



**Πανεπιστήμιο Πελοποννήσου**  
**Σχολή Θετικών Επιστημών και Τεχνολογίας**  
**Τμήμα Επιστήμης και Τεχνολογίας Υπολογιστών**

## **Μεταπτυχιακή Διπλωματική Εργασία**

**ΤΙΤΛΟΣ**

**ΕΝΗΜΕΡΩΣΗ ΣΤΟΙΧΕΙΩΝ ΠΡΟΣΩΠΙΚΗΣ ΟΝΤΟΛΟΓΙΑΣ ΑΠΟ**

**ΚΙΝΗΤΟ: ΕΦΑΡΜΟΓΗ ΣΕ ANDROID**

**Ονόματα**

**ΓΚΑΝΑΣ ΙΩΑΝΝΗΣ 2010008**  
**ΓΙΑΝΝΟΠΟΥΛΟΣ ΝΙΚΟΛΑΟΣ 2010006**

**Επιβλέπων:**  
**Λέπυρας Γεώργιος**

**Τρίπολη, Μάιος 2013**

## Περίληψη

Στόχος της εργασίας αυτής είναι η δημιουργία λογισμικού που θα επιτρέπει την ενημέρωση των στοιχείων της προσωπικής οντολογίας με δεδομένα που συλλέγονται από το προσωπικό κινητό τηλέφωνο του χρήστη. Η συγκεκριμένη εφαρμογή θα εκτελείται σε κινητά με λειτουργικό σύστημα android.

## Summary

The objective of this project is to select personal data from an android-based phone and update the personal ontology accordingly.

## Περιεχόμενα

Πίνακας Εικόνων.....	5
Πίνακας Πινάκων .....	6
1. Εισαγωγή .....	7
2. Βιβλιογραφική έρευνα .....	8
2.1. PROMPT .....	8
2.1.1. Graph-based ontology mapping .....	8
2.2. FCA-MERGE .....	9
2.3. Context Dimension Tree .....	12
2.4. FLAME2008 .....	13
2.5. Quick Ontology Mapping (QOM).....	15
2.5.1. Χαρακτηριστικά .....	16
2.5.2. Μέτρηση ομοιότητας.....	16
2.6. SPARQL.....	17
3. Υλοποίηση.....	19
3.1. Μοντέλο.....	19
3.2. Αρχιτεκτονική.....	19
3.3. Λειτουργίες .....	22
3.3.1. Content.....	22
3.3.2. Import .....	22
3.3.3. Export .....	23
3.4. Components.....	23
3.4.1. Cell Phone Application .....	23
3.4.2. Cell Phone Application Library.....	24
3.4.2. Αρχεία RDF .....	24
3.5. Επίδειξη Λειτουργίας.....	24
4. Συμπεράσματα.....	29
5. Βιβλιογραφία .....	31
6. Παράρτημα Α' - FTP Server .....	32
7. Παράρτημα Β' - Android Emulator .....	34
7.1. Δημιουργία Emulator .....	34
7.2. Εισαγωγή Τεστ Δεδομένων.....	34
7.2.1. Google Account .....	34
7.2.2. Επαφές .....	35

7.2.3. Μηνύματα.....	35
7.2.3. Ημερολόγιο .....	35
8. Παράρτημα Γ' - RDF Αρχεία .....	36
8.1. Επαφές.....	36
8.2. Μηνύματα.....	38
8.3. Εγγραφές Ημερολογίου .....	39
9. Παράρτημα Δ' - FileZilla Logs.....	41

## Πίνακας Εικόνων

Εικόνα 1, Λειτουργία συγχώνευσης στο iPrompt.....	9
Εικόνα 2, Λειτουργία FCA MERGE.....	10
Εικόνα 3, Παράδειγμα FCA-MERGE .....	11
Εικόνα 4, Παράδειγμα FCA-MERGE (συνέχεια).....	11
Εικόνα 5, Παράδειγμα Context Dimension Tree .....	12
Εικόνα 6, Context Integrator .....	13
Εικόνα 7, Παράδειγμα ER διάγραμμα μοντέλου χρήστη.....	14
Εικόνα 8, Παράδειγμα Quick Ontology Mapping .....	15
Εικόνα 9, Πίνακας σύγκρισης .....	17
Εικόνα 10, Λειτουργία SPARQL.....	18
Εικόνα 11, Αλληλεπίδραση εφαρμογής με κινητό τηλέφωνο.....	20
Εικόνα 12, Ανταλλαγή αρχείων μεταξύ κινητού τηλεφώνου και άλλης συσκευής.....	22
Εικόνα 13, Κάρτα μνήμης κινητού τηλεφώνου πριν την διαδικασία των import και export .	25
Εικόνα 14, Επιλογή μενού Status.....	25
Εικόνα 15, Επιλογή μενού Export και πάτημα κουμπιού Export .....	26
Εικόνα 16, Επιλογή μενού Import και πάτημα κουμπιού Import .....	27
Εικόνα 17, Επιλογή μενού Settings.....	28
Εικόνα 18, Κάρτα μνήμης κινητού τηλεφώνου μετά την διαδικασία των import και export	28
Εικόνα 19, Αρχική οθόνη.....	32
Εικόνα 20, Εισαγωγή / τροποποίηση χρήση.....	33
Εικόνα 21, Εισαγωγή / τροποποίηση φακέλων.....	33
Εικόνα 22, Android emulator.....	34

## Πίνακας Πινάκων

Πίνακας 1, Παράδειγμα Quick Ontology Mapping .....	15
Πίνακας 2, Λειτουργίες υλοποίησης.....	19
Πίνακας 3, Αρχεία που χρησιμοποιεί η υλοποίηση.....	21

## 1. Εισαγωγή

Τη σημερινή εποχή διαδραματίζονται σημαντικές αλλαγές καθώς ο ερχομός του ιντερνέτ και της ανάπτυξης των επικοινωνιών έχουν αλλάξει δραματικά το ενδιαφέρον στον χώρο της ανάκτησης πληροφοριών, από την μείωση της πολυπλοκότητας στην ανάκτηση πληροφοριών και της οργάνωσης τους, σε αυτό της δραματικής μείωσης της ροής διαθέσιμων πληροφοριών.

Είναι γεγονός ότι παραδοσιακά προβλήματα όπως της ενοποίησης δεδομένων (data integration) δεν έχουν ακόμα επιλυθεί και πολλά ακόμα ζητήματα παραμένουν ανοιχτά: ενοποίηση πηγών πληροφοριών με άγνωστα εκ των προτέρων στοιχεία, αυτοματοποιημένη εξαγωγή σημασιολογικών στοιχείων, φιλτράρισμα δεδομένων κλπ. Την ίδια στιγμή η κινητικότητα ανθρώπων και δεδομένων έχει γίνει σημαντικός παράγοντας της ζωής των ανθρώπων καθώς τονίζει την ανάγκη να επιλυθούν παραδοσιακά ζητήματα αναμειγνύοντας τα με καινούρια.

Σε αυτό το ζήτημα που μας απασχολεί σημαντικό ρόλο διαδραματίζουν οι οντολογίες. Ευρέως διαδεδομένες είναι οι τεχνολογίες RDF και OWL που χρησιμοποιούνται όλο και περισσότερο κυρίως στο Web. Τα τελευταία χρόνια όμως το τοπίο έχει αρχίσει να αλλάζει καθώς χρησιμοποιούνται όλο και περισσότερο smart phones τα οποία έχουν ολοένα και αυξανόμενες δυνατότητες σύνδεσης με άλλες πλατφόρμες (π.χ. Outlook, Gmail, κλπ.). Τα smart phones έχουν γίνει αναπόσπαστο κομμάτι της καθημερινότητάς μας και ολοένα και περισσότεροι άνθρωποι βασίζονται σε αυτά όχι μόνο για επικοινωνία αλλά και για την δουλειά τους.

Η λύση που έχει υλοποιηθεί για κινητά τηλέφωνα με λειτουργικό σύστημα Android αποτελείται από τα ακόλουθα τρία (3) components:

1. Cell Phone Application
2. Cell Phone Application Library
3. Αρχεία RDF

Ο πυρήνας της υλοποίησης είναι το **Cell Phone Application** το οποίο χρησιμοποιεί το **Cell Phone Application Library** και παράγει τα **Αρχεία RDF**. Ο χρήστης της Android συσκευής χρειάζεται να εγκαταστήσει μόνο το Cell Phone Application. Τόσο το Cell Phone Application Library όσο και τα Αρχεία RDF δεν είναι εμφανή στο χρήστη και δεν αντιλαμβάνεται την ύπαρξή τους.

## 2. Βιβλιογραφική έρευνα

Οντολογίες - ενέργειες διαχείρισης:

- Συγχώνευση
- Ευθυγράμμιση
- Διαχείριση εκδόσεων

Εργαλεία για:

- Διαχείριση πολλαπλών οντολογιών
- Παραπάνω ενέργειες διαχείρισης

Αλγόριθμοι για τον εντοπισμό συσχετίσεων μεταξύ εννοιών σε διαφορετικές οντολογίες.

### 2.1. PROMPT

Το PROMPT διαθέτει ένα γνωσιακό μοντέλο και μια σειρά εργαλείων για να υποστηρίξει τα παραπάνω. Συγκεκριμένα, το iPrompt είναι ένα διαδραστικό εργαλείο για την συγχώνευση οντολογιών. Το εργαλείο αυτό κατευθύνει τον χρήστη σε μία διαδικασία συγχώνευσης προτείνοντας τι πρέπει να συγχωνευθεί, ανιχνεύοντας μη συμβατότητες μεταξύ των οντοτήτων και πιθανά προβλήματα. Παράλληλα προτείνει στρατηγικές επίλυσης των προβλημάτων αυτών. Το iPrompt χρησιμοποιεί την δομή των εννοιών σε μία οντολογία και τις συσχετίσεις μεταξύ τους καθώς επίσης και τη πληροφορία που παίρνει από τις ενέργειες του χρήστη (πχ αν το εργαλείο εκτιμά ότι δύο κλάσεις ενδεχομένως να είναι όμοιες και ο χρήστης έχει ήδη ενώσει κάποιες από τις υποκλάσεις τους το εργαλείο θα είναι ακόμα πιο σίγουρο ότι οι κλάσεις είναι όμοιες).

#### 2.1.1. Graph-based ontology mapping

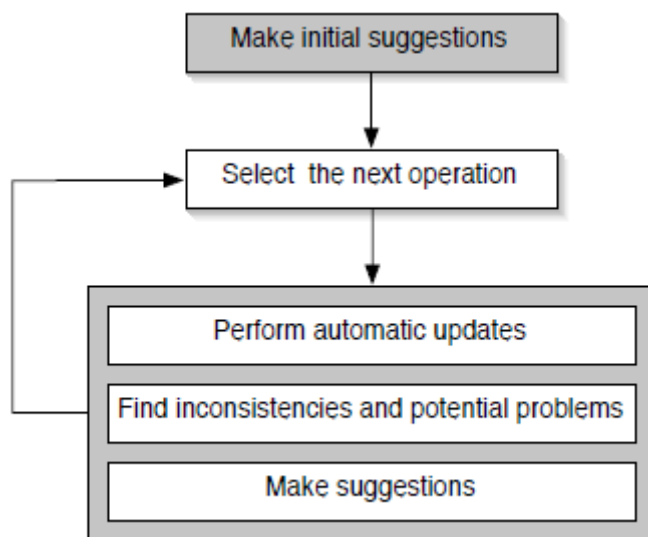
Το iPrompt δημιουργεί πρώτα μία αρχική λίστα από διασταυρώσεις (matches) βασισμένο σε λεκτική ομοιότητα των ονομάτων των κλάσεων. Κατόπιν εκτελεί ένα κύκλο εργασιών:

1. Ο χρήστης επιλέγει μία από τις προτάσεις του iPrompt ή
2. Χρησιμοποιεί κάποιο εργαλείο επεξεργασίας οντοτήτων για να κατευθύνει ο ίδιος τις τροποποιήσεις

Το εργαλείο εκτελεί την διαδικασία ένωσης, εκτελεί αυτόματα τις περαιτέρω αλλαγές ως επίπτωση της συγκεκριμένης απόφασης, δημιουργεί μία λίστα υποψήφιας διαμορφώσεων στην οντολογία με βάση την αρχική δομή της, τις παραμέτρους και αφού εντοπίσει ασυμβατότητες προτείνει και συγκεκριμένες λύσεις.

Κατά τη διάρκεια της συγχώνευσης το εργαλείο (αυτόματα ή με τη συμβολή του χρήστη) αντιγράφει πλαίσια (frames) από τις οντότητες-πηγές προς την οντολογία προς συγχώνευση, ενώνει πλαίσια μεταξύ τους και κάνει αλλαγές στα πλαίσια της οντολογίας προς συγχώνευση για κάθε πλαίσιο στις οντολογίες-πηγές και κρατάει το ίχνος της στο αντίστοιχο πλαίσιο στην οντολογία προς συγχώνευση.





Εικόνα 1, Λειτουργία συγχώνευσης στο iPrompt

Το σύνολο των λειτουργιών που εκτελούνται κατά τη συγχώνευση περιλαμβάνει τις εργασίες που πραγματοποιούνται σε μία παραδοσιακή οντολογία καθώς και εκείνες που σχετίζονται με συγχώνευση και προτεραιοποίηση: συγχώνευση κλάσεων, συγχώνευση ιδιοτήτων (slots), συγχώνευση αντικειμένων (instances) και αντιγραφή σε βάθος ή σε ρηχό επίπεδο μίας κλάσης από μία οντολογία στην άλλη.

Η συγχώνευση των οντολογιών δεν είναι πάντα στις προτιμήσεις του χρήστη. Μερικές φορές ο χρήστης προτιμά να διατηρεί την βασική οντολογία χωριστή για διαφορετικούς λόγους: συμβατότητα με δεδομένα και εργαλεία χρησιμοποιώντας τις βασικές οντολογίες, συντήρηση των συσχετίσεων μεταξύ οντολογιών όταν οι βασικές οντολογίες μεταβάλλονται και άλλα.

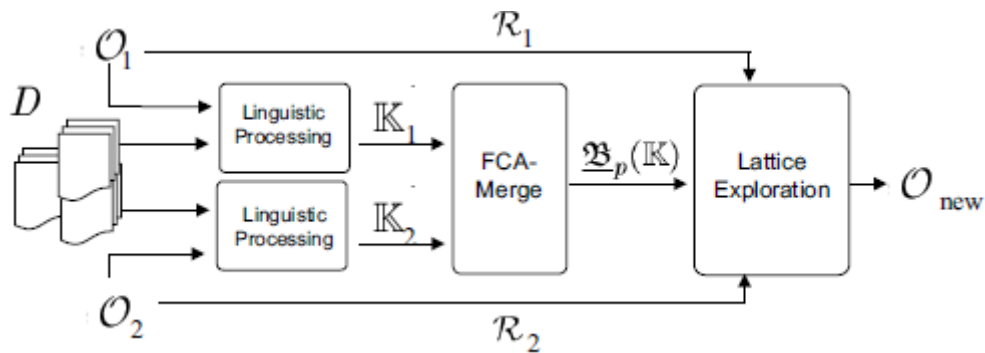
Η καρδιά της διαδικασίας συγχώνευσης του IPROMPT είναι η ακόλουθη:

- Προσδιορίζονται το σύνολο των εργασιών που απαιτούνται για συγχώνευση ή τακτοποίηση
- Για κάθε ενέργεια ορίζονται 1) οι αλλαγές που εκτελούνται αυτόματα, 2) οι αλλαγές που προτείνονται στον χρήστη και 3) πιθανά προβλήματα που θα προκύψουν και απαιτούν το ενδιαφέρον του χρήστη
- Εφόσον ο χρήστης επιλέγει μία από τις ενέργειες, το iPrompt δημιουργεί τα μέλη των 3 αυτών συνόλων βασισμένο στις εκάστοτε παραμέτρους.
- Κατά τη συγχώνευση το εργαλείο αντιγράφει πλαίσια από τις αρχικές οντολογίες στην οντολογία προς συγχώνευση, ενώνει πλαίσια ανάμεσα τους και κάνει αλλαγές στα πλαίσια της οντολογίας προς συγχώνευση.

## 2.2. FCA-MERGE

Σε πρώτο βήμα η μέθοδος δημιουργεί για κάθε οντολογία ένα αφηρημένο πλαίσιο εννοιών (formal context). Το σύνολο των κειμένων λαμβάνεται υπόψη ως σύνολο αντικειμένων και το σύνολο των εννοιών λαμβάνεται υπόψη ως σύνολο ιδιοτήτων. Η απεικόνιση αυτή γίνεται εύκολα και γρήγορα, ωστόσο το δύσκολο βήμα είναι η δημιουργία των δυαδικών

σχέσεων μεταξύ των δύο συνόλων. Η σχέση μεταξύ ενός κειμένου-αντικείμενου και της έννοιας-ιδιότητας διατηρείται όταν ένα αντικείμενο περιέχει ένα ορισμό ιδιότητας.



Εικόνα 2, Λειτουργία FCA MERGE

Ο παραπάνω υπολογισμός εφαρμόζει γλωσσολογικές τεχνικές οι οποίες έχουν επιπλέον αναμειχθεί με μία προσέγγιση εξαγωγής πληροφοριών η οποία έχει υλοποιηθεί πάνω από ένα επεξεργαστή κειμένου στη Γερμανική γλώσσα (SMES - Saarbrücken Message Extraction System). Η αρχιτεκτονική του συστήματος αποτελείται από ένα επεξεργαστή συμβόλων που βασίζεται σε κανόνες κανονικών εκφράσεων, ένα λεξικό αναλυτή που βασίζεται σε ένα λεξικό γενικευμένων αλλά και εξειδικευμένων λέξεων ανά πεδίο, και ένα συντακτικό αναλυτή για μπλοκ εκφράσεων. Ο επεξεργαστής συμβόλων ανιχνεύει το κείμενο έτσι ώστε να εντοπίσει όρια λέξεων και σύνθετες εκφράσεις.

Το λεξικό περιέχει περισσότερες από 120000 καταχωρήσεις από ρίζες λέξεων και περισσότερα από 12000 πλαίσια κατηγοριοποίησης αυτών των λέξεων. Τα πλαίσια αυτά είναι χρήσιμα στην λεξική ανάλυση και στη συντακτική ανάλυση μπλοκ εκφράσεων. Πολύ περισσότερο το εξειδικευμένο μέρος ανά πεδίο του λεξικού περιέχει λεκτικές καταχωρήσεις που δηλώνουν αναπαραστάσεις εννοιών και συσχετίσεων μεταξύ τους σε φυσική γλώσσα. Οι λεκτικές καταχωρήσεις μπορεί να αναφέρονται σε πολλές έννοιες ή σχέσεις ενώ μία έννοια ή σχέση μπορεί να αναφέρεται από πολλές λεξικές καταχωρήσεις.

Η λεκτική ανάλυση χρησιμοποιεί το λεξικό για να εκτελέσει:

1. μορφολογική ανάλυση, δηλαδή προσδιορισμό της ρίζας μίας σειράς λέξεων και ανάλυση σύνθετων μορφών.
2. αναγνώριση εννοιών που έχουν ονομαστεί
3. κατηγοριοποίηση της κάθε λέξης ως μέρος του λόγου και
4. ανάκτηση εξειδικευμένης πληροφορίας για κάθε λέξη στο πεδίο της εφαρμογής

Ενώ τα τρία πρώτα βήματα θεωρούνται ως μία κλασική προσέγγιση για εξαγωγή πληροφορίας, το τέταρτο βήμα έχει ιδιαίτερο ενδιαφέρον για την μέθοδο FCA-MERGE όσον αφορά την εξαγωγή παρουσίας εννοιών. Πράγματι αυτό το βήμα αντιστοιχίζει μεμονωμένες λέξεις ή σύνθετες εκφράσεις με μία έννοια της οντολογίας εφόσον υπάρχει αντίστοιχη καταχώρηση στο μέρος του λεξικού που αφορά το πεδίο της εφαρμογής.

Παρακάτω απεικονίζεται παράδειγμα της εκτέλεσης του πρώτου βήματος σε μία εφαρμογή που παρέχει τουριστικές πληροφορίες (πχ ονόματα ξενοδοχείων). Οι δύο πίνακες ορίζουν

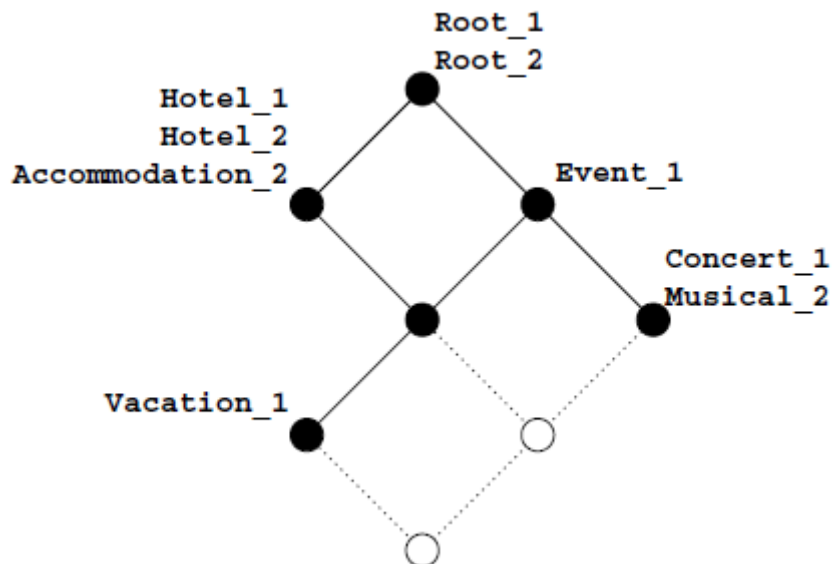
δύο διαφορετικά πλαίσια εννοιών δύο διαφορετικών οντολογιών. Σε κάθε κείμενο που εντοπίζονται συνδυασμοί παρουσίας εννοιών απεικονίζονται στη κάθε εγγραφή.

$I_1$	Vacation	Hotel	Event	Concert	Root
doc1	x	x	x	x	x
doc2	x	x	x	x	x
doc3	x	x	x		x
doc4	x	x	x	x	x
doc5			x	x	x
doc6		x	x	x	x
doc7		x			x
doc8	x	x	x	x	x
doc9	x	x	x		x
doc10	x	x	x		x
doc11	x	x	x	x	x
doc12		x			x
doc13		x	x	x	x
doc14	x	x	x		x

$I_2$	Hotel	Accommodation	Musical	Root
doc1	x	x	x	x
doc2	x	x		x
doc3	x	x	x	x
doc4	x	x	x	x
doc5			x	x
doc6	x	x	x	x
doc7	x	x		x
doc8	x	x	x	x
doc9	x	x		x
doc10	x	x		x
doc11	x	x	x	x
doc12	x	x		x
doc13	x	x	x	x
doc14	x	x		x

Εικόνα 3, Παράδειγμα FCA-MERGE

Στο δεύτερο βήμα της μεθόδου δημιουργείται ένα δίκτυο εννοιών που έχει συμπυκωθεί μετά τη διαδικασία της συγχώνευσης. Για να επιτευχθεί αυτό γίνεται φιλτράρισμα των αμφισημιών που εντοπίζονται στα σύνολα των ιδιοτήτων (χαρακτηριστικών). Η κάθε αφηρημένη έννοια αναπαριστάται από ένα κόμβο. Οι κενοί κόμβοι δείχνουν σε έννοιες που έχουν φιλτραριστεί και δεν προβάλλονται στον χρήστη. Επίσης, μία έννοια είναι υποκλάση μίας άλλης έννοιας αν και μόνο αν μπορούμε να φτάσουμε σε αυτή την έννοια διασχίζοντας ένα μονοπάτι συγγενικών σχέσεων.

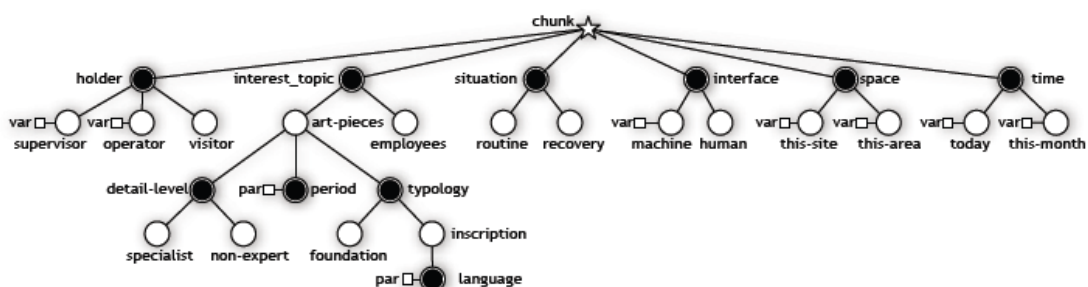


Εικόνα 4, Παράδειγμα FCA-MERGE (συνέχεια)

Το επόμενο βήμα της παραγωγής της οντολογίας προς συγχώνευση από το παραπάνω δίκτυο απαιτεί αλληλεπίδραση με τον χρήστη καθώς βασίζεται σε μεγάλο βαθμό στην εμπειρία και τη γνώση του ειδικού στο συγκεκριμένο πεδίο εφαρμογών.

### 2.3. Context Dimension Tree

Το σύστημα Context-ADDICT προσπαθεί να αντιμετωπίσει αρκετά προβλήματα που αφορούν την ενοποίηση πληροφοριών δίνοντας ιδιαίτερη έμφαση στην ιδέα του πλαισίου (context) σχεδιασμού βάσεων δεδομένων για εφαρμογές κινητών συσκευών.



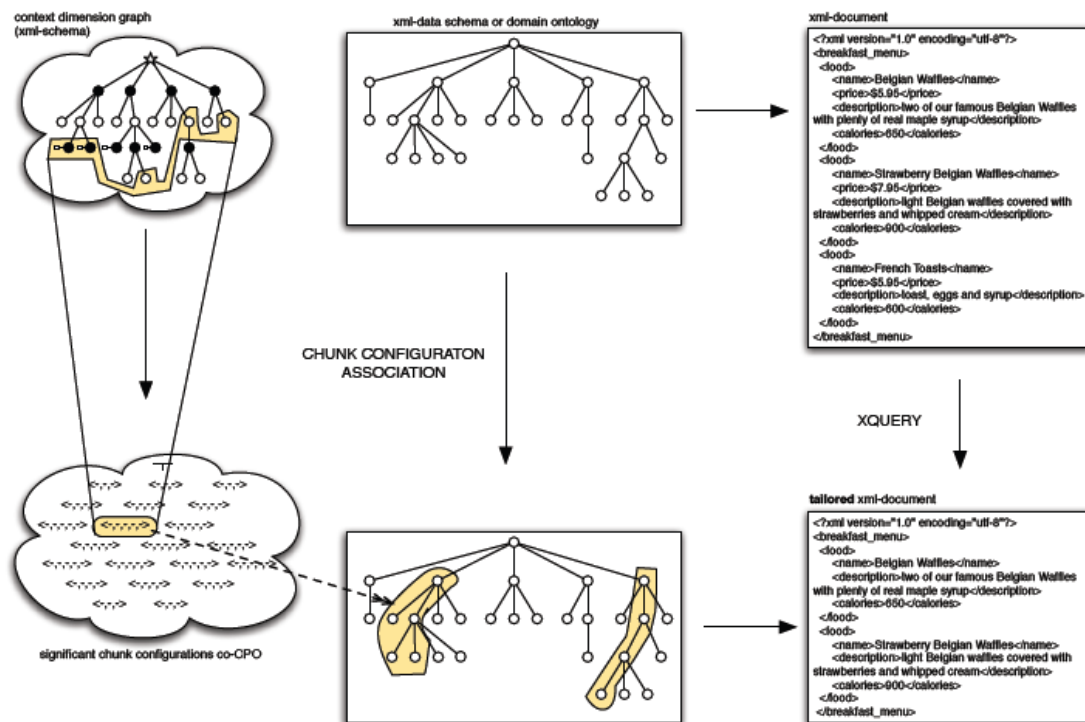
Εικόνα 5, Παράδειγμα Context Dimension Tree

Το προτεινόμενο μοντέλο για τον ορισμό ενός πλαισίου πληροφοριών σε ένα συγκεκριμένο πεδίο εφαρμογών ονομάζεται Context Dimension Tree. Το μοντέλο αυτό παίζει σημαντικό ρόλο στο ράψιμο στοχευόμενων δεδομένων μίας εφαρμογής και σύμφωνα με τις ανάγκες του χρήστη. Το δέντρο μπορεί να συμβολίζει είτε μία οντολογική αναπαράσταση σε OWL είτε μία XML αναπαράσταση. Στο Context Dimension Tree, τα παιδιά κάτω από την ρίζα ορίζουν τις διαστάσεις του πλαισίου που συμβολίζουν τα διαφορετικά χαρακτηριστικά των χρηστών των συσκευών και του πλαισίου μέσα στο οποίο ενεργούν. Προσδιορίζοντας μία τιμή για κάθε διάσταση, ένα σημείο στον πολυδιάστατο χώρο μπορεί να οριστεί ανάμεσα σε πολλούς άλλους συνδυασμούς. Η τιμή μίας διάστασης μπορεί να αναλυθεί ακόμη περισσότερο δημιουργώντας τα επόμενα επίπεδα του δέντρου. Η δομή που προκύπτει ένα συγκεκριμένο πλαίσιο ή ένα μπλοκ παραμετροποίησης καθορίζοντας ένα κομμάτι του συνόλου των δεδομένων που σχεδιάζεται αρχικά για να χρησιμοποιηθεί κατά την εκτέλεση της εφαρμογής.

Η οντολογία ενός συγκεκριμένου πεδίου συλλαμβάνει τις κύριες έννοιες του πληροφοριακού πεδίου έτσι ώστε να παρέχει μία γενικά αποδεκτή ταξινόμια εννοιών και σχέσεων μεταξύ τους. Από τη μία η οντολογία του πληροφοριακού πεδίου αναλύει γνώσεις που συγκεντρώνονται απαντώντας στις ερωτήσεις:

- ποιες είναι οι ενέργειες που πραγματοποιούν οι χρήστες καθώς και τα ενδιαφέροντα τους σε αυτό το πεδίο εφαρμογών;
- Ποιο μέρος του πεδίου είναι σχετικό με τα παραπάνω σε ένα συγκεκριμένο πλαίσιο;

Η διασύνδεση της πληροφορίας απαντώντας στα παραπάνω γίνεται από συγκεκριμένο εργαλείο του προτεινόμενου συστήματος: ο Context Integrator είναι ένα διαδραστικό εργαλείο που δίνει τη δυνατότητα στον αναλυτή να ορίσει τη σχέση ανάμεσα στο πλαίσιο και το σχήμα του πληροφοριακού πεδίου.

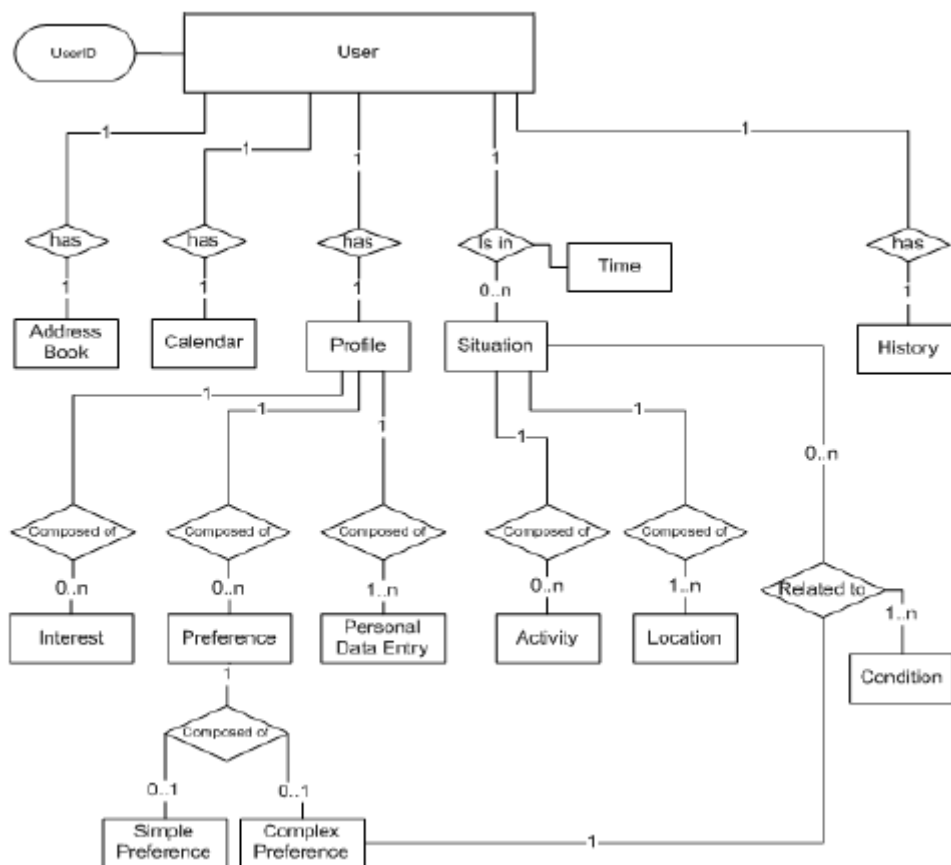


Εικόνα 6, Context Integrator

## 2.4. FLAME2008

Το σύστημα FLAME2008 επικεντρώνεται στην παροχή προσωποποιημένων πληροφοριακών υπηρεσιών μέσω της διεπαφής κινητών συσκευών. Αυτό αυτόματα συνεπάγεται τη διατήρηση πληροφοριών για τους χρήστες ως ένα προφίλ. Η δομή ενός προφίλ περιλαμβάνει συνήθως στοιχεία όπως όνομα, διεύθυνση, προτιμήσεις/ενδιαφέροντα, προσωπικές επαφές, ημερολόγιο δραστηριοτήτων κλπ. Επιπλέον ένα πλαίσιο πληροφοριών για τον χρήστη περιλαμβάνει την τοποθεσία του χρήστη σε μία ορισμένη χρονική στιγμή. Για παράδειγμα η παροχή χώρο-κεντρικών υπηρεσιών βασίζονται στην χρήση πληροφοριών σε ένα πλαίσιο για την εξαγωγή περισσότερων σημασιολογικών στοιχείων.

Το ER διάγραμμα ενός μοντέλου χρήστη απεικονίζεται παρακάτω:



Εικόνα 7, Παράδειγμα ER διάγραμμα μοντέλου χρήστη

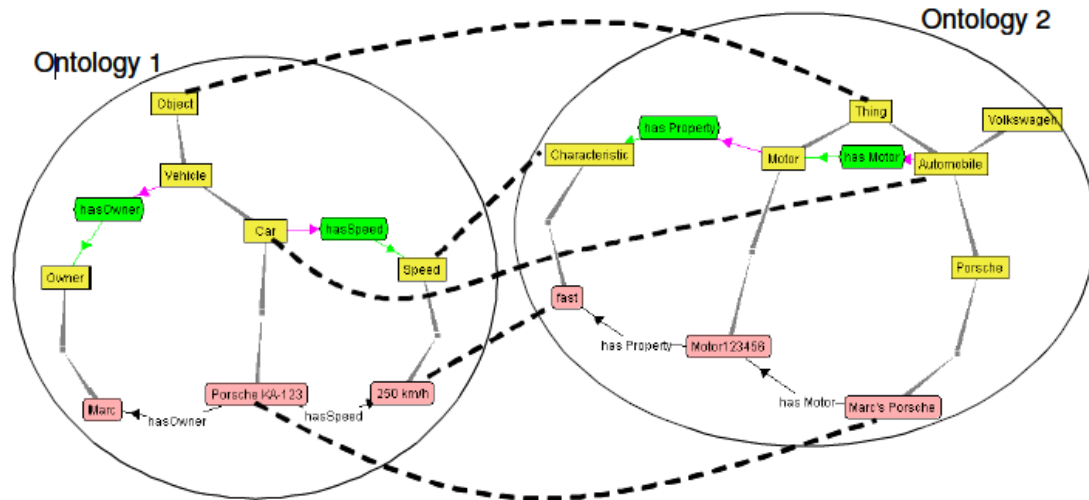
Για παράδειγμα η σχέση ενός χρήστη με ένα άλλο χρήστη ορίζει μία εγγραφή στο βιβλίο επαφών. Συσχετίσεις μεταξύ χρηστών προκύπτουν επίσης από τα στοιχεία επικοινωνίας (πχ μηνύματα). Επίσης το ημερολόγιο περιέχει στοιχεία για το τι ο κάθε χρήστης προγραμματίζει να κάνει. Για την αναπαράσταση των παραπάνω σε OWL εφαρμόζονται γενικευμένες βιβλιοθήκες οντολογιών:

- Οντολογία τοποθεσίας/χώρου
- Οντολογία χρόνου
- Οντολογία περιεχομένου
- Οντολογίες εργασιών (υπηρεσίες, καταστάσεις)

Με βάση τις παραπάνω οντολογίες η δυναμική πληροφορία στο προφίλ του χρήστη και στο πλαίσιο πληροφοριών του χρήστη συνδυάζονται για να προσδιορίσουν μια σειρά καταστάσεων και υπηρεσιών που σχετίζονται εκείνη τη στιγμή με τον χρήστη. Το προφίλ της συγκεκριμένης κατάστασης που δημιουργείται εκείνη τη στιγμή για τον χρήστη συγκρίνεται σημασιολογικά με όλα τα γενικευμένα προφίλ καταστάσεων που υπάρχουν στο σύστημα. Αυτό πετυχαίνεται χρησιμοποιώντας ξεχωριστούς τελεστές σύγκρισης για κάθε διάσταση οι οποίοι λαμβάνουν υπόψη τις συσχετίσεις μεταξύ των τιμών της υποψήφιας κατάστασης και των διαθέσιμων. Αυτές οι συσχετίσεις ορίζονται ως αξιώματα στην οντολογία.

## 2.5. Quick Ontology Mapping (QOM)

Δύο οντολογίες O1 και O2 που περιγράφουν το πεδίο της αγοράς αυτοκινήτων δίνονται στο παρακάτω παράδειγμα. Οι διακεκομμένες γραμμές εμφανίζουν μία λογική απεικόνιση ανάμεσα στις δύο αυτές οντολογίες.



Εικόνα 8, Παράδειγμα Quick Ontology Mapping

Οντολογία 1	Οντολογία 2
Αντικείμενο	Πράγμα
Αυτοκίνητο	Όχημα
Porsche KA-123	Marc's Porsche
Ταχύτητα	Χαρακτηριστικό
250 km/h	γρήγορο

Πίνακας 1, Παράδειγμα Quick Ontology Mapping

Η διαδικασία που προτείνεται υλοποιεί μία –προς – μία απεικονίσεις όπως η παραπάνω ως εξής:

1. Τεχνολογία Χαρακτηριστικών (Feature Engineering) μετατρέπει την αρχική αναπαράσταση των οντολογιών σε μία αναπαράσταση που επιτρέπει τον υπολογισμό ομοιοτήτων. Στο βήμα αυτό μπορεί να είναι απαραίτητο να ασχοληθεί η διαδικασία μόνο με ένα υποσύνολο από RDFS συστατικά ή μπορεί να είναι χρήσιμη η εφαρμογή κανόνων κατηγοριοποίησης για πιο σύνθετους μετασχηματισμούς.
2. Επιλογή των επόμενων στοιχείων αναζήτησης: τα στοιχεία αυτά αναζητούνται σε ένα ευρύτερο πεδίο αναζήτησης από υποψήφιας αντιστοιχίες
3. Ο υπολογισμός ομοιοτήτων επιστρέφει τιμές από την σύγκριση υποψήφιων αντιστοιχιών με βάση τους ορισμούς των δύο οντολογιών
4. Για κάθε υποψήφιο ζεύγος οντοτήτων από τις δύο οντολογίες θα υπάρχουν πολλαπλές τιμές που υποδεικνύουν ομοιότητες. Οι τιμές αυτές συγκεντρώνονται σε μία τιμή.

5. Η διαδικασία της μετάφρασης χρησιμοποιεί τις ατομικές ή συγκεντρωτικές τιμές για να δημιουργήσει αντιστοιχίες ανάμεσα σε οντότητες των δύο οντολογιών. Διάφοροι μηχανισμοί στη περίπτωση αυτοί εφαρμόζουν κατώτατα όρια για συγκρίσεις ομοιότητας, ή εφαρμόζουν δομημένη γνώση και κριτήρια ομοιότητας.
6. Επαναλήψεις: αρκετοί αλγόριθμοι επαναλαμβάνουν τα παραπάνω βήματα έτσι ώστε να ενισχύσουν τη δομημένη γνώση. Οι επαναλήψεις σταματάνε όταν δεν προτείνονται νέες αντιστοιχίες.

### 2.5.1. Χαρακτηριστικά

Τα χαρακτηριστικά πηγάζουν από τις ιδιότητες των οντοτήτων (έννοιες, συσχετίσεις, αντικείμενα) και από άλλα πιθανά χαρακτηριστικά όπως:

- Αναγνωριστικά: πχ συμβολοσειρές με ειδικό format πχ URIs ή RDF ετικέτες.
- RDF/S τύποι: όπως ιδιότητες ή σχέσεις μεταξύ υποκλάσεις
- Εξαγόμενα στοιχεία: αυτά περιορίζουν ή επεκτείνουν απλούς RDFS τύπους (π.χ. mostspecific-class-of-instance)
- Συγκεντρωτικά χαρακτηριστικά: π.χ. η συγκέντρωση περισσότερων του ενός RDFS τύπου (πχ sibling)
- OWL τύποι: πχ ορισμός ομοιοτήτων με βάση τον ορισμό sameAs συσχετίσεων
- Εξειδικευμένα χαρακτηριστικά που εφαρμόζονται σε ένα συγκεκριμένο πεδίο με μία οντολογία που προϋπάρχει και είναι κοινή.

Επιπλέον οι ετικέτες (Labels) είναι αναγνωριστικά που χρησιμοποιούν οι άνθρωποι για οντότητες συνήθως όσων μοιράζονται μία κοινά αποδεκτή γλώσσα. Οι έννοιες είναι τακτοποιημένες σε μία ταξινομία και ένας κανόνας αντιστοίχισης θα ήταν: αν οι εν μέρει έννοιες είναι ίδιες τότε και οι κύριες έννοιες θα είναι ίδιες. Όσο πιο κοντά είναι σε ομοιότητες οι εν μέρει έννοιες στις κύριες έννοιες τόσο πιο στενά συμπεράσματα μπορούν να εξαχθούν για τις ομοιότητες τους. Επιπλέον, οι έννοιες είναι γενικεύσεις των αντικειμένων. Επομένως αν δύο έννοιες αποτελούνται από όμοια αντικείμενα τότε είναι ίδιες.

### 2.5.2. Μέτρηση ομοιότητας

Αυτό μπορεί να βασιστεί σε διάφορες μετρικές όπως:

- Η ισοδυναμία αντικειμένων βασίζεται σε λογικούς υπολογισμούς, ειδικά υπολογισμούς που βασίζονται στα αποτελέσματα των προηγούμενων επαναλήψεων
- Η σαφής ισοδυναμία ελέγχει την ύπαρξη κάποιου λογικού υπολογισμού που επιβάλει δύο οντότητες να είναι όμοιες
- Ισοδυναμία λεκτικών είναι ένα αυστηρό μέτρο για την σύγκριση συμβολοσειρών με την έννοια ότι πρέπει όλοι οι χαρακτήρες να είναι όμοιοι
- Η ομοιότητα λεκτικών είναι πιο χαλαρό μέτρο που ζυγίζει την ομοιότητα σε μία κλίμακα από 0 έως 1
- Dice Coefficient συγκρίνει δύο σύνολα οντοτήτων
- Simset είναι μία στατιστική προσέγγιση πολυδιάστατης αναπαράστασης των ομοιοτήτων κάνοντας χρήση διανυσμάτων. Υπολογίζοντας την απόσταση ανάμεσα



στα διανύσματα δύο οντοτήτων μπορούμε να γνωρίζουμε και την απόσταση της ομοιότητας τους

Οι παραπάνω μετρικές λαμβάνονται ως είσοδο στη φόρμουλα της συγκέντρωσης των συγκρίσεων σε μία ενιαία τιμή και με βάση αυτή τη τιμή ακολουθεί μία στρατηγική για να καταλήξει στο συμπέρασμα αν δύο οντότητες είναι όμοιες ή όχι.

Comparing	No.	Feature	Similarity Measure
Concepts	1	(label, $X_1$ )	string similarity( $X_1, X_2$ )
	2	( $URI_1$ )	string equality( $URI_1, URI_2$ )
	3	( $X_1, sameAs, X_2$ ) relation	explicit equality( $X_1, X_2$ )
	4	(direct properties, $Y_1$ )	SimSet( $Y_1, Y_2$ )
	→5a	(properties of direct super-concepts, $Y_1$ )	SimSet( $Y_1, Y_2$ )
	→6a	(direct super-concepts, $Y_1$ )	SimSet( $Y_1, Y_2$ )
	→7a	(direct sub-concepts, $Y_1$ )	SimSet( $Y_1, Y_2$ )
	8	(concept siblings, $Y_1$ )	SimSet( $Y_1, Y_2$ )
	9	(direct instances, $Y_1$ )	SimSet( $Y_1, Y_2$ )
	→10a	(instances of direct sub-concepts, $Y_1$ )	SimSet( $Y_1, Y_2$ )
Relations	1	(label, $X_1$ )	string similarity( $X_1, X_2$ )
	2	( $URI_1$ )	string equality( $URI_1, URI_2$ )
	3	( $X_1, sameAs, X_2$ ) relation	explicit equality( $X_1, X_2$ )
	4	(domain, $X_{d1}$ ) and (range, $X_{r1}$ )	object equality( $X_{d1}, X_{d2}$ ), ( $X_{r1}, X_{r2}$ )
	→5a	(direct super-properties, $Y_1$ )	SimSet( $Y_1, Y_2$ )
	→6a	(direct sub-properties, $Y_1$ )	SimSet( $Y_1, Y_2$ )
	7	(property siblings, $Y_1$ )	SimSet( $Y_1, Y_2$ )
	8	(property instances, $Y_1$ )	SimSet( $Y_1, Y_2$ )
Instances	1	(label, $X_1$ )	string similarity( $X_1, X_2$ )
	2	( $URI_1$ )	string equality( $URI_1, URI_2$ )
	3	( $X_1, sameAs, X_2$ ) relation	explicit equality( $X_1, X_2$ )
	→4a	(direct parent-concepts, $Y_1$ )	SimSet( $Y_1, Y_2$ )
	5	(property instances, $Y_1$ )	SimSet( $Y_1, Y_2$ )
Property-Instances	1	(domain, $X_{d1}$ ) and (range, $X_{r1}$ )	object equality( $X_{d1}, X_{d2}$ ), ( $X_{r1}, X_{r2}$ )
	2	(parent property, $Y_1$ )	SimSet( $Y_1, Y_2$ )

Εικόνα 9, Πίνακας σύγκρισης

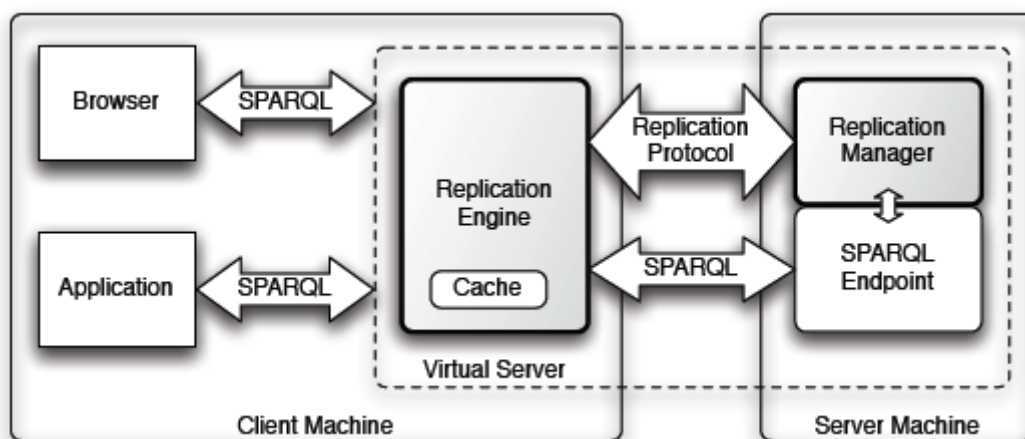
## 2.6. SPARQL

Στην παρακάτω αρχιτεκτονική το ενδιαφέρον εστιάζεται στον συγχρονισμό των δεδομένων μεταξύ δύο διαφορετικών σημείων (πχ συσκευών) που αποκαλούνται SPARQL endpoints. Ο μηχανισμός αυτός επιτρέπει τον επιλεκτικό συγχρονισμό δεδομένων μέσω του συγχρονισμού RDF γράφων. Οι αλγόριθμοι που εφαρμόζονται παρέχουν υποκειμενική ταξινόμηση RDF τριάδων από δεδομένα λαμβάνοντας υπόψη δομημένα και σημασιολογικά χαρακτηριστικά, προτιμήσεις και πλαίσιο χρήσης των δεδομένων.

Η δομή που λαμβάνεται υπόψη είναι αυτή των γραφημάτων. Ένας πρώτος τρόπος σύγκρισης σε μη πραγματικό χρόνο είναι μέσω της σύγκρισης των στοιχείων των γραφημάτων. Ο τρόπος αυτός δεν λαμβάνει υπόψη το RDF μοντέλο και τις οντολογίες που εξετάζονται συνήθως από τη δομή της οντολογίας και τα ερωτήματα (queries).

Συμπληρωματικός τρόπος αναπαράστασης των οντολογιών είναι μέσω της OWL γλώσσας έτσι ώστε να ορίζονται οι οντολογίες υπό τη μορφή κλάσεων, οντοτήτων, και ιδιοτήτων. Επομένως οι δομές και οι σημασιολογικές εκφράσεις αναπαρίστανται ως δομικά στοιχεία της OWL γλώσσας (πχ subClassOf, equivalentProperty). Από την ανάλυση αυτών των εκφράσεων αντιλαμβανόμαστε τη σημασία των δεδομένων των αντικειμένων κατά την σύγκριση ομοιοτήτων και ποια είναι τα περιττά δεδομένα που δεν πρέπει να λαμβάνονται υπόψη κατά την αντιγραφή δεδομένων.

SPARQL ερωτήματα χρησιμοποιούνται για τη πρόσβαση σε RDF δεδομένα. Ο συγχρονισμός των δεδομένων λαμβάνει υπόψη το πλαίσιο των δραστηριοτήτων του χρήστη καθώς επίσης τις προτιμήσεις του.



Εικόνα 10, Λειτουργία SPARQL

## 3. Υλοποίηση

### 3.1. Μοντέλο

Το μοντέλο που χρησιμοποιήθηκε στην υλοποίησή μας είναι συνδυασμός των PROMPT και SPARQL. Χρησιμοποιούμε την SPARQL για τον συγχρονισμό<sup>1</sup> του περιεχομένου και το PROMPT για την επίλυση των προβλημάτων που προκύπτουν κατά το συγχρονισμό. Με την χρήση των SPARQL ερωτημάτων εντοπίζουμε το περιεχόμενο που έχει τροποποιηθεί (προσθεθεί, διαγραφεί ή ενημερωθεί). Στην συνέχεια βασιζόμενοι στο PROMPT ενημερώνουμε τον χρήστη για το περιεχόμενο που έχει τροποποιηθεί και εκείνος αποφασίζει εάν θέλει να κρατήσει το περιεχόμενο που ήδη έχει ή να το ενημερώσει με τις τροποποιήσεις που έχουν γίνει. Η εικόνα 1 του παρόντος κειμένου που περιγράφει την διαδικασία συγχώνευσης από το iPrompt έχει υλοποιηθεί και στην δικιά μας υλοποίηση. Η διαφορά έγκειται στο ότι το περιεχόμενο που θα συγχωνευτεί το έχουμε βρει χρησιμοποιώντας από την SPARQL.

### 3.2. Αρχιτεκτονική

Σκοπός της παρούσας υλοποίησης είναι η ανάπτυξη μιας εφαρμογής που θα κάνει export περιεχόμενο από κινητό και θα μπορεί να το συγχρονίζει με μία άλλη συσκευή. Το μόνο που θα χρειάζεται να κάνει ο χρήστης του κινητού είναι να δηλώσει τα στοιχεία για τον FTP Server (IP, username και password) που χρησιμοποιεί η άλλη συσκευή. Το κινητό θα πρέπει να έχει λειτουργικό σύστημα Android και θα συγχρονίζεται περιεχόμενο επαφών, μηνυμάτων και εγγραφών ημερολογίου.

Για την παρούσα υλοποίηση χρησιμοποιήσαμε την βιβλιοθήκη Apache Jena το οποίο είναι ένα framework για την ανάπτυξη Semantic Web Applications. Η βιβλιοθήκη Apache Jena έχει γίνει port στο λειτουργικό σύστημα Android με την ονομασία androidjena. Για να χρησιμοποιήσουμε την μηχανή ερωτημάτων SPARQL κάναμε port την βιβλιοθήκη ARQoid του Android. Επιπλέον, χρησιμοποιήσαμε και την βιβλιοθήκη ftp4j για να κάνουμε upload και download από FTP Server τα αρχεία RDF.

Στον ακόλουθο πίνακα παρουσιάζονται οι λειτουργίες που επιτελεί η υλοποίησή μας.

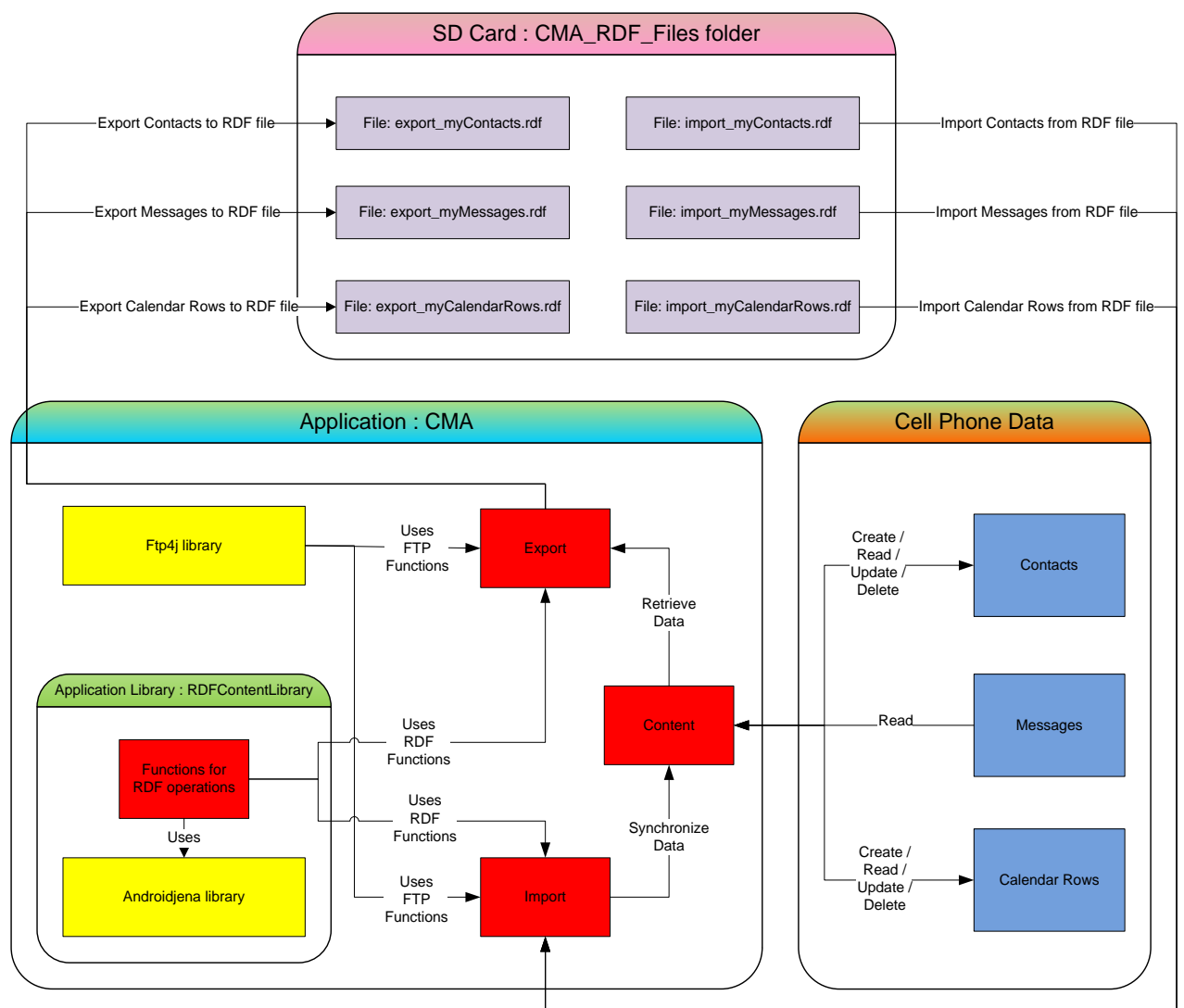
Component	Περιγραφή	APIs / Βιβλιοθήκες
Content	Διάβασμα του περιεχομένου του κινητού.	Android API
Import	Λήψη από FTP Server αρχείων RDF που περιέχουν περιεχόμενο και συγχρονισμό με το υπάρχων περιεχόμενο της συσκευής.	Android API androidjena ARQoid ftp4j
Export	Δημιουργία αρχείων RDF από το περιεχόμενο της συσκευής και αποστολή τους στον FTP Server.	Android API androidjena ftp4j

Πίνακας 2, Λειτουργίες υλοποίησης

**Σημείωση:** Δεν έχει αναπτυχθεί όλη η λειτουργία του Import παρά μόνο η λήψη των αρχείων RDF. Όλες οι άλλες λειτουργίες έχουν αναπτυχθεί πλήρως.

<sup>1</sup> Δεν έχει υλοποιηθεί ο συγχρονισμός στην παρούσα έκδοση της εφαρμογής, μόνο περιγράφεται το πως μπορεί να υλοποιηθεί.

Στην εικόνα που ακολουθεί σας παρουσιάζουμε σχηματικά την υλοποίησή μας και πως αυτή αλληλεπιδρά με το κινητό τηλέφωνο του χρήστη. Το component Content εκτελεί λειτουργίες CRUD (create, read update και delete) στο περιεχόμενο του κινητού τηλεφώνου, με εξαίρεση τα μηνύματα (SMS) τα οποία μόνο διαβάζει. Το component export μετασχηματίζει το περιεχόμενο του κινητού που έχει διαβάσει το component Content σε RDF μοντέλα και τα αποθηκεύει σε αρχεία στην εξωτερική κάρτα μνήμης του κινητού τηλεφώνου. Στην συνέχεια στέλνει τα αυτά τα αρχεία σε μια άλλη συσκευή με FTP. Στον αντίποδα το component Import λαμβάνει από την άλλη συσκευή τα αρχεία RDF με FTP. Τα αρχεία που έχουν ληφθεί στην εξωτερική κάρτα μνήμης το component Import τα διαβάζει και τα συγχρονίζει με το περιεχόμενο του κινητού τηλεφώνου. Τα component Import και Export χρησιμοποιούν την βιβλιοθήκη που έχουμε αναπτύξει (RDFContentLibrary) και η οποία τους παρέχει τις απαραίτητες μεθόδους για την επεξεργασία RDF μοντέλων. Η λήψη και η αποστολή των αρχείων με FTP γίνεται με την βοήθεια της βιβλιοθήκης ftp4j.



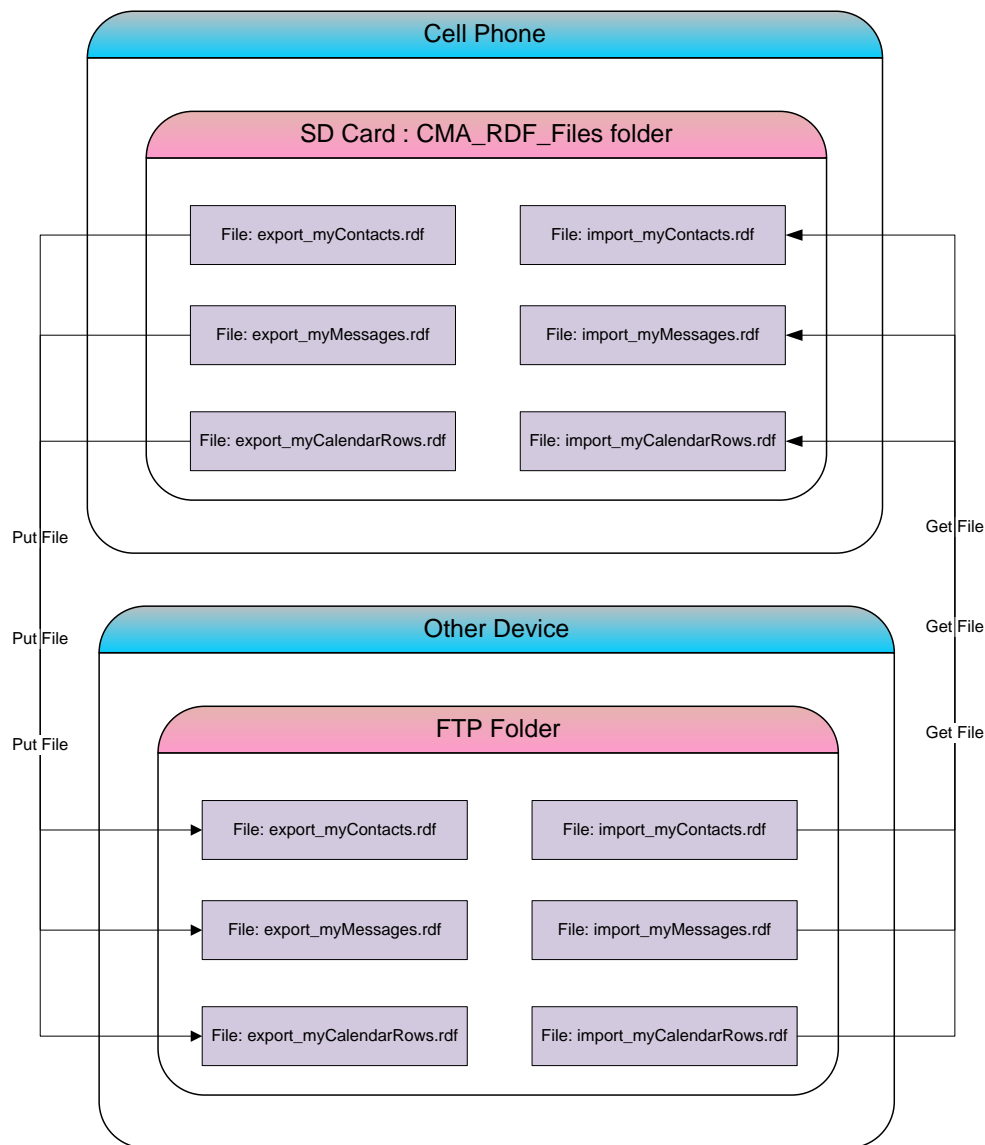
Εικόνα 11, Αλληλεπίδραση εφαρμογής με κινητό τηλέφωνο

Ο συγχρονισμός του περιεχομένου μεταξύ του κινητού τηλεφώνου του χρήστη και μιας άλλης συσκευής γίνεται με την αποστολή και λήψη RDF αρχείων σε FTP φάκελο της άλλης συσκευής. Τόσο η αποστολή όσο και η λήψη των RDF αρχείων γίνεται από την εφαρμογή που έχουμε αναπτύξει. Ο επόμενος πίνακας περιέχει τα αρχεία που χρησιμοποιούνται από την υλοποίησή μας.

Αρχεία	Λειτουργία	Περιγραφή
export_myContacts.rdf	Export Content	Διαβάζονται τα στοιχεία των επαφών και αποθηκεύονται στην κάρτα μνήμης του κινητού. Στην συνέχεια αυτό το αρχείο στέλνεται με FTP σε μία άλλη συσκευή.
export_myMessages.rdf	Export Content	Διαβάζονται τα στοιχεία των μηνυμάτων και αποθηκεύονται στην κάρτα μνήμης του κινητού. Στην συνέχεια αυτό το αρχείο στέλνεται με FTP σε μία άλλη συσκευή.
export_myCalendarRows.rdf	Export Content	Διαβάζονται τα στοιχεία των εγγραφών ημερολογίου και αποθηκεύονται στην κάρτα μνήμης του κινητού. Στην συνέχεια αυτό το αρχείο στέλνεται με FTP σε μία άλλη συσκευή.
import_myContacts.rdf	Import Content	Είναι ένα αρχείο στην κάρτα μνήμης που περιέχει τις επαφές που έχουν ληφθεί με FTP από την άλλη συσκευή. Αυτές οι επαφές συγχρονίζονται με τις υπάρχουσες του κινητού τηλεφώνου.
import_myMessages.rdf	Import Content	Είναι ένα αρχείο στην κάρτα μνήμης που περιέχει τα μηνύματα που έχουν ληφθεί με FTP από την άλλη συσκευή. Δεν συγχρονίζονται με τα υπάρχοντα μηνύματα του κινητού τηλεφώνου.
import_myCalendarRows.rdf	Import Content	Είναι ένα αρχείο στην κάρτα μνήμης που περιέχει τις εγγραφές ημερολογίου που έχουν ληφθεί με FTP από την άλλη συσκευή. Αυτές οι εγγραφές ημερολογίου συγχρονίζονται με τις υπάρχουσες του κινητού τηλεφώνου.

**Πίνακας 3, Αρχεία που χρησιμοποιεί η υλοποίηση**

Στην επόμενη εικόνα παρουσιάζεται αναλυτικά η ανταλλαγή αρχείων μεταξύ της εφαρμογής μας και μιας άλλης συσκευής. Η διαδικασία της ανταλλαγής αρχείων γίνεται μέσω FTP.



Εικόνα 12, Ανταλλαγή αρχείων μεταξύ κινητού τηλεφώνου και άλλης συσκευής

### 3.3. Λειτουργίες

Όλες οι λειτουργίες γίνονται με την χρήση της βιβλιοθήκης που έχουμε αναπτύξει καθώς επίσης και με την χρήση της βιβλιοθήκης ftp4j.

#### 3.3.1. Content

Η εφαρμογή διαβάζει το περιεχόμενο του κινητού που πρόκειται να συγχρονιστεί (επαφές, μηνύματα και εγγραφές ημερολογίου) και το αποθηκεύει στην μνήμη του κινητού. Η παρούσα λειτουργία χρησιμοποιείται και στις επόμενες δύο (Import & Export).

#### 3.3.2. Import

Η εφαρμογή κατεβάζει τα αρχεία RDF από τον FTP Server της άλλης συσκευής και τα συγχρονίζει με το περιεχόμενο του κινητού. Αυτή η λειτουργία χρησιμοποιεί τα ακόλουθα αρχεία που έχουν ληφθεί με FTP:

1. import\_myContacts.rdf
2. import\_myMessages.rdf

3. import\_myCalendarRows.rdf

### 3.3.3. Export

Η εφαρμογή εξάγει το περιεχόμενο της συσκευής σε αρχεία RDF και τα αποστέλλει στον FTP Server της άλλης συσκευής. Αυτή η λειτουργία παράγει τα ακόλουθα αρχεία και τα στέλνει με FTP:

1. export\_myContacts.rdf
2. export\_myMessages.rdf
3. export\_myCalendarRows.rdf

## 3.4. Components

### 3.4.1. Cell Phone Application

Η υλοποίησή μας έχει τις ακόλουθες δυνατότητες:

1. export περιεχομένου από το κινητό σε αρχεία RDF,
2. αποστολή των αρχείων RDF σε FTP Server και
3. χρήση βιβλιοθήκης βασισμένη στο Apache Jena που μπορεί να γίνει port σε άλλες εφαρμογές που τρέχουν Java.

Τα αρχεία γράφονται και διαβάζονται από προκαθορισμένη θέση στην εξωτερική κάρτα μνήμης του κινητού.

Η εφαρμογή τρέχει σε κινητά τηλέφωνα με έκδοση Android 2.2 και πάνω και ονομάζεται Content Management Application (CMA). Υπάρχει ένα μενού με τις ακόλουθες επιλογές:

1. Status: Όταν ξεκινάει η εφαρμογή δημιουργεί ένα Local Copy των περιεχομένων του κινητού που θα συγχρονίζονται (από εδώ και στο εξής θα αναφέρονται ως MyContent). Ο χρήστης έχει την δυνατότητα ανά πάσα χρονική στιγμή να κάνει refresh το MyContent.
2. Import: Γίνονται import τα αρχεία RDF από τον FTP Server και συγχρονίζονται με το MyContent.
3. Export: Γίνεται export το MyContent σε αρχεία RDF και στην συνέχεια τα αρχεία RDF γίνονται upload στον FTP Server.
4. Settings
5. FAQ
6. About

Τα περιεχόμενα του κινητού που συγχρονίζονται θα είναι τα ακόλουθα:

1. Επαφές
  - a. ID (επαφής στο κινητό)
  - b. Ονοματεπώνυμο
  - c. Λίστα με τα τηλέφωνα (αριθμός και τύπος)
  - d. Λίστα με τα email (διεύθυνση και τύπος)
  - e. Όνομα Λογαριασμού (που ανήκει η επαφή)

- f. Τύπος Λογαριασμού (που ανήκει η επαφή)
- 2. Μηνύματα
  - a. Κείμενο
  - b. Διεύθυνση (αριθμός)
  - c. Τύπος (εάν το μήνυμα είναι εισερχόμενο ή εξερχόμενο, δεδομένου ότι δεν υπάρχει τεκμηρίωση από την Google θα χρησιμοποιείται η αριθμητική τιμή που επιστρέφεται από το κινητό)
  - d. Ημερομηνία
- 3. Εγγραφές Ημερολογίου
  - a. Τίτλος
  - b. Ημερομηνία έναρξης
  - c. Ημερομηνία λήξης
  - d. Id Ημερολογίου (που ανήκει η εγγραφή)
  - e. Όνομα Ημερολογίου (που ανήκει η εγγραφή)

Η εφαρμογή έχει υλοποιηθεί με την χρήση του Singleton Pattern, έτσι ώστε να υπάρχει ένα instance κάθε χρονική στιγμή του MyContent. Όλα τα αρχεία που χρησιμοποιούνται από την εφαρμογή (αυτά που γίνονται export και import) αποθηκεύονται στην εξωτερική κάρτα μνήμης του κινητού τηλεφώνου σε ένα φάκελο με όνομα CMA\_RFD\_Files (στο root της κάρτας μνήμης) που δημιουργεί η ίδια η εφαρμογή.

#### 3.4.2. Cell Phone Application Library

Η βιβλιοθήκη που έχουμε αναπτύξει περιέχει:

1. τα λεξικά που χρησιμοποιούνται για την δημιουργία των μοντέλων RDF και
2. τις μεθόδους για τη δημιουργία μοντέλων RDF που χρησιμοποιεί η υλοποίησή μας.

Όποιος επιθυμεί μπορεί να αναπτύξει περαιτέρω την βιβλιοθήκη εμπλουτίζοντας με μεθόδους για την επεξεργασία και την σύγκριση RDF μοντέλων βασιζόμενων στα χρησιμοποιούμενα λεξικά. Επίσης, μπορεί να δημιουργήσει αντίστοιχη βιβλιοθήκη για Java applications με την χρησιμοποίηση του source κώδικα και των βιβλιοθηκών του Apache Jena (και όχι του androidjena που χρησιμοποιούμε εμείς για το Android).

#### 3.4.2. Αρχεία RDF

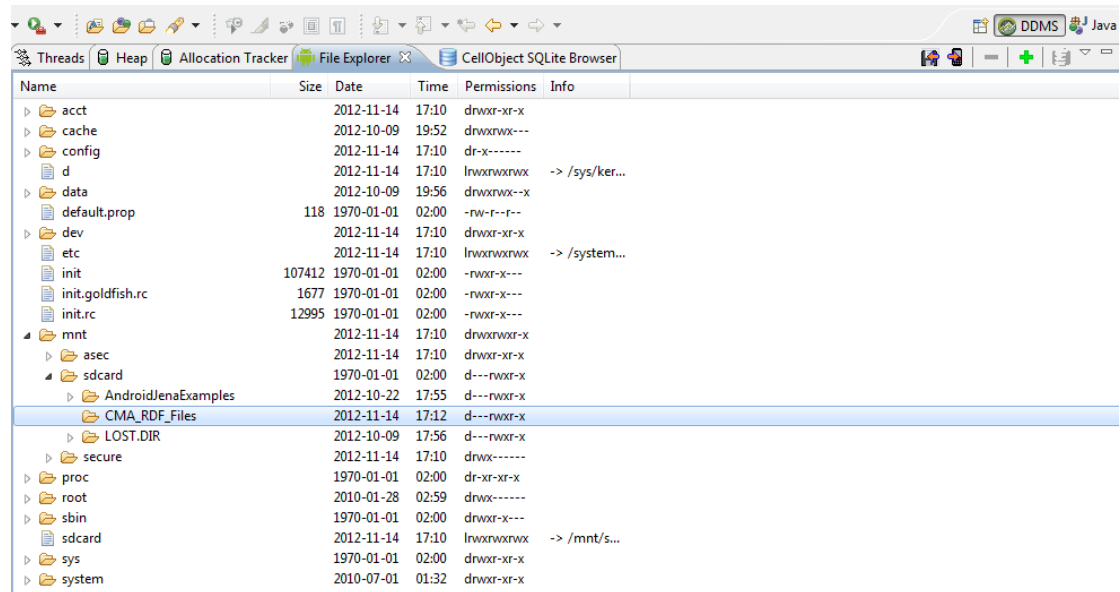
Υπάρχει ένα αρχείο RDF για κάθε τύπο περιεχομένου. Τα αρχεία RDF αποθηκεύονται στην εξωτερική κάρτα μνήμης του κινητού τηλεφώνου. Έχουν κατάληξη RDF για να ξεχωρίζουν από τα υπόλοιπα αρχεία και η δομή τους είναι XML like. Στο παράρτημα Γ υπάρχουν τα αρχεία που δημιουργεί η εφαρμογή για τις επαφές, τα μηνύματα και τις εγγραφές του ημερολογίου. Όλα τα αρχεία RDF γίνονται validate με την online υπηρεσία που παρέχεται από το W3C στο ακόλουθο URL <http://www.w3.org/RDF/Validator/>.

### 3.5. Επίδειξη Λειτουργίας

Στην συνέχεια σας παραθέτουμε ορισμένα print screens από την λειτουργία της εφαρμογής. Στο ακόλουθο print screen παρουσιάζεται η κάρτα μνήμης του emulator μέσα από το perspective DDMS. Όπως μπορείτε να παρατηρήσετε ο φάκελος CMA\_RDF\_Files της

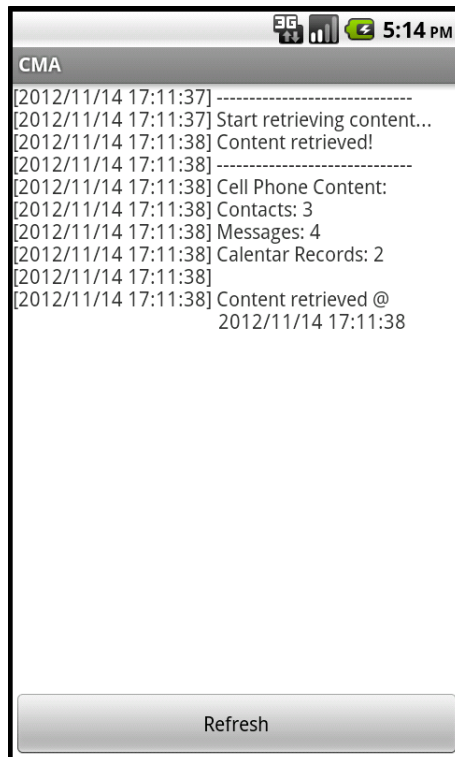


κάρτας μνήμης είναι άδειος. Ο λόγος που έχουμε αδειάσει αυτό τον φάκελο είναι για να δείτε τα περιεχόμενά του μετά την εκτέλεση των λειτουργιών Import και Export.



Εικόνα 13, Κάρτα μνήμης κινητού τηλεφώνου πριν την διαδικασία των import και export

Όταν ξεκινάει η εφαρμογή διαβάζει κατευθείαν το περιεχόμενο από το κινητό τηλέφωνο. Όπως μπορείτε να παρατηρήσετε η βασική οθόνη της εφαρμογής έχει τη μορφή ενός Log αρχείου. Σε αυτή την οθόνη παρουσιάζονται όλες οι ενέργειες που έχει κάνει ο χρήστης.



Εικόνα 14, Επιλογή μενού Status

Για να εξαγάγετε το περιεχόμενο που έχει διαβαστεί από το κινητό τηλέφωνο πρέπει να πάτε στο μενού Export και να πατήσετε το κουμπί Export. Όταν τελειώσει η διαδικασία του

Export θα ενημερωθείτε κατάλληλα στην οθόνη. Με το πέρας της διαδικασίας Export τα αρχεία RDF αποστέλλονται με FTP στην άλλη συσκευή που έχει οριστεί από τις ρυθμίσεις της εφαρμογής. Κατά το τέλος οποιασδήποτε διαδικασίας η εφαρμογή σας ενημερώνει για το αποτέλεσμα της στην βασική οθόνη.



Εικόνα 15, Επιλογή μενού Export και πάτημα κουμπιού Export

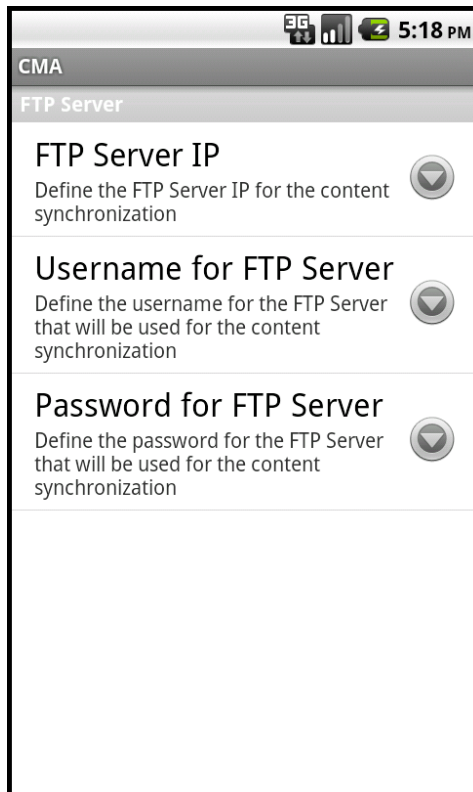
Για να εισάγετε το περιεχόμενο από μία άλλη συσκευή και να το συγχρονίσετε με το υπάρχων της συσκευής σας πρέπει να πάτε στο μενού Import και να πατήσετε το κουμπί Import. Όταν τελειώσει η διαδικασία του Import θα ενημερωθείτε κατάλληλα στην οθόνη. Όπως ήδη σας έχουμε ενημερώσει από την διαδικασία του Import έχει υλοποιηθεί μόνο η λήψη των αρχείων και όχι ο συγχρονισμός του περιεχομένου. Γι' αυτό ακριβώς το λόγο εμφανίζεται το λεκτικό TODO στην οθόνη του κινητού.



Εικόνα 16, Επιλογή μενού Import και πάτημα κουμπιού Import

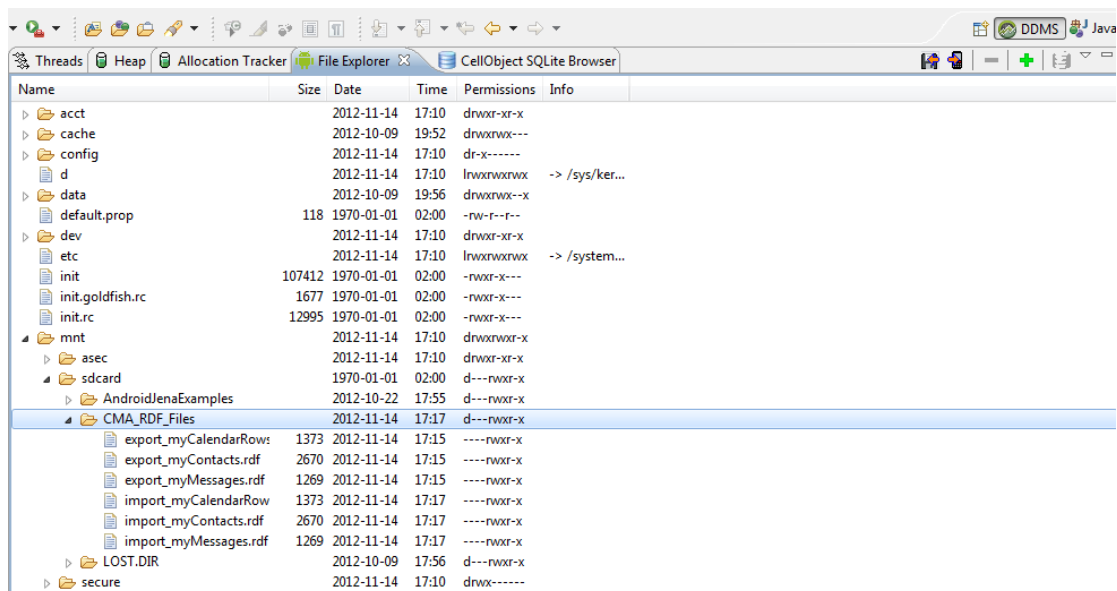
Για να μπορέσουν να αποσταλούν και να ληφθούν αρχεία RDF κατά τις λειτουργίες Export και Import αντίστοιχα πρέπει στο μενού ρυθμίσεις τις εφαρμογές για να έχουν τεθεί τα ακόλουθα στοιχεία για τον FTP Server της άλλης συσκευής:

1. IP
2. username
3. password



Εικόνα 17, Επιλογή μενού Settings

Μετά το πέρας των λειτουργιών Import και Export στον φάκελο CMA\_RDF\_Files της κάρτας μνήμης του κινητού τηλεφώνου υπάρχουν τα αρχεία που έχουν παραχθεί / αποσταλεί και ληφθεί, όπως μπορείτε να δείτε στην εικόνα που ακολουθεί.



Εικόνα 18, Κάρτα μνήμης κινητού τηλεφώνου μετά την διαδικασία των import και export

Στο παράρτημα Δ. βρίσκονται τα logs τα logs από τον FileZilla Server που δείχνουν την αποστολή και την λήψη των RDF αρχείων.

## 4. Συμπεράσματα

Μελετώντας τα μοντέλα που παρουσιάσαμε στο κεφάλαιο της βιβλιογραφικής έρευνας καταλήξαμε στο μοντέλο που χρησιμοποιήσαμε στην υλοποίησή μας και είναι συνδυασμός των μοντέλων PROMPT και SPARQL. Ουσιαστικά δημιουργήσαμε ένα υβριδικό μοντέλο των προηγούμενων δύο εκμεταλλευόμενοι τόσο τα καλύτερα χαρακτηριστικά τους όσο και τις δυνατότητες της τεχνολογίας. Βασικό ρόλο στην υλοποίηση του υβριδικού μας μοντέλου έχουν και οι δυνατότητες της πλατφόρμας του Android. Λόγω της στενής συγγένειας του Android με την γλώσσα προγραμματισμού Java αρκετές από τις βιβλιοθήκες που τρέχουν σε Java έχουν γίνει port σε αυτό. Χαρακτηριστικό παράδειγμα είναι η βιβλιοθήκη androidjena που χρησιμοποιήσαμε.

Χρησιμοποιήσαμε από το PROMPT την επίλυση των προβλημάτων που προκύπτουν από τον συγχρονισμό. Αν και δεν υλοποιήσαμε το συγχρονισμό, ωστόσο η διαδικασία της επίλυσης προβλημάτων που χρησιμοποιεί το PROMPT είναι από τις πιο άρτιες που βρήκαμε στα μοντέλα που μελετήσαμε. Με τις δυνατότητες που προσφέρει η πλατφόρμα του Android για την δημιουργία γραφικών διεπαφών με το χρήστη μπορούμε πολύ εύκολα να υλοποιήσουμε τα βήματα που περιγράφονται στην εικόνα 1. Αυτό που χρειάζεται είναι να γνωρίζουμε τα προβλήματα και να μπορούμε να τα απεικονίσουμε στο χρήστη. Η γνώση για τα προβλήματα προέρχεται από την SPARQL.

Από την στιγμή που έχουμε το περιεχόμενο που θέλουμε να συγχρονίσουμε σε αρχεία RDF είναι πολύ εύκολο να εκτελέσουμε SPARQL ερωτήματα. Με τα SPARQL ερωτήματα μπορούμε να βρούμε το περιεχόμενο που πρέπει να χρησιμοποιηθεί. Η βιβλιοθήκη ARQoid περιέχει την μηχανή ερωτημάτων SPARQL σε Android του Apache Jena project. Στην παρούσα υλοποίηση δεν χρησιμοποιήσαμε την παραπάνω βιβλιοθήκη, ωστόσο κρίνοντας από την εμπειρία μας χρησιμοποιώντας την βιβλιοθήκη androidjena δεν θα είχαμε κάποιο πρόβλημα. Τα ερωτήματα SPARQL είναι ο μηχανισμός ο οποίος θα μας ενημερώσει για το περιεχόμενο που πρέπει να ενημερωθεί. Για να ενημερώσουμε αυτό το περιεχόμενο θα χρησιμοποιήσουμε την επίλυση προβλημάτων του PROMPT που αναφέρουμε παραπάνω.

Το υβριδικό μοντέλο που χρησιμοποιήσαμε θεωρούμε ότι ενδείκνυται για εφαρμογές σε κινητά τηλέφωνα με λειτουργικό σύστημα Android γιατί:

1. περιέχει διαδικασία επίλυσης προβλημάτων
2. χρησιμοποιεί μηχανή ερωτημάτων ειδικά σχεδιασμένη για RDF
3. βασίζεται στις δυνατότητες που έχει το Android και ειδικά στο τρόπο ανάπτυξης εφαρμογών σε αυτό
4. χρησιμοποιεί βιβλιοθήκες που είναι ευρέως διαδεδομένες (στην γλώσσα προγραμματισμού Java) και υποστηρίζονται από την κοινότητα ανάπτυξης λογισμικού (Apache Software Foundation)
5. οι βιβλιοθήκες που έχουμε αναπτύξει μπορούν να χρησιμοποιηθούν και σε εφαρμογές που έχουν γραφεί στην γλώσσα προγραμματισμού Java (εφόσον γίνουν οι κατάλληλες τροποποιήσεις)
6. η διαδικασία FTP που χρησιμοποιούμε για την ανταλλαγή των RDF αρχείων είναι ευρέως διαδεδομένη και μπορεί εύκολα να αντικατασταθεί από web services

Τα επόμενα βήματα που πρέπει να γίνουν μελλοντικά για να ολοκληρωθεί η εφαρμογή και να μπορεί να υποστηριχθεί το μοντέλο μας είναι:

1. η ολοκλήρωση της εφαρμογής
2. η δημιουργία client σε host machine
3. όταν παγιωποιηθούν οι βιβλιοθήκες androidjena και ARQoid να εισαχθούν στην εφαρμογή
4. η αντικατάσταση της FTP διαδικασίας είτε από web services ή από peer to peer ανταλλαγή RDF αρχείων (π.χ. Bluetooth)

## 5. Βιβλιογραφία

Perez de Laborda, C. & Conrad, S., *Relational.OWL - A Data and Schema Representation Format Based on OWL*. Heinrich-Heine-Universität, Düsseldorf.

Noy, N. & Musen, M. (2003). *The PROMPT Suite: Interactive Tools For Ontology Merging And Mapping*. Stanford Medical Informatics, Stanford University.

Stumme, G. & Maedche, A. *FCA-MERGE: Bottom-Up Merging of Ontologies*. Institute for Applied Computer Science and Formal Description Methods, University of Karlsruhe.

Tanca, L. *Context-Based Data Tailoring for Mobile Users*. Dipartimento di Elettronica e Informazione, Politecnico di Milano.

Calvanese, D. & Lenzerini, M. & Nardi, D. (1999). *Unifying Class-Based Representation Formalisms*. Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza".

Weißenberg, N. & Voisard, A. & Gartmann, R. *Using Ontologies in Personalized Mobile Applications*. Fraunhofer ISST, Dortmund.

Ehrig, M. & Staab, S. *QOM - Quick Ontology Mapping*. Institute AIFB, University of Karlsruhe

Schndl, B. & Zander, S. *A Framework for Adaptive RDF Graph Replication for Mobile Semantic Web Applications*. Department of Distributed and Multimedia Systems, University of Vienna.

Bolchini, C. & Curino, C. & Schreiber, A. & Tanca, L. (2006). *Context integration for mobile data tailoring*. Portonovo (AN), Italy.

*ftp4j* [Internet]. Sauron Software. Available from: <http://www.sauronsoftware.it/projects/ftp4j/> [Accessed 8 December 2012]

*RDF Tutorial* [Internet]. w3schools. Available from: <http://www.w3schools.com/rdf/> [Accessed 8 December 2012]

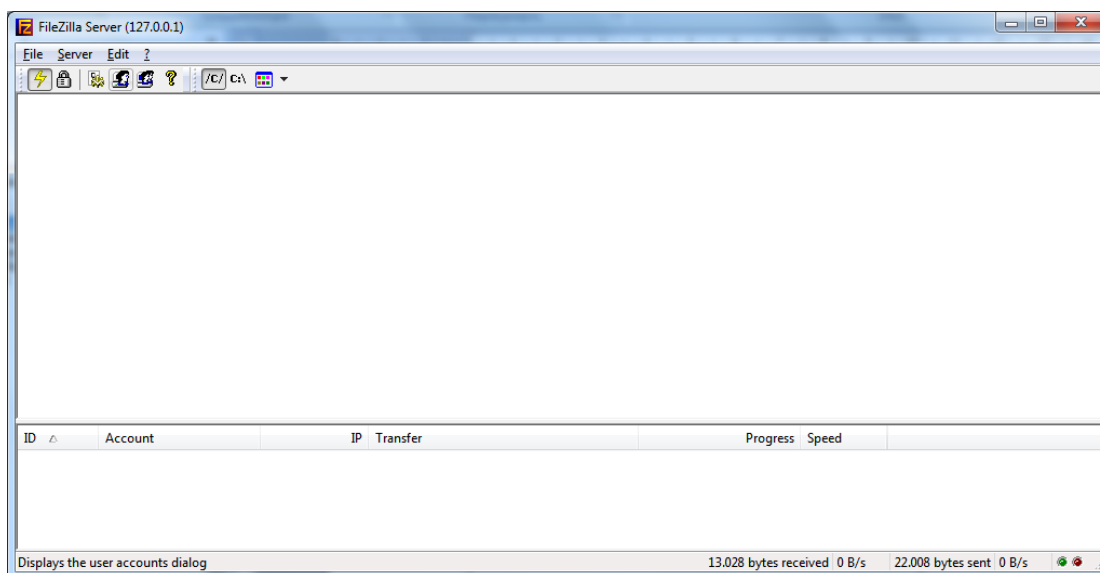
*Apache Jena* [Internet]. Apache Software Foundation. Available from: <http://jena.apache.org/index.html> [Accessed 8 December 2012]

*androidjena* [Internet]. Google. Available from: <http://code.google.com/p/androidjena/> [Accessed 8 December 2012]

## 6. Παράρτημα Α' - FTP Server

Για να δοκιμάσουμε το λειτουργικότητα FTP της εφαρμογής εγκαταστήσαμε τον FileZilla Server στον υπολογιστή μας. Δημιουργήσαμε έναν χρήστη με όνομα test και κωδικό test; που έχει πρόσβαση για ανάγνωση και εγγραφή σε έναν φάκελο του υπολογιστή μας.

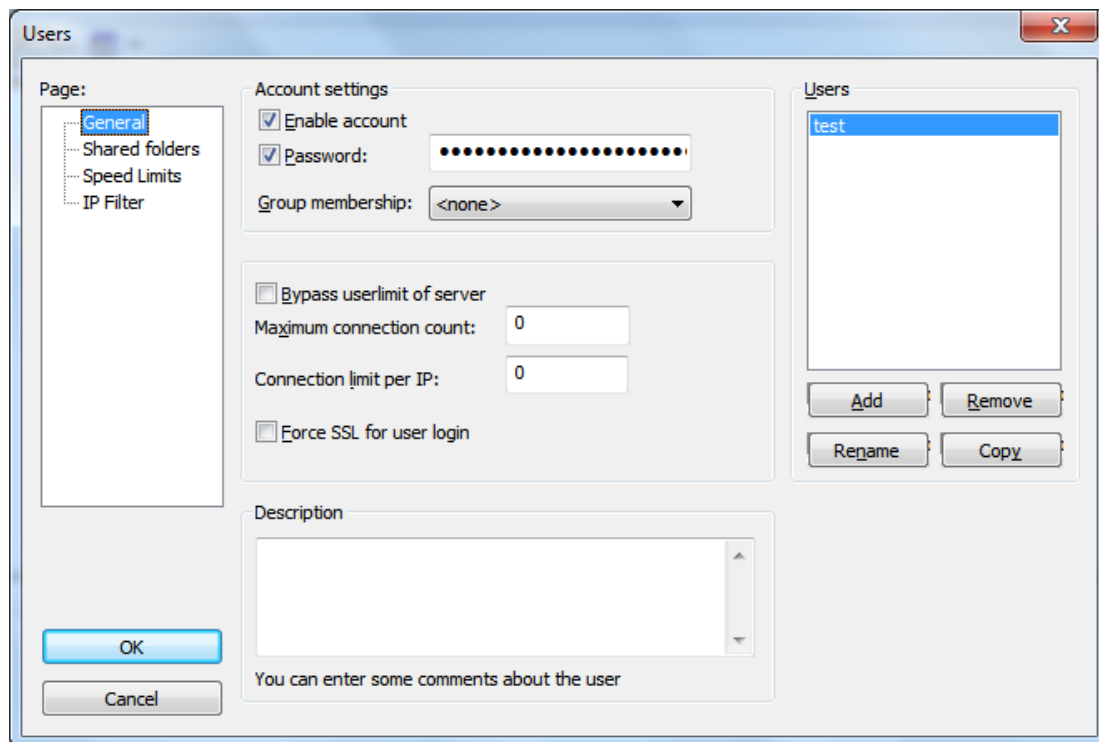
Για την δημιουργία χρήστη επιλέγουμε το τέταρτο εικονίδιο από αριστερά.



Εικόνα 19, Αρχική οθόνη

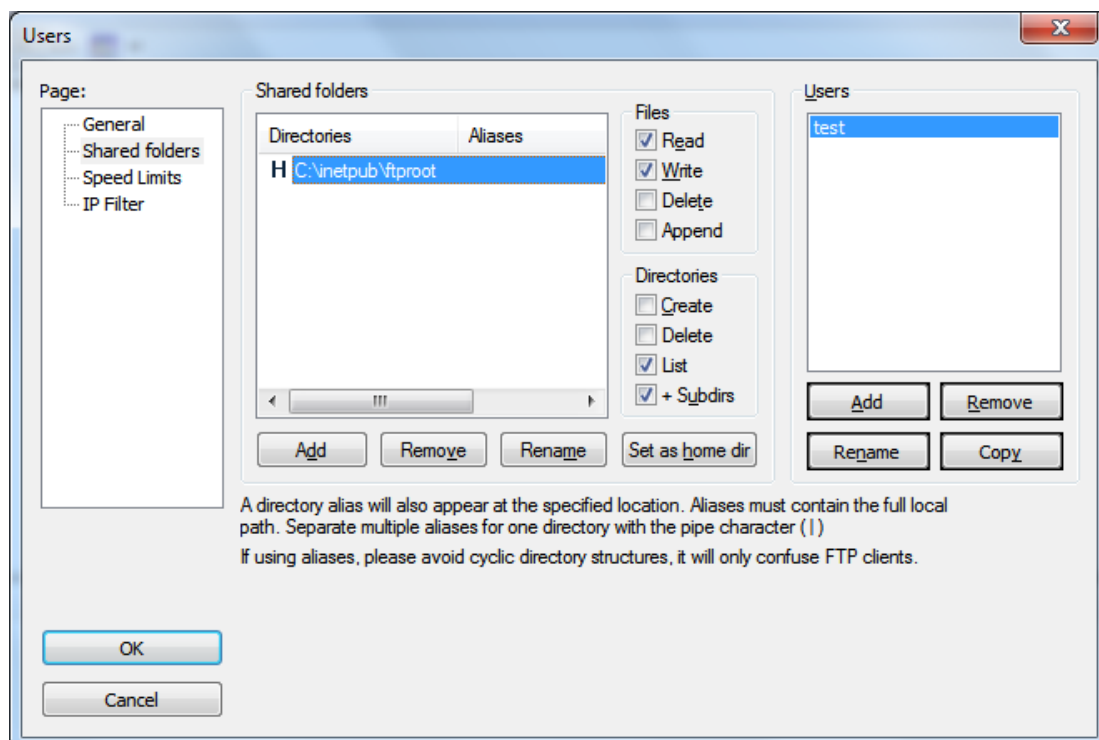
Στην συνέχεια δημιουργούμε τον χρήστη που επιθυμούμε από την Page General. Το όνομα και ο κωδικός του χρήστη μπορούν να αλλάξουν από το μενού Settings της εφαρμογής στο κινητό τηλέφωνο. Συνεπώς, μπορείτε να δημιουργήσετε / χρησιμοποιήσετε οποιονδήποτε χρήστη επιθυμείτε.





Εικόνα 20, Εισαγωγή / τροποποίηση χρήστη

Τέλος, από την Page Shared Folders ορίζουμε τους καταλόγους που θέλουμε να έχει πρόσβαση ο χρήστης. Για την δικιά μας εφαρμογή απαιτείται ένας κατάλογος στον οποίο μπορεί να γίνεται ανάγνωση και εγγραφή στα αρχεία του.

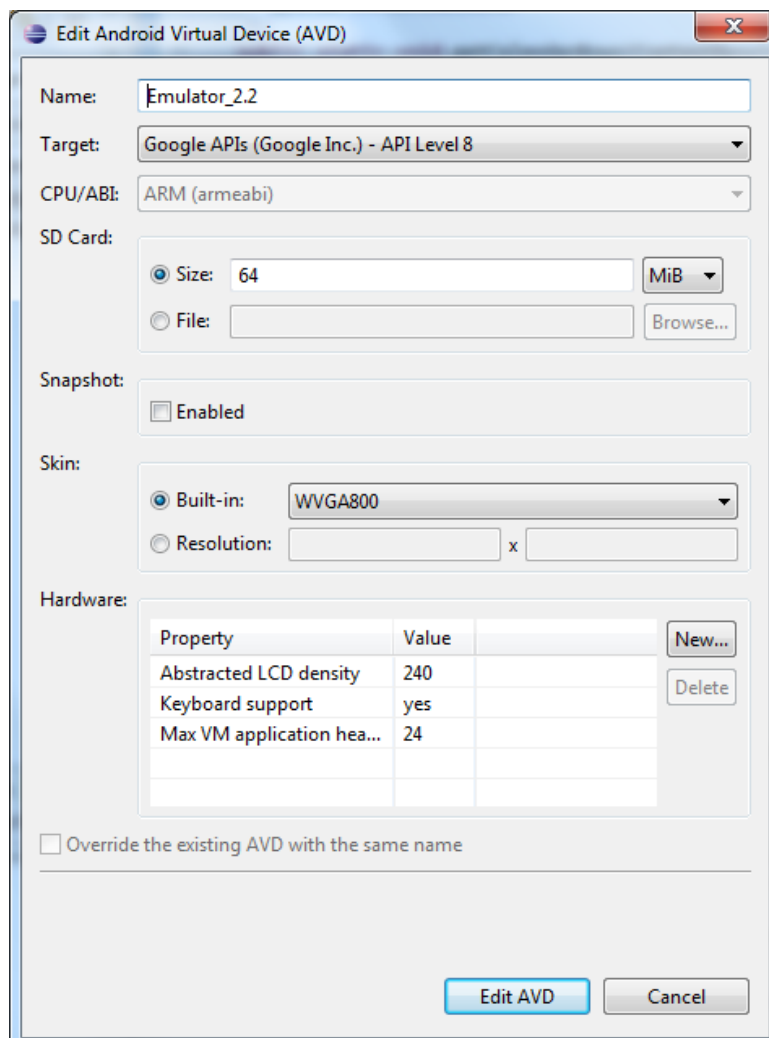


Εικόνα 21, Εισαγωγή / τροποποίηση φακέλων

## 7. Παράρτημα Β' - Android Emulator

### 7.1. Δημιουργία Emulator

Ο Android Emulator θα πρέπει να έχει τα χαρακτηριστικά που περιγράφονται στην ακόλουθη εικόνα.



Εικόνα 22, Android emulator

### 7.2. Εισαγωγή Τεστ Δεδομένων

#### 7.2.1. Google Account

Δημιουργήθηκε λογαριασμός στην Google με τα ακόλουθα χαρακτηριστικά:

- Όνομα: TestFirstName01
- Επώνυμο: TestLastName01
- Email: testfirstname01@gmail.com
- Password: testaccount01
- Ερώτηση ασφαλείας: FavoriteFood με απάντηση Patsa

Ο λογαριασμός αυτός θα χρησιμοποιηθεί κυρίως για το ημερολόγιο.

### 7.2.2. Επαφές

Η εισαγωγή επαφών στον Android Emulator γίνεται με τον ίδιο τρόπο όπως και σε ένα κινητό τηλέφωνο με OS Android. Μπορείτε να χρησιμοποιήσετε την εφαρμογή Contacts του Android.

### 7.2.3. Μηνύματα

Μπορείτε να προσομοιώσετε την αποστολή δεδομένων στον Android Emulator χρησιμοποιώντας την εφαρμογή Messaging, όπως ακριβώς και σε ένα κινητό τηλέφωνο με OS Android. Για την λήψη δεδομένων μία από τις επιλογές που έχετε είναι χρησιμοποιώντας το Eclipse. Θα πρέπει να έχετε εκκινήσει τον Android Emulator μέσα από το Eclipse και από το Perspective DDMS (Menu Window > Open Perspective > Other > DDMS) να χρησιμοποιήσετε το View Emulator Control (Menu Window > Show View > Other > Android > Emulator Control). Στο View Emulator Control συμπληρώνετε στο πεδίο Incoming Number τον αριθμό που επιθυμείτε, επιλέγετε SMS, γράφετε το κείμενο που επιθυμείτε και πατάτε το κουμπί Send. Στον Android Emulator θα δείτε να λαμβάνετε το μήνυμα που έχετε πληκτρολογήσει από τον αριθμό που έχετε θέσει.

### 7.2.3. Ημερολόγιο

Θα πρέπει να ακολουθηθούν οι οδηγίες που περιγράφονται στα ακόλουθα URLs, αφού έχει τεθεί στον emulator λογαριασμός στην Google:

- <http://blog.stefan-macke.com/2011/12/04/how-to-add-the-calendar-application-to-the-android-emulator-running-android-2-2-froyo/> , για να εκτελέσετε την εντολή adb πρέπει να περιηγηθείτε στο φάκελο που βρίσκεται η εφαρμογή adb.exe (βρίσκεται είτε στο φάκελο "\android-sdk\platform-tools" είτε στο φάκελο "\android-sdk\tools")
- <http://blog.stefan-macke.com/2011/12/10/how-to-add-a-local-calendar-to-the-android-2-2-froyo-emulator/> , θα πρέπει να περιμένετε λίγη ώρα μέχρι να εμφανιστεί το ημερολόγιο

## 8. Παράρτημα Γ' - RDF Αρχεία

### 8.1. Επαφές

```
<rdf:RDF
```

```
  xmlns:Email="http://myContentImplementation/android/email#"
```

```
  xmlns:Contact="http://myContentImplementation/android/contact#"
```

```
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
```

```
  xmlns:Phone="http://myContentImplementation/android/phone#" >
```

```
<rdf:Description rdf:nodeID="A0">
```

```
  <Phone:PhoneType>TYPE_MOBILE</Phone:PhoneType>
```

```
  <Phone:PhoneNumber>22</Phone:PhoneNumber>
```

```
</rdf:Description>
```

```
<rdf:Description rdf:nodeID="A1">
```

```
  <Phone:PhoneType>TYPE_MOBILE</Phone:PhoneType>
```

```
  <Phone:PhoneNumber>12</Phone:PhoneNumber>
```

```
</rdf:Description>
```

```
<rdf:Description rdf:nodeID="A2">
```

```
  <Email:EmailType>TYPE_WORK</Email:EmailType>
```

```
  <Email:EmailData>12@</Email:EmailData>
```

```
</rdf:Description>
```

```
<rdf:Description rdf:nodeID="A3">
```

```
  <Phone:PhoneType>TYPE_HOME</Phone:PhoneType>
```

```
  <Phone:PhoneNumber>31</Phone:PhoneNumber>
```

```
</rdf:Description>
```

```
<rdf:Description rdf:about="http://myContentImplementation/android/contact#2">
```

```
  <Contact:Email rdf:nodeID="A4"/>
```

```
  <Contact:Phone rdf:nodeID="A5"/>
```

```
  <Contact:Phone rdf:nodeID="A0"/>
```

```
  <Contact:FullName>First 2 Last 2</Contact:FullName>
```

```
<Contact:ID>2</Contact:ID>
</rdf:Description>
<rdf:Description rdf:about="http://myContentImplementation/android/contact#1">
  <Contact:Email rdf:nodeID="A2"/>
  <Contact:Email rdf:nodeID="A6"/>
  <Contact:Phone rdf:nodeID="A1"/>
  <Contact:Phone rdf:nodeID="A7"/>
  <Contact:FullName>First 1 Last 1</Contact:FullName>
  <Contact:ID>1</Contact:ID>
</rdf:Description>
<rdf:Description rdf:nodeID="A6">
  <Email:EmailType>TYPE_HOME</Email:EmailType>
  <Email:EmailData>11@</Email:EmailData>
</rdf:Description>
<rdf:Description rdf:about="http://myContentImplementation/android/contact#3">
  <Contact:Email rdf:nodeID="A8"/>
  <Contact:Phone rdf:nodeID="A3"/>
  <Contact:AccountType>com.google</Contact:AccountType>
  <Contact:AccountName>testfirstname01@gmail.com</Contact:AccountName>
  <Contact:FullName>First 3 Last 3</Contact:FullName>
  <Contact:ID>3</Contact:ID>
</rdf:Description>
<rdf:Description rdf:nodeID="A8">
  <Email:EmailType>TYPE_HOME</Email:EmailType>
  <Email:EmailData>31@</Email:EmailData>
</rdf:Description>
<rdf:Description rdf:nodeID="A5">
```

```

<Phone:PhoneType>TYPE_HOME</Phone:PhoneType>

<Phone:PhoneNumber>21</Phone:PhoneNumber>

</rdf:Description>

<rdf:Description rdf:nodeID="A4">

  <Email:EmailType>TYPE_HOME</Email:EmailType>

  <Email:EmailData>21@</Email:EmailData>

</rdf:Description>

<rdf:Description rdf:nodeID="A7">

  <Phone:PhoneType>TYPE_HOME</Phone:PhoneType>

  <Phone:PhoneNumber>11</Phone:PhoneNumber>

</rdf:Description>

</rdf:RDF>

```

## 8.2. Μηνύματα

```

<rdf:RDF

  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

  xmlns:Message="http://myContentImplementation/android/message#" >

  <rdf:Description
rdf:about="http://myContentImplementation/android/message#1349814940312_4">

    <Message:Date>1349814940312</Message:Date>

    <Message:Type>1</Message:Type>

    <Message:Address>4</Message:Address>

    <Message:Body>Test 4</Message:Body>

  </rdf:Description>

  <rdf:Description
rdf:about="http://myContentImplementation/android/message#1349801869107_2">

    <Message:Date>1349801869107</Message:Date>

    <Message:Type>1</Message:Type>

    <Message:Address>2</Message:Address>

    <Message:Body>Test 2</Message:Body>

```

```

</rdf:Description>

<rdf:Description
rdf:about="http://myContentImplementation/android/message#1349802070420_12">

  <Message:Date>1349802070420</Message:Date>

  <Message:Type>2</Message:Type>

  <Message:Address>12</Message:Address>

  <Message:Body>Test 3</Message:Body>

</rdf:Description>

<rdf:Description
rdf:about="http://myContentImplementation/android/message#1349801863635_1">

  <Message:Date>1349801863635</Message:Date>

  <Message:Type>1</Message:Type>

  <Message:Address>1</Message:Address>

  <Message:Body>Test 1</Message:Body>

</rdf:Description>

</rdf:RDF>

```

### 8.3. Εγγραφές Ημερολογίου

```

<rdf:RDF

  xmlns:CalendarRecord="http://myContentImplementation/android/calendarRecord#"
  xmlns:MobileCalendar="http://myContentImplementation/android/mobileCalendar#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" >

  <rdf:Description rdf:nodeID="A0">

    <MobileCalendar:Name>testfirstname01@gmail.com</MobileCalendar:Name>

    <MobileCalendar:ID>1</MobileCalendar:ID>

  </rdf:Description>

  <rdf:Description
rdf:about="http://myContentImplementation/android/message#1_2012Oct10,20:16:21_20
12Oct10,21:16:21">

    <CalendarRecord:MobileCalendar rdf:nodeID="A0"/>

```

```
<CalendarRecord:End>2012 Oct 10, 21:16:21</CalendarRecord:End>

<CalendarRecord:Begin>2012 Oct 10, 20:16:21</CalendarRecord:Begin>

<CalendarRecord:Title>Test 01</CalendarRecord:Title>

</rdf:Description>

<rdf:Description
rdf:about="http://myContentImplementation/android/message#1_2012Oct18,03:00:00_20
12Oct19,03:00:00">

  <CalendarRecord:MobileCalendar rdf:nodeID="A1"/>

  <CalendarRecord:End>2012 Oct 19, 03:00:00</CalendarRecord:End>

  <CalendarRecord:Begin>2012 Oct 18, 03:00:00</CalendarRecord:Begin>

  <CalendarRecord:Title>Test 02</CalendarRecord:Title>

</rdf:Description>

<rdf:Description rdf:nodeID="A1">

  <MobileCalendar:Name>testfirstname01@gmail.com</MobileCalendar:Name>

  <MobileCalendar:ID>1</MobileCalendar:ID>

</rdf:Description>

</rdf:RDF>
```



## 9. Παράρτημα Δ' - FileZilla Logs

FileZilla Server version 0.9.41 beta

Copyright 2001-2012 by Tim Kosse (tim.kosse@filezilla-project.org)

Connecting to server...

Connected, waiting for authentication

Logged on

(000001)14/11/2012 17:15:58 μμ - (not logged in) (127.0.0.1)> Connected, sending welcome message...

(000001)14/11/2012 17:15:58 μμ - (not logged in) (127.0.0.1)> 220-FileZilla Server version 0.9.41 beta

(000001)14/11/2012 17:15:58 μμ - (not logged in) (127.0.0.1)> 220-written by Tim Kosse (Tim.Kosse@gmx.de)

(000001)14/11/2012 17:15:58 μμ - (not logged in) (127.0.0.1)> 220 Please visit <http://sourceforge.net/projects/filezilla/>

(000001)14/11/2012 17:15:58 μμ - (not logged in) (127.0.0.1)> USER test

(000001)14/11/2012 17:15:58 μμ - (not logged in) (127.0.0.1)> 331 Password required for test

(000001)14/11/2012 17:15:58 μμ - (not logged in) (127.0.0.1)> PASS \*\*\*\*\*

(000001)14/11/2012 17:15:58 μμ - test (127.0.0.1)> 230 Logged on

(000001)14/11/2012 17:15:58 μμ - test (127.0.0.1)> FEAT

(000001)14/11/2012 17:15:58 μμ - test (127.0.0.1)> 211-Features:

(000001)14/11/2012 17:15:58 μμ - test (127.0.0.1)> MDTM

(000001)14/11/2012 17:15:58 μμ - test (127.0.0.1)> REST STREAM

(000001)14/11/2012 17:15:58 μμ - test (127.0.0.1)> SIZE

(000001)14/11/2012 17:15:58 μμ - test (127.0.0.1)> MLST type\*;size\*;modify\*;

(000001)14/11/2012 17:15:58 μμ - test (127.0.0.1)> MLSD

(000001)14/11/2012 17:15:58 μμ - test (127.0.0.1)> UTF8

(000001)14/11/2012 17:15:58 μμ - test (127.0.0.1)> CLNT

(000001)14/11/2012 17:15:58 μμ - test (127.0.0.1)> MFMT

(000001)14/11/2012 17:15:58 μμ - test (127.0.0.1)> 211 End

(000001)14/11/2012 17:15:59 μμ - test (127.0.0.1)> OPTS UTF8 ON

(000001)14/11/2012 17:15:59 μμ - test (127.0.0.1)> 200 UTF8 mode enabled

(000001)14/11/2012 17:15:59 μμ - test (127.0.0.1)> TYPE I

(000001)14/11/2012 17:15:59 μμ - test (127.0.0.1)> 200 Type set to I

(000001)14/11/2012 17:15:59 μμ - test (127.0.0.1)> PASV

(000001)14/11/2012 17:15:59 μμ - test (127.0.0.1)> 227 Entering Passive Mode  
(127,0,0,1,203,39)

(000001)14/11/2012 17:15:59 μμ - test (127.0.0.1)> REST 0

(000001)14/11/2012 17:15:59 μμ - test (127.0.0.1)> 350 Rest supported. Restarting at 0

(000001)14/11/2012 17:15:59 μμ - test (127.0.0.1)> STOR export\_myContacts.rdf

(000001)14/11/2012 17:15:59 μμ - test (127.0.0.1)> 150 Connection accepted

(000001)14/11/2012 17:15:59 μμ - test (127.0.0.1)> 226 Transfer OK

(000001)14/11/2012 17:15:59 μμ - test (127.0.0.1)> QUIT

(000001)14/11/2012 17:15:59 μμ - test (127.0.0.1)> 221 Goodbye

(000001)14/11/2012 17:15:59 μμ - test (127.0.0.1)> disconnected.

(000002)14/11/2012 17:15:59 μμ - (not logged in) (127.0.0.1)> Connected, sending welcome message...

(000002)14/11/2012 17:15:59 μμ - (not logged in) (127.0.0.1)> 220-FileZilla Server version 0.9.41 beta

(000002)14/11/2012 17:15:59 μμ - (not logged in) (127.0.0.1)> 220-written by Tim Kosse (Tim.Kosse@gmx.de)

(000002)14/11/2012 17:15:59 μμ - (not logged in) (127.0.0.1)> 220 Please visit <http://sourceforge.net/projects/filezilla/>

(000002)14/11/2012 17:15:59 μμ - (not logged in) (127.0.0.1)> USER test

(000002)14/11/2012 17:15:59 μμ - (not logged in) (127.0.0.1)> 331 Password required for test

(000002)14/11/2012 17:15:59 μμ - (not logged in) (127.0.0.1)> PASS \*\*\*\*\*

(000002)14/11/2012 17:15:59 μμ - test (127.0.0.1)> 230 Logged on

(000002)14/11/2012 17:15:59 μμ - test (127.0.0.1)> FEAT

(000002)14/11/2012 17:15:59 μμ - test (127.0.0.1)> 211-Features:

(000002)14/11/2012 17:15:59 μμ - test (127.0.0.1)> MDTM  
(000002)14/11/2012 17:15:59 μμ - test (127.0.0.1)> REST STREAM  
(000002)14/11/2012 17:15:59 μμ - test (127.0.0.1)> SIZE  
(000002)14/11/2012 17:15:59 μμ - test (127.0.0.1)> MLST type\*;size\*;modify\*;  
(000002)14/11/2012 17:15:59 μμ - test (127.0.0.1)> MLSD  
(000002)14/11/2012 17:15:59 μμ - test (127.0.0.1)> UTF8  
(000002)14/11/2012 17:15:59 μμ - test (127.0.0.1)> CLNT  
(000002)14/11/2012 17:15:59 μμ - test (127.0.0.1)> MFMT  
(000002)14/11/2012 17:15:59 μμ - test (127.0.0.1)> 211 End  
(000002)14/11/2012 17:15:59 μμ - test (127.0.0.1)> OPTS UTF8 ON  
(000002)14/11/2012 17:15:59 μμ - test (127.0.0.1)> 200 UTF8 mode enabled  
(000002)14/11/2012 17:15:59 μμ - test (127.0.0.1)> TYPE I  
(000002)14/11/2012 17:15:59 μμ - test (127.0.0.1)> 200 Type set to I  
(000002)14/11/2012 17:15:59 μμ - test (127.0.0.1)> PASV  
(000002)14/11/2012 17:15:59 μμ - test (127.0.0.1)> 227 Entering Passive Mode  
(127,0,0,1,203,45)  
(000002)14/11/2012 17:15:59 μμ - test (127.0.0.1)> REST 0  
(000002)14/11/2012 17:15:59 μμ - test (127.0.0.1)> 350 Rest supported. Restarting at 0  
(000002)14/11/2012 17:15:59 μμ - test (127.0.0.1)> STOR export\_myMessages.rdf  
(000002)14/11/2012 17:15:59 μμ - test (127.0.0.1)> 150 Connection accepted  
(000002)14/11/2012 17:15:59 μμ - test (127.0.0.1)> 226 Transfer OK  
(000002)14/11/2012 17:15:59 μμ - test (127.0.0.1)> QUIT  
(000002)14/11/2012 17:15:59 μμ - test (127.0.0.1)> 221 Goodbye  
(000002)14/11/2012 17:15:59 μμ - test (127.0.0.1)> disconnected.  
(000003)14/11/2012 17:15:59 μμ - (not logged in) (127.0.0.1)> Connected, sending welcome  
message...  
(000003)14/11/2012 17:15:59 μμ - (not logged in) (127.0.0.1)> 220-FileZilla Server version  
0.9.41 beta

(000003)14/11/2012 17:15:59 μμ - (not logged in) (127.0.0.1)> 220-written by Tim Kosse  
(Tim.Kosse@gmx.de)

(000003)14/11/2012 17:15:59 μμ - (not logged in) (127.0.0.1)> 220 Please visit  
<http://sourceforge.net/projects/filezilla/>

(000003)14/11/2012 17:15:59 μμ - (not logged in) (127.0.0.1)> USER test

(000003)14/11/2012 17:15:59 μμ - (not logged in) (127.0.0.1)> 331 Password required for  
test

(000003)14/11/2012 17:15:59 μμ - (not logged in) (127.0.0.1)> PASS \*\*\*\*\*

(000003)14/11/2012 17:15:59 μμ - test (127.0.0.1)> 230 Logged on

(000003)14/11/2012 17:15:59 μμ - test (127.0.0.1)> FEAT

(000003)14/11/2012 17:15:59 μμ - test (127.0.0.1)> 211-Features:

(000003)14/11/2012 17:15:59 μμ - test (127.0.0.1)> MDTM

(000003)14/11/2012 17:15:59 μμ - test (127.0.0.1)> REST STREAM

(000003)14/11/2012 17:15:59 μμ - test (127.0.0.1)> SIZE

(000003)14/11/2012 17:15:59 μμ - test (127.0.0.1)> MLST type\*;size\*;modify\*;

(000003)14/11/2012 17:15:59 μμ - test (127.0.0.1)> MLSL

(000003)14/11/2012 17:15:59 μμ - test (127.0.0.1)> UTF8

(000003)14/11/2012 17:15:59 μμ - test (127.0.0.1)> CLNT

(000003)14/11/2012 17:15:59 μμ - test (127.0.0.1)> MFMT

(000003)14/11/2012 17:15:59 μμ - test (127.0.0.1)> 211 End

(000003)14/11/2012 17:15:59 μμ - test (127.0.0.1)> OPTS UTF8 ON

(000003)14/11/2012 17:15:59 μμ - test (127.0.0.1)> 200 UTF8 mode enabled

(000003)14/11/2012 17:15:59 μμ - test (127.0.0.1)> TYPE I

(000003)14/11/2012 17:15:59 μμ - test (127.0.0.1)> 200 Type set to I

(000003)14/11/2012 17:15:59 μμ - test (127.0.0.1)> PASV

(000003)14/11/2012 17:15:59 μμ - test (127.0.0.1)> 227 Entering Passive Mode  
(127,0,0,1,203,51)

(000003)14/11/2012 17:15:59 μμ - test (127.0.0.1)> REST 0

(000003)14/11/2012 17:15:59 μμ - test (127.0.0.1)> 350 Rest supported. Restarting at 0

(000003)14/11/2012 17:15:59 μμ - test (127.0.0.1)> STOR export\_myCalendarRows.rdf  
(000003)14/11/2012 17:15:59 μμ - test (127.0.0.1)> 150 Connection accepted  
(000003)14/11/2012 17:16:00 μμ - test (127.0.0.1)> 226 Transfer OK  
(000003)14/11/2012 17:16:00 μμ - test (127.0.0.1)> QUIT  
(000003)14/11/2012 17:16:00 μμ - test (127.0.0.1)> 221 Goodbye  
(000003)14/11/2012 17:16:00 μμ - test (127.0.0.1)> disconnected.  
(000004)14/11/2012 17:17:19 μμ - (not logged in) (127.0.0.1)> Connected, sending welcome message...  
(000004)14/11/2012 17:17:19 μμ - (not logged in) (127.0.0.1)> 220-FileZilla Server version 0.9.41 beta  
(000004)14/11/2012 17:17:19 μμ - (not logged in) (127.0.0.1)> 220-written by Tim Kosse (Tim.Kosse@gmx.de)  
(000004)14/11/2012 17:17:19 μμ - (not logged in) (127.0.0.1)> 220 Please visit <http://sourceforge.net/projects/filezilla/>  
(000004)14/11/2012 17:17:19 μμ - (not logged in) (127.0.0.1)> USER test  
(000004)14/11/2012 17:17:19 μμ - (not logged in) (127.0.0.1)> 331 Password required for test  
(000004)14/11/2012 17:17:19 μμ - (not logged in) (127.0.0.1)> PASS \*\*\*\*\*  
(000004)14/11/2012 17:17:19 μμ - test (127.0.0.1)> 230 Logged on  
(000004)14/11/2012 17:17:19 μμ - test (127.0.0.1)> FEAT  
(000004)14/11/2012 17:17:19 μμ - test (127.0.0.1)> 211-Features:  
(000004)14/11/2012 17:17:19 μμ - test (127.0.0.1)> MDTM  
(000004)14/11/2012 17:17:19 μμ - test (127.0.0.1)> REST STREAM  
(000004)14/11/2012 17:17:19 μμ - test (127.0.0.1)> SIZE  
(000004)14/11/2012 17:17:19 μμ - test (127.0.0.1)> MLST type\*;size\*;modify\*;  
(000004)14/11/2012 17:17:19 μμ - test (127.0.0.1)> MLSD  
(000004)14/11/2012 17:17:19 μμ - test (127.0.0.1)> UTF8  
(000004)14/11/2012 17:17:19 μμ - test (127.0.0.1)> CLNT  
(000004)14/11/2012 17:17:19 μμ - test (127.0.0.1)> MFMT

(000004)14/11/2012 17:17:19 μμ - test (127.0.0.1)> 211 End

(000004)14/11/2012 17:17:19 μμ - test (127.0.0.1)> OPTS UTF8 ON

(000004)14/11/2012 17:17:19 μμ - test (127.0.0.1)> 200 UTF8 mode enabled

(000004)14/11/2012 17:17:19 μμ - test (127.0.0.1)> TYPE I

(000004)14/11/2012 17:17:19 μμ - test (127.0.0.1)> 200 Type set to I

(000004)14/11/2012 17:17:19 μμ - test (127.0.0.1)> PASV

(000004)14/11/2012 17:17:19 μμ - test (127.0.0.1)> 227 Entering Passive Mode  
(127,0,0,1,203,216)

(000004)14/11/2012 17:17:19 μμ - test (127.0.0.1)> REST 0

(000004)14/11/2012 17:17:19 μμ - test (127.0.0.1)> 350 Rest supported. Restarting at 0

(000004)14/11/2012 17:17:19 μμ - test (127.0.0.1)> RETR export\_myContacts.rdf

(000004)14/11/2012 17:17:19 μμ - test (127.0.0.1)> 150 Connection accepted

(000004)14/11/2012 17:17:19 μμ - test (127.0.0.1)> 226 Transfer OK

(000004)14/11/2012 17:17:19 μμ - test (127.0.0.1)> QUIT

(000004)14/11/2012 17:17:19 μμ - test (127.0.0.1)> 221 Goodbye

(000004)14/11/2012 17:17:19 μμ - test (127.0.0.1)> disconnected.

(000005)14/11/2012 17:17:21 μμ - (not logged in) (127.0.0.1)> Connected, sending welcome message...

(000005)14/11/2012 17:17:21 μμ - (not logged in) (127.0.0.1)> 220-FileZilla Server version 0.9.41 beta

(000005)14/11/2012 17:17:21 μμ - (not logged in) (127.0.0.1)> 220-written by Tim Kosse (Tim.Kosse@gmx.de)

(000005)14/11/2012 17:17:21 μμ - (not logged in) (127.0.0.1)> 220 Please visit <http://sourceforge.net/projects/filezilla/>

(000005)14/11/2012 17:17:21 μμ - (not logged in) (127.0.0.1)> USER test

(000005)14/11/2012 17:17:21 μμ - (not logged in) (127.0.0.1)> 331 Password required for test

(000005)14/11/2012 17:17:21 μμ - (not logged in) (127.0.0.1)> PASS \*\*\*\*\*

(000005)14/11/2012 17:17:21 μμ - test (127.0.0.1)> 230 Logged on

(000005)14/11/2012 17:17:21 μμ - test (127.0.0.1)> FEAT

(000005)14/11/2012 17:17:21 μμ - test (127.0.0.1)> 211-Features:  
(000005)14/11/2012 17:17:21 μμ - test (127.0.0.1)> MDTM  
(000005)14/11/2012 17:17:21 μμ - test (127.0.0.1)> REST STREAM  
(000005)14/11/2012 17:17:21 μμ - test (127.0.0.1)> SIZE  
(000005)14/11/2012 17:17:21 μμ - test (127.0.0.1)> MLST type\*;size\*;modify\*;  
(000005)14/11/2012 17:17:21 μμ - test (127.0.0.1)> MLSD  
(000005)14/11/2012 17:17:21 μμ - test (127.0.0.1)> UTF8  
(000005)14/11/2012 17:17:21 μμ - test (127.0.0.1)> CLNT  
(000005)14/11/2012 17:17:21 μμ - test (127.0.0.1)> MFMT  
(000005)14/11/2012 17:17:21 μμ - test (127.0.0.1)> 211 End  
(000005)14/11/2012 17:17:22 μμ - test (127.0.0.1)> OPTS UTF8 ON  
(000005)14/11/2012 17:17:22 μμ - test (127.0.0.1)> 200 UTF8 mode enabled  
(000005)14/11/2012 17:17:22 μμ - test (127.0.0.1)> TYPE I  
(000005)14/11/2012 17:17:22 μμ - test (127.0.0.1)> 200 Type set to I  
(000005)14/11/2012 17:17:22 μμ - test (127.0.0.1)> PASV  
(000005)14/11/2012 17:17:22 μμ - test (127.0.0.1)> 227 Entering Passive Mode  
(127,0,0,1,203,225)  
(000005)14/11/2012 17:17:22 μμ - test (127.0.0.1)> REST 0  
(000005)14/11/2012 17:17:22 μμ - test (127.0.0.1)> 350 Rest supported. Restarting at 0  
(000005)14/11/2012 17:17:22 μμ - test (127.0.0.1)> RETR export\_myMessages.rdf  
(000005)14/11/2012 17:17:22 μμ - test (127.0.0.1)> 150 Connection accepted  
(000005)14/11/2012 17:17:22 μμ - test (127.0.0.1)> 226 Transfer OK  
(000005)14/11/2012 17:17:22 μμ - test (127.0.0.1)> QUIT  
(000005)14/11/2012 17:17:22 μμ - test (127.0.0.1)> 221 Goodbye  
(000005)14/11/2012 17:17:22 μμ - test (127.0.0.1)> disconnected.  
(000006)14/11/2012 17:17:22 μμ - (not logged in) (127.0.0.1)> Connected, sending welcome  
message...  
(000006)14/11/2012 17:17:22 μμ - (not logged in) (127.0.0.1)> 220-FileZilla Server version  
0.9.41 beta

(000006)14/11/2012 17:17:22 μμ - (not logged in) (127.0.0.1)> 220-written by Tim Kosse  
(Tim.Kosse@gmx.de)

(000006)14/11/2012 17:17:22 μμ - (not logged in) (127.0.0.1)> 220 Please visit  
<http://sourceforge.net/projects/filezilla/>

(000006)14/11/2012 17:17:22 μμ - (not logged in) (127.0.0.1)> USER test

(000006)14/11/2012 17:17:22 μμ - (not logged in) (127.0.0.1)> 331 Password required for  
test

(000006)14/11/2012 17:17:22 μμ - (not logged in) (127.0.0.1)> PASS \*\*\*\*\*

(000006)14/11/2012 17:17:22 μμ - test (127.0.0.1)> 230 Logged on

(000006)14/11/2012 17:17:22 μμ - test (127.0.0.1)> FEAT

(000006)14/11/2012 17:17:22 μμ - test (127.0.0.1)> 211-Features:

(000006)14/11/2012 17:17:22 μμ - test (127.0.0.1)> MDTM

(000006)14/11/2012 17:17:22 μμ - test (127.0.0.1)> REST STREAM

(000006)14/11/2012 17:17:22 μμ - test (127.0.0.1)> SIZE

(000006)14/11/2012 17:17:22 μμ - test (127.0.0.1)> MLST type\*;size\*;modify\*;

(000006)14/11/2012 17:17:22 μμ - test (127.0.0.1)> MLSL

(000006)14/11/2012 17:17:22 μμ - test (127.0.0.1)> UTF8

(000006)14/11/2012 17:17:22 μμ - test (127.0.0.1)> CLNT

(000006)14/11/2012 17:17:22 μμ - test (127.0.0.1)> MFMT

(000006)14/11/2012 17:17:22 μμ - test (127.0.0.1)> 211 End

(000006)14/11/2012 17:17:22 μμ - test (127.0.0.1)> OPTS UTF8 ON

(000006)14/11/2012 17:17:22 μμ - test (127.0.0.1)> 200 UTF8 mode enabled

(000006)14/11/2012 17:17:22 μμ - test (127.0.0.1)> TYPE I

(000006)14/11/2012 17:17:22 μμ - test (127.0.0.1)> 200 Type set to I

(000006)14/11/2012 17:17:22 μμ - test (127.0.0.1)> PASV

(000006)14/11/2012 17:17:23 μμ - test (127.0.0.1)> 227 Entering Passive Mode  
(127,0,0,1,203,228)

(000006)14/11/2012 17:17:23 μμ - test (127.0.0.1)> REST 0

(000006)14/11/2012 17:17:23 μμ - test (127.0.0.1)> 350 Rest supported. Restarting at 0



(000006)14/11/2012 17:17:23 μμ - test (127.0.0.1)> RETR export\_myCalendarRows.rdf

(000006)14/11/2012 17:17:23 μμ - test (127.0.0.1)> 150 Connection accepted

(000006)14/11/2012 17:17:23 μμ - test (127.0.0.1)> 226 Transfer OK

(000006)14/11/2012 17:17:23 μμ - test (127.0.0.1)> QUIT

(000006)14/11/2012 17:17:23 μμ - test (127.0.0.1)> 221 Goodbye

(000006)14/11/2012 17:17:23 μμ - test (127.0.0.1)> disconnected.