

**ΣΧΕΔΙΑΣΜΟΣ & ΥΛΟΠΟΙΗΣΗ ΜΟΝΤΕΛΟΥ
ΦΥΣΙΚΗΣ ΓΛΩΣΣΑΣ ΓΙΑ ΤΗΝ ΔΙΑΧΕΙΡΙΣΗ
TV-ANYTIME ΠΛΗΡΟΦΟΡΙΩΝ ΣΕ
ΠΕΡΙΒΑΛΛΟΝΤΑ ΨΗΦΙΑΚΗΣ ΤΗΛΕΟΡΑΣΗΣ ΚΑΙ
ΚΙΝΗΤΩΝ ΣΥΣΚΕΥΩΝ.**

Καρναναστάση Αναστασία

Μία εργασία που παρουσιάστηκε στο Πολυτεχνείο Κρήτης σε εκπλήρωση
των απαιτήσεων για την απόκτηση Διπλώματος στο Τμήμα Ηλεκτρονικών
Μηχανικών και Μηχανικών Υπολογιστών

Πολυτεχνείο Κρήτης

Χανιά, Ελλάδα 2003

ΠΕΡΙΛΗΨΗ

Το πρότυπο TV-Anytime καθορίζει δομές από κατηγορίες μεταδεδομένων, που περιγράφουν το περιεχόμενο προγραμμάτων ψηφιακής τηλεόρασης, καθώς και δομές για μεταδεδομένα που περιγράφουν τα ενδιαφέροντα ενός χρήστη (προφίλ) για προγράμματα ψηφιακής τηλεόρασης. Σε αυτή την εργασία παρουσιάζεται ένα μοντέλο φυσικής γλώσσας και η υλοποίησή του, για τη διαχείριση αυτών των μεταδεδομένων, ώστε ο χρήστης να μπορεί, χρησιμοποιώντας φυσική γλώσσα, να αναζητεί προγράμματα τηλεόρασης, να προγραμματίζει εγγραφές και διαγραφές προγραμμάτων, να διαχειρίζεται το προφίλ του και να προγραμματίζει τι θέλει να δει κατά τις μετακινήσεις του. Το μοντέλο που παρουσιάζεται είναι πλήρες σε σχέση με το TV-Anytime, εφόσον διαχειρίζεται όλες τις δομές και τις κατηγορίες μεταδεδομένων του TV-Anytime. Το μοντέλο είναι, επίσης, επεκτάσιμο και μπορεί να εκμεταλλευθεί περισσότερες κατηγορίες, που μπορεί να δημιουργηθούν στο μέλλον, αλλά και τυχόν υπάρχουσες οντολογίες των εφαρμογών.

Σε σύγκριση με προσπάθειες που έγιναν στο παρελθόν για φιλική πρόσβαση σε βάσεις δεδομένων, μέσω φυσικής γλώσσας, που είχαν αντιμετωπίσει το απαγορευτικό κόστος των διευκρινιστικών διαλόγων για την πλήρη αποσαφήνιση της ερώτησης, στο περιβάλλον των εφαρμογών του TV-Anytime ο χώρος της πληροφορίας είναι πολύ πιο δομημένος και καθορίζεται από τις δομές του προτύπου TV-Anytime (και του MPEG-7 για το προφίλ του χρήστη) με αποτέλεσμα την μείωση κατά πολύ της αβεβαιότητας και της ανάγκης διευκρινιστικών ερωτήσεων.

Στη γενική περίπτωση, όμως, μια μερική φράση από την ερώτηση του χρήστη μπορεί να πάρει περισσότερες από μία σημασίες στην ιεραρχία του TV-Anytime. Στο σύστημα που αναπτύχθηκε, δεν χρησιμοποιήθηκαν διευκρινιστικοί διάλογοι για την αποσαφήνιση των ερωτήσεων των χρηστών και όλα τα πιθανά αποτελέσματα επιστρέφονται στον χρήστη. Αντί των διευκρινιστικών διαλόγων, δίνεται ένα βάρος στα αποτελέσματα, που εκφράζει την πιθανότητα να ταιριάζει η ερώτηση με την απάντηση (συμβατότητα). Το βάρος επιτρέπει την ταξινόμηση των απαντήσεων, σύμφωνα με τον

βαθμό συμβατότητάς τους, ώστε να αυξάνεται η πιθανότητα να βρει ο χρήστης αυτό που θέλει στις πρώτες απαντήσεις. Το βάρος υπολογίζεται λαμβάνοντας υπόψη το πόσο ταιριάζει η ερώτηση με τις προτιμήσεις του χρήστη, όπως αυτές έχουν καταγραφεί στο TV-Anytime προφίλ του, αλλά και λαμβάνοντας υπόψη υπάρχουσες οντολογίες εφαρμογών. Ο αλγόριθμος που αναπτύχθηκε εξετάζει τις καλύτερες δυνατές συσχετίσεις και ενοποιεί τα αποτελέσματα αναθέτοντας το κανονικοποιημένο βάρος στις τιμές των χαρακτηριστικών κατηγοριών.

Το μοντέλο φυσικής γλώσσας που κατασκευάστηκε, χρησιμοποιήθηκε και για τη δημιουργία μιας διεπαφής (interface) για τη διαχείριση της TV-Anytime πληροφορίας από κινητές συσκευές (mobile phones, PDA), για τα οποία οι διεπαφές με φυσική γλώσσα είναι ιδιαίτερα σημαντικές.

Η παρούσα διπλωματική εργασία έγινε στα πλαίσια του ευρωπαϊκού ερευνητικού προγράμματος **UP-TV**, στο οποίο συμμετέχει το εργαστήριο Διανεμημένων Πληροφοριακών Συστημάτων και Εφαρμογών (MUSIC), του Πολυτεχνείου Κρήτης.

ΕΥΧΑΡΙΣΤΗΡΙΑ

Ιδιαίτερες ευχαριστίες θα ήθελα να δώσω στον καθηγητή κ. Σταύρο Χριστοδουλάκη για την επίβλεψη και την καθοδήγησή του στην ολοκλήρωση της διπλωματικής εργασίας και για τις πολύτιμες εμπειρίες που μου προσέφερε στα πλαίσια της εργασίας μου στο Εργαστήριο Διανεμημένων Πληροφοριακών Συστημάτων και Εφαρμογών του Πολυτεχνείου Κρήτης.

Θα ήθελα, επίσης, να ευχαριστήσω τους καθηγητές κκ. Κουμπάράκη Μανώλη και Πετράκη Ευρυπίδη για τον χρόνο που διέθεσαν για την ανάγνωση του κειμένου και τις εποικοδομητικές παρατηρήσεις τους.

Ιδιαίτερα, θα ήθελα να ευχαριστήσω τους κκ. Καζάση Φώτη και Παππά Νίκο για την επίβλεψη, την συμπαράσταση και την πολύτιμη βοήθειά τους στην διπλωματική αυτή εργασία και τον κ. Μουμουτζή Νεκτάριο για την συμβολή του σε όλα τα στάδια της ανάπτυξης της.

Τέλος, θα ήθελα να ευχαριστήσω ολόκληρη την ομάδα των τελειοφοίτων και συμφοιτητών μου, που εργάστηκαν πάνω στην εξέλιξη και ανάπτυξη του προγράμματος *UP-TV* και ιδιαίτερα τον Κοτόπουλο Γιώργο και την Φραντζή Μαρία για την συνεργασία, την υποστήριξη και κυρίως, την υπομονή τους.

Καρναναστάση Αναστασία

Πολυτεχνείο Κρήτης

Σεπτέμβριος, 2003

Αφιέρωση

Στην οικογένεια μου ...

Περιεχόμενα

Κεφάλαιο 1	1
Εισαγωγή	1
1.1 Αναγκαιότητα.....	1
1.2 Το πρόγραμμα UP-TV	2
1.3 Στόχοι της διπλωματικής εργασίας.....	4
1.4 Δομή της εργασίας	5
Κεφάλαιο 2.....	7
Επισκόπηση Σχετικής Έρευνας.....	7
2.1 Φυσική Γλώσσα.....	7
2.1.1 Προκαταρκτικά	7
2.1.2 Συστήματα Διαλόγου.....	8
2.1.3 Δομές Ιδιοτήτων και Ενοποίηση	10
2.1.4 Γραμματικοί φορμαλισμοί και συντακτική ανάλυση.....	12
2.1.4.1 Γραμματική χωρίς συμφραζόμενα	12
2.1.4.2 Φορμαλισμός PATR-II.....	13
2.1.4.3 Ανάλυση Διαγραμμάτων (Chart Parsing).....	16
2.1.4.4 Λεξικό και Γραμματική	17
2.2 JavaChart	18
2.3 Συστήματα Διαλόγου σε περιβάλλον ψηφιακής τηλεόρασης.....	23
2.3.1 Το σύστημα NOKIA – TV Guide	24
2.3.2 Το σύστημα NOKIA – Program Guide Information System	24
Ανακεφαλαίωση.....	26
Κεφάλαιο 3	27
Το πρότυπο TV-Anytime.....	27
3.1 Η γενική οργάνωση ενός συστήματος TV-Anytime.....	27
3.2 Μεταδεδομένα στο TV-Anytime.....	29
3.3 Βασικά στοιχεία των προτιμήσεων του χρήστη	31
UserPreferences DS	31
UserIdentifier DS	32
FilteringAndSearchPreferences DS.....	33
CreationPreferences DS	33
ClassificationPreferences DS.....	35
SourcePreferences DS.....	37
PreferenceCondition DS	38
BrowsingPreferences DS	38
SummaryPreferences DS	39
3.4 Βασικά στοιχεία περιγραφής προγραμμάτων	40
ProgramDescription DS	41
ProgramInformation DS.....	41
BasicDescription DS.....	42
BroadcastEvent DS	43
Ανακεφαλαίωση.....	44
Κεφάλαιο 4	45
Το μοντέλο φυσικής γλώσσας σε περιβάλλον ψηφιακής τηλεόρασης	45
4.1 Μεθοδολογία	45
4.2 Προδιαγραφές Απαιτήσεων	47

4.3 Κύριο σώμα εργασίας (Corpus Work)	49
4.3.1 Εισαγωγικές Φράσεις	50
4.3.2 Φράσεις αναζήτησης.....	50
4.3.3 Φράσεις Στόχου	52
4.3.4 Χρονικές Φράσεις	53
4.3.5 Φράσεις περιλήψης.....	54
4.4 Σχεδιασμός (Design)	55
4.4.1 Το μεταμοντέλο των εκφράσεων της φυσικής γλώσσας	55
4.4.2 Απεικόνιση πληροφορίας	56
4.5 Λειτουργικότητα.....	61
4.5.1 Σύνταξη φράσεων για παροχή λειτουργικότητας	61
4.5.2 Ταξινόμηση πληροφορίας από το σύστημα	66
4.5.3 Αλγόριθμος για την Επίλυση Ασαφειών	70
4.6 Θεμελιώδης σχεδιασμός της αρχιτεκτονικής του μοντέλου Φυσικής Γλώσσας	75
4.7 USE CASES.....	77
Use Case 1: Ανάκτηση εξατομικευμένης πληροφορίας περιεχομένου μεταδεδομένων.....	77
Use Case 2: Διαχείριση εξατομικευμένου περιεχομένου μεταδεδομένων	78
Use Case 3: Δημιουργία εξατομικευμένων περιλήψεων περιεχομένου μεταδεδομένων.....	80
Use Case 4: Δημιουργία προφίλ χρήστη.....	81
Ανακεφαλαίωση.....	82
Κεφάλαιο 5	83
Υλοποίηση μοντέλου φυσικής γλώσσας	83
5.1 Γενική Αρχιτεκτονική.....	83
5.2 Η Υπομονάδα Ανάλυσης Διαγραμμάτων (Chartparser Module).....	85
5.2.2 Λεξιλόγιο.....	85
5.2.3 Γραμματική.....	87
5.3 Η Υπομονάδα Διαχείρισης Διαλόγου (Dialogue Manager Module)	92
5.4 Η Υπομονάδα Επίλυσης Ασαφειών (Ambiguities Resolver Module).....	95
5.4.1 Date/Time Resolver	95
5.4.2 TVA Semantics Resolver	95
5.4.3 User's Profile Resolver.....	96
5.5 Η Υπομονάδα Διαχείρισης Απόκρισης (Response Manager Module).....	97
5.6 System Flow : Δύο παραδείγματα	103
Ανακεφαλαίωση.....	108
Κεφάλαιο 6	109
Σχεδιασμός και Υλοποίηση εφαρμογής	109
6.1 Τεχνολογία Java για κινητές συσκευές	109
6.1 Αρχιτεκτονική και Υλοποίηση της σχεδίασης της εφαρμογής.....	110
6.2 Λειτουργικότητα.....	113
6.3 Γραφικός Σχεδιασμός	113
Ανακεφαλαίωση.....	116
Κεφάλαιο 7	117
Ανακεφαλαίωση Διπλωματικής Εργασίας.....	117
Μελλοντικές επεκτάσεις.....	117
7.1 Ανακεφαλαίωση Διπλωματικής εργασίας	117
7.2 Μελλοντικές Επεκτάσεις.....	121

Πίνακας Σχημάτων

Σχήμα 1: Κατάταξη συστημάτων διαλόγου ανάλογα με το βαθμό πολυπλοκότητας	9
Σχήμα 2: Μία απλή δομή ιδιοτήτων με ατομικές τιμές, που παριστάνεται με έναν πίνακα με τιμές ιδιοτήτων.....	10
Σχήμα 3: Μία δομή ιδιοτήτων που περιέχει υπο-δομές.....	10
Σχήμα 4: Η δομή ιδιοτήτων σαν Κατευθυνόμενος Ακυκλικός Γράφος (ΚΑΓ).....	11
Σχήμα 5: Η δομή ιδιοτήτων σαν εξίσωση. Το '0' αναφέρεται στην κορυφή της δομής.....	11
Σχήμα 6: Ενοποίηση δύο δομών ιδιοτήτων σε μία που περιέχει την ένωση της πληροφορίας.	12
Σχήμα 7: Δείγμα γραμματικής χωρίς συμφραζόμενα για τη φράση 'John likes football'	13
Σχήμα 8: Ένα συντακτικό δέντρο που παράγεται από μία γραμματική χωρίς συμφραζόμενα.....	13
Σχήμα 9: Γραμματική χωρίς συμφραζόμενα με περιορισμούς ενοποίησης PATR-II.	15
Σχήμα 10: Προσομοίωση του συστήματος Talkative-TV μέσω στιγμιότυπων	25
Σχήμα 11: Λογική κατάταξη ενός TV-Anytime συστήματος σε επιμέρους υποσυστήματα	27
Σχήμα 12 : Κατηγορίες και πορεία των μεταδεδομένων μέσα στο TV-Anytime	30
Σχήμα 13: Το διάγραμμα εργασίας για την ανάπτυξη του συστήματος (Degerstedt & Johnsson, 2001)	46
Σχήμα 14: Δομή ιδιοτήτων που παράγεται από τον αναλυτή διαγραμμάτων του συστήματος	58
Σχήμα 15: Η μηχανή καταστάσεων που χρησιμοποιείται για τη διαχείριση των καταστάσεων ενός προγράμματος μέσα στην εξατομικευμένη λίστα.....	69
Σχήμα 16: Θεμελιώδης σχεδιασμός της αρχιτεκτονικής του μοντέλου	75
Σχήμα 17: Παράδειγμα του λεξικού με τις ρίζες των λέξεων.....	87
Σχήμα 18: Παράδειγμα του mini λεξικού με τις καταλήξεις κατηγορίας ουσιαστικών	87
Σχήμα 19: Παράδειγμα από την γραμματική.....	88
Σχήμα 20: Δομές ιδιοτήτων που παράγονται από τον αναλυτή διαγραμμάτων.....	91
Σχήμα 21: Δομή ιδιοτήτων για χρονική φράση με ασάφεια	93
Σχήμα 22: XML έγγραφο που σχηματίζεται από τον διαχειριστή διάλογου με την πληροφορία από την εισαγωγή του χρήστη	105
Σχήμα 23: Επίπεδα λογισμικού κατά την εφαρμογή στην πλατφόρμα J2ME.....	110
Σχήμα 24: Η αρχιτεκτονική client-server	111
Σχήμα 25: Παράδειγμα ακολουθίας μηνυμάτων σε περίπτωση εισαγωγής του χρήστη στο σύστημα με χρήση του προσωπικού του λογαριασμού	112
Σχήμα 26: Ο σχεδιασμός της υλοποίησης της εφαρμογής.....	112
Σχήμα 27: Ο γραφικός σχεδιασμός της οθόνης του κινητού τηλεφώνου.....	115
Σχήμα 28: Ένα παράδειγμα μέσα από στιγμιότυπα της εφαρμογής σε ένα κινητό τηλέφωνο.....	116

Κεφάλαιο 1

Εισαγωγή

1.1 Αναγκαιότητα

Η επίδραση και η παρουσία της πληροφορικής στο περιβάλλον του σπιτιού μας συνεχώς αυξάνει και συνεπακόλουθα υπάρχει μία αναπτυσσόμενη ανάγκη για αξιόπιστη αλληλεπίδραση και ευκολία στη χρήση της. Πολλά από τα σπίτια σήμερα έχουν πρόσβαση στο Διαδίκτυο και διαθέτουν πολλές τεχνολογίες αναψυχής, που την εκμεταλλεύονται. Έτσι, είναι αναμενόμενο ότι πολλές συσκευές και συστήματα πληροφορίας σε ένα 'e-home' περιβάλλον θα συνταιριαστούν στο άμεσο μέλλον.

Σε περιβάλλοντα ψηφιακής τηλεόρασης, υπάρχει η ανάγκη για την ύπαρξη μίας φυσικής γλώσσας που θα επιτρέπει στον χρήστη κάνοντας χρήση οποιασδήποτε συσκευής, είτε αυτή είναι η τηλεόρασή του, είτε ο υπολογιστής του, είτε η κινητή του συσκευή (mobile phone, PDA) να επικοινωνεί με τη ψηφιακή συσκευή καταγραφής PDR (Personal Digital Recorder) που διαθέτει σπίτι του, με σκοπό:

- να καθορίσει τους κανόνες διαχείρισης των δεδομένων ψηφιακής τηλεόρασης (ψηφιακά τηλεοπτικά προγράμματα, μεταδεδομένα τηλεοπτικών προγραμμάτων),
- να ανακτήσει δεδομένα για τα ψηφιακά τηλεοπτικά προγράμματα και τα μεταδεδομένα τηλεοπτικών προγραμμάτων βάσει οποιασδήποτε πληροφορίας για το είδος τους.
- να εκφράσει τις προτιμήσεις του και επιθυμίες για το είδος των τηλεοπτικών προγραμμάτων που θα αποθηκευτούν,
- να διαχειριστεί τις εξατομικευμένες λίστες των τηλεοπτικών προγραμμάτων που έχουν επιλεγεί για αποθήκευση,
- να τροποποιήσει οτιδήποτε από τα παραπάνω.

Τα παραπάνω εξυπηρετούν τον χρήστη αλλά και το ίδιο το σύστημα, δεδομένου ότι μπορεί να χρησιμοποιήσει την πληροφορία αυτή για να κάνει καλύτερη δρομολόγηση των πόρων του. Η ανάπτυξη ενός μοντέλου φυσικής γλώσσας σε περιβάλλον ψηφιακής τηλεόρασης, με δυνατότητα εφαρμογής σε διάφορες συσκευές, επιλύει και το πρόβλημα της αυξημένης πολυπλοκότητας που υπάρχει στην ανάπτυξη ενός γραφικού σχεδιασμού (user interface) που να καλύπτει την παραπάνω λειτουργικότητα.

Το μοντέλο φυσικής γλώσσας, που θα κατασκευαστεί, καθορίζει τα χαρακτηριστικά του χρήστη, που είναι παρόμοια με το προφίλ του. Η διαφοροποίηση είναι ότι ο όρος προφίλ χρήστη περιλαμβάνει τα ενδιαφέροντα του χρήστη σχετικά με το περιεχόμενο των τηλεοπτικών προγραμμάτων, ενώ οι κανόνες της γλώσσας καθορίζουν τη μορφή συμπεριφοράς του χρήστη κατά την αλληλεπίδρασή του στο περιβάλλον της ψηφιακής τηλεόρασης.

1.2 Το πρόγραμμα UP-TV

Το πρόγραμμα UP-TV είναι ένα ολοκληρωμένο πρόγραμμα έρευνας και ανάπτυξης βασικών τεχνολογιών για συστήματα και υπηρεσίες ψηφιακής τηλεόρασης. Οι δυνατότητες που στοχεύει να προσφέρει είναι:

- Εξατομίκευση (Personalization) : Κάθε χρήστης θα μπορεί να ορίζει παραμέτρους με τις προτιμήσεις του ώστε να έχει ευκολότερη πρόσβαση στο υλικό που τον ενδιαφέρει.
- Αλληλεπιδραστικότητα (Interactivity) : Ο χρήστης θα μπορεί να πλοηγείται, να οργανώνει και να εκτελεί ερωτήσεις στο παρεχόμενο υλικό.
- Ανεξαρτησία από χώρο και χρόνο (Ubiquity) : Πρόσβαση ανεξάρτητα από το χώρο, περιλαμβανομένης και της εμβέλειας και το χρόνο μετάδοσης του υλικού.

Βάσει των συστημάτων που αναπτύσσονται στα πλαίσια του προγράμματος είναι η αρχιτεκτονική που προτείνεται από το TV-Anytime. Βασικό δομικό στοιχείο της αρχιτεκτονικής αυτής είναι τα Προσωπικά Συστήματα Καταγραφής Βίντεο (Personal Video Recorders-PVRs, Personal Digital Recorders-PDRs) τα οποία αποτελούν την εξέλιξη των συμβατικών συσκευών VCR. Τα συστήματα PVR εκμεταλλεύονται τη διαρκώς αυξανόμενη χωρητικότητα των τυπικών εμπορικών αποθηκευτικών μέσων, καθώς και τις προηγμένες

τεχνολογίες κωδικοποίησης (MPEG 1.2.4) για να παρέχουν βασικές δυνατότητες εξατομικευμένης πρόσβασης σε τηλεοπτικές μεταδόσεις ανεξάρτητα από το χρόνο, αποθηκεύοντας για μετέπειτα χρήση ολόκληρα ή τμήματα τηλεοπτικών προγραμμάτων που ενδιαφέρουν το χρήστη. Η λειτουργικότητα αυτή επιτυγχάνεται μέσω της χρήσης μεταδεδομένων, που συνοδεύουν τα μεταδιδόμενα προγράμματα. Βάσει αυτών, υπολογίζεται τοπικά η σχετικότητα κάθε προγράμματος ως προς ένα υπάρχον προφίλ προτιμήσεων του χρήστη, επιτρέποντας έτσι τη βέλτιστη χρήση του αποθηκευτικού χώρου, «κρατώντας» μόνο υλικό άμεσου ενδιαφέροντος. Τα μεταδεδομένα μεταδίδονται είτε ταυτόχρονα με τα τηλεοπτικά προγράμματα, είτε πριν από αυτά, ανεξάρτητα από αυτήν και ίσως και μέσω διαφορετικού διαύλου επικοινωνίας (internet). Στην τελευταία περίπτωση, αυτή της ετεροχρονισμένης μετάδοσης, τα μεταδεδομένα μπορούν να χρησιμοποιηθούν από το PVR για τη δημιουργία Ηλεκτρονικών Οδηγών Προγράμματος (Electronic Program Guides), οι οποίοι είναι το ηλεκτρονικό ανάλογο των παραδοσιακών τηλεοπτικών οδηγών. Ο χρήστης μπορεί μέσω της συσκευής του να πλοηγηθεί στο περιεχόμενο τους και να εντοπίσει αντικείμενα ενδιαφέροντος, τα οποία η συσκευή αναλαμβάνει στη συνέχεια να αποθηκεύσει τοπικά.

Το πρόγραμμα UP-TV επιδιώκει να επεκτείνει τη λειτουργικότητα της παραπάνω βασικής αρχιτεκτονικής, εισάγοντας δίκτυα καταναμημένων TV-Anytime εξυπηρετητών, οι οποίοι θα μπορούν να παρέχουν υψηλότερου επιπέδου υπηρεσίες. Το δίκτυο των TV-Anytime εξυπηρετητών παρέχει επιπλέον αποθηκευτικό χώρο πέραν της τοπικής συσκευής και πρόσβαση σε μεγαλύτερο αριθμό τηλεοπτικών καναλιών. Το σύστημα είναι σε θέση να εντοπίζει τις μεταδόσεις ενδιαφέροντος για κάθε χρήστη και να κάνει προτάσεις πάνω σε αυτές, μέσω σύνθετων προφίλ προτιμήσεων και αντίστοιχων πλούσιων σε πληροφορία μεταδεδομένων. Τα επιλεγμένα αντικείμενα αποθηκεύονται στο δίκτυο (ή στην τοπική συσκευή PVR) με βελτιστοποιημένο τρόπο, ώστε να είναι «κοντά» στον τελικό χρήστη, ενώ το σύστημα σχεδιάστηκε ειδικά για να υποστηρίζει χρήστες που αλλάζουν γεωγραφική θέση και που συνεπώς επιθυμούν να έχουν γρήγορη πρόσβαση στο υλικό που τους ενδιαφέρει από τον εκάστοτε τόπο διαμονής τους.

1.3 Στόχοι της διπλωματικής εργασίας

Στα πλαίσια της διπλωματικής αυτής εργασίας σχεδιάστηκε και υλοποιήθηκε ένα μοντέλο φυσικής γλώσσας για περιβάλλον ψηφιακής τηλεόρασης, που ικανοποιεί τις ακόλουθες κατηγορίες κανόνων:

1. Ανάκτηση εξατομικευμένης πληροφορίας περιεχομένου μεταδεδομένων.
2. Διαχείριση εξατομικευμένου περιεχομένου μεταδεδομένων.
3. Διαχείριση εξατομικευμένων λιστών με προγράμματα
4. Δημιουργία εξατομικευμένων περιλήψεων περιεχομένου μεταδεδομένων.
5. Δημιουργία προφίλ χρήστη.

Επίσης, η εργασία αυτή ασχολείται με τον σχεδιασμό και την υλοποίηση μίας εφαρμογής του μοντέλου της φυσικής γλώσσας. Η εφαρμογή του μοντέλου αυτού γίνεται διαμέσου κινητών συσκευών, που μέσα από το HTTP πρωτόκολλο θα επικοινωνούν με το σύστημα και θα αποτελούν το μέσο εισαγωγής των εκφράσεων του χρήστη.

Οι βασικές αρχές που ακολουθήθηκαν για τους δύο στόχους που μόλις αναφέρθηκαν, είναι οι εξής:

- Ως προς το μοντέλο της φυσικής γλώσσας, πρέπει αυτό να

- Υποστηρίζει το σύνολο της δομής των μεταδεδομένων υλικού του TV-Anytime.
- Εισάγει και να εξάγει πληροφορία σε μορφή εγγράφων XML κάνοντας χρήση ενός Java API, που έχει ήδη αναπτυχθεί για να εξάγει την διαθέσιμη λειτουργικότητα του εξυπηρετητή TV-Anytime (υπηρεσίες φιλτραρίσματος, ανάκτησης και δημιουργίας περιλήψεων).
- «Κρύβει» την πολυπλοκότητα της δομής των μεταδεδομένων, κατά το δυνατόν περισσότερο, από τον χρήστη και να του παρουσιάζει μία ευκολονόητη εικόνα
- Είναι αξιόπιστο και ανθεκτικό σε πιθανά λάθη του χρήστη.
- Εξασφαλίζει ανάνηψη πληροφορίας σε περίπτωση κατάρρευσης συστήματος ή κατά λάθος μη αποθήκευση της από τον χρήστη.

- Παρέχει χρήσιμες αναφορές στον χρήστη, ώστε να μπορεί εύκολα να καταλάβει πιθανό λάθος του.

-Ως προς τον TV-Anytime Server, το μοντέλο πρέπει να

- Ενσωματώνει τις οντότητες του μοντέλου μεταδεδομένων που ορίζεται από το TV-Anytime.
- Εξασφαλίζει την ασφάλεια των δεδομένων που αποθηκεύει.
- Είναι όσο το δυνατόν πιο αποδοτικό και γρήγορο στην ανάκτηση και αποθήκευση της πληροφορίας.

-Ως προς την εφαρμογή σε κινητές συσκευές, πρέπει αυτή να

- Είναι εύχρηστη και να ακολουθεί τις βασικές αρχές σχεδίασης και ανάπτυξης User Interface, ώστε να μην κουράζει τον χρήστη αλλά να τον βοηθά όσο το δυνατόν περισσότερο στην αλληλεπίδρασή του με το σύστημα.

1.4 Δομή της εργασίας

Μετά το τρέχον –πρώτο- κεφάλαιο, το οποίο αποτελεί και την εισαγωγή του συγγράμματος, ακολουθούν πέντε ακόμα κεφάλαια. Συνοπτικά, αναφέρονται το αντικείμενο του καθενός:

- **Κεφάλαιο 2 – Επισκόπηση σχετικής έρευνας:** παρουσιάζει το θεωρητικό υπόβαθρο της ανάπτυξης πρωτοτύπων φυσικής γλώσσας, τον αναλυτή διαγραμμάτων (chart parser) που χρησιμοποιήθηκε στα πλαίσια αυτής της εργασίας και σχετικές εργασίες, που κυμαίνονται στους χώρους της ψηφιακής τηλεόρασης, χρησιμοποιώντας μοντέλα φυσικής γλώσσας.
- **Κεφάλαιο 3 – Το πρότυπο TV-Anytime:** Παρουσιάζεται το πρότυπο TV-Anytime και τα περιγραφικά σχήματά του, με τα οποία είναι συμβατή η εξαγόμενη πληροφορία από το μοντέλο της φυσικής γλώσσας και στα οποία έχουν στηριχθεί οι κανόνες της γλώσσας.
- **Κεφάλαιο 4 – Το μοντέλο φυσικής γλώσσας σε περιβάλλον ψηφιακής τηλεόρασης:** παρουσιάζει το θεωρητικό μοντέλο της φυσικής γλώσσας και την σχεδίαση του συστήματος, πάνω στα οποία στηρίχθηκε η περαιτέρω υλοποίησή του, τα Use Cases που ικανοποιεί, καθώς επίσης και η λειτουργικότητα του μοντέλου.

- **Κεφάλαιο 5 – Υλοποίηση μοντέλου φυσικής γλώσσας:** αναλύει λεπτομερώς την υλοποίηση του μοντέλου και της επικοινωνίας του με το υπόλοιπο σύστημα.
- **Κεφάλαιο 6 – Σχεδιασμός και υλοποίηση εφαρμογής:** αναλύει τον σχεδιασμό και την υλοποίηση της εφαρμογής για κινητές συσκευές και τις τεχνολογίες που χρησιμοποιήθηκαν.
- **Κεφάλαιο 7 – Ανακεφαλαίωση, συμπεράσματα, μελλοντικές επεκτάσεις:** συνοψίζει αυτήν την εργασία και προτείνει κάποιες μελλοντικές επεκτάσεις.

Κεφάλαιο 2

Επισκόπηση Σχετικής Έρευνας

Στο κεφάλαιο αυτό εξετάζονται έννοιες και ορισμοί πάνω στην φυσική γλώσσα, που θα βοηθήσουν να γίνει κατανοητή η λογική ανάπτυξης του μοντέλου φυσικής γλώσσας. Έτσι, ο αναγνώστης αποκτά μια οικειότητα με την αιχμή της τεχνολογίας (state of the art) πάνω σε θέματα φυσικής γλώσσας, σε συστήματα διαλόγων και στις δομές που διαχειρίζονται τέτοια συστήματα. Επίσης, γίνεται μία αναλυτική παρουσίαση του συντακτικού αναλυτή (open-source λογισμικό) ο οποίος χρησιμοποιήθηκε στην υλοποίηση του μοντέλου της φυσικής γλώσσας (JavaChart). Τέλος, γίνεται μία αναφορά σε υπάρχοντα συστήματα διαλόγων που έχουν αναπτυχθεί σε περιβάλλοντα ψηφιακής τηλεόρασης.

2.1 Φυσική Γλώσσα

Στις παρακάτω παραγράφους θα δώσουμε το θεωρητικό υπόβαθρο για την ομαλή εισαγωγή μας σε έννοιες και λειτουργίες που αφορούν στη φυσική γλώσσα.

2.1.1 Προκαταρκτικά

Προκειμένου να καταστεί το υπόλοιπο αυτής της διατριβής συνεπές και κατανοητό, θα πρέπει να διευκρινιστούν μερικές βασικές έννοιες. Αυτή η εργασία εξετάζει τις πτυχές της γλωσσικής τεχνολογίας, και όχι τη λεκτική τεχνολογία. Η λεκτική τεχνολογία αναφέρεται στην επεξεργασία γλωσσών στο επίπεδο της φωνολογίας και της φωνητικής και θεμελιώνεται στην Ακουστική, την Ηλεκτρομηχανική και την Πληροφορική.

Η γλωσσική τεχνολογία είναι η διαδικασία της ανάλυσης της εισαγωγής κειμένου από την άποψη της σύνταξης, της μορφολογίας, και της σημασιολογίας. Τα θεμέλια για αυτήν βρίσκονται στην υπολογιστική γλωσσολογία, την ψυχολογία και την πληροφορική. Τα

συστήματα που περιγράφονται εδώ είναι συστήματα διαλόγου που περιορίζονται στην επεξεργασία της γραπτής - ή δακτυλογραφημένος - αλληλεπίδρασης [4].

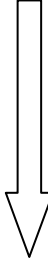
Έκφραση (utterance) είναι ένας γενικός όρος που αναφέρεται σε οποιοδήποτε τρόπο επικοινωνίας. Δηλαδή, για παράδειγμα, μια έκφραση μπορεί να δακτυλογραφηθεί ή να υπαγορευτεί [19].

Διάλογος (dialogue) είναι μια κοινή αλληλεπιδραστική δραστηριότητα μεταξύ δύο οντοτήτων (π.χ. ένας χρήστης και ένα σύστημα υπολογιστή). Αυτό που καθιστά τους διάλογους διαφορετικούς από άλλους τύπους ομιλιών είναι ότι αφορούν στην εγκατάσταση μιας βάσης αμοιβαίας συνεννόησης. Σε αυτήν την εργασία, η αλληλεπίδραση από την μεριά του χρήστη δακτυλογραφείται σε ένα πληκτρολόγιο, και το σύστημα που περιγράφεται επιβάλλει την αυστηρή αλληλεπίδραση, δηλαδή οι διακοπές στη συνομιλία και οι παράλληλες εκφράσεις δεν επιτρέπονται.

Η έννοια της φυσικής γλώσσας (natural language) αφορά στις ανθρώπινες γλώσσες που μπορούν να γραφτούν ή/και να μιληθούν (π.χ. Αγγλικά, Ελληνικά).

2.1.2 Συστήματα Διαλόγου

Ο γενικός ορισμός ενός συστήματος διαλόγου (μερικές φορές αποκαλούμενου συνομιλητικό σύστημα) αφορά σε ένα σύστημα με μια διεπαφή φυσικής γλώσσας, που επιτρέπει στο χρήστη να επικοινωνήσει στη φυσική γλώσσα του/της, και περιλαμβάνει λιγότερο ή περισσότερο σύνθετους διαλόγους [6]. Τα συστήματα διαλόγου, ανάλογα με τον βαθμό της πολυπλοκότητάς τους μπορούν να ταξινομηθούν, όπως φαίνεται στο σχήμα 1:

Τεχνική	Παράδειγμα Εργασίας	Πολυπλοκότητα Εργασίας	Φαινόμενα Διαλόγου που αντιμετωπίζονται
Κείμενο Πεπερασμένων Καταστάσεων (Finite-state Script)	Υπεραστική συνδιάλεξη	μικρή πολυπλοκότητα	Ο χρήστης απαντά σε ερωτήσεις
Βασισμένη σε πλαίσια (Frame-based)	Πληροφορία για την άφιξη και την αναχώρηση ενός τρένου		Ο χρήστης κάνει ερωτήσεις που διευκρινίζει το σύστημα, με απλό τρόπο
Σύνολο από πλαίσια (Sets of Contexts)	Τάξιδικωτικό πρακτορείο		Μεταθέσεις ανάμεσα σε προκαθορισμένα θέματα
Μοντέλα βασισμένα σε σχέδιο (Plan-based)	Σύμβουλος στο σχεδιασμό κουζινών		Θεματικές δομές, δυναμικά παραγόμενες, συνεργατικοί υπο-διάλογοι για διαπραγματεύσεις.
Μοντέλα βασισμένα σε πράκτορες (Agent-based)	Διαχείριση φυσικών καταστροφών	μεγάλη πολυπλοκότητα	Διαφορετικές υποστάσεις (π.χ. πραγματικός κόσμος και ιδεατός κόσμος)

Σχήμα 1: Κατάταξη συστημάτων διαλόγου ανάλογα με το βαθμό πολυπλοκότητας

Τα περισσότερα συστήματα διαλόγου που κατασκευάζονται μέχρι σήμερα ανήκουν στις δύο πρώτες κατηγορίες, όπως και το σύστημα που υλοποιήθηκε στην παρούσα εργασία. Επιπλέον, το χάσμα μεταξύ των θεωρητικών (δηλ. μη-υλοποιημένων) συστημάτων διαλόγου και των υπάρχοντων συστημάτων διαλόγου είναι απέραντο. Υπάρχουν διάφορες δυνατότητες που τα συστήματα των διάφορων πολυπλοκότητων θα έπρεπε να υποστηρίζουν, όπως η παραγωγή απάντησης. Δυστυχώς, είναι πολύ λίγα εφαρμοσμένα συστήματα διαλόγου που μέχρι σήμερα παρέχουν όλες τις παραπάνω δυνατότητες. Μπορεί επομένως να φανεί ότι τα συστήματα διαλόγου μικρής πολυπλοκότητας δεν αξίζουν το όνομα συστήματα διαλόγου, τουλάχιστον όχι υπό την έννοια "διάλογος" κανονικά χρησιμοποιημένης. Επίσης, δεδομένου ότι υπάρχει ένας διαχωρισμός μεταξύ της γλωσσικής τεχνολογίας και της λεκτικής τεχνολογίας, το σύστημα διαλόγου άλλες φορές υποδεικνύει προφορική, άλλες φορές γραπτή, και μερικές φορές πολύμορφη αλληλεπίδραση. Σε αυτή την εργασία συζητάμε μόνο για δακτυλογραφημένη αλληλεπίδραση.

2.1.3 Δομές Ιδιοτήτων και Ενοποίηση

Δύο κεντρικές έννοιες της υπολογιστικής γλωσσολογίας (computational linguistics) είναι οι δομές ιδιοτήτων και η ενοποίηση (feature structures and unification). Γενικά, ανήκουν σε μια προσέγγιση υπεραπλούστευσης, που επιτρέπει την εξήγηση της συμπεριφοράς μιας μεγάλης δομής ως λειτουργία των συνδυασμένων συμπεριφορών των υποδομών της [4]. Μια δομή μπορεί να θεωρηθεί ως αντιπροσώπευση ενός αντικειμένου, το οποίο μπορεί να έχει σύνθετα σύνολα ιδιοτήτων συνημμένα. Τα πρότυπα που χρησιμοποιούν τέτοιες δομές καλούνται φορμαλισμοί βασισμένοι σε περιορισμούς.

Όταν εφαρμόζεται ένας φορμαλισμός βασισμένος σε περιορισμούς, τότε χρησιμοποιείται ένα πίνακας με τιμές ιδιοτήτων, που ονομάζεται δομή ιδιοτήτων. Το σχήμα 2 παρουσιάζει ένα παράδειγμα μιας απλής δομής ιδιοτήτων:

$$\begin{pmatrix} \text{ΙΔΙΟΤΗΤΑ}_1 & \text{ΤΙΜΗ}_1 \\ \text{ΙΔΙΟΤΗΤΑ}_2 & \text{ΤΙΜΗ}_2 \\ \dots & \dots \\ \text{ΙΔΙΟΤΗΤΑ}_n & \text{ΤΙΜΗ}_n \end{pmatrix}$$

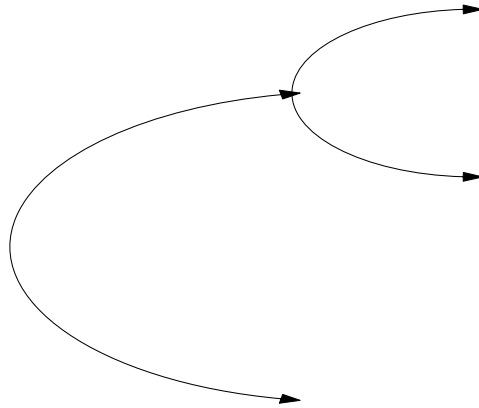
Σχήμα 2: Μία απλή δομή ιδιοτήτων με ατομικές τιμές, που παριστάνεται με έναν πίνακα με τιμές ιδιοτήτων

Τα πεδία των τιμών στο προηγούμενο σχήμα μπορούν να είναι και υποδομές, όπως στο σχήμα 3.

$$\begin{pmatrix} \text{ΙΔΙΟΤΗΤΑ}_1 & \begin{pmatrix} \text{ΙΔΙΟΤΗΤΑ}_{1\alpha} & \text{ΤΙΜΗ}_{1\alpha} \\ \text{ΙΔΙΟΤΗΤΑ}_{2\beta} & \text{ΤΙΜΗ}_{2\beta} \end{pmatrix} \\ \text{ΙΔΙΟΤΗΤΑ}_2 & \text{ΤΙΜΗ}_2 \end{pmatrix}$$

Σχήμα 3: Μία δομή ιδιοτήτων που περιέχει υπο-δομές

Μία δομή ιδιοτήτων μπορεί να παρασταθεί με γράφο, που θα παριστάνει ένα χαρακτηριστικό μονοπάτι μέσα στη δομή. Αυτός ονομάζεται Κατευθυνόμενος Ακυκλικός Γράφος (ΚΑΓ) και φαίνεται στο σχήμα 4.



Σχήμα 4: Η δομή ιδιοτήτων σαν Κατευθυνόμενος Ακυκλικός Γράφος (ΚΑΓ)

Ένας τρίτος τρόπος για την αναπαράσταση μίας δομής ιδιοτήτων είναι με τη μορφή εξίσωσης, όπως φαίνεται και στο σχήμα 5. Χρήση αυτής της αναπαράστασης γίνεται, όταν θέλουμε να συνενώσουμε τη δομή με ανεξάρτητους από συμφραζόμενα γραμματικούς κανόνες.

$$\begin{aligned} \text{Ιδιότητα}_1 \text{ Ιδιότητα}_{1\alpha} &= \text{Τιμή}_{1\alpha} \\ \text{Ιδιότητα}_1 \text{ Ιδιότητα}_{1\beta} &= \text{Τιμή}_{1\beta} \\ \text{Ιδιότητα}_2 &= \text{Τιμή}_2 \end{aligned}$$

Σχήμα 5: Η δομή ιδιοτήτων σαν εξίσωση. Το '0' αναφέρεται στην κορυφή της δομής

Οι δομές ιδιοτήτων μπορούν να χρησιμοποιηθούν για την αναπαράσταση της πληροφορίας για ένα αντικείμενο και τις ιδιότητές του. Στην γλωσσική τεχνολογία, οι δομές ιδιοτήτων χρησιμοποιούνται και στην ανάλυση και στην λογική του πεδίου ορισμού.

Το κύριο πλεονέκτημα της χρήσης δομών ιδιοτήτων είναι ότι πολλές τέτοιες δομές μπορούν να συνενωθούν μέσω της ενοποίησης. Η διαδικασία της ενοποίησης συγκρίνει δύο δομές ιδιοτήτων και επιστρέφει, αν είναι δυνατό, μία ενοποιημένη δομή. Δύο δομές ιδιοτήτων μπορούν να ενοποιηθούν αν οι δομές που περιέχουν είναι συμβατές. Για παράδειγμα

$$[\text{ιδιοτηταB τιμη2}] \cup [\text{ιδιοτηταB τιμη2}] = [\text{ιδιοτηταB τιμη2}]$$

$$[\text{ιδιοτηταB τιμη2}] \cup [\text{ιδιοτηταB []}] = [\text{ιδιοτηταB τιμη2}]$$

Στο σχήμα φαίνεται μία επιτυχημένη ενοποίηση 2 δομών ιδιοτήτων.

$$\left(\begin{matrix} \text{ιδιοτηταA τιμη1} \\ \text{ιδιοτηταB τιμη2} \end{matrix} \right) \cup \left(\begin{matrix} \text{ιδιοτηταB τιμη2} \\ \text{ιδιοτηταΓ τιμη3} \end{matrix} \right) = \left(\begin{matrix} \text{ιδιοτηταA τιμη1} \\ \text{ιδιοτηταB τιμη2} \\ \text{ιδιοτηταΓ τιμη3} \end{matrix} \right)$$

Σχήμα 6: Ενοποίηση δύο δομών ιδιοτήτων σε μία που περιέχει την ένωση της πληροφορίας.

2.1.4 Γραμματικοί φορμαλισμοί και συντακτική ανάλυση

Υπάρχει ένας μεγάλος αριθμός από διαφορετικούς γραμματικούς φορμαλισμούς που μπορούν να χρησιμοποιηθούν σε συστήματα διαλόγου για φυσική γλώσσα. Για να δώσουμε ένα υπόβαθρο ώστε να προσεγγίσουμε την ανάλυση διαγραμμαμάτων θα αναφερθούμε σε δύο γραμματικούς φορμαλισμούς, στη γραμματική χωρίς συμφραζόμενα και στον φορμαλισμό PATR-II.

2.1.4.1 Γραμματική χωρίς συμφραζόμενα

Η γραμματική χωρίς συμφραζόμενα (Context Free Grammar (CFG)) είναι ένα πεπερασμένο σύνολο από κανόνες της μορφής

$$\alpha \rightarrow \beta$$

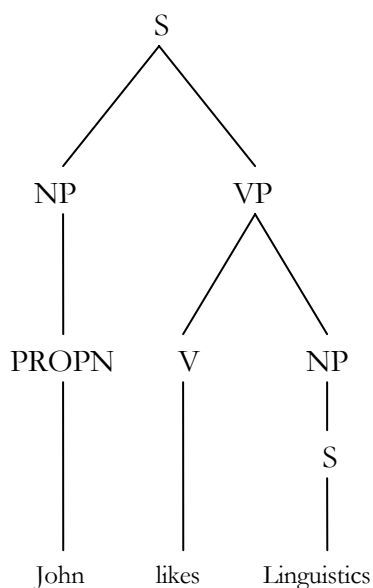
όπου $\alpha \in V_N$ και $\beta \in (V_N \cup V_T)^*$ με V_N ένα πεπερασμένο σύνολο από μη τερματικά (non-terminal) σύμβολα, ή συντακτικές κατηγορίες, V_T ένα πεπερασμένο σύνολο από τερματικά (terminal) σύμβολα ή πρωταρχικά (primitive) σύμβολα. Οι γραμματικές χωρίς συμφραζόμενα είναι σημαντικές γιατί είναι αρκετά δυνατές, για να περιγράψουν την σύνταξη των γλωσσών προγραμματισμού, και αρκετά απλές για να επιτρέψουν την κατασκευή αξιόπιστων αλγορίθμων ανάλυσης, ώστε για μία συμβολοσειρά να καθοριστεί το κατά πόσο και πως μπορεί να παραχθεί από την γραμματική.

Η ιδέα της γραμματικής χωρίς συμφραζόμενα τυποποιήθηκε από τον Τσόμσκι και περιέχει ένα σύνολο από κανόνες για προκαθορισμένες κατηγορίες [4]. Κάθε κανόνας εκφράζει τον τρόπο με τον οποίο οι κατηγορίες ομαδοποιούνται και διατάσσονται. Στο σχήμα 7 βλέπουμε μία απλή γραμματική χωρίς συμφραζόμενα.

$S \rightarrow NP VP$	Abbreviations:
$NP \rightarrow N$	S = sentence
$NP \rightarrow PROP N$	P = phrase
$VP \rightarrow V NP$	N = noun
$N \rightarrow \text{Linguistics}$	V = verb
$PROP N \rightarrow \text{Abraham}$	PROP N = proper noun
$V \rightarrow \text{likes}$	

Σχήμα 7: Δείγμα γραμματικής χωρίς συμφραζόμενα για τη φράση 'John likes Linguistics'

Η πρόταση «John likes Linguistics» αναλύεται σαν μία πρόταση που ανήκει στην κατηγορία S, αφού περιέχει μία φράση ουσιαστικού (κατηγορία NP), με κύριο όνομα (PROP N) «John», και μία φράση ρήματος (κατηγορία VP), που περιέχει ένα ρήμα (κατηγορία V) «likes» και το ουσιαστικό (κατηγορία N) «Linguistics». Αυτό μπορεί να παρασταθεί σε ένα συντακτικό δέντρο, όπως στο σχήμα 8.



Σχήμα 8: Ένα συντακτικό δέντρο που παράγεται από μία γραμματική χωρίς συμφραζόμενα

2.1.4.2 Φορμαλισμός PATR-II

Ο φορμαλισμός PATR-II μπορεί να θεωρηθεί ως μία γλώσσα υπολογιστή για κωδικοποίηση γλωσσολογικής πληροφορίας, χωρίς να προϋποθέτει κάποια συγκεκριμένη θεωρία για την σύνταξη. Αναπτύχθηκε από τον Stuart M. Shieber στο πανεπιστήμιο του Stanford, στις αρχές του 1980 [17]. Μία γραμματική PATR-II περιέχει ένα σύνολο από κανόνες και ένα λεξικό.

Κάθε κανόνας περιέχει έναν κανόνα χωρίς συμφραζόμενα για τη δομή της φράσης και ένα σύνολο από περιορισμούς βασισμένους στην ενοποίηση, που αναφέρονται στις δομές ιδιοτήτων, που σχετίζονται με τις συνιστώσες των κανόνων για τη δομή των φράσεων. Το λεξικό παρέχει τα αντικείμενα που μπορούν να αντικαταστήσουν τα τερματικά σύμβολα των κανόνων για τη δομή των φράσεων. Ουσιαστικά είναι μία γραμματική χωρίς συμφραζόμενα με δομές ιδιοτήτων σαν κατηγορίες. Αυτό σημαίνει ότι οι κατηγορίες δε χρειάζεται να είναι ατομικές, αλλά μπορεί να αποτελούνται από ιδιότητες με συνημμένες τιμές. Η γενική, λοιπόν, σημειογραφία που χρησιμοποιείται για να περιγράψει την γραμματική PATR-II, είναι η εξής:

$$\beta_0 \rightarrow \beta_1 \dots \beta_n$$

{σύνολο από περιορισμούς με μορφή Κατευθυνόμενου Ακυκλικού Γράφου σε μορφή εξίσωσης}

Οι καθορισμένοι περιορισμοί έχουν μία από τις ακόλουθες μορφές

$$< \beta_i \text{ χαρακτηριστικό μονοπάτι} > = \text{ατομική τιμή}$$

$$< \beta_i \text{ χαρακτηριστικό μονοπάτι} > = < \beta_j \text{ χαρακτηριστικό μονοπάτι} >$$

Η σημειογραφία $< \beta_i \text{ χαρακτηριστικό μονοπάτι} >$ δηλώνει ένα χαρακτηριστικό μονοπάτι μέσα από μία δομή ιδιοτήτων, που σχετίζεται με το στοιχείο β_i του μέρους του κανόνα, χωρίς συμφραζόμενα. Στην πρώτη μορφή των περιορισμών έχουμε ότι η τιμή που βρίσκεται στο τέλος του δοσμένου μονοπατιού πρέπει να ενοποιηθεί με την συγκεκριμένη ατομική τιμή. Η δεύτερη μορφή δηλώνει ότι οι τιμές που βρίσκονται στο τέλος των δύο δοσμένων μονοπατιών πρέπει να μπορούν να ενοποιηθούν [4].

Εφόσον λοιπόν, η χρήση των δομών ιδιοτήτων επιτρέπει την ενοποίηση, μας είναι πολύ χρήσιμη όταν θέλουμε να συγχωνεύσουμε μονομερείς δομές πληροφορίας. Αυτό σημαίνει ότι η νέα πληροφορία μπορεί να προστεθεί στη δομή αρκεί να είναι συμβατή. Στο σχήμα 9 βλέπουμε μία απλή γραμματική, με ένα λεξικό, που συνδέει λέξεις με Κατευθυνόμενους Ακυκλικούς Γράφους (Κ.Α.Γ.), δείγμα από κανόνα PATR-II.

$$S \rightarrow NP VP$$

$$\langle VP \text{ agr} \rangle = \langle NP \text{ agr} \rangle$$

$$VP \rightarrow V NP$$

$$\langle VP \text{ agr} \rangle = \langle V \text{ agr} \rangle$$

Uther :

$$\langle cat \rangle = np$$

$$\langle agr \text{ number} \rangle = singular$$

$$\langle agr \ person \rangle = third$$

Arthur :

$$\langle cat \rangle = np$$

$$\langle agr \ number \rangle = sin \ gular$$

$$\langle agr \ person \rangle = third$$

knights :

$$\langle cat \rangle = v$$

$$\langle agr \ number \rangle = sin \ gular$$

$$\langle agr \ person \rangle = third$$

Σχήμα 9: Γραμματική χωρίς συμφραζόμενα με περιορισμούς ενοποίησης PATR-II.

Αυτή η γραμματική, μαζί με το λεξικό, αφορά στις δύο προτάσεις “Uther knights Arthur” και “Arthur knights Uther” με *cat* σαν κατηγορία, το *number* να υποδηλώνει τον ενικό ή πληθυντικό αριθμό και το *person* το πρόσωπο. Αν θεωρήσουμε ότι έχουμε την πρώτη από τις δύο προτάσεις, η δομή της φράσης (bracketed notation) έχει ως εξής:

$$[s[NP \ Uther][VP[v \ knights \]][NP \ Arthur]]$$

Ο κανόνας VP απαιτεί ότι ο Κ.Α.Γ. που αφορά στο VP, πρέπει να είναι ο ίδιος (να μπορεί να ενοποιηθεί) με το V. Έτσι, το VP θα έχει σαν τιμή τον ίδιο κόμβο με το V, και επακόλουθα τις ίδιες τιμές για το *person* και το *number*. Ομοίως, για την ενοποίηση που αφορά στον κανόνα S, το NP θα έχει τις ίδιες τιμές με το VP, και επακόλουθα με το V. Θα πρέπει να επισημάνουμε εδώ, ότι η διαδικασία της ενοποίησης είναι ανεξάρτητη από σειρά (order-independent). Δηλαδή, θα έχουμε το ίδιο αποτέλεσμα, είτε η διαδικασία της ενοποίησης ξεκινήσει από τη ρίζα του αναλυμένου δέντρου, είτε από τα φύλλα του. Και στις δύο περιπτώσεις ο Κ.Α.Γ. που αφορά, για παράδειγμα, στον κόμβο VP θα είναι ο παρακάτω:

[*cat* : *vp*

agr : [*person* : *third*

number : *singular*]]

2.1.4.3 Ανάλυση Διαγραμμάτων (Chart Parsing)

Όπως αναφέραμε νωρίτερα, ανάλυση (parsing) είναι η διαδικασία εκείνη της ανάθεσης μέρους ιδιοτήτων του λόγου σε κάθε λέξη, μέσα σε μία πρόταση [19]. Αυτό σημαίνει την αυτόματη ανάλυση ενός κειμένου, με δεδομένο ένα λεξικό και ένα σύνολο από γραμματικούς κανόνες. Ο στόχος μιας υπο-μονάδας ανάλυσης είναι να λάβει την εισαγωγή του χρήστη, να την ερμηνεύσει και να παράγει μία δομή δεδομένων για την αναπαράστασή της.

Η ανάλυση διαγραμμάτων είναι ένα αλγοριθμικό σχήμα για την ανάλυση των φορμαλισμών σχετικών με τις ανεξάρτητες από τα συμφραζόμενα γραμματικές. Ουσιαστικά, η ανάλυση διαγραμμάτων νοείται σαν μια μέθοδος εξαντλητικής ανάλυσης, που αποτελεί το σύνολο από επιμέρους αναλύσεις που λειτουργούν για όλες τις πιθανές εισαγωγές. Οι επιμέρους αυτές αναλύσεις συγκεντρώνονται σε ένα, αποκαλούμενο, διάγραμμα [16].

Ο βασικός αλγόριθμος που χρησιμοποιείται για την ανάλυση διαγραμμάτων είναι ο αλγόριθμος Early [5]. Ο βασικός πυρήνας του αλγορίθμου αυτού είναι το πέρασμα από αριστερά προς τα δεξιά που γεμίζει έναν πίνακα, δηλαδή το διάγραμμα μας (chart), με $N+1$ εισαγωγές. Για κάθε λέξη στην αναλυμένη πρόταση, το διάγραμμα περιέχει μία λίστα από καταστάσεις που απεικονίζουν τα μερικώς αναλυμένα δέντρα, που παράγονται μέχρι εκείνη τη στιγμή. Στόχος του αλγορίθμου είναι να αποτρέψει την επαναλαμβανόμενη ανάλυση της ίδιας συμβολοσειράς. Οι ανεξάρτητες καταστάσεις που περιέχονται σε κάθε εισαγωγή στο διάγραμμα περιέχουν τρία είδη πληροφορίας: ένα υπο-δέντρο σχετικό με έναν γραμματικό κανόνα, πληροφορία για την πρόοδο που έχει γίνει για την ολοκλήρωση του υπο-δέντρου, και την θέση του υπο-δέντρου σε σχέση με την είσοδο. Γραφικά, θα χρησιμοποιήσουμε μία τελεία (dot) στη δεξιά πλευρά της κατάστασης ενός γραμματικού κανόνα για να δηλώσουμε την πρόοδο στην αναγνώρισή του. Η δομή που προκύπτει ονομάζεται κανόνας τελείας (dotted rule). Η θέση μιας κατάστασης, εξαρτημένη από την είσοδο, απεικονίζεται με δύο αριθμούς, που δηλώνουν το που ξεκινά η κατάσταση και το που βρίσκεται η τελεία. Ας το δούμε όμως, μέσα από ένα παράδειγμα καταστάσεων:

'Book that flight'

$$S \rightarrow \bullet VP, [0,0]$$

$$NP \rightarrow Det \bullet No \min al, [1,2]$$

$$VP \rightarrow V NP^\bullet, [0,3]$$

Η πρώτη κατάσταση, με την τελεία στα αριστερά της συνιστώσας, παριστά μία καθοδική πρόβλεψη για το συγκεκριμένο είδος του S. Το πρώτο 0 δηλώνει ότι η συνιστώσα, που προβλέπεται σε αυτή την κατάσταση, πρέπει να ξεκινά από την αρχή της εισόδου. Η δεύτερη κατάσταση, που δημιουργείται σε επόμενο στάδιο στην επεξεργασία αυτής της πρότασης, δηλώνει ότι ένα NP ξεκινά από την θέση 1, ότι ένα Det έχει αναλυθεί επιτυχώς και ότι αναμένεται ένα Nominal. Η τρίτη κατάσταση, με την τελεία στα δεξιά των δύο συνιστωσών της, παριστά την επιτυχημένη ανακάλυψη ενός δέντρου, που αντιστοιχεί σε ένα VP, που καλύπτει όλη την είσοδο. Αυτές οι καταστάσεις μπορούν να αναπαρασταθούν γραφικά, σε ένα σχήμα όπου οι καταστάσεις είναι κορυφές ή τόξα, ενώ το διάγραμμα (chart) είναι ένας Κατευθυνόμενος Ακυκλικός Γράφος (Κ.Α.Γ.) [4].



Όπως είπαμε, η βασική λειτουργία ενός αναλυτή Earley, είναι να καλύψει $N+1$ σύνολα από καταστάσεις στο διάγραμμα από τα αριστερά προς τα δεξιά, και να επεξεργαστεί με τη σειρά, αυτές τις καταστάσεις, μέσα σε κάθε σύνολο. Ο αλγόριθμος κινείται μέσα στο διάγραμμα και δεν επιστρέφει σε καταστάσεις και σε εισαγωγές του διαγράμματος, από τις οποίες έχει περάσει. Η παρουσία της κατάστασης $S \rightarrow a\bullet, [0, N]$ στη λίστα των καταστάσεων στην τελευταία εισαγωγή στο διάγραμμα δηλώνει την επιτυχημένη ανάλυση.

2.1.4.4 Λεξικό και Γραμματική

Ο αναλυτής χρησιμοποιεί μία λίστα από λέξεις, το λεξικό. Το λεξικό πρέπει θεωρητικά να αποτελείται από όλους τους λεξικολογικούς όρους που εμφανίζονται στην γλώσσα [4], πράγμα αδύνατο, εφόσον κάτι τέτοιο απαιτεί μεγάλο όγκο πληροφορίας και υπολογιστική δύναμη.

Ένα πρόβλημα κατά την κατασκευή των λεξικών στα αγγλικά είναι η έννοια των ανοικτών ονομαστικών ενώσεων (open nominal compounds), όπως "action movie", και τα φραστικά

ρήματα (phrasal verbs), όπως "write down". Θα ήταν επιθυμητό μέσα από το λεξικό να μπορούν να υποστηριχθούν ονομαστικές ενώσεις και κύρια ονόματα (proper nouns), αλλιώς κάτι τέτοιο θα πρέπει να υποστηρίζεται μέσα από γραμματικούς κανόνες.

Το απλούστερο, αλλά ταυτοχρόνως, το μεγαλύτερο πιθανό λεξικό για μία φυσική γλώσσα, θα ήταν ένας κατάλογος όλων των υπαρχουσών λέξεων. Αυτό θα ήταν ένας πολύ μεγάλος κατάλογος λέξεων, ο οποίος δεν θα απεικόνιζε τη μορφολογική δομή της γλώσσας, και τις δυναμικές πτυχές της, όπως οι παραγωγές και οι ενώσεις. Για την αντιμετώπιση, ιδιαίτερα του τελευταίου, ένα υπολογιστικό λεξικό ή μίνι-λεξικό συστήνεται από τους Jurafsky και Martin. Το υπολογιστικό λεξικό αποτελείται από έναν κατάλογο με τις ρίζες των λέξεων, και καθολικούς μορφολογικούς κανόνες για την συγκεκριμένη φυσική γλώσσα. Το κύριο πλεονέκτημα αυτού είναι το γεγονός ότι όλες οι λέξεις που ανήκουν σε έναν ορισμένο τύπο λέξης που συμπεριφέρονται με τον ίδιο τρόπο μπορούν να χρησιμοποιηθούν κατά την οικοδόμηση του λεξικού. Παραδείγματος χάριν, όλα τα ουσιαστικά που τελειώνουν με ένα '-y' στον ενικό αριθμό (π.χ. "sky", "city") συμπεριφέρονται με τον ίδιο τρόπο όταν μετασχηματίζονται στον πληθυντικό, παίρνοντας την κατάληξη '-ies' ("skies", "cities"). Έτσι, αντί μιας λίστας από όλα τα πιθανά παράγωγα από μια ρίζα λέξης, το υπολογιστικό λεξικό περιέχει μόνο τη ρίζα, μαζί με τις πληροφορίες για το τι τύπος λέξης είναι.

Η γραμματική στην ανάλυση διαγραμμάτων ακολουθεί τον φορμαλισμό βασισμένο στην ενοποίηση και αποτελείται από τους κανόνες που αποφασίζουν πώς οι Κατευθυνόμενοι Ακυκλικοί Γράφοι που συνδέονται με τις λέξεις στο λεξικό πρέπει να ομαδοποιηθούν και να ενοποιηθούν.

Τα κύρια πλεονεκτήματα της μείωσης του μεγέθους του λεξικού και της γραμματικής είναι ότι ο χρόνος ανάπτυξης και ο χρόνος υπολογισμού, μειώνονται [20].

2.2 JavaChart

Ο αναλυτής διαγραμμάτων JavaChart, του Lars Degerstedt [15], είναι ένα open-source λογισμικό, που δημιουργήθηκε για την ικανοποίηση των νέων αναγκών, που προκύπτουν σε συστήματα διαλόγων, όπως

- ενοποίηση με εφαρμογές Java και open-source λογισμικό,

- μια ευέλικτη διεπαφή εφαρμογής, που απλοποιεί την αλληλεπίδραση με την υπο-μονάδα JavaChart server και
- μία τυποποιημένη, μερική ανάλυση σημασιολογίας με εφαρμογές που ικανοποιούν τον ανεπίσημο τρόπο που χρησιμοποιείται μία φυσική γλώσσα, όπως ημερομηνίες, ακέραιοι αριθμοί, άγνωστα κύρια ονόματα κ.τ.λ.

Η ιδέα για τον συγκεκριμένο αναλυτή προέρχεται από έναν αναλυτή βασισμένο σε LISP, τον Flexchart, που δημιουργήθηκε και χρησιμοποιήθηκε στο εργαστήριο Επεξεργασίας Φυσικής Γλώσσας (NLPLAB), του πανεπιστημίου Linköping της Σουηδίας.

Το εργαστήριο αυτό, ανήκει στο τμήμα Επιστήμης Υπολογιστών και Πληροφορικής, και ασχολείται με την επεξεργασία φυσικής γλώσσας και με τομείς σχετικούς με την Υπολογιστική Γλωσσολογία και την Γλωσσική Επιστήμη.

Ο γραμματικός φορμαλισμός που χρησιμοποιείται και από τους δύο είναι ο PATR-II, τον οποίο αναλύσαμε σε προηγούμενη παράγραφο. Ο τύπος του λεξικού που χρησιμοποιεί ο JavaChart είναι βασισμένο στο αποκαλούμενο μοντέλο mini lexicon, στο οποίο το λεξικό χωρίζεται σε δύο μέρη, στο λεξικό που περιέχει τις ρίζες των λέξεων και σε ένα που περιέχει καταλήξεις.

Τυπική Σύνταξη: Σαν κατηγορία ονομάζουμε ένα γραμματικά συμβολικό όνομα. Τα μονοπάτια ιδιοτήτων (feature paths) και οι τιμές ιδιοτήτων (feature values) μαζί, ονομάζονται έκφραση ιδιοτήτων. Μία εξίσωση ιδιοτήτων $f_1 = f_2$ είναι μία εξίσωση που περιέχει δύο εκφράσεις ιδιοτήτων f_1 και f_2 . Η εμφάνιση μιας κατηγορίας $c(n)$ είναι το ζεύγος μιας κατηγορίας c και ενός μη-αρνητικού αριθμού n . Ένας γραμματικός κανόνας βασισμένος σε ιδιότητες (feature-based), r της μορφής $h \rightarrow B, E$, περιέχει την εμφάνιση μιας κατηγορίας h , μία όχι άδεια αλληλουχία από εμφανίσεις κατηγοριών $B = b_1, \dots, b_n$ και μία, ίσως και άδεια, αλληλουχία από εξισώσεις ιδιοτήτων. Μία λεξιλογική είσοδος l της μορφής $(c = t, E)$ είναι μία τριάδα από μία κατηγορία c , ένα σύμβολο (token) και μία αλληλουχία από εξισώσεις ιδιοτήτων E .

Απεικόνιση του Διαγράμματος: Μια ακμή ((chart) edge) είναι μία τετράδα (i_s, i_e, r, F) , όπου i_s και i_e είναι ακέραιοι αριθμοί, r είναι ένας κανόνας τελείας (dotted rule), όπως τον

περιγράψαμε στην παράγραφο 2.1.4.3. και F είναι μία δομή ιδιοτήτων (feature structure), όπως την περιγράψαμε στην παράγραφο 2.1.3.

Στρατηγική Bottom-Up: Το σχήμα της ανάλυσης διαγραμμάτων είναι μία εξαντλητική διαδικασία βασισμένη σε τρεις διαδικασίες: σάρωση, πρόβλεψη και συνδυασμός. Ο αλγόριθμος είναι ο εξής, ακολουθώντας την στρατηγική Bottom-Up:

Είσοδος: Μία ακολουθία από σύμβολα (tokens).

Έξοδος: Ένα διάγραμμα.

Μέθοδος: Επανάλαβε τα ακόλουθα βήματα για κάθε θέση $i = 1, \dots, n+1$ μέχρι να μη μπορούν να προστεθούν στο διάγραμμα άλλες ακμές.

Σάρωση: Για κάθε λεξιλογική είσοδο της μορφής $(c = t_i, E)$, πρόσθεσε μία ακμή $(i, i+1, c- > t_i, E)$.

Πρόβλεψη: Για κάθε ακμή της μορφής $(i, j, c- > B_1, E)$ και για κάθε κανόνα της μορφής $h- > cB_2, E$, πρόσθεσε την ακμή $(i, j, h- > cB_2, E)$, αν η ακμή είναι καινούρια.

Συνδυασμός: Για κάθε ακμή της μορφής $(i, j, h- > B_1 \cdot b(n)B_2, E_1)$ και για κάθε ακμή της μορφής $(j, k, b(0)- > B, E_2)$, πρόσθεσε την ακμή $i, k, h- > B_1 b \cdot B_2, \text{unify}(E_1, \{b(n): E_2(b(0))\})$, αν πετύχει η ενοποίηση και η ακμή είναι καινούρια.

Το σύστημα JavaChart περιέχει έναν μεταγλωττιστή jchartc που παίρνει μία γραμματική (feature-based grammar), ένα ή περισσότερα λεξικά που περιέχουν ρίζες των λέξεων, και ένα ονομαζόμενο μίνι-λεξικό, που περιέχει τις καταλήξεις, που συμπληρώνουν τις ρίζες και δημιουργούν τις λέξεις. Ο μεταγλωττιστής παράγει ‘αφηρημένο κώδικα’ σε αρχείο που χρησιμοποιείται από τον εκτελεστή, που υλοποιεί τον βασικό αλγόριθμο ανάλυσης διαγραμμάτων. Ο εκτελεστής έχει δύο διεπαφές, μία μέσω γραμμής εντολών και μία μέσω ενός Java API. Και οι δύο διεπαφές παίρνουν σαν είσοδο μία έκφραση. Για το παρόν παράδειγμα χρησιμοποιούμε την διεπαφή μέσω γραμμής εντολών.

Παρακάτω βλέπουμε μία απλή γραμματική, που χρησιμοποιεί σύνταξη JavaChart:

```
s -> np vp : 0 cat = s :
              0 components np = 1 :
              0 components vp = 2 .

np -> arg *{0,1}: 0 cat = np :
                  0 components = 1 .

vp -> v np : 0 cat = vp :
              0 components v = 1 :
              0 components np = 2 .
```

Στη γραμματική αυτή υπάρχουν 3 γραμματικοί κανόνες που ακολουθούνται από αντίστοιχους κατευθυνόμενους ακυκλικούς γράφους (Κ.Α.Γ.), οι οποίοι ορίζουν και τους κανόνες της ενοποίησης. Η τελεία στο τέλος κάθε Κ.Α.Γ υπονοεί το τέλος κάθε γραμματικού κανόνα. Η έκφραση $\ast\{0,1\}$ σημαίνει ότι στο τέλος του np κανόνα χρειάζεται το λιγότερο 0 και το περισσότερο 1 ανώνυμη λέξη.

- ✓ Ο πρώτος κανόνας δηλώνει ότι μία πρόταση s μπορεί να περιέχει μία φράση ουσιαστικού (np) και μία φράση ρήματος. Οι ανέραυλοι θετικοί στα δεξιά του κανόνα ορίζουν τις θέσεις μέσα στον γραμματικό κανόνα και αναφέρονται στη δομή που θα παραχθεί όταν μία πρόταση εισαχθεί στο σύστημα και αναλυθεί από τον αναλυτή διαγραμμάτων. Για την πρόταση s ορίζεται ότι είναι τύπου κατηγορίας (cat) που περιέχει ως συστατικά στοιχεία μία φράση ουσιαστικού, που ορίζεται από την αντίστοιχη φράση μέσα στον κανόνα και είναι στην θέση 1, και μία φράση ρήματος, που ορίζεται πάλι από την αντίστοιχη φράση μέσα στον κανόνα και είναι στη θέση 2.
- ✓ Ο δεύτερος γραμματικός κανόνας ορίζει ότι μία φράση ουσιαστικού μπορεί να περιέχει ένα ή δύο ορίσματα, είναι κατηγορίας φράση ουσιαστικού (0 cat = np) και τα συστατικά στοιχεία της ορίζονται από τα ορίσματα αυτά.
- ✓ Ο τρίτος γραμματικός κανόνας αναφέρεται στη φράση ρήματος, η οποία αποτελείται από ένα όρισμα κατηγορίας v, δηλαδή ένα ρήμα και από μία φράση ρήματος. Η κατηγορία της φράσης είναι φράση ρήματος (0 cat = vp) και τα συστατικά στοιχεία της είναι ένα ρήμα που προέρχεται από την θέση ένα μέσα στον κανόνα και μία φράση ρήματος, που ο αναλυτής διαγραμμάτων θα αναζητήσει στη θέση 2.

Ακολουθεί ένα λεξικό, που χρησιμοποιεί σύνταξη JavaChart:

```
pelle = arg (stop) :    0 lex = pelle .
maja = arg (stop) :    0 lex = maja .
like = v (v1) :        0 lex = like .
```

Στην προκειμένη περίπτωση υπάρχουν δύο ορίσματα, *pelle* και *maja* (arguments) και ένα ρήμα *like* (verb). Σε ένα λεξικό με σύνταξη JavaChart ορίζονται οι κατηγορίες στις οποίες ομαδοποιούνται οι λέξεις, όπως γίνεται και εδώ (*arg* και *v*), ενώ στα δεξιά βλέπουμε σε ποια λέξη αντιστοιχίζεται η εισαγωγή του χρήστη μέσα στο λεξικό (*lex*).

Τέλος, υπάρχει και ένα μινι-λεξικό με σύνταξη JavaChart :

```
V1
_ = * (stop):          0 vform = pres :
                       0 active = yes .

ing = * (stop):        0 vform = pres :
                       0 active = yes .

s = * (stop):          0 vform = pres :
                       0 active = no .

ed = * (stop):         0 vform = pret .
@
```

Μετά από κάθε κατηγορία καταλήξεων όπως αυτή ορίζεται να ξεκινά με το όνομα (V1) και να τελειώνει με το σύμβολο @ υπάρχει η κατάληξη, η οποία μπορεί να αντιστοιχηθεί με την ρίζα λέξης. Στην προκειμένη περίπτωση διακρίνονται οι εξής καταλήξεις:

- ✓ το σύμβολο ‘_’ χρησιμοποιείται όταν δε χρειάζεται να υπάρξει κατάληξη στη λέξη γιατί ορίζεται ήδη από την ρίζα της, για παράδειγμα το ρήμα ‘tell’, όπου στον ενεστώτα δεν έχει άλλη διακριτική κατάληξη,
- ✓ η κατάληξη ‘-ing’ που υποδηλώνει γερούνδιο
- ✓ η κατάληξη ‘-s’ για το τρίτο πρόσωπο ενικού
- ✓ η κατάληξη ‘-ed’ για τους παρελθοντικούς χρόνους.

Ο αστερίσκος με τη λέξη *stop* εντός παρενθέσεως δηλώνει ότι η κατάληξη θα είναι αυτή και μόνο αυτή. Η πληροφορία που ακολουθεί δεξιά αφορά στον χρόνο και τη φωνή που χαρακτηρίζουν την κάθε κατάληξη.

Αυτά τα αρχεία μπορούν να τρέξουν από τη γραμμή εντολών (Unix) ως εξής. Έστω η ενδεχόμενη πρόταση ‘*maja likes pelle*’

```
% jchartc.sh -g ex_grammar.jcg -l ex_lexicon.jcl -m ex_minilexicon.jcm
Lexicon saved to file: lexicon.obj
Mini lexicon saved to file: minilexicon.obj
Grammar saved to file: grammar.obj

% jchart.sh -cat s maja likes pelle
s:
[[cat: s ]
 [components: [vp: [cat: vp ]
                   [components: [v: [lex: like]]
                                   [vform: pres]
                                   [active: no]
                                   [$token: likes]]]
 [np: [components: [lex: pelle]
                   [$token: pelle]]
       [cat: np ]]]]
 [np: [cat: np ]
       [components: [lex: maja]
                     [$token: maja]]]]]
```

Παράχεται μία δομή ιδιοτήτων η οποία παράχεται από τους κανόνες ενοποίησης που ορίζονται στη γραμματική και τα λεξικά. Έχει τη μορφή λίστας και περιέχει τις κατηγορίες στις οποίες ανήκουν τα συστατικά μέλη της πρότασης, δηλαδή την πρόταση s με τα συστατικά της στοιχεία, τα οποία είναι η φράση ρήματος και ουσιαστικού. Ανάλογα με την πληροφορία που ενοποιείται στη γραμματική και φθάνει μέχρι το μίνι-λεξικό, που περιέχει τις καταλήξεις των λέξεων, φθάνει σε βάθος και η δομή ιδιοτήτων.

2.3 Συστήματα Διαλόγου σε περιβάλλον ψηφιακής τηλεόρασης

Σε αυτήν την ενότητα, θα αναφερθούμε σε ένα project με πεδίο δράσης την περιοχή των συστημάτων διαλόγων με εφαρμογές στην ψηφιακή τηλεόραση. Αυτό είναι το MIINA (Multimodal Interaction for Information Appliances) project. Το MIINA είναι ένα project ανάμεσα στο Natural Language Processing Group του τμήματος Επιστήμης Υπολογιστών και Πληροφορικής, του πανεπιστημίου Linköping και της Nokia Home Communications. Ο στόχος του συγκεκριμένου project είναι να ερευνήσει και να αναπτύξει πολυμορφικά (multimodal) αλληλεπιδραστικά συστήματα για το μελλοντικό 'e-home' και συγκεκριμένα, για ένα σύνολο από εργασίες σχετικές με τη διαχείριση της πληροφορίας μέσα σε ένα σπίτι. Η προσέγγιση που ακολουθείται σε αυτό το project βασίζεται σε μια αποδοτική στρατηγική αναζήτησης σε συγκεκριμένες εργασίες, με τελική κατεύθυνση την υλοποίηση ενός γενικότερου πλαισίου.

Πριν από κάποιο καιρό, η NHC ανέπτυξε το Nokia Mediaterminal, ένα σύστημα που ενοποιεί σε μία υπο-μονάδα το Internet και την ψηφιακή τηλεόραση. Αυτή η τεχνολογία παρέχει στον χρήστη την ευκολία να μπορεί να πλοηγείται στο internet, να βλέπει τηλεόραση, να διαβάζει και να στέλνει e-mail, να παίζει μουσική, όλα με τη χρήση ενός τηλεχειριστηρίου. Ο τελικός χρήστης του Mediaterminal είναι ένα άνθρωπος στο σπίτι του, που χρησιμοποιεί το Mediaterminal επί καθημερινής βάσεως. Αφού, λοιπόν η ομάδα-στόχος για την συσκευή αυτή είναι τόσο μεγάλο και πολύπλευρο, και η ποσότητα της πληροφορίας που προσφέρεται στον χρήστη είναι τεράστια, υπάρχει ανάγκη για την ευκολία στην χρήση μιας διεπαφής για τις ερωτήσεις των χρηστών για το πρόγραμμα της τηλεόρασης. Θα πρέπει να αναφέρουμε ότι το σύστημα αυτό δεν βγήκε ποτέ στην αγορά από την NOKIA.

2.3.1 Το σύστημα NOKIA – TV Guide

Σα πλαίσια του project MIINA, αναπτύχθηκε ένα σύστημα που επιτρέπει στους χρήστες να εξαγάγουν σχετική και σωστή πληροφορία, με τρόπο αξιόπιστο, από το set-top-box Mediaterminal της Nokia. Αυτό πραγματοποιήθηκε με μία διεπαφή φυσικής γλώσσας σε μία βάση δεδομένων, που περιέχει προγράμματα τηλεόρασης. Σε ένα τέτοιο σύστημα, ο χρήστης μπορεί να κάνει ερωτήσεις για τα προγράμματα τηλεόρασης, τα κανάλια, τις κατηγορίες προγραμμάτων και τις ώρες που ξεκινά η μετάδοσή τους, σε φυσική γλώσσα και να παίρνει σωστές απαντήσεις από το σύστημα.

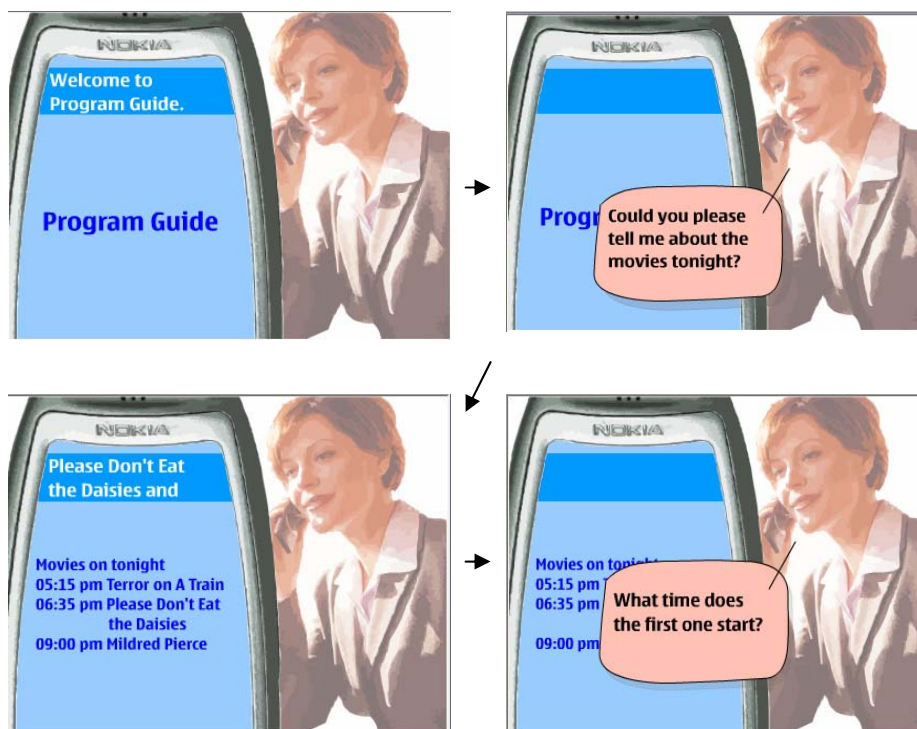
2.3.2 Το σύστημα NOKIA – Program Guide Information System

Το σύστημα Program Guide Information (NOKIA) [22] είναι μία ηλεκτρονική εφαρμογή τηλεφωνικής κλήσης που προσφέρει μία πληθώρα πληροφοριών πάνω σε προγράμματα τηλεόρασης. Ένας χρήστης μπορεί να καλέσει τον οδηγό προγραμμάτων από οποιοδήποτε τηλέφωνο και να ανακτήσει την επιθυμητή πληροφορία, συνδιαλεγόμενος με το σύστημα μέσω φωνητικής συνομιλίας, κάνοντας χρήση απλών φυσικών εκφράσεων.

Από την πλευρά του χρήστη, ακολουθούνται δύο βήματα για την ανάκτηση της πληροφορίας. Αρχικά, ο χρήστης μπορεί να ζητήσει πληροφορία πάνω σε τηλεοπτικά προγράμματα. Έπειτα, μπορεί είτε να επιλέξει κάποιο από τα προγράμματα που επεστράφησαν και να ζητήσει επιπρόσθετες πληροφορίες, είτε να προχωρήσει σε μία νέα αναζήτηση.

Το σύστημα Program Guide Information περιέχει επίσης, χαρακτηριστικά προσωποποίησης. Ο χρήστης μπορεί να θέσει τις προτιμήσεις του, όπως τύπους προγραμμάτων, κανάλια κ.α. μέσω ενός δικτυακού τόπου, και να κάνει χρήση αυτών την επόμενη φορά που θα χρησιμοποιήσει το σύστημα.

Θα πρέπει να επισημάνουμε στο σημείο αυτό, ότι σημαντικό στοιχείο της υπηρεσίας αποτελεί η χρέωσή της, που εύκολα επιτυγχάνεται λόγω της χρήσης του κινητού τηλεφώνου. Παρακάτω, παρουσιάζουμε μέσα από μία σειρά στιγμιότυπων από το φωνητικό αυτό σύστημα φυσικής γλώσσας της Nokia, τη μερική λειτουργικότητα που παρέχει στους χρήστες του. Η πληροφορία που φαίνεται στην οθόνη του κινητού, δεν υπάρχει σε γραπτή μορφή αλλά το σύστημα συνομιλεί με τον χρήστη μέσω φωνητικών μηνυμάτων.



Σχήμα 6: Προσομοίωση του συστήματος Talkative-TV μέσω στιγμιότυπων

Στην πρώτη εικόνα, ο χρήστης αφού έχει καλέσει το νούμερο της υπηρεσίας, καλωσορίζεται στο σύστημα που περιμένει την εισαγωγή της ερώτησής του. Στην δεύτερη εικόνα, ο χρήστης εισάγει την ερώτησή του στα Αγγλικά (το σύστημα προσφέρει τη λειτουργικότητα του στα Αγγλικά και στα Σουηδικά), που αφορά στις ταινίες που μεταδίδονται το ίδιο βράδυ από την ψηφιακή του τηλεόραση. Στην τρίτη εικόνα, το σύστημα απαντά στον χρήστη τα αποτελέσματα της αναζήτησής του. Τέλος, στην τέταρτη εικόνα, ο χρήστης εισάγει ερώτηση

πάνω στα αποτελέσματα της πρώτης του ερώτησης, για την ώρα μετάδοσης μίας ταινίας από την λίστα των αποτελεσμάτων.

Ανακεφαλαίωση

Στο παρόν κεφάλαιο μελετήσαμε αρχές, ορισμούς και αλγορίθμους που εφαρμόζονται σε συστήματα επεξεργασίας φυσικής γλώσσας. Έγινε εκτενής παρουσίαση του αναλυτή διαγραμμάτων που χρησιμοποιήθηκε στην υλοποίηση του συστήματός μας (JavaChart) και τέλος, παρουσιάστηκαν τα υπάρχοντα συστήματα διαλόγου, εφαρμοσμένα σε περιβάλλοντα ψηφιακής τηλεόρασης. Στο επόμενο κεφάλαιο θα ασχοληθούμε με το πρότυπο TV-Anytime και την περιγραφή των σχημάτων, πάνω στα οποία στηρίχθηκε η οργάνωση της πληροφορίας και των αποτελεσμάτων.

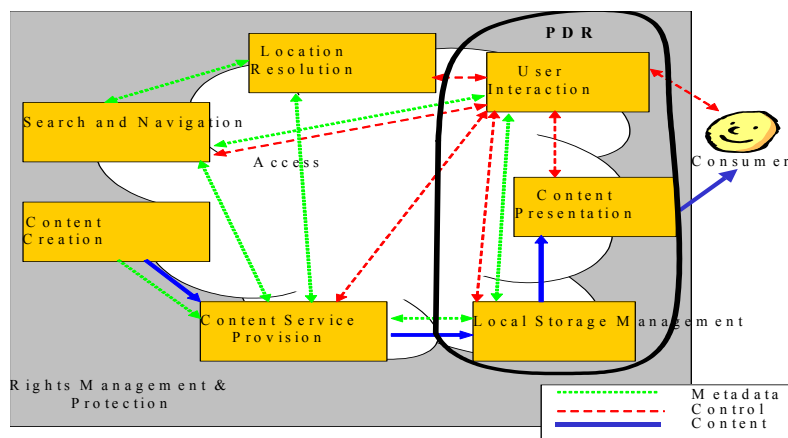
Κεφάλαιο 3

Το πρότυπο TV-Anytime

Όσο κατά την θεωρητική μελέτη στα πλαίσια της Διπλωματικής εργασίας, τόσο και κατά τον σχεδιασμό και την υλοποίηση του μοντέλου της φυσικής γλώσσας, μελετήθηκε και χρησιμοποιήθηκε το πρότυπο TV-Anytime. Στο κεφάλαιο αυτό γίνεται μία αναφορά στο πρότυπο γενικότερα, στην έννοια των μεταδεδομένων και στα περιγραφικά σχήματα, τα οποία είναι συμβατά με την πληροφορία που διαχειρίζεται το σύστημα για να παρέχει την απαιτούμενη λειτουργικότητα στο χρήστη.

3.1 Η γενική οργάνωση ενός συστήματος TV-Anytime

Ένα σύστημα TVA απαρτίζεται από τρία βασικά δομικά στοιχεία: Τον παροχέα των TVA υπηρεσιών (Service Provider), τον παροχέα των μηχανισμών πρόσβασης στο περιεχόμενο (transport provider) και την αποθηκευτική συσκευή που χρησιμοποιείται από τον τελικό χρήστη, η οποία αποθηκεύει το οπτικοακουστικό υλικό και αναλαμβάνει την αναπαραγωγή του. Τα παραπάνω συνοψίζονται στο παρακάτω σχήμα, όπου παρουσιάζεται η λογική κατάτμηση ενός TVA συστήματος σε επιμέρους υποσυστήματα.



Σχήμα 11: Λογική κατάτμηση ενός TV-Anytime συστήματος σε επιμέρους υποσυστήματα

Το υλικό που χρησιμοποιείται από τις TVA υπηρεσίες παρέχεται από κινηματογραφικά στούντιο και εταιρίες παραγωγής οπτικοακουστικών προγραμμάτων και αποτελεί τη λειτουργικότητα «δημιουργία του υλικού» (content creation).

Η «παροχή υπηρεσιών πάνω στο περιεχόμενο» (content service provider) πραγματοποιείται από οργανισμούς, όπως τα τηλεοπτικά κανάλια, που αναλαμβάνουν την προετοιμασία του περιεχομένου για μετάδοση, την προσθήκη μεταδεδομένων σε αυτό και την διάθεσή του στους χρήστες του συστήματος.

Τα τμήματα που εμφανίζονται στην παραπάνω εικόνα εντός του πλαισίου αντιστοιχούν σε μονάδες λειτουργικότητας που παρέχονται από το Προσωπικό Σύστημα Καταγραφής Video (Personal Video Recorders-PVRs και Personal Digital Recorders-PDRs) που έχει στην κατοχή του ο χρήστης και το οποίο παρέχει:

- Αναζήτηση και πλοήγηση (Search and Navigation) στο διαθέσιμο υλικό βάσει ποικίλων τρόπων όπως εικόνες, περιλήψεις κ.α.
- Εντοπισμό του Υλικού (Location Resolution) που έχει επιλέξει ο χρήστης να δει. Ο εντοπισμός γίνεται με τους Κωδικούς Αναφοράς στο Περιεχόμενο (Content Reference ID-CRID), που χαρακτηρίζουν την ταυτότητα του αντικείμενου ανεξάρτητα από τις φυσικές παραμέτρους της πρόσβασης στο ίδιο το αντικείμενο. Σημειώνεται ότι το κάθε πρόγραμμα χαρακτηρίζεται από έναν CRID.
- Αλληλεπίδραση με το χρήστη (User Interaction): Ο χρήστης χρησιμοποιεί τη λειτουργικότητα ενός PVR μέσω κατάλληλων υποσυστημάτων αλληλεπίδρασης (user interfaces).
- Παρουσίαση του υλικού (Content Presentation): Το PVR θα παρέχει εξελιγμένες δυνατότητες πρόσβασης στο αποθηκευμένο υλικό.
- Διαχείριση Αποθηκευτικού Χώρου (Local Storage Management): Ο χρήστης θα πρέπει να έχει τη δυνατότητα να επιλέγει ποια αντικείμενα θα παραμείνουν αποθηκευμένα στη συσκευή και ποια θα διαγραφούν ανάλογα με τις προτιμήσεις του.

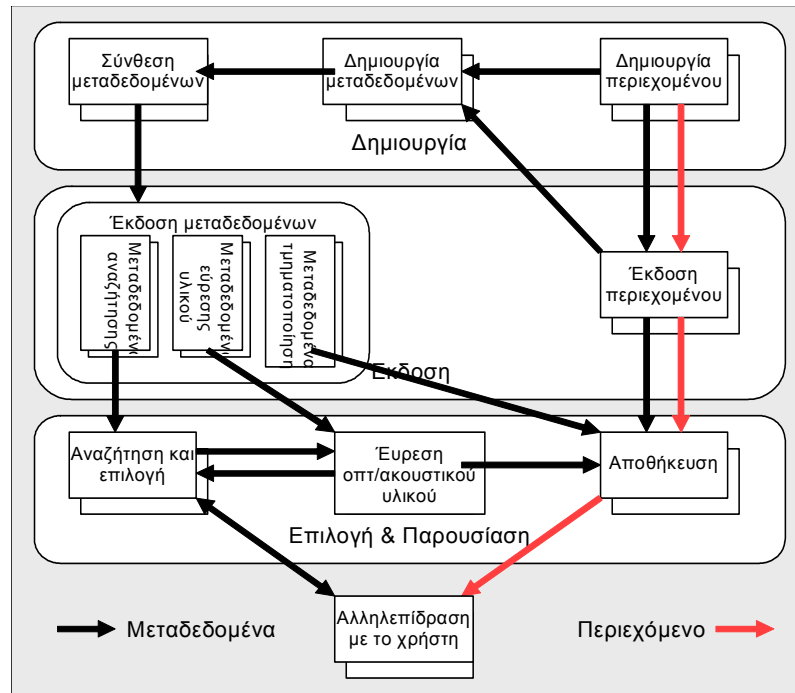
3.2 Μεταδεδομένα στο TV-Anytime

Τα μεταδεδομένα, γενικά ορίζονται σαν την «πληροφορία για την πληροφορία». Το TV-Anytime σύστημα μεταδεδομένων επιτρέπει στον καταναλωτή να βρει, να πλοηγηθεί και να διαχειριστεί υλικό από μία ποικιλία από εσωτερικές και εξωτερικές πηγές. Καθορίζει ένα συγκεκριμένο τρόπο για την περιγραφή των προφίλ των καταναλωτών, που περιλαμβάνει προτιμήσεις αναζήτησης, ώστε να διευκολύνει το αυτόματο φιλτράρισμα και την απόκτηση του υλικού, μέσω πρακτόρων για λογαριασμό των καταναλωτών.

Το μοντέλο του TV-Anytime, διαφοροποιεί τα μεταδεδομένα σε τρεις κατηγορίες, ανάλογα με τον τρόπο δημιουργίας τους:

- **Μεταδεδομένα περιγραφής περιεχομένου** (content description metadata): γενικές πληροφορίες που δεν αλλάζουν ανεξάρτητα από τον τρόπο διάθεσης του περιεχομένου.
- **Μεταδεδομένα περιγραφής στιγμιότυπων** (instance description metadata): πληροφορίες για το συγκεκριμένο τύπο διάθεσης του περιεχομένου.
- **Μεταδεδομένα περιγραφής καταναλωτή** (consumer metadata): πληροφορίες για τις κινήσεις του χρήστη, τις προτιμήσεις του και προσωπικές σημειώσεις.

Στο σχήμα που ακολουθεί φαίνονται οι σχέσεις μεταξύ των κατηγοριών μεταδεδομένων τόσο μεταξύ τους όσο και με το περιεχόμενο:



Σχήμα 12 : Κατηγορίες και πορεία των μεταδεδομένων μέσα στο TV-Anytime

Παρατηρούμε ότι ταυτόχρονα με τη δημιουργία του περιεχομένου, δημιουργούνται και τα μεταδεδομένα περιγραφής περιεχομένου, τα οποία συνδέονται και παρουσιάζονται μαζί με το περιεχόμενο (π.χ. οι ηθοποιοί μιας ταινίας). Σε αυτό το στάδιο τα μεταδεδομένα δεν είναι, πιθανότατα, σε μορφή τέτοια ώστε να παρουσιαστούν στο χρήστη κατευθείαν. Μόλις δημιουργηθεί το υλικό είναι έτοιμο προς έκδοση από τους παροχείς προγραμμάτων. Σε αυτό το σημείο ορίζονται μεταδεδομένα που αφορούν συγκεκριμένες μεταδόσεις του υλικού (π.χ. ώρα μετάδοσης), αλλά και συγκεκριμένα τμήματά του (segmentation metadata). Σε αυτό το στάδιο, τα μεταδεδομένα περιγραφής περιεχομένου συνδυάζονται με αυτά της περιγραφής της έκδοσης που δημιουργούνται τώρα, ώστε να είναι δυνατές η αναζήτηση και η επιλογή του υλικού. Μόλις επιλεγεί το υλικό, με τα μεταδεδομένα εύρεσης υλικού βρίσκεται η φυσική του θέση και προβάλλεται στο χρήστη. Στο τρίτο στάδιο, έχουμε τη δημιουργία των μεταδεδομένων ιστορικού χρήσης, που παράγονται κατά την αλληλεπίδραση του χρήστη με το υλικό, τα οποία και θα χρησιμοποιηθούν στην επόμενη διαδικασία αναζήτησης και επιλογής (π.χ. αν κάποιος επιλέξει να δει ένα ντοκιμαντέρ, στην επόμενη πλοήγησή του στο υλικό πρέπει να βρει ευκολότερα παρόμοιο υλικό).

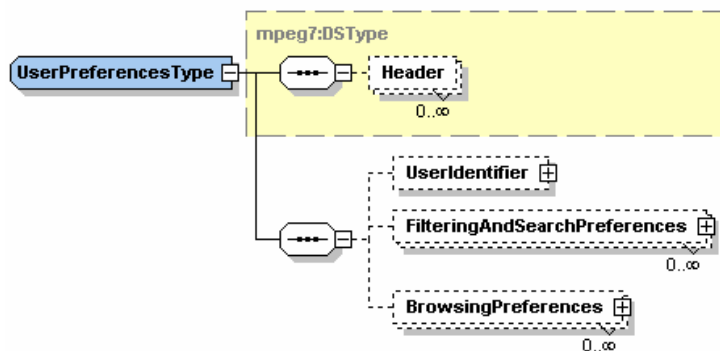
Το TV-Anytime για να περιγράψει με σαφήνεια τη δομή των μεταδεδομένων αυτών έχει χρησιμοποιήσει και αυτό την XML Schema, όπως και το MPEG7. Όντας πιο εξειδικευμένη

και μεταγενέστερη εφαρμογή από το MPEG7, χρησιμοποιεί αρκετούς τύπους βασικών στοιχείων του MPEG7 (π.χ. ελεγχόμενοι όροι, media locators), ακολουθώντας ωστόσο τη δική του, εντελώς διαφορετική, δομή.

Στη συγκεκριμένη εργασία, θα πρέπει να επισημάνουμε ότι από τη μεριά της οργάνωσης της πληροφορίας, όπως αυτή εισάγεται στο σύστημα από τον χρήστη, το ενδιαφέρον εστιάζεται στα μεταδεδομένα καταναλωτή και συγκεκριμένα στις προτιμήσεις του χρήστη (User Preferences), γιατί όλη η λειτουργικότητα που παρέχεται αφορά σε εξατομικευμένη πληροφορία πάνω στο υλικό [11]. Από την μεριά, όμως, της ανάκτησης, της οργάνωσης και της παρουσίασης της πληροφορίας, όπως αυτή έρχεται από τον TV-Anytime server, μετά το φιλτράρισμα της πληροφορίας του χρήστη, αφορά στα μεταδεδομένα περιγραφής περιεχομένου και στα μεταδεδομένα περιγραφής στιγμιότυπων. Στη συνέχεια θα δούμε αναλυτικά τα σχήματα για την περιγραφή των προτιμήσεων του χρήστη και για την περιγραφή του περιεχομένου και των στιγμιότυπων.

3.3 Βασικά στοιχεία των προτιμήσεων του χρήστη

UserPreferences DS



Η περιγραφή των προτιμήσεων του χρήστη μπορεί να συσχετιστεί με την περιγραφή του υλικού για την αναζήτηση, το φιλτράρισμα και την κατανάλωση του επιθυμητού υλικού. Το UserPreferences DS συνδέεται με έναν συγκεκριμένο χρήστη (ή ένα σύνολο χρηστών) διαμέσου του UserIdentifier DS. Η κύρια οντότητα στο διάγραμμα, το UsagePreferences DS, περιέχει δύο κύρια συστατικά, το BrowsingPreferences DS και το FilteringAndSearchPreferences DS. Το BrowsingPreferences DS μπορεί να χρησιμοποιηθεί

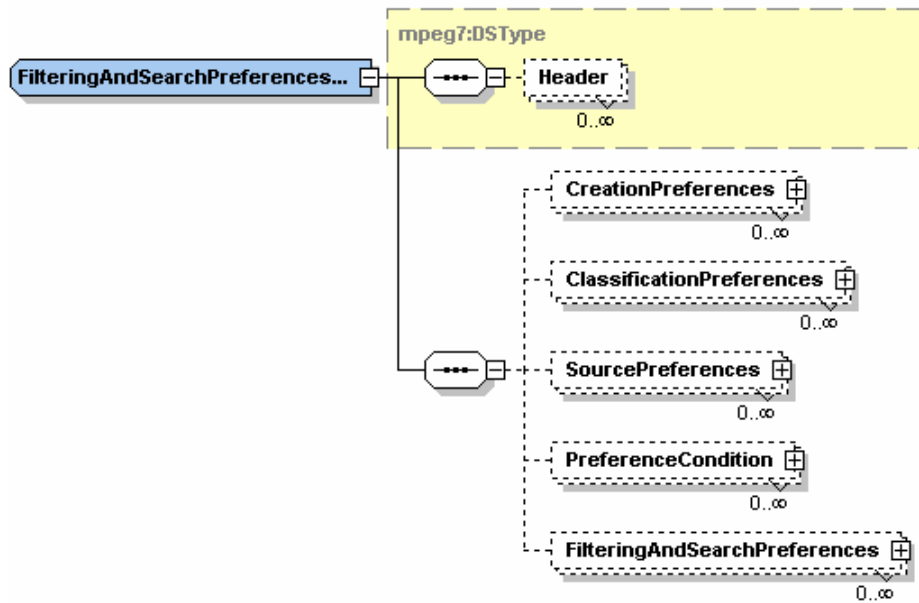
για να καθορίσει τις προτιμήσεις για τον τρόπο που καταναλώνεται το υλικό, και περιέχει τα SummaryPreferences. Το FilteringAndSearchPreferences DS μπορεί να χρησιμοποιηθεί για να καθορίσει τις προτιμήσεις για τον τύπο του υλικού, για να μπορεί να οδηγήσει στην αναζήτηση, το φιλτράρισμα, την επιλογή και την κατανάλωσή του. Αυτό το σχήμα περιέχει το ClassificationPreferences DS, CreationPreferences DS και το SoursePreferences DS.

Το UserPreferences DS χρησιμοποιείται για να περιγράψει τις προτιμήσεις ενός χρήστη όσον αφορά στην κατανάλωση του υλικού. Η αντιστοίχιση της πληροφορίας για τις προτιμήσεις του χρήστη και της περιγραφή του υλικού, επιτρέπει την εξατομίκευση της προσπέλασης και κατανάλωσης του υλικού.

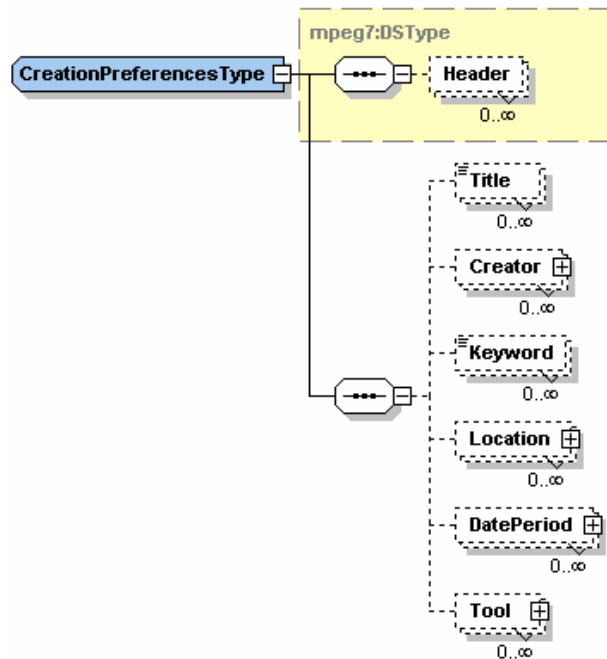
UserIdentifier DS



Το UserIdentifier DS μπορεί να χρησιμοποιηθεί για την συσχέτιση ενός συγκεκριμένου χρήστη (ή σύνολο από χρήστες) με ένα συγκεκριμένο user preference description, ή να αναγνωρίσει ένα συγκεκριμένο user preference description, ή να μπορεί να ξεχωρίσει πολλαπλά user preference descriptions. Το στοιχείο Name μπορεί να περιέχει το πραγματικό όνομα του χρήστη, ένα ψευδώνυμο, το όνομα του λογαριασμού του χρήστη και γενικά οποιοδήποτε όνομα. Έτσι, ο χρήστης μπορεί να έχει πολλά διαφορετικά user preference descriptions, που να τα χρησιμοποιεί σε διαφορετικές καταστάσεις.

FilteringAndSearchPreferences DS

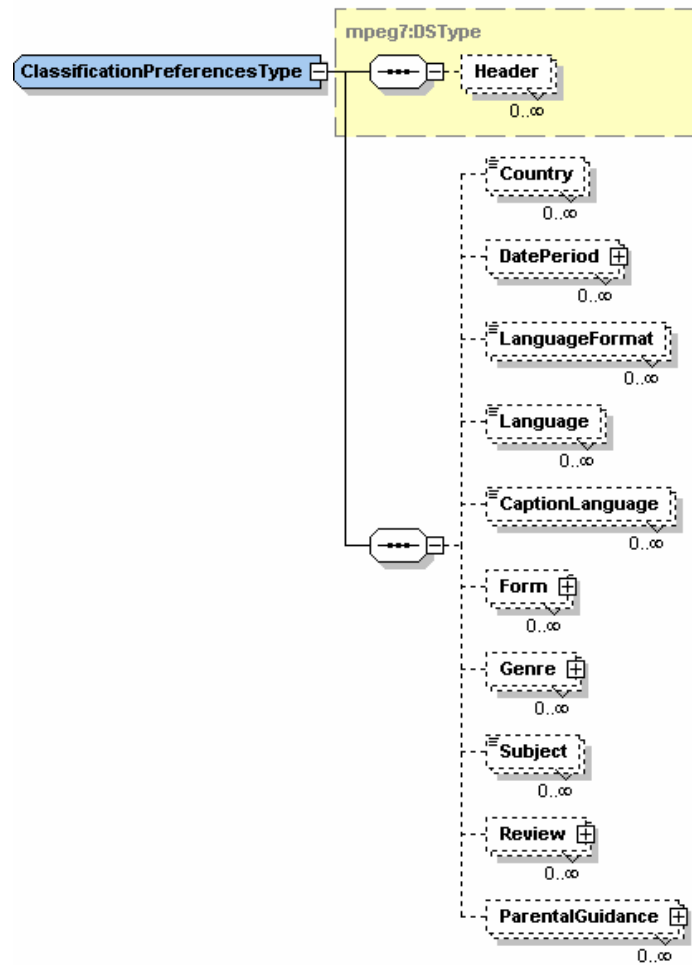
Το **FilteringAndSearchPreferences DS** καθορίζει τις προτιμήσεις του χρήστη για το φιλτράρισμα και/ή την αναζήτηση του οπτικο-ακουστικού υλικού. Αυτές οι προτιμήσεις μπορούν να καθοριστούν στα πλαίσια των συσχετιζόμενων ιδιοτήτων, όσον αφορά στη δημιουργία (creation), την ταξινόμηση (classification) και την πηγή (source) του υλικού.

CreationPreferences DS

Το CreationPreferences DS καθορίζει τις προτιμήσεις του χρήστη για τις ιδιότητες συσχετιζόμενες με τη δημιουργία του οπτικο-ακουστικού υλικού, όπως αγαπημένοι ηθοποιοί κ.α. Αναλυτικά, στον παρακάτω πίνακα βλέπουμε τη σημασιολογική ανάλυση των επιμέρους συστατικών ενός CreationPreferences DS.

Όνομα	Ορισμός
Title	Περιγράφει τις προτιμήσεις του χρήστη για τον τίτλο του υλικού.
Creator	Περιγράφει τις προτιμήσεις του χρήστη για τον συντελεστή (π.χ. σκηνοθέτη ή ηθοποιό) του υλικού.
Keyword	Περιγράφει μία λέξη κλειδί που χαρακτηρίζει το υλικό που προτιμά ο χρήστης.
Location	Περιγράφει την προτίμηση του χρήστη για τον τόπο δημιουργίας του υλικού.
DatePeriod	Περιγράφει την προτίμηση του χρήστη για την χρονική περίοδο δημιουργίας του υλικού.
preferenceValue	Περιγράφει την σχετική προτίμηση του χρήστη με ανάθεση βάρους στο συγκεκριμένο creation preference description, σε σχέση με άλλα που μπορεί να υπάρχουν.

Ένα χαρακτηριστικό preferenceValue μπορεί να συσχετιστεί με το κάθε ένα από τα παραπάνω στοιχεία για να δηλώσει την σχετική προτεραιότητα σε σχέση με άλλα στοιχεία του ίδιου τύπου.

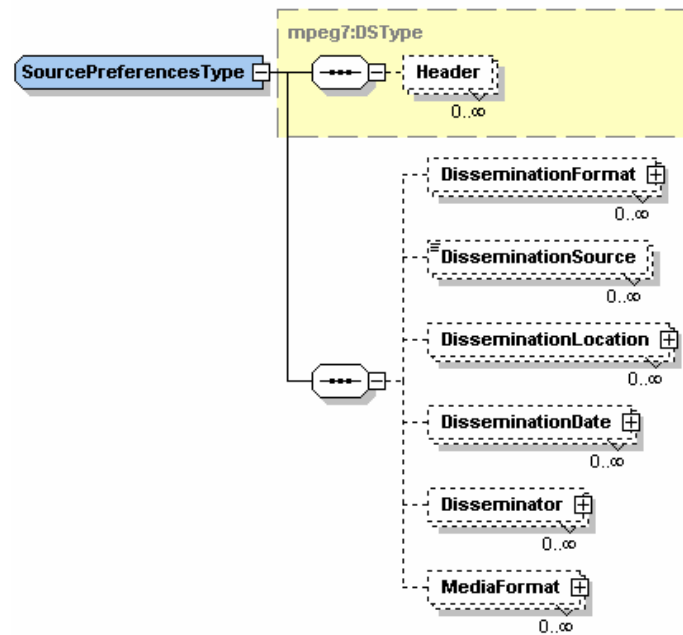
ClassificationPreferences DS

Το ClassificationPreferences DS χρησιμοποιείται για να μεταφέρει τις προτιμήσεις του χρήστη πάνω σε διάφορες ταξινομήσεις του υλικού, όπως, για παράδειγμα προτίμηση στη γλώσσα. Αναλυτικά, στον παρακάτω πίνακα βλέπουμε τη σημασιολογική ανάλυση των επιμέρους συστατικών ενός ClassificationPreferences DS.

Όνομα	Ορισμός
Country	Περιγράφει τις προτιμήσεις του χρήστη για την χώρα προέλευσης του υλικού.
DatePeriod	Περιγράφει τις προτιμήσεις του χρήστη για την χρονική περίοδο πρώτης έκδοσης του υλικού.
Language	Περιγράφει τις προτιμήσεις του χρήστη για την γλώσσα προέλευσης του υλικού.
Genre	Περιγράφει τις προτιμήσεις του χρήστη για το είδος του υλικού.

Όνομα	Ορισμός
Subject	Περιγράφει τις προτιμήσεις του χρήστη για το θέμα του οπτικο-ακουστικού υλικού. Το θέμα ταξινομεί το υλικό από την άποψη του τύπου του, χωρίς να λαμβάνει υπόψη του το είδος του (genre classification).
MediaReview	Περιγράφει τις προτιμήσεις του χρήστη για τις κριτικές πάνω στο υλικό.
ParentalGuidance	Περιγράφει τις προτιμήσεις του χρήστη για κατάταξη του υλικού σε σχέση με το κοινό στο οποίο απευθύνεται, βάση ενός σχήματος, που μπορεί να διαφέρει ανάμεσα στους οργανισμούς ή στις διάφορες χώρες. Το ParentalGuidance περιλαμβάνει τις προδιαγραφές της χώρας, το σχήμα κατάταξης και την τιμή κατάταξης βάση του συγκεκριμένου σχήματος. Επίσης, περιλαμβάνει την κατάταξη βάση ταξινόμησης ανά ηλικία για συγκεκριμένα γκρουπ θεατών.
preferenceValue	Περιγράφει την σχετική προτίμηση του χρήστη με ανάθεση βάρους στο συγκεκριμένο classification preference description, σε σχέση με άλλα που μπορεί να υπάρχουν.

Ένα χαρακτηριστικό preferenceValue μπορεί να σχετιστεί με το κάθε ένα από τα παραπάνω στοιχεία, εκτός του ParentalGuidance, για να δηλώσει την σχετική προτεραιότητα σε σχέση με άλλα στοιχεία του ίδιου τύπου.

SourcePreferences DS

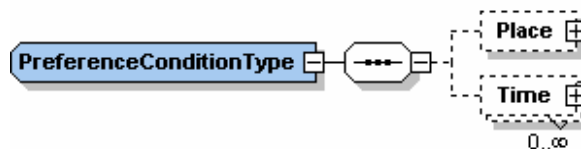
Το SourcePreference DS χρησιμοποιείται για να μεταφέρει τις προτιμήσεις του χρήστη για την πηγή του οπτικο-ακουστικού υλικού, όπως ο εκδότης, το κανάλι ή ο διανομέας. Αναλυτικά, στον παρακάτω πίνακα βλέπουμε τη σημασιολογική ανάλυση των επιμέρους συστατικών ενός SourcePreference DS.

Όνομα	Ορισμός
PublicationType	Περιγράφει τις προτιμήσεις του χρήστη για το μέσο έκδοσης ή τον μηχανισμό παράδοσης του υλικού.
PublicationSource	Περιγράφει τις προτιμήσεις του χρήστη για μια συγκεκριμένη πηγή του υλικού, όπως για παράδειγμα ένα κανάλι μετάδοσης ή έναν server.
PublicationPlace	Περιγράφει τις προτιμήσεις του χρήστη για τον τόπο διανομής του υλικού.
PublicationDate	Περιγράφει τις προτιμήσεις του χρήστη για την ώρα και την ημέρα διάθεσης του υλικού, όπως για την ώρα και την ημέρα μετάδοσης ή διάθεσης του υλικού.
Publisher	Περιγράφει τις προτιμήσεις του χρήστη για τον εκδότη του υλικού, δηλαδή το πρόσωπο, γκρουπ ή οργανισμό που διαθέτει το υλικό για κατανάλωση από τον υπεύθυνο μετάδοσης ή τον διανομέα.
MediaFormat	Περιγράφει τις προτιμήσεις του χρήστη για τον τύπο του υλικού, π.χ. οπτικο-ακουστικό, ακουστικό και λοιπά.
noRepeat	Δηλώνει το αν ο χρήστης επιθυμεί το υλικό να μην είναι επανάληψη υλικού που ήταν ξανά διαθέσιμο (προαιρετικό).
noEncryption	Δηλώνει το αν ο χρήστης επιθυμεί το υλικό να μην

Όνομα	Ορισμός
	είναι κρυπτογραφημένο (προαιρετικό).
noPayPerUse	Δηλώνει το αν ο χρήστης επιθυμεί το υλικό να μην είναι διαθέσιμο επί πληρωμή (προαιρετικό).
preferenceValue	Περιγράφει την σχετική προτίμηση του χρήστη με ανάθεση βάρους στο συγκεκριμένο source preference description, σε σχέση με άλλα που μπορεί να υπάρχουν.

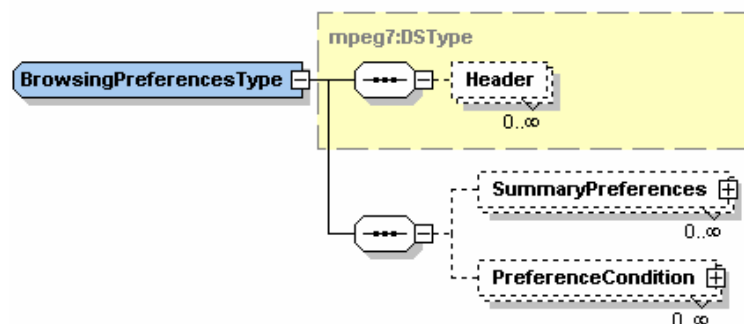
Ένα χαρακτηριστικό preferenceValue μπορεί να σχετιστεί με το κάθε ένα από τα παραπάνω στοιχεία, εκτός από τα noRepeat, noEncryption και noPayPerUse, για να δηλώσει την σχετική προτεραιότητα σε σχέση με άλλα στοιχεία του ίδιου τύπου.

PreferenceCondition DS

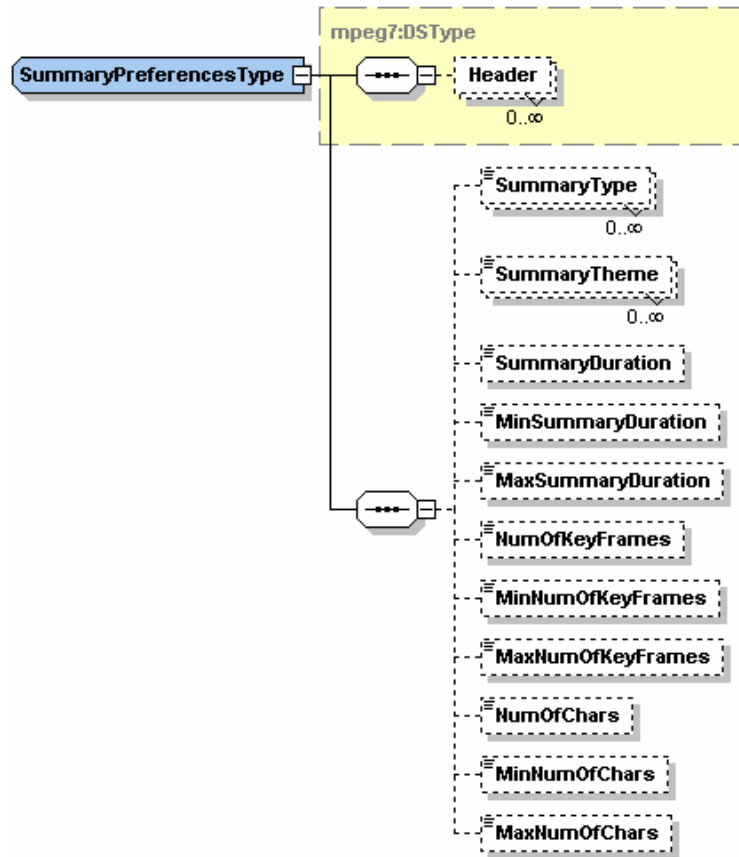


Το PreferenceCondition DS χρησιμοποιείται για να καθορίσει έναν συνδυασμό από χρόνο και/ή τόπο, που συσχετίζεται με ένα συγκεκριμένο σετ από προτιμήσεις του χρήστη.

BrowsingPreferences DS



Το BrowsingPreferences DS χρησιμοποιείται για να καθορίσει τις προτιμήσεις ενός χρήστη, όσον αφορά στην πλοήγηση και την πρόσβαση στο οπτικο-ακουστικού υλικού. Το BrowsingPreferences DS περιέχει SummaryPreferences.

SummaryPreferences DS

Το SummaryPreferences DS περιγράφει τις προτιμήσεις του χρήστη για μη σειριακή πλοήγηση του υλικού. Αναλυτικά, στον παρακάτω πίνακα βλέπουμε τη σημασιολογική ανάλυση των επιμέρους συστατικών ενός SummaryPreferences DS.

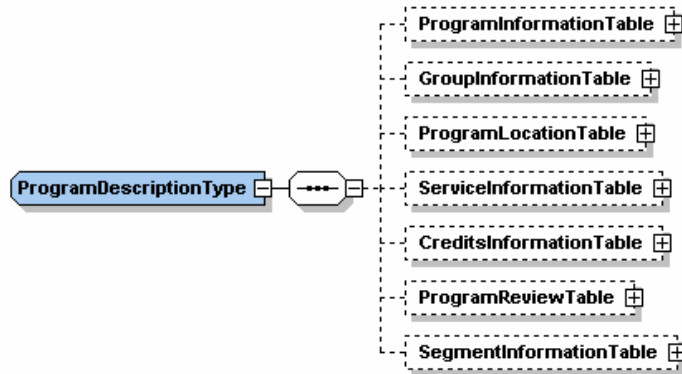
Όνομα	Ορισμός
SummaryTypePreference	Περιγράφει την προτίμηση του χρήστη για τον τύπο της περίληψης, σύμφωνα με τον τύπο των στοιχείων μέσα σε ένα media summary description.
PreferredSummaryTheme	Περιγράφει την προτίμηση του χρήστη για το όνομα της περίληψης ή το όνομα/θεματολογία για τα τμήματα του υλικού, όπου τα ονόματα/θεματολογίες περιλαμβάνονται στο summary and segment media descriptions.
NumOfKeyFrames	Περιγράφει την προτίμηση του χρήστη για τον αριθμό των εικονικών στιγμιотύπων-κλειδιών σε μία οπτική περίληψη.
MinNumOfKeyFrames	Περιγράφει την προτίμηση του χρήστη για τον ελάχιστο αριθμό των εικονικών στιγμιотύπων-κλειδιών σε μία οπτική περίληψη.
MaxNumOfKeyFrames	Περιγράφει την προτίμηση του χρήστη για τον

Όνομα	Ορισμός
	μέγιστο αριθμό των εικονικών στιγμιотύπων-κλειδιών σε μία οπτική περίληψη.
SummaryDuration	Περιγράφει την προτίμηση του χρήστη για την χρονική διάρκεια της οπτικο-ακουστικής περίληψης.
MinSummaryDuration	Περιγράφει την προτίμηση του χρήστη για την ελάχιστη χρονική διάρκεια της οπτικο-ακουστικής περίληψης.
MaxSummaryDuration	Περιγράφει την προτίμηση του χρήστη για την μέγιστη χρονική διάρκεια της οπτικο-ακουστικής περίληψης.
NumOfChars	Περιγράφει την προτίμηση του χρήστη για το μήκος, σε αριθμό χαρακτήρων, μίας γραπτής περίληψης.
MinNumOfChars	Περιγράφει την προτίμηση του χρήστη για το ελάχιστο μήκος, σε αριθμό χαρακτήρων, μίας γραπτής περίληψης.
MaxNumOfChars	Περιγράφει την προτίμηση του χρήστη για το μέγιστο μήκος, σε αριθμό χαρακτήρων, μίας γραπτής περίληψης.
preferenceValue	Περιγράφει την σχετική προτίμηση του χρήστη με ανάθεση βάρους στο συγκεκριμένο source preference description, σε σχέση με άλλα που μπορεί να υπάρχουν.

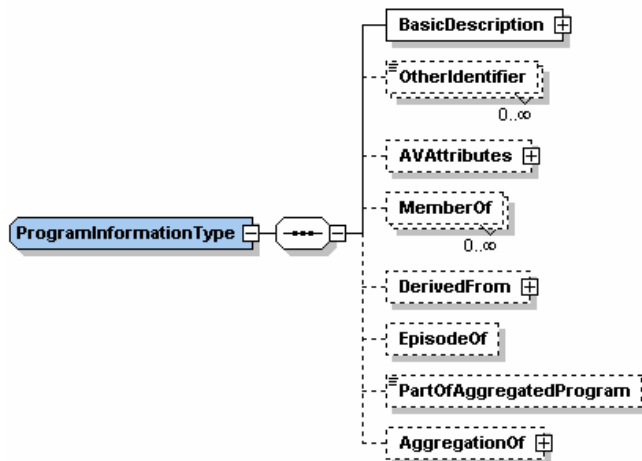
Ένα χαρακτηριστικό preferenceValue μπορεί να σχετιστεί με το SummaryTypePreference και το PreferredSummaryTheme, για να δηλώσει την σχετική προτεραιότητα σε σχέση με άλλα στοιχεία του ίδιου τύπου.

3.4 Βασικά στοιχεία περιγραφής προγραμμάτων

Το βασικό στοιχείο που έχει όλη την πληροφορία για τα προγράμματα είναι το ProgramDescription.

ProgramDescription DS

Από το ProgramDescription χρησιμοποιούμε το ProgramInformationTable που περιέχει το ProgramInformation, και το ProgramLocationTable που περιέχει το BroadcastEvent. Ξεκινώντας από το ProgramInformation έχουμε το παρακάτω περιγραφικό σχήμα

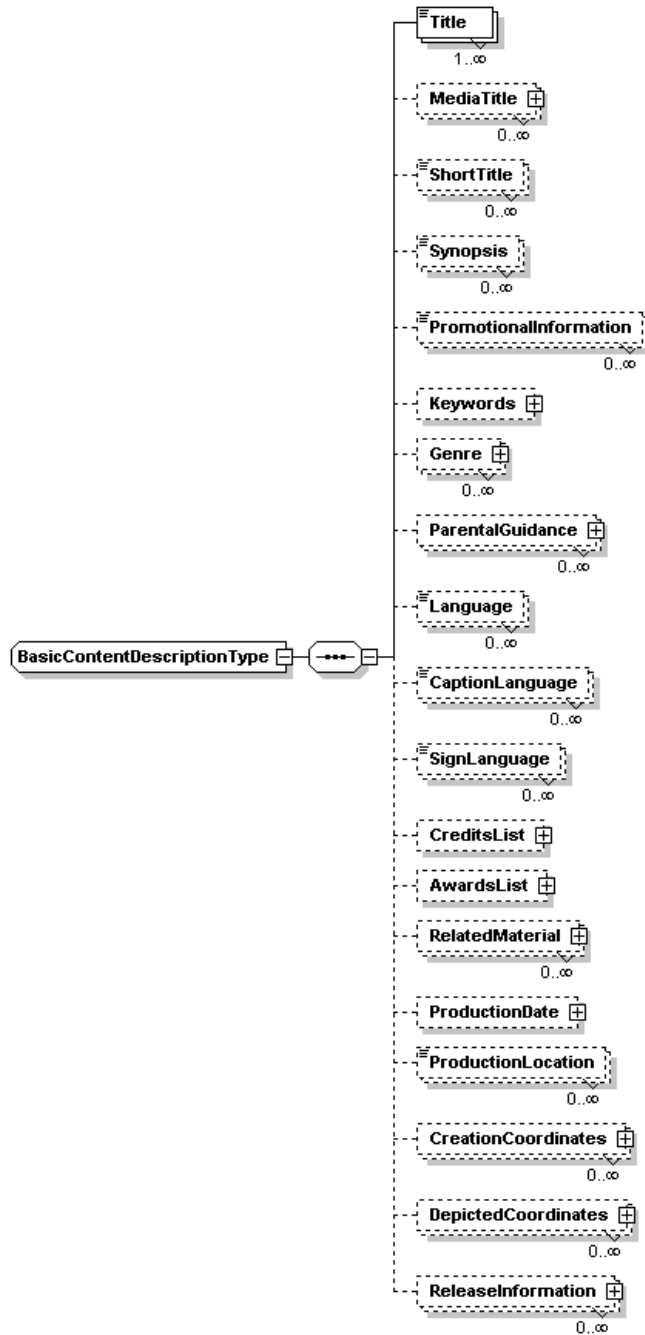
ProgramInformation DS

Τα στοιχεία που θα χρειαστούν περιγράφονται πιο αναλυτικά παρακάτω:

- BasicDescription (Η βασική περιγραφή του προγράμματος)

BasicDescription DS

Από αυτά θα εξεταστεί πιο αναλυτικά το BasicDescription όπου βρίσκεται η βασική



περιγραφή του προγράμματος.

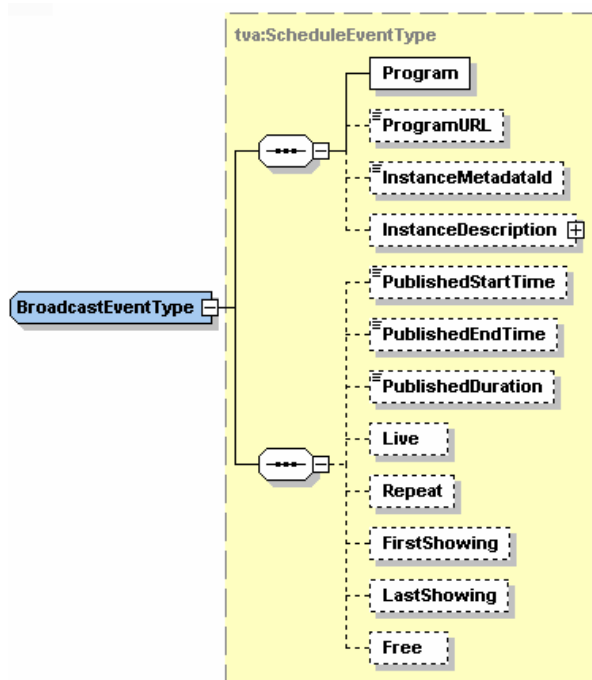
Αναλυτικά τα στοιχεία που χρησιμοποιούνται επεξηγούνται παρακάτω:

Όνομα	Ορισμός
Title	Ο τίτλος του προγράμματος. Ένα πρόγραμμα μπορεί να έχει περισσότερους από ένα τίτλους, π.χ. σε διαφορετικές γλώσσες
ShortTitle	Σύντομος τίτλος που χρησιμοποιείται για λόγους παρουσίασης

Όνομα	Ορισμός
Synopsis	Μία σύντομη του προγράμματος (είναι σε μορφή κειμένου)
Keywords	Μία λίστα από λέξεις ή φράσεις που περιγράφουν επιγραμματικά το πρόγραμμα
Genre	Θεματική κατηγορία. Ορίζεται από το TVA ένας θησαυρός από θεματικές κατηγορίες
Language	Περιγράφει μία ομιλούσα γλώσσα του προγράμματος

Συνεχίζοντας, έχουμε το περιγραφικό σχήμα για το BroadcastEvent

BroadcastEvent DS



και από τα παραπάνω χρησιμοποιούμε τα παρακάτω, στην παρουσίαση των αποτελεσμάτων στο χρήστη.

Όνομα	Ορισμός
PublishedStartTime	Η χρονική στιγμή που ξεκινά η μετάδοση του προγράμματος
PublishedEndTime	Η χρονική στιγμή που τελειώνει η μετάδοση του προγράμματος
PublishedDuration	Η χρονική διάρκεια μετάδοσης του προγράμματος

Ανακεφαλαίωση

Στο παρόν κεφάλαιο, παρουσιάσαμε τη γενική οργάνωση ενός TV-Anytime συστήματος και αναλύσαμε την έννοια των μεταδεδομένων εξετάζοντας τις κατηγορίες στις οποίες αυτά χωρίζονται. Στη συνέχεια, για να μπορέσουμε να εξηγήσουμε το σκεπτικό πάνω στο οποίο στηρίχθηκε το μοντέλο της φυσικής γλώσσας, αναλύσαμε τα βασικά στοιχεία που χαρακτηρίζουν τις προτιμήσεις ενός χρήστη, πάνω στα οποία θα στηρίζουμε τη δομή της πληροφορίας που εισάγεται στο σύστημα. Έπειτα, αναλύσαμε τα βασικά στοιχεία που περιγράφουν ένα πρόγραμμα, τα οποία αποτελούν και τα στοιχεία που συνοδεύουν ένα πρόγραμμα κατά την παρουσίασή του στον χρήστη. Στο επόμενο κεφάλαιο, θα ασχοληθούμε διεξοδικά με την ανάπτυξη και τον σχεδιασμό του μοντέλου της φυσικής γλώσσας, όπως αυτό προκύπτει για ένα περιβάλλον ψηφιακής τηλεόρασης.

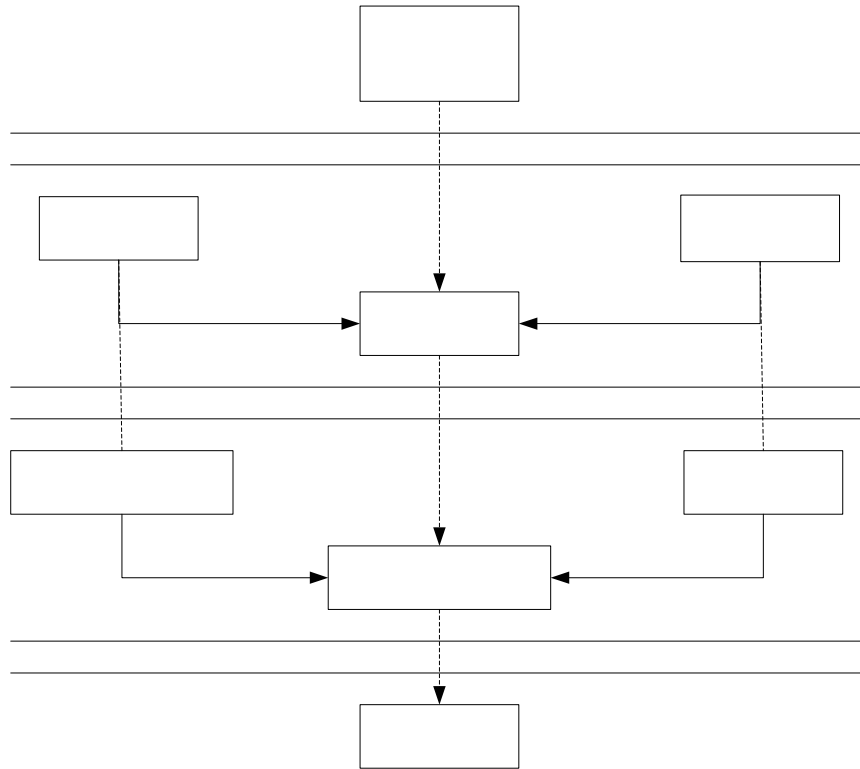
Κεφάλαιο 4

Το μοντέλο φυσικής γλώσσας σε περιβάλλον ψηφιακής τηλεόρασης

Στο κεφάλαιο αυτό παρουσιάζουμε το μοντέλο της φυσικής γλώσσας που σχεδιάστηκε, ποιες απαιτήσεις όρισαν τις παραμέτρους σχεδιασμού και πως μέσα από τον σχεδιασμό καλύπτεται η προτεινόμενη λειτουργικότητα. Μέσα από την παρουσίαση της δομής της πληροφορίας γίνεται η σύνδεση με το πρότυπο TV-Anytime, ενώ ιδιαίτερη αναφορά γίνεται στην στρατηγική αντιμετώπισης των ασαφειών στην εισαγωγή του χρήστη. Επίσης, μέσα από μία σειρά Use Cases παρουσιάζεται η ροή μέσα στο σύστημα.

4.1 Μεθοδολογία

Η υλοποίηση ενός συστήματος διαλόγου για μία νέα εφαρμογή, μπορεί να χωριστεί σε δύο ξεχωριστά βήματα: στον θεμελιώδη σχεδιασμό και στην προσαρμογή της δομής (Degerstedt & Johnsson, 2001). Ο σχεδιασμός και ο κώδικας, αποτελούν βήματα πάνω στη διαδικασία της ανάπτυξης, και θεωρείται ότι γίνονται το ένα μετά το άλλο, ξεκινώντας από τον σχεδιασμό. Συνοπτικά, μπορούμε στο παρακάτω σχήμα να δούμε το διάγραμμα ροής για την ανάπτυξη του συστήματος της συγκεκριμένης διατριβής.



Σχήμα 7: Το διάγραμμα εργασίας για την ανάπτυξη του συστήματος (Degerstedt & Johnsson, 2001)

Στο πρώτο στάδιο, τον θεμελιώδη σχεδιασμό, εμπλέκεται το θεωρητικό υπόβαθρο και οι προτεινόμενες τεχνολογίες από την αιχμή της τεχνολογίας (state of the art) πάνω σε φυσικές γλώσσες και οι προδιαγραφές των απαιτήσεων του συστήματος. Σε συνδυασμό με το κύριο σώμα εργασίας της φυσικής γλώσσας (δηλαδή το σύνολο της γραμματικής και των λεξικών) οδηγούμαστε στο σχεδιασμό του μοντέλου. Στο δεύτερο στάδιο, το στάδιο της υλοποίησης έχουμε την ενοποίηση του μοντέλου με το υπάρχον σύστημα και με υπο-μονάδες που προσάπτονται σε αυτό.

Θεωρία

Παρακάτω συνοψίζουμε τις προδιαγραφές:

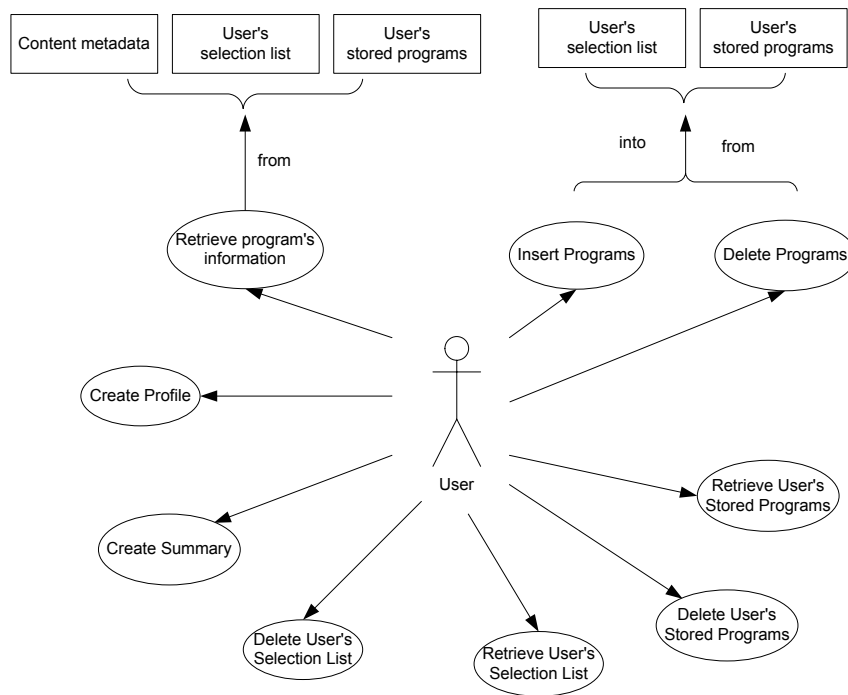
- Η αλληλεπίδραση είναι πληκτρολογημένη και είναι στα Αγγλικά
- Το σύστημα τρέχει πάνω στην πλατφόρμα του UP-TV
- Στην παρούσα του μορφή υποστηρίζεται η διαχείριση διαλόγου σε αρχική μορφή.

Η πολυπλοκότητα του συστήματος βρίσκεται ανάμεσα στα δύο επίπεδα του πίνακα 1, μιας και το σύστημα δεν υποστηρίζει διασαφηνιστικές ερωτήσεις.

Υπάρχον Σύστη

4.2 Προδιαγραφές Απαιτήσεων

Το μοντέλο που αναπτύχθηκε στην εργασία αυτή, καλύπτει μία σειρά από απαιτήσεις, που προσδιορίζουν την τελική του λειτουργικότητα. Στο παρακάτω διάγραμμα παρουσιάζουμε τις λειτουργίες που θα πρέπει το σύστημα να ικανοποιεί:



Επιγραμματικά θα αναφέρουμε τις απαιτήσεις του συστήματος, εφόσον αυτές αναλύονται σε βάθος στην παράγραφο της λειτουργικότητας του παρόντος κεφαλαίου. Έτσι, λοιπόν το σύστημα θα πρέπει να ικανοποιεί τις απαιτήσεις του χρήστη για

- Ανάκτηση εξατομικευμένης πληροφορίας περιεχομένου μεταδεδομένων
 - ✓ το σύνολο των μεταδεδομένων των τηλεοπτικών προγραμμάτων,
 - ✓ την εξατομικευμένη λίστα προγραμμάτων του χρήστη,
 - ✓ ο σύνολο των αποθηκευμένων προγραμμάτων στο PDR του χρήστη.

Μέσα από μία τέτοια λειτουργικότητα, προσφέρεται στον χρήστη ένας οδηγός προγραμμάτων, είτε για προγράμματα που περιέχονται στη βάση του συστήματος ή σε

κάποιον on-demand server του συστήματος, είτε στην εξατομικευμένη λίστα προγραμμάτων του χρήστη (selection list), που ταιριάζουν στις προτιμήσεις που έχει δηλώσει στο προφίλ του, είτε στα αποθηκευμένα προγράμματα στο PDR του.

- Διαχείριση τηλεοπτικών προγραμμάτων, μέσω πληροφορίας μεταδεδομένων
 - ✓ στην εξατομικευμένη λίστα προγραμμάτων του χρήστη (selection list),
 - ✓ στο σύνολο των αποθηκευμένων προγραμμάτων στο PDR του χρήστη.

Θα πρέπει στο σημείο αυτό να εξηγήσουμε την λειτουργία του selection list. Ο χρήστης μπορεί, ανά πάσα στιγμή να δημιουργήσει ένα ή περισσότερα προφίλ για λογαριασμό του, στα οποία να δηλώσει τις προτιμήσεις του. Τότε, το σύστημα αναλαμβάνει να ταιριάζει τις προτιμήσεις του χρήστη με πληροφορία που προέρχεται από τα μεταδεδομένα των προγραμμάτων και να αποθηκεύσει σε μία λίστα την πληροφορία για τα προγράμματα αυτά. Μέσα στην λίστα αυτή, για κάθε ένα πρόγραμμα υπάρχει και πληροφορία για την κατάσταση στην οποία βρίσκεται σε σχέση πάντα με το συγκεκριμένο προφίλ. Οι καταστάσεις που μπορούμε να διακρίνουμε είναι οι εξής:

- ✓ **Undefined:** Κατάσταση στην οποία βρίσκεται ένα πρόγραμμα όταν δεν έχει αποφασιστεί από το σύστημα ή τον χρήστη αν θα γραφεί ή όχι στο PDR.
 - ✓ **ToBeRecorded:** Κατάσταση στην οποία βρίσκεται ένα πρόγραμμα όταν έχει δοθεί εντολή από το σύστημα ή τον χρήστη για εγγραφή.
 - ✓ **Recorded:** Κατάσταση στην οποία βρίσκεται ένα πρόγραμμα όταν έχει γραφεί στο PDR.
 - ✓ **ToBeDeleted:** Κατάσταση στην οποία βρίσκεται ένα πρόγραμμα όταν έχει δοθεί εντολή για διαγραφή από το PDR.
- Δημιουργία προφίλ χρήστη.
 - Δημιουργία περιλήψεων.
 - Ανάκτηση όλης της εξατομικευμένης λίστας προγραμμάτων του χρήστη.

- Διαγραφή όλης της εξατομικευμένης λίστας προγραμμάτων του χρήστη.
- Ανάκτηση του συνόλου των αποθηκευμένων προγραμμάτων του χρήστη στο PDR του.
- Διαγραφή του συνόλου των αποθηκευμένων προγραμμάτων του χρήστη στο PDR του.

4.3 Κύριο σώμα εργασίας (Corpus Work)

Σε αυτό το σημείο θα καθορίσουμε το κύριο σώμα της φυσικής γλώσσας που υλοποιήθηκε και μέσα από εδώ θα φανούν και οι προδιαγραφές του συστήματος. Η συλλογή του διαλόγου περιέχει έναν αριθμό από εκφράσεις, τις οποίες θα πρέπει το σύστημα να είναι σε θέση να απαντήσει. Τα συστατικά των εκφράσεων αυτών μπορούν να διακριθούν στις παρακάτω κατηγορίες:

- Εισαγωγικές φράσεις (Introduction phrases)
- Φράσεις αναζήτησης (Search phrases)
- Φράσεις στόχου (Target phrases)
- Χρονικές φράσεις (Temporal phrases)
- Φράσεις περίληψης (Summary phrases)

Ο στόχος στο σημείο αυτό, είναι μέσα από τις εκφράσεις αυτές να μοντελοποιήσουμε την λειτουργικότητα που μας παρέχεται από το πρότυπο TV-Anytime αλλά και σε τυχόν προεκτάσεις του να μπορούμε εύκολα και χωρίς να επέμβουμε στη δομή, να επεκτείνουμε το παρόν μοντέλο.

Η μικρότερη έκφραση περιέχει είτε μία εισαγωγική φράση και μία φράση αναζήτησης, είτε μία εισαγωγική φράση και μία φράση στόχου. Οι φράσεις περίληψης είναι ανεξάρτητες από όλες τις προηγούμενες, εκτός από τις χρονικές φράσεις που με τον συνδυασμό αυτών των δύο έχουμε μία έκφραση, χωρίς να είναι αναγκαία η παρουσία κάποιας άλλης κατηγορίας.

4.3.1 Εισαγωγικές Φράσεις

Οι εισαγωγικές φράσεις παρέχουν πληροφορία μόνο για το αν η ενέργεια που θα πρέπει να γίνει από την πλευρά του συστήματος αφορά στην ανάκτηση πληροφορίας για κάποιο πρόγραμμα ή κατηγορία προγραμμάτων, στην εισαγωγή ή διαγραφή κάποιου προγράμματος ή κατηγορίας προγραμμάτων στην/από την λίστα των προγραμμάτων που αναφέρονται στις προσωπικές του προτιμήσεις, και στην δημιουργία του προσωπικού προφίλ του χρήστη. Έτσι, μπορούμε και υποστηρίζουμε την λειτουργικότητα που μας παρέχει το πρότυπο TV-Anytime, μέσα από το μοντέλο μας. Στην ανάπτυξη της γραμματικής υπάρχουν τέσσερα είδη εισαγωγικών εκφράσεων.

- **Τύπος 1:** Ο χρήστης θέλει να ανακτήσει πληροφορία πάνω σε προγράμματα, αρχίζοντας την πρόταση με φράσεις, όπως “is there a”, “what”, “show me”.
- **Τύπος 2:** Ο χρήστης θέλει να εισάγει ένα πρόγραμμα στην προσωπική λίστα προγραμμάτων του ή να το γράψει στο κουτί του, αρχίζοντας την πρόταση με φράσεις, όπως “I would like to (store|add|insert|record)”.
- **Τύπος 3:** Ο χρήστης θέλει να διαγράψει ένα πρόγραμμα από την προσωπική λίστα προγραμμάτων του ή από το κουτί του, αρχίζοντας την πρόταση με φράσεις, όπως “I would like to (remove|delete)”
- **Τύπος 4:** Ο χρήστης θέλει να δηλώσει κάποια προτίμηση του που αφορά στο προφίλ του, αρχίζοντας την πρόταση με φράσεις, όπως “I (prefer|like)”

4.3.2 Φράσεις αναζήτησης

Από το μπλοκ της φράσης αυτής μπορούμε και εξάγουμε την κυρίως πληροφορία. Είναι υποχρεωτικό να υπάρχει τουλάχιστον μία φράση αναζήτησης. Ανάλογα με το πόσο συγκεκριμένος μπορεί να είναι ένας χρήστης, η πληροφορία μπορεί να εξαχθεί από όσες φράσεις αναζήτησης θέλει ο χρήστης. Οι φράσεις αναζήτησης αφορούν στο ίδιο πρόγραμμα ή στην ίδια κατηγορία προγραμμάτων.

Για τον τύπο των φράσεων αυτών χρησιμοποιήθηκαν 10 κύριες κατηγορίες ανάλογα με την ζητούμενη πληροφορία, χωρίς αυτό να αποτρέπει τη χρήση και άλλων κατηγοριών. Κάθε τέτοια φράση σε διαφορετική κατηγορία πληροφορίας, αντιστοιχίζεται με την πληροφορία που αναλύσαμε στα περιγραφικά σχήματα του TV-Anytime, του κεφαλαίου 3. Αναλυτικά:

- **Τύπος 1:** Περιέχει πληροφορία για τον τίτλο προγραμμάτων γενικά ή προγραμμάτων μιας συγκεκριμένης κατηγορίας, όπως ‘with title Gladiator’. Αντιστοιχίζεται με το πεδίο Title (CreationPreferences DS) του προτύπου TV-Anytime.
- **Τύπος 2:** Περιέχει πληροφορία για το όνομα και το ρόλο (όπως ηθοποιός, σκηνοθέτης, παραγωγός) που συμμετέχει σε προγράμματα γενικά ή σε προγράμματα μιας συγκεκριμένης κατηγορίας, όπως ‘starring Mel Gibson’. Αντιστοιχίζεται με το πεδίο Creator (CreationPreferences DS) του προτύπου TV-Anytime.
- **Τύπος 3:** Περιέχει πληροφορία για μία σειρά από λέξεις που χαρακτηρίζουν κάποια προγράμματα γενικά ή κάποιας συγκεκριμένης κατηγορίας, ‘about World War 2’. Αντιστοιχίζεται με το πεδίο Keyword (CreationPreferences DS) του προτύπου TV-Anytime.
- **Τύπος 4:** Περιέχει πληροφορία για την χρονική περίοδο δημιουργίας προγραμμάτων γενικά ή μιας συγκεκριμένης κατηγορίας, όπως ‘created after 1995’. Αντιστοιχίζεται με το πεδίο DatePeriod (ClassificationPreferences DS) του προτύπου TV-Anytime.
- **Τύπος 5:** Περιέχει πληροφορία για την χώρα προέλευσης προγραμμάτων γενικά ή μιας συγκεκριμένης κατηγορίας, όπως ‘created in Spain’. Αντιστοιχίζεται με το πεδίο Country (ClassificationPreferences DS) του προτύπου TV-Anytime.
- **Τύπος 6:** Περιέχει πληροφορία για την γλώσσα προγραμμάτων γενικά ή μιας συγκεκριμένης κατηγορίας, όπως ‘English spoken’. Αντιστοιχίζεται με το πεδίο Language (ClassificationPreferences DS) του προτύπου TV-Anytime.
- **Τύπος 7:** Περιέχει πληροφορία για το είδος, την κατηγορία των προγραμμάτων, όπως ‘comedy movies’. Αντιστοιχίζεται με το πεδίο Genre (ClassificationPreferences DS) του προτύπου TV-Anytime.

- **Τύπος 8:** Περιέχει πληροφορία για το κανάλι ή τον διακομιστή μετάδοσης προγραμμάτων γενικά ή μιας συγκεκριμένης κατηγορίας, όπως ‘broadcasted by ERT’. Αντιστοιχίζεται με το πεδίο DisseminationSource (SourcePreferences DS) του προτύπου TV-Anytime.
- **Τύπος 9:** Περιέχει πληροφορία για τον τόπο διανομής προγραμμάτων γενικά ή μιας συγκεκριμένης κατηγορίας, όπως ‘broadcasted from London’. Αντιστοιχίζεται με το πεδίο DisseminationLocation (SourcePreferences DS) του προτύπου TV-Anytime.
- **Τύπος 10:** Περιέχει πληροφορία για την ημερομηνία και την χρονική περίοδο μετάδοσης ή διαθεσιμότητας προγραμμάτων γενικά ή μιας συγκεκριμένης κατηγορίας, όπως ‘broadcasted in January’. Αντιστοιχίζεται με το πεδίο DisseminationDate (SourcePreferences DS) του προτύπου TV-Anytime.

4.3.3 Φράσεις Στόχου

Από τις φράσεις στόχου παίρνουμε πληροφορία για το που θέλει ο χρήστης να στοχεύσει η λειτουργία που θα εκτελέσει το σύστημα. Διακρίνουμε τρεις κατηγορίες φράσεων στόχου:

- **Τύπος 1:** Στοχεύει στην εξατομικευμένη λίστα προγραμμάτων του χρήστη, όπως ‘from my selection list’.
- **Τύπος 2:** Στοχεύει στα περιεχόμενα του PDR του χρήστη, όπως ‘from my PDR’
- **Τύπος 3:** Στοχεύει στο προφίλ του χρήστη, όπως ‘in my profile’.

Όταν στην έκφραση του χρήστη δεν υπάρχει φράση στόχου, τότε την πληροφορία την παίρνουμε από την εισαγωγική φράση, που όταν φανερώνει ανάγκη τότε γίνεται με γνώμονα όλα τα μεταδεδομένα του συστήματος.

4.3.4 Χρονικές Φράσεις

Οι χρονικές φράσεις που χρησιμοποιούνται στο σώμα της αίτησης του χρήστη, αφορούν στον χρόνο μετάδοσης και στο χρόνο δημιουργίας ενός ή περισσότερων προγραμμάτων και στο χρόνο παράδοσης κάποιας περίληψης του χρήστη. Οι περιπτώσεις που καλύπτονται μέσα από τη συγκεκριμένη κατηγορία φράσεων είναι οι παρακάτω:

- **Τύπος 1:** Η πληροφορία που δίνει ο χρήστης αφορά στο συγκεκριμένο χρονικό στιγμιότυπο.

- ✓ **Day:** Αναφέρεται στην ημέρα της εβδομάδος (π.χ. on Monday).
- ✓ **Month:** Αναφέρεται στον μήνα του χρόνου (π.χ. in August).
- ✓ **Year:** Αναφέρεται στην χρονολογία (π.χ. in 1995)
- ✓ **Time:** Αναφέρεται στην ώρα της μέρας (π.χ. at 6).
- ✓ **Date:** Αναφέρεται στην ημερομηνία, μέσα από έναν αριθμό και προηγείται πάντα φράσης, που αφορά σε μήνα, Month (π.χ. at 6 November).
- ✓ **Time Indicator:** Αναφέρεται στον δείκτη της ώρας όταν αυτή παρέχεται με αριθμό ανάμεσα στο 1 και το 12 (π.χ. at 6 am).
- ✓ **Day Indicator:** Αναφέρεται σε επαναλήψεις μίας δεδομένης χρονικής στιγμής (π.χ. every Monday).
- ✓ **Other:** Αναφέρεται σε ασαφείς περιγραφές του χρόνου, που αποκωδικοποιούνται και δίνουν τιμές σε κάποιες από τις παραπάνω κατηγορίες. Για παράδειγμα, αν ο χρήστης δώσει τη φράση 'tonight', τότε αυτό θα αποκωδικοποιηθεί σε φράση τύπου 4 και θα δώσει τιμή στην κατηγορία 'after' και 'time', 5 και στην κατηγορία 'before' και 'time', 12 και στην κατηγορία 'time indicator' την τιμή *pm*, θεωρώντας ότι η φράση 'tonight' χρονικά αντιστοιχίζεται ανάμεσα στις 5 – 12 μ.μ.

- **Τύπος 2:** Η πληροφορία που δίνει ο χρήστης αφορά σε οποιοδήποτε χρονικό στιγμιότυπο πριν τη δοσμένη χρονική στιγμή. Στον συγκεκριμένο τύπο δεν υπάρχει έννοια του day indicator. Για παράδειγμα η φράση ‘before 5 pm’.
- **Τύπος 3:** Η πληροφορία που δίνει ο χρήστης αφορά σε οποιοδήποτε χρονικό στιγμιότυπο μετά τη δοσμένη χρονική στιγμή. Στον συγκεκριμένο τύπο δεν υπάρχει έννοια του day indicator. Για παράδειγμα η φράση ‘after 5 pm’.
- **Τύπος 4:** Η πληροφορία που δίνει ο χρήστης αφορά σε οποιοδήποτε χρονικό στιγμιότυπο ανάμεσα σε δύο χρονικές στιγμές. Στον συγκεκριμένο τύπο δεν υπάρχει έννοια του day indicator. Για παράδειγμα η φράση ‘between 5 and 12 pm’.

4.3.5 Φράσεις περίληψης

Οι φράσεις περίληψης αφορούν στην δυνατότητα του χρήστη να ορίσει τις προτιμήσεις του για χαρακτηριστικά των περιλήψεων οπτικο-ακουστικού περιεχομένου που επιθυμεί, με τα χαρακτηριστικά που επιθυμεί ο ίδιος. Τα στοιχεία που χαρακτηρίζουν την κάθε περίληψη είναι τα παρακάτω:

- **Type:** Αφορά στον τύπο της περίληψης, για το αν αυτός θα είναι ακουστικός (audio), κείμενο (textual) ή οπτικός (visual). Αντιστοιχίζεται με το πεδίο SummaryType (SummaryPreferences DS) του προτύπου TV-Anytime.
- **Theme:** Αφορά στο κύριο θέμα της περίληψης (π.χ. about the tornado in Florida). Αντιστοιχίζεται με το πεδίο SummaryTheme (SummaryPreferences DS) του προτύπου TV-Anytime.
- **The exact:** Αφορά στον ακριβή
ή
- **The minimum:** Αφορά στον μικρότερο
ή
- **The maximum:** Αφορά στον μεγαλύτερο
ή
- **The minimum and maximum:** Αφορά στον μικρότερο και μεγαλύτερο

✓ **Number of characters:** αριθμό από χαρακτήρες

ή

✓ **Number of frames:** αριθμό από εικόνες

ή

✓ **Number of minutes or seconds:** αριθμό από λεπτά ή δευτερόλεπτα

που δίνει ο χρήστης και αντιστοιχίζεται στον τύπο της περίληψης (π.χ. of minimum 100 and maximum 200 chars). Η πληροφορία στους τύπους των φράσεων αυτών αντιστοιχίζεται με τα πεδία NumOfChars, MinNumOfChars, MaxNumOfChars, NumOfKeyFrames, MinNumOfKeyFrames, MaxNumOfKeyFrames, SummaryDuration, MinSummaryDuration, MaxSummaryDuration (SummaryPreferences DS) του προτύπου TV-Anytime.

- **Genre:** Αφορά στην κατηγορία ή το είδος του προγράμματος από το οποίο θα σχηματιστεί η περίληψη (π.χ. news summary). Αντιστοιχίζεται με το πεδίο Genre (PreferenceCondition DS του BrowsingPreference DS) του προτύπου TV-Anytime.

Ένα παράδειγμα φράσης περίληψης αποτελεί το παρακάτω:

‘a visual basketball summary of minimum 100 and maximum 200 characters about Knicks’ last game’.

4.4 Σχεδιασμός (Design)

Κατά τον σχεδιασμό του μοντέλου οι δύο αντικειμενικοί στόχοι ήταν ο ορισμός της συμπεριφοράς του συστήματος ανάλογα με την πληροφορία που συλλέγεται από την εισαγωγή της έκφρασης του χρήστη και μέσα από αυτό, η σχεδίαση των υπο-μονάδων του συστήματος και ο καθορισμός του ρόλου της καθεμιάς.

4.4.1 Το μεταμοντέλο των εκφράσεων της φυσικής γλώσσας

Όταν ο χρήστης εισάγει μία έκφραση, το σύστημα προσπαθεί μέσα από τον συνδυασμό των φράσεων που έχει χρησιμοποιήσει ο χρήστης, να οργανώσει την πληροφορία που δίνεται. Από την εισαγωγή της έκφρασης του χρήστη στο σύστημα δημιουργείται μία δομή ιδιοτήτων, που περιέχει όλη την πληροφορία σε μορφή, που το υπόλοιπο σύστημα μπορεί να επεξεργαστεί.

Στην προσπάθεια να δοθεί μία γενική δομή σε όλες τις πιθανές εκφράσεις, που μπορεί να χρησιμοποιήσει ένας χρήστης, δημιουργήθηκε το παρακάτω μοντέλο, που ουσιαστικά αποτελεί το μεταμοντέλο για τη δημιουργία των τύπων ή των επιτρεπτών εκφράσεων της φυσικής γλώσσας. Μία έκφραση μπορεί να αποτελεί μία ή περισσότερες οντότητες (entities), καθεμία από τις οποίες έχει διαφορετικά πεδία (fields), οι τιμές (values) των οποίων προέρχονται από την είσοδο του χρήστη. Ο τύπος των φράσεων, ουσιαστικά, ορίζει τον τύπο του αντικειμένου (entity), και αναλόγως αυτού ορίζονται τα πεδία που το αποτελούν.

4.4.2 Απεικόνιση πληροφορίας

Αναλυτικά, η δομή ιδιοτήτων που δημιουργείται και περιέχει μία ή περισσότερες παραμέτρους, παρουσιάζεται στο παρακάτω σχήμα. Όπως παρατηρούμε αποτελείται από τις οντότητες, τα πεδία και τις τιμές. Στις οντότητες γίνεται ο διαχωρισμός σε κατηγορίες που περιλαμβάνουν πληροφορία προερχόμενη από το πρότυπο TV-Anytime (properties) και κατηγορίες που περιλαμβάνουν πληροφορία για να κατευθύνουν το σύστημα στη σωστή διαχείριση του προτύπου και της υπόλοιπης πληροφορίας (markers), δηλαδή ένας διαχωρισμός για το αν η ενέργειες που θα ακολουθήσουν αφορούν στο σύνολο των μεταδεδομένων των τηλεοπτικών προγραμμάτων, στην εξατομικευμένη λίστα του χρήστη (selection list) ή στη λίστα των αποθηκευμένων προγραμμάτων του. Η κατηγορία temporal περιέχει πληροφορία που αφορά στο πρότυπο TV-Anytime, αλλά λόγω της ιδιαιτερότητας στη διαχείριση των χρονικών φράσεων στις φυσικές γλώσσες, αποτελεί ξεχωριστή κατηγορία. Τα πεδία, αντιστοιχίζουν τις τιμές με τις κατηγορίες του TV-Anytime στην περίπτωση της οντότητας properties και γενικότερα κάνουν συγκεκριμένη την πληροφορία και βοηθούν στη διαχείρισή της. Το πεδίο τιμή περιλαμβάνει όλες τις πιθανές τιμές που περιέχονται στις φράσεις, που εισάγονται από τον χρήστη, σύμφωνα με τις κατηγορίες φράσεων που αναφέρθηκαν πριν.

Οντότητα	Πεδίο	Τιμή
markers	action	retrieve insert delete

			profile
	target		box list profile
temporal	time		1 ... 24
	day		Monday ... Sunday
	month		January ... December
	year		<YYYY>
	before	time	1 ... 24
		day	Monday ... Sunday
		month	January ... December
		year	<YYYY>
	after	time	1 ... 24
		day	Monday ... Sunday
		month	January ... December
		year	<YYYY>
	time indicator		pm or am
	day indicator		weekly
properties	genre		<list of genres from the TVA Specification>
	title		string of arbitrary length
	keyword		string of arbitrary length
	creator		string of arbitrary length
	name		string of arbitrary length
	country		string of arbitrary length
	date period		no value
	language		string of arbitrary length
	dissemination date		no value
	dissemination location		string of arbitrary length
	dissemination source		string of arbitrary length
	type		audio

		visual
		textual
	theme	string of arbitrary length
	format	characters frames minutes seconds
	length	string of arbitrary length
	minlength	string of arbitrary length
	maxlength	string of arbitrary length

Σχήμα 8: Δομή ιδιοτήτων που παράγεται από τον αναλυτή διαγραμμάτων του συστήματος

Στο πεδίο οντότητες, παρατηρούμε τρία βασικά αντικείμενα με ονόματα markers, temporal και properties. Η οντότητα markers ορίζεται και παίρνει τιμές από τις εισαγωγικές φράσεις και τις φράσεις στόχου. Συγκεκριμένα, από τις εισαγωγικές φράσεις ανακτούμε πληροφορία για το action, δηλαδή ποια είναι η λειτουργία που θέλει να εκτελέσει ο χρήστης χρησιμοποιώντας το σύστημα, και από τις φράσεις στόχου, ανακτούμε πληροφορία για το target, δηλαδή για το που θέλουμε να λάβει χώρα η προηγούμενη λειτουργία. Έστω, λοιπόν, η εισαγωγική φράση:

-Are there any (Τύπος 1)

Από την φράση αυτή το πεδίο action της οντότητας markers παίρνει την τιμή retrieve. Επίσης, από τη φράση στόχου

-from my selection list (Τύπος 1)

το πεδίο target της οντότητας markers παίρνει την τιμή list.

Το αντικείμενο temporal παίρνει τιμές μέσω των χρονικών φράσεων. Για παράδειγμα, από την φράση:

-at 5 pm (Τύπος 2)

το πεδίο time της οντότητας temporal παίρνει την τιμή 5 και το πεδίο time indicator την τιμή pm.

Στην οντότητα *properties*, μέσα από τις φράσεις αναζήτησης και τις φράσεις περίληψης, παίρνουν τιμές τα διάφορα πεδία που αναγνωρίζονται στις φράσεις αυτές. Για παράδειγμα, η φράση αναζήτησης

- **comedy movies created in Spain** (Τύπος 7 + Τύπος 5)

δίνει στο πεδίο *genre* τις τιμές <comedy, movies> και στο πεδίο *country* την τιμή Spain, της οντότητας *properties*.

Σύμφωνα και με τα προαναφερθέντα, στην περίπτωση της μικρότερης δυνατής έκφρασης που θα οδηγήσει και στην μικρότερη δομή, όπου έχουμε μία εισαγωγική φράση και μία φράση στόχου, τελικώς, η δομή που θα παραχθεί θα περιέχει μόνο το αντικείμενο *markers* με συμπληρωμένα και τα δύο πιθανά πεδία *action* και *target*. Δηλαδή, η έκφραση

- **I want to see what is in my list**

αποτελείται από μία εισαγωγική φράση (I want to see what is) και μια φράση στόχου (in my list) και δίνει τιμές στα πεδία *action* και *target* της οντότητας *markers*, *retrieval* και *list* αντίστοιχα.

Φυσικά, η δομή μπορεί να περιέχει και τα τρία αντικείμενα με διάφορα πεδία συμπληρωμένα. Οι συνδυασμοί που μπορεί να προκύψουν, ανάλογα με την έκφραση του χρήστη είναι οι παρακάτω:

- **Τύπος 1: Markers**

με σχετικό παράδειγμα έκφρασης το αμέσως προηγούμενο

- **Τύπος 2: Markers – Properties**

με παράδειγμα έκφρασης

- **I would like you to show me movies starring Mel Gibson**

όπου έχουμε την τιμή *retrieval* στο πεδίο *action* της οντότητας *markers*, την τιμή *movies* στο πεδίο *genre*, την τιμή Mel Gibson στο πεδίο *name* και την τιμή *actor* στο πεδίο *creator* της οντότητας *properties*.

▪ **Τύπος 3: Markers – Properties – Temporal**

με παράδειγμα έκφρασης

- **Insert English spoken mystery movies broadcasted at midnight into my selection list**

όπου έχουμε την τιμή insert στο πεδίο action και την τιμή list στο πεδίο target της οντότητας markers, τις τιμές <mystery, movies> στο πεδίο genre, την τιμή English στο πεδίο language, ενεργοποιημένο το πεδίο dissemination date αλλά χωρίς τιμή, έτσι όπως το έχουμε ορίσει στη δομή της πληροφορίας, της οντότητας properties, καθώς επίσης την τιμή 24 στο πεδίο time και am στο πεδίο time indicator.

Όπως παρατηρούμε, το αντικείμενο Markers υπάρχει και στις τρεις περιπτώσεις και αυτό είναι λογικό, εφόσον το ένα από τα δύο πεδία του συμπληρώνεται μέσα από την εισαγωγική φράση, που απαρτίζει κατά κανόνα, κάθε πιθανή έκφραση. Επίσης, δε θα συναντήσουμε δομή αποτελούμενη από τα αντικείμενα Markers και Temporal μόνο, γιατί ο χρόνος δεν έχει νόημα σε εκφράσεις που αφορούν σε ανάκτηση πληροφορίας πάνω στα αποθηκευμένα προγράμματα του χρήστη και στην εξατομικευμένη λίστα των προγραμμάτων του, που ουσιαστικά οδηγούν σε δομές του πρώτου τύπου. Δηλαδή, δε μπορούμε να έχουμε μία έκφραση όπως το παράδειγμα που ακολουθεί

- **I would like to see after 5 pm**

Σε περίπτωση εισαγωγής, όμως σχετικής πρότασης από τον χρήστη, αυτός θα ενημερωθεί κατάλληλα για το λάθος του, όπως θα αναλύσουμε παρακάτω με χρήση αντίστοιχης υπομονάδας της αρχιτεκτονικής.

Στον πίνακα που περιγράψαμε τη δομή της πληροφορίας και συγκεκριμένα στο πεδίο Τιμή αναφέρουμε τις επιτρεπτές τιμές σε κάθε αντίστοιχο πεδίο. Σε κάποια πεδία επιτρέπεται οποιαδήποτε συμβολοσειρά, ενώ σε άλλες το σύνολο των λέξεων από τις οποίες συμπληρώνεται η τιμή είναι συγκεκριμένο, όπως για παράδειγμα στο πεδίο month της οντότητας temporal, που παίρνει τιμή από τους μήνες. Σε αυτή την τελευταία περίπτωση όταν ο χρήστης εισάγει την τιμή, υπάρχει ένας έλεγχος σε επίπεδο λεξικού και αν δεν ταιριάζει με τις προβλεπόμενες, ο χρήστης ενημερώνεται κατάλληλα, όπως θα δούμε και στο επόμενο κεφάλαιο, το κεφάλαιο της υλοποίησης.

4.5 Λειτουργικότητα

Σε αυτή την ενότητα θα δώσουμε αναλυτικά, την πλήρη λειτουργικότητα που προσφέρεται στον χρήστη. Ο χρήστης αρχικά μπορεί να δηλώσει επιθυμητό μέγιστο αριθμό αποτελεσμάτων που θα του επιστρέψει το σύστημα.

4.5.1 Σύνταξη φράσεων για παροχή λειτουργικότητας

Σε αυτό το σημείο παρουσιάζονται αναλυτικά οι τέσσερις βασικές λειτουργίες που επιτελεί το σύστημά μας, ξεκινώντας από την ανάκτηση εξατομικευμένου περιεχομένου μεταδεδομένων.

Αυτό επιτυγχάνεται με template έκφρασης του χρήστη που περιλαμβάνει τις παρακάτω επιμέρους φράσεις. Μετά το template κάθε έκφρασης έπεται παράδειγμα:

- **Introduction Phrase** (Τύπος 1) + * **Search Phrase**

‘Give me movies created in England’ ή

Introduction Phrase (Τύπος 1) + * **Search Phrase** + **Temporal Phrase**

‘Give me movies created in England broadcasted after 5 pm’

για ανάκτηση από τα μεταδεδομένα του server.

- **Introduction Phrase** (Τύπος 1) + * **Search Phrase** + **Target phrase** (Τύπος 1)

‘Give me movies created in England from my selection list’ ή

Introduction Phrase (Τύπος 1) + * **Search Phrase** + **Temporal Phrase** + **Target phrase** (Τύπος 1)

‘Give me movies created in England broadcasted after 5 pm from my selection list’

για ανάκτηση από την εξατομικευμένη λίστα μεταδεδομένων προγραμμάτων του χρήστη (selection list).

- **Introduction Phrase** (Τύπος 1) + * **Search Phrase** + **Target phrase** (Τύπος 2)

‘Give me movies created in England from my PDR’ ή

- **Introduction Phrase** (Τύπος 1) + * **Search Phrase** + **Temporal Phrase** + **Target phrase** (Τύπος 2)

‘Give me movies created in England broadcasted after 5 pm from my PDR’

για ανάκτηση από τα αποθηκευμένα προγράμματα του χρήστη, η πληροφορία για τα οποία βρίσκεται μέσα στην εξατομικευμένη λίστα μεταδεδομένων προγραμμάτων του χρήστη. Με τον αστερίσκο (*) θέλουμε να δηλώσουμε το πλήθος των εκφράσεων αναζήτησης, που μπορεί να είναι μία ή περισσότερες.

- **Introduction Phrase** (Τύπος 1) + **Target phrase** (Τύπος 1)

‘Show me the contents of my selection list’ ή

Introduction Phrase (Τύπος 1) + **Target phrase** (Τύπος 2)

‘Show me my stored contents’

για ανάκτηση όλης της λίστας των προγραμμάτων που περιέχονται στην εξατομικευμένη λίστα των μεταδεδομένων στην πρώτη περίπτωση και όλης της λίστας των αποθηκευμένων προγραμμάτων στην δεύτερη.

Η δεύτερη λειτουργία που προσφέρεται στον χρήστη είναι η διαχείριση εξατομικευμένου περιεχομένου μεταδεδομένων, δηλαδή ένα σύνολο από λειτουργίες πάνω στο selection list του.

Αυτό επιτυγχάνεται με template έκφρασης του χρήστη που περιλαμβάνει τις παρακάτω επιμέρους φράσεις:

- **Introduction Phrase** (Τύπος 2) + * **Search Phrase** ή
 ‘I would like to insert comedy movies’
- **Introduction Phrase** (Τύπος 2) + * **Search Phrase** + **Temporal Phrase**
 ‘I would like to insert comedy movies broadcasted tonight’ ή
- **Introduction Phrase** (Τύπος 2) + * **Search Phrase** + **Target phrase** (Τύπος 1)
 ‘Insert comedy movies into my selection list’ ή
Introduction Phrase (Τύπος 2) + * **Search Phrase** + **Temporal Phrase** + **Target phrase** (Τύπος 1)
 ‘Insert comedy movies broadcasted tonight into my selection list’
 για εισαγωγή προγραμμάτων στο selection list του χρήστη.
- **Introduction Phrase** (Τύπος 2) + * **Search Phrase**
 ‘I would like to record comedy movies’ ή
- **Introduction Phrase** (Τύπος 2) + * **Search Phrase** + **Temporal Phrase**
 ‘I would like to record comedy movies broadcasted tonight’ ή
- **Introduction Phrase** (Τύπος 2) + * **Search Phrase** + **Target phrase** (Τύπος 2)
 ‘I would like to insert comedy movies into my PDR’
ή
- **Introduction Phrase** (Τύπος 2) + * **Search Phrase** + **Temporal Phrase** + **Target phrase** (Τύπος 2)

‘I would like to insert comedy movies broadcasted tonight into my PDR’

Για εισαγωγή προγραμμάτων στα αποθηκευμένα του PDR.

▪ **Introduction Phrase (Τύπος 3) + * Search Phrase**

‘I would like to delete comedy movies’ ή

▪ **Introduction Phrase (Τύπος 3) + * Search Phrase + Temporal Phrase**

‘I would like to delete comedy movies broadcasted tonight’ ή

▪ **Introduction Phrase (Τύπος 3) + * Search Phrase + Target phrase (Τύπος 1)**

‘Delete comedy movies from my selection list’ ή

▪ **Introduction Phrase (Τύπος 3) + * Search Phrase + Temporal Phrase + Target phrase (Τύπος 1)**

‘Delete comedy movies broadcasted tonight from my selection list’

για διαγραφή προγραμμάτων από το selection list του χρήστη.

▪ **Introduction Phrase (Τύπος 3) + * Search Phrase + Target phrase (Τύπος 1)**

‘Delete my selection list’

για διαγραφή όλων των προγραμμάτων που βρίσκονται στο selection list του χρήστη.

▪ **Introduction Phrase (Τύπος 3) + * Search Phrase**

‘Delete comedy movies’ ή

▪ **Introduction Phrase (Τύπος 3) + * Search Phrase + Temporal Phrase**

‘Delete comedy movies broadcasted tonight’ ή

- **Introduction Phrase** (Τύπος 3) + * **Search Phrase** + **Target phrase** (Τύπος 2)

‘Delete comedy movies from my PDR’ ή

- **Introduction Phrase** (Τύπος 3) + * **Search Phrase** + **Temporal Phrase** + **Target phrase** (Τύπος 2)

‘Delete comedy movies broadcasted tonight from my PDR’

για διαγραφή προγραμμάτων από τα αποθηκευμένα προγράμματα του χρήστη.

- **Introduction Phrase** (Τύπος 3) + **Target phrase** (Τύπος 2)

‘Delete the contents of my PDR’

για διαγραφή όλων των αποθηκευμένων προγραμμάτων του χρήστη.

Η τρίτη λειτουργία που προσφέρεται στον χρήστη είναι η δημιουργία του προφίλ του. Αυτό επιτυγχάνεται με template έκφρασης του χρήστη που περιλαμβάνει τις παρακάτω επιμέρους φράσεις:

- **Introduction Phrase** (Τύπος 4) + * **Search Phrase**

‘I prefer English spoken movies created in Ireland’ ή

- **Introduction Phrase** (Τύπος 4) + * **Search Phrase** + **Temporal Phrase**

‘I prefer English spoken movies created in Ireland broadcasted every Monday’

ή

- **Introduction Phrase** (Τύπος 4) + * **Search Phrase** + **Target phrase** (Τύπος 3)

‘I want English spoken movies created in Ireland for my profile’ ή

- **Introduction Phrase** (Τύπος 4) + * **Search Phrase** + **Temporal Phrase** + **Target phrase** (Τύπος 3)

‘I prefer English spoken movies created in Ireland broadcasted every Monday for my profile’

Τελειώνοντας, έχουμε την τέταρτη κύρια λειτουργία, δηλαδή τη δημιουργία εξατομικευμένων περιλήψεων περιεχομένου μεταδεδομένων. Ο χρήστης μπορεί να ζητήσει από το σύστημα την δημιουργία περιλήψεων βασισμένων σε κριτήρια που ο ίδιος δηλώνει, καθώς και την ώρα παράδοσης της περίληψης αυτής. Αυτό επιτυγχάνεται με template έκφρασης του χρήστη που περιλαμβάνει τις παρακάτω επιμέρους φράσεις:

Introduction Phrase (Τύπος 1) + **Summary Phrase** + **Temporal Phrase**

‘I want a textual news summary of minimum 500 chars at 5 pm’

4.5.2 Ταξινόμηση πληροφορίας από το σύστημα

Οι τρεις πρώτες λειτουργίες που προσφέρονται στον χρήστη, δηλαδή η ανάκτηση εξατομικευμένης πληροφορίας περιεχομένου μεταδεδομένων, η διαχείριση εξατομικευμένου περιεχομένου και η δημιουργία προφίλ του χρήστη περιέχουν στο σώμα της έκφρασης που τις ορίζει, μία ή περισσότερες φράσεις αναζήτησης. Όταν θα γίνει ανάλυση από το σύστημα της εισαγωγής του χρήστη, αυτό που θα εισαχθεί στην βάση του συστήματος για να γίνει και η απαραίτητη εκτέλεση λειτουργίας και ανάκτηση αποτελεσμάτων, είναι ένα TVA XML Doc, σύμφωνα με το UserPreferences DS και συγκεκριμένα, θα περιέχει πληροφορία για το UserIdentifier και το FilteringAndSearchPreferences. Οι φράσεις, λοιπόν, αναζήτησης θα πρέπει να αντιστοιχίζουν τιμές με τις κατηγορίες που ορίζονται από το ClassificationPreferences, CreationPreferences και SourcePreferences. Σύμφωνα, λοιπόν, με αυτά που αναφέραμε στην παράγραφο 3.3 οι κατηγορίες που καλύπτονται σε αυτή τη φάση, από το μοντέλο είναι οι παρακάτω:

CreationPreferences	Title
	Creator
	Keyword
	DatePeriod
	Country
	Language

ClassificationPreferences	Country
	Language
	Genre
SourcePreferences	DisseminationDate
	DisseminationLocation
	DisseminationSource

Εξετάζοντας την περίπτωση της τέταρτης κύριας λειτουργίας, δηλαδή στον ορισμό προτιμήσεων του χρήστη για τα χαρακτηριστικά των περιλήψεων οπτικο-ακουστικού περιεχομένου καθίσταται ανάγκη περαιτέρω ανάλυσης του περιεχομένου του Summary Phrase. Ουσιαστικά, αυτό που στο τέλος συντάσσεται από το σύστημα είναι ένα TVA XML Doc, σύμφωνα με το UserPreferences DS και συγκεκριμένα, θα περιέχει πληροφορία για το UserIDentifier και το BrowsingPreferences, με τιμές στα πεδία του SummaryPreferences και του PreferenceCondition, για τον χρόνο παράδοσης. Ο χρήστης δεν είναι υποχρεωμένος να δηλώσει τον τύπο της περίληψης, textual, audio, visual, γιατί η προεπιλεγμένη τιμή είναι visual. Τα πεδία που μπορούν να πάρουν τιμές από τη δήλωση του χρήστη είναι τα παρακάτω, σύμφωνα με το SummaryPreference DS.

SummaryPreferences	SummaryType
	SummaryTheme
	SummaryDuration
	MinSummaryDuration
	MaxSummaryDuration
	NumOfKeyFrames
	MinNumOfKeyFrames
	MaxNumOfKeyFrames
	NumOfChars
	MinNumOfChars
	MaxNumOfChars

Σε περίπτωση που ο χρήστης κάνει κάποιο λάθος κατά την δήλωση της περίληψης, όπως να δηλώσει σαν τύπο textual και να ζητήσει αριθμό από frames το σύστημα αναλαμβάνει να τον ενημερώσει κατάλληλα, αλλιώς συντάσσεται το έγγραφο και εισάγεται στην βάση και επιστρέφεται κατάλληλο μήνυμα επιτυχίας στον χρήστη.

Στην περίπτωση που δεν έχουμε ασάφειες από την πλευρά του χρήστη, ανακτούνται τα προγράμματα και ανάλογα με τη λειτουργία του χρήστη:

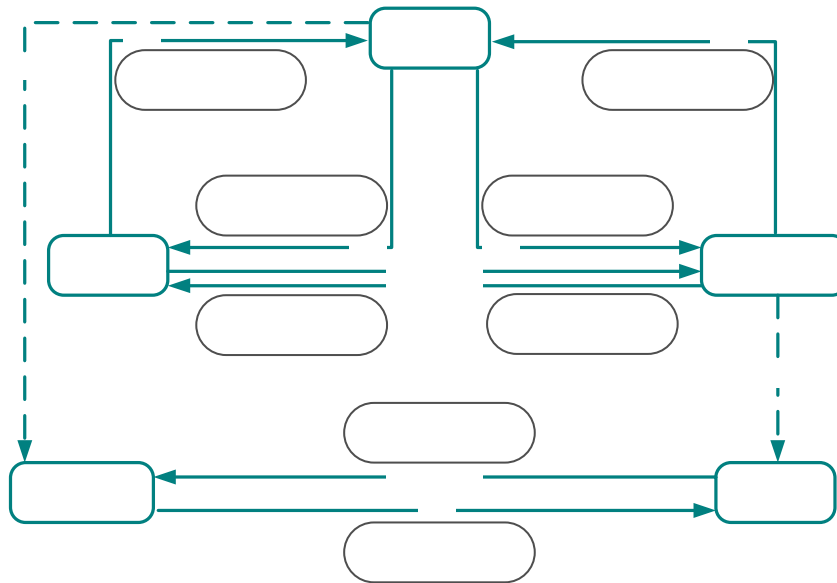
- Επιστρέφεται μία λίστα προγραμμάτων και κατάλληλο μήνυμα στον χρήστη.

- Ενημερώνεται ο χρήστης για τον αριθμό των προγραμμάτων που εισήχθησαν στο selection list του και με ποιο status έγινε αυτό.
- Ενημερώνεται ο χρήστης για την επιτυχή δημιουργία του προφίλ του.

Σε περίπτωση που δεν υπάρχουν προγράμματα με την συγκεκριμένη πληροφορία που ζήτησε ο χρήστης του επιστρέφεται κατάλληλο μήνυμα, ενώ σε περίπτωση που η τιμή που έχει εισάγει δε βρίσκει αντίκρισμα στην βάση στην κατηγορία που την έχει δηλώσει ο χρήστης, τότε παίρνει πάλι κατάλληλο μήνυμα.

Στις περιπτώσεις ανάκτησης και διαχείρισης εξατομικευμένου περιεχομένου το TVA XML Doc, που εισάγεται στη βάση του συστήματος, αυτή μετά το πέρας της διαχείρισης της πληροφορίας, εφόσον ο ρόλος του είναι προσωρινός και στόχο έχει την διεκπεραίωση της αίτησης του χρήστη για ανάκτηση πληροφορίας ή λειτουργίας πάνω στο selection list του. Θα πρέπει να τονίσουμε ότι στις δύο περιπτώσεις αυτές η εισαγωγή στην βάση του εγγράφου περιέχει προσωρινή πληροφορία για το Name του UserIdentifier του χρήστη, γιατί εάν περιείχε το όνομα του χρήστη που μπορεί να είχε δημιουργήσει ένα προφίλ με το ίδιο όνομα, τότε το σύστημα θα έσβηνε το προηγούμενο προφίλ και θα χανόταν πολύτιμη πληροφορία. Ο λόγος που συντάσσεται η πληροφορία κατά αυτόν τον τρόπο, είναι για την εκμετάλλευση της λειτουργικότητας του συστήματος για γρήγορο matching με τα μεταδεδομένα των προγραμμάτων.

Όταν ο χρήστης θέλει να εισάγει ή να διαγράψει από το selection list ή από τα αποθηκευμένα προγράμματα του PDR του κάποια προγράμματα ανάλογα της πληροφορίας που έχει εισάγει, όταν από την βάση γυρίσουν τα αποτελέσματα, δε χρειάζεται να γίνει κάποιου είδους ταξινόμηση, αλλά το σύστημα αναλαμβάνει και στις δύο περιπτώσεις να ενημερώσει την λίστα του χρήστη. Σχηματικά το σύνολο των ενεργειών που μπορούν να γίνουν πάνω στη λίστα φαίνονται παρακάτω:



Σχήμα 9: Η μηχανή καταστάσεων που χρησιμοποιείται για τη διαχείριση των καταστάσεων ενός προγράμματος μέσα στην εξατομικευμένη λίστα

Ο χρήστης, όμως δεν μπορεί να γνωρίζει χωρίς να δει την λίστα του, ποια προγράμματα περιέχονται και με ποιο status. Οπότε πάλι το σύστημα αναλαμβάνει να εκτελέσει τη σωστή λειτουργία για λογαριασμό του χρήστη. Δηλαδή ανάλογα με την πληροφορία που μπορούμε να συλλέξουμε από κάθε πιθανή έκφραση του χρήστη μπορούμε να διακρίνουμε τις παρακάτω κατηγορίες και τις λειτουργίες που αναλαμβάνει το σύστημα:

Εντολή	Στόχος	Ενέργεια
Insert	List	InsertIntoUserSelectionList με status Undefined
Insert	PDR	Αν υπήρχαν στο selection list UpdateUserSelectionList με status ToBeDeleted Αν δεν υπήρχαν InsertIntoUserSelectionList με status ToBeRecorded
Insert	-	InsertIntoUserSelectionList με status Undefined
Delete	List	DeleteFromUserSelectionList
Delete	PDR	UpdateUserSelectionList με status ToBeDeleted
Delete	-	Αν υπήρχαν στο selection list με status Recorded τότε UpdateUserSelectionList με status ToBeDeleted Αν υπήρχαν στο selection list με status οποιοδήποτε εκτός του Recorded τότε DeleteFromUserSelectionList
Store Record	-	InsertIntoUserSelectionList με status ToBeRecorded

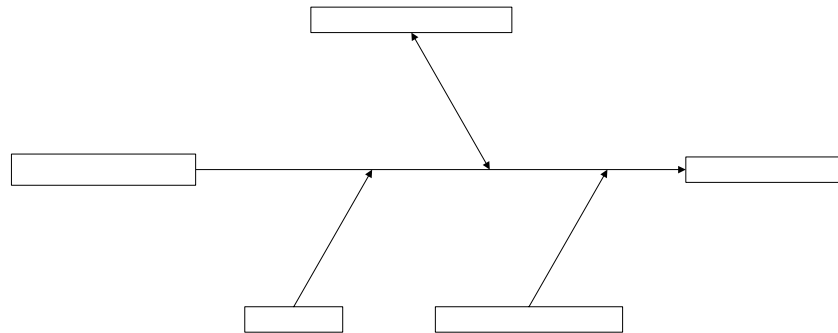
Στον χρήστη επιστρέφεται μήνυμα που περιέχει τον αριθμό των προγραμμάτων που πήραν μέρος στη λειτουργία που διέταξε μαζί με το μήνυμα επιτυχίας.

4.5.3 Αλγόριθμος για την Επίλυση Ασαφειών

Όπως τονίσαμε και προηγούμενα, το σύστημα προσπαθεί να λύσει τυχόν ασάφειες που προέρχονται από την εισαγωγή του χρήστη. Αυτό γίνεται με τους παρακάτω τρόπους. Όταν ο χρήστης εισάγει τιμή για κάποια από τις κατηγορίες του TVA, πρώτα γίνεται ένας έλεγχος σε μια καθολική βάση δεδομένων με υπάρχουσες οντολογίες, που περιέχουν πληροφορία για ψηφιακή τηλεόραση για να διαπιστωθεί το αν υπάρχει τιμή για το συγκεκριμένο πεδίο και αν υπάρχει συμπληρώνεται ορθά κατά την σύνταξη του εγγράφου. Στην περίπτωση εισαγωγής χρονικών ειφράσεων, περιέχονται μηχανισμοί από το σύστημα για μετατροπή τους σε MPEG-7 μορφή, συμβατής με το TVA Schema.

Σε περίπτωση που ο χρήστης εισάγει πληροφορία για την οποία δεν υπάρχουν στοιχεία σε ποια από τις TV-Anytime κατηγορίες αντιστοιχεί, τότε γίνονται τα εξής βήματα. Αρχικά, συλλέγεται το πλήθος της πληροφορίας, σαν ένα σύνολο από συμβολοσειρές (strings) που θα αντιμετωπιστούν ξεχωριστά. Έπειτα περνούν τα strings από μία λίστα λέξεων, που κόβει λέξεις που ταιριάζουν με τις δοσμένες και δεν αποτελούν σημαντική πληροφορία, όπως in, about, the και άλλες. Αν η λίστα δεν κόψει καμία λέξη από αυτές που εισήγαγε ο χρήστης, το κρατάμε σαν πληροφορία για να το εκμεταλλευτούμε στα αποτελέσματα, όπως θα εξηγήσουμε παρακάτω. Οι λέξεις and και or κόβονται από την λίστα αυτή, αλλά κρατιέται πληροφορία για την παρουσία τους για να χρησιμοποιηθεί αργότερα στην ταξινόμηση των αποτελεσμάτων. Έπειτα, το σύνολο πάλι των strings που απέμειναν, φιλτράρονται από την βάση του συστήματος, η οποία γυρνά ένα TVA XML Doc συμβατό με το FilteringAndSearchPreferences DS, από όπου δίνεται η σημασιολογία των τιμών. Τέλος, το σύνολο των strings, φιλτράρονται από το προφίλ του χρήστη, αν αυτό υπάρχει για να δώσουμε καλύτερα ταξινομημένα αποτελέσματα στον χρήστη. Ο λόγος που η ταξινόμηση των αποτελεσμάτων είναι τόσο σημαντική, είναι επειδή ο χρήστης μπορεί και δηλώνει τον αριθμό των αποτελεσμάτων και θέλουμε τα πρώτα αποτελέσματα να περιέχουν την πληροφορία που

αναζητούσε ο χρήστης. Συνοψίζοντας, μπορούμε να παραστήσουμε την πορεία που ακολουθείται με το παρακάτω σχήμα:



Τώρα, συντάσσεται το έγγραφο με τιμές στις κατηγορίες που βρέθηκαν από το φιλτράρισμα που έγινε στη βάση του συστήματος. Αφού συνταχθεί το TVA XML Doc, εισάγεται στην βάση δεδομένων και ξεκινά η διαδικασία για το ταίριασμα της πληροφορίας που περιέχει, με την πληροφορία των μεταδεδομένων των προγραμμάτων.

Όπως είδαμε και στην αμέσως παραπάνω παράγραφο, στην περίπτωση που ο χρήστης εισάγει πληροφορία για την οποία δεν έχουμε στοιχεία για την αντιστοίχισή της με τις TVA κατηγορίες, όταν αυτές περνούν από την λίστα των λέξεων που κόβουν τις ασήμαντες, κρατιέται η πληροφορία για το αν υπήρχε η λέξη and ή η λέξη or μέσα σε αυτές. Υποχρέωση του συστήματος είναι να επιστρέψει στον χρήστη όλα τα πιθανά αποτελέσματα. Εκτός, λοιπόν από το προσωρινό έγγραφο που συντάσσεται και εισάγεται στην βάση, εισάγεται προσωρινά και το έγγραφο που συντάχθηκε όταν η ασαφής πληροφορία του χρήστη φιλτραρίστηκε από το προφίλ του χρήστη. Έτσι, επιστρέφονται και από αυτή την περίπτωση αποτελέσματα πάνω σε προγράμματα. Στην προσπάθεια να γίνουμε συγκεκριμένοι και ακριβείς, μπορούμε να εκμεταλλευτούμε τον τρόπο με τον οποίο τα αποτελέσματα εμφανίζονται στον χρήστη, και έτσι κάνουμε χρήση κάποιων κανόνων για την ταξινόμησή τους, ανάλογα με την εισαγωγή του χρήστη, και τις διακρίνουμε στις εξής παρακάτω:

- Αν η Stop List δε κόψει καθόλου λέξεις κλειδιά
- Αν η βάση δεδομένων γυρίσει αποτελέσματα και το προφίλ του χρήστη βρει τιμές που έχουν δηλωθεί και στις προτιμήσεις του χρήστη τότε
 - ✓ Τα αποτελέσματα βασίζονται σε όλη τη σειρά των λέξεων που εισήγαγε ο χρήστης και ταξινομούνται πρώτα με τα αποτελέσματα στα προγράμματα που

Vector of Keywords

προέρχονται από το προσωρινό μερικό προφίλ του χρήστη και έπειτα από τα αποτελέσματα της βάσης.

- Αν μόνο η βάση γυρίσει αποτελέσματα τότε
 - ✓ Τα αποτελέσματα βασίζονται σε όλη τη σειρά των λέξεων που εισήγαγε ο χρήστης.

Παράδειγμα για την πρώτη περίπτωση

Έστω, ότι έχουμε τη φράση: **‘...with Nicole Kidman’**

και έστω ότι από την ανάκτηση της σημασιολογίας μέσα από τη βάση του συστήματος έχουμε τα παρακάτω αποτελέσματα σε ζεύγη ονόματος και ρόλου

Creator	Role
Nicole Rosselle	actor
Mary Kidman	director
Nicole Kidman	producer
Nicole Kidman	actor

Έπειτα από το φιλτράρισμα που γίνεται μέσα από το προφίλ του χρήστη έχουμε τα παρακάτω αποτελέσματα με ένα βάρος που υποδηλώνει τον βαθμό προτίμησής του.

Creator	Role	PVal
Nicole Rosselle	actor	80
Nicole Kidman	producer	50
Nicole Kidman	actor	70

Τα αποτελέσματα που θα επιστραφούν στο χρήστη βασίζονται σύμφωνα με τα παραπάνω στην εξής ταξινόμηση:

	Creator	Role
1	Nicole Kidman	actor
2	Nicole Kidman	producer
3	Nicole Rosselle	director
4	Mary Kidman	actor

- Αν η Stop List κόψει κάποια από τις λέξεις κλειδί “and” ή “or”

- Αν η βάση γυρίσει αποτελέσματα και το προφίλ του χρήστη βρει τιμές που έχουν δηλωθεί και στις προτιμήσεις του χρήστη τότε
 - ✓ Τα αποτελέσματα βασίζονται στην TVA κατηγορία που ανήκει η κάθε τιμή και ταξινομούνται πρώτα με τα αποτελέσματα που οι κατηγορίες είναι ίδιες για όλες τις τιμές.
- Αν μόνο η βάση γυρίσει αποτελέσματα τότε
 - ✓ Τα αποτελέσματα βασίζονται στην TVA κατηγορία που ανήκει η κάθε τιμή και ταξινομούνται πρώτα με τα αποτελέσματα που οι κατηγορίες είναι ίδιες για όλες τις τιμές.

Παράδειγμα για την πρώτη περίπτωση

Έστω, ότι έχουμε τη φράση: **‘...with Frank and English’**

και έστω ότι από την ανάκτηση της σημασιολογίας μέσα από τη βάση του συστήματος έχουμε τα παρακάτω αποτελέσματα σε ζεύγη ονόματος και ρόλου

Creator	Role
Frank English	actor
Holger Franke	director
Frank... 66 results	various
Language	english

Έπειτα από το φιλτράρισμα που γίνεται μέσα από το προφίλ του χρήστη έχουμε τα παρακάτω αποτελέσματα με ένα βάρος που υποδηλώνει τον βαθμό προτίμησής του.

Language	PVal
english	80

Τα αποτελέσματα που θα επιστραφούν στο χρήστη βασίζονται σύμφωνα με τα παραπάνω στην εξής ταξινόμηση:

	Creator	Role
1	Frank / English	actor/actor
2	...	actor/actor
3	Franke / English	director/actor

- Αν η StopList κόψει λέξεις, αλλά όχι την “and” ή την “or”
- Αν η βάση γυρίσει αποτελέσματα και το προφίλ του χρήστη βρει τιμές που έχουν δηλωθεί και στις προτιμήσεις του χρήστη τότε
 - ✓ Τα αποτελέσματα βασίζονται στην TVA κατηγορία που ανήκει η κάθε τιμή και ταξινομούνται με τα αποτελέσματα στα προγράμματα που προέρχονται από το προσωρινό μερικό προφίλ του χρήστη και έπειτα με τα αποτελέσματα που οι κατηγορίες είναι ίδιες για όλες τις τιμές .
- Αν μόνο η βάση γυρίσει αποτελέσματα τότε
 - ✓ Τα αποτελέσματα βασίζονται στην TVA κατηγορία που ανήκει η κάθε τιμή και ταξινομούνται πρώτα με τα αποτελέσματα που οι κατηγορίες είναι ίδιες για όλες τις τιμές.

Παράδειγμα για την πρώτη περίπτωση

Έστω, ότι έχουμε τη φράση: **‘...with Frank in English’**

και έστω ότι από την ανάκτηση της σημασιολογίας μέσα από τη βάση του συστήματος έχουμε τα παρακάτω αποτελέσματα σε ζεύγη ονόματος και ρόλου

Creator	Role
Frank English	actor
Holger Franke	director
Frank... 66 results	various
Language	english

Έπειτα από το φιλτράρισμα που γίνεται μέσα από το προφίλ του χρήστη έχουμε τα παρακάτω αποτελέσματα με ένα βάρος που υποδηλώνει τον βαθμό προτίμησής του.

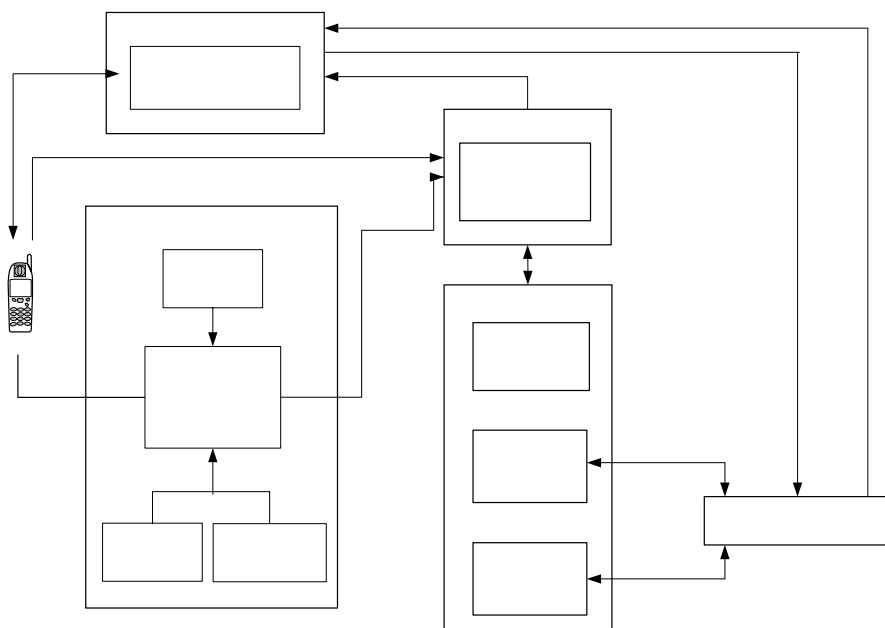
Language	PVal
english	80

Τα αποτελέσματα που θα επιστραφούν στο χρήστη βασίζονται σύμφωνα με τα παραπάνω στην εξής ταξινόμηση:

	Creator	Role
1	Frank / English	actor / language
2	...	actor/actor
3	Franke / English	director/actor

4.6 Θεμελιώδης σχεδιασμός της αρχιτεκτονικής του μοντέλου Φυσικής Γλώσσας

Ο καθορισμός των υπο-μονάδων του συστήματος απαιτεί και τον απόλυτο καθορισμό των αρμοδιοτήτων της καθεμίας και της επικοινωνίας μεταξύ αυτών. Οι υπο-μονάδες που σχεδιάστηκαν για το μοντέλο της φυσικής γλώσσας παρουσιάζονται στο παρακάτω σχήμα:



Σχήμα 14: Θεμελιώδης σχεδιασμός της αρχιτεκτονικής του μοντέλου

Η έκφραση που εισάγει ο χρήστης αποτελεί την είσοδο στον αναλυτή διαγραμμάτων (**Chart Parser**). Ο αναλυτής διαγραμμάτων χρησιμοποιείται για να παράγει σαν έξοδο τη δομή ιδιοτήτων που παρουσιάσαμε παραπάνω, την οποία θα πάρει για να επεξεργαστεί το σύστημα, και κάνει χρήση ενός δεδομένου λεξικού και μίας γραμματικής.

Ο διαχειριστής διαλόγου (**Dialogue Manager**) παίρνει σαν είσοδο τη δομή ιδιοτήτων από τον αναλυτή διαγραμμάτων και τα προσωπικά στοιχεία του χρήστη από την εισαγωγή του στο σύστημα, τη διαμορφώνει κατάλληλα, ώστε να μπορούν να τη διαχειριστούν ανάλογα όλες οι υπο-μονάδες του συστήματος και χρησιμοποιώντας την υπο-μονάδα, που είναι υπεύθυνη για την επίλυση ασαφειών (**Ambiguities Resolver**), επικοινωνεί με τον **TVA server**. Ο **TVA server** ενημερώνεται με πληροφορία που ανακτάται από μία σχεσιακή βάση δεδομένων με TV-Anytime πληροφορία και με οντολογίες καθολικής πληροφορίας σε περιεχόμενο ψηφιακής τηλεόρασης, όπως πίνακες με όλα τα υπάρχοντα κανάλια ή όλους τους συντελεστές τηλεοπτικών προγραμμάτων.

Η υπο-μονάδα **Ambiguities Resolver** περιέχει τρεις διαδικασίες οι οποίες αναλαμβάνουν η καθεμία να επιλύσουν ασάφειες ανάλογα της εισαγωγής του χρήστη. Ο **Date/Time Resolver** παίρνει ως είσοδο τη δομημένη πληροφορία χρονικών φράσεων όπως έρχεται από τη δομή ιδιοτήτων και τροφοδοτεί τον διαχειριστή διαλόγου με την πληροφορία αυτή διαμορφωμένη κατάλληλα (με σύνταξη όπως ορίζει το πρότυπο TV-Anytime). Ο **TVA Semantics Resolver** αναλαμβάνει να αποδώσει σημασιολογία σε ένα σύνολο από συμβολοσειρές για τις οποίες δεν υπήρξε αντιστοίχιση με κάποια TV-Anytime κατηγορία και επιστρέφει τις συμβολοσειρές και τις κατηγορίες στις οποίες βρέθηκε αντιστοίχιση για κάθε μία από αυτές. Ο **User's Profile Resolver** παίρνει ως είσοδο από τον διαχειριστή διαλόγου το ίδιο σύνολο από συμβολοσειρές που παρουσιάζουν ασάφεια, φιλτράρει τα δεδομένα της έκφρασης μέσα από τις τυχόν υπάρχουσες προτιμήσεις του χρήστη, δηλαδή το προφίλ του και επιστρέφει ένα TVA XML έγγραφο, συμβατό με το UserPreferences DS που αναλύσαμε στο προηγούμενο κεφάλαιο. Το έγγραφο αυτό περιέχει ως στοιχεία μόνο εκείνα στα οποία βρέθηκε τιμή ίδια με κάποια συμβολοσειρά από αυτές που εισήχθησαν στον **User's Profile Resolver**. Ο στόχος αυτού του εγγράφου είναι να βοηθήσει στην ταξινόμηση των αποτελεσμάτων.

Ο **TVA Server**, στον οποίο θα αναφερθούμε διεξοδικά στο επόμενο κεφάλαιο, επιστρέφει τα αποτελέσματα στην υπο-μονάδα **Response Manager**, που αναλαμβάνει τη μορφοποίηση των αποτελεσμάτων έτσι ώστε να μπορούν να παρουσιαστούν στον χρήστη. Επίσης, προσαρμόζει και ένα πρότυπο μηνύματος, που σε συνδυασμό με πληροφορία που του προσφέρει η υπο-μονάδα **Dialogue Manager**, το παρέχει μαζί με το σετ των αποτελεσμάτων όταν αυτό υπάρχει.

4.7 USE CASES

Στο σημείο αυτό θα αναφερθούμε στα βασικά χαρακτηριστικά της λειτουργικότητας του συστήματος, κάνοντας χρήση των Use Cases [13]. Χρησιμοποιώντας το μοντέλο των Use Cases έχουμε έναν τρόπο «εύκολης» παρουσίασης όλων των βασικών χαρακτηριστικών και βημάτων μίας διαδικασίας σε μορφή πινάκων. Δείχνουν τις ιδιότητες της συγκεκριμένης χρήσης, όλα τα βήματα ορθής αποπεράτωσής της, αλλά και τον λανθασμένο τερματισμό της.

Αποτελούν το πρώτο βήμα κατά τον σχεδιασμό οποιασδήποτε εφαρμογής και κατόπιν, χρησιμοποιούνται κατά την υλοποίησή της σαν οδηγός, ώστε το τελικό σύστημα να μπορεί να αντεπεξέλθει στις προϋποθέσεις χρήσης του και να φέρει τα επιθυμητά αποτελέσματα.

Στη συνέχεια, παρουσιάζουμε τα τέσσερα βασικά Use Cases που εξυπηρετεί το μοντέλο της φυσικής γλώσσας στο περιβάλλον της ψηφιακής τηλεόρασης της εφαρμογής μας που εξηγούν αναλυτικά αλλά και με κατανοητό τρόπο την λειτουργικότητα που απαιτήθηκε και τελικά υποστηρίζει η εφαρμογή μας.

Use Case 1: Ανάκτηση εξατομικευμένης πληροφορίας περιεχομένου μεταδεδομένων

USE CASE 1	Ανάκτηση εξατομικευμένης πληροφορίας περιεχομένου μεταδεδομένων	
Goal In Context	Ο χρήστης ζητά πληροφορία που αφορά μεταδεδομένα προγραμμάτων	
Scope & Level	Σύστημα , Primary Task	
Precondition		
Success End Condition	Ο χρήστης λαμβάνει τις σωστές πληροφορίες για τα μεταδεδομένα προγραμμάτων που ταιριάζουν στις παραμέτρους που εισήγαγε στο σύστημα.	
Failed End Condition	Ο χρήστης δε λαμβάνει καμία πληροφορία από το σύστημα (ο χρήστης δεν θέλει ή system crash)	
Primary, Secondary Actor	Χρήστης	
Trigger	Ο χρήστης ξεκινάει την εφαρμογή	
Description	Step	Action
	1	Ο χρήστης δίνει κριτήρια για το επιθυμητό περιεχόμενο των προγραμμάτων, μέσω έκφρασης που εισάγεται στον αναλυτή διαγραμμάτων

		(Chart Parser).
	2	Ο αναλυτής διαγραμμάτων παράγει την ή τις δομές ιδιοτήτων που εισάγονται στον διαχειριστή διαλόγου (Dialogue Manager).
	3	Ο διαχειριστής διαλόγου αλληλεπιδρά με τον επιλυτή ασαφειών για πλήρη αποσαφήνιση της πληροφορίας.
	4	Ο διαχειριστής διαλόγου εισάγει την πληροφορία στον διαχειριστή αποκρίσεων.
		Ο διαχειριστής αποκρίσεων εισάγει την πληροφορία για τα κριτήρια πάνω στο περιεχόμενο των προγραμμάτων στον TVA Server.
	5	Ο TVA Server επιστρέφει τα αποτελέσματα στον διαχειριστή αποκρίσεων.
	6	Ο διαχειριστής αποκρίσεων διαμορφώνει τα αποτελέσματα κατάλληλα, βάση και της πληροφορίας που παίρνει από τον διαχειριστή διαλόγου και παρουσιάζει τα αποτελέσματα στον χρήστη.
Extensions	Step	Branching Action
	3α	Δεν υπάρχει κανένα πρόγραμμα βάση των κριτηρίων του χρήστη. 3α1. Ο χρήστης ενημερώνεται και καλείται να δώσει νέα στοιχεία προς αναζήτηση.
	3β	Δεν υπάρχουν μεταδεδομένα για το επιλεχθέν πρόγραμμα. Ο χρήστης ενημερώνεται 3β1. Ενημερώνεται ο χρήστης.

Use Case 2: Διαχείριση εξατομικευμένου περιεχομένου μεταδεδομένων

USE CASE 2	Διαχείριση εξατομικευμένου περιεχομένου μεταδεδομένων
Goal In Context	Ο χρήστης ζητά να διαχειριστεί την εξατομικευμένη λίστα του που περιέχει προσωπικές επιλογές και επιλογές που προτείνονται από το σύστημα, βάση του προφίλ του
Scope & Level	Σύστημα , Primary Task
Precondition	Ο χρήστης έχει εξατομικευμένη λίστα προγραμμάτων
Success End Condition	Η ενέργεια του χρήστη πάνω στη εξατομικευμένη λίστα των προγραμμάτων λαμβάνει χώρα ορθώς.

Failed End Condition	Η ενέργεια του χρήστη αποτυγχάνει(ο χρήστης δεν θέλει ή system crash).	
Primary, Secondary Actor	Χρήστης	
Trigger	Ο χρήστης ξεκινάει την εφαρμογή.	
Description	Step	Action
	1	Ο χρήστης εισάγει στον αναλυτή διαγραμμάτων έκφραση με την ενέργεια (insert, delete) πάνω στο selection list και τα κριτήρια για το περιεχόμενο των προγραμμάτων.
	2	Ο αναλυτής διαγραμμάτων παράγει την ή τις δομές ιδιοτήτων που εισάγονται στον διαχειριστή διαλόγου (Dialogue Manager).
	3	Ο διαχειριστής διαλόγου αλληλεπιδρά με τον επιλυτή ασαφειών για πλήρη αποσαφήνιση της πληροφορίας, όσον αφορά στα κριτήρια του περιεχομένου.
	4	Ο διαχειριστής διαλόγου εισάγει την πληροφορία στον διαχειριστή αποκρίσεων.
	5	Ο διαχειριστής αποκρίσεων εισάγει την πληροφορία στον TVA Server για το περιεχόμενο των προγραμμάτων και εισάγει στον διαχειριστή αποκρίσεων την ενέργεια για το selection list.
	5	Ο TVA Server επιστρέφει τα προγράμματα που ταιριάζουν στα κριτήρια του χρήστη στον διαχειριστή αποκρίσεων.
	6	Ο διαχειριστής αποκρίσεων εκτελεί τη λειτουργία στο selection list με τα αποτελέσματα του Server.
	7	Ο διαχειριστής αποκρίσεων επιστρέφει στον χρήστη κατάλληλο μήνυμα για την επιτυχία της λειτουργίας.
Extensions	Step	Branching Action
	5α	Δεν υπάρχουν προγράμματα στα μεταδεδομένα του server που να ταιριάζουν στα κριτήρια του χρήστη και η ενέργεια για εισαγωγή προγραμμάτων στη λίστα του δε πραγματοποιείται. 5α1. Ο χρήστης ενημερώνεται και καλείται να δώσει νέα κριτήρια.
	5β	Δεν υπάρχουν προγράμματα στη λίστα του χρήστη που να ταιριάζουν στα κριτήρια

		του και η ενέργεια για διαγραφή δε πραγματοποιείται. 5β1. Ενημερώνεται ο χρήστης.
Sub-variations		Branching Action
	1α	Ο χρήστης μπορεί να δώσει ως στόχο της λειτουργίας το PDR του το προσωπικό του selection list

***Use Case 3:** Δημιουργία εξατομικευμένων περιλήψεων περιεχομένου μεταδεδομένων*

USE CASE 3	Δημιουργία εξατομικευμένων περιλήψεων περιεχομένου μεταδεδομένων.	
Goal In Context	Ο χρήστης δίνει κριτήρια και πληροφορία για τη δημιουργία περιλήψεων πάνω σε μεταδεδομένα προγραμμάτων.	
Scope & Level	Σύστημα , Primary Task	
Precondition		
Success End Condition	Το σύστημα λαμβάνει την πληροφορία από τον χρήστη και την αποθηκεύει στην βάση.	
Failed End Condition	Η ενέργεια του χρήστη αποτυγχάνει(ο χρήστης δεν θέλει ή system crash).	
Primary, Secondary Actor	Χρήστης	
Trigger	Ο χρήστης ξεκινάει την εφαρμογή.	
Description	Step	Action
	1	Ο χρήστης δίνει τις προτιμήσεις του για τα χαρακτηριστικά περίληψης οπτικο-ακουστικού περιεχομένου, μέσω έκφρασης που εισάγεται στον αναλυτή διαγραμμάτων.
	2	Ο αναλυτής διαγραμμάτων παράγει την ή τις δομές ιδιοτήτων που εισάγονται στον διαχειριστή διαλόγου.
	3	Ο διαχειριστής διαλόγου αλληλεπιδρά με τον επιλυτή ασαφειών χρονικών εκφράσεων για πλήρη αποσαφήνιση της πληροφορίας.
	4	Ο διαχειριστής διαλόγου εισάγει την πληροφορία στον διαχειριστή αποκρίσεων.
	5	Ο διαχειριστής αποκρίσεων εισάγει την πληροφορία στον TVA Server και επιστρέφει κατάλληλο μήνυμα επιτυχούς έκβασης της λειτουργίας στον χρήστη.

Extensions	Step	Branching Action
	1α	Ο χρήστης δίνει αταίριαστα κριτήρια. 1α1. Ο χρήστης ενημερώνεται και καλείται να δώσει νέα κριτήρια.

Use Case 4: Δημιουργία προφίλ χρήστη.

USE CASE 3	Δημιουργία προφίλ χρήστη	
Goal In Context	Ο χρήστης δίνει κριτήρια και πληροφορία για τη δημιουργία του προσωπικού του προφίλ.	
Scope & Level	Σύστημα , Primary Task	
Precondition		
Success End Condition	Το σύστημα λαμβάνει την πληροφορία από τον χρήστη και την αποθηκεύει στην βάση.	
Failed End Condition	Η ενέργεια του χρήστη αποτυγχάνει(ο χρήστης δεν θέλει ή system crash).	
Primary, Secondary Actor	Χρήστης	
Trigger	Ο χρήστης ξεκινάει την εφαρμογή.	
Description	Step	Action
	1	Ο χρήστης δίνει κριτήρια για το επιθυμητό περιεχόμενο των προγραμμάτων που ταιριάζουν στις προτιμήσεις του, μέσω έκφρασης που εισάγεται στον αναλυτή διαγραμμάτων.
	2	Ο αναλυτής διαγραμμάτων παράγει την ή τις δομές ιδιοτήτων που εισάγονται στον διαχειριστή διαλόγου.
	3	Ο διαχειριστής διαλόγου αλληλεπιδρά με τον επιλυτή ασαφειών για πλήρη αποσαφήνιση της πληροφορίας.
	4	Ο διαχειριστής διαλόγου εισάγει την πληροφορία στον διαχειριστή αποκρίσεων.
	5	Ο διαχειριστής αποκρίσεων εισάγει την πληροφορία στον TVA Server και επιστρέφει κατάλληλο μήνυμα επιτυχούς έκβασης της λειτουργίας στον χρήστη.
Extensions	Step	Branching Action
	1α	Ο χρήστης δίνει αταίριαστα κριτήρια. 1α1. Ο χρήστης ενημερώνεται και καλείται να δώσει νέα κριτήρια.

Sub-variations	1α	Δημιουργία μεταδεδομένων.
-----------------------	-----------	---------------------------

Ανακεφαλαίωση

Στο παρόν κεφάλαιο, μελετήθηκε διεξοδικά το μοντέλο της φυσικής γλώσσας ξεκινώντας από τις προδιαγραφές των απαιτήσεων σε ένα περιβάλλον ψηφιακής τηλεόρασης. Ακολούθησε η ανάπτυξη του κύριου σώματος εργασίας της φυσικής γλώσσας, που αφορά στη σύνταξη των φράσεων και οδήγησε στην παρουσίαση του μεταμοντέλου και την απεικόνιση της πληροφορίας, όπως αυτή εξάγεται στη δομή ιδιοτήτων από τον αναλυτή διαγραμμάτων. Στην συνέχεια, έγινε παρουσίαση της λειτουργικότητας που παρέχει το σύστημα, που αφορά στο συνδυασμό των φράσεων με τελικό στόχο την ικανοποίηση των απαιτήσεων, στην ταξινόμηση της πληροφορίας βάση του προτύπου TV-Anytime που παρουσιάστηκε στο κεφάλαιο 3 και ανάλυση της στρατηγικής και του βασικού αλγορίθμου που ακολουθείται στην περίπτωση ασαφούς πληροφορίας από το χρήστη, όπου δε μπορούμε να έχουμε ταξινόμηση σε κάποια από τις TV-Anytime κατηγορίες. Στη συνέχεια παρουσιάστηκε ο θεμελιώδης σχεδιασμός της αρχιτεκτονικής του συστήματος που αναπτύχθηκε στα πλαίσια της εργασίας αυτής για την υποστήριξη του μοντέλου της φυσικής γλώσσας, ενώ ακολούθησαν τα Use Cases μέσα από τα οποία παρουσιάστηκε η λειτουργικότητα, σε συνδυασμό με τις υπο-μονάδες της αρχιτεκτονικής. Στο επόμενο κεφάλαιο, θα αναπτυχθεί η υλοποίηση του μοντέλου της φυσικής γλώσσας.

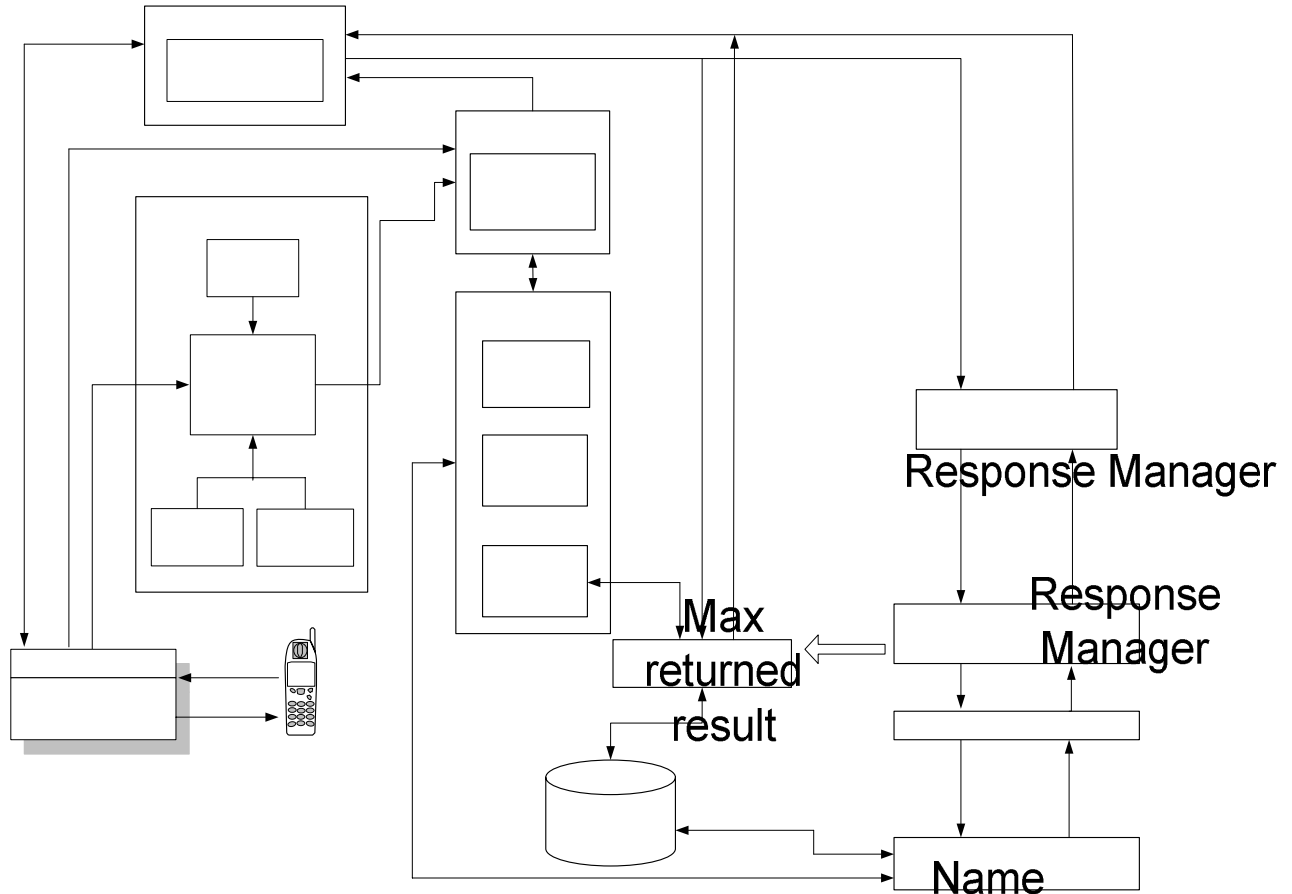
Κεφάλαιο 5

Υλοποίηση μοντέλου φυσικής γλώσσας

Στο κεφάλαιο αυτό ασχολούμαστε διεξοδικά με την επεξήγηση της υλοποίησης της αρχιτεκτονικής του μοντέλου της φυσικής γλώσσας, μέσα από την ανάλυση των υπο-μονάδων της και του τρόπου που αυτές επικοινωνούν μεταξύ τους. Στο τέλος του κεφαλαίου παρατίθενται δύο παραδείγματα χρήσης του συστήματος, στα οποία φαίνεται η ροή της πληροφορίας και πως αυτή σχηματίζεται κατάλληλα για να χρησιμοποιηθεί από τις εκάστοτε υπο-μονάδες του συστήματος.

5.1 Γενική Αρχιτεκτονική

Όπως είχαμε δείξει και στην παράγραφο 4.6, στο θεμελιώδη σχεδιασμό της αρχιτεκτονικής του μοντέλου της φυσικής γλώσσας, καταλήγουμε στην παρακάτω:



Η αρχιτεκτονική αυτή ακολουθεί μία πολυεπίπεδη (multi-tier) προσέγγιση και έχει τρία διαφορετικά επίπεδα. Το κατώτερο επίπεδο ασχολείται με τη διαχείριση των μεταδεδομένων. Το μεσαίο επίπεδο περιέχει όλη τη λογική για την επικοινωνία του συστήματος με τον έξω κόσμο. Το επίπεδο της εφαρμογής ή της χρήσης ενεργοποιεί την συναλλαγή ανάμεσα στον server και στους ετερογενείς clients, μέσω διαφορετικών διόδων.

Για την υλοποίηση της αρχιτεκτονικής του πρώτου επιπέδου, δηλαδή της επικοινωνίας του χρήστη, μέσω κινητής συσκευής, με το υπόλοιπο σύστημα, θα μιλήσουμε αναλυτικά στο επόμενο, 7^ο κεφάλαιο, Σχεδιασμός και Υλοποίησης εφαρμογής.

Η σχεσιακή βάση δεδομένων του συστήματος περιέχει όλη την TVAM πληροφορία, καθώς και οντολογίες, που περιέχουν πληροφορία σχετική με την ψηφιακή τηλεόραση, όπως καθολικούς πίνακες με όλους τους συντελεστές, όλα τα τηλεοπτικά κανάλια κ.α. Το RDBMS διαχειρίζεται την προσπέλαση των συνδιαλλαγών στην βάση. Επίσης χρησιμοποιείται ένα Java API που υλοποιήθηκε για την εξαγωγή όλης της διαθέσιμης λειτουργικότητας για τις υπηρεσίες φιλτραρίσματος, ανάκτησης και περιλήψεων και συνεργάζεται με το XML-DB middleware. Το XML-DB middleware είναι ένα σύνολο από από στοιχεία λογισμικού

(software components) που είναι υπεύθυνα για την διαχείριση των TV-Anytime XML εγγράφων και την αντιστοίχιση του TVAM XML σχήματος με το υποκείμενο σχεσιακό σχήμα.

Στη συνέχεια θα περιγράψουμε την υλοποίηση του πρωτοτύπου της φυσικής γλώσσας με την περιγραφή κάθε υπο-μονάδας. Θα πρέπει να επισημάνουμε ότι η υλοποίηση όλου του μοντέλου φυσικής γλώσσας, εκτός από τα λεξικά και την γραμματική που είναι αρχεία τύπου txt, η επικοινωνία με τον TVA server και η εφαρμογή μέσω κινητών συσκευών έγινε με χρήση της Java γλώσσας [10], [14].

5.2 Η Υπομονάδα Ανάλυσης Διαγραμμάτων (Chartparser Module)

Η υπο-μονάδα ανάλυσης διαγραμμάτων περιέχει τον αναλυτή JavaChart, για τον οποίο μιλήσαμε σε εισαγωγικό κεφάλαιο, το stem lexicon και το mini lexicon και την γραμματική, και είναι υπεύθυνη για τη δημιουργία μίας δομής που θα απεικονίζει την εισαγωγή του χρήστη στο σύστημα.

5.2.2 Λεξικό

Το λεξικό που ακολουθεί την σύνταξη του JavaChart, όπως αναφέραμε και νωρίτερα, αποτελείται από δύο επιμέρους λεξικά. Το stem lexicon, περιέχει τις ρίζες των λέξεων και λέξεις που βοηθούν στο να αποδοθούν σωστά οι τιμές στις κατηγορίες του TVA. Όταν εισάγουμε μία λέξη στο stem lexicon ορίζουμε σε ποια κατηγορία ανήκει όπως στο παρακάτω παράδειγμα το prn (profile nouns), ούτως ώστε σε γραμματικό κανόνα να αντιστοιχίζεται με την κατηγορία αυτή. Θα πρέπει να επισημάνουμε ότι στο λεξικό περιλάβαμε και τις κατηγορίες προγραμμάτων που περιέχονται στο πρότυπο TV-Anytime (genres), γιατί είναι συγκεκριμένες και οι περισσότερες εκφράσεις περιέχουν προσδιορισμό της κατηγορίας των προγραμμάτων στην οποία αναφέρονται. Έτσι, δε χρειάζεται να γίνεται αναζήτηση και ανάκτηση από τον TV-Anytime server του συστήματος των κατηγοριών ενός τηλεοπτικού προγράμματος, με επιβάρυνση της απόδοσης του συστήματος.

Ας εξηγήσουμε την σύνταξη μέσα στο λεξικό για να μπορούμε να κατανοήσουμε καλύτερα τα παραδείγματα που ακολουθούν. Αυτό που γίνεται μέσα στο λεξικό με τις ρίζες των λέξεων (stem lexicon) είναι μία κατηγοριοποίηση των λέξεων που περιέχονται, με χαρακτηριστικά

ονόματα, συν τον τύπο από την κατάληξη ο οποίος ταιριάζει με κάθε λέξη, και περιέχεται στο mini lexicon, ώστε να αναγνωρίζονται αργότερα μέσα από τους γραμματικούς κανόνες. Ας δούμε μία κατηγορία λέξεων μέσα από το stem lexicon

//** SELECTION LIST VERBS **//

```
stor = addslv (V2) :      0 lex = store .
add = addslv (V1) :      0 lex = add .
remov = delslv (V1) :    0 lex = remove .
insert = addslv (V1) :   0 lex = insert .
delet = delslv (V2) :    0 lex = delete .
record = addslv (V1) :   0 lex = record .
```

Εδώ βλέπουμε ρίζες ρημάτων τα οποία, ουσιαστικά χρησιμοποιούνται στη διαχείριση της εξατομικευμένης λίστας του χρήστη, όπου αντιστοιχίζονται με ένα χαρακτηριστικό όνομα (π.χ. addslv) με το οποίο χρησιμοποιούνται μέσα στους γραμματικούς κανόνες. Μετά το όνομα που τους δίνουμε έχουμε τον τύπο της κατάληξης, που ταιριάζει με την κάθε λέξη. Ο ακέραιος 0 υποδηλώνει τη θέση της λέξης στην κύρια σχέση που περιγράφεται. Για παράδειγμα, στην πρώτη περίπτωση το 0 αφορά στο stor και αυτό που θέλει να δηλώσει η σχέση είναι ότι για κάθε λέξη που ξεκινά από stor και ανήκει σε αυτές που ικανοποιούν τον κανόνα stor + V2, ο αναλυτής θα την αντιλαμβάνεται σαν τη λέξη store. Ας δούμε, όμως, λίγο τι συμβαίνει στο mini lexicon και για παράδειγμα, ας δούμε τον τύπο V2, ο οποίος περιγράφεται ως εξής:

V2

```
e = * (stop):  0 vform = pres :
                0 active = yes .
```

```
ing = * (stop): 0 vform = pres :
                0 finite = yes :
                0 active = yes .
```

```
es = * (stop):  0 vform = pres :
                0 finite = yes :
                0 active = no .
```

```
ed = * (stop):  0 vform = past :
                0 finite = yes .
```

@

Σχήμα 16: Παράδειγμα του mini λεξικού με τις καταλήξεις κατηγορίας ρημάτων

Εδώ, λοιπόν φαίνονται οι καταλήξεις που μπορούν να αντιστοιχηθούν με τη ρίζα του ρήματος stor. Ο αστερίσκος με τη λέξη stor σημαίνει ότι η κατάληξη θα είναι αυτή και μόνο αυτή. Η πληροφορία που ακολουθεί δεξιά αφορά στον χρόνο και τη φωνή που χαρακτηρίζουν αυτήν την κατάληξη. Για παράδειγμα το e ανήκει στον ενεστώτα (present) και έχει ενεργητική φωνή (active). Το γεγονός ότι ο τύπος αυτός ξεινιά με το όνομα της κατηγορίας και τελειώνει με το σύμβολο @, επιβάλλεται από την JavaChart σύνταξη για το mini lexicon.

Μέσα από το λεξικό δίνονται και τιμές σε κάποιες από τις παραμέτρους της δομής ιδιοτήτων, που προκύπτει μετά τη διέλευση της πρότασης από την υπο-μονάδα της ανάλυσης. Για παράδειγμα, από το entity markers, το field target παίρνει τιμή μέσα από το λεξικό και ο τρόπος που δηλώνεται αυτό φαίνεται στο παρακάτω παράδειγμα, που αποτελεί μέρος του λεξικού με τις ρίζες των λέξεων:

```
profil = pm (N3) :
    0 lex = profile :
    0 markers target = profile .
```

Σχήμα 17: Παράδειγμα του λεξικού με τις ρίζες των λέξεων

Εδώ το νέο στοιχείο που έχουμε είναι η τρίτη γραμμή στην οποία παίρνουμε την τιμή (value), για το πεδίο (field) target, του αντικειμένου (entity) markers.

Το N3 είναι η κατηγορία του mini lexicon που δηλώνει τις καταλήξεις, που μπορούν να αντιστοιχιστούν με την ρίζα profil και αυτές είναι οι παρακάτω:

```
N3
e = * (gen):    0 agr numb = sg .
es = * (gen):   0 agr numb = pl .
```

Σχήμα 18: Παράδειγμα του mini λεξικού με τις καταλήξεις κατηγορίας ουσιαστικών

Το sg αφορά στον ενικό (singular) και το pl στον πληθυντικό (plural). Έτσι, ικανοποιούνται και ο ενικός και ο πληθυντικός αυτής και όμοιας με αυτήν λέξης.

5.2.3 Γραμματική

Η γραμματική λειτουργεί σαν ένα δέντρο με κύριους κανόνες με πολλούς επιμέρους που μπορεί να τους ικανοποιούν. Αρχικά, έχουμε τις πράξεις του λόγου, που αναφέρουν πως

μπορούν να συνδυαστούν μεταξύ τους οι 4 μεγάλες κατηγορίες φράσεων που αναλύσαμε και παραπάνω και ανάλογα σε ποιο entity, αντιστοιχεί η καθεμία, γίνεται και η αντίστοιχη ανάθεση. Για παράδειγμα, ο κανόνας:

```
s -> retintro multisearch temp target :
    0 markers = 1 markers :
    0 markers = 4 markers :
    0 properties = 2 properties :
    0 temporal = 3 temporal.
```

Σχήμα 19: Παράδειγμα από την γραμματική

αναφέρεται στις εκφράσεις που αφορούν σε αναζητήσεις με εισαγωγική φράση τύπου 1, φράση πολλαπλής αναζήτησης, χρονική φράση και φράση στόχου. Θα πρέπει λίγο να εξηγήσουμε πως λειτουργεί η σύνταξη στην γραμματική. Οποδήποτε υπάρχει αριστερά από το βέλος αφορά σε κατηγορία φράσεων, εκτός από το s που είναι ακόμα γενικότερο και αφορά σε κατηγορία έκφρασης. Δεξιά από το βέλος μπορούμε να έχουμε είτε κατηγορία φράσεων που στη συνέχεια, θα περιγραφεί από άλλον γραμματικό κανόνα είτε κατηγορία από το λεξικό (stem lexicon) και η ανάλυση της θα περάσει στον έλεγχο του λεξικού. Μετά από την αντιστοίχιση ακολουθεί μία ανάθεση για τα αντικείμενα και τα πεδία αυτών. Στο παραπάνω παράδειγμα, αυτό που βλέπουμε είναι ότι η συγκεκριμένη έκφραση θα χαρακτηρίζεται από τα αντικείμενα markers, properties, temporal και τα πεδία του καθενός από αυτά, θα πάρουν τιμές με τον εξής τρόπο. Το αντικείμενο markers που θα χαρακτηρίσει αυτήν την έκφραση θα συμπληρώσει αντίστοιχα, τα διάφορα πεδία του από τις κατηγορίες φράσεων retintro και target. Έτσι φαίνεται και ο ρόλος των ακεραίων αριθμών στην αντιστοίχιση. Αντίστοιχα, τα πεδία του αντικειμένου properties θα πάρουν τιμές μέσα από την φράση multisearch και τα πεδία του temporal από το temp.

Στην περίπτωση που θέλουμε να επιτρέπεται μία σειρά από φράσεις του ίδιου είδους, όπως γίνεται και στην περίπτωση των φράσεων αναζήτησης, αυτό γίνεται από το συνδυασμό των δύο παρακάτω κανόνων:

```
multisearch -> multisearch multisearch :
    0 properties = 1 properties :
    0 properties = 2 properties .
```

```
multisearch -> search :
```


0 properties = 1 properties .

όπου το search αποτελεί κατηγορία φράσης. Η ανάθεση των τιμών στα διάφορα πεδία γίνεται από λέξεις ή σειρά λέξεων που εισάγει ο χρήστης. Κάτι τέτοιο ικανοποιείται από κανόνα της παρακάτω μορφής:

Θα πρέπει να επισημάνουμε ότι ανάλογα με την φράση που έχουμε από τον χρήστη και την ακολουθία των λέξεων που την χαρακτηρίζει δηλώνουμε τα πεδία για κάθε αντικείμενο, τα οποία ουσιαστικά ανήκουν σε αυτά που περιγράψαμε στην παράγραφο 4.4.2- Απεικόνιση της πληροφορίας.

sumphrase -> det type genre summary pr min val cc max val length pr theme :

0 properties type = 2 lex :

0 properties genre = 3 lex :

0 properties format = 11 lex :

0 properties minlength = 7 properties value :

0 properties maxlength = 10 properties value :

0 properties theme = 13 properties value .

value -> *{0,1} : 0 properties value = 1 \$tokens .

theme -> * : 0 properties value = 1 \$tokens .

Εδώ βλέπουμε έναν κανόνα που ικανοποιεί φράση περίληψης, στην μεγαλύτερη δυνατή μορφή, όπου ο χρήστης δηλώνει τύπο, κατηγορία περίληψης, μέγιστο και ελάχιστο duration ή αριθμό frames ή χαρακτήρων, ανάλογα με τον τύπο της περίληψης και το θέμα της. Ο τύπος (type) και η κατηγορία (genre) της παίρνουν τιμές από λέξεις που υπάρχουν στο λεξικό (lex), οι λέξεις αυτές θα είναι στη σειρά η δεύτερη και η τρίτη μέσα στην φράση αυτή και θα ικανοποιούν τις προβλεπόμενες από το TVA τιμές. Αυτό σημαίνει ότι γίνεται μία επιβεβαίωση της λέξης που εισάγει ο χρήστης, ώστε να υπάρχει στο λεξικό. Η τιμή για το μέγιστο ή το ελάχιστο έρχεται από την εισαγωγή του χρήστη και μπορεί να είναι ένας οποιοσδήποτε αριθμός. Το θέμα της παίρνει πάλι τιμή από την οποιαδήποτε ακολουθία λέξεων, που θα εισάγει ο χρήστης. Ας δούμε αναλυτικά τι συμβαίνει στην περίπτωση του value και του theme και πως παίρνουν τις τιμές τους. Ο συμβολισμός *{0,1} σημαίνει ότι η τιμή μπορεί να περιέχει οποιαδήποτε μία ή δύο λέξεις εισάγει ο χρήστης. Ο συμβολισμός * σημαίνει ότι ο αριθμός των λέξεων δεν είναι συγκεκριμένος. Ο συμβολισμός \$tokens σημαίνει ότι η τιμή θα έρθει από την εισαγωγή του χρήστη και οι λέξεις που θα την απαρτίζουν δε χρειάζεται να βρίσκονται στο λεξικό.

Όπως είπαμε, όταν υπάρχει πεδίο που παίρνει την τιμή του από λέξη που υπάρχει στο λεξικό και ο χρήστης εισάγει κάποια άλλη λέξη, τότε ο χρήστης ενημερώνεται κατάλληλα.

Το πρόβλημα που αντιμετωπίσαμε στην γραμματική είναι ότι μπορεί μία έκφραση του χρήστη να ικανοποιείται από περισσότερους από έναν γραμματικούς κανόνες και κυρίως, λόγω του κανόνα που ικανοποιεί την κατηγορία του TVA για τα keywords (CreationPreferences DS). Όπως παρατηρούμε, ο γραμματικός κανόνας που αντιστοιχίζεται σε αυτά είναι:

search -> ByKeyword :
 $0 \text{ properties} = 1 \text{ properties} .$

ByKeyword -> pr keyword:
 $0 \text{ properties keyword} = 2 \text{ properties value} .$

keyword -> * : $0 \text{ properties value} = 1 \$\text{tokens} .$

Ας δούμε πως δουλεύει αυτός ο γραμματικός κανόνας. Σε μία φράση αναζήτησης, που ξεκινά από επίρρημα (κατηγορία pr από το λεξικό με τις ρίζες των λέξεων) και ακολουθείται από ένα σύνολο συμβολοσειρών (*), οι συμβολοσειρές αυτές αναγνωρίζονται από το σύστημα και εισάγονται σαν τιμές στο πεδίο keyword της οντότητας properties (Κεφ. 4).

Αντίστοιχα, ο γραμματικός κανόνας που ικανοποιεί, για παράδειγμα φράσεις αναζήτησης βασισμένες στο τίτλο ενός τηλεοπτικού προγράμματος, είναι ο ακόλουθος.

search -> ByTitle :
 $0 \text{ properties} = 1 \text{ properties} .$

ByTitle -> pr t keyword:
 $0 \text{ properties title} = 3 \text{ properties value} .$

keyword -> * : $0 \text{ properties value} = 1 \$\text{tokens} .$

Η διαφορά με τον προηγούμενο γραμματικό κανόνα είναι ότι η φράση αναζήτησης μετά το επίρρημα ακολουθείται από την λέξη 'title', την οποία αναγνωρίζει η γραμματική μας, οπότε η ακολουθία των συμβολοσειρών μετά την λέξη 'title', γίνεται τιμή στο πεδίο 'title' της οντότητας properties.

Έτσι, για την απλή πρόταση ‘show me drama movies with title the pianist’ η έξοδος από τον αναλυτή είναι οι δύο παρακάτω δομές ιδιοτήτων:

```
[ [ temporal : [ ] ]
  [ properties : [ keyword : [ 2 : pianist ]
                    [ 1 : the ]
                    [ 0 : title ] ]
    [genre2 : movies ]
    [genre1 : drama ] ]
[ markers : [ act : ret ] ] ]
```

```
[ [ temporal : [ ] ]
  [ properties : [ title : [ 1 : pianist ]
                    [ 0 : the ] ]
    [genre2 : movies ]
    [genre1 : drama ] ]
[ markers : [ act : ret ] ] ]
```

Σχήμα 20: Δομές ιδιοτήτων που παράγονται από τον αναλυτή διαγραμμάτων

Ας εξηγήσουμε λίγο τη σύνταξη της δομής ιδιοτήτων. Αυτό που παράγεται από τον αναλυτή διαγραμμάτων έχει τη μορφή λίστας και περιέχει τα αντικείμενα που ορίζονται από τον τύπο της έκφρασης. Κάθε αντικείμενο περιέχει μία άλλη λίστα από τα πεδία τα οποία αναγνωρίστηκαν από τις επιμέρους φράσεις και αποτελούν τις διάφορες TVA κατηγορίες. Κάθε πεδίο περιέχει μία λίστα με την τιμή ή τις τιμές του. Στην περίπτωση που οι τιμές προέρχονται από την έκφραση του χρήστη αυτή καθεαυτή και λαμβάνονται από το σύστημα με τον τρόπο που δείξαμε πριν (\$tokens) τότε η εισαγωγή αυτή σπάει σε ένα σύνολο από συμβολοσειρές, που με την κατάλληλη αρίθμηση απαρτίζουν την τιμή. Έτσι, στη δεύτερη δομή ιδιοτήτων βλέπουμε ότι η φράση ‘the pianist’ σπάει σε δύο συμβολοσειρές, την 0: the και την 1: pianist, όπου οι ακέραιοι υποδηλώνουν τη σειρά μέσα στη φράση.

Αυτό που παρατηρούμε εδώ είναι ότι στην πρώτη δομή ιδιοτήτων την υπο-πρόταση ‘title the pianist’ την έλαβε ως keyword, ενώ στην δεύτερη, σωστά έλαβε ως τίτλο την φράση ‘the pianist’. Τέτοιου είδους προβλήματα αντιμετωπίζονται όταν οι δομές ιδιοτήτων, εισέρχονται στην επόμενη υπο-μονάδα, διαχείρισης διαλόγου.

5.3 Η Υπομονάδα Διαχείρισης Διαλόγου (Dialogue Manager Module)

Η υπο-μονάδα διαχείρισης διαλόγου αποτελεί τον κορμό του συστήματος και είναι εξαιρετικής σημασίας. Επικοινωνεί με όλες τις υπο-μονάδες, αναλαμβάνει να φέρει στην κατάλληλη μορφή την δομή ιδιοτήτων που προέρχεται από την εισαγωγή του χρήστη, να λύσει τυχόν ασάφειες με τον καλύτερο δυνατό τρόπο χρησιμοποιώντας την υπο-μονάδα επίλυσης ασαφειών, να αλληλεπιδράσει με τον server για την ανάκτηση αποτελεσμάτων και να τροφοδοτήσει την υπο-μονάδα απόκρισης πληροφορία για την σύνταξη μηνυμάτων προς τον χρήστη.

Όπως είπαμε και στην προηγούμενη παράγραφο, μία λίστα από μία ή περισσότερες δομές ιδιοτήτων έρχεται στον διαχειριστή διαλόγου από τον αναλυτή. Ο πρώτος στόχος είναι μέσα από μία σειρά ελέγχων να μπορέσουμε να περιορίσουμε την λίστα αυτή σε μία δομή ιδιοτήτων με την πληροφορία του χρήστη. Η διαδικασία που ακολουθείται είναι η ακόλουθη: Πρώτα κοιτάμε αν η λίστα περιέχει έστω και μία δομή ιδιοτήτων με το entity properties. Αν δεν υπάρχει τότε ο χρήστης έχει ζητήσει ανάκτηση είτε όλου του selection list, είτε των αποθηκευμένων στο PDR προγραμμάτων του. Αν υπάρχει παίρνουμε την λίστα και για κάθε μία δομή ιδιοτήτων κοιτάμε αν υπάρχει, για παράδειγμα το πεδίο 'title' ή συνώνυμες λέξεις ('name') σε αυτή. Αν για κάποιες από αυτές υπάρχει, τότε διαγράφουμε αυτές στις οποίες δεν υπάρχει, γιατί σε αυτές θα έχει συμπεριλάβει τη λέξη 'title' στη σειρά λέξεων του 'keyword'. Έτσι, μέσα από έλεγχους για διάφορες κατηγορίες καταλήγουμε σε μία δομή ιδιοτήτων.

Όταν υπάρχει χρονική φράση στην εισαγωγή του χρήστη, τότε γίνεται χρήση ενός αντικειμένου Calendar, για να τεθούν οι αντίστοιχες τιμές στα κατάλληλα πεδία του και, έπειτα, καλείται η κλάση του επιλυτή ασαφειών που είναι υπεύθυνη να δώσει την κατάλληλη μορφή στις τιμές των πεδίων αυτών (Date/Time Resolver), ώστε να μπορούν να χρησιμοποιηθούν από τον TVA server. Για παράδειγμα, όταν η εισαγωγή του χρήστη είναι 'show me drama movies for tonight' η δομή ιδιοτήτων για το entity temporal θα είναι η εξής:

```
[ temporal : [ timeind : pm ]
               [ before : [ time : 12 ] ]
               [ after : [ time : 5 ] ] ]
```

Σχήμα 21: Δομή ιδιοτήτων για χρονική φράση με ασάφεια

Ο διαχειριστής διαλόγου θα θέσει στο αντικείμενο Calendar σαν ημερομηνία, την ημερομηνία του συστήματος, σαν ώρα την τιμή 17 και θα κρατήσει και μία τιμή για το duration σε λεπτά, τον χρόνο που μεσολαβεί ανάμεσα στις 5 και στις 12. Σε περίπτωση που ο χρήστης εισάγει σαν χρονική φράση ‘after 5’ στην οποία δεν υπάρχει time indicator, τότε το σύστημα λαμβάνει υπόψη του ότι η εισαγωγή του χρήστη περιείχε σαν τιμή το 5 και όχι το 17, που υποδηλώνει pm.

Στην περίπτωση που η εισαγωγή του χρήστη έγινε κατά τέτοιο τρόπο, ώστε να μην υπάρχουν στοιχεία για τις κατηγορίες στις οποίες αναφέρεται και να υπάρχει στο entity properties πεδίο keyword με μία λίστα από μία ή περισσότερες λέξεις, ακολουθείται η παρακάτω διαδικασία: Δημιουργείται ένα διάνυσμα από τις λέξεις αυτές. Γίνεται ένας έλεγχος για το αν περιέχονται οι λέξεις ‘and’ ή ‘or’ . Έπειτα, περνάει από την stop list. Επιστρέφεται το νέο διάνυσμα με τις ‘ουσιαστικές’ λέξεις για το σύστημα. Καλείται από την υπο-μονάδα επίλυσης ασαφειών η κλάση που είναι υπεύθυνη για την απόδοση σημασιολογίας TVA (TVA Semantics Resolver). Από την διαδικασία αυτή επιστρέφεται η σημασιολογία για κάθε συμβολοσειρά του διανύσματος. Όπως είπαμε και στην αρχή του κεφαλαίου η σχεσιακή βάση του συστήματος περιέχει TV-Anytime μεταδεδομένα και οντολογίες με όλη την υπάρχουσα πληροφορία σε περιβάλλον ψηφιακής τηλεόρασης. Έτσι, κατά την αναζήτηση της σημασιολογίας δεν υπάρχει ενδεχόμενο να μην επιστραφούν αποτελέσματα. Η πληροφορία αυτή κρατείται, ενώ από τον διαχειριστή διαλόγου καλείται και η κλάση, του επιλυτή ασαφειών, υπεύθυνη για το φιλτράρισμα της πληροφορίας μέσα από το προφίλ του χρήστη, για να χρησιμοποιηθούν τα αποτελέσματά της στην καλύτερη ταξινόμηση των αποτελεσμάτων.

Στο σημείο αυτό ο διαχειριστής διαλόγου προσπαθεί μέσα από ένα σύνολο ενεργειών να χρησιμοποιήσει τα στοιχεία που συγκέντρωσε για την καλύτερη ταξινόμηση των αποτελεσμάτων. Όπως είπαμε κοιτάμε αν μέσα στις λέξεις που έχει εισάγει ο χρήστης υπάρχει το ‘and’ ή το ‘or’. Έπειτα περνάμε το διάνυσμα με τις λέξεις αυτές από την stop list και διακρίνουμε τις παρακάτω περιπτώσεις:

-Δε κόβονται λέξεις από το stop list

Η ανάκτηση από την βάση δεδομένων του συστήματος γίνεται για κάθε μία λέξη και οι πίνακες με αποτελέσματα από ίδια κατηγορία TVA ενώνονται σε έναν με αποτελέσματα που ήταν κοινά στους πίνακες αυτούς.

-Κόβονται λέξεις από το stop list, αλλά δεν υπάρχει ανάμεσά τους το ‘and’ ή το ‘or’

Γίνεται ανάκτηση πληροφορίας και συντάσσεται το XML έγγραφο με την πληροφορία συμβατό με το FilteringAndSearchPreferences DS.

-Κόβονται λέξεις από το stop list και υπάρχει ανάμεσά τους το ‘and’ ή το ‘or’

Από τα αποτελέσματα που ανακτώνται υπάρχει πάλι η ενοποίηση όλων των πινάκων που περιέχεται πληροφορία ίδιας κατηγορίας TVA. Για παράδειγμα, όλοι οι Creators συγκεντρώνονται σε έναν πίνακα, όλα τα Countries σε κάποιον άλλο πίνακα και έχουμε την σύνταξη διαφορετικών XML εγγράφων ανά κατηγορία, συμβατών με το FilteringAndSearchPreferences DS.

Έπειτα, ο διαχειριστής διαλόγου διαμορφώνει την πληροφορία σε κατάλληλη μορφή για να μπορεί να την διαχειριστεί σε αλληλεπίδραση με τον TVA server. Η δομή αυτή περιέχει δύο strings, που παίρνουν τιμές από το entity markers και τα πεδία action, target και ένα ή περισσότερα TVA XML έγγραφα, βασισμένα στο UserPreferences DS. Στην περίπτωση, που ο χρήστης εισάγει πληροφορία για τη δημιουργία περίληψης περιέχει πληροφορία συμβατή με το UserIdentifier DS και το BrowsingPreferences DS, όπως για παράδειγμα η έκφραση

‘I want a textual news summary of 1000 characters at 5 pm’,

ενώ σε όλες τις άλλες των περιπτώσεων συμβατή με το UserIdentifier DS και το FilteringAndSearchPreferences DS, όπως για παράδειγμα η έκφραση

‘I am looking for movies with title Roxanne’.

Η δομή αυτή που δημιουργείται, σε συνδυασμό με πληροφορία για τον προσωπικό λογαριασμό του χρήστη (username, password), χρησιμοποιείται για την ανάκτηση της πληροφορίας από τον server.

5.4 Η Υπομονάδα Επίλυσης Ασαφειών (Ambiguities Resolver Module)

Η υπο-μονάδα αυτή περιέχει τρεις κλάσεις, την Date/Time Resolver, την TVA Semantics Resolver και την User's Profile Resolver, τις οποίες θα αναλύσουμε λεπτομερώς.

5.4.1 Date/Time Resolver

Από το πρότυπο TV-Anytime ορίζεται ότι στην περίπτωση που αναφερόμαστε σε χρόνους, υπάρχει υποχρεωτικά τιμή στο πεδίο TimePoint και όχι υποχρεωτικά τιμή στο πεδίο Duration. Από τον διαχειριστή διαλόγου έρχεται ένα αντικείμενο Calendar και στο σημείο αυτό γίνεται η μετατροπή της πληροφορίας που υπάρχει σε αυτό σε μορφή συμβατής με το πρότυπο MPEG-7, που ακολουθεί και το TV-Anytime (reference).

Δηλαδή για την ημερομηνία '1 January 2003' και ώρα '14:00' η τελική μορφή είναι η εξής:
01-01-2003T14:00:00

Η ίδια διαδικασία εκτελείται και στην περίπτωση του Duration

Δηλαδή, για χρονική περίοδο 1501 λεπτά, η τελική μορφή είναι η εξής:

P1DT1H1M

δηλαδή μία μέρα, μία ώρα και ένα λεπτό.

5.4.2 TVA Semantics Resolver

Όταν επιστρέφεται το νέο διάνυσμα με τις λέξεις που δεν κόπηκαν από το stop list, είχαμε δει ότι η επόμενη κίνηση από τον διαχειριστή διαλόγου είναι να βρει τη σημασιολογία των λέξεων κάνοντας χρήση του TVA Semantics Resolver. Για κάθε μία από τις λέξεις γίνεται μία αναζήτηση μέσω αντίστοιχων SQL Queries και στο τέλος επιστρέφεται ένας διάνυσμα από επιμέρους διανύσματα, που το κάθε ένα από αυτό περιέχει πληροφορία για την τιμή σε διαφορετική TV-Anytime κατηγορία, όπως αυτή προέρχεται από τη βάση.

Οι κύριες κατηγορίες στις οποίες ψάχνουμε να αποδώσουμε σημασιολογία στις ασαφείς αυτές λέξεις είναι οι εξής: Creator, Country, Language, DisseminationSource. Δεν συμπεριλάβαμε άλλες κατηγορίες επειδή στη μορφή των φράσεων που εισάγει ο χρήστης δε θα μπορούσε να αποδοθεί κάποιου άλλου είδους σημασιολογία στις λέξεις αυτές. Σε περίπτωση, όμως επέκτασης του μοντέλου μας με προσθήκη και άλλων ειδών φράσεων η διαδικασία μπορεί να επεκταθεί και σε άλλες κατηγορίες TVA.

Ο TVA Semantics Resolver περιέχει συναρτήσεις για την αναζήτηση στην βάση κατηγοριών πέρα από αυτές τις τέσσερις που χρησιμοποιούνται για τις ασαφείς λέξεις και , που χρησιμοποιούνται για τον έλεγχο της ορθότητας της εισαγωγής του χρήστη. Για παράδειγμα αν ο χρήστης εισάγει την πρόταση 'Give me drama movies created in Franc' η SQL ερώτηση θα ψάξει στη βάση χώρες παραγωγής δραματικών ταινιών με το string "%franc%", που υπολογίζει ότι μπροστά και πίσω από το string μπορεί να υπάρχουν και άλλοι χαρακτήρες, οπότε στα αποτελέσματα θα υπάρχει η σωστή πληροφορία για την χώρα που είναι France, και ας έκανε λάθος ο χρήστης.

5.4.3 User's Profile Resolver

Η χρήση του επιλυτή μέσω του προφίλ του χρήστη, που αλληλεπιδρά μέσω φυσικής γλώσσας με το σύστημα, γίνεται για να μπορούμε να εμφανίσουμε καλύτερα ταξινομημένα αποτελέσματα στον χρήστη. Όπως αναφέραμε και προηγούμενα, ο κάθε χρήστης μπορεί να έχει ένα ή περισσότερα προφίλ, που προσδιορίζουν τα εκάστοτε ενδιαφέροντά του. Ο επιλυτής εδώ κάνει χρήση του Java API που παρέχεται από το σύστημα για την επικοινωνία με τον TVA server και παρέχοντας το διάνυσμα με τις λέξεις που δεν κόπηκαν από το stop list, καθώς και το username του χρήστη, φιλτράρει το προφίλ του χρήστη για να βρει αν κάποιες από τις ασαφείς λέξεις που έχει εισάγει στο σύστημα έχουν δηλωθεί στο προφίλ του. Αν πράγματι υπάρχει τέτοιου είδους πληροφορία, του επιστρέφεται ένα XML έγγραφο συμβατό με το FilteringAndSearchPreferences DS, που αποτελεί κομμάτι του προφίλ του χρήστη και περιέχει μόνο τις κατηγορίες στις οποίες βρέθηκαν τιμές, ίδιες με αυτές που εισήγαγε ο επιλυτής. Το έγγραφο αυτό παρέχεται από τον επιλυτή στον διαχειριστή διαλόγου.

5.5 Η Υπομονάδα Διαχείρισης Απόκρισης (Response Manager Module)

Ο διαχειριστής αποκρίσεων είναι υπεύθυνος για την διαχείριση των αποτελεσμάτων και τη δημιουργία μηνυμάτων με τα οποία τροφοδοτεί τελικώς, τον χρήστη. Τα μηνύματα πρέπει να είναι τέτοια ώστε ο χρήστης να καταλάβει τι ακριβώς πήγε ή δεν πήγε καλά στην προσπάθειά του να αλληλεπιδράσει με το σύστημα, και εμπλουτίζονται, όποτε χρειάζεται, με πληροφορία που παρέχει ο διαχειριστής διαλόγου. Τα μηνύματα θα μπορούσαν να ομαδοποιηθούν στις παρακάτω κατηγορίες:

Τύπος 1: Ο χρήστης εισάγει πληροφορία για ανάκτηση αποτελεσμάτων βάση μίας κατηγορίας TVA και το σύστημα δε βρίσκει αποτελέσματα. Για παράδειγμα: Στην εισαγωγή του χρήστη

‘Find me Spanish spoken drama movies’

η απόκριση του συστήματος θα είναι

"There are no available programs spoken in this language."

Τύπος 2: Ο χρήστης εισάγει πληροφορία για ανάκτηση αποτελεσμάτων βάση μίας κατηγορίας TVA και το σύστημα δε βρίσκει στη βάση την τιμή που έθεσε ο χρήστης για την συγκεκριμένη κατηγορία. Για παράδειγμα: Στην εισαγωγή του χρήστη

‘Find me drama movies broadcasted by Kydon’

η απόκριση του συστήματος θα είναι

"There is no broadcast channel or server with name Kydon. Please try again."

Τύπος 3: Ο χρήστης εισάγει πληροφορία για ανάκτηση αποτελεσμάτων και ανάλογα με το αν το σύστημα ανέκτησε ή όχι αποτελέσματα έχουμε καταφατικό μήνυμα με την λίστα των αποτελεσμάτων ή αρνητικό μήνυμα. Για παράδειγμα: Στην εισαγωγή του χρήστη

‘Find me drama movies broadcasted by ERT’

η απόκριση του συστήματος θα είναι

"The results of your search are:" + λίστα αποτελεσμάτων ή

"No results were found. Please try again."

Τύπος 4: Όπως έχουμε ξαναπεί ο χρήστης μπορεί να κάνει διάφορες ενέργειες πάνω στην εξατομικευμένη λίστα του ή στα αποθηκευμένα προγράμματα το PDR του και να δημιουργήσει και νέο προφίλ. Όμως αυτό δεν εμποδίζει τον χρήστη να ζητήσει από το σύστημα λειτουργίες τις οποίες δε μπορεί να ικανοποιήσει χρησιμοποιώντας το σύστημα, όπως να σβήσει περιεχόμενο από το ήδη υπάρχον προφίλ του. Στην συγκεκριμένη κατηγορία, λοιπόν, μηνυμάτων συμπεριλαμβάνουμε όλα τα μηνύματα που έχουν σχέση με Insert, Update, Delete από το selection list και το user's profile.

Ας δούμε και ένα παράδειγμα για την παραπάνω περίπτωση. Στην περίπτωση που ο χρήστης παρέχει πληροφορία για την εγγραφή κάποιων προγραμμάτων (InsertIntoUserSelectionList με status Recorded) και κάποια από τα προγράμματα υπήρχαν μέσα στο selection list του, αλλά δεν είχαν status recorded (UpdateUserSelectionList με status Recorded) το μήνυμα που του επιστρέφεται από τον διαχειριστή αποκρίσεων είναι το παρακάτω, θεωρώντας ότι αναγνώρισε 4 προγράμματα εκτός selection list και 2 εντός.

“4 programs were found and added into your selection list and 2 programs from your selection list changed status, so to be recorded.”

Τύπος 5: Ο χρήστης εισάγει πληροφορία για την δημιουργία μίας περίληψης που θα είναι διαθέσιμη μετά από δεδομένη χρονική στιγμή και το σύστημα αποκρίνεται ως εξής, αν θεωρήσουμε ότι ο χρήστης θέλει την περίληψη του μετά τις 5 το απόγευμα.

“The summary you asked will be available after 5 pm.”

Τύπος 6: Ο χρήστης ζητά να δει τα περιεχόμενα του selection list ή τα αποθηκευμένα προγράμματα. Στις δύο αυτές περιπτώσεις τα μηνύματα είναι τα εξής:

"The contents of your selection list are: " + λίστα αποτελεσμάτων

"The stored contents of your selection list are: " + λίστα αποτελεσμάτων

όταν υπάρχουν αποτελέσματα και

"Your selection list is empty. "

"There are no recorded programs in your PDR. "

όταν δεν υπάρχουν αποτελέσματα.

Όπως είδαμε και στην αρχιτεκτονική ο διαχειριστής διαλόγου παρέχει στον διαχειριστή αποκρίσεων τη δομή που δημιουργείται από την εισαγωγή του χρήστη (action, target, list of XML έγγραφα συμβατά με το UserIdentifier DS και το FilteringAndSearchPreferences DS)

ή (XML έγγραφο συμβατό με το UserIdentifier DS και το BrowsingPreferences DS), μαζί με τα στοιχεία του (username, password), καθώς επίσης και με το XML έγγραφο, συμβατό με το FilteringAndSearchPreferences DS, που προήλθε από το φιλτράρισμα της πληροφορίας από το προφίλ του χρήστη, αν αυτό υπάρχει. Για να μπορέσουμε αναλυτικά να δούμε πως συμπεριφέρεται ο διαχειριστής αποκρίσεων μετά το σημείο αυτό θα πρέπει να διακρίνουμε όλες τις περιπτώσεις που καλείται να αντιμετωπίσει.

Αν ξεκινήσουμε με την περίπτωση που ο χρήστης εισάγει στοιχεία για δημιουργία περίληψης. Στην απλή αυτή περίπτωση ο διαχειριστής αποκρίσεων εισάγει το έγγραφο στη βάση, μέσω του XML-DB Middleware και επιστρέφει μήνυμα στον χρήστη ενημερώνοντάς τον για την επιτυχή έκβαση της ενέργειάς του.

Συνεχίζοντας, θα αναλύσουμε τις ενέργειες που γίνονται όταν ο χρήστης ζητήσει την ανάκτηση ή την διαγραφή όλης της πληροφορίας που βρίσκεται στην εξατομικευμένης λίστα του (selection list) ή των αποθηκευμένων στο PDR του προγραμμάτων. Αυτό συμβαίνει όταν η λίστα από τα XML έγγραφα που θα έπρεπε να περιέχει η δομή που προέρχεται από τον διαχειριστή διαλόγου είναι κενή. Ο διαχειριστής αποκρίσεων κάνει χρήση συγκεκριμένων συναρτήσεων που παρέχονται από ένα Java API, που έχει αναπτυχθεί στα πλαίσια του project UP-TV και στο οποίο έχουμε αναφερθεί στην αρχή του κεφαλαίου αυτού, ανακτά την πληροφορία και την επιστρέφει στο χρήστη μαζί με το κατάλληλο μήνυμα.

Τις υπόλοιπες περιπτώσεις θα τις ταξινομήσουμε ανάλογα με το action και το target που παίρνουν τις τιμές τους από τις εισαγωγικές φράσεις και τις φράσεις στόχου της εισαγωγής του χρήστη.

Action: Insert

Target: Profile

Ο διαχειριστής αποκρίσεων εισάγει στη βάση, μέσω του XML-DB Middleware, το XML έγγραφο με την πληροφορία συμβατή με το UserIdentifier DS και το FilteringAndSearchPreferences DS και επιστρέφει στον χρήστη μήνυμα για την επιτυχή δημιουργία του νέου του προφίλ.

Target: List

Ο διαχειριστής αποκρίσεων εισάγει στη βάση, μέσω του XML-DB Middleware, το XML έγγραφο με την πληροφορία συμβατή με το UserIdentifier DS και το FilteringAndSearchPreferences DS με προσωρινά στοιχεία του χρήστη, ώστε να μη σβήσει κάποιο ήδη υπάρχον προφίλ του και κάνει ανάκτηση των μεταδεδομένων που ταιριάζουν με την πληροφορία που έχει εισάγει ο χρήστης. Έπειτα χρησιμοποιεί την πληροφορία αυτή και κάνοντας χρήση του Java API εισάγει το υλικό στην εξατομικευμένη λίστα (selection list) του χρήστη με status undefined, ώστε να αποφασιστεί αργότερα από το υπόλοιπο σύστημα ή τον ίδιο τον χρήστη πως θα διαχειριστεί αυτά τα προγράμματα. Ο χρήστης ενημερώνεται κατάλληλα.

Target: PDR

Ο διαχειριστής αποκρίσεων εισάγει στη βάση, μέσω του XML-DB Middleware, το XML έγγραφο με την πληροφορία συμβατή με το UserIdentifier DS και το FilteringAndSearchPreferences DS με προσωρινά στοιχεία του χρήστη, ώστε να μη σβήσει κάποιο ήδη υπάρχον προφίλ του και κάνει ανάκτηση των μεταδεδομένων που ταιριάζουν με την πληροφορία που έχει εισάγει ο χρήστης. Έπειτα ανακτά την πληροφορία για τα μεταδεδομένα των προγραμμάτων που υπάρχουν στο selection list του χρήστη και γίνεται ένας έλεγχος με τα αποτελέσματα που ανέκτησε από την γενικότερη αναζήτηση. Αν δεν υπάρχουν όμοια προγράμματα τα εισάγει στην λίστα, κάνοντας χρήση του API και ενημερώνει τον χρήστη για τον αριθμό των προγραμμάτων που εισήχθησαν στην λίστα του με status ToBeRecorded. Αν στη λίστα βρεθούν κάποια προγράμματα από αυτά της αναζήτησης και το status τους δεν είναι Recorded ή ToBeRecorded γίνεται, πάλι μέσω του API ένα update στη λίστα με status ToBeRecorded και για όσα δεν υπήρχαν στην λίστα εισάγονται στη λίστα με status ToBeRecorded. Ο χρήστης ενημερώνεται κατάλληλα.

Target: -

Όταν δεν υπάρχει πληροφορία για το target από τον χρήστη, θεωρούμε ότι εννοεί την εξατομικευμένη του λίστα (selection list). Έτσι, ακολουθείται η ίδια διαδικασία με αυτήν που το target είναι list.

Action: Delete**Target: Profile**

Μία τέτοια ενέργεια δε μπορεί να υποστηριχθεί από το παρόν σύστημα, δηλαδή να διαγραφεί μερική πληροφορία από το προφίλ του χρήστη, οπότε ένα μήνυμα ενημερώνει κατάλληλα τον χρήστη.

Target: List

Ο διαχειριστής αποκρίσεων εισάγει στη βάση, μέσω του XML-DB Middleware, το XML έγγραφο με την πληροφορία συμβατή με το UserIdentifier DS και το FilteringAndSearchPreferences DS με προσωρινά στοιχεία του χρήστη, ώστε να μη σβήσει κάποιο ήδη υπάρχον προφίλ του και κάνει ανάκτηση των μεταδεδομένων που ταιριάζουν με την πληροφορία που έχει εισάγει ο χρήστης. Έπειτα χρησιμοποιεί την πληροφορία αυτή και κάνοντας χρήση του Java API διαγράφει όλα τα μεταδεδομένα των προγραμμάτων της εξατομικευμένης λίστας (selection list) του χρήστη που ταιριάζουν με τα αποτελέσματα της αναζήτησης και έχουν status Undefined ή ToBeRecorded. Ο χρήστης ενημερώνεται κατάλληλα.

Target: PDR

Ο διαχειριστής αποκρίσεων εισάγει στη βάση, μέσω του XML-DB Middleware, το XML έγγραφο με την πληροφορία συμβατή με το UserIdentifier DS και το FilteringAndSearchPreferences DS με προσωρινά στοιχεία του χρήστη, ώστε να μη σβήσει κάποιο ήδη υπάρχον προφίλ του και κάνει ανάκτηση των μεταδεδομένων που ταιριάζουν με την πληροφορία που έχει εισάγει ο χρήστης. Έπειτα χρησιμοποιεί την πληροφορία αυτή και κάνοντας χρήση του Java API θέτει το status των αποθηκευμένων προγραμμάτων της εξατομικευμένης λίστας (selection list) του χρήστη που ταιριάζουν με τα αποτελέσματα της αναζήτησης σε ToBeDeleted. Ο χρήστης ενημερώνεται κατάλληλα.

Target: -

Ο διαχειριστής διαλόγου εκτελεί σε συνδυασμό τις δύο παραπάνω ενέργειες, αυτές δηλαδή για το target ίσο με List και το target ίσο με PDR.

Action: Retrieve**Target: Profile**

Επειδή από το σύστημα, τα μεταδεδομένα των προγραμμάτων που ταιριάζουν με το προφίλ του χρήστη αποθηκεύονται στο selection list του αυτό που γίνεται σε αυτήν την περίπτωση είναι να ανακτηθούν τα αποτελέσματα από την πληροφορία του χρήστη, όπως και προηγουμένως και όσα από τα προγράμματα ταιριάζουν με αυτά που περιέχονται στο selection list του χρήστη και έχουν status Undefined ή ToBeRecorded, παρουσιάζονται σε αυτόν με το κατάλληλο μήνυμα. Σε περίπτωση που η αναζήτηση έχει γίνει με χρήση μόνο μίας κατηγορίας TVA και δεν βρέθηκαν αποτελέσματα, τότε ο χρήστης λαμβάνει ένα μήνυμα τύπου 2.

Target: List

Ακολουθείται ακριβώς η ίδια διαδικασία με την προηγούμενη περίπτωση, όπου το target είναι το Profile.

Target: PDR

Ακολουθείται ακριβώς η ίδια διαδικασία με την προηγούμενη περίπτωση, όπου το target είναι το Profile και το status των προγραμμάτων είναι Recorded.

Target: -

Ο διαχειριστής αποκρίσεων εισάγει στη βάση, μέσω του XML-DB Middleware, το XML έγγραφο με την πληροφορία συμβατή με το UserIdentifier DS και το FilteringAndSearchPreferences DS με προσωρινά στοιχεία του χρήστη, ώστε να μη σβήσει κάποιο ήδη υπάρχον προφίλ του και κάνει ανάκτηση των μεταδεδομένων που ταιριάζουν με την πληροφορία που έχει εισάγει ο χρήστης. Έπειτα, αν υπάρχουν, παρουσιάζει τα αποτελέσματα στον χρήστη με το κατάλληλο μήνυμα. Σε περίπτωση που η αναζήτηση έχει γίνει με χρήση μόνο μίας κατηγορίας TVA και δεν βρέθηκαν αποτελέσματα, τότε ο χρήστης λαμβάνει ένα μήνυμα τύπου 2.

Action: Profile

Target: Profile

Αναφέρεται σε εισαγωγικές φράσεις, όπως 'I prefer', 'I like'. Εδώ έχουμε την ίδια αντιμετώπιση με την περίπτωση όπου το action είναι Insert και το target ίσο με Profile.

Target: List

Ίδια αντιμετώπιση πάλι με την περίπτωση όπου το action είναι Insert και το target ίσο με List.

Target: PDR

Ίδια αντιμετώπιση πάλι με την περίπτωση όπου το action είναι Insert και το target ίσο με PDR.

Target: -

Ίδια αντιμετώπιση με την περίπτωση όπου το action είναι Insert και το target ίσο με Profile.

Όπως είχαμε τονίσει και στην ανάλυση του διαχειριστή διαλόγου, όταν η τιμή που έχει αποδοθεί σε μία κατηγορία TVA από τον χρήστη δεν υπάρχει στη βάση, τότε η πληροφορία αυτή περνάει αμέσως στον διαχειριστή αποκρίσεων ο οποίος τροφοδοτεί τον χρήστη με μήνυμα τύπου 1.

Στο σημείο αυτό θα πρέπει να αναλύσουμε λίγο πως ο διαχειριστής αποκρίσεων διαχειρίζεται και ταξινομεί τα αποτελέσματα που προκύπτουν από την αναζήτηση και τα παρουσιάζει στον χρήστη. Όπως είπαμε και στην αρχή ο διαχειριστής διαλόγου τροφοδοτεί, όταν υπάρχει, τον διαχειριστή αποκρίσεων με το έγγραφο εκείνο που περιέχει την πληροφορία που εισήγαγε ο χρήστης και ταιριάζει μερικώς με το προφίλ του. Αν αυτό δεν υπάρχει και έχουμε αποτελέσματα μόνο από την αναζήτηση, τότε στηριζόμαστε στην επεξεργασία που έχει ήδη γίνει από τον διαχειριστή διαλόγου και εμφανίζουμε τα αποτελέσματα. Αν υπάρχει έγγραφο τότε η διαδικασία είναι η εξής. Ο διαχειριστής αποκρίσεων εισάγει στη βάση, μέσω του XML-DB Middleware, το XML έγγραφο που προέρχεται από το προφίλ με προσωρινά στοιχεία του χρήστη, ώστε να μη σβήσει κάποιο ήδη υπάρχον προφίλ του και κάνει ανάκτηση των μεταδεδομένων που ταιριάζουν με την πληροφορία που έχει εισάγει ο χρήστης. Έπειτα εισάγει το έγγραφο με την πληροφορία του χρήστη. Τα αποτελέσματα που προέρχονται από το προφίλ και ταιριάζουν με μερίδα των αποτελεσμάτων της γενικής αναζήτησης παρουσιάζονται πρώτα και ακολουθούν αυτά που περισεύουν.

5.6 System Flow : Δύο παραδείγματα

Για να συνοψίσουμε το κεφάλαιο αυτό θα παρουσιάσουμε δύο παραδείγματα και θα δούμε αναλυτικά τις φάσεις από τις οποίες περνάει η εισαγωγή του χρήστη μέσα από τις διάφορες υπο-μονάδες τις οποίες περιγράψαμε και πως παρουσιάζεται στο τέλος στον χρήστη.

Παράδειγμα 1

-Ας θεωρήσουμε ότι ο χρήστης εισάγει στο σύστημα μια ερώτηση για το αν υπάρχουν ταινίες με τον ηθοποιό Frankie Muniz που προβάλλονται μετά τις 5 το απόγευμα. Ο χρήστης πληκτρολογεί την παρακάτω πρόταση.

‘Are there any movies with Frankie Muniz after 5 pm’

Εισάγεται η πρόταση στην υπο-μονάδα ανάλυσης διαγραμμάτων από την οποία εξέρχεται η παρακάτω δομή ιδιοτήτων που περνά στην υπο-μονάδα διαχείρισης διαλόγου.

```
[[ properties: [ keyword: [ 1: Muniz
                        [ 0: Frankie]]
               [ genre1: movies]]
 [ temporal: [ timeind: pm]
               [ after: [ time : 5 ]]]
 [ markers: [ act: ret ]]]
```

Ο διαχειριστής διαλόγου αναγνωρίζει τιμή στο πεδίο keyword και κάνει τα εξής βήματα. Ελέγχει αν υπάρχει and ή or ανάμεσα στις λέξεις και βλέπει ότι δεν υπάρχει. Δημιουργεί ένα διάνυσμα που περιέχει τις τιμές του keyword. Τις περνά από την stop list η οποία δεν κόβει καμμία λέξη. Καλεί τον επιλυτή ασαφειών και κάνει χρήση του TVA Semantics Resolver ο οποίος για κάθε μία λέξη επιστρέφει ένα διάνυσμα που περιέχει ένα σύνολο από διανύσματα με την εξής πληροφορία.

```
[[ACTOR, Frankie Muniz]]
```

Η δομή που δημιουργείται και περνά στον διαχειριστή αποκρίσεων, μαζί με τα στοιχεία του χρήστη, είναι της μορφής

String action = retrieval **String target** = null XML doc

όπου το XML doc είναι το παρακάτω:

```
<?xml version="1.0" encoding="UTF-8"?>
<tva:TVAMain xmlns:tva="http://www.tv-anytime.org/2001/08/metadata"
xmlns:xm1="http://www.w3.org/XML/1998/namespace" xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001">
<tva:UserDescription>
  <tva:UserPreferences>
    <mpeg7:UserIdentifier>
      <mpeg7:Name>__temp_faspmatching_</mpeg7:Name>
    </mpeg7:UserIdentifier>
    <mpeg7:FilteringAndSearchPreferences preferenceValue="100">
      <mpeg7:CreationPreferences preferenceValue="100">
        <mpeg7:Creator preferenceValue="100">
          <mpeg7:Role href="urn:mpeg:MPEG7RoleCS:ACTOR" />
          <mpeg7:Agent xsi:type="mpeg7:PersonType">
            <mpeg7:Name>
              <mpeg7:GivenName>FrankieMuniz</mpeg7:GivenName>
            </mpeg7:Name>
          </mpeg7:Agent>
        </mpeg7:Creator>
        <mpeg7:DatePeriod preferenceValue="100">
          <mpeg7:TimePoint>2003-08-21T17:00:00</mpeg7:TimePoint>
          <mpeg7:Duration></mpeg7:Duration>
        </mpeg7:DatePeriod>
      </mpeg7:CreationPreferences>
      <mpeg7:ClassificationPreferences preferenceValue="100">
        <mpeg7:Genre href="urn:mpeg:TVAnytime_v0.1ContentCS:6" preferenceValue="100">
          <mpeg7:Name>Movies</mpeg7:Name>
          <mpeg7:Definition>Film, Cinema</mpeg7:Definition>
        </mpeg7:Genre>
      </mpeg7:ClassificationPreferences>
      <mpeg7:SourcePreferences preferenceValue="100" />
    </mpeg7:FilteringAndSearchPreferences>
  </tva:UserPreferences>
</tva:UserDescription>
</tva:TVAMain>
```

Σχήμα 10: XML έγγραφο που σχηματίζεται από τον διαχειριστή διαλόγου με την πληροφορία από την εισαγωγή του χρήστη

Ας θεωρήσουμε στο παρόν παράδειγμα ότι από το φιλτράρισμα που γίνεται από τον διαχειριστή διαλόγου στο προφίλ του χρήστη, δεν έχουμε παραγόμενο έγγραφο, που σημαίνει ότι δεν υπήρχε αντίστοιχη πληροφορία στο προφίλ του χρήστη.

Όταν ο διαχειριστής αποκρίσεων λάβει τη δομή ελέγχει πρώτα τα πεδία action και target. Εφόσον είναι λειτουργία ανάκτησης χωρίς στόχο, σημαίνει ότι η αναζήτηση πρέπει να γίνει στο σύνολο των μεταδεδομένων μέσα στο TVA server. Έτσι έχουμε την εισαγωγή του εγγράφου στην βάση μέσω του XML-BD Middleware και με προσωρινή τιμή στο πεδίο Name του UserIdentifier. Ένας μηχανισμός matching που παρέχεται από το σύστημα, καλείται από τον διαχειριστή αποκρίσεων, που ταιριάζει την πληροφορία με τα μεταδεδομένα και επιστρέφει το σύνολο των προγραμμάτων, με ένα βάρος που υποδηλώνει το πόσο σχετικό είναι το αποτέλεσμα με την πληροφορία. Ο διαχειριστής αποκρίσεων από την λίστα αυτή παίρνει τόσα προγράμματα όσα είχε δηλώσει ο χρήστης στην αρχή της εφαρμογής. Στο παρόν παράδειγμα, επιστρέφει όλα τα αποτελέσματα.

Οι πληροφορίες που ανακτούμε σε πρώτη φάση αφορούν στο crid του προγράμματος (το αποτέλεσμα της διαδικασίας αναζήτησης και επιλογής που αναφέρεται μοναδικά σε ένα κομμάτι του υλικού), τον τίτλο του, ένας βαθμός (rank) που υποδηλώνει τη σχετικότητα του ανάλογα με τις παραμέτρους της αναζήτησης και το status με το οποίο μπορεί το πρόγραμμα να περιέχεται στο selection list του χρήστη.

```
crid: crid://www.pps.de/213-0-1133697
Title: Mit den Augen einer Toten
Rank: 20
Status: undefined
```

Το μήνυμα που θα συνοδέψει την πληροφορία μαζί με τη λίστα από τους τίτλους των προγραμμάτων που παρουσιάζονται στον χρήστη ακολουθεί:

```
The results of your search are:
Mit den Augen einer Toten
```

Παράδειγμα 2

-Αν ο χρήστης θέλει να δει τα περιεχόμενα της εξατομικευμένης λίστας του (selection list) τότε θα μπορούσε να εισάγει στο σύστημα την παρακάτω πρόταση:

‘Give me the contents of my selection list’

Εισάγεται η πρόταση στην υπο-μονάδα ανάλυσης διαγραμμάτων από την οποία εξέρχεται η παρακάτω δομή ιδιοτήτων που περνά στην υπο-μονάδα διαχείρισης διαλόγου.

```
[ [ markers : [ act : ret ]
      [ target : list ] ] ]
```

Ο διαχειριστής διαλόγου ελέγχει αν υπάρχει το entity properties στην δομή ιδιοτήτων.

Εφόσον δεν υπάρχει δημιουργείται η παρακάτω δομή, που περνά στον διαχειριστή αποκρίσεων, μαζί με τα στοιχεία του χρήστη:

String action = retrieval **String target** = list null

Ο διαχειριστής αποκρίσεων καλεί την αντίστοιχη συνάρτηση από το Java API για την ανάκτηση του selection list, που του επιστρέφει τα παρακάτω αποτελέσματα σε μορφή πίνακα.

Οι πληροφορίες που αναζητούμε σε πρώτη φάση αφορούν στο crid του προγράμματος, τον τίτλο του, το rank και το status με το οποίο είναι στο selection list.

```
crid: crid://www.pps.de/213-0-1133697
Title: Mit den Augen einer Toten
Rank: 20
Status: undefined
```

```
crid: crid://www.pps.de/213-0-3577331
Title: Friends
Rank: 20
Status: undefined
```

```
crid: crid://www.pps.de/213-0-7988949
Title: Terminator 2
Rank: 20
Status: undefined
```

```
crid: crid://www.pps.de/213-0-8036521
Title: Matrix Reloaded
Rank: 20
Status: undefined
```

Το μήνυμα που θα συνοδεύει την πληροφορία μαζί με τη λίστα από τους τίτλους των προγραμμάτων που παρουσιάζονται στον χρήστη ακολουθεί:

The contents of your selection list are :

Mit den Augen einer Toten
Friends
Terminator 2
Matrix Reloaded

Ο λόγος που σε πρώτη φάση παρουσιάζουμε μόνο τους τίτλους των προγραμμάτων δεν είναι άλλος από το ότι θα παρουσιάσουμε τα αποτελέσματα στην περιορισμένη οθόνη των κινητών συσκευών, που όμως μέσα από την εφαρμογή παρέχεται η λειτουργικότητα της ανάκτησης όλης της πληροφορίας που αφορά στα προγράμματα αυτά.

Ανακεφαλαίωση

Μέσα από την ανάλυση της υλοποίησης του συστήματος, γίνεται φανερός ο τρόπος με τον οποίο επικοινωνούν όλες οι υπο-μονάδες και ο τρόπος που μετασχηματίζεται η πληροφορία ανάλογα με τις ανάγκες του εισαγωγής του χρήστη. Ανάλογα με τη δομή της πληροφορίας, αναλύονται όλοι οι τρόποι με τους οποίους αντιμετωπίζονται τα προβλήματα που προκύπτουν, όπως στην περίπτωση με την εξαγωγή δύο ή περισσότερων δομών ιδιοτήτων από τον αναλυτή διαγραμμάτων (Chart Parser), ενώ αναλυτικά παρουσιάστηκε και ο βασικός αλγόριθμος και η στρατηγική που ακολουθείται για την καλύτερη αντιμετώπιση των ασαφειών της εισαγωγής του χρήστη. Θα πρέπει να επισημανθεί ότι ανεξάρτητα από την ασαφή εισαγωγή του χρήστη, λόγω της χρήσης οντολογιών για την ανάκτηση δεδομένων υπάρχουν πάντα αποτελέσματα προς διαχείριση ικανοποιώντας τις απαιτήσεις του χρήστη. Στο τέλος παρουσιάστηκε ο τρόπος με τον οποίο το σύστημα απαντά στον χρήστη, ανάλογα με την ορθότητα ή όχι των ερωτήσεών του και η επιλογή που γίνεται στην ταξινόμηση των αποτελεσμάτων βάση των γενικότερων προτιμήσεων του χρήστη, που έχουν ήδη δηλωθεί στο προφίλ του. Στο επόμενο κεφάλαιο θα δούμε πως εφαρμόζεται το σύστημα σε περιβάλλον κινητών συσκευών.

Κεφάλαιο 6

Σχεδιασμός και Υλοποίηση εφαρμογής

Το μοντέλο της φυσικής γλώσσας που αναπτύχθηκε στα πλαίσια της διπλωματικής αυτής εργασίας μπορεί να έχει διάφορες εφαρμογές και σε διαφορετικές πλατφόρμες. Η εφαρμογή όμως, σε κινητές συσκευές αποδεικνύεται ιδιαίτερα χρήσιμη, λόγω της έλλειψης χώρου και πόρων των συσκευών αυτών. Έτσι, ένα απαιτητικό σε μέγεθος γραφικό περιβάλλον για ψηφιακή τηλεόραση μπορεί και συνοψίζεται στις ελάχιστες προδιαγραφές κινητών συσκευών μέσω ενός συστήματος φυσικής γλώσσας. Παρακάτω, παρουσιάζεται η τεχνολογία που χρησιμοποιήθηκε για την ανάπτυξη της εφαρμογής, η αρχιτεκτονική και η υλοποίηση, η λειτουργικότητα που παρέχεται και τέλος ο γραφικός σχεδιασμός της εφαρμογής.

6.1 Τεχνολογία Java για κινητές συσκευές

Όπως έχουμε αναφέρει σε πολλά σημεία της εργασίας αυτής, η εφαρμογή που επιλέξαμε να συνδέσει τον τελικό χρήστη με το υπόλοιπο σύστημα, που του προσφέρει την συνδιαλλαγή του μέσω φυσικής γλώσσας, αφορά σε κινητές συσκευές. Με την εισαγωγή της πλατφόρμας J2ME, ο ρόλος των κινητών ασύρματων συσκευών αναβαθμίστηκε από συσκευές φωνητικής επικοινωνίας με περιορισμένη λειτουργικότητα, σε συσκευές με επεκτάσιμη λειτουργικότητα σε υπηρεσίες προσφερόμενες από το internet. Με την χρήση της τεχνολογίας Java σε κινητές συσκευές, μπορούμε μέσα από την εφαρμογή να παραδώσουμε δυναμικά περιεχόμενο, ενώ έχουμε ικανοποιητική αλληλεπίδραση με τον χρήστη. Επίσης, διαφυλάσσουμε την συμβατότητα ανάμεσα στις πλατφόρμες που χρησιμοποιούνται από το σύστημα.

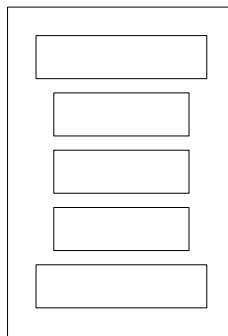
Από το J2ME περιβάλλον ορίζονται δύο θεμελιώδεις αρχές πάνω στις οποίες στηρίχθηκε η υλοποίηση της εφαρμογής :

- **Configuration:** Το configuration προσδιορίζει τα χαρακτηριστικά της γλώσσας προγραμματισμού Java, που υποστηρίζεται, της εικονικής μηχανής (virtual machine) και των βασικών βιβλιοθηκών και σύνολο από APIs. Το CLDC (Connected, Limited Device Configuration) αφορά σε low-end συσκευές, όπως κινητά τηλέφωνα, organizers και PDAs. Αυτό το configuration περιλαμβάνει κάποιες νέες κλάσεις, που

δεν προέρχονται από το J2SE API, αλλά είναι σχεδιασμένες αποκλειστικά για να ικανοποιούν τις ανάγκες μικρών, σε μέγεθος, συσκευών.

- **Profile:** Το profile εξυπηρετεί, κυρίως, στο να παρέχει ένα ολοκληρωμένο εργαλείο για την υλοποίηση των εφαρμογών σε συγκεκριμένες συσκευές, όπως τα κινητά τηλέφωνα. Το MIDP (Mobile Information Device Profile) είναι μία αρχιτεκτονική και ένα σύνολο από Java βιβλιοθήκες, που δημιουργούν ένα ανοιχτό, third-party περιβάλλον ανάπτυξης εφαρμογών για μικρές, περιορισμένες από πηγή λειτουργίας κινητές συσκευές πληροφορίας, τις MIDs. Επίσης, θα πρέπει να επισημάνουμε ότι η βασική μονάδα εκτέλεσης ενός MIDP, είναι η κλάση MIDlet.

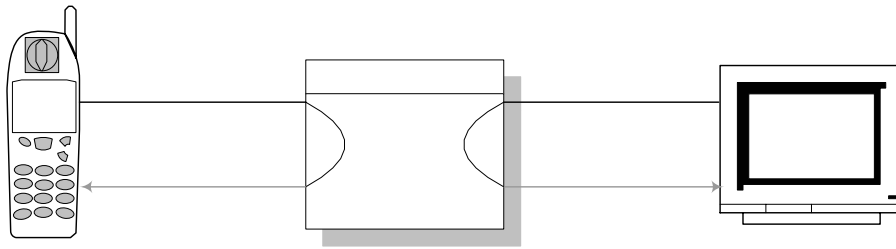
Συνοψίζοντας, στο παρακάτω σχήμα βλέπουμε πως τα παραπάνω ιεραρχούνται σε μία εφαρμογή.



Σχήμα 23: Επίπεδα λογισμικού κατά την εφαρμογή στην πλατφόρμα J2ME

6.1 Αρχιτεκτονική και Υλοποίηση της σχεδίασης της εφαρμογής

Η εφαρμογή είναι βασισμένη στην αρχιτεκτονική client-server, όπως φαίνεται και στο παρακάτω σχήμα.

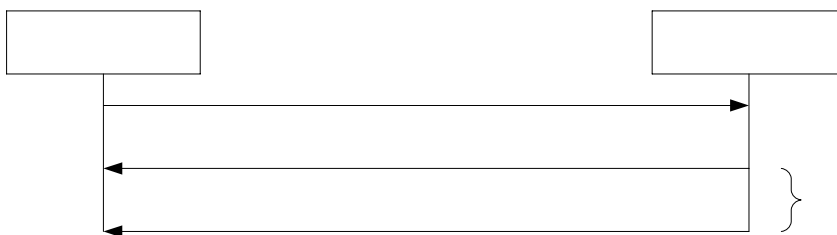


Σχήμα 24: Η αρχιτεκτονική client-server

Στην παρούσα αρχιτεκτονική βλέπουμε έναν client με κινητό τηλέφωνο και έναν server βασισμένο σε Java Servlet, που παίζει το ρόλο της διαπαφής ανάμεσα στα χρήστες και στον UP-TV database server ή στο PDR.

- *UIMIDlet*, ο client τρέχει σε ένα MIDP-enabled κινητό τηλέφωνο. Παρέχει ένα γραφικό σχεδιασμό (user interface) στον χρήστη, ώστε να συνδέεται, μέσω του πρωτοκόλλου του λογαριασμού στο PDR του ή σε κάποιον συγκεκριμένο UP-TV server.
- *MobileServlet*, ο server βασισμένος σε Java Servlet τρέχει σε μία μηχανή βασισμένη στο HTTP πρωτόκολλο, όπως ο Apache's Jakarta Tomcat.

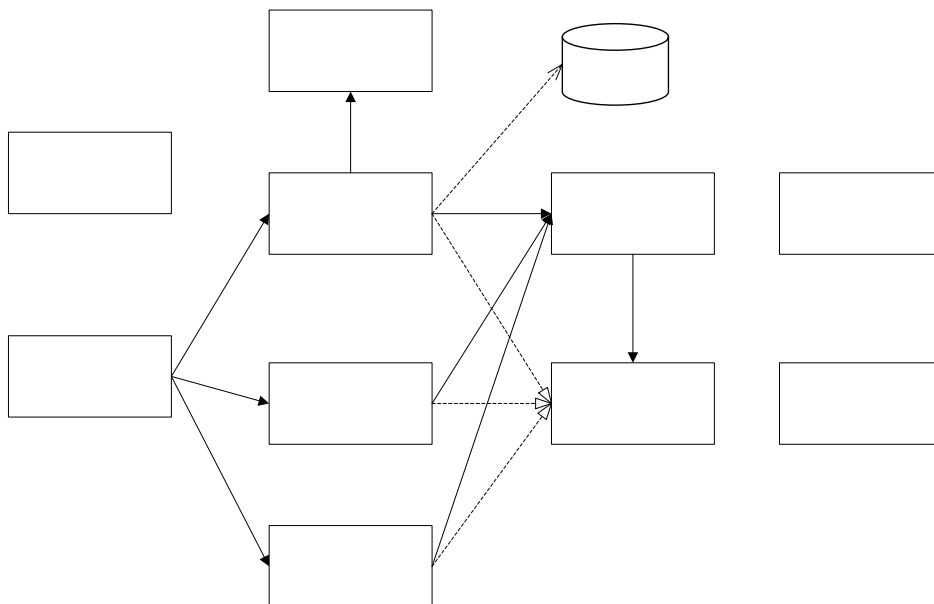
Το πρωτόκολλο της εφαρμογής που χρησιμοποιείται από το MobileServlet είναι ένα τυπικό πρωτόκολλο, υλοποιημένο πάνω στο HTTP. Η επικοινωνία ανάμεσα στο MIDlet και στο Servlet, γίνεται με μηνύματα που μεταφέρονται στο σώμα των HTTP αιτήσεων και απαντήσεων. Αυτό το σώμα έχει τη μορφή ενός byte stream, οπότε το MIDlet και το Servlet πρέπει να συμφωνούν στον τρόπο διερμηνείας αυτών των bytes. Η μορφή του μηνύματος που χρησιμοποιείται, είναι τέτοια ώστε να τελειώνει κάθε μέρος του μηνύματος με έναν χαρακτήρα νέας γραμμής. Για παράδειγμα, μια αλληλουχία μηνυμάτων έχει την παρακάτω μορφή.



Σχήμα 25: Παράδειγμα ακολουθίας μηνυμάτων σε περίπτωση εισαγωγής του χρήστη στο σύστημα με χρήση του προσωπικού του λογαριασμού

Όταν ξεκινά το MIDlet και ο χρήστης εισάγει το όνομά του (username) και τον κωδικό του (password) στην αντίστοιχη οθόνη, το MIDlet στέλνει μία αίτηση login στο servlet, το οποίο επιβεβαιώνει τα δύο στοιχεία αυτά με ένα μήνυμα απόκρισης, είτε login-OK για επιτυχία είτε login-error για αποτυχία.

Η εφαρμογή, όπως φαίνεται και στο παρακάτω σχήμα, χρησιμοποιεί το MIDP Record Store API (RMS) για την αποθήκευση πληροφορίας, όπως τα στοιχεία του λογαριασμού του χρήστη. Στην συγκεκριμένη εφαρμογή το MIDlet δεν αποθηκεύει πολλαπλούς λογαριασμούς χρηστών. Οι κλάσεις που αναλαμβάνουν να πιάσουν αυτή την πληροφορία έχουν υλοποιηθεί βάση του Singleton design pattern.



Σχήμα 26: Ο σχεδιασμός της υλοποίησης της εφαρμογής

Το UIMIDlet χρησιμοποιεί μία διαδικασία βασισμένη σε sockets για να επικοινωνήσει με το MobileServlet, δημιουργώντας ένα αντικείμενο HttpP και χρησιμοποιώντας το σαν αναφορά πίσω στον εαυτό του (callbacks). Ένα αντικείμενο HttpP, δημιουργημένο από το MIDlet περιέχεται σαν παράμετρος στον constructor όλων των οθονών, εκτός της WelcomeScreen. Κάθε οθόνη καλεί το αντικείμενο HttpP, ώστε να στείλει μία αίτηση στο servlet και κάθε οθόνη περνά, επίσης, στο αντικείμενο HttpP μία αναφορά πίσω στον εαυτό του, με τον HttpPListener. Αυτό επιτρέπει στο αντικείμενο HttpP να χρησιμοποιεί τα callbacks όταν

φθάνει η απόκριση. Όλες οι οθόνες που χρησιμοποιούν το αντικείμενο HttpP, χρησιμοποιούν και ένα αντικείμενο VGenerator για να δημιουργήσουν τις αιτήσεις με το πρωτόκολλο που δείξαμε προηγουμένως, και να τις στείλουν στο servlet. Επίσης, χρησιμοποιούν ένα αντικείμενο VParser, ώστε να αναλύσουν τις αποκρίσεις του servlet.

6.2 Λειτουργικότητα

Ο παρακάτω πίνακας περιγράφει την λειτουργικότητα που παρέχεται από την εφαρμογή.

Connect to	My Box	Σύνδεση χρήστη με το PDR του μέσω ενός συγκεκριμένου url
	A Remote Server	Σύνδεση χρήστη με ένα PDR μέσω του αντίστοιχου url
Login	Username	Γαυτοποίηση (login) του χρήστη. Μετά την πρώτη εισαγωγή του χρήστη στο σύστημα το username κρατείται από το Record Store.
	Password	
General Preferences	Language	Ο χρήστης μπορεί να θέσει τη γλώσσα επικοινωνίας με το σύστημα, διαλέγοντας μέσα από μία λίστα.
	Set Max Return Results	Ο χρήστης θέτει τον επιθυμητό αριθμό αποτελεσμάτων που θα του επιστρέψει το σύστημα.
Text Input		Ο χρήστης εισάγει μία έκφραση στη φυσική του γλώσσα (Αγγλικά).
Results		Επιστρέφεται στον χρήστη ένα μήνυμα και τα αποτελέσματα, αν αυτά υπάρχουν.
Get Program Details		Επιστρέφονται λεπτομέρειες για ένα πρόγραμμα, όπως σύνοψη, ημέρα και ώρα μετάδοσης, διάρκεια, κανάλι μετάδοσης.

6.3 Γραφικός Σχεδιασμός

Σε αυτό το σημείο θα δώσουμε τις λεπτομέρειες που αφορούν στον γραφικό σχεδιασμό της εφαρμογής. Αρχικά θα αναφέρουμε κάποιες από τις βασικές αρχές σχεδιασμού, τις οποίες λάβαμε υπόψη μας κατά τον σχεδιασμό και θα παραθέσουμε και κάποια στιγμιότυπα από συγκεκριμένη παρεχόμενη υπηρεσία.

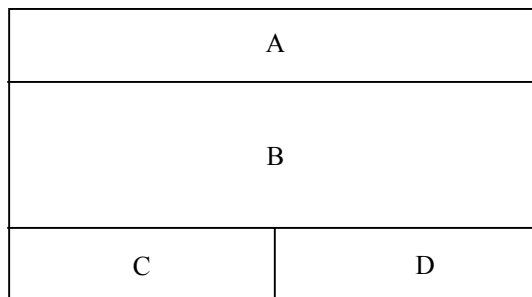
Γραφικός σχεδιασμός συσκευής κινητού τηλεφώνου

Συγκεκριμένες βασικές αρχές σχεδιασμού για κινητές συσκευές και ειδικότερα για κινητά τηλέφωνα, χρησιμοποιήθηκαν και αναφέρονται παρακάτω:

- **Keep soft key labels to 7 characters or less:** Πολλές συσκευές δεν μπορούν να απεικονίσουν στην οθόνη τους ονόματα για τα κουμπιά τους πάνω από 7 χαρακτήρες, με αποτέλεσμα να περικλύβουν τους παραπάνω χαρακτήρες ή να συντομεύουν τα ονόματα με αυθαίρετο τρόπο.
- **Keep the content that appears above select and input fields to 1 or 2 lines max:** Περιεχόμενο που ξεπερνά τις 2 γραμμές μπορεί να κοπεί από μικρότερες συσκευές, όταν χρησιμοποιείται σε συνδυασμό με πεδία για εισαγωγή κειμένου ή πεδία για επιλογή. Αυτό μπορεί να οδηγήσει σε απώλεια πολύτιμης πληροφορίας.
- **Reliability:** Οι οθόνες πρέπει να δουλεύουν μαζί. Για παράδειγμα, σε μία εφαρμογή όταν πληροφορία επιστρέφεται σε μία οθόνη, θα πρέπει να μπορεί να συνεχιστεί η πλοήγηση του χρήστη στις επόμενες οθόνες χωρίς να χρειάζεται να επιστρέψει στην συγκεκριμένη οθόνη για να πάρει ξανά την πληροφορία. Κάτι τέτοιο είναι αναποτελεσματικό και πολύπλοκο για έναν χρήστη να χρειάζεται να εισάγει από μόνος του πληροφορία που η εφαρμογή θα έπρεπε να είχε κρατήσει από μόνη της. Οι χρήστες θα πρέπει να μπορούν να προβλέψουν τι θα γίνει στη συνέχεια όταν κάνουν κάποια ενέργεια, γιατί σπάνια κοιτούν στις οδηγίες χρήσης κάποιας εφαρμογής.
- **Functionality Versus Simplicity:** Χρησιμοποιείται ο κανόνας 80/20, δηλαδή προσδιόρισε το 20% της λειτουργικότητας που συναντά το 80% των αναγνών του χρήστη και βελτιστοποίησε ανάλογα τον σχεδιασμό. Αφού το 20% έχει υποστηριχθεί μπορεί μετά να αποφασιστεί, αν κάποιο από το υπόλοιπο 80% της λειτουργικότητας θα υποστηριχθεί, βάση άλλων κριτηρίων. Μην αλλάζεις τον σχεδιασμό ώστε να παρέχεις τον ίδιο βαθμό προσπέλασης σε αυτές τις λειτουργίες πάνω από το κρίσιμο 20%.
- **Make Everything Interruptible:** Πρέπει να παρέχεται στον χρήστη η δυνατότητα να μπορεί να διακόπτει την εφαρμογή, ανεξάρτητα σε ποια κατάσταση βρίσκεται.

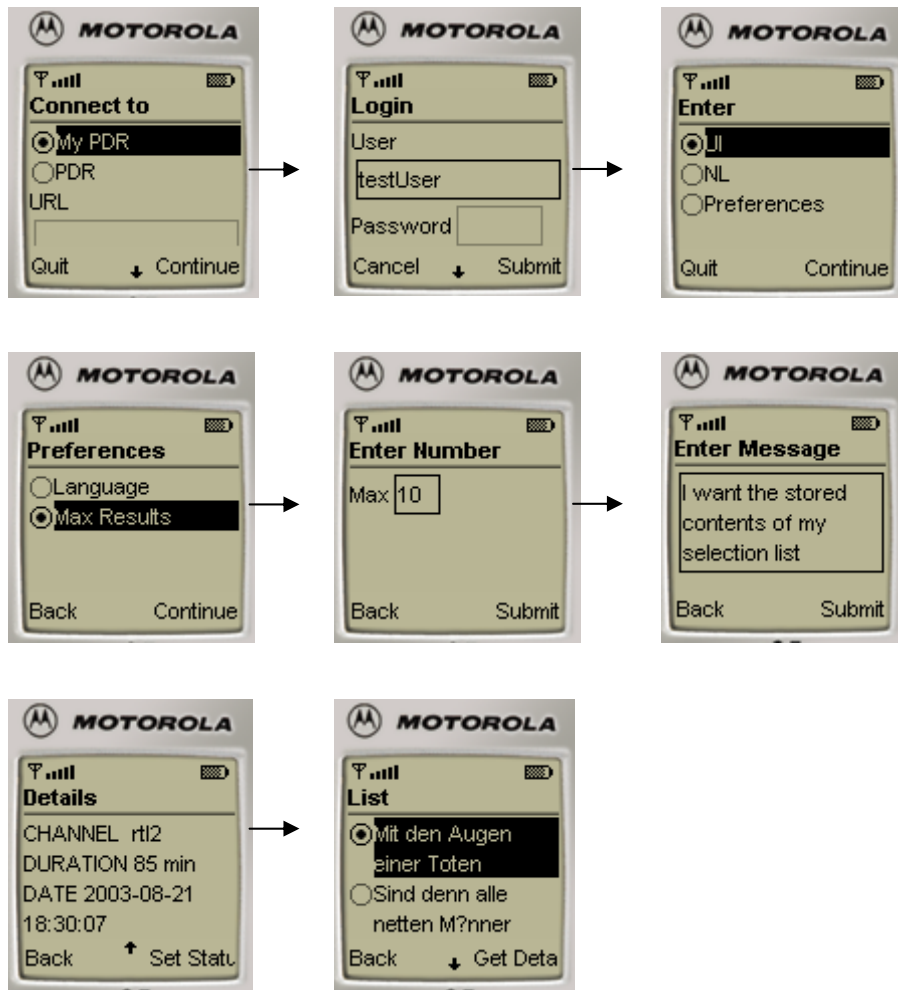
- **Implications of the MIDP Environment:** Δημιούργησε και σχεδίασε τα συνιστώμενα μέρη του MIDP γραφικού σχεδιασμού με τέτοιο τρόπο ώστε τα MIDlets να μοιάζουν σαν φυσικές εφαρμογές. Για παράδειγμα, ας υποθέσουμε ότι μία συσκευή έχει έναν συγκεκριμένο τρόπο να παριστά μία λίστα στον χρήστη. Ο MIDP υλοποιητής πρέπει να διασφαλίσει ότι η MIDP υλοποίηση παριστά τη λίστα με τον ίδιο τρόπο (μέσα στα όρια του MIDP specification).

Το επόμενο σχήμα παριστά την σχεδίαση του γραφικού σχεδιασμού για την προσπέλαση από απόσταση. Η εφαρμογή χρησιμοποιεί έναν ομοιόμορφο σχεδιασμό για όλες τις οθόνες, στις οποίες παριστάνεται η λειτουργικότητα που αναφέραμε πιο πάνω. Κάθε οθόνη χωρίζεται σε 4 μέρη. Το μέρος A χρησιμοποιείται για τον τίτλο της τρέχουσας οθόνης, το μέρος B για την παρουσίαση των περιεχομένων που σχετίζονται με την λειτουργικότητα που προσφέρεται από την τρέχουσα οθόνη, το μέρος C για εντολές πλοήγησης ('continue', 'back', 'quit') και το μέρος D περιέχει εντολές για την επιλογή του περιεχομένου της οθόνης, που περιέχεται στο μέρος B. Για παράδειγμα, αν στο μέρος B έχουμε μία λίστα, τότε το μέρος D είναι η εντολή select.



Σχήμα 27: Ο γραφικός σχεδιασμός της οθόνης του κινητού τηλεφώνου

Ας θεωρήσουμε το παρακάτω σενάριο: Ένας χρήστης που ονομάζεται testUser χρησιμοποιεί το κινητό του τηλέφωνο που είναι συμβατό με Java και θέλει να δει τα προγράμματα που περιέχονται στο selection list του στο PDR του σπιτιού του. Στην πρώτη οθόνη ο χρήστης συνδέεται στο PDR του σπιτιού του. Στην δεύτερη οθόνη ο χρήστης παρέχει τις πληροφορίες του προσωπικού του λογαριασμού. Στις επόμενες τρεις οθόνες παρέχει πληροφορίες για τις προτιμήσεις του πάνω στον μέγιστο αριθμό των αποτελεσμάτων που θα του επιστρέψει το σύστημα. Στην έκτη οθόνη ο χρήστης εισάγει το μήνυμά του για να πάρει τα περιεχόμενα του selection list του. Στην έβδομη οθόνη φαίνονται τα περιεχόμενα της λίστας αυτή.



Σχήμα 28: Ένα παράδειγμα μέσα από στιγμιότυπα της εφαρμογής σε ένα κινητό τηλέφωνο

Ανακεφαλαίωση

Η επικοινωνία του χρήστη με το σύστημα φυσικής γλώσσας, γίνεται μέσω κινητών συσκευών και συγκεκριμένα κινητών τηλεφώνων, συμβατών με το MIDP προφίλ της J2ME πλατφόρμας. Έγινε μία παρουσίαση της αλληλεπίδρασης της εφαρμογής με το υπόλοιπο σύστημα και πως σχεδιάστηκε αυτή ώστε να ικανοποιεί τη λειτουργικότητα που παρέχει το σύστημα φυσικής γλώσσας στο χρήστη, καθώς επίσης και ο τρόπος με τον οποίο έγινε ο γραφικός σχεδιασμός της εφαρμογής και των κανόνων στους οποίους αυτός έγκειται. Στο επόμενο και τελευταίο κεφάλαιο συνοψίζουμε την διπλωματική αυτή εργασία, τονίζοντας τις δυσκολίες και τις συνεισφορές αυτής, προτείνοντας και κάποιες μελλοντικές επεκτάσεις.

Κεφάλαιο 7

Ανακεφαλαίωση Διπλωματικής Εργασίας

Μελλοντικές επεκτάσεις

7.1 Ανακεφαλαίωση Διπλωματικής εργασίας

Η παρούσα διπλωματική εργασία αποτέλεσε μία πρόταση για τη δημιουργία μίας φυσικής γλώσσας σε περιβάλλοντα ψηφιακής τηλεόρασης, που θα επιτρέπει στον χρήστη, κάνοντας χρήση οποιασδήποτε συσκευής, να επικοινωνεί με την ψηφιακή συσκευή καταγραφής (PDR) που διαθέτει στο σπίτι του, με σκοπό:

- να καθορίσει τους κανόνες διαχείρισης των δεδομένων ψηφιακής τηλεόρασης (ψηφιακά τηλεοπτικά προγράμματα, μεταδεδομένα τηλεοπτικών προγραμμάτων),
- να ανακτήσει δεδομένα για τα ψηφιακά τηλεοπτικά προγράμματα και τα μεταδεδομένα τηλεοπτικών προγραμμάτων βάσει οποιασδήποτε πληροφορίας για το είδος τους.
- να εκφράσει τις προτιμήσεις του και επιθυμίες για το είδος των τηλεοπτικών προγραμμάτων που θα αποθηκευτούν,
- να διαχειριστεί τις εξατομικευμένες λίστες των τηλεοπτικών προγραμμάτων που έχουν επιλεγεί για αποθήκευση,
- να τροποποιήσει οτιδήποτε από τα παραπάνω.

Για την ομαλή εισαγωγή του αναγνώστη σε έννοιες και ορισμούς της φυσικής γλώσσας έγινε μία επισκόπηση σχετικής έρευνας με μελέτη αλγορίθμων και τεχνολογιών, που εφαρμόζονται σε συστήματα επεξεργασίας φυσικής γλώσσας. Έγινε εκτενής παρουσίαση του αναλυτή διαγραμμάτων που χρησιμοποιήθηκε στην υλοποίηση του συστήματός μας (JavaChart) και τέλος, παρουσιάστηκαν τα υπάρχοντα συστήματα διαλόγου, εφαρμοσμένα σε περιβάλλοντα ψηφιακής τηλεόρασης.

Το μοντέλο που αναπτύχθηκε στηρίχθηκε στο πρότυπο TV-Anytime και συγκεκριμένα, με τη διαχείριση της πληροφορίας πάνω στις διάφορες κατηγορίες μεταδεδομένων προγραμμάτων και στα μεταδεδομένα του προφίλ του χρήστη. Το πρότυπο TV-Anytime καθορίζει δομές από κατηγορίες μεταδεδομένων, που περιγράφουν το περιεχόμενο προγραμμάτων ψηφιακής τηλεόρασης, καθώς και δομές για μεταδεδομένα που περιγράφουν τα ενδιαφέροντα ενός χρήστη (προφίλ) για προγράμματα ψηφιακής τηλεόρασης. Παρουσιάστηκε, λοιπόν, η γενική οργάνωση ενός TV-Anytime συστήματος και αναλύθηκε η έννοια των μεταδεδομένων εξετάζοντας τις κατηγορίες στις οποίες αυτά χωρίζονται. Στη συνέχεια, για να γίνει κατανοητό το σκεπτικό πάνω στο οποίο στηρίχθηκε το μοντέλο της φυσικής γλώσσας, αναλύθηκαν τα βασικά στοιχεία που χαρακτηρίζουν τις προτιμήσεις ενός χρήστη, πάνω στα οποία στηρίχθηκε η δομή της πληροφορίας που εισάγεται στο σύστημα. Έπειτα, αναλύθηκαν τα βασικά στοιχεία που περιγράφουν ένα πρόγραμμα, τα οποία αποτελούν και τα στοιχεία που συνοδεύουν ένα πρόγραμμα κατά την παρουσίασή του στον χρήστη.

Στη συνέχεια, μελετήθηκε διεξοδικά το μοντέλο της φυσικής γλώσσας ξεκινώντας από τις προδιαγραφές των απαιτήσεων σε ένα περιβάλλον ψηφιακής τηλεόρασης. Μείζονος σημασίας η ανάπτυξη του κύριου σώματος εργασίας της φυσικής γλώσσας, που αφορά στη σύνταξη και κατηγοριοποίηση των φράσεων και οδήγησε στην παρουσίαση του μεταμοντέλου της φυσικής γλώσσας και την απεικόνιση της πληροφορίας, όπως αυτή εξάγεται στη δομή ιδιοτήτων από τον αναλυτή διαγραμμάτων. Η κατηγοριοποίηση των φράσεων έγινε με γνώμονα τις προδιαγραφές των απαιτήσεων σε ένα περιβάλλον ψηφιακής τηλεόρασης και αφορούν σε:

- ✓ Εισαγωγικές φράσεις, για καθορισμό της λειτουργίας του συστήματος.
- ✓ Φράσεις αναζήτησης, για καθορισμό της TV-Anytime πληροφορίας.
- ✓ Φράσεις στόχου, για καθορισμό του στόχου της λειτουργίας .
- ✓ Χρονικές φράσεις, για καθορισμό γενικά χρονικών φράσεων.
- ✓ Φράσεις περίληψης, για καθορισμό των χαρακτηριστικών περιλήψεων οπτικο-ακουστικού περιεχομένου.

Όσον αφορά στο μεταμοντέλο της φυσικής γλώσσας, για τη δημιουργία των επιτρεπτών εκφράσεων του χρήστη, αποφασίστηκε ότι μία έκφραση μπορεί να αποτελεί μία ή περισσότερες οντότητες. Οι οντότητες αυτές διαχωρίζουν την πληροφορία, συμβατή με το TV-Anytime για τα μεταδεδομένα των προγραμμάτων, με τις λειτουργίες που πρέπει να διατελέσει το σύστημα για την οργάνωση αυτής. Στην συνέχεια, έγινε παρουσίαση της λειτουργικότητας που παρέχει το σύστημα, που αφορά στο συνδυασμό των φράσεων με τελικό στόχο την ικανοποίηση των απαιτήσεων, στην ταξινόμηση της πληροφορίας βάση του

προτύπου TV-Anytime και ανάλυση της στρατηγικής και του βασικού αλγορίθμου που ακολουθείται στην περίπτωση ασαφούς πληροφορίας από το χρήστη, όπου δε μπορούμε να έχουμε ταξινόμηση σε κάποια από τις TV-Anytime κατηγορίες.

Το μοντέλο που παρουσιάστηκε είναι πλήρες σε σχέση με το TV-Anytime, εφόσον διαχειρίζεται όλες τις δομές και τις κατηγορίες μεταδεδομένων του TV-Anytime. Το μοντέλο είναι, επίσης, επεκτάσιμο και μπορεί να εικμεταλλευθεί περισσότερες κατηγορίες, που μπορεί να δημιουργηθούν στο μέλλον, εφόσον στην απεικόνιση της πληροφορίας η πληροφορία που αφορά στο TV-Anytime, σχετίζεται με συγκεκριμένη οντότητα, και ενδεχόμενες επεκτάσεις στο TV-Anytime, προσθέτουν πεδία, που αντιστοιχίζουν σε TVA κατηγορίες στην οντότητα αυτή.

Το θέμα της επίλυσης των ασαφειών από μία φράση του χρήστη, αποτελεί πολύ σημαντικό κομμάτι για την εργασία αυτή. Στη γενική περίπτωση, λοιπόν, που μια μερική φράση από την ερώτηση του χρήστη μπορεί να πάρει περισσότερες από μία σημασίες στην ιεραρχία του TV-Anytime, το σύστημα δε χρησιμοποιεί διευκρινιστικούς διαλόγους για την αποσαφήνιση των ερωτήσεων των χρηστών και όλα τα πιθανά αποτελέσματα επιστρέφονται στον χρήστη. Αντί των διευκρινιστικών διαλόγων, γίνεται μία αναζήτηση σε υπάρχουσες οντολογίες με πληροφορία σχετική με ψηφιακή τηλεόραση και αποδίδεται σημασιολογία στις ασαφείς φράσεις αυτές. Για την καλύτερη ταξινόμηση των αποτελεσμάτων αυτών χρησιμοποιείται το TV-Anytime προφίλ του χρήστη και αποδίδεται ένα βάρος σύμφωνα με τις προτιμήσεις του χρήστη, ούτως ώστε τα πρώτα αποτελέσματα να είναι και τα πλέον σχετικά με τις απαιτήσεις του χρήστη. Ο αλγόριθμος που αναπτύχθηκε εξετάζει τις καλύτερες δυνατές συσχετίσεις και ενοποιεί τα αποτελέσματα αναθέτοντας το κανονικοποιημένο βάρος στις τιμές των χαρακτηριστικών κατηγοριών.

Στη συνέχεια παρουσιάστηκε ο θεμελιώδης σχεδιασμός της αρχιτεκτονικής του συστήματος που αναπτύχθηκε στα πλαίσια της εργασίας αυτής για την υποστήριξη του μοντέλου της φυσικής γλώσσας με συνοπτική ανάλυση των υπο-μονάδων που την απαρτίζουν, ενώ ακολούθησαν τα Use Cases μέσα από τα οποία παρουσιάστηκε η λειτουργικότητα, και πως αυτή εξυπηρετείται με χρήση των υπο-μονάδων της αρχιτεκτονικής.

- ✓ Μέσα από την ανάλυση της υλοποίησης του συστήματος, γίνεται φανερός ο τρόπος με τον οποίο επικοινωνούν όλες οι υπο-μονάδες και ο τρόπος που μετασχηματίζεται η

πληροφορία ανάλογα με τις ανάγκες του εισαγωγής του χρήστη. Παρουσιάζεται το σύνολο των γραμματικών κανόνων βασισμένο στο μοντέλο της φυσικής γλώσσας και το σύνολο των λεξικών, δηλαδή ενός λεξικού που περιέχει ρίζες λέξεων (stem lexicon) και ενός που περιέχει καταλήξεις για κατηγορίες λέξεων (mini-lexicon) με τελικό στόχο την μείωση του όγκου της πληροφορίας μέσα στο λεξικό. Ανάλογα με τη δομή της πληροφορίας, αναλύονται όλοι οι τρόποι με τους οποίους αντιμετωπίζονται τα προβλήματα που προκύπτουν, όπως στην περίπτωση με την εξαγωγή δύο ή περισσότερων δομών ιδιοτήτων από τον αναλυτή διαγραμμάτων (Chart Parser), ενώ αναλυτικά παρουσιάστηκε και ο βασικός αλγόριθμος από την πλευρά της υλοποίησης και η στρατηγική που ακολουθείται για την καλύτερη αντιμετώπιση των ασαφειών της εισαγωγής του χρήστη. Ένα βασικό στοιχείο στην ανάλυση αυτή αποτελεί το γεγονός της ύπαρξης ή όχι λέξης 'and' ή 'or' ανάμεσα στη συμβολοσειρά που παρουσιάζει ασάφεια. Θα πρέπει να επισημανθεί ότι ανεξάρτητα από την ασαφή εισαγωγή του χρήστη, λόγω της χρήσης οντολογιών για την ανάκτηση δεδομένων υπάρχουν πάντα αποτελέσματα προς διαχείριση ικανοποιώντας τις απαιτήσεις του χρήστη. Στο τέλος παρουσιάστηκε ο τρόπος με τον οποίο το σύστημα απαντά στον χρήστη, ανάλογα με την ορθότητα ή όχι των ερωτήσεων του και η επιλογή που γίνεται στην ταξινόμηση των αποτελεσμάτων βάση των γενικότερων προτιμήσεων του χρήστη, που έχουν ήδη δηλωθεί στο προφίλ του. Οι βασικές αρχές για την ταξινόμηση αυτή, αφορούν στη σημασιολογία που αποδίδεται στις λέξεις που αποτελούν την συμβολοσειρά που παρουσιάζει ασάφεια και στη σημασιολογία που παίρνουν αυτές μέσα από τις προτιμήσεις του χρήστη, όπως δηλώνονται αυτές στο προφίλ του.

Οι βασικοί στόχοι της εργασίας μας ήταν αφενός να σχεδιαστεί και να υλοποιηθεί το μοντέλο με αξιοπιστία ως προς τα δεδομένα του χρήστη, αφετέρου να σχεδιαστεί και να κατασκευαστεί μία εφαρμογή που θα αναλαμβάνει την επικοινωνία του χρήστη με το υπόλοιπο σύστημα για την εισαγωγή των αιτήσεων και την σωστή απεικόνιση των αποτελεσμάτων. Έτσι, το μοντέλο φυσικής γλώσσας που κατασκευάστηκε, χρησιμοποιήθηκε και για τη δημιουργία μιας διεπαφής (interface) για τη διαχείριση της TV-Anytime πληροφορίας από κινητές συσκευές (mobile phones, PDA), για τα οποία οι διεπαφές με φυσική γλώσσα είναι ιδιαίτερα σημαντικές, λόγω των περιοριστικών τους πόρων. Η Java τεχνολογία που χρησιμοποιήθηκε είναι συμβατή με την J2ME πλατφόρμα για κινητές συσκευές και το MIDP προφίλ, που ορίζεται από αυτή. Έτσι, ένα απαιτητικό σε μέγεθος γραφικό περιβάλλον για ψηφιακή

τηλεόραση μπορεί και συνοψίζεται στις ελάχιστες προδιαγραφές κινητών συσκευών μέσω ενός συστήματος φυσικής γλώσσας.

7.2 Μελλοντικές Επεκτάσεις

Το μοντέλο της φυσικής γλώσσας που αναπτύχθηκε στα πλαίσια της διπλωματικής αυτής εργασίας, όπως είδαμε και στο κεφάλαιο 2, ανήκει σε βαθμό πολυπλοκότητας ανάμεσα στις 2 πρώτες κατηγορίες του σχήματος 1, λόγω του ότι δεν υποστηρίζονται διευκρινιστικοί διάλογοι ανάμεσα στο χρήστη και το σύστημα. Ένας μελλοντικός στόχος, λοιπόν αποτελεί η υποστήριξη τέτοιου είδους διαλόγων με στόχο την μερική άρση της ασάφειας σε επίπεδο φυσικής γλώσσας και όχι σε επίπεδο επεξεργασίας της δομής ιδιοτήτων που παράγεται από την επεξεργασία της εισαγωγής του χρήστη, από τον αναλυτή διαγραμμάτων.

Από πλευράς λειτουργικότητας, μελλοντική επέκταση αποτελεί η σχεδίαση και υλοποίηση κατηγοριών από κανόνες, που αφορούν στην κινητικότητα των χρηστών, δηλαδή κανόνες για τη δημιουργία και διαχείριση εγγράφων κινητικότητας (mobility records) που αντιστοιχούν σε ένα ενδεχόμενο πλάνο ταξιδιού του χρήστη. Επιπλέον περιλαμβάνουν κανόνες για τη διαχείριση του προφίλ του χρήστη στους διάφορους κόμβους του ταξιδιού του ανάλογα με τον προορισμό και τον τύπο του ταξιδιού. Επίσης, μπορούν να υποστηριχθούν κανόνες που αφορούν στη διαχείριση των χρηστών, δηλαδή κανόνες για τη διαχείριση των επιπέδων πρόσβασης του κάθε χρήστη στο περιεχόμενο του PDR, καθώς και του επιπέδου πρόσβασης στα προφίλ που βρίσκονται σε αυτό.

Το μοντέλο που αναπτύχθηκε στηρίχθηκε στο πρότυπο TV-Anytime και συγκεκριμένα, με τη διαχείριση της πληροφορίας πάνω στις διάφορες κατηγορίες μεταδεδομένων προγραμμάτων και στα μεταδεδομένα του προφίλ του χρήστη. Προέκταση αυτού, αποτελεί η υποστήριξη μέσα από το μοντέλο του προτύπου MPEG-7. Το MPEG-7 είναι ένα πρότυπο μεταδεδομένων για να καλύψει τις αυξανόμενες ανάγκες για μια αυστηρά ορισμένη περιγραφή οπτικο-ακουστικού περιεχομένου, σε περιβάλλον πολυμέσων.

Βιβλιογραφία

- [1] Johansson, P., Degerstedt, L., Jönsson, A. (2002). *Iterative Development of an Information-Providing Dialogue System*. In Proceedings of the 7th Workshop on User Interfaces for All. Chantilly, France.
- [2] Ibrahim, A., Johansson, P. (2002). *Multimodal Dialogue Systems: A Case Study for Interactive TV*. In Proceedings of the 7th Workshop on User Interfaces for All. Chantilly, France.
- [3] Ibrahim, A., Johansson, P. (2002). *Multimodal Dialogue Systems for Interactive TV Applications*. In Proceedings of 4th IEEE International Conference on Multimodal Interfaces, Pittsburgh, USA. pp. 117-222.
- [4] Jurafsky, D., Martin, J.H. (2000). *Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. New Jersey: Prentice Hall.
- [5] Early, J. (1970). *An efficient context-free parsing algorithm*. Communications of the ACM, 6, 451-455.
- [6] Allen, J., Byron, D., Dzikovska, M., Ferguson, G., Galescu, L., Stent, A. (1998). *An Architecture for a Generic Dialogue Shell*. In Natural Language Engineering, 1, 1-15.
- [7] Allen, J., Ferguson, G., Stent, A. (2001a). *An Architecture For More Realistic Conversational Systems*. In Proceedings of IUI '01, 14-17. Santa Fe, New Mexico, USA.
- [8] Allen, J.F., Byron, D.K., Dzikovska, M., Ferguson, G., Galescu, L., Stent, A. (2001b). *Towards Conversational Human-Computer Interaction*. In AI Magazine, 2001, 1-9.
- [9] Kazasis, F.G., Moumoutzis, N., Pappas, N., Karanastasi, A., Christodoulakis, S. (2003). *Designing Ubiquitous Personalized TV-Anytime Services*. In the International Workshop on Ubiquitous Mobile Information and Collaboration Systems (UMICS), Klagenfurt/Velden, Austria.
- [10] Riggs, Roger., Taivalssari, A., VandenBrink, M., (2001). *Programming Wireless Devices with the Java™ 2 Platform, Micro Edition*. Addison-Wesley.
- [11] TV-Anytime Specification Series: S3 on Metadata (Normative). <http://www.tv-anytime.org>
- [12] MPEG Group, MPEG-7 (Multimedia Content Description Interface). <http://mpeg.telecomitalia.com/standards/mpeg-7/mpeg-7.htm>

- [13] Fowler, M., Scott, K. (1997). UML Distilled Applying the Standard Object Modelling Language, ADDISON-WESLEY.
- [14] Java™ 2 Platform – Standard Edition, v. 1.4.1 API Specification, <http://java.sun.com/j2se/1.4.1/docs/api/>
- [15] Degerstedt, L. (2002). *JavaChart User Manual*, <http://nlpfarm.sourceforge.net/javachart/>
- [16] Arnold, D. (2002). *Chart Parsing*, Computational Linguistics I, University Of Essex.
- [17] Shieber, S. M. (1985), *The design of a Computer Language for Linguistic Information*. In Proceedings of the 22nd Annual Meeting of the Association for Computational Linguistics.
- [18] Flycht-Eriksson, A., Jonsson, A. (2000). *Dialogue and Domain Knowledge Management in Dialogue Systems*. In Proceedings of the 1st SigDial Workshop, Hong-Kong.
- [19] Russel, S., Norvig, P. (1995). *Artificial Intelligence. A Modern Approach*. New Jersey : Prentice Hall.
- [20] Jonsson, A., Stromback, L. (1998). *Robust Interaction through Partial Interpretation and Dialogue Management*. In Proceedings of Coling-ACL '98, Montreal, Canada.
- [21] Multimodal Interaction for Information Appliances (MIINA Project). <http://www.ida.liu.se/~nlplab/miina/>
- [22] NOKIA – Talkative TV-Guide, Program Guide Information System. <http://www.nokia.com/nokia/0,,27727,00.html>
- [23] Tsinaraki, C., Fatourou, E., Christodoulakis, S. *An Ontology-Driven Framework for the Management of Semantic Metadata describing Audiovisual Information*. In the proceedings of the 15th International Conference on Advanced Information Systems Engineering (CAiSE), Klagenfurt/Velden, Austria, June 2003.
- [24] Pappas, N., Kazasis, F., Moumoutzis, M., Tsinaraki, C., and Christodoulakis, C.,. *Personalized and ubiquitous information services for TV programs*. In the proceedings of the International Workshop on Multimedia in Digital Libraries (MDL), Greece, June 2003
- [25] Tsinaraki, C., Fatourou, E., Christodoulakis, S., *An Interoperability Framework for the Management of Semantic Metadata in order to support Personalized TV Services*. In the proceedings of the 2nd

- [26] Tsinaraki, C., Polydoros, P., Kazasis, F.G., Christodoulakis, S. *Ontology-based Semantic Indexing for MPEG-7 and TV-Anytime Audiovisual Content*. Multimedia Tools and Applications International Journal, Special issue on "Video Segmentation for Semantic Annotation and Transcoding.", Kluwer Academic Publishers, to appear.
- [27] Moumoutzis, N., Kazasis, F., Pappas, N. *Generic Architecture for handling TVA Metadata using relational database technologies*. Technical Report, TUC/MUSIC, 2002.
- [28] Kazasis F., Moumoutzis N., Pappas N., *Informative Annex: Generic Architecture for handling TVA Metadata using relational database technologies*, contribution TV-Anytime Forum/Metadata WG, 2nd Implementer's workshop, Geneva July 2002.
- [29] Κοτόπουλος, Γ., *Σύστημα Διήθησης TV-Anytime Περιεχομένων Προγραμμάτων Ψηφιακής Τηλεόρασης σύμφωνα με τα Ενδιαφέροντα των Χρήστων*. Διπλωματική Εργασία, Τμήμα Ηλεκτρονικών Μηχανικών και Μηχανικών Η/Υ, Πολυτεχνείο Κρήτης, Χανιά 2003.
- [30] Κοτόπουλος, Ι.Α., *Σύστημα Αυτόματης Δημιουργίας και Αναπροσαρμογής Προφίλ Χρηστών σε Περιβάλλον Ψηφιακής Τηλεόρασης*. Διπλωματική Εργασία, Τμήμα Ηλεκτρονικών Μηχανικών και Μηχανικών Η/Υ, Πολυτεχνείο Κρήτης, Χανιά 2003.