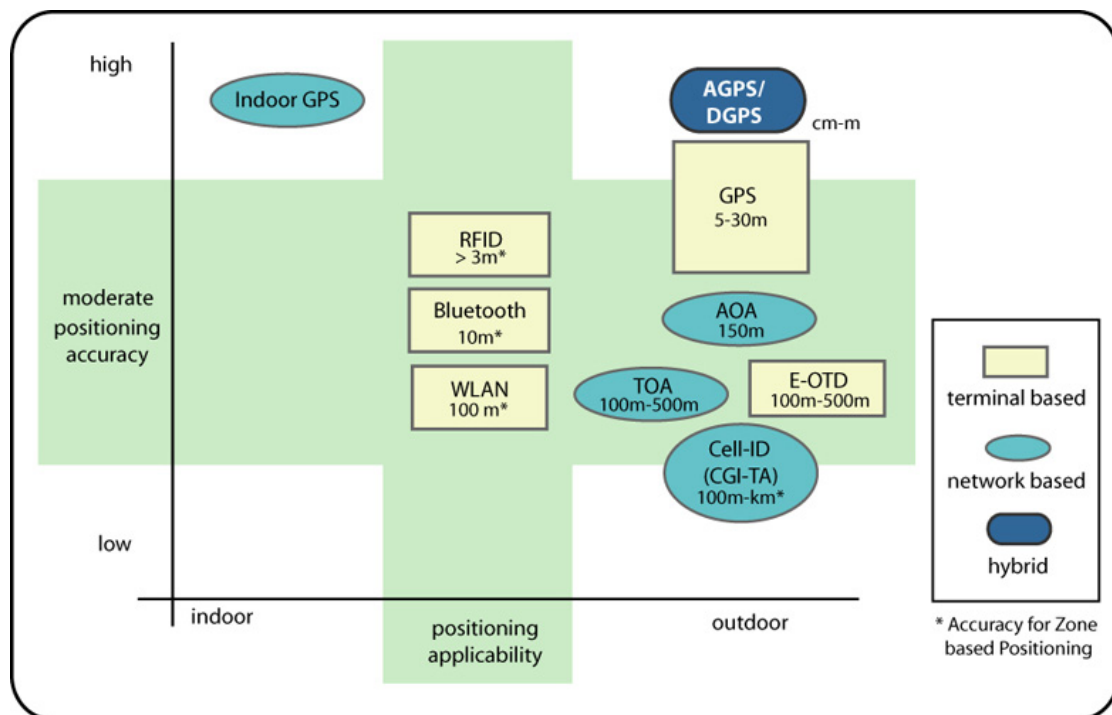
Cell of origin (COO), location signature, location beacons: Το αναγνωριστικό (id) της κυψέλης είναι συνήθως το αναγνωριστικό του πλησιέστερου σταθμού βάσης, π.χ. μια κεραία κινητής τηλεφωνίας. Με αυτή τη τεχνική, η θέση είναι γνωστή σε έναν ορισμένο κύκλο ή κυψέλη γύρω από τον σταθμό βάσεως που βρίσκεται φυσικά σε γνωστή θέση. Σταθμοί μετάδοσης (beacons) (π.χ. με υπέρυθρες ή RFID) χρησιμοποιούνται κυρίως σε εσωτερικούς χώρους. Τα beacons έχουν ένα αναγνωριστικό ή μεταδίδουν την ακριβή τους τοποθεσία στη κινητή συσκευή που πλησιάζει.

Time of Arrival (TOA): Καθώς τα ηλεκτρομαγνητικά σήματα διαδίδονται με την ταχύτητα του φωτός, γνωρίζοντας την ταχύτητα και την διαφορά χρόνου μεταξύ αποστολής και λήψης, μπορεί να υπολογιστεί η απόσταση. Η ταχύτητα του φωτός είναι περίπου 300.000km/s και έτσι ο χρόνος εκτέλεσης είναι πολύ μικρός και για αυτό απαιτούνται χρονόμετρα ακριβείας. Η ίδια αρχή μπορεί να χρησιμοποιηθεί για πιο αργά σήματα όπως τα υπερηχητικά.

Time Difference of Arrival (TDOA), Enhanced Observed Time Difference (E-OTD): Και αυτές οι τεχνικές υπολογίζουν την απόσταση μετρώντας το χρόνο εκτέλεσης, αλλά χρησιμοποιούν τη διαφορά χρόνου μεταξύ των σημάτων από τρεις (συνήθως) διαφορετικούς σταθμούς βάσεως. Έτσι, έχοντας σήματα από διαφορετικούς γειτονικούς σταθμούς βάσεως, η θέση μπορεί να υπολογιστεί (triangulation). Στην περίπτωση του

TDOA, ο υπολογισμός της θέσης γίνεται από τον πάροχο του δικτύου, ενώ στην περίπτωση του E-ODT γίνεται από την κινητή συσκευή.

Angle of Arrival (AOA), Direction of Arrival (DOA): Χρησιμοποιώντας κεραίες με συγκεκριμένα χαρακτηριστικά, μπορεί να ανιχνευθεί η γωνία άφιξης (angle of arrival) στην κινητή συσκευή. Επειδή για μία κινητή συσκευή αυτό δεν είναι ικανοποιητικά ακριβές, μια εναλλακτική είναι πολλοί σταθμοί βάσης που έχουν περισσότερες της μίας κεραίες (συνήθως 2-4) που διαιρούν τον περιγεγραμμένο κύκλο των σταθμών βάσεων σε τμήματα των 90,120 ή 180 μοιρών.



**Σχήμα 1.5:** Μέθοδοι εντοπισμού θέσης, ακρίβεια και εφαρμογή (AGPS: Assisted GPS, AOA: Angle of Arrival, TOA: Time of Arrival, E-OTD: Enhanced Observed Time Difference) [6]

Οι δύο πιο διαδεδομένες τεχνικές εντοπισμού είναι το GPS και η επαλήθευση θέσης κάνοντας χρήση Cell-DI από τον πλησιέστερο πομποδέκτη σταθμού βάσεως.

## **2. ΟΡΙΣΜΟΣ ΠΡΟΦΙΛ ΧΡΗΣΤΗ**

### **2.1. Εισαγωγή**

Για την καθημερινή μας πρόσβαση σε πληροφορίες με σκοπό την εργασία, την ενημέρωση, την ψυχαγωγία και την επικοινωνία με άλλους, χρησιμοποιούμε μια ποικιλία συσκευών, δικτυακών διεπαφών (networking interfaces), εφαρμογών και πρωτοκόλλων. Η ανάγκη για μια μοναδική, εξατομικευμένη (personalized), προσανατολισμένη στο χρήστη προσέγγιση σε αυτό το θέμα συντείνει στην εμφάνιση της έννοιας των προφίλ (profiles), η οποία μπορεί να περιγράψει με σαφήνεια το πλαίσιο μέσα στο οποίο μπορεί να γίνει ανταλλαγή πληροφοριών. Μέχρι σήμερα, διάφορες ομάδες εργασίας έχουν ασχοληθεί με τον ορισμό και την εφαρμογή προφίλ, ενώ έχουν γίνει και απόπειρες για προτυποποίηση από σχετικούς οργανισμούς.

#### **2.1.1. Πρότυπα για τον ορισμό προφίλ**

Η διεξοδικότερη εργασία στο εν λόγω πεδίο έχει γίνει από το European Telecommunications Standards Institute (ETSI), το οποίο έχει εκδώσει οδηγίες για τη Διαχείριση Προφίλ Χρήστη (User Profile Management) [13], όπου εισάγονται και περιγράφονται οι έννοιες: προφίλ χρήστη (user profile), profile agents, stakeholders ρόλων (roles), κανόνων (rules) και ενεργοποίησης προφίλ (profile activation). Σύμφωνα με το ETSI, ένα προφίλ χρήστη περιέχει πληροφορίες του χρήστη και των προσωπικών του απαιτήσεων σε τέτοια μορφή που να μπορούν να χρησιμοποιηθούν από το σύστημα, ώστε αυτό να επιδεικνύει την απαιτούμενη συμπεριφορά, ενώ διακρίνει τρεις τύπους προφίλ. Επιπρόσθετα, το ETSI έχει εκδώσει μια σειρά από τεχνικές προδιαγραφές [14], [15], [16] που ορίζουν το Γενικό Προφίλ Χρήστη - Generic User Profile (GUP) για το κινητό σύστημα 3GPP. Ο απώτερος στόχος των προδιαγραφών αυτών είναι να καταστεί δυνατή η αρμονική, συνδυασμένη χρήση πληροφοριών σχετικών με το χρήστη από διαφορετικούς τομείς (domains). Σκοπός είναι η διευκόλυνση της διαχείρισης προτιμήσεων του χρήστη, της εξατομικεύσης υπηρεσιών, της ανταλλαγής πληροφοριών και της διαχείρισης δυνατοτήτων του τερματικού.

Εκτός του ETSI, το W3C έχει επίσης εκδώσει οδηγία αναφορικά με προφίλ. Πιο συγκεκριμένα, το έγγραφο “Composite Capabilities / Preference Profiles (CC/PP):

Structure and Profiles 1.0” [17] ορίζει ένα προφίλ CC/PP ως την περιγραφή των δυνατοτήτων της συσκευής και των προτιμήσεων του χρήστη. Ας σημειωθεί ότι στο [17] ο όρος profile δεν αναφέρεται σε κάποιο υποσύνολο κάποιων συγκεκριμένων προδιαγραφών, αλλά στο έγγραφο που περιγράφει τις δυνατότητες μιας συσκευής. Το πλαίσιο Resource Description Framework (RDF) [18] χρησιμοποιείται για τη δημιουργία προφίλ. Αναφορικά με τη δομή, ένα προφίλ CC/PP κατασκευάζεται ως μια ιεραρχία δύο επιπέδων: ένα προφίλ με ένα ή περισσότερα συστατικά (π.χ. πλατφόρμα hardware, πλατφόρμα software, μια εφαρμογή όπως ένας browser), με κάθε συστατικό να έχει τουλάχιστον ένα γνώρισμα (δηλαδή ένα υποδένδρο, οι κλάδοι του οποίου είναι οι δυνατότητες ή προτιμήσεις σχετικές με το συγκεκριμένο συστατικό). Παρότι η μέχρι πρότινος χρήση κυρίως εστιάζεται στις δυνατότητες των συσκευών, το προφίλ CC/PP μπορεί να μεταφέρει πληροφορία σχετική με προτιμήσεις του χρήστη.

Σε μια απόπειρα να συγκρίνουμε το πρότυπο του ETSI με αυτό του W3C, μπορούμε να πούμε ότι το πρώτο είναι πολύ περισσότερο προσανατολισμένο στις τηλεπικοινωνίες (telecom oriented)[19], ενώ το δεύτερο στο διαδίκτυο (web oriented). Τα χαρακτηριστικά που προτείνονται από το ETSI καθιστούν το πρότυπο αυτό μια κατάλληλη βάση για τη δική μας περίπτωση, ήτοι την παροχή εξατομικευμένων υπηρεσιών (personalized services) για κινητά τερματικά.

### **2.1.2. Υπάρχουσα έρευνα στον ορισμό προφίλ**

Έρευνα πάνω στον ορισμό προφίλ έχει δημοσιευθεί και σε μνημόνια της IETF (IETF memoranda). Στο [20] (work in progress) προτείνεται ένα πλαίσιο που ορίζει τρεις τύπους προφίλ (τοπικού δικτύου, συσκευής και χρήστη) και περιγράφει, μεταξύ άλλων, τον κύκλο ζωής προφίλ (profile life cycle), τις διαδικασίες αλλαγής και ανάκτησης του προφίλ, κ.α.

Ο ορισμός προφίλ περιλαμβάνεται και στις εργασίες διαφόρων ερευνητικών και αναπτυξιακών έργων. Το έργο DAIDALOS [21] του οποίου στόχος είναι «να δώσει στους κινητούς χρήστες αρραγή (seamless), διεισδυτική (pervasive) πρόσβαση σε περιεχόμενο και υπηρεσίες μέσω ετερογενών δικτύων, υποστηρίζοντας τις προτιμήσεις τους και την περίσταση (context)» είναι ένα χαρακτηριστικό παράδειγμα.

Ακολούθως, το Parlay/OSA [22] αποτελεί μια προσπάθεια να αναπτυχθεί ένα ανοιχτό API για ανάπτυξη υπηρεσιών και ασχολείται επίσης με διαχείριση προφίλ χρήστη.



Τέλος, το IST FP6 project MAGNET Beyond το οποίο εστιάζει σε user-centricity, personalization and personal networking, service ubiquity and federation, αποτελεί το έργο, σε τμήματα του οποίου μπορεί να εφαρμοστεί και να αξιολογηθεί η μεθοδολογία που προτείνεται στις επόμενες ενότητες. Συγκεκριμένα, θα παρουσιαστεί ένας πλήρης ορισμός ενός προφίλ χρήστη (user profile) με συγκεκριμένες προδιαγραφές, ενώ θα γίνει και αναφορά στη διαχείριση του προτεινόμενου προφίλ.

## **2.2. Σχεδιασμός ορισμού προφίλ κατά τις προδιαγραφές του ETSI**

Σύμφωνα με το ETSI [13], η ύπαρξη προφίλ χρήστη καθίσταται απαραίτητη όταν ένας χρήστης επιθυμεί να προσδώσει πιο προσωπικό χαρακτήρα στη χρήση μιας συσκευής ή υπηρεσίας. Ξεκινώντας από αυτό, το προφίλ χρήστη είναι μια καταγραφή προτιμήσεων, κανόνων, ρυθμίσεων και εν γένει πληροφορίες σχετικές με το χρήστη. Οι πληροφορίες αυτές αποθηκεύονται και μπορούν να αλλάζουν δυναμικά, ώστε να επιτυγχάνεται η κατάλληλη συμπεριφορά της συσκευής και των διάφορων υπηρεσιών, ανάλογα με την περίπτωση και σύμφωνα πάντα με τις ανάγκες του χρήστη.

Το προφίλ χρήστη θα πρέπει να περιέχει όλα εκείνα τα στοιχεία, τα οποία προσδιορίζουν τα χαρακτηριστικά, τις δυνατότητες, τις ανάγκες ενός χρήστη και, οπωσδήποτε, κάθε αλλαγή που συμβαίνει στην κατάστασή του (status). Το ζήτημα των προφίλ περιπλέκεται, αν λάβουμε υπόψη το γεγονός ότι η πλειονότητα των χρηστών επιλέγουν να έχουν πολλαπλά προφίλ, ώστε καθένα από τα επιμέρους προφίλ να αντιστοιχεί σε κάποιο «ρόλο» ή σε κάποια κατάσταση στη ζωή του χρήστη. Επομένως, οι διαφορετικές περιστάσεις στις οποίες βρίσκεται ένας χρήστης μπορούν να οδηγήσουν στη δημιουργία διαφορετικών προφίλ (π.χ. προφίλ γραφείου, προφίλ σπιτιού, προφίλ βραδινής εξόδου), κάτι που δημιουργεί περαιτέρω την ανάγκη υλοποίησης μηχανισμών που θα ενεργοποιούν τα κατάλληλα προφίλ ανάλογα με την περίπτωση.

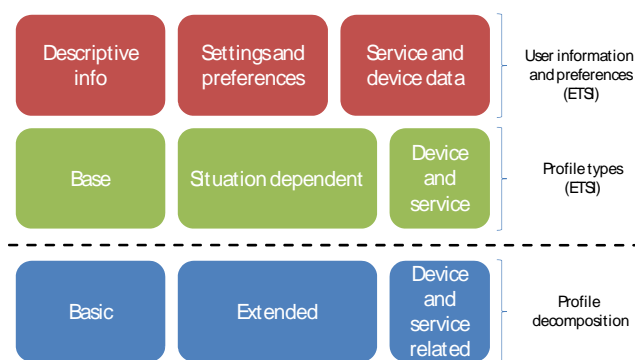
Ακολουθώντας την παραπάνω περιγραφή, στη συνέχεια θα γίνει απόπειρα να προσδιοριστούν οι διαφορετικές κατηγορίες προφίλ και να οριστούν templates για τη διαχείριση προφίλ. Οι ορισμοί που παρατίθενται είναι σύμφωνοι με τις οδηγίες του ETSI

για τον ορισμό, την εφαρμογή και τη διαχείριση προφίλ, όπως αυτές δίνονται στο έγγραφο “Human Factors (HF); User Profile Management”[13].

Σύμφωνα με το πρότυπο του ETSI, υπάρχουν τρεις τύποι προφίλ:

1. Το «**Βασικό Προφίλ**» (Base Profile), το οποίο περιέχει περιγραφικές πληροφορίες και ορισμένες κατηγορίες γενικών ρυθμίσεων και προτιμήσεων.
2. Το «**Προφίλ Συσκευής και Υπηρεσίας**» (Device and Service Profile), το οποίο σχετίζεται με μια συγκεκριμένη συσκευή και υπηρεσία και περιέχει τα αντίστοιχα δεδομένα.
3. Το «**Προφίλ Περιστασης**» (Situation Dependent Profile), το οποίο διαμορφώνεται ανάλογα με τις διαφορετικές περιστάσεις στις οποίες βρίσκεται ο χρήστης.

Βάσει αυτής της ανάλυσης, θα γίνει μια «αποσύνθεση» των πληροφοριών του προφίλ σε ομάδες χαρακτηριστικών, ώστε να γίνει εμφανής η συνολική δομή ενός προφίλ χρήστη (σχήμα 2.1).



**Σχήμα 2.1:** Ορισμός τύπων προφίλ κατά ETSI [13] και οι σχετιζόμενες πληροφορίες

Οι βασικές κατηγορίες πληροφοριών που συνεισφέρουν στη διαμόρφωση του προφίλ είναι:

- Οι *βασικές πληροφορίες χρήστη* (basic user information)
- Οι *προχωρημένες πληροφορίες χρήστη* (extended user information).
- Οι *πληροφορίες σχετικές με συσκευή και υπηρεσίες* (device and service related information).

Το περιεχόμενο κάθε κατηγορίας θα αναλυθεί στη συνέχεια.

## 2.3. Αποσύνθεση προφίλ

### 2.3.1. Βασικές πληροφορίες χρήστη

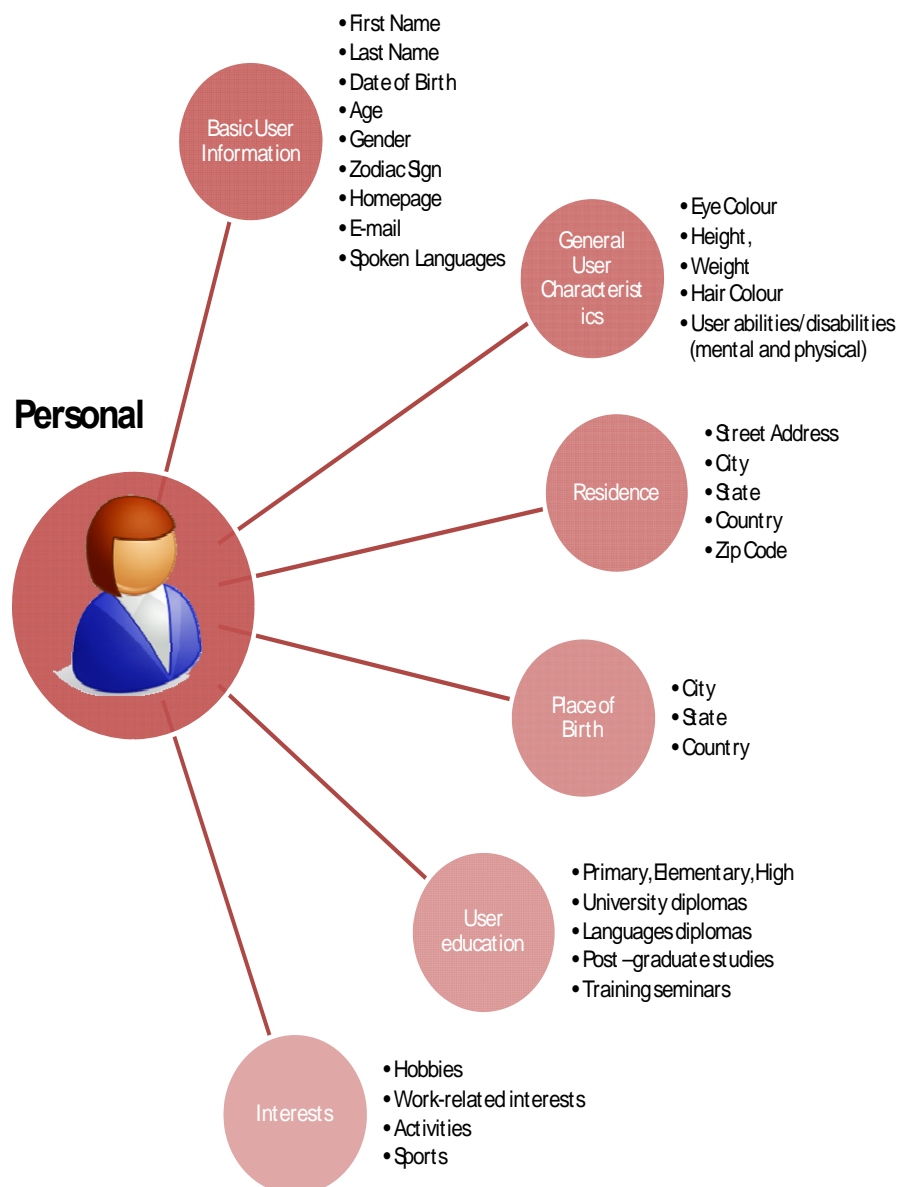
Οι βασικές πληροφορίες χρήστη συνίστανται σε μια ομάδα χαρακτηριστικών τα οποία περιγράφουν την προσωπικότητα ενός χρήστη και υπογραμμίζουν την ατομικότητά του. Τα χαρακτηριστικά αυτά μπορούν να θεωρηθούν σταθερά/μόνιμα, με την έννοια ότι δεν αλλάζουν συχνά και δεν επηρεάζονται από εξωτερικούς παράγοντες. Μπορούμε να διακρίνουμε τρεις κύριες υποκατηγορίες:

1. *Προσωπικές πληροφορίες χρήστη* (Personal user information), που βασίζονται σε προσωπικά στοιχεία του χρήστη.
2. *Επαγγελματικές πληροφορίες χρήστη* (Professional user information), που σχετίζονται με την επαγγελματική ταυτότητα του χρήστη.
3. *Πληροφορίες σχέσης χρήστη – συσκευής* (Comportment information), που σχετίζονται με τον τρόπο με τον οποίο ο χρήστης χρησιμοποιεί και αλληλεπιδρά με την κινητή του συσκευή.

Όσο αφορά στις προσωπικές και επαγγελματικές πληροφορίες χρήστη, μπορούν να διαιρεθούν σε ιδιωτικό και δημόσιο μέρος, όπου το πρώτο περιέχει πληροφορίες, στις οποίες δεν έχουν πρόσβαση (όλοι ή οι περισσότεροι) άλλοι χρήστες, και το δεύτερο περιέχει ευρέως διαθέσιμες (και ενδεχομένως κάποιες φορές «διαφημιζόμενες») πληροφορίες. Ωστόσο, αξίζει να σημειωθεί ότι η αντίληψη σχετικά με το τι είναι ιδιωτικό και τι δημόσιο δεν είναι ίδια για όλους τους χρήστες, ενώ μπορεί ακόμα και να αλλάζει ανάλογα με την περίπτωση. Πρέπει, επομένως, τα θέματα ιδιωτικότητας (privacy issues) να αντιμετωπίζονται με ιδιαίτερη προσοχή, καθώς εξαρτώνται σε μεγάλο βαθμό από την υποκειμενική θεώρηση του χρήστη. Σε επόμενες υποενότητες, εκφράζονται κάποιες ιδέες σχετικά με πιθανούς τρόπους αντιμετώπισης του ζητήματος αυτού.

#### 2.3.1.1. Προσωπικές πληροφορίες χρήστη

Οι προσωπικές πληροφορίες χρήστη συνίστανται σε όλα εκείνα τα στοιχεία τα οποία τον/την χαρακτηρίζουν ως άτομο [23]. Αυτές φαίνονται στο σχήμα 2.2:.



Σχήμα 2.2: Προσωπικές πληροφορίες χρήστη

Οι πληροφορίες μπορούν να ομαδοποιηθούν σε έξι κατηγορίες, ταυτοποιώντας πλήρως το χρήστη:

- *Βασικές πληροφορίες (Basic information)*, που περιέχουν όλες τις πληροφορίες που μπορούν να χρησιμοποιηθούν ως είσοδος σε φόρμες εγγραφής ή ερωτηματολόγια. Ενδεικτικά πεδία είναι τα: όνομα, επώνυμο, τόπος γέννησης, φύλο, κ.λπ.
- *Γενικά χαρακτηριστικά χρήστη (General User Characteristics)*, τα οποία περιέχουν πιο προσωπικές πληροφορίες, οι οποίες μπορούν να χρησιμοποιηθούν για να ταυτοποιηθούν ιδιαιτερότητες (π.χ. κάποια αναπηρία). Οι πληροφορίες

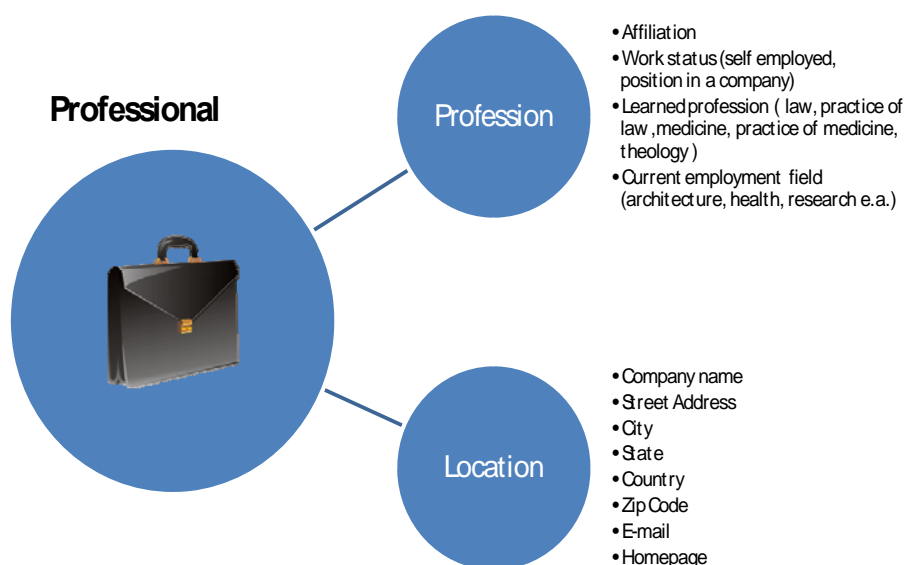
αυτές περιέχουν πεδία όπως το ύψος, το χρώμα ματιών, χρώμα μαλλιών, κ.α. και λόγω του ευαίσθητου της φύσης τους δεν είναι δημόσια διαθέσιμες, εκτός αν υπάρχει η συγκατάθεση του χρήστη.

- *Τόπος Κατοικίας (Residence)*, όπου περιέχονται πληροφορίες σχετικά με τον τόπο διαμονής, μαζί με διευθύνσεις και τρόπους επικοινωνίας. Οι πληροφορίες αυτές μπορούν να είναι ιδιωτικές ή δημόσιες, ανάλογα με την προτίμηση του χρήστη.
- *Τόπος γέννησης (Place of birth)*, όπου περιέχονται όλες οι πληροφορίες που χαρακτηρίζουν την εθνικότητα του χρήστη και την τοποθεσία γέννησής του. Οι πληροφορίες αυτές μπορεί να θεωρούνται από κάποιους χρήστες ιδιωτικές (π.χ. λόγω φόβου ρατσισμού), γι' αυτό και τις ξεχωρίζουμε από τις βασικές πληροφορίες.
- *Εκπαίδευση (User education)*, όπου περιέχονται όλες οι πληροφορίες σχετικά με το μορφωτικό επίπεδο του χρήστη, πτυχία και τυχόν επιπρόσθετη κατάρτιση, π.χ. σεμινάρια κ.λπ.
- *Ενδιαφέροντα (Interests)*, όπου περιέχονται πληροφορίες για χόμπι, δραστηριότητες, αθλητισμό, κ.α. Πληροφορίες τέτοιου τύπου είναι πολύ σημαντικές για υπηρεσίες, όπου γίνεται ταίριασμα των χρηστών ανάλογα με τα ενδιαφέροντά τους, ωστόσο επειδή μπορεί να χρησιμοποιηθούν για στοχευμένη διαφήμιση (targeted advertising), ο δημόσιος ή ιδιωτικός τους χαρακτήρας θα πρέπει να καθορίζεται από το χρήστη.

Εκτός των πληροφοριών που περιγράφονται στις έξι παραπάνω κατηγορίες, ενδεχομένως να υπάρχουν και άλλα στοιχεία που θα έπρεπε να περιληφθούν στο προφίλ χρήστη. Παράδειγμα αποτελούν οι θρησκευτικές πεποιθήσεις, οι οποίες σαφώς ανήκουν στη δεύτερη κατηγορία. Η παράλειψή τους έγινε σκόπιμα, ώστε να καταδειχθεί ότι ο ορισμός των συγκεκριμένων έξι κατηγοριών έγινε έτσι, ώστε κάθε επιπλέον στοιχείο να μπορεί να περιληφθεί σε μία από τις υπάρχουσες κατηγορίες.

### 2.3.1.2. Επαγγελματικές πληροφορίες χρήστη

Οι επαγγελματικές πληροφορίες χρήστη συνίστανται σε όλες εκείνες τις πληροφορίες που χαρακτηρίζουν το χρήστη στον επαγγελματικό χώρο. Οι πληροφορίες αυτές συμπληρώνουν τις προσωπικές πληροφορίες χρήστη, σε επαγγελματικά ζητήματα. Στο σχήμα 2.3 φαίνονται τα στοιχεία που αποτελούν τις επαγγελματικές πληροφορίες χρήστη.



**Σχήμα 2.3:** Επαγγελματικές πληροφορίες χρήστη

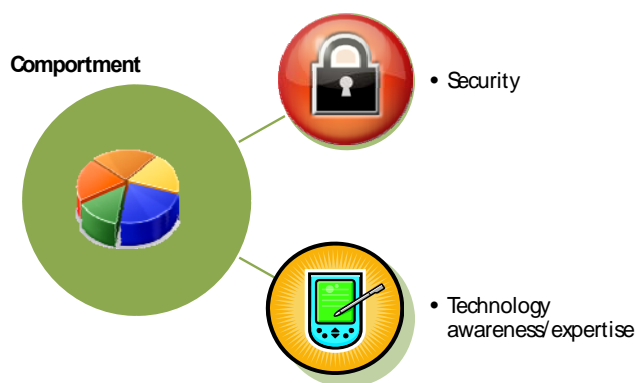
Οι επαγγελματικές πληροφορίες μπορούν να διακριθούν σε δύο κατηγορίες:

- *Επάγγελμα (Profession)*, όπου περιέχονται πληροφορίες σχετικές με την επαγγελματική κατάσταση του χρήστη, την ειδικότητά του, τις θέσεις στις οποίες έχει εργαστεί. Να σημειωθεί ότι περιέχονται πληροφορίες που περιγράφουν τις δεξιότητες και την τρέχουσα επαγγελματική κατάσταση του χρήστη, και όχι τρόπους επικοινωνίας (διευθύνσεις κ.λπ). Εδώ ο χρήστης επιλέγει ποιες πληροφορίες είναι ιδιωτικές και ποιες δημόσιες.
- *Τοποθεσία (Location)*, όπου περιέχονται όλες οι απαραίτητες πληροφορίες για επικοινωνία με το χρήστη σε επαγγελματικά πλαίσια. Οι πληροφορίες εδώ είναι γενικά δημόσιες και το γεγονός ότι είναι διαχωρισμένες από τις υπόλοιπες επαγγελματικές πληροφορίες συνεισφέρει στην ελαχιστοποίηση προβλημάτων spam (στοχευμένων διαφημίσεων προς συγκεκριμένες ειδικότητες, για παράδειγμα).

Αξίζει να σημειωθεί ότι κυρίως για την τοποθεσία (αλλά εν μέρει και για το επάγγελμα), μπορούν να υπάρχουν περισσότερα του ενός προφίλ για ένα χρήστη, εάν αυτός έχει περισσότερες από μία επαγγελματικές ταυτότητες. Σε αυτές τις περιπτώσεις ο χρήστης επιλέγει το κατάλληλο προφίλ ανάλογα με την κατάσταση.

### 2.3.1.3. Πληροφορίες σχέσης χρήστη – συσκευής

Στην κατηγορία αυτή περιέχονται τα στοιχεία που χαρακτηρίζουν προτιμήσεις, οι οποίες δε σχετίζονται με αυτό καθαυτό το χρήστη, αλλά με τη «στάση» του απέναντι στη χρήση της συσκευής. Διακρίνουμε δύο κύριες περιοχές: Ασφάλεια (security) και επίπεδο τεχνολογικής γνώσης/εξειδίκευσης (technology awareness/expertise).



**Σχήμα 2.4:** Πληροφορίες σχέσης χρήστη – συσκευής (comportment information)

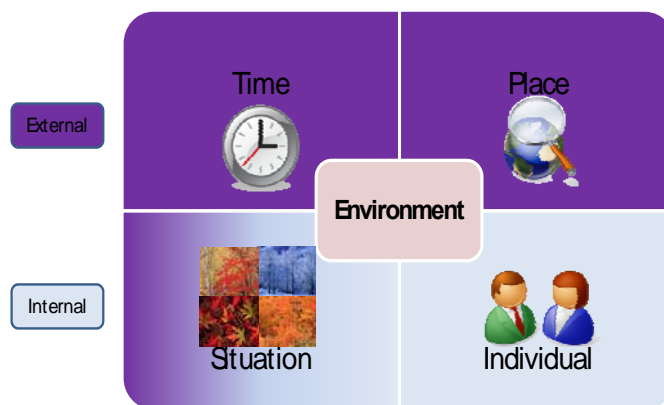
Οι παράμετροι αυτές περιγράφουν το πλαίσιο της αλληλεπίδρασης του χρήστη με τη συσκευή και δείχνουν το επίπεδο εμπιστοσύνης, κατανόησης τεχνικών θεμάτων και απαίτησης για έλεγχο. Έτσι, είναι δυνατό να καταγραφούν οι διαφορετικοί τύποι χρηστών και να προβλεφθούν οι στάσεις και οι ανάγκες τους. Προς το παρόν, δεν υπάρχει κάποιο πρότυπο ορισμού πληροφοριών comportment. Αυτό οφείλεται κυρίως στο ότι αυτό το τμήμα του προφίλ συνδέεται στενά με την προσφερόμενη υπηρεσία και το πλαίσιο στο οποίο αυτή προσφέρεται. Επιπρόσθετα, οι έννοιες της ιδιωτικότητας (privacy) και της τεχνολογικής (επί)γνωσης είναι άκρως υποκειμενικές και εφαρμογή τέτοιων χαρακτηρισμών θα επηρέαζε όλες τις υπόλοιπες πληροφορίες.

Επομένως, η σαφής διαμόρφωση του comportment profile πιθανόν να μην είναι δυνατή χωρίς να ληφθούν υπόψη η ασφάλεια / ιδιωτικότητα / τεχνολογική (επί)γνωση στις υπόλοιπες κατηγορίες του προφίλ. Ωστόσο, στην ενότητα αυτή δεν θα εστιάσουμε σε τέτοια θέματα.

### 2.3.2. Προχωρημένες πληροφορίες χρήστη

Οι προχωρημένες πληροφορίες χρήστη περιέχουν δεδομένα από τέσσερις ομάδες προτιμήσεων που διαδραματίζουν σημαντικό ρόλο σε όλες τις όψεις της ζωής του χρήστη. Επίσης, εξατομικεύουν την αλληλεπίδραση του χρήστη με τις συσκευές του και εκεί έγκειται κυρίως η σημασία τους. Οι προχωρημένες πληροφορίες χρήστη περιέχουν ορισμένες γενικές ρυθμίσεις βασισμένες στην ατομικότητα του χρήστη, οι οποίες δεν είναι μόνιμες και μπορούν να αλλάξουν ανάλογα με τις επιθυμίες και τις ανάγκες του χρήστη.

Αν σκεφτούμε από τι αποτελείται το περιβάλλον στο οποίο βρισκόμαστε κάθε χρονική στιγμή, μπορούμε να διακρίνουμε δύο παράγοντες: τους εξωτερικούς (external), που δεν μπορούμε να ελέγξουμε και τους εσωτερικούς (internal), την τιμή των οποίων μπορούμε να ελέγξουμε ή αποφασίσουμε. Το σχήμα 2.5 απεικονίζει τους δύο παράγοντες, καθώς και τη διάσπαση σε τέσσερις κατηγορίες, ήτοι χρόνο (time), χώρο (space), κατάσταση (situation) και άτομο (individual).



**Σχήμα 2.5:** Τα τέσσερα συστατικά στοιχεία του περιβάλλοντος του χρήστη

Ο χρόνος, ο χώρος και η κατάσταση (εν μέρει) θεωρούνται εξωτερικά, αφού ο χρήστης δεν μπορεί να αλλάξει αυτές τις παραμέτρους, ενώ το άτομο και η κατάσταση (εν μέρει) σχετίζονται με ρυθμίσεις του χρήστη.

Προσπαθώντας να μετατρέψουμε τα παραπάνω σε γενικές ρυθμίσεις και προτιμήσεις, έχουμε τα εξής:



- *Ρύθμιση/Προτίμηση Χρόνου (Time Setting/Preference)*. Οι ρυθμίσεις χρόνου αναφέρονται σε ταυτοποίηση χρονικών παραμέτρων για το παρόν και μπορούν να περιέχουν τα ακόλουθα για να περιγράψουν πλήρως τον παρόντα χρόνο:
  - Ώρα της ημέρας
  - Μέρα της εβδομάδας
  - Μήνας
  - Χρόνος
  - Ειδικές μέρες για τη συγκεκριμένη γεωγραφική περιοχή/χώρα
- *Ρύθμιση/Προτίμηση Τόπου (Location Setting/Preference)*. Οι ρυθμίσεις τόπου σχετίζονται με την περιοχή διαμονής του χρήστη και τις ιδιαιτερότητες που αυτή παρουσιάζει αναφορικά με τη χρήση της συσκευής και των προσφερόμενων υπηρεσιών. Για την πλήρη περιγραφή του τόπου, μπορούν να χρησιμοποιηθούν οι ακόλουθες παράμετροι:
  - Ακριβής γεωγραφική θέση
    - Γεωγραφικό πλάτος (latitude),
    - Γεωγραφικό μήκος (longitude),
    - Υψόμετρο (Altitude)
  - Προσεγγιστική γεωγραφική θέση
    - Σημείο αναφοράς (κέντρο),
    - Απόσταση από σημείο αναφοράς (ακτίνα),
    - Αναγνωριστικό σημείου πρόσβασης (access point ID)
- *Γενικές Ρυθμίσεις/Προτιμήσεις (Generic Setting/Preference)*. Αυτές αναφέρονται στις γενικές προτιμήσεις του χρήστη ανάλογα με τα χαρακτηριστικά, τις επιλογές και τις ανάγκες του [24] :
  - Προτιμήσεις στο φαγητό (π.χ. διατροφικοί περιορισμοί)
  - Αγαπημένα είδη ταινιών
  - Αγαπημένα είδη μουσικής
  - Προτιμήσεις σχετικά με θέματα υγείας (π.χ. ουσίες στις οποίες είναι αλλεργικός)
  - Ρυθμίσεις σχετικές με θρήσκευμα (π.χ. αργίες, νηστείες)
  - Προτιμήσεις τιμής (αν υπάρχουν πολλές διαθέσιμες συνδέσεις, ο χρήστης μπορεί να προτιμά την πιο οικονομική σύνδεση).
  - Προτιμήσεις σε μεθόδους πληρωμής.

### 2.3.3. Πληροφορίες σχετικές με συσκευή και υπηρεσίες

#### 2.3.3.1. Πληροφορίες σχετικές με τη συσκευή

Οι πληροφορίες σχετικές με συσκευή σχετίζονται με τις δυνατότητες της συσκευής του χρήστη. Επίσης εξατομικεύουν την αλληλεπίδραση χρήστη – συσκευής. Και σε αυτήν την περίπτωση ορίζονται οι βασικές πληροφορίες σχετικές με τη συσκευή και οι προχωρημένες πληροφορίες σχετικές με τη συσκευή, με αντίστοιχο τρόπο με αυτόν που ορίστηκαν ανωτέρω οι βασικές και προχωρημένες πληροφορίες χρήστη. Οι βασικές πληροφορίες σχετικές με τη συσκευή αποτελούνται από όλα τα χαρακτηριστικά που ταυτοποιούν τη συσκευή και δε σχετίζονται με παραμέτρους του περιβάλλοντος ή της περίπτωσης. Οι προχωρημένες πληροφορίες σχετικές με τη συσκευή (σχήμα 2.6) περιλαμβάνουν όλες τις παραμέτρους οι οποίες υπόκεινται σε αλλαγές ανάλογα με εξωτερικά ερεθίσματα ή/και κανόνες ή τρόπους λειτουργίας.



Σχήμα 2.6: Πληροφορίες σχετικές με τη συσκευή

Τα στοιχεία που περιέχονται εδώ μπορούν να ομαδοποιηθούν σε τρεις κατηγορίες, περιγράφοντας πλήρως τη συσκευή του χρήστη:

- *Εξοπλισμός υλικού (Hardware equipment)*, που περιέχει όλες τις πληροφορίες που σχετίζονται με τις δυνατότητες της συσκευής σε υλικό και σε διαθέσιμες διεπαφές

(interfaces). Οι δυνατότητες αυτές επηρεάζουν την εμπειρία του χρήστη, καθώς αποτελούν το πλαίσιο στο οποίο εκτελούνται οι υπηρεσίες.

- *Λογισμικό, υπηρεσίες και πρωτόκολλα (Software, services and protocols)*, που περιέχει πληροφορίες για το software, firmware και το λειτουργικό σύστημα της συσκευής, καθώς και τις υποστηριζόμενες εφαρμογές και τα σχετικά features. Αυτά τα στοιχεία επηρεάζουν την εμπειρία του χρήστη με πιο άμεσο τρόπο, καθώς παρέχουν το άμεσο interface με το χρήστη, μέσω των εφαρμογών που αυτός χρησιμοποιεί.
- *Σύνδεση δικτύου (Network communication)*, που καλύπτουν τις δυνατότητες συνδεσιμότητας της συσκευής, καθώς και την κατάσταση της τρέχουσας σύνδεσης.

#### 2.3.3.2. Πληροφορίες σχετικές με υπηρεσίες

Εδώ περιέχονται όλες οι πληροφορίες που σχετίζονται με την εκτέλεση και χρήση των προσφερόμενων υπηρεσιών. Ως τέτοιες μπορούμε να θεωρήσουμε όλες τις ρυθμίσεις που σχετίζονται με την τρέχουσα κατάσταση του περιβάλλοντος και με τις υπηρεσίες στις οποίες έχει πρόσβαση ο χρήστης. Τέτοιες πληροφορίες είναι οι ακόλουθες:

- Εκδηλώσεις/γεγονότα στην περιοχή
- Προσφερόμενες υπηρεσίες
- Ήχος/θόρυβος στο background
- Multi-tasking
- Καιρός
- Δυνατότητες/διαθεσιμότητα δικτύου

## **2.4. Διαχείριση προφίλ**

Σύμφωνα με το ETSI [13], για την αποτελεσματική διαχείριση των προφίλ χρήστη, είναι απαραίτητη η ανάπτυξη profile agents / entities (οντοτήτων προφίλ), οι οποίες θα παρέχουν αποθήκευση δεδομένων σχετικών με το προφίλ, επεξεργασία, προσαρμογή συμπεριφοράς, αλλά και ενεργοποίηση και απενεργοποίηση προφίλ ανάλογα με τις περιστάσεις. Η ανάγκη αυτή οδηγεί στον ορισμό τεσσάρων agents / entities, καθεμία εκ των οποίων είναι υπεύθυνη για την αποθήκευση (storage), επισκόπηση / τροποποίηση (viewing/editing), επεξεργασία (processing) και ενεργοποίηση (activation) των προφίλ.

Σε μια προσπάθεια να παράσχουμε υποστήριξη για το μοντέλο προφίλ χρήστη που παρουσιάζουμε εδώ, ορίζουμε δύο διαφορετικές οντότητες (entities), η καθεμία από τις οποίες ενσωματώνει τη λειτουργικότητα δύο agents: την οντότητα διαχείρισης προφίλ (profile handling entity) και την οντότητα εφαρμογής προφίλ (profile enforcement entity), οι οποίες περιγράφονται κατωτέρω.

### **2.4.1. Η οντότητα διαχείρισης προφίλ**

Η οντότητα αυτή είναι υπεύθυνη για τη διαχείριση του προφίλ χρήστη και για την εκτέλεση όλων των απαραίτητων ενεργειών για την αποθήκευση πληροφοριών σχετικών με το χρήστη και κανόνων ενεργοποίησης/απενεργοποίησης, καθώς και για την επισκόπηση και τροποποίηση του προφίλ. Επομένως, η οντότητα διαχείρισης προφίλ υλοποιεί τις λειτουργίες αποθήκευσης (storage) και επισκόπησης/τροποποίησης (viewing/editing), όπως περιγράφονται από το ETSI [13].

### **2.4.2. Η οντότητα εφαρμογής προφίλ**

Η οντότητα αυτή είναι υπεύθυνη για την εφαρμογή ενός προφίλ ανάλογα με την περίπτωση και τις προτιμήσεις του χρήστη. Ο ρόλος της έγκειται στην επεξεργασία των πληροφοριών που σχετίζονται με ρυθμίσεις χρήστη και περίστασης και την εφαρμογή του κατάλληλου προφίλ σύμφωνα με ένα σύνολο κανόνων. Για το σκοπό αυτό, η οντότητα αυτή λαμβάνει επίσης υπόψη εξωτερικές πληροφορίες (συνδεδεμένες συσκευές και στοιχεία για αυτές) και έχει πρόσβαση σε δεδομένα αποθηκευμένα έξω από τη

συσκευή του χρήστη. Επομένως, η οντότητα εφαρμογής προφίλ υλοποιεί τις λειτουργίες της επεξεργασίας (processing) και ενεργοποίησης (activation), όπως περιγράφονται από το ETSI [13].

#### **2.4.3. Εφαρμογή ενός προφίλ χρήστη**

Για να τεθεί σε ισχύ/εφαρμογή ένα συγκεκριμένο (ή περισσότερα του ενός) προφίλ, είναι αναγκαία η ύπαρξη μιας διαδικασίας που θα συνδυάζει τις πληροφορίες του προφίλ και θα επιτρέπει στην profile enforcement entity να αποφανθεί για το κατάλληλο προφίλ.

Για την εφαρμογή ενός προφίλ είναι απαραίτητη η συνεργασία του processing agent και του activation agent parts της profile enforcement entity. Και οι δύο agents βασίζονται σε ένα σύνολο κανόνων, οι οποίοι προκύπτουν από την επεξεργασία των παρακάτω πληροφοριών του προφίλ:

- Ρυθμίσεις ενεργοποίησης
- Ρυθμίσεις απενεργοποίησης
- Διάρκεια ζωής (lifetime)

Οι κανόνες ορίζονται ως ένα σύνολο από συνθήκες που αντιστοιχίζουν τις ρυθμίσεις ενεργοποίησης/απενεργοποίησης σε ένα σύνολο πιθανών ενεργειών.

Αναφορικά με το ποιες πληροφορίες πρέπει να συνυπολογιστούν, πρόκειται για ένα υποσύνολο όλων των πληροφοριών που περιέχονται στο προφίλ και σύμφωνα με τον ορισμό του ETSI [13] «περιλαμβάνουν ημερομηνία και ώρα, ομάδα, υπηρεσία, τερματικό, πληροφορίες παρουσίας και δραστηριότητας, ρυθμίσεις γλώσσας και τοπικές ρυθμίσεις, ρυθμίσεις προσβασιμότητας, προτιμήσεις επικοινωνίας, διάθεση, δίκτυο και διαχείριση ρυθμίσεων προφίλ άλλων χρηστών». Οι πληροφορίες αυτές βασίζονται σε βασικά δεδομένα του προφίλ, μαζί με μια παράμετρο διάρκειας ζωής.

Συγκεφαλαιώνοντας, οι πληροφορίες που ορίστηκαν ως μέρη του προφίλ χρήστη δίνουν μια κατά το δυνατό σφαιρική εικόνα της προσωπικής και επαγγελματικής ζωής του χρήστη, καθώς και των δυνατοτήτων της συσκευής του. Πολλές από τις παραμέτρους που ορίστηκαν αποτελούν απαραίτητες πληροφορίες για την παροχή εξατομικευμένων υπηρεσιών βασισμένων στη θέση. Στις επόμενες ενότητες, θα μελετήσουμε το ζήτημα της ιδιωτικότητας ορισμένων εκ των πληροφοριών αυτών (γεωγραφική θέση, προσωπικές προτιμήσεις σε διάφορους τομείς όπως ταινίες, μουσική, φαγητό, κ.α.), θα προτείνουμε ένα μοντέλο ποσοτικοποίησής της με βάση την εντροπία με στόχο την επίτευξη ισορροπίας ανάμεσα στην ποιότητα της παρεχόμενης υπηρεσίας και την επιθυμία για ιδιωτικότητα, και, τέλος, θα περιγράψουμε την υπό ανάπτυξη πλατφόρμα PLASMA, στην οποία θα εφαρμοστούν οι προτεινόμενες μεθοδολογίες.

### **3. ΙΔΙΩΤΙΚΟΤΗΤΑ ΠΡΟΣΩΠΙΚΩΝ ΔΕΔΟΜΕΝΩΝ ΣΕ ΕΞΑΤΟΜΙΚΕΥΜΕΝΕΣ ΥΠΗΡΕΣΙΕΣ**

#### **3.1. Γενικά**

Η παροχή εξατομικευμένων υπηρεσιών βασισμένων στη θέση (personalized, location-based services) προϋποθέτει τη γνωστοποίηση στον πάροχο (της υπηρεσίας) από το χρήστη διαφόρων πληροφοριών, οι οποίες θα χρησιμοποιηθούν ώστε να παρασχεθεί η υπηρεσία με εξατομικευμένο τρόπο. Οι πληροφορίες αυτές χωρίζονται σε δύο κατηγορίες:

- Πληροφορίες σχετικές με τη γεωγραφική θέση του χρήστη (location)
- Πληροφορίες σχετικές με τις προτιμήσεις του χρήστη (personal preferences).

Είναι προφανές ότι ο server πρέπει να γνωρίζει τη γεωγραφική θέση του χρήστη, ώστε να μπορεί να απαντήσει σε ερωτήματα του τύπου «Ποιο είναι το πιο κοντινό εστιατόριο;» ή «Ενημέρωσε με αν υπάρχει κυκλοφοριακή συμφόρηση στα επόμενα 10 λεπτά της διαδρομής μου». Αν, μάλιστα, προχωρήσουμε ακόμα ένα βήμα και πάμε σε μια ακόμα πιο εξατομικευμένη υπηρεσία (π.χ. ερώτημα του τύπου «Δείξε μου τα κοντινότερα bar που παίζουν κάποιο είδος μουσικής της αρεσκείας μου»), τότε γίνεται εμφανής και η ανάγκη ύπαρξης πληροφοριών σχετικά με τις προτιμήσεις του χρήστη (στο συγκεκριμένο παράδειγμα, π.χ., ποια είδη μουσικής αρέσουν στο χρήστη). Είναι, επομένως, σαφές ότι οι δύο αυτές ομάδες πληροφοριών αποτελούν το «θεμέλιο λίθο» κάθε personalized, location-based υπηρεσίας.

Από την άλλη πλευρά, ωστόσο, είναι αντιληπτό ότι και οι δύο αυτές κατηγορίες δεδομένων διακρίνονται από τον προσωπικό, ιδιωτικό τους χαρακτήρα. Με τη γνωστοποίηση λοιπόν τέτοιων πληροφοριών σε τρίτους ανακύπτουν ζητήματα διατήρησης / παραβίασης της ιδιωτικότητας (privacy). Η παράνομη παρακολούθηση του γεωγραφικού στίγματος του χρήστη αποτελεί σοβαρή παραβίαση της ιδιωτικότητάς του. Για παράδειγμα, ένας εργοδότης μπορεί να ελέγχει τον υπάλληλό του παρακολουθώντας πού ακριβώς βρίσκεται κάθε στιγμή της μέρας. Στις ΗΠΑ μάλιστα είχε απασχολήσει τα ΜΜΕ η ιστορία ενός άντρα που παρακολουθούσε μια γυναίκα με την οποία σχετιζόταν στο παρελθόν, λαμβάνοντας το σήμα του GPS της [25], [26]. Όσο αφορά στη δεύτερη

κατηγορία πληροφοριών, και εδώ προκύπτουν απειλές για την ιδιωτικότητα του χρήστη. Αποκάλυψη προσωπικών προτιμήσεων / ρυθμίσεων σε τρίτους μπορεί να οδηγήσει σε εκμετάλλευση των στοιχείων από αυτούς, παραβιάζοντας το ιδιωτικό των δεδομένων. Π.χ. διατροφικές συνήθειες ή περιορισμοί, ή κάποια δεδομένα που σχετίζονται με την κατάσταση της υγείας του χρήστη θα μπορούσαν να χρησιμοποιηθούν από ασφαλιστικές εταιρίες, ενώ πληροφορίες για τις καταναλωτικές προτιμήσεις του χρήστη, μπορούν να οδηγήσουν σε στοχευμένη διαφήμιση (targeted advertising) διαφόρων προϊόντων. Ο χρήστης, συνεπώς, «ανταλλάσσει», ως ένα βαθμό, ιδιωτικότητα με την ποιότητα της υπηρεσίας που θέλει να χρησιμοποιήσει. Προκύπτει, λοιπόν, το ζήτημα της προστασίας της ιδιωτικότητας και της αντιμετώπισης αυτών των απειλών παραβίασής της, διατηρώντας όμως ταυτόχρονα την ποιότητα στην υπηρεσία. Το ζήτημα αυτό απασχολεί την επιστημονική κοινότητα και έχουν προταθεί διάφορα μοντέλα, για ορισμένα από τα οποία θα μιλήσουμε στη συνέχεια.

### **3.2. Μοντέλα που έχουν προταθεί για την προστασία της ιδιωτικότητας**

Κατωτέρω παρατίθενται μοντέλα που έχουν προταθεί για την προστασία της ιδιωτικότητας προσωπικών πληροφοριών (κυρίως της θέσης):

#### **3.2.1. Εισαγωγή «θορύβου»**

Κατ' αρχήν, έχουν προταθεί διάφορες μέθοδοι για την προστασία της ιδιωτικότητας της γεωγραφικής θέσης (location privacy) του χρήστη, βασισμένες σε «θόλωμα» (blurring) ή εισαγωγή «θορύβου» [27]. Ένα χαρακτηριστικό παράδειγμα τέτοιων τεχνικών δίνεται στο [28], όπου οι Kido, Yanagisawa και Satoh προτείνουν τη μετάδοση, εκτός της πραγματικής θέσης, ψευδών (“dummy”) θέσεων στον πάροχο της υπηρεσίας, ώστε αυτός να επιστρέψει μια απάντηση για καθεμία από τις υποβληθείσες θέσεις, χωρίς να γνωρίζει ποια πραγματικά αντιστοιχεί στο χρήστη. Εντούτοις, η χρήση τεχνικών «θολώματος» ή εισαγωγής «θορύβου» επιβάλλει επιπλέον υπολογιστικό φόρτο στον πάροχο (πολλά περιττά δεδομένα). Σε περιπτώσεις κινητών συσκευών και ασύρματων συνδέσεων όμως, αυτή η απαίτηση για μεγάλο αριθμό επιπρόσθετων άχρηστων ουσιαστικά δεδομένων αυξάνει σημαντικά το κόστος, ενώ εισάγει πολύ μεγαλύτερες καθυστερήσεις στις προσπελάσεις στις βάσεις δεδομένων. Επιπρόσθετα, όταν ακολουθούνται τέτοιες



τεχνικές, πρέπει να ληφθεί μέριμνα για μια σειρά από θέματα, όπως, για παράδειγμα, στην περίπτωση πολλαπλών διαδοχικών ερωτήσεων, θα πρέπει η τομή των συνόλων των ψευδών θέσεων που αποστέλλονται κάθε φορά να μην είναι κενή ή τόσο μικρή, ώστε να είναι δυνατό να αποκαλυφθεί η πραγματική θέση του χρήστη.

### 3.2.2. **k-ανωνυμία (k-anonymity)**

Μια άλλη μέθοδος συνίσταται στην απόκρυψη της ακριβούς θέσης του χρήστη μέσω της «ανάμιξης» της πληροφορίας αυτής με αντίστοιχες πληροφορίες θέσης από άλλους χρήστες, με στόχο την επίτευξη ενός είδους «ανωνυμίας» (“anonymity”) [29]. Σύμφωνα με τους Pfitzmann και Koehntopp, ανωνυμία είναι εκείνη η κατάσταση στην οποία το υποκείμενο δεν μπορεί να ταυτοποιηθεί σε ένα σύνολο υποκειμένων, το σύνολο ανωνυμίας, δηλαδή όταν ένας τρίτος δεν μπορεί να ξεχωρίσει μεταξύ τους τα υποκείμενα που ανήκουν στο σύνολο ανωνυμίας (anonymity set) [30]. Στο [31] οι Gruteser and Grunwald προτείνουν έναν αλγόριθμο «συγκάλυψης» με βάση ένα region quad-tree (region quad-tree cloaking algorithm), προσπαθώντας να επιτύχουν **k-ανωνυμία (k-anonymity)**, (δηλαδή ανωνυμία, με πληθάρημο συνόλου ανωνυμίας ίσο με  $k$ ) από χωρική και χρονική άποψη. Ο αλγόριθμος κάνει μια αναδρομική διαίρεση των δεδομένων θέσης γύρω από το χρήστη, ωστόσο η δημιουργηθείσα περιοχή να περιέχει αριθμό χρηστών μικρότερο από  $k_{min}$  και, κατόπιν, χρησιμοποιεί το αμέσως προηγούμενο επίπεδο ως περιοχή συγκάλυψης (cloaking region). Αυτή η προσέγγιση, ωστόσο, δε λαμβάνει υπόψη θέματα ιδιωτικότητας σχετικά με τις προσωπικές προτιμήσεις του χρήστη, ενώ δεν υπάρχει και κάποιο «κάτω φράγμα» για την ποιότητα των επιστρεφόμενων αποτελεσμάτων. Επομένως, η περιοχή που απαιτείται για να επιτευχθεί ένα ικανοποιητικό επίπεδο ανωνυμίας μπορεί να είναι αρκετά μεγάλη, κυρίως στην περίπτωση όπου η παράμετρος  $k$  επιλέγεται από το χρήστη. Αυτό συμβαίνει επειδή οι απαιτούμενοι χρήστες με παρεμφερείς προτιμήσεις, οι οποίοι θα μπορούν να συγκροτήσουν ένα  $k$ -ανώνυμο σύνολο, ενδεχομένως να είναι διεσπαρμένοι σε μια πολύ μεγάλη περιοχή. Επιπλέον, η μέθοδος απαιτεί τη δημιουργία και εγκαθίδρυση ενός «εξυπηρετητή ανωνυμοποίησης» (anonymizing server), ο οποίος συγκεντρώνει πληροφορίες από πολλούς χρήστες και λειτουργεί ως proxy για τις αιτήσεις προς τον πάροχο της υπηρεσίας, εισάγοντας έτσι επιπρόσθετη πολυπλοκότητα και καθυστερήσεις στις πιθανές υλοποιήσεις συστημάτων που βασίζονται σε αυτήν την προσέγγιση.

Στο [32] προτείνεται από τους Gedik και Liu ο αλγόριθμος Clique Cloak. Ο αλγόριθμος λειτουργεί σε επίπεδο χρήστη (per-user basis), λαμβάνοντας υπόψη προσωπικές ρυθμίσεις ιδιωτικότητας και απαιτήσεις σε ποιότητα υπηρεσίας (Quality of Service – QoS). Όλες οι αιτήσεις των χρηστών «ανωνυμοποιούνται» με τη χρήση ενός μη κατευθυνόμενου γράφου, ο οποίος αποτελείται από αιτήματα χρηστών που δεν έχουν ανωνυμοποιηθεί ακόμα. Και σε αυτήν την περίπτωση χρησιμοποιείται ένας anonymizing server, ο οποίος «συγκαλύπτει» ένα αίτημα συνδυάζοντάς το με άλλα, ήδη υπάρχοντα αιτήματα στην ίδια περιοχή. Επομένως, η ύπαρξη παρόμοιων αιτημάτων αποτελεί προϋπόθεση για την εφαρμογή του αλγορίθμου. Η μέθοδος είναι αποτελεσματική για μικρό αριθμό χρηστών (περίπου 5, κάτι που θεωρείται ικανοποιητικό επίπεδο anonymity), ωστόσο η πολυπλοκότητα του αλγορίθμου είναι υψηλή, ενώ στις περιπτώσεις στις οποίες δεν είναι εφικτή η επίτευξη ανωνυμίας, τα αιτήματα των χρηστών ακυρώνονται.

Στο [33], ο Mokbel, Chow και Aref ακολουθούν το μοντέλο k-ανωνυμίας, σε μια απόπειρα να δημιουργήσουν ένα πλαίσιο (το αποκάλεσαν «Casper») το οποίο θα ανταποκρίνεται στις ανάγκες για ιδιωτικότητα (privacy) και ποιότητα (quality) σε ερωτήματα προς βάσεις δεδομένων σχετικά με location based υπηρεσίες. Το μοντέλο τους βασίζεται στη χρήση ενός location anonymizer («ανωνυμοποιητή θέσης») και ενός privacy-aware query processor («επεξεργαστή ερωτημάτων με αντίληψη ιδιωτικότητας») για την απόκρυψη πληροφοριών σχετικών με τη θέση του χρήστη. Μια και το μοντέλο Casper αποτελεί μια πολύ καλή προσπάθεια δημιουργίας ενός συστήματος που θα συνδυάζει ιδιωτικότητα και ποιότητα υπηρεσίας, κρίνεται σκόπιμο να προβούμε σε μια διεξοδικότερη περιγραφή του.

### 3.2.2.1. Περιγραφή συγκεκριμένης υλοποίησης: CASPER

Κάθε χρήστης που εγγράφεται στο Casper συντάσσει ένα «προφίλ ιδιωτικότητας» (privacy profile), το οποίο αναπαριστά το ύψος της απαίτησης του κάθε χρήστη για ιδιωτικότητα. Συγκεκριμένα, το προφίλ αυτό συνίσταται στη δυάδα τιμών  $(k, A_{\min})$ , όπου το  $k$  δείχνει ότι χρήστης επιθυμεί να είναι k-ανώνυμος (κατά την έννοια που αναλύθηκε παραπάνω, δηλαδή μη διαχωρίσιμος από k-1 άλλους χρήστες), ενώ το  $A_{\min}$  συμβολίζει την ελάχιστη επιφάνεια της περιοχής που ο χρήστης αποδέχεται να αναφερθεί ως η γεωγραφική του θέση και είναι ιδιαίτερα χρήσιμο σε περιοχές με μεγάλη πυκνότητα

χρηστών, όπου ακόμα και ένα μεγάλο  $k$  πιθανόν να μην παρέχει ικανοποιητική προστασία της ιδιωτικότητας. Ο χρήστης έχει τη δυνατότητα να αλλάξει το προφίλ αυτό όποτε επιθυμεί.

Το σύστημα Casper αποτελείται από δύο κύρια στοιχεία: τον location anonymizer και τον privacy-aware query processor. Ο πρώτος λαμβάνει συνεχώς στίγματα γεωγραφικών θέσεων από χρήστες κινητών συσκευών, «θολώνει» τις θέσεις τους ανάλογα με το προφίλ του καθενός και στέλνει τις θολωμένες-συγκαλυμμένες (cloaked) θέσεις στον εξυπηρετητή – βάση δεδομένων της υπηρεσίας (location-based database server). Ταυτόχρονα, διαγράφει οποιοδήποτε αναγνωριστικό ταυτότητας του χρήστη για να εξασφαλίσει την ψευδωνυμία (pseudonymity) της πληροφορίας θέσης [30]. Παρόμοια, ο location anonymizer «θολώνει» την πληροφορία θέσης και των ερωτημάτων (queries) προτού τα στείλει στη βάση δεδομένων.

Ο privacy-aware query processor είναι ενσωματωμένος στον location-based database server ώστε η επεξεργασία των ερωτημάτων να γίνεται σε επίπεδο ευρύτερων ή «θολωμένων» περιοχών και όχι ακριβών θέσεων. Αντί να επιστρέφει μια ακριβή απάντηση, ο privacy-aware query processor επιστρέφει μια λίστα με υποψήφιες απαντήσεις στο ερώτημα μέσω του location anonymizer. Οι χρήστες τοπικά εξετάζουν τη λίστα και βάσει του ερωτήματος που έχουν θέσει, βρίσκουν την απάντηση. Ο privacy-aware query processor εγγυάται ότι η λίστα έχει το ελάχιστο μέγεθος και ότι περιέχει την ακριβή απάντηση στο ερώτημα. Το μέγεθος της λίστας εξαρτάται σε μεγάλο βαθμό από το προφίλ ιδιωτικότητας του χρήστη. Ένα αυστηρό σε απαιτήσεις προφίλ οδηγεί σε εκτενή λίστα. Ρυθμίζοντας το προφίλ τους, οι χρήστες μπορούν να προσδιορίσουν το ισοζύγιο ιδιωτικότητας έναντι ποιότητα υπηρεσίας. Ερωτήματα βασισμένα στη θέση (location-based queries) λαμβάνονται τόσο από κινητούς χρήστες όσο και από δημόσιους διαχειριστές (public administrators). Ερωτήματα προερχόμενα από χρήστες θεωρούνται ιδιωτικά ερωτήματα και πρέπει να περάσουν από τον location anonymizer ώστε να κρυφτεί η ταυτότητα προέλευσης του ερωτήματος και να «θολωθεί» η θέση του χρήστη που υποβάλλει το ερώτημα. Ερωτήματα προερχόμενα από δημόσιους διαχειριστές θεωρούνται δημόσια ερωτήματα και δε χρειάζεται να περάσουν από τον location anonymizer, αλλά προωθούνται απευθείας στον location-based database server. Ο server αυτός απαντά σε τέτοια δημόσια ερωτήματα με βάση αποθηκευμένες «θολωμένες» πληροφορίες προερχόμενες από το σύνολο των χρηστών.

### Περιγραφή του location anonymizer

Η βασική ιδέα είναι να χρησιμοποιηθεί μια δομή με μορφή πυραμίδας με πλέγματα (grid-based), η οποία κατακερματίζει το χώρο σε  $H$  το πλήθος επίπεδα, όπου το πλέγμα ενός επιπέδου  $h$  χωρίζεται σε  $4^h$  κελιά (cells). Η κορυφή της πυραμίδας έχει ύψος μηδέν και αποτελείται από ένα μόνο κελί, το οποίο καλύπτει ολόκληρο το χώρο. Κάθε κελί στην πυραμίδα αναπαρίσταται από ένα ζεύγος (cid, N), όπου cid είναι το αναγνωριστικό του κελιού και N είναι ο αριθμός των χρηστών που υπάρχουν εντός των ορίων της περιοχής στην οποία αντιστοιχεί το κελί. Η δομή της πυραμίδας ενημερώνεται κατάλληλα ώστε να διατηρείται πάντοτε ο τρέχων αριθμός χρηστών σε κάθε κελί, και να αποθηκεύεται το cid του κελιού στο οποίο βρίσκεται κάθε χρήστης (στο χαμηλότερο επίπεδο της πυραμίδας). Ο αλγόριθμος που προτείνεται από τους συγγραφείς για το «θόλωμα» της θέσης του χρήστη και τον προσδιορισμό της περιοχής που πληροί τα κριτήρια ( $k, A_{\min}$ ) του προφίλ του κάθε χρήστη παρατίθεται παρακάτω:

#### Bottom-up cloaking algorithm

```

1: Function BottomUp-Cloaking( $k, A_{\min}, cid$ )
2: if  $cid.N \geq k$  and  $cid.Area \geq A_{\min}$  then
3:   return Area( $cid$ );
4: end if
5:  $cidV \leftarrow$  The vertical neighbor cell of  $cid$ .
6:  $cidH \leftarrow$  The horizontal neighbor cell of  $cid$ .
7:  $NV = cid.N + cidV.N$ ,  $NH = cid.N + cidH.N$ 
8: if ( $NV \geq k$  OR  $NH \geq k$ ) AND  $2cid.Area \geq A_{\min}$  then
9:   if ( $NH \geq k$  AND  $NV \geq k$  AND  $NH \leq NV$ ) OR  $NV < k$  then
10:    return Area( $cid$ )  $\square$  Area( $cidH$ );
11:   else
12:    return Area( $cid$ )  $\square$  Area( $cidV$ );
13:   end if
14: else
15:   BottomUp-Cloaking( $(k, A_{\min}), Parent(cid)$ );
16: end if

```

Τα ορίσματα που παίρνει ως είσοδο ο αλγόριθμος είναι το προφίλ ιδιωτικότητας του χρήστη ( $k, A_{\min}$ ) και το κελί cid που αναπαριστά τη θέση του. Ο αλγόριθμος ελέγχει εάν πληρούνται τα κριτήρια  $k$ -ανωνυμίας (τουλάχιστον  $k$  συνολικά χρήστες στο κελί) και ελάχιστης επιφάνειας  $A_{\min}$  (επιφάνεια κελιού μεγαλύτερη ή ίση με  $A_{\min}$ ). Εάν ναι, επιστρέφει το τρέχον κελί, αλλιώς ελέγχει αν ο συνδυασμός του κελιού με κάποιο από τα γειτονικά του κελιά πληροί τα κριτήρια και αν ναι, επιστρέφει αυτόν σαν αποτέλεσμα. Εάν και πάλι δεν πληρούνται τα κριτήρια, ο αλγόριθμος εκτελείται αναδρομικά για το

κελί-γονέα (parent cell) του τρέχοντος κελιού (ανεβαίνει, δηλαδή, ένα επίπεδο στην πυραμίδα). Η διαδικασία συνεχίζεται μέχρι να βρεθεί το κατάλληλο κελί.

#### Περιγραφή privacy-aware query processor

Ο κύριος σκοπός του privacy-aware query processor είναι να παράσχει ακριβείς, αποτελεσματικές και «ανώνυμες» υπηρεσίες βασιζόμενος στις «συγκεκριμένες»-«θολωμένες» (cloaked) περιοχές που προέρχονται από τον location anonymizer, αντί σε ακριβείς θέσεις. Στον location-based database server αποθηκεύονται δύο τύποι πληροφοριών: δημόσιες (π.χ. νοσοκομεία, εστιατόρια, βενζινάδικα), οι οποίες αποθηκεύονται ως έχουν, και ιδιωτικές (θέσεις χρηστών), οι οποίες αποθηκεύονται ως cloaked regions. Οι Mokbel, Chow και Aref διακρίνουν τρεις διαφορετικές κατηγορίες ερωτημάτων που μπορούν να γίνουν στον privacy-aware query processor:

- ιδιωτικά ερωτήματα για δημόσιες πληροφορίες, π.χ. ένας χρήστης ρωτάει πού βρίσκεται το κοντινότερο σε αυτόν βενζινάδικο
- δημόσια ερωτήματα για ιδιωτικές πληροφορίες, π.χ. ένας administrator ρωτάει ποιος είναι ο αριθμός των χρηστών σε μια συγκεκριμένη περιοχή
- ιδιωτικά ερωτήματα για ιδιωτικές πληροφορίες, π.χ. ένας χρήστης ρωτάει ποιος από τους φίλους του βρίσκεται πιο κοντά μια συγκεκριμένη χρονική στιγμή.

Οι συγγραφείς εστιάζουν στην πρώτη και στην τρίτη κατηγορία, θεωρώντας τη δεύτερη ειδική περίπτωση της τρίτης, και μάλιστα σε ερωτήματα πλησιέστερου γείτονα (nearest neighbor). Η βασική ιδέα συνίσταται στην επιλογή κάποιων αντικειμένων-στόχων (target objects), με τη βοήθεια των οποίων προσδιορίζεται ένας χώρος αναζήτησης  $A_{EXT}$ , ο οποίος καλύπτει όλες τις πιθανές περιοχές που ενδεχομένως περιέχουν μια απάντηση στο ερώτημα. Τελικά, όλα τα αντικείμενα-στόχοι επιστρέφονται στον πελάτη ως μια λίστα υποψήφιων απαντήσεων, η οποία όπως αποδεικνύουν έχει το ελάχιστο μέγεθος και περιέχει την ακριβή απάντηση στο ερώτημα.

Σε μια απόπειρα αξιολόγησης του μοντέλου Casper μπορούμε κατ' αρχήν να πούμε ότι παρέχει ένα πολύ καλό πλαίσιο για την αποτελεσματική απόκρυψη της θέσης του χρήστη. Ωστόσο η προσέγγιση παρουσιάζει ορισμένα μειονεκτήματα. Πρώτα απ' όλα, όπως και άλλες τεχνικές που αναφέρθηκαν ανωτέρω, και η μέθοδος αυτή στηρίζεται στην

περιοδική συλλογή πληροφοριών από άλλους χρήστες, και άρα προϋποθέτει συνεχή επικοινωνία για αναφορά θέσεων από αυτούς. Ακολούθως, από την περιγραφή της αρχιτεκτονικής προκύπτει ότι το μοντέλο απαιτεί την ύπαρξη ειδικής υποδομής (location anonymizer, privacy-aware query processor), γεγονός που ενδεχομένως καθιστά το σύστημα τρωτό σε επιθέσεις. Επιπλέον, αναφορικά με την ποιότητα των επιστρεφόμενων αποτελεσμάτων, τα κριτήρια βάσει των οποίων επιλέγεται το κατάλληλο επίπεδο αφαίρεσης / αοριστίας για την αναφορά θέσης δε λαμβάνουν υπόψη τον αριθμό των επιστρεφόμενων αποτελεσμάτων, κάτι που είναι σημαντικό για την αποτελεσματικότητα της υπηρεσίας.

### 3.2.3. Άλλες μέθοδοι

Άλλες μέθοδοι βασίζονται στη χρήση επιπέδων αφαίρεσης / αοριστίας για την αναφορά της θέσης του χρήστη. Η αφαίρεση αυτή ισοδυναμεί με την αύξηση της εντροπίας στην αναφορά γεωγραφικής θέσης [34]. Όσο υψηλότερο είναι το επίπεδο της εντροπίας, τόσο μεγαλύτερη προστασία της ιδιωτικότητας του χρήστη επιτυγχάνεται. Για το σκοπό αυτό, πολλές προτάσεις συμπεριλαμβάνουν της ανάπτυξη μιας κεντρικής οντότητας, π.χ. ενός anonymizing server. Ο χρήστης αναφέρει την πραγματική ακριβή του θέση σε αυτήν την οντότητα, η οποία στη συνέχεια «κατασκευάζει» μια ευρύτερη περιοχή που περιέχει την ακριβή θέση, αυξάνοντας έτσι την εντροπία γεωγραφικής θέσης του χρήστη. Και οι τεχνικές αυτές, ωστόσο παρουσιάζουν τα μειονεκτήματα που έχουν προαναφερθεί: την εξάρτηση, δηλαδή, της αποτελεσματικότητας από την ύπαρξη ή μη επαρκούς αριθμού άλλων χρηστών, καθώς και των καθυστερήσεων που εισάγει η επικοινωνία με τους χρήστες αυτούς και την επεξεργασία των πληροφοριών τους, αλλά και την ανάγκη ύπαρξης υποδομής (anonymizing server), η οποία μπορεί να αποτελέσει πηγή αστοχίας (failure) του συστήματος, σε περιπτώσεις κακής χρήσης ή ανεπαρκών μέτρων ασφάλειας.

## 4. ΜΑΘΗΜΑΤΙΚΗ ΠΕΡΙΓΡΑΦΗ ΠΟΣΟΤΙΚΟΠΟΙΗΣΗΣ ΤΗΣ ΙΔΙΩΤΙΚΟΤΗΤΑΣ ΜΕ ΒΑΣΗ ΤΗΝ ΕΝΤΡΟΠΙΑ

### 4.1. Εισαγωγή

Είναι σαφές ότι για την παροχή εξατομικευμένων υπηρεσιών βασισμένων στη θέση (personalized, location-based services) από κάποιον εξυπηρετητή είναι απαραίτητο ο τελευταίος να γνωρίζει τη θέση του χρήστη, καθώς και να διαθέτει πληροφορίες για τις προσωπικές του προτιμήσεις, ώστε να μπορεί να του παράσχει ακριβείς, διεξοδικές και εξατομικευμένες πληροφορίες, να μπορεί δηλαδή να εγγυηθεί την *ποιότητα* (quality) αυτών των υπηρεσιών. Από την άλλη μεριά, η μετάδοση από το χρήστη αναλυτικών πληροφοριών σχετικών με την ακριβή του θέση και τις προσωπικές του προτιμήσεις εγείρει ζητήματα **ιδιωτικότητας (privacy)**, όπως έχει αναφερθεί σε προηγούμενη ενότητα. Επομένως, υπάρχουν δύο αντικρουόμενες τάσεις: ποιότητα και ιδιωτικότητα. Πρόκειται για δύο πλευρές των οποίων η σημασία για τον τελικό χρήστη είναι αδιαμφισβήτητη.

Σκοπός εδώ είναι η δημιουργία ενός πλαισίου, στο οποίο ο χρήστης θα κρίνει πόσο θα συμβάλει στο σύστημα καθεμία από αυτές τις δύο αντίρροπες δυνάμεις (ποιότητα και ιδιωτικότητα), οι οποίες οδηγούν την παρεχόμενη υπηρεσία σε μια κατάσταση «ισορροπίας», ανάλογα με τη σημασία που αποδίδει ο ίδιος σε καθεμία από τις δύο αυτές πλευρές.

Πριν προχωρήσουμε στην εξέταση της σχέσης μεταξύ ιδιωτικότητας και ποιότητας με στόχο την επίτευξη του βέλτιστου αποτελέσματος, πρέπει πρώτα να βρούμε έναν τρόπο συστηματικής περιγραφής – ποσοτικοποίησης της έννοιας της ιδιωτικότητας. Σύμφωνα με τον Westin [35], η ιδιωτικότητα των προσωπικών πληροφοριών μπορεί να οριστεί ως «το αίτημα ατόμων ή ομάδων να προσδιορίζουν οι ίδιοι πότε, πώς και σε ποιο βαθμό θα κοινοποιούνται προσωπικές τους πληροφορίες σε τρίτους». Μέχρι σήμερα, έχουν γίνει ορισμένες προσπάθειες στην κατεύθυνση της συστηματικής περιγραφής της ιδιωτικότητας, όπως έχει αναλυθεί σε προηγούμενη ενότητα. Σε όλες τις περιπτώσεις πάντως, το ζήτημα της ιδιωτικότητας προσεγγίζεται από δύο πλευρές: αυτήν του εντοπισμού της γεωγραφικής θέσης του χρήστη και εκείνη των προσωπικών

προτιμήσεων και ρυθμίσεων. Στην ακόλουθη προσέγγιση που προτείνουμε, επιχειρούμε την ενσωμάτωση και των δύο πλευρών σε μια παράμετρο που θα σχετίζεται με την ιδιωτικότητα. Επομένως, η ιδιωτικότητα του χρήστη θα αντιμετωπισθεί ακολούθως σαν ένα ενιαίο ζήτημα γεωγραφικής θέσης – προτιμήσεων, ενώ θα προταθεί και ένας μηχανισμός απόφασης σχετικά με το για ποια από τις δύο πλευρές θα πρέπει να δίνονται πιο συγκεκριμένες πληροφορίες ώστε να βελτιώνονται τα παρεχόμενα αποτελέσματα.

## 4.2. Το προτεινόμενο μοντέλο για την ποσοτικοποίηση της ιδιωτικότητας

### 4.2.1. Χρήση της εντροπίας για τη μέτρηση της ιδιωτικότητας

Προσπαθώντας να βρούμε την κατάλληλη μέθοδο για να περιγράψουμε μαθηματικά το επίπεδο της ιδιωτικότητας με έναν ενιαίο τρόπο, καταλήξαμε στη χρήση του μεγέθους της **εντροπίας** (**entropy** -  $H$ ), ως μέτρου της αοριστίας, και άρα δυσκολίας στην ταυτοποίηση των προσωπικών ρυθμίσεων και γεωγραφικών πληροφοριών του χρήστη. Ο Shannon ανέπτυξε τη Μαθηματική Θεωρία της Επικοινωνίας (The Mathematical Theory of Communication) [36] σε μια προσπάθεια να ποσοτικοποιήσει (να περιγράψει μαθηματικά) την αβεβαιότητα (ή τυχαιότητα) μιας πηγής πληροφορίας (information source). Δεδομένου ότι:

- σκοπός μας είναι να ποσοτικοποιήσουμε την αβεβαιότητα (της γεωγραφικής θέσης του χρήστη, αλλά και των προσωπικών του προτιμήσεων),
- ιδιότητες της μετρικής που αναζητούμε παραπέμπουν σε μεγάλο βαθμό σε κάποιες από τις ιδιότητες που ο Shannon θεωρούσε απαραίτητες σε μια μετρική της αβεβαιότητας της πληροφορίας και που διαθέτει η εντροπία, όπως για παράδειγμα:
  - (i) με ισοπίθανες επιλογές, υπάρχει μεγαλύτερη αβεβαιότητα όταν υπάρχει μεγαλύτερος αριθμός επιλογών (μονοτονία)
  - (ii) αν ένα σύνολο ορίζεται ως συνδυασμός δύο ή περισσότερων ξένων υποσυνόλων, η εντροπία  $H$  του νέου συνόλου πρέπει να ισούται με το σταθμισμένο μέσο των τιμών της εντροπίας  $H$  των υποσυνόλων.



Δηλαδή για ένα σύνολο  $S_c$  που συντίθεται από δύο υποσύνολα  $S_a$  και  $S_b$ , θα ισχύει:

$$H(S_c) = H(a,b) + aH(S_a) + bH(S_b),$$

όπου  $a = |S_a|/|S_c|$  και  $b = |S_b|/|S_c|$  (αναδρομικότητα),

- η πληροφοριακή εντροπία (information entropy) χρησιμοποιείται σε πολλά επιστημονικά πεδία, όπως: στην οικολογία για να προσδιορίσει την ποικιλότητα των ειδών [37], [38], [39], στην κοινωνιολογία για τη μελέτη της κοινωνικής εξέλιξης [40], στην ταξινομική για την αξιολόγηση μεθόδων κατηγοριοποίησης (classification) [41], [42], αλλά και στη ρομποτική για τη μέτρηση της ποικιλότητας συστημάτων πολλών πρακτόρων (multiagent system diversity) [43],

η υιοθέτηση του μοντέλου πληροφοριακής εντροπίας του Shannon μπορεί να μας παράσχει τη βάση για το μοντέλο μας.

Στη συνέχεια θα παρουσιαστεί το απαραίτητο μαθηματικό υπόβαθρο στο οποίο βασίζεται η μεθοδολογία μας.

Η κατά Shannon πληροφοριακή εντροπία ενός συστήματος  $X$  δίνεται από τον τύπο:

$$H(X) = -K \sum_{j=1}^M p_j \log(p_j) \quad (4.1)$$

όπου:

- $M$  είναι ο αριθμός των πιθανών τιμών / καταστάσεων / υποσυνόλων του συστήματος υπό εξέταση,
- $p_j$  είναι η πιθανότητα μια μεταβλητή του συστήματος να πάρει την  $j$ -στη τιμή (να βρεθεί στην  $j$ -στη κατάσταση / να ανήκει στο  $j$ -στο υποσύνολο, κ.λπ), με  $\sum_{j=1}^M p_j = 1$ ,
- $K$  είναι μια σταθερά που αντιστοιχεί στην επιλογή μονάδων μέτρησης (ο Shannon θέτει  $K=1$  κάτι που αρχικά θα κάνουμε και εμείς, ωστόσο αργότερα θα συζητηθεί η χρήση του  $K$  στο ρύθμιση παραμέτρων του μοντέλου, ώστε να γίνεται πιο «ευαίσθητο» σε κάποιες προσωπικές παραμέτρους)

Θα εφαρμόσουμε τώρα το μοντέλο της εντροπίας και στις δύο παραμέτρους που ταυτόχρονα αποτελούν θεμέλιο λίθο κάθε εξατομικευμένης υπηρεσίας για κινητές

συσκευές και απειλή για την ιδιωτικότητα των δεδομένων του χρήστη: γεωγραφική θέση και προσωπικές προτιμήσεις. Με αυτόν τον τρόπο, αυτές οι δύο φαινομενικά διαφορετικές παράμετροι θα τοποθετηθούν σε κοινή βάση από άποψη συνεισφοράς στην απειλή της ιδιωτικότητας του χρήστη. Θα καταστεί, επομένως, δυνατή η αντιμετώπιση με ενιαίο τρόπο των επιδράσεων των δύο αυτών πλευρών στην ιδιωτικότητα του χρήστη μέσω της χρήσης μίας μοναδικής αριθμητικής παραμέτρου και, στη συνέχεια, η χρήση της παραμέτρου αυτής για τη βελτίωση της ποιότητας της παρεχόμενης εξατομικευμένης υπηρεσίας με ταυτόχρονο έλεγχο της έκτασης του κινδύνου (ως προς την ιδιωτικότητα) στον οποίο εκτίθεται ο χρήστης.

#### 4.2.2. Εντροπία εντοπισμού θέσης

Κάθε κινητή υπηρεσία βασισμένη στη θέση χρησιμοποιεί απαραίτητως ένα σύστημα χαρτών, όπως για παράδειγμα τα Windows Live Local (a.k.a. MSN Virtual Earth) [44], Google Maps [45], Yahoo! Maps [46], κ.α. Τα περισσότερα από αυτά τα συστήματα βασίζονται στη χρήση της προβολής Mercator (Mercator projection) [47]. Το σημαντικότερο χαρακτηριστικό των συστημάτων αυτών συνίσταται στην έμφυτη υποστήριξη διαφορετικών επιπέδων αφαίρεσης / ακρίβειας. Στα συστήματα που είναι οργανωμένα σε ιεραρχική δομή με τη χρήση tiles (τετραγωνίδια, «πλακίδια») όπως τα Virtual Earth [48] και Google Maps [45], υποστηρίζεται ένας αριθμός επιπέδων λεπτομέρειας (συνήθως 20-23). Σε κάθε επίπεδο, ο παγκόσμιος χάρτης χωρίζεται σε tiles με συγκεκριμένο μέγεθος σε pixels, και κάθε tile έχει ένα αναγνωριστικό (ID), π.χ. στο επίπεδο 1, υπάρχουν 4 tiles, με IDs για παράδειγμα 0, 1, 2, 3. Στο επόμενο επίπεδο, η περιοχή του χάρτη που αντιστοιχεί σε κάθε tile διασπάται περαιτέρω σε tiles (και πάλι συνήθως 4), π.χ. το tile 2 του επιπέδου 1 χωρίζεται στο επίπεδο 2 στα tiles με IDs: 20, 21, 22, 23. Αυτή η διαδικασία συνεχίζεται για όλα τα επίπεδα, οδηγώντας στη δημιουργία μιας ιεραρχικής δομής (στο παράδειγμά μας, μιας ιεραρχικής δενδρικής δομής τύπου quad-tree), όπου η ακρίβεια θέσης αυξάνεται (και, επομένως, η ιδιωτικότητα μειώνεται), όσο το επίπεδο λεπτομέρειας γίνεται μεγαλύτερο. Για παράδειγμα, εάν ένας χρήστης βρίσκεται σε μια περιοχή που αναπαρίσταται από το tile 12202102331213 (στο 14<sup>ο</sup> επίπεδο λεπτομέρειας), τότε ως θέση του θα μπορούσε να αναφέρει εναλλακτικά το tile 1220210233121 (13<sup>ο</sup> επιπέδου) ή το 122021023312 (12<sup>ο</sup> επιπέδου), με την ακρίβεια να μειώνεται και την αοριστία να αυξάνεται κάθε φορά. Αξίζει να σημειωθεί ότι η προτεινόμενη μεθοδολογία είναι ανεξάρτητη από το γεωγραφικό σύστημα

αναπαράστασης που επιλέγεται, αρκεί να βασίζεται σε μια ιεραρχική δομή, (π.χ. ένα region quad-tree [49], [50]) για να οργανώσει το χώρο.

Θα εφαρμόσουμε τώρα το μοντέλο εντροπίας του Shannon (1) στην περίπτωση ενός συστήματος γεωγραφικής θέσης (geographical location system) βασισμένου σε κάποια ιεραρχική δομή. Θα υποθέσουμε ότι για τον εντοπισμό της θέσης του χρήστη, θα χρησιμοποιηθεί το Global Positioning system (GPS), αφού αυτό είναι το ακριβέστερο σύστημα εντοπισμού παγκοσμίως. Ας συμβολίσουμε με  $s_{GPS}$  το εμβαδό της περιοχής που αναφέρει το GPS ως θέση του χρήστη (ιδανικά αυτό ισούται με  $1 \text{ m}^2$  περίπου, αφού τότε είναι η επιφάνεια που καταλαμβάνει ένας άνθρωπος [51], όμως εφόσον εδώ εμπλέκεται το ζήτημα της ακρίβειας του GPS, η τιμή του θα είναι μεγαλύτερη – πιο συγκεκριμένα θα συζητηθεί παρακάτω). Ας συμβολίσουμε με  $A$  το πραγματικό εμβαδό που αντιστοιχεί στην περιοχή του χάρτη την οποία ο χρήστης αναφέρει σε ένα LBS επεξεργαστή ερωτημάτων ως θέση του, δηλαδή στην προαναφερθείσα περίπτωση το πραγματικό εμβαδό που αντιστοιχεί σε ένα tile κάποιου επιπέδου λεπτομέρειας. Στην περίπτωση αυτή, το  $M$  ισούται με τον αριθμό των διακριτών πιθανών θέσεων, οι οποίες μπορούν να αντιστοιχούν στην πραγματική ακριβή θέση του χρήστη στην περιοχή. Επομένως,

$$M = \lceil A / s_{GPS} \rceil \quad (4.2)$$

Κατά συνέπεια,  $p_j$  είναι η πιθανότητα κάθε μια από τις πιθανές ακριβείς θέσεις στην αναφερόμενη περιοχή να συνιστά την πραγματική ακριβή θέση του χρήστη και θα ισούται επομένως με:

$$p_j = s_{GPS} / A \quad (4.3)$$

Από τις (4.1), (4.2), (4.3), προκύπτει ότι η εντροπία εντοπισμού θέσης δίνεται από τον τύπο:

$$H = - \sum_{j=1}^M p_j \log(p_j) = - \sum_{j=1}^{\lceil A/s_{GPS} \rceil} \frac{s_{GPS}}{A} \log\left(\frac{s_{GPS}}{A}\right) \quad (4.4)$$

Η τιμή του  $A$  εξαρτάται από το επίπεδο λεπτομέρειας (με άλλα λόγια, το επίπεδο ακρίβειας). Συνεπώς, η τιμή της εντροπίας  $H$  μεταβάλλεται όταν tiles διαφορετικών επιπέδων ακρίβειας στην ιεραρχική δομή (π.χ. κόμβοι διαφορετικού βάθους στο quad-tree) επιλέγονται από το χρήστη για να προσδιορίσουν τη θέση του. Ένα συγκεκριμένο παράδειγμα υπολογισμού εντροπίας εντοπισμού θέσης δίνεται στην υποενότητα 4.3.

### 4.2.3. Εντροπία προσωπικών προτιμήσεων

Η γνωστοποίηση στον server των προσωπικών προτιμήσεων του χρήστη είναι ζωτικής σημασίας για την αποτελεσματική, ποιοτική παροχή μιας εξατομικευμένης υπηρεσίας. Οι προτιμήσεις αυτές μπορεί να περιλαμβάνουν μια πληθώρα παραμέτρων. Για παράδειγμα, το αγαπημένο είδος ταινιών ενός χρήστη αποτελεί απαραίτητη πληροφορία για μια υπηρεσία η οποία κάνει προτάσεις στο χρήστη, όταν αυτός δηλώσει ότι θέλει να πάει στον κινηματογράφο. Το αγαπημένο είδος μουσικής και η προτίμηση σε κάποια κουζίνα (π.χ. γαλλική, ινδική, κ.λπ.) συνιστούν δύο ακόμα χαρακτηριστικά παραδείγματα. Ωστόσο, και οι προτιμήσεις μπορούν να περιγραφούν είτε πιο συγκεκριμένα, οδηγώντας σε πιο ακριβή και «καλύτερης ποιότητας» αποτελέσματα, θυσιάζοντας την ιδιωτικότητα, είτε πιο αφηρημένα, γενικά, εξασφαλίζοντας έτσι μεγαλύτερη ιδιωτικότητα, αλλά μειώνοντας την ακρίβεια και ποιότητα των αποτελεσμάτων. Αναφορικά με το προηγούμενο παράδειγμα με τον κινηματογράφο, ένας χρήστης μπορεί να δηλώσει πολύ συγκεκριμένα την προτίμησή του σε ταινίες για φυσικές καταστροφές ή, πολύ πιο αόριστα, σε ταινίες δράσης.

Για να εφαρμόσουμε το μοντέλο (4.1) στις παραμέτρους προτιμήσεων (είδη ταινιών, είδη μουσικής, αθλήματα, κ.λπ.), πρέπει να ομαδοποιήσουμε τις πιθανές προτιμήσεις για κάθε παράμετρο σε ομάδες διαφορετικών επιπέδων ακρίβειας. Δεν υπάρχει ένας μόνο συγκεκριμένος τρόπος να γίνει αυτό. Μπορεί να γίνει αποκλειστικά με βάση τη λογική, τη γνώση ή την εμπειρία. Ωστόσο, μια περισσότερο συστηματική μέθοδος, όπως ένας αλγόριθμος συσταδοποίησης (clustering algorithm), αποτελεί μάλλον καλύτερη λύση. Στην ενότητα 5 παρουσιάζεται ένα πλαίσιο για τη δημιουργία κοινωνικών ομάδων ανάμεσα σε χρήστες, βάσει των προτιμήσεών τους οι οποίες δηλώνονται ως  $n$  λέξεις-κλειδιά. Συγκρίνονται αλγόριθμοι k-means, hierarchical clustering και spectral clustering και ο τελευταίος αποδεικνύεται πιο αποτελεσματικός για την περίπτωση. Έτσι, αν κάθε χρήστης επιλέξει  $n$  λέξεις-κλειδιά που περιγράφουν τις προτιμήσεις του για κάποια παράμετρο (π.χ.  $n$  διαφορετικά είδη ταινιών που του αρέσει να βλέπει), και εκτελέσουμε έναν αλγόριθμο spectral clustering για δημιουργία ομάδων μεταξύ των χρηστών, κάθε βήμα του αλγορίθμου θα δίνει έναν αριθμό clusters (συστάδων, ομάδων), που θα περιέχουν συγγενικά μεταξύ τους είδη ταινιών το καθένα, βάσει πραγματικών στοιχείων από χρήστες. Όσο ο αλγόριθμος προχωρά σε επόμενα βήματα, ο αριθμός των clusters θα

αυξάνεται και η πληθικότητά τους θα μειώνεται. Θα έχουμε έτσι οργανώσει τις πιθανές προτιμήσεις των χρηστών σε επίπεδα διαφορετικής ακρίβειας.

Η εφαρμογή του μοντέλου εντροπίας (4.1) είναι απλή και προκύπτει από τη χρήση της αναδρομικής ιδιότητας ως ακολούθως:

Ας υποθέσουμε ότι ένα cluster  $C_j^l$  (το  $j$ -στο cluster του επιπέδου  $l$ ) με πληθικότητα  $N_j^l$  διασπάται στο επόμενο επίπεδο ( $l+1$ ) και τα  $N_j^l$  στοιχεία του διασπείρονται σε τρία διαφορετικά clusters επιπέδου  $l+1$ : τα  $C_x^{l+1}$ ,  $C_y^{l+1}$  και  $C_z^{l+1}$ . Πιο συγκεκριμένα,  $n_x$  από τα  $N_j^l$  στοιχεία πηγαίνουν στο cluster  $C_x^{l+1}$ ,  $n_y$  πηγαίνουν στο cluster  $C_y^{l+1}$  και  $n_z$  πηγαίνουν στο cluster  $C_z^{l+1}$  ( $n_x + n_y + n_z = N_j^l \equiv n$ ). Η εντροπία  $H_j^l$  του cluster  $C_j^l$  ισούται με:

$$H_j^l = H\left(\frac{n_x}{n}, \frac{n_y}{n}, \frac{n_z}{n}\right) + \frac{n_x}{n} H_x^{l+1} + \frac{n_y}{n} H_y^{l+1} + \frac{n_z}{n} H_z^{l+1} \quad (4.5)$$

Ας σημειωθεί ότι στην ειδική (αλλά μάλλον αρκετά συχνή) περίπτωση μιας καθαρά ιεραρχικής δομής, όπου δηλαδή κάθε cluster στο επίπεδο  $l$  είναι υποσύνολο κάποιου cluster του επιπέδου  $l-1$  (αυτό θα συνέβαινε, για παράδειγμα, αν εφαρμόζαμε αλγόριθμο ιεραρχικής συσταδοποίησης - hierarchical clustering), το  $n_x$  ισούται με την πληθικότητα του cluster  $C_x^{l+1}$ , το  $n_y$  ισούται με την πληθικότητα του cluster  $C_y^{l+1}$  και το  $n_z$  με την πληθικότητα του  $C_z^{l+1}$ .

#### 4.2.4. Ενοποίηση μετρήσεων εντροπίας και χρήση σε ερωτήματα LBS

Έχοντας εφαρμόσει το μοντέλο εντροπίας τόσο στη γεωγραφική θέση όσο και στις προσωπικές προτιμήσεις, και συνεπώς έχοντας μοντελοποιήσει αριθμητικά την ιδιωτικότητα που σχετίζεται και με τις δύο αυτές παραμέτρους, μπορούμε τώρα να ενσωματώσουμε τις δύο αυτές «πηγές» εντροπίας / αοριστίας σε μία παράμετρο, η οποία αντιστοιχεί στο επίπεδο της συνολικής / σφαιρικής ιδιωτικότητας του χρήστη.

Ας υποθέσουμε ότι το ιεραρχικό γεωγραφικό σύστημα χαρτών που χρησιμοποιούμε υποστηρίζει  $g_{\max}$  επίπεδα λεπτομέρειας / ακρίβειας, όπου το  $g_{\max}$  είναι το επίπεδο μέγιστης ακρίβειας. Ας υποθέσουμε επίσης ότι μια συγκεκριμένη παράμετρος προσωπικών προτιμήσεων (π.χ. αγαπημένο είδος ταινιών) που είναι απαραίτητη για μια συγκεκριμένη υπηρεσία (π.χ. «βρες έναν κινηματογράφο στην περιοχή που είμαι ο οποίος παίζει μια ταινία του είδους που προτιμώ») έχει υποβληθεί σε διαδικασία

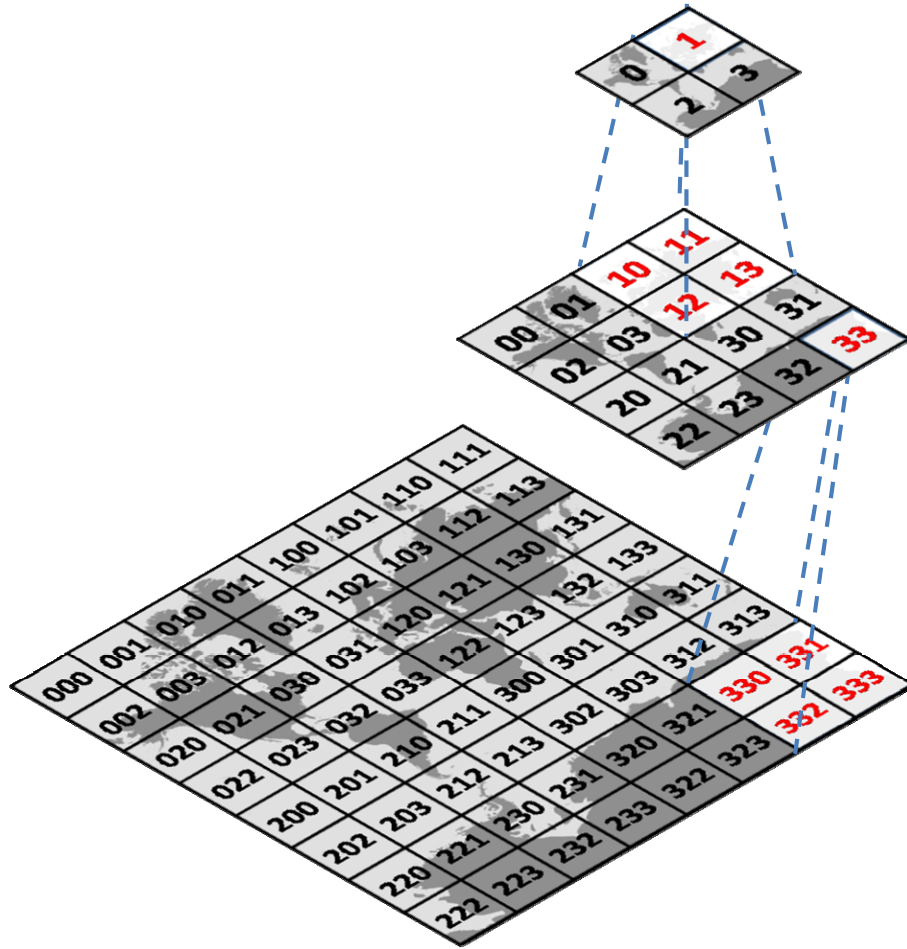
ομαδοποίησης (π.χ. με έναν αλγόριθμο συσταδοποίησης), με αποτέλεσμα την οργάνωση των πιθανών προτιμήσεων σε  $p_{\max}$  διακριτά επίπεδα, όπου το  $p_{\max}$  είναι το επίπεδο μέγιστης ακρίβειας. Όπως έχει ήδη αναφερθεί, η γεωγραφική θέση του χρήστη αντιστοιχεί σε έναν κόμβο καθενός από τα  $g_{\max}$  επίπεδα του ιεραρχικού γεωγραφικού συστήματος, και οι προτιμήσεις του χρήστη για την υπό εξέταση παράμετρο αντιστοιχούν σε ένα cluster για καθένα από τα  $p_{\max}$  επίπεδα της σχετικής οργάνωσης σε clusters. Υπολογίζοντας τις τιμές της εντροπίας εντοπισμού θέσης και προσωπικών προτιμήσεων για κάθε επίπεδο  $g_i$  και  $p_j$  αντίστοιχα, και ταξινομώντας τις σε αύξουσα σειρά, μπορούμε να χρησιμοποιήσουμε αυτήν τη διάταξη για να προσδιορίσουμε διαφορετικά επίπεδα ιδιωτικότητας, τα οποία ο χρήστης έχει τη δυνατότητα να επιλέξει όταν διατυπώνει LBS ερωτήματα προς τον server που παρέχει την υπηρεσία.

Επομένως, ο χρήστης χρειάζεται απλά να επιλέξει το επίπεδο της ιδιωτικότητας από ένα σύνολο τιμών, και, κατόπιν, χρησιμοποιείται το κατάλληλο επίπεδο ακρίβειας σε γεωγραφική θέση και προσωπικές προτιμήσεις ανάλογα με τις συνιστώσες της επιλεγείσας τιμής.

### 4.3. Ένα πρακτικό παράδειγμα

Για να καταστεί σαφέστερη η προτεινόμενη μεθοδολογία, παρουσιάζεται ένα πρακτικό παράδειγμα. Το γεωγραφικό σύστημα που θα χρησιμοποιηθεί είναι το Microsoft Virtual Earth (Windows Local Live) και η παράμετρος προσωπικών προτιμήσεων οι προτιμήσεις του χρήστη σε κινηματογραφικά είδη ταινιών. Ωστόσο, όπως αναφέρθηκε ανωτέρω, η προτεινόμενη μεθοδολογία δεν εξαρτάται από το συγκεκριμένο γεωγραφικό σύστημα.

Το Virtual Earth «τεμαχίζει» τον παγκόσμιο χάρτη σε τετραγωνίδια (tiles) μεγέθους 256x256 pixels. Προς το παρόν, υποστηρίζονται 23 επίπεδα ακρίβειας. Στο επίπεδο 1, ο παγκόσμιος χάρτης έχει διαστάσεις 512x512 pixels και αποτελείται από 4 tiles (tiles 0, 1, 2 και 3) [48]. Στο σχήμα 4.1 αναπαρίσταται γραφικά το σύστημα tiles του Virtual Earth.



**Σχήμα 4.1:** Αναπαράσταση της οργάνωσης σε δομή quad-tree των tiles στο Virtual Earth

Ξεκινάμε από το επίπεδο μέγιστης ακρίβειας, αναπαριστώντας δηλαδή τη θέση με ένα tile επιπέδου 23. Βαθμιαία θα κάνουμε “zoom out”, δηλαδή θα αναπαριστούμε τη θέση με tiles σε χαμηλότερα επίπεδα ακρίβειας, και σε κάθε βήμα θα υπολογίζουμε την αντίστοιχη εντροπία.

Το πραγματικό μήκος  $u_i$  της πλευράς (σε m) της περιοχής που αναπαριστά ένα tile επιπέδου  $i$  ( $i=1,2,..23$ ) ισούται με το μέγεθος ground resolution («ανάλυση εδάφους», ήτοι την απόσταση στο έδαφος που αναπαρίσταται από ένα μόνο pixel στο χάρτη) του συγκεκριμένου επιπέδου, πολλαπλασιασμένου επί 256 (pixels ανά tile). Για δύο διαδοχικά βήματα / επίπεδα έχουμε:

$$\frac{u_{i-1}}{u_i} = \frac{256 * ((\cos(\text{latitude} * \pi / 180) * 2 * \pi * 6378137 \text{ meters}) / (256 * 2^{i-1}))}{256 * ((\cos(\text{latitude} * \pi / 180) * 2 * \pi * 6378137 \text{ meters}) / (256 * 2^i))} = 2 \quad (4.6)$$

Όσον αφορά στην πραγματική επιφάνεια  $\alpha_i$  (σε  $m^2$ ) που αντιστοιχεί σε ένα tile στο επίπεδο  $i$ , αυτή ισούται με  $u_i^2$ , και για δύο διαδοχικά επίπεδα ισχύει:

$$\frac{\alpha_{i-1}}{\alpha_i} = 4 \quad (4.7)$$

Για να υπολογίσουμε την εντροπία, πρέπει πρώτα να υπολογίσουμε το  $A$  (το πραγματικό εμβαδό που αντιστοιχεί στην περιοχή του χάρτη την οποία ο χρήστης αναφέρει σε ένα LBS επεξεργαστή ερωτημάτων ως θέση του) και το  $s_{GPS}$  (την επιφάνεια της περιοχής που αναφέρει το GPS ως θέση του χρήστη). Έχουμε:

$$A_i = \alpha_i \quad (4.8)$$

Λαμβάνοντας υπόψη ότι το  $\alpha_{23}$  είναι η μικρότερη δυνατή περιοχή (καθώς το επίπεδο 23 είναι το επίπεδο μέγιστης ακρίβειας), η (4.7) δίνει:

$$\alpha_i = 4^{23-i} * \alpha_{23} \quad (4.9)$$

Οι (4.8) και (4.9) τώρα δίνουν:

$$A_i = 4^{23-i} * \alpha_{23} \quad (4.10)$$

Όσον αφορά στο  $s_{GPS}$ , πρέπει να προσδιοριστεί η ακρίβεια εντοπισμού του GPS. Δεδομένου ότι οι εμπορικές διαφημίσεις διατείνονται για ακρίβεια μέχρι και 1m, ενώ πρακτικές δοκιμές καταδεικνύουν ότι στην πράξη η ακρίβεια των συσκευών είναι σημαντικά μικρότερη, προσπαθήσαμε να κάνουμε μια υπόθεση για τη μέση ακρίβεια που θα προσεγγίζει κατά το δυνατό πραγματικές καταστάσεις. Προς τούτο, χρησιμοποιήθηκαν τα αποτελέσματα της δουλειάς των Wing, Eklund, Kellogg στο [52], όπου δοκιμάζουν την ακρίβεια και αξιοπιστία των εμπορικών συσκευών GPS σε μια ποικιλία τοποθεσιών (πόλεις, δάση, βουνά, κ.λπ). Σύμφωνα με τα αποτελέσματα των δοκιμών τους, η καλύτερη ακρίβεια, που επετεύχθη σε συνθήκες ανοιχτού ουρανού, μετρήθηκε κατά μέσο όρο στα 5m περίπου. Λαμβάνοντας υπόψη το γεγονός ότι η επιφάνεια που καλύπτει ένα tile στο επίπεδο 23 (επίπεδο μέγιστης ακρίβειας) του συστήματος Virtual Earth System είναι  $22,9m^2$  ( $0.0187meters/pixel * 256pixels$ )<sup>2</sup> [48], αποφασίσαμε να θεωρήσουμε την επιφάνεια ακρίβειας GPS ίση με την επιφάνεια που καλύπτει ένα tile επιπέδου 23 (περίπου ισοδύναμο με ένα τετράγωνο πλευράς σχεδόν 5m). Θεωρούμε, δηλαδή, ότι:

$$s_{GPS} = \alpha_{23} \quad (4.11)$$



Λαμβάνοντας υπόψη τις (4.10) και (4.11), η (4.4) δίνει τον ακόλουθο τύπο για τον υπολογισμό της εντροπίας εντοπισμού θέσης  $H_g(i)$  στο επίπεδο  $i$  ( $i=1,2,..,23$ ):

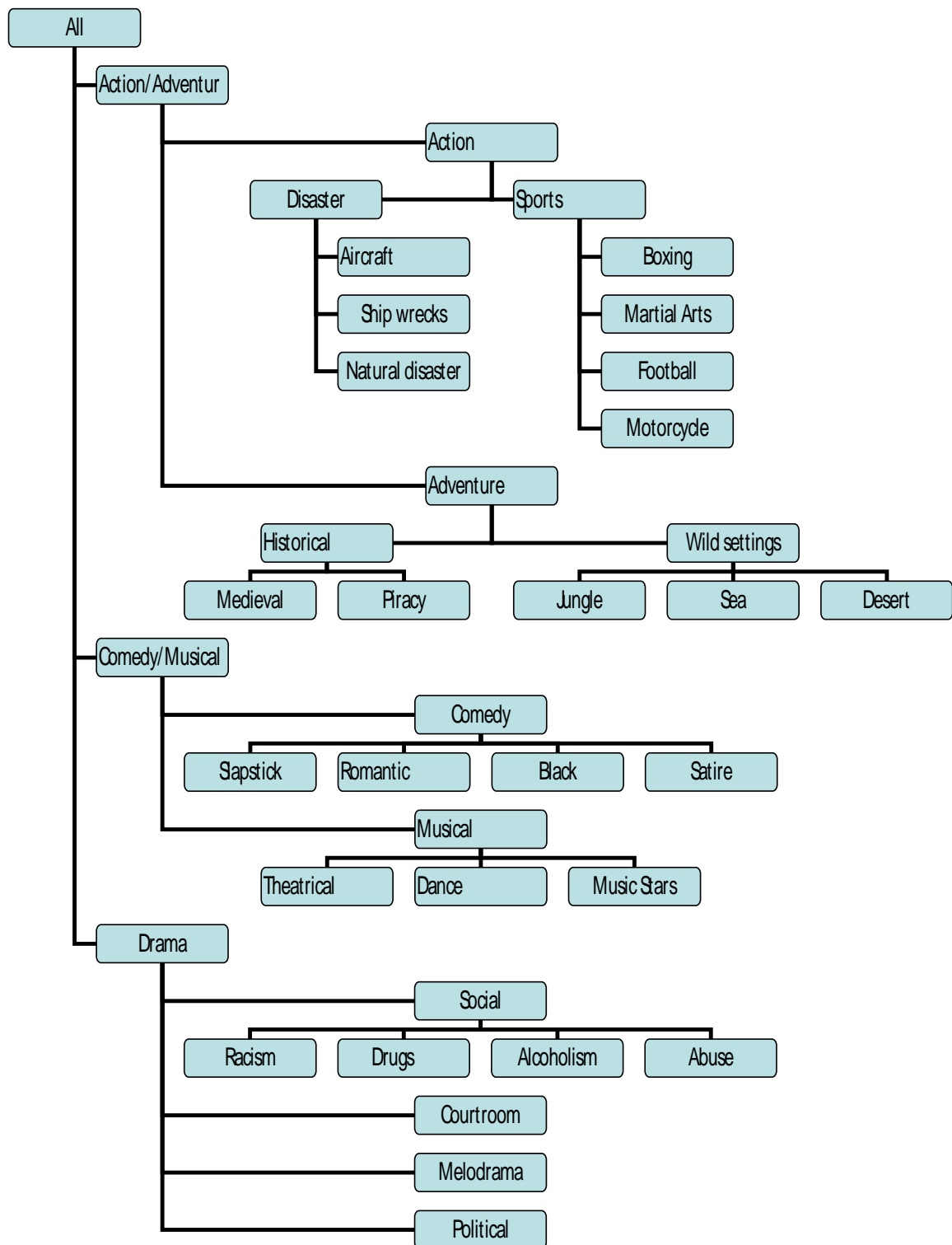
$$\begin{aligned} H_g(i) &= - \sum_{j=1}^{4^{23-i}} \frac{a_{23}}{4^{23-i} * a_{23}} \log\left(\frac{a_{23}}{4^{23-i} * a_{23}}\right) \\ &= -4^{23-i} * \left(\frac{1}{4^{23-i}} \log\left(\frac{1}{4^{23-i}}\right)\right) = \log(4^{23-i}) = (23-i) \log 4 \end{aligned} \quad (4.12)$$

Επομένως, κάθε φορά που μεγαλώνει η περιοχή που ορίζει τη θέση του χρήστη (με την άνοδο κατά ένα επίπεδο στο δέντρο του Virtual Earth System), η προστιθέμενη εντροπία ισούται με  $\log 4$ :

$$\Delta H_g = H_g(i) - H_g(i+1) = \log 4 = 1.386 \quad (4.13)$$

Σε αυτό το σημείο, πρέπει να σημειωθεί ότι ο Shannon χρησιμοποιεί λογάριθμο βάσης 2 για την πληροφοριακή εντροπία, καθώς ασχολείται με binary data. Στην παρούσα μελέτη, επιλέξαμε να χρησιμοποιήσουμε το φυσικό (νεπέριο) λογάριθμο με βάση  $e$ , εφόσον είναι αυτός που εμφανίζεται πιο συχνά στην περιγραφή φυσικών διεργασιών (δηλαδή εδώ είναι  $\log \equiv \log_e \equiv \ln$ ). Εντούτοις, η επιλογή της βάσης του λογαρίθμου δεν επηρεάζει τη μεθοδολογία μας.

Όσον αφορά στις προτιμήσεις σε είδη ταινιών, δημιουργήσαμε μια οργανωμένη, ιεραρχική δομή, η οποία περιέχει αρκετά είδη και υπο-είδη ταινιών, βάσει μιας εξειδικευμένης ιστοσελίδας [53]. Η δομή αναμφίβολα δεν είναι εξαντλητική, δηλαδή περιέχει ορισμένες μόνο κατηγορίες και υποκατηγορίες και έχει σαν μοναδικό στόχο να χρησιμεύσει ως βάση για την περιγραφή της διαδικασίας με την οποία η προτεινόμενη μεθοδολογία μπορεί να εφαρμοστεί στην πράξη. Όπως έχει εξηγηθεί ανωτέρω, μπορούν (και προτείνεται) να χρησιμοποιηθούν άλλες μέθοδοι ομαδοποίησης των ειδών ταινιών, όπως αλγόριθμοι συσταδοποίησης. Στο σχήμα 4.2 φαίνεται η ιεραρχική οργάνωση των ειδών ταινιών που χρησιμοποιούνται στο παράδειγμά μας.



Σχήμα 4.2: Ιεραρχική οργάνωση clusters με είδη ταινιών

Στη συγκεκριμένη περίπτωση είναι  $p_{\max}=5$ . Θα υπολογίσουμε την εντροπία βάσει της (5) για κάθε κόμβο του δέντρου.

Κατ'αρχήν είναι,  $H=0$  για όλα τα φύλλα του δέντρου, εφόσον στις περιπτώσεις αυτές προσδιορίζονται με απόλυτη ακρίβεια οι προτιμήσεις του χρήστη από το εκάστοτε υπο-είδος ταινιών. Για τα άλλα, έχουμε:

$$H(\text{"Disaster"}) = H(1/3, 1/3, 1/3) + 1/3 * H(\text{"Aircraft"}) + 1/3 * H(\text{"Ship Wrecks"}) + 1/3 * H(\text{"Natural Disasters"}) = \log 3 + 0 + 0 + 0 = 1.099$$

$$H(\text{"Sports"}) = H(1/4, 1/4, 1/4, 1/4) + 1/4 * H(\text{"Boxing"}) + 1/4 * H(\text{"Martial Arts"}) + 1/4 * H(\text{"Football"}) + 1/4 * H(\text{"Motorcycle"}) = 1.386$$

$$H(\text{"Historical Adventures"}) = H(1/2, 1/2) + 1/2 * H(\text{"Medieval"}) + 1/2 * H(\text{"Wild Settings"}) = \log 2 + 0 + 0 = 0.693$$

$$H(\text{"Wild Settings"}) = H(1/3, 1/3, 1/3) + 1/3 * H(\text{"Jungle"}) + 1/3 * H(\text{"Sea"}) + 1/3 * H(\text{"Disaster"}) = 1.099$$

$$H(\text{"Comedy"}) = H(1/4, 1/4, 1/4, 1/4) + 1/4 * H(\text{"Slapstick"}) + 1/4 * H(\text{"Romantic Comedy"}) + 1/4 * H(\text{"Black"}) + 1/4 * H(\text{"Satire"}) = 1.386$$

$$H(\text{"Musical"}) = H(1/3, 1/3, 1/3) + 1/3 * H(\text{"Theatrical"}) + 1/3 * H(\text{"Dance"}) + 1/3 * H(\text{"Music Stars"}) = 1.099$$

$$H(\text{"Social"}) = H(1/4, 1/4, 1/4, 1/4) + 1/4 * H(\text{"Racism"}) + 1/4 * H(\text{"Drugs"}) + 1/4 * H(\text{"Alcoholism"}) + 1/4 * H(\text{"Abuse"}) = 1.386$$

$$H(\text{"Action"}) = H(3/7, 4/7) + 3/7 * H(\text{"Disaster"}) + 4/7 * H(\text{"Sports"}) = 1.946$$

$$H(\text{"Adventure"}) = H(2/5, 3/5) + 2/5 * H(\text{"Historical Adventures"}) + 3/5 * H(\text{"Wild Settings"}) = 1.610$$

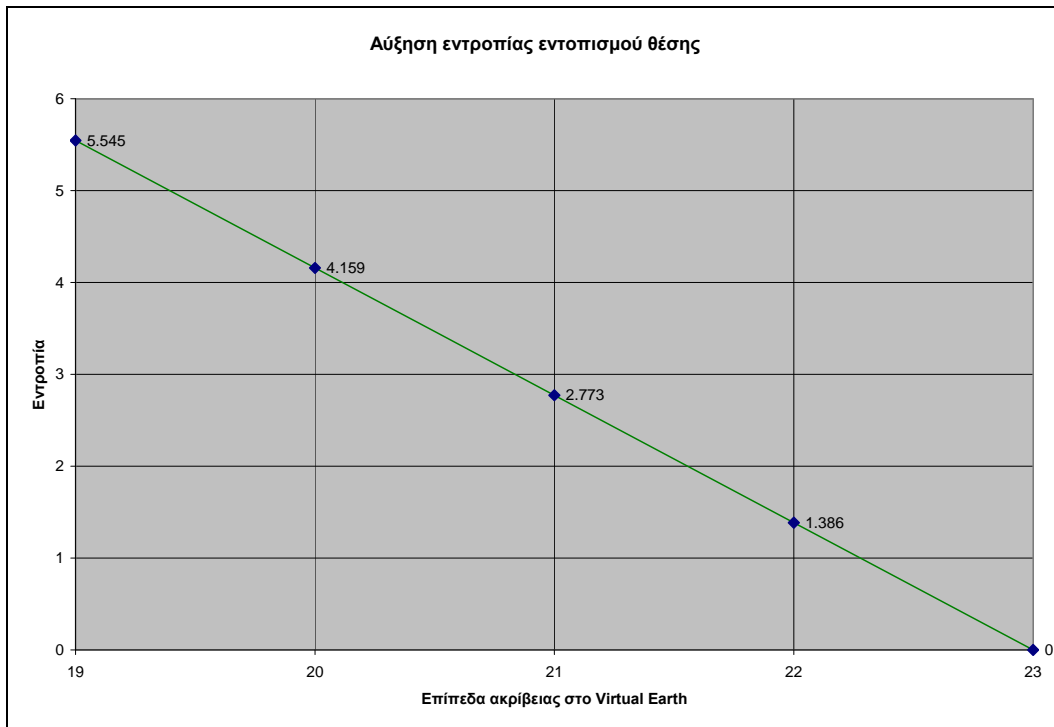
$$H(\text{"Action/Adventure"}) = H(7/12, 5/12) + 7/12 * H(\text{"Action"}) + 5/12 * H(\text{"Adventure"}) = 2.485$$

$$H(\text{"Comedy/Musical"}) = H(4/7, 3/7) + 4/7 * H(\text{"Comedy"}) + 3/7 * H(\text{"Musical"}) = 1.946$$

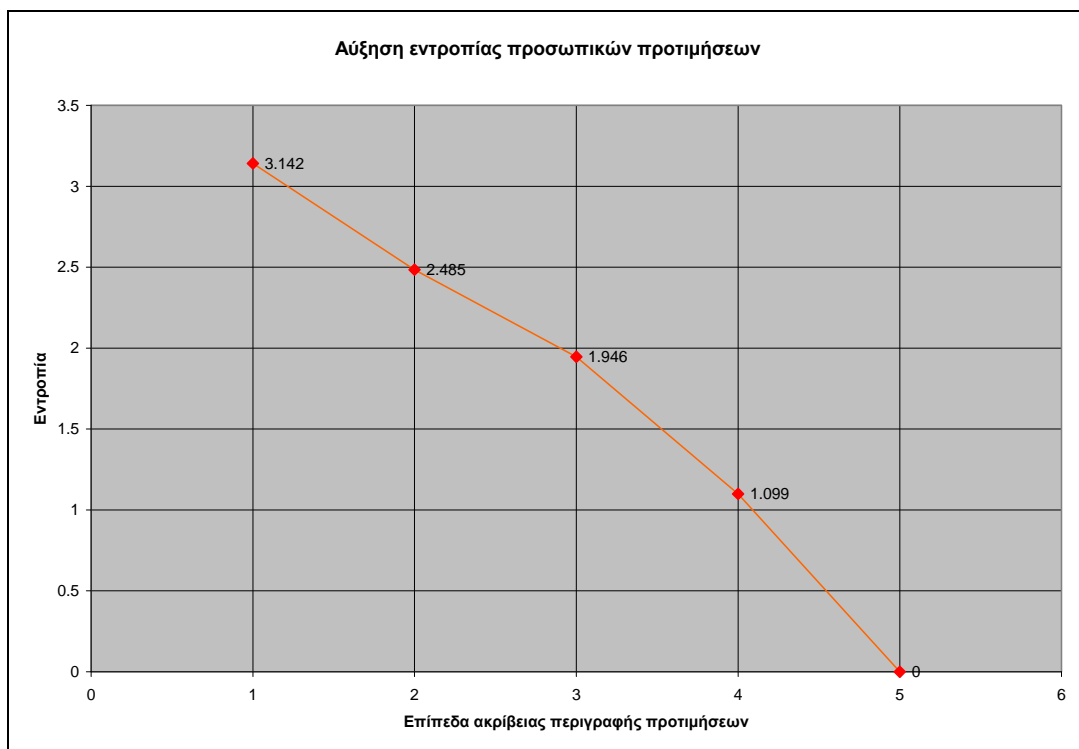
$$H(\text{"Drama"}) = H(4/7, 1/7, 1/7, 1/7) + 4/7 * H(\text{"Social"}) + 1/7 * H(\text{"Courtroom"}) + 1/7 * H(\text{"Melodrama"}) + 1/7 * H(\text{"Political"}) = 1.513$$

$$H(\text{"All Movies"}) = H(12/26, 7/26, 7/26) + 12/26 * H(\text{"Action/Adventure"}) + 7/26 * H(\text{"Comedy/Musical"}) + 7/26 * H(\text{"Drama"}) = 3.142$$

Στους παραπάνω υπολογισμούς, θεωρούμε ότι το εκάστοτε ποσοστό συνεισφοράς ισούται με το λόγο της πληθικότητας κάθε υποσυνόλου προς την πληθικότητα του νέου υπερσυνόλου.

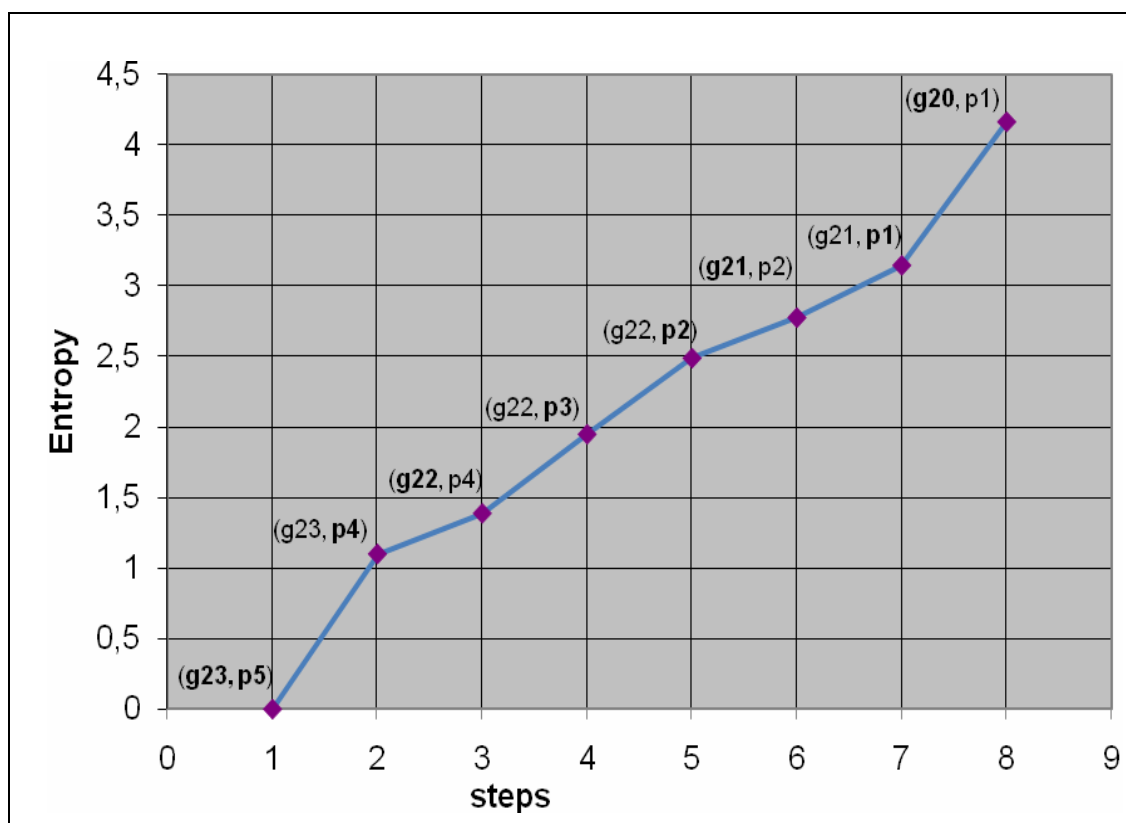


Σχήμα 4.3: Εντροπία εντοπισμού θέσης συναρτήσει επιπέδου ακρίβειας



Σχήμα 4.4: Εντροπία προσωπικών προτιμήσεων συναρτήσει επιπέδου ακρίβειας περιγραφής προσωπικών προτιμήσεων

Έχοντας υπολογίσει όλες τις πιθανές τιμές της εντροπίας αναφορικά με τη θέση (σχήμα 4.3) και τις προσωπικές προτιμήσεις (για είδη ταινιών) (σχήμα 4.4) και θεωρώντας ότι η προτίμηση του χρήστη εκφρασμένη με τον ακριβέστερο τρόπο είναι σε ταινίες «natural disaster», ταξινομούμε αυτές τις τιμές σε αύξουσα σειρά και κατασκευάζουμε το διάγραμμα που φαίνεται στο σχήμα 4.5. Στο διάγραμμα αυτό, ο άξονας X δείχνει τα διαφορετικά βήματα προς την αύξηση της εντροπίας, της οποίας η τιμή αναπαρίσταται στον άξονα Y.



**Σχήμα 4.5:** Τιμές της εντροπίας για διαφορετικά επίπεδα ακρίβειας των πληροφοριών του χρήστη

Οι ετικέτες στα σημεία του διαγράμματος δείχνουν τις τιμές  $g_i$  και  $p_j$  για τη συγκεκριμένη κατάσταση. Ανάμεσα σε δύο διαδοχικά βήματα, ένα εκ των  $g_i$  ή  $p_j$  γίνεται  $g_{i-1}$  ή  $p_{j-1}$  αντίστοιχα. Η αλλαγή που οδηγεί στη μικρότερη ολική εντροπία καθορίζει ποιο από τα δύο πρέπει να μεταβληθεί. Στο διάγραμμα, η παράμετρος που αλλάζει σε σχέση με το προηγούμενο βήμα αναγράφεται με έντονα γράμματα.

Έχοντας κατασκευάσει το διάγραμμα, μπορούμε να «αναθέσουμε» διαφορετικά επίπεδα ιδιωτικότητας / ακρίβειας στα διάφορα βήματα, με το βήμα 1 να αντιστοιχεί σε ελάχιστη

ακρίβεια / μέγιστη ιδιωτικότητα και το βήμα 8 σε μέγιστη ιδιωτικότητα / ελάχιστη ακρίβεια. Επομένως, ο χρήστης πρέπει απλώς να επιλέξει το επίπεδο ιδιωτικότητας που επιθυμεί. Ο κατάλληλος βαθμός λεπτομέρειας / αοριστίας για γεωγραφική θέση και προσωπικές προτιμήσεις υπολογίζονται αυτόματα, αποκρύπτοντας από το χρήστη την πολυπλοκότητα των ζητημάτων αυτών.

Επιπρόσθετα, το παραπάνω μοντέλο μπορεί να χρησιμοποιηθεί και στην πλευρά του server, εφόσον το ερώτημα του χρήστη μπορεί να μην επιστρέψει τον επιθυμητό αριθμό αποτελεσμάτων. Για παράδειγμα, ένα ερώτημα για ένα συγκεκριμένο είδος ταινίας που να παίζεται στην άμεση περιοχή του χρήστη, μπορεί να μην επιστρέψει καμία τιμή. Χρησιμοποιώντας την ίδια μεθοδολογία, ο server μπορεί να αρχίσει να ψάχνει για εναλλακτικές, ακολουθώντας την αύξηση της εντροπίας και, άρα, αποφασίζοντας ποια από τις δύο παραμέτρους: κοντινή περιοχή ή μεγαλύτερη ακρίβεια στην προτίμηση του χρήστη στο είδος ταινίας, θα πρέπει να «χαλαρωθεί» σε κάθε βήμα, ώσπου να επιστραφεί ένας αποδεκτός αριθμός αποτελεσμάτων. Αυτό θα γίνει είτε διευρύνοντας τη γεωγραφική περιοχή αναζήτησης είτε αυξάνοντας τα είδη ταινιών που εξετάζονται, προσθέτοντας άλλα είδη, παρεμφερή με αυτά του αρχικού ερωτήματος.

Ένα σημαντικό θέμα εδώ είναι η βαρύτητα καθεμίας από τις δύο παραμέτρους  $g_i$  και  $p_i$ , μια και για διαφορετικούς χρήστες ενδεχομένως να έχουν διαφορετικό βάρος. Γι' αυτό, η σταθερά  $K$  στον υπολογισμό της εντροπίας όπως παρουσιάστηκε στην (4.1) χρησιμοποιείται για τον καθορισμό της σημασίας κάθε παραμέτρου, ώστε να εξυπηρετηθούν ακόμα καλύτερα οι προτιμήσεις του χρήστη. Σε μία από τις δύο παραμέτρους – δεν επηρεάζει σε ποια, έτσι χωρίς βλάβη της γενικότητας επιλέγουμε τη γεωγραφική – αναθέτουμε την τιμή  $K=1$ . Για να υπολογίσουμε τώρα τη σταθερά  $K$  που θα αποτελεί το συντελεστή βαρύτητας της ιδιωτικότητας των προσωπικών προτιμήσεων έναντι της ιδιωτικότητας του εντοπισμού θέσης, πρέπει να προσδιοριστεί η σημασία που έχει για το χρήστη η μια παράμετρος σε σχέση με την άλλη. Μια απλή προσέγγιση για τον υπολογισμό του  $K$  θα ήταν να εξισώσουμε τις εντροπίες στα υψηλότερα επίπεδα των δύο τύπων, ήτοι  $g_i$  και  $p_i$ . Στις προσωπικές προτιμήσεις, το υψηλότερο επίπεδο είναι αυτό του ενός και μοναδικού cluster, το οποίο περιέχει όλους τους πιθανούς τύπους επιλογών του χρήστη (στο παράδειγμα, όλα τα είδη ταινιών), δηλαδή το επίπεδο  $p_1$ . Όσον αφορά στη γεωγραφική θέση, ως υψηλότερο επίπεδο δεν πρέπει να θεωρηθεί το επίπεδο 1 της

ιεραρχικής δομής (καθώς αυτό αντιστοιχεί στον παγκόσμιο χάρτη), αλλά στο επίπεδο εκείνο που περιέχει μια περιοχή αρκετά μεγάλη, ώστε να θεωρεί ο χρήστης ότι προστατεύει σε μέγιστο βαθμό την ιδιωτικότητα της θέσης τους. Όταν ο χρήστης αποφασίσει ποια είναι αυτή η περιοχή, το αντίστοιχο επίπεδο στην ιεραρχία του quad-tree επιλέγεται ως το ψηλότερο επίπεδο αφαίρεσης (έστω ότι είναι το  $g_M$ ), και υπολογίζεται η εντροπία  $H(g_M)$  που αντιστοιχεί. Εξισώνοντας την εντροπία του  $g_M$  με αυτήν του  $p_1$  και λύνοντας ως προς  $K$ , λαμβάνουμε τη ζητούμενη τιμή του  $K$ :

$$H(g_M) = KH(p_1) \Rightarrow K = \frac{H(g_M)}{H(p_1)} \Rightarrow K = \frac{(23 - M) \log 4}{H(p_1)} \quad (4.14)$$

Μπορούμε λοιπόν στο εξής να θεωρήσουμε ότι η εντροπία προσωπικών προτιμήσεων ισούται με:

$$H'(p_j) = K H(p_j) \quad (4.15)$$

Με τον τρόπο αυτό, μπορούμε να προχωρήσουμε στην επιλογή των βημάτων προς την αύξηση της εντροπίας με τη διαδικασία που περιγράφηκε ανωτέρω.

#### 4.4. Συμπεράσματα

Στην ενότητα αυτή παρουσιάσαμε μια μεθοδολογία για την αντιμετώπιση των αντικρουόμενων τάσεων της ιδιωτικότητας και της ακρίβειας των LBS ερωτημάτων, μέσω ενός απλού, εύχρηστου τρόπου. Η μεθοδολογία βασίστηκε στη χρήση της εντροπίας για τον προσδιορισμό του κατάλληλου επιπέδου ιδιωτικότητας / ακρίβειας ανάλογα με τις ανάγκες του χρήστη. Η ίδια μεθοδολογία μπορεί να χρησιμοποιηθεί στην πλευρά του server ενός LBS συστήματος με σκοπό τη βελτίωση της παρεχόμενης υπηρεσίας.

Η μεθοδολογία είναι ανεξάρτητη από δεσμεύσεις αρχιτεκτονικής, σε αντίθεση με άλλα μοντέλα που έχουν κατά καιρούς προταθεί και ορισμένα από τα οποία αναφέραμε σε προηγούμενη ενότητα (π.χ. anonymizers), ενώ χρησιμοποιεί ένα απλό interface για την εισαγωγή των ρυθμίσεων ιδιωτικότητας του χρήστη και τη μετατροπή τους στα κατάλληλα επίπεδα αφαίρεσης / ακρίβειας. Επιπλέον, δεν απαιτεί συσχέτιση με άλλες πληροφορίες του χρήστη για την παροχή ιδιωτικότητας στις προσωπικές πληροφορίες,

για και αυτό επιτυγχάνεται μέσω της μίξης της αοριστίας των προσωπικών προτιμήσεων και αφαίρεσης στον εντοπισμό θέσης, ώστε να επιτευχθεί η επιθυμητή τιμή εντροπίας.

Επί του παρόντος, η προτεινόμενη μεθοδολογία χρησιμοποιείται στην πράξη για την παροχή εξατομικευμένων, προσαρμοζόμενων στο περιβάλλον (context aware) κινητών υπηρεσιών μέσω της πλατφόρμας PLASMA [54], περιγραφή της οποίας παρατίθεται στην ενότητα 7.



## 5. ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗ ΧΡΗΣΤΩΝ ΒΑΣΕΙ ΠΡΟΣΩΠΙΚΩΝ ΠΡΟΤΙΜΗΣΕΩΝ ΜΕ ΧΡΗΣΗ ΜΕΘΟΔΩΝ CLUSTERING

### 5.1. Εισαγωγή

Στην ενότητα 4, στην οποία εξετάστηκε η σχέση της ποιότητας μιας εξατομικευμένης (personalized) υπηρεσίας με την προστασία της ιδιωτικότητας των προσωπικών πληροφοριών του χρήστη κατέστη σαφής η ανάγκη ύπαρξης μιας δομημένης οργάνωσης των προτιμήσεων σε ομάδες που θα περιέχουν συναφείς επιλογές και μάλιστα σε διαφορετικά επίπεδα αφαίρεσης / ακρίβειας. Το ζήτημα, ωστόσο, της δημιουργίας κοινωνικών ομάδων μεταξύ των χρηστών με βάση τις προσωπικές τους προτιμήσεις και τα ενδιαφέροντά τους εμφανίζεται και σε ευρύτερο φάσμα διαδικτυακών και κινητών υπηρεσιών.

Ο σχεδιασμός ενός πλαισίου που θα επιτρέπει την αξιοποίηση των κοινών χαρακτηριστικών μεταξύ των χρηστών όσον αφορά σε προτιμήσεις και ενδιαφέροντα με σκοπό τη δημιουργία εικονικών κοινοτήτων (virtual communities) που συμβάλλουν αποτελεσματικά στην κοινωνικοποίηση (socializing) αποτελεί ένα ιδιαίτερα ενδιαφέρον αντικείμενο εργασίας, καθώς υπάρχει μεν πληθώρα εφαρμογών που προωθούν την «κοινωνική δικτύωση» (social networking), όπως τα newsgroups, chat rooms, weblogs, η οποία όμως δεν συνοδεύεται από τρόπους δημιουργίας σχέσεων συνάφειας και συνεργασίας. Φυσικά, οι έννοιες: ubiquitous κινητές επικοινωνίες και εξατομικευμένες (personalized) υπηρεσίες προσδίδουν πολλαπλάσια αξία στο προαναφερθέν αντικείμενο, καθιστώντας την κοινωνική δικτύωση (social networking) σε κινητές επικοινωνίες έναν πολλά υποσχόμενο τομέα για έρευνα.

Στην ενότητα αυτή, συζητείται ένα πλαίσιο (βασισμένο σε ιδέες και εργασία που έχει γίνει στα [55][56]) μιας υπηρεσίας βασισμένης στα προφίλ των χρηστών (user profile based service), η οποία έχει ως σκοπό την αυτοματοποιημένη δημιουργία ομάδων ενδιαφερόντων ανάμεσα στους χρήστες, να οργανώσει, δηλαδή, τους χρήστες σε ομάδες με παρεμφερή ενδιαφέροντα, με βάση τα προσωπικά προφίλ προτιμήσεών τους.

## 5.2. Μοντελοποίηση προφίλ

Τα προσωπικά προφίλ προτιμήσεων μπορούν να προκύπτουν έμμεσα, π.χ. από την επισκόπηση των δραστηριοτήτων του χρήστη [57], των ιστοσελίδων που επισκέπτεται [58], [59], ή, βεβαίως, να συντάσσονται άμεσα από το χρήστη (στην ενότητα 2, άλλωστε, παρουσιάστηκε αναλυτικά δουλειά που έχει γίνει για τον ορισμό προφίλ). Προς χάριν της γενικότητας, θα θεωρήσουμε στην παρακάτω ανάλυση ότι η αναπαράσταση ενός προφίλ χρήστη συνίσταται σε μια λίστα  $n$  λέξεων-κλειδιών που καταδεικνύουν την προσωπικότητα, τις προτιμήσεις, τα ενδιαφέροντα του χρήστη – μπορούν είτε να αναφέρονται σε μια συγκεκριμένη παράμετρο (π.χ. ταινίες) είτε να αφορούν ευρύτερο φάσμα ενδιαφερόντων, ανάλογα με το είδος των ομάδων που θέλουμε να δημιουργήσουμε.

Αναφορικά με την επίδραση της αναπαράστασης του προφίλ στην ποιότητα της κοινωνικής ομαδοποίησης (social grouping), θα μελετηθούν δύο πλευρές της πρώτης: το μέγεθος του προφίλ και η κατανομή πιθανότητας των λέξεων-κλειδιών. Σχετικά με το πρώτο, θα μελετηθεί πώς επηρεάζει ο αριθμός  $n$  των λέξεων-κλειδιών που χρησιμοποιούνται για να περιγράψουν τις προτιμήσεις του χρήστη την ποιότητα του αποτελέσματος της διαδικασίας δημιουργίας ομάδων χρηστών εφαρμόζοντας αλγορίθμους συσταδοποίησης (clustering algorithms). Όσον αφορά στο δεύτερο, θεωρούμε ότι η συχνότητα εμφάνισης των λέξεων-κλειδιών μπορεί να ακολουθεί: i) το νόμο του Zipf, κάτι που διαισθητικά μεταφράζεται στο ότι λίγες λέξεις εμφανίζονται πολύ συχνά, ενώ η πλειονότητα των λέξεων εμφανίζεται σπάνια και ii) ομοιόμορφη κατανομή (uniform distribution). Η πρώτη υπόθεση βασίζεται στην ευρεία χρήση του νόμου του Zipf σε εργασίες μοντελοποίησης προτιμήσεων χρηστών [60], [61], [62]. Η κατανομή Zipf δίνεται από τον τύπο:

$$P(k, a) = 1/[k^a \cdot \zeta(a)] \quad (5.1)$$

και εκφράζει την πιθανότητα εμφάνισης της  $k$ -στης (κατά φθίνουσα συχνότητα εμφάνισης) λέξης του «λεξικού» από το οποίο αντλούνται λέξεις-κλειδιά για τη δημιουργία προφίλ. Το  $\zeta(a)$  είναι η συνάρτηση ζήτα του Riemann και το  $a$  είναι η παράμετρος skewness («λόξωσης») της κατανομής. Στα πειράματα που διεξήχθησαν, τα προφίλ δημιουργήθηκαν από ένα σύνολο 5000 πιθανών λέξεων με  $a = 1.5$ , βάσει άλλων μελετών για το θέμα [63]. Η δεύτερη υπόθεση (ομοιόμορφη κατανομή) γίνεται για να

μελετηθεί η περίπτωση έντονης διασποράς των προτιμήσεων των χρηστών: διεξήχθησαν, δηλαδή, τα πειράματα θεωρώντας ισοπίθανες όλες τις λέξεις-κλειδιά.

### 5.3. Μετρικές κοινωνικής δικτύωσης

Εδώ θα περιγραφούν οι μετρικές που χρησιμοποιούνται για τον ορισμό σχέσεων μεταξύ διαφορετικών προφίλ. Η εύρεση μιας μετρικής για τη μοντελοποίηση της ανομοιότητας μεταξύ δυο προφίλ είναι απαραίτητη, καθώς θα είναι το μέγεθος στο οποίο θα βασιστούν οι μέθοδοι clustering. Προτείνεται λοιπόν η έννοια της *σημασιολογικής απόστασης* (*semantic distance*) μεταξύ δύο προφίλ χρηστών  $u_i$  and  $u_j$ , η οποία ορίζεται ως εξής:

$$d_{ij} \equiv d(u_i, u_j) = 1 - \frac{a_{ij}}{|u|} = 1 - s_{ij} \quad (5.2)$$

όπου το  $a_{ij}$  συμβολίζει τον αριθμό των κοινών στοιχείων των προφίλ  $u_i$  and  $u_j$ , το  $|u|$  εκφράζει την πληθικότητα των προφίλ (σε κάθε υλοποίηση του πλαισίου τα προφίλ θεωρούνται ίσης πληθικότητας).

Αντίστοιχα, εισάγεται και η αντίθετη έννοια, η *σημασιολογική εγγύτητα* (*semantic proximity*) μεταξύ δυο προφίλ  $u_i$  and  $u_j$ , η οποία ορίζεται ως εξής:

$$s_{ij} = \frac{a_{ij}}{|u|} \quad (5.3)$$

δηλαδή ο λόγος των κοινών λέξεων-κλειδιών προς την πληθικότητα των προφίλ. Ο ορισμός αυτής της μετρικής είναι απαραίτητος για την εφαρμογή ενός αλγορίθμου spectral clustering που χρησιμοποιεί σαν είσοδο έναν πίνακα ομοιότητας (similarity matrix)  $S = [s_{ij}]$ , όπου  $s_{ij}$  εκφράζει την ομοιότητα των προφίλ  $u_i$  και  $u_j$ . Είναι σαφές ότι ο όρος  $a_{ij}/|u|$  είναι κανονικοποιημένος, δηλαδή παίρνει τιμές από 0 ως 1.

Τέλος, η έννοια του *κέντρου* (*centroid*) μιας ομάδας – συστάδας είναι απαραίτητη για τους αλγορίθμους k-means και αρκετές παραλλαγές hierarchical clustering. Ένα cluster centroid ορίζεται ως ένα σημείο του οποίου οι συντεταγμένες ισούνται με τη μέση τιμή των συντεταγμένων των μελών του αντίστοιχου cluster. Για την παρούσα περίπτωση, όμως, ο ορισμός αυτός θα τροποποιηθεί, ώστε να θεωρείται ότι ένα cluster centroid ή

αλλιώς εκπρόσωπος ομάδας (*group representative*) μιας ομάδας  $k = 1, 2, \dots, R$ , είναι το μέλος εκείνος της ομάδας για το οποίο ισχύει:

$$\mu_k = \arg \min_i \sum_{j \in k} d_s^{ij} \quad (5.4)$$

ήτοι, το σημείο του οποίου το άθροισμα των σημασιολογικών αποστάσεων από τα υπόλοιπα μέλη της ομάδας είναι το ελάχιστο.

## 5.4. Μέθοδοι clustering για δημιουργία κοινοτήτων χρηστών

Από τις διάφορες τεχνικές που έχουν προταθεί για τη δημιουργία κοινοτήτων χρηστών, οι μέθοδοι συσταδοποίησης είναι από τις πιο ευρέως χρησιμοποιούμενες. Εδώ συζητούνται τρεις από τους γνωστότερους τύπους αλγορίθμων συσταδοποίησης (clustering algorithms), οι: k-means, hierarchical clustering και spectral clustering.

### 5.4.1. K-means clustering

Ο αλγόριθμος k-means [63] είναι ένας απλός μη επιβλεπόμενος αλγόριθμος συσταδοποίησης, ο οποίος χρησιμοποιεί ως είσοδο τον αριθμό των clusters και ένα centroid (central point) για κάθε cluster. Ο αλγόριθμος προχωρά αντιστοιχίζοντας επαναληπτικά κάθε σημείο σε ένα centroid (και άρα σε ένα cluster), και κατόπιν επανυπολογίζοντας νέα centroids, κ.ο.κ. μέχρι να συγκλίνει. Ο τελικός στόχος είναι η ελαχιστοποίηση μιας αντικειμενικής συνάρτησης, η οποία εκφράζει το άθροισμα των τετραγώνων των αποστάσεων ανάμεσα στα σημεία και στα κέντρα των clusters:

$$J = \sum_{j=1}^R \sum_{i=1}^N \|x_i^{(j)} - \mu_j\|^2 \quad (5.5)$$

όπου  $R$  είναι ο αριθμός των clusters,  $N$  ο αριθμός των σημείων (χρηστών),  $x_i^{(j)}$  το  $i$ -στο σημείο που έχει αντιστοιχηθεί στο centroid  $\mu_j$  και το  $\|\cdot\|$  αναπαριστά μια μετρική απόστασης (στην περίπτωσή μας τη σημασιολογική απόσταση).

### 5.4.2. Hierarchical clustering

Το hierarchical clustering (ιεραρχική συσταδοποίηση) [64] είναι μια απλή τεχνική, η οποία διαιρείται σε δύο κύριες υποκατηγορίες αλγορίθμων: divisive και agglomerative. Η πρώτη κατηγορία αποσυνθέτει σταδιακά το σύνολο των σημείων (τα οποία θεωρούνται αρχικά μια ομάδα) σε ολοένα μικρότερα clusters, ώσπου κάθε σημείο να αποτελεί ένα cluster μόνο του. Η δεύτερη κατηγορία (που χρησιμοποιείται στη μελέτη, καθώς ταιριάζει καλύτερα στις πραγματικές περιπτώσεις χρήσεις) λειτουργεί κατά την αντίθετη κατεύθυνση, ομαδοποιώντας δηλαδή σημεία σε ολοένα και μεγαλύτερα clusters, ώσπου όλα να ενωθούν σε ένα και μοναδικό cluster. Οι agglomerative αλγόριθμοι διαιρούνται περαιτέρω ανάλογα με τη μετρική απόστασης που χρησιμοποιούν, π.χ. σε single linkage hierarchical clustering όπου χρησιμοποιείται η ελάχιστη απόσταση μεταξύ δύο σημείων ως απόσταση δύο clusters και σε average linkage hierarchical clustering, όπου χρησιμοποιείται η μέση απόσταση. Εδώ χρησιμοποιείται και πάλι η μετρική της σημασιολογικής απόστασης.

### 5.4.3. Spectral clustering

Το spectral clustering [65] είναι μια νεότερη μη επιβλεπόμενη τεχνική μάθησης, η οποία χρησιμοποιεί το συμμετρικό πίνακα ομοιότητας των σημείων (χρηστών) και τον αριθμό των clusters ως είσοδο. Ο πίνακας αυτός  $S = [s_{ij}]$  προκύπτει από τις τιμές της σημασιολογικής εγγύτητας  $s_{ij}$  ανάμεσα σε δυο προφίλ  $u_i$  και  $u_j$ .

Ένας διαισθητικός τρόπος αντίληψης του spectral clustering είναι η γραφοθεωρητική θεώρηση. Τα προς συσταδοποίηση σημεία (χρήστες) αναπαρίστανται ως κορυφές του γράφου και οι πλευρές του γράφου αναπαριστούν τη σύνδεσή τους με βάρη τις τιμές του μέτρου ομοιότητας μεταξύ των αντίστοιχων κορυφών. Το spectral clustering βελτιστοποιεί την τιμή του  $R$ -way normalized cut. Το normalized cut ισούται με το λόγο του αθροίσματος των βαρών των πλευρών που «φεύγουν» από το cluster προς το άθροισμα των συνολικών βαρών των κορυφών του cluster. Το  $R$ -way normalized cut δίνεται από τον τύπο:

$$C_R = \sum_{r=1}^R \frac{\sum_{i \in A_r, j \notin A_r} s_{ij}}{\sum_{i \in A_r, j \in V} s_{ij}} \quad (5.6)$$

όπου  $A_r$  είναι το  $r$ -στο cluster,  $V$  το σύνολο των κορυφών του γράφου και  $s_{ij}$  η σημασιολογική εγγύτητα των κορυφών  $i$  και  $j$ .

Έχουν προταθεί πολλές παραλλαγές αλγορίθμων, όπως οι [66], [67], [68]. Εδώ υιοθετείται ο [68] λόγω της αποδεδειγμένης αποτελεσματικότητάς του και της ευκολίας εφαρμογής του. Τα βήματα κατηγοριοποίησης των προφίλ σε ομάδες είναι εν συντομία:

- Υπολογίζουμε τον πίνακα:

$$\mathbf{P} = \mathbf{Z}^{-1} \cdot \mathbf{S} \quad (5.7)$$

όπου  $\mathbf{Z} = \text{diag}(z_i)$ ,  $z_i = \sum_{j=1}^N s_{ij}$ ,  $N$  ο συνολικός αριθμός χρηστών.

- Κατόπιν, υπολογίζουμε τα  $R$  πρώτα ιδιοδιανύσματα  $v^1, v^2, \dots, v^R$  του  $\mathbf{P}$ , όπου  $R$  είναι ο αριθμός των clusters.
- Τέλος, οι γραμμές του πίνακα  $\mathbf{V} = [v^1, v^2, \dots, v^R]$  συσταδοποιούνται σαν σημεία στον  $R$ -διάστατο χώρο, χρησιμοποιώντας fuzzy (ασαφή) c-means clustering.

## 5.5. Αξιολόγηση αλγορίθμων

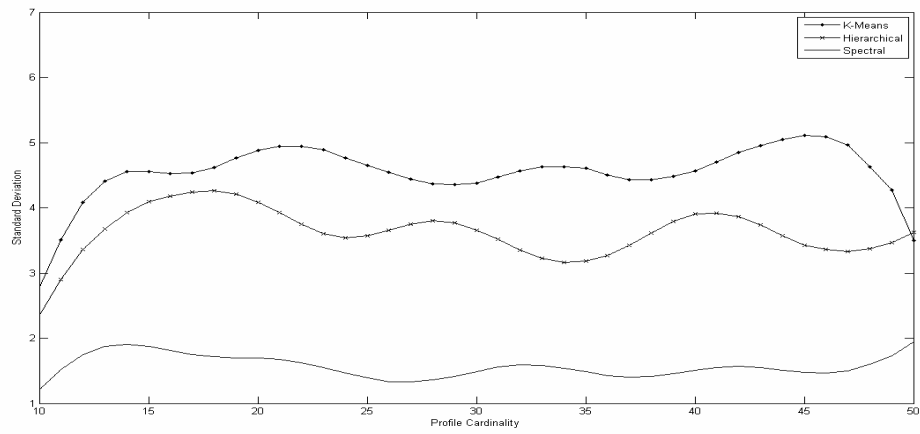
Στην υποενότητα αυτή θα γίνει αξιολόγηση της συμπεριφοράς του συστήματος στην κατηγοριοποίηση των χρηστών από διαφορετικούς αλγορίθμους συσταδοποίησης με διαφορετικά μεγέθη προφίλ και με διαφορετικές υποθέσεις κατανομής των προτιμήσεων (Zipf και ομοιόμορφη).

Η αξιολόγηση μιας υπηρεσίας social networking βέβαια δεν είναι ιδιαίτερα απλή υπόθεση, καθώς λόγω της φύσης της υπηρεσίας η αξιολόγησή της ενέχει μεγάλο βαθμό υποκειμενικότητας. Ωστόσο, προτείνεται μια μετρική κατά το δυνατόν αντικειμενικά η οποία θα χρησιμοποιηθεί στην εκτίμηση των αποτελεσμάτων, η *Μέση Σημασιολογική Εγγύτητα Συστήματος* (Average System Semantic Proximity  $\overline{P_S}$ ):

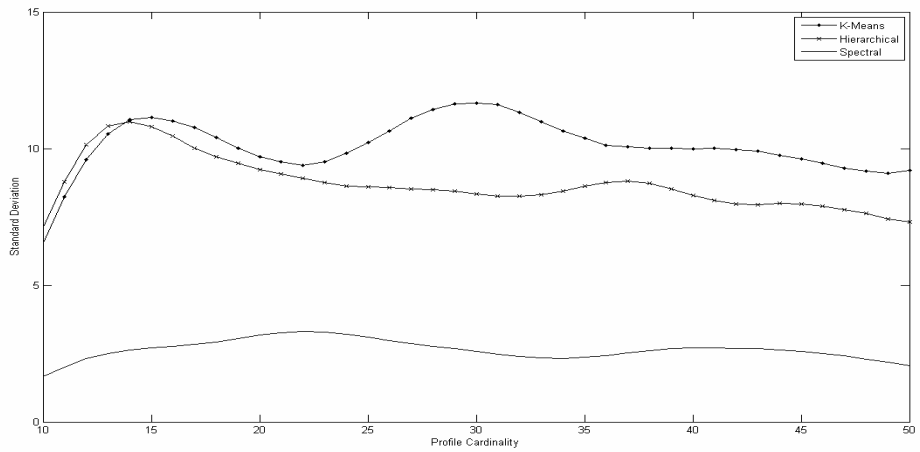
$$\overline{P}_S = \frac{1}{R} \sum_{k=1}^R \frac{1}{|A_k|} \sum_{i,j \in A_k} \frac{a_{ij}}{|u|} = \frac{1}{R} \sum_{k=1}^R \frac{1}{|A_k|} \sum_{i,j \in A_k} S_{ij} \quad (5.8)$$

όπου  $R$  ο αριθμός των clusters,  $a_{ij}$  ο αριθμός των κοινών στοιχείων των προφίλ  $i$  and  $j$ ,  $|u|$  το μέγεθος του προφίλ και  $|A_k|$  η πληθικότητα του  $k$ -στου cluster. Η  $\overline{P}_S$  παρέχει μια αντικειμενική ένδειξη για τη μέση τιμή για όλα τα clusters του μέσου αριθμού των κοινών στοιχείων που μοιράζεται κάθε χρήστης. Και πάλι χρησιμοποιείται μια κανονικοποιημένη μετρική (διαίρεση με το μέγεθος του προφίλ  $|u|$ ). Μια τέτοια μετρική είναι ενδεικτική της επίδοσης του συστήματος συνολικά και δεν υπόκειται στη γνώμη των χρηστών για την υπηρεσία. Η μετρική αυτή θα χρησιμοποιηθεί για την αξιολόγηση της επίδοσης των συγκρινόμενων αλγορίθμων συσταδοποίησης, αλλά και της επίδρασης του μεγέθους του προφίλ στην απόδοση.

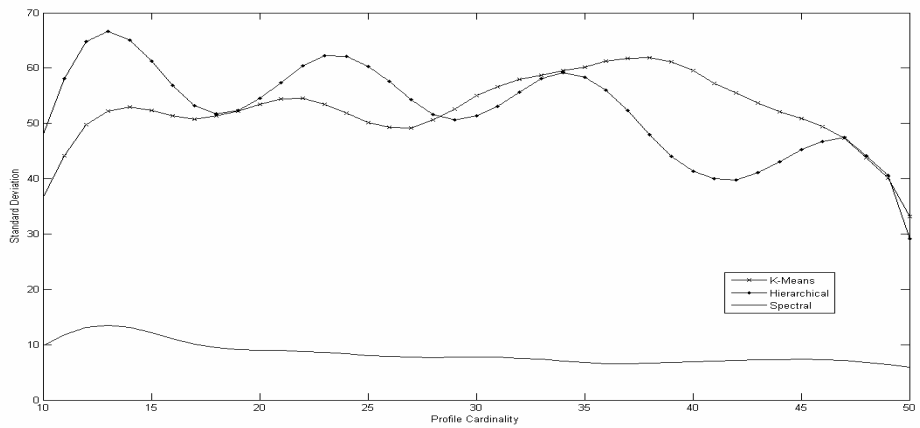
Πέραν της Μέσης Σημασιολογικής Εγγύτητας Συστήματος, ως μέτρα αξιολόγησης του συστήματος, θα χρησιμεύσει και η *τυπική απόκλιση* (*standard deviation*): όσο χαμηλότερη είναι η τυπική απόκλιση τόσο πιο κοντά είναι η πληθικότητα κάθε ομάδας στη μέση πληθικότητα (η παράμετρος αυτή ενδιαφέρει όταν είναι επιθυμητό οι δημιουργούμενες ομάδες να έχουν παρόμοια μεγέθη, κάτι που αποτελεί αίτημα σε αρκετές περιπτώσεις social networking). Μια ακόμα μετρική που χρησιμοποιήθηκε για την αξιολόγηση του συστήματος είναι η *normalized cut value* της προκύπτουσας συσταδοποίησης.



(α)



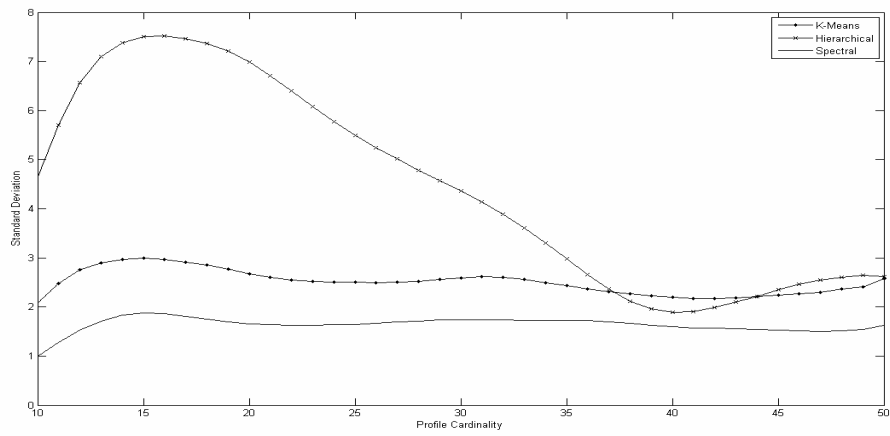
(β)



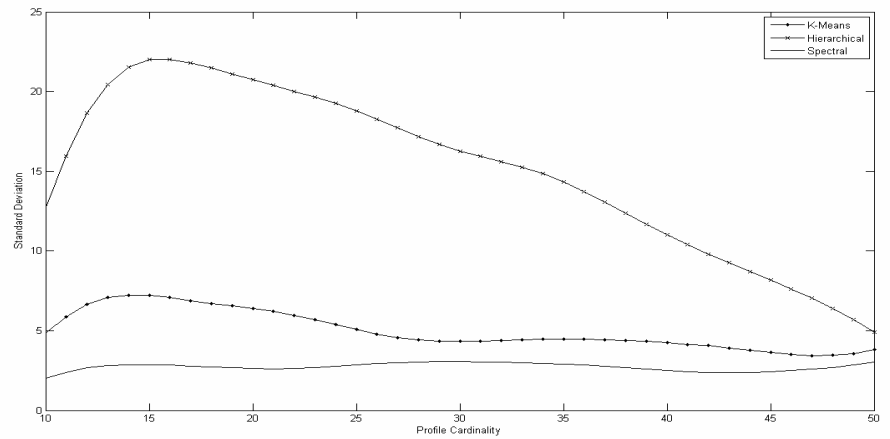
(γ)

**Σχήμα 5.1:** Τυπική απόκλιση συναρτήσει του μεγέθους προφίλ για συσταδοποίηση σε 10 clusters με τους τρεις αλγορίθμους για κατανομή προτιμήσεων Zipf και αριθμό χρηστών (α) 50, (β) 100 και (γ) 500.

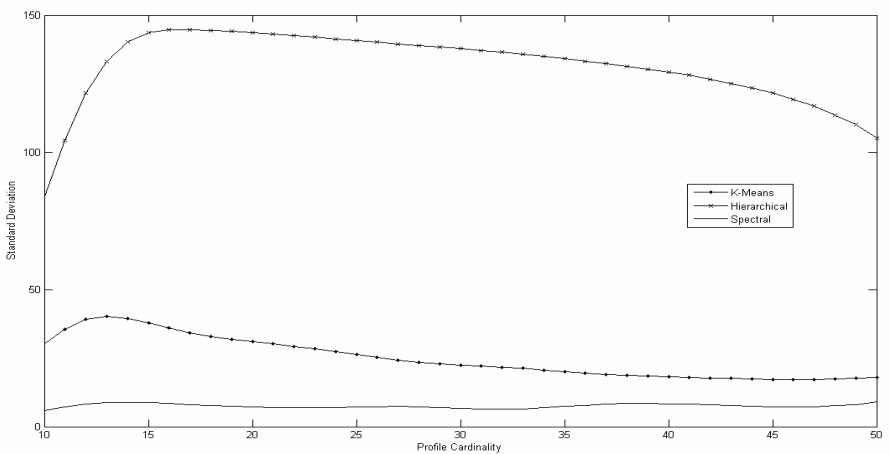




(α)



(β)



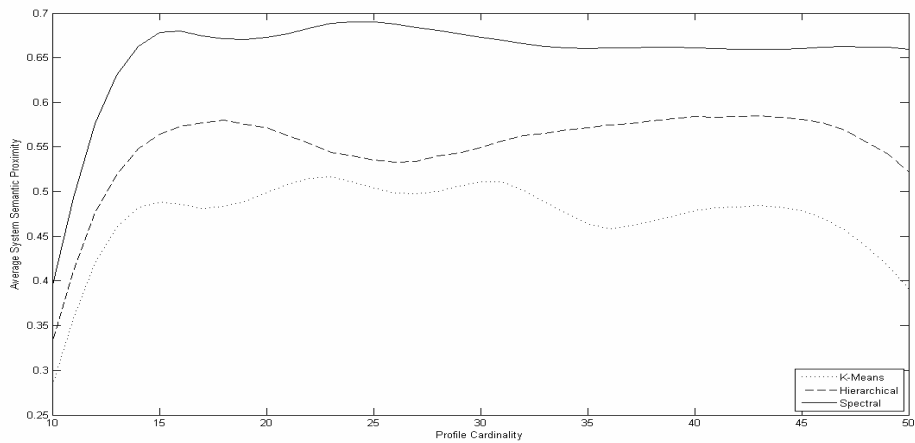
(γ)

**Σχήμα 5.2:** Τυπική απόκλιση συναρτήσεως του μεγέθους προφίλ για συσταδοποίηση σε 10 clusters με τους τρεις αλγορίθμους για ομοιόμορφη (uniform) κατανομή προτιμήσεων και αριθμό χρηστών (α) 50, (β) 100 και (γ) 500.

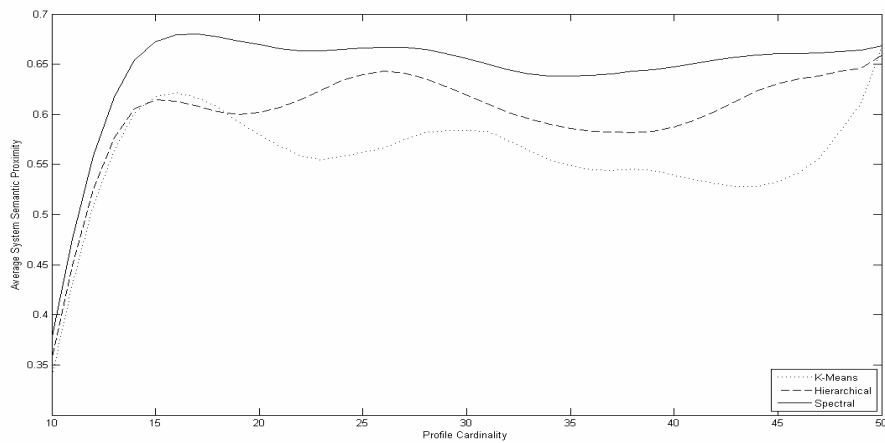
Στο σχήμα 5.1 φαίνονται οι τυπικές αποκλίσεις που προκύπτουν από την εφαρμογή αλγορίθμων k-means, hierarchical και spectral clustering σε σχέση με το μέγεθος του προφίλ, και υποθέτοντας κατανομή Zipf για τις λέξεις-κλειδιά των προφίλ. Έγιναν τρία πειράματα, για 50, 100 και 500 χρήστες. Το σχήμα 5.2 απεικονίζει τα αποτελέσματα του ίδιου πειράματος, υποθέτοντας όμως ομοιόμορφη κατανομή. Και τα δύο σχήματα καταδεικνύουν υπεροχή του spectral clustering, αφού χαρακτηρίζεται από αισθητά χαμηλότερη τυπική απόκλιση.

Τα σχήματα 5.1 και 5.2 παρέχουν επίσης μια πρώτη εικόνα σχετικά με το ρόλο του μεγέθους του προφίλ στην ποιότητα της συσταδοποίησης, ο οποίος φαίνεται να συνδέεται με το είδος της υποτιθέμενης κατανομής. Και τα τρία διαγράμματα στο σχήμα 5.1 δείχνουν ότι η τυπική απόκλιση παραμένει μάλλον ανεπηρέαστη από την αύξηση του μεγέθους του προφίλ, σε αντίθεση με το σχήμα 5.2, όπου οι αλγόριθμοι k-means και hierarchical clustering δίνουν χαμηλότερη τυπική απόκλιση όσο αυξάνεται το μέγεθος του προφίλ (αριθμός λέξεων-κλειδιών).

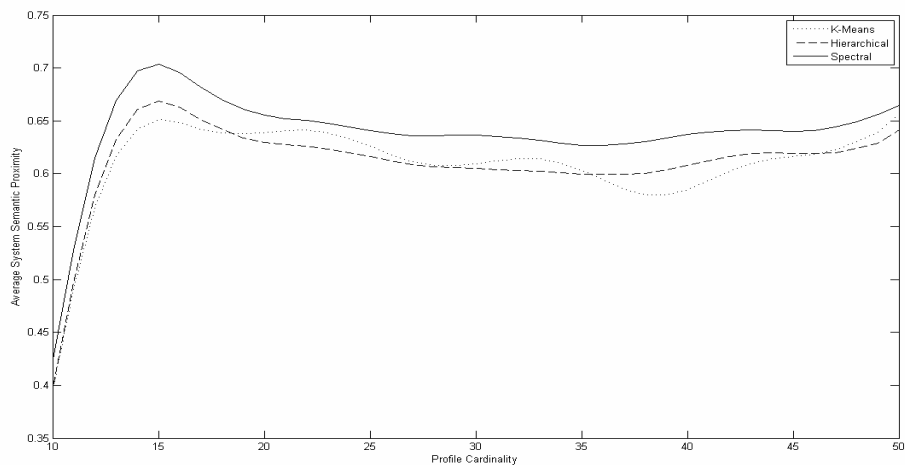
Τα επόμενα δύο σχήματα εστιάζουν στην αξιολόγηση της Μέσης Σημασιολογικής Εγγύτητας Συστήματος (Average System Semantic Proximity) σε σχέση με το μέγεθος προφίλ. Στο σχήμα 5.3 φαίνονται τα αποτελέσματα υποθέτοντας κατανομή προτιμήσεων Zipf και στο σχήμα 5.4 υποθέτοντας ομοιόμορφη κατανομή.



(α)

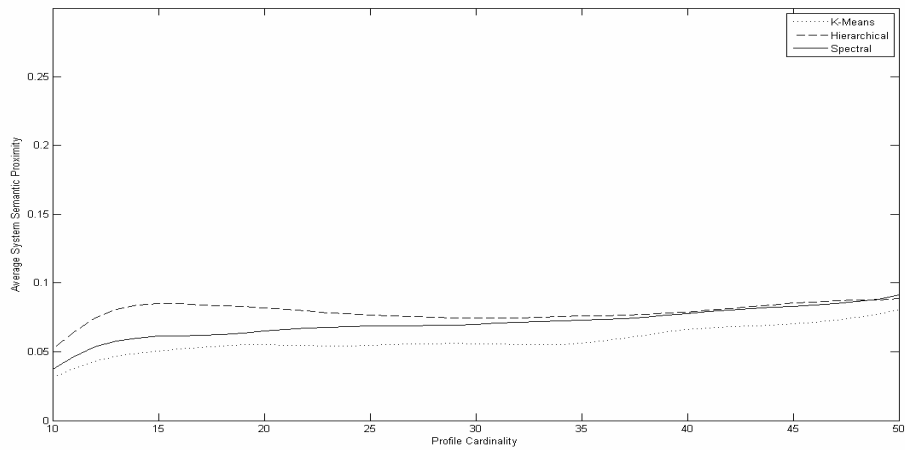


(β)

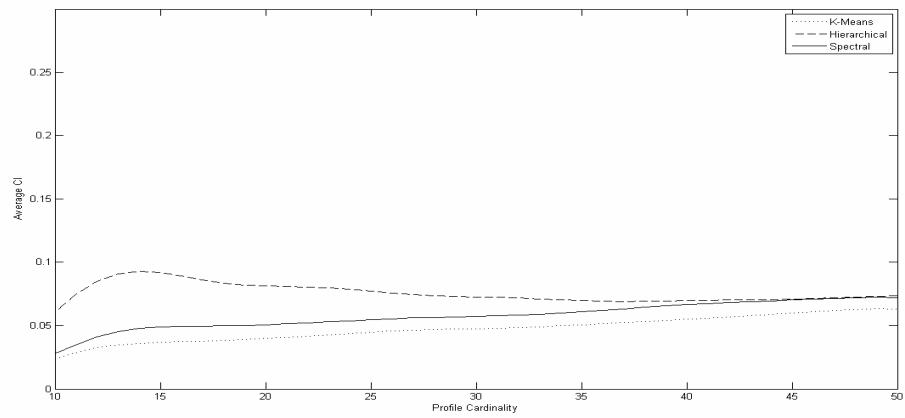


(γ)

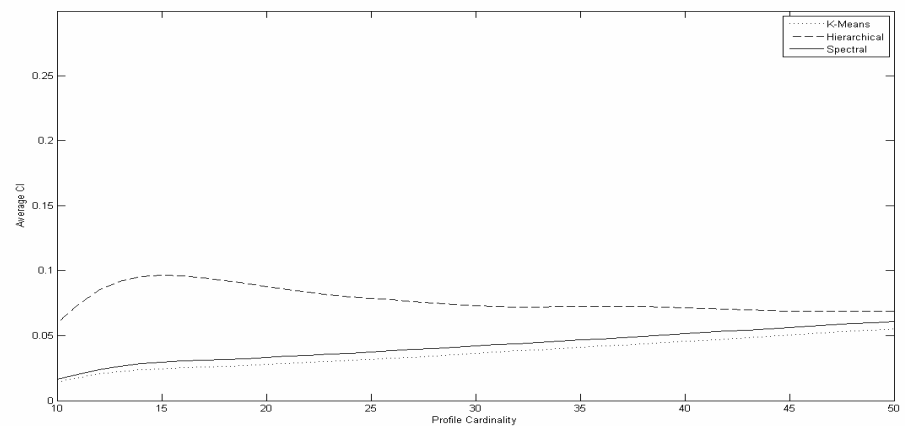
**Σχήμα 5.3:** Μέση Σημασιολογική Εγγύτητα Συστήματος συναρτήσει του μεγέθους προφίλ για συσταδοποίηση σε 10 clusters με τους τρεις αλγορίθμους για κατανομή προτιμήσεων Zipf και αριθμό χρηστών (α) 50, (β) 100 και (γ) 500.



(α)



(β)

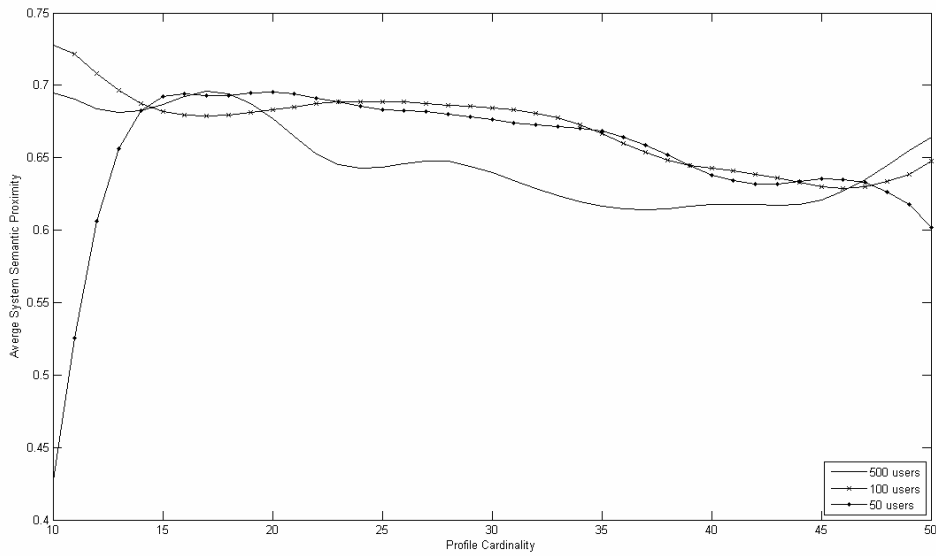


(γ)

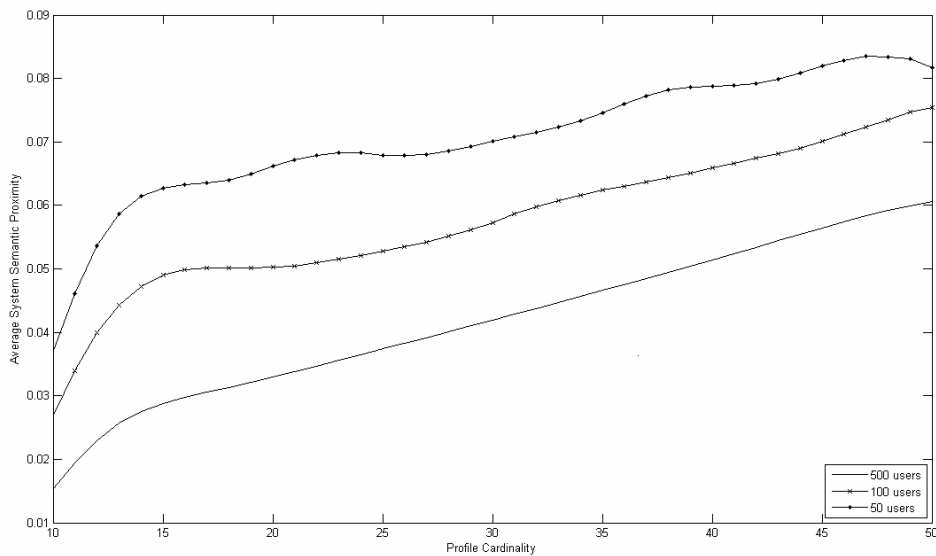
**Σχήμα 5.4:** – Μέση Σημασιολογική Εγγύτητα Συστήματος συναρτήσεως του μεγέθους προφίλ για συσταδοποίηση σε 10 clusters με τους τρεις αλγορίθμους για ομοιόμορφη (uniform) κατανομή προτιμήσεων και αριθμό χρηστών (α) 50, (β) 100 και (γ) 500.

Το σχήμα 5.3 καθιστά σαφές ότι το spectral clustering οδηγεί σε ομαδοποίηση καλύτερης ποιότητας, αφού για όλα τα μεγέθη προφίλ, τα μέλη των ομάδων που προκύπτουν με την εφαρμογή του spectral clustering παρουσιάζουν κατά μέσο όρο περισσότερα κοινά ενδιαφέροντα σε σύγκριση με τις ομαδοποιήσεις που προκύπτουν από τους άλλους δύο αλγορίθμους. Επίσης, παρατηρείται ότι όσο λιγότεροι είναι οι χρήστες, τόσο μεγαλύτερη είναι η διαφορά στην απόδοση μεταξύ του spectral clustering και των άλλων μεθόδων. Από το σχήμα 5.3 μπορούν ακόμα να εξαχθούν συμπεράσματα για τη σχέση απόδοσης και μεγέθους (πληθικότητας) προφίλ, από τη σκοπιά του  $\overline{P}_s$ : όταν η συχνότητα εμφάνισης των λέξεων-κλειδιών στα προφίλ ακολουθεί κατανομή Zipf, η αύξηση του μεγέθους του προφίλ (του αριθμού των λέξεων-κλειδιών που το απαρτίζουν) δεν συνεπάγεται απαραίτητα βελτίωση της απόδοσης (δηλαδή του αμέσου αριθμού κοινών λέξεων-κλειδιών στις ομάδες). Η παρατήρηση αυτή σχετίζεται με τα ειδικά χαρακτηριστικά της κατανομής Zipf. Εφόσον ένας μικρός αριθμός λέξεων χρησιμοποιείται πολύ συχνά και ένας μεγάλος αριθμός σπάνια, το ποσοστό των λέξεων-κλειδιών που θα είναι κοινές σε μια ομάδα θα παραμένει λίγο-πολύ σταθερό, ενώ η αύξηση στο μέγεθος του προφίλ απλώς θα αυξήσει την πιθανότητα εμφάνισης των σπάνιων λέξεων, χωρίς να προκαλεί αξιοσημείωτη μεταβολή (ίσως ελαφρά μείωση) στο ποσοστό κοινών λέξεων.

Στο σχήμα 5.4 φαίνονται τα αποτελέσματα με υπόθεση ομοιόμορφης (uniform) κατανομής. Εδώ παρατηρεί κανείς ότι οι διαφορές στην απόδοση των τριών αλγορίθμων είναι πρακτικά αμελητέες. Οι τρεις μέθοδοι δημιουργούν ομάδες χρηστών με παρόμοια ποσοστά κοινών ενδιαφερόντων, με ελαφρά καλύτερες τιμές αυτές του hierarchical clustering. Ωστόσο για μεγέθη προφίλ μεγαλύτερα από 30 λέξεις, οι διαφορές πρακτικά δεν μπορούν να εντοπιστούν. Μια άλλη παρατήρηση που μπορεί να γίνει είναι ότι, σε αντίθεση με την περίπτωση Zipf κατανομής, στην ομοιόμορφη κατανομή η αύξηση του μεγέθους (πληθικότητας) του προφίλ συνεπάγεται βελτίωση της ποιότητας της συσταδοποίησης (από πλευράς  $\overline{P}_s$ ). Παρότι η αύξηση φαίνεται μικρή, αν λάβει κανείς υπόψη και τα αποτελέσματα του σχήματος 5.3, καθίσταται σαφές ότι η κατανομή που ακολουθούν οι προτιμήσεις των χρηστών διαδραματίζει σημαντικό ρόλο στη συνολική συμπεριφορά του συστήματος.



(α)



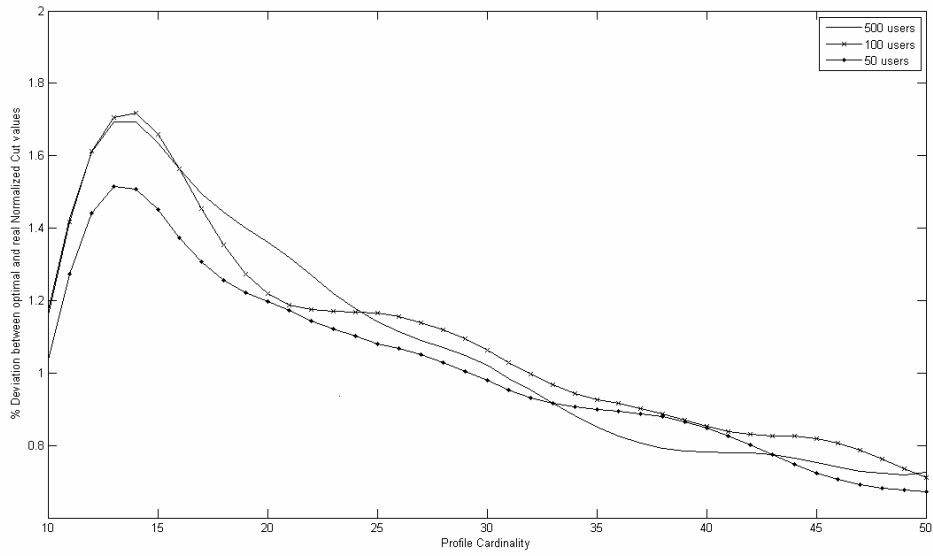
(β)

**Σχήμα 5.5:** Μέση Σημασιολογική Εγγύτητα Συστήματος συναρτήσει του μεγέθους προφίλ για συσταδοποίηση για (α) Zipf και (β) ομοιόμορφη (uniform) κατανομή προτιμήσεων κατόπιν εφαρμογής μεθόδου spectral clustering

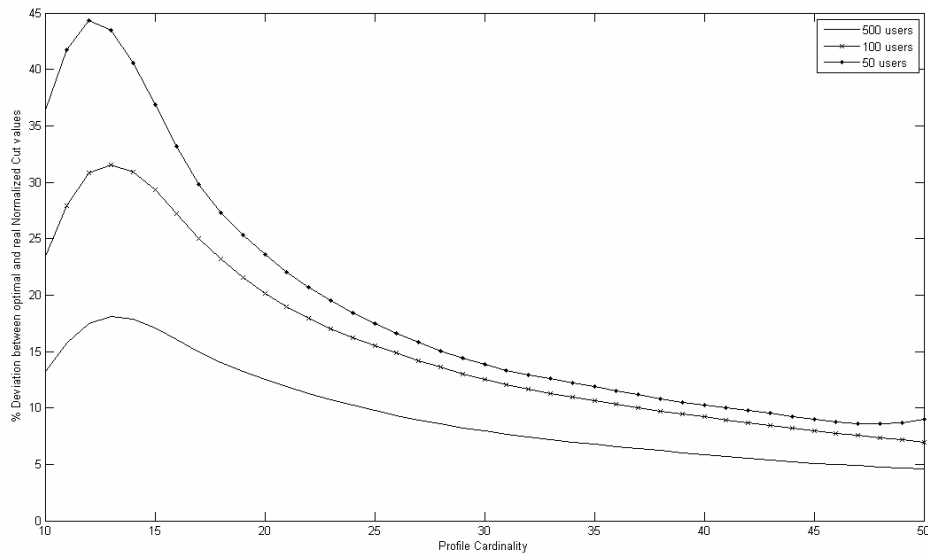
Κατόπιν μελετάται η ποιότητα συσταδοποίησης στην οποία οδηγεί ο αλγόριθμος spectral clustering (ο οποίος φαίνεται να υπερέχει των άλλων) σε σχέση με το μέγεθος του προφίλ, από τη σκοπιά του normalized cut value. Η μεθοδολογία βασίζεται στη σύγκριση του θεωρητικά βέλτιστου normalized cut value με τα πειραματικά αποτελέσματα. Στο [69], οι συγγραφείς δείχνουν ότι το βέλτιστο normalized cut value επιτυγχάνεται όταν ο πίνακας  $P$  έχει ιδιοδιανύσματα, καθένα από στοιχεία των οποίων θα έχει μια συγκεκριμένη τιμή που θα αντιστοιχεί σε ένα συγκεκριμένο cluster. Στην περίπτωση αυτή, το spectral clustering δίνει τη βέλτιστη τιμή για το normalized cut, η οποία ισούται με:

$$C_R^{opt} = R - \sum_{i=1}^R \lambda_i \quad (5.9)$$

όπου  $R$  ο αριθμός των clusters και  $\lambda_i, \quad i = 1, 2, \dots, R$  είναι οι  $R$  μεγαλύτερες ιδιοτιμές του πίνακα  $P$ . Σε λιγότερο ιδανικές καταστάσεις, η πραγματική τιμή του normalized cut αποκλίνει από το  $C_R^{opt}$ .



(α)



(β)

**Σχήμα 5.6:** Απόκλιση της πραγματικής τιμής normalized cut από τη βέλτιστη συναρτήσει του μεγέθους προφίλ για (α) Zipf και (β) ομοιόμορφη (uniform) κατανομή προτιμήσεων



Στο σχήμα 5.6 παρουσιάζονται τα αποτελέσματα ενός πειράματος που αξιολογεί την επίδραση του μεγέθους προφίλ στην ποιότητα της συσταδοποίησης που παρέχει το spectral clustering υποθέτοντας πάλι (α) Zipf και (β) ομοιόμορφη κατανομή για 50/100/500 χρήστες και 10 clusters. Ο οριζόντιος άξονας αναπαριστά το μέγεθος προφίλ σε λέξεις και ο κατακόρυφος την ποσοστιαία απόκλιση της πραγματικής τιμής του normalized cut  $C_R^{real}$  από το  $C_R^{opt}$ , ήτοι την τιμή  $|C_R^{real} - C_R^{opt}| / C_R^{opt}$ . Τα αποτελέσματα επιβεβαιώνουν προηγούμενα συμπεράσματα: στην περίπτωση της κατανομής Zipf, η αύξηση του μεγέθους προφίλ συνεπάγεται πολύ μικρές αποκλίσεις απόδοσης, ενώ στην περίπτωση ομοιόμορφης κατανομής, η απόκλιση είναι σημαντικά μεγαλύτερη. Παρότι τα αποτελέσματα κατά την εξέταση του  $\bar{P}_s$  δεν καταδεικνύουν τόσο μεγάλη επίδραση, το σχήμα 5.6 παρέχει μια ακόμα ένδειξη του συσχετισμού της ποιότητας συσταδοποίησης με το μέγεθος του προφίλ στην περίπτωση ομοιόμορφης κατανομής προτιμήσεων των χρηστών.

## 5.6. Συμπεράσματα

Συμπερασματικά, μπορεί κανείς να πει ότι το spectral clustering οδηγεί σε καλύτερη ομαδοποίηση των χρηστών από την άποψη μεγαλύτερου ποσοστού κοινών ενδιαφερόντων μεταξύ των μελών μιας ομάδας (αλλά και από την άποψη δημιουργίας ομάδων παρόμοιων μεγεθών, κάτι που είναι χρήσιμο σε ορισμένες υπηρεσίες social networking). Επιπλέον, κατέστη σαφές ότι η επίδραση του μεγέθους προφίλ στην ποιότητα της ομαδοποίησης εξαρτάται σε μεγάλο βαθμό από την κατανομή που ακολουθούν οι προτιμήσεις των χρηστών (λέξεις-κλειδιά στα προφίλ τους). Αναφορικά με την εφαρμογή του μοντέλου στην παροχή εξατομικευμένων υπηρεσιών με προστασία ιδιωτικότητας και εξασφάλιση ποιότητας υπηρεσίας, δηλαδή στο πλαίσιο που περιγράψαμε στην ενότητα 4, μπορούμε να πούμε τα εξής: συγκεντρώνοντας (π.χ. μέσω ερωτηματολογίων) τις προτιμήσεις ενός μεγάλου αριθμού ανθρώπων σχετικά με ένα συγκεκριμένο τομέα, π.χ. αγαπημένες ταινίες ή είδη ταινιών (διαδικασία που είναι, μάλιστα, σε εξέλιξη στα πλαίσια της ανάπτυξης της πλατφόρμας PLASMA) είμαστε σε θέση να συντάξουμε τα προφίλ των ανθρώπων αυτών κατά την έννοια που περιγράφηκε ανωτέρω αναφορικά με μια συγκεκριμένη παράμετρο προτιμήσεων (στο εν λόγω

παράδειγμα, των κινηματογραφικών προτιμήσεων). Κατόπιν, με την επαναληπτική εφαρμογή μεθόδων clustering (με όποια κρίνεται καταλληλότερη για την περίπτωση – τα περιγραφέντα πειράματα πάντως κατέδειξαν μια αισθητή υπεροχή του spectral clustering), λαμβάνουμε ως αποτέλεσμα ομαδοποιήσεις των προτιμήσεων / χρηστών σε διαφορετικούς αριθμούς ομάδων. Έχουμε, έτσι, την οργάνωση των πιθανών προτιμήσεων των χρηστών για μια προσωπική παράμετρο σε μια δομή με διαφορετικά επίπεδα ακρίβειας. Η διαδικασία αυτή επαναλαμβάνεται για κάθε προσωπική παράμετρο, ώστε τελικά η οργανωμένη αυτή δομή να υπάρχει για κάθε πιθανή προσωπική παράμετρο, η οποία ενδέχεται να εμπλέκεται στην παροχή κάποιας υπηρεσίας.

## 6. VIRTUAL EARTH TILE SYSTEM

Είναι σαφές ότι η ύπαρξη ενός συστήματος χαρτών αποτελεί απαραίτητη προϋπόθεση για την ολοκληρωμένη παροχή εξατομικευμένων, location based υπηρεσιών. Υπάρχουν αρκετά τέτοια συστήματα, όπως το Windows Live Local (a.k.a. Virtual Earth) [44], το Google Maps [45], το Yahoo! Maps [46], κ.α. Για την ανάπτυξη της πλατφόρμας PLASMA, χρησιμοποιούμε τον χάρτη του Virtual Earth, το οποίο διαθέτει αρκετά χαρακτηριστικά χρήσιμα για την ανάπτυξη εφαρμογών. Στη συνέχεια θα περιγράψουμε τον τρόπο δημιουργίας του Virtual Earth, καθώς και το σύστημα οργάνωσης των χαρτών του, το Virtual Earth Tile System. Η περιγραφή αυτή βασίζεται σε σχετικά ενημερωτικά άρθρα της Microsoft [48].

### 6.1. Προβολή χάρτη (Map projection)

Η ανάγκη να είναι ο χάρτης ενιαίος και «αρραγής» (seamless) και να εξασφαλιστεί ότι οι αεροφωτογραφίες που προέρχονται από διαφορετικές πηγές στοιχίζονται σωστά, οδηγεί στη χρήση μίας και μοναδικής προβολής για ολόκληρο τον πλανήτη. Η προβολή που επελέγη είναι η Μερκατορική Προβολή (Mercator projection), η οποία φαίνεται στο παρακάτω σχήμα:



**Σχήμα 6.1:** Μερκατορική προβολή του παγκόσμιου χάρτη (πηγή [48])

Μολονότι η Μερκατορική προβολή παραμορφώνει την κλίμακα και το εμβαδό (κυρίως κοντά στους πόλους), έχει δυο σημαντικές ιδιότητες, οι οποίες υπερκαλύπτουν την παραμόρφωση κλίμακας:

1. Είναι μια *σύμμορφη* (conformal) προβολή, που σημαίνει ότι διατηρεί το σχήμα των σχετικά μικρών αντικειμένων. Αυτό αποκτά ιδιαίτερη αξία όταν γίνεται απεικόνιση αεροφωτογραφιών (aerial), αφού πρέπει να αποφευχθεί η παραμόρφωση του σχήματος των κτιρίων. Τα τετραγωνικά κτίρια πρέπει να φαίνονται τετραγωνικά και όχι ορθογώνια παραλληλόγραμμα.
2. Είναι μια *κυλινδρική* (cylindrical) προβολή, που σημαίνει ότι ο βορράς και ο νότος απεικονίζονται πάντα πάνω και κάτω και η δύση και η ανατολή απεικονίζονται αριστερά και δεξιά αντίστοιχα.

Η Μερκατορική Προβολή απειρίζεται στους πόλους, επομένως στην πραγματικότητα δεν απεικονίζει ολόκληρο τον κόσμο. Χρησιμοποιώντας τετραγωνικό λόγο διαστάσεων (square aspect ratio) για το χάρτη, το μέγιστο γεωγραφικό πλάτος που απεικονίζεται είναι περίπου οι 85 μοίρες και 5 λεπτά.

Προς απλοποίηση των υπολογισμών, χρησιμοποιήθηκε η σφαιρική μορφή της προβολής και όχι η ελλειψοειδής. Αφού η προβολή χρησιμοποιείται από το Virtual Earth μόνο για απεικόνιση χαρτών και όχι για απεικόνιση αριθμητικών συντεταγμένων, θεωρήθηκε περιττή η επιπρόσθετη ακρίβεια της ελλειψοειδούς προβολής. Η σφαιρική προβολή προκαλεί παραμόρφωση της κλίμακας του άξονα Y της τάξης του 0.33%, η οποία δεν γίνεται οπτικά αντιληπτή.

## **6.2. Εδαφική ανάλυση (Ground resolution) και Κλίμακα χάρτη (Map scale)**

Για τον πλήρη «ορισμό» ενός χάρτη, εκτός από τον τύπο της προβολής πρέπει να προσδιοριστεί και η εδαφική ανάλυση ή η κλίμακα του χάρτη. Στο χαμηλότερο επίπεδο (level of detail) ακρίβειας / λεπτομέρειας (Επίπεδο 1), ο χάρτης έχει μέγεθος 512 × 512 pixels. Σε κάθε επόμενο επίπεδο ακρίβειας, το πλάτος και το ύψος του χάρτη μεγαλώνουν

κατά 2 φορές: στο Επίπεδο 2, το μέγεθος του χάρτη είναι  $1024 \times 1024$  pixels, στο Επίπεδο 3 είναι  $2048 \times 2048$  pixels, στο Επίπεδο 4 είναι  $4096 \times 4096$  pixels, κ.ο.κ. Σε κάθε περίπτωση το πλάτος (width) και το ύψος (height) του χάρτη (σε pixels) υπολογίζεται από τον τύπο:

$$\text{map width} = \text{map height} = 256 * 2^{\text{level}} \text{ pixels} \quad (6.1)$$

Η *εδαφική ανάλυση* (ground resolution) είναι η απόσταση στο έδαφος η οποία αναπαρίσταται από 1 pixel στο χάρτη. Για παράδειγμα, με εδαφική ανάλυση 10meters/pixel, κάθε pixel αναπαριστά απόσταση στο έδαφος ίση με 10 μέτρα. Η εδαφική ανάλυση μεταβάλλεται με το επίπεδο ακρίβειας και το γεωγραφικό πλάτος στο οποίο μετράται. Θεωρώντας την ακτίνα της Γης ίση με 6378137 μέτρα, η εδαφική ανάλυση (ground resolution) (σε μέτρα ανά pixel) υπολογίζεται από τον τύπο:

$$\begin{aligned} \text{ground resolution} &= \cos(\text{latitude} * \pi/180) * \text{earth circumference} / \text{map width} \\ &= (\cos(\text{latitude} * \pi/180) * 2 * \pi * 6378137 \text{ meters}) / (256 * 2^{\text{level}} \text{ pixels}) \quad (6.2) \end{aligned}$$

Η *κλίμακα χάρτη* (map scale) δείχνει την αναλογία ανάμεσα στην απόσταση στο χάρτη και την απόσταση στο έδαφος, μετρημένες με τις ίδιες μονάδες. Για παράδειγμα, σε κλίμακα 1 : 100000, μια απόσταση μίας ίντσας στο χάρτη αναπαριστά απόσταση 100000 ιντσών στο έδαφος. Όπως και η εδαφική ανάλυση, η κλίμακα μεταβάλλεται με το επίπεδο ακρίβειας και το γεωγραφικό πλάτος της μέτρησης. Υπολογίζεται με βάση την εδαφική ανάλυση ως ακολούθως, έχοντας δεδομένη την ανάλυση της οθόνης σε dots per inch (συνήθως 96 dpi):

$$\begin{aligned} \text{map scale} &= 1 : \text{ground resolution} * \text{screen dpi} / 0.0254 \text{ meters/inch} \\ &= 1 : (\cos(\text{latitude} * \pi/180) * 2 * \pi * 6378137 * \text{screen dpi}) / (256 * 2^{\text{level}} * 0.0254) \quad (6.3) \end{aligned}$$

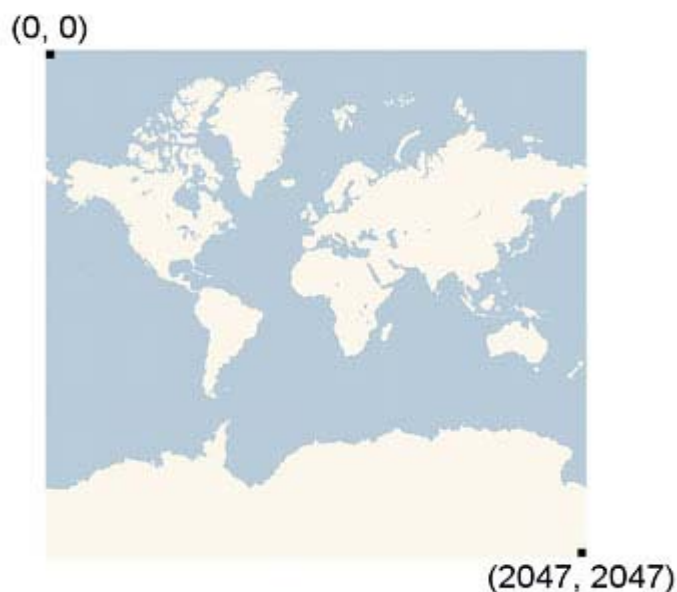
Στον παρακάτω πίνακα φαίνονται οι τιμές των παραπάνω μεγεθών σε κάθε επίπεδο ακρίβειας, όπως μετρήθηκαν στον ισημερινό (όπως αναφέρθηκε, η εδαφική ανάλυση και η κλίμακα εξαρτώνται και από το γεωγραφικό πλάτος).

Level of Detail	Map Width and Height (pixels)	Ground Resolution (meters / pixel)	Map Scale (at 96 dpi)
1	512	78,271.5170	1 : 295,829,355.45
2	1,024	39,135.7585	1 : 147,914,677.73
3	2,048	19,567.8792	1 : 73,957,338.86
4	4,096	9,783.9396	1 : 36,978,669.43
5	8,192	4,891.9698	1 : 18,489,334.72
6	16,384	2,445.9849	1 : 9,244,667.36
7	32,768	1,222.9925	1 : 4,622,333.68
8	65,536	611.4962	1 : 2,311,166.84
9	131,072	305.7481	1 : 1,155,583.42
10	262,144	152.8741	1 : 577,791.71
11	524,288	76.4370	1 : 288,895.85
12	1,048,576	38.2185	1 : 144,447.93
13	2,097,152	19.1093	1 : 72,223.96
14	4,194,304	9.5546	1 : 36,111.98
15	8,388,608	4.7773	1 : 18,055.99
16	16,777,216	2.3887	1 : 9,028.00
17	33,554,432	1.1943	1 : 4,514.00
18	67,108,864	0.5972	1 : 2,257.00
19	134,217,728	0.2986	1 : 1,128.50
20	268,435,456	0.1493	1 : 564.25
21	536,870,912	0.0746	1 : 282.12
22	1,073,741,824	0.0373	1 : 141.06
23	2,147,483,648	0.0187	1 : 70.53

**Πίνακας 6.1:** Διαστάσεις χάρτη, κλίμακα και εδαφική ανάλυση στα διαφορετικά επίπεδα ακρίβειας του Virtual Earth (πηγή [48])

### 6.3. Συντεταγμένες Pixel

Με δεδομένη την επιλεγείσα προβολή και την κλίμακα του χάρτη, μπορούν να μετατραπούν οι γεωγραφικές συντεταγμένες σε συντεταγμένες pixel (pixel coordinates). Αφού οι διαστάσεις του χάρτη είναι διαφορετικές σε κάθε επίπεδο, το ίδιο θα ισχύει και για τις συντεταγμένες pixel. Το pixel στην πάνω-αριστερά γωνία του χάρτη έχει πάντα συντεταγμένες pixel (0, 0). Το pixel στην κάτω-δεξιά γωνία έχει συντεταγμένες pixel (width-1, height-1), ή, χρησιμοποιώντας την (6.1),  $(256 * 2^{\text{level}} - 1, 256 * 2^{\text{level}} - 1)$ . Για παράδειγμα, στο Επίπεδο 3, οι συντεταγμένες pixel κυμαίνονται από (0, 0) έως (2047, 2047), όπως φαίνεται στο σχήμα 6.2:



**Σχήμα 6.2:** Συντεταγμένες pixel στον παγκόσμιο χάρτη Επιπέδου 3 (πηγή [48])

Δεδομένων του γεωγραφικού πλάτους και του γεωγραφικού μήκους σε μοίρες και του επιπέδου ακρίβειας, οι συντεταγμένες pixel, pixelX και pixelY, υπολογίζονται από τους τύπους:

$$\sinLatitude = \sin(\text{latitude} * \pi/180) \quad (6.4)$$

$$\text{pixelX} = ((\text{longitude} + 180) / 360) * 256 * 2^{\text{level}} \quad (6.5\alpha)$$

$$\text{pixelY} = (0.5 - \log((1 + \sinLatitude) / (1 - \sinLatitude)) / (4 * \pi)) * 256 * 2^{\text{level}} \quad (6.5\beta)$$

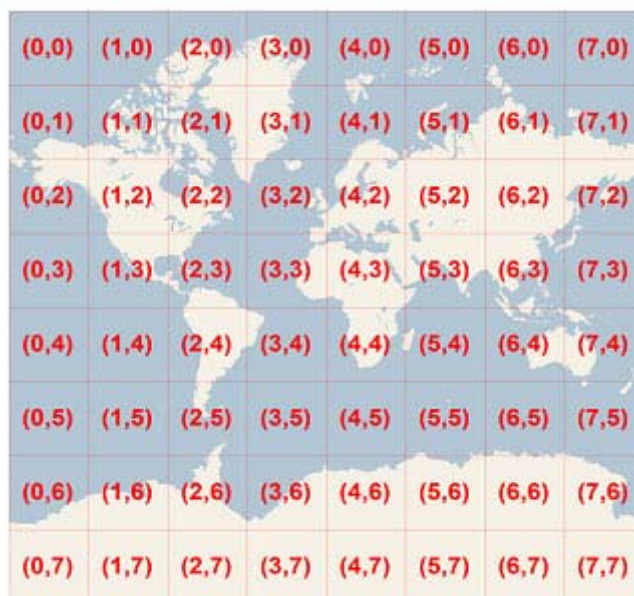
Το γεωγραφικό πλάτος και μήκος βρίσκονται σε συμφωνία με το WGS 84 (World Geodetic System - Παγκόσμιο Γεωδαιτικό Σύστημα Αναφοράς). Παρότι γίνεται χρήση σφαιρικής προβολής, οι γεωγραφικές συντεταγμένες πρέπει να μετατραπούν με βάση κάποια κοινή βάση και ως τέτοια επελέγη το WGS 84. Το γεωγραφικό μήκος κυμαίνεται από -180 έως +180 μοίρες, και το γεωγραφικό πλάτος προσαρμόζεται ώστε να κυμαίνεται από -85.05112878 ως 85.05112878 μοίρες. Αυτό συμβάλλει στην αποφυγή ανωμαλιών στους πόλους και στο τετραγωνικό σχήμα του προβαλλόμενου χάρτη.

#### 6.4. Συντεταγμένες Tile και Quadkeys

Για την αποτελεσματικότερη ανάκτηση και εμφάνιση του χάρτη, ο χάρτης που έχει κατά τα ως άνω δημιουργηθεί, «τεμαχίζεται» σε τετραγωνικά «τεμάχια»-«πλακίδια» (tiles) διαστάσεων  $256 \times 256$  pixels το καθένα. Αφού ο αριθμός των pixels είναι διαφορετικός σε κάθε επίπεδο ακρίβειας, θα είναι διαφορετικός και ο αριθμός των tiles:

$$\text{map width} = \text{map height} = 2^{\text{level}} \text{ tiles} \quad (6.6)$$

Σε κάθε tile αποδίδονται συντεταγμένες tile (tile coordinates) XY από (0, 0) στο πάνω-αριστερά, μέχρι  $(2^{\text{level}}-1, 2^{\text{level}}-1)$  στο κάτω-δεξιά. Για παράδειγμα, στο Επίπεδο 3, οι συντεταγμένες tile κυμαίνονται από (0, 0) έως (7, 7), ως εξής:



**Σχήμα 6.3:** Συντεταγμένες tile στον παγκόσμιο χάρτη Επιπέδου 3 (πηγή [48])



Με δεδομένο ένα ζεύγος συντεταγμένων pixel  $XY$ , μπορούν εύκολα να προσδιοριστούν οι συντεταγμένες tile  $XY$  του tile που περιέχει το συγκεκριμένο pixel:

$$\text{tileX} = \text{floor}(\text{pixelX} / 256) \quad (6.7\alpha)$$

$$\text{tileY} = \text{floor}(\text{pixelY} / 256) \quad (6.7\beta)$$

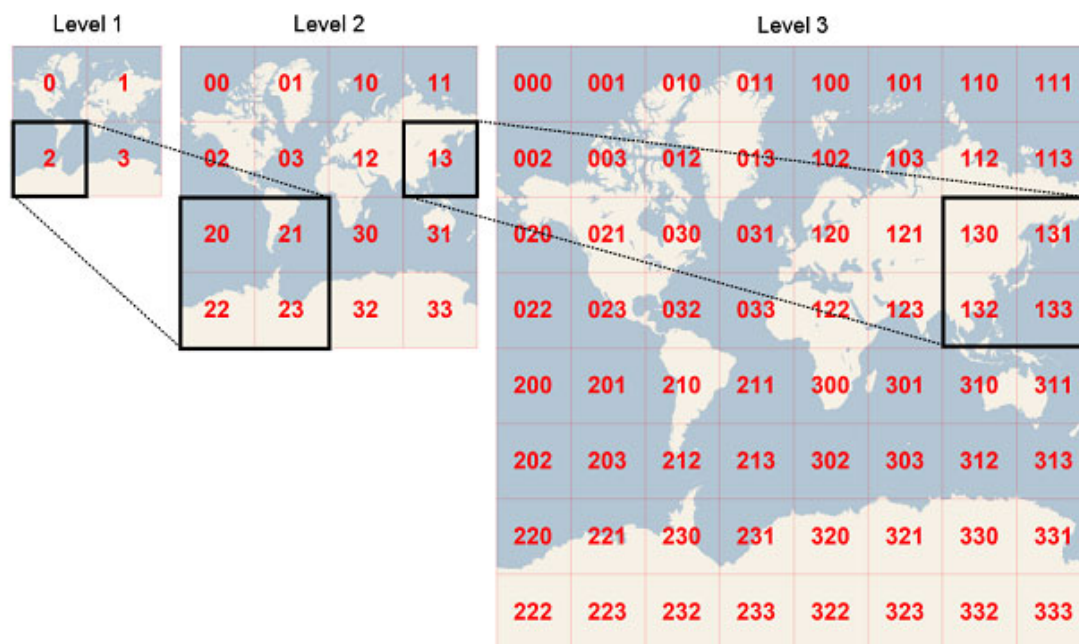
Για τη βελτιστοποίηση της δεικτοδότησης και δημιουργίας ευρετηρίου (indexing) και της αποθήκευσης των tiles, οι διδιάστατες συντεταγμένες tile  $XY$  (tile  $XY$  coordinates) συνδυάζονται σε μονοδιάστατες συμβολοσειρές (strings) που ονομάζονται quadtree keys, ή “quadkeys” για συντομία. Κάθε quadkey αντιστοιχεί μονοσήμαντα σε ένα μοναδικό tile σε ένα συγκεκριμένο επίπεδο ακρίβειας και μπορεί να χρησιμοποιηθεί σαν κλειδί σε ευρετήρια B-δέντρων για βάσεις δεδομένων. Για τη μετατροπή των συντεταγμένων tile σε quadkey, τα bits των  $Y$  και  $X$  συντεταγμένων διεμπλέκονται και το αποτέλεσμα ερμηνεύεται σαν ένας αριθμός βάσης 4 (base4) διατηρώντας τα αρχικά μηδενικά, και μετατρέπεται σε string. Για παράδειγμα, με δεδομένες συντεταγμένες tile  $XY$  τις (3, 5) στο Επίπεδο 3, το quadkey υπολογίζεται ως εξής:

$$\text{tileX} = 3 = 011_2$$

$$\text{tileY} = 5 = 101_2$$

$$\text{quadkey} = 100111_2 = 213_4 = \text{“213”}$$

Τα quadkeys έχουν πολλές ενδιαφέρουσες ιδιότητες. Πρώτον, το μήκος του quadkey (ο αριθμός των ψηφίων που το αποτελούν) ισούται με το επίπεδο ακρίβειας του αντίστοιχου tile. Δεύτερον, το quadkey κάθε tile ξεκινά με το quadkey του tile-γονέα του (δηλαδή εκείνο που περιέχει το παρόν tile στο προηγούμενο επίπεδο). Όπως φαίνεται στο ακόλουθο παράδειγμα, το tile 2 είναι ο γονέας των tiles 20 έως 23, και το tile 13 είναι ο γονέας των tiles 130 έως 133:



**Σχήμα 6.4:** Σχέση tiles γονέων-παιδιών και quadkeys (πηγή [48])

Τέλος, τα quadkeys παρέχουν ένα μονοδιάστατο κλειδί ευρετηρίου, το οποίο συνήθως διατηρεί την εγγύτητα των tiles στο χώρο XY. Με άλλα λόγια, δύο tiles που έχουν κοντινές συντεταγμένες tile XY συνήθως έχουν quadkeys που είναι αρκετά κοντά μεταξύ τους. Το χαρακτηριστικό αυτό είναι σημαντικό για την απόδοση της βάσης δεδομένων, μια και γειτονικά tiles ζητούνται συνήθως κατά ομάδες και έτσι είναι επιθυμητό τα tiles αυτά να βρίσκονται στο ίδιο μπλοκ του δίσκου, ώστε να ελαχιστοποιείται ο αριθμός των προσπελάσεων στο δίσκο.

## 6.5. Virtual Earth Tile Servers

Τα tiles είναι αποθηκευμένα σε τέσσερις servers, τους Virtual Earth Tile Servers (υπάρχει αντίτυπο κάθε tile και στους τέσσερις) και είναι προσβάσιμα μέσω των URL της μορφής:

[http://r\[S\].ortho.tiles.virtualearth.net/tiles/r\[quadkey\].png?g=1](http://r[S].ortho.tiles.virtualearth.net/tiles/r[quadkey].png?g=1),

όπου [S] = 0, 1, 2 ή 3, ο αριθμός του server και

[quadkey]: το quadkey του tile.

Για παράδειγμα το tile Επιπέδου 3 με quadkey 1221002 μπορεί να προσπελαστεί από τα URL:

<http://r0.ortho.tiles.virtualearth.net/tiles/r1221002.png?g=1>

<http://r1.ortho.tiles.virtualearth.net/tiles/r1221002.png?g=1>

<http://r2.ortho.tiles.virtualearth.net/tiles/r1221002.png?g=1>

<http://r3.ortho.tiles.virtualearth.net/tiles/r1221002.png?g=1>

Επισκεπτόμενοι κάποιο από αυτά τα URL, παίρνουμε το tile:



**Σχήμα 6.5:** Το tile με quadkey 1221002

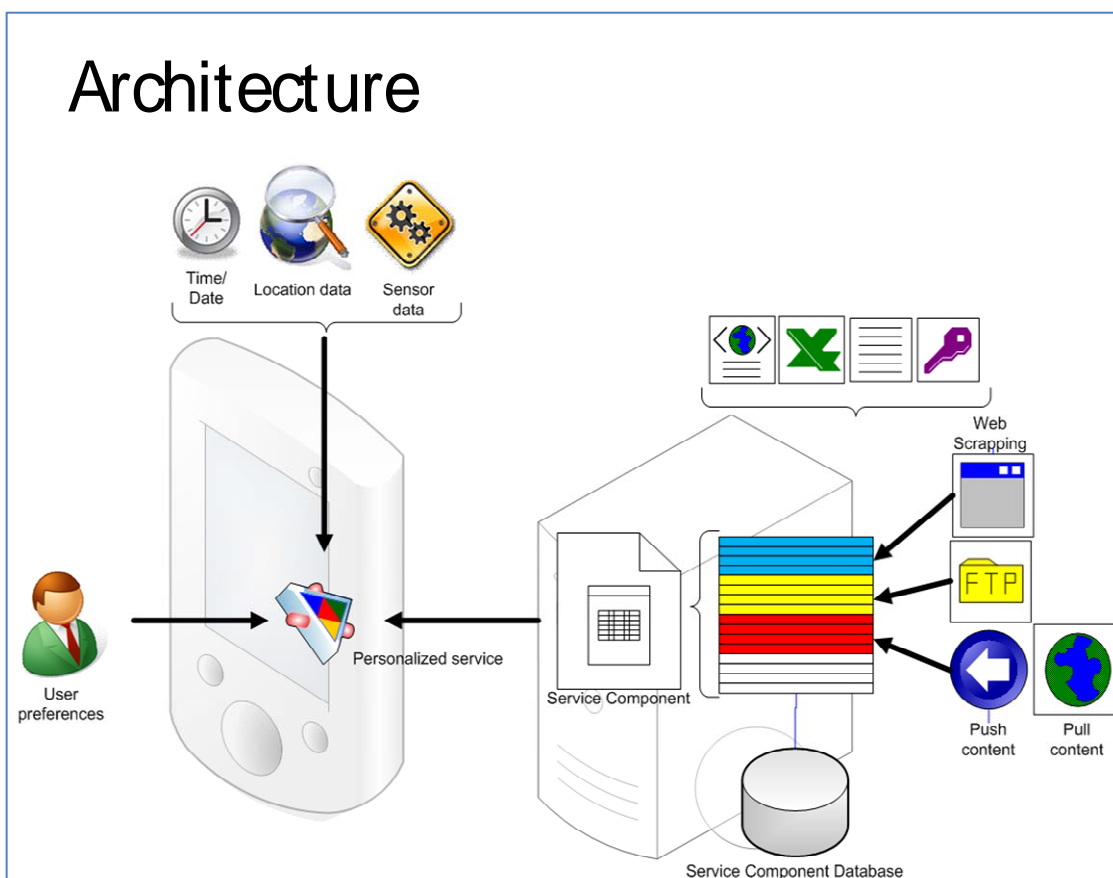
## 7. Η ΠΛΑΤΦΟΡΜΑ PLASMA

Η ανάπτυξη της πλατφόρμας **PLASMA** (Personalized, Location Aware Services over Mobile Architectures) είναι μια προσπάθεια σχεδιασμού και υλοποίησης ενός πλαισίου υπηρεσιών για κινητές συσκευές (smart phones, PDAs, PocketPCs), το οποίο προσφέρει εξατομικευμένες (personalized), προσαρμοστικές (context aware) (με έμφαση στο location awareness) υπηρεσίες. Πρόκειται για ένα έργο που βρίσκεται σε εξέλιξη [54]. Οι υπηρεσίες δεν προσφέρονται με τον παραδοσιακό τρόπο, αλλά βασίζονται σε μια ανοιχτή αρχιτεκτονική, η οποία επιτρέπει στο χρήστη να σχεδιάσει, να δοκιμάσει, να εκτελέσει και να μοιραστεί Δομικά Στοιχεία της Υπηρεσίας - ΔΣΥ (service components).

Το πλαίσιο επιτρέπει την εξατομικευμένη ανάπτυξη υπηρεσιών, οι οποίες μπορούν να προσαρμοστούν σύμφωνα με τις απαιτήσεις του χρήστη (customization) ως προς το επίπεδο προστασίας της ιδιωτικότητας (privacy) των προσωπικών πληροφοριών και την ακρίβεια των αποτελεσμάτων.

Η υποστήριξη εξατομικευμένων προσαρμοστικών υπηρεσιών βασίζεται στη χρήση πληροφοριών σχετικών με τη θέση, την κατάσταση, το χρόνο και τις προσωπικές προτιμήσεις του χρήστη για την υποβολή ερωτημάτων στον LBS server. Για να μπορεί να συντάξει ερωτήματα (queries), η εφαρμογή της κινητής συσκευής μπορεί να αναγνωρίσει όλες τις διαθέσιμες πληροφορίες στον LBS server, χρησιμοποιώντας το σχετικό API. Όταν ο χρήστης θελήσει να υποβάλει ένα ερώτημα για ένα συγκεκριμένο θέμα, χρησιμοποιούνται οι προσωπικές ρυθμίσεις χρήστη για να προσδιοριστεί το κατάλληλο επίπεδο ακρίβειας / αοριστίας σύμφωνα με τη μεθοδολογία που αναπτύχθηκε στην ενότητα 4. Όταν αυτό προσδιοριστεί, ο χρήστης στέλνει στον LBS server το ερώτημα, το οποίο περιέχει μεν προσωπικές πληροφορίες, αλλά προστατεύει την ιδιωτικότητα της ακριβούς θέσης και των ακριβών προτιμήσεων. Η πλατφόρμα, επομένως, εγγυάται την προστασία της ιδιωτικότητας, αφού οι αναφερόμενες πληροφορίες για τη γεωγραφική θέση και τις προσωπικές προτιμήσεις του χρήστη διακρίνονται από τον κατάλληλο (και εγκρινόμενο πάντα από το χρήστη) βαθμό ακρίβειας, με τεχνικές που αναπτύσσονται στη συσκευή, βάσει της μεθοδολογίας που περιγράφηκε στην ενότητα 4.

Το πλαίσιο στηρίζεται στην ανάπτυξη ΔΣΥ, τα οποία αποθηκεύονται σε έναν κεντρικό server και στα οποία έχουν πρόσβαση οι κινητές συσκευές. Κάθε ΔΣΥ επεξεργάζεται πληροφορίες που βρίσκονται στον εξυπηρετητή, «φιλτράρεται» από τις προσωπικές ρυθμίσεις / προτιμήσεις του χρήστη και παραμέτρους σχετικές με τις συνθήκες που επικρατούν (context), όπως θέση (location), χρόνος (time/date), πληροφορίες από αισθητήρες (sensor data), ώστε να παραχθεί μια εξατομικευμένη «έκδοση» της υπηρεσίας.



Σχήμα 7.1: Η αρχιτεκτονική της πλατφόρμας PLASMA

Ορισμένα παραδείγματα σεναρίων χρήσης είναι τα ακόλουθα:

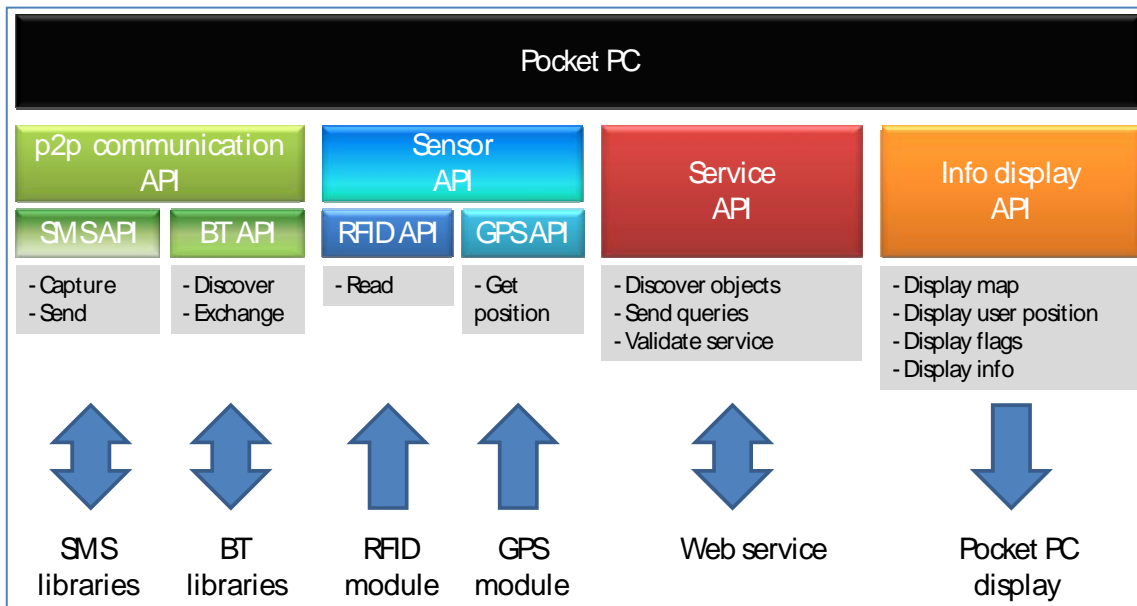
Ο χρήστης Α επιθυμεί να «χτίσει» (build) ένα ΔΣΥ, το οποίο όταν τίθεται σε εφαρμογή θα επιστρέφει τα πλησιέστερα βενζινάδικα. Τότε, επικοινωνεί με το server, επιλέγει τις πληροφορίες που θα χρησιμοποιηθούν και κατασκευάζει ένα ΔΣΥ, το οποίο όταν εκτελείται επιστρέφει το πλησιέστερο βενζινάδικο, το οποίο προσφέρει την καλύτερη τιμή στην αμόλυβδη βενζίνη και στον ιδιοκτήτη του οποίου δεν έχει επιβληθεί ποτέ πρόστιμο από τις αρχές για παραβίαση νόμων σχετικών με την προστασία των

καταναλωτών (π.χ. νοθευμένη βενζίνη, «πειραγμένοι» μετρητές). Αφού έχει χτίσει το ΔΣΥ, μπορεί να το αναπτύξει / εκτελέσει, όποτε χρειάζεται βενζίνη. Τότε, θα εμφανίζεται στο χάρτη στην κινητή συσκευή του χρήστη η λίστα με τα αποτελέσματα που ικανοποιούν τις προσωπικές του ρυθμίσεις και προτιμήσεις.

Ο χρήστης Β επιστρέφοντας στο σπίτι από τη δουλειά, πρέπει να περάσει από ένα φαρμακείο για να προμηθευτεί ασπιρίνες. Αρχικά, «χτίζει» το ΔΣΥ που θα εκτελεί περιοδικούς ελέγχους και θα τον ειδοποιεί όποτε ένα φαρμακείο είναι σε κοντινή απόσταση από το χρήστη (η έννοια του «κοντά» ορίζεται από τον ίδιο το χρήστη). Φεύγοντας από τη δουλειά, ενεργοποιεί το ΔΣΥ και κάθε φορά που βρίσκεται κοντά σε ένα φαρμακείο που πληροί τις προϋποθέσεις (π.χ. να διανυκτερεύει εκείνη τη μέρα, αν είναι βράδυ), εμφανίζεται ένα μήνυμα που του υπενθυμίζει την αγορά ασπιρινών εμφανίζοντας στο χάρτη τη θέση τού εν λόγω κοντινού φαρμακείου.

Οι χρήστες Γ και Δ επιθυμούν να συναντηθούν ένα απόγευμα. Συμφωνούν να συναντηθούν σε μια μεγάλη πλατεία με πολλές καφετέριες, όπου συνήθως επικρατεί πολυκοσμία. Αφού δεν ξέρουν σε ποια καφετέρια θα βρουν άδειο τραπέζι, συμφωνούν να επικοινωνήσουν όταν φτάσουν και ο πρώτος που θα φτάσει, να κρατήσει θέση. Ο χρήστης Δ φτάνει πρώτος και βρίσκει θέση στην καφετέρια X. Ενεργοποιεί τότε το P2P user location information module, το οποίο στέλνει την ακριβή θέση του και κάποιο κείμενο που πιθανόν πληκτρολογεί (π.χ. «Είμαι στο X»). Οι πληροφορίες στέλνονται με SMS, επομένως δεν παρεμβάλλεται κεντρικός server, επομένως διασφαλίζεται η ιδιωτικότητα της θέσης. Μόλις το SMS φτάσει στη συσκευή του χρήστη Γ, αποθηκεύεται και μόλις ενεργοποιηθεί η σχετική εφαρμογή, η πληροφορία για την ακριβή θέση του Δ απεικονίζεται στο χάρτη. Στη γενική περίπτωση, βέβαια, κάποιος χρήστης θα μπορεί να επιλέγει το επίπεδο ιδιωτικότητας, και θα αναφέρει τη θέση του σε αυτή τη βάση, π.χ. με ακρίβεια 200 τ.μ.

Αυτά είναι ορισμένα μόνο από τα πιθανά σενάρια λειτουργίας που θα υποστηρίζει η τελική πλατφόρμα.



Σχήμα 7.2: Δομικά στοιχεία εφαρμογής στην κινητή συσκευή

Η τελική εφαρμογή ενσωματώνει διάφορες διεπαφές επικοινωνίας (communication interfaces) στην κινητή συσκευή, υποστηρίζοντας επικοινωνία κινητής τηλεφωνίας (2-3G, SMS), ασύρματο LAN, αλλά και διεπαφές προσωπικών δικτύων (Personal Networking interfaces). Όσον αφορά τους αισθητήρες (sensors), υποστηρίζονται GPS και RFID, ενώ για το interface με το χρήστη χρησιμοποιείται το Windows Live Local (Virtual Earth) για την απεικόνιση θέσεων στο χάρτη (θα είναι δυνατή και η υποστήριξη και άλλων interfaces, λόγω του modular design που γίνεται).

## 8. ΒΑΣΕΙΣ ΓΙΑ ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ

Η έρευνα που έχει ξεκινήσει με αφορμή την εκπόνηση της παρούσας διπλωματικής εργασίας προσφέρει γόνιμο έδαφος για την περαιτέρω συνέχιση της δουλειάς, τόσο προς την κατεύθυνση της ολοκλήρωσης του προτεινόμενου μοντέλου, όσο και προς την κατεύθυνση της εφαρμογής του στο έργο PLASMA.

Αρχικά, το μοντέλο που προτάθηκε στις ενότητες 4 και 5 θα επεκταθεί, ώστε να καλύψει ένα μεγάλο εύρος δραστηριοτήτων του χρήστη. Θα πρέπει η μεθοδολογία κατηγοριοποίησης σε διαφορετικά επίπεδα ακρίβειας των πιθανών επιλογών του χρήστη για κάποια παράμετρο προσωπικών προτιμήσεων να εφαρμοστεί για ένα πλήθος παραμέτρων που ορίζονται στο προφίλ του. Το πρακτικό μέρος αυτής της διαδικασίας περιλαμβάνει τη συλλογή πραγματικών στοιχείων από πιθανούς χρήστες σχετικά με τις προτιμήσεις τους και την εφαρμογή μεθόδων συσταδοποίησης σύμφωνα με την προτεινόμενη μεθοδολογία (ενότητα 5), ώστε να δημιουργηθούν ομάδες προτιμήσεων σε διάφορα επίπεδα ακρίβειας / αοριστίας, με απώτερο στόχο τη δυνατότητα χρήσης αυτών στην εξασφάλιση ισορροπίας ανάμεσα στην ποιότητα της προσφερόμενης υπηρεσίας και της προστασίας της ιδιωτικότητας των προσωπικών πληροφοριών, κατά τα περιγραφόμενα στην ενότητα 4.

Ένα άλλο ενδιαφέρον, όσο και χρήσιμο για την εφαρμογή, πεδίο συνίσταται στην ενασχόληση με την πλευρά του Human – Computer Interaction (HCI). Κρίνεται, δηλαδή, σκόπιμη η δημιουργία ενός interface, μέσω του οποίου ο χρήστης θα μπορεί εύκολα και απλά να ορίζει το απαιτούμενο επίπεδο προστασίας της ιδιωτικότητας των προσωπικών του δεδομένων ή/και το επιθυμητό ύψος της ακρίβειας της υπηρεσίας. Κατόπιν αυτού, θα προσδιορίζονται και θα εξειδικεύονται αυτόματα για κάθε παράμετρο του προσωπικού προφίλ του τα κατάλληλα επίπεδα αφαίρεσης / ακρίβειας, βάσει των υπολογισμών των εκάστοτε τιμών της εντροπίας.

Τέλος, θα συνεχιστεί η έρευνα σχετικά με το θεωρητικό πλαίσιο για το συνδυασμό ακρίβειας αποτελεσμάτων και ιδιωτικότητας προσωπικών πληροφοριών. Η ιδέα για την περαιτέρω έρευνα βασίζεται στην Αρχή της Αβεβαιότητας του Heisenberg, σύμφωνα με την οποία, στην Κβαντομηχανική, το γινόμενο των τυπικών αποκλίσεων της θέσης και της ορμής ενός συστήματος υπό εξέταση υπόκειται σε ένα κάτω φράγμα, το οποίο απαγορεύει την ταυτόχρονη ακριβή μέτρηση της θέσης και της ορμής. Βασιζόμενοι στην



αρχή αυτή, μπορούμε να «αντιστοιχίσουμε» την αβεβαιότητα με την ιδιωτικότητα, υπολογίζοντας το γινόμενο των τυπικών αποκλίσεων διάφορων πλευρών της ιδιωτικότητας του χρήστη (δηλαδή για διάφορες προσωπικές παραμέτρους) και θέτοντας ένα κάτω φράγμα κατάλληλο για τις ρυθμίσεις ιδιωτικότητας του χρήστη. Πιο συγκεκριμένα, το μέγεθος «θέση» στην αρχή της αβεβαιότητας θα αντιστοιχιστεί με τη γεωγραφική θέση του χρήστη (η οποία μπορεί να αναπαρασταθεί σε πολλά διαφορετικά επίπεδα ακρίβειας), ενώ το μέγεθος «ορμή» θα αντιστοιχιστεί με την παράμετρο ενδιαφέροντος του χρήστη (και πάλι σε διαφορετικά επίπεδα ακρίβειας) για ένα συγκεκριμένο LBS ερώτημα προς τον server.

## ΠΑΡΑΡΤΗΜΑ: Κώδικας

Ακολουθεί ο κώδικας που έχει γραφτεί στην κατεύθυνση της υλοποίησης του client. Ο κώδικας γράφτηκε σε γλώσσα C# στο περιβάλλον Microsoft Visual Studio 2005, ενώ χρησιμοποιήθηκαν τα Windows Mobile 5.0 Pocket PC SDK, Windows Mobile 5.0 Smartphone SDK, .NET Compact Framework 2.0.

Ο παρατιθέμενος κώδικας καλύπτει τις ακόλουθες βασικές λειτουργίες:

- λήψη από το GPS των συντεταγμένων θέσης και μετατροπή αυτών τελικά σε tile quadkeys
- «κατέβασμα» από τους Virtual Earth Tile Servers των απαραίτητων tiles για την απεικόνιση του χάρτη και αποθήκευση των εκάστοτε πιο πρόσφατων tiles στη μνήμη (της συσκευής ή τυχόν storage card) για γρηγορότερη μελλοντική πρόσβαση
- απεικόνιση του χάρτη της περιοχής στην οποία βρίσκεται ο χρήστης σε δεδομένο επίπεδο ακρίβειας (zoom level)
- λειτουργίες zoom in, zoom out, scrolling, panning υλοποιημένες με τρόπους φιλικούς προς το χρήστη
- δυνατότητα εμφάνισης χάρτη περιοχής της οποίας τις συντεταγμένες παρέχει ο χρήστης
- δυνατότητα αποθήκευσης επαφών του χρήστη (ονόματα, διευθύνσεις, τηλέφωνα χρηστών – φίλων, γνωστών)
- αποστολή μηνυμάτων SMS.

Εκτός των κλάσεων που παρατίθενται στη συνέχεια, χρησιμοποιούνται και οι κλάσεις: GPS.cs, GpsDeviceState.cs, GpsPosition.cs, DeviceStateChangedEventArgs.cs, DegreesMinutesSeconds.cs, LocationChangedEventArgs.cs, Utils.cs, οι οποίες υπάρχουν στο πακέτο του Windows Mobile 5.0 Pocket PC SDK και συγκεκριμένα στον κατάλογο WindowsMobile5.0PocketPCSDK\Samples\Cs\Gps. Οι κλάσεις αυτές σχετίζονται με τη λήψη του σήματος GPS και τη μετατροπή του σε συντεταγμένες στην κατάλληλη μορφή. Επίσης, χρησιμοποιούνται οι κλάσεις ContactEditor.cs, ContactSelector.cs, που σχετίζονται με τη διαχείριση επαφών και βρίσκονται στο ίδιο πακέτο, στο φάκελο WindowsMobile5.0PocketPCSDK\Samples\Cs\Pocketoutlooksample. Τέλος, χρησιμοποιείται και η κλάση VirtualEarthTileSystem.cs [48] για τη μετατροπή γεωγραφικών συντεταγμένων σε συντεταγμένες pixel και tile και τελικά σε tile quadkeys.

Tile.cs

```
#region Using directives

using System;
using System.Drawing;

#endregion

namespace PLASMA
{
    /// <summary>
    /// An image tile and its corresponding URL.
    public class Tile
    {
        /// <summary>
        /// The URL of the image tile.
        /// </summary>
        private string url;

        /// <summary>
        /// The bitmap of the image tile.
        /// </summary>
        private Bitmap bitmap;

        public string Url
        {
            get { return url; }
        }

        public Bitmap Bitmap
        {
            get { return bitmap; }
        }

        public Tile(string url, Bitmap bitmap)
        {
            this.url = url;
            this.bitmap = bitmap;
        }
    }
}
```

Cache.cs

```
#region Using directives

using System;
using System.Collections;
using System.Drawing;
using System.IO;
using System.Net;
using System.Text;
using System.Windows.Forms;

#endregion

namespace PLASMA
{
    public class Cache
    {
        /// <summary>
        /// The cache of Tiles.
        /// </summary>
        private static ArrayList tiles = new ArrayList();

        /// <summary>
        /// The file cache.
        /// </summary>
        private ArrayList files = new ArrayList();

        /// <summary>
        /// The maximum number of files (with tile bitmaps) to cache in
memory
        /// </summary>
        public int MaxCachedFiles = 20;

        /// <summary>
        /// The maximum number of tiles (bitmaps) to cache in the LIST.
        /// </summary>
        private int maxCachedTiles = 4;

        /// <summary>
        /// The directory in persistent storage where the cached files
are stored.
        /// </summary>
        private string persistentStoragePath;

        /// <summary>
        /// The directory on the storage card where the cached files are
stored.
        /// </summary>
        private string storageCardPath;

        /// <summary>
        /// The bitmap to show when the real bitmap cannot be gotten
        /// from the server.
        /// </summary>
        private static Bitmap badBitmap = new Bitmap(256, 256);

        [System.Runtime.InteropServices.DllImport("coredll.dll")]
```

```
private static extern bool SHGetSpecialFolderPath(IntPtr
hwndOwner, StringBuilder lpszPath, int nFolder, bool fCreate);
```

```
/// <summary>
/// Name and creation time of cached file
/// </summary>
```

```
public struct CachedFileInfo
{
    public string Name;
    public DateTime CreationTime;

    public CachedFileInfo(string name, DateTime time)
    {
        Name = name;
        CreationTime = time;
    }
}
```

```
/// <summary>
/// Compares two FileInfos based on CreationTime.
/// </summary>
```

```
public class TimeSorter : IComparer
{
    int IComparer.Compare(Object x, Object y)
    {
        CachedFileInfo f1 = (CachedFileInfo)x;
        CachedFileInfo f2 = (CachedFileInfo)y;
        if (f1.CreationTime < f2.CreationTime)
        {
            return -1;
        }
        else if (f1.CreationTime == f2.CreationTime)
        {
            return 0;
        }
        else
        {
            return +1;
        }
    }
}
```

```
public Cache()
{
    // Set up file cache.
    DirectoryInfo di = new DirectoryInfo(CachePath);
    // Read in all the pre-existing files.
    foreach (FileInfo fileinfo in di.GetFiles())
    {
        files.Add(new CachedFileInfo(fileinfo.FullName,
fileinfo.CreationTime));
    }
    files.Sort(new TimeSorter());
}
```

```
/// <summary>
```

```

    /// The directory where the file cache lives if persistent
storage is used.
    /// </summary>

    private string PersistentStoragePath
    {
        get
        {
            if (persistentStoragePath == null)
            {
                int MAX_PATH = 260;
                int CSIDL_APPDATA = 26;
                StringBuilder sb = new StringBuilder(MAX_PATH);
                SHGetSpecialFolderPath(IntPtr.Zero, sb,
CSIDL_APPDATA, false);
                persistentStoragePath = sb.ToString() + "\\PLASMA";
                if (!Directory.Exists(persistentStoragePath))
                {
                    Directory.CreateDirectory(persistentStoragePath);
                }
            }
            return persistentStoragePath;
        }
    }

    /// <summary>
    /// The directory where the file cache lives if a storage card
is used.
    /// </summary>
    private string StorageCardPath
    {
        get
        {
            if (storageCardPath == null)
            {
                DirectoryInfo rootDir = new DirectoryInfo(@"");

                foreach (DirectoryInfo di in
rootDir.GetDirectories())
                {
                    if ((di.Attributes & FileAttributes.Temporary)
== FileAttributes.Temporary)
                    {
                        storageCardPath = "\\" + di.Name +
"\\PLASMA";

                        if (!Directory.Exists(storageCardPath))
                        {
                            Directory.CreateDirectory(storageCardPath);
                        }
                        break;
                    }
                }
            }
            return storageCardPath;
        }
    }
    /// <summary>

```

```

    /// The directory where the file cache lives.
    /// Checks if memory card is used, and returns appropriate path
    /// </summary>
    private string CachePath
    {
        get
        {
            int useStorageCard =
(int)Registry.GetValue(Registry.GetRootKey(Registry.HKey.CURRENT_USER),
                        "Software\\PLASMA", "UseStorageCard");

            //detecting if card is used. calling registry methods
            if (useStorageCard != 0)
            {
                string stgCardPath = StorageCardPath;
                if (Directory.Exists(stgCardPath))
                {
                    return stgCardPath;
                }
            }
            return PersistentStoragePath;
        }
    }

    /// <summary>
    /// Convert a URL to a local filename where the content of the
    /// URL was or will be stored.
    /// </summary>
    /// <param name="url">The url.</param>
    /// <returns>A filename in the local filesystem.</returns>
    private string UrlToFilename(string url)
    {
        // Extract the local filename from the url.
        int start = url.LastIndexOf('/');
        int end = url.LastIndexOf('?');
        string filename = CachePath + "\\\" + url.Substring(start +
1, end - start - 1);
        return filename;
    }

    /// <summary>
    /// Looks up the bitmap corresponding to the given url.
    /// Checks if tile is in the list; if not, checks in the file
    /// cache; if not, downloads the tile from VE Tile Server
    /// </summary>
    /// <param name="url">The url of the tile bitmap.</param>
    /// <returns>The bitmap, or null if not found.</returns>
    public Bitmap LookupUrl(string url)
    {
        Bitmap bitmap = LookupCachedBitmap(url);
        if (bitmap != null)
        {
            return bitmap;
        }

        bitmap = LookupCachedFile(url);
        if (bitmap != null)
        {

```

```

        return bitmap;
    }

    bitmap = LookupBitmapFromServer(url);
    if (bitmap != null)
    {
        return bitmap;
    }

    // Last resort: Return an empty bitmap
    this.Add(url, badBitmap);
    return badBitmap;
}

/// <summary>
/// Looks up the cached file corresponding to the given url.
/// If found, the resulting bitmap is cached in the LIST.
/// </summary>
/// <param name="url">The url of the tile bitmap</param>
/// <returns>The bitmap, or null if not found.</returns>
private Bitmap LookupCachedFile(string url)
{
    Bitmap bitmap;
    string filename = UrlToFilename(url);

    if (!File.Exists(filename))
    {
        return null;
    }

    FileStream stm = File.Open(filename, FileMode.Open);
    if (stm != null)
    {
        bitmap = new Bitmap(stm);
        stm.Close();
        // Cache the image IN THE LIST tiles, so that if we use
        // it again soon we needn't search in the file cache
        this.Add(url, bitmap);
    }
    else
    {
        bitmap = null;
    }
    return bitmap;
}

/// <summary>
/// Get a bitmap from the Virtual Earth server. If found, cache
/// the file and the bitmap.
/// </summary>
/// <param name="url">The url to retrieve.</param>
/// <returns>The bitmap.</returns>
private Bitmap LookupBitmapFromServer(string url)
{
    WebResponse response = null;
    Stream stream = null;

    // Get bitmap from VirtualEarth server.
    WebRequest req = WebRequest.Create(url);

```



```

        Cursor.Current = Cursors.WaitCursor;
        try
        {
            response = req.GetResponse();
            stream = response.GetResponseStream();
        }
        catch (WebException exc)
        {
            #if false
                MessageBox.Show("Could not get the image from server ");
            #endif
        }
        Cursor.Current = Cursors.Default;

        if (stream != null)
        {
            // Create the local file
            string filename = UrlToFilename(url);
            FileStream fileStream = File.Open(filename,
FileMode.CreateNew);
            byte[] bytes = new byte[1024];
            int bytesRead;
            while ((bytesRead = stream.Read(bytes, 0, 1024)) != 0)
            {
                fileStream.Write(bytes, 0, bytesRead);
            }
            fileStream.Close();
            response.Close();

            // Cache the local file
            files.Add(new CachedFileInfo(filename, DateTime.Now));

            // Delete the least recently-used file if necessary
            while (files.Count > MaxCachedFiles)
            {
                File.Delete(((CachedFileInfo) files[0]).Name);
                files.RemoveAt(0);
            }

            // Now get a bitmap from the cached file we just created.
            return LookupCachedFile(url);
        }
    }

```

```

    /// <summary>
    /// Looks up the cached bitmap corresponding to the given url.
    /// </summary>
    /// <param name="url">The url of the tile bitmap.</param>
    /// <returns>The bitmap, or null if not found.</returns>
    public Bitmap LookupCachedBitmap(string url)
    {
        Tile foundTile = null;
        Bitmap bitmap;

        foreach (Tile t in tiles)
        {
            if (t.Url == url)
            {
                foundTile = t;
            }
        }
    }

```

```

    }
}
if (foundTile != null)
{
    // Since someone is looking up this tile, move it to the
    // end of the list, which marks it as the most recently
    // used tile.
    tiles.Remove(foundTile);
    tiles.Add(foundTile);
    bitmap = foundTile.Bitmap;
}
else
{
    bitmap = null;
}
return bitmap;
}

```

```

/// <summary>
/// Adds a tile to the cache.
/// </summary>
/// <param name="url"></param>
/// <param name="bitmap"></param>
public void Add(string url, Bitmap bitmap)
{
    // Create a new tile and add it to the end of the list,
    // which marks it as the most recently used.
    tiles.Add(new Tile(url, bitmap));

    // Remove older tiles if necessary.
    if (tiles.Count > maxCachedTiles)
    {
        // Remove from the front of the list, where the
        // least recently used tile is.
        tiles.RemoveAt(0);

        // To avoid OutOfMemory exceptions.
        GC.Collect();
        GC.WaitForPendingFinalizers();
    }
}

```

```

/// <summary>
/// Remove from the cache any tiles whose bitmap is bad.
/// </summary>
public static void PurgeBadBitmaps()
{
    // List of bad tiles to remove.
    ArrayList bads = new ArrayList();

    foreach (Tile t in tiles)
    {
        if (t.Bitmap == badBitmap)
        {
            bads.Add(t);
        }
    }
    foreach (Tile t in bads)

```

```
        {
            tiles.Remove(t);
        }
    }

    public void PurgeAllBitmaps()
    {
        // Throw away the old cache and create a new one.
        tiles = new ArrayList();
    }
}
}
```

Map.cs

```

using System;
using System.Collections.Generic;
using System.Text;
using System.Drawing;
using System.Net;
using System.IO;
using System.Windows.Forms;

namespace PLASMA
{
    public class Map {

        public enum MapStyle
        {
            Road,
            Aerial,
            Hybrid
        }

        private double longitude;
        private double latitude;
        private int zoomLevel;
        private string mapStyle;
        private static Cache tileCache = new Cache();

        /// <summary>
        /// viewPort Width and Height
        /// depends on device and screen resolution
        /// </summary>
        private int viewPortW;
        private int viewPortH;

        /// <summary>
        /// The X-coordinate of the center of the map, in pixels, in
        world coordinates (at the current zoom level).
        /// </summary>
        private int x;

        /// <summary>
        /// The Y-coordinate of the center of the map, in pixels, in
        world coordinates (at the current zoom level).
        /// </summary>
        private int y;

        private static int tileSize = 256;
        private const int earthRadius = 6378137;

        public Map (double latitude, double longitude, int
zoomLevel, string mapStyle, int vPortW, int vPortH)
        {
            this.zoomLevel = zoomLevel;
            this.latitude = latitude;
            this.longitude = longitude;
            this.mapStyle = mapStyle;
            this.viewPortW=vPortW;
            this.viewPortH = vPortH;
        }
    }
}

```

```

public double Latitude
{
    get
    {
        if (double.IsNaN(latitude))
        {
            // When panning, must recalculate the latitude.
            latitude =
VirtualEarthTileSystem.YToLatitudeAtZoom(Y, zoomLevel);
        }
        return latitude;
    }
    set
    {
        latitude = value;
        y = VirtualEarthTileSystem.LatitudeToYAtZoom(latitude,
zoomLevel);
    }
}

public double Longitude
{
    get
    {
        if (double.IsNaN(longitude))
        {
            // When panning, must recalculate the longitude.
            longitude =
VirtualEarthTileSystem.XToLongitudeAtZoom(X, zoomLevel);
        }
        return longitude;
    }
    set
    {
        longitude = value;
        x = VirtualEarthTileSystem.LongitudeToXAtZoom(longitude,
zoomLevel);
    }
}

public int ZoomLevel
{
    get
    { return zoomLevel; }

    set { zoomLevel = value; }
}

public void setLatitude(double lat)
{
    latitude=lat;
}

public void setLongitude (double longi)
{
    longitude=longi;
}

public void setZoomLevel(int zoom)

```

```

    {
        zoomLevel=zoom;
    }

    public void setMapStyle(string style)
    {
        mapStyle=style;
    }

    public double getLatitude() { return latitude; }
    public double getLongitude() { return longitude; }
    public int getZoomLevel() { return zoomLevel; }

    /// <summary>
    /// Create the URL of the tile given its quadkey
    /// </summary>
    public string GetUrl(string quadKey)
    {
        string mapType = null;
        string mapExtension = null;

        mapType = "r";
        mapExtension = ".png";
        string url = String.Concat(new object[] { "http://",
mapType, quadKey[quadKey.Length - 1],
".ortho.tiles.virtualearth.net/tiles/", mapType, quadKey, mapExtension,
"?g=", 1 });
        return url;
    }

    /// <summary>
    /// Draws the map on the window
    /// </summary>

    public void Paint(Graphics g)
    {
        int pixelX, pixelY;
        VirtualEarthTileSystem.LatLongToPixelXY(Latitude, Longitude,
ZoomLevel, out pixelX, out pixelY);

        X = pixelX;
        Y = pixelY;

        // Where we are currently drawing on the window
        int drawingX, drawingY;

        // The coordinate x in pixels (on the world map at a given
        // zoom level) of the upper left point (point A) of the
        // visible rectangular on our screen
        // (the center of the rectangular is the point of interest P)
        int originX, originY;

        // Get the X,Y coordinates (in world coordinates) of the
        // origin of the window.
        // X is the coordinate x in pixels of the point of interestP
        // values: 0 to 256*(2^level)-1
        originX = X - this.viewPortW / 2;
        originY = Y - this.viewPortH / 2;
    }

```

```

int s = tileSize; //256

// Calculate how much we need to shift the first tile (and
// therefore all tiles) by to line up (originX, originY)
// with the origin of the window.
// The position in pixels of point A in its tile
int drawingOffsetX = originX % s;
int drawingOffsetY = originY % s;

int b = 0;
int w = this.viewPortW; //Screen (or Container) Width
int h = this.viewPortH;

// Calculate the first and last tile indices needed to fill
// the whole window.

//Tile indices (0 to 2^level-1) on the world map in given
//zoom level of the upper left tile of the viewport
int txStart = (int)Math.Floor((originX - b) / s);
int tyStart = (int)Math.Floor((originY - b) / s);

//Tile indices (0 to 2^level-1) on the world map in given
//zoom level of the down right left tile of the viewport
int txEnd = (int)Math.Floor((originX + w + b) / s);
int tyEnd = (int)Math.Floor((originY + h + b) / s);

txStart = Math.Max(0, txStart);
tyStart = Math.Max(0, tyStart);

// Loop through all the tiles needed to fill the window.

Console.WriteLine(txStart + " " + tyStart + " " + txEnd + "
" + " " + tyEnd + " " + originX + " " + originY);

for (int tx = txStart; tx <= txEnd; tx++)
{
    drawingX = (tx - txStart) * tileSize - drawingOffsetX;
    for (int ty = tyStart; ty <= tyEnd; ty++)
    {
        drawingY = (ty-tyStart) * tileSize - drawingOffsetY;

        String quadKey =
VirtualEarthTileSystem.TileXYToQuadKey(tx, ty, zoomLevel);
        String url = GetUrl(quadKey);
        url = GetUrl(quadKey);

        Bitmap bmp = tileCache.LookupUrl(url);

        g.DrawImage(bmp, drawingX, drawingY);
    }
}

}

/// <summary>
/// The X-coordinate of the center of the map, in pixels,
/// in world coordinates (at the current zoom level).

```

```

    /// </summary>
    public int X
    {
        get { return x; }
        set
        {
            x = value;
            // We don't recalculate the longitude until asked for.
            // So right now, we just invalidate it.
            longitude = double.NaN;
        }
    }

    /// <summary>
    /// The Y-coordinate of the center of the map, in pixels,
    /// in world coordinates (at the current zoom level).
    /// </summary>
    public int Y
    {
        get { return y; }
        set
        {
            y = value;
            // We don't recalculate the latitude until asked for.
            // So right now, we just invalidate it.
            latitude = double.NaN;
        }
    }

    /// <summary>
    /// Maximum allowed zoom level
    /// </summary>
    public int MaximumZoom
    {
        get { return 23; }
    }

    /// <summary>
    /// Minimum allowed zoom level
    /// </summary>
    public int MinimumZoom
    {
        get { return 3; }
    }
}
}
}

```



Program.cs

```
using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace PLASMA
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [MTAThread]
        static void Main()
        {
            Main mainObj = new Main();
            Application.Run(mainObj);
        }
    }
}
```

```

Phone.cs

using System;
using System.Text;
using System.Runtime.InteropServices;

namespace Microsoft.Wireless {
    /// <summary>
    /// Identifies the phone number type specified.
    /// </summary>
    public enum AddressType {
        /// <summary>Unknown phone number type.</summary>
        Unknown,
        /// <summary>International phone number.</summary>
        International,
        /// <summary>National phone number.</summary>
        National,
        /// <summary>Network-specific phone number.</summary>
        NetworkSpecific,
        /// <summary>Subscriber phone number.</summary>
        Subscriber,
        /// <summary>Alphanumeric phone number.</summary>
        Alphanumeric,
        /// <summary>Abbreviated phone number.</summary>
        Abbreviated
    }

    /// <summary>
    /// Information about the phone number.
    /// </summary>
    public struct PhoneAddress {
        /// <summary>The address type.</summary>
        public AddressType AddressType;
        /// <summary>The phone number in string format.</summary>
        public String Address;
    }

    /// <summary>
    /// Short Message Service.
    /// </summary>
    public class SMS {
        private static string SMS_MSGTYPE_TEXT
        = "Microsoft Text SMS Protocol";
        private static long SMS_MODE_SEND
        = 0x00000002;
        private static long SMS_OPTION_DELIVERY_NONE
        = 0x00000000;
        private static long SMS_OPTION_DELIVERY_NO_RETRY
        = 0x00000001;
        private static long PS_MESSAGE_OPTION_NONE
        = 0x00000000;

        private enum SMS_DATA_ENCODING {
            SMSDE_OPTIMAL=0,
            SMSDE_GSM,
            SMSDE_UCS2,
        }

        private enum PROVIDER_SPECIFIC_MESSAGE_CLASS {
            PS_MESSAGE_CLASS0 = 0,
            PS_MESSAGE_CLASS1,
        }
    }
}

```

```

        PS_MESSAGE_CLASS2,
        PS_MESSAGE_CLASS3,
    }

    private enum PROVIDER_SPECIFIC_REPLACE_OPTION {
        PSRO_NONE = 0,
        PSRO_REPLACE_TYPE1,
        PSRO_REPLACE_TYPE2,
        PSRO_REPLACE_TYPE3,
        PSRO_REPLACE_TYPE4,
        PSRO_REPLACE_TYPE5,
        PSRO_REPLACE_TYPE6,
        PSRO_REPLACE_TYPE7,
        PSRO_RETURN_CALL,
        PSRO_DEPERSONALIZATION,
    }

    private struct TEXT_PROVIDER_SPECIFIC_DATA {
        public IntPtr dwMessageOptions;
        public PROVIDER_SPECIFIC_MESSAGE_CLASS psMessageClass;
        public PROVIDER_SPECIFIC_REPLACE_OPTION
psReplaceOption;
    }

    [DllImport("sms.dll")]
    private static extern IntPtr SmsOpen(String
ptsMessageProtocol, IntPtr dwMessageModes, ref IntPtr psmsHandle,
IntPtr phMessageAvailableEvent);

    [DllImport("sms.dll")]
    private static extern IntPtr SmsSendMessage(IntPtr
smshHandle, IntPtr psmsaSMSCAddress, IntPtr psmsaDestinationAddress,
IntPtr pstValidityPeriod, byte[] pbData, IntPtr dwDataSize,
byte[] pbProviderSpecificData, IntPtr
dwProviderSpecificDataSize, SMS_DATA_ENCODING smsdeDataEncoding,
IntPtr dwOptions, IntPtr psmsmidMessageID);

    [DllImport("sms.dll")]
    private static extern IntPtr SmsClose(IntPtr smshHandle);

    /// <summary>
    /// Sends a SMS message to the phone number specified.
    /// </summary>
    /// <param name="sPhoneNumber">The phone number of the
recipient</param>
    /// <param name="sMessage">The message to send.</param>
    unsafe public static void SendMessage(string sPhoneNumber,
string sMessage) {
        IntPtr hSms = IntPtr.Zero;

        try {
            IntPtr res = SmsOpen(SMS_MSGTYPE_TEXT,
(IntPtr)SMS_MODE_SEND, ref hSms, IntPtr.Zero);
            if (res != IntPtr.Zero)
                throw new Exception("Could not open
SMS.");

            Byte[] bDest = new Byte[516];
            fixed (byte* pAddr = bDest) {
                byte *pCurrent = pAddr;

```



Main.cs

```

using System;
using System.IO;
using System.Collections;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Drawing.Imaging;
using System.Text;
using System.Reflection;
using System.Windows.Forms;
using Microsoft.WindowsMobile.Samples.Location;
using Microsoft.WindowsMobile.DirectX;
using Microsoft.WindowsMobile.DirectX.Direct3D;

namespace PLASMA
{
    /// <summary>
    /// Class for the main menu of the application.
    /// Zoom in/out, panning, scrolling, options are handled here
    /// </summary>

    public partial class Main : Form
    {
        private System.Windows.Forms.MainMenu mainMenu1;
        public const string AppName = "PLASMA";
        private int maxZoomLevel = 18;
        private int minZoomLevel = 1;
        private MenuItem menuLatLng;
        private MenuItem menuGps;
        private MenuItem menuSms;
        private MenuItem menuItem2;
        private MenuItem menuZoomIn;
        private MenuItem menuZoomOut;
        private ImageButton zoomInButton;
        private ImageButton zoomOutButton;
        private MenuItem menuExit;
        private GpsDeviceState device;
        private GpsPosition position;
        private Gps gps;
        private Map map;

        public Main()
        {
            InitializeComponent();
            //GPS stuff
            device = null;
            position = null;
            // Obtain lat/long from the GPS device
            gps = new Gps();
            map = new Map(37.573857, 23.4443495, 10, "r",
this.Size.Width, this.Size.Height);
            RedrawMap();
        }

        private void InitializeComponent()
        {

```

```

PresentParameters present;

// Give the application a way to be closed.
// This must be done before the device is created
this.MinimizeBox = false;

present = new PresentParameters();
present.Windowed = true;
present.SwapEffect = SwapEffect.Discard;

this.mainMenu1 = new System.Windows.Forms.MainMenu();
this.menuLatLng = new System.Windows.Forms.MenuItem();
this.menuGps = new System.Windows.Forms.MenuItem();
this.menuSms = new System.Windows.Forms.MenuItem();
this.menuZoomIn = new System.Windows.Forms.MenuItem();
this.menuZoomOut = new System.Windows.Forms.MenuItem();
this.menuExit = new System.Windows.Forms.MenuItem();
this.menuItem2 = new System.Windows.Forms.MenuItem();
this.menuItem2.Text = "Menu";
//
// menuLatLng
//
this.menuLatLng.Text = "Go to Lat/Long";
this.menuLatLng.Click += new
System.EventHandler(this.menuLatLng_Click);

//
// menuGps
//
this.menuGps.Text = "GPS";
this.menuGps.Click += new
System.EventHandler(this.menuGps_Click);

//
// SMS
//
this.menuSms.Text = "SMS";
this.menuSms.Click += new
System.EventHandler(this.menuSms_Click);

//
// zoomIn
//
this.menuZoomIn.Text = "Zoom In";
this.menuZoomIn.Click += new
System.EventHandler(this.menuZoomIn_Click);

//
//zoomOut
//
this.menuZoomOut.Text = "Zoom Out";
this.menuZoomOut.Click += new
System.EventHandler(this.menuZoomOut_Click);

//
// menuExit
//
this.menuExit.Text = "Exit";
// this.menuExit.Click += new
System.EventHandler(this.menuExit_Click);

```

```

        this.mainMenu1.MenuItems.Add(menuItem2);
        this.menuItem2.MenuItems.Add(menuZoomIn);
        this.menuItem2.MenuItems.Add(menuZoomOut);
        this.menuItem2.MenuItems.Add(menuLatLng);
        this.menuItem2.MenuItems.Add(menuGps);
        this.menuItem2.MenuItems.Add(menuSms);
        this.menuItem2.MenuItems.Add(menuExit);

        //
        // FormPocketEarth
        //
        this.ClientSize = new System.Drawing.Size(176, 180);
        this.Menu = this.mainMenu1;
        this.Text = "Virtual Earth Mobile";

        this.Deactivate += new
System.EventHandler(this.Main_Deactivate);
        this.Paint += new
System.Windows.Forms.PaintEventHandler(this.Main_Paint);
        this.KeyDown += new
System.Windows.Forms.KeyEventHandler(this.Main_KeyDown);

        this.MouseDown += new
MouseEventHandler(this.Main_MouseDown);
        this.Load += new System.EventHandler(this.Main_Load);
        this.KeyPreview=true;

        //zoomOutButton
        zoomOutButton = new ImageButton();
        zoomOutButton.Image = new
Bitmap(Assembly.GetExecutingAssembly().GetManifestResourceStream("PLASMA
.minus.gif"));
        zoomOutButton.Location = new Point(0, Height - 25);
        zoomOutButton.Size = new Size(25, 25);
        //Hook up into click event
        zoomOutButton.Click += new
EventHandler(zoomOutButton_Click);
        this.Controls.Add(zoomOutButton);

        //zoomInButton
        zoomInButton = new ImageButton();
        zoomInButton.Image = new
Bitmap(Assembly.GetExecutingAssembly().GetManifestResourceStream("PLASMA
.plus.gif"));
        zoomInButton.Location = new Point(Width - 25, Height - 25);
        zoomInButton.Size = new Size(25, 25);
        //Hook up into click event
        zoomInButton.Click += new EventHandler(zoomInButton_Click);
        this.Controls.Add(zoomInButton);
    }

    private void menuSms_Click(object sender, EventArgs e)
    {
        FormSMS dlg = new FormSMS(map.Longitude,map.Latitude);
        if (dlg.ShowDialog() != DialogResult.Cancel)
        {

```

```

    }
    dlg.Dispose();
}

private void menuGps_Click(object sender, EventArgs e)
{
    if (position == null || !position.LongitudeValid ||
!position.LatitudeValid)
    {
        GpsDeviceState dState = gps.GetDeviceState();

        MessageBox.Show("GPS not ready.Device
state:"+dState.DeviceState+"Service state:"+dState.ServiceState,
Main.AppName);
        gps.GetPosition();

    }
    else
    {
        map.Latitude = position.Latitude;
        map.Longitude = position.Longitude;
        RedrawMap();
    }
}

private void menuLatLong_Click(object sender, EventArgs e)
{
    FormLatLong dlg = new FormLatLong();
    // Pre-populate the dialog with the current lat/long
    dlg.Latitude = map.Latitude;
    dlg.Longitude = map.Longitude;
    dlg.ZoomLevel = map.ZoomLevel;

    if (dlg.ShowDialog() != DialogResult.Cancel)
    {
        // Set the map to lat/long specified by the user.
        try
        {
            // map object's fields update
            map.Latitude = dlg.Latitude;
            map.Longitude = dlg.Longitude;
            map.ZoomLevel = dlg.ZoomLevel;
        }
        catch (NullReferenceException eww) {
MessageBox.Show("error" + eww.Message); }

        RedrawMap();
    }
    dlg.Dispose();
}

private void Main_MouseDown(object sender, MouseEventArgs e)
{
    int deltaX = Size.Width / 4;
    int deltaY = Size.Height / 4;

```



```

int newMapX, newMapY;

if (e.X > Size.Width / 2)
    newMapX = map.X + (e.X - Size.Width / 2);
else
    newMapX = map.X - (Size.Width / 2 - e.X);

if (e.Y > Size.Width / 2)
    newMapY = map.Y + (e.Y - Size.Height / 2);
else
    newMapY = map.Y - (Size.Height / 2 - e.Y);

map.X = newMapX;
map.Y = newMapY;

RedrawMap();
}

private void Main_KeyDown(object sender, KeyEventArgs e)
{
    // Pan by one quarter the size of the map.
    int deltaX = Size.Width / 4;
    int deltaY = Size.Height / 4;
    int newZoom;

    // Pan or zoom the map.
    switch (e.KeyCode)
    {
        case Keys.Enter:
        case (Keys)120: // Pound key
        case Keys.D5:
            newZoom = newZoom = map.ZoomLevel + 1;
            if (newZoom < this.maxZoomLevel)
            {
                // we have invalidated long/lat so calculate with the
old zoom level
                // before changing it
                map.Longitude =
VirtualEarthTileSystem.XToLongitudeAtZoom(map.X, map.ZoomLevel);
                map.Latitude =
VirtualEarthTileSystem.YToLatitudeAtZoom(map.Y, map.ZoomLevel);
                map.ZoomLevel = newZoom;
                RedrawMap();
            }

            break;

        case (Keys)119: // Star key
            // Zoom out.
            newZoom = map.ZoomLevel - 2;
            map.ZoomLevel = newZoom;
            RedrawMap();

            break;

        case Keys.Up:
        case Keys.D2:
            map.Y -= deltaY;

```

```

        RedrawMap ();
        break;

    case Keys.Down:
    case Keys.D8:
        map.Y += deltaY;
        RedrawMap ();
        break;

    case Keys.Left:
    case Keys.D4:
        map.X -= deltaX;
        RedrawMap ();
        break;

    case Keys.Right:
    case Keys.D6:
        map.X += deltaX;
        RedrawMap ();
        break;

    case Keys.D1:
        map.Y -= deltaY;
        map.X -= deltaX;
        RedrawMap ();
        break;

    case Keys.D3:
        map.Y -= deltaY;
        map.X += deltaX;
        RedrawMap ();
        break;

    case Keys.D7:
        map.Y += deltaY;
        map.X -= deltaX;
        RedrawMap ();
        break;

    case Keys.D9:
        map.Y += deltaY;
        map.X += deltaX;
        RedrawMap ();
        break;
    }
}

private void Main_Paint(object sender, PaintEventArgs e)
{
    Graphics gx = e.Graphics;
    //Bitmap bmp = null;
    map.Paint (gx);

    Bitmap pushPin = new
Bitmap(Assembly.GetExecutingAssembly().GetManifestResourceStream("PLASMA
.PushPin.png"));

    ImageAttributes imgattr = new ImageAttributes();
    imgattr.SetColorKey(Color.White, Color.White);

```

```

        gx.DrawImage(pushPin, new Rectangle(Size.Width/2-
pushPin.Width/2, Size.Height/2-pushPin.Height/2,
        pushPin.Width,pushPin.Height), 0, 0, pushPin.Width,
pushPin.Height,
        GraphicsUnit.Pixel, imgattr);

    }

    private void Main_Deactivate(object sender, EventArgs e)
    {
    }

    private void Main_Load(object sender, EventArgs e)
    {
        this.Text = Main.AppName;

        gps.DeviceStateChanged += new
DeviceStateChangedEventHandler(gps_DeviceStateChanged);
        gps.LocationChanged += new
LocationChangedEventHandler(gps_LocationChanged);
        if (!gps.Opened) gps.Open();
    }

    private void RedrawMap()
    {
        // Clean up the cache so we will try to refetch any bad
bitmaps.
        Cache.PurgeBadBitmaps();
        // Invalidate the window.
        Invalidate();
    }

    protected void gps_LocationChanged(object sender,
LocationChangedEventArgs args)
    {
        position = args.Position;
    }

    void gps_DeviceStateChanged(object sender,
DeviceStateChangedEventArgs args)
    {
        device = args.DeviceState;
    }

    private void zoomOutButton_Click(object sender, EventArgs e)
    {
        zoomOut();
    }

    private void zoomInButton_Click(object sender, EventArgs e)
    {
        zoomIn();
    }

    private void zoomIn()

```

```

        {
            int newZoom = newZoom = map.ZoomLevel + 2;
            if (newZoom < this.maxZoomLevel)
            {
                // we have invalidated long/lat so calculate with the
old zoom level
                // before increasing it
                map.Longitude =
VirtualEarthTileSystem.XToLongitudeAtZoom(map.X, map.ZoomLevel);
                map.Latitude =
VirtualEarthTileSystem.YToLatitudeAtZoom(map.Y, map.ZoomLevel);
                map.ZoomLevel = newZoom;
                RedrawMap();
            }
        }

private void zoomOut ()
{
    int newZoom = newZoom = map.ZoomLevel - 2;
    if (newZoom >= this.minZoomLevel)
    {
        // we have invalidated long/lat so calculate with the
old zoom level
        // before decreasing it
        map.Longitude =
VirtualEarthTileSystem.XToLongitudeAtZoom(map.X, map.ZoomLevel);
        map.Latitude =
VirtualEarthTileSystem.YToLatitudeAtZoom(map.Y, map.ZoomLevel);
        map.ZoomLevel = newZoom;
        RedrawMap();
    }
}

private void menuZoomIn_Click(object sender, EventArgs e)
{
    zoomIn();
}

private void menuZoomOut_Click(object sender, EventArgs e)
{
    zoomOut();
}
}
}

```

```

ContactForm.cs

using [...];
using Microsoft.WindowsMobile.PocketOutlook;

namespace PLASMA
{
    public partial class ContactForm : Form
    {
        private OutlookSession outlookSession;
        private Contact contactToSelect;
        private String mobileTelephoneNumber;

        public ContactForm()
        {
            this.outlookSession = new OutlookSession();
            this.InitializeComponent();
            InitializeComboBox();
        }

        public String MobileTelephoneNumber
        {
            get
            { return mobileTelephoneNumber; }
            set
            { mobileTelephoneNumber = value; }
        }

        private void InitializeComboBox()
        {
            this.comboBox1.DataSource = null;
            this.comboBox1.DataSource =
this.outlookSession.Contacts.Items;
            this.comboBox1.DisplayMember = "FileAs";
            this.comboBox1.ValueMember = "ItemId";
        }

        private void comboBox1_SelectedIndexChanged(object sender,
EventArgs e)
        {
            contactToSelect = (Contact)this.comboBox1.SelectedItem;
this.mobileTelephoneNumber=contactToSelect.MobileTelephoneNumber;
        }

        private void menuItem2_Click(object sender, EventArgs e)
        {
            contactToSelect = (Contact)this.comboBox1.SelectedItem;
            this.mobileTelephoneNumber =
contactToSelect.MobileTelephoneNumber;
            DialogResult = DialogResult.OK;
        }

        private void menuItem1_Click(object sender, EventArgs e)
        {
            DialogResult = DialogResult.Cancel;
        }

        private void ContactForm_Load(object sender, EventArgs e) {}
    }
}

```

FormLatLng.cs

```

using [...];

namespace PLASMA
{

    public partial class FormLatLng : Form
    {
        public FormLatLng()
        {
            InitializeComponent();
        }

        public double Latitude
        {
            get { return Convert.ToDouble(LatitudeBox.Text); }
            set { LatitudeBox.Text = value.ToString(); }
        }

        public double Longitude
        {
            get { return Convert.ToDouble(LongitudeBox.Text); }
            set { LongitudeBox.Text = value.ToString(); }
        }

        public int ZoomLevel
        {
            get { return Convert.ToInt32(DetailBox.Text); }
            set { DetailBox.Text = value.ToString(); }
        }

        private void OkButton_Click(object sender, EventArgs e)
        {
            Latitude = Double.Parse(LatitudeBox.Text);
            Longitude = Double.Parse(LongitudeBox.Text);
            ZoomLevel = Int32.Parse(DetailBox.Text);
            this.Close();
        }

        private void LatitudeBox_TextChanged(object sender, EventArgs e) {}

        private void label1_ParentChanged(object sender, EventArgs e) {}

        private void ClearButton_Click(object sender, EventArgs e)
        {this.Close();}

        private void menuItem2_Click(object sender, EventArgs e)
        {
            Latitude = Double.Parse(LatitudeBox.Text);
            Longitude = Double.Parse(LongitudeBox.Text);
            ZoomLevel = Int32.Parse(DetailBox.Text);
            this.Close();
        }

        private void menuItem1_Click(object sender, EventArgs e)
        {this.DialogResult = DialogResult.Cancel;}
    }
}

```

FormSMS.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Drawing.Imaging;
using System.Text;
using System.Reflection;
using System.Windows.Forms;
using Microsoft.WindowsMobile.PocketOutlook;
using Microsoft.WindowsMobile.Samples.ContactSample;

namespace PLASMA
{
    public partial class FormSMS : Form
    {
        public double latitude;
        public double longitude;

        public FormSMS(double lon, double lat)
        {
            Latitude = lat;
            Longitude = lon;
            InitializeComponent();
        }

        public double Latitude
        {
            get
            {
                return latitude;
            }
            set
            {
                latitude = value;
            }
        }

        public double Longitude
        {
            get
            {
                return longitude;
            }
            set
            {
                longitude = value;
            }
        }

        private void button1_Click(object sender, EventArgs e)
        {
            SmsMessage s = new SmsMessage(txtPhoneNumber.Text,
txtMessage.Text);
            s.Send();
            this.Close();
        }
    }
}

```

```

e)     private void txtPhoneNumber_TextChanged(object sender, EventArgs
        {
        }

        private void txtMessage_TextChanged(object sender, EventArgs e)
        {
        }

        private void Contact_Click(object sender, EventArgs e)
        {
            ContactForm dlg = new ContactForm();
            if (dlg.ShowDialog() != DialogResult.Cancel)
            {
                if (dlg.MobileTelephoneNumber != null)
                    this.txtPhoneNumber.Text =
dlg.MobileTelephoneNumber;
                this.Update();
            }
            dlg.Dispose();
        }

        private void menuItem1_Click(object sender, EventArgs e)
        {
            DialogResult = DialogResult.Cancel;
        }

        private void menuItem2_Click(object sender, EventArgs e)
        {
            String message =
"PLASMA#" + latitude + "#" + longitude + "#" + txtMessage.Text;
            SmsMessage s = new SmsMessage(txtPhoneNumber.Text, message);
            s.Send();
            this.Close();
        }

        private void FormSMS_Load(object sender, EventArgs e) {}
    }
}

```



**ΒΙΒΛΙΟΓΡΑΦΙΑ**

- [1] Ubiquitous computing, Wikipedia, the free encyclopedia [Online]. Available: [http://en.wikipedia.org/wiki/Ubiquitous\\_computing](http://en.wikipedia.org/wiki/Ubiquitous_computing) [accessed on August 26<sup>th</sup>, 2007]
- [2] A. Dey, "Understanding and using context," *Personal and Ubiquitous Computing*, vol. 5(1), pp. 47-52, 2001.
- [3] A. Dey and G. Abowd, "Towards a better understanding of context and context-awareness," in *Proceedings of the CHI 2000 Workshop on "The What, Who, Where, When, Why and How of Context-Awareness"*, 2000.
- [4] G. Abowd, A. Dey, P. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness," in *Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, 1999, pp. 304-307, Springer-Verlag.
- [5] T. Reichenbacher, "Adaptive methods for mobile cartography," in *Proceedings of the 21<sup>st</sup> International Cartographic Conference ICC: Cartographic Renaissance*, 10-16 August 2003, Durban, South Africa, pp. 1311-1321.
- [6] S. Stieniger, M. Neun, A. Edwardes, "Foundations of location based services," Project CartouCHE - Lecture Notes on LBS, version 1.0 [Online]. Available: [http://www.geo.unizh.ch/publications/cartouche/lbs\\_lecturenotes\\_steinigeretal2006.pdf](http://www.geo.unizh.ch/publications/cartouche/lbs_lecturenotes_steinigeretal2006.pdf) [accessed on August 26<sup>th</sup>, 2007].
- [7] A. Beresford, "Location privacy in ubiquitous computing," Ph.D. dissertation, University of Cambridge, Computer Laboratory, 2005.
- [8] K. Lyytinen and Y. Yoo, "Issues and challenges in ubiquitous computing," *Communications of the ACM*, vol. 45(12), pp. 62-65, 2002.
- [9] M. Weiser, "The computer for the 21<sup>st</sup> century," *Scientific American*, vol. 365(3), pp. 94-104, September 1991.
- [10] B. Schilit, N. Adams, and R. Want, "Context-aware computing applications," in *Proceedings of the Workshop on Mobile Computing Systems and Applications*, 1993, pp. 85-90.
- [11] D. Tennenhouse, "Proactive computing," *Communications of the ACM*, vol. 43(5), pp. 43-50, 2000.

- [12] K. Virrantaus, J. Markkula, A. Garmash, Y. V. Terziyan, “Developing GIS-Supported Location Based Services,” in *Proceedings of WGIS’2001 – First International Workshop on Web Geographical Information Systems*, Kyoto, Japan, 2001, pp. 423-432.
- [13] ETSI, “Human factors (HF); User profile management,” EG 202 325, v1.1.1, October 2005 [Online]. Available: [http://webapp.etsi.org/action/PU/20051018/eg\\_202325v010101p.pdf](http://webapp.etsi.org/action/PU/20051018/eg_202325v010101p.pdf) [accessed on August 26<sup>th</sup>, 2007]
- [14] ETSI, “Universal Mobile Telecommunications System (UMTS); Service requirements for 3GPP Generic User Profile (GUP); Stage 1,” (3GPP TS 22.240 Release 6).
- [15] ETSI, “Universal Mobile Telecommunications System (UMTS); Service Requirements for 3GPP Generic User Profile (GUP); Stage 2,” (3GPP TS 23.240 Release 6).
- [16] ETSI, “Universal Mobile Telecommunications System (UMTS); Service Requirements for 3GPP Generic User Profile (GUP); Stage 3,” (3GPP TS 29.240 Release 6).
- [17] W3C, “Composite Capabilities / Preference Profiles (CC/PP): Structure and Vocabularies 1.0”, Recommendation, 15 January 2004 [Online]. Available: <http://www.w3.org/TR/CCPP-struct-vocab/> [accessed on August 26<sup>th</sup>, 2007].
- [18] Resource Description Framework [Online]. Available: <http://www.w3.org/RDF/>, [accessed on August 18<sup>th</sup>, 2007]
- [19] I. Jørstad, T. van Do, S. Dustdar, “Personalization of next generation mobile services,” in *4th International Workshop on Ubiquitous Mobile Information and collaboration Systems (UMICS)*, co-located with CAiSE 2006, Luxemburg, 5 - 6 June 2006.
- [20] D. Petrie, “A Framework for Session Initiation Protocol User Agent Profile Delivery (draft-ietf-sipping-config-framework-11),” March 3<sup>rd</sup>, 2007 [Online]. Available: <http://www.ietf.org/internet-drafts/draft-ietf-sipping-config-framework-11.txt> [accessed on May 15<sup>th</sup>, 2007].

- [21] R. Pascotto, “Designing Advanced network Interfaces for the Delivery and Administration of Location independent, Optimised personal Services (DAIDALOS),” [Online]. Available: <http://www.ist-daidalos.org> [accessed on May 15<sup>th</sup>, 2007].
- [22] OSA Phase 4, Parley/OSA Specifications [Online]. Available: <http://www.parlay.org>, <http://portal.etsi.org/docbox/TISPAN/Open/OSA/Parlay60.html>, [accessed on May 15<sup>th</sup>, 2007].
- [23] M. Golemati, A. Katifori, C. Vassilakis, G. Lepouras, and C. Halatsis, “Creating an ontology for the user profile: method and applications,” in *Proceedings of the First IEEE International Conference on Research Challenges in Information Science (RCIS)*, Morocco, 2007.
- [24] IST-027396 MAGNET Beyond Deliverable D1.2.1, “The conceptual structure of user profiles,” September 2006 [Online]. Available: <http://www.ist-magnet.org/public+deliverables> [accessed on May 15<sup>th</sup>, 2007].
- [25] Fox News. “Man Accused of Stalking Ex-Girlfriend With GPS” [Online]. Available: <http://www.foxnews.com/story/0,2933,131487,00.html>, [accessed on August 26<sup>th</sup>, 2007].
- [26] USAToday, “Authorities: GPS system used to stalk woman” [Online]. Available: [http://www.usatoday.com/tech/news/2002-12-30-gps-stalker\\_x.htm](http://www.usatoday.com/tech/news/2002-12-30-gps-stalker_x.htm), [accessed on August 26<sup>th</sup>, 2007].
- [27] M. Duckham and L. Kulik, “A formal model of obfuscation and negotiation for location privacy,” in *Pervasive Computing*, Springer Berlin / Heidelberg, 2005, pp. 152-170.
- [28] H. Kido, Y. Yanagisawa, and T. Satoh, “Protection of location privacy using dummies for location-based services,” in *Proceedings of the 21st International Conference on Data Engineering Workshops*, 2005, p. 1248.
- [29] L. Sweeney, “Achieving k-anonymity privacy protection using generalization and suppression,” *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, vol. 10(5), pp. 571–588, 2002.

- [30] A. Pfitzmann and M. Koehntopp, “Anonymity, unobservability, and pseudonymity – a proposal for terminology,” in *Proceedings of the International Workshop on Design Issues in Anonymity and Unobservability*, vol. 2009 of LNCS, Springer, 2000.
- [31] M. Gruteser and D. Grunwald, “Anonymous usage of location-based services through spatial and temporal cloaking,” presented at the First International Conference on Mobile Systems, Applications, and Services, 2003.
- [32] B. Gedik and L. Liu, “Location privacy in mobile systems: a personalized anonymization model,” in *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems*, 2005, pp. 620-629.
- [33] M. F. Mokbel, C. Chow, and W. G. Aref, “The new Casper: query processing for location services without compromising privacy,” in *Proceedings of the 32nd International Conference on Very Large Data Bases*, Seoul, Korea, 2006, pp. 763-774.
- [34] X. Jiang and J. A. Landay, “Modeling privacy control in context-aware systems,” *Pervasive Computing*, IEEE, vol. 1(3), July-September 2002, pp. 59 – 63.
- [35] A. F. Westin, *Privacy and Freedom*, New York NY: Atheneum, 1967.
- [36] C. E. Shannon, “The Mathematical Theory of Communication,” *The Bell System Technical Journal*, vol. 27, 1948, pp. 379-423, 623-656.
- [37] D. Lurie, J. Valls, and J. Wagensberg, “Thermodynamic approach to biomass distributions in ecological systems,” in *Bulletin of Mathematical Biology*, Vol. 45(5), 1983, pp. 869-872.
- [38] D. Lurie and J. Wagensberg, “Information theory and ecological diversity,” in *Systems Far from Equilibrium*, Berlin, West Germany, 1980, Sitges Conf. on Statistical Mechanics, Springer-Verlag, pp. 290-303.
- [39] A. E. Magurran, *Ecological Diversity and Its Measurement*, Princeton University Press, 1988.
- [40] K. Bailey, *Social Entropy Theory*, State University of New York Press, Albany, 1990.
- [41] P. H. E. Sneath and R. R. Sokal, *Numerical Taxonomy: The Principles and Practice of Numerical Classification*, Freeman, San Francisco, 1973.

- [42] N. Jardine and R. Sibson, *Mathematical Taxonomy*, John Wiley and sons, 1971.
- [43] T. Balch, “Hierarchical social entropy: an information theoretic measure of robot group diversity,” in *Autonomous Robots*, vol. 8, no 3, Springer Netherlands, 2000, pp. 209-238.
- [44] Live Search (Windows local live homepage) [Online]. Available: <http://local.live.com/> [accessed on August 27<sup>th</sup>, 2007].
- [45] Google Maps [Online]. Available: <http://maps.google.com/> [accessed on August 27<sup>th</sup>, 2007].
- [46] Yahoo! Maps [Online]. Available: <http://maps.yahoo.com/> [accessed on August 27<sup>th</sup>, 2007].
- [47] Mercator projection - Wikipedia, the free encyclopedia [Online]. Available: [http://en.wikipedia.org/wiki/Mercator\\_projection](http://en.wikipedia.org/wiki/Mercator_projection) [accessed on August 27<sup>th</sup>, 2007].
- [48] Virtual Earth Tile System (V4) [Online]. Available: <http://msdn2.microsoft.com/en-us/library/bb259689.aspx> [accessed on August 27<sup>th</sup>, 2007].
- [49] H. Samet, “The quadtree and related hierarchical data structures,” *ACM Computing Surveys*, vol.16(2), pp. 187-260, 1984.
- [50] A. Klinger, “Patterns and search statistics,” in *Optimizing Methods in Statistics*, J.S. Rustagi, FEd., Academic Press, New York, 1971, pp. 303-337.
- [51] R. de Neufville, “Defining Capacity of Airport Passenger Buildings,” lecture notes for the course *Airport Systems Planning, Design, and Management*, Massachusetts Institute of Technology, [http://ardent.mit.edu/airports/ASP\\_current\\_lectures/ASP%2004/Defining\\_Capacity\\_04.pdf](http://ardent.mit.edu/airports/ASP_current_lectures/ASP%2004/Defining_Capacity_04.pdf), [accessed on August 27<sup>th</sup>, 2007].
- [52] M. Wing, A. Eklund, and L. Kellogg, “Consumer-grade global positioning system (GPS) accuracy and reliability,” *Journal of Forestry*, vol. 103, no 4, pp. 169-173(5), 2005.
- [53] T. Dirks, “Film Genres” [Online]. Available: <http://www.filmsite.org/genres.html>, [accessed on August 27<sup>th</sup>, 2007].

- [54] “The PLASMA project: Personalized, Location Aware Services over Mobile Architectures” [Online]. Available: <http://www.telecom.ntua.gr/~bpatr/staticcontent/PLASMA.php> [accessed on August 27<sup>th</sup>, 2007].
- [55] P. N. Karamolegkos, Ch. Z. Patrikakis, N. D. Doulamis, “Adaptive user communities assessment in personal networking applications”, to be presented at the 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'07), Athens, Greece, September 2007.
- [56] P. N. Karamolegkos, Ch. Z. Patrikakis, N. D. Doulamis, E. Tragos, “User-profile based communities assessment using clustering methods,” to be presented at the 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'07), Athens, Greece, September 2007.
- [57] S. Schiaffino and A. Amandi, “Polite personal agent,” *IEEE Intelligent Systems*, vol. 21, no.1, pp.12-19, 2006.
- [58] D.S. Phatak and R. Mulvaney, “Clustering for personalized mobile web usage,” in *Proceedings of the 2002 IEEE International Conference on Fuzzy Systems*, vol. 1, USA, 2002, pp. 705-710.
- [59] B. Liu, H. Wang, and A. Feng, “Applying information agent in open bookmark service,” *Advances in Engineering Software*, vol. 32, no. 7, pp. 519-525, 2001.
- [60] J.L. Huang, M.S. Chen, and H.P. Hung, “A QoS aware transcoding proxy using on demand data broadcasting,” in *Proceedings of IEEE INFOCOM 2004*, Hong Kong, 2004, vol. 2, pp. 2050-2059.
- [61] A. Metwally, D. Agrawal, and A. E. Abbadi, “Efficient computation of frequent and top-k elements in data streams,” in *Database Theory*, Springer Lecture Notes in Computer Science, vol. 3363/2004, 2005.
- [62] E. Tonkin, “Searching the long tail,” in *Proceedings of the 17th Workshop of the American Society for Information Science and Technology Special Interest Group in Classification Research*, Austin, Texas, USA, 2006.
- [63] A. Kobsa and J. Fink, “An LDAP-based user modeling server and its evaluation,” *User Modeling and User-Adapted Interaction*, Kluwer Academic Publishers, vol. 16, no. 2, pp. 129-169, 2006.
- [64] A. Rencher, *Methods of Multivariate Analysis*, Wiley, 2002.

- [65] U. von Luxburg, “A tutorial on spectral clustering,” *Max Planck Institute for Biological Cybernetics Technical Report*, No. TR-149.
- [66] F. Bach and M. Jordan, “Learning spectral clustering,” in *Advances in Neural Information Processing Systems 16 (NIPS)*, Cambridge, MA: MIT Press, 2004.
- [67] B. Hendrickson and R. Leland, “An improved spectral graph partitioning algorithm for mapping parallel computations”, *SIAM J. on Scientific Computing*, vol. 16, pp. 452–469, 1995.
- [68] M. Meila and J. Shi, “A random walks view of spectral segmentation,” in *Proceedings of 8<sup>th</sup> International Workshop on Artificial Intelligence and Statistics (AISTATS)*, 2001.
- [69] M. Meila and L. Xu, “Multiway cuts and spectral clustering,” *U. Washington Department of Statistics Tech Report 442*, 2004.