

ΣΥΣΤΗΜΑ ΔΙΗΘΗΣΗΣ TV-ANYTIME
ΠΕΡΙΕΧΟΜΕΝΟΥ ΠΡΟΓΡΑΜΜΑΤΩΝ ΨΗΦΙΑΚΗΣ
ΤΗΛΕΟΡΑΣΗΣ ΣΥΜΦΩΝΑ ΜΕ ΤΑ
ΕΝΔΙΑΦΕΡΟΝΤΑ ΤΩΝ ΧΡΗΣΤΩΝ

Από τον Γεώργιο Κοτόπουλο

Πολυτεχνείο Κρήτης

Τμήμα Ηλεκτρονικών Μηχανικών και Μηχανικών Ηλεκτρονικών Υπολογιστών

Μια διπλωματική εργασία που παρουσιάστηκε στο Πολυτεχνείο Κρήτης για την εκπλήρωση
των απαιτήσεων απόκτησης διπλώματος Ηλεκτρονικού Μηχανικού και Μηχανικού

Ηλεκτρονικών Υπολογιστών

Χανιά 2003

*Στην Αδερφή μου
και στο αγέννητο
ανιψάκι μου*

*Της δ' αρετῆς ιδρώτα θεοί
προπάρουθεν ἔθηκαν*

Ἡσιόδος

ΠΕΡΙΛΗΨΗ

Η επικράτηση της ψηφιακής τηλεόρασης σήμερα κάνει διαθέσιμα στο σπίτι χιλιάδες ψηφιακά κανάλια με τεράστιο ποσό πληροφορίας. Γίνεται όμως πρακτικά αδύνατο για τους χρήστες να ψάξουν και να βρουν προγράμματα που τους ενδιαφέρουν, και που ταυτόχρονα εκπέμπονται σε χρόνο που μπορεί να τα παρακολουθήσει. Το TV-Anytime forum καθόρισε αρχιτεκτονικές ψηφιακής τηλεόρασης που βασίζονται σε μεγάλους δίσκους που ενσωματώνονται στην ψηφιακή τηλεόραση ώστε να μπορεί να αποθηκευτεί αυτόματα ένας αριθμός προγραμμάτων.

Το TV-Anytime forum καθορίζει, επίσης, δομές μεταδεδομένων που επιτρέπουν στους χρήστες να αναζητούν, να πλοηγούνται και να διαχειρίζονται περιεχόμενο από μία ποικιλία εσωτερικών και εξωτερικών πηγών που περιλαμβάνουν, π.χ., αναμεταδόσεις, διαδραστική τηλεόραση, διαδίκτυο και τοπική αποθήκευση. Επιπλέον, ορίζει ένα συγκεκριμένο τρόπο περιγραφής προφίλ χρηστών, συμπεριλαμβανόμενων προτιμήσεων αναζήτησης, για να διευκολύνει την αυτόματη διήθηση και απόκτηση του περιεχομένου (προγράμματα) από πράκτορες για λογαριασμό του χρήστη.

Στην παρούσα διπλωματική διατριβή, αναπτύχθηκαν μηχανισμοί ευφυούς διήθησης TV-Anytime περιεχομένου προγραμμάτων. Συγκεκριμένα, σχεδιάστηκε ένα μεταφραστικό μοντέλο του σχήματος μεταδεδομένων του TV-Anytime για τις προτιμήσεις των χρηστών σε ασαφείς δυαδικές εκφράσεις ώστε από ένα αφηρημένο χώρο προτιμήσεων να υπάρχει μία πλήρως καθορισμένη συντακτικά και σημασιολογικά ερώτηση. Οι δυαδικές εκφράσεις χρησιμοποιήθηκαν από ένα πλαίσιο ασαφούς λογικής ώστε τελικά να υπολογιστούν εκτιμήσεις για το περιεχόμενο σύμφωνα με τα ενδιαφέροντα των χρηστών. Επίσης, για να κατανοεί ο χρήστης καλύτερα την συμπεριφορά του συστήματος διατηρείται πληροφορία για το τι χαρακτηριστικά του προφίλ ταιριάζουν με τα μεταδεδομένα κάθε προγράμματος. Έτσι, μπορεί να διαπιστώσει ότι το σύστημα δεν συμπεριφέρεται αυθαίρετα αν επιστρέψει ένα πρόγραμμά που τελικά δεν του άρεσε αλλά σύμφωνα με τις προτιμήσεις του. Επιπλέον, κατασκευάστηκε μηχανισμός συνεργατικής διήθησης ο οποίος διαχειρίζεται πληροφορίες για τις αρέσκειες των χρηστών για να εκμαιεύσει ομοιότητες και να κάνει προτάσεις

προγραμμάτων προς τους χρήστες. Δόθηκε έμφαση στο να είναι σε θέση ο μηχανισμός έστω και με ελάχιστα δεδομένα να μπορεί να κάνει προβλέψεις. Αυτό γίνεται κάνοντας χρήση του προφίλ των χρηστών για πάρει τα πρώτα δεδομένα ενώ παράλληλα δεν δεσμεύει τους χρήστες να χρειάζεται να δίνουν όλοι τα δεδομένα τους αλλά μπορεί να δώσει εξίσου καλά αποτελέσματα με ένα μικρότερο σύνολο χρηστών.

Κατασκευάστηκε, επίσης, ένα γραφικό εργαλείο για την κατασκευή και επεξεργασία προφίλ χρηστών από τους ίδιους. Το εργαλείο αυτό υποστηρίζει πλήρως το σχήμα του TV-Anytime για τα μεταδεδομένα των χρηστών. Δόθηκε έμφαση στην, κατά το δυνατόν, διευκόλυνση του χρήστη στο χειρισμό του εργαλείου δεδομένου της περιπλοκότητας του σχήματος. Γενικά, ο σχεδιασμός του εργαλείου αυτού είναι βασισμένος στις αρχές των Ερωτήσεων Δια Παραδείγματος (Query by Example). Επιπλέον, προσφέρει τη δυνατότητα στο χρήστη να χρησιμοποιεί πληροφορία από προγράμματα που έχει δει και του άρεσαν για να επεξεργαστεί ή ακόμα και να κατασκευάσει το προφίλ του χωρίς να εισέλθει σε περιττές για αυτόν λεπτομέρειες τις οποίες όμως διαφορετικά δεν θα μπορούσε να αποφύγει.

ΕΥΧΑΡΙΣΤΙΕΣ

Ο συγγραφέας θα ήθελε να ευχαριστήσει τον καθηγητή κ. Σταύρο Χριστοδουλάκη για την επίβλεψη και καθοδήγησή του για την εκπόνηση αυτής της διπλωματικής εργασίας, καθώς και για τις σημαντικές εμπειρίες που του προσέφερε κατά την διάρκεια της εργασίας του στο Εργαστήριο Κατανεμημένων Πληροφοριακών Συστημάτων και Εφαρμογών.

Θα ήθελε επίσης να ευχαριστήσει τους αναγνώστες της διπλωματικής κ. Μανόλη Κουμπράκη και κ. Ευριπίδη Πετράκη για τον χρόνο που αφιέρωσαν στην ανάγνωσή της και για τις παρατηρήσεις τους.

Επίσης, οφείλει ένα ιδιαίτερο ευχαριστώ στον Νεκτάριο Μουμουτζή για την συνεργασία και την αρωγή που του προσέφερε στο σχεδιασμό της εργασίας αυτής καθώς και τις συμβουλές του σε τεχνικά θέματα.

Ένα μεγάλο ευχαριστώ στους συνεργάτες που δούλεψαν στο πρόγραμμα UP-TV Νίκο Παπά, Φώτη Καζάση, Γιάννη Κοτόπουλο και Νατάσσα Καρναναστάση, Μαρία Φραντζή των οποίων η έγκαιρη και άρτια εργασία οδήγησε στην ταχύτερη εκπλήρωση της διπλωματικής αυτής διατριβής.

Η διπλωματική αυτή εργασία χρηματοδοτήθηκε στα πλαίσια του Ευρωπαϊκού Ερευνητικού Προγράμματος UP-TV, στο οποίο συμμετέχει το εργαστήριο Διανεμημένων Πληροφοριακών Συστημάτων και Εφαρμογών (MUSIC), του Πολυτεχνείου Κρήτης.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ.....	IV
ΠΕΡΙΕΧΟΜΕΝΑ	VII
ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ.....	X
ΕΙΣΑΓΩΓΗ.....	1
Αναγκαιότητα	1
Το πρόγραμμα UP-TV	3
Στόχοι της Εργασίας	5
Δομή της Εργασίας.....	6
ΤΕΧΝΟΛΟΓΙΕΣ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ – ΣΥΣΧΕΤΙΖΟΜΕΝΕΣ ΕΡΓΑΣΙΕΣ	9
Εισαγωγή	9
MPEG-7	9
Το TV-Anytime	11
Μεταδεδομένα στο TV-Anytime	13
Το Query by Example	14
Συσχετιζόμενες Εργασίες	19
Σερφάροντας στο Ψηφιακό Κύμα (Surfing the Digital Wave).....	19
Εξερευνητής Τηλεόρασης (TV Scout).....	19
Εξατομικευμένες Προτάσεις Τηλεοπτικών Προγραμμάτων (Personalized TV Show Recommender).....	20
Εξατομικευση στην Ψηφιακή Τηλεόραση (Personalization in Digital TV)	20
Σύγκριση	21
Περίληψη	21
ΤΟ ΣΧΗΜΑ ΜΕΤΑΔΕΔΟΜΕΝΩΝ ΤΟΥ TV-ANYTIME	23
Εισαγωγή	23
Μεταδεδομένα των Περιγραφής Περιεχομένου	23
ProgramInformation	24
BasicDescription	24
Μεταδεδομένα Περιγραφής Στιγμιότυπων	27
Μεταδεδομένα Περιγραφής Τμημάτων Προγραμμάτων	30
Μεταδεδομένα για τα Προφίλ των Χρηστών.....	31
Filtering and Search Preferences	32
Creation Preferences	33
Classification Preferences	34
Source Preferences.....	36
Preference Condition	37
Παράδειγμα Filtering And Search Preferences.....	37
Αντιστοιχίσεις μεταξύ χαρακτηριστικών του Filtering and Search Preferences και των μεταδεδομένων των προγραμμάτων	39
Περίληψη	42
ΜΗΧΑΝΙΣΜΟΙ ΕΥΦΥΟΥΣ ΔΙΕΘΗΣΗΣ ΠΕΡΙΕΧΟΜΕΝΟΥ.....	43
Εισαγωγή	43
Αρχιτεκτονική	44
Η Βασισμένη στο Περιεχόμενο Μονάδα Διήθησης (Content Based Filtering Module, CBFM)	48
Το πλαίσιο ασαφούς λογικής για τις εκτιμήσεις προτιμήσεων	48
Περιγραφή του Μοντέλου Ασαφούς Λογικής	49
Ενσωματώνοντας αρνητικά βάρη ερωτήσεων στο WFIRS	52
Υλοποίηση του p-Norm Extended Boolean Μοντέλου σε σχεσιακή βάση δεδομένων.....	53
Μειώνοντας τις επιβαρύνσεις του χώρου.....	56
Ειδικές Περιπτώσεις	60

1 ^η Ειδική Περίπτωση: Όλα τα w_i 's είναι ίσα με το 1	61
2 ^η Ειδική Περίπτωση: Όλα τα a_i 's είναι ίσα με το 1	61
Μεταφραστικό σχήμα του προφίλ σε δυαδικό δέντρο	62
Επιπλέον Σημασιολογία του προφίλ	62
Ένα γενικό μεταφραστικό σχήμα από ένα φίλτρο σε ένα δυαδικό συντακτικό δέντρο	64
Το GWFIRS που υλοποιεί τη συσχέτιση μεταξύ FASP και των μεταδεδομένων του περιεχομένου	66
Υλοποίηση	67
Το σχήμα της Βάσης	67
Υπολογισμός Μεμονωμένων Κόμβων	69
Υπολογισμός των Επιμέρους Φίλτρων	70
Υπολογισμός των Μεμονωμένων Προτιμήσεων	71
Υπολογισμός των Επιμέρους Προτιμήσεων	72
Τελικός υπολογισμός κόμβου	73
Υπολογισμός του συνολικού προφίλ του χρήστη	74
Υπολογισμός Ένωσης Τένων	76
Υπολογισμός Ένωσης Τένων με Πατέρα	77
Επέκταση του TV-Anytime	78
Διατήρηση των χαρακτηριστικών του προφίλ που ταιριάζουν με κάθε πρόγραμμα	79
Ειδικά θέματα που αφορούν την απόδοση	80
Εφαρμογές της βασισμένης στο περιεχόμενο μονάδας διήθησης	81
Η Μονάδα Συνεργατικής Διήθησης (Collaboration Filtering Module, CFM)	82
Η Μονάδα Εύρεσης Παρόμοιων Καταναλωτών (Find Similar Consumers Module, FSCM)	83
Χρησιμοποιώντας Προτιμήσεις Περιεχομένου για την Εύρεση Παρόμοιων Χρηστών	84
Χρησιμοποιώντας τις Προτιμήσεις Διήθησης και Αναζήτησης για την Εύρεση Παρόμοιων Χρηστών	85
Η Μονάδα Συνεργατικών Προτάσεων (The Collaborative Suggestions Module, FSCM)	86
Ξεπερνώντας Προβλήματα Προστασίας Προσωπικών Δεδομένων Χρηστών	87
Υλοποίηση	89
Υπολογισμός Ομοιοτήτων με χρήση του συντελεστή Pearson	89
Υπολογισμός Προτάσεων προς τους Χρήστες	90
Περίληψη	91
ΓΡΑΦΙΚΟ ΠΕΡΙΒΑΛΛΟΝ ΔΙΑΧΕΙΡΙΣΗΣ ΠΡΟΦΙΛ ΧΡΗΣΤΗ	93
Εισαγωγή	93
Γραφικός Σχεδιασμός της Εφαρμογής	94
Βασικές Αρχές Σχεδιασμού (User Interface Guidelines)	95
Αναπαράσταση Filtering and Search Preferences με Query by Example	97
Δενδρική Αναπαράσταση της Ιεραρχίας των Φίλτρων	98
Λειτουργικότητα	98
Γενικά	98
Use Cases	99
Use Case 1: Επεξεργασία προφίλ χρήστη	101
Use Case 2: Επεξεργασία ιεραρχίας φίλτρου	102
Use Case 3: Επεξεργασία χαρακτηριστικών φίλτρου	103
Use Case 4: Δημιουργία νέου προφίλ με χρήση wizard	104
Use Case 5: Δημιουργία νέου φίλτρου	104
Use Case 6: Πρόσθεση τιμών μεταδεδομένων προγραμμάτων σε πεδία φίλτρου	105
Use Case 7: Επεξεργασία των τιμών πεδίου φίλτρου	106
Υλοποίηση	106
Επεξεργασία Προφίλ	107
Το μενού	109
Τα Tabs με τα Preferences	110
Η Ιεραρχία	114
Κατασκευή Νέων Κόμβων	115
Επεξεργασία Χαρακτηριστικών	116
Προσθήκη χαρακτηριστικών από μεταδεδομένα επιλεγμένων προγραμμάτων που ενδιαφέρουν τον χρήστη	119
Χρήση του Wizard	121

Το σύστημα βοήθειας.....	126
Περίληψη	127
ΑΝΑΚΕΦΑΛΑΙΩΣΗ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ.....	129
Συμπεράσματα - Ανακεφαλαίωση	129
Μελλοντικές Επεκτάσεις.....	131
Επεκτάσεις αναφορικά με τους Μηχανισμούς Ευφυούς Διήθησης Περιεχομένου.....	132
Επεκτάσεις αναφορικά με το Γραφικό Περιβάλλον.....	132
ΒΙΒΛΙΟΓΡΑΦΙΑ	135

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Σχήμα 1: Λογική κατάτμηση ενός TV-Anytime συστήματος σε επιμέρους υποσυστήματα	12
Σχήμα 2: Το σχήμα των μεταδεδομένων περιγραφής στιγμιότυπων	27
Σχήμα 3: Σχήμα που περιγράφει την δομή των μεταδεδομένων ενός προγράμματος.....	30
Σχήμα 4: Το στοιχείο Description, που κρατάει την πληροφορία για την περιγραφή τμημάτων προγραμμάτων	31
Σχήμα 5: UML διάγραμμα που περιγράφει την δομή του Filtering and Search Preferences.	32
Σχήμα 6: Δενδρικό παράδειγμα αναπαράστασης ενός προφίλ χρήστη.....	39
Σχήμα 7: Η Αρχιτεκτονική του Μηχανισμού Ευφυούς Διήθησης Περιεχομένου	45
Σχήμα 8: Παράδειγμα Boolean αναπαράστασης του Filtering and Search Preferences του σχήματος 6.....	63
Σχήμα 9: Τα τέσσερα επίπεδα στα οποία χωρίζεται η διαδικασία ταιριάσματος των μεταδεδομένων των προγραμμάτων με τους κόμβους FASP.....	70
Σχήμα 10: Σχηματική αναπαράσταση ιεραρχικού προφίλ όπου φαίνονται τα επίπεδα του δέντρου.....	75
Σχήμα 11: Διάγραμμα παρεχόμενης λειτουργικότητας από το γραφικό περιβάλλον.....	99

*Ἄνευ αἰτίου καὶ ἀρχῆς
αδύνατον τι εἶναι ἢ
γενέσθαι.*

Ἀριστοτέλης

Κ ε φ α λ α ι ο Ι

ΕΙΣΑΓΩΓΗ

Αναγκαιότητα

Ο σύγχρονος οπτικοακουστικός τομέας χαρακτηρίζεται από μια τεράστια ποικιλία και όγκο προσφερόμενης πληροφορίας με χιλιάδες κανάλια να μεταδίδουν επί 24ώρου βάσεως τηλεοπτικό ή/και ακουστικό περιεχόμενο. Η συνεχής διεύρυνση των προσφερόμενων επιλογών είναι μια διαρκώς αυξανόμενη τάση. Σύντομα ο κάθε τηλεθεατής θα έχει να διαλέξει ανάμεσα σε χιλιάδες κανάλια πράγμα που σημαίνει ότι και μόνο η απλή επισκόπηση των προσφερόμενων προγραμμάτων με τη μορφή τηλεοπτικών οδηγών θα απαιτεί δεκάδες ώρες. Ο σύγχρονος τηλεθεατής επιθυμεί να τιθασεύσει αυτό το χείμαρρο πληροφοριών με τον πιο αποτελεσματικό τρόπο ώστε να μπορεί ανά πάσα στιγμή να επιλέγει την τηλεοπτική εκπομπή που ταιριάζει περισσότερο με τα προσωπικά ή οικογενειακά του ενδιαφέροντα. Αυτή την ανάγκη θα περιγράψουμε με τη φράση «εξατομικευμένη πρόσβαση στην πληροφορία». Μέχρι σήμερα, αυτή η ανάγκη, δεν καλύπτονταν ή καλύτερα καλύπτονταν με τρόπους μη συστηματικούς που βασίζονταν στη δημοσίευση τηλεοπτικών οδηγών τους οποίους ο τηλεθεατής ήταν υποχρεωμένος να μελετά πριν κάνει την επιλογή του σταθμού της αρεσκείας του.

Τα μεταδεδομένα είναι η «πληροφορία για την πληροφορία». Τα μεταδεδομένα είναι συμπληρωματικά προς την καθαυτό πληροφορία, πολλές φορές δημιουργούνται και αποθηκεύονται εντελώς ανεξάρτητα από αυτήν και έχουν ως σκοπό να την καταστήσουν

ευκολότερα προσβάσιμη και να διευκολύνουν τη διαχείρισή της. Η ύπαρξη των μεταδεδομένων είναι απαραίτητη, ιδιαίτερα για τη λειτουργία συστημάτων που διαχειρίζονται οπτικοακουστική πληροφορία. Σήμερα, μεγάλος όγκος οπτικοακουστικού υλικού αποθηκεύεται σε ψηφιακή μορφή και διατίθεται σε χρήστες, μέσω συστημάτων ψηφιακών βιβλιοθηκών ή και του Παγκόσμιου Ιστού. Το οπτικοακουστικό υλικό, είναι πλούσιο τόσο σε δομή όσο και σε περιεχόμενο, ωστόσο η ιδιαίτερη φύση του και η διαφοροποίηση του σε σχέση με το γραπτό λόγο καθιστά αναγκαία τη χρήση νέων τεχνικών για την ανάκτηση του, αφού οι κλασικές τεχνικές που έχουν αναπτυχθεί για κείμενα δεν επαρκούν. Στην περίπτωση αυτή, συσχετίζονται με την οπτικοακουστική πληροφορία μεταδεδομένα, τα οποία έχουν τέτοια δομή και περιεχόμενο, ώστε να καθίσταται αποτελεσματική η πρόσβαση στην πληροφορία αυτή και η διαχείρισή της.

Τα μεταδεδομένα, κατά συνέπεια, είναι ο τρόπος με τον οποίο μπορούμε να ενημερωθούμε ουσιαστικά για το οπτικοακουστικό υλικό και παρόλο που τα υπάρχοντα συστήματα ανάκτησης, χρησιμοποιούν μεθόδους που βασίζονται σε αυτά, η διαφορετική δομή τους καθιστά αδύνατη τη διαφανή πρόσβαση σε περιεχόμενο πολλών συστημάτων. Προσπάθειες προς αυτή την κατεύθυνση έχουν γίνει με τον ορισμό προτύπων, τα οποία είναι ανεξάρτητα συγκεκριμένων εφαρμογών και έχουν ως στόχο την περιγραφή μεταδεδομένων με τρόπο γενικό, ώστε να μπορούν να υιοθετηθούν από διαφορετικά συστήματα.

Το υπό έκδοση διεθνές πρότυπο TV-Anytime είναι η πιο ολοκληρωμένη προσπάθεια σε αυτόν τον τομέα. Παρέχει ένα μοντέλο ορισμένο σε XML Schema, ικανό να περιγράψει από χαμηλού επιπέδου χαρακτηριστικά, όπως το χρωματικό φάσμα μιας εικόνας, μέχρι πολύ υψηλού επιπέδου, όπως τη σημασιολογική πληροφορία, δηλαδή την πληροφορία που δίνει ένας άνθρωπος όταν ρωτάται «τι συμβαίνει» στο οπτικοακουστικό υλικό.

Με εργαλείο το TV-Anytime μπορούν να αναπτυχθούν συστήματα τα οποία θα μπορούν να εξυπηρετούν ανάγκες του τηλεθεατή, όπως της επιλογής από έναν κυκλώνα προσφερόμενων τηλεοπτικών προγραμμάτων. Ο χρήστης έχει μέσω του TV-Anytime την δυνατότητα να ορίζει τις προτιμήσεις του. Κάποιο σύστημα που έχει αυτήν την πληροφορία (τις προτιμήσεις του χρήστη) θα είναι σε θέση να προτείνει, με αυτοματοποιημένο τρόπο, εξατομικευμένα στον χρήστη τα προγράμματα που ταιριάζουν καλύτερα στις προτιμήσεις του.

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

Βάση των συστημάτων που θα αναπτυχθούν 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 a.k.a 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 προσωποποιημένης πρόσβασης σε τηλεοπτικές μεταδόσεις ανεξάρτητα από το χρόνο, αποθηκεύοντας για μετέπειτα χρήση ολόκληρα τηλεοπτικά προγράμματα 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 προκαθορισμένο προφίλ προτιμήσεων του χρήστη, επιτρέποντας έτσι τη βέλτιστη χρήση του αποθηκευτικού χώρου, «κρατώντας» μόνο υλικό άμεσου ενδιαφέροντος. Τα μεταδεδομένα μεταδίδονται είτε ταυτόχρονα με την πληροφορία είτε πριν από αυτήν, 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 οι συσκευή αναλαμβάνει στη συνέχεια να αποθηκεύσει τοπικά.

Το πρόγραμμα UP-TV επιδιώκει να επεκτείνει τη λειτουργικότητα της παραπάνω βασικής αρχιτεκτονικής, εισάγοντας δίκτυα κατανεμημένων 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 καναλιών. Το σύστημα θα 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 θα 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 υποστηρίξει χρήστες που αλλάζουν γεωγραφική θέση και που συνεπώς επιθυμούν να έχουν γρήγορη πρόσβαση στο υλικό που τους ενδιαφέρει από τον ελάχιστοτε τόπο διαμονής τους.

Τα αντικείμενα ενδιαφέροντος που θα αποθηκεύονται, μπορούν να είναι είτε ολόκληρα τηλεοπτικά προγράμματα, είτε τμήματα που αντιστοιχούν σε συγκεκριμένες νοηματικές ενότητες. Το σύστημα, θα δύναται για παράδειγμα να διαχειριστεί από ένα δελτίο ειδήσεων τα αθλητικά ή τις πολιτικές ειδήσεις, από έναν αγώνα ποδοσφαίρου τα ενδιαφέροντα γεγονότα ή από μια ταινία τις ενδιαφέρουσες σκηνές. Η λειτουργικότητα αυτή, απαιτεί την εφαρμογή τεχνικών ανάκτησης οπτικοακουστικού υλικού βάσει περιεχομένου και θέτει ιδιαίτερες απαιτήσεις στον σχεδιασμό των μεταδεδομένων του συστήματος. Στο σύστημα UP-TV, τα μεταδεδομένα επιτρέπουν τη δεικτοδότηση του διαθέσιμου υλικού βάσει του περιεχομένου του, θέτοντας τη βάση για τη μη γραμμική περιήγηση στα περιεχόμενα ενός προγράμματος (ανά σκηνή, θέμα ή γεγονός ενδιαφέροντος), αλλά και για την αυτοματοποιημένη δημιουργία περιλήψεων.

Στόχοι της Εργασίας

Η εργασία αυτή πραγματοποιήθηκε στα πλαίσια του προγράμματος UP-TV και έχει τέσσερις στόχους:

- Την μελέτη της δομής και της σημασιολογίας των προτιμήσεων διήθησης (filtering and search preferences) του MPEG7/TVA και τον προσδιορισμό της αντιστοιχίας τους σε λογικές ερωτήσεις.
- Την υλοποίηση μηχανισμού προτάσεων προγραμμάτων προς τους χρήστες βασισμένες στο προφίλ τους και το περιεχόμενο (Content based Filtering)
- Την υλοποίηση μηχανισμών συνεργατικής αναζήτησης (Collaboration Filtering) και
- Την υλοποίηση ενός γραφικού περιβάλλοντος που θα δίνει την δυνατότητα στους χρήστες να διαχειρίζονται το προφίλ τους.

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

- Ως προς τον μηχανισμό προτάσεων προγραμμάτων προς τους χρήστες βασισμένο στο περιεχόμενο, πρέπει να
 - Υποστηρίζει το σύνολο της δομής των μεταδεδομένων του προφίλ των χρηστών καθώς και των προγραμμάτων του TV-Anytime.
 - Είναι όσο το δυνατόν πιο αποδοτικός και γρήγορος στην ανάκτηση της πληροφορίας.

- Ως προς τους μηχανισμούς συνεργατικής αναζήτησης, αυτοί πρέπει να
 - Δίνουν αποτελέσματα βασισμένοι τόσο στα προγράμματα που είδε ο χρήστης.
 - Υπερικεράσουν προβλήματα προσωπικών δεδομένων και λίγων δεδομένων.
- Ως προς την γραφική εφαρμογή, αυτή πρέπει να
 - Υποστηρίζει το σύνολο της δομής των μεταδεδομένων του προφίλ των χρηστών του TV-Anytime
 - Εισάγει και να εξάγει πληροφορία σε μορφή εγγράφων XML, συμβατών με το TV-Anytime, καθώς επίσης και στην βάση δεδομένων του UP-TV.
 - “Κρύβει” την πολυπλοκότητα της δομής των μεταδεδομένων, όσο γίνεται, από τον χρήστη και να του παρουσιάζει μία ευκολονόητη εικόνα
 - Είναι εύχρηστο και να ακολουθεί βασικές αρχές User Interface ώστε να μην κουράζει τον χρήστη αλλά να τον βοηθά όσο το δυνατόν περισσότερο στην εργασία του
 - Είναι αξιόπιστο και ανθεκτικό σε πιθανά λάθη του χρήστη
 - Παρέχει μία χρήσιμη βοήθεια στον χρήστη ώστε να μπορεί εύκολα να του εξηγηθεί οποιαδήποτε πιθανή απορία.

Δομή της Εργασίας

Στο δεύτερο (II) κεφάλαιο αναλύονται οι τεχνολογίες που χρησιμοποιήθηκαν και αναφέρονται συσχετιζόμενες εργασίες. Συγκεκριμένα, αναλύεται το μοντέλο μεταδεδομένων του TVA, όσο αφορά τόσο τα μεταδεδομένα των προγραμμάτων όσο και του προφίλ των χρηστών με

μεγαλύτερη έμφαση στο δεύτερο. Επίσης, περιγράφεται και το Query by Example, το οποίο χρησιμοποιείται στην κατασκευή του γραφικού περιβάλλοντος.

Στο τρίτο (III) κεφάλαιο περιγράφεται εκτενώς το σχήμα μεταδεδομένων του TV-Anytime. Πρόκειται για τα μεταδεδομένα που περιγράφουν τόσο προγράμματα όσο προτιμήσεις χρηστών. Στο τέλος μελετούνται οι αντιστοιχίες μεταξύ τους όπως θα χρησιμοποιηθούν από τους μηχανισμούς που αναπτύσσονται στο επόμενο κεφάλαιο.

Στο τέταρτο (IV) κεφάλαιο περιγράφεται ο σχεδιασμός και η υλοποίηση του μηχανισμού προτάσεων προγραμμάτων στους χρήστες με βάση το προφίλ τους και στο περιεχόμενο καθώς επίσης και οι μηχανισμοί της συνεργατικής αναζήτησης.

Στο πέμπτο (V) κεφάλαιο περιγράφεται το γραφικό περιβάλλον για την διαχείριση προφίλ από τους χρήστες.

Στο έκτο (VI) και τελευταίο κεφάλαιο γίνεται μία ανακεφαλαίωση, αναφέρονται κάποια συμπεράσματα καθώς επίσης γίνονται και κάποιες επισημάνσεις για μελλοντικές επεκτάσεις.

*[...] οὕτω δει και τους παιδείας
ορεγομένους μηδενός μεν απείρως έχειν,
πανταχόθεν δε τα χρήσιμα συλλέγειν.*

Ισοκράτης

Κ ε φ ά λ α ι ο Ι Ι

ΤΕΧΝΟΛΟΓΙΕΣ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ – ΣΥΣΧΕΤΙΖΟΜΕΝΕΣ ΕΡΓΑΣΙΕΣ

Εισαγωγή

Αυτό το κεφάλαιο ασχολείται, αρχικά, με τις τεχνολογίες που μελετήθηκαν και χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής. Όσο κατά την θεωρητική μελέτη, τόσο και κατά τον σχεδιασμό και την υλοποίηση της εφαρμογής, μελετήθηκαν και χρησιμοποιήθηκαν ένα πλήθος τεχνολογιών, όπως, MPEG-7, TV-Anytime, Query by Example και άλλες. Ακολουθεί μία μικρή αναφορά σε όλες εκείνες που είναι απαραίτητες ώστε να γίνουν κατανοητά τα επόμενα κεφάλαια.

Κατόπιν, παρουσιάζονται κάποιες σχετικές εργασίες και εφαρμογές που έχουν ήδη αναπτυχθεί στον τομέα της ψηφιακής τηλεόρασης και γίνεται σύγκριση με την παρούσα εργασία.

MPEG-7

Το MPEG-7 είναι ένα πρότυπο μεταδεδομένων, δημιουργημένο από την επιτροπή Moving Picture Experts Group, για να καλύψει τις αυξανόμενες ανάγκες για μια αυστηρά ορισμένη περιγραφή οπτικοακουστικού περιεχομένου σε περιβάλλον πολυμέσων, με όλα τα πλεονεκτήματα που προσφέρει κάτι τέτοιο στην ταχύτερη ανάπτυξη συστημάτων ανάκτησης

και πλοήγησης στο υλικό καθώς και τη δυνατότητα επικοινωνίας μεταξύ διαφορετικών συστημάτων. Δεν έρχεται να αντικαταστήσει προηγούμενα πρότυπα για οπτικοακουστικό υλικό, όπως το MPEG-1, MPEG-2 και MPEG-4, αλλά να προσφέρει μία υψηλότερου επιπέδου περιγραφή με δεικτοδότηση προς αυτά. Για παράδειγμα, ίσως η περιγραφή ενός σχήματος σε MPEG-4 να είναι χρήσιμη και στο MPEG-7, όπως και ένα διάγραμμα κίνησης από τα MPEG-1 ή και MPEG-2.

Με τον όρο ‘περιγραφή οπτικοακουστικού περιεχομένου σε περιβάλλον πολυμέσων’ αναφερόμαστε στην περιγραφή εικόνας, ήχου, λόγου, γραφικών, τρισδιάστατων μοντέλων, video, αλλά και την περιγραφή για το πως κάποια από τα παραπάνω συσχετίζονται μεταξύ τους για να παράγουν το τελικό αποτέλεσμα. Η πληροφορία που παρέχεται από το MPEG-7 ανάγεται στους τομείς:

- Της δημιουργίας και των διαδικασιών παραγωγής του υλικού (π.χ. σκηνοθέτης, τίτλος, ταινία μικρού μήκους).
- Του συσχετιζόμενου με το περιεχόμενο υλικό (copyright, ιστορία χρήσης, πρόγραμμα μετάδοσης).
- Των χαρακτηριστικών αποθήκευσης του περιεχομένου (format, κωδικοποίηση)
- Της δομής του περιεχομένου και των επιμέρους τμημάτων του (σκηνές, τμηματοποίηση του video)
- Των χαμηλού επιπέδου χαρακτηριστικών του περιεχομένου (χρώμα, υφή της εικόνας)
- Της σημασιολογικής πληροφορίας από την ‘πραγματικότητα’ που απεικονίζεται στο περιεχόμενο (αντικείμενα, πρόσωπα, σχέσεις μεταξύ τους)
- Της πλοήγησης στο περιεχόμενο
- Των συλλογών αντικειμένων
- Της αλληλεπίδρασης του χρήστη με το περιεχόμενο.

Όλες αυτές οι περιγραφές δομούνται σε XML έγγραφα που συνοδεύουν και δεικτοδοτούν το υλικό και που, όπως προαναφέρθηκε, υπακούουν σε ένα σχήμα δομημένο με βάση την XML Schema.

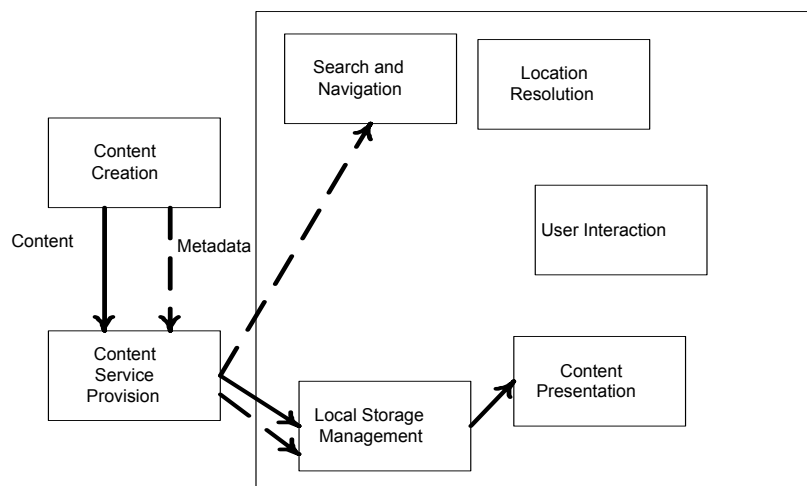
Στη συγκεκριμένη διατριβή παρουσιάζει ενδιαφέρον ο τρόπος με τον οποίο το MPEG7 περιγράφει τη σημασιολογική πληροφορία που υπάρχει στο οπτικοακουστικό υλικό καθώς επίσης και την πληροφορία για την αλληλεπίδραση του χρήστη με το περιεχόμενο. Με τον όρο “σημασιολογική πληροφορία” εννοείται την πληροφορία που θα μας έδινε ένας άνθρωπος αν τον ρωτούσαμε τι υπάρχει σε κάποιο οπτικοακουστικό υλικό ή απλά την πληροφορία που έχει ένας οδηγός τηλεόρασης. Για παράδειγμα, για μία ταινία, θα μπορούσαμε να ενημερωθούμε για τα πρόσωπα, πράγματα και συμβάντα που παρουσιάζει, αλλά και τις σχέσεις μεταξύ τους και τις έννοιες και καταστάσεις που πηγάζουν μέσα από αυτά. Το πρόβλημα που προκύπτει είναι ότι κάθε άνθρωπος θα επέλεγε διαφορετική γλώσσα, σύνταξη και λεξιλόγιο, καθιστώντας αδύνατη την αποδοτική ανάκτηση βάσει σημασιολογικού περιεχομένου στην περίπτωση που συνδέονταν απλά ένα κείμενο με το οπτικοακουστικό αντικείμενο. Από την άλλη η αλληλεπίδραση του χρήστη με το περιεχόμενο επιτυγχάνεται με την διατήρηση πληροφορίας τόσο για τις προτιμήσεις του χρήστη όσο και για το ιστορικό χρήσης του, οι οποίες να μπορούν να συσχετιστούν με την σημασιολογική πληροφορία. Έτσι, το MPEG-7 παρέχει δομές ώστε αφενός το οπτικοακουστικό μέσο να περιγράφεται ικανά και με ενιαία μορφή και αφετέρου να καθίσταται δυνατή η ανάκτηση του οπτικοακουστικού μέσου αυτοματοποιημένα μεν αλλά εξατομικευμένα.

To TV-Anytime

Το TV-Anytime Forum έχει εκδώσει ένα σύνολο από πρότυπα (specifications) που περιγράφουν την οργάνωση και τη λειτουργία συστημάτων ψηφιακής τηλεόρασης.

Ένα σύστημα TVA περιλαμβάνει τρία βασικά δομικά στοιχεία: Τον παροχέα των TVA υπηρεσιών (Service Provider), τον παροχέα των μηχανισμών πρόσβασης στο περιεχόμενο (transport provider-για παράδειγμα ο παροχέας των τηλεπικοινωνιακών δικτύων και μηχανισμών) και την αποθηκευτική συσκευή που χρησιμοποιείται από τον τελικό χρήστη, η

οποία αποθηκεύει το οπτικοακουστικό υλικό και αναλαμβάνει την αναπαραγωγή του. Όλα αυτά φαίνονται στην παρακάτω εικόνα όπου τα εικονιζόμενα παραλληλόγραμμα παρουσιάζουν μια λογική κατάτμηση ενός TVA συστήματος σε επιμέρους υποσυστήματα.



Σχήμα 1: Λογική κατάτμηση ενός TV-Anytime συστήματος σε επιμέρους υποσυστήματα

Η λειτουργικότητα «δημιουργία του υλικού» (content creation) παρέχεται από κινηματογραφικά στούντιο και εταιρίες παραγωγής οπτικοακουστικών προγραμμάτων. Εκφράζει την πηγή του υλικού πάνω στο οποίο χτίζονται οι TVA υπηρεσίες.

Η «παροχή υπηρεσιών πάνω στο υλικό» (content service provider) πραγματοποιείται από οργανισμούς (π.χ. τηλεοπτικά κανάλια) που αναλαμβάνουν την προετοιμασία του υλικού για μετάδοση, την προσθήκη μεταδεδομένων και την διάθεση του υλικού στους τελικούς χρήστες.

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 a.k.a Personal Digital Recorders-PVRs) 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 αντικειμένου 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.

Μεταδεδομένα στο TV-Anytime

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

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

Το TV-Anytime για να περιγράψει με σαφήνεια τη δομή των μεταδεδομένων αυτών έχει χρησιμοποιήσει 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, το ενδιαφέρον εστιάστηκε κυρίως στα μεταδεδομένα περιγραφής των προτιμήσεων των χρηστών αλλά και σε αυτά της περιγραφής του περιεχομένου καθώς επίσης και στις μεταξύ τους αντιστοιχίσεις. Αυτά θα περιγραφούν εκτενώς στο επόμενο κεφάλαιο.

To Query by Example

Το Query by Example (QBE) είναι ένας πολύ απλός τρόπος να δοθεί στον χρήστη η δυνατότητα να εισάγει ένα σύνολο από κριτήρια ώστε να απευθύνει μία ερώτηση μετά σε ένα σύνολο δεδομένων (παράδειγμα ενός τέτοιου συνόλου είναι ένα πίνακας μίας βάσης δεδομένων ή ένας αριθμός από στήλες με δεδομένα στο MS Excel) .

Η βασική του ιδέα είναι ότι μία ερώτηση προς το σύνολο των δεδομένων αποτελείται από and, or και διάφορα conditions – που μπορεί να περιλαμβάνουν από απλούς τελεστές (όπως <, >, <=, not, like, κλπ) έως aggregate functions (όπως sum, min, count, κλπ) σε συνδυασμό με απλούς τελεστές (π.χ. Sum(Salary) < 1000€).

Για τις ανάγκες της περιγραφής του QBE θα δώσουμε ένα σύνολο δεδομένων ώστε παρακάτω να γίνονται πιο ευνόητα τα παραδείγματα. Ένα ενδεικτικό σύνολο δεδομένων παρουσιάζεται στον επόμενο πίνακα και προέρχεται από πωλήσεις ασφαλειών:

Salesperson	Type	Sales
Davolio	Life	2000
Davolio	Fire	100
Davolio	Car	5500
Buchanan	Life	1100
Buchanan	Fire	1000
Buchanan	Car	3000
Suyama	Life	500
Smith	Car	2500

Πίνακας 1: Δεδομένα από ασφαλιστική εταιρία αναφορικά με τους πωλητές ασφαλειών.

Το interface βασίζεται στην έννοια του πίνακα. Συγκεκριμένα τα κριτήρια εισάγονται για σε διαφορετική γραμμή όταν πρόκειται για OR. Δηλαδή ο παρακάτω πίνακας σημαίνει ότι θέλω τα δεδομένα που οι πωλητές τους είναι Davolio OR Buchanan OR Suyama:

Salesperson
Davolio
Buchanan
Suyama

Πίνακας 2: 1^ο παράδειγμα QBE

Τα αποτελέσματα που προκύπτουν είναι:

Salesperson	Type	Sales
Davolio	Life	2000
Davolio	Fire	100
Davolio	Car	5500
Buchanan	Life	1100
Buchanan	Fire	1000
Buchanan	Car	3000
Suyama	Life	500

Πίνακας 3: Αποτελέσματα 1^{ου} παραδείγματος QBE

Εισάγονται, δε πάλι, στην ίδια σειρά όταν έχουμε AND. Δηλαδή ο παρακάτω πίνακας σημαίνει ότι θέλω τα δεδομένα που ο πωλητής είναι ο Davolio AND ο τύπος είναι Life AND οι πωλήσεις ξεπερνούν τα 1000.

Type	Salesperson	Sales
Life	Davolio	>1000

Πίνακας 4: 2^ο παράδειγμα QBE

Τα αποτελέσματα που προκύπτουν είναι:

Salesperson	Type	Sales
Davolio	Life	2000
Davolio	Fire	100
Davolio	Car	5500
Buchanan	Life	1100
Buchanan	Fire	1000
Buchanan	Car	3000
Suyama	Life	500
Smith	Car	2500

Πίνακας 5: : Αποτελέσματα 2^{ου} παραδείγματος QBE

Είναι, πλέον, φανερό ότι ο παρακάτω πίνακας σημαίνει ότι θέλω τα δεδομένα που ο πωλητής είναι ο Davolio OR ο τύπος είναι Life OR οι πωλήσεις ξεπερνούν τα 1000.

Type	Salesperson	Sales
Life		
	Davolio	
		>1000

Πίνακας 6: 3^ο παράδειγμα QBE

Τα αποτελέσματα που προκύπτουν είναι:

Salesperson	Type	Sales
Davolio	Life	2000
Davolio	Fire	100
Davolio	Car	5500
Buchanan	Life	1100
Buchanan	Fire	1000
Buchanan	Car	3000
Suyama	Life	500
Smith	Car	2500

Πίνακας 7: Αποτελέσματα 3^{ου} παραδείγματος QBE

Παρακάτω φαίνεται η χρήση της aggregate function average. Είναι προφανές ότι θέλω τα δεδομένα που οι πωλήσεις τους είναι μεγαλύτερες από τον μέσο όρο.

Sales
> AVERAGE(Sales)

Πίνακας 8: 4^ο παράδειγμα QBE

Τα αποτελέσματα που προκύπτουν είναι:

Salesperson	Type	Sales
Davolio	Life	2000
Davolio	Fire	100
Davolio	Car	5500
Buchanan	Life	1100
Buchanan	Fire	1000
Buchanan	Car	3000
Suyama	Life	500
Smith	Car	2500

Πίνακας 9: Αποτελέσματα 4^{ου} παραδείγματος QBE. Ο μέσος όρος των πωλήσεων είναι 1962

Μία ακόμη πολύ ενδιαφέρουσα δυνατότητα είναι η χρήση του ίδιου attribute περισσότερες από μια φορές. Φαίνεται αρκετά ξεκάθαρα με τον παρακάτω πίνακα.

Sales	Sales
>5000	<8000
<500	

Πίνακας 10: 5^ο παράδειγμα QBE

Τα αποτελέσματα που προκύπτουν είναι:

Salesperson	Type	Sales
Davolio	Life	2000
Davolio	Fire	100
Davolio	Car	5500
Buchanan	Life	4100
Buchanan	Fire	4000
Buchanan	Car	3000
Suyama	Life	500
Smith	Car	2500

Πίνακας 11: Αποτελέσματα 5^{ου} παραδείγματος QBE.

Αν συνυπολογίσει κανείς και την δυνατότητα να κάνει χρήση του παραπάνω σαν multi-valued attribute όπως υποδεικνύει το παράδειγμα, τότε φθάνουμε στο συμπέρασμα ότι το QBE είναι ένα πανίσχυρο εργαλείο με παροιμιακή απλότητα όσον αφορά την χρησιμοποίησή του. Φαίνεται παρακάτω μία αρκετά περίπλοκη ερώτηση αλλά ο αναγνώστης μπορεί κατανοεί γρήγορα πια τι αποτελέσματα θέλει να του επιστραφούν:

Type	Salesperson	Sales	
Car	Davolio	>5000	<8000
		<500	

Πίνακας 12: 6^ο παράδειγμα QBE

Τα αποτελέσματα που προκύπτουν είναι:

Salesperson	Type	Sales
Davolio	Life	2000
Davolio	Fire	400
Davolio	Car	5500
Buchanan	Life	4100
Buchanan	Fire	4000
Buchanan	Car	3000
Suyama	Life	500
Smith	Car	2500

Πίνακας 13: Αποτελέσματα 6^{ου} παραδείγματος QBE.

Συσχετιζόμενες Εργασίες

Σε αυτήν την παράγραφο θα παρουσιαστούν εργασίες που σχετίζονται με την δουλειά που έγινε στην παρούσα διατριβή. Λόγω της πληθώρας των υλοποιήσεων που έχουν γίνει πάνω σε διήθηση πληροφοριών θα γίνει σύγκριση μόνο με τις εργασίες που υλοποιήθηκαν για περιβάλλοντα ψηφιακής τηλεόρασης. Αρχικά θα παρουσιαστούν τα κύρια χαρακτηριστικά κάθε μίας και μετά θα γίνει η σύγκρισή.

Σερφάροντας στο Ψηφιακό Κύμα (Surfing the Digital Wave)

Σε αυτήν την εργασία υλοποιήθηκε ένα σύστημα το οποίο δημιουργεί εξατομικευμένους τηλεοπτικούς καταλόγους χρησιμοποιώντας συνεργατικές (Collaborative) και βασισμένες σε περιπτώσεις (Case-Based) προτάσεις. Το σύστημα για να εξάγει τις προτάσεις υλοποιεί ένα υβριδικό σύστημα αποτελούμενο από δύο κομμάτια. Το πρώτο κομμάτι προτείνει προγράμματα που άρεσαν σε όμοιους χρήστες (συνεργατικές προτάσεις) και το άλλο προγράμματα όμοια με αυτά που είδε ο χρήστης στο παρελθόν και του άρεσαν (βασισμένο σε περιπτώσεις). Επίσης, υποστηρίζεται περιορισμένης έκτασης προφίλ χρηστών και ένα γραφικό περιβάλλον που δίνει την δυνατότητα στους χρήστες να το διαχειριστούν. Για περισσότερες λεπτομέρειες ο αναγνώστης παραπέμπετε στο [27].

Εξερευνητής Τηλεόρασης (TV Scout)

Το TV Scout είναι ένα σύστημα τηλεοπτικών προτάσεων βασισμένο στο παγκόσμιο ιστό. Παρουσιάζεται στους χρήστες όχι σαν ένα σύστημα διήθησης, αλλά σαν ένα σύστημα ανάκτησης όπου όλη η προσπάθεια των χρηστών οδηγεί σε άμεσα αποτελέσματα, δηλ. οι χρήστες απλά επιλέγουν τα προγράμματα που θέλουν να δουν. Όσο οι χρήστες ασχολούνται με αυτό, το σύστημα συγκεντρώνει πληροφορίες για αυτούς για να μετατραπεί τελικά σε ένα σύστημα διήθησης. Για περισσότερες λεπτομέρειες ο αναγνώστης παραπέμπετε στο [28].

Εξατομικευμένες Προτάσεις Τηλεοπτικών Προγραμμάτων (Personalized TV Show Recommender)

Εδώ αναπτύχθηκε ένα εξατομικευμένο σύστημα που προτείνει στους χρήστες τηλεοπτικά προγράμματα βασισμένο στις προτιμήσεις τους. Εκτός από το σύστημα που κάνει τις προτάσεις κατασκευάστηκε και ένα γραφικό περιβάλλον που αξιοποιεί όλες τις δυνατότητες του μηχανισμού των προτάσεων. Ο μηχανισμός προτάσεων βασίζεται σε προφίλ χρηστών τα οποία αλλάζουν είτε από τους ίδιους τους χρήστες, μέσω του γραφικού περιβάλλοντος, είτε από το σύστημα βάση του ιστορικού χρήσης και χρησιμοποιεί νευρωνικά δίκτυα για υπολογίσει τα αποτελέσματα. Τα προφίλ των χρηστών περιλαμβάνουν λίγα στοιχεία σε σχέση με τα προτεινόμενα από το TV-Anytime. Για περισσότερες λεπτομέρειες ο αναγνώστης παραπέμπεται στο [29].

Εξατομίκευση στην Ψηφιακή Τηλεόραση (Personalization in Digital TV)

Σε αυτήν την εργασία κατασκευάστηκε ένα σύστημα βασισμένο σε πράκτορες (Agents) για εξατομίκευση σε περιβάλλοντα ψηφιακής τηλεόρασης. Συγκεκριμένα το σύστημα παρέχει μέσα από μηχανισμούς διήθησης περιεχομένου (τόσο βασισμένους στο περιεχόμενο όσο και συνεργατικούς) προτάσεις προς τους χρήστες. Οι χρήστες διατηρούν προφίλ το οποίο αλλάζει από το σύστημα ανάλογα με τις αλληλεπιδράσεις του χρήστη με το σύστημα και από ερωτήσεις που γίνονται στον χρήστη από το σύστημα. Έχει κατασκευαστεί, επίσης, ένα εξατομικευμένο γραφικό περιβάλλον το οποίο προσφέρει υπηρεσίες στους χρήστες ανάλογα με τις ανάγκες τους (π.χ. για τα παιδιά και τους ηλικιωμένους προσφέρονται απλοποιημένα περιβάλλοντα). Επίσης, χρησιμοποιείται μικρό κομμάτι από το TV-Anytime (μόνο τίτλος και κατηγορία από τα μεταδεδομένα περιγραφή περιεχομένου). Για περισσότερες λεπτομέρειες ο αναγνώστης παραπέμπεται στο [30].

Σύγκριση

Στον παρακάτω πίνακα φαίνονται οι πέντε κύριοι τομείς όπου έγινε δουλειά στην εργασία αυτή και οι τέσσερις εργασίες που παρουσιάστηκαν παραπάνω. Για κάθε μία εργασία φαίνεται κατά πόσον υποστήριξαν αυτούς τους τομείς.

	Surfing the Digital Wave	TV Scout	Personalized TV Show Recommender	Personalization in Digital TV
Υποστήριξη προφίλ χρηστών	Περ. ¹	Ναι	Ναι	Ναι
Διήθηση περιεχομένου	Όχι	Ναι	Ναι	Ναι
Συνεργατική διήθηση	Ναι	Όχι	Όχι	Ναι
Περιβάλλον κατασκευής προφίλ	Περ. ¹	Ναι	Ναι	Όχι
Υποστήριξη TV-Anytime	Όχι	Όχι	Όχι	Περ. ¹

Πίνακας 14: Συγκριτικός πίνακας των μερών της παρούσας διατριβής με συσχετιζόμενες εργασίες.

Περίληψη

Σε αυτό το κεφάλαιο μελετήθηκαν υπάρχοντα συστήματα που έχουν αναπτυχθεί στο παρελθόν με παρεμφερή θεματολογία με την παρούσα διατριβή και επίσης τεχνολογίες που χρησιμοποιήθηκαν από την εργασία αυτή. Αυτές είναι το Query by Example, MPEG7 ως υπερσύνολο του TV-Anytime και το ίδιο TV-Anytime το οποίο αναλύεται σε μεγαλύτερη έκταση στην επόμενο κεφάλαιο.

¹ Περιορισμένης έκτασης υποστήριξη

*Δος μοι πα στω και
ταν γαν κινάσσω*

Αρχιμήδης

Κε φ á λ α ι ο Ι Ι Ι

ΤΟ ΣΧΗΜΑ ΜΕΤΑΔΕΔΟΜΕΝΩΝ ΤΟΥ TV-ANYTIME

Εισαγωγή

Σε αυτό το κεφάλαιο θα παρουσιαστεί το σχήμα μεταδεδομένων TV-Anytime. Σκοπός του είναι κυρίως να περιγράψει το μοντέλο για το προφίλ των χρηστών και τις αντιστοιχίσεις των τύπων χαρακτηριστικών, οι οποίοι συνιστούν το προφίλ, με τους τύπους χαρακτηριστικών των οπτικοακουστικών μέσων, τα οποία εφεξής θα καλούνται απλά προγράμματα. Για να γίνουν κατανοητές όμως οι αντιστοιχίσεις θα αναπτυχθεί και το μοντέλο του TV-Anytime για εκείνα τα μεταδεδομένα των προγραμμάτων που γίνεται η αντιστοίχιση.

Μεταδεδομένα των Περιγραφής Περιεχομένου

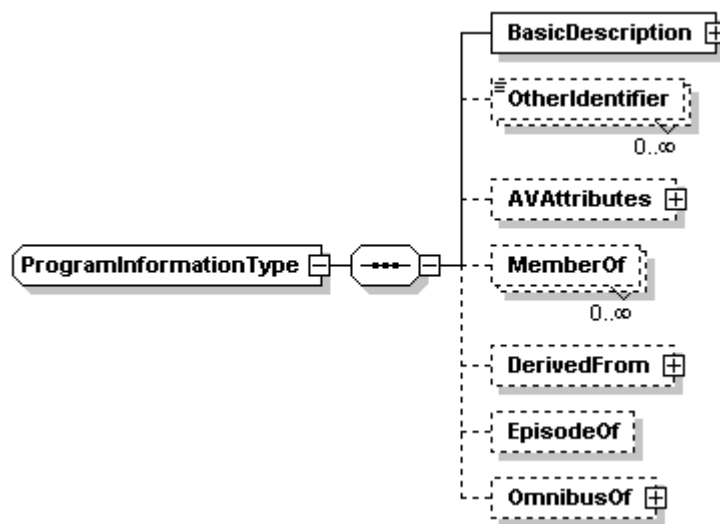
Κάθε πρόγραμμα έχει ένα σύνολο από μεταδεδομένα που το περιγράφουν. Το TV-Anytime προτείνει τη δομή καθώς και το είδος των μεταδεδομένων που θα κρατούνται ώστε αυτή η περιγραφή να είναι επαρκής για όλες τις εφαρμογές που υλοποιούν περιβάλλοντα ψηφιακής τηλεόρασης.

Είναι απαραίτητο να εξεταστεί το είδος και η δομή των μεταδεδομένων που κρατούνται ώστε αργότερα να μπορούν να γίνουν οι απαραίτητες συσχετίσεις με τα επιμέρους στοιχεία του προφίλ των χρηστών. Επειδή είναι αρκετά εκτενές το σχήμα των μεταδεδομένων των προγραμμάτων, θα περιγράψουν μόνον όσα στοιχεία είναι απαραίτητα για την κατανόηση της

ανάλυσης που θα ακολουθήσει στην τελευταία παράγραφο. Για τα υπόλοιπα ο φιλομαθής αναγνώστης παραπέμπεται στο TV-Anytime Metadata Specification [8] και στο MPEG-7 Metadata Specification [9].

ProgramInformation

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

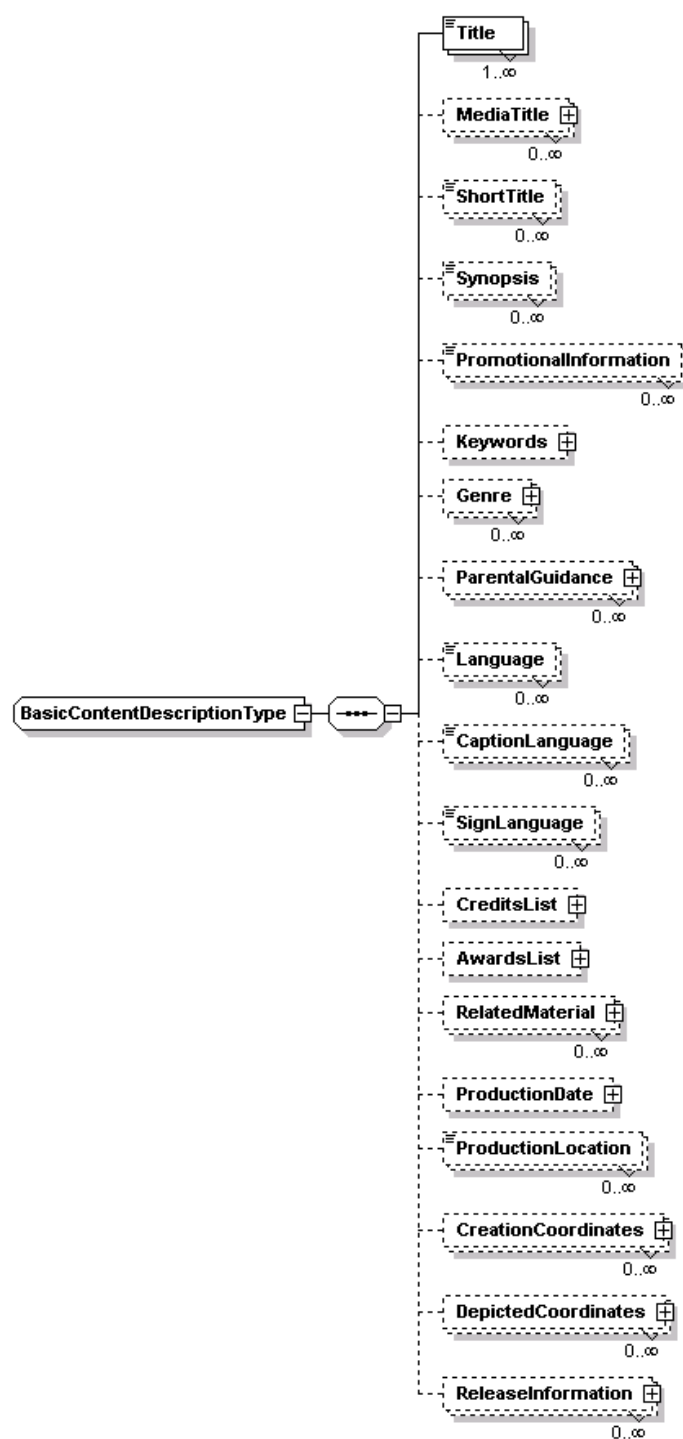


Τα στοιχεία που θα χρειαστούν περιγράφονται πιο αναλυτικά παρακάτω:

- BasicDescription (Η βασική περιγραφή του προγράμματος)
- AVAttributes (Οπτικοακουστικά χαρακτηριστικά του προγράμματος όπως αυτό δημιουργήθηκε)

BasicDescription

Από αυτά θα εξεταστεί πιο αναλυτικά το BasicDescription όπου βρίσκεται η βασική περιγραφή του προγράμματος. Σχηματικά φαίνεται η δομή του στην επόμενη σελίδα.



Αναλυτικά τα στοιχεία που χρησιμοποιούνται επεξηγούνται παρακάτω:

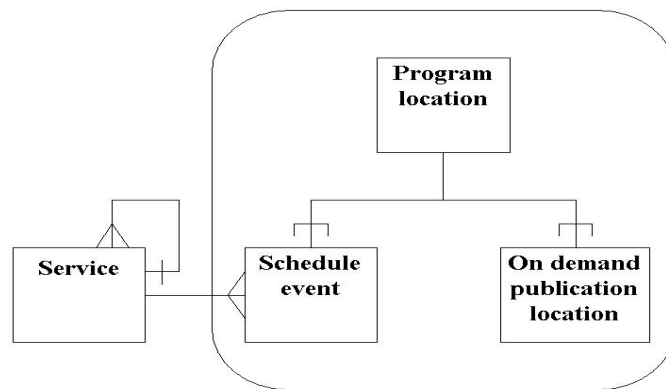
- Title (ο τίτλος του προγράμματος. Ένα πρόγραμμα μπορεί να έχει περισσότερους από ένα τίτλους, π.χ. σε διαφορετικές γλώσσες)

- ShortTitle (Σύντομος τίτλος που χρησιμοποιείται για λόγους παρουσίασης)
- 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)
- ParentalGuidance (Ένας κωδικός που αφορά γονική αξιολόγηση)
- Language (Περιγράφει μία ομιλούσα γλώσσα του προγράμματος)
- CaptionLanguage (Αναφέρεται σε μία γλώσσα υποτίτλων)
- SignLanguage (Περιγράφει τη γλώσσα κωφάλαλων του προγράμματος και κατά πόσον αυτή είναι η κύρια γλώσσα του προγράμματος ή μετάφραση)
- CreditsList (Η λίστα με τα ονόματα των δημιουργών και των συνεργατών (π.χ. ηθοποιοί, σκηνοθέτες, κ.λπ.) του προγράμματος.
- AwardList (Μία λίστα με επαίνους και βραβεία που αξιώθηκε το πρόγραμμα)
- ProductionDate (Η ημερομηνία ή η χρονική περίοδος που παρήχθη το πρόγραμμα)
- ProductionLocation (Η χώρα που παρήχθη το πρόγραμμα)
- ReleaseInformation (Πληροφορίες σχετικά με την ημερομηνία και την χώρα που έγινε η διάθεση/ κυκλοφορία του προγράμματος).

Μεταδεδομένα Περιγραφής Στιγμιότυπων

Στην προηγούμενη ενότητα, εξετάστηκαν τα μεταδεδομένα περιγραφής προγραμμάτων, τα οποία συνδέουν μεταδεδομένα με ένα πρόγραμμα. Σε αυτήν τη ενότητα θα περιγραφούν μεταδεδομένα περιγραφής στιγμιότυπων. Τα μεταδεδομένα περιγραφής στιγμιότυπων συνδέουν το πρόγραμμα με μια περιγραφή για το γεγονός του στιγμιότυπου, π.χ. πότε και που συνέβη.

Το γενικό σχήμα που το περιγράφει είναι το ακόλουθο:



Σχήμα 2: Το σχήμα των μεταδεδομένων περιγραφής στιγμιότυπων

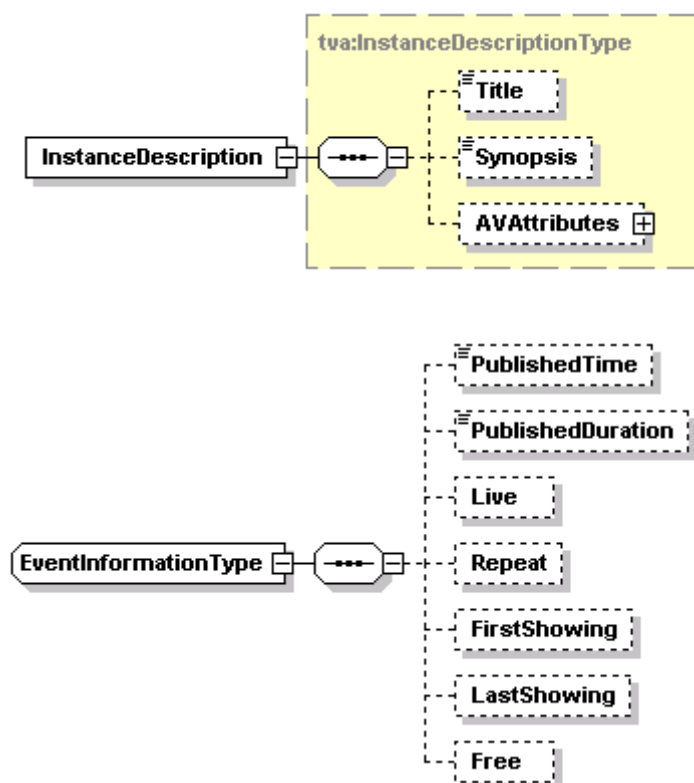
Υπάρχουν τρία στοιχεία που συνιστούν το σχήμα των μεταδεδομένων περιγραφής στιγμιότυπων:

- **Program location** – η τοποθεσία του προγράμματος αντικατοπτρίζει μια γενική τοποθεσία του προγράμματος, παραδείγματα είναι υπηρεσίες αναμετάδοσης και το δίκτυο.
- **Schedule event** – το χρονοδιάγραμμα γεγονότων είναι μια συγκεκριμένου τύπου τοποθεσία προγράμματος που περιγράφει τοποθεσίες αναμετάδοσης προγραμμάτων. Το χρονοδιάγραμμά γεγονότων συσχετίζει μία δεδομένη τοποθεσία αναμετάδοσης (υπηρεσία, χρόνος και διάρκεια) με ένα πρόγραμμα

- **Service** – Η υπηρεσία αντιπροσωπεύει μία διακριτή (αναφορικά με το περιεχόμενο) ροή υλικού αναμετάδοσης. Μια υπηρεσία διακινείται μέσα από κάποιες μορφές φυσικά κανάλια αλλά δεν είναι συνώνυμες οι έννοιες αφού μία υπηρεσία μπορεί να μεταδοθεί με μέσω μιας πληθώρας φυσικών καναλιών.

Χωρίς να εισέλθουμε σε εξαντλητικές λεπτομέρειες που ξεφεύγουν από τους σκοπούς του κεφαλαίου αυτού, θα περιγραφούν τα στοιχεία που διατηρούν τα μεταδεδομένα που μας ενδιαφέρουν και όχι ολόκληρη η δομή, την οποία ο αναγνώστης μπορεί να δει στο [8].

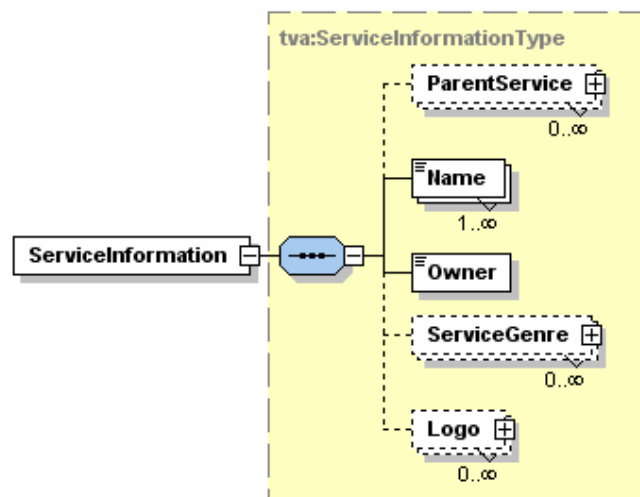
Το InstanceDescription περιγράφει πληροφορία που αφορά ένα στιγμιότυπο. Το EventInformation περιγράφει πληροφορία για ένα γεγονός αναμετάδοσης. Και τα δύο βρίσκονται στα στοιχεία Event, BroadcastEvent και onDemandBroadcastEvent τα οποία είναι κληρονόμοι του schedule event που είδαμε παραπάνω. Παρακάτω φαίνονται σχηματικά τα πεδία τους:



Αναλυτικά τα στοιχεία που χρησιμοποιούνται επεξηγούνται παρακάτω:

- Title (ο τίτλος του προγράμματος. Ένα πρόγραμμα μπορεί να έχει περισσότερους από ένα τίτλους, π.χ. σε διαφορετικές γλώσσες)
- Synopsis (Μία σύνοψη του προγράμματος (είναι σε μορφή κειμένου))
- AVAttributes (Οπτικοακουστικά χαρακτηριστικά του συγκεκριμένου στιγμιότυπου προγράμματος)
- PublishedTime (Η ώρα που έχει διαφημιστεί ότι αρχίζει η αναμετάδοση)
- PublishedDuration (Η διάρκεια που έχει διαφημιστεί ότι θα κρατήσει η αναμετάδοση)
- Repeat (Ένας δείκτης για το αν το πρόγραμμά είναι επανάληψη)
- Free (Ένας δείκτης για το αν το συγκεκριμένο στιγμιότυπο προγράμματος είναι δωρεάν)

Στη συνέχεια μελετάται η πληροφορία που περιγράφει ένα Service. Αυτή κρατείται στο ServiceInformation το οποίο φαίνεται παρακάτω:



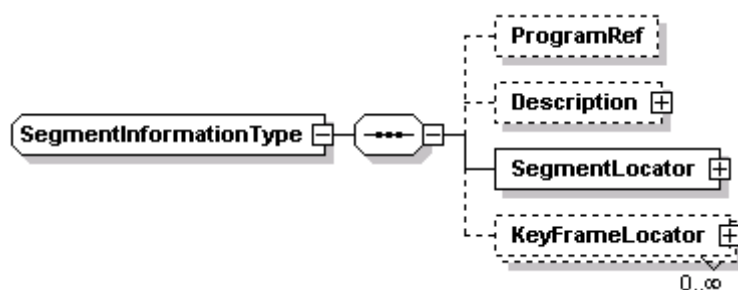
Αναλυτικά τα στοιχεία που χρησιμοποιούνται επεξηγούνται παρακάτω:

- Name (το όνομα της υπηρεσίας)
- Owner (Η φίρμα του ιδιοκτήτη της υπηρεσίας)

Μεταδεδομένα Περιγραφής Τμημάτων Προγραμμάτων

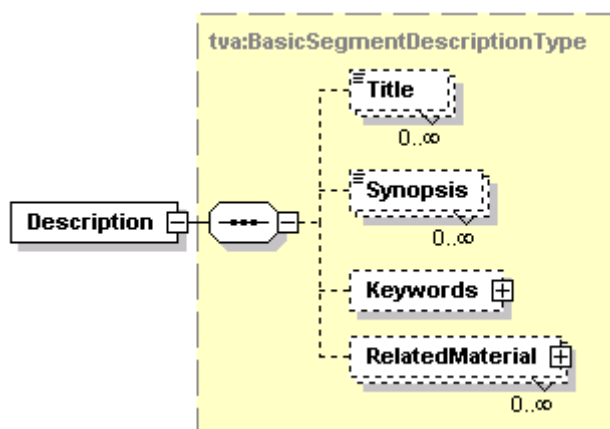
Τα μεταδεδομένα τμημάτων προγραμμάτων αφορούν την περιγραφή και δεικτοδότηση τμημάτων προγραμμάτων, ώστε για παράδειγμα να υπάρχει η πληροφορία ότι στο συγκεκριμένο κομμάτι ενός προγράμματος παίζει ένας συγκεκριμένος ηθοποιός. Εδώ θα περιγραφεί συνοπτικά το κομμάτι εκείνο που αφορά τα μεταδεδομένα που περιγράφουν το περιεχόμενο των τμημάτων.

Συγκεκριμένα έχουμε το παρακάτω σχήμα που περιγράφει την δομή των μεταδεδομένων ενός τμήματος προγράμματος:



Σχήμα 3: Σχήμα που περιγράφει την δομή των μεταδεδομένων ενός προγράμματος

Από τα οποία επιμέρους πεδία του SegmentInformation αυτό που διατηρεί όλη την πληροφορία για τα τμήματα είναι το Description το περιγράφεται σχηματικά πιο κάτω:



Σχήμα 4: Το στοιχείο Description, που κρατάει την πληροφορία για την περιγραφή τμημάτων προγραμμάτων

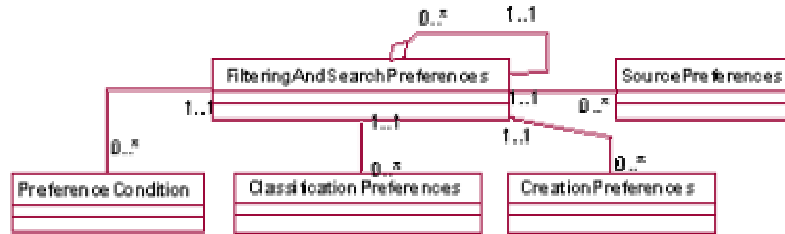
Αναλυτικά τα στοιχεία που χρησιμοποιούνται επεξηγούνται παρακάτω:

- Title (Ο τίτλος του τμήματος του προγράμματος. Ένα πρόγραμμα μπορεί να έχει περισσότερους από ένα τίτλους, π.χ. σε διαφορετικές γλώσσες)
- Synopsis (Μία σύνοψη του τμήματος του προγράμματος (είναι σε μορφή κειμένου))
- Keywords (Μία λίστα από λέξεις ή φράσεις που περιγράφουν επιγραμματικά το τμήμα του προγράμματος)

Μεταδεδομένα για τα Προφίλ των Χρηστών

Το προφίλ του χρήστη αποτελείται από φίλτρα τα οποία ονομάζονται Filtering and Search Preferences και σε αυτά διατηρούνται όλα τα δεδομένα που χρειάζονται για να περιγράψουν τις προτιμήσεις των χρηστών για τηλεοπτικά προγράμματα. Αυτά τα δεδομένα έχουν να κάνουν τόσο με το οπτικοακουστικό υλικό αυτό καθαυτό όσο και με και με δεδομένα καθαρά για εφαρμογές τηλεόρασης (π.χ. χρόνος αναμετάδοσης του προγράμματος). Παρακάτω θα περιγράψουμε λεπτομερώς το είδος των δεδομένων που κρατούνται καθώς και της δομής των φίλτρων.

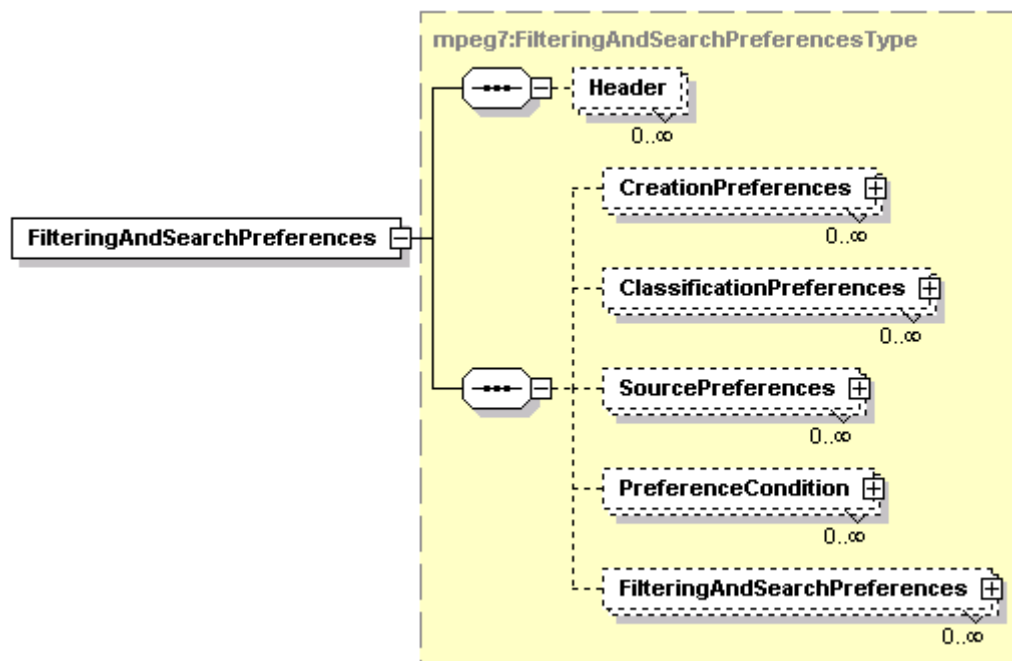
Το UML διάγραμμα που ακολουθεί περιγράφει την βασική δομή του Filtering and Search Preferences (FASP):



Σχήμα 5: UML διάγραμμα που περιγράφει την δομή του Filtering and Search Preferences.

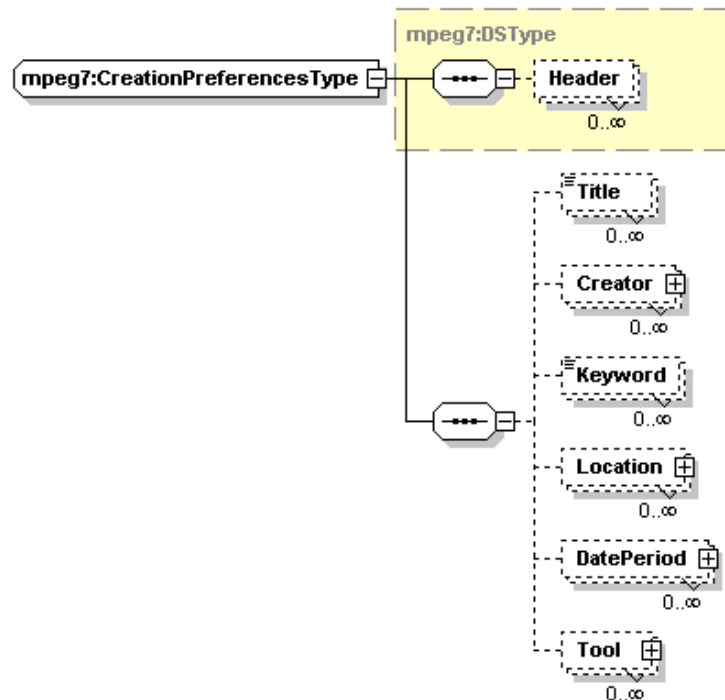
Filtering and Search Preferences

Τα Filtering and Search Preferences (FASP) χρησιμοποιούνται για να ορίσουν τις προτιμήσεις του χρήστη για την επιλογή και αναζήτηση περιεχομένου πολυμέσων με βάση τη δημιουργία (Creation), κατηγοριοποίηση (Classification) και προέλευση (Source) του περιεχομένου. Οργανώνονται ιεραρχικά από το γενικότερο, στην κορυφή της ιεραρχίας, στο ειδικότερο, στη βάση της ιεραρχίας. Σχηματικά:



Παρακάτω θα εξεταστούν ένα προς ένα τα στοιχεία αυτά και τα πεδία τους.

Creation Preferences

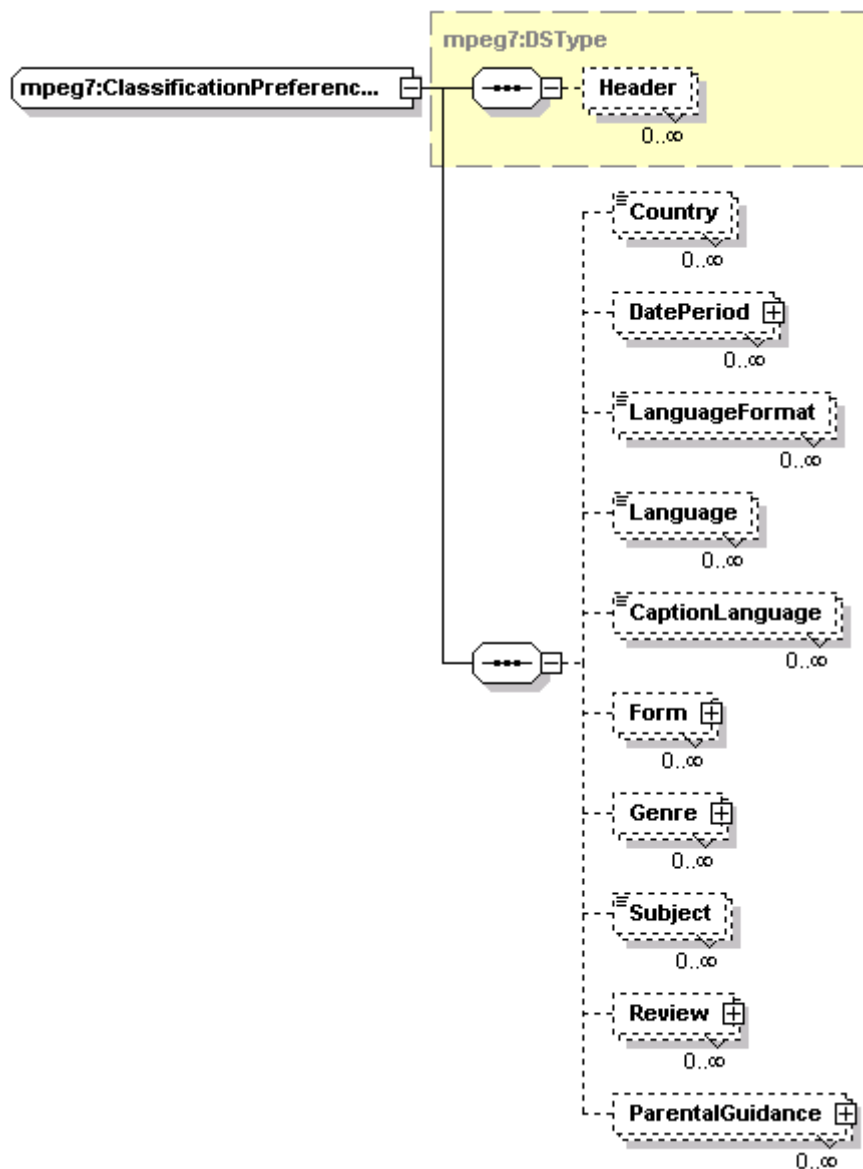


Περιέχει τις προτιμήσεις του χρήστη σε σχέση με τη δημιουργία του περιεχομένου. Περιλαμβάνει τα στοιχεία:

- Title (Περιγράφει τις προτιμήσεις του χρήστη για τον τίτλο της οπτικοακουστικής (Α/Υ) πληροφορίας)
- Creator (Περιγράφει τις προτιμήσεις του χρήστη για τους δημιουργούς και τους συνεργάτες (π.χ. ηθοποιοί, σκηνοθέτες, κ.λπ.) του προγράμματος)
- Keyword (Περιγράφει τις προτιμήσεις του χρήστη για λέξεις ή φράσεις που περιγράφουν επιγραμματικά το πρόγραμμα)
- Location (Περιγράφει τις προτιμήσεις του χρήστη για την τοποθεσία που δημιουργήθηκε το πρόγραμμα)

- Date period (Περιγράφει τις προτιμήσεις του χρήστη για την χρονική περίοδο που δημιουργήθηκε το πρόγραμμα)

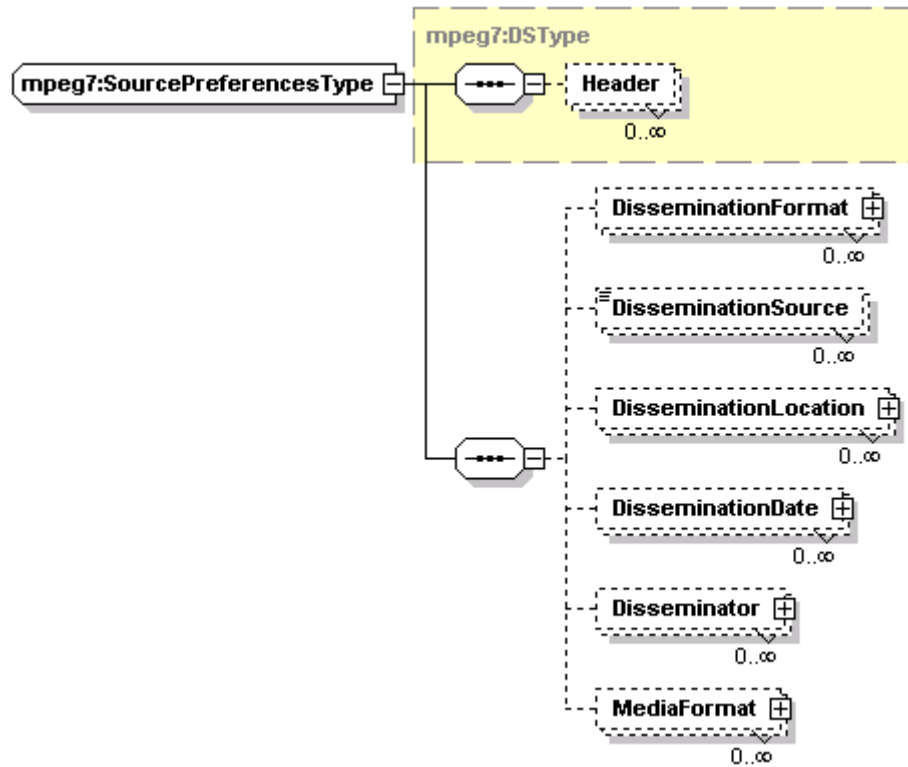
Classification Preferences



Περιέχει τις προτιμήσεις του χρήστη σε σχέση με τη ταξινόμηση του A/V περιεχομένου.
Περιλαμβάνει:

- Country (Περιγράφει τις προτιμήσεις του χρήστη για την χώρα κατασκευής του προγράμματος)
- Date period (Περιγράφει τις προτιμήσεις του χρήστη για την χρονική περίοδο της κυκλοφορίας του προγράμματος)
- Language Format (Περιγράφει τις προτιμήσεις του χρήστη για την μορφή της γλώσσας του προγράμματος)
- Language (Περιγράφει τις προτιμήσεις του χρήστη για την γλώσσα του προγράμματος)
- Caption Language (Περιγράφει τις προτιμήσεις του χρήστη για την γλώσσα υπότιτλων του προγράμματος)
- Genre (Περιγράφει τις προτιμήσεις του χρήστη για την κατηγορία του προγράμματος)
- Subject (Περιγράφει τις προτιμήσεις του χρήστη για το θέμα των πολυμέσων του προγράμματος)
- Media review (Περιγράφει τις προτιμήσεις του χρήστη για τις κριτικές του προγράμματος)
- Parental guidance (Περιγράφει τις προτιμήσεις του χρήστη για την γονική καθοδήγηση, αναφέρεται π.χ. σε μη προβολή ακατάλληλου περιεχομένου στα παιδιά)

Source Preferences



Περιέχει τις προτιμήσεις του χρήστη σε σχέση με τη πηγή του Α/V περιεχομένου.
Περιλαμβάνει:

- Dissemination Format (Περιγράφει τις προτιμήσεις του χρήστη για την μορφή διάχυσης του προγράμματος π.χ. επίγειες αναμεταδόσεις, δικτυακές αναμεταδόσεις, CD-ROM, κ.λπ.)
- Dissemination Source (Περιγράφει τις προτιμήσεις του χρήστη για την πηγή διάχυσης του προγράμματος, π.χ. ένα κανάλι αναμετάδοσης ή ένας διακομιστής)
- Dissemination Location (Περιγράφει τις προτιμήσεις του χρήστη για την τοποθεσία που διανέμεται το πρόγραμμα)

- Dissemination Date (Περιγράφει τις προτιμήσεις του χρήστη για την χρονική περίοδο είναι διαθέσιμο το πρόγραμμα, π.χ. ο χρόνος αναμετάδοσης του προγράμματος)
- Disseminator (Περιγράφει τις προτιμήσεις του χρήστη για την το πρόσωπο, την ομάδα ή τον οργανισμό που διανέμει το πρόγραμμα)
- Media Format (Περιγράφει τις προτιμήσεις του χρήστη για την μορφή που διανέμεται το πρόγραμμα, π.χ. μορφή βίντεο, μορφή ήχου, αναλογία διαστάσεων (aspect ratio) κ.λπ.)

Τόσο τα βασικά (Creation, Classification, κ.λπ) όσο και τα στοιχειώδη (Title, Creator, Country, κ.λπ) στοιχεία συσχετίζονται με ένα *preference value* $[-100,100]$ που αντικατοπτρίζει, όπως γίνεται φανερό, το βαθμό αρεσκείας του χρήστη προς αυτό σε σχέση πάντα με τα υπόλοιπα όμοια στοιχεία.

Preference Condition



Αναφέρετε στο πότε και που ο χρήστης θέλει να ενεργοποιείται το συγκεκριμένο φίλτρο, π.χ. τον ενδιαφέρει τον φίλτρο να ενεργοποιείται κάθε μέρα από τις 8 το πρωί μέχρι τις 12 το μεσημέρι στο γραφείο του.

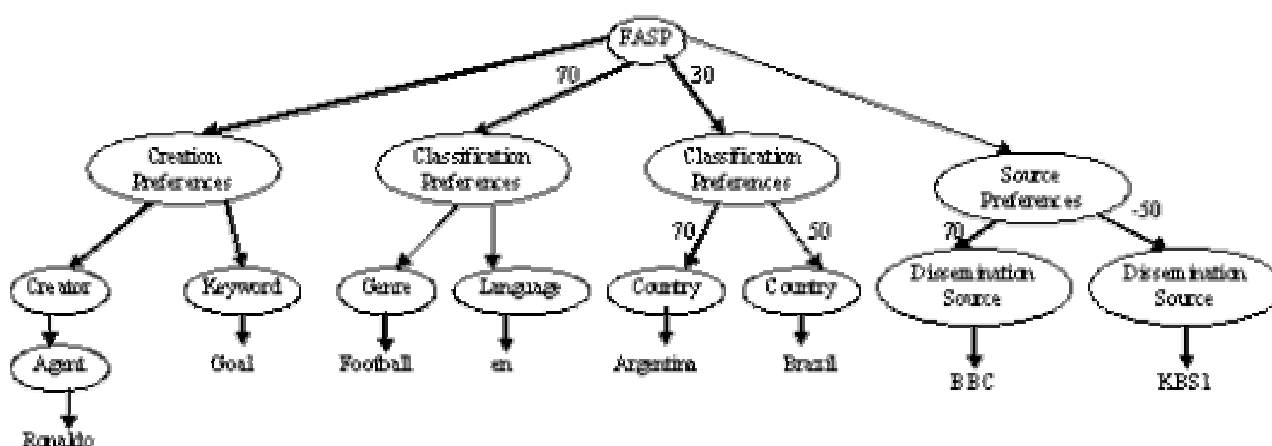
Παράδειγμα Filtering And Search Preferences

Παρακάτω φαίνεται ένα παράδειγμα xml εγγράφου που περιγράφει ένα δέντρο FASP:

```
<FilteringAndSearchPreferences>
  <CreationPreferences>
    <Creator>
      <Agent xsi:type="PersonType">
        <Name><GivenName>
```

```
Ronaldo
  </GivenName></Name>
</Agent>
</Creator>
<Keyword>
  Goal
</Keyword>
</CreationPreferences>
<ClassificationPreferences preferenceValue="70">
  <Genre href="urn:mpeg:GenreCS">
    <Name>Football</Name>
  </Genre>
  <Language>
    en
  </Language>
</ClassificationPreferences>
<ClassificationPreferences preferenceValue="30">
  <Country preferenceValue="70">
    Argentina
  </Country>
  <Country preferenceValue="50">
    Brazil
  </Country>
</ClassificationPreferences>
<SourcePreferences>
  <DisseminationSource preferenceValue="70">
    BBC
  </DisseminationSource>
  <DisseminationSource preferenceValue="-50">
    KBS1
  </DisseminationSource>
</SourcePreferences>
</FilteringAndSearchPreferences>
```

Χάριν απλότητας στην αναπαράσταση χρησιμοποιούμε την παρακάτω γραφική μορφή. Το FASP απεικονίζεται σαν ένα δέντρο όπου στις ακμές του αναφέρετε ο βαθμός προτίμησης του χρήστη. Για το παράδειγμα το δέντρο είναι:



Σχήμα 6: Δενδρικό παράδειγμα αναπαράστασης ενός προφίλ χρήστη.

Αντιστοιχίες μεταξύ χαρακτηριστικών του Filtering and Search

Preferences και των μεταδεδομένων των προγραμμάτων

Κάθε πρόγραμμα έχει μία πληθώρα από χαρακτηριστικά που το περιγράφουν όπως μελετήθηκε παραπάνω. Από την άλλη ένα FASP έχει και αυτό πολλά χαρακτηριστικά. Δεν είναι τόσο σαφές μερικές φορές πια χαρακτηριστικά των προγραμμάτων αντιστοιχούν στα πεδία του FASP ώστε να μπορεί να γίνει η σύγκριση και να βρεθεί αν ένα πρόγραμμα πληροί τις προϋποθέσεις που θέτονται από το προφίλ το οποίο είναι δομημένο σαν FASP. Παρακάτω γίνεται μία αντιστοίχιση μεταξύ των μεταδεδομένων των προγραμμάτων με αυτών του προφίλ.

Οι αντιστοιχίες των πεδίων του Creation Preferences με τα μεταδεδομένα των προγραμμάτων φαίνονται στον παρακάτω πίνακα:

<i>Element/attribute Name</i>	<i>Mapping</i>
Title	ProgramInformationTable/ProgramInformation/BasicDescription/Title ProgramInformationTable/ProgramInformation/BasicDescription/ShortTitle SegmentInformationTable/SegmentInformation/Description/Title
Creator	ProgramInformationTable/ProgramInformation/BasicDescription/CreditsList/CreditsItem

<i>Element/attribute Name</i>	<i>Mapping</i>
Keyword	ProgramInformationTable/ProgramInformation/BasicDescription/Title ProgramInformationTable/ProgramInformation/BasicDescription/ShortTitle ProgramInformationTable/ProgramInformation/BasicDescription/Keyword ProgramInformationTable/ProgramInformation/BasicDescription/Synopsis SegmentInformationTable/SegmentInformation/Description/Title SegmentInformationTable/SegmentInformation/Description/Keyword SegmentInformationTable/SegmentInformation/Description/Synopsis
Location	ProgramInformationTable/ProgramInformation/BasicDescription/CreationCoordinates/CreationLocation
DatePeriod	ProgramInformationTable/ProgramInformation/BasicDescription/CreationCoordinates/CreationDate
Tool	<i>Δεν χρησιμοποιείται.</i>

Πίνακας 15: Αντιστοιχίσεις μεταξύ Creation Preferences και μεταδεδομένων προγραμμάτων.

Οι αντιστοιχίσεις των πεδίων του Classification Preferences με τα μεταδεδομένα των προγραμμάτων φαίνονται στον παρακάτω πίνακα:

<i>Element/attribute Name</i>	<i>Mapping</i>
Country	ProgramInformationTable/ProgramInformation/BasicDescription/ReleaseLocation
DatePeriod	ProgramInformationTable/ProgramInformation/BasicDescription/ReleaseDate
LanguageFormat	ProgramInformationTable/ProgramInformation/BasicDescription/Language ProgramInformationTable/ProgramInformation/BasicDescription/CaptionLanguage ProgramInformationTable/ProgramInformation/BasicDescription/SignLanguage
Language	ProgramInformationTable/ProgramInformation/BasicDescription/Language
CaptionLanguage	ProgramInformationTable/ProgramInformation/BasicDescription/CaptionLanguage
Form	<i>This element is not used in this specification.</i>
Genre	ProgramInformationTable/ProgramInformation/BasicDescription/Genre

<i>Element/ attribute Name</i>	<i>Mapping</i>
Subject	ProgramInformationTable/ProgramInformation/BasicDescription/Title ProgramInformationTable/ProgramInformation/BasicDescription/ShortTitle ProgramInformationTable/ProgramInformation/BasicDescription/Keyword ProgramInformationTable/ProgramInformation/BasicDescription/Synopsis SegmentInformationTable/SegmentInformation/Description/Title SegmentInformationTable/SegmentInformation/Description/Keyword SegmentInformationTable/SegmentInformation/Description/Synopsis
Review	ProgramReviewTable/ProgramReviews/Review
ParentalGuidance	ProgramInformationTable/ProgramInformation/BasicDescription/ParentalGuidance

Πίνακας 16: Αντιστοιχίσεις μεταξύ Classification Preferences και μεταδεδομένων προγραμμάτων.

Οι αντιστοιχίσεις των πεδίων του Source Preferences με τα μεταδεδομένα των προγραμμάτων φαίνονται στον παρακάτω πίνακα:

<i>Element/ attribute Name</i>	<i>Mapping</i>
DisseminationFormat	Δεν χρησιμοποιείται.
DisseminationSource	ServiceInformationTable/ServiceInformation/Name
DisseminationLocation	Δεν χρησιμοποιείται.
DisseminationDate	ProgramLocationTable/BroadcastEvent/PublishedTime & PublishedDuration ProgramLocationTable/Schedule/ScheduleEvent/PublishedTime & PublishedDuration ProgramLocationTable/OnDemandProgram/StartOfAvailability & EndOfAvailability ProgramLocationTable/OnDemandService/OnDemandProgram/StartOfAvailability & EndOfAvailability
Disseminator	ServiceInformationTable/ServiceInformation/Owner

<i>Element/ attribute Name</i>	<i>Mapping</i>
MediaFormat	ProgramInformationTable/ProgramInformation/AVAttributes ProgramLocationTable/BroadcastEvent/InstanceDescription/AVAttributes ProgramLocationTable/Schedule/ScheduleEvent/InstanceDescription/AVAttributes ProgramLocationTable/OnDemandProgram/InstanceDescription/AVAttributes ProgramLocationTable/OnDemandService/OnDemandProgram/InstanceDescription/AVAttributes
noRepeat	ProgramLocationTable/BroadcastEvent/Repeat ProgramLocationTable/Schedule/ScheduleEvent/Repeat
noEncryption	<i>Δεν χρησιμοποιείται.</i>
noPayPerUse	ProgramLocationTable/BroadcastEvent/Free ProgramLocationTable/Schedule/ScheduleEvent/Free

Πίνακας 17: Αντιστοιχίσεις μεταξύ Source Preferences και μεταδεδομένων προγραμμάτων

Περίληψη

Σε αυτό το κεφάλαιο μελετήθηκε πλήρως, το προσφερόμενο σαν εργαλείο, σχήμα μεταδεδομένων του TV-Anytime ώστε κατά την επερχόμενη ανάλυση να γίνουν κατανοητές όλες οι ενέργειες που εφαρμόστηκαν για την υλοποίηση του συστήματος διήθησης περιεχομένου προγραμμάτων σύμφωνα με τα ενδιαφέροντα των χρηστών. Συγκεκριμένα μελετήθηκε ενδελεχώς η δομή ενός προφίλ χρήστη και τί ακριβώς χαρακτηριστικά περιεχομένου είναι σε θέση περιγράψει αυτό το προφίλ. Παρατηρείται, δε, στις περισσότερες περιπτώσεις ένα πεδίο του FilteringAndSearchPreferences να αντιστοιχίζεται με περισσότερα του ενός μεταδεδομένα προγραμμάτων πράγμα που λήφθηκε υπ' όψιν κατά τον σχεδιασμό αλλά και την υλοποίηση των μηχανισμών ευφυούς διήθησης περιεχομένου που θα περιγραφούν στο επόμενο κεφάλαιο.

*Αυτοί που χρυσάφι αναζητούν,
γη πολλή σκάβουν και λίγο
βρίσκουν.*

Ηράκλειτος

Κεφάλαιο IV

ΜΗΧΑΝΙΣΜΟΙ ΕΥΦΥΟΥΣ ΔΙΗΘΗΣΗΣ ΠΕΡΙΕΧΟΜΕΝΟΥ

Εισαγωγή

Το UP-TV βασίζεται στις προδιαγραφές του TV-Anytime όσον αφορά τα μεταδεδομένα τόσο περιεχομένου όσο και των προτιμήσεων των χρηστών. Στο προηγούμενο κεφάλαιο περιγράφηκαν τα μεταδεδομένα τόσο του περιεχομένου όσο και των προτιμήσεων των χρηστών, ώστε τώρα να είναι δυνατόν να περιγραφούν με λεπτομέρεια οι μηχανισμοί του ταιριάσματος των μεταδεδομένων περιεχομένου με τις προτιμήσεις των χρηστών καθώς επίσης και τεχνικές συνεργατικής διήθησης. Οι μηχανισμοί διήθησης περιεχομένου (information filtering) που αναπτύχθηκαν έχουν τα ακόλουθα χαρακτηριστικά:

- Τα τηλεοπτικά προγράμματα θεωρούνται Αντικείμενα Πληροφορίας που περιγράφονται από μια σειρά από διαφορετικούς τύπους χαρακτηριστικών. Κάθε τύπος χαρακτηριστικού αντιστοιχεί σε ένα συγκεκριμένο μέρος της περιγραφής των μεταδεδομένων προγραμμάτων (π.χ. το Genre του προγράμματος είναι ένας τύπος χαρακτηριστικού, το Creator είναι ένα άλλο κ.λπ.).
- Αιτήσεις ανάκτησης όπως και προτιμήσεις χρηστών που σχετίζονται με προφίλ χρηστών ή είναι προκαθορισμένες (stereotypes) όλα μοντελοποιούνται χρησιμοποιώντας το FilteringAndSearchPreferences περιγραφικό σχήμα του MPEG-7, το οποίο επίσης χρησιμοποιείται από τις προδιαγραφές του TV-Anytime. Τέτοια φίλτρα συνθέτονται από όλους τους πιθανούς τύπους χαρακτηριστικών. Ο

προσεκτικός αναγνώστης θα θυμάται ότι οι τύποι χαρακτηριστικών των μεταδεδομένων περιεχομένου και των προτιμήσεων των χρηστών δεν συμπίπτουν πάντα ακριβώς. Το αποτέλεσμα της διαδικασίας του ταιριάσματος είναι ένα σύνολο από προγράμματα που ικανοποιούν τις προτιμήσεις διήθησης και αναζήτησης του χρήστη. Επίσης, διατηρείται πληροφορία με τις προτιμήσεις του χρήστη που ικανοποιούνται για κάθε πρόγραμμα που επέστρεψε στο αποτέλεσμα, ώστε να μπορούν να χρησιμοποιηθούν για να κατανοεί ο χρήστης γιατί κάποιο συγκεκριμένο πρόγραμμα βρίσκεται στα αποτελέσματα και να αποκτάει εμπιστοσύνη στις προβλέψεις του συστήματος.

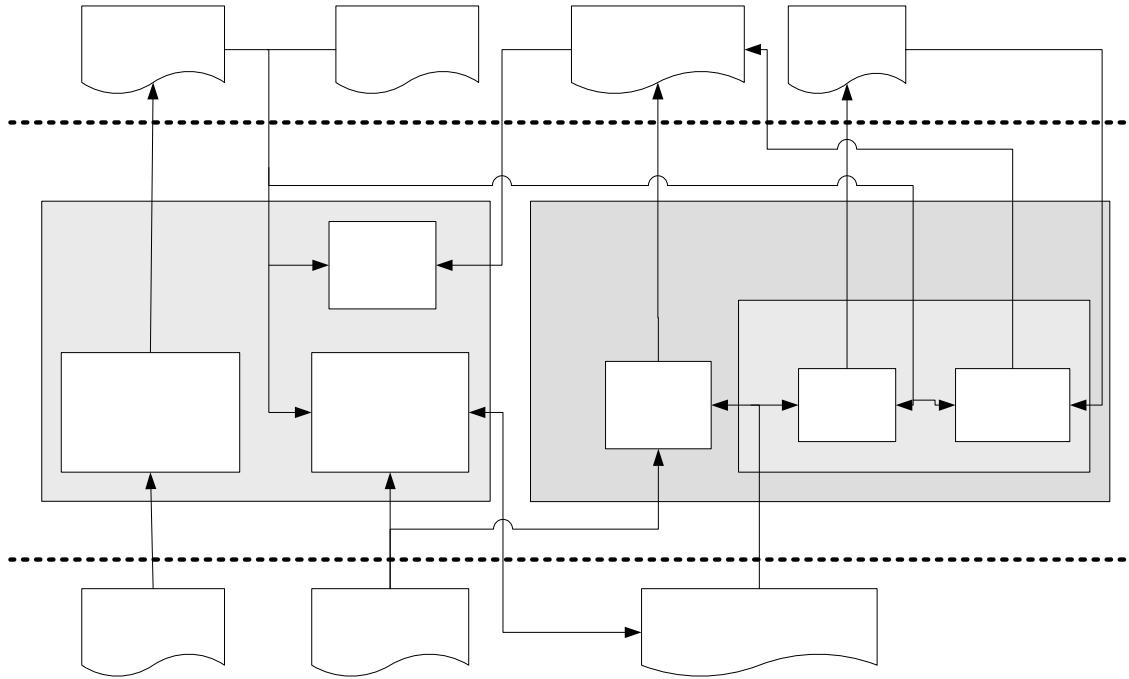
- Οι αιτήσεις συνεργατικής διήθησης (Collaborative Filtering) βασίζονται σε εκτιμήσεις ομοιότητας μεταξύ χρηστών. Αυτή η ομοιότητα υπολογίζεται από το ιστορικό χρήσης και τις προτιμήσεις διήθησης και αναζήτησης κάθε ζευγαριού χρηστών. Ειδικότερα, δύο χρήστες θεωρούνται όμοιοι αν τα προγράμματα στα τα οποία αρέσκονται (ή δεν αρέσκονται) έχουν μία κοινή τομή και/ή οι “προτιμήσεις διήθησης και αναζήτησης” τους μοιάζουν. Για έναν συγκεκριμένο χρήστη, το σύστημα είναι σε θέση να προτείνει προγράμματα που όμοιοι με αυτόν χρήστες είδαν και τους άρεσαν.
- Συνδυασμένες αιτήσεις ανάκτησης χρησιμοποιούν και αναζήτηση βασισμένη στο περιεχόμενο αλλά και συνεργατική διήθηση για να ανακτήσουν προγράμματα και εφαρμόζουν ένα σχήμα στατιστικού μέσου για να υπολογίσουν ένα τελικό βάρος προτίμησης για κάθε πρόγραμμα.

Στην συνέχεια του κεφαλαίου θα παρουσιαστεί η αρχιτεκτονική, η μεθοδολογία καθώς και οι λεπτομέρειες της υλοποίησης των μηχανισμών ευφυούς διήθησης περιεχομένου.

Αρχιτεκτονική

Οι αυτοτελές μονάδες λογισμικού που είναι υπεύθυνες για την ευφυή διήθηση περιεχομένου είναι δύο: Ο Πράκτορας Προτάσεων – Διήθησης (Recommender – Filtering Agent (R-FA)) και η Μονάδα Προσαρμογής Προτιμήσεων (Preference Adaptation Module (PAM)). Για να

εξηγηθεί σε βάθος η λειτουργικότητα, αυτές οι μονάδες έχουν επιπλέον αναλυθεί όπως απεικονίζει το παρακάτω σχήμα.



Σχήμα 7: Η Αρχιτεκτονική του Μηχανισμού Ευφυούς Διήθησης Περιεχομένου

Τα συνιστώντα μέρη της αρχιτεκτονικής διαιρούνται σε τρία κύρια τμήματα:

TV-Anytime XML έγγραφα που περιέχουν την απαιτούμενη πληροφορία. Αυτή η πληροφορία αποτυπώνεται στην βάση δεδομένων του UP-TV. Υπάρχουν τρία κύρια είδη έγκυρων TV-Anytime XML εγγράφων: Περιγραφές περιεχομένου (content descriptions) (που αναφέρεται σε πληροφορία τόσο των προγραμμάτων όσο και τμημάτων αυτών), ιστορικά χρήσης (usage histories) και “filtering and search preferences” των χρηστών (το summary and browsing preferences δεν μελετάται εδώ).

Εκτιμήσεις προτιμήσεων διαφόρων ειδών. Κάποιες υπολογίζονται από το σύστημα και κάποιες άλλες δίνονται κατευθείαν από τους χρήστες.

Τμήματα λογισμικού που υλοποιούν τους μηχανισμούς ευφυούς διήθησης. Αυτά τα τμήματα αποτελούνται από δύο κύρια μέρη: (1) Το R-FA που είναι υπεύθυνο να προβλέπει τις προτιμήσεις του χρήστη για μελλοντικό περιεχόμενο με το ταίριασμα των “filtering and search preferences” με περιγραφές περιεχομένου ή με την ανεύρεση παρόμοιων χρηστών ως προς κάποιον συγκεκριμένο και την πρόταση περιεχομένου που βρέθηκε να ενδιαφέρει τους πρώτους. (2) Το PAM που είναι υπεύθυνο να διαχειρίζεται ιστορικά χρηστών για να παρέχει έμμεσες προτιμήσεις περιεχομένου και να χρησιμοποιεί προτιμήσεις περιεχομένου (άμεσες ή έμμεσες) για να διαμορφώνει τις “filtering and search preferences” των χρηστών. Η ανάλυση και υλοποίηση του PAM δεν αποτελούν αντικείμενο της παρούσας διατριβής· ο αναγνώστης παραπέμπεται στο [10].

Η υλοποίηση των μηχανισμών διήθησης περιεχομένου βασίζεται σε ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων ακολουθώντας μία προσέγγιση παρόμοια με το [6] αλλά πιο γενική και περίπλοκη. Το σύστημα UP-TV χρησιμοποιεί μια κανονικοποιημένη βάση δεδομένων για να αποθηκεύσει:

- Περιγραφές περιεχομένου (μεταδεδομένα τόσο του περιεχομένου όσο και των τμημάτων καθώς και των στιγμιότυπων των προγραμμάτων)
- Προτιμήσεις Διήθησης και Αναζήτησης (Filtering and Search Preferences) που σχετίζονται με αιτήσεις ανάκτησης, προφίλ χρηστών αλλά και προκαθορισμένες τιμές (stereotypes).
- Εκτιμήσεις προτιμήσεων κάθε είδους (δηλ. ανεκδήλωτες αλλά και ρητές προτιμήσεις περιεχομένου, μελλοντικές εκτιμήσεις προτιμήσεων και εκτιμήσεις ομοιότητας χρηστών)

Το ταίριασμα μεταξύ των προτιμήσεων διήθησης και αναζήτησης και των μεταδεδομένων των προγραμμάτων βασίζεται στο εκτεταμένο δυαδικό μοντέλο (extended Boolean model) και υλοποιείται σαν ένα σύνολο από ερωτήσεις SQL που ενώνουν τους πίνακες που διατηρούν την

πληροφορία για τα μεταδεδομένα των προγραμμάτων με αυτά που την διατηρούν για τις προτιμήσεις διήθησης και αναζήτησης.

Διατηρείται πληροφορία με τις προτιμήσεις του χρήστη που ικανοποιούνται για κάθε πρόγραμμα που βρίσκεται στις προβλέψεις.

Οι τεχνικές της συνεργατικής διήθησης αποτελούνται από δύο διακριτά μέρη:

1) *Εύρεση χρηστών όμοιων με κάποιον συγκεκριμένο.* Αυτό το μέρος υλοποιεί μία διαδικασία ταιριάσματος μεταξύ των χρηστών βασισμένη είτε στις προτιμήσεις περιεχομένου τους είτε στις προτιμήσεις διήθησης και αναζήτησής τους. Η υλοποίηση αυτού του ταιριάσματος βασίζεται σε συγκεκριμένες ερωτήσεις SQL που εφαρμόζουν ένα self join είτε μεταξύ προτιμήσεων διήθησης και αναζήτησης είτε μεταξύ προτιμήσεων περιεχομένου.

2) *Συνεργατικές προτάσεις.* Αυτό το μέρος υλοποιεί αναζήτηση σε προγράμματα ρητά ή ανειδήλωτα βαθμολογημένα από χρήστες όμοιους προς τον χρήστη U και που ποτέ δεν τα έχει παρακολουθήσει ο U, με στόχο να του τα προτείνουν. Η εφαρμογή των παραπάνω γίνεται με ερωτήσεις SQL που αναζητούν προγράμματα που βρέθηκαν να ενδιαφέρουν τους όμοιους με τον U χρήστες και που δεν τα έχει παρακολουθήσει ο U.

Από τα παραπάνω προκύπτει ότι υποστηρίζονται τρεις διαφορετικοί τύποι προτιμήσεων περιεχομένου: ρητές (δοσμένες από τους χρήστες), ανειδήλωτες (υπολογισμένες από τα ιστορικά χρήσης, βλέπε [10]) και από πρόβλεψη (αναφέρονται σε μελλοντικό περιεχόμενο και υπολογίζονται με βασισμένες στο περιεχόμενο τεχνικές συνεργατικής διήθησης).

Η Βασισμένη στο Περιεχόμενο Μονάδα Διήθησης (Content Based Filtering Module, CBFM)

Το πλαίσιο ασαφούς λογικής για τις εκτιμήσεις προτιμήσεων

Όλες οι εκτιμήσεις προτιμήσεων που υπολογίζονται από τους ευφυείς μηχανισμούς διήθησης περιεχομένου του UP-TV μπορούν να θεωρηθούν σαν αιτήσεις ανάκτησης πληροφοριών βασισμένες σε ομοιότητα. Όλες αυτές οι διαφορετικών ειδών εκτιμήσεις προτιμήσεων μπορούν να μοντελοποιηθούν χρησιμοποιώντας το ίδιο γενικό μαθηματικό πλαίσιο βασισμένο στην θεωρία ανάκτησης πληροφοριών. Αυτό το πλαίσιο περιγράφεται εδώ.

Γενικά, κάθε αίτηση για ανάκτηση αντικειμένων που ανήκουν σε ένα σύμπαν I που περιγράφεται σε ένα χώρο χαρακτηριστικών F , αντιστοιχεί σε μία ερώτηση q που συγκροτείται από ένα (πιθανώς) δομημένο σύνολο χαρακτηριστικών F' , τα οποία, με την σειρά τους, είναι υποσύνολο του F . Αυτό το γενικό σχήμα μπορεί να χρησιμοποιηθεί για να περιγράψει τα παρακάτω είδη λειτουργικότητας (βλέπε την στήλη “Ερμηνεία”):

Σύμπαν I	Χώρος Χαρακτηριστικών F	Ερωτήσεις Q	Ερμηνεία
Προγράμματα / τμήματα	Χαρακτηριστικά περιγραφής περιεχομένου (όπως genre, creators, channels etc).	«Filtering and Search Preferences» των χρηστών.	Εκτιμήσεις προτιμήσεων για μελλοντικό περιεχόμενο (διήθηση βασισμένη στο περιεχόμενο)
Προγράμματα / τμήματα	Τύποι ενεργειών χρηστών.	Ενέργειες που πραγματοποιούνται από τους χρήστες.	Υποκρυπτόμενες εκτιμήσεις περιεχομένου
«Filtering and Search Preferences» των χρηστών	Χαρακτηριστικά περιγραφής περιεχομένου (όπως genre, creators, channels etc).	«Filtering and Search Preferences» των χρηστών	Εκτιμήσεις ομοιότητας χρηστών

Σύμπαν I	Χώρος Χαρακτηριστικών F	Ερωτήσεις Q	Ερμηνεία
Προτιμήσεις περιεχομένου (υποκρυπτόμενες ή κατηγορηματικές)	Προγράμματα / τμήματα	Προτιμήσεις περιεχομένου (υποκρυπτόμενες ή κατηγορηματικές)	Εκτιμήσεις ομοιότητας χρηστών
Προτιμήσεις περιεχομένου (υποκρυπτόμενες ή κατηγορηματικές)	Εκτιμήσεις ομοιότητας χρηστών	Προτιμήσεις περιεχομένου (υποκρυπτόμενες ή κατηγορηματικές)	Εκτιμήσεις προτιμήσεων για μελλοντικό περιεχόμενο (συνεργατική (collaborative) διήθηση)

Πίνακας 18: Διαφορετικά είδη λειτουργικότητας εκφραζόμενα από ένα γενικό σύστημα ανάκτησης πληροφορίας.

Περιγραφή του Μοντέλου Ασαφούς Λογικής

Τα κλασικά συστήματα ανάκτησης πληροφοριών (information retrieval systems) μοντελοποιούν τόσο αντικείμενα πληροφορίας όσο και τις ερωτήσεις ως πίνακες αποτελούμενους από στοιχεία (βλέπε [4] και [5] για λεπτομέρειες). Το σύνολο των αντικειμένων της πληροφορίας που ικανοποιούν μία ερώτηση περιλαμβάνει εκείνα τα αντικείμενα που περιέχουν οποιοδήποτε στοιχείο περιέχετε στον πίνακα της ερώτησης.

Αντίθετα, στην παραδοσιακή Δυαδική λογική (Boolean logic), μία ερώτηση αναπαριστάται ως μία δυαδική έκφραση που αποτελείτε από αναπόσπαστα στοιχεία και τους τρεις δυαδικούς τελεστές (AND, OR, NOT) ως συνδέσμους. Το σύνολο των αντικειμένων πληροφορίας που συνιστούν την απάντηση σε μία ερώτηση ικανοποιούν την δυαδική έκφραση. Το ελάττωμα αυτής της προσέγγισης είναι ότι ένα αντικείμενο πληροφορίας θεωρείτε είτε σχετικό (relevant) είτε μη-σχετικό (non-relevant) ως προς μία δεδομένη ερώτηση. Δεν υφίσταται, ως εκ τούτου, μέθοδος κατάταξης των αντικειμένων αναφορικά προς τον βαθμό που ταιριάζουν στην ερώτηση.

Για να ξεπεραστεί το παραπάνω μειονέκτημα, το Εκτεταμένο Δυαδικό Μοντέλο (Extended Boolean Model) εισήχθη στην βιβλιογραφία της ανάκτησης πληροφορίας (information retrieval). Αυτό το μοντέλο είναι γενίκευση της δυαδικής λογικής και στηρίζεται στη θεωρία των ασαφούς συνόλων (fuzzy set theory). Μας παρέχει με τύπους για τον υπολογισμό

περίπλοκων δυαδικών εκφράσεων ούτως ώστε στα αντικείμενα πληροφορίας να δίδετε μια βαθμολογία στο διάστημα $[0,1]$ αντί μίας απλής δυαδικής απάντησης (true/false). Ποικίλες μελέτες (όπως [1] και [2]) αποδεικνύουν την ανωτερότητα της απόδοσής του σε σύγκριση με τις παραδοσιακές μεθόδους ανάκτησης πληροφοριών. Παρακάτω περιγράφεται ένας ορισμός του μοντέλου, έτσι όπως παρουσιάστηκε στην [1], κατάλληλα διασκευασμένο για τις ανάγκες της εργασίας:

Ορισμός: Weighted Fuzzy Information Retrieval System (WFIRS):

Ένα WFIRS C ορίζεται σαν μία τετράδα $C = \langle F, I, Q, E \rangle$ όπου:

F είναι ένα σύνολο χαρακτηριστικών (features) που χρησιμοποιούνται για να περιγράψουν αντικείμενα πληροφορίας και να σχηματίζουν ερωτήσεις.

I είναι ένα σύνολο αντικειμένων πληροφορίας (information items). Κάθε αντικείμενο πληροφορίας αντιστοιχεί σε μία συνάρτηση από τα F στο διάστημα $[0,1]$. Αυτό ισοδυναμεί με την έκφραση $I \subseteq [0,1]^F$.

Q είναι ένα σύνολο ερωτήσεων (queries) που ταυτοποιούνται από το σύστημα. Ανταποκρίνονται σε δυαδικές εκφράσεις που χρησιμοποιούν χαρακτηριστικά (F) σαν άτομα και τους τελεστές AND, OR, NOT ως συνδέσμους. Επιπρόσθετα κάθε υπό-ερώτηση συσχετίζεται με ένα σχετικό βάρος από το διάστημα $[0,1]$ που αντιπροσωπεύει τη σχετική “βαρύτητα” του αναφορικά προς τις υπόλοιπα υπό-ερωτήσεις. Παραδείγματα έγκυρων ερωτήσεων είναι: $\langle q_1, w_1 \rangle AND \langle q_2, w_2 \rangle$ και $\langle q_1, w_1 \rangle OR \langle q_2, w_2 \rangle$.

E είναι μία συνάρτηση αξιολόγησης (evaluation) $E : Q \times I \rightarrow [0,1]$ που δίνει μία τιμή στο $[0,1]$ σε κάθε έγκυρη ερώτηση από το Q όσον αναφορά κάθε αντικείμενο πληροφορίας. Έτσι, $E(q, i)$ είναι η κατάταξη ομοιότητας του αντικειμένου πληροφορίας i όσον αναφορά την ερώτηση q . Ε ορίζεται έτσι αναδρομικά στηριγμένο σε συναρτήσεις αξιολόγησης των λογικών (logical) τελεστών:

$$E(\langle q_1, w_1 \rangle AND \langle q_2, w_2 \rangle, i) = f_{AND}(\langle E(q_1, i), w_1 \rangle, \langle E(q_2, i), w_2 \rangle)$$

$$E(\langle q_1, w_1 \rangle OR \langle q_2, w_2 \rangle, i) = f_{OR}(\langle E(q_1, i), w_1 \rangle, \langle E(q_2, i), w_2 \rangle)$$

$$E(NOT q_1, i) = f_{NOT}(E(q_1, i))$$

$$E(f, i) = i(f) \text{ (παρατηρήστε ότι } i \text{ είναι μία συνάρτηση από το } F \text{ στο } [0,1]).$$

Για μία ερώτηση $q \in Q$, το υποσύνολο $ANS(q)$ του I που περιέχει όλα τα στοιχεία του I για τα οποία η συνάρτηση αξιολόγησης F έχει θετική (>0) τιμή, είναι απάντηση στο q . Αυτό το υποσύνολο ορίζεται ως:

$$ANS(q) = \{ \langle i, E(q, i) \rangle \in I \times (0,1] \mid E(q, i) > 0 \}.$$

Παρατηρήστε ότι στην πράξη ενδέχεται να πάρουμε πάρα πολλές πιθανές απαντήσεις, τις περισσότερες με πολύ μικρές τιμές. Σ' αυτήν την περίπτωση μπορούμε να θεωρήσουμε μόνο τις τιμές με $E(q, I) > E_{min}$, για κάποιο κατάλληλο για την εφαρμογή E_{min} , ή να πάρουμε τα N πρώτα αντικείμενα με τις υψηλότερες καταχωρήσεις, πάλι για κάποιο κατάλληλο N .

Για να αξιολογήσει κανείς τις ερωτήσεις που αναγνωρίζονται από ένα WFIRS, πρέπει να του δοθούν οι συναρτήσεις αξιολόγησης f_{NOT} , f_{AND} , και f_{OR} . Υπάρχει μία πληθώρα ορισμών τέτοιων συναρτήσεων. Ο παρακάτω πίνακας παρουσιάζει τους ορισμούς που αντιστοιχούν στο p-norm Extended Boolean Model, το οποίο χρησιμοποιείται από το σύστημα. Παρατηρήστε ότι οι συναρτήσεις f_{AND} , και f_{OR} είναι n-αδικές αντί δυαδικές. Αυτό οφείλεται στο γεγονός ότι δεν είναι μεταβατικές.

$f_{AND}((a_1, w_1), \dots, (a_n, w_n))$	$f_{OR}((a_1, w_1), \dots, (a_n, w_n))$	$f_{NOT}(a)$
$1 - \left(\frac{\sum_{i=1}^n (1 - a_i)^p \cdot w_i^p}{\sum_{i=1}^n w_i^p} \right)^{1/p} \quad 1 \leq p \leq \infty$	$\left(\frac{\sum_{i=1}^n a_i^p \cdot w_i^p}{\sum_{i=1}^n w_i^p} \right)^{1/p} \quad 1 \leq p \leq \infty$	$1 - a$

Πίνακας 19. Σταθμητές (Weighted) n-αδικές βασικές συναρτήσεις αξιολόγησης του p-Norm Extended Boolean Model.

Ενσωματώνοντας αρνητικά βάρη ερωτήσεων στο WFIRS

Το TV-Anytime χρησιμοποιεί εκτεταμένα αρνητικές τιμές προτίμησης για να μοντελοποιήσει την δυσαρέσκεια. Για να διαχειριστεί αυτή η κατάσταση στο UP-TV (βλέπε για λεπτομέρειες το [31], [32] όπου αναλύονται διεξοδικώς), ο ορισμός του WFIRS πρέπει να διασκευαστεί ώστε να επιτρέπει αρνητικά βάρη στις ερωτήσεις. Υποθέτουμε ότι ένας όρος με αρνητικό βάρος ερώτησης ισοδυναμεί με τον ίδιο όρο με θετικό βάρος και ένα τελεστή NOT να εφαρμόζεται σ' αυτόν. Συνεπώς, μια ερώτηση της μορφής $q = \langle f_1, 1 \rangle \text{AND} \langle f_2, 0.5 \rangle \text{AND} \langle f_3, -0.3 \rangle \text{AND} \langle f_4, -1 \rangle$ σημαίνει ότι τα αντικείμενα που ανταποκρίνονται στην ερώτηση πρέπει να έχουν τα χαρακτηριστικά f_1, f_2 και να μην έχουν ούτε το f_3 ούτε το f_4 .

Ένας σαφής τρόπος να ενσωματωθούν αυτά τα βάρη στο WFIRS είναι με τον ακόλουθο ορισμό:

Ορισμός: Generalized WFIRS (GWFIRS):

Ένα Γενικευμένο (Generalized) WFIRS \mathbf{C} ορίζετε ως ένα WFIRS $\mathbf{C} = \langle F, I, Q, E \rangle$ του οποίου τα βάρη των ερωτήσεων του συνόλου Q παίρνουν πραγματικές τιμές στο διάστημα $[-1, 1]$.

Επιπρόσθετα, η μέθοδος αξιολόγησης πρέπει να αναθεωρηθεί ούτως ώστε να ενσωματωθούν τα αρνητικά βάρη. Η αναθεώρηση βασίζεται στην παρακάτω αντικατάσταση των a_i 's του πίνακα 19.

$$a_i \leftarrow \frac{|\{w_i\}| - \{w_i\}}{2} + \{w_i\} \cdot a_i$$

$$1 - a_i \leftarrow \frac{|\{w_i\}| + \{w_i\}}{2} - \{w_i\} \cdot a_i$$

όπου $\{.\}$ ορίζετε ως εξής:

$$\{x\} = \begin{cases} -1, & x < 0 \\ 0, & x = 0 \\ +1, & x > 0 \end{cases}$$

Είναι απλό να δούμε ότι οι παραπάνω ορισμοί οδηγούν:

$$\frac{|\{w_i\}| - \{w_i\}}{2} + \{w_i\} \cdot a_i = \begin{cases} 1 - a_i, & w_i < 0 \\ 0, & w_i = 0 \\ a_i, & w_i > 0 \end{cases}$$

$$\frac{|\{w_i\}| + \{w_i\}}{2} - \{w_i\} \cdot a_i = \begin{cases} a_i, & w_i < 0 \\ 0, & w_i = 0 \\ 1 - a_i, & w_i > 0 \end{cases}$$

Βασισμένοι στις παραπάνω ισότητες, οι συναρτήσεις αξιολόγησης του p-Norm μοντέλου έχουν ως ακολούθως:

$F_{AND}((A_D, W_D), \dots, (A_N, W_N))$	$F_{OR}((A_D, W_D), \dots, (A_N, W_N))$	$F_{NOT}(A)$
$1 - \left(\frac{\sum_{i=1}^n \left(\frac{ \{w_i\} + \{w_i\}}{2} - \{w_i\} \cdot a_i \right)^p w_i ^p}{\sum_{i=1}^n w_i ^p} \right)^{\frac{1}{p}}$	$\left(\frac{\sum_{i=1}^n \left(\frac{ \{w_i\} - \{w_i\}}{2} + \{w_i\} \cdot a_i \right)^p w_i ^p}{\sum_{i=1}^n w_i ^p} \right)^{\frac{1}{p}}$	$1 - a$

Πίνακας 20: Συναρτήσεις αξιολόγησης του p-Norm GWFIRS.

Υλοποίηση του p-Norm Extended Boolean Μοντέλου σε σχεσιακή βάση δεδομένων

Υπάρχουν μία σειρά από στρατηγικές που μπορούν να ακολουθηθούν για να υλοποιηθεί ένα GWFIRS πάνω σε μία σύστημα διαχείρισης σχεσιακής βάσης δεδομένων (RDBMS) (βλέπε [7] για επισκόπηση). Μολαταύτα, υπάρχει μία άμεση υλοποίηση του p-Norm GWFIRS πάνω

σε ένα RDBMS στην περίπτωση που οι ερωτήσεις που γίνονται δεκτές από το σύστημα έχουν απλή μορφή σχετική με το [6] και αναλύονται καλύτερα στο [31]. Για να παρουσιάσουμε αυτήν την τεχνική, ας υποθέσουμε ότι οι ερωτήσεις που γίνονται δεκτές από το σύστημα είναι απλές διαζευκτικές ερωτήσεις της μορφής:

$$q = \langle t_{q1}, w_{q1} \rangle OR \dots OR \langle t_{qn}, w_{qn} \rangle, \quad -1 \leq w_{qi} \leq 1, i = 1, \dots, n .$$

Και ας υποθέσουμε, ότι έχουμε το παρακάτω σχεσιακό σχήμα (τα πρωτεύοντα κλειδιά είναι **bold**, ξένα (foreign) κλειδιά είναι υπογραμμισμένα) για να αποθηκεύουμε την πληροφορία που διαχειρίζεται το GWFIRS:

FEATURES(**FeatureID**, FeatureName) αποθηκεύει τα χαρακτηριστικά του συστήματος.

ITEMS(**ItemID**, ItemName) αποθηκεύει το σύνολο των αντικειμένων πληροφορίας.

ITEMS_FEATURES(ItemID, FeatureID, Weight) αποθηκεύει τη σχέση μεταξύ αντικειμένων και χαρακτηριστικών (πχ. Οι συναρτήσεις $i:F \rightarrow [0,1]$).

QUERIES(**QueryID**, FeatureID, QueryWeight) αποθηκεύει τις ερωτήσεις του συστήματος.

Η παρακάτω SQL ερώτηση μπορεί να χρησιμοποιηθεί για να υπολογίζει τα αποτελέσματα για κάθε διαζευκτική ερώτηση που αναγνωρίζετε από το σύστημα (βλέπε τον πίνακα 20 για τον ορισμό της συνάρτησης f_{OR} ενός GWFIRS που ακολουθεί το p-Norm μοντέλο):

SQL [1]:

```
SELECT Q.QueryID, IF.ItemID,
    POWER(SUM(POWER((ABS(SGN(Q.QueryWeight))-
    SGN(Q.QueryWeight))/2+SGN(Q.QueryWeight)*IF.Weight,p)*
    POWER(ABS(Q.QueryWeight),p))
    /SUM(POWER(ABS(Q.QueryWeight),p)),1/p)
FROM ITEMS_FEATURES AS IF, QUERIES AS Q
WHERE IF.FeatureID=Q.FeatureID
GROUP BY Q.QueryID, IF.ItemID
```


Στην παραπάνω SQL δήλωση ενώνοντας τους πίνακες ITEMS_FEATURES και QUERIES πάνω στο πεδίο FeatureID έχουμε για το σύνολο των ερωτήσεων τα αντικείμενα που τις ικανοποιούν. Αυτό που θέλουμε είναι να υπολογίσουμε την f_{OR} ανά ερώτηση και ανά αντικείμενο, έτσι τα ομαδοποιούμε ανά ερώτηση και ανά αντικείμενο. Στο τρίτο πεδίο του SELECT έχουμε τον καθαυτό τύπο της f_{OR} .

Είναι αρκετά απλό να υλοποιήσουμε όμοιες SQL ερωτήσεις στην περίπτωση που ερωτήσεις που αναγνωρίζονται από το σύστημα είναι απλές συζευκτικές ερωτήσεις της μορφής:

$$q = \langle t_{q1}, w_{q1} \rangle \text{ AND } \dots \text{ AND } \langle t_{qn}, w_{qn} \rangle, \quad -1 \leq w_{qi} \leq 1, i = 1, \dots, n$$

Η αντίστοιχη SQL ερώτηση έχει ως ακολούθως (βλέπε τον πίνακα 20 για τον ορισμό της συνάρτησης f_{AND} του p-Norm μοντέλου):

SQL [2]:

```
SELECT Q.QueryID, IF.ItemID,
      1 - POWER(SUM(POWER((ABS(SGN(Q.QueryWeight))+
      SGN(Q.QueryWeight))/2-SGN(Q.QueryWeight)*IF.Weight,p)*
      POWER(ABS(Q.QueryWeight),p))
      /SUM(POWER(ABS(Q.QueryWeight),p)),1/p)
FROM ITEMS_FEATURES AS IF, QUERIES AS Q
WHERE IF.FeatureID=Q.FeatureID
GROUP BY Q.QueryID, IF.ItemID
```

Στην παραπάνω SQL δήλωση ενώνοντας τους πίνακες ITEMS_FEATURES και QUERIES πάνω στο πεδίο FeatureID έχουμε για το σύνολο των ερωτήσεων τα αντικείμενα που τις ικανοποιούν. Αυτό που θέλουμε είναι να υπολογίσουμε την f_{AND} ανά ερώτηση και ανά αντικείμενο, έτσι τα ομαδοποιούμε ανά ερώτηση και ανά αντικείμενο. Στο τρίτο πεδίο του SELECT έχουμε τον καθαυτό τύπο της f_{AND} .

Ο προσεκτικός αναγνώστης θα παρατήρησε ότι η διαφορά του *SQL[2]* από το *SQL[1]* βρίσκεται στο τρίτο πεδίο του **SELECT**, αυτού που υπολογίζει τα βάρη του απαντητικού συνόλου.

Μειώνοντας τις επιβαρύνσεις του χώρου

Είναι αξιοσημείωτο ότι όλες οι χρήσιμες εφαρμογές ενός GWFIRS αντιστοιχούν σε περιβάλλοντα που η πλειονότητα των σχέσεων Item-Feature έχουν μηδενικό βάρος. Η τεχνική που παρουσιάστηκε ως τώρα, βασίζεται στην αποθήκευση όλων των σχέσεων Item-Feature (των μηδενικών όσο και των μη) στον πίνακα ITEMS_FEATURES. Ένας τρόπος να μειωθούν αυτές οι τεράστιες επιβαρύνσεις του χώρου είναι να αποθηκεύονται μόνο οι μη μηδενικές σχέσεις. Σε αυτήν την περίπτωση, εισάγεται μία εικασία στο σχεδιασμό της βάσης δεδομένων η οποία λέει: “Όλες οι σχέσεις μεταξύ αντικειμένων και χαρακτηριστικών Item-Feature που δεν είναι ρητά αποθηκευμένες στη βάση έχουν μηδενική τιμή”. Μολαταύτα, αυτό το τέχνασμα έχει ένα σοβαρό αντίκτυπο στο σύνολο των αποτελεσμάτων: Οι παραπάνω SQL ερωτήσεις δίνουν λάθος αποτελέσματα αν οι επιμέρους οροί της ερώτησης που έχουν μηδενικές τιμές παραλειφθούν από τον πίνακα ITEMS_FEATURES. Αποκαλούμε αυτού του είδους τα προβλήματα “πρόβλημα εξάρτησης του μηδενός” (“*zero dependency problem*”). Εύκολα παρατηρεί κανείς ότι οι μέθοδοι αξιολόγησης που υποφέρουν από αυτό το πρόβλημα είναι αυτές που δεν ικανοποιούν την εξής ιδιότητα: $f(\langle 0, w \rangle, \langle a_1, w_1 \rangle) = f(\langle a_1, w_1 \rangle)$. Κάποιος μπορεί εύκολα να επιβεβαιώσει, από τον πίνακα 20, ότι σε ένα GWFIRS που ακολουθεί το p-Norm μοντέλο τόσο η f_{AND} όσο και η f_{OR} υποφέρουν από το πρόβλημα εξάρτησης του μηδενός.

Υπάρχει παρόλα αυτά μία τεχνική για να ξεπεράσουμε το πρόβλημα εξάρτησης του μηδενός με την κατάλληλη επανεγγραφή κάθε προβληματικού τύπου: Ξεκινώντας από τον αρχικό τύπο και χωρίζοντας το άθροισμα του αριθμητή σε δύο κομμάτια: το πρώτο αθροίζει όλα τα a_i 's με μηδενικές τιμές και το δεύτερο τα μη μηδενικά. Μετά γίνονται όλοι οι απαραίτητοι αλγεβρικοί χειρισμοί έτσι ώστε στο τελικό τύπο μόνο τα μη μηδενικά a_i 's να παρίστανται. Παρατηρήστε ακόμα ότι:

$$\left(\frac{|\{w_i\}| \pm \{w_i\}}{2} \right)^p = \left(\frac{|\{w_i\}| \pm \{w_i\}}{2} \right), p > 0$$

το οποίο θα αποδειχτεί πολύ χρήσιμο στην απλοποίηση του τύπου.

Ας εφαρμόσουμε την τεχνική αυτή για να υπολογίσουμε την συνάρτηση αξιολόγησης f_{AND} του πίνακα 20:

$$\begin{aligned}
 f_{AND}(\langle a_1, w_1 \rangle, \dots, \langle a_n, w_n \rangle) &= 1 - \left(\frac{\sum_{i=1}^n \left(\frac{|\{w_i\}| + \{w_i\}}{2} - \{w_i\} \cdot a_i \right)^p |w_i|^p}{\sum_{i=1}^n |w_i|^p} \right)^{\frac{1}{p}} = \\
 &= 1 - \left(\frac{\sum_{\substack{i=1 \\ a_i=0}}^n \left(\frac{|\{w_i\}| + \{w_i\}}{2} \right) \cdot |w_i|^p + \sum_{\substack{i=1 \\ a_i \neq 0}}^n \left(\frac{|\{w_i\}| + \{w_i\}}{2} - \{w_i\} \cdot a_i \right)^p |w_i|^p}{\sum_{i=1}^n |w_i|^p} \right)^{\frac{1}{p}} = \\
 &= 1 - \left(\frac{\sum_{i=1}^n \left(\frac{|\{w_i\}| + \{w_i\}}{2} \right) \cdot |w_i|^p - \sum_{\substack{i=1 \\ a_i \neq 0}}^n \left(\frac{|\{w_i\}| + \{w_i\}}{2} \right) \cdot |w_i|^p + \sum_{\substack{i=1 \\ a_i \neq 0}}^n \left(\frac{|\{w_i\}| + \{w_i\}}{2} - \{w_i\} \cdot a_i \right)^p |w_i|^p}{\sum_{i=1}^n |w_i|^p} \right)^{\frac{1}{p}} = \\
 &= 1 - \left(\frac{\sum_{i=1}^n \frac{|\{w_i\}| + \{w_i\}}{2} |w_i|^p + \sum_{\substack{i=1 \\ a_i \neq 0}}^n \left(\left(\frac{|\{w_i\}| + \{w_i\}}{2} - \{w_i\} \cdot a_i \right)^p - \frac{|\{w_i\}| + \{w_i\}}{2} \right) \cdot |w_i|^p}{\sum_{i=1}^n |w_i|^p} \right)^{\frac{1}{p}}
 \end{aligned}$$

Η συνάρτηση αξιολόγησης f_{OR} αναθεωρείται ως εξής:

$$f_{OR}(\langle a_1, w_1 \rangle, \dots, \langle a_n, w_n \rangle) = \left(\frac{\sum_{i=1}^n \left(\frac{|\{w_i\}| - \{w_i\}}{2} + \{w_i\} \cdot a_i \right)^p |w_i|^p}{\sum_{i=1}^n |w_i|^p} \right)^{\frac{1}{p}} =$$

$$\begin{aligned}
 &= \left(\frac{\sum_{\substack{i=1 \\ a_i=0}}^n \left(\frac{|\{w_i\}| - \{w_i\}}{2} \right) \cdot |w_i|^p + \sum_{\substack{i=1 \\ a_i \neq 0}}^n \left(\frac{|\{w_i\}| - \{w_i\}}{2} + \{w_i\} \cdot a_i \right)^p |w_i|^p}{\sum_{i=1}^n |w_i|^p} \right)^{\frac{1}{p}} \\
 &= \left(\frac{\sum_{i=1}^n \left(\frac{|\{w_i\}| - \{w_i\}}{2} \right) \cdot |w_i|^p - \sum_{\substack{i=1 \\ a_i \neq 0}}^n \left(\frac{|\{w_i\}| - \{w_i\}}{2} \right) \cdot |w_i|^p + \sum_{\substack{i=1 \\ a_i \neq 0}}^n \left(\frac{|\{w_i\}| - \{w_i\}}{2} + \{w_i\} \cdot a_i \right)^p |w_i|^p}{\sum_{i=1}^n |w_i|^p} \right)^{\frac{1}{p}} \\
 &= \left(\frac{\sum_{i=1}^n \frac{|\{w_i\}| - \{w_i\}}{2} |w_i|^p + \sum_{\substack{i=1 \\ a_i \neq 0}}^n \left(\left(\frac{|\{w_i\}| - \{w_i\}}{2} + \{w_i\} \cdot a_i \right)^p - \frac{|\{w_i\}| - \{w_i\}}{2} \right) \cdot |w_i|^p}{\sum_{i=1}^n |w_i|^p} \right)^{\frac{1}{p}}
 \end{aligned}$$

Ανακεφαλαιώνοντας έχουμε:

$f_{AND}((a_1, w_1), \dots, (a_n, w_n))$	$1 - \left(\frac{\sum_{i=1}^n \frac{ \{w_i\} + \{w_i\}}{2} w_i ^p + \sum_{\substack{i=1 \\ a_i \neq 0}}^n \left(\left(\frac{ \{w_i\} + \{w_i\}}{2} - \{w_i\} \cdot a_i \right)^p - \frac{ \{w_i\} + \{w_i\}}{2} \right) \cdot w_i ^p}{\sum_{i=1}^n w_i ^p} \right)^{\frac{1}{p}}$
$f_{OR}((a_1, w_1), \dots, (a_n, w_n))$	$\left(\frac{\sum_{i=1}^n \frac{ \{w_i\} - \{w_i\}}{2} w_i ^p + \sum_{\substack{i=1 \\ a_i \neq 0}}^n \left(\left(\frac{ \{w_i\} - \{w_i\}}{2} + \{w_i\} \cdot a_i \right)^p - \frac{ \{w_i\} - \{w_i\}}{2} \right) \cdot w_i ^p}{\sum_{i=1}^n w_i ^p} \right)^{\frac{1}{p}}$

Πίνακας 21: Αναθεωρημένες συναρτήσεις αξιολόγησης του p-Norm GWFIRS για την υπέρβαση του προβλήματος εξάρτησης του μηδενός.

Η υλοποίηση σε SQL θέλει κάποια προσοχή. Παρατηρήστε ότι ο παρονομαστής των αναθεωρημένων τύπων f_{OR} και f_{AND} αθροίζει όλους τους όρους της ερώτησης. Ο αριθμητής έχει ένα κομμάτι που αθροίζει όλους τους όρους της ερώτησης και ένα άλλο που αθροίζει τους όρους με μη μηδενικά βάρη.

Για να πραγματοποιούμε αυτό το θέμα, οι αναθεωρημένες SQL ερωτήσεις για των υπολογισμό απλών διαζευκτικών (ή συζευκτικών) ερωτήσεων, γράφονται σαν μία σειρά από δύο SQL ερωτήσεων: Η πρώτη διατρέχει όλους τους όρους της ερώτησης για να υπολογίσει τον παρονομαστή και τον πρώτο όρο του αριθμητή. Η δεύτερη κάνει την ένωση (join) μεταξύ των πινάκων ITEMS_FEATURES και QUERIES και υπολογίζει το τελικό αποτέλεσμα.

SQL [3]:

```
SELECT @denominator=sum(POWER(ABS(QueryWeight),p)),
        @numerator1=sum((ABS(SGN(QueryWeight))-SGN(QueryWeight))/
        2*POWER(ABS(QueryWeight),p))
FROM QUERIES
WHERE Q.QueryID=@qid;

SELECT Q.QueryID, IF.ItemID,
        (POWER(@numerator1+
        SUM(POWER((ABS(SGN(Q.QueryWeight))-
        SGN(Q.QueryWeight))/2+SGN(Q.QueryWeight)*IF.Weight,p)-
        (ABS(SGN(Q.QueryWeight))-SGN(Q.QueryWeight))/2)*
        POWER(ABS(Q.QueryWeight),p))
        /@denominator,1/p)
FROM ITEMS_FEATURES AS IF, QUERIES AS Q
WHERE IF.FeatureID=Q.FeatureID AND Q.QueryID=@qid
GROUP BY Q.QueryID, IF.ItemID
```

SQL [4]:

```
SELECT @denominator=sum(POWER(ABS(QueryWeight),p)),
        @numerator1=sum((ABS(SGN(QueryWeight))+SGN(QueryWeight))/
        2*POWER(ABS(QueryWeight),p))
FROM QUERIES
WHERE Q.QueryID=@qid;

SELECT Q.QueryID, IF.ItemID,
        1 - (POWER(@numerator1+
        SUM(POWER((ABS(SGN(Q.QueryWeight))+
        SGN(Q.QueryWeight))/2-SGN(Q.QueryWeight)*IF.Weight,p)-
        (ABS(SGN(Q.QueryWeight))+SGN(Q.QueryWeight))/2)*
        POWER(ABS(Q.QueryWeight),p))
        /@denominator,1/p)
FROM ITEMS_FEATURES AS IF, QUERIES AS Q
WHERE IF.FeatureID=Q.FeatureID AND Q.QueryID=@qid
GROUP BY Q.QueryID, IF.ItemID
```

Ειδικές Περιπτώσεις

Τέλος, ανεπτυχθήσαν συναρτήσεις αξιολόγησης για δυο ειδικές περιπτώσεις ενός GWFIRS βασισμένο στο p-Norm. Αυτές οι ειδικές περιπτώσεις υπάρχουν στο UP-TV σύστημα.

1^η Ειδική Περίπτωση: Όλα τα w_i 's είναι ίσα με το 1.

Αυτό αντιστοιχεί σε ένα σύστημα που δέχεται ερωτήσεις χωρίς βάρη. Αντικαθιστώντας στις συναρτήσεις του πίνακα 21 παίρνουμε τα αποτελέσματα που φαίνονται στο παρακάτω πίνακα:

$F_{AND}((A_p 1), \dots, (A_N 1))$	$F_{OR}((A_p 1), \dots, (A_N 1))$
$1 - \left(\frac{n - \sum_{\substack{i=1 \\ a_i \neq 0}}^n ((1 - a_i)^p - 1)}{n} \right)^{\frac{1}{p}}$	$\left(\frac{\sum_{\substack{i=1 \\ a_i \neq 0}}^n a_i^p}{n} \right)^{\frac{1}{p}}$

Πίνακας 22: Συναρτήσεις αξιολόγησης για το p-Norm GWFIRS στην ειδική περίπτωση που όλα τα βάρη των ερωτήσεων είναι ίσα με το 1. Οι τύποι είναι αυτοί που δεν υποφέρουν από το πρόβλημα εξάρτησης του μηδενός.

2^η Ειδική Περίπτωση: Όλα τα a_i 's είναι ίσα με το 1.

Άλλη μία ειδική περίπτωση υπάρχει όταν όλα τα a_i 's είναι ίσα με το 1. Αυτή είναι η περίπτωση όπου τα αντικείμενα πληροφορίας συνδέονται με χαρακτηριστικά χωρίς βάρη. Οι τύποι για το *AND* και το *OR* μπορούν να αναπτυχθούν λίγο ώστε να απλοποιηθούν. Ο Παρακάτω πίνακας ανακεφαλαιώνει τα τελικά αποτελέσματα.

$F_{AND}((1, w_1), \dots, (1, w_N))$	$F_{OR}((1, w_1), \dots, (1, w_N))$
$1 - \left(\frac{\sum_{i=1}^n \frac{ \{w_i\} + \{w_i\}}{2} w_i ^p - \sum_{\substack{i=1 \\ a_i \neq 0}}^n \{w_i\} \cdot w_i ^p}{\sum_{i=1}^n w_i ^p} \right)^{\frac{1}{p}}$	$\left(\frac{\sum_{i=1}^n \frac{ \{w_i\} - \{w_i\}}{2} w_i ^p + \sum_{\substack{i=1 \\ a_i \neq 0}}^n \{w_i\} \cdot w_i ^p}{\sum_{i=1}^n w_i ^p} \right)^{\frac{1}{p}}$

Πίνακας 23: Συναρτήσεις αξιολόγησης για το p-Norm GWFIRS στην ειδική περίπτωση που όλα τα βάρη μεταξύ των αντικειμένων και των χαρακτηριστικών είναι ίσα με 1. Οι τύποι είναι αυτοί που δεν υποφέρουν από το πρόβλημα εξάρτησης του μηδενός.

Είναι σχετικά απλή η ανάπτυξη των αντίστοιχων SQL ερωτήσεων για κάθε μία από τις ειδικές περιπτώσεις με κατάλληλες απλοποιήσεις των *SQL*[3] και *SQL*[4].

Μεταφραστικό σχήμα του προφίλ σε δυαδικό δέντρο

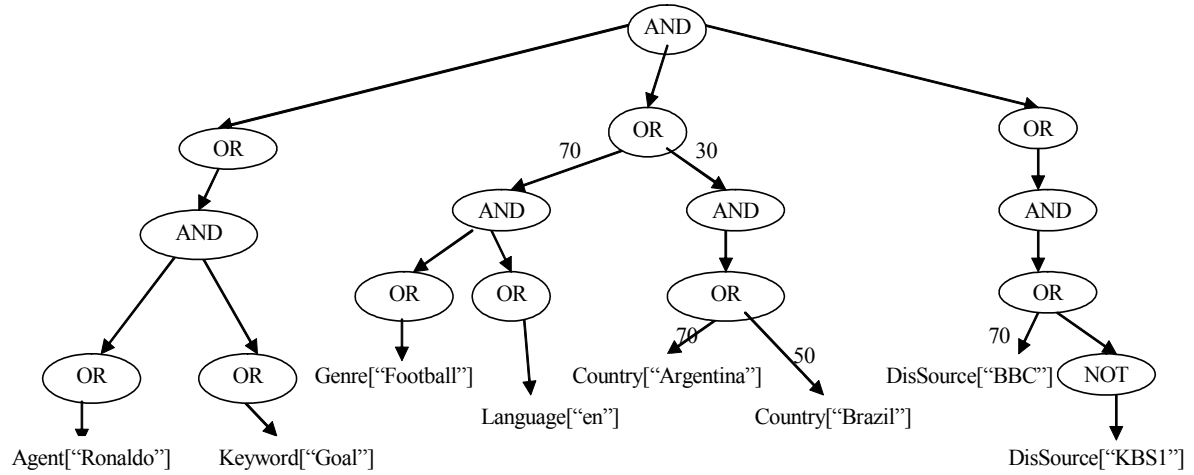
Σε αυτήν την παράγραφο θα μελετηθεί η μέθοδος με την οποία εφαρμόζεται το p-Norm GWFIRS στο σύστημα. Συγκεκριμένα θα αναλυθεί η σημασιολογία των μεταδεδομένων του TV-Anytime για το προφίλ, το οποίο έχει περιγραφεί λεπτομερώς στο κεφάλαιο III, ειδικά για το σύστημα. Στην συνέχεια, θα περιγραφεί το μεταφραστικό μοντέλο που χρησιμοποιείται για την μετατροπή του σε δυαδικό συντακτικό δέντρο. Τέλος, θα παρουσιαστεί το GWFIRS που αξιοποιεί το δυαδικό συντακτικό δέντρο για να υπολογίσει τα μεταδεδομένα περιεχομένου που ικανοποιούν το προφίλ.

Επιπλέον Σημασιολογία του προφίλ

Το σύστημα θέτει επιπλέον σημασιολογία στην ιεραρχία των φίλτρων του προφίλ (δηλ. των FilteringAndSearchPreferences ή για συντομία θα αναφέρεται εφεξής FASP) με σκοπό να παράγει ένα δυαδικό συντακτικό δέντρο από τη δομή του FASP. Αυτό το δέντρο αντιστοιχεί σε μια δυαδική έκφραση που αποτελείται από τους συνδέσμους AND, OR, και NOT. Η Γενική ιδέα που κυριαρχεί είναι ότι κάθε στοιχείο αντιστοιχεί σε μία λογική σύζευξη (AND) των τμημάτων της. Αν κάποιο κομμάτι ενός στοιχείου είναι μία λίστα από περισσότερα του ενός στοιχεία ίδιου τύπου, τότε αυτή η λίστα αντιστοιχεί σε μία λογική διάζευξη (OR). Ένα στοιχείο με αρνητική τιμή προτίμησης αντιστοιχεί σε λογική άρνηση (NOT).

Οι τιμές προτίμησης είναι σημαντικές μόνο στην περίπτωση λιστών από στοιχεία ίδιου τύπου (αντιστοιχεί, όπως προαναφερθεί, σε λογική διάζευξη). Οι τιμές προτίμησης αντιπροσωπεύουν την σχετική βαρύτητα κάθε στοιχείου στην λίστα ως προς τα υπόλοιπα στοιχεία της λίστας.

Το παρακάτω γράφημα απεικονίζει το συντακτικό δέντρο που αντιστοιχεί στο FASP του παραδείγματος της σελίδας 37 λαμβάνοντας υπ' όψιν την σημασιολογία του UP-TV που αναφέρθη παραπάνω:



Σχήμα 8: Παράδειγμα Boolean αναπαράστασης του Filtering and Search Preferences του σχήματος 6.

Το παραπάνω γράφημα είναι μία δυαδική αναπαράσταση του FASP. Δηλαδή:

$$AND \left(\begin{array}{l} AND((Agent[\"Ronaldo\"], 1), (Keyword[\"Goal\"], 1)), \\ OR \left(\begin{array}{l} \left(AND \left(\begin{array}{l} (Genre[\"Football\"], 1), \\ (Language[\"En\"], 1) \end{array} \right), 70 \right), \\ \left(OR \left(\begin{array}{l} (Country[\"Argentina\"], 70), \\ (Country[\"Brasil\"], 50) \end{array} \right), 30 \right) \end{array} \right), \\ OR((DisSource[\"BBC\"], 70), (DisSource[\"KBS1\"], NOT)) \end{array} \right)$$

Στην επόμενη παράγραφο θα μελετηθεί διεξοδικά το μεταφραστικό σχήμα που οδήγησε στην παραπάνω παράσταση.

Ένα γενικό μεταφραστικό σχήμα από ένα φίλτρο σε ένα δυαδικό συντακτικό δέντρο

Ένα γενικός μεταφραστικός μηχανισμός θα οριοθετηθεί εδώ. Αυτός ο μηχανισμός παίρνει ένα φίλτρο, δηλαδή μία δομή FASP (Filtering and Search Preferences), και το μεταμορφώνει σε μία δυαδική έκφραση χρησιμοποιώντας τους τελεστές AND, OR, και NOT του δυαδικού (Boolean) μοντέλου. Η προκύπτουσα δυαδική έκφραση μπορεί να εφαρμοστεί απευθείας στο Εκτεταμένο Δυαδικό (Extended Boolean) μοντέλο της ασαφούς λογικής ώστε να παραγάγει διατεταγμένα αποτελέσματα. Η διαδικασία μετάφρασης είναι ακριβώς αυτή που περιγράφηκε άτυπα στην προηγούμενη υπό-παράγραφο και υπολογίζεται σε ένα συρμό από πάνω προς τα κάτω (top down).

Το πρώτο βήμα είναι να συσχετιστεί ένας κόμβος FASP fs με την σύζευξη των τεσσάρων κλάσεων των παιδιών του, δηλαδή της creation preferences $\{ cr_i \mid i=1,..,n_{cr} \}$, της classification preferences $\{ cl_i \mid i=1,..,n_{cl} \}$, της source preferences $\{ sr_i \mid i=1,..,n_{sr} \}$ και των παιδιών κόμβων FASP $\{ fs_i \mid i=1,..,n_{fs} \}$ ώστε να οριστεί η αντίστοιχη ερώτηση $Q(fs)$:

$$Q(fs) = AND \left(\left(OR_{i=1}^{n_{cr}} (Q(cr_i), cr_i.p), w_{cr} \right), \right. \\ \left(OR_{i=1}^{n_{cl}} (Q(cl_i), cl_i.p), w_{cl} \right), \\ \left(OR_{i=1}^{n_{sr}} (Q(sr_i), sr_i.p), w_{sr} \right), \\ \left. \left(OR_{i=1}^{n_{fs}} (Q(fs_i), fs_i.p), w_{fs} \right) \right)$$

Το αποτέλεσμα της παραπάνω μεταφραστικής μεθόδου είναι μια σύζευξη τεσσάρων διαζευκτικών εκφράσεων κάθε μία εκ των οποίων αντιστοιχεί στα creation preferences, στα classification preferences, στα source preferences και στα παιδιά κόμβους FASP του αρχικού FASP. Στον παραπάνω τύπο είναι $w_{cr} = w_{cl} = w_{sr} = w_{fs} = 1$. Οι αντίστοιχες ερωτήσεις για τα παιδιά FASP $Q(fs_i)$ δημιουργούνται αναδρομικά χρησιμοποιώντας την ίδια μεταφραστική μέθοδο μέχρι τους κόμβους φύλλα FASP.

Ο παρακάτω τύπος δίνει την διαδικασία μετάφρασης για τον ορισμό των ερωτήσεων $Q(cr)$, $Q(cl)$, και $Q(sr)$ που αντιστοιχούν στις δομές creation, classification και source preference:

$$\begin{aligned}
 Q(cr) &= \underset{f \in \{Title, Creator, Keyword, Location, DatePeriod, Tool\}}{AND} \left(\left(\underset{i=1}{OR}^{n_f} (Q(f_i), f_i \cdot p), w_f \right) \right) \\
 Q(cl) &= \underset{f \in \{Country, DatePeriod, LanguageFormat, Language, CaptionLanguage, Form, Genre, Subject, Review, ParentalGuidance\}}{AND} \left(\left(\underset{i=1}{OR}^{n_f} (Q(f_i), f_i \cdot p), w_f \right) \right) \\
 Q(sr) &= \underset{f \in \{DisseminationFormat, DisseminationSource, DisseminationLocation, DisseminationDate, Disseminator, MediaFormat\}}{AND} \left(\left(\underset{i=1}{OR}^{n_f} (Q(f_i), f_i \cdot p), w_f \right) \right)
 \end{aligned}$$

Σε κάθε μία από τις παραπάνω περιπτώσεις, η καινούργια ερώτηση ορίζεται ως σύζευξη συνόλων αποτελούμενα από διαζευκτικές ερωτήσεις, τα βάρη των συζεύξεων είναι όλα 1, $w_f = 1$ $\forall f$. Κάθε μία από τις διαζευκτικές ερωτήσεις αντιστοιχεί σε μία λίστα από χαρακτηριστικά ενός συγκεκριμένου τύπου. Για κάθε ένα από τα ξεχωριστά χαρακτηριστικά σε αυτές τις ερωτήσεις η μετάφραση έχει ως ακολούθως:

$$Q(f) = type(f) \text{ op } f$$

όπου $type(f)$ είναι ο τύπος του χαρακτηριστικού f , op είναι ο τελεστής σύγκρισης (πχ. '=', '<', 'LIKE') που εξαρτάται από τον τύπο του f και την συγκεκριμένη ερμηνεία της τιμής του f . Στις περισσότερες περιπτώσεις, χρησιμοποιείται μόνον ο τελεστής ισότητας '=' και το αποτέλεσμα είναι 1 αν η συνθήκη είναι αληθής (TRUE) ή 0 αν η έκφραση αποτιμάται σε ψευδής (FALSE). Αυτός ο τύπος τερματίζει την αναδρομική διαδικασία κατασκευής.

Αυτή η μεταφραστική διαδικασία χρησιμοποιήθηκε για να παραγάγει δυαδική έκφραση που απεικονίζεται σαν συντακτικό δέντρο στο σχήμα 8 από την δομή FASP του σχήματος 6.

Το GWFIRS που υλοποιεί τη συσχέτιση μεταξύ FASP και των μεταδεδομένων του περιεχομένου

Αφού ορίσθηκε και τυπικά η μετάφραση ενός FASP σε μία έκφραση του Γενικευμένου Εκτεταμένου Δυναδικού Μοντέλου (Generalized Extended Boolean Model) –αυτού, δηλαδή, που δέχεται όχι μόνο θετικά αλλά και αρνητικά βάρη και τα μεταφράζει σε αρνήσεις – είναι απαραίτητο να ορίσουμε το GWFIRS που αξιολογεί τα μεταδεδομένα πληροφορίας.

Αυτό είναι σχετικά σαφές: Υποθέτοντας ότι υπάρχει ένα σύνολο *Filters* (φίλτρων) που περιέχουν όλους τους κόμβους FASP που πρέπει να συσχετιστούν. Το GWFIRS που υπολογίζει την ομοιότητα μεταξύ των κόμβων FASP του F και των προγραμμάτων ορίζεται ως $\mathbf{C} = \langle F, I, Q, E \rangle$ όπου:

$F = \{\text{όλα τα πιθανά χαρακτηριστικά}\},$

$I = \{\text{όλα τα προγράμματα σαν συναρτήσεις των χαρακτηριστικών στο διάστημα } \{0,1\}, \text{ πχ. τι χαρακτηριστικό υπάρχει σε κάθε πρόγραμμα}\},$

$Q = \{Q(fasp) \mid \text{για κάθε } fasp \in Filters\},$

E είναι η συνάρτηση αξιολόγησης για το GWFIRS.

Υλοποίηση

Για να διαφωτιστούν πλήρως οι λεπτομέρειες της υλοποίησης των μηχανισμών διήθησης που εφαρμόστηκαν, είναι πρώτα απαραίτητο να περιγράψουμε το σχήμα της βάσης (τους σχεσιακούς πίνακες) που χρησιμοποιούνται για να αποθηκεύουν τόσο τα μεταδεδομένα των προγραμμάτων όσο και την πληροφορία των FASP. Το πλήρες σχήμα περιγράφεται στο [38] και στο [39]. Για να περιγραφεί, ωστόσο, η υλοποίηση της βασισμένης στο περιεχόμενο μονάδας διήθησης αρκεί να υιοθετηθεί ένα απλοποιημένο μοντέλο βάσει του οποίου κάθε πρόγραμμα περιγράφεται από μία λίστα από συγκεκριμένου τύπου χαρακτηριστικά και κάθε κόμβος FASP μοντελοποιείται σαν μία λίστα από επί μέρους φίλτρα κάθε ένα εκ των οποίων έχει, επίσης, συγκεκριμένου τύπου χαρακτηριστικά.

Το σχήμα της Βάσης

Το απλοποιημένο σχήμα της βάσης έχει ως ακολούθως. Για κάθε τύπου χαρακτηριστικό (Feature) F υπάρχει ένας πίνακας της μορφής:

Program_Feature(**CRID**, **FeatureID**)

Αυτός ο πίνακας συσχετίζει κάθε πρόγραμμα (παραπέμπεται με το CRID του) με χαρακτηριστικά συγκεκριμένου τύπου, $Feature \in \{Title, Creator, \text{κλπ.}\}$. Αυτή η σχέση χρησιμοποιεί έμμεσα ένα μοντέλο όπου κάθε πρόγραμμα-χαρακτηριστικό έχει βάρος είτε 0 είτε 1. Αν το βάρος είναι 0 (δηλ. Το αντίστοιχο χαρακτηριστικό δεν σχετίζεται με το συγκεκριμένο πρόγραμμα) τότε δεν υπάρχει εγγραφή στον πίνακα Program_Feature. Αν τώρα το βάρος είναι 1, υπάρχει εγγραφή στον πίνακα. Παρατηρήστε ότι αυτό είναι μόνον το γενικό σχήμα τέτοιων πινάκων. Οι ειδικοί πίνακες για κάθε τύπου χαρακτηριστικό μπορεί να είναι πιο περίπλοκοι όπως ισχύει στην περίπτωση για τα χαρακτηριστικά που αναφέρονται σε ημερομηνίες (π.χ. η χρονική περίοδος που δημιουργήθηκε ένα πρόγραμμα). Παρόλα αυτά, η αναλυτική έκθεση των λεπτομερειών της υλοποίησης δεν επηρεάζεται από την υπόθεση ότι όλοι οι τύποι χαρακτηριστικών περιγράφονται από με το παραπάνω απλοποιημένο σχήμα.

Για την αποθήκευση των επιμέρους φίλτρων των κόμβων FASP, η βάση έχει ένα πίνακα για κάθε τύπου χαρακτηριστικό F. Επειδή τα φίλτρα συνδέονται με τους κόμβους FASP μέσω των Classification, Creation και Source Preferences θα χρειαστούμε δυο είδη πινάκων. Ο πρώτος πίνακας θα συνδέει χαρακτηριστικά με προτιμήσεις και θα έχει την μορφή:

Preference_Feature (**PreferenceID**, **FeatureID**, Pvalue)

Όπου Preference \in {Classification, Creation, Source} και Feature \in {Title, Creator, κλπ.} ανάλογα με το Preference.

Παρατηρήστε ότι οι τύποι των χαρακτηριστικών των κόμβων FASP δεν αντιστοιχίζονται μοναδικά με τους τύπους των χαρακτηριστικών των προγραμμάτων (π.χ. ο τύπος Title των κόμβων FASP αντιστοιχίζεται με τους τύπους Title, ShortTitle, κλπ. των προγραμμάτων). Μολαταύτα, και πάλι μπορούμε να θεωρήσουμε ότι αντιστοιχίζονται μοναδικά, για την ανάλυση της υλοποίησης, αν υποθέσουμε ότι θα κατασκευάζονται κατάλληλοι πίνακες (αντίστοιχοι με τους Program_Feature) οι οποίοι θα περιέχουν την συσχέτιση ενός προγράμματος με όλους τους τύπους χαρακτηριστικών εκείνων που αντιστοιχίζονται με ένα τύπο χαρακτηριστικών κόμβου FASP.

Για την αποθήκευση των επιμέρους προτιμήσεων των κόμβων FASP, η βάση έχει ένα πίνακα για κάθε τύπου προτίμησης P, ο οποίος έχει την μορφή:

FASP_Preference(**FASPID**, **PreferenceID**, Pvalue)

Και πάλι Preference \in {Classification, Creation, Source}

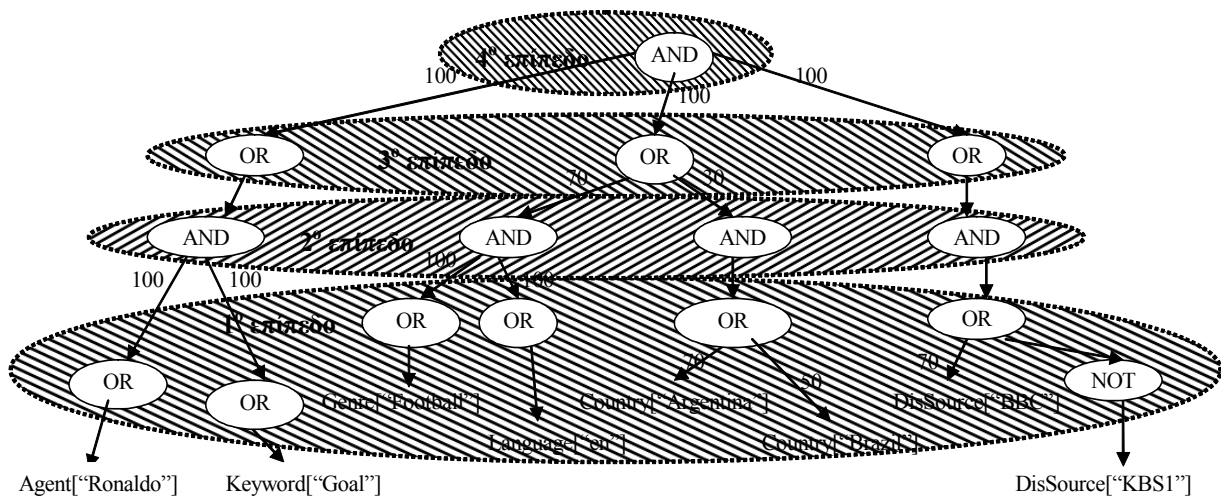
Ακόμα χρειάζεται ένας ακόμα πίνακας για την ιεραρχία των κόμβων FASP, ο οποίος έχει την μορφή:

FASP(UserID, **FASPID**, ParentFASPID, Pvalue)

Υπολογισμός Μεμονωμένων Κόμβων

Το ταίριασμα των κόμβων FASP με τα μεταδεδομένα των προγραμμάτων γίνεται σε τέσσερα διακριτά επίπεδα τα οποία αντιστοιχούν σε τέσσερις SQL δηλώσεις. Η πρώτη υπολογίζει τα επιμέρους φίλτρα, δηλ. την διάζευξη μεταξύ όμοιου τύπου χαρακτηριστικών, χρησιμοποιώντας την συνάρτηση f_{OR} για την ειδική περίπτωση που τα βάρη μεταξύ αντικείμενων και χαρακτηριστικών είναι ίσα με 1 (ειδική περίπτωση 2, βλέπε πίνακα 23). Η δεύτερη δήλωση SQL υπολογίζει τις μεμονωμένες προτιμήσεις, δηλ. την σύζευξη μεταξύ των επιμέρους φίλτρων ομαδοποιημένους ανά Classification, Creation και Source Preferences εφαρμόζοντας τον τύπο f_{AND} για την ειδική περίπτωση που τα βάρη της ερώτησης είναι 1 (βλέπε πίνακα 22). Η Τρίτη, τώρα, δήλωση SQL υπολογίζει τις επιμέρους προτιμήσεις, την διάζευξη μεταξύ των μεμονωμένων προτιμήσεων ομαδοποιημένων στις εξής τρεις ομάδες: Classification, Creation και Source Preferences, χρησιμοποιώντας την συνάρτηση f_{OR} για την γενική περίπτωση του πίνακα 21. Η τέταρτη και τελευταία δήλωση SQL υπολογίζει το τελικό αποτέλεσμα, δηλ. την σύζευξη μεταξύ των επιμέρους προτιμήσεων, εφαρμόζοντας τον τύπο f_{AND} της ειδικής περίπτωσης που όλα τα βάρη της ερώτησης είναι 1 (βλέπε πίνακα 21).

Σχηματικά τα επίπεδα φαίνονται στο επόμενο σχήμα. Έχουν προστεθεί και βάρη στα AND, είναι 100 και συμβολίζουν το μοναδιαίο βάρος των συζεύξεων. Αργότερα θα παρουσιαστεί και μία διαφορετική προσέγγιση όπου τα βάρη των συζεύξεων θα μπορούν να παίρνουν και μη μοναδιαίες τιμές.



Σχήμα 9: Τα τέσσερα επίπεδα στα οποία χωρίζεται η διαδικασία ταιριάσματος των μεταδεδομένων των προγραμμάτων με τους κόμβους FASP.

Υπολογισμός των Επιμέρους Φίλτρων

Στην παρακάτω δήλωση SQL πρέπει να οριστούν δύο παράμετροι πριν την εκτέλεση της δήλωσης:

@p αντιστοιχεί στην παράμετρο p του p-Norm μοντέλου. Δίνοντας διαφορετικές τιμές σε αυτήν την παράμετρο είναι δυνατόν να αλλάζουμε το αποτέλεσμα του τύπου.

@userid αντιστοιχεί στο id του χρήστη που θα υπολογιστεί.

Το αποτέλεσμα της εκτέλεσης αποθηκεύεται σε έναν προσωρινό πίνακα PartialFilter(FASPID, PreferenceID, CRID,Pvalue) έτσι ώστε η δεύτερη SQL δήλωση να μπορεί να διατυπωθεί.

SQL [5]:

```
INSERT INTO Temp1(FASPID, PreferenceID, denominator, addend)
SELECT FP.FASPID, P. PreferenceID
        sum(power(abs(P.Pvalue)/100,@p)),
        sum(power(if(P.Pvalue<0,-P.Pvalue,0)/100,@p))
FROM FASP as F, FASP_PREFERENCE as FP, Preference_Feature as P
WHERE F.UserID = @userid
        AND F.FASPID = FP.FASPID
        AND FP.PreferenceID = P. PreferenceID
GROUP BY FP.FASPID, P. PreferenceID;
```

```
INSERT INTO PartialFilter(FASPID, PreferenceID, CRID,Pvalue)
SELECT FP.FASPID, Pr. PreferenceID, P.CRID,
        100*power((T.addend+sum(sign(F.Pvalue)*
        power(abs(F.Pvalue)/100,@p)))/ T.denominator,1/@p)
FROM FASP as F, FASP_Preference as FP, Preference_Feature as Pr,
        Temp1 as T, Program_Feature as P
WHERE F.UserID= @userid
        AND F.FASPID = FP.FASPID
        AND Pr.PreferenceID = T.PreferenceID
        AND F.PreferenceID = Pr. PreferenceID
        AND Pr.FeatureID=P.FeatureID
GROUP BY FP.FASPID, Pr. PreferenceID, P.CRID;
```

Υπολογισμός των Μεμονωμένων Προτιμήσεων

Στην παρακάτω δήλωση SQL πρέπει να οριστεί μία παράμετρος και ένας πίνακας πριν την εκτέλεση της δήλωσης:

@p αντιστοιχεί στην παράμετρο p του p-Norm μοντέλου. Δίνοντας διαφορετικές τιμές σε αυτήν την παράμετρο είναι δυνατόν να αλλάζουμε το αποτέλεσμα του τύπου.

PreferenceCount(FASPID, PreferenceID, Count) αντιστοιχεί στο αριθμό των επιμέρους φίλτρων κάθε προτίμησης του κόμβου FASP που υπολογίζεται.

Το αποτέλεσμα της εκτέλεσης αποθηκεύεται σε έναν προσωρινό πίνακα `DistinctPreferences(FASPID, PreferenceID, CRID, Pvalue)` έτσι ώστε η τρίτη SQL δήλωση να μπορεί να διατυπωθεί.

SQL [6]:

```
INSERT INTO DistinctPreferences(FASPID, PreferenceID, CRID,Pvalue)
SELECT P.FASPID, P. PreferenceID, P.CRID,
        1-power(C.Count-sum(power(1-power(P.Pvalue,1/@p),@p)-1)/C.Count,
        1/@p)
FROM PartialFilter as P,
        PreferenceCount as C
WHERE P.FASPID = C.FASPID
        P. PreferenceID =C. PreferenceID
GROUP BY P.FASPID, P. PreferenceID, P.CRID;
```

Υπολογισμός των Επιμέρους Προτιμήσεων

Στην παρακάτω δήλωση SQL πρέπει να οριστούν δύο παράμετροι πριν την εκτέλεση της δήλωσης:

@p αντιστοιχεί στην παράμετρο *p* του *p-Norm* μοντέλου. Δίνοντας διαφορετικές τιμές σε αυτήν την παράμετρο είναι δυνατόν να αλλάζουμε το αποτέλεσμα του τύπου.

@userid αντιστοιχεί στο *id* του χρήστη που θα υπολογιστεί.

Το αποτέλεσμα της εκτέλεσης αποθηκεύεται σε έναν προσωρινό πίνακα `PartialPreferences(FASPID, CRID, Pvalue)` έτσι ώστε η τέταρτη SQL δήλωση να μπορεί να διατυπωθεί.

SQL [7]:

```
INSERT INTO Temp2(FASPID, denominator, addend)
SELECT FP.FASPID,
       sum(power(abs(F.Pvalue)/100,@p)),
       sum(power(if(F.Pvalue<0,-F.Pvalue,0)/100,@p))
FROM FASP as F, FASP_Preference as FP
WHERE F.UserID = @userid
      AND F.FASPID= FP.FASPID
GROUP BY FP.FASPID;

INSERT INTO PartialPreferences(FASPID, CRID,Pvalue)
SELECT FP.FASPID, P.CRID,
       power((T.addend +sum(if(FP.Pvalue>0, power(FP.Pvalue/100, @p),
                                power(1-FP.Pvalue/100, @p)-1)*
                                power(P.Pvalue, @p)))/T.denominator, 1/@p)*100
FROM FASP as F, FASP_Preference as FP, DistinctPreferences as P, Temp2 as T
WHERE F.UserID = @userid
      AND F.FASPID= FP.FASPID
      AND F.PreferenceID = Pr. PreferenceID
      AND T.FASPID = F.FASPID
GROUP BY FP.FASPID, P.CRID;
```

Τελικός υπολογισμός κόμβου

Στην παρακάτω δήλωση SQL πρέπει να οριστεί μία παράμετρος και ένας πίνακας πριν την εκτέλεση της δήλωσης:

@p αντιστοιχεί στην παράμετρο p του p-Norm μοντέλου. Δίνοντας διαφορετικές τιμές σε αυτήν την παράμετρο είναι δυνατόν να αλλάζουμε το αποτέλεσμα του τύπου.

PreferenceFASPCount(FASPID, Count) αντιστοιχεί στο αριθμό των επιμέρους προτιμήσεων του χρήστη του οποίου οι κόμβοι FASP υπολογίζονται.

Το αποτέλεσμα της εκτέλεσης αποθηκεύεται σε έναν προσωρινό πίνακα FASP_Program(FASPID, CRID, Pvalue) έτσι ώστε οι SQL δηλώσεις για την ιεραρχία να μπορούν να διατυπωθούν.

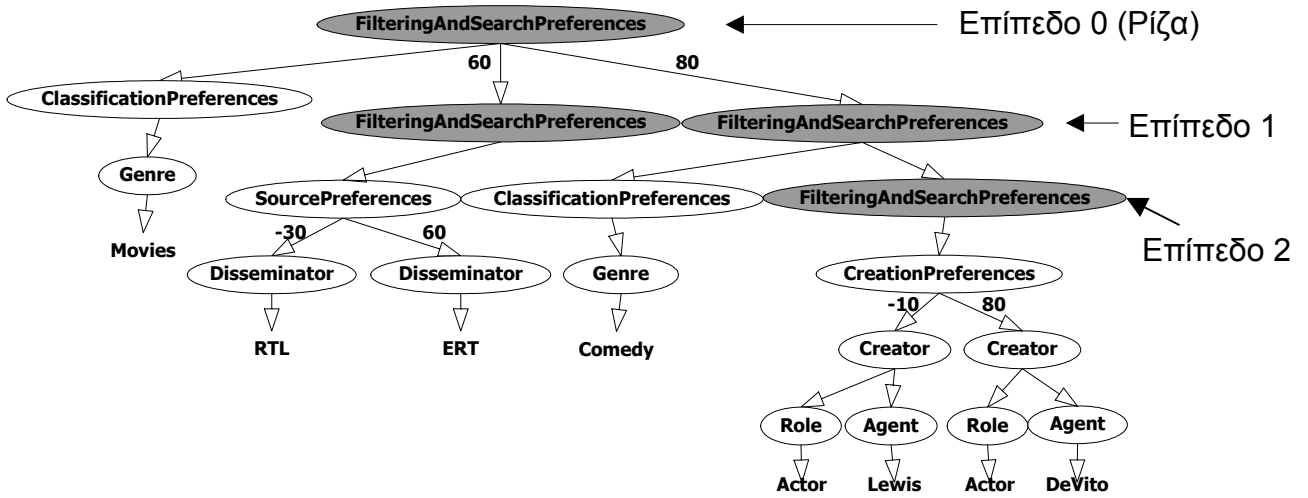
SQL [8]:

```
INSERT INTO FASP_Program(FASPID, CRID, Pvalue)
SELECT P.FASPID, P.CRID,
       100*(1-power((C.Count-sum(power(1-P.Pvalue/100,@p)-1))
                    /C.Count,1/@p))
FROM PartialPreferences as P, PreferenceFASPCount as C
WHERE P.FASPID = C.FASPID
GROUP BY P.FASPID, P.CRID;
```

Υπολογισμός του συνολικού προφίλ του χρήστη

Το τελικό αποτέλεσμα προκύπτει αν τα αποτελέσματα από τους κόμβους FASP τα συνενώσουμε κατά τον τρόπο που περιγράφεται στο μεταφραστικό σχήμα σε αυτό το κεφάλαιο. Δηλαδή όλα τα παιδιά ενός κόμβου ενώνονται με διάζευξη, ενώ μετά ενώνονται με τον πατέρα με σύζευξη. Έχουμε, δηλαδή, δύο διακριτά επίπεδα καθένα εκ των οποίων αντιστοιχεί σε μία δήλωση SQL. Η πρώτη υπολογίζει την διάζευξη των παιδιών με την συνάρτηση f_{OR} για την γενική περίπτωση (βλέπε πίνακα 21). Η δεύτερη δήλωση SQL υπολογίζει την σύζευξη του πατέρα με τα προηγούμενα αποτελέσματα (εφαρμόζοντας τον τύπο f_{AND} για την ειδική περίπτωση που τα βάρη της ερώτησης είναι 1 (βλέπε πίνακα 22)).

Για την ορθή κατανόηση της ανάλυσης που ακολουθεί ο αναγνώστης θα ήταν σκόπιμο να έχει κατά νου ότι η διαδικασία επαναλαμβάνεται για κάθε ένα επίπεδο του δέντρου με φορά από τα φύλλα στην ρίζα. Σχηματικά φαίνεται παρακάτω:



Σχήμα 10: Σχηματική αναπαράσταση ιεραρχικού προφίλ όπου φαίνονται τα επίπεδα του δέντρου

Δηλαδή η διαδικασία τρέχει μία φορά για το επίπεδο 2, μετά για το επίπεδο 1, ενώ για την ρίζα (επίπεδο 0) τρέχει μόνο η δήλωση SQL που αφορά την διάζευξη των τέκνων. Ο αναγνώστης μπορεί να θυμηθεί από το κεφάλαιο 2 ότι ένα προφίλ μπορεί να έχει στο επίπεδο της ρίζας οποιονδήποτε αριθμό από κόμβους FASP.

Σε αυτό το σημείο θα χρειαστούμε τον πίνακα:

FASP_L(FASPID, ParentFASPID, Pvalue, level)

ο οποίος διατηρεί όλους τους κόμβους FASP του τρέχοντος χρήστη έχοντας υπολογίσει και το επίπεδο που βρίσκεται καθένας εκ των οποίων στο δέντρο του προφίλ. Ο υπολογισμός γίνεται αν τρέξει η παρακάτω δήλωση SQL για $n = \lceil \log_2(\#FASP) \rceil$ φορές

SQL [9]:

```

REPLACE INTO FASP_L(FASPID, ParentFASPID, Pvalue, level)
SELECT C.FASPID, P.ParentFASPID, C.Pvalue,
        P.level+C.level
FROM FASP_L as P, FASP_L as C
WHERE C.ParentFASPID=P.FASPID;
  
```

Υπολογισμός Ένωσης Τέκνων

Στην παρακάτω δήλωση SQL πρέπει να οριστούν δύο παράμετροι πριν την εκτέλεση της δήλωσης:

@p αντιστοιχεί στην παράμετρο *p* του *p*-Norm μοντέλου. Δίνοντας διαφορετικές τιμές σε αυτήν την παράμετρο είναι δυνατόν να αλλάξουμε το αποτέλεσμα του τύπου.

@level αντιστοιχεί στο τρέχον επίπεδο της εκτέλεσης.

FASP_L(FASPID, ParentFASPID, Pvalue, level) αντιστοιχεί στους κόμβους FASP του τρέχοντος χρήστη με υπολογισμένο το επίπεδο (*level*) που βρίσκεται στο δέντρο του προφίλ.

Το αποτέλεσμα της εκτέλεσης αποθηκεύεται σε έναν προσωρινό πίνακα *Children(ParentFASPID, CRID, Pvalue)* έτσι ώστε η επόμενη SQL δήλωση να μπορεί να διατυπωθεί.

SQL [10]:

```
INSERT INTO Temp3(ParentFASPID, denominator, addend)
SELECT F.ParentFASPID,
    ifnull(sum(power(abs(F.Pvalue)/100,@p)),0),
    sum(power(if(F.Pvalue<0,-F.Pvalue,0)/100,@p))
FROM FASP_L as F
WHERE F.level = @level
    AND F.level > 0
GROUP BY F.ParentFASPID;
```

```
INSERT INTO Children(ParentFASPID, CRID,Pvalue)
SELECT F. ParentFASPID, P.CRID,
    power((T.addend +sum(if(F.Pvalue>0, power(F.Pvalue/100, @p),
        power(1-F.Pvalue/100, @p)-1)*
        power(abs(P.Pvalue), @p)))/T.denominator, 1/@p)*100
FROM FASP_L as F, FASP_Program as P, Temp3 as T
WHERE F.FASPID= P.FASPID
    AND T.ParentFASPID = F. ParentFASPID
GROUP BY F.ParentFASPID, P.CRID;
```

Υπολογισμός Ένωσης Τέκνων με Πατέρα

Στην παρακάτω δήλωση SQL πρέπει να οριστούν δύο παράμετροι πριν την εκτέλεση της δήλωσης:

@a, *@b* αντιστοιχούν σε διαφορετικά βάρη ως προς την σημασία που δίνετε στον πατέρα έναντι των παιδιών. Δίνοντας διαφορετικές τιμές σε αυτήν την παράμετρο είναι δυνατόν να αλλάξουμε το αποτέλεσμα του τύπου.

@level αντιστοιχεί στο τρέχον επίπεδο της εκτέλεσης.

FASP_L(FASPID, ParentFASPID, Pvalue, level) αντιστοιχεί στους κόμβους FASP του τρέχοντος χρήστη με υπολογισμένο το επίπεδο (*level*) που βρίσκεται στο δέντρο του προφίλ.

SQL [11]:

```
REPLACE INTO FASP_Program
SELECT P.FASPID, P.CRID,
    if(C.CRID IS NULL, P.Pvalue ,(C.Pvalue*@b+P.Pvalue *@a)/(@a+@b))
FROM FASP_Program as P LEFT JOIN
    Children as C ON (P.CRID =C.CRID),
    FASP_L as F
WHERE F.FASPID=P.FASPID
    AND (S.ParentFASPID=P.FASPID OR C.ParentFASPID IS NULL)
    AND F.level = @level-1;
```

Επέκταση του TV-Anytime

Το TV-Anytime δεν έχει προβλέψει την ύπαρξη βαρών μεταξύ των τύπων χαρακτηριστικών (μεταξύ π.χ. του Genre και του Creator) καθώς επίσης και μεταξύ των Creation, Classification και Source Preferences. Εκτός από ενδιαφέρον, είναι σε μερικές περιπτώσεις και χρήσιμο να παρέχεται από το σύστημα αυτή η δυνατότητα ώστε το μοντέλο να γίνει ακόμα πιο ισχυρό. Σε αυτήν την περίπτωση οι δηλώσεις SQL που πρέπει να αλλάξουν είναι η 6 και η 8 και να γίνουν σύμφωνα με τον τύπο f_{AND} του πίνακα 21. Οι πίνακες της βάσης αλλάζουν ως εξής για να διατηρούν την επιπλέον πληροφορία:

Ο FASP γίνεται:

FASP(UserID, **FASPID**, ParentFASPID, Pvalue, CreationPreferencesWeight,
ClassificationPreferencesWeight, SourcePreferencesWeight)

Ενώ οι πίνακες FASP_Preference όπου Preference \in {Creation, Classification, Source} γίνονται

FASP_Preference(**FASPID**, **PreferenceID**, Pvalue, FeatureType1Weight,
FeatureType2Weight, ...)

Όπου τα FeatureType1 ,2 κλπ. είναι τα Genre, Title, Keyword, κλπ. ανάλογα με το Preference.

Για τον υπολογισμό των μεμονωμένων προτιμήσεων έχουμε:

Ο πίνακας *PreferenceWeight*(**FASPID**, **PreferenceID**, *toadd*, *denominator*) έχει την πληροφορία για τον αριθμό που πρέπει να προστεθεί για διαμορφωθεί ο αριθμητής (*toadd*) καθώς επίσης και τον αριθμό του παρονομαστή (*denominator*). Αυτός ο πίνακας υπολογίζεται σταδιακά.

Επίσης, ο πίνακας PartialFilter κατά τον υπολογισμό του από την SQL δήλωση 5 έχει και την πληροφορία FeatureWeight ανάλογα με το Feature από όπου προέρχεται.

SQL [12]:

```
INSERT INTO DistinctPreferences(FASPID, PreferenceID, CRID,Pvalue)
SELECT P.FASPID, P. PreferenceID, P.CRID,
        1-power(W.toadd+sum(if(FeatureWeight > 0, power(1-P.Pvalue,@p)-1,
        power(P.Pvalue, @p))*power(abs(P.FeatureWeight), @p))/ W.denominator,
        1/@p)
FROM PartialFilter as P,
        PreferenceWeight as W
WHERE P.FASPID = W.FASPID
        P. PreferenceID =W. PreferenceID
GROUP BY P.FASPID, P. PreferenceID, P.CRID;
```

Για τον τελικό υπολογισμό των κόμβων έχουμε:

Ο πίνακας *PreferenceFASPWeight(FASPID, toadd, denominator)* έχει την πληροφορία για τον αριθμό που πρέπει να προστεθεί για διαμορφωθεί ο αριθμητής (toadd) καθώς επίσης και τον αριθμό του παρονομαστή (denominator). Αυτός ο πίνακας, όπως και προηγουμένως, υπολογίζεται σταδιακά.

Επίσης, ο πίνακας *PartialPreferences* κατά τον υπολογισμό του από την SQL δήλωση 7 έχει και την πληροφορία *PreferenceWeight* ανάλογα με το *Preference* από όπου προέρχεται.

SQL [13]:

```
INSERT INTO FASP_Program(FASPID, CRID, Pvalue)
SELECT P.FASPID, P.CRID,
        100*(1-power(W.toadd+sum(if(P.PreferenceWeight > 0, power(1-
        P.Pvalue,@p)-1, power(P.Pvalue, @p))*power(abs(P. PreferenceWeight), @p))/
        W.denominator, 1/@p))
FROM PartialPreferences as P, PreferenceFASPWeight as W
WHERE P.FASPID = W.FASPID
GROUP BY P.FASPID, P.CRID;
```

Διατήρηση των χαρακτηριστικών του προφίλ που ταιριάζουν με κάθε πρόγραμμα

Όπως αναφέρθηκε στην εισαγωγή διατηρείται πληροφορία για κάθε πρόγραμμα που επιστρέφεται από το ταίριασμα. Αυτή η πληροφορία αφορά τα χαρακτηριστικά του προφίλ που ταιριάζουν με τα μεταδεδομένα του προγράμματος ώστε να προταθεί ως ενδιαφέρον για

τον χρήστη. Αργότερα, αυτή η πληροφορία μπορεί να χρησιμοποιηθεί από μία διεπαφή (User Interface) δείχνοντας, ουσιαστικά, τον λόγο για τον οποίο το σύστημα του προτείνει αυτό το πρόγραμμα. Έτσι ο χρήστης κατανοεί καλύτερα την λειτουργία του συστήματος και αφενός μαθαίνει να φτιάχνει καλύτερα το προφίλ του, ενώ, αφετέρου, καλλιεργείται η εμπιστοσύνη του προς αυτό, αφού διαπιστώνει, με αυτόν τον τρόπο, ότι δεν συμπεριφέρεται αυθαίρετα αλλά σύμφωνα με τις προτιμήσεις του.

Η πληροφορία διατηρείται σε πίνακες της μορφής:

Matching_Feature (UserID, CRID, FeatureID)

Όπου Feature \in {Creator, Keyword, Country, DatePeriod, ...}. Οι πίνακες αυτοί προκύπτουν με SQL δηλώσεις της μορφής:

SQL [14]:

```
INSERT INTO Matching_Feature (UserID, CRID, FeatureID)
SELECT FP.FASPID, Pr. PreferenceID, P.CRID
FROM FASP as F, FASP_Preference as FP, Preference_Feature as Pr,
      Program_Feature as P
WHERE F.FASPID = FP.FASPID
      AND Pr.PreferenceID = T.PreferenceID
      AND F.PreferenceID = Pr. PreferenceID
      AND Pr.FeatureID=P.FeatureID;
```

Ειδικά θέματα που αφορούν την απόδοση

Το σύστημα υλοποιήθηκε όπως παρουσιάστηκε στις παραπάνω παραγράφους πάνω από την σχεσιακή βάση δεδομένων του UP-TV ([38], [39]) με οδηγό πάντοτε τις αντιστοιχίσεις των χαρακτηριστικών των μεταδεδομένων των προγραμμάτων με τα χαρακτηριστικά του προφίλ του χρήστη. Δόθηκε ιδιαίτερη έμφαση κατά την υλοποίηση στην βελτιστοποίηση της απόδοσης του συστήματος ώστε ο συνολικός χρόνος να είναι επιτρεπτός. Για αυτόν τον σκοπό, τοποθετήθηκαν, όπου κρίθηκε απαραίτητο, οι κατάλληλοι δείκτες (indexes) στην βάση. Η απόδοση ήταν ικανοποιητική τόσο για την περίπτωση που το σύστημα λειτουργεί

κατ' απαίτηση (online) όσο και στην περίπτωση που υπολογίζει τα αποτελέσματα σε κάποιο άσχετο χρόνο (offline).

Εφαρμογές της βασισμένης στο περιεχόμενο μονάδας διήθησης

Το σύστημα δύναται να χρησιμοποιηθεί για μία πληθώρα σκοπών. Συγκεκριμένα:

- Για να βρίσκει όλα τα προγράμματα που ενδιαφέρουν όλους τους χρήστες που είναι στην βάση (κύρια εφαρμογή). Τρέχει σε στιγμές που το υπόλοιπο σύστημα δεν χρησιμοποιείται και όταν ο χρήστης ζητήσει να δει τα προγράμματα που ταιριάζουν στο προφίλ του και είναι διαθέσιμα, αυτά έχουν ήδη υπολογιστεί.
- Για να βρίσκει όλα τα προγράμματα που ενδιαφέρουν ένα συγκεκριμένο χρήστη. Συνήθως αυτή η εφαρμογή βρίσκει απήχηση στην αναζήτηση βάση κάποιων κριτηρίων που θέτει ο χρήστης και όχι ειδικά για το προφίλ του. Εδώ υπολογίζει τα αποτελέσματα την στιγμή που θα του ζητηθούν. Είναι πολύ φυσικό στην περίπτωση αυτή θέματα απόδοσης να είναι ζωτικής σημασίας.
- Για να βρίσκει την ομοιότητα μεταξύ χρηστών βασισμένη στα προφίλ τους. Το σκεπτικό σε αυτή την περίπτωση είναι αντί να συγκριθούν δύο δενδρικές δομές θα μπορούσαν να συγκριθούν τα αποτελέσματα που προκύπτουν από την διήθηση. Αυτή είναι ορθότερη και απλούστερη προσέγγιση μιας και δύο αρκετά διαφορετικά προφίλ είναι πιθανόν να επιστρέφουν παρόμοια σύνολα προγραμμάτων. Εξάλλου, η δομή που προτείνει το MPEG-7 έχει την δυνατότητα να εκφραστούν ίδια προφίλ με πολλούς διαφορετικούς τρόπους. Οπότε για να συγκριθούν δυο δομές θα πρέπει να κατασκευαστούν περίπλοκοι αλγόριθμοι. Αυτή η προσέγγιση μπορεί να χρησιμοποιηθεί σε συνεργατική διήθηση (collaboration filtering). Περισσότερες πληροφορίες θα ακολουθήσουν στην επόμενη ενότητα.
- Τέλος, μέρος της διαδικασίας μπορεί να χρησιμοποιηθεί για να ομαδοποιηθεί ένα σύνολο προγραμμάτων με βάση, για παράδειγμα, των Genre με κάποιο κατάλληλο προκαθορισμένο προφίλ.

Η Μονάδα Συνεργατικής Διήθησης (Collaboration Filtering Module, CFM)

Οι μηχανισμοί ευφυούς διήθησης δεν περιορίζονται στην βασισμένη στο περιεχόμενο διήθηση που περιγράφηκε παραπάνω, αλλά χρησιμοποιούν και την συνεργατική διήθηση. Αυτή η υβριδική προσέγγιση (βλέπε [11],[12],[13]) για περιγραφή τέτοιων υβριδικών συστημάτων) αναμένεται να δίνει καλύτερες προτάσεις για τους χρήστες και να υπερβεί διάφορα προβλήματα που σχετίζονται με μη-υβριδικές προσεγγίσεις (δηλ. μεμονωμένη διήθηση είτε βασισμένη στο περιεχόμενο είτε συνεργατική). Οι μηχανισμοί συνεργατικής διήθησης χειρίζονται την πληροφορία του προφίλ του χρήστη και την συνδυάζουν με πληροφορία ερχόμενη από άλλους χρήστες με παρόμοια ενδιαφέροντα και προτιμήσεις).

Τα παραδοσιακά συστήματα διήθησης περιεχομένου βρίσκουν ένα σύνολο από παρόμοιους χρήστες (“φίλοι”) βασισμένα σε βαθμολογήσεις χρηστών πάνω σε κοινά αντικείμενα πληροφορίας. Το σύστημα χειρίζεται την πληροφορία για τις ενέργειες των φίλων (π.χ. αξιολογήσεις που κάνανε για αντικείμενα πληροφορίας) με σκοπό να προτείνει περισσότερα αντικείμενα πληροφορίας στον συγκεκριμένο χρήστη. Αυτή η προσέγγιση έχει το πρόβλημα της “κρύας εκκίνησης” (cold start): όταν δεν υπάρχουν ικανός αριθμός ενεργειών στο σύστημα από τους φίλους, το σύστημα δεν μπορεί να δώσει αξιόπιστες προτάσεις για αρκετά αντικείμενα πληροφορίας.

Η προσέγγιση που ακολουθείται εδώ υιοθετεί την βασική ιδέα της συνεργατικής διήθησης, δηλ. προτείνει αντικείμενα πληροφορίας (δηλ. Προγράμματα ή τμήματα αυτών) που παρόμοιοι χρήστες έχουν δει και αξιολογήσει ως ενδιαφέροντα. Σε αντίθεση όμως με την παραδοσιακή συνεργατική διήθηση όμοιοι χρήστες βρίσκονται με σύγκριση και των προφίλ τους αντί μόνο των αξιολογήσεων σε κοινά προγράμματα. Κατά αυτόν τον τρόπο ο βαθμός ομοιότητας δύο χρηστών A και B μπορεί να είναι και η ομοιότητα των προφίλ τους P_A και P_B δηλ. $\text{Sim}(A, B) = \text{Sim}(P_A, P_B)$.

Αυτή η ιδιαίτερη μεθοδολογία δίνει λύση όχι μόνο στο πρόβλημα “κρύας εκκίνησης” αλλά και σε άλλα γνωστά προβλήματα της παραδοσιακής συνεργατικής και της βασισμένης στο περιεχόμενο διήθησης όπως αυτό της “υπερ-εξειδίκευσης”, κ.λπ. Σε αντίθεση με την

συνεργατική διήθηση δεν απαιτεί από τους χρήστες να έχουν αξιολογήσει ακριβώς τα ίδια αντικείμενα (exact matching) αλλά παρόμοια. Επιπρόσθετα, αυτή η διαδικασία είναι ανοιχτή και επεκτάσιμη διότι δεν βασίζεται εξολοκλήρου σε ρητή ανάδραση (feedback) αλλά χρησιμοποιεί διαφόρων ειδών έμμεσων αναδράσεων (π.χ. τύποι ενεργειών) παρέχοντας την ευκαιρία να οριστούν διαφορετικές κλίμακες ικανοποίησης και διαφορετικοί τρόποι μετάφρασης της ανάδρασης.

Η υιοθετημένη μεθοδολογία εστιάζει στις προτάσεις λαμβάνοντας υπ' όψιν τις προτιμήσεις των χρηστών που σχετίζονται με το περιεχόμενο των προγραμμάτων. Είναι ξεκάθαρο ότι οι υποδείξεις του συστήματος δύνανται να ικανοποιούν και ένα σύνολο προτιμήσεων που εκφράζονται με τις προτιμήσεις διήθησης και αναζήτησης του χρήστη. Για παράδειγμα, ας θεωρήσουμε ότι το σύστημα έχει συνάγει (από το ιστορικό χρήσης) ότι ο χρήστης αρέσκεται σε ιστορικές ταινίες. Ο χρήστης έχει, επίσης, δηλώσει ότι προτιμά προγράμματα που παρήχθησαν τα 3 τελευταία και που προσφέρονται σε υψηλής ποιότητας βίντεο. Το σύστημα πρώτα θα βρει παρόμοιους χρήστες βασισμένο στα προφίλ. Στη συνέχεια θα βρει τα προγράμματα που αυτοί οι χρήστες βρήκαν ενδιαφέροντα και παράλληλα έχουν κυκλοφορήσει τα 3 τελευταία χρόνια και διατίθενται σε υψηλή ποιότητα. Τελικά, θα προτείνει τα προγράμματα που έχουν αξιολογηθεί ως πολύ ενδιαφέροντα.

Η Μονάδα Εύρεσης Παρόμοιων Καταναλωτών (Find Similar Consumers Module, FSCM)

Αυτή η μονάδα είναι υπεύθυνη για τον υπολογισμό του βαθμού ομοιότητας μεταξύ των χρηστών. Τρέχει κυρίως σε έναν διακομιστή UP-TV όπου πληροφορία που αφορά υποικρυπτόμενες και ρητές βαθμολογήσεις και/ή προτιμήσεις διήθησης και αναζήτησης διαφορετικών χρηστών.

Χρησιμοποιώντας Προτιμήσεις Περιεχομένου για την Εύρεση Παρόμοιων Χρηστών

Έχοντας τον πίνακα των προτιμήσεων περιεχομένου, ο οποίος συνδέει ένα σύνολο χρηστών με ένα σύνολο προγραμμάτων, κάποιος μπορεί να χρησιμοποιήσει διάφορους τύπους για υπολογίσει τους βαθμούς ομοιότητας μεταξύ των χρηστών. Ένα από τους πλέον διαδεδομένους τρόπους να γίνει το παραπάνω είναι με τον συντελεστή του Pearson ο τύπος του οποίου δίδεται παρακάτω:

$$s_{u,v} = \frac{\sum_{i \in A} (R_{u,i} - \bar{R}_u) \cdot (R_{v,i} - \bar{R}_v)}{\sqrt{\sum_{i \in A} (R_{u,i} - \bar{R}_u)^2} \cdot \sqrt{\sum_{i \in A} (R_{v,i} - \bar{R}_v)^2}}, \quad A = \left\{ \begin{array}{l} \text{Σύνολο αντικείμενων που έχουν} \\ \text{βαθμολογηθεί και από τους δύο} \\ \text{χρήστες } u \text{ και } v \end{array} \right\}$$

Αυτός ο τύπος υπολογίζει την ομοιότητα $s_{u,v}$ κάθε ζεύγους χρηστών u και v .

Χρησιμοποιείται, συμπληρωματικά, και ένας εναλλακτικός τύπος, ο οποίος είναι απλούστερος και χρησιμοποιεί τις διαφορές μεταξύ των βαθμολογιών των χρηστών u και v για να υπολογίσει την απόσταση (ή ανομοιότητα) μεταξύ τους και μετά αφαιρεί από το 1 την ανομοιότητα και έχει την ομοιότητα μεταξύ των u και v :

$$s_{u,v} = \frac{\sum_{i \in A} |R_{u,i} - R_{v,i}|}{\#(A) \cdot \max\{|R_{u,i} - R_{v,i}|\}}, \quad A = \left\{ \begin{array}{l} \text{Σύνολο αντικειμένων} \\ \text{που έχουν βαθμολογηθεί} \\ \text{και από τους δύο} \\ \text{χρήστες } u \text{ και } v \end{array} \right\}$$

Παρατηρείστε ότι το $\#(A)$ υποδηλώνει την πληθυκότητα του συνόλου A , δηλ. τον αριθμό των στοιχείων του. Ο τελευταίος τύπος, έκτος από την απλότητα του συγκρινόμενος με τον συντελεστή Pearson, δίνει μόνο θετικούς βαθμούς ομοιότητας ενώ ο συντελεστής του Pearson δίνετε να παραγάγει και αρνητικούς. Αυτό καταδεικνύεται καλύτερα με το παρακάτω παράδειγμα.

Έστω ένα σύνολο πέντε προγραμμάτων και δύο χρήστες u και v που έχουν δώσει τις βαθμολογίες του παρακάτω πίνακα:

Program ID	Ratings of user u	Ratings of user v
1	-	20
2	20	20
3	10	30
4	30	10
5	20	-

Πίνακας 24: Παράδειγμα βαθμολογιών δύο χρηστών για πέντε προγράμματα.

Με τον συντελεστή του Pearson ο βαθμός ομοιότητάς τους είναι -1 , ενώ με τον εναλλακτικό τύπο είναι 0 .

Χρησιμοποιώντας τις Προτιμήσεις Διήθησης και Αναζήτησης για την Εύρεση Παρόμοιων Χρηστών

Όπως προαναφέρθηκε, το σύστημα εκμεταλλεύεται τις “Προτιμήσεις Διήθησης και Αναζήτησης” (Filtering and Search Preferences ή FASP) των χρηστών με στόχο την εύρεση του βαθμού ομοιότητας των παραπάνω. Αυτό είναι εξαιρετικά χρήσιμο αν θέλουμε να επιληφθούμε το πρόβλημα της κρύας εκκίνησης που εγείρεται όταν δεν υπάρχουν τόσο πολλά προγράμματα που να έχουν ειδη βαθμολογηθεί από τους χρήστες.

Μία σαφής προσέγγιση για την εύρεση της ομοιότητας μεταξύ ενός χρήστη u και ενός άλλου με τα FASP τους είναι να συγκριθούν για την ομοιότητά τους οι δυαδικές εκφράσεις των δομών FASP. Αυτό θα μπορούσε να είναι εφικτό στην περίπτωση απλών δομών FASP, αλλά θα ήταν πολύ δύσκολο και στερούμενο πρακτικότητας όταν οι δομές FASP είναι περίπλοκες, εξαιτίας της χρονικής πολυπλοκότητας. (Παρατηρήστε ότι το πρόβλημα της εύρεσης αν δύο αυθαίρετες δυαδικές εκφράσεις είναι ισοδύναμες είναι NP-Complete).

Μια άλλη προσέγγιση, είναι να χρησιμοποιηθεί ένα σταθερό σύνολο αντικειμένων (δηλ. Προγραμμάτων) I , και με την χρήση των FASP των χρηστών να βρεθεί η σχετικότητά τους ως προς τους χρήστες. Αυτές οι χρήστη προς πρόγραμμα προτιμήσεις μπορούν να

χρησιμοποιηθούν για να υπολογιστούν οι ομοιότητες μεταξύ των χρηστών. Στην περίπτωση που υπάρχουν και υποκρυπτόμενες και ρητές προτιμήσεις περιεχομένου, αυτές οι προτιμήσεις μπορούν να συνδυαστούν με τις προκύψασες από την βασισμένη στο περιεχόμενο διήθηση όπου οι πρώτες λαμβάνονται με υψηλότερη εμπιστοσύνη. Αυτή η παραλλαγή μεταξύ πραγματικών και προτεινόμενων προτιμήσεων περιεχομένου υλοποιείται με την υποβάθμιση των βασισμένων στο περιεχόμενο προβλέψεων (δηλ. με τον πολλαπλασιασμό τους με μία σταθερά $0 < c < 1$) πριν χρησιμοποιηθούν και με το να ληφθούν υπ' όψιν όσες προβλέψεις αναφέρονται σε αντικείμενα που έχουν είδη αξιολογηθεί από τον χρήστη.

Η Μονάδα Συνεργατικών Προτάσεων (The Collaborative Suggestions Module, FSCM)

Όταν είναι διαθέσιμοι οι βαθμοί ομοιότητας των χρηστών, τότε μπορούν να γίνουν προτάσεις για το περιεχόμενο για κάθε χρήστη u . Για να γίνει αυτό, το σύστημα χρειάζεται να γνωρίζει το σύνολο των προγραμμάτων για τα οποία θα υπολογιστούν οι συνεργατικές προτάσεις. Αυτό το σύνολο μπορεί να αποτελείται από τα προγράμματα που ανταποκρίνονται στο προφίλ του χρήστη ή από αυτά που οι πιο όμοιοι με τον u χρήστες έχουν δει και (άμεσα ή έμμεσα) έχουν αξιολογήσει. Τότε για κάθε τέτοιο πρόγραμμα i , η κατάταξη που θα προταθεί προκύπτει από την εφαρμογή του παρακάτω τύπου:

$$P_{u,i} = \frac{\sum_{v \in \left\{ \begin{array}{l} \text{k best friends of } u \\ \text{having rated item } i \end{array} \right\}} (s_{u,v} \cdot R_{v,i})}{\sum_{v \in \left\{ \begin{array}{l} \text{k best friends of } u \\ \text{having rated item } i \end{array} \right\}} |s_{u,v}|}$$

Αυτός ο τύπος υπολογίζει ένα στατιστικό μέσο όλων των αξιολογήσεων των k πιο όμοιων χρηστών σε σχέση με τον u που έχουν βαθμολογήσει το συγκεκριμένο πρόγραμμα i . Παρατηρήστε ότι ο παρανομαστής είναι ένα άθροισμα των απολύτων τιμών των βαθμών ομοιότητας μεταξύ των φίλων και του χρήστη u . Η ανάγκη να χρησιμοποιηθούν απόλυτες τιμές δικαιολογείται από το γεγονός ότι ο συντελεστής του Pearson μεταξύ δύο χρηστών μπορεί να είναι και αρνητικός, μία κατάσταση που δεν συμβαίνει με τον εναλλακτικό τύπο που περιγράφηκε στο προηγούμενη υπό-ενότητα.

Ξεπερνώντας Προβλήματα Προστασίας Προσωπικών Δεδομένων Χρηστών

Πρέπει να παρατηρηθεί ότι η μέχρι τούδε ακολουθούμενη προσέγγιση υποθέτει την ύπαρξη κεντρικής βάσης δεδομένων που να περιέχει τις ενέργειες όλων των χρηστών (ή τις υποκρυπτόμενες προτιμήσεις περιεχομένου που εξήχθησαν από τις ενέργειες αυτές) όπως και των προφίλ τους. Αυτό σημαίνει ότι ο χρήστης αποδέχτηκε να διαθέσει τόσο το προφίλ του όσο και το ιστορικό χρήσης του.

Υπάρχει, μολαταύτα, μία ενδιαφέρουσα εναλλακτική που μπορεί να ξεπεράσει το πρόβλημα της εξαγωγής του προφίλ από το PVR του χρήστη. Βασίζεται πάνω στην αυτοαποκαλούμενη βασισμένη στα αντικείμενα συνεργατική διήθηση (Item-based collaboration filtering) όπως περιγράφεται στο [17]. Οι βασισμένες στα αντικείμενα τεχνικές διερευνούν τις σχέσεις μεταξύ των αντικειμένων αντί των σχέσεων μεταξύ χρηστών. Προτάσεις προς τους χρήστες υπολογίζονται με την εύρεση αντικειμένων παρόμοιων με άλλα αντικείμενα που άρεσαν στον χρήστη. Επειδή οι σχέσεις μεταξύ των αντικειμένων είναι σχετικά στατικές, οι βασισμένες στα αντικείμενα τεχνικές μπορεί να είναι σε θέση να δώσουν την ίδια ποιότητα με τις βασισμένες στους χρήστες κάνοντας χρήση λιγότερων υπολογιστικών πόρων.

Μία σύντομη περιγραφή αυτών των τεχνικών είναι η εξής: Όταν το σύστημα θέλει να προβλέψει την προτίμηση ενός συγκεκριμένου χρήστη u (δηλ του ενεργού χρήστη) για ένα αντικείμενο i , πρώτα ελέγχει το σύνολο των αντικειμένων που ήδη ο ενεργός χρήστης έχει βαθμολογήσει (ρητά ή υποκρυπτόμενα). Το σύστημα συνεχίζει με τον υπολογισμό της ομοιότητας αυτών των αντικειμένων με το αντικείμενο i . Μετά, επιλέγει τα k πιο όμοια αντικείμενα $\{i_1, i_2, \dots, i_k\}$ μαζί με τις αντίστοιχες τους ομοιότητες $\{s_{i,i1}, s_{i,i2}, \dots, s_{i,ik}\}$ προς το αντικείμενο i . Όταν τα πλέον όμοια αντικείμενα βρεθούν, γίνεται η πρόβλεψη παίρνοντας τον στατιστικό μέσο των βαθμολογιών του ενεργού χρήστη για αυτά τα όμοια αντικείμενα.

Η παραπάνω διαδικασία για να λειτουργήσει πρέπει δύο πράγματα να αποσαφηνιστούν: Ο υπολογισμός της ομοιότητας μεταξύ των αντικειμένων καθώς επίσης και της γένεσης προβλέψεων. Η βασική ιδέα στον υπολογισμό των ομοιοτήτων μεταξύ δύο αντικειμένων i και j είναι πρώτα να βρεθούν οι χρήστες που βαθμολόγησαν και τα δύο αυτά αντικείμενα και μετά να εφαρμοστεί μία τεχνική υπολογισμού ομοιοτήτων απαράλλακτη με αυτήν του υπολογισμού

ομοιοτήτων μεταξύ χρηστών, ώστε να υπολογιστεί η ομοιότητα $s_{i,j}$. Ένας πιθανός τύπος για τον υπολογισμό ομοιοτήτων δεν είναι άλλος από μία προσαρμογή του συντελεστή Pearson, η οποία δίνεται παρακάτω:

$$s_{i,j} = \frac{\sum_{u \in U} (R_{u,i} - \bar{R}_u) \cdot (R_{u,j} - \bar{R}_u)}{\sqrt{\sum_{u \in U} (R_{u,i} - \bar{R}_u)^2} \cdot \sqrt{\sum_{u \in U} (R_{u,j} - \bar{R}_u)^2}}$$

Σε αυτόν τον τύπο, $R_{u,i}$ είναι η αξιολόγηση του αντικειμένου i από τον χρήστη u και \bar{R}_u είναι ο μέσος όρος των αξιολογήσεων που έχει κάνει ο χρήστης u . Η πρόβλεψη υπολογίζεται με τον παρακάτω τύπο:

$$P_{u,i} = \frac{\sum_{j \in \{\text{items similar to } i \text{ rated by } u\}} (s_{i,j} \cdot R_{u,j})}{\sum_{j \in \{\text{items similar to } i \text{ rated by } u\}} |s_{i,j}|}$$

Τα πειραματικά αποτελέσματα που δίνονται στο [17] καταδεικνύουν ότι αυτή η μέθοδος συμπεριφέρεται καλύτερα ακόμα και από την κλασική βασισμένη σε χρήστες συνεργατική διήθηση. Παρόλα αυτά, αυτό που την καθιστά αρκετά ενδιαφέρουσα είναι το γεγονός ότι μπορεί να υλοποιηθεί χωρίς την αναγκαιότητα όλοι οι χρήστες να μοιραστούν τις προτιμήσεις περιεχομένων τους σε κάποιο κεντρικό διακομιστή. Παρατηρήστε πρώτα ότι οι ομοιότητες μεταξύ αντικειμένων μπορούν να υπολογιστούν εξολοκλήρου πριν ακόμα χρησιμοποιηθούν έστω και για μία πρόβλεψη. Παρατηρείστε, ακόμα, ότι οι ομοιότητες μεταξύ αντικειμένων αναμένεται να παραμένουν αμετάβλητες, έτσι αν υπολογιστούν από ένα υποσύνολο χρηστών (θα αποκαλούνται *εισηγητές* εφεξής) τότε θα παραμείνουν πάνω κάτω οι ίδιες ασχέτως των επιπλέον χρηστών που ενδεχομένως να χρησιμοποιηθούν για να τις βελτιώσουν. Αυτή η εξέταση πρέπει να έχει δώσει ήδη την ιδέα: Ένας διακομιστής μπορεί να χειρίζεται τις προτιμήσεις περιεχομένου ενός επιλεγμένου συνόλου εισηγητών που προθυμοποιούνται να μοιραστούν αυτήν την πληροφορία για να υπολογίσει τις ομοιότητες μεταξύ αντικειμένων και να τις προωθήσει σε όλους τους χρήστες όπου ο μπορεί να χρησιμοποιηθεί ο τύπος της πρόβλεψης ώστε να γίνουν οι προτάσεις προς τον ενεργό χρήστη. Περιοδικά ο διακομιστής

μπορεί να στέλνει καινούργια πληροφορία για τις ομοιότητες για να αντικατοπτρίσει την ενσωμάτωση νέων αντικείμενων στο περιβάλλον της ψηφιακής τηλεόρασης.

Υλοποίηση

Η υλοποίηση των τεχνικών της συνεργατικής διήθησης (τόσο της βασισμένης στους χρήστες όσο και της βασισμένης στα αντικείμενα) είναι σχετικά ξεκάθαρη και γίνεται πάνω από ένα σύστημα διαχείρισης βάσεων δεδομένων, όπως αυτό της προηγούμενης ενότητας.

Η υλοποίηση ξεκινάει θεωρώντας ότι υπάρχουν οι προτιμήσεις περιεχομένου (αυτές μπορούν να έχουν δοθεί κατευθείαν από τους χρήστες, μπορεί να έχουν υπολογιστεί από το ιστορικό χρήσης, βλέπε [10], ή να υπολογιστούν με χρήση της μονάδας διήθησης βασισμένη στο περιεχόμενο όπως περιγράφηκε παραπάνω). Αυτές οι προτιμήσεις περιεχομένου κρατούνται στην βάση σε έναν πίνακα της μορφής:

$$\text{Program_User}(\mathbf{CRID}, \mathbf{UserID}, \text{Pvalue})$$

Υπολογισμός Ομοιοτήτων με χρήση του συντελεστή Pearson

Ο παραπάνω πίνακας είναι αρκietός για να υπολογιστεί ο συντελεστής του Pearson με το ακόλουθη δήλωση SQL. Το αποτέλεσμα αποθηκεύεται στον πίνακα:

UserCorrelation (UserID1, UserID2, Sim) ώστε να χρησιμοποιηθεί αργότερα για τις προτάσεις.

SQL [15]:

```
INSERT INTO TempMeanRating (UserID, Mean)
SELECT UserID, sum(Pvalue)/count(Pvalue)
FROM Program_User
GROUP BY UserID;

INSERT INTO UserCorrelation (UserID1, UserID2, Sim)
SELECT A.UserID, U.UserID,
        sum((A.Pvalue-Ma.Mean)*(U.Pvalue-Mu.Mean))/
        sqrt(sum(power(A.Pvalue-Ma.Mean,2))*
        sum(power(U.Pvalue-Mu.Mean,2)))
FROM Program_User as A,
        Program_User as U,
        TempMeanRating as Ma,
        TempMeanRating as Mu
WHERE A.CRID=U. CRID
        AND Ma.UserID=A.UserID
        AND Mu.UserID=U.UserID
        AND A.UserID<>U.UserID
GROUP BY A.UserID, U.UserID;
```

Διευκρινίζεται σε αυτό το σημείο ότι με εντελώς όμοιο τρόπο είναι και οι δηλώσεις SQL για τον υπολογισμό ομοιοτήτων μεταξύ προγραμμάτων.

Υπολογισμός Προτάσεων προς τους Χρήστες

Έχοντας υπολογίσει τις ομοιότητες μπορούμε παρακάτω να κάνουμε προτάσεις σε έναν ενεργό χρήστη. Η παρακάτω μεταβλητή χρησιμοποιείται:

@userid αντιστοιχεί στο id του ενεργού χρήστη.

SQL [16]:

```
INSERT INTO TempMeanRating (UserID, Mean)
SELECT UserID, sum(Pvalue)/count(Pvalue)
FROM Program_User
GROUP BY UserID;

SELECT P.CRID,
        round(Ma.Mean+sum((P.Pvalue-Mu.Mean)*UC.Sim)/sum(UC.Sim)) as Rank
FROM ProgramPreference as P,
        UserCorrelation as UC,
        TempMeanRating as Ma,
        TempMeanRating as Mu,
WHERE PU.UserID=@userid
        AND PU.UserID=UC.UserID2
        AND UC.UserID1=Ma.UserID
        AND Ma.UserID=UP.UserID
        AND UC.UserID2=Mu.UserID
GROUP BY UP.CRID
HAVING Rank>0;
```

Όμοια και εδώ γίνονται οι υπολογισμοί αν είχαμε ομοιότητες μεταξύ προγραμμάτων.

Περίληψη

Σε αυτό το κεφάλαιο μελετήθηκαν μηχανισμοί ευφυούς διήθησης TV-Anytime περιεχομένου προγραμμάτων. Συγκεκριμένα αναπτύχθηκε ένας μηχανισμός που αναλαμβάνει να βρει το βαθμό ομοιότητας του περιεχομένου ενός προγράμματος σε σχέση με το προφίλ του χρήστη. Υποστηρίχθηκε πλήρως το σχήμα μεταδεδομένων του TV-Anytime ώστε οι ερωτήσεις ανάκτησης και διήθησης που μπορεί να απαντήσει το σύστημα να περιλαμβάνουν ένα ευρύ σύνολο από χαρακτηριστικά αξιοποιώντας έτσι όλη την δυνατή πληροφορία που υπάρχει για τα προγράμματα, είτε αυτή προέρχεται από κάποια TV-Anytime εφαρμογή είτε ακόμα και από το διαδίκτυο. Μέχρι τώρα ο χρήστης ήταν σε θέση να κάνει ερωτήσεις στο σύστημα ή να έχει ένα προφίλ με ελάχιστα χαρακτηριστικά (π.χ. μόνο Genres και Keywords) αφήνοντας ουσιαστικά ανεκμεταλλευτή την τεράστια πληροφορία που υπάρχει για το οπτικοακουστικό υλικό. Ειδικά, σχεδιάστηκε ένα μεταφραστικό μοντέλο του σχήματος μεταδεδομένων του

TV-Anytime για τις προτιμήσεις των χρηστών σε ασαφείς δυαδικές εκφράσεις ώστε από ένα αφηρημένο χώρο προτιμήσεων να υπάρχει μία πλήρως καθορισμένη συντακτικά και σημασιολογικά ερώτηση. Επίσης, για να κατανοεί ο χρήστης καλύτερα την λειτουργία του συστήματος ώστε να μπορεί να φτιάχνει καλύτερα το προφίλ του διατηρείται πληροφορία με τι χαρακτηριστικά του προφίλ του ταιριάζουν τα μεταδεδομένα κάθε προγράμματος. Με αυτόν τον τρόπο, καλλιεργείται εμπιστοσύνη από τον χρήστη ως προς το σύστημα, αφού διαπιστώνει, ότι το σύστημα δεν συμπεριφέρεται αυθαίρετα αλλά σύμφωνα με τις προτιμήσεις του.

Επιπλέον, κατασκευάσθηκε μηχανισμός συνεργατικής διήθησης ο οποίος διαχειρίζεται πληροφορίες για τις αρέσκειες των χρηστών για να εκμαιεύσει ομοιότητες και να κάνει προτάσεις προγραμμάτων προς τους χρήστες. Ο μηχανισμός που υλοποιήθηκε δεν παρέμεινε στα στενά όρια της συνηθισμένης συνεργατικής διήθησης αλλά, χωρίς να μειώσει την απόδοσή του, ξεπέρασε το πρόβλημα τόσο του κρύου ξεκινήματος, τροφοδοτώντας με δεδομένα που προέρχονται από τα προφίλ των χρηστών, όσο και των προσωπικών δεδομένων των χρηστών, με υπολογισμό των απαραίτητων δεδομένων από ένα κατά πολύ μικρότερο υποσύνολο χρηστών, απαγκιστρώνοντάς τους έτσι από την υποχρεωτικό διαμοιρασμό των δεδομένων τους σε ένα κεντρικό εξυπηρετητή.

*Ου το ζην περί πλείστον
ποιητέον, αλλά το εν ζην.*

Πλάτωνας

Κεφάλαιο V

ΓΡΑΦΙΚΟ ΠΕΡΙΒΑΛΛΟΝ ΔΙΑΧΕΙΡΙΣΗΣ ΠΡΟΦΙΛ ΧΡΗΣΤΗ

Εισαγωγή

Σε περιβάλλον ψηφιακής τηλεόρασης είναι μείζονος σημασίας η δυνατότητα των χρηστών να μπορούν να δηλώνουν μία σειρά από προτιμήσεις ώστε να εξυπηρετούνται σύμφωνα με αυτές. Αυτή η σειρά από τις προτιμήσεις συνιστούν το προφίλ του χρήστη.

Είναι λοιπόν εύλογο σε ένα περιβάλλον ψηφιακής τηλεόρασης το σύστημα να προσπαθεί να προβλέψει το προφίλ των χρηστών, αν δεν έχει δηλωθεί, ή να το τροποποιεί κατάλληλα ανάλογα με τις καινούργιες προτιμήσεις του. Εναλλακτικά, πρέπει να παρέχεται η δυνατότητα στους χρήστες να μπορούν να επεξεργαστούν το προφίλ τους μόνοι τους. Καθοριστική, λοιπόν, καθίσταται η κατασκευή διεπαφών (interfaces) ώστε να παρουσιάζεται το προφίλ με έναν απλό και κατανοητό για τους χρήστες τρόπο. Η υλοποίηση της διεπαφής καθορίζεται από ένα σύνολο παραγόντων όπως:

- Το επίπεδο εξοικείωσης των χρηστών αφενός με το σύστημα και αφετέρου με τις έννοιες της πληροφορικής (π.χ. όροι όπως OR και AND για ερωτήσεις - queries – προς το σύστημα είναι δύσκολο να κατανοηθούν από αδαείς χρήστες).
- Τη φύση και την πολυπλοκότητα της δομής των πληροφοριών που πρέπει να αντληθούν από τους χρήστες. Αν ζητηθεί από τους χρήστες, επί παραδείγματι, να

ενημερώσουν ένα δέντρο με αρκετά κλαδιά χωρίς να κατανοούν, ενδεχομένως, την σημασιολογική υπόσταση της κάθε διακλάδωσης, το αποτέλεσμα σίγουρα δεν θα είναι το επιθυμητό.

Οπότε εύκολα συμπεραίνουμε ότι το προκύπτον UI πρέπει, για να απευθύνετε σε αδαείς χρήστες, να αποκρύπτει την ενδεχόμενη πολυπλοκότητα της δομής παρέχοντας ταυτόχρονα, αν όχι το σύνολο, ένα μεγάλο μέρος των επιλογών που δύναται να έχει ο χρήστης. Ταυτόχρονα θα πρέπει να διακρίνεται η λειτουργία της απλής και η λειτουργία της σύνθετης κατασκευής δομής προτιμήσεων έτσι ώστε οι πολυπληθείς αδαείς χρήστες να μπορούν να χρησιμοποιήσουν τον απλούστερο μηχανισμό ενώ οι πιο έμπειροι να μπορούν να αξιοποιήσουν τις επαυξημένες δυνατότητες του συστήματος. Επίσης, αυτή η προσέγγιση βοηθά τον αδαή χρήστη να εξοικειωθεί γρηγορότερα και να μεταβεί στη σύνθετη λειτουργικότητα σταδιακά καθώς θα αρχίσει σιγά σιγά να κατανοεί τις λεπτομέρειες του συστήματος.

Στην συγκεκριμένη μελέτη θα εστιάσουμε στην κατασκευή UI για τον προσδιορισμό, από τους χρήστες, προφίλ με τα κριτήρια επιλογής προγραμμάτων που τους ενδιαφέρει να παρακολουθούν στην τηλεόραση. Αρχικά, θα περιγράψουμε το μοντέλο που χρησιμοποιείτε για την δομή των κριτηρίων (είναι το τμήμα Filtering And Search Preferences του TV-Anytime). Στη συνέχεια, θα παρατηρήσουμε τα προβλήματα χρήσης και κατανόησής του από τους απλούς χρήστες και τέλος, θα προτείνουμε κάποια μέτρα για την λύση του προβλήματος.

Γραφικός Σχεδιασμός της Εφαρμογής

Η εφαρμογή υλοποιήθηκε εξολοκλήρου σε Java κυρίως επειδή είναι διαπλατφορμική. Αποφασίσθηκε να υλοποιηθεί ως εφαρμογή γραφείου αντί για εφαρμογή δικτύου (Web Application). Παρόλα αυτά η εφαρμογή είναι δυνατόν να τρέξει και ως applet δίνοντας έτσι την δυνατότητα σε όλους τους χρήστες του διαδικτύου να την εκμεταλλευτούν.

Βασικές Αρχές Σχεδιασμού (User Interface Guidelines)

Κατά το σχεδιασμό του user interface της εφαρμογής έγινε προσπάθεια να εφαρμοστούν οι παρακάτω βασικές αρχές σχεδιασμού interface :

- **Ορατότητα της Κατάστασης του Συστήματος (Visibility of System Status):** Το interface που υλοποιήθηκε ενημερώνει διαρκώς τον χρήστη σχετικά με το που βρίσκεται και τι κάνει. Αυτό επιτυγχάνεται με χρήση του status bar αλλά και με ορθών τίτλων στους διαλόγους εφαρμογής.
- **Αντιστοίχιση μεταξύ Συστήματος και Πραγματικού Κόσμου (Match Between System and the Real World):** Κατά το σχεδιασμό του User Interface ελήφθην υπ' όψιν το κοινό στο οποίο αυτό απευθύνεται και έτσι χρησιμοποιήθηκαν όροι οικείοι και κατανοητοί για το σύνολο των ενδιαφερομένων.
- **Έλεγχος και Ελευθερία Χρήστη (User Control and Freedom):** Ο χρήστης στις περισσότερες περιπτώσεις έχει την ελευθερία να ενεργήσει με τον τρόπο και τη σειρά που θέλει ο ίδιος, χωρίς να είναι δεσμευμένος να ακολουθήσει μια ακριβή και συγκεκριμένη σειρά ενεργειών. Μειώθηκαν οι διάλογοι που ενημερώνουν τον χρήστη μόνο στους απαραίτητους, δηλαδή, σε εκείνους όπου εγκυμονεί κίνδυνος να απολεσθούν δεδομένα από κατά λάθος ενέργειες.
- **Συνέπεια και Πρότυπα (Consistency and Standards):** Σε όλη την εφαρμογή ακολουθείται η ίδια λογική σχεδιασμού (π.χ. ίδια χρώματα, κοινή ονομασία κουμπιών που εκτελούν την ίδια ενέργεια). Επίσης έγινε προσπάθεια ώστε η εφαρμογή να ακολουθεί τα πρότυπα άλλων ευρέως χρησιμοποιούμενων εφαρμογών ώστε να είναι όσο το δυνατόν πιο οικεία στους χρήστες.
- **Αναγνώριση παρά Μνήμη (Recognition rather than Recall):** Ο χρήστης δεν είναι υποχρεωμένος να θυμάται πληροφορία μεταξύ των ενεργειών που κάνει. Με αυτή την καθοδήγηση κατά νου, σε κάθε σημείο στην εφαρμογή υπάρχουν επαρκείς επεξηγήσεις ώστε ο χρήστης να καθοδηγείται στο πως θα κάνει αυτά που θέλει να

κάνει. Οι επεξηγήσεις είναι σύντομα tool tips ή ακόμα και σύντομα labels που επεξηγούν την διαδικασία που μπορεί να ακολουθηθεί.

- **Αποτροπή Λαθών (Error Prevention):** Ο χρήστης εμποδίζεται από το να κάνει λάθη με το να του προσφέρονται όλες οι δυνατές επιλογές όπου χρειάζεται ώστε τότε να μην ενεργεί αυθαίρετα, στο οποίο κατά κύριο λόγο οφείλεται η πρόκληση λαθών.
- **Ευελιξία και Αποδοτικότητα Χρήσης (Flexibility and Efficiency of Use):** Το σύστημα απευθύνεται τόσο σε έμπειρους χρήστες όσο και σε λιγότερο έμπειρους ή ακόμα και αρχάριους επιτρέποντας τη ταυτόχρονη εξυπηρέτηση όλων των χρηστών, δίχως η εξυπηρέτηση των μεν να επιβαρύνει τους δε. Έχει γίνει αρκετή προσπάθεια να έχει η εφαρμογή καλή απόκριση. Οι όποιες καθυστερήσεις συμβαίνουν οφείλονται κατά κύριο λόγο στην τεράστια ποσότητα πληροφορίας που πρέπει να φορτωθεί από την Βάση. Για την καλύτερη απόδοση κατά την χρήση της εφαρμογής η πληροφορία, αυτή, φορτώνεται κατά την εκκίνηση της εφαρμογής και με χρήση πολλών threads ώστε να παραλληλοποιηθεί και να επιταχυνθεί κατά το δυνατόν η χρονοβόρα αυτή διαδικασία. Επίσης, μετατοπίστηκαν και όλες οι άλλες ενέργειες, που μπορούν να εκτελεστούν κατά την εκκίνηση, στην αρχή.
- **Αισθητική και Μινιμαλιστική Σχεδίαση (Aesthetic and Minimalistic Design):** Το φαίνεσθαι (Look & Feel) της εφαρμογής είναι το ίδιο πάντα το ίδιο με αυτό του λειτουργικού συστήματος, ώστε να είναι πιο κοντά στις αισθητικές προτιμήσεις του χρήστη. Εκτός αυτού, στο χρήστη δεν παρέχονται περιττές πληροφορίες και οι επεξηγήσεις παραμένουν σύντομες, διότι αυτό θα εμποδίσει τον χρήστη να εστιάσει την προσοχή του στην πληροφορία. Ακόμη και στην περίπτωση των λαθών, η επισήμανση αυτών γίνεται με τρόπο σύντομο ώστε να φαίνεται ξεκάθαρα το μήνυμα που θέλουν να δώσουν στο χρήστη.
- **Βοήθεια Χρηστών στην Αναγνώριση, Διάγνωση και Ανάνηψη των Λαθών (Help Users Recognize, Diagnose, and Recover from Errors):** Τα μηνύματα λάθους είναι λακωνικά, εύκολα κατανοητά και απόλυτα ακριβή έτσι ώστε να τα καταλαβαίνει αμέσως ο χρήστης και να μπορεί να διορθώσει εύκολα τα λάθη του.

Αναπαράσταση Filtering and Search Preferences με Query by Example

Στο τρίτο κεφάλαιο μελετήθηκε το Query by Example ή QBE. Είναι τώρα ενδιαφέρον να εξετάσουμε πώς μπορεί αυτή η τόσο απλή και κατανοητή μεθοδολογία να αξιοποιηθεί κατάλληλα. Τα Filtering and Search Preferences ή FASP έχουν μια αριετά περίπλοκη δομή έστω κι αν δεν εξεταστεί καθόλου η ιεραρχία των FASP, δηλαδή σαν να ήταν το FASP ένα στοιχείο που δεν είχε παιδιά. Αυτό είναι ενδιαφέρον διότι η ιεραρχία θα μπορούσε να παρουσιαστεί σαν μία δενδρική δομή, όπως θα δείχτεί στην επόμενη παράγραφο, όπου χρειάζεται μία μέθοδος που να παρουσιάζει την πληροφορία ενός FASP απαλλαγμένου από την ιεραρχία. Γεννήθηκε έτσι η ιδέα να παρουσιαστεί το FASP με QBE.

Θα εξεταστούν, τώρα, τρόποι για να αναπαρασταθεί με την σημασιολογία του QBE η πληροφορία που διαθέτει ένα FASP. Για να γίνει αυτό θα χρησιμοποιηθεί το παράδειγμα της σελίδας 37. Καταρχάς αποφασίστηκε τα Classification, Creation και Source Preferences να εμφανίζονται σε τρεις διαφορετικούς πίνακες. Για τον κάθε τύπο ξεχωριστά έχουμε:

Creator	Keyword
<i>Ronaldo</i>	<i>Goal</i>

Πίνακας 25: Ο πίνακας για τα Classification Preferences

Country	Lang	Genre
<i>Argentina</i>		
<i>Brazil</i>		
	<i>En</i>	<i>Football</i>

Πίνακας 26: Ο πίνακας για τα Creation Preferences

Dissemination Source
<i>BBC</i>
NOT <i>KBS1</i>

Πίνακας 27: Ο πίνακας για τα Source Preferences

Είναι πλέον κατανοητό το πώς μπορούμε να χρησιμοποιήσουμε το QBE για να κατασκευάσουμε ένα απλό interface για μεμονωμένα FASP.

Δενδρική Αναπαράσταση της Ιεραρχίας των Φίλτρων

Η ιεραρχία των φίλτρων έχει δενδρική δομή, όπως έχει αναφερθεί πολλές φορές. Ως εκ τούτου, είναι ξεκάθαρο ότι η ενδεικνυόμενη αναπαράστασή της είναι πάλι με δενδρική δομή. Το δέντρο είναι μία αρκετά κατανοητή δομή για όλους τους χρήστες, οφειλόμενη κυρίως στην εξοικείωσή τους με τους διαχειριστές αρχείων (File Managers), όπως π.χ. ο MS Windows Explorer.

Λειτουργικότητα

Σε αυτήν την παράγραφο θα αναλυθεί λεπτομερώς η λειτουργικότητα που υποστηρίζεται από την εφαρμογή. Για τον σκοπό αυτό θα χρησιμοποιηθούν τόσο διαγράμματα όσο και Use Cases.

Γενικά

Το αντικείμενο της εφαρμογής που υλοποιήθηκε είναι η δημιουργία και η επεξεργασία προφίλ χρηστών.

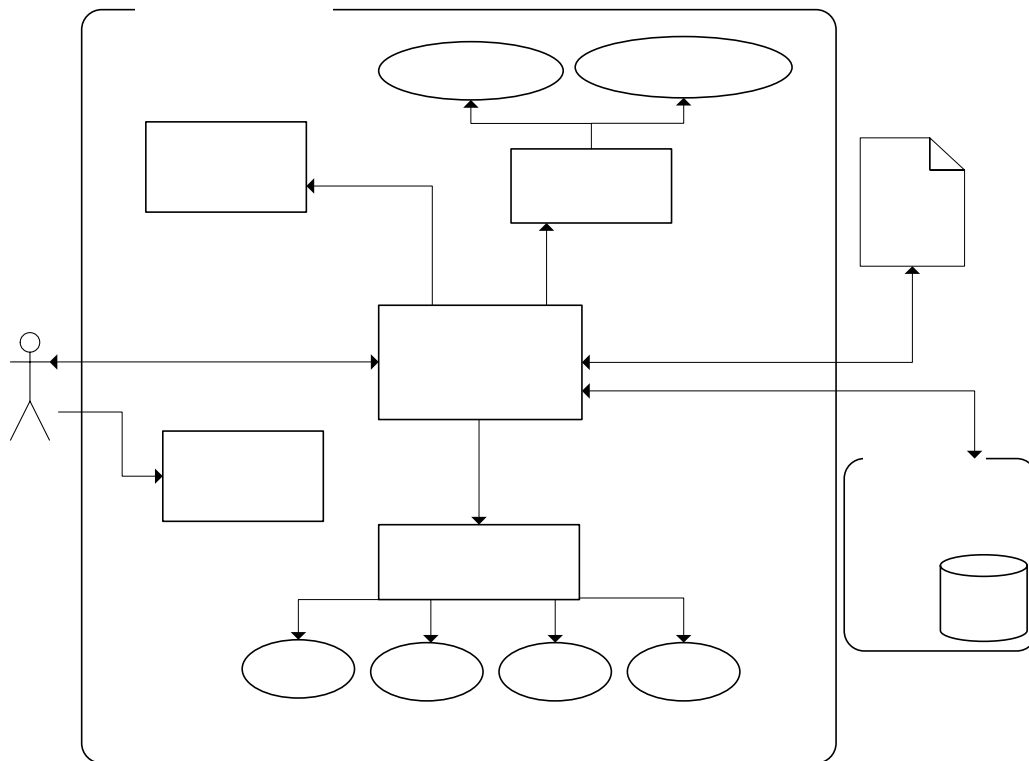
Πιο αναλυτικά, αρχικά ο χρήστης μπορεί είτε να φορτώσει προφίλ που είναι ήδη αποθηκευμένο στη Βάση Δεδομένων ή σε ένα XML έγγραφο, είτε να δημιουργήσει ένα νέο προφίλ. Το νέο αυτό προφίλ που θα δημιουργηθεί είτε θα είναι άδειο είτε μπορεί να κατασκευαστεί αυτόματα με Wizard και ο χρήστης να το τροποποιήσει στη συνέχεια.

Έτσι, αφού ο χρήστης έχει ένα προφίλ ανοιχτό, μπορεί να δημιουργεί, να προσθέτει, να επεξεργάζεται ή και να σβήνει πληροφορία που βρίσκεται στο προφίλ του.

Κατόπιν, αφού τελειώσει η προσθαφαίρεση της πληροφορίας, μπορεί να γίνει αποθήκευση ξανά στην Βάση ή σε ένα XML έγγραφο.

Επίσης, μπορεί να βλέπει τα προγράμματα που του προτείνει το σύστημα βασισμένο στο προφίλ του και τα χαρακτηριστικά του προφίλ του εξαιτίας των οποίων το κάθε πρόγραμμα προτείνεται.

Η όλη λειτουργικότητα του εργαλείου παρουσιάζεται γραφικά στο παρακάτω διάγραμμα.



Σχήμα 11: Διάγραμμα παρεχόμενης λειτουργικότητας από το γραφικό περιβάλλον.

Παρακάτω, παρουσιάζονται τα βασικά Use Cases της εφαρμογής, τα οποία οριοθετούν την λειτουργικότητά της με σαφή και ολοκληρωμένο τρόπο.

Use Cases

Τα Use Cases είναι ένας εύκολος και συνοπτικός τρόπος παρουσίασης της λειτουργικότητας της εφαρμογής. Με τα Use Cases μπορούν να παρουσιαστούν όλα τα βήματα μίας

Graphical User Interface

διαδικασίας τόσο αν εκτελεστεί σωστά όσο και στις περιπτώσεις που κάτι δεν εξελιχτεί όπως αναμενόταν.

Use Case 1: Επεξεργασία προφίλ χρήστη

USE CASE 1	Επεξεργασία προφίλ χρήστη	
Goal In Context	Ο χρήστης θέλει να δημιουργήσει ή να τροποποιήσει το προφίλ του	
Scope & Level	Σύστημα , Primary Task	
Precondition		
Success End Condition	Ο χρήστης αποθηκεύει το προφίλ του στην Βάση ή σε XML έγγραφο	
Failed End Condition	Το προφίλ δεν αποθηκεύεται (ο χρήστης δεν θέλει ή system crash)	
Primary, Secondary Actor	Χρήστης	
Trigger	Ο χρήστης ξεκινάει την εφαρμογή	
Steps	1	Ανάκτηση από την Βάση Δεδομένων του επιλεγέντος από τον χρήστη προφίλ με βάση το όνομά του (User Name).
	2	Επεξεργασία της ιεραρχίας των φίλτρων (FASP) (USE CASE 2)
	3	Επεξεργασία των χαρακτηριστικών κάθε φίλτρου (FASP) (USE CASE 3)
	4	Αποθήκευση του προφίλ στην Βάση Δεδομένων
Extensions	1α	Η σύνδεση με την Βάση δεν είναι δυνατή. 1α1. Ο χρήστης ενημερώνεται και καλείται να δώσει νέα στοιχεία προς αναζήτηση
	1β	Δεν υπάρχει το επιλεγθέν προφίλ στη Βάση. Ο χρήστης ενημερώνεται 1β1. επαναλαμβάνει το βήμα 1 1β2. δημιουργεί ένα νέο προφίλ
Sub-Variations	1α	Ανάκτηση προφίλ από ένα έγγραφο XML 1α1. Η πληροφορία μέσα στο XML έγγραφο δεν περιέχει έγκυρο προφίλ. 1α1α. Ο χρήστης ενημερώνεται.
	1β	Δημιουργία νέο προφίλ
	1γ	Δημιουργία νέου προφίλ με χρήση wizard (USE CASE 4)
	2α	Ο χρήστης επιλέγει να μην επεξεργαστεί την ιεραρχία των φίλτρων
	3α	Ο χρήστης επιλέγει να μην επεξεργαστεί τα χαρακτηριστικά κάποιων ή όλων των φίλτρων

	4α	Ο χρήστης επιλέγει να μην αποθηκεύσει τα μεταδεδομένα του τρέχοντος προφίλ. Επιλέγει έξοδο από το πρόγραμμα ή εκκίνηση νέου project
	5α1.	Το πρόγραμμα ενημερώνει τον χρήστη
	4β	Ο χρήστης επιλέγει να αποθηκεύσει τα μεταδεδομένα του τρέχοντος προφίλ σε έγγραφο XML.

Πίνακας 28: Use Case 1: Επεξεργασία προφίλ χρήστη

Use Case 2: Επεξεργασία ιεραρχίας φίλτρου.

USE CASE 2	Επεξεργασία ιεραρχίας φίλτρων	
Goal In Context	Ο χρήστης θέλει να δημιουργήσει ή να τροποποιήσει την ιεραρχία των φίλτρων του προφίλ του	
Scope & Level	Σύστημα , sub-function	
Precondition	Να έχει ξεκινήσει η εφαρμογή	
Success End Condition	Ο χρήστης αποθηκεύει την ιεραρχία στο προφίλ	
Failed End Condition	Η ιεραρχία δεν αποθηκεύεται (ο χρήστης δεν θέλει)	
Primary, Secondary Actor	Χρήστης	
Trigger		
Steps	1	Φόρτωση της ιεραρχίας των φίλτρων από το προφίλ του χρήστη.
	2	Ο χρήστης δημιουργεί ένα ή περισσότερα νέα φίλτρα στην ιεραρχία. (USE CASE 5)
	3	Ο χρήστης αλλάζει το όνομα ενός ή περισσότερων φίλτρων.
	4	Ο χρήστης μετακινεί ένα ή περισσότερα φίλτρα στην ιεραρχία, με χρήση drag & drop.
	5	Ο χρήστης διαγράφει ένα ή περισσότερα φίλτρα.
	6	Η ιεραρχία αποθηκεύεται στο προφίλ.
Extensions	1α	Δεν υπάρχει στο τρέχον προφίλ κανένα φίλτρο. Ο χρήστης μπορεί μόνο να δημιουργήσει νέο.
Sub-Variations	2α	Ο χρήστης δεν θέλει να δημιουργήσει κανένα νέο φίλτρο.
	3α	Ο χρήστης δεν θέλει να αλλάξει το όνομα κανενός φίλτρου.
	4α	Ο χρήστης δεν θέλει να μετακινήσει κανένα φίλτρο.
	5α	Ο χρήστης δεν θέλει να διαγράψει κανένα φίλτρο.
	6α	Ο χρήστης αποφασίζει να μην αποθηκεύσει τις αλλαγές.

Πίνακας 29: Use Case 2: Επεξεργασία ιεραρχίας φίλτρου.

Use Case 3: Επεξεργασία χαρακτηριστικών φίλτρου.

USE CASE 3	Επεξεργασία χαρακτηριστικών φίλτρου	
Goal In Context	Ο χρήστης θέλει να επεξεργαστεί ένα φίλτρο του προφίλ του	
Scope & Level	Σύστημα, Sub-function.	
Precondition	Να έχει ξεκινήσει η εφαρμογή.	
Success End Condition	Ο χρήστης συμπληρώνει την πληροφορία του φίλτρου.	
Failed End Condition	Ο χρήστης δεν θέλει να προσθέσει την πληροφορία του φίλτρου.	
Primary, Secondary Actor	Χρήστης.	
Trigger		
Steps	1	Επιλογή φίλτρου και φόρτωσή του από το προφίλ του χρήστη.
	2	Επιλογή πεδίων ενδιαφέροντος για επεξεργασία.
	3	Ο χρήστης προσθέτει αυτόματα τιμές από τα μεταδεδομένα προγραμμάτων που επιλέγει καθολικά σε ολόκληρο το φίλτρο (Creator, Title, Keyword κ.α.). (USE CASE 6)
	4	Ο χρήστης προσθέτει αυτόματα τιμές από τα μεταδεδομένα προγραμμάτων που επιλέγει μεμονωμένα στα στοιχεία του φίλτρου (Creator, Title, Keyword κ.α.). (USE CASE 6)
	5	Ο χρήστης επεξεργάζεται (προσθέτει, διαγράφει, αλλάζει βάρη) τα πεδία των Preference Condition, Creation, Classification και Source Preferences του φίλτρου. (USE CASE 7)
Extensions	1α	Δεν υπάρχει στο τρέχον προφίλ κανένα φίλτρο. Ο χρήστης μπορεί μόνο να δημιουργήσει νέο. (USE CASE 2)
Sub-Variations	2α	Ο χρήστης δεν θέλει επιλέξει συγκεκριμένα πεδία.
	2α1	Τα προκαθορισμένα πεδία επιλέγονται
	3α	Ο χρήστης δεν θέλει προσθέσει αυτόματα τιμές από προγράμματα καθολικά στο φίλτρο.
	4α	Ο χρήστης δεν θέλει προσθέσει αυτόματα τιμές από προγράμματα σε μεμονωμένα πεδία του φίλτρου.
	5α	Ο χρήστης δεν θέλει να επεξεργαστεί τα πεδία του φίλτρου.

Πίνακας 30: Use Case 3: Επεξεργασία χαρακτηριστικών φίλτρου.

Use Case 4: Δημιουργία νέου προφίλ με χρήση wizard

USE CASE 4	Δημιουργία νέου προφίλ με χρήση wizard	
Goal In Context	Ο χρήστης θέλει να δημιουργήσει ένα νέο προφίλ χρησιμοποιώντας wizard.	
Scope & Level	Σύστημα , Primary Task	
Precondition	Να έχει ξεκινήσει η εφαρμογή	
Success End Condition	Ο χρήστης κατασκευάζει αυτόματα το προφίλ του	
Failed End Condition	Το προφίλ δεν κατασκευάστηκε (ο χρήστης δεν θέλει ή system crash)	
Primary, Secondary Actor	Χρήστης	
Trigger	Ο χρήστης ξεκινάει τον Wizard	
Steps	1	Ο χρήστης επιλέγει ένα όνομα.
	2	Ο χρήστης επιλέγει αν θέλει να κατασκευαστεί επίπεδο προφίλ ή ιεραρχικό με βάση το Genre.
	3	Επιλογή των πεδίων που θα αποτελούν το προφίλ.
	4	Επιλογή των προγραμμάτων των οποίων τα μεταδεδομένα θα χρησιμοποιηθούν για την κατασκευή του προφίλ.
	5	Κατασκευή προφίλ.
Extensions	5	Ο χρήστης ανυρώνει την όλη διαδικασία.

Πίνακας 31: Use Case 4: Δημιουργία νέου προφίλ με χρήση wizard

Use Case 5: Δημιουργία νέου φίλτρου.

USE CASE 5	Δημιουργία νέου φίλτρου	
Goal In Context	Ο χρήστης θέλει να δημιουργήσει ένα νέο φίλτρο στο προφίλ του	
Scope & Level	Σύστημα, Sub-function	
Precondition	Να έχει ξεκινήσει η εφαρμογή	
Success End Condition	Δημιουργείται ένα νέο φίλτρο.	
Failed End Condition	Δεν δημιουργείται ένα νέο φίλτρο. (ο χρήστης δεν θέλει)	
Primary, Secondary Actor	Χρήστης	
Trigger	Ο χρήστης ζητά να δημιουργήσει ένα νέο φίλτρο σε κάποιο σημείο της ιεραρχίας των φίλτρων.	
Steps	1	Ο χρήστης επιλέγει ένα όνομα για το φίλτρο.

	2	Επιλέγονται τα Genres που σχετίζονται με το φίλτρο.
	3	Επιλέγονται οι μέρες της εβδομάδας που θα ισχύει το καινούργιο αυτό φίλτρο.
	4	Κατασκευή νέου φίλτρου.
Extensions	4α	Ο χρήστης ακυρώνει την όλη διαδικασία.
	4β	Το όνομα είναι κενό: Ακυρώνεται η διαδικασία
Sub-Variations	1α	Ο χρήστης δεν θέλει να επιλέξει όνομα.
	1α1	Το όνομα φτιάχνεται αυτόματα από τα βήματα 3, 4.
	2α	Δεν επιλέγονται Genres.
	3α	Δεν επιλέγονται μέρες.

Πίνακας 32: Use Case 5: Δημιουργία νέου φίλτρου.

Use Case 6: Πρόσθεση τιμών μεταδεδομένων προγραμμάτων σε πεδία φίλτρου.

USE CASE 6	Πρόσθεση τιμών μεταδεδομένων προγραμμάτων σε πεδία φίλτρου.	
Goal In Context	Αυτόματη πρόσθεση τιμών που προέρχονται από τα μεταδεδομένα επιλεγμένων προγραμμάτων σε πεδία φίλτρου.	
Scope & Level	Σύστημα, Sub-function.	
Precondition	Να έχει ξεκινήσει η εφαρμογή. Να είναι επιλεγμένο κάποιο φίλτρο.	
Success End Condition	Να προστεθούν στα πεδία του επιλεγμένου φίλτρου οι τιμές των μεταδεδομένων των προγραμμάτων που αφορούν αυτό το χαρακτηριστικό.	
Failed End Condition	Δεν προστίθεται καμία τιμή.	
Primary, Secondary Actor	Χρήστης, Βάση Δεδομένων.	
Trigger		
Steps	1	Επιλογή ενός ή περισσότερων προγραμμάτων.
	2	Επιλογή ενός πεδίου του φίλτρου.
	3	Ενημέρωση πεδίου από τα μεταδεδομένα του/ των προγραμμάτων.
Extensions	3	Δεν υπάρχουν μεταδεδομένα στα προγράμματα που να σχετίζονται με το επιλεγμένο πεδίο: Δεν προστίθεται καμία τιμή.
Sub-Variations	2α	Ο χρήστης επιλέγει καθολικά το φίλτρο (όλα τα ενεργά πεδία).

Πίνακας 33: Use Case 6: Πρόσθεση τιμών μεταδεδομένων προγραμμάτων σε πεδία φίλτρου.

Use Case 7: Επεξεργασία των τιμών πεδίου φίλτρου.

USE CASE 7	Επεξεργασία των τιμών πεδίου φίλτρου.	
Goal In Context	Πρόσθεση, διαγραφή και αλλαγή βάρους προτίμησης τιμών πεδίου φίλτρου.	
Scope & Level	Σύστημα, Sub-function.	
Precondition	Να έχει ξεκινήσει η εφαρμογή. Να είναι επιλεγμένο κάποιο φίλτρο.	
Success End Condition	Να γίνει η επεξεργασία των τιμών του πεδίου.	
Failed End Condition	Δεν γίνεται καμία αλλαγή (ο χρήστης δεν θέλει).	
Primary, Secondary Actor	Χρήστης.	
Trigger	Ο χρήστης ξεκινάει την επεξεργασία.	
Steps	1	Πρόσθεση καινούργιων τιμών στο πεδίο.
	2	Διαγραφή τιμών από το πεδίο.
	3	Αλλαγή των βαρών προτίμησης των τιμών του πεδίου.
	4	Ανάθεση των επεξεργασμένων τιμών στο πεδίο.
Extensions	3	Δεν υπάρχουν βάρη προτίμησης για τις τιμές του πεδίου.
	4	Ο χρήστης ακυρώνει την όλη διαδικασία.
Sub-Variations	1α	Δεν προστίθενται καινούργιες τιμές στο πεδίο.
	2α	Δεν διαγράφονται τιμές από το πεδίο.
	3α	Δεν γίνεται καμία αλλαγή των βαρών προτίμησης των τιμών του πεδίου.

Πίνακας 34: Use Case 7: Επεξεργασία των τιμών πεδίου φίλτρου.

Use Case 8: Δημιουργία νέου προφίλ με χρήση wizard

USE CASE 8	Επισκόπηση προτεινομένων προγραμμάτων από το σύστημα
Goal In Context	Ο χρήστης θέλει δει τα προτεινόμενα προγράμματα και τα χαρακτηριστικά του προφίλ που ταιριάζουν με τα μεταδεδομένα του προγράμματος
Scope & Level	Σύστημα , Primary Task
Precondition	
Success End Condition	Ο χρήστης βλέπει τα προγράμματα
Failed End Condition	Ο χρήστης δεν βλέπει τα προγράμματα (ο χρήστης δεν θέλει ή system crash)

Primary, Secondary Actor	Χρήστης	
Trigger	Ο χρήστης ζητά να δει τα προγράμματα	
Steps	1	Ο χρήστης επιλέγει ένα όνομα.
	2	Επισκόπηση των προγραμμάτων και των χαρακτηριστικών του προφίλ.
Extensions	1	Ο χρήστης ακυρώνει την όλη διαδικασία.

Πίνακας 35: Use Case 4: Δημιουργία νέου προφίλ με χρήση wizard

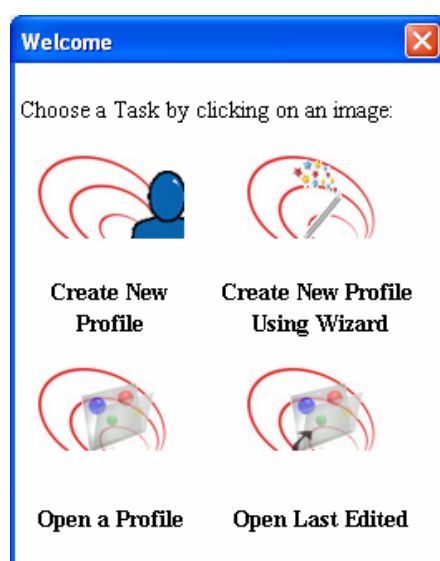
Υλοποίηση

Ο Profile Editor διαχειρίζεται το προφίλ σαν μία δενδρική δομή που αντιστοιχεί ακριβώς στο θεωρητικό μοντέλο του TV-Anytime για το προφίλ (UserPreferences) που παρουσιάστηκε στο τρίτο κεφάλαιο.

Παρακάτω, αναλύονται όλες οι ενέργειες που αναφέρονται στο σχήμα 11 και στα Use Cases και αντιπροσωπεύουν την λειτουργικότητα του εργαλείου, από την δημιουργία ενός νέου προφίλ έως και την αποθήκευσή του. Βεβαίως, τα ενδιάμεσα βήματα μπορούν να επαναληφθούν όσες φορές είναι απαραίτητο και με οποιαδήποτε σειρά.

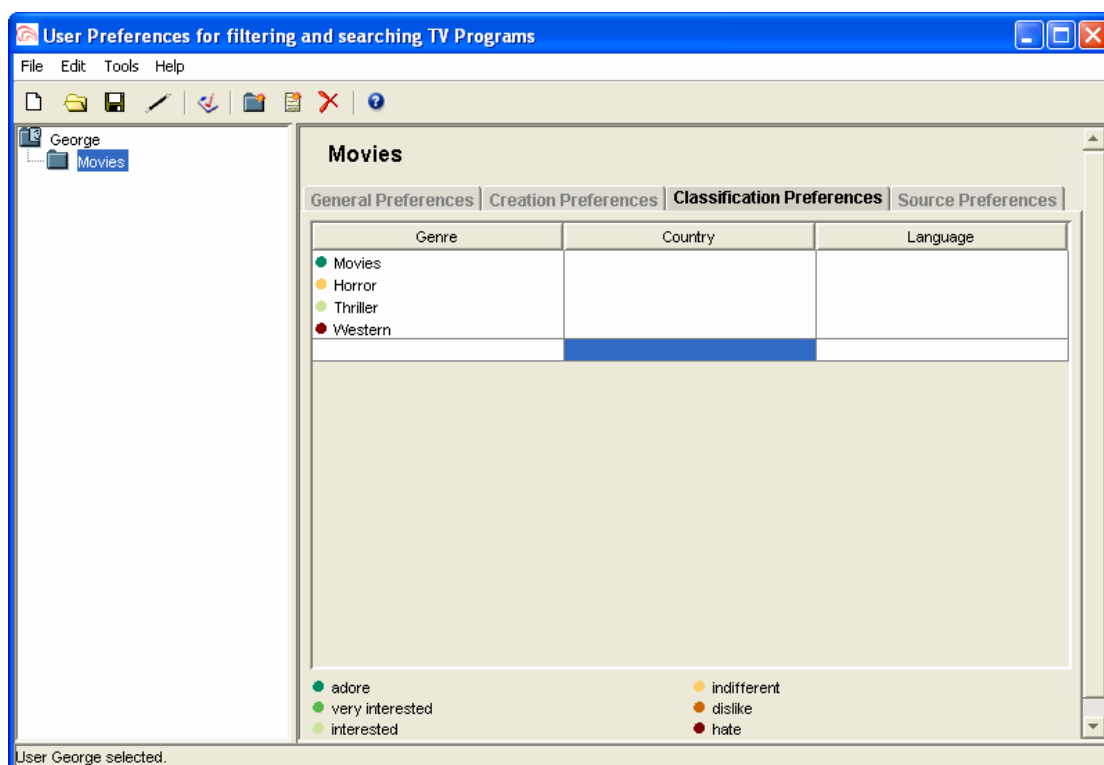
Επεξεργασία Προφίλ

Κατά την εκκίνηση της εφαρμογής ο χρήστης καλείται, είτε μέσω μίας οθόνης καλωσορίσματος είτε μέσω του μενού, να δημιουργήσει ένα καινούργιο προφίλ ή να φορτώσει ένα από την βάση δεδομένων ή ακόμα και από ένα XML έγγραφο. Η οθόνη καλωσορίσματος φαίνεται παρακάτω:



Εικόνα 1: Η οθόνη καλωσορίσματος

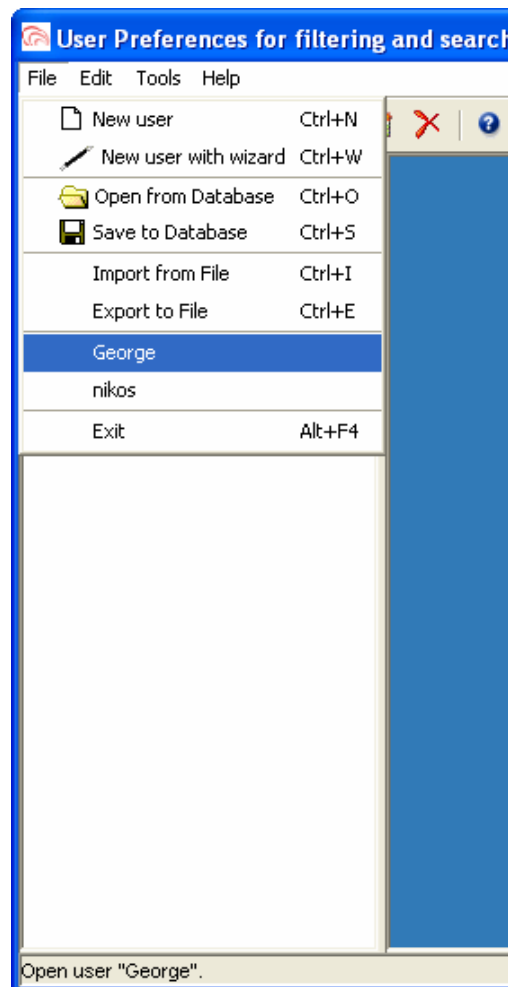
Μετά από αυτό είναι σε θέση να το επεξεργαστεί. Κατά την διαδικασία της επεξεργασίας το κύριο περιβάλλον που του προσφέρεται φαίνεται παρακάτω:



Εικόνα 2: Γενική άποψη του περιβάλλοντος

Το μενού

Υπάρχει το κεντρικό μενού από το οποίο δίνονται στο χρήστη μία πληθώρα επιλογών. Ας σημειωθεί σε αυτό το σημείο ότι οι περισσότερες επιλογές συνδυάζονται και με συντομεύσεις πληκτρολογίου. Παρακάτω φαίνεται παράδειγμα του File μενού:



Εικόνα 3: Το File μενού και το Status Bar.

Ο παρακάτω πίνακας παρουσιάζει το μενού επιγραμματικά με τις συντομεύσεις, όπου υπάρχουν.

Όνομα μενού	Λειτουργία	Συντόμευση
New User	Φτιάχνει ένα καινούργιο χρήστη	Ctrl+N
New User with Wizard	Φτιάχνει ένα καινούργιο χρήστη με wizard	Ctrl+W

Όνομα μενού	Λειτουργία	Συντόμευση
Open User	Φορτώνει έναν χρήστη από την βάση	Ctrl+O
Save to Database	Σώζει τον χρήστη στην βάση	Ctrl+S
Import from file	Ανοίγει έναν χρήστη από ένα XML αρχείο	Ctrl+I
Export to file	Σώζει έναν χρήστη σε ένα XML αρχείο	Ctrl+E
Recent users	Ανοίγει πρόσφατα σωσμένους χρήστες	
Exit	Τερματίζει την εφαρμογή	Alt+F4
Options	Ανοίγει τον διάλογο για την επιλογή των τύπων των χαρακτηριστικών που θα εμφανίζονται	Ctrl+C
Select from programs	Ανοίγει τον διάλογο από τον οποίο ο χρήστης μπορεί να επιλέγει από προγράμματα για εισάγει χαρακτηριστικά	Ctrl+P
View selection list	Ξεινάει την επισκόπηση των προγραμμάτων που προτείνει το σύστημα στον χρήστη	Ctrl+L
Rename tree node	Μετονομάζει τον επιλεγμένο κόμβο του δέντρου	
New Child search filter	Κατασκευάζει έναν καινούργιο κόμβο στο δέντρο	Ctrl+F
New Summary filter	Κατασκευάζει ένα καινούργιο κόμβο που αφορά αυτόματες κατασκευές περιλήψεων	Ctrl+B
Delete	Διαγράφει το επιλεγμένο φίλτρο	
Edit table cell	Ανοίγει τον διάλογο για επεξεργασία του επιλεγμένου κελιού στο δέντρο	
Help	Ανοίγει την βοήθεια	F1
About	Ανοίγει ένα μήνυμα που αφορά την εφαρμογή	

Πίνακας 36: Το προσφερόμενο μενού επιλογών με τις αντίστοιχες συντομεύσεις.

Τα Tabs με τα Preferences

Δεξιά βρίσκετε η περιοχή που δίνει την δυνατότητα στους χρήστες να επεξεργαστούν τον επιλεγμένο κόμβο. Έχει τέσσερα tabs:

Το πρώτο λέγεται General Preferences και όπως μαρτυράει ο τίτλος του δίνει την δυνατότητα στους χρήστες να επεξεργαστούν προτιμήσεις που αφορούν γενικά τον κόμβο. Συγκεκριμένα

περιέχει δυνατότητα αλλαγής του βάρους του κόμβου και το που και πότε ο κόμβος θα εφαρμόζεται (π.χ. τις Σάββατα τα απογεύματα)

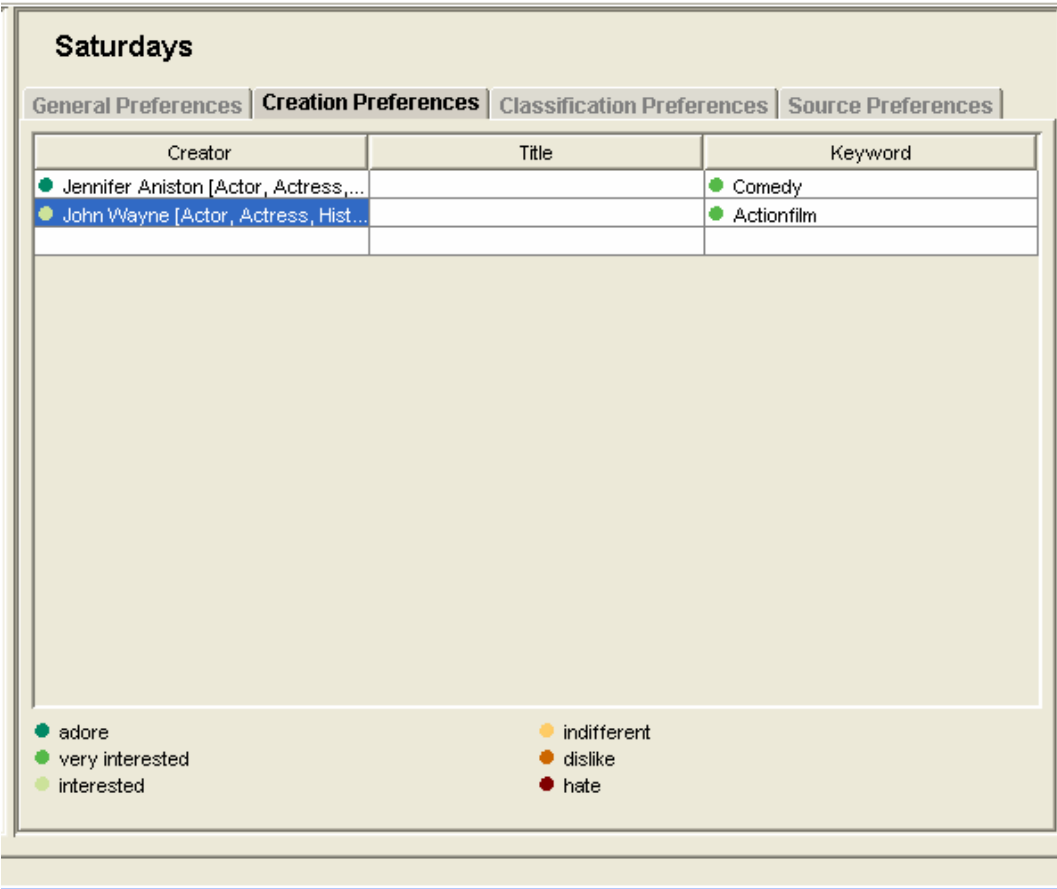
The screenshot shows a web interface for managing user preferences. The main heading is 'Saturdays'. Below it are four tabs: 'General Preferences' (selected), 'Creation Preferences', 'Classification Preferences', and 'Source Preferences'. Under 'General Preferences', there is a 'Preference value' dropdown set to 'very interested' and a 'Protected' checkbox. Below this is a 'Preference Conditions' section containing a table with two columns: 'Place' and 'Time'. The table has one row with the value 'every Saturday 18:05 to 22:05' in the 'Time' column. A tooltip message at the bottom of the table reads: 'To edit the values you see or add new ones just double click here or left click and select edit.'

Place	Time
	every Saturday 18:05 to 22:05

Εικόνα 4: Το General Preferences tab.

Η επεξεργασία κάθε πεδίου των πινάκων γίνεται είτε με διπλό πάτημα του αριστερού πλήκτρου του ποντικιού, είτε με πάτημα του δεξιού πλήκτρου, είτε, τέλος από το μενού. Επεξηγηματικό μήνυμα εμφανίζεται για να βοηθήσει καταλάβουν οι χρήστες τι πρέπει να κάνουν.

Μετά είναι Creation Preferences tab που δίνει την δυνατότητα μέσω ενός Query by Example περιβάλλοντος να επεξεργαστούν τα χαρακτηριστικά που αφορούν προτιμήσεις αναφορικά με την δημιουργία των προγραμμάτων (περιέχει χαρακτηριστικά όπως Creator, Title, κ.λπ.).



Saturdays

General Preferences | **Creation Preferences** | Classification Preferences | Source Preferences

Creator	Title	Keyword
● Jennifer Aniston [Actor, Actress, ...]		● Comedy
● John Wayne [Actor, Actress, Hist...]		● Actionfilm

● adore ● indifferent
 ● very interested ● dislike
 ● interested ● hate

Εικόνα 5: Το Creation Preferences tab. Με παράδειγμα που χρησιμοποιούνται δύο γραμμές του πίνακα.

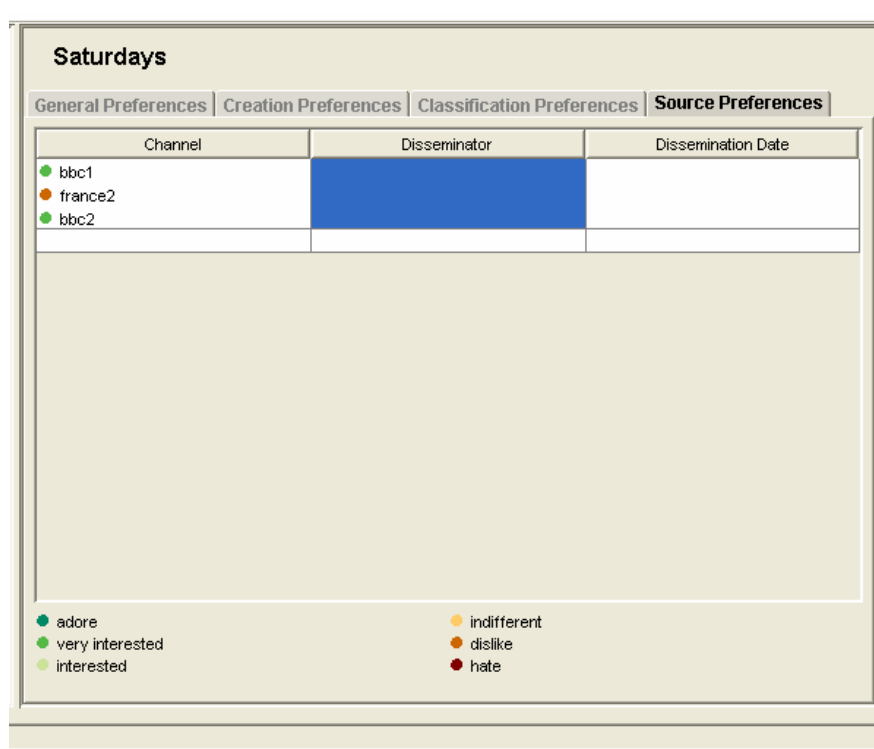
Στην παραπάνω εικόνα ο χρήστης επέλεξε να βλέπει προγράμματα που να έχουν Creator την Jennifer Aniston και Keyword Comedy ή Creator τον John Wayne και Keyword Actionfilm.

Όμοια είναι και tabs Classification και Source Preferences τα οποία περιέχουν αντίστοιχα χαρακτηριστικά. Από κάτω φαίνεται η λεζάντα με τα χρώματα των bullets που αντιπροσωπεύουν το βάρος προτίμησης.

Genre	Country	Language
● Movies		

● adore
● very interested
● interested
● indifferent
● dislike
● hate

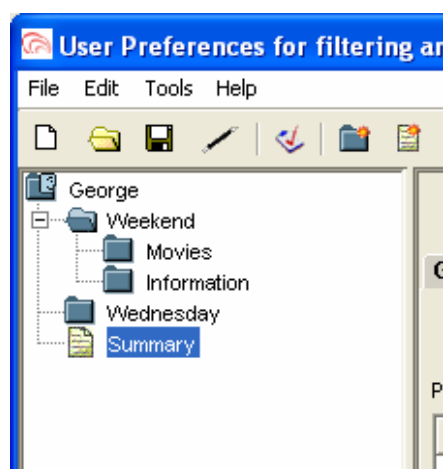
Εικόνα 6: Το Classification Preferences tab. Η λεζάντα με τα χρώματα των bullets εμφανίζεται στο κάτω μέρος του tab.



Εικόνα 7: To Source Preferences tab.

Η Ιεραρχία

Αριστερά της εφαρμογής (βλέπε εικόνα 2) είναι το δέντρο της ιεραρχίας το οποίο δίνει την δυνατότητα στους χρήστες να επεξεργαστούν την ιεραρχία του προφίλ. Φαίνεται καλύτερα παρακάτω:

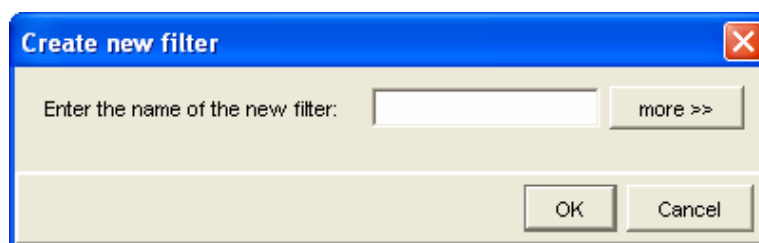


Εικόνα 8: Παράδειγμα Ιεραρχίας

Ο κάθε κόμβος αντιπροσωπεύει ένα κόμβο FASP. Ο χρήστης εδώ έχει την δυνατότητα να κατασκευάσει καινούργιους κόμβους είτε με το δεξί πλήκτρο του ποντικιού και την ανάλογη επιλογή από το μενού που εμφανίζεται είτε από το Edit μενού και την ανάλογη επιλογή είτε ακόμα και με το πλήκτρο Insert. Επίσης έχει την δυνατότητα να σβήσει κόμβους πάλι με τα παραπάνω αλλά και με χρήση του πλήκτρου Delete. Μήνυμα επιβεβαίωσης πάντα πριν την διαγραφή εμφανίζεται στον χρήστη. Ακόμα μπορεί να μετονομάσει τον κόμβο με τους ίδιους πάλι τρόπους αλλά και με τριπλό πάτημα του αριστερού πλήκτρου του ποντικιού. Με διπλό πάτημα οι κόμβοι ανοίγουν (δείχνουν τα παιδιά τους) και αν είναι ανοιχτοί κλείνουν. Επίσης δίνεται η δυνατότητα ο χρήστης να πάρει (drag) έναν κόμβο και αφήνοντάς τον πάνω από κάποιον άλλο (drop) να μετακινηθεί.

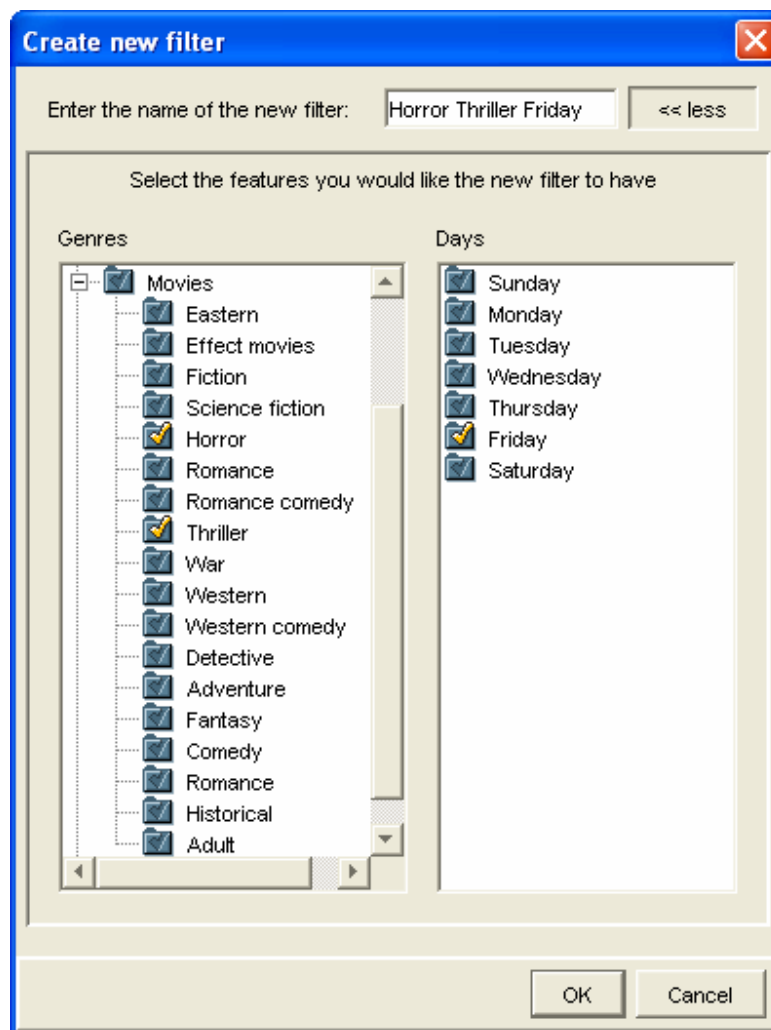
Κατασκευή Νέων Κόμβων

Για την δημιουργία καινούργιων κόμβων έχει κατασκευαστεί ειδικός διάλογος που σκοπό έχει να βοηθήσει τους χρήστες αφενός μεν να κατανοήσουν την χρησιμότητα καινούργιων κόμβων και αφετέρου να κατασκευάσουν πρότυπους κόμβους. Ο διάλογος φαίνεται παρακάτω:



Εικόνα 9: Αρχικός Διάλογος για δημιουργία νέου φίλτρου

Αρχικά ο χρήστης μπορεί να συμπληρώσει μόνο το όνομα του νέου φίλτρου (εικόνα 8). Αν επιλέξει το κουμπί “more>>” τότε ο διάλογος αλλάζει (εικόνα 9).



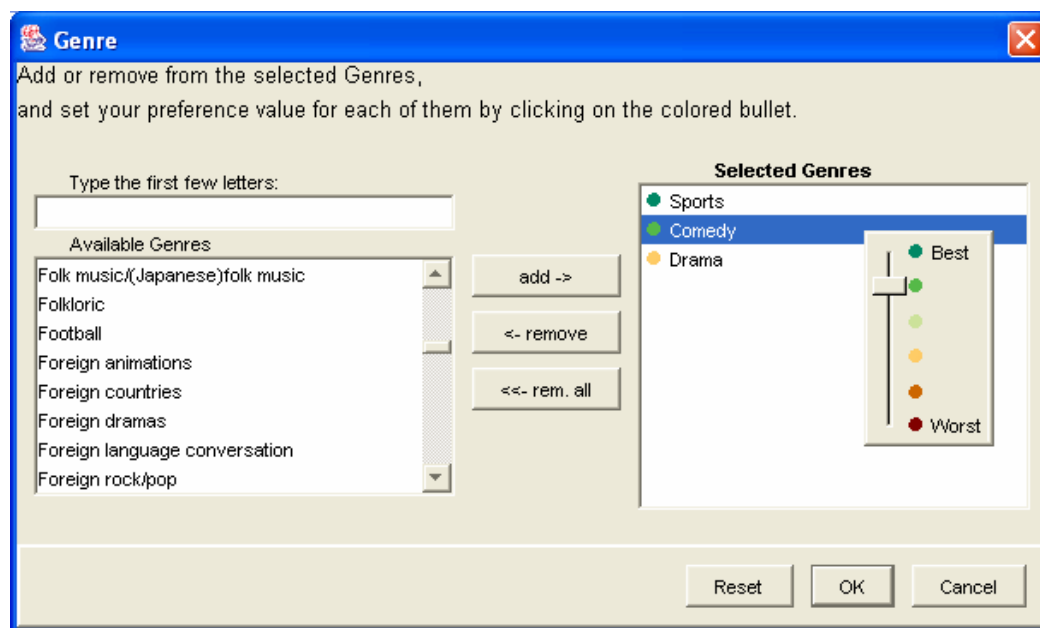
Εικόνα 10: Ανεπτυγμένος διάλογος για δημιουργία νέου προφίλ

Τώρα μπορεί να επιλέξει τις ημέρες της εβδομάδας και τις κατηγορίες (Genres) που τον ενδιαφέρουν για να φτιάξει το καινούργιο φίλτρο.

Επεξεργασία Χαρακτηριστικών

Για επεξεργαστεί ο χρήστης κάποια χαρακτηριστικά ενός φίλτρου αρχικά επιλέγει τον κόμβο που τον ενδιαφέρει από το δέντρο. Στην δεξιά, τώρα, μεριά του παραθύρου εμφανίζονται τα tabs με την πληροφορία που αφορούν τον κόμβο (εικόνα 2) από εκεί μπορεί να επεξεργαστεί κάθε κελί του πίνακα. Η επεξεργασία κάθε κελιού ξεκινάει με διπλό πάτημα του αριστερού πλήκτρου του ποντικιού, επίσης, με πάτημα του δεξιού πλήκτρου και επιλογή από το μενού

που εμφανίζεται και, τέλος, από το κεντρικό μενού. Ο χρήστης είναι όταν θα θελήσει να επεξεργαστεί ένα κελί είναι οικείο σε αυτόν από πολλές εφαρμογές να την ξεκινάει με διπλό πάτημα, εξάλλου αν δεν του φαίνεται εύκολο πάντα υπάρχει και η εναλλακτική του μενού όπου θα καταφύγει στην πρώτη δυσκολία ενώ η εμφάνιση tool tip πάνω από το κελί μπορεί να του δώσει διέξοδο. Στην γενική περίπτωση εμφανίζεται το παρακάτω παράθυρο:

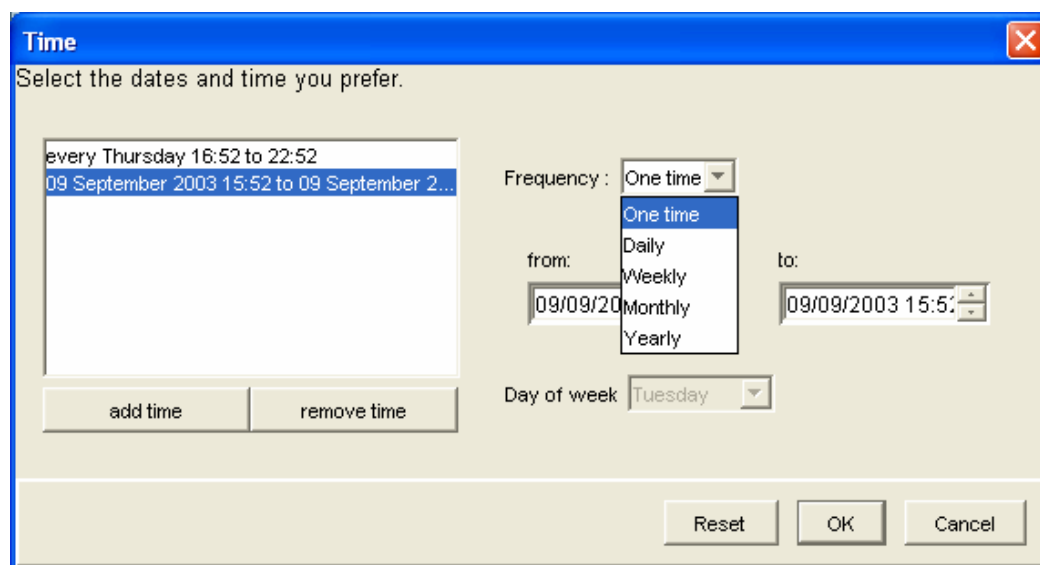


Εικόνα 11: Επεξεργασία γενικών χαρακτηριστικών.

Στην αριστερή λίστα εμφανίζονται όλα τα διαθέσιμα στοιχεία (είτε αυτά είναι Genre είτε Title) από τα οποία ο χρήστης μπορεί να επιλέξει να εισάγει στην δεξιά λίστα, όπου κρατούνται οι επιλογές του. Αυτή η υλοποίηση έγινε με το σκεπτικό να απαλλαγεί ο χρήστης από την περιττή πολυπλοκότητα της δήλωσης των περιπλοκών πεδίων των χαρακτηριστικών, αποφεύγοντας έτσι και τα ενδεχόμενα λάθη που θα έκανε αν είχε να ασχοληθεί με αυτό (τα λάθη θα μπορούσαν να είναι ορθογραφικά, να μην έδινε όλη την απαραίτητη πληροφορία ώστε να μπορούν να συσχετιστούν με όμοια της βάσης δεδομένων, κλπ). Έγινε η θεώρηση ότι όλη η πληροφορία για τα προγράμματα υπάρχει σε μία καθολική βάση. Συνεχίζοντας την περιγραφή, επισημαίνεται η δυνατότητα των χρηστών να αλλάζουν το βάρος προτίμησης προς κάποιο χαρακτηριστικό επιλέγοντας μία από τις έξι τιμές του χρωματικού κώδικα. Το slider εμφανίζεται, όπως αναφέρεται και στην ετικέτα, είτε με πάτημα του αριστερού πλήκτρου του ποντικιού πάνω στο χρωματιστό bullet, είτε με δεξί οπουδήποτε πάνω στην τιμή. Αντί για να

δίνεται στον χρήστη η επιλογή να δίνει μία αριθμητική τιμή από το -100 έως το 100, αναγκάζεται να επιλέξει μία από έξι προεπιλεγμένες τιμές. Αυτό φαίνεται αρχικά λάθος αλλά κάνει το περιβάλλον πιο εύχρηστο, ενώ ο χρήστης μπορεί ανά πάσα στιγμή να κατανοήσει το βάρος με το χρώμα. Στο γενικό περιβάλλον (βλέπε εικόνα 2) υπάρχει υπόμνημα όπου χρειάζεται ώστε ο χρήστης να μην χρειάζεται να θυμάται την βαρύτητα του κάθε χρώματος.

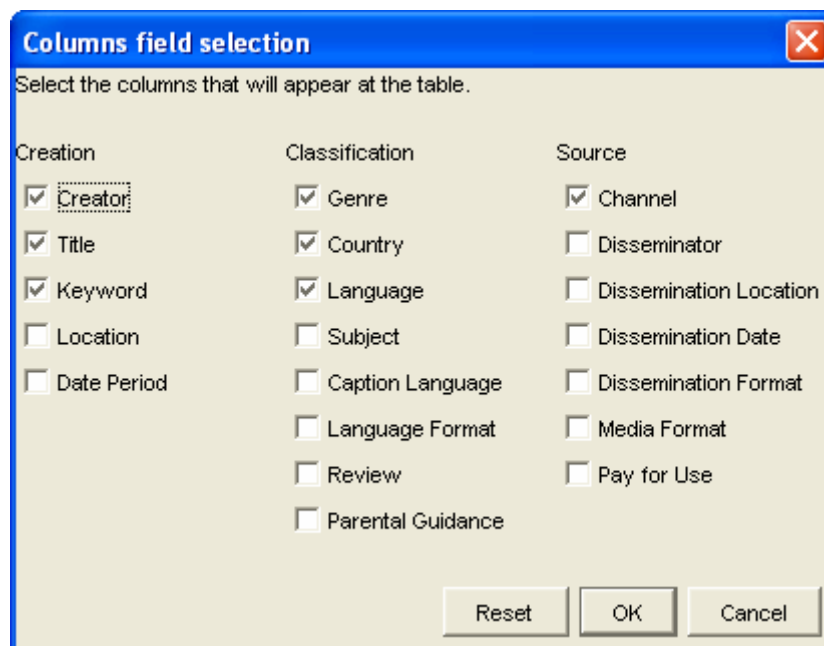
Για τις ημερομηνίες και τις χρονικές περιόδους εμφανίζονται το παρακάτω παράθυρο ανάλογα με την περίπτωση σε δύο παραλλαγές. Η πρώτη φαίνεται στο παρακάτω σχήμα και οι χρόνοι που μπορεί να δηλώσει μπορεί να έχουν περιοδικότητα (ημερήσια, εβδομαδιαία, κ.λπ.). Ενώ στην άλλη παραλλαγή οι χρόνοι δεν έχουν περιοδικότητα και απλά ο χρήστης έχει λιγότερα πεδία να συμπληρώσει.



Εικόνα 12: Διάλογος για επεξεργασία χρονικών περιόδων

Ο τρόπος με τον οποίο εμφανίζονται οι ημερομηνίες είναι τέτοιος που ο χρήστης να κατανοεί άμεσα τη σημασία τους.

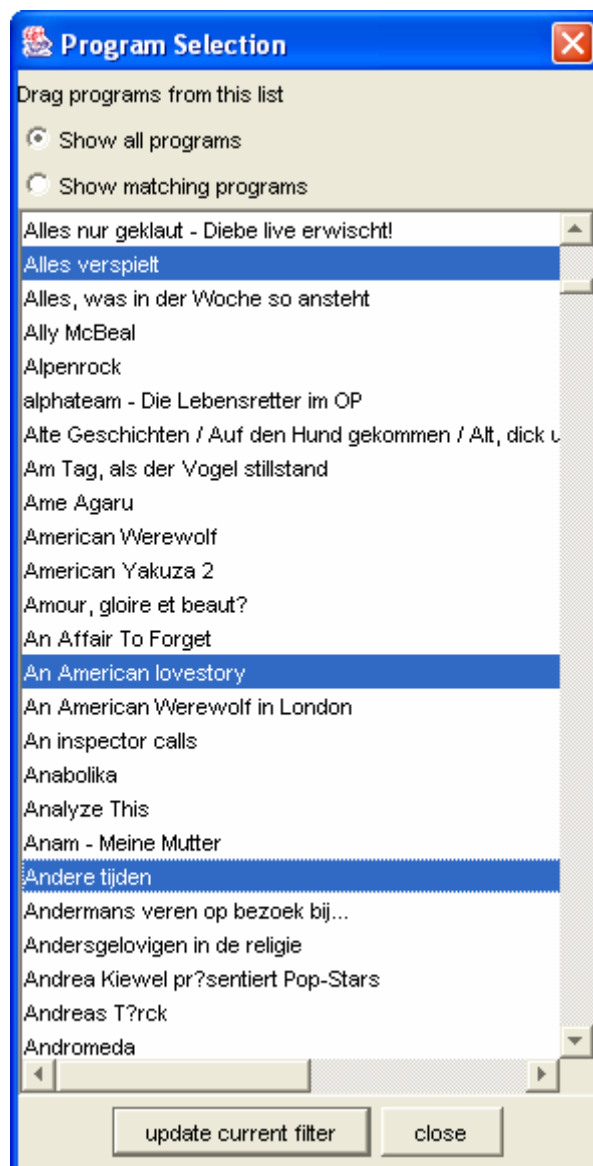
Επιπρόσθετα, ο χρήστης μπορεί να επιλέγει ποιους τύπους χαρακτηριστικών θέλει να βλέπει και να επεξεργάζεται μέσω του παρακάτω διαλόγου, τον οποίο μπορεί να εμφανίσει από το κύριο μενού:



Εικόνα 13: Διάλογος για επιλογή των τύπων χαρακτηριστικών που εμφανίζονται

Προσθήκη χαρακτηριστικών από μεταδεδομένα επιλεγμένων προγραμμάτων που ενδιαφέρουν τον χρήστη

Μία καινοτόμος ιδέα είναι ο χρήστης να έχει την δυνατότητα να προσθέτει σε πεδία που τον ενδιαφέρουν χαρακτηριστικά από τα μεταδεδομένα προγραμμάτων που αρέσκεται. Αυτά μπορεί να είναι μία λίστα προγραμμάτων που έχει δει στο παρελθόν, όχι αναγκαστικά από τηλεόραση και όχι αναγκαστικά με το PVR του TV-Anytime. Συμβαίνει όμως το PVR του TV-Anytime να ξέρει γι' αυτά καθώς κρατάει μεταδεδομένα για τα προγράμματα που είδε ο χρήστης. Ιδανική είναι η περίπτωση που φτιάχνει ένα καινούργιο φίλτρο για τα σαββατοκύριακα και το μόνο που έχει να κάνει είναι να επιλέξει τα προγράμματα που τον ενδιαφέρει να βλέπει τότε και αυτόματα να κατασκευάζεται το φίλτρο. Μετά έχοντας κατανοήσει την σημασιολογία των πινάκων (βλέποντας τις τιμές που εισήγαγε το σύστημα) μπορεί να βγάλει όσες δεν τον ενδιαφέρουν πραγματικά, να προσθέσει άλλες κ.λπ.. Η παρακάτω εικόνα δείχνει τον διάλογο που δίνει αυτή την δυνατότητα στον χρήστη.



Εικόνα 14: Διάλογος που επιτρέπει στον χρήστη να προσθέτει τιμές, στα χαρακτηριστικά που τον ενδιαφέρουν, οι οποίες προέρχονται από τα μεταδεδομένα των προγραμμάτων.

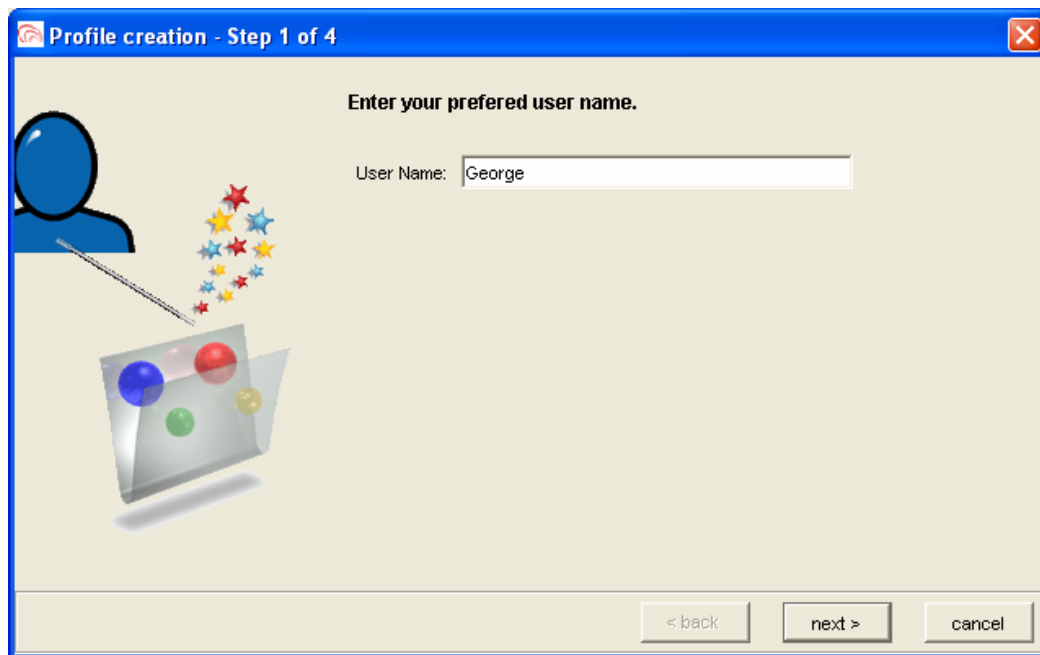
Ο χρήστης επιλέγει τα προγράμματα που τον ενδιαφέρουν και πατώντας το κουμπί “update current filter” προστίθενται αυτόματα τα μεταδεδομένα των προγραμμάτων στα επιλεγμένα πεδία του φίλτρου. Η διαδικασία της πρόσθεσης των μεταδεδομένων των προγραμμάτων περιλαμβάνει ένα μηχανισμό ανεύρεσης των μεταδεδομένων των προγραμμάτων και την κατάταξη τους βαθμολογικά σύμφωνα με τον αριθμό των εμφανίσεών τους στα προγράμματα αν είναι κύρια ή δευτερεύοντα στην περιγραφή του προγράμματος κλπ. Λέγοντας επιλεγμένα πεδία υπονοούνται αυτά που έχουν επιλεγεί να φαίνονται από τον διάλογο της εικόνας 13.

Επιπρόσθετα δίνεται η δυνατότητα στον χρήστη να επιλέξει ένα σύνολο προγραμμάτων, να το “τραβήξει” (drag) και να το “ρίξει” (drop) είτε σε ένα κελί του πίνακα (προσθέτει τιμές μόνον σε αυτό το κελί) είτε σε ένα κόμβο του δέντρου (αλλάζει όλο το φίλτρο όπως περιγράφηκε παραπάνω). Επιπλέον ο χρήστης μπορεί να δει τα χαρακτηριστικά κάθε προγράμματος με διπλό πάτημα του ποντικιού.

Χρήση του Wizard

Μια ιδέα πολύ κοντινή σε αυτήν της προηγούμενης παραγράφου είναι να κατασκευάζονται προφίλ από τα μεταδεδομένα των προγραμμάτων. Για αυτόν τον σκοπό κατασκευάστηκε ένας wizard ο οποίος είναι ιδανικός για αδαείς χρήστες, ενώ προσφέρει και ενδιαφέρουσες επιλογές για πιο εξοικειωμένους. Ο wizard έχει 4 βήματα.

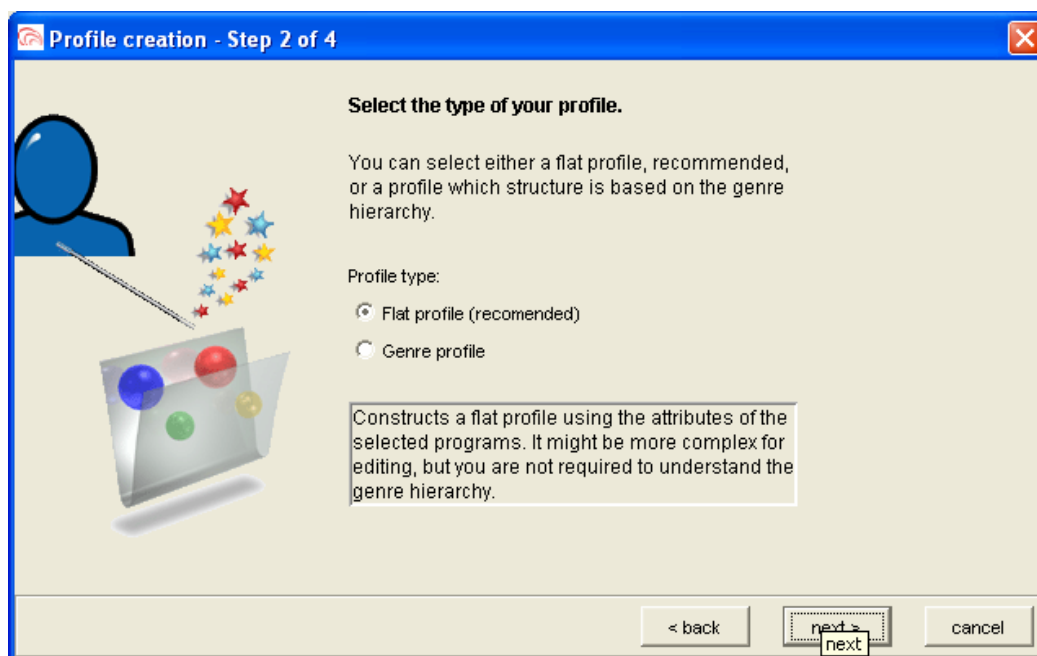
Στο πρώτο ο χρήστης βάζει το όνομά του ή ενδεχομένως το όνομα που θέλει να δώσει στο προφίλ:



Εικόνα 15: 1ο βήμα του wizard.

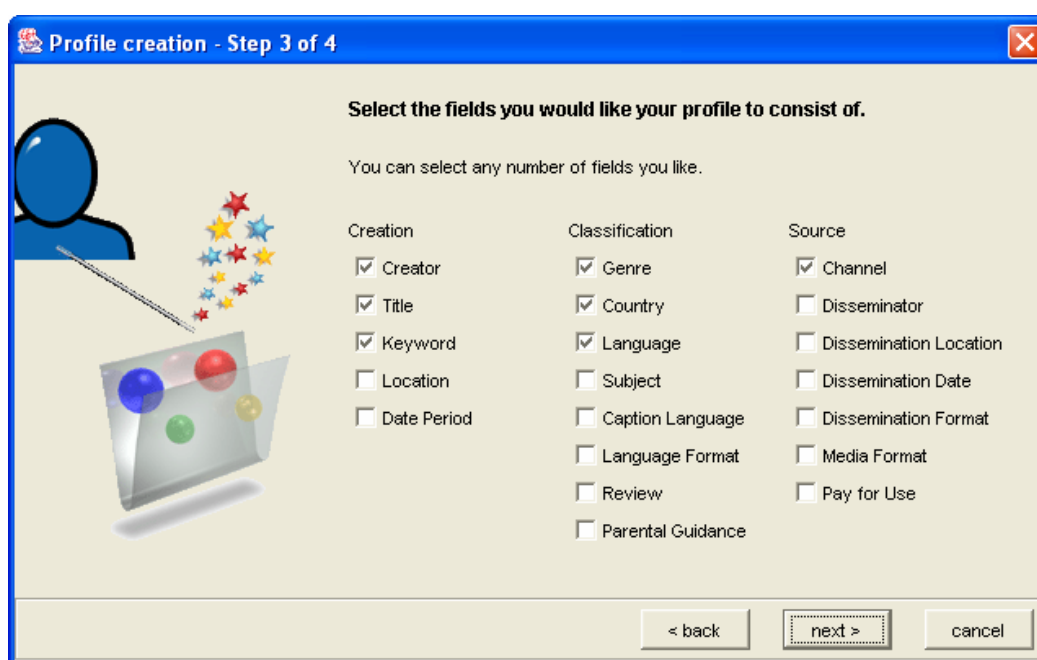
Στο δεύτερο βήμα ο χρήστης επιλέγει αν το προφίλ που θα δημιουργηθεί θα έχει μόνο ένα επίπεδο ή θα είναι ιεραρχικό με βάση την κατηγορία. Ιεραρχικό με βάση την κατηγορία

(Genre) σημαίνει ότι ο κάθε κόμβος του προφίλ που θα δημιουργηθεί θα είναι μία κατηγορία και επειδή η δομή των κατηγοριών είναι ιεραρχική (π.χ. η Movies έχει σαν παιδιά την Comedy και την Thriller) και το προφίλ θα είναι και αυτό ιεραρχικό.



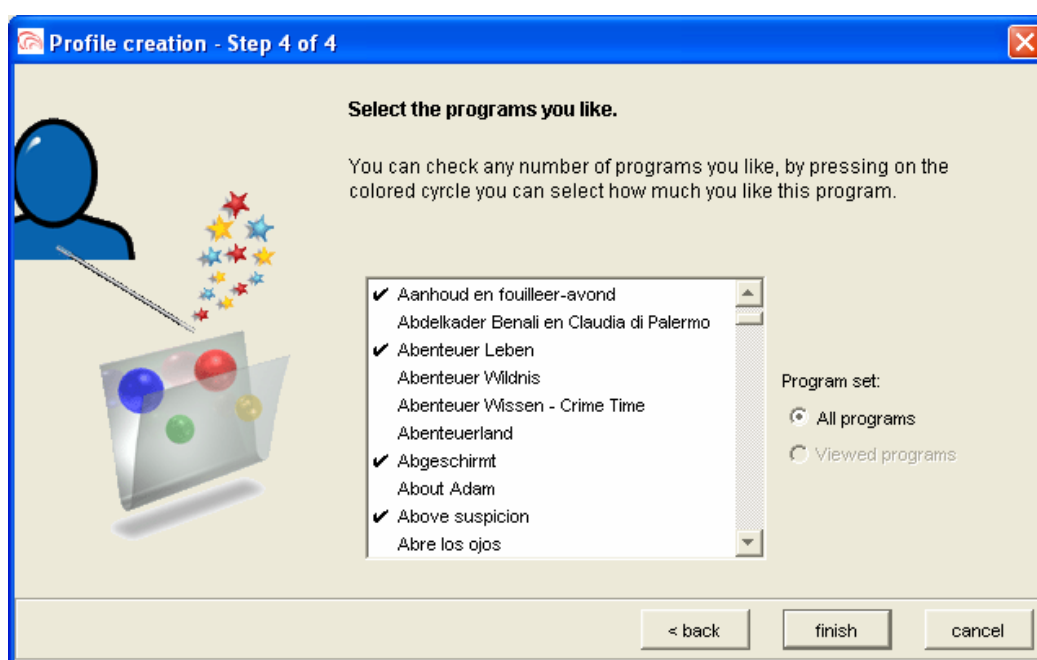
Εικόνα 16: 2ο βήμα του wizard.

Στο τρίτο βήμα ο χρήστης επιλέγει τους τύπους χαρακτηριστικών που τον ενδιαφέρει να αντλήσει από τα προγράμματα.



Εικόνα 17: 3ο βήμα του wizard.

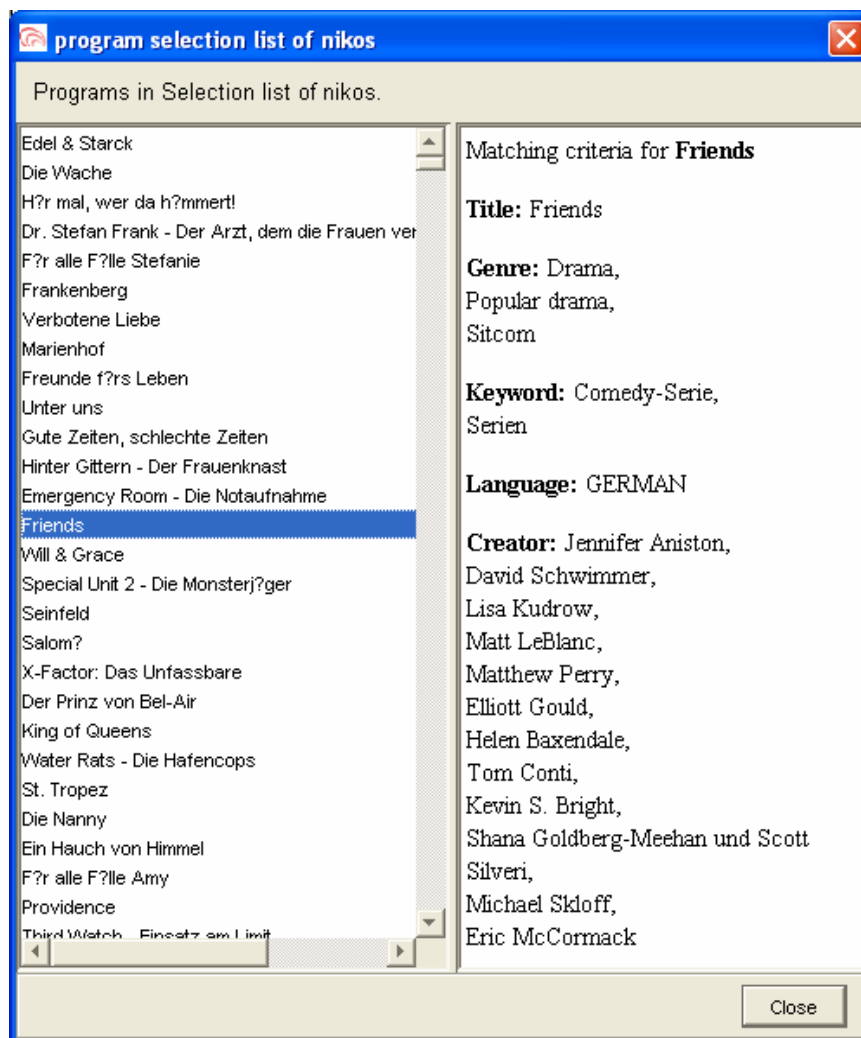
Στο τέταρτο και τελευταίο βήμα ο χρήστης επιλέγει τα προγράμματα που τον ενδιαφέρουν από μία λίστα από όλα τα προγράμματα. Αν στο όνομα του (το έχει δηλώσει στο πρώτο βήμα) υπάρχει στην βάση πληροφορία ότι έχει δει κάποια προγράμματα τότε του δίνεται η δυνατότητα να επιλέξει προγράμματα από αυτά που έχει δει και όχι από ένα κατά πολύ μεγαλύτερο σύνολο.



Εικόνα 18: 4ο βήμα του wizard.

Επισκόπηση προτεινομένων προγραμμάτων

Όπως έχει αναφερθεί στο προηγούμενο κεφάλαιο διατηρείται πληροφορία για κάθε πρόγραμμα που προτείνεται στον χρήστη από το σύστημα πια χαρακτηριστικά του προφίλ του ταιριάζουν με αυτό. Αυτό γίνεται για να κατανοεί ο χρήστης την συμπεριφορά του συστήματος και εξαιτίας αυτού να το εμπιστεύεται για τις εκτιμήσεις που του προτείνονται. Για να μπορεί, λοιπόν, ο χρήστης να δει τα προγράμματα που του προτείνει το σύστημα καθώς επίσης και τα χαρακτηριστικά του προφίλ του που ταιριάζουν με τα μεταδεδομένα κάθε προγράμματος προσφέρεται ο παρακάτω διάλογος:

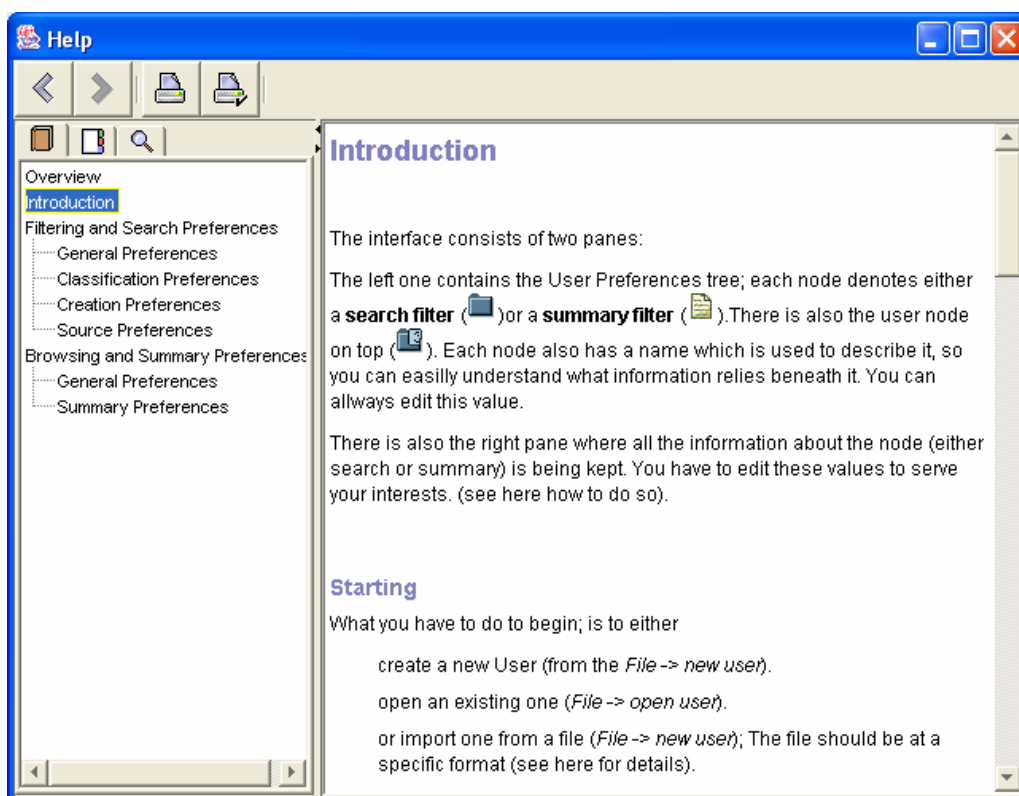


Εικόνα 19: Ο διάλογος για την επισκόπηση των προγραμμάτων.

Στα αριστερά φαίνεται η λίστα από τα προγράμματα ενώ στα δεξιά εμφανίζονται τα χαρακτηριστικά του προφίλ που ταιριάζουν στο πρόγραμμα που είναι επιλεγμένο από την λίστα. Επιπλέον δίνεται στο χρήστη η δυνατότητα με διπλό πάτημα του ποντικιού στο κάθε πρόγραμμα να δει τα χαρακτηριστικά του σε ξεχωριστό παράθυρο.

Το σύστημα βοήθειας

Η εφαρμογή υποστηρίζει και ένα πλήρες σύστημα βοήθειας, το οποίο εξηγεί αναλυτικά όλες τις λειτουργίες της και παρέχει απαντήσεις στις τυχόν απορίες του χρήστη. Είναι βασισμένο στο JavaHelp 2.0 πακέτο της Sun. Μια αντιπροσωπευτική άποψη της βοήθειας φαίνεται παρακάτω.



Εικόνα 20: Το σύστημα βοήθειάς.

Περίληψη

Σε αυτό το κεφάλαιο αναλύθηκε το γραφικό περιβάλλον που υλοποιήθηκε για να καλυφθούν οι ανάγκες των χρηστών για την κατασκευή και αναπροσαρμογή του προφίλ τους. Η σχεδίαση του γραφικού περιβάλλοντος έγινε επί τη βάση της πλήρους κάλυψης του TV-Anytime σχήματος για τα μεταδεδομένα των χρηστών. Αυτή η ανάγκη της κάλυψης δημιούργησε ένα περιβάλλον όχι τόσο φιλικό όσο θα σχεδιάζονταν για λιγότερα χαρακτηριστικά (π.χ. μόνο για Genres και Keywords). Αυτή η έλλειψη της φιλικότητας υπερκεράστηκε με την παροχή δυνατότητας προς τους χρήστες να κατασκευάζουν και να τροποποιούν επιμέρους φίλτρα χρησιμοποιώντας μεταδεδομένα προγραμμάτων που είδε ο χρήστης στο παρελθόν και του άρεσαν. Επίσης δόθηκε η δυνατότητα στους χρήστες να κατασκευάζουν από την αρχή προφίλ χωρίς να χρησιμοποιούν το περιβάλλον, για μεγαλύτερη διευκόλυνση άπειρων χρηστών, μέσω ενός wizard ο οποίος κατασκευάζει τόσο ιεραρχικά όσο και απλά προφίλ βασισμένος στις προτιμήσεις των χρηστών για τα προγράμματα. Επίσης, του προσφέρεται η δυνατότητα να βλέπει τα προγράμματα που του προτείνει το σύστημα βασισμένο στο προφίλ του καθώς και την αιτία της πρότασης (τα χαρακτηριστικά του προφίλ του που ταίριαζαν με τα μεταδεδομένα του εκάστοτε προγράμματος). Κατά την υλοποίηση της εφαρμογής τηρήθηκαν βασικές αρχές σχεδιασμού ώστε το περιβάλλον να είναι όσο κατά το δυνατόν πιο εύχρηστο.

*Αλοίμονο, είναι δυνατόν κανείς,
στον κόσμο αυτόν που τον φωτίζει ο
ήλιος, μέσα στην ίδια μέρα να τα
μάθει όλα αυτά;*

Ευριπίδης

Κ ε φ ά λ α ι ο V I

ΑΝΑΚΕΦΑΛΑΙΩΣΗ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

Συμπεράσματα - Ανακεφαλαίωση

Η ολοκλήρωση της παρούσας διατριβής μας οδηγεί στο κύριο συμπέρασμα πως ενώ τα μεταδεδομένα, έτσι όπως είναι οργανωμένα μέσα στο TV- Anytime, είναι ένας πλήρης τρόπος περιγραφής ενός οποιουδήποτε οπτικοακουστικού μέσου και κάθε προτίμησης χρήστη για εφαρμογές ψηφιακής τηλεόρασης, ταυτόχρονα είναι και αρκετά πολύπλοκα με αποτέλεσμα να απαιτούν προσεκτική μεταχείριση ώστε να γίνεται ευκολότερη η διαχείρισή τους. Η αξιοποίησή τους δίνει ένα ισχυρό εργαλείο με το οποίο οι χρήστες μπορούν, χρησιμοποιώντας το προφίλ τους, να αναζητούν και να βρίσκουν τα προγράμματα που προτιμούν από μία πληθώρα συνεχώς προβαλλόμενων.

Αρχικά, εξετάστηκε το σχήμα μεταδεδομένων του TV-Anytime για τα μεταδεδομένα τόσο του περιεχομένου των προγραμμάτων όσο και του προφίλ των χρηστών. Καθορίστηκαν επακριβώς οι αντιστοιχίσεις μεταξύ των χαρακτηριστικών των προγραμμάτων και του προφίλ για να είναι δυνατή η αξιοποίηση των προφίλ των χρηστών από μηχανισμούς διήθησης που αναπτύχθηκαν στη συνέχεια.

Μελετήθηκαν μηχανισμοί ευφρούς διήθησης TV-Anytime περιεχομένου προγραμμάτων. Συγκεκριμένα αναπτύχθηκε ένας μηχανισμός που αναλαμβάνει να βρει το βαθμό ομοιότητας του περιεχομένου ενός προγράμματος σε σχέση με το προφίλ χρηστών. Υποστηρίχθηκε πλήρως το σχήμα μεταδεδομένων του TV-Anytime ώστε οι ερωτήσεις ανάκτησης και

διήθησης που μπορεί να απαντήσει το σύστημα να περιλαμβάνουν ένα ευρύ σύνολο από χαρακτηριστικά αξιοποιώντας έτσι όλη την δυνατή πληροφορία που υπάρχει για τα προγράμματα, είτε αυτή προέρχεται από κάποια TV-Anytime εφαρμογή είτε ακόμα και από το διαδίκτυο. Μέχρι τώρα ο χρήστης ήταν σε θέση να κάνει ερωτήσεις στο σύστημα ή να έχει ένα προφίλ με ελάχιστα χαρακτηριστικά (π.χ. μόνο Genres και Keywords) αφήνοντας ουσιαστικά ανεκμετάλλευτη την τεράστια πληροφορία που υπάρχει για το οπτικοακουστικό υλικό. Ειδικά, σχεδιάστηκε ένα μεταφραστικό μοντέλο του σχήματος μεταδεδομένων του TV-Anytime για τις προτιμήσεις των χρηστών σε ασαφείς δυαδικές εκφράσεις ώστε από ένα αφηρημένο χώρο προτιμήσεων να υπάρχει μία πλήρως καθορισμένη συντακτικά και σημασιολογικά ερώτηση. Επίσης, για να κατανοεί ο χρήστης καλύτερα την συμπεριφορά του συστήματος διατηρείται πληροφορία με τι χαρακτηριστικά του προφίλ ταιριάζουν τα μεταδεδομένα κάθε προγράμματος. Έτσι, καλλιεργείται εμπιστοσύνη από τον χρήστη ως προς το σύστημα, αφού διαπιστώνει, ότι το σύστημα δεν συμπεριφέρεται αυθαίρετα αλλά σύμφωνα με τις προτιμήσεις του, από τις οποίες τώρα μπορεί να εξαιρέσει όσες δεν δίνουν ικανοποιητικές για αυτόν προτάσεις από το σύστημα.

Επιπλέον, κατασκευάστηκε μηχανισμός συνεργατικής διήθησης ο οποίος διαχειρίζεται πληροφορίες για τις αρέσκειες των χρηστών για να εκμαιεύσει ομοιότητες και να κάνει προτάσεις προγραμμάτων προς τους χρήστες. Ο μηχανισμός που υλοποιήθηκε δεν παρέμεινε στα στενά όρια της συνηθισμένης συνεργατικής διήθησης αλλά, χωρίς να μειώσει την απόδοσή του, ξεπέρασε το πρόβλημα τόσο του κρύου ξεκινήματος, τροφοδοτώντας με δεδομένα που προέρχονται από τα προφίλ των χρηστών, όσο και των προσωπικών δεδομένων των χρηστών, με υπολογισμό των απαραίτητων δεδομένων από ένα κατά πολύ μικρότερο υποσύνολο χρηστών, απαγκιστρώνοντάς τους έτσι από την υποχρεωτικό διαμοιρασμό των δεδομένων τους σε ένα κεντρικό εξυπηρετητή.

Στη συνέχεια κατασκευάστηκε ένα γραφικό περιβάλλον για την κάλυψη των αναγκών των χρηστών για την κατασκευή και επεξεργασία του προφίλ τους. Η σχεδίαση του γραφικού περιβάλλοντος έγινε επί τη βάση της πλήρους υποστήριξης του TV-Anytime σχήματος για τα μεταδεδομένα των χρηστών ώστε έμπειροι χρήστες όταν το θελήσουν, να μπορούν να ικανοποιηθούν και οι πιο λεπτομερείς απαιτήσεις τους στην διαδικασία της καταγραφής των προτιμήσεών τους. Η υποστήριξη αυτή δημιούργησε ένα περιβάλλον όχι τόσο φιλικό όσο θα

σχεδιάζονταν για λιγότερα χαρακτηριστικά (π.χ. μόνο για Genres και Keywords) παρόλο που κρύβεται από τον χρήστη, όσο το δυνατόν, η πολυπλοκότητα του σχήματος. Η έλλειψη της φιλικότητας υπερκεράστηκε με την παροχή δυνατότητας προς τους χρήστες να κατασκευάζουν και να τροποποιούν επιμέρους φίλτρα και χαρακτηριστικά χρησιμοποιώντας μεταδεδομένα προγραμμάτων που είδε ο χρήστης στο παρελθόν και του άρεσαν. Επίσης δόθηκε η δυνατότητα στους χρήστες να κατασκευάζουν από την αρχή προφίλ χωρίς να χρησιμοποιούν το περιβάλλον, για μεγαλύτερη διευκόλυνση άπειρων χρηστών, μέσω ενός wizard ο οποίος κατασκευάζει τόσο ιεραρχικά όσο και απλά προφίλ βασισμένος στις προτιμήσεις των χρηστών για τα προγράμματα. Κατά την υλοποίηση της εφαρμογής τηρήθηκαν βασικές αρχές σχεδιασμού ώστε το περιβάλλον να είναι όσο κατά το δυνατόν πιο εύχρηστο.

Η εργασία αυτή ήταν μέρος του ευρωπαϊκού ερευνητικού προγράμματος UP-TV, που έχει σαν στόχο την ανάπτυξη συστημάτων που θα υποστηρίζουν ψηφιακή και αλληλεπιδραστική τηλεόραση και βασίζεται στις αρχές και τα πρότυπα που επιβάλλει το TV-Anytime Forum. Ως εκ τούτου, είναι δομική μονάδα στο σύστημα του UP-TV για τον υπολογισμό προτάσεων προγραμμάτων στους χρήστες. Επίσης, χρησιμοποιήθηκε σε τμήματα της μονάδας που αναλαμβάνει την αυτόματη δημιουργία και αναπροσαρμογή του προφίλ (βλέπε [10]) ενώ αξιοποιείται και από το μοντέλο φυσικής γλώσσας κυρίως για να απαντήσει της ερωτήσεις των χρηστών (βλέπε [40]).

Η υλοποίηση των παραπάνω, έγινε σε MySQL και σε Java.

Μελλοντικές Επεκτάσεις

Η παρούσα διατριβή μπορεί να επεκταθεί σε διάφορους ενδιαφέροντες τομείς αξιοποιώντας, έτσι, την ήδη υπάρχουσα εργασία.

Επεκτάσεις αναφορικά με τους Μηχανισμούς Ευφυούς Διήθησης Περιεχομένου

Αρχικά θα μπορούσε να κατασκευαστεί ένας εξατομικευμένος ηλεκτρονικός τηλεοπτικός οδηγός. Η λογική του οποίου είναι: να κατασκευάζεται ένα ημερήσιο πρόγραμμα για τον χρήστη το οποίο θα περιλαμβάνει προγράμματα που θα προκύπτουν με μηχανισμούς ευφυούς διήθησης περιεχομένου. Τα προγράμματα θα μπορεί να είναι από διαφορετικά κανάλια. Δηλαδή θα κατασκευάζονται εξατομικευμένοι τηλεοπτικοί οδηγοί.

Θα μπορούσε να αναπτυχθεί μηχανισμός ο οποίος να βοηθά την βασισμένη σε αντικείμενα συνεργατική διήθηση να ξεπεράσει το πρόβλημα εντοπίζεται για προγράμματα τηλεόρασης που εμφανίζονται για πρώτη φορά. Η φιλοσοφία του μηχανισμού βασίζεται στην εύρεση για ομοιότητας με βάση το περιεχόμενο. Αυτή η προσέγγιση θα έχει πολύ καλή απόκριση για επεισόδια τηλεοπτικών σειρών ή για εβδομαδιαίες ενημερωτικές εκπομπές, κ.λπ.

Επεκτάσεις αναφορικά με το Γραφικό Περιβάλλον

Όσο αφορά το γραφικό περιβάλλον, κρίνεται απαραίτητο να γίνουν δοκιμές ευχρηστίας (Usability Test). Με αυτές τις δοκιμές θα βρεθούν λάθη ή παραλήψεις που δυσχεραίνουν τη δουλειά των χρηστών, ενώ θα μπορούσαν να γίνουν και προτάσεις για καινούργια εργαλεία ή δυνατότητες που θα κάνουν ακόμα πιο εύκολη την πλοήγηση της εφαρμογής.

Επιπλέον, μπορεί να κατασκευαστεί γραφικό περιβάλλον για την διαχείριση του προφίλ σε περιβάλλον ψηφιακής τηλεόρασης. Επειδή σε τέτοιο περιβάλλον η μόνη συσκευή για είσοδο στο σύστημα που προσφέρεται στον χρήστη είναι το τηλεχειριστήριο πρέπει να μελετηθούν εναλλακτικοί τρόποι παρουσίασης του προφίλ.

Επίσης, το γραφικό περιβάλλον θα μπορούσε να επεκταθεί με τεχνικές μάθησης ώστε παρατηρώντας τις κινήσεις των χρηστών να κατανοεί τι θέλουν να κάνουν και να τους προσφέρει αυτές τις επιλογές ώστε και ο πλέον αδαής χρήστης να μπορεί με ευκολία να περιηγηθεί στο γραφικό περιβάλλον.

Επιπρόσθετα, θα μπορούσαν να υποστηριχτούν αυτόματες κατασκευές στερεοτύπων προφίλ ώστε να διευκολύνονται οι χρήστες χρησιμοποιώντας τους. Αυτό που γίνεται μέχρι στιγμής είναι ο χρήστης να μπορεί να ανοίξει ένα στερεότυπο, να το επεξεργαστεί και μετά να το αποθηκεύσει με άλλο όνομα. Αυτόματοι τρόποι θα μπορούσαν να εκμαιεύουν πληροφορίες από τους χρήστες και να το κατασκευάζουν αυτόματα, όπως γίνεται για παράδειγμα με τον wizard, ο οποίος υποστηρίζει κατασκευή ιεραρχικού προφίλ με βάση την κατηγορία (genre).

*Πη παρέβην; Τι δ' έρεξα; Τι μοι
δέον ουκ ετελέσθη;*

Πυθαγόρας

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Joon Ho Lee, “*Properties of Extended Boolean Models in Information Retrieval*”, In Proceedings of the 17th ACM SIGIR International Conference on Research and Development in Information Retrieval, 1994, 182-190.
- [2] Joon Ho Lee, W. Y. Kim, M. H. Kim, Y. J. Lee “*On the evaluation of boolean operators in the extended boolean framework*”, In Proceedings of the 16th ACM SIGIR International Conference on Research and Development in Information Retrieval, 1993, 291-297.
- [3] R. Kruse, J. Gebhardt, F. Klawonn: “*Foundations of Fuzzy Systems*”, John Wiley & Sons Ltd., ISBN 0-471-94243-X, 1994.
- [4] Salton, G., Buckley, C. “*Introduction to Modern Information Retrieval*”, McGraw-Hill Book Company, New York, 1982.
- [5] Yates, R.B., Neto, B.R.: *Modern Information Retrieval*. ACM Press (1999).
- [6] D. A. Grossman, O. Frieder, D. O. Holmes, and D. C. Roberts: “*Integrating structured data and text: A relational approach*”, Journal of the American Society for Information Sciences, 48(2):122–132, 1997.
- [7] Sriram Raghavan, Hector Garcia-Molina: “*Integrating Diverse Information Management Systems: A Brief Survey*”, IEEE Data Engineering Bulletin, Vol. 24, No. 4, pp. 44-52, December 2001.
- [8] The TV-Anytime Forum, *Specification Series: S-3 on: Metadata (Normative)*, www.tvanytime.org, June 2002
- [9] Text of ISO/IEC 15938-5 *Information Technology - Multimedia content description interface - Part 5 Multimedia Description Schemes*, 2001.
- [10] Ιωάννης Αριστοτέλης Κοτόπουλος – Σύστημα Αυτόματης Δημιουργίας και Αναπροσαρμογής Προφίλ Χρηστών Σε περιβάλλον Ψηφιακής Τηλεόρασης, Πολυτεχνείο Κρήτης, 2003
- [11] Konstan J. A., Miller B. N., Maltz D., Herlocker J. L., Gordon L. R., Riedl J.: “*GroupLens: Applying collaborative filtering to Usenet News*”, Communications of the ACM, March, 40(3), 77-87.
- [12] Balabanovic M., Shoham Y.: “*Fab: Content-Based, Collaborative Recommendation*”, Communications of the ACM, March, 40(3), 66-72.
- [13] Terveen L., Hill W., Amento B., McDonald D., Creter J.: “*PHOAKS: A System for Sharing Recommendations*”, Communications of the ACM, March, 40(3), 66-72.
- [14] Bollacker K. D., Lawrence S., Lee Giles C.: “*CiteSeer: An Autonomous Web Agent for Automatic Retrieval and Identification of Interesting Publications*”, 2nd International ACM Conference on Autonomous Agents, pp. 116-123, May, 1998.
- [15] Middleton, S.E., De Roure, D. C., and Shadbolt, N.R.: “*Capturing Knowledge of User Preferences: ontologies on recommender systems*”, In Proceedings of the First International Conference on Knowledge Capture (K-CAP 2001), Oct 2001, Victoria, B.C. Canada.
- [16] Herlocker J. L., Konstan J. A., Borchers A., Riedl J.: “*An Algorithmic Framework for Performing Collaborative Filtering*”, In Proceedings of the 22nd ACM SIGIR International Conference on Research and Development in Information Retrieval, 1999, pp. 230-237.

- [17] Sarwar B., Karypis G., Konstan J., Riedl J.: *Item-Based Collaborative Filtering Recommendation Algorithms*, WWW01, pp. 285-295, Hong Kong, 2001.
- [18] Patrick Baudisch: *Joining Collaborative and Content-Based Filtering*, Online Proceedings of the CHI '99 Workshop.
- [19] *MySQL Optimization*, MySQL Reference Manual, 300-340, 2002.
- [20] Abraham Silberschatz, Henry F. Korth, S. Sudarshan - *Database System Concepts Third Edition*, The McGraw-Hill Companies, Inc.
- [21] Java™ 2 Platform - *Standard Edition, v 1.4.1 API Specification*, <http://java.sun.com/j2se/1.4.1/docs/api/>
- [22] Cay Horstmann, Gary Cornell – *Core Java 2, Volume II-Advanced Features*, Sun Microsystems, 2002.
- [23] Deitel & Deitel - *Java How to Program, Introducing Swing*, PRENTICE HALL, 1999.
- [24] Deborah J. Mahew - *Principles and Guidelines in Software User Interface Design*, Prentice Hall PTR-Englewood, New Jersey 07632, 1992.
- [25] Martin Fowler, Kendall Scott – *UML Distilled Applying the Standard Object Modeling Language*, ADDISON-WESLEY, 1997.
- [26] David C. Fallside (IBM) - *XML Schema Part 0: Primer*, <http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/>, May 2001.
- [27] Bary Smyth, Paul Cotter: *Surfing the Digital Wave*, ICCBR 1999: 561-571
- [28] Patrick Baudisch, Lars Brueckner: *TV Scout: Lowering the entry barrier to personalized TV program recommendation*, Proceedings of AH2002.
- [29] Ann Buczak, John Zimmerman, Kaushal Kurapati: *Personalization: Improving Ease-of-Use, Trust and Accuracy of a TV Show Recommender* Proceedings of the AHB'2002 Workshop
- [30] Samuli Niiranen, Artur Lugmayr, Seppo Kalli: *Agent-Based Personalization in Digital Television*, Proceedings of the NORSIG2002.
- [31] Νεκτάριος Μουμουτζής *Σχεδιασμός Συστήματος Ανάπτυξης Αλληλεπιδραστικών Γεωγραφικών Εφαρμογών* Πολυτεχνείο Κρήτης, 1998
- [32] Νεκτάριος Μουμουτζής: *Mechanisms for Intelligent Content Filtering in the UP-TV System*. MUSIC/ Πολυτεχνείο Κρήτης, 2003.
- [33] Τσιναράκη, Φατούρου, Χριστοδουλάκης: *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
- [34] Καζάσης, Μουμουτζής, Παππάς, Καραναστάση, Χριστοδουλάκης: *Designing Ubiquitous Personalized TV-Anytime Services* In the proceedings of the International Workshop on Ubiquitous Mobile Information and Collaboration Systems (UMICS), Klagenfurt/Velden, Austria, June 2003
- [35] Παππάς, Καζάσης, Μουμουτζής, Τσιναράκη, Χριστοδουλάκης: *Personalized and ubiquitous information services for TV programs*, In the proceedings of the International Workshop on Multimedia in Digital Libraries (MDL), Greece, June 2003

- [36] Τσιναράκη, Φατούρου, Χριστοδουλάκης: *An Interoperability Framework for the Management of Semantic Metadata in order to support Personalized TV Services*, In the proceedings of the 2nd Hellenic Data Management Symposium (HDMS '03), Greece, September 2003
- [37] Τσιναράκη, Πολύδωρος, Καζάσης, Χριστοδουλάκης: *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.
- [38] Μουμουτζής, Καζάσης, Παππάς: *Generic Architecture for handling TV/A Metadata using relational database technologies*, Technical Report, TUC/MUSIC, 2002
- [39] Καζάσης, Μουμουτζής, Παππάς: *Informative Annex: Generic Architecture for handling TV/A Metadata using relational database technologies*, contribution TV-Anytime Forum/Metadata WG, 2nd Implementer's workshop, Geneva July 2002
- [40] Αναστασία Καραναστάση – Σχεδιασμός και Υλοποίηση Μοντέλου Φυσικής Γλώσσας για την Διαχείριση TV-Anytime Πληροφοριών σε Περιβάλλον Ψηφιακής Τηλεόρασης και Κινητών Συσκευών, Πολυτεχνείο Κρήτης, 2003

