



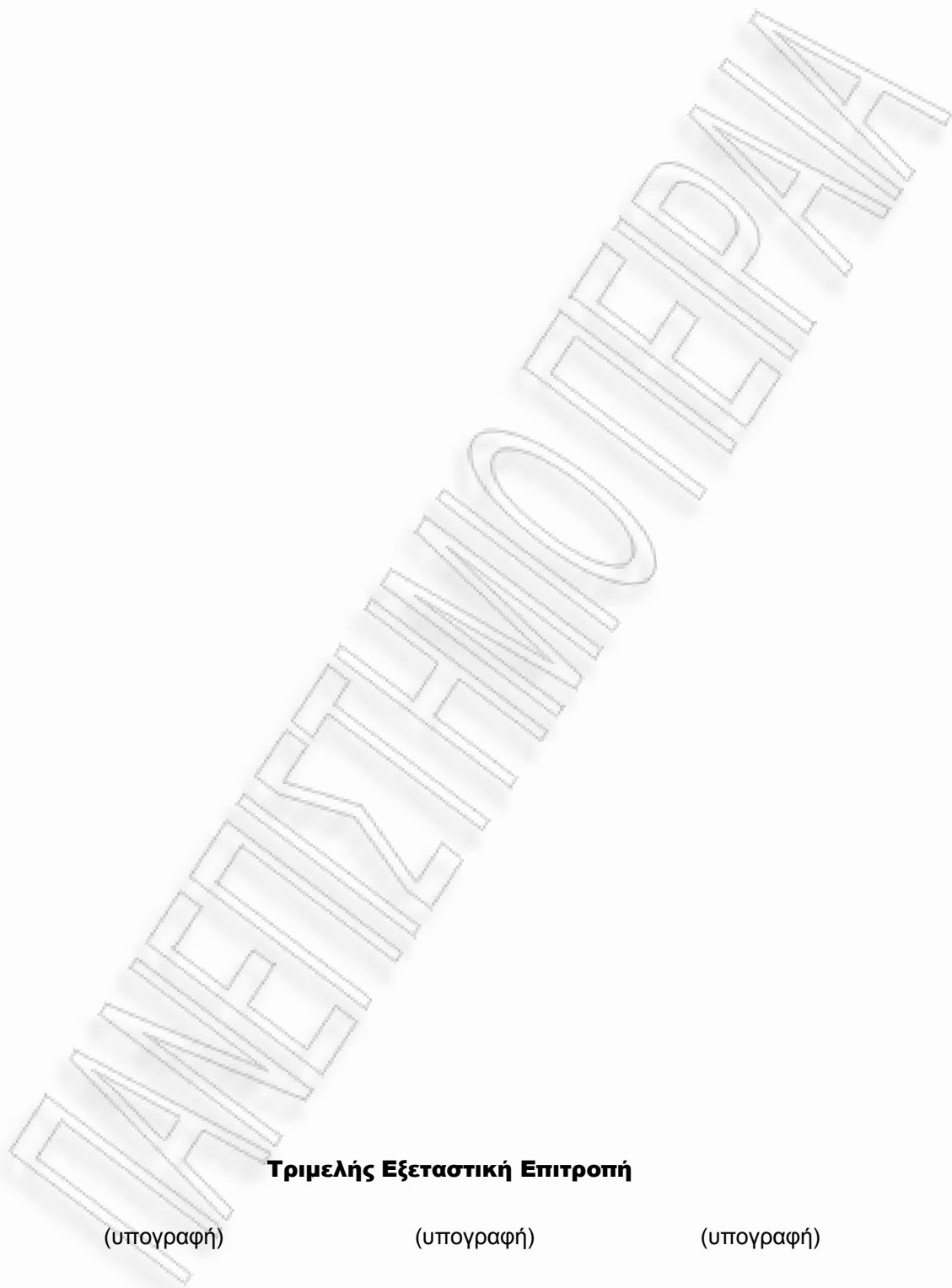
Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής  
Πρόγραμμα Μεταπτυχιακών Σπουδών  
«Πληροφορική»

**Μεταπτυχιακή Διατριβή**

Τίτλος Διατριβής	<b>Εξατομικευμένο Σύστημα Συστάσεων</b>
Όνοματεπώνυμο Φοιτητή	<b>Ιωάννης Μαρκόγιαννης</b>
Πατρώνυμο	<b>Γεώργιος</b>
Αριθμός Μητρώου	<b>ΜΠΠΛ/ 09029</b>
Επιβλέπων	<b>Μαρία Βίρβου, Καθηγήτρια</b>

Ημερομηνία Παράδοσης **Νοέμβριος 2011**

---



**Τριμελής Εξεταστική Επιτροπή**

(υπογραφή)

(υπογραφή)

(υπογραφή)

Μαρία Βίρβου  
Καθηγήτρια

Γεώργιος Τσιχριντζής  
Καθηγητής

Ευάγγελος Φούντας  
Καθηγητής

## ΠΕΡΙΛΗΨΗ

Σκοπός αυτής της διπλωματικής εργασίας είναι ο σχεδιασμός και η ανάπτυξη ενός επεκτάσιμου πυρήνα λογισμικού για την παροχή εξατομικευμένων συστάσεων, πληροφοριών και στοιχείων βοήθειας ενσωματώνοντας αλγορίθμους εξατομίκευσης, εξόρυξης γνώσης και εμπλουτισμού των μεταδεδομένων. Στα πλαίσια παρουσίασης του πυρήνα υλοποιήθηκε ένα προηγμένο πληροφοριακό σύστημα διαχείρισης πολυμέσων, το οποίο συνδυάζει τα χαρακτηριστικά - προφίλ του χρήστη με τη συμπεριφορά, τις ενέργειες του και το ιστορικό των συναλλαγών του έτσι ώστε να παρέχει υπηρεσίες προστιθέμενης αξίας σε αυτόν.

Οι χρήστες και οι φορείς παροχής των πολυμέσων σήμερα αντιμετωπίζουν ένα κοινό πρόβλημα: πώς να αντιμετωπίσουν την " κρίση της επιλογής " που υπάρχει κατά τη στιγμή όπου πάρα πολλές διαφορετικές μορφές πολυμέσων παρουσιάζονται σε αυτούς. Υπό αυτές τις συνθήκες η αναζήτηση δεν είναι μια ολοκληρωμένη λύση και η παροχή συστινόμενου υλικού μπορεί να βελτιώσει την εμπειρία του χρήστη.

Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκε το περιβάλλον προγραμματισμού Visual Studio .Net 2010, η γλώσσα προγραμματισμού C# και το σύστημα διαχείρισης βάσεων δεδομένων SQL Server 2008. Οι διαδικτυακές υπηρεσίες που χρησιμοποιήθηκαν προσδιορίστηκαν ως SOAP/XML web services.

Λέξεις κλειδιά: Εξατομίκευση, Σύστημα Συστάσεων, Ευφυές Σύστημα Βοήθειας, Πολυμέσα, Εξόρυξη γνώσης, Κατηγοριοποίηση, Αλγόριθμος Συσχετίσεων

**ABSTRACT**

The purpose of this thesis is to design and develop an extensible software framework to provide personalized recommendations, information and assistance integrating personalization algorithms, data mining and enrichment of metadata. Within the presentation of the framework implemented an advanced multimedia information management system, which combines features - profile of user behavior, actions and history of transactions to provide value added services to him.

Users and providers of multimedia today face a common problem: how to deal the "crisis of choice" which exists at a time when so many different forms of media presented to them. Under these conditions, the search is not a complete solution and providing recommended material can improve the user experience. To implement the application we used application development platform Visual Studio. Net 2010, the C # programming language and database management system SQL Server 2008. The web services used were identified as SOAP/XML Lweb services.

Keywords: Personalization, Recommendation System, Intelligent Help System, Media, Data Mining, Classification, Association Algorithms

**ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ**

<b>ΠΕΡΙΛΗΨΗ</b> .....	<b>3</b>
<b>ABSTRACT</b> .....	<b>4</b>
<b>ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ</b> .....	<b>5</b>
<b>ΠΙΝΑΚΑΣ ΣΧΗΜΑΤΩΝ</b> .....	<b>7</b>
<b>1 ΕΙΣΑΓΩΓΗ</b> .....	<b>9</b>
1.1 Αντικείμενο της διπλωματικής.....	9
1.2 Οργάνωση του τόμου.....	9
<b>2 ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΧΡΗΣΤΩΝ ΚΑΙ ΕΥΦΥΗ ΣΥΣΤΗΜΑΤΑ ΒΟΗΘΕΙΑΣ</b> .....	<b>11</b>
2.1 Ευφυή Συστήματα Βοήθειας και Προσαρμοστικές Διεπαφές.....	11
2.2 Software Agents.....	12
2.3 Software Intelligent Agents.....	13
2.4 Interface Agents.....	14
<b>3 ΣΥΣΤΗΜΑΤΑ ΣΥΣΤΑΣΕΩΝ</b> .....	<b>15</b>
3.1 Δομή του Συστήματος Συστάσεων.....	16
3.1.1 Η είσοδος.....	16
3.1.2 Η έξοδος.....	17
3.1.3 Η μέθοδος.....	18
3.1.4 Ο βαθμός εξατομίκευσης.....	19
3.2 Η Προτεινόμενη Προσέγγιση.....	21
<b>4 ΑΝΑΛΥΣΗ ΚΑΙ ΣΧΕΔΙΑΣΗ</b> .....	<b>22</b>
4.1 Εισαγωγή.....	22
4.1.1 UML (Unified Modeling Language).....	22
4.1.2 RUP (Rational Unified Process).....	22
4.2 Έγγραφο Προδιαγραφών Απαιτήσεων από το Λογισμικό.....	22
4.2.1 Σκοπός.....	22
4.2.2 Γενική περιγραφή.....	23
4.2.3 Απαιτήσεις.....	24
4.2.4 Ανάλυση περιπτώσεων χρήσης.....	25
4.3 Έγγραφο περιγραφής αρχιτεκτονικού σχεδίου.....	34
4.3.1 Εισαγωγή.....	34
4.3.2 Σκοπός.....	34
4.3.3 Ενδιαφερόμενοι.....	34
4.3.4 Αναφορές.....	35
4.3.5 Αρχιτεκτονικές αποφάσεις.....	35
4.4 Έγγραφο περιγραφής του λεπτομερούς σχεδίου.....	37
4.4.1 Σκοπός.....	37
4.4.2 Ενδιαφερόμενοι.....	37
4.4.3 Βιβλιογραφία.....	37
4.4.4 Σχεδιαστικές όψεις.....	37
4.4.5 Σχεδιασμός της βάσης δεδομένων.....	41
4.5 Εξατομικευμένο σύστημα συστάσεων.....	46
4.5.1 Αλγόριθμοι και μοντέλα εξόρυξης δεδομένων.....	46

4.6	Προσαρμοστικές Διεπαφές .....	66
4.6.1	Αλλαγή Theme .....	68
4.6.2	Προσαρμογή Οθόνης .....	69
4.7	Ευφύες Σύστημα Βοήθειας .....	70
4.7.1	Microsoft Agent Core Components .....	70
4.7.2	Υλοποίηση Ευφούς Συστήματος Βοήθειας.....	72
<b>5</b>	<b>ΥΛΟΠΟΙΗΣΗ.....</b>	<b>83</b>
5.1	Πλατφόρμες και προγραμματιστικά εργαλεία.....	83
5.1.1	Το .Net Framework.....	83
5.1.2	Common Language Runtime.....	83
5.1.3	Η Βιβλιοθήκη κλάσεων του .NET Framework.....	85
5.2	Σύστημα Διαχείρισης Βάσεων Δεδομένων.....	86
5.3	Οδηγός Εγκατάστασης .....	86
5.3.1	Απαιτήσεις Συστήματος.....	86
5.3.2	Εγκατάσταση .....	86
<b>6</b>	<b>ΕΠΙΛΟΓΟΣ .....</b>	<b>92</b>
6.1	Σύνοψη - Σύγκριση και Συμπεράσματα .....	92
6.2	Μελλοντικές επεκτάσεις .....	92
<b>7</b>	<b>ΒΙΒΛΙΟΓΡΑΦΙΑ.....</b>	<b>94</b>

**ΠΙΝΑΚΑΣ ΣΧΗΜΑΤΩΝ**

Εικόνα 1 Χαρακτηριστικά ενός intelligent agent .....	13
Εικόνα 2 Δομή του Συστήματος Συστάσεων .....	16
Εικόνα 3 Ταξινόμηση των τεχνικών RSs .....	20
Εικόνα 4 Use case diagram .....	24
Εικόνα 5 Περίπτωση χρήσης Register .....	25
Εικόνα 6 Activity Diagram – Εγγραφή και Είσοδος στο σύστημα .....	26
Εικόνα 7 Profile Information .....	27
Εικόνα 8 Movies Shop .....	28
Εικόνα 9 Activity Diagram – Movies Shop .....	29
Εικόνα 10 Movie Rentals .....	29
Εικόνα 11 Movies .....	30
Εικόνα 12 Customers .....	30
Εικόνα 13 Movies KPI .....	31
Εικόνα 14 Customers KPI .....	31
Εικόνα 15 Top Movies .....	32
Εικόνα 16 Top Customers .....	32
Εικόνα 17 Model Viewer .....	33
Εικόνα 18 New Model .....	34
Εικόνα 19 Deployment Diagram .....	36
Εικόνα 20 User Interface User Role .....	36
Εικόνα 21 Αποσυνθετική όψη του UI του συστήματος Smart Media (επιπεδο 0) .....	38
Εικόνα 22 Αποσυνθετική όψη του UI του συστήματος Smart Media (επιπεδο 1) .....	39
Εικόνα 23 Αποσυνθετική όψη του DAL του συστήματος Smart Media (επιπεδο 0) .....	39
Εικόνα 24 Αποσυνθετική όψη του DAL του συστήματος Smart Media (επιπεδο 1) .....	40
Εικόνα 25 Αποσυνθετική όψη του BLL του συστήματος Smart Media .....	40
Εικόνα 26 Αποσυνθετική όψη του Framework του συστήματος Smart Media .....	41
Εικόνα 27 Διάγραμμα Οντοτήτων-Συσχέισων ER .....	42
Εικόνα 28 Πίνακας «Actors» - Οντότητα .....	42
Εικόνα 29 Πίνακας «Shopping Basket» - Οντότητα .....	43
Εικόνα 30 Πίνακας «Channels» - Οντότητα .....	43
Εικόνα 31 Πίνακας «Criteria» - Οντότητα .....	43
Εικόνα 32 Πίνακας «Customers» - Οντότητα .....	43
Εικόνα 33 Πίνακας «Directors» - Οντότητα .....	44
Εικόνα 34 Πίνακας «Hobbies» - Οντότητα .....	44
Εικόνα 35 Πίνακας «Genres» - Οντότητα .....	44
Εικόνα 36 Πίνακας «Movies» - Οντότητα .....	44
Εικόνα 37 Πίνακας «Technologies» - Οντότητα .....	44
Εικόνα 38 Σχισιακό Σχήμα .....	45
Εικόνα 39 Recommended Movies By Demographics .....	49
Εικόνα 40 Male User .....	50
Εικόνα 41 Recommended Movies By Gender for John .....	51
Εικόνα 42 Female User .....	51
Εικόνα 43 Recommended Movies By Gender for Maria .....	52

Εικόνα 44 Microsoft Association Rules .....	56
Εικόνα 45 Recommended Movies By Demographics & Transactions History .....	58
Εικόνα 46 Male User Profile .....	58
Εικόνα 47 Ιστορικό Συναλλαγών John .....	59
Εικόνα 48 Recommended Movies By Profile & Transactions for John.....	59
Εικόνα 49 Maria User Profile .....	60
Εικόνα 50 Maria Transactions History .....	60
Εικόνα 51 Recommended Movies By Profile & Transactions for Maria.....	61
Εικόνα 52 Related Movies.....	65
Εικόνα 53 Movie Associations .....	66
Εικόνα 54 Αλλαγή Theme Dark Style.....	68
Εικόνα 55 Αλλαγή Theme DevExpress Style .....	68
Εικόνα 56 Customize Layout.....	69
Εικόνα 57 Μετακίνηση Πεδίων.....	69
Εικόνα 58 Μετονομασία Πεδίων .....	70
Εικόνα 59 Agent Character Options.....	71
Εικόνα 60 Agent Character Editor .....	72
Εικόνα 61 Profile Information Help for New User .....	81
Εικόνα 62 Profile Information Help for Advanced User .....	82
Εικόνα 63 Movies Shop Screen Help.....	82
Εικόνα 64 .NET Framework.....	83
Εικόνα 65 Common language runtime.....	84
Εικόνα 66 .Net Framework Class Library.....	85
Εικόνα 67 Εγκατάσταση βήμα 1 .....	87
Εικόνα 68 Εγκατάσταση βήμα 2 .....	88
Εικόνα 69 Εγκατάσταση βήμα 3 .....	89
Εικόνα 70 Εγκατάσταση βήμα 4 .....	90
Εικόνα 71 Εγκατάσταση βήμα 5 .....	91



# 1 ΕΙΣΑΓΩΓΗ

## 1.1 Αντικείμενο της διπλωματικής

Σκοπός της παρούσας εργασίας είναι ο σχεδιασμός και η ανάπτυξη ενός επεκτάσιμου πυρήνα λογισμικού για την παροχή εξατομικευμένων συστάσεων, πληροφοριών και στοιχείων βοήθειας ενσωματώνοντας αλγορίθμους εξατομίκευσης, εξόρυξης γνώσης και εμπλουτισμού των μεταδεδομένων. Στα πλαίσια παρουσίασης του πυρήνα υλοποιήθηκε ένα προηγμένο πληροφοριακό σύστημα διαχείρισης πολυμέσων, το οποίο συνδυάζει τα χαρακτηριστικά - προφίλ του χρήστη με τη συμπεριφορά, τις ενέργειες του και το ιστορικό των συναλλαγών του έτσι ώστε να παρέχει υπηρεσίες προστιθέμενης αξίας σε αυτόν.

Οι χρήστες και οι φορείς παροχής των πολυμέσων σήμερα αντιμετωπίζουν ένα κοινό πρόβλημα: πώς να αντιμετωπίσουν την "κρίση της επιλογής" που υπάρχει κατά τη στιγμή όπου πάρα πολλές διαφορετικές μορφές πολυμέσων παρουσιάζονται σε αυτούς. Υπό αυτές τις συνθήκες η αναζήτηση δεν είναι μια ολοκληρωμένη λύση και η παροχή συστινόμενου υλικού μπορεί να βελτιώσει την εμπειρία του χρήστη.

Αντίστοιχα οι προγραμματιστές των εφαρμογών πολυμέσων σήμερα βρίσκονται αντιμέτωποι με το δικό τους κοινό πρόβλημα: την ανάγκη να επεξεργαστούν και να παρουσιάσουν πολλές διαφορετικές μορφές πολυμέσων. Για τον προγραμματιστή όπου επεξεργάζεται τα πολυμέσα και διαχειρίζεται το λογισμικό, αυτό σημαίνει απαιτήσεις όπως ευελιξία και επεκτασιμότητα σε βαθμό όσο το δυνατόν περισσότερο καθώς είναι δύσκολο να επιτευχθεί λόγω της αδύνατης πρόβλεψης των τάσεων αλλαγών στα πολυμέσα.

Τη λύση στα δύο παραπάνω προβλήματα έρχεται να δώσει το framework που υλοποιήσαμε καλύπτοντας τόσο τις ανάγκες των χρηστών για εξατομικευμένο και στοχευμένο περιεχόμενο και των προγραμματιστών για την εφαρμογή των υλοποιημένων αλγορίθμων και μεθόδων του framework σε οποιοδήποτε πληροφοριακό σύστημα ή μοντέλο παρουσίασης και διαχείρισης πολυμέσων. Στη πραγματικότητα πρόκειται για ένα δυναμικό σύστημα δημιουργίας μοντέλων εξατομίκευσης, παραμετροποίησής τους και εφαρμογής τους σε οποιοδήποτε σύνολο δεδομένων. Το framework έχει υλοποιηθεί ώστε να παρέχει στοιχεία εξατομίκευσης διατηρώντας την ευελιξία του και απόδοση με σκόπο τη δυναμική δημιουργία διαφορετικών σεναρίων εξατομίκευσης. Παράλληλα έχει ενσωματωθεί ένα ευφυές σύστημα βοήθειας με τη δημιουργία ενός αλγορίθμου ο οποίος προσδιορίζει από τις ενέργειες και το ιστορικό του χρήστη το βαθμό εμπειρίας του, τον κατηγοριοποιεί και παρουσιάζεται ανάλογη σε λεπτομέρεια πληροφορία βοήθειας διαχείρισης ενός συστήματος.

## 1.2 Οργάνωση του τόμου

Ο τόμος της διπλωματικής εργασίας αποτελείται από 7 κεφάλαια. Στα κεφάλαια αυτά γίνεται αναλυτική περιγραφή της διαδικασίας η οποία ακολουθήθηκε κατά την εκπόνηση της εργασίας. Συγκεκριμένα, περιγράφονται έννοιες σχετικές με το θέμα, όπως για παράδειγμα οι αλγόριθμοι εξατομίκευσης. Επιπλέον, φαίνεται η θεωρητική μελέτη η οποία έγινε, παρουσιάζονται τα συμπεράσματα στα οποία φθάσαμε και περιγράφεται ο τρόπος υλοποίησης του συστήματος.

Στο 1<sup>ο</sup> κεφάλαιο γίνεται μια αναφορά στο αντικείμενο της διπλωματικής εργασίας και ακολουθεί μια συνοπτική παρουσίαση της οργάνωσης αυτού του τόμου ώστε ο αναγνώστης να αποκτήσει μια γενική εικόνα της διπλωματικής εργασίας.

Το 2<sup>ο</sup> κεφάλαιο αρχικά περιγράφει συνοπτικά την μοντελοποίηση χρηστών και στη συνέχεια αναφέρεται στις προσαρμοστικές διεπαφές και στα ευφυή συστήματα βοήθειας. Εκτενέστερη αναφορά γίνεται για τους intelligent software agents καθώς έχουν ενσωματωθεί και στο προτεινόμενο framework.

Το 3<sup>ο</sup> κεφάλαιο περιέχει τη μελέτη των συστημάτων συστάσεων. Αναλύει και εξηγεί τα χαρακτηριστικά τους, τις δυνατότητες τους, τις παραμέτρους τους και τη δομή τους επιμερώς. Γίνεται αναφορά στο βαθμό εξατομίκευσης που επιτυγχάνεται αλλά και στα κριτήρια τα οποία

τον διαμορφώνουν. Στο τέλος του κεφαλαίου παρουσιάζεται συνοπτικά η προτεινόμενη προσέγγιση που ακολουθηθηκε στα πλαίσια της μεταπτυχιακής διατριβής.

Το 4<sup>ο</sup> κεφάλαιο αναφέρεται στην ανάλυση και το σχεδιασμό του συστήματος. Έχοντας πλέον μελετήσει θεωρητικά το θέμα, μένει να γίνει σχεδιασμός του συστήματος, να προσδιοριστούν, δηλαδή, οι απαιτήσεις, να δοθεί η αρχιτεκτονική και να γίνει λεπτομερής σχεδίαση του συστήματος. Το κεφάλαιο αυτό, λοιπόν, αρχικά εισάγει τον αναγνώστη στις μεθοδολογίες ανάπτυξης λογισμικού που χρησιμοποιήθηκαν και στη συνέχεια παρατίθενται τα έγγραφα της τεχνολογίας λογισμικού. Συγκεκριμένα περιλαμβάνεται το «Έγγραφο Προδιαγραφών των απαιτήσεων από το Λογισμικό Σύστημα», το «Έγγραφο Περιγραφής της Αρχιτεκτονικής του Συστήματος» και το «Έγγραφο Περιγραφής της Λεπτομερούς Σχεδίασης του Συστήματος».

Το 5<sup>ο</sup> κεφάλαιο παρουσιάζει την υλοποίηση της εφαρμογής. Αρχικά, περιγράφονται συνοπτικά οι πλατφόρμες και τα προγραμματιστικά εργαλεία τα οποία χρησιμοποιήθηκαν ενώ στη συνέχεια φαίνονται οι λεπτομέρειες της υλοποίησης με την περιγραφή της δομής του κώδικα μέσω των κυριότερων πακέτων (packages). Τέλος, παρέχονται οδηγίες για την εγκατάσταση του συστήματος.

Το 6<sup>ο</sup> κεφάλαιο αποτελεί μια σύνοψη της παρούσας διπλωματικής εργασίας και παρατίθενται συμπεράσματα που προέκυψαν. Επιπλέον, αναφέρονται πιθανές μελλοντικές επεκτάσεις του συστήματος που υλοποιήθηκε.

Το 7<sup>ο</sup> κεφάλαιο περιλαμβάνει μια παρουσίαση της βιβλιογραφίας η οποία χρησιμοποιήθηκε κατά την εκπόνηση της εργασίας.

## 2 ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΧΡΗΣΤΩΝ ΚΑΙ ΕΥΦΥΗ ΣΥΣΤΗΜΑΤΑ ΒΟΗΘΕΙΑΣ

Η περιοχή της μοντελοποίησης χρηστών, σχετίζεται με τρεις ευρείες κατηγορίες προβλημάτων:

- Πώς μπορούν να αποκτηθούν οι πληροφορίες που αφορούν στον χρήστη.
- Τί είδη πληροφοριών μπορούν να αποκτηθούν.
- Πώς μπορούν να χρησιμοποιηθούν οι πληροφορίες που αποκτήθηκαν, για να βελτιώσουν την επίδοση του συστήματος.

Οι πληροφορίες γύρω από τον χρήστη, μπορεί να αποκτηθούν άμεσα ή μπορεί να βγουν συμπερασματικά ή και με τους δυο αυτούς τρόπους. Συνήθως υπάρχει ένας μηχανισμός που αποκτά πραγματικές γνώσεις γύρω από το χρήστη και μετά ένας μηχανισμός εξαγωγής συμπεράσματος που γεννάει υποθέσεις για το χρήστη βασιζόμενος όμως στις πραγματικές γνώσεις που αποκτήθηκαν. Το είδος των πληροφοριών που αποκτήθηκαν σχετικά με τον χρήστη, μπορεί να διαφέρει πολύ ανάλογα με τον βαθμό λεπτομέρειας που έχει το μοντέλο του χρήστη. Για παράδειγμα, το μοντέλο χρήστη μπορεί να κωδικοποιεί (πληροφορίες) για μια ευρεία τάξη 'τυπικών' χρηστών ή μπορεί (το μοντέλο χρήστη) να κωδικοποιεί πληροφορίες για ένα συγκεκριμένο χρήστη, που βασίζονται στην παρατήρηση της συμπεριφοράς του. Οι πληροφορίες για το χρήστη, που πρέπει να μοντελοποιηθούν και ο τρόπος απόκτησης αυτών των πληροφοριών, μπορεί να αποτελούν ένα σύνολο καθοριστικών χαρακτηριστικών για το μοντέλο του χρήστη.

Από τη στιγμή που έχει κατασκευαστεί, ένα μοντέλο χρήστη πρέπει να χρησιμοποιείται από το σύστημα για να βελτιώσει την αλληλεπιδραστική συμπεριφορά του. Ο τρόπος με τον οποίο ένα μοντέλο χρήστη μπορεί να χρησιμοποιηθεί εξαρτάται από τις ανάγκες της εφαρμογής. Για παράδειγμα, σε μια εκπαιδευτική εφαρμογή το μοντέλο χρήστη μπορεί να χρησιμεύει για να αντιπροσωπεύει τις γνώσεις του χρήστη για το διδαχθέν θέμα. Ως εκ τούτου, το σύστημα μπορεί να χρησιμοποιεί αυτές τις πληροφορίες για να προσαρμόζει τις στρατηγικές διδασκαλίας του στο επίπεδο γνώσεων του μαθητή. Από την άλλη μεριά σε ένα σύστημα ανάκτησης πληροφοριών το μοντέλο χρήστη μπορεί να καταγράφει τα ενδιαφέροντα του χρήστη για διάφορες περιοχές και μπορεί να χρησιμοποιηθεί για να τον βοηθήσει να ανακτήσει τις πληροφορίες για τις οποίες ενδιαφέρεται περισσότερο. Ανάλογα με την εφαρμογή, ο όρος «μοντέλο χρήστη» μπορεί επίσης να αλλάζει ελαφρά το όνομα του. Για παράδειγμα, συχνά βρίσκουμε τον όρο «μοντέλο\_\_ μαθητή» ή «μοντέλο πράκτορας». Οι μαθητές είναι ένας τύπος χρήστη και οι χρήστες είναι ένας τύπος πράκτορα.

### 2.1 Ευφυή Συστήματα Βοήθειας και Προσαρμοστικές Διεπαφές

Σε οποιοδήποτε μεγάλο ή σύνθετο σύστημα υπολογιστή, οι χρήστες είναι συνήθως εξοικειωμένοι με κάποιο υποσύνολο της διατιθέμενης λειτουργικότητας, δείχνοντας πείρα σε κάποιες εφαρμογές και μη έχοντας πείρα σε άλλες. Επιπρόσθετα, διαφορετικοί χρήστες θα έχουν διαφορετικές ανάγκες και επίπεδα κατανόησης. Τα Ευφυή Συστήματα Βοήθειας (Intelligent Help Systems) προσπαθούν να αντιμετωπίσουν αυτά τα προβλήματα με την προσαρμογή της βοήθειας, στον κάθε ξεχωριστό χρήστη ο οποίος την ζητάει, και με το να προτείνουν ενεργά, εναλλακτικές πορείες δράσης για τις οποίες ο χρήστης μπορεί να μην είναι ενήμερος (Dix et al., 1993).

Τα συστήματα ΕΣΒ και ΕΣΔ έχουν παρά πολλά κοινά στοιχεία σχετικά με την αρχιτεκτονική και την λειτουργία τους. Αμφότερα ενδιαφέρονται για την ευφυή επικοινωνία μεταξύ ενός υπολογιστή και ενός χρήστη, έχουν την ίδια αρχιτεκτονική και βασίζονται κατά πολύ στο τμήμα της μοντελοποίησης χρηστών για να παράγουν συμβουλές για χρήστες και να προσαρμόσουν τη λειτουργία τους σ' αυτούς. Ένα μοντέλο χρήστη σε ένα ΕΣΒ γενικά αναφέρεται στην αναπαράσταση της γνώσης του χρήστη και / ή στις πεποιθήσεις του για το πεδίο γνώσης.

Η αναγνώριση και η επεξήγηση παρανοήσεων και λαθών ενός χρήστη ήταν πάντα ένα μεγάλο θέμα για την μοντελοποίηση χρηστών στα ΕΣΒ. Το μοντέλο χρήστη μπορεί επίσης να παράγει προβλέψεις για το χρήστη. Όπως επισημαίνουν κάποιοι ερευνητές, προβλέψεις για τα πιθανά συμπεράσματα του χρήστη, μπορεί να χρησιμοποιηθούν για να μεταφέρουν έμμεσα πληροφορίες.

Ένα ΕΣΒ μπορεί να θεωρηθεί σαν μια ειδική περίπτωση των προσαρμοστικών διεπαφών του χρήστη. Όπως επισημαίνουν ερευνητές από το χώρο Της Αλληλεπίδρασης Ανθρώπου- Υπολογιστή, ότι προσαρμοστικότητα είναι η αυτόματη προσαρμογή της διεπαφής του χρήστη, από το σύστημα. Οι αποφάσεις για προσαρμογή μπορεί να βασίζονται στην πείρα του χρήστη ή στην παρατηρηθείσα επανάληψη συγκεκριμένων εργασιών. Σημειώνουν επίσης ότι η διαφορά μεταξύ *προσαρμοστικότητας* και *προσαρμοσιμότητας* είναι στο γεγονός ότι στην προσαρμοσιμότητα ο χρήστης παίζει έναν άμεσο ρόλο στην ικανότητα προσαρμογής ενώ στην προσαρμοστικότητα ο ρόλος του είναι περισσότερο έμμεσος. Το προσαρμοστικό λογισμικό παρέχει στους χρήστες την δυνατότητα να το προσαρμόζουν στις ανάγκες τους με δική τους πρωτοβουλία και ευθύνη για την εκτέλεση της προσαρμογής. Αντίθετα το προσαρμοστικό λογισμικό προσαρμόζεται αυτόματα στις ανάγκες ενός χρήστη με πρωτοβουλία του ίδιου του συστήματος και δική του ευθύνη για την εκτέλεση της προσαρμογής.

Ένα προσαρμοστικό σύστημα, μπορεί να εκπαιδευτεί για να αναγνωρίζει τη συμπεριφορά ενός έμπειρου ή αρχάριου και ακολούθως να ρυθμίζει τον έλεγχο του διάλογου του ή το σύστημα βοήθειας του αυτόματα, για να αντιμετωπίσει τις ανάγκες του παρόντα χρήστη. Ανάλογα με το είδος της διεπαφής του χρήστη, η προσαρμοστικότητα μπορεί να αναφέρεται σε συντομεύσεις (shortcuts, στα GUIs) ή στην πλοήγηση σε συστήματα υπερκείμενων κ.τ.λ.

## 2.2 Software Agents

Οι software agents χρονολογούνται πίσω στις πρώτες μέρες εργασίας της Τεχνητής Νοημοσύνης (Artificial Intelligence) και συγκεκριμένα στο μοντέλο του Carl Hewitt 'actor'. Σ' αυτό το μοντέλο, ο Hewitt πρότεινε την ιδέα ενός αντικειμένου ανεξάρτητου, αλληλεπιδρών διαλογικού και το οποίο δρούσε παράλληλα με άλλα αντικείμενα. Αυτό το αντικείμενο είχε ενσωματωμένο ένα είδος εσωτερικής κατάστασης και μπορούσε να απαντήσει σε μηνύματα που προερχόταν από άλλα παρόμοια αντικείμενα.

Στις μέρες μας, οι agents είναι αντικείμενο μελέτης πολλών ομάδων ερευνητών και η τεχνολογία τους αναπτύσσεται ραγδαία. Ωστόσο λόγω της πολυμορφίας και της διαφορετικότητας των σημερινών agents, όσον αφορά τη χρήση και τη σχεδίαση τους, είναι σχεδόν αδύνατη η ύπαρξη ενός κοινά αποδεκτού ορισμού για την έννοια αυτή. Ενδεικτικά αναφέρονται δύο ορισμοί:

- "Agent: Ένα άτομο ή μια οντότητα η οποία δρα ή είναι ικανή να δράσει ή εξουσιοδοτημένη να δράσει εκ μέρους κάποιου άλλου". (Ερμηνεία του λεξικού The Webster's New World Dictionary )
- "Agent: Μια υπολογιστική μονάδα η οποία πραγματοποιεί κάποια εξειδικευμένα καθήκοντα αυτόνομα".

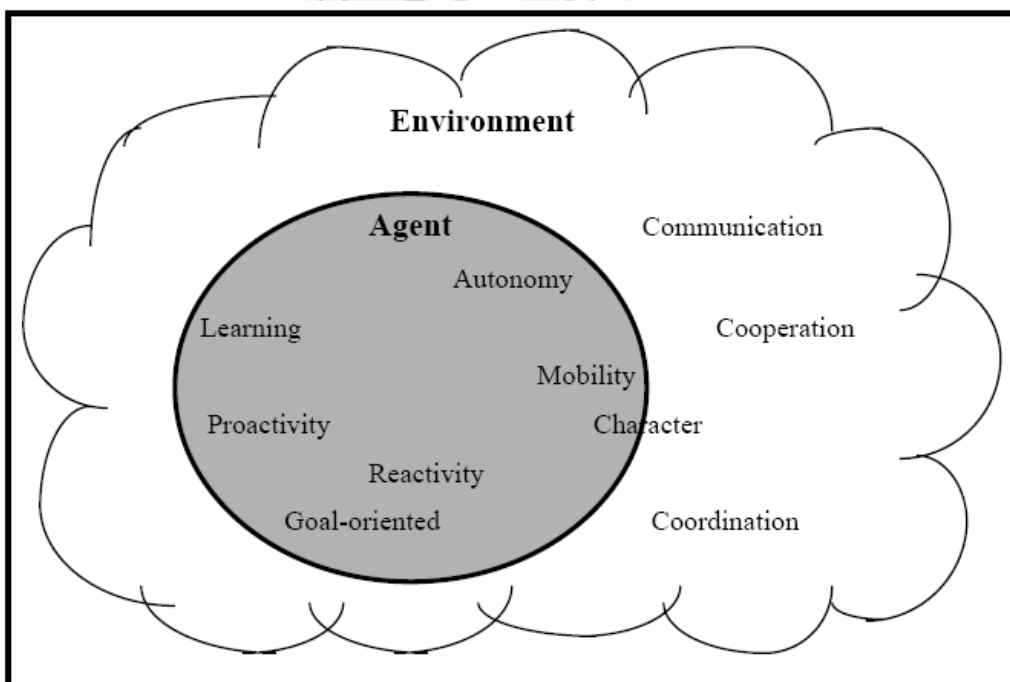
Τα βασικά χαρακτηριστικά ενός software agent είναι κυρίως τρία:

- **Αυτονομία (autonomy)**, με την έννοια ότι λειτουργεί ανεξάρτητα από τον χρήστη και ευθύνεται ο ίδιος για τον έλεγχο των πράξεων του.
- **Δυνατότητα αντίδρασης και προσαρμοστικότητας (reactive and adaptive)**, με την έννοια ότι αισθάνεται συνεχώς το εξωτερικό του περιβάλλον και προσαρμόζει συνεχώς την συμπεριφορά του σύμφωνα με τις αλλαγές που γίνονται σ' αυτό.
- **Δυνατότητα επικοινωνίας (communication ability)**, με την έννοια ότι συνεργάζεται με άλλες οντότητες είτε αυτές είναι κάποιιοι χρήστες, είτε είναι άλλοι agents, είτε βάσεις δεδομένων κ.α.

### 2.3 Software Intelligent Agents

Ένας agent τις περισσότερες φορές απαιτεί ένα ποσοστό ευφυΐας για να εκτελέσει μια εργασία. Έτσι, ένας software agent μπορεί να χαρακτηριστεί σαν **intelligent agent** αν δρα με λογική, δηλαδή χρησιμοποιεί τη γνώση, την πληροφορία και τη λογική για να πραγματοποιεί λογικές ενέργειες που αποσκοπούν στην εκπλήρωση ενός ή περισσότερων σκοπών. Ένας intelligent agent πρέπει να είναι ικανός να αναγνωρίζει κάποια συμβάντα, να καθορίζει την σημασία τους, και τότε να ενεργεί εκ μέρους του χρήστη.[1]

Για να κατανοήσει κανείς τις εργασίες και τις λειτουργίες των intelligent agents, είναι απαραίτητο να γνωρίσει και να κατανοήσει τις χαρακτηριστικές ιδιότητες τους που τους κάνουν να ξεχωρίζουν από τα απλά παραδοσιακά προγράμματα και τους απλούς agents. Τα χαρακτηριστικά ενός **intelligent agent** φαίνονται στην Εικόνα 1. Θα πρέπει να σημειωθεί ότι δεν είναι απαραίτητο κάθε intelligent agent να έχει όλες τις παρακάτω ιδιότητες. Μάλιστα, στην πράξη υπάρχουν πολλά συστήματα με διαφορετικό επίπεδο πολυπλοκότητας. Έτσι, απλοί agents πολύ μικρής πολυπλοκότητας μπορεί να έχουν μόνο ένα ή δύο από τα παρακάτω χαρακτηριστικά, ενώ πολύ υψηλού επιπέδου συστήματα agent στην ιδανική περίπτωση μπορεί να υποστηρίζουν (τουλάχιστον μερικώς) όλες τις παρακάτω ιδιότητες.



Εικόνα 1 Χαρακτηριστικά ενός intelligent agent

Τα χαρακτηριστικά των intelligent agents μπορούν να ταξινομηθούν σε δύο μεγάλες κατηγορίες των εσωτερικών και των εξωτερικών ιδιοτήτων. Οι εσωτερικές ιδιότητες είναι αυτές που σχηματίζουν τον 'εσωτερικό κόσμο' ενός agent, δηλαδή οι ιδιότητες που καθορίζουν τις ενέργειες που γίνονται μέσα στον agent. Οι εσωτερικές ιδιότητες περιλαμβάνουν την δυνατότητα μάθησης, την δυνατότητα αντίδρασης, την αυτονομία και τον προσανατολισμό σε ένα στόχο.[1]

Οι εξωτερικές δυνατότητες περιλαμβάνουν όλα εκείνα τα χαρακτηριστικά που επηρεάζουν την αλληλεπίδραση μεταξύ των agents ή την επικοινωνία μεταξύ agent και ανθρώπου. Για παράδειγμα, τέτοιες είναι δυνατότητες όπως η επικοινωνία ή η συνεργασία. Ωστόσο, δεν μπορούν όλα τα χαρακτηριστικά να ανήκουν μόνο σε ένα από τα δύο γκρουπ, μερικά μπορούν να ανήκουν και στα δύο γκρουπ. Μία τέτοια ιδιότητα είναι η προσωπικότητα ενός agent, της οποίας κάποια μέρη καθορίζουν την εσωτερική συμπεριφορά ενός agent αλλά επίσης παίζουν έναν σημαντικό ρόλο στην εξωτερική επικοινωνία.

## 2.4 Interface Agents

Οι Interface Agents εργάζονται ως προσωπικοί βοηθοί για να βοηθούν τον χρήστη στην πραγματοποίηση καθηκόντων. Οι Interface agents συνήθως χρησιμοποιούν την εκμάθηση για να προσαρμοστούν στις συνήθειες της δουλειάς και στις προτιμήσεις του χρήστη. Η Patty Maes (1994) του MIT αναγνωρίζει τέσσερις τρόπους εκμάθησης που μπορεί να συμβούν.

- Κατά πρώτον ένας agent μπορεί να μάθει παρακολουθώντας στενά τον χρήστη, προσέχοντας τις ενέργειες του και μιμούμενος τον χρήστη.
- Δεύτερον, ένας agent μπορεί να προσφέρει συμβουλές ή να ενεργεί για λογαριασμό του χρήστη και μετά να μαθαίνει από το αποτέλεσμα που λαμβάνει από τον χρήστη όσον αφορά την ικανοποίηση του ή όχι.
- Τρίτον, ένας agent μπορεί να δέχεται σαφείς οδηγίες από τον χρήστη για παράδειγμα αν γίνει αυτό, τότε κάνει εκείνο.
- Τέλος, με το να ζητά τις συμβουλές άλλων agents οπότε να μαθαίνει από την δική τους εμπειρία.

Θα πρέπει να σημειωθεί ότι οι interface agents συνεργάζονται κυρίως με τον χρήστη και όχι με άλλους agents (εξάιρεση αποτελεί η ερώτηση για συμβουλές). Χρησιμοποιώντας διάφορους μηχανισμούς εκμάθησης, οι interface agents προσαρμόζουν στις απαιτήσεις του χρήστη το interface ενός συστήματος υπολογιστή ή μιας σειράς εφαρμογών σύμφωνα με την ιδιαιτερότητα του χρήστη και το προσωπικό του στυλ εργασίας. Αν οι interface agents μπορούν να συνεργάζονται και να μοιράζονται την γνώση τους για το πως να επιτύχουν ένα έργο, τότε όταν ένα άτομο από μια ομάδα εργασίας ανακαλύψει πως πρέπει να ενεργήσει για να πετύχει κάτι, αυτή η γνώση μεταφέρεται και στους άλλους χρήστες της ομάδας εργασίας μέσω των δικών τους interface agents. Τότε το κέρδος όσον αφορά την παραγωγικότητα είναι μεγάλο.[1]

### 3 ΣΥΣΤΗΜΑΤΑ ΣΥΣΤΑΣΕΩΝ

Τα συστήματα συστάσεων (recommendations systems, recommender systems ή RSs) αποτελούν ίσως την πιο δημοφιλή μορφή εξατομίκευσης και τείνουν να μετατραπούν στις μέρες μας σε σημαντικά επιχειρησιακά εργαλεία. Αρχικά, εμφανίστηκαν στα μέσα της δεκαετίας του '90 και χρησιμοποιούνταν από λίγα web sites. Σήμερα δεν νοούνται εφαρμογές που να μην ενσωματώνουν τεχνολογίες RSs προκειμένου να βοηθήσουν τους πελάτες στις αγορές τους όπως: προτάσεις για βιβλία, CDs και άλλα προϊόντα στο Amazon.com, προτάσεις για κινηματογραφικές ταινίες στο MovieLens, κλπ., ενώ ακόμα και εταιρίες λογισμικού όπως η Microsoft ενσωμάτωσαν recommendation χαρακτηριστικά στα προϊόντα τους π.χ. στον Commerce Server. Με αυτό τον τρόπο, τα e-shops μπορούν να βελτιώσουν την απόδοσή τους καλύπτοντας τις ανάγκες και τις προτιμήσεις κάθε πελάτη ξεχωριστά, να αυξήσουν την ικανοποίησή του (satisfaction), να εδραιώσουν την e-εμπιστοσύνη (e-trust) και την e-πίστη του (e-loyalty) και να καθιερώσουν ένα-προς-ένα σχέσεις (1-1 relationships).

Τα συστήματα αυτά χρησιμοποιούν γνώση που παρέχεται είτε από ειδικούς ή που εξάγεται από την παρακολούθηση της συμπεριφοράς των πελατών για να οδηγήσουν τους τελευταίους στον εντοπισμό των προϊόντων-υπηρεσιών που τους ενδιαφέρουν. Οι προτάσεις μπορεί να βασίζονται: στις μεγαλύτερες συνολικές πωλήσεις του site, στα δημογραφικά δεδομένα του πελάτη, στην ανάλυση της προηγούμενης αγοραστικής συμπεριφοράς του σαν μια πρόβλεψη της μελλοντικής συμπεριφοράς του, σε δεδομένα από κοινότητες χρηστών (user communities), κλπ. Οι τύποι των προτάσεων περιλαμβάνουν: εξατομικευμένες προσφορές, τιμές, προϊόντα ή υπηρεσίες, προσθήκη ή παράλειψη θεματικών ενοτήτων ή παραγράφων, ταξινόμηση, κρύψιμο, εμφάνιση, προσθήκη, παράλειψη ή έμφαση συνδέσμων, προαιρετικές εξηγήσεις ή αναλυτικές πληροφορίες, σύνοψη απόψεων μιας κοινότητας, παροχή κριτικών μιας κοινότητας, κλπ. Η όλη διαδικασία μπορεί να υλοποιηθεί χειροκίνητα π.χ. συνδυασμός προϊόντων (cross-selling) ή αυτόματα από τον ηλεκτρονικό υπολογιστή π.χ. συσχέτιση πελατών με παρόμοια αγοραστική συμπεριφορά. [4]

Ένα RS ενισχύει τις πωλήσεις ενός καταστήματος e-εμπορίου με τους ακόλουθους τρόπους: μετατρέπει τους επισκέπτες σε αγοραστές, αυξάνει το cross-selling και το up-selling και «χτίζει» την εμπιστοσύνη και την πίστη των πελατών. Σε έναν κόσμο όπου οι ανταγωνιστές είναι μόνο ένα «κλικ» μακριά, το να κερδηθεί η εμπιστοσύνη των πελατών αποτελεί μια σημαντική επιχειρηματική στρατηγική. Τα RSs βελτιώνουν την εμπιστοσύνη αφού δημιουργούν μια ιδιαίτερη σχέση μεταξύ του πελάτη και του site. Τα sites που επενδύουν στο να μάθουν τους πελάτες τους, χρησιμοποιούν αυτά τα συστήματα ώστε να παρουσιάσουν interfaces που ταιριάζουν στις ανάγκες των χρηστών τους. Οι πελάτες από τη μεριά τους ανταποδίδουν αυτές τις ευκολίες με το να ξαναγυρίσουν στα sites που τους ικανοποιούν σε μεγαλύτερο βαθμό. Όσο περισσότερο ο πελάτης χρησιμοποιεί το σύστημα (το οποίο μαθαίνει τι αυτός χρειάζεται), τόσο μεγαλύτερη είναι και η εμπιστοσύνη του σε αυτό το site. Ακόμη, δίνοντας τη δυνατότητα δημιουργίας σχέσεων μεταξύ των πελατών π.χ. μέσω μιας κοινότητας χρηστών με κοινά χαρακτηριστικά και ενδιαφέροντα, αυξάνεται η πιθανότητα οι πελάτες να επιστρέψουν σε αυτό το site αφού το προτείνουν και άλλοι χρήστες με τους οποίους έχουν επικοινωνία.

Γενικά, τα RSs μπορούν να ταξινομηθούν σε τρεις βασικές κατηγορίες:

- **content based filtering (φιλτράρισμα με βάση το περιεχόμενο),**
- **collaborative filtering (συνεργατικό φιλτράρισμα) και**
- **hybrid approaches (υβριδικές προσεγγίσεις).**

Το content based filtering λαμβάνει υπόψη του τις παραμέτρους του χρήστη με τις πληροφορίες των προϊόντων (μεταδεδομένα, λέξεις κλειδιά, κλπ.), ενώ το collaborative filtering (CF) την «ομοιότητα» των χρηστών προκειμένου να παράγουν συστάσεις. Οι δύο αυτές μέθοδοι μπορούν να συνδυαστούν σε υβριδικές προσεγγίσεις.[3]

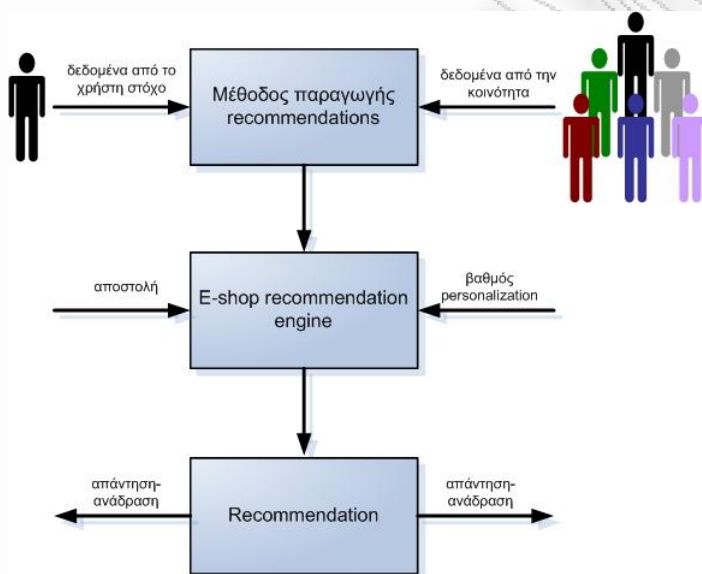
### 3.1 Δομή του Συστήματος Συστάσεων

Τα βασικά δομικά στοιχεία ενός RS είναι η **είσοδος**, η **έξοδος** και η **μέθοδος παραγωγής των προτάσεων**. Επιπρόσθετα, μπορούμε να θεωρήσουμε τον **τρόπο αποστολής των προτάσεων** και το **βαθμό εξατομίκευσης** (Εικόνα 2). Αυτά τα στοιχεία δεν είναι ανεξάρτητα μεταξύ τους αφού για παράδειγμα συγκεκριμένες σχεδιαστικές επιλογές απαιτούν συγκεκριμένες εισόδους ή συγκεκριμένες έξοδοι μπορούν να παραχθούν μόνο με κάποιες και όχι με όλες τις μεθόδους προτάσεων.

#### 3.1.1 Η είσοδος

Ένα RS δέχεται σαν είσοδο μια συλλογή από δεδομένα που μπορεί να περιλαμβάνουν μεταξύ άλλων τις προτιμήσεις των πελατών, τα χαρακτηριστικά των προϊόντων, τις συσχετίσεις των συναλλαγών, κλπ. ώστε να παράγει εξόδους-προτάσεις. Τα δεδομένα αυτά καλύπτουν ένα μεγάλο εύρος και μπορούν να διακριθούν σε σχέση με την προέλευσή τους σε

- α) **δεδομένα για τον πελάτη** στον οποίο θέλουμε να προτείνουμε προϊόντα και
- β) **γενικά δεδομένα από τις κοινότητες** ή τη **συγκεκριμένη κοινότητα που ανήκει ο πελάτης**.



Εικόνα 2 Δομή του Συστήματος Συστάσεων

#### Προσανατολισμένα στον πελάτη δεδομένα εισόδου

Τα δεδομένα αυτά εισάγονται στο RS ώστε να παραχθούν προσανατολισμένες στον πελάτη προτάσεις. Ένα σύστημα που δεν χρησιμοποιεί πληροφορίες για κάθε πελάτη ξεχωριστά, μπορεί να εμφανίσει μόνο γενικές προτάσεις. Οι κυριότεροι τύποι δεδομένων αφορούν:

#### Υπονοούμενη πλοήγηση (implicit navigation).

Τα δεδομένα σε αυτή την περίπτωση εξάγονται από την πλοηγητική συμπεριφορά του πελάτη χωρίς αυτός να είναι ενήμερος για την παρουσία του RS. Μπορεί να περιλαμβάνουν τα συγκεκριμένα προϊόντα που ο πελάτης βλέπει την τρέχουσα στιγμή ή αυτά που περιέχονται στο



καλάθι αγορών του (shopping cart). Επίσης την κατηγορία ή το χαρακτηριστικό στο οποίο ο πελάτης πλοηγείται.

#### **Σαφής πλοήγηση (explicit navigation).**

Σε αντίθεση με την υπονοούμενη, στη σαφή πλοήγηση οι πελάτες εισάγουν τα δεδομένα με σκοπό να πληροφορηθούν το σύστημα για τις προτιμήσεις τους. Ο πιο απλός τρόπος είναι με τη χρήση ερωτηματολογίων ή φορμών που συγκεντρώνουν δημογραφικά ή άλλα στοιχεία.

#### **Λέξεις κλειδιά (keywords) και χαρακτηριστικά των προϊόντων (attributes).**

Συχνά τα δεδομένα από τον πελάτη δεν περιορίζονται σε μια απλή κατηγορία ή κάποιο προϊόν που τον ενδιαφέρει. Μπορεί επίσης να χρησιμοποιηθούν λέξεις κλειδιά ή χαρακτηριστικά των προϊόντων είτε με σαφή τρόπο μέσω π.χ. μιας μηχανής αναζήτησης, είτε με υπονοούμενο τρόπο π.χ. από τα προϊόντα που ο πελάτης βλέπει την τρέχουσα στιγμή. Σε κάθε περίπτωση, αυτά τα δεδομένα μεταφράζονται σαν είσοδος που υποδεικνύει τα τρέχοντα ενδιαφέροντα του πελάτη.

#### **Βαθμολόγηση (rating).**

Οι πελάτες παρέχουν τις πιο χρήσιμες και σαφείς πληροφορίες στο RS όταν αξιολογούν τα προϊόντα που έχουν ήδη αγοράσει. Η αξιολόγηση μπορεί να γίνει με βαθμούς π.χ. «βαθμολόγησε ένα προϊόν σε μια κλίμακα από 1-5» ή με απλή ερώτηση της μορφής «σου άρεσε αυτό;».

#### **Ιστορικό συναλλαγών (transactions history).**

Το ιστορικό μπορεί να χρησιμοποιηθεί σαν μια μορφή υπονοούμενης βαθμολόγησης. Με βάση αυτό μπορούν να συσχετισθούν προϊόντα και να εξαχθούν συμπεράσματα για τις προτιμήσεις των πελατών.

#### **Δεδομένα από τις κοινότητες**

Τα δεδομένα αυτά περιλαμβάνουν μια ποικιλία πληροφοριών για τον τρόπο που τα διάφορα μέλη της κοινότητας ή ολόκληρη η κοινότητα αντιλαμβάνεται τα προϊόντα. Δεδομένα εισόδου που αντανakλούν τις απόψεις μιας κοινότητας είναι εκείνα τα **χαρακτηριστικά των προϊόντων (item attributes)** στα οποία αποδίδονται ετικέτες ή κατηγορίες π.χ. το είδος ενός βιβλίου ή μιας ταινίας. Ομοίως, η **δημοτικότητα ενός προϊόντος (item popularity)** μπορεί να εκφράζει τις συνολικές πωλήσεις ή μια λίστα με best-sellers. Με τον ίδιο τρόπο που χρησιμοποιείται το ιστορικό συναλλαγών για να παραχθούν προτάσεις για έναν πελάτη, μπορεί να χρησιμοποιηθεί και το **ιστορικό συναλλαγών μιας κοινότητας (community transactions history)**. Τα παραπάνω μπορούν να συνδυασθούν για να παραχθεί η λίστα των καλύτερων προϊόντων ενός site ή να εξαχθούν συμπεράσματα σχετικά με την κίνηση των πωλήσεων ή τις ομοιότητες/διαφορές των προϊόντων. Παράλληλα, πολλά sites ενθαρρύνουν τα **σχόλια κειμένου (text comments)** από τους πελάτες τους, τα οποία αν και θεωρούνται πολύ χρήσιμα, εντούτοις απαιτούν αρκετό χρόνο επεξεργασίας, αφού κάποιος πρέπει να διαβάσει μια παράγραφο για να καταλάβει αν το σχόλιο είναι θετικό ή αρνητικό. Για να διευκολυνθεί αυτή η διαδικασία τα περισσότερα RSs παρέχουν δυνατότητες **βαθμολόγησης (rating)** των προϊόντων. [3][4]

### **3.1.2 Η έξοδος**

Η έξοδος ενός RS εξαρτάται από τον τύπο, την ποσότητα και την ποιότητα των δεδομένων που ανέλυσε. Ο πιο κοινός τύπος εξόδου είναι οι **προτάσεις (suggestions)**. Αυτές συνήθως έχουν τη μορφή «δοκίμασε αυτό» ή «αυτό» στη σελίδα που ο χρήστης πλοηγείται. Ένας δεύτερος τύπος αφορά σε **προβλέψεις (predictions)** για τη βαθμολόγηση που θα έδινε ο πελάτης σε ένα προϊόν. Μπορούν να παρουσιάζονται σαν μια μεμονωμένη πρόταση ή σαν μια λίστα προτάσεων ή να εμφανίζονται στη σελίδα παρουσίασης του προϊόντος. Όταν οι κοινότητες είναι

μικρές ή τα μέλη γνωρίζονται μεταξύ τους είναι χρήσιμο να παρουσιάζονται οι **βαθμολογίες (ratings)** συγκεκριμένων μελών ώστε κάθε χρήστης να μπορεί να εξάγει τα δικά του συμπεράσματα για τη σημασία της πρότασης. Αυτή η τεχνική είναι χρήσιμη όταν ο πελάτης μπορεί να επιλέξει γνωστά μέλη της κοινότητας ή όταν οι βαθμολογίες συνοδεύονται και από **κριτικές (reviews)**. Οι κριτικές είναι παραδείγματα προτάσεων που περιέχουν εκτιμήσεις τις οποίες δεν μπορεί να κατανοήσει πλήρως ένα υπολογιστικό σύστημα. Πράγματι, είναι δύσκολο να διακριθούν τα θετικά ή τα αρνητικά σχόλια για ένα προϊόν και για αυτό το λόγο συχνά συνοδεύονται και από αριθμητικές βαθμολογίες. [3]

### 3.1.3 Η μέθοδος

Στη συνέχεια παρουσιάζονται οι πιο γνωστές μέθοδοι που χρησιμοποιούν τα RSs. Κάθε μια αντιπροσωπεύει μια οικογένεια αλγορίθμων και τεχνικών, ενώ πρέπει να σημειωθεί ότι είναι δυνατός ο συνδυασμός περισσότερων της μιας μεθόδου για την παραγωγή των προτάσεων.

#### Raw retrieval.

Στη μέθοδο αυτή παρέχεται στους πελάτες ένα interface αναζήτησης μέσω του οποίου μπορούν να κάνουν ερωτήσεις στη βάση δεδομένων των προϊόντων. Στη συνέχεια, το RS με απλή συντακτική διαδικασία επιστρέφει στον πελάτη ό,τι ζήτησε. Για παράδειγμα, όταν ο πελάτης ζητά από ένα onlineβιβλιοπωλείο ένα βιβλίο ενός συγγραφέα, τότε το σύστημα επιστρέφει μια λίστα με όλα τα βιβλία του, δίνοντας τη δυνατότητα στον πελάτη να βρει ίσως κάποιο βιβλίο του συγγραφέα που δεν γνώριζε.

#### Manual selection.

Ορισμένα RSs παράγουν προτάσεις με βάση δεδομένα που έχουν συλλεχθεί χειρωνακτικά από συντάκτες, καλλιτέχνες, κριτικούς ή άλλους ειδικούς. Αυτές οι προτάσεις αντανakλούν τις προτιμήσεις, τα ενδιαφέροντα, τους υποκειμενικούς λόγους και σκοπούς των δημιουργών τους και μπορεί συχνά να συνοδεύονται από σχόλια κειμένου. Για παράδειγμα, αν ο χρήστης επιλέξει ένα συγκεκριμένο είδος ταινιών π.χ. θρίλερ, τότε εμφανίζεται μια λίστα ταινιών τις οποίες κάποιος κριτικός θεωρεί σαν τα καλύτερα θρίλερ όλων των εποχών. Η μέθοδος δεν κάνει υπολογιστική επεξεργασία, απλώς αναπαράγει ό,τι θα εμφανιζόταν σε μια στήλη περιοδικού κριτικής ταινιών.

#### Statistical summaries.

Αρκετά RSs παρέχουν στατιστικές αναλύσεις των απόψεων μιας κοινότητας. Αυτά τα στοιχεία μπορεί να περιλαμβάνουν μετρήσεις δημοτικότητας μέσα στην κοινότητα (π.χ. ποσοστό ανθρώπων που τους αρέσει ή αγόρασαν ένα συγκεκριμένο προϊόν) ή συνοπτικές βαθμολογίες (π.χ. ο αριθμός των ατόμων που προτείνουν ένα προϊόν ή η μέση βαθμολογία για ένα προϊόν). Επίσης, διάφορα συστήματα παρουσιάζουν μέσες τιμές βαθμολογιών αγοραστών αλλά και πωλητών. Οι πιθανοί πελάτες μπορούν να συμβουλευτούν τις μέσες τιμές και τις εκτιμήσεις αλλά δεν μπορούν να δουν βαθμολογίες από πελάτες με τους οποίους συμφωνούν. Με αυτό τον τρόπο παράγονται μόνο μη-εξατομικευμένες προτάσεις (δηλαδή ίδιες για όλους τους πελάτες).

#### Attribute-based technologies.

Αυτά τα RSs βασίζονται στα συντακτικά χαρακτηριστικά των προϊόντων. Αν και στην πιο απλή μορφή τους εκφυλίζονται στη μέθοδο raw retrieval, όσα συστήματα χρησιμοποιούν αυτό το μοντέλο προχωράνε ένα βήμα παραπέρα από το απλό ερώτημα. Για παράδειγμα, ένα πελάτης πλοηγείται στο τμήμα «κλασική μουσική» ενός e-shop και έχει στο καλάθι του διάφορα CDs ειδικής προσφοράς. Τότε μπορεί να λάβει μια πρόταση για έκπτωση σε κάποιο CD κλασικής μουσικής. Άλλα συστήματα για να παράγουν προτάσεις χρησιμοποιούν τα προφίλ των χρηστών που υποδεικνύουν τις προτιμήσεις τους π.χ. μπορεί ένας πελάτης να αγοράζει μόνο CDs που έχουν έκπτωση, ενώ κάποιος άλλος ποτέ δεν αγοράζει μουσική της δεκαετίας του '70.

### Item-to-item correlation.

Πολλά RSs αναγνωρίζουν τα προϊόντα που εμφανίζονται πιο συχνά μαζί με τα προϊόντα για τα οποία ο χρήστης έχει εκφράσει κάποιο ενδιαφέρον. Η συσχέτιση μπορεί να βασίζεται σε δεδομένα αγορών, προτιμήσεων από κοινούς πελάτες, ή άλλες μετρικές. Στην πιο απλή μορφή της, μπορεί να χρησιμοποιηθεί για να αναγνωρίσει ταιριαστά είδη για ένα συγκεκριμένο προϊόν όπως άλλου είδους ρουχισμός που αγοράζεται συνήθως μαζί με ένα παντελόνι π.χ. πουκάμισο ή πουλόβερ. Πιο προηγμένα συστήματα συσχετίζουν ένα ολόκληρο σύνολο από προϊόντα π.χ. αυτά που έχει ο χρήστης στο καλάθι του για να παράγουν τις κατάλληλες προτάσεις. Συνήθως, χρησιμοποιούνται οι τρέχουσες πωλήσεις ή άλλα τρέχοντα ενδιαφέροντα αντί για τη μακροπρόθεσμη ιστορία του πελάτη γεγονός που τις κάνει πολύ χρήσιμες για προτάσεις δώρων. Τέλος, ο πελάτης θα πρέπει να υποδείξει και κάποια πρόσθετα χαρακτηριστικά ή προϊόντα που μπορεί να αρέσουν σε αυτόν που θα λάβει το δώρο.

### User-to-user correlation.

Πολλά RSs προτείνουν προϊόντα σε έναν πελάτη με βάση τη συσχέτισή του με άλλους πελάτες που έχουν αγοράσει προϊόντα από το e-shop. Αυτή η τεχνολογία είναι γνωστή και σαν collaborative filtering ή social filtering και προέρχεται από την τεχνική information filtering που χρησιμοποιεί απόψεις ομάδων για να προτείνει προϊόντα σε διάφορα άτομα. Το My CDNOW είναι ένα σύστημα που χρησιμοποιεί user-to-user correlation για να αναγνωρίσει μια κοινότητα από πελάτες που τείνουν να έχουν και να τους αρέσουν τα ίδια σύνολα από CDs. Η τεχνική στηρίζεται στην αρχή ότι εάν διάφορα μέλη της κοινότητας που ανήκει ο πελάτης έχουν και τους αρέσει το τελευταίο άλμπουμ του Sting τότε είναι αρκετά πιθανό να αρέσει και σε αυτόν.

Ένα σημαντικό θέμα που θα πρέπει να προσεχθεί ιδιαίτερα σχετικά με τη μέθοδο recommendation είναι αν ο υπολογισμός θα γίνει εξολοκλήρου **online**, όταν δηλαδή ο πελάτης αλληλεπιδρά με το e-shop, ή μέρη του να υπολογίζονται **offline**, κυρίως για λόγους απόδοσης. Οι online προτάσεις προτιμώνται γιατί απαντούν απευθείας στο χρήστη. Οι raw retrieval, manual selection, statistical summaries, attribute-based μέθοδοι εκτελούν απλούς υπολογισμούς που συνήθως γίνονται στη διάρκεια της αλληλεπίδρασης του πελάτη με το site. Οι item-to-item correlation και user-to-user correlation τεχνικές είναι περισσότερο πολύπλοκες υπολογιστικά και συχνά απαιτούν μια offline επεξεργασία η οποία θα προετοιμάσει το μοντέλο που θα εκτελεσθεί online. Μια πρόκληση στο σχεδιασμό του μοντέλου είναι να διασφαλισθεί ότι το online σύστημα θα παράγει αποτελέσματα που θα ανταποκρίνονται όσο το δυνατόν περισσότερο στην αλληλεπιδραστική είσοδο του χρήστη. [3][4]

### 3.1.4 Ο βαθμός εξατομίκευσης

Τα RSs μπορούν να παράγουν προτάσεις με διαφορετικό **βαθμό εξατομίκευσης (personalization degree)**. Όταν το σύστημα προτείνει όμοια recommendations σε κάθε πελάτη τότε η εφαρμογή θεωρείται ότι είναι **μη-εξατομικευμένη (non-personalized)**. Τα συγκεκριμένα recommendations μπορεί να βασίζονται σε επιλογή με το χέρι, στατιστικά στοιχεία, ή άλλες τεχνικές. Οι μεγαλύτερες πωλήσεις ενός προϊόντος, οι επιλογές των κριτικών, οι μέσοι όροι, τα ακατέργαστα σχόλια, κλπ. παρουσιάζουν τις ίδιες προτάσεις σε όλους τους πελάτες του e-shop. Τα RSs που χρησιμοποιούν δεδομένα των πελατών για να προσαρμόσουν τις προτάσεις στα ενδιαφέροντά τους παρέχουν **εφήμερη εξατομίκευση (ephemeral personalization)**. Στην περίπτωση αυτή, οι προτάσεις ανταποκρίνονται στην πλοήγηση του πελάτη και στις επιλογές του. Αυτά τα συστήματα βασίζονται συνήθως σε item-to-item correlation, attribute-based recommendation ή και τα δυο μαζί. Τέλος, οι εφαρμογές που χρησιμοποιούν **επίμονη εξατομίκευση (persistent personalization)** μπορούν να δημιουργήσουν προτάσεις που είναι διαφορετικές για διαφορετικούς πελάτες ακόμα και αν αυτοί κοιτάνε τα ίδια προϊόντα. Συνήθως χρησιμοποιούν item-to-item correlation, user-to-user correlation ή attribute-based recommendation. Απαιτούν από τους πελάτες να συντηρούν μόνιμες **ταυτότητες (profiles)** αλλά τους επιβραβεύουν με μεγαλύτερο βαθμό εξατομίκευσης. [3][4]

Όπως έχουμε ήδη αναφέρει τα RSs μπορούν να ταξινομηθούν σε τρεις βασικές κατηγορίες: content-based filtering (φιλτράρισμα με βάση το περιεχόμενο), collaborative filtering (συνεργατικό φιλτράρισμα) και hybrid approaches (υβριδικές προσεγγίσεις). Τα τελευταία χρόνια

σημαντική έρευνα έχει γίνει στην περιοχή. Η εικόνα 3 παρουσιάζει μια ταξινόμηση των τεχνικών RSs καθώς και αντιπροσωπευτικά ερευνητικά παραδείγματα

Κατηγορίες τεχνικών RSs	Ευριστικές (heuristic-based)	Μοντέλου (model-based)
<b>Φιλτράρισμα με βάση το περιεχόμενο ή content-based filtering</b>	<p>Τεχνικές:</p> <ul style="list-style-type: none"> <li>• TF-IDF (Information Retrieval)</li> <li>• Clustering</li> </ul> <p>Αντιπροσωπευτικά παραδείγματα:</p> <ul style="list-style-type: none"> <li>• Lang, 1995</li> <li>• Balabanovic &amp; Shoham, 1997</li> <li>• Pazzani &amp; Billsus, 1997</li> </ul>	<p>Τεχνικές:</p> <ul style="list-style-type: none"> <li>• Bayesian classifiers</li> <li>• Clustering</li> <li>• Decision trees</li> <li>• Artificial neural networks</li> </ul> <p>Αντιπροσωπευτικά παραδείγματα:</p> <ul style="list-style-type: none"> <li>• Pazzani &amp; Billsus, 1997</li> <li>• Mooney et al., 1998</li> <li>• Mooney &amp; Roy, 1999</li> <li>• Billsus &amp; Pazzani, 1999, 2000</li> </ul>
<b>Συνεργατικό φιλτράρισμα ή collaborative filtering</b>	<p>Τεχνικές:</p> <ul style="list-style-type: none"> <li>• Nearest neighbor (cosine, correlation)</li> <li>• Clustering</li> <li>• Graph theory</li> </ul> <p>Αντιπροσωπευτικά παραδείγματα:</p> <ul style="list-style-type: none"> <li>• Resnick et al., 1994</li> <li>• Hill et al., 1995</li> <li>• Shardanand &amp; Maes, 1995</li> <li>• Breese et al., 1998</li> <li>• Nakamura &amp; Abe, 1998</li> <li>• Aggarwal et al., 1999</li> <li>• Delgado &amp; Ishii, 1999</li> <li>• Pennock &amp; Horwitz, 1999</li> <li>• Sarwar et al., 2001</li> </ul>	<ul style="list-style-type: none"> <li>• Zhang et al., 2002</li> </ul> <p>Τεχνικές:</p> <ul style="list-style-type: none"> <li>• Bayesian networks</li> <li>• Clustering</li> <li>• Artificial neural networks</li> <li>• Linear regression</li> <li>• Probabilistic models</li> </ul> <p>Αντιπροσωπευτικά παραδείγματα:</p> <ul style="list-style-type: none"> <li>• Billsus &amp; Pazzani, 1998</li> <li>• Breese et al., 1998</li> <li>• Ungar &amp; Foster, 1998</li> <li>• Chien &amp; George, 1999</li> <li>• Getoor &amp; Sahami, 1999</li> <li>• Pennock &amp; Horwitz, 1999</li> <li>• Goldberg et al., 2001</li> <li>• Kumar et al., 2001</li> <li>• Pavlov &amp; Pennock, 2002</li> <li>• Shani et al., 2002</li> <li>• Yu et al., 2002, 2004</li> <li>• Hofmann 2003, 2004</li> <li>• Marlin, 2003</li> </ul>
<b>Υβριδικές προσεγγίσεις ή hybrid approaches</b>	<p>Τεχνικές:</p> <ul style="list-style-type: none"> <li>• Linear combination of predicted ratings</li> <li>• Various voting schemes</li> <li>• Incorporating one component as a part of the heuristic for the other</li> </ul> <p>Αντιπροσωπευτικά παραδείγματα:</p> <ul style="list-style-type: none"> <li>• Balabanovic &amp; Shoham, 1997</li> <li>• Craypool et al., 1999</li> <li>• Good et al., 1999</li> <li>• Pazzani, 1999</li> <li>• Billsus &amp; Pazzani, 2000</li> <li>• Tran &amp; Cohen, 2000</li> <li>• Melville et al., 2002</li> </ul>	<p>Τεχνικές:</p> <ul style="list-style-type: none"> <li>• Incorporating one component as a part of the model for the other</li> <li>• Building one unifying model</li> </ul> <p>Αντιπροσωπευτικά παραδείγματα:</p> <ul style="list-style-type: none"> <li>• Basu et al., 1998</li> <li>• Condliff et al., 1999</li> <li>• Soboroff &amp; Nicholas, 1999</li> <li>• Ansari et al., 2000</li> <li>• Popescul et al., 2001</li> <li>• Schein et al., 2002</li> </ul>

Εικόνα 3 Ταξινόμηση των τεχνικών RSs

### 3.2 Η Προτεινόμενη Προσέγγιση

Το πρόβλημα που μελετάται στα πλαίσια της μεταπτυχιακής διατριβής είναι αυτό της παραγωγής προτάσεων σε μια εφαρμογή ενοικίασης και αγοράς ταινιών. Συγκεκριμένα, η προτεινόμενη υβριδική προσέγγιση στοχεύει στην παραγωγή αποτελεσματικών συστάσεων για τους πελάτες ενός καταστήματος που νοικιάζει κινηματογραφικές ταινίες, επεκτείνοντας τις παραδοσιακές τεχνικές για τα αντικείμενα προς σύσταση. Η γνώση για τους πελάτες και τα προϊόντα προκύπτει από δεδομένα χρήσης (usage data), τη δομή του προφίλ του χρήστη σε συνδυασμό με το ιστορικό συναλλαγών τόσο του ίδιου όσο και των υπολοίπων πελατών καθώς και από την εφαρμογή τεχνικών ταιριάσματος (matching) «όμοιων» (similar) πελατών. Το πιλοτικό πληροφοριακό σύστημα διαχείρισης, ενοικίασης και αγορών ταινιών αποτελείται από μια συλλογή οθονών που περιέχουν πληροφορίες για τις ταινίες, τους πελάτες, στατιστικά στοιχεία, δείκτες αξιολόγησης αλλά και το υποσύστημα δυναμικής δημιουργίας νέου μοντέλου συστάσεων διαφορετικών παραμέτρων και ελεγχού του στο σύστημα. Το υλοποιημένο framework μπορεί να εφαρμοστεί και σε άλλου τύπου e-shops π.χ. βιβλιοπωλεία, δισκοπωλεία, κλπ.

Τα αρχικά στοιχεία ανακτήθηκαν από τη βάση δεδομένων Internet Movie Database (<http://www.imdb.com>) και στη συνέχεια οργανώθηκαν σε ένα οντολογικό σχήμα προκειμένου να αποτελέσουν ένα πλούσιο σύνολο δεδομένων (superset) το οποίο χρησιμοποιήθηκε στη διαδικασία ανακάλυψης γνώσης (knowledge discovery process). Συγκεκριμένα, αποτέλεσε την είσοδο στη φάση εξόρυξης γνώσης ώστε να συνδυαστούν και να μετασχηματιστούν τα υπονοούμενα/ρητά δεδομένα (κλάση χρηστών, ιστορικό, προφίλ, δομή και περιεχόμενο) σε νέες μορφές.

Παράλληλα το μοντέλο δημιουργεί συνδέσεις μεταξύ των ταινιών σύμφωνα με τα χαρακτηριστικά τους (attributes). Χρησιμοποιώντας το ιστορικό των συναλλαγών (transactions history file) του συγκεκριμένου πελάτη, οι προτιμήσεις του μπορούν να προβλεφτούν. Για παράδειγμα, μπορεί να βρεθεί εάν σε ένα πελάτη αρέσει ή όχι μια ταινία μιας συγκεκριμένης κατηγορίας, ενός συγκεκριμένου σκηνοθέτη, ενός συγκεκριμένου ηθοποιού ή συνδυασμών άλλων χαρακτηριστικών. Όταν ένα ή περισσότερα κριτήρια ταιριάσματος ικανοποιούνται, τότε άλλες ταινίες μπορούν να προσδιοριστούν σύμφωνα με το μοντέλο που έχουν παρόμοια χαρακτηριστικά με αυτές που ο πελάτης έχει ήδη νοικιάσει. Στην περίπτωση ενός νέου πελάτη όπου το ιστορικό του είναι κενό, οι πληροφορίες από την αίτηση εγγραφής (registration form) του στο e-shop αναλύονται ώστε να ταξινομηθεί σε μια συγκεκριμένη κλάση πελατών (user class), ενώ οι προτάσεις παράγονται με βάση το μοντέλο. Αυτή η ολοκλήρωση παρέχει πρόσθετη γνώση για τις προτιμήσεις των πελατών και επιτρέπει την παραγωγή πιο επιτυχημένων συστάσεων. Ακόμη και στην περίπτωση του «cold-start problem» όπου δεν είναι διαθέσιμη αρχική πληροφορία για τη συμπεριφορά του πελάτη, η προσέγγιση μπορεί να παράγει λογικές και σχετικές συστάσεις.

## 4 ΑΝΑΛΥΣΗ ΚΑΙ ΣΧΕΔΙΑΣΗ

Στο παρόν κεφάλαιο περιγράφεται η ανάλυση και η σχεδίαση του συστήματος και του framework του οποίου την γενική περιγραφή δώσαμε παραπάνω. Η ανάλυση και σχεδίαση αυτή γίνεται με τη βοήθεια των εγγράφων τεχνολογίας λογισμικού για το προς ανάπτυξη σύστημα. Αρχικά κάνουμε μια εισαγωγή με την περιγραφή της γλώσσας μοντελοποίησης UML καθώς και των μεθοδολογιών ανάπτυξης λογισμικού RUP οι οποίες χρησιμοποιήθηκαν. Στη συνέχεια παραθέτουμε τα έγγραφα της τεχνολογίας λογισμικού που προέκυψαν. Σημειώνουμε στο σημείο αυτό ότι για τα περισσότερα από τα σχέδια που έγιναν για τις ανάγκες των εγγράφων της τεχνολογίας λογισμικού χρησιμοποιήθηκε το πρόγραμμα Microsoft Office Visio 2007.

### 4.1 Εισαγωγή

#### 4.1.1 UML (Unified Modeling Language)

Η ενοποιημένη γλώσσα μοντελοποίησης UML (Unified Modeling Language) είναι μια γραφική (graphical) γλώσσα μοντελοποίησης γενικής χρήσης. Προέκυψε από την ενοποίηση πολλών υπάρχοντων και διαφορετικών μεταξύ τους γλωσσών μοντελοποίησης, που είχαν χρησιμοποιηθεί στην πράξη στο πλαίσιο διάφορων μεθοδολογιών ανάπτυξης λογισμικών συστημάτων. Συμπεριλαμβάνει ότι καλό είχαν αυτές οι προηγούμενες γλώσσες μοντελοποίησης. Ήρθε να καλύψει την ανάγκη ύπαρξης μιας μοναδικής γλώσσας μοντελοποίησης με όλα τα καλά που αυτό συνεπάγεται. Έτσι, δικαιολογείται και η ευρεία αποδοχή της σήμερα. Βέβαια, πρωταρχικά απευθύνεται στη μοντελοποίηση λογισμικών συστημάτων. Όμως, μπορεί να χρησιμοποιηθεί και για μοντελοποίηση άλλων κατηγοριών συστημάτων, όπως των τεχνητών συστημάτων. Η UML είναι τώρα βιομηχανικό πρότυπο και υποστηρίζεται από το μεγάλο οργανισμό OMG [<http://www.omg.org/>], ανήκει δε στην κατηγορία των αντικειμενοστραφών γλωσσών μοντελοποίησης.

#### 4.1.2 RUP (Rational Unified Process)

Η μεθοδολογία RUP (Rational Unified Process) αποτελεί μια βιομηχανική μεθοδολογία λογισμικού η οποία αναπτύχθηκε από την εταιρεία Rational Software Corporation. Η RUP βασίζεται σε ένα σύνολο από αρχές ανάπτυξης λογισμικού όπως για παράδειγμα:

- Ανάπτυξη λογισμικού επαναληπτικά, δηλαδή τα μέρη του λογισμικού αναπτύσσονται, ελέγχονται και συμπληρώνονται διαρκώς.
- Χειρισμός των απαιτήσεων των χρηστών, δηλαδή καθορισμός των αναγκών των χρηστών και αλλαγή των απαιτήσεων όταν οι ανάγκες αυτές αλλάζουν
- Χρήση αρχιτεκτονικής με components
- Οπτική μοντελοποίηση του λογισμικού
- Επαλήθευση της ποιότητας του λογισμικού
- Έλεγχος των αλλαγών στο λογισμικό

### 4.2 Έγγραφο Προδιαγραφών Απαιτήσεων από το Λογισμικό

#### 4.2.1 Σκοπός

Σκοπός αυτού του εγγράφου είναι η παρουσίαση των προδιαγραφών των απαιτήσεων για το πληροφοριακό σύστημα "Smart Media". Ο χρήστης του συστήματος θα έχει τη δυνατότητα να εγγραφεί στο σύστημα, να διαμορφώσει το προφίλ του και τα χαρακτηριστικά του, να πληροφορηθεί για οποιαδήποτε ταινία επιθυμεί καθώς και να την αγοράσει. Στο σύστημα γίνεται Εξατομικευμένο Σύστημα Συστάσεων

κυρίως επίδειξη των δυνατοτήτων του framework το οποίο έχει τη δυνατότητα να προσφέρει στο χρήστη προσωποποιημένες συστάσεις όπως ταινίες που ταιριάζουν περισσότερο στο προφίλ του ανάλογα και με τις επιλεγμένες ταινίες παρόμοιων χρηστών αλλά και ταινίες που σχετίζονται τόσο με το προφίλ του αλλά και με το ιστορικό των αγορών του. Όλες οι πληροφορίες σχετικά με τις λεπτομέρειες των ταινιών, το ιστορικό των αγορών του χρήστη, τα ενδιαφέροντα του κάθε χρήστη και το προφίλ του θα αποθηκεύονται σε μια βάση δεδομένων μέσω της οποίας εφαρμόζονται και οι αλγόριθμοι του framework και τεχνικές εξόρυξης γνώσης για την εξαγωγή των συστάσεων. Οι πληροφορίες της κάθε ταινίας θα παρέχονται μέσω ενός web service το οποίο επικοινωνεί με τη διαδικτυακή βάση δεδομένων [www.imdb.com](http://www.imdb.com).

### Εμβέλεια

Το παρόν έγγραφο παρουσιάζει τις περιπτώσεις χρήσης του συστήματος και περιλαμβάνει περαιτέρω ανάλυση αυτών με τη βοήθεια κατάλληλων διαγραμμάτων σε UML. Το έγγραφο απευθύνεται σε οποιονδήποτε επιθυμεί να χρησιμοποιεί τις παραπάνω υπηρεσίες, καθώς και στον αρχιτέκτονα και κατασκευαστή του συστήματος.

### Ορισμοί, Ακρωνύμια, Συντομογραφίες

RS: Συστήματα Συστάσεων

DM: Εξόρυξη Γνώσης

### Αναφορές

Χρησιμοποιήθηκαν οι πηγές που αναφέρονται στο κεφάλαιο για τη Βιβλιογραφία.

## 4.2.2 Γενική περιγραφή

### Μοντέλο περιπτώσεων χρήσης

Το μοντέλο περιπτώσεων χρήσης περιγράφει μια σειρά από ενέργειες που εκτελεί το σύστημα ώστε να αποδώσει χρήσιμα αποτελέσματα σε ένα ρόλο του συστήματος. Με τον όρο «ρόλος του συστήματος» αναφέρεται ένας χρήστης ή ένα εξωτερικό σύστημα (λογισμικό) που αλληλεπιδρά με το παρόν σύστημα λογισμικού για να λάβει κάποιες πληροφορίες. Με βάση τα παραπάνω δημιουργούνται οι εξής δεκαοχτώ (13) περιπτώσεις χρήσης του συστήματός μας:

#### User Role: User

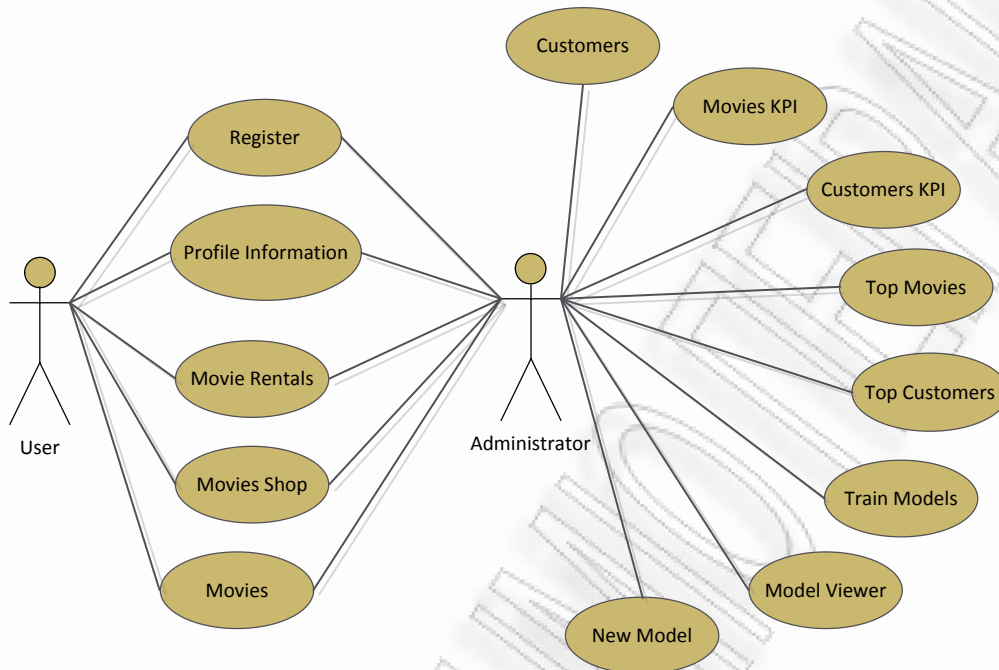
1. Register (Εγγραφή Χρήστη)
2. Profile Information (Προφίλ Χρήστη)
3. Movies Shop (Εικονικό Κατάστημα για την αγορά ταινιών)
4. Movie Rentals (Ιστορικό Αγορών)
5. Movies (Κατάλογος ταινιών)

#### User Role: Administrator

6. Customers (Κατάλογος Χρηστών)
7. Movies KPI (Δείκτης Αξιολόγησης Ταινιών με βάση τις αγορές)
8. Customers KPI (Δείκτης Αξιολόγησης Χρηστών με βάση τις αγορές)
9. Top Customers (Οι χρήστες με τις περισσότερες σε αξία αγορές)
10. Top Movies (Οι ταινίες με τις περισσότερες σε αξία αγορές)
11. Model Viewer (Γραφικό περιβάλλον απεικόνισης του μοντέλου εξατομικευμένων συστάσεων)
12. Train Models (Εκπαίδευση του μοντέλου εξατομικευμένων συστάσεων με τα νέα δεδομένα)
13. New Model (Δυνατότητα δυναμικής δημιουργίας νέου μοντέλου εξατομικευμένων συστάσεων)



Οι παραπάνω περιπτώσεις χρήσης του συστήματός μας φαίνονται στο παρακάτω διάγραμμα. Στη συνέχεια του κειμένου θα γίνει ανάλυση της καθεμιάς από τις περιπτώσεις χρήσης με τη βοήθεια διαγραμμάτων δραστηριοτήτων (activity diagrams).



Εικόνα 4 Use case diagram

### 4.2.3 Απαιτήσεις

#### Προδιαγραφές περιπτώσεων χρήσης

Κάθε περίπτωση χρήσης αναλύεται και τεκμηριώνεται στη συνέχεια του παρόντος εγγράφου στην ενότητα «Ανάλυση Περιπτώσεων Χρήσης».

#### Λειτουργικότητα

Ακολουθως περιγράφονται τα γνωρίσματα που απαιτείται να έχει το πληροφοριακό σύστημα από πλευράς λειτουργικότητας προς το χρήστη. Τα γνωρίσματα αυτά σχετίζονται με τις περιπτώσεις χρήσης που προαναφέρθηκαν.

#### User Interface

Απαιτείται η ύπαρξη GUI (Graphical User Interface) μέσω του οποίου ο χρήστης θα έχει πρόσβαση στις υπηρεσίες. Το GUI θα πρέπει να είναι ενιαίο για την πρόσβαση σε όλες τις διαθέσιμες υπηρεσίες. Ο χρήστης θα δίνει τις απαραίτητες πληροφορίες στην εφαρμογή μέσω των διαφόρων φορμών του GUI. Επίσης στο framework έχει υλοποιηθεί η δυνατότητα προσαρμοστικών διεπαφών ώστε το γραφικό περιβάλλον να προσαρμόζεται στο χρήστη.

#### Απαιτήσεις χρήσης

Το λογισμικό θα πρέπει να είναι απλό στη χρήση και να μην απαιτεί κάποια ιδιαίτερη εκμάθηση ή ιδιαίτερες γνώσεις για κάποιον που θέλει να το χρησιμοποιήσει. Για το λόγο αυτό, όπως προαναφέραμε, το GUI του συστήματος θα πρέπει να είναι φιλικό προς το χρήστη και να βασίζεται σε γραφικό περιβάλλον. Επίσης έχει υλοποιηθεί στο framework ένα σύστημα ευφυούς



βοήθειας για την παρουσίαση στο χρήστη κατάλληλων οδηγιών ώστε να βοηθηθεί στην κατανόηση του συστήματος.

### Απαιτήσεις αξιοπιστίας

Για να είναι το σύστημα διαθέσιμο διαρκώς στο χρήστη, θα πρέπει να είναι διαθέσιμα τα web services που χρησιμοποιούνται. Η παραπάνω απαίτηση προϋποθέτει τη διαθεσιμότητα online σύνδεσης στο Internet για τη πρόσβαση στις διαδικτυακές υπηρεσίες, καθώς και τη διαθεσιμότητα της βάσης δεδομένων που είναι αποθηκευμένα όλα τα δεδομένα.

### Απαιτήσεις ασφάλειας

Το σύστημά μας θα πρέπει να εγγυάται πιστοποίηση χρήστη για την αποθήκευση των δεδομένων του και του προφίλ του στη βάση δεδομένων. Ο χρήστης για να μπορεί να χρησιμοποιεί τις υπηρεσίες θα πρέπει αρχικά να πιστοποιείται.

## 4.2.4 Ανάλυση περιπτώσεων χρήσης

### Περίπτωση χρήσης 1: Register (Εγγραφή Χρήστη)

Η συγκεκριμένη περίπτωση χρήσης αφορά την εγγραφή του χρήστη στο σύστημα δίνοντας τα κατάλληλα στοιχεία και εισάγωντας πληροφορίες όπως αγαπημένοι σκηνοθέτες, ταινίες, ενδιαφέροντα, τεχνολογίες, ηθοποιοί και κανάλια.

Η οθόνη εγγραφής του χρήστη με τις επιμέρους καρτέλες παρουσιάζεται παρακάτω:

Field	Value
Username:	test
Password:	test
Firstname:	John
Lastname:	Markogiannis
Age:	
Gender:	
Education Level:	
Marital Status:	
PPV Freq:	
Prerec Buying Freq:	
Prerec Format:	
Prerec Renting Freq:	
Home Ownership:	
Internet Connection:	
Movie Selector:	
Num Bathrooms:	
Num Bedrooms:	
Num Cars:	
Num Children:	
Num TVs:	
Theater Freq:	
TV Movie Freq:	
TV Signal:	
Prerec Viewing Freq:	

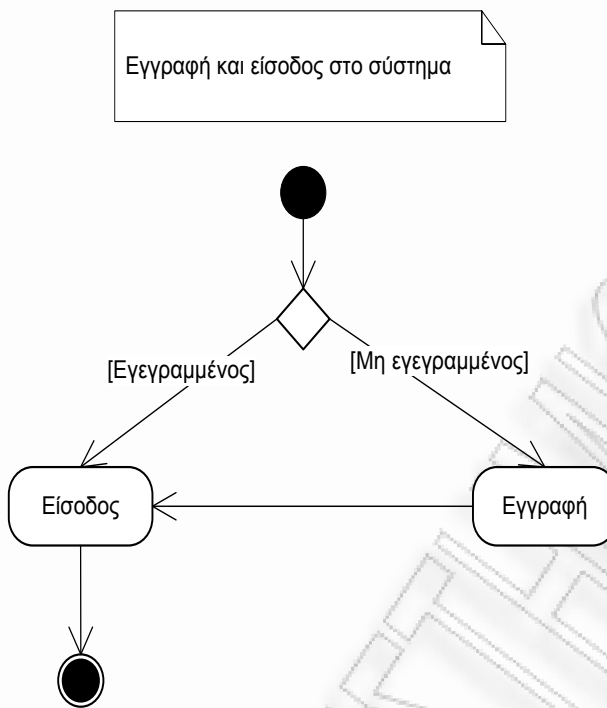
Εικόνα 5 Περίπτωση χρήσης Register

### Ροή γεγονότων

Η βασική ροή γεγονότων περιγράφει τη διαδικασία κατά την οποία η περίπτωση χρήσης εκτελείται και ολοκληρώνεται με επιτυχία.

Στην αρχή ο χρήστης εγγράφεται στο σύστημα εφόσον δεν είναι εγγεγραμμένος. Η εγγραφή πραγματοποιείται με το πάτημα του κουμπιού Register και έπειτα ο χρήστης εισέρχεται στο σύστημα με τα αναγνωριστικά του στοιχεία. Από αυτή τη στιγμή το σύστημα αρχίζει να καταγράφει τις κινήσεις και τις ενέργειες του χρήστη στο σύστημα διαμορφώνοντας το βαθμό εμπειρίας του για τη λειτουργία του ευφυούς συστήματος βοήθειας.

Η παραπάνω ροή παρουσιάζεται στο παρακάτω διάγραμμα ροής:



Εικόνα 6 Activity Diagram – Εγγραφή και Είσοδος στο σύστημα

## Περίπτωση χρήσης 2: Profile Information

### Περιγραφή περίπτωσης χρήσης

Η συγκεκριμένη περίπτωση χρήσης αφορά στην οθόνη όπου παρουσιάζονται τα χαρακτηριστικά του χρήστη, ο οποίος έχει τη δυνατότητα τροποποίησης τους διαμορφώνοντας έτσι το προφίλ του.

The screenshot shows a software interface for managing customer profiles. The 'Customer Layout' window is open, displaying a 'Customer' profile. The 'Profile Information' tab is selected and highlighted with a red box. The form contains the following fields:

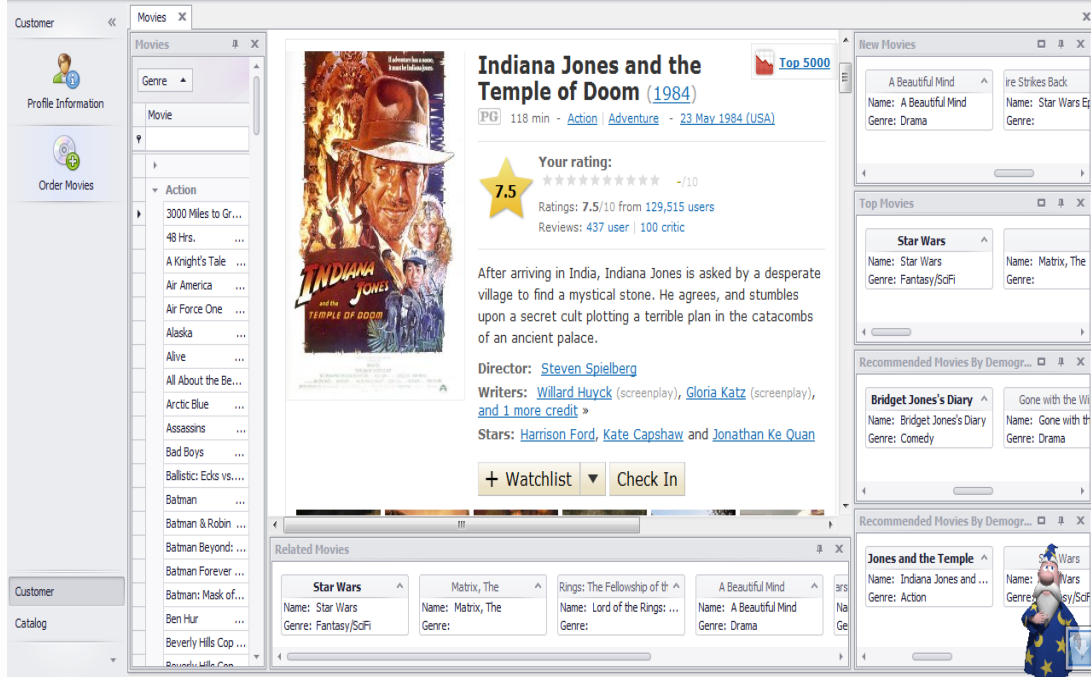
Username:	test	Home Ownership:	Rent
Password:	test	Internet Connection:	DSL
Firstname:	test	Movie Selector:	Me
Lastname:	test	Num Bathrooms:	1
Age:	12	Num Bedrooms:	1
Gender:	Female	Num Cars:	1
Education Level:	Doctorate	Num Children:	1
Marital Status:	Divorced	Num TVs:	1
PPV Freq:	Monthly	Theater Freq:	Monthly
Prerec Buying Freq:	Daily	TV Movie Freq:	Daily
Prerec Format:	Betamax	TV Signal:	Digital Satellite
Prerec Renting Freq:	Daily	Prerec Viewing Freq:	Daily

Εικόνα 7 Profile Information

### Περίπτωση χρήσης 3: Movies Shop

#### Περιγραφή περίπτωσης χρήσης

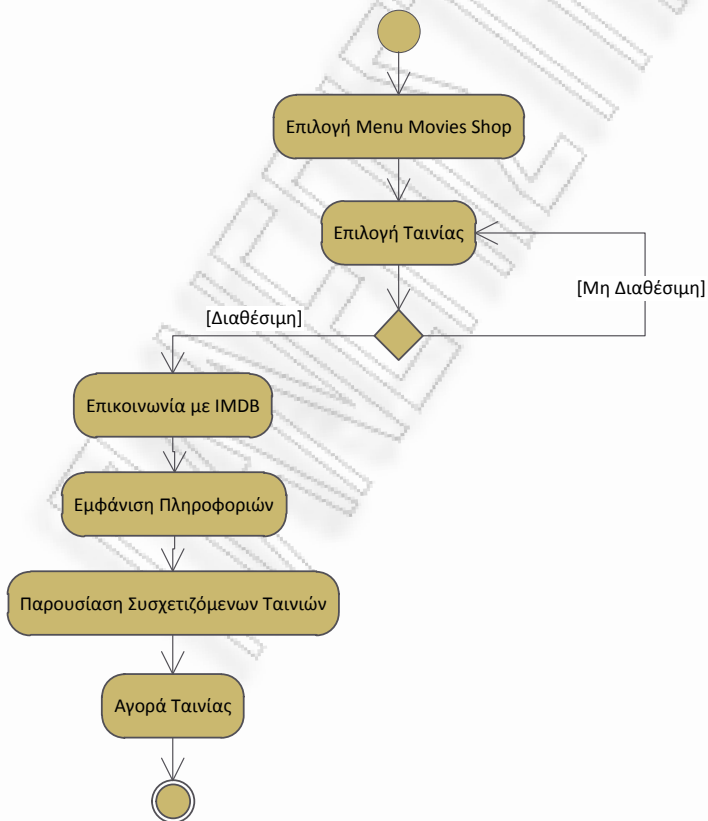
Η περίπτωση χρήσης "Movies Shop" του συστήματός μας αφορά ουσιαστικά τη δυνατότητα του χρήστη να πραγματοποιήσει μία αγορά της επιθυμητής ταινίας. Στο σημείο αυτό βρίσκει εφαρμογή το σύστημα των εξατομικευμένων συστάσεων το οποίο αλληλεπιδρά με το πληροφοριακό σύστημα για την κατηγοριοποίηση του χρήστη και την εξόρυξη γνώσης ώστε τελικά να του προτείνει τις ταινίες που αντιστοιχούν στο προφίλ του. Ο χρήστης μπορεί να επιλέξει από τον κατάλογο στα αριστερά της οθόνης την ταινία της επιλογής του ή κάποια από τις ταινίες που του προτείνονται και εμφανίζονται στα πλαίσια δεξιά του. Ανάλυση του συστήματος συστάσεων θα γίνει αναλυτικά παρακάτω. Επίσης εμφανίζονται οι καινούριες ταινίες καθώς και οι ταινίες με τις περισσότερες πωλήσεις. Όταν ο χρήστης επιλέξει μία ταινία τότε το σύστημα επικοινωνεί με τη διαδικτυακή βάση δεδομένων imdb και απεικονίζει τις πληροφορίες και λεπτομέρειες της ταινίας μέσω του web service.



Εικόνα 8 Movies Shop

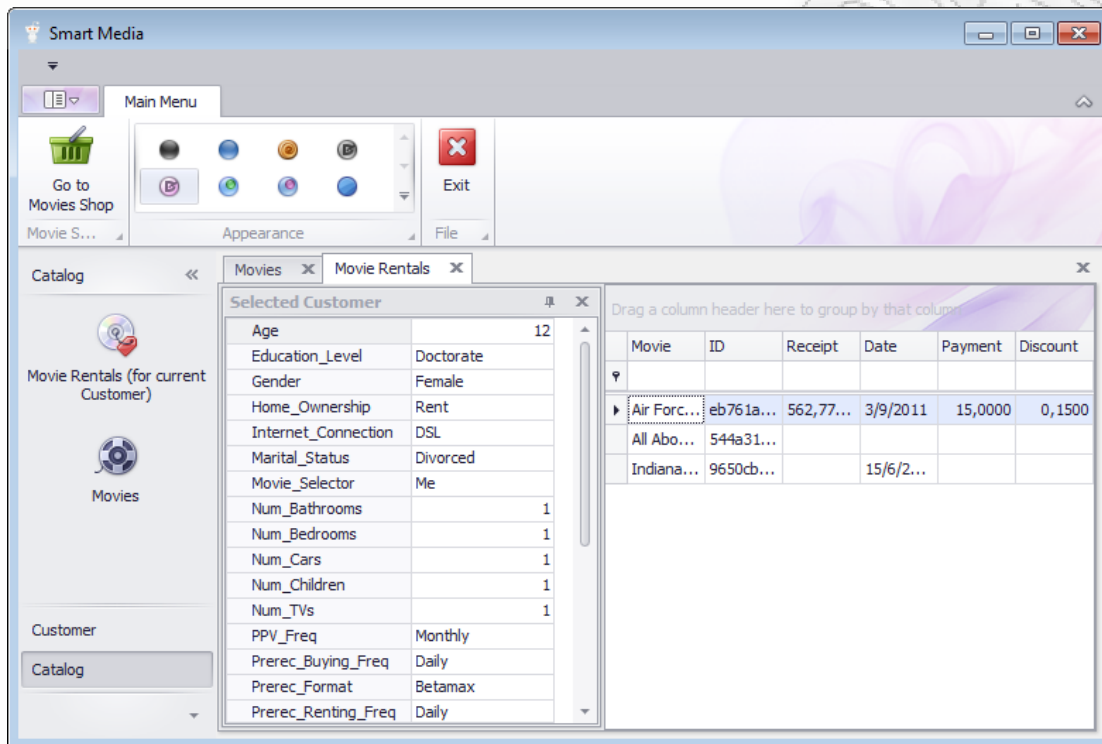
Ροή γεγονότων

Βασική Ροή

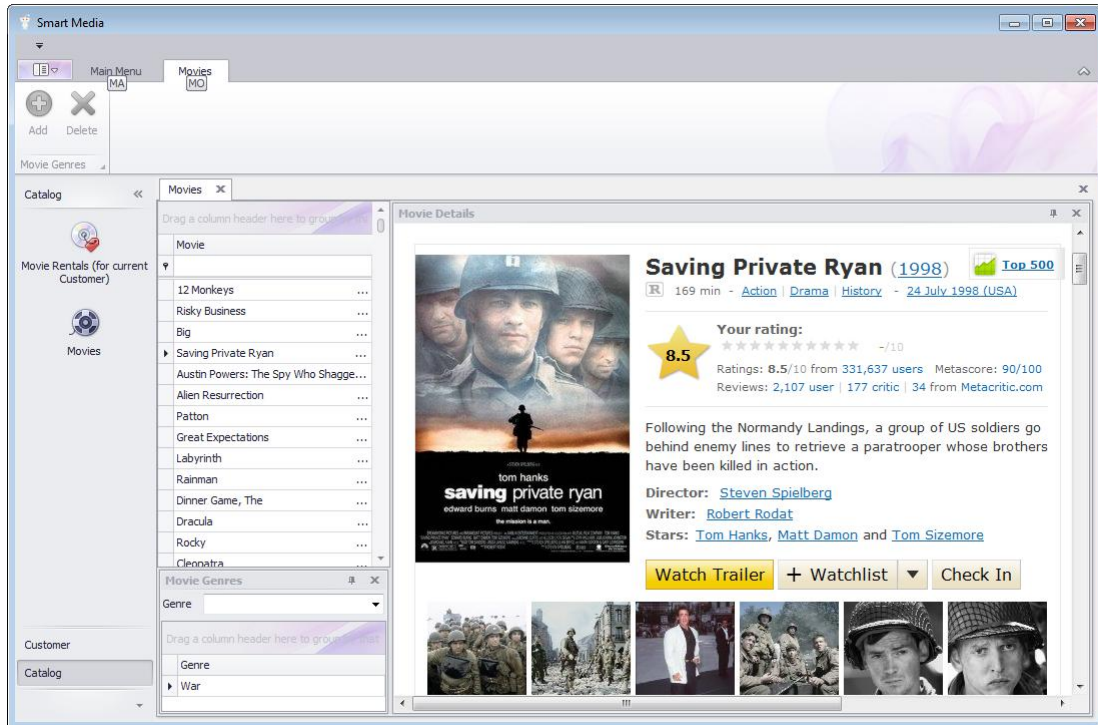


**Εικόνα 9 Activity Diagram – Movies Shop****Περίπτωση χρήσης 4: Movie Rentals****Περιγραφή περίπτωσης χρήσης**

Η περίπτωση χρήσης “Movie Rentals” του συστήματός μας αφορά τη δυνατότητα του χρήστη να παρακολουθήσει το ιστορικό των αγορών του μέσω της κατάλληλης οθόνης.

**Εικόνα 10 Movie Rentals****Περίπτωση χρήσης 5: Movies****Περιγραφή περίπτωσης χρήσης**

Η περίπτωση χρήσης “Movies” του συστήματός μας αποτελεί τον κατάλογο όλων των ταινιών ώστε ο χρήστης να μπορεί να πληροφορηθεί για οποιαδήποτε ταινία επιθυμεί.

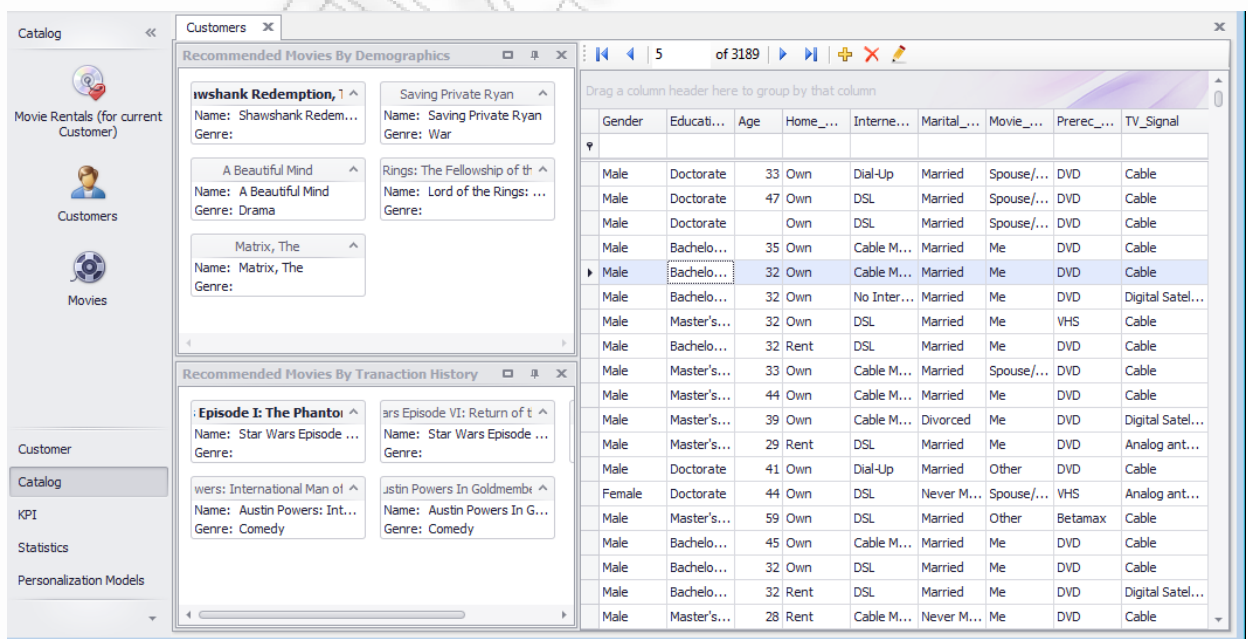


Εικόνα 11 Movies

### Περίπτωση χρήσης 6: Customers

#### Περιγραφή περίπτωσης χρήσης

Η περίπτωση χρήσης “Customers” του συστήματός μας αποτελείται από την οθόνη όλων των χρηστών του συστήματος και για αυτό το λόγο είναι προσβάσιμη μόνο από το διαχειριστή. Εδώ ο διαχειριστής έχει το δικαίωμα της προσθήκης, τροποποίησης και διαγραφής ενός χρήστη. Επίσης ο διαχειριστής έχει τη δυνατότητα παρακολούθησης των αποτελεσμάτων του μοντέλου εξατομικευμένων συστάσεων για κάθε χρήστη.



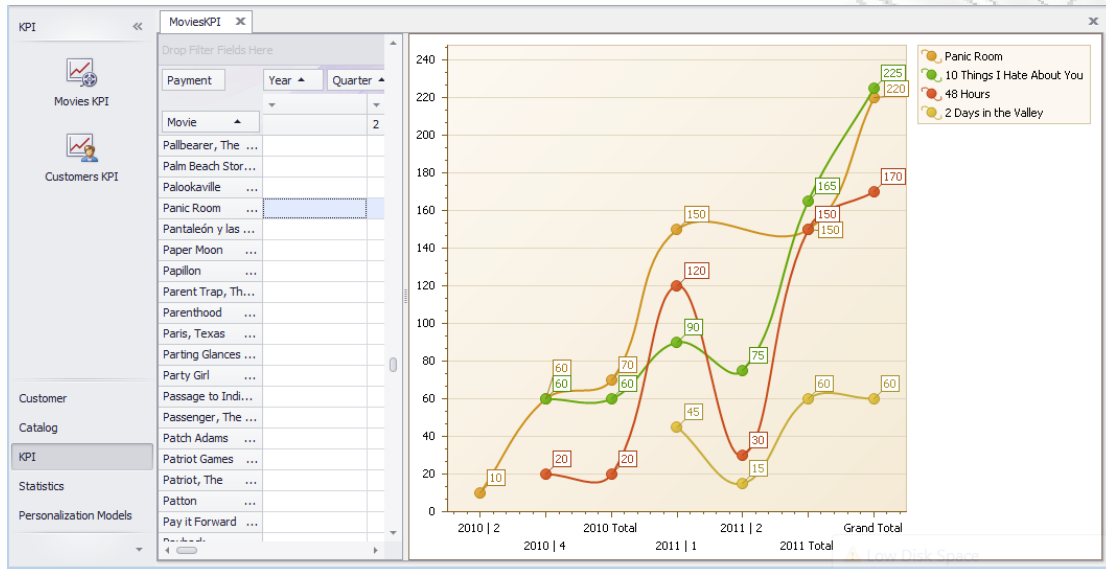
Εικόνα 12 Customers



## Περίπτωση χρήσης 7: Movies KPI

### Περιγραφή περίπτωσης χρήσης

Η περίπτωση χρήσης "Movies KPI" του συστήματός μας αποτελείται από την επιλογή του διαχειριστή της προβολής του δείκτη αξιολόγησης των ταινιών ανάλογα με την αξία των πωλήσεων της κάθε ταινίας σε σχέση με το χρόνο.

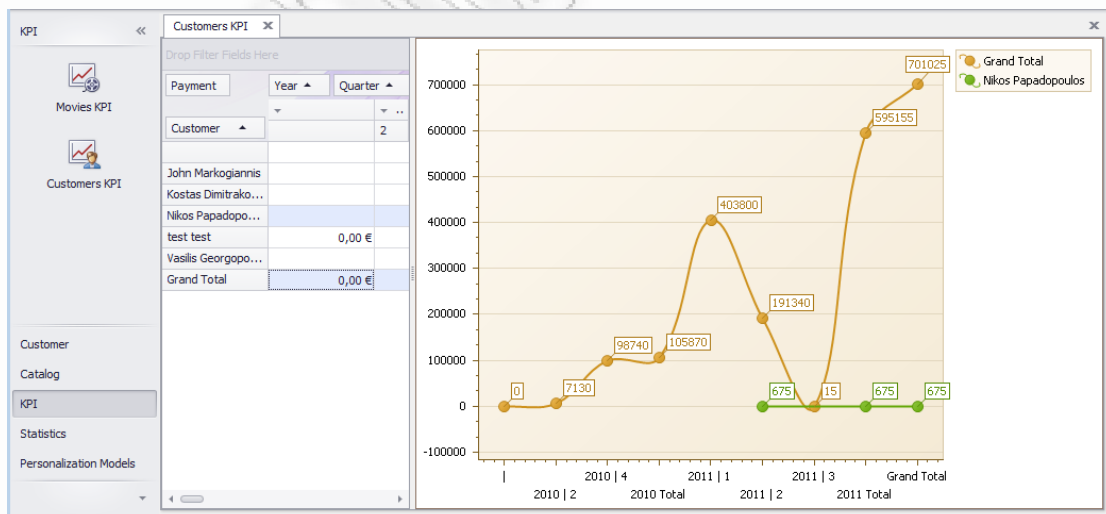


Εικόνα 13 Movies KPI

## Περίπτωση χρήσης 8: Customers KPI

### Περιγραφή περίπτωσης χρήσης

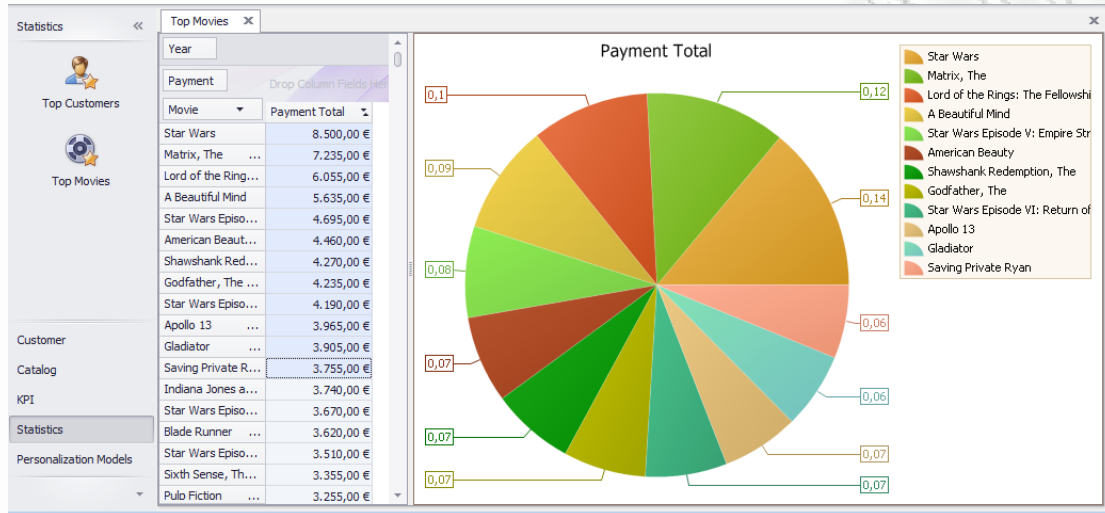
Η περίπτωση χρήσης "Customers KPI" του συστήματός μας αποτελείται από την επιλογή του διαχειριστή της προβολής του δείκτη αξιολόγησης των χρηστών ανάλογα με την αξία των αγορών του κάθε χρήστη σε σχέση με το χρόνο.



Εικόνα 14 Customers KPI

**Περίπτωση χρήσης 9: Top Movies**  
**Περιγραφή περίπτωσης χρήσης**

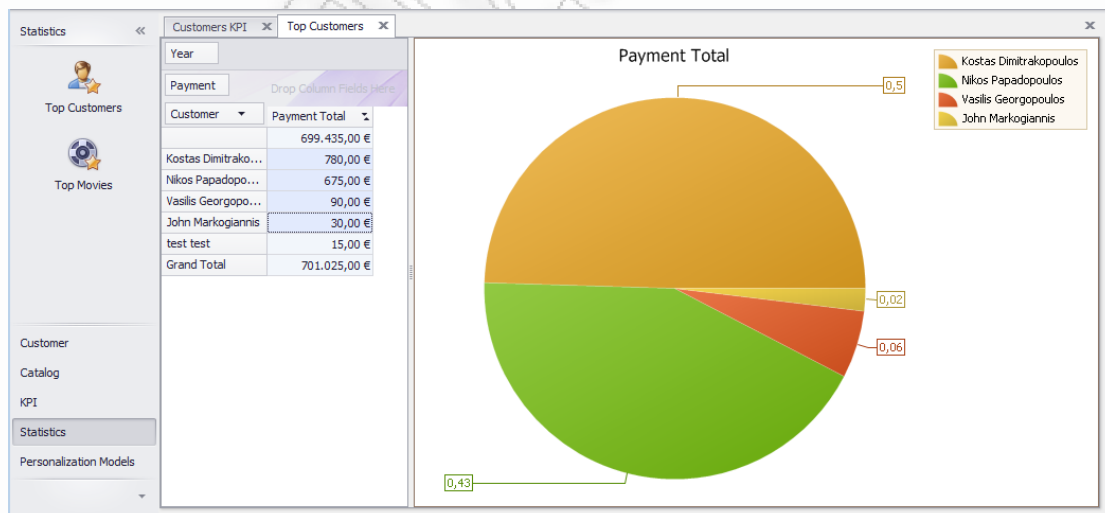
Η περίπτωση χρήσης “Top Movies” του συστήματός μας αποτελείται από την επιλογή του διαχειριστή της προβολής των ταινιών με την μεγαλύτερη αξία σε πωλήσεις.



**Εικόνα 15 Top Movies**

**Περίπτωση χρήσης 10: Top Customers**  
**Περιγραφή περίπτωσης χρήσης**

Η περίπτωση χρήσης “Top Customers” του συστήματός μας αποτελείται από την επιλογή του διαχειριστή της προβολής των χρηστών με την μεγαλύτερη αξία σε αγορές.



**Εικόνα 16 Top Customers**



## Περίπτωση χρήσης 11: Train Models

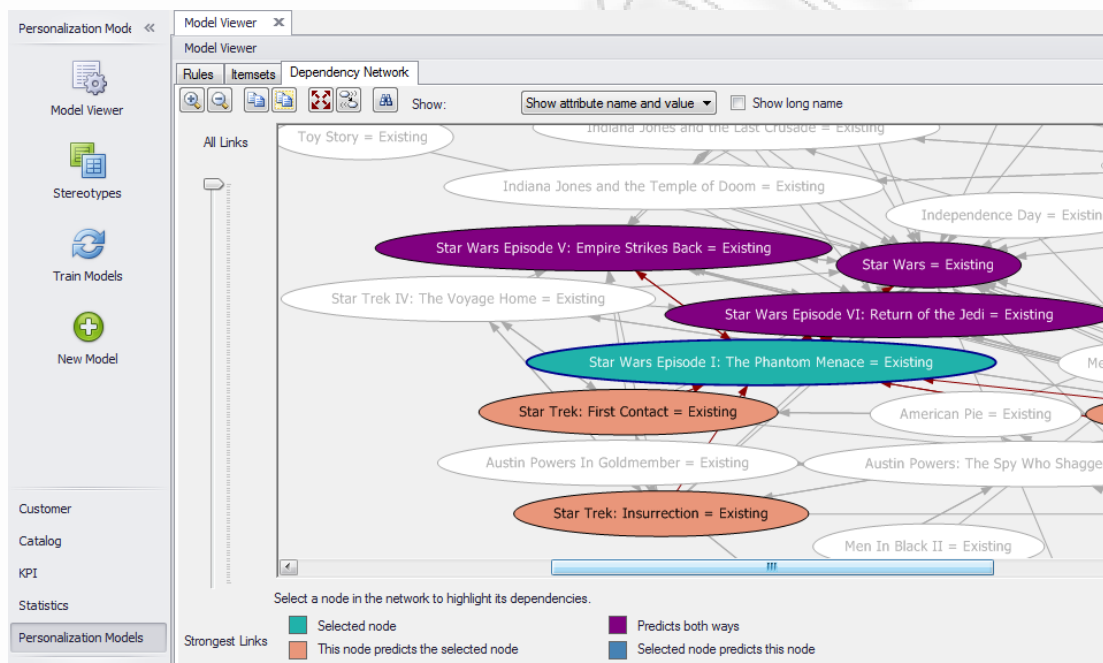
### Περιγραφή περίπτωσης χρήσης

Η περίπτωση χρήσης “Train Models” του συστήματός μας αποτελείται από την επιλογή του διαχειριστή της εκπαίδευσης του μοντέλου εξατομικευμένων συστάσεων λαμβάνοντας υπόψη τα νέα δεδομένα της βάσης.

## Περίπτωση χρήσης 12: Model Viewer

### Περιγραφή περίπτωσης χρήσης

Η περίπτωση χρήσης “Model Viewer” του συστήματός μας αποτελείται από την οθόνη του διαχειριστή της απεικόνισης των αποτελεσμάτων του μοντέλου εξατομικευμένων συστάσεων ώστε μέσω αυτής να απεικονίζονται οι συσχετίσεις των ταινιών με σκοπό την ευκολότερη επαλήθευση και έλεγχο του μοντέλου.

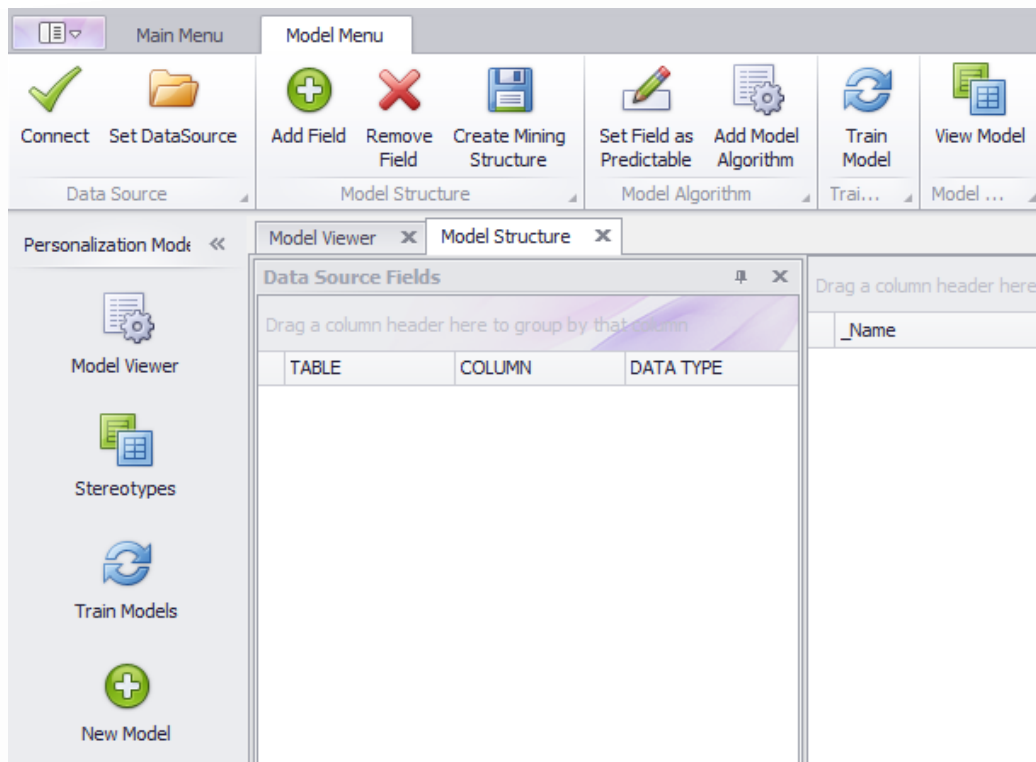


Εικόνα 17 Model Viewer

## Περίπτωση χρήσης 12: New Model

### Περιγραφή περίπτωσης χρήσης

Η περίπτωση χρήσης “New Model” του συστήματός μας αποτελείται τη δυνατότητα του διαχειριστή μέσω του ειδικά δημιουργημένου σύστημα δυναμικής δημιουργίας μοντέλων εξατομικευμένων συστάσεων να δημιουργήσει το δικό του μοντέλο με τον αλγόριθμο που επιθυμεί και με τις παράμετρους που αυτός διαμορφώνει.



Εικόνα 18 New Model

## 4.3 Έγγραφο περιγραφής αρχιτεκτονικού σχεδίου

### 4.3.1 Εισαγωγή

Στο συγκεκριμένο έγγραφο γίνεται παρουσίαση της τεκμηρίωσης του αρχιτεκτονικού σχεδίου ανάπτυξης του συστήματος “Smart Media”. Η υλοποίηση του συστήματος στηρίζεται σε έναν υπολογιστή ο οποίος πρέπει να έχει κάποια σύνδεση στο Internet για να μπορέσει να έχει πρόσβαση στη βάση δεδομένων των ταινιών. Η βάση δεδομένων του συστήματος δεν είναι αναγκαίο να βρίσκεται στον ίδιο υπολογιστή. Η τεκμηρίωση που ακολουθεί παρέχει μια αναλυτική περιγραφή του αρχιτεκτονικού σχεδίου του συστήματος “Smart Media” μέσω της περιγραφής των σημαντικότερων όψεών του.

### 4.3.2 Σκοπός

Σκοπός του εγγράφου αυτού είναι η παρουσίαση της αρχιτεκτονικής του πληροφοριακού συστήματος “Smart Media”, το οποίο παρέχει εξατομικευμένες υπηρεσίες συστάσεων ταινιών. Στο παρόν έγγραφο δίνεται από μακροσκοπική σκοπιά η σχεδίαση του πληροφοριακού συστήματος καθώς και η απεικόνιση των απαιτήσεων που έχουν τεθεί στο έγγραφο «Προδιαγραφών Απαιτήσεων από το Λογισμικό», σε αρχιτεκτονικές όψεις. Είναι σημαντικό να σημειώσουμε ότι η περιγραφή του αρχιτεκτονικού σχεδίου αφορά μόνο στο κομμάτι που αποτελεί τον πυρήνα του συστήματός μας.

### 4.3.3 Ενδιαφερόμενοι

Αποδέκτες του εγγράφου είναι αφενός οι χρήστες των υπηρεσιών που θα παρέχονται, και αφετέρου οι κατασκευαστές του πληροφοριακού συστήματος “Smart Media”.

#### 4.3.4 Αναφορές

Το έγγραφο «Προδιαγραφών Απαιτήσεων από το Λογισμικό Σύστημα» το οποίο παρουσιάστηκε στην προηγούμενη παράγραφο καθώς και η σχετική βιβλιογραφία στο αντίστοιχο κεφάλαιο του παρόντος τόμου.

#### 4.3.5 Αρχιτεκτονικές αποφάσεις

Στη συνέχεια δίνονται οι σημαντικότερες αρχιτεκτονικές αποφάσεις που χρειάστηκε να ληφθούν για την υλοποίηση του λογισμικού συστήματος.

##### Πλατφόρμα (υλιστική, λογισμική)

Ως υλιστική πλατφόρμα, το σύστημά μας αποτελείται από έναν προσωπικό υπολογιστή (PC). Ο προσωπικός υπολογιστής λειτουργεί ως client της εφαρμογής μας και απεικονίζει και διαχειρίζεται τα δεδομένα που λαμβάνει είτε μέσω του web service διασύνδεσης με τη βάση δεδομένων imdb είτε μέσω της επικοινωνίας του με τη βάση δεδομένων του πληροφοριακού συστήματος. Μια εποπτική εικόνα των υλιστικών συνιστωσών δίνεται στη συνέχεια μέσω της πλατφορμικής όψης.

Ως λογισμική πλατφόρμα ανάπτυξης και εκτέλεσης χρησιμοποιούνται:

Ο ηλεκτρονικός υπολογιστής στον οποίο θα εκτελείται η εφαρμογή θα πρέπει να μπορεί να εκτελέσει προγράμματα Microsoft .Net, δηλαδή θα πρέπει να υπάρχει το Microsoft .Net Framework 4, να έχει κάποια σύνδεση στο Internet υποστηρίζοντας Web Services και εφόσον στον ίδιο υπολογιστή εγκατασταθεί η βάση δεδομένων πρέπει να είναι εγκατεστημένος ο Sql Server 2008. Για τη συντήρηση θα χρησιμοποιηθεί η ίδια λογισμική πλατφόρμα με αυτήν της ανάπτυξης και της εκτέλεσης. Δεν προβλέπεται χρησιμοποίηση κάποιου ιδιαίτερου λογισμικού για τη συντήρηση.

##### Περιβάλλον προγραμματισμού

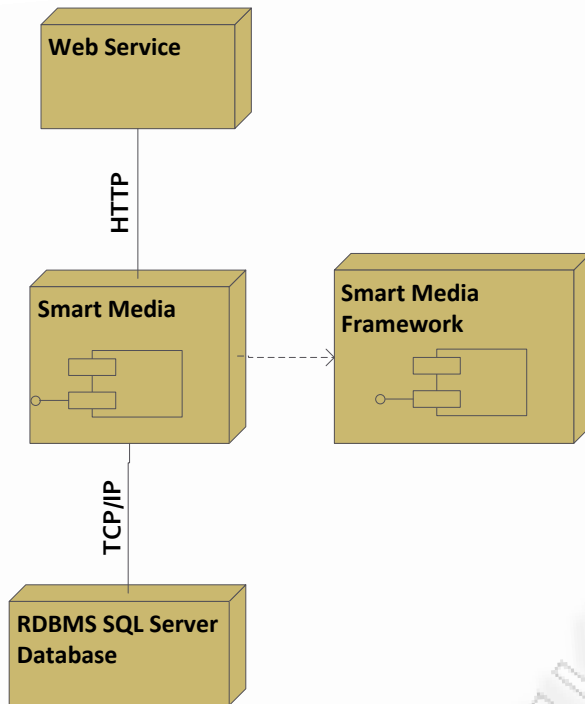
Στην ανάπτυξη του εν λόγω λογισμικού χρησιμοποιήθηκε το αντικειμενοστραφές περιβάλλον προγραμματισμού Visual Studio .Net 2010. Ως γλώσσα προγραμματισμού θα χρησιμοποιηθεί η C#. Ο λόγος που επιλέχθηκε το Visual Studio .Net είναι ότι το συγκεκριμένο περιβάλλον υποστηρίζει την ανάπτυξη εφαρμογών που υποστηρίζονται από όλους τους προσωπικούς υπολογιστές. Έτσι, το σύστημα που θα αναπτυχθεί θα μπορεί να τρέχει σε οποιαδήποτε πλατφόρμα και πάνω από οποιοδήποτε λειτουργικό σύστημα εφόσον έχει εγκατεστημένο το Microsoft .Net Framework 4.

##### Γλώσσα προγραμματισμού

Η γλώσσα προγραμματισμού είναι η C# μαζί με τις διάφορες διεπαφές (interfaces) που απαιτούνται από τις απαραίτητες βιβλιοθήκες για την υποστήριξη γραφικού περιβάλλοντος και την επικοινωνία μεταξύ εφαρμογής και των διαδικτυακών υπηρεσιών.

##### Πλατφορμική

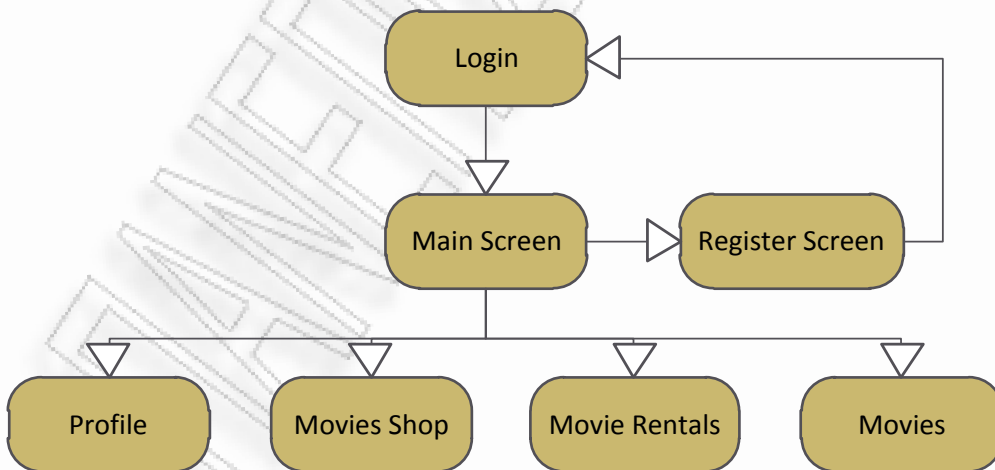
Στο επόμενο διάγραμμα παρουσιάζουμε την πλατφορμική όψη του συστήματός μας. Η πλατφορμική όψη περιγράφει με ένα διάγραμμα διανομής (deployment diagram) της UML, τους υπολογιστές καθώς και τις άλλες συσκευές που λαμβάνουν μέρος στην εκτέλεση του Smart Media καθώς και τα πρωτόκολλα της μεταξύ τους επικοινωνίας. Ακολουθεί λοιπόν το διάγραμμα διανομής με βάση τα όσα ειπώθηκαν στην ενότητα για λογισμική και υλισμική πλατφόρμα.



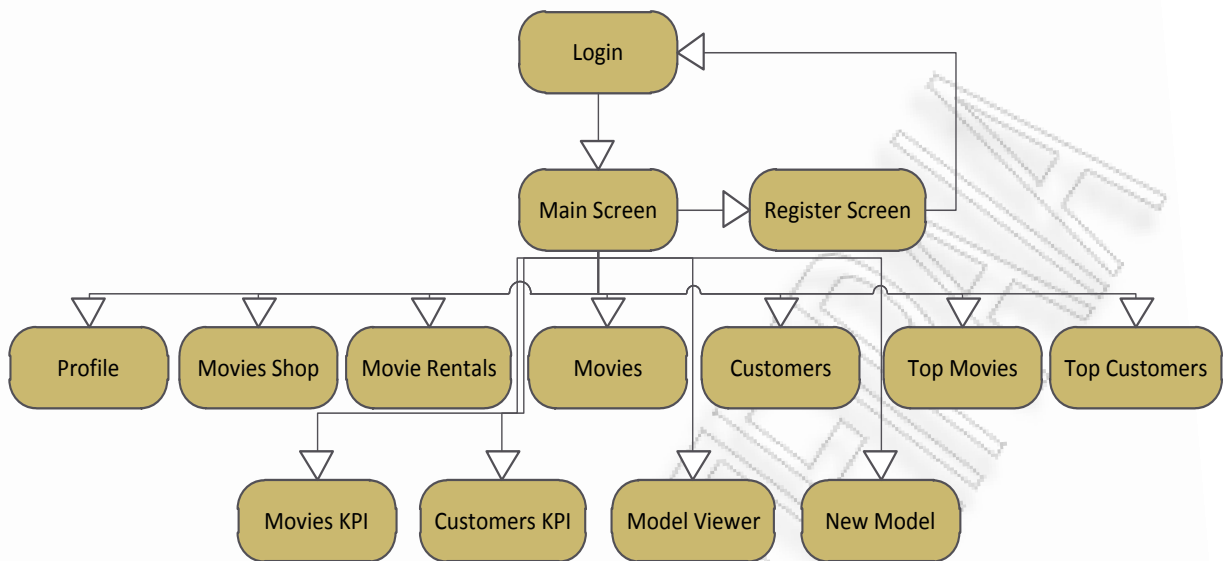
Εικόνα 19 Deployment Diagram

**Χρηστική**

Σε αυτή την ενότητα παρουσιάζονται οι διάφορες οθόνες που στην πραγματικότητα θα χρησιμοποιεί ο χρήστης του συστήματος Smart Media για να του παρέχονται οι διάφορες υπηρεσίες. Παρουσιάζεται έτσι το σύστημα από τη σκοπιά του τρόπου χρήσης του. Στο σχήμα που ακολουθεί φαίνεται το σύνολο των οθονών του User Interface του συστήματος ανάλογα με το ρόλο του συστήματος.



Εικόνα 20 User Interface User Role



Εικόνα 48 User Interface Administrator Role

## 4.4 Έγγραφο περιγραφής του λεπτομερούς σχεδίου

### 4.4.1 Σκοπός

Σκοπός αυτού του εγγράφου είναι η παρουσίαση του Λεπτομερούς Σχεδίου του συστήματος Smart Media.

### 4.4.2 Ενδιαφερόμενοι

Ενδιαφερόμενοι αυτού του εγγράφου είναι ο αρχιτέκτων του συστήματος, οι σχεδιαστές οι προγραμματιστές και οι ελεγκτές.

### 4.4.3 Βιβλιογραφία

Τα έγγραφα «Έγγραφο Προδιαγραφών Απαιτήσεων από το Λογισμικό Σύστημα», «Έγγραφο Περιγραφής της Αρχιτεκτονικής του Συστήματος».

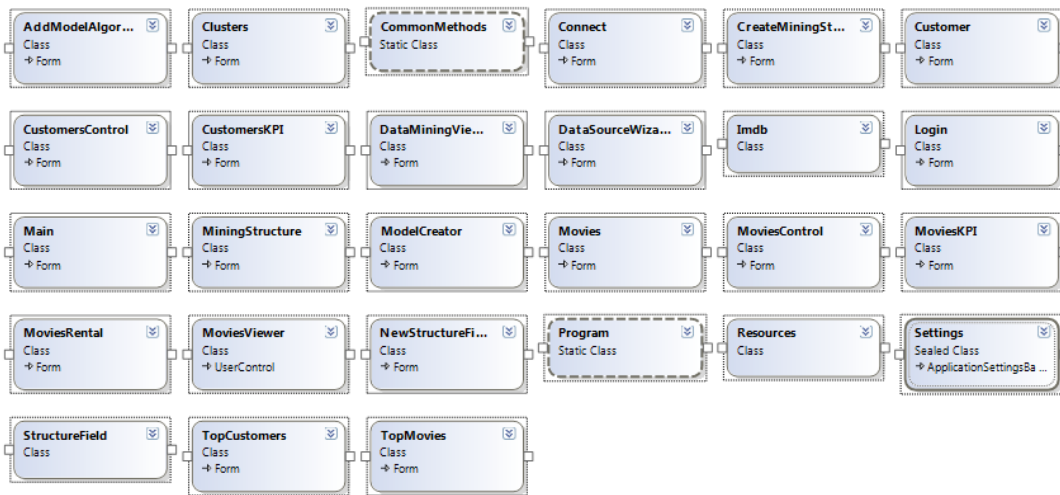
### 4.4.4 Σχεδιαστικές όψεις

#### Αποσυνθετική

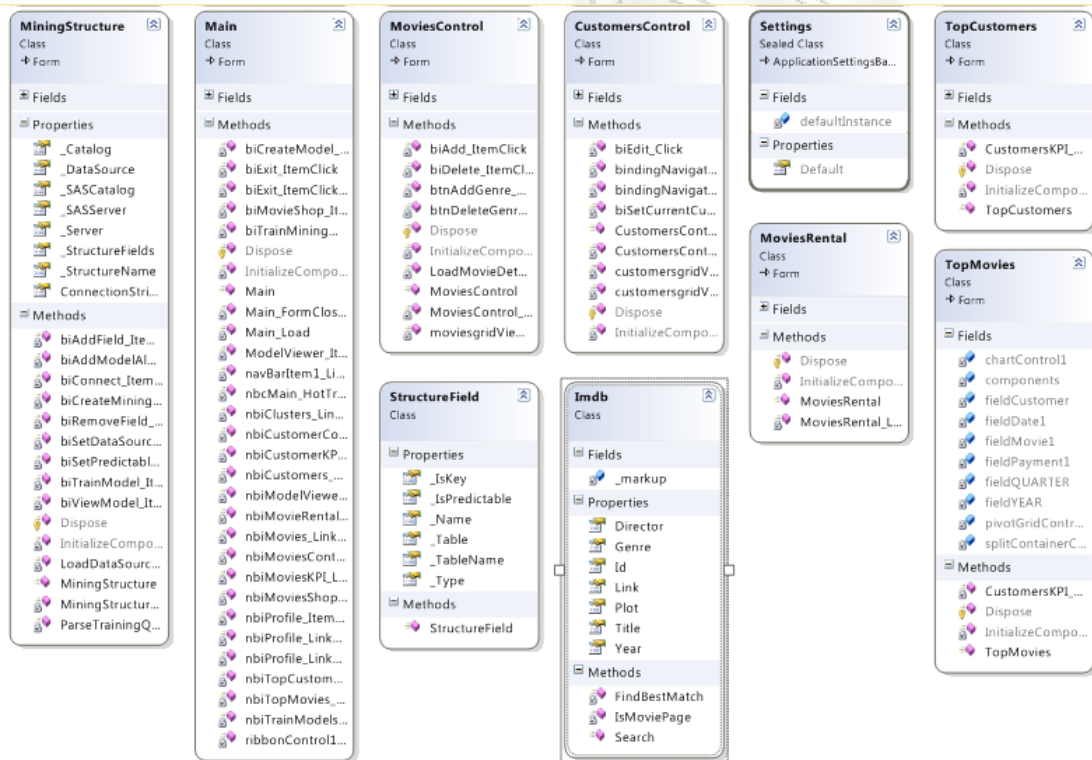
Η αποσυνθετική όψη των υπολογιστών του συστήματός μας όπως αυτοί ορίστηκαν και περιγράφηκαν στο «Έγγραφο Περιγραφής της Αρχιτεκτονικής του Συστήματος» φαίνονται στη συνέχεια.

#### Smart Media

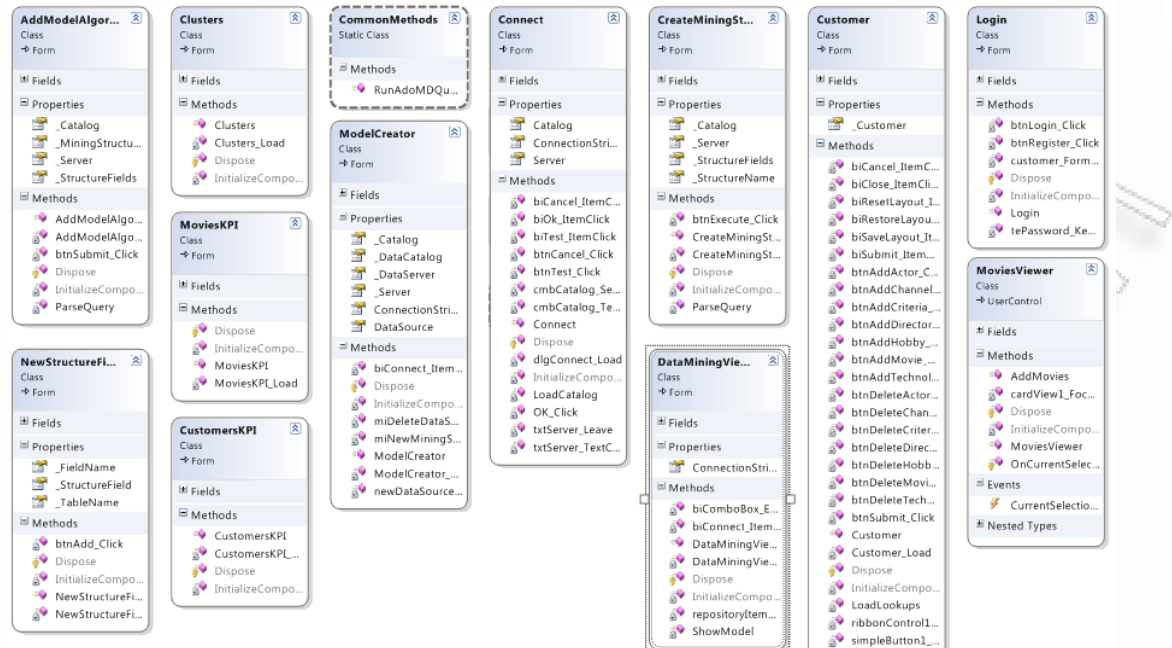
Η αρχιτεκτονική οντότητα UI (User Interface) του ειδικού μέρους αποσυντίθεται στις κλάσεις - classes που φαίνονται στο ακόλουθο διάγραμμα.



Εικόνα 21 Αποσυνθετική όψη του UI του συστήματος Smart Media (επιπεδο 0)



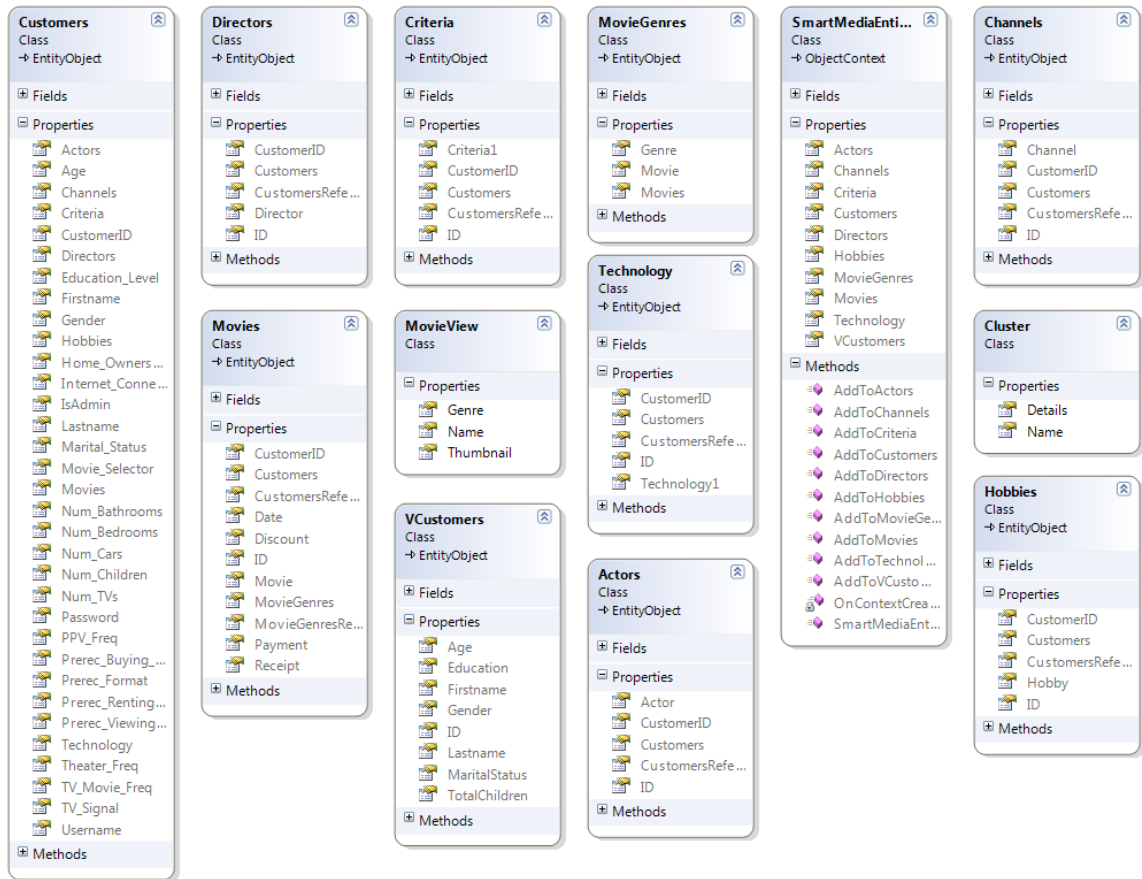




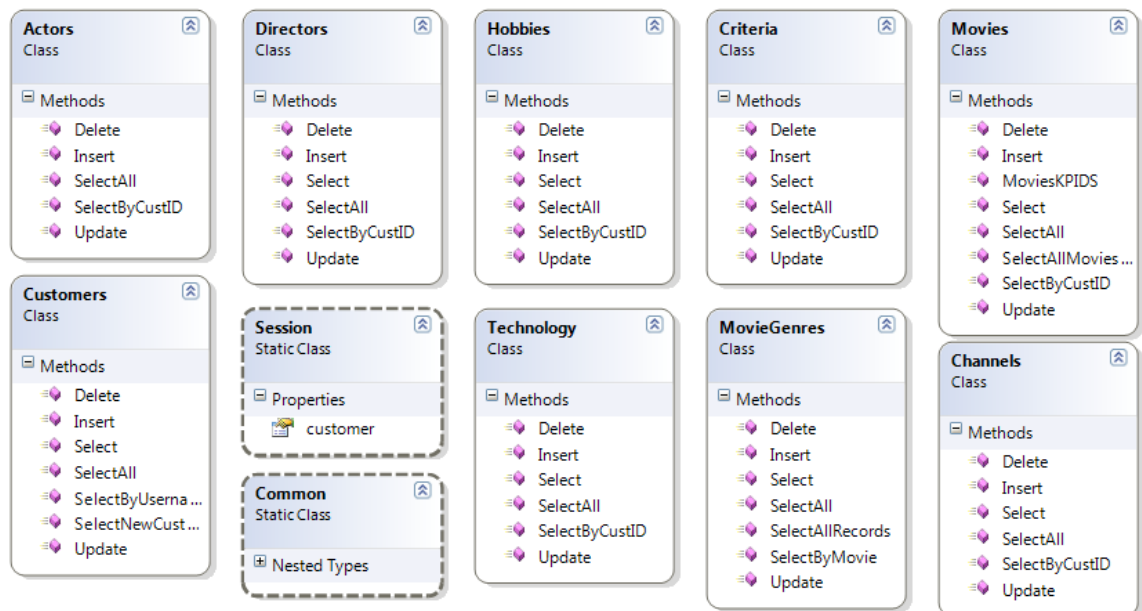
Εικόνα 22 Αποσυνθετική όψη του UI του συστήματος Smart Media (επιπεδο 1)



Εικόνα 23 Αποσυνθετική όψη του DAL του συστήματος Smart Media (επιπεδο 0)

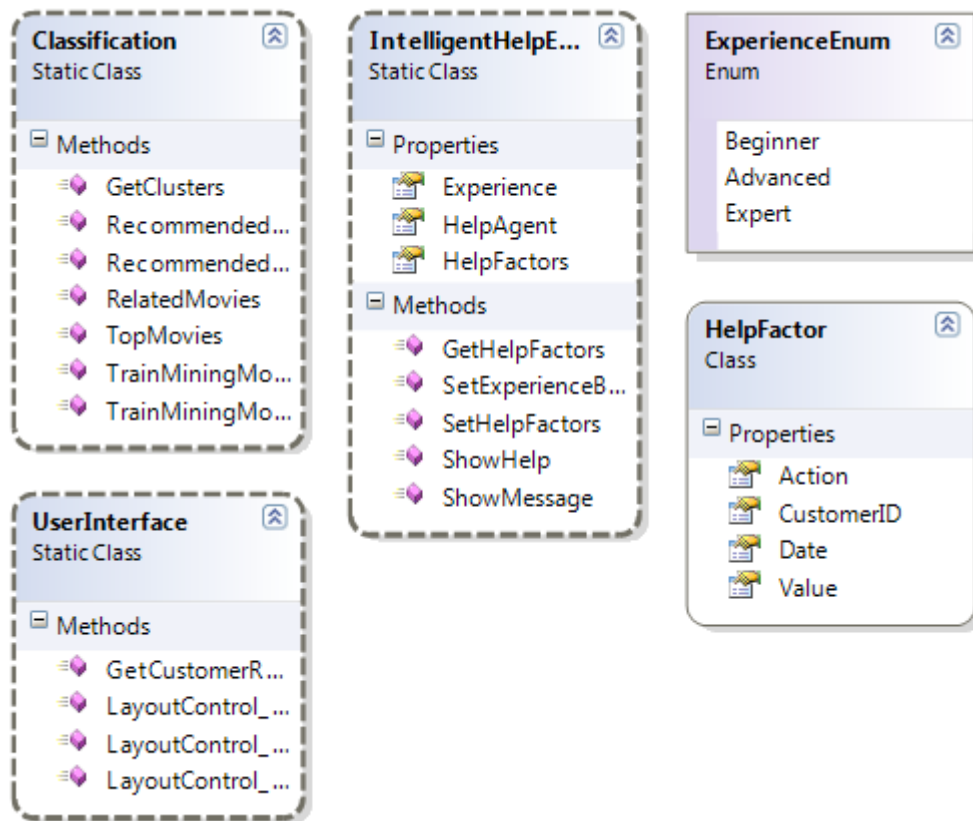


Εικόνα 24 Αποσυνθετική όψη του DAL του συστήματος Smart Media (επίπεδο 1)



Εικόνα 25 Αποσυνθετική όψη του BLL του συστήματος Smart Media



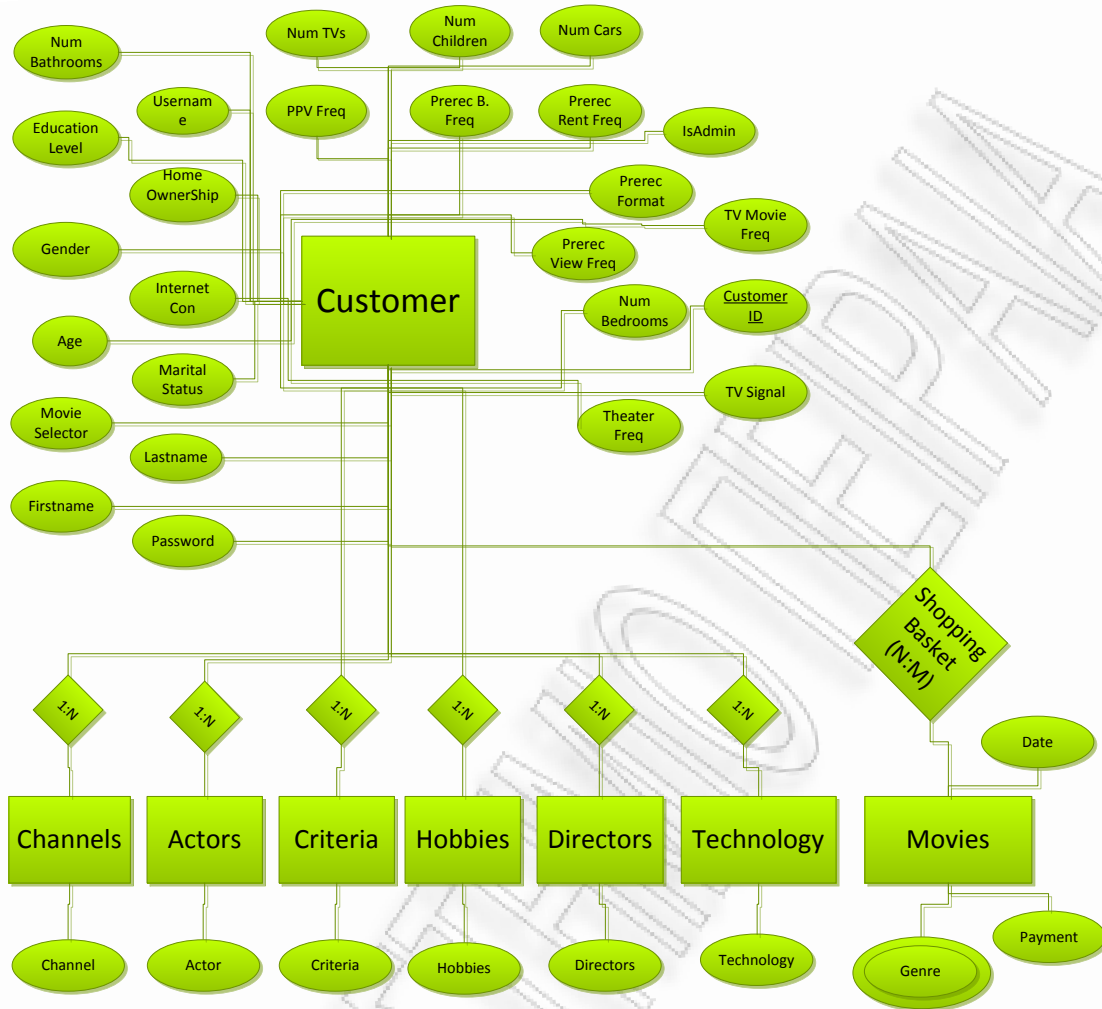


Εικόνα 26 Αποσυνθετική όψη του Framework του συστήματος Smart Media

#### 4.4.5 Σχεδιασμός της βάσης δεδομένων

##### Εννοιολογικός σχεδιασμός

Η φάση αυτή περιλαμβάνει την δημιουργία του εννοιολογικού σχήματος για την βάση δεδομένων, με χρήση εννοιολογικού μοντέλου υψηλού επιπέδου. Το εννοιολογικό σχήμα είναι μια περιεκτική περιγραφή των απαιτήσεων των χρηστών σχετικά με τα δεδομένα και περιλαμβάνει λεπτομερείς περιγραφές των τύπων δεδομένων, των συσχετίσεων και των περιορισμών. Για τον εννοιολογικό σχεδιασμό της βάσης δεδομένων χρησιμοποιείται το μοντέλο οντοτήτων συσχετίσεων (Entity Relationship Model).



Εικόνα 27 Διάγραμμα Οντοτήτων-Συσχείσεων ER

**Λογικός Σχεδιασμός**

Η φάση του λογικού σχεδιασμού περιλαμβάνει την δημιουργία ενός λογικού σχήματος για την βάση δεδομένων με τη χρήση του σχεσιακού μοντέλου (Relational Model), έτσι όπως αυτό προτάθηκε από τον E.F.Codd (1970). Ο λογικός σχεδιασμός, ο οποίος αποτελεί συνέχεια του εννοιολογικού σχεδιασμού, αναπαριστά τους πίνακες από τους οποίους αποτελείται η βάση δεδομένων της εφαρμογής Smart Media

Πίνακας «Actors» - Οντότητα

	Column Name	Data Type	Allow Nulls
	CustomerID	int	<input type="checkbox"/>
	Actor	nvarchar(40)	<input checked="" type="checkbox"/>
	ID	uniqueidentifier	<input type="checkbox"/>

Εικόνα 28 Πίνακας «Actors» - Οντότητα

Πίνακας «Shopping Basket» - Οντότητα

	Column Name	Data Type	Allow Nulls
▶	CustomerID	int	<input type="checkbox"/>
	Movie	nvarchar(100)	<input checked="" type="checkbox"/>
🔑	ID	uniqueidentifier	<input type="checkbox"/>

Εικόνα 29 Πίνακας «Shopping Basket» - Οντότητα

## Πίνακας «Channels» - Οντότητα

	Column Name	Data Type	Allow Nulls
▶	CustomerID	int	<input type="checkbox"/>
	Channel	nvarchar(50)	<input checked="" type="checkbox"/>
🔑	ID	uniqueidentifier	<input type="checkbox"/>

Εικόνα 30 Πίνακας «Channels» - Οντότητα

## Πίνακας «Criteria» - Οντότητα

	Column Name	Data Type	Allow Nulls
▶	CustomerID	int	<input type="checkbox"/>
	Criteria	nvarchar(50)	<input checked="" type="checkbox"/>
🔑	ID	uniqueidentifier	<input type="checkbox"/>

Εικόνα 31 Πίνακας «Criteria» - Οντότητα

## Πίνακας «Customers» - Οντότητα

	Column Name	Data Type	Allow Nulls
▶	Age	int	<input checked="" type="checkbox"/>
	[Education Level]	nvarchar(25)	<input checked="" type="checkbox"/>
	Gender	nvarchar(10)	<input checked="" type="checkbox"/>
	[Home Ownership]	nvarchar(5)	<input checked="" type="checkbox"/>
	[Internet Connection]	nvarchar(25)	<input checked="" type="checkbox"/>
	[Marital Status]	nvarchar(15)	<input checked="" type="checkbox"/>
	[Movie Selector]	nvarchar(15)	<input checked="" type="checkbox"/>
	[Num Bathrooms]	real	<input checked="" type="checkbox"/>
	[Num Bedrooms]	int	<input checked="" type="checkbox"/>
	[Num Cars]	int	<input checked="" type="checkbox"/>
	[Num Children]	int	<input checked="" type="checkbox"/>
	[Num TVs]	int	<input checked="" type="checkbox"/>
	[PPV Freq]	nvarchar(10)	<input checked="" type="checkbox"/>
	[Prerec Buying Freq]	nvarchar(10)	<input checked="" type="checkbox"/>
	[Prerec Format]	nvarchar(25)	<input checked="" type="checkbox"/>
	[Prerec Renting Freq]	nvarchar(10)	<input checked="" type="checkbox"/>
	[Prerec Viewing Freq]	nvarchar(10)	<input checked="" type="checkbox"/>
🔑	CustomerID	int	<input type="checkbox"/>
	[Theater Freq]	nvarchar(10)	<input checked="" type="checkbox"/>
	[TV Movie Freq]	nvarchar(10)	<input checked="" type="checkbox"/>
	[TV Signal]	nvarchar(25)	<input checked="" type="checkbox"/>
	Password	nvarchar(50)	<input checked="" type="checkbox"/>
	Firstname	nvarchar(50)	<input checked="" type="checkbox"/>
	Lastname	nvarchar(50)	<input checked="" type="checkbox"/>
	IsAdmin	bit	<input checked="" type="checkbox"/>

Εικόνα 32 Πίνακας «Customers» - Οντότητα

## Πίνακας «Directors» - Οντότητα

	Column Name	Data Type	Allow Nulls
▶	CustomerID	int	<input type="checkbox"/>
	Director	nvarchar(40)	<input checked="" type="checkbox"/>
🔑	ID	uniqueidentifier	<input type="checkbox"/>

Εικόνα 33 Πίνακας «Directors» - Οντότητα

## Πίνακας «Hobbies» - Οντότητα

	Column Name	Data Type	Allow Nulls
▶	CustomerID	int	<input type="checkbox"/>
	Hobby	nvarchar(50)	<input checked="" type="checkbox"/>
🔑	ID	uniqueidentifier	<input type="checkbox"/>

Εικόνα 34 Πίνακας «Hobbies» - Οντότητα

## Πίνακας «Genres» - Οντότητα

	Column Name	Data Type	Allow Nulls
▶🔑	Movie	nvarchar(100)	<input type="checkbox"/>
	Genre	nvarchar(25)	<input checked="" type="checkbox"/>

Εικόνα 35 Πίνακας «Genres» - Οντότητα

## Πίνακας «Movies» - Οντότητα

	Column Name	Data Type	Allow Nulls
▶	CustomerID	int	<input type="checkbox"/>
	Movie	nvarchar(100)	<input checked="" type="checkbox"/>
🔑	ID	uniqueidentifier	<input type="checkbox"/>
	Receipt	nvarchar(50)	<input checked="" type="checkbox"/>
	Date	datetime	<input checked="" type="checkbox"/>
	Payment	money	<input checked="" type="checkbox"/>
	Discount	money	<input checked="" type="checkbox"/>

Εικόνα 36 Πίνακας «Movies» - Οντότητα

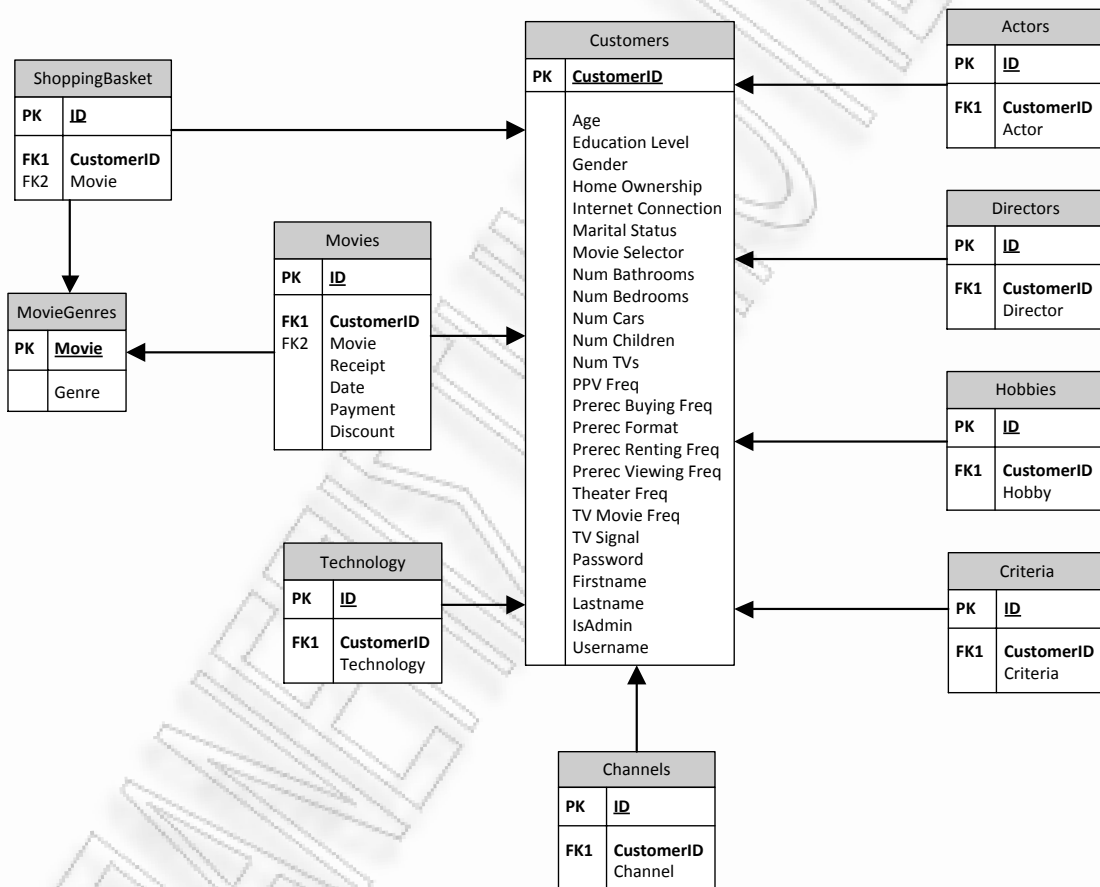
## Πίνακας «Technologies» - Οντότητα

	Column Name	Data Type	Allow Nulls
▶	CustomerID	int	<input type="checkbox"/>
	Technology	nvarchar(50)	<input checked="" type="checkbox"/>
🔑	ID	uniqueidentifier	<input type="checkbox"/>

Εικόνα 37 Πίνακας «Technologies» - Οντότητα

### Σχεσιακό Σχήμα

Στο σχεσιακό περιβάλλον βάσης δεδομένων, οι πίνακες πρέπει να είναι κανονικοποιημένοι τουλάχιστον στην πρώτη κανονική μορφή, δηλαδή να μην περιέχουν επαναλαμβανόμενες ομάδες δεδομένων. Ισοδύναμα η στήλη του κάθε πίνακα δε νοείται να αντιστοιχεί σε πεδίο που παίρνει πολλαπλές και όχι μια τιμή. Ένας πίνακας που είναι σε πρώτη κανονική μορφή (1Normal Form) είναι πίνακας σύμφωνα με το σχεσιακό μοντέλο, μπορεί όμως να παρουσιάζει προβληματική συμπεριφορά στις εισαγωγές, στις ενημερώσεις και στις διαγραφές των δεδομένων του. "Προβληματική συμπεριφορά" σημαίνει είτε δυνατότητα να συμβεί λάθος στην καταχώρηση - ενημέρωση των δεδομένων είτε σπατάλη χώρου καταχώρησης των δεδομένων στον δίσκο. Η θεωρία της κανονικοποίησης αναπτύχθηκε με στόχο να ανιχνεύεται και να προλαμβάνεται τέτοιου είδους προβληματική κατάσταση όταν η εφαρμογή είναι ακόμη στο στάδιο του σχεδιασμού της. Στην Κανονικοποίηση με διάσπαση, η κάθε περίπτωση μη κανονικοποιημένου πίνακα αντιμετωπίζεται με διαδοχικές διασπάσεις που καταλήγουν σε ένα ισοδύναμο σύνολο πινάκων όπου ο καθένας είναι σε τρίτη κανονική μορφή (3NF).



Εικόνα 38 Σχεσιακό Σχήμα

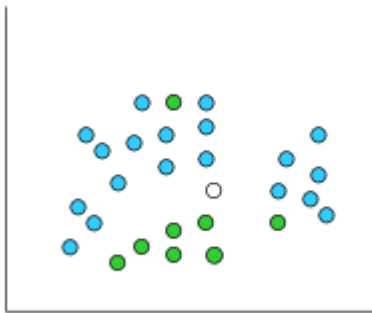
## 4.5 Εξατομικευμένο σύστημα συστάσεων

Ένα από τα σημαντικότερα χαρακτηριστικά των συστημάτων συστάσεων είναι ότι μπορούν να λειτουργήσουν εξατομικευμένα, δηλαδή γνωρίζοντας ποιον χρήστη εξυπηρετούν και υπό ποιες συνθήκες και για ποιο σκοπό. Συγκεκριμένα, η προτεινόμενη υβριδική προσέγγιση του συστήματος συστάσεων στοχεύει στην παραγωγή αποτελεσματικών συστάσεων για τους πελάτες ενός καταστήματος που νοικιάζει κινηματογραφικές ταινίες, επεκτείνοντας τις παραδοσιακές τεχνικές για τα αντικείμενα προς σύσταση. Η γνώση για τους πελάτες και τα προϊόντα προκύπτει από τα δεδομένα χρήσης (usage data), τη δομή του προφίλ του χρήστη σε συνδυασμό με το ιστορικό των συναλλαγών τόσο του ίδιου όσο και των υπολοίπων πελατών καθώς και από την εφαρμογή τεχνικών ταιριάσματος (matching) «όμοιων» (similar) πελατών. Παρακάτω γίνεται αναφορά στους αλγόριθμους εξόρυξης γνώσης που χρησιμοποιήθηκαν και πως αυτοί υλοποιήθηκαν και εφαρμόστηκαν.

### 4.5.1 Αλγόριθμοι και μοντέλα εξόρυξης δεδομένων

#### Αλγόριθμος Κατηγοριοποίησης με Δέντρα Απόφασης (classification)

Με τον όρο κατηγοριοποίηση εννοούμε την εκμάθηση μιας τεχνικής να προβλέπει την κλάση ενός στοιχείου επιλέγοντας από προκαθορισμένες τιμές.



Έτσι αν μας δοθεί μια βάση δεδομένων  $D=\{t_1, t_2, \dots, t_i\}$  και ένα σύνολο κατηγοριών -"κλάσεων"  $C=\{C_1, \dots, C_j\}$ , το **πρόβλημα κατηγοριοποίησης** έγκειται στον ορισμό μιας απεικόνισης  $f: D \rightarrow C$  όπου κάθε εγγραφή  $t_i$  ανατίθεται σε μία κλάση  $C_j$ . Ουσιαστικά, η κατηγοριοποίηση διαμερίζει τη  $D$  σε **κλάσεις ισοδυναμίας**. Η **πρόβλεψη** είναι παρόμοιο πρόβλημα, αλλά μπορεί να θεωρηθεί ότι έχει άπειρο αριθμό κλάσεων. Οι πιο κοινές τεχνικές κατηγοριοποίησης είναι τα δένδρα αποφάσεων, τα νευρωνικά δίκτυα και τεχνικές βασισμένες σε απόσταση ή σε στατιστικές μεθόδους. Πρώτα δημιουργείτε ένα μοντέλο μέσω της αξιολόγησης ενός συνόλου δεδομένων εκπαίδευσης (training data) (ή μέσω της γνώσης ειδικών του πεδίου) και ύστερα εφαρμόζεται το μοντέλο σε νέα δεδομένα. Ένας αλγόριθμος που χρησιμοποιείται συχνά για την επίλυση της κατηγοριοποίησης είναι τα δέντρα απόφασης. Ο αλγόριθμος των δέντρων απόφασης υποστηρίζει τόσο την ταξινόμηση και την οπισθοδρόμηση και λειτουργεί καλά για την προγνωστική μοντελοποίηση. Χρησιμοποιώντας τον αλγόριθμο, μπορούμε να προβλέψουμε τόσο διακριτά όσο και συνεχή χαρακτηριστικά. Κατά τη δημιουργία του μοντέλου, ο αλγόριθμος εξετάζει πώς κάθε χαρακτηριστικό εισόδου από το σύνολο των δεδομένων επηρεάζει το αποτέλεσμα του προβλεπόμενου χαρακτηριστικού και στη συνέχεια χρησιμοποιεί τα χαρακτηριστικά εισόδου με την ισχυρότερη σχέση για να δημιουργήσει μια σειρά από κόμβους. Δεδομένου ότι οι νέοι κόμβοι προστίθενται στο μοντέλο, μια δομή δέντρου αρχίζει να σχηματίζεται. Ο κορυφαίος κόμβος του δέντρου περιγράφει την κατανομή του προβλεπόμενου χαρακτηριστικού πάνω στο συνολικό πληθυσμό. Κάθε επιπλέον κόμβος δημιουργείται με βάση την κατανομή των καταστάσεων του προβλεπόμενου χαρακτηριστικού σε σύγκριση με τα χαρακτηριστικά εισόδου. Εάν ένα χαρακτηριστικό εισόδου θεωρείται ότι προκαλεί το προβλεπόμενο χαρακτηριστικό ώστε να αλλάξει κατάσταση, τότε ένας νέος κόμβος προστίθεται στο μοντέλο. Το μοντέλο συνεχίζει να



αυξάνεται έως ότου κανένα από τα υπόλοιπα χαρακτηριστικά δημιουργήσουν μια διάσπαση που παρέχει μια βελτιωμένη πρόγνωση πάνω στο υπάρχον κόμβο. Το μοντέλο επιδιώκει να βρει έναν συνδυασμό χαρακτηριστικών και καταστάσεων ώστε να δημιουργεί μια δυσανάλογη κατανομή των καταστάσεων στο προβλεπόμενο χαρακτηριστικό, συνεπώς, επιτρέποντάς τη πρόβλεψη του αποτελέσματος του προβλεπόμενου χαρακτηριστικού.[2][5]

Παρακάτω παρουσιάζεται ο κώδικας της δημιουργίας ενός μοντέλου εξόρυξης γνώσης που βασίζεται στον αλγόριθμο των δέντρων απόφασης και τα αποτελεσμάτα του έτσι όπως αυτά εμφανίζονται στον τελικό χρήστη μέσω του πιλοτικού πληροφοριακού συστήματος Smart Media.

```
// Create datasource for queries
CALL ASSprocs.CreateDataSource('SmartMedia',
  'Provider=SQLNCLI10.1;Data Source=localhost;
  Integrated Security=SSPI;Initial Catalog=SmartMedia',
  'Default','','')
;
GO

// Listing 7.7 Associative Model Creation
CREATE MINING MODEL RecommendedMovies
(
  CustomerID          LONG KEY,
  Gender              TEXT DISCRETE,
  MaritalStatus      TEXT DISCRETE,
  Age                LONG DISCRETE,
  Movies              TABLE PREDICT
  (
    Movie TEXT KEY
  )
)
USING Microsoft_Decision_Trees
GO

// Listing 7.8 Associative Model Training
INSERT INTO RecommendedMovies
( CustomerId, Gender, MaritalStatus, Age,
  Movies (SKIP, Movie))
SHAPE
{
  OPENQUERY (SmartMedia,
  'SELECT CustomerId, Gender, [Marital Status], Age
  FROM Customers ORDER BY CustomerID')
}
APPEND
(
  {OPENQUERY (SmartMedia,
  'SELECT CustomerId, Movie
  FROM Movies
  WHERE NOT Movie IS NULL
  ORDER BY CustomerID')}
  RELATE CustomerID TO CustomerID
) AS Movies
GO
```

```
// Listing 7.9 Associative Singleton Prediction
```

```

SELECT
    t.CustomerID,
    Predict(RecommendedMovies.Movies,5, $AdjustedProbability) AS
Recommendation
FROM
    RecommendedMovies
NATURAL PREDICTION JOIN
(SELECT '' AS CustomerID, 'Female' AS Gender,
'Divorced' AS MaritalStatus, '27' AS Age,
(SELECT '' AS Movie)
    AS Movies) AS t

GO

// Listing 7.10 Decision Tree Stored Procedures
CALL System.GetTreeScores('RecommendedMovies')

GO

CALL System.DTGetNodes('RecommendedMovies')

GO

CALL System.DTGetNodeGraph('RecommendedMovies', 60)

GO

CALL System.DTAddNodes('RecommendedMovies', '36;34',
'99;282;20;261;26;201;33;269;30;187')

public static List<string>
RecommendedMoviesByDemographics(DAL.Customers customer)
{
    List<string> recommendedMovies = new List<string>();

    string DMX = @"SELECT
        t.CustomerID,
        Predict(RecommendedMovies.Movies,5,
$AdjustedProbability) AS Recommendation
        FROM
            RecommendedMovies
        NATURAL PREDICTION JOIN
        (
            SELECT '' + customer.CustomerID + @'' AS
CustomerID,
                '' + customer.Gender + @'' AS Gender,
                '' + customer.Marital_Status + @'' AS
MaritalStatus,
                '' + customer.Age + @'' AS Age,
                (SELECT '' AS Movie)
                AS Movies) AS t";

    AdomdConnection conn = new
        AdomdConnection("Data Source=localhost;
Catalog=SmartMedia; Integrated Security=SSPI");
    try
    {

```



```

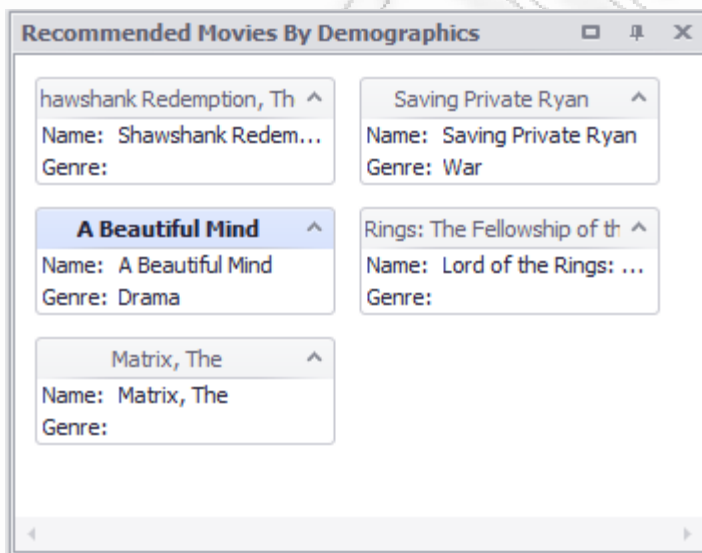
conn.Open();
AdomdCommand cmd = (AdomdCommand) conn.CreateCommand();
cmd.CommandText = DMX;
IDataReader dr = null;
dr = cmd.ExecuteReader();

//display the result in the listbox
while(dr.Read())
{
    IDataReader dr1 = (IDataReader) dr.GetValue(1);
    while (dr1.Read())
    {
        recommendedMovies.Add(dr1.GetString(0));
    }
}

return recommendedMovies;
}
catch(Exception ex)
{
    System.Windows.Forms.MessageBox.Show(ex.Message);
    return null;
}
finally
{
    conn.Close();
}
}

```

Τα αποτελέσματα του παραπάνω μοντέλου το οποίο βασίζεται σε κάποια δημογραφικά στοιχεία του χρήστη εμφανίζονται στο πληροφοριακό σύστημα στη παρακάτω οθονή:



Εικόνα 39 Recommended Movies By Demographics

### Επίδειξη Αλγορίθμου Δέντρων Αποφάσεων

Για την επίδειξη του αλγορίθμου των δέντρων αποφάσεων έχουμε κατασκευάσει ένα μοντέλο εξόρυξης γνώσης βασισμένο στην οικογενειακή κατάσταση και στο φύλο του χρήστη. Αρχικά κάνουμε σύνδεση στο σύστημα με έναν χρήστη αρσενικού φύλου με τα παρακάτω στοιχεία.

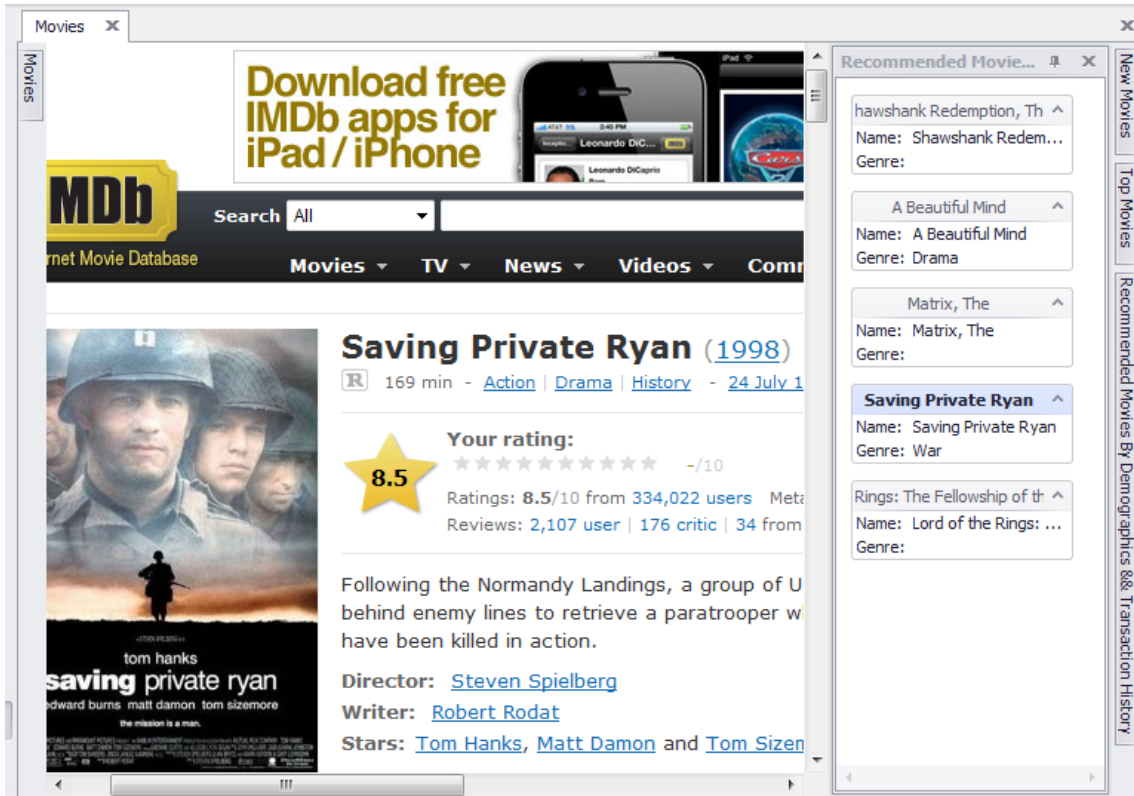
Εξατομικευμένο Σύστημα Συστάσεων

The screenshot shows a software window titled 'Customer' with a 'Customer Layout' tab. The window contains a form with various fields and dropdown menus. The fields are organized into two columns. The left column contains personal and contact information, while the right column contains household and media preferences. The form is set against a light purple background with a floral pattern.

Username:	giannis	Home Ownership:	Own
Password:	giannis	Internet Connection:	DSL
Firstname:	John	Movie Selector:	Children
Lastname:	Markogiannis	Num Bathrooms:	1
Age:	26	Num Bedrooms:	2
Gender:	Male	Num Cars:	1
Education Level:	Master's Degree	Num Children:	1
Marital Status:	Married	Num TVs:	1
PPV Freq:	Never	Theater Freq:	Daily
Prerec Buying Freq:	Monthly	TV Movie Freq:	Monthly
Prerec Format:	Laserdisk	TV Signal:	Digital Satellite
Prerec Renting Freq:	Daily	Prerec Viewing Freq:	Daily

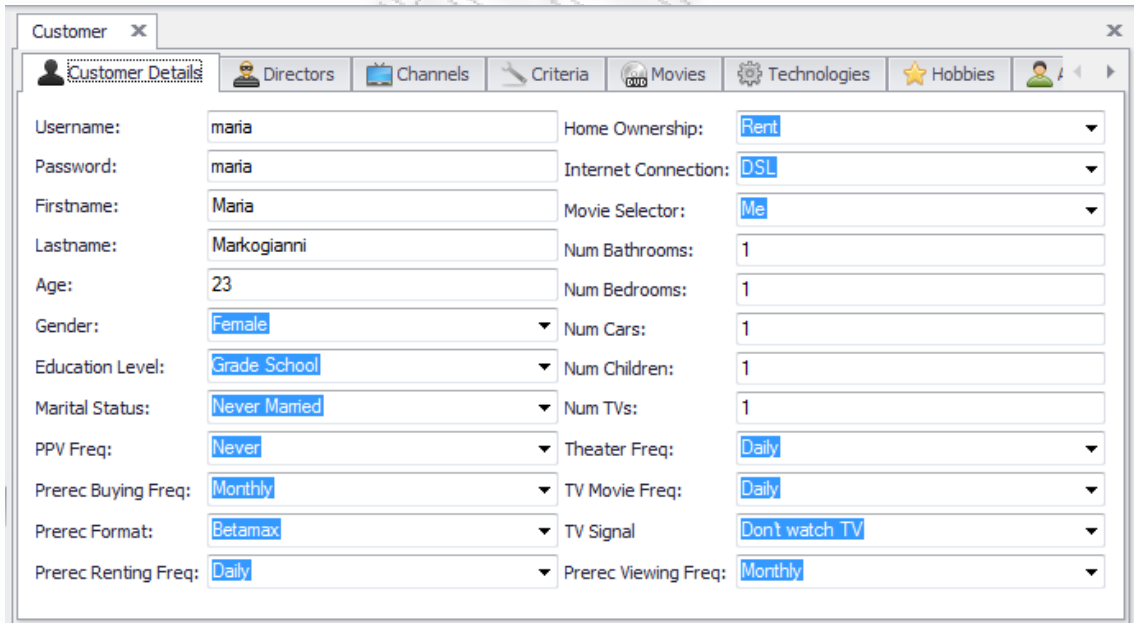
**Εικόνα 40 Male User**

Στη συνέχεια μεταφερόμαστε στην οθόνη του ηλεκτρονικού καταστήματος και παρατηρούμε τις συστημένες ταινίες σύμφωνα με τα δημογραφικά που προτίνει το σύστημα στον συγκεκριμένο χρήστη.



Εικόνα 41 Recommended Movies By Gender for John

Έπειτα συνδεόμαστε με έναν χρήστη φύλου θηλυκού με τα παρακάτω χαρακτηριστικά.



Εικόνα 42 Female User

Τέλος μεταφερόμαστε στην οθόνη του ηλεκτρονικού καταστήματος και παρατηρούμε τις αλλαγές με τις προηγούμενες συστάσεις στο είδος των ταινιών που προτείνει το σύστημα. Ενώ στον χρήστη John το σύστημα προτείνει ταινίες περιεχομένου πολεμικού, στο χρήστη Μαρία οι ταινίες ανήκουν στη κατηγορία ρομαντικές.

Εικόνα 43 Recommended Movies By Gender for Maria

### Αλγόριθμος Κανόνων Συσχετίσεων

Ο αλγόριθμος κανόνων συσχετίσεων έχει σχεδιαστεί ειδικά για χρήση σε αναλύσεις καλαθιού αγοράς. Ο αλγόριθμος θεωρεί κάθε χαρακτηριστικό / τιμή ζευγάρι (όπως το προϊόν / ποδήλατο) ως ένα στοιχείο (item). Ένα σύνολο στοιχείων (itemset) είναι ένας συνδυασμός των στοιχείων σε μία μόνο συναλλαγή. Ο αλγόριθμος σαρώνει το σύνολο δεδομένων προσπαθώντας να εντοπίσει itemsets που τείνουν να εμφανίζονται σε πολλές συναλλαγές.

Η παράμετρος support καθορίζει πόσες συναλλαγές του itemset πρέπει να εμφανίζονται, ώστε να θεωρείται σημαντικό. Για παράδειγμα, ένα συχνό itemset μπορεί να περιέχει {Φύλο = "Γυναίκα", Οικογενειακή Κατάσταση = "Παντρεμένη", Ηλικία = "30-35"}. Κάθε itemset έχει ένα μέγεθος, που είναι ο αριθμός των στοιχείων που περιέχει. Στην περίπτωση αυτή, το μέγεθος είναι 3. Συχνά τα μοντέλα συσχετίσεων δουλεύουν με σύνολα δεδομένων που περιέχουν ένθετους πίνακες, όπως μια λίστα των πελατών που ακολουθείται από έναν πίνακα με αγορές. Εάν ένας ένθετος πίνακας υπάρχει στο σύνολο δεδομένων, κάθε ένθετο κλειδί (όπως το προϊόν στον πίνακα αγορών) θεωρείται ένα στοιχείο. Ο αλγόριθμος συσχετίσεων βρίσκει επίσης τους κανόνες που σχετίζονται με τα σύνολα στοιχείων. Ένας κανόνας σε ένα μοντέλο συσχετίσεων αποτυπώνεται ως εξής:  $A, B \Rightarrow C$  (συνδέεται με μια πιθανότητα να συμβεί), όπου A, B, C είναι όλα τα συχνά itemsets. Η σχέση ' $\Rightarrow$ ' σημαίνει ότι το C προβλέπεται από το A και B. Το «κατώτατο όριο πιθανότητα» είναι μια παράμετρος που καθορίζει την ελάχιστη πιθανότητα να μπορεί να θεωρηθεί ένας κανόνας. Η πιθανότητα αυτή ονομάζεται επίσης "εμπιστοσύνη" (confidence) στην εξόρυξη δεδομένων. Τα μοντέλα συσχετίσεων είναι επίσης χρήσιμα για τις πολλαπλές πωλήσεις ή φίλτράρισμα. Για παράδειγμα, μπορούμε να χρησιμοποιήσουμε ένα μοντέλο συσχετίσεων για

να προβλέψουμε τα προϊόντα που ο χρήστης μπορεί να θέλει να αγοράσει με βάση άλλα προϊόντα στο καλάθι του.[2][5]

Παρακάτω παρουσιάζεται ο κώδικας της δημιουργίας ενός μοντέλου εξόρυξης γνώσης που βασίζεται στον αλγόριθμο των συσχετίσεων και τα αποτελεσμά του έτσι όπως αυτά εμφανίζονται στον τελικό χρήστη μέσω του πιλοτικού πληροφοριακού συστήματος Smart Media. Ο παραπάνω αλγόριθμος έχει χρησιμοποιηθεί τόσο στις εξατομικευμένες συστάσεις με βάση το προφίλ του χρήστη και το ιστορικό των αγορών του αλλά και στην εμφάνιση συσχετιζόμενων ταινιών με βάση τις προτιμήσεις των υπόλοιπων χρηστών.

```
// Create datasource for queries
CALL ASSprocs.CreateDataSource('SmartMedia,
    'Provider=SQLNCLI10.1;Data Source=localhost;
    Integrated Security=SSPI;Initial Catalog=SmartMedia',
    'Default','','')
;
GO

// Association Rules model analyzing relationships between movies
// and demographics. The first statement only creates a mining
// structure
// The modeling entity (case level concept) is Customer
// Simple Customer attributes, the demographics, appear as case level
// columns
// (e.g. Gender and Marital_Status)
// The nested table contains customer's Movie purchases
CREATE MINING STRUCTURE RelatedMovies (
    Customer_Id LONG KEY,
    Gender TEXT DISCRETE,
    Marital_Status TEXT DISCRETE,
    Age LONG DISCRETE,
    MoviePurchase TABLE(
        Movie_Name TEXT KEY
    )
)
GO

// An Association Rules mining model analyzing the information
// in the newly created structure
ALTER MINING STRUCTURE RelatedMovies
ADD MINING MODEL RelatedMovies (
    Customer_Id,
    Gender PREDICT,
    Marital_Status PREDICT,
    Age PREDICT,
    MoviePurchase PREDICT (
        Movie_Name
    )
)
GO

USING Microsoft_Association_Rules (Minimum_Support = 0.02,
    Minimum_Probability = 0.40)
GO
```

```

// Training a mining structure containing demographics and movie
purchases
//
INSERT INTO MINING STRUCTURE [RelatedMovies]
(
  Customer_Id, Gender, Marital_Status, Age,
  MoviePurchase (SKIP, [Movie_Name])
)
SHAPE
{
  OPENQUERY ([SmartMedia],
  'SELECT CustomerId, Gender, [Marital Status], Age FROM Customers
  ORDER BY CustomerId')
}
APPEND
(
  {
    OPENQUERY ([SmartMedia],
    'SELECT CustomerID, Movie
    FROM Movies
    WHERE NOT Movie IS NULL
    ORDER BY CustomerID')
  }
  RELATE CustomerId To CustomerId
)
AS Movie_Purchase
GO
// Retrieving all the frequent itemsets
//
SELECT Node_Description FROM RelatedMovies.CONTENT
WHERE Node_Type = 7
GO

// Retrieving all the rules
//
SELECT Node_Description FROM RelatedMovies.CONTENT
WHERE Node_Type = 8
GO

// Retrieving recommendations based on demographics
//
SELECT T.CustomerID, Predict (MoviePurchase, 5) AS Recommendation
FROM RelatedMovies
NATURAL PREDICTION JOIN
  OPENQUERY([SmartMedia],
  'SELECT TOP 500 CustomerID, Gender, [Marital Status], Age FROM
Customers') AS T
GO
// Retrieving recommendations based on demographics
// and transaction history
//

```



```

SELECT T.CustomerID, Predict (MoviePurchase, 5) AS Recommendation
FROM RelatedMovies
PREDICTION JOIN
SHAPE
{
  OPENQUERY ([SmartMedia],
  'SELECT CustomerId, Gender, [Marital Status],Age FROM Customers
  ORDER BY CustomerId')
}
APPEND
(
  {
    OPENQUERY ([SmartMedia],
    'SELECT CustomerID, Movie FROM Movies
    ORDER BY CustomerID')
  }
  RELATE CustomerId To CustomerId
)
AS Movie_Purchase AS T
ON
  RelatedMovies.Gender = t.Gender
AND RelatedMovies.Marital_Status = t.[Marital Status]
AND RelatedMovies.Age = t.Age
AND RelatedMovies.MoviePurchase.Movie_Name = t.Movie_Purchase.Movie
GO

// Using Predict flags to get additional information about
// the recommendation results for a customer
//
// The predictions returning a non-NULL NodeID are based on
// rules. The ones with a NULL Node ID are based on popularity
SELECT FLATTENED PredictAssociation(
  MoviePurchase,
  5,
  INPUT_ONLY,
  INCLUDE_STATISTICS,
  INCLUDE_NODE_ID)
FROM RelatedMovies NATURAL PREDICTION JOIN
(SELECT 'Male' AS Gender,
'Married' AS Marital_Status,
'27' AS Age,
(SELECT 'Alien' AS Movie_Name UNION
SELECT 'Raiders of the Lost Ark' AS Movie_Name ) AS
MoviePurchase) AS T

```

CustomerID	Recommendation						
877687	<input type="checkbox"/> Recommendation <table border="1"> <thead> <tr> <th>Movie_Name</th> </tr> </thead> <tbody> <tr> <td>Star Wars</td> </tr> <tr> <td>Matrix, The</td> </tr> <tr> <td>Lord of the Rings: The Fellowship of the Ring, The</td> </tr> <tr> <td>A Beautiful Mind</td> </tr> <tr> <td>Star Wars Episode V: Empire Strikes Back</td> </tr> </tbody> </table>	Movie_Name	Star Wars	Matrix, The	Lord of the Rings: The Fellowship of the Ring, The	A Beautiful Mind	Star Wars Episode V: Empire Strikes Back
Movie_Name							
Star Wars							
Matrix, The							
Lord of the Rings: The Fellowship of the Ring, The							
A Beautiful Mind							
Star Wars Episode V: Empire Strikes Back							
877723	<input type="checkbox"/> Recommendation						
877757	<input type="checkbox"/> Recommendation						
877792	<input type="checkbox"/> Recommendation						

Εικόνα 44 Microsoft Association Rules

Τα αποτελέσματα του παραπάνω μοντέλου το οποίο βασίζεται σε κάποια δημογραφικά στοιχεία του χρήστη αλλά και στο ιστορικό των αγορών του εμφανίζονται στο πληροφοριακό σύστημα στη παρακάτω οθονή:

```

public static List<string>
RecommendedMoviesByDemographicsAndTransactionHistory(int customerID)
{
    List<string> recommendedMovies = new List<string>();

    string DMX = @"SELECT T.CustomerID, Predict (MoviePurchase,
5) AS Recommendation
FROM RelatedMovies
PREDICTION JOIN
SHAPE
{
    OPENQUERY ([SmartMedia],
'SELECT CustomerId, Gender, [Marital
Status],Age FROM Customers
WHERE CustomerId = " + customerID +
@"
ORDER BY CustomerId')
}
APPEND
(
{
    OPENQUERY ([SmartMedia],
'SELECT CustomerID, Movie FROM
Movies
ORDER BY CustomerID')
}
RELATE CustomerId To CustomerId
)
AS Movie_Purchase AS T
ON
RelatedMovies.Gender = t.Gender

```



```
AND RelatedMovies.Marital_Status =
t.[Marital Status]
AND RelatedMovies.Age = t.Age
AND
RelatedMovies.MoviePurchase.Movie_Name = t.Movie_Purchase.Movie
";

AdomdConnection conn = new
AdomdConnection("Data Source=localhost;
Catalog=SmartMedia; Integrated Security=SSPI");
try
{
conn.Open();
AdomdCommand cmd = (AdomdCommand)conn.CreateCommand();
cmd.CommandText = DMX;
IDataReader dr = null;
dr = cmd.ExecuteReader();

//display the result in the listbox
while (dr.Read())
{
IDataReader dr1 = (IDataReader)dr.GetValue(1);
while (dr1.Read())
{
recommendedMovies.Add(dr1.GetString(0));
}
}

return recommendedMovies;
}
catch (Exception ex)
{
System.Windows.Forms.MessageBox.Show(ex.Message);
return null;
}
finally
{
conn.Close();
}
}
```



Εικόνα 45 Recommended Movies By Demographics & Transactions History

### Επίδειξη Αλγορίθμου Κανόνων Συσχετίσεων

Δημιουργήσαμε ένα μοντέλο με μία δομή που περιλαμβάνει το φύλο του χρήστη, την οικογενειακή του κατάσταση και το ιστορικό των συναλλαγών του. Στην αρχή χρησιμοποιούμε ένα χρήστη φύλου άνδρα με τα παρακάτω χαρακτηριστικά και ιστορικό συναλλαγών.

Customer Details		Directors	Channels	Criteria	Movies	Technologies	Hobbies	Actors
Username:	giannis	Home Ownership:	Own					
Password:	giannis	Internet Connection:	DSL					
Firstname:	John	Movie Selector:	Children					
Lastname:	Markogiannis	Num Bathrooms:	1					
Age:	26	Num Bedrooms:	2					
Gender:	Male	Num Cars:	1					
Education Level:	Master's Degree	Num Children:	1					
Marital Status:	Married	Num TVs:	1					
PPV Freq:	Never	Theater Freq:	Daily					
Prerec Buying Freq:	Monthly	TV Movie Freq:	Monthly					
Prerec Format:	Laserdisk	TV Signal:	Digital Satellite					
Prerec Renting Freq:	Daily	Prerec Viewing Freq:	Daily					

Εικόνα 46 Male User Profile

Movie	Date	Payment	Discount
The Apartment	8/10/2011	15,0000	0,1500
Repo Man	8/10/2011	15,0000	0,1500
The Princess Diaries	8/10/2011	15,0000	0,1500
Ali G: Indahouse	8/10/2011	15,0000	0,1500
200 Cigarettes	8/10/2011	15,0000	0,1500
Smoke	8/10/2011	15,0000	0,1500

Εικόνα 47 Ιστορικό Συναλλαγών John

Έπειτα μεταβαίνουμε στην οθόνη του ηλεκτρονικού καταστήματος και παρατηρούμε τις ταινίες που μας προτείνει το σύστημα σύμφωνα με τα δημογραφικά και στο ιστορικό των συναλλαγών.

The screenshot shows the IMDb website interface. At the top, there are navigation tabs for 'Movie Rentals' and 'Movies'. A banner for 'Download free IMDb apps for iPad/iPhone' is visible. The main search bar is set to 'All'. Below the search bar, there are navigation links for 'Movies', 'TV', 'News', 'Videos', and 'Com'. The central content area features a large movie poster for 'The Lord of the Rings: The Fellowship of the Ring (2001)'. To the right of the poster, the movie title is displayed in large text, followed by its rating (8.8) and a brief synopsis. Below the synopsis, the director's name 'Peter Jackson' is listed. On the right side of the page, there is a 'Recommended Movies' sidebar. This sidebar lists several movies with their names and genres: 'Star Wars' (Fantasy/SciFi), 'Matrix, The' (Genre:), 'The Fellowship of the Ring' (Genre:), 'A Beautiful Mind' (Drama), and 'Star Wars Episode V: Empire Strikes Back' (Genre:). The sidebar also includes navigation links for 'New Movies', 'Recommended Movies By Demographics', and 'Top Movies'.

Εικόνα 48 Recommended Movies By Profile &amp; Transactions for John

Στη συνέχεια κάνουμε σύνδεση στο σύστημα με τον χρήστη Μαρία που είναι γυναίκα με τα παρακάτω χαρακτηριστικά και ιστορικό συναλλαγών.

Customer Details

Username: maria Home Ownership: Rent

Password: maria Internet Connection: DSL

Firstname: Maria Movie Selector: Me

Lastname: Markogianni Num Bathrooms: 1

Age: 23 Num Bedrooms: 1

Gender: Female Num Cars: 1

Education Level: Grade School Num Children: 1

Marital Status: Never Married Num TVs: 1

PPV Freq: Never Theater Freq: Daily

Prerec Buying Freq: Monthly TV Movie Freq: Daily

Prerec Format: Betamax TV Signal: Don't watch TV

Prerec Renting Freq: Daily Prerec Viewing Freq: Monthly

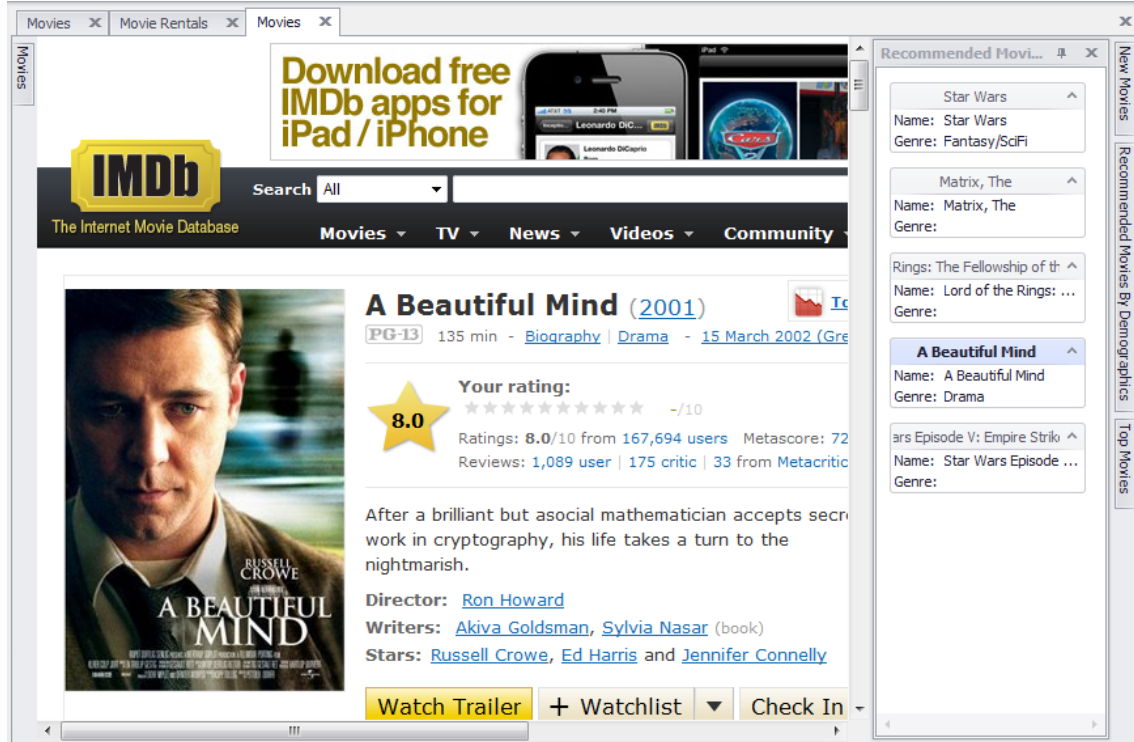
Εικόνα 49 Maria User Profile

Drag a column header here to group by that column

Movie	Date	Payment	Discount
When Harry Met Sally	8/10/2011	15,0000	0,1500
Breakdown ...	8/10/2011	15,0000	0,1500
Bridget Jones's Diary	8/10/2011	15,0000	0,1500
The Muppet Movie ...	8/10/2011	15,0000	0,1500
Bandit Queen ...	8/10/2011	15,0000	0,1500
The Muppet Movie ...	8/10/2011	15,0000	0,1500
Driving Miss Daisy ...	8/10/2011	15,0000	0,1500
After Life ...	8/10/2011	15,0000	0,1500
Funny Girl ...	8/10/2011	15,0000	0,1500
Naked Lunch ...	8/10/2011	15,0000	0,1500

Εικόνα 50 Maria Transactions History

Τέλος μεταβαίνουμε στην οθόνη του ηλεκτρονικού καταστήματος και παρατηρούμε τις συστηνόμενες ταινίες για τον χρήστη Maria.



Εικόνα 51 Recommended Movies By Profile & Transactions for Maria

Υλοποίηση αλγορίθμου συσχετιζόμενων ταινιών με βάση τις προτιμήσεις των χρηστών.

```
// Create datasource for queries
CALL ASSprocs.CreateDataSource('Smart Media',
    'Provider=SQLNCLI10.1;Data Source=localhost;
    Integrated Security=SSPI;Initial Catalog=SmartMedia',
    'Default','','')
;
GO

// Association Rules model analyzing relationships between movies
// The nested table contains customer's Movie purchases
CREATE MINING STRUCTURE MovieAssociation (
    Customer_Id LONG KEY,
    MoviePurchase TABLE(
        Movie_Name TEXT KEY
    )
)
GO

// An Association Rules mining model analyzing the information
// in the newly created structure
ALTER MINING STRUCTURE MovieAssociation
```

```
ADD MINING MODEL MovieAssociation (  
    Customer_Id,  
    MoviePurchase PREDICT (  
        Movie_Name  
    )  
)  
USING Microsoft_Association_Rules (Minimum_Support = 0.01,  
    Minimum_Probability = 0.30)  
GO
```

```
//Training a mining structure containing movie purchases
```

```
INSERT INTO MINING STRUCTURE [MovieAssociation]
```

```
(  
    Customer_Id,  
    MoviePurchase (SKIP, [Movie_Name])  
)  
SHAPE
```

```
{  
    OPENQUERY ([SmartMedia],  
        'SELECT CustomerId FROM Customers  
        ORDER BY CustomerId')  
}
```

```
APPEND
```

```
(  
    {  
        OPENQUERY ([SmartMedia],  
            'SELECT CustomerID, Movie FROM Movies  
            ORDER BY CustomerID')  
    }  
    RELATE CustomerId To CustomerId  
)  
AS Movie_Purchase
```

```
GO
```

```
// Retrieving all the frequent itemsets
```

```
SELECT Node_Description FROM MovieAssociation.CONTENT  
WHERE Node_Type = 7
```

```
GO
```

```
// Retrieving all the rules
```

```
SELECT Node_Description FROM MovieAssociation.CONTENT  
WHERE Node_Type = 8
```

```
GO
```

```
// Associative Batch prediction Query
```

```
SELECT
```

```
Εξατομικευμένο Σύστημα Συστάσεων
```

```

t.CustomerID,
  Predict (MoviePurchase, $AdjustedProbability, 5) AS Recommendation
FROM
  MovieAssociation
PREDICTION JOIN
  SHAPE {
    OPENQUERY ([SmartMedia],
      'SELECT
        CustomerID
      FROM
        Customers
      ORDER BY
        CustomerID')
  }
  APPEND ({
    OPENQUERY ([SmartMedia],
      'SELECT Movie, CustomerID
      FROM
        Movies
      ORDER BY
        CustomerID')
  })
  RELATE
    CustomerID TO CustomerID)
  AS
    NewCustomerPurchase AS t
ON
  MovieAssociation.MoviePurchase.Movie_Name =
  t.NewCustomerPurchase.Movie

// Singleton Associative Prediction
SELECT FLATTENED
  (Predict (MoviePurchase, 6)) AS [Recommendation]
FROM
  MovieAssociation
NATURAL PREDICTION JOIN
  (
    SELECT
      (
        (SELECT 'Jaws' AS Movie)
        UNION
        (SELECT 'Matrix' AS Movie)
      )
    AS Movies)
  AS NewCustomer

```



```

public static List<string> RelatedMovies(List<string> movies)
{
    List<string> relatedMovies = new List<string>();

    StringBuilder sb = new StringBuilder();
    foreach (string movie in movies)
    {
        sb.Append(@" (SELECT '"' + movie.Replace("'", "") + @"'
AS Movie) UNION");
    }
    //Remove Last UNION
    sb.Remove(sb.Length - 6, 6);

    string DMX = @"SELECT FLATTENED
(Predict(MoviePurchase, 6)) AS
[Recommendation]
FROM
    MovieAssociation
NATURAL PREDICTION JOIN
(
    SELECT
(
    " + sb.ToString() + @"
)
AS Movies)
AS NewCustomer
";

    AdomdConnection conn = new
    AdomdConnection("Data Source=localhost;
Catalog=SmartMedia; Integrated Security=SSPI");
    try
    {
        conn.Open();
        AdomdCommand cmd = (AdomdCommand)conn.CreateCommand();
        cmd.CommandText = DMX;
        IDataReader dr = null;
        dr = cmd.ExecuteReader();

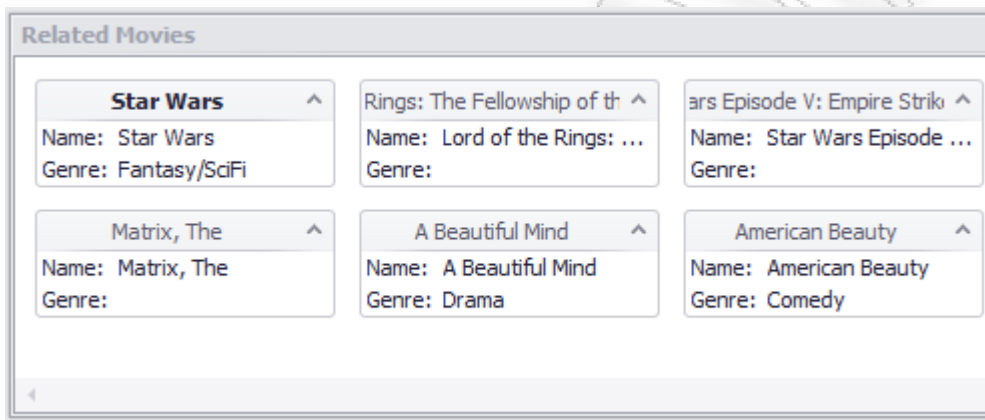
        //display the result in the listbox
        while (dr.Read())
        {
            relatedMovies.Add(dr.GetString(0));
        }
    }
}

```



```
        return relatedMovies;
    }
    catch (Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message);
        return null;
    }
    finally
    {
        conn.Close();
    }
}
```

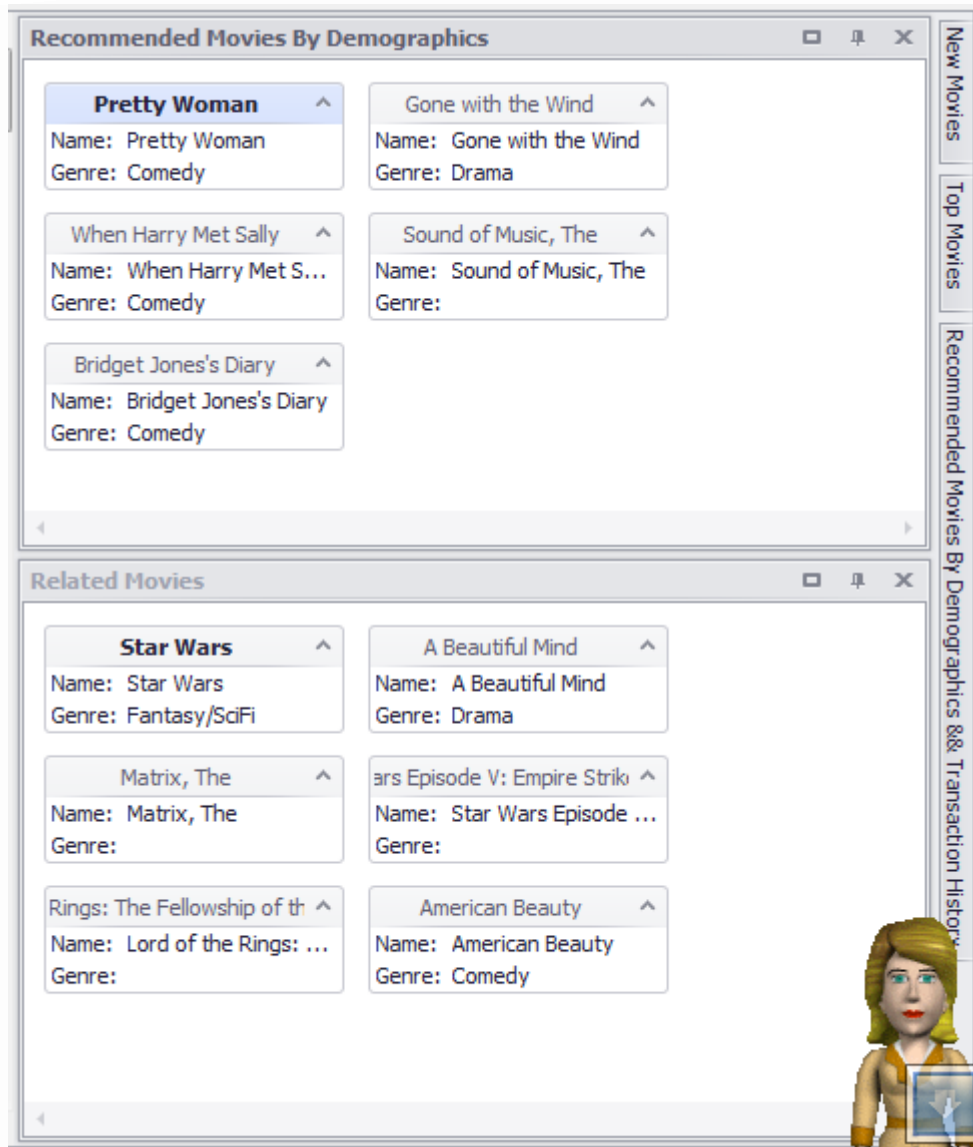
Τα αποτελέσματα του παραπάνω μοντέλου το οποίο βασίζεται στο ιστορικό των αγορών του κάθε χρήστη εμφανίζονται στο πληροφοριακό σύστημα στη παρακάτω οθονή:



Εικόνα 52 Related Movies

### Επίδειξη Αλγορίθμου Συσχετίσεων για Συσχετιζόμενες Ταινίες

Επιπλέον στο πληροφοριακό σύστημα έχουμε υλοποιήσει ένα μοντέλο συσχετιζόμενων ταινιών με τη χρήση του αλγορίθμου συσχετίσεων λαμβάνοντας υπόψη το ιστορικό των συναλλαγών των ταινιών. Οι συσχετιζόμενες ταινίες μίας επιλεγμένης ταινίας προκύπτουν από το ιστορικό των συναλλαγών των χρηστών που έχουν αγοράσει την επιλεγμένη ταινία και όχι από τη κατηγορία της επιλεγμένης ταινίας. Το αποτέλεσμα ενός παραδείγματος χρήσης με επιλεγμένη μία ταινία παρουσιάζεται παρακάτω.



Εικόνα 53 Movie Associations

Στο σημείο αυτό να τονίσουμε πως σε κάθε παραπάνω μοντέλο έχει γίνει μία απλή προσέγγιση με ορισμένες παραμέτρους για την επίδειξη των δυνατοτήτων του framework.

#### 4.6 Προσαρμοστικές Διεπαφές

Μια προσαρμοστική (adaptive) διεπαφή, δεν είναι τίποτε άλλο από μια διεπαφή που λαμβάνει υπόψη τις (δηλαδή «προσαρμόζεται») στις διαθέσιμες πληροφορίες, προκειμένου να παρουσιάσει μια βελτιστοποιημένη έκδοση της στο χρήστη. Μια προσαρμοστική διεπαφή λαμβάνει υπόψη διάφορους παράγοντες (συμπεριλαμβανομένων και των προτιμήσεων των χρηστών ή των περιβαλλοντικών συνθηκών) και λαμβάνει τις αποφάσεις σχετικά με τη σχεδίασή της στο χρόνο εκτέλεσης (runtime). Παραδείγματος χάριν, απεικονίζει τα κουμπιά μεγαλύτερα, καθιστά την οθόνη σκοτεινότερη, ή ανεβάζει την ένταση του ήχου). Ο στόχος είναι να διευκολυνθεί η χρήση του interface κάτω από τις τρέχουσες περιστάσεις. Είναι, συνεπώς, ένα δυναμικό interface, το οποίο όμως δεν αλλάζει τον τρόπο επικοινωνίας αυτόν καθ' εαυτόν όπως συμβαίνει με τα multimodal συστήματα, παρά μόνο τις λεπτομέρειες (οπτικές, ακουστικές, κτλ.)

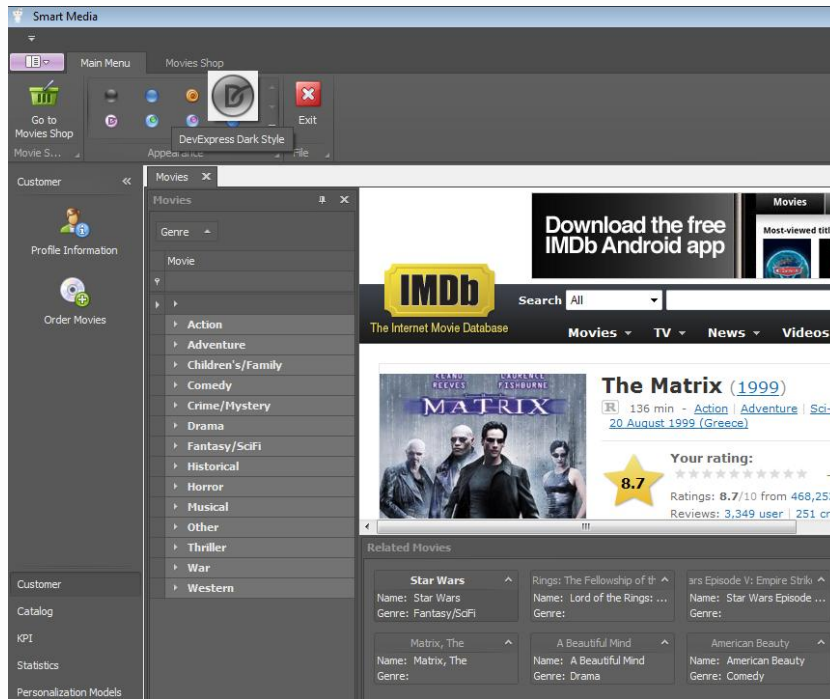
της παρουσίασης του περιεχομένου, χρησιμοποιώντας τον τρόπο επικοινωνίας που έχει (προ)επιλεγεί. Οι προσαρμοστικές διεπαφές επιτυγχάνονται στο συστημά μας μέσω ειδικού κώδικα που δίνει στο χρήστη τη δυνατότητα να προσαρμόζει τη κάθε οθόνη (π.χ. τη διάταξη των πεδίων και κάποιες ιδιοτητές τους). Παρακάτω παρουσιάζεται ο κώδικας και ένα παράδειγμα χρήσης μέσω screenshots.

```
public static void LayoutControl_SaveLayoutToXml(ref
LayoutControl layoutControl, string saveDirectory, string controlName)
{
    if (!Directory.Exists(saveDirectory))
    {
        Directory.CreateDirectory(saveDirectory);
    }
    layoutControl.SaveLayoutToXml(saveDirectory + @"\" +
controlName + "_Layout.xml");
}

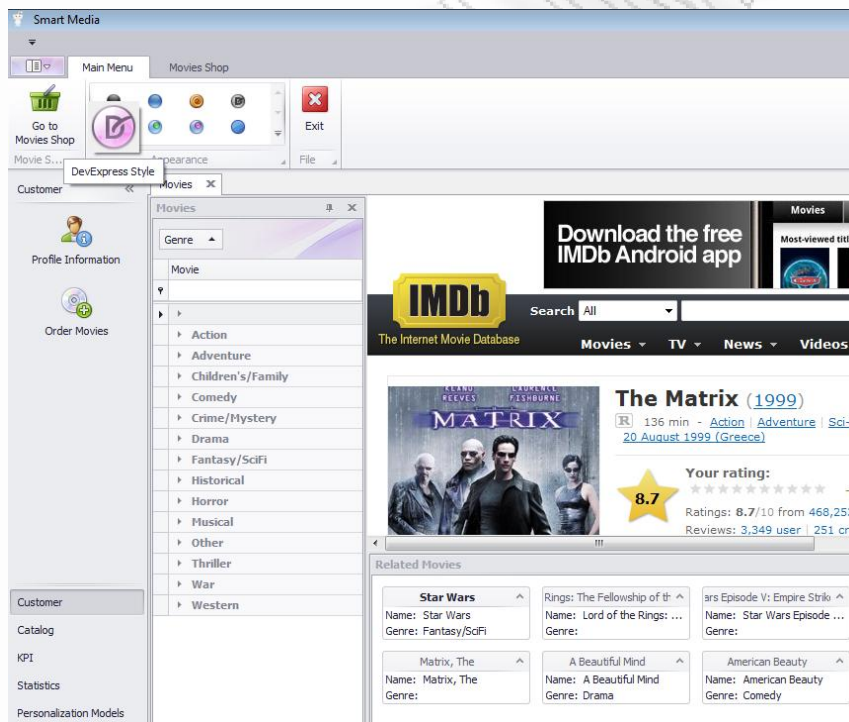
public static void LayoutControl_RestoreLayoutFromXml(ref
LayoutControl layoutControl, string saveDirectory, string controlName)
{
    if (File.Exists(saveDirectory + @"\" + controlName +
"_Layout.xml"))
    {
        layoutControl.RestoreLayoutFromXml(saveDirectory +
@"\" + controlName + "_Layout.xml");
    }
}

public static void LayoutControl_ResetLayoutToDefault(ref
LayoutControl layoutControl,
string saveDirectory, string controlName)
{
    if (File.Exists(String.Format(@"{0}\{1}_Layout.xml",
saveDirectory, controlName)))
    {
        File.Delete(saveDirectory + @"\" + controlName +
"_Layout.xml");
    }
}
```

### 4.6.1 Αλλαγή Theme

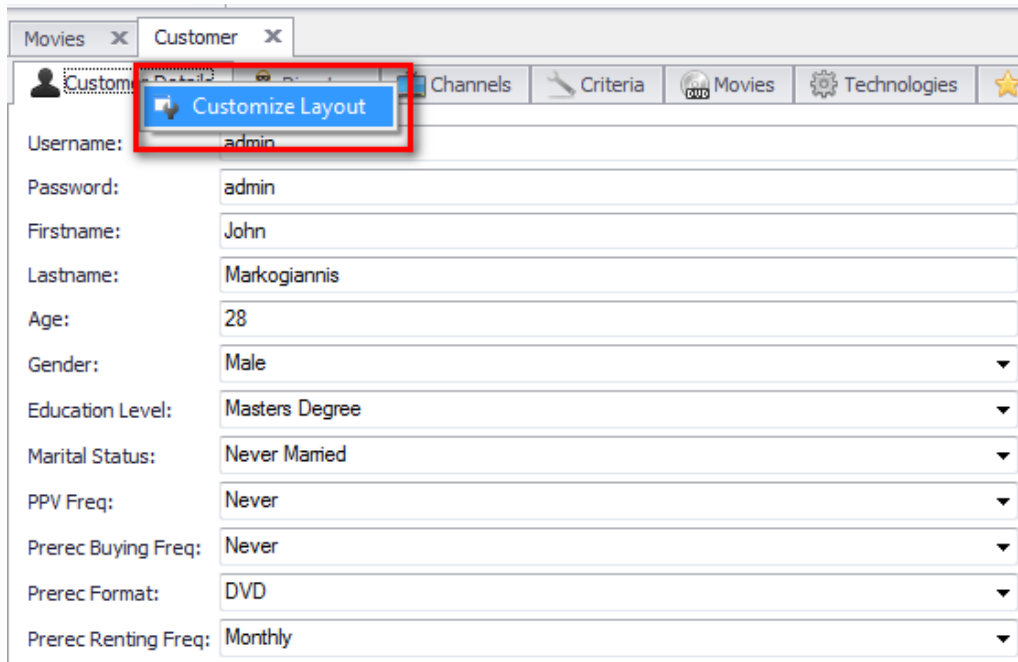


Εικόνα 54 Αλλαγή Theme Dark Style

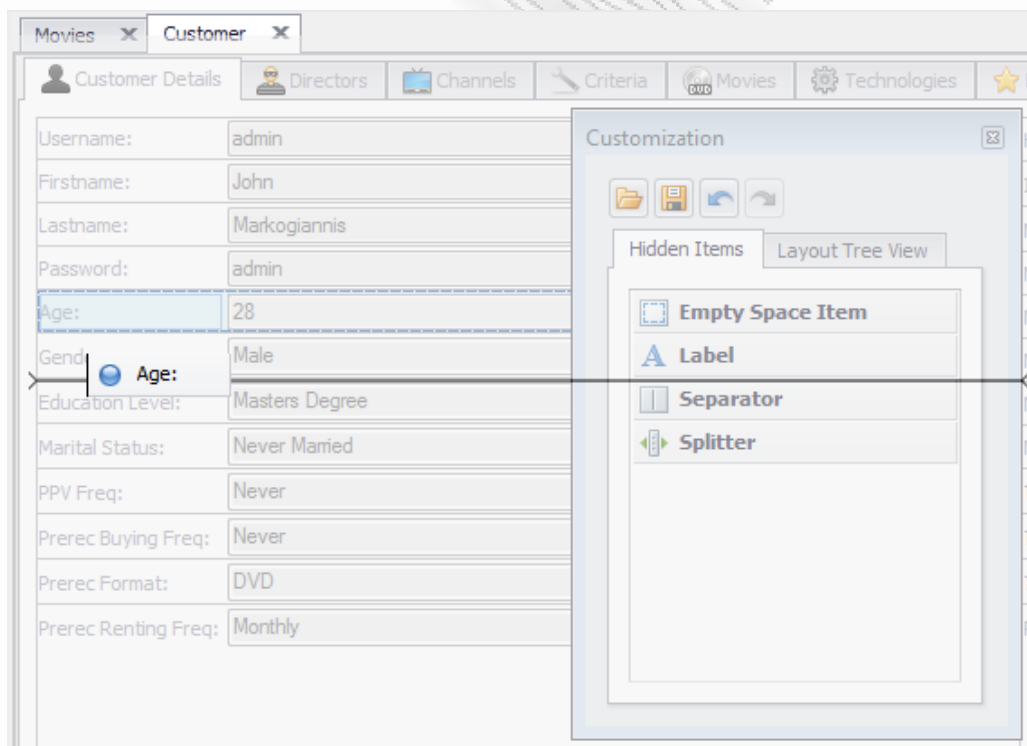


Εικόνα 55 Αλλαγή Theme DevExpress Style

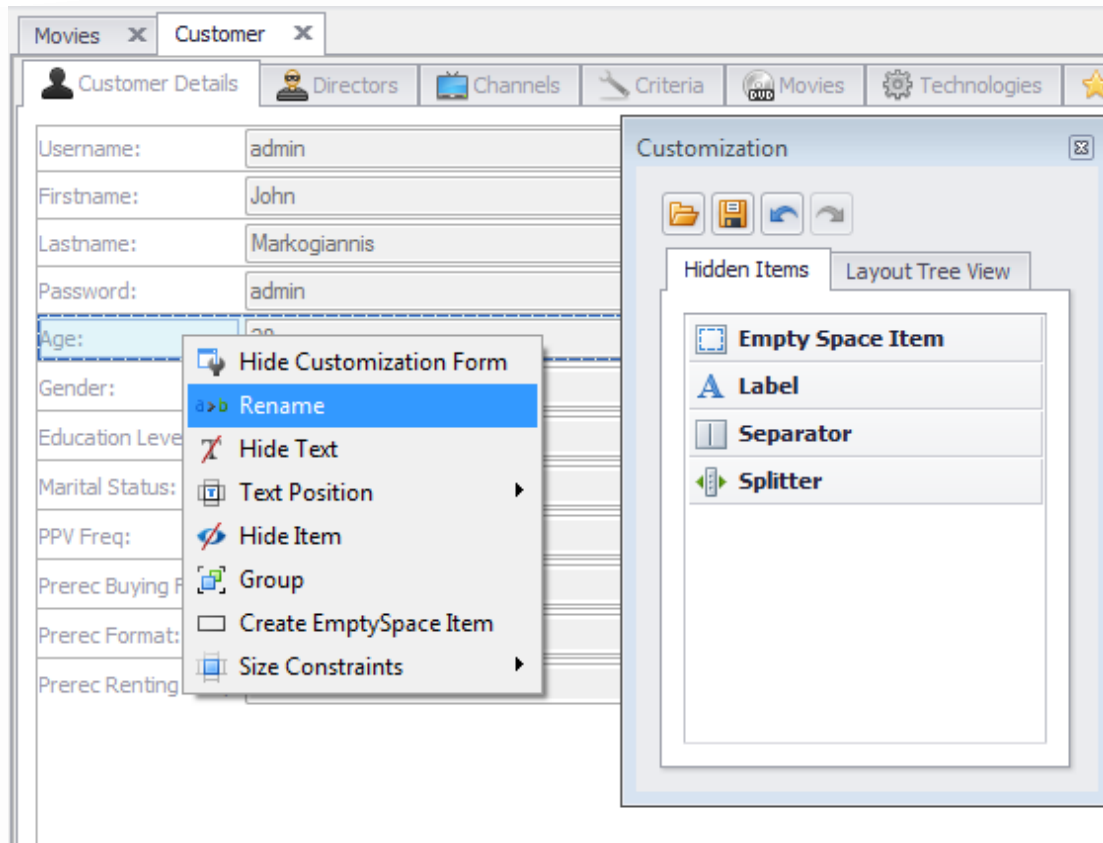
## 4.6.2 Προσαρμογή Οθόνης



Εικόνα 56 Customize Layout



Εικόνα 57 Μετακίνηση Πεδίων



Εικόνα 58 Μετονομασία Πεδίων

## 4.7 Ευφές Σύστημα Βοήθειας

Η λειτουργία του συστήματος παροχής βοήθειας βασίζεται σε έναν agent. Για την υλοποίηση του Agent έχουμε χρησιμοποιήσει τα εργαλεία Microsoft Agent Core Components, Text-to-speech engine (TTS) και Speech recognition engine. Παρακάτω παρουσιάζεται κάθε ένα από τα παραπάνω εργαλεία αναλυτικά καθώς και η διαδικασία υλοποίησης του agent.

### 4.7.1 Microsoft Agent Core Components

Οι Microsoft Agents είναι τρισδιάστατοι χαρακτήρες οι οποίοι περπατούν-πετούν στην οθόνη και έχουν αντίληψη της θέσης των διαφόρων παραθύρων και ενσωματώνουν τεχνολογίες αιχμής όπως η αναγνώριση φωνητικών εντολών και η σύνθεση φωνής, με έμφαση μάλιστα στο συγχρονισμό της κίνησης των χειλιών του χαρακτήρα και των ήχων που ακούγονται να βγαίνουν από το στόμα τους (lip sync). [1]

Κατευθύνονται από κάποιο αυτόματο γνωσιολογικό σύστημα, το οποίο παρακολουθεί και μπορεί να συσχετίσει την τρέχουσα κατάσταση του προγράμματος και την χρήση που κάνουμε σε αυτό με την πληροφορία που έχει σε κάποια βάση γνώσης, ώστε να κρίνει και αποφασίσει να μας προσφέρει τις κατάλληλες συμβουλές την στιγμή που κατά πάσα πιθανότητα τις χρειαζόμαστε, προτού καν το ρωτήσουμε.

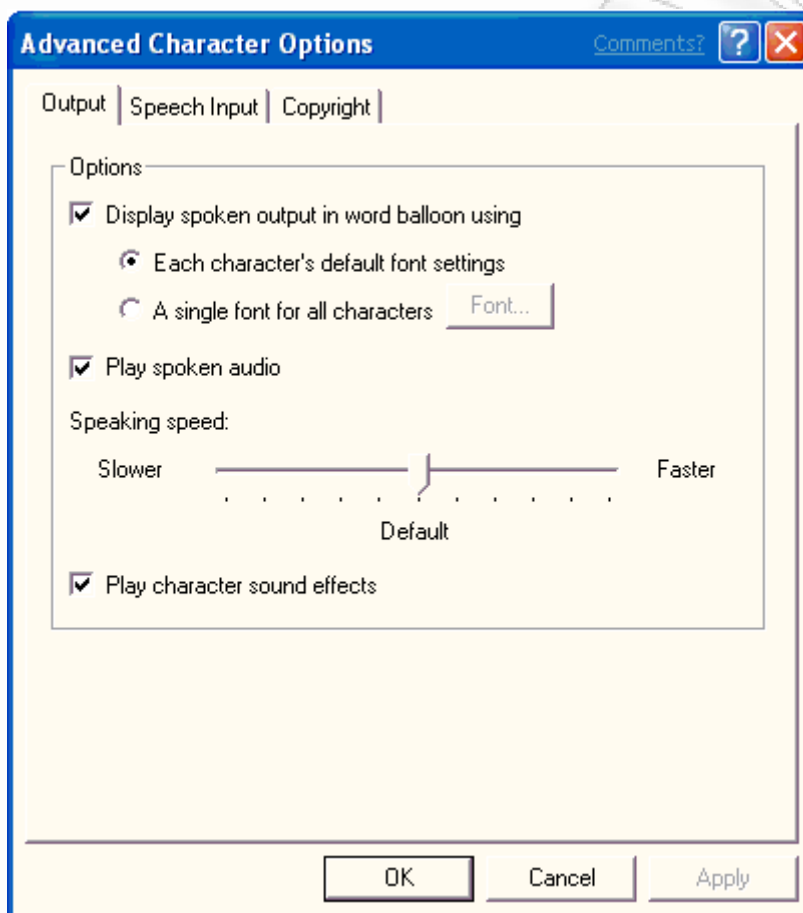
Η διεπαφή προγραμματισμού των Microsoft Agents είναι βασισμένη στο COM(Component Object Model). Με τη χρησιμοποίηση του COM, ο πράκτορας είναι σε θέση να παρέχει μια διεπαφή που επιτρέπει στις πολλαπλές εφαρμογές, ή πελάτες(clients), να χρησιμοποιήσουν τις animation υπηρεσίες ταυτόχρονα. Αυτές οι υπηρεσίες περιλαμβάνουν τη δυνατότητα φόρτωσης ενός χαρακτήρα, παραγωγής ενός συγκεκριμένου animation, σύνθεσης φωνής, και απόκρισης στις εντολές του χρήστη. Η αυτόματη lip-to-sync υποστήριξη παρέχεται για τη παραγωγή φωνής.





Για να χρησιμοποιήσει τις υπηρεσίες πρακτόρων, μια εφαρμογή πελατών πρέπει πρώτα να εγκαταστήσει μια σύνδεση με τον κεντρικό υπολογιστή πρακτόρων. Υπάρχουν τυποποιημένες διεπαφές COM που καθορίζονται για αυτόν το λόγο, αλλά ο πράκτορας περιλαμβάνει επίσης έναν ActiveX control που απλοποιεί τη διαδικασία.

Ο Microsoft Agent διατηρεί ορισμένες γενικές ρυθμίσεις που επιτρέπουν σε έναν χρήστη να ελέγξει την αλληλεπίδραση με όλους τους χαρακτήρες. Το προηγμένο παράθυρο επιλογών χαρακτήρα επιδεικνύει αυτές τις επιλογές και τις τρέχουσες ρυθμίσεις, και μπορεί να παρουσιαστεί από οποιαδήποτε εφαρμογή φιλοξενίας χρησιμοποιώντας τη διεπαφή προγραμματισμού πρακτόρων της Microsoft.[1]



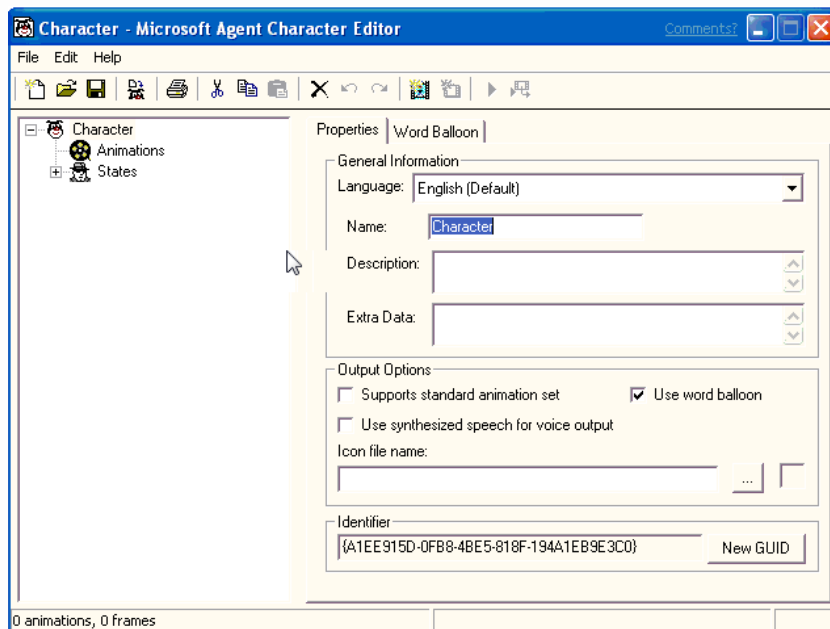
**Εικόνα 59 Agent Character Options**

Αυτή η σελίδα περιλαμβάνει τις γενικές ιδιότητες ελεγχου του χαρακτήρα. Παραδείγματος χάριν, ο χρήστης μπορεί να καθορίσει τη γραμματοσειρά εμφάνισης της προφορικής βοήθειας, τη ταχύτητα ομιλίας του agent, καθώς και τα ειδικά ηχητικά εφέ του κάθε agent.

Ένας χρήστης μπορεί να ρυθμίσει τις επιλογές φωνητικών εντολών σε αυτήν την σελίδα. Ο χρήστης μπορεί να θέσει εκτός λειτουργίας τις φωνητικές εντολές, να θέσει το πλήκτρο εισαγωγής ακούσματος, να επιλέξει την εμφάνιση του παραθύρου των φωνητικών

εντολών, και να επιλέξει να παίξει έναν τόνο midi για να δείξει πότε οι φωνητικές εντολές είναι διαθέσιμες

Επίσης με τη βοήθεια του Agent Character Editor έχουμε τη δυνατότητα δημιουργίας των δικών μας χαρακτήρων agent με τις εντολές και τα animation της επιλογής μας.[1]



Εικόνα 60 Agent Character Editor

#### 4.7.2 Υλοποίηση Ευφυούς Συστήματος Βοήθειας

Η παραπάνω υλοποίηση του agent είναι ανεξάρτητη εφαρμογής οπότε έχει τη δυνατότητα να προσαρτηθεί σε οποιοδήποτε ευφες σύστημα βοήθειας επιθυμούμε είτε αυτό αφορά μια online είτε μια desktop εφαρμογή.

Παρακάτω θα παρουσιάσουμε την υλοποίηση ενός συστήματος βοήθειας το οποίο αφορά το πληροφοριακό σύστημα Smart Media που παρουσιάσαμε παραπάνω.

Βασικό χαρακτηριστικό του συστήματος πρέπει να είναι η προσαρμοστικότητα. Ένα προσαρμοστικό σύστημα, μπορεί να εκπαιδευτεί για να αναγνωρίζει τη συμπεριφορά ενός έμπειρου ή αρχάριου και ακολούθως να ρυθμίζει τον έλεγχο του διάλογου του ή το σύστημα βοήθειας του αυτόματα, για να αντιμετωπίσει τις ανάγκες του παρόντα χρήστη. Έτσι οι δυο κύριοι στόχοι του συστήματος είναι:

1. Αναγνώριση Προτιμήσεων και Συνηθειών του χρήστη.
2. Αναγνώριση Σχεδίων και Στόχων του Χρήστη.

#### Κατηγοριοποίηση με βάση το ρόλο

Η κατηγοριοποίηση με βάση το ρόλο του χρήστη επιτυγχάνεται με τη παρακολούθηση των ενεργειών του χρήστη στο περιβάλλον της εφαρμογής αλλά και με τη καταγραφή των ενεργειών του χρήστη στη βάση δεδομένων και τον έλεγχο της βάσης δεδομένων. Για να επιδεχθούν οι παραπάνω λειτουργίες έχουμε δημιουργήσει μια κλάση User Profile για να μπορούμε να εξαγάμε και να αποθηκεύουμε το προφίλ του κάθε χρήστη όσο αφορά το ρόλο του στο σύστημα.



### Κατηγοριοποίηση με βάση την εμπειρία

Για την κατηγοριοποίηση των χρηστών με βάση την εμπειρία τους και την ικανότητα χειρισμού του συστήματος χρησιμοποιούμε δυο μεθόδους. Η πρώτη είναι να παρακολουθούμε τις ενέργειες του χρήστη (επιλογή ταινίας, αλλαγή προφίλ κ.α.) και η δεύτερη να παρακολουθούμε τις κινήσεις του χρήστη στο περιβάλλον της εφαρμογής (κίνηση ποντικιού).

Στη πρώτη μέθοδο ελέγχουμε τις επιλογές του χρήστη δηλαδή για παράδειγμα εάν έχει επιλέξει από το menu της εφαρμογής κάποια συγκεκριμένη οθόνη και στη δεύτερη μέθοδο ελέγχουμε τη κίνηση του χρήστη πάνω σε μία συγκεκριμένη οθόνη ώστε να προτείνουμε συγκεκριμένη βοήθεια για τη επιλεγμένη επιλογή. Έτσι κάθε χρήστης αποκτά ένα βαθμό εμπειρίας χρήσης του συστήματος ο οποίος προσαρμόζεται κάθε φορά με την αντίστοιχη χρήση του με σκοπό τη τελική προσαρμογή των λεπτομερειών βοήθειας.

Οι παραπάνω μέθοδοι συνδυάζονται μεταξύ τους μέσω της παρακάτω διεπαφής για την επίτευξη της προσαρμοστικής βοήθειας.

```
namespace SmartMedia.BI
{
    [Serializable]
    public class HelpFactor
    {
        public int CustomerID { get; set; }
        public int Value { get; set; }
        public string Action { get; set; }
        public DateTime Date { get; set; }
    }

    public static class IntelligentHelpEngine
    {
        public static List<HelpFactor> HelpFactors { get; set; }
        public static ExperienceEnum Experience { get; set; }
        public static AgentObjects.IAgentCtlCharacter HelpAgent {
get; set; }

        //MoveTo( int x , int y , objectspeed speed);
        //Play(string animation);
        // this function make the character to play some animation
        //The possible animations that the character can play it are

        /*****
        RestPose, Wave, DontRecognize, Uncertain, Decline, Sad,
        StopListening, GetAttention, GetAttentionReturn, Blink,
Idle3_2,
        Surprised, Congratulate_2, Reading, Announce, Read
, ReadReturn, Idle2_2 ,
        Writing , Write , WriteReturn , Congratulate , Confused , Suggest
, MoveRight,
        MoveLeft, Idle2_1, MoveUp, MoveDown,
        StartListening, WriteContinued, DoMagic1,
```

```

    DoMagic2, Idle1_1, LookDown, LookDownBlink, LookDownReturn,
    LookLeft,
    LookLeftBlink, LookLeftReturn, Idle1_3, LookRight,
    LookRightBlink,
    LookRightReturn, LookUp, LookUpBlink, LookUpReturn, Idle1_2,
    ReadContinued, Pleased, GetAttentionContinued, Process,
    Search,
    Think, Idle1_4, Greet, Idle3_1, GestureUp, GestureDown, GestureLeft,
    GestureRight, Show, Hide, Hearing_4, Hearing_1, Hearing_2, Hearin,
    Alert, Explain, Processing, Thinking, Searching, Acknowledge
    *****/
public static void ShowHelp(string formName)
{
    if (formName == "Login")
    {
        switch (Experience)
        {
            case ExperienceEnum.Beginner:
                HelpAgent.Speak("If you are new User please
Register first else Login with your account");
                break;
            case ExperienceEnum.Advanced:
                break;
            case ExperienceEnum.Expert:
                break;
            default:
                break;
        }
    }
    if (formName == "Main")
    {
        switch (Experience)
        {
            case ExperienceEnum.Beginner:
                HelpAgent.Speak("Hello New User " +
BLL.Session.customer.Firstname);
                //HelpAgent.Play("Suggest");
                HelpAgent.Speak("You can view or edit your
profile in Profile Information");
                //HelpAgent.Play("Suggest");
                HelpAgent.Speak("If you want to buy a new
Movie just click the Movies Shop option");
                //HelpAgent.Play("Thinking");
                break;
            case ExperienceEnum.Advanced:
                HelpAgent.Speak("Hello Advanced User " +
BLL.Session.customer.Firstname);

```

```
        HelpAgent.Speak("Your Transaction History is
under the Catalog Menu");
        break;
        case ExperienceEnum.Expert:
            HelpAgent.Speak("Hello Expert User " +
BLL.Session.customer.Firstname);
            HelpAgent.Speak("There are some new
recommendations for you in Movies Shop");
            break;
        default:
            break;
    }
}
if (formName == "Customer")
{
    switch (Experience)
    {
        case ExperienceEnum.Beginner:
            HelpAgent.Speak("Here you can change your
profile information");
            break;
        case ExperienceEnum.Advanced:
            HelpAgent.Speak("Edit your favorite movies,
actors and others");
            break;
        case ExperienceEnum.Expert:
            HelpAgent.Speak("If you change your profile,
you affect your recommendations in movie shop");
            break;
        default:
            break;
    }
}
if (formName == "Customers")
{
    switch (Experience)
    {
        case ExperienceEnum.Beginner:
            HelpAgent.Speak("See all system customers and
their profiles");
            break;
        case ExperienceEnum.Advanced:
            HelpAgent.Speak("Notice recommendations for
each selected user");

            break;
        case ExperienceEnum.Expert:
```

```
HelpAgent.Speak("You can check recommendation
system for each user");
        break;
    default:
        break;
    }
}
if (formName == "CustomersKPI")
{
    switch (Experience)
    {
        case ExperienceEnum.Beginner:
            HelpAgent.Speak("See Customers Total
Payments");
            break;
        case ExperienceEnum.Advanced:
            HelpAgent.Speak("Find your best customers per
time");
            break;
        case ExperienceEnum.Expert:
            HelpAgent.Speak("Compare customers");
            break;
        default:
            break;
    }
}
if (formName == "TopCustomers")
{
    switch (Experience)
    {
        case ExperienceEnum.Beginner:
            HelpAgent.Speak("Best Customers per
payments");
            break;
        case ExperienceEnum.Advanced:
            HelpAgent.Speak("Find the top customers");
            break;
        case ExperienceEnum.Expert:
            HelpAgent.Speak("Compare customers");
            break;
        default:
            break;
    }
}
if (formName == "Movies")
```

```
{
    switch (Experience)
    {
        case ExperienceEnum.Beginner:
            HelpAgent.Speak("Search for a movies and see
its details");
            break;
        case ExperienceEnum.Advanced:
            HelpAgent.Speak("Change movie genres");
            break;
        case ExperienceEnum.Expert:
            HelpAgent.Speak("Here you can add,edit or
delete a movie");
            break;
        default:
            break;
    }
}
if (formName == "MoviesShop")
{
    switch (Experience)
    {
        case ExperienceEnum.Beginner:
            HelpAgent.Speak("Find your favorite movie and
buy it. Notice the recommendations for you and related movies.");
            break;
        case ExperienceEnum.Advanced:
            HelpAgent.Speak("User recommendations for the
best movie for you");
            break;
        case ExperienceEnum.Expert:
            HelpAgent.Speak("Check recommendation
algorithms for movies");
            break;
        default:
            break;
    }
}
if (formName == "MoviesKPI")
{
    switch (Experience)
    {
        case ExperienceEnum.Beginner:
            HelpAgent.Speak("Best Movies per time");
            break;
        case ExperienceEnum.Advanced:
```

```
HelpAgent.Speak("Movies payments per time");

        break;
    case ExperienceEnum.Expert:
        HelpAgent.Speak("Compare movies");
        break;
    default:
        break;
    }
}
if (formName == "MoviesRental")
{
    switch (Experience)
    {
        case ExperienceEnum.Beginner:
            HelpAgent.Speak("See all movies
transactions");

            break;
        case ExperienceEnum.Advanced:
            HelpAgent.Speak("Check each user
transactions");

            break;
        case ExperienceEnum.Expert:
            HelpAgent.Speak("Find a transaction");
            break;
        default:
            break;
    }
}
if (formName == "TopMovies")
{
    switch (Experience)
    {
        case ExperienceEnum.Beginner:
            HelpAgent.Speak("Top Movies per payments");
            break;
        case ExperienceEnum.Advanced:
            HelpAgent.Speak("Find top seller Movie");

            break;
        case ExperienceEnum.Expert:
            HelpAgent.Speak("Compare movies per
payments");

            break;
        default:
            break;
    }
}
```

```
    }
    if (formName == "MiningStructure")
    {
        switch (Experience)
        {
            case ExperienceEnum.Beginner:
                HelpAgent.Speak("Create your own mining model
for recommendations");
                break;
            case ExperienceEnum.Advanced:
                HelpAgent.Speak("Change parameters for
recommendations mining models");

                break;
            case ExperienceEnum.Expert:
                HelpAgent.Speak("Create a new model and check
it live for recommendations");
                break;
            default:
                break;
        }
    }
}

public static void ShowMessage(string message)
{
    HelpAgent.Speak(message);
    HelpAgent.Play("Suggest");
}

public static List<HelpFactor> GetHelpFactors()
{
    List<HelpFactor> helpFactors = new List<HelpFactor>();

    try
    {
        if (BLL.Session.customer != null)
        {
            XmlSerializer s = new
XmlSerializer(typeof(List<HelpFactor>));
            XmlReader reader =
XmlReader.Create(BLL.Session.customer.CustomerID +
"_HelpFactors.xml");
            helpFactors =
(List<HelpFactor>)s.Deserialize(reader);
        }

        HelpFactors = helpFactors;
    }
}
```

```

        int expValue = helpFactors.Select(record =>
record.Value).Sum();

        SetExperienceByExpValue(expValue);

        return HelpFactors;
    }
    catch (Exception)
    {
        HelpFactors = helpFactors;
        SetExperienceByExpValue(0);

        return helpFactors;
    }
}

public static void SetExperienceByExpValue(int expValue)
{
    if (expValue <= 50)
    {
        Experience = ExperienceEnum.Beginner;
    }
    if (expValue > 50 && expValue<=100)
    {
        Experience = ExperienceEnum.Advanced;
    }
    if (expValue > 100)
    {
        Experience = ExperienceEnum.Expert;
    }
}

public static void SetHelpFactors()
{
    if (BLL.Session.customer != null)
    {
        if (File.Exists(BLL.Session.customer.CustomerID +
        "_HelpFactors.xml"))
        {
            File.Delete(BLL.Session.customer.CustomerID +
            "_HelpFactors.xml");
        }

        XmlSerializer s = new
        XmlSerializer(typeof(List<HelpFactor>));
        XmlWriter writer =
        XmlWriter.Create(BLL.Session.customer.CustomerID +
        "_HelpFactors.xml");
    }
}

```



```

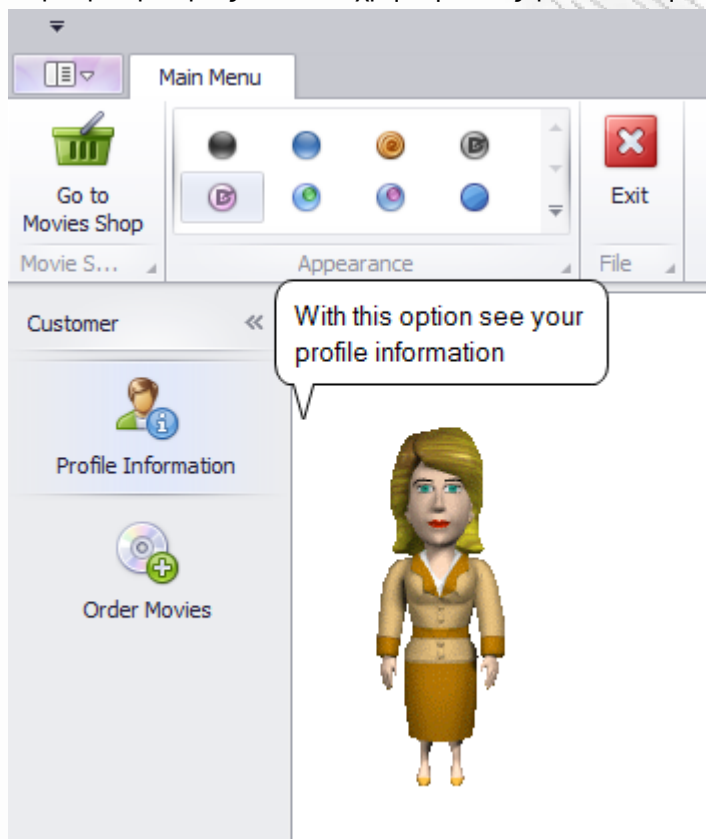
        s.Serialize(writer, HelpFactors);
    }
}

public enum ExperienceEnum
{
    Beginner,
    Advanced,
    Expert
}
}

```

### Επίδειξη Συστήματος Ευφούς Βοήθειας

Αρχικά με κάθε σύνδεση στο σύστημα φορτώνεται το ιστορικό των ενεργειών του χρήστη και προσδιορίζεται ο βαθμός εμπειρίας του. Έπειτα το σύστημα παρακολουθεί τις ενέργειες του χρήστη στο κεντρικό μενού επιλογών και εμφανίζει τη κατάλληλη βοήθεια που αντιστοιχεί στο επίπεδό του. Έτσι για την ίδια επιλογή διαφοροποιείται η πληροφορία της βοήθειας ανάλογα με το βαθμό εμπειρίας του κάθε χρήστη, όπως φαίνεται στη παρακάτω οθόνη.

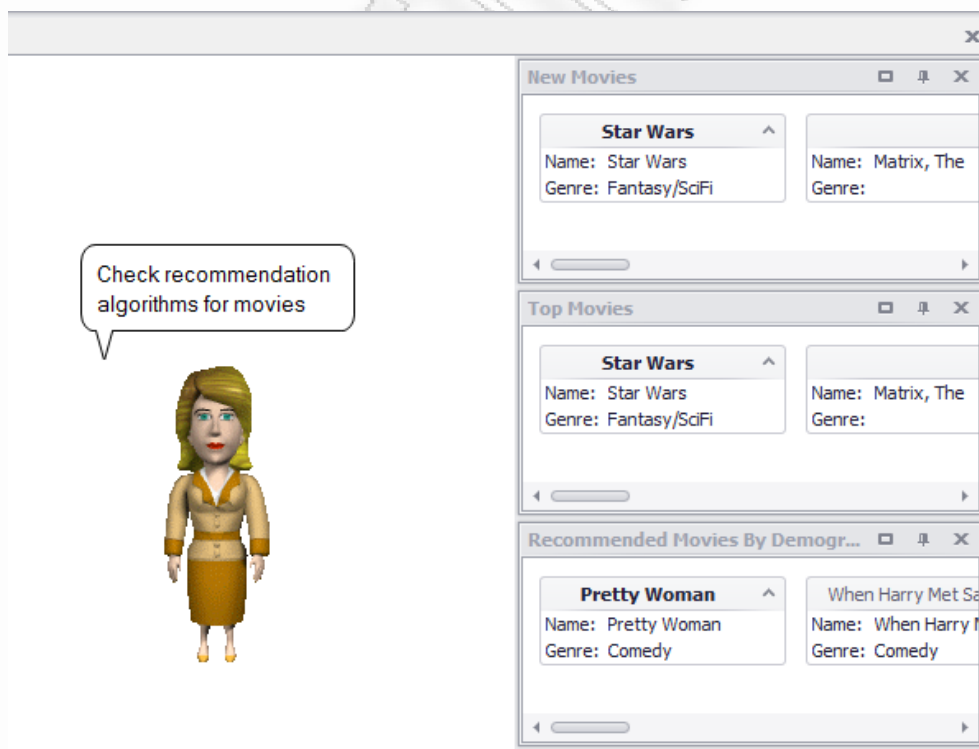


Εικόνα 61 Profile Information Help for New User



**Εικόνα 62 Profile Information Help for Advanced User**

Επίσης με την εισαγωγή του χρήστη στην κάθε οθόνη, υπολογίζεται από τη καταγραφή του ιστορικού των ενεργειών του, ο βαθμός εμπειρίας του στη χρήση της συγκεκριμένης οθόνης και εμφανίζεται σχετική σε αυτόν βοήθεια.



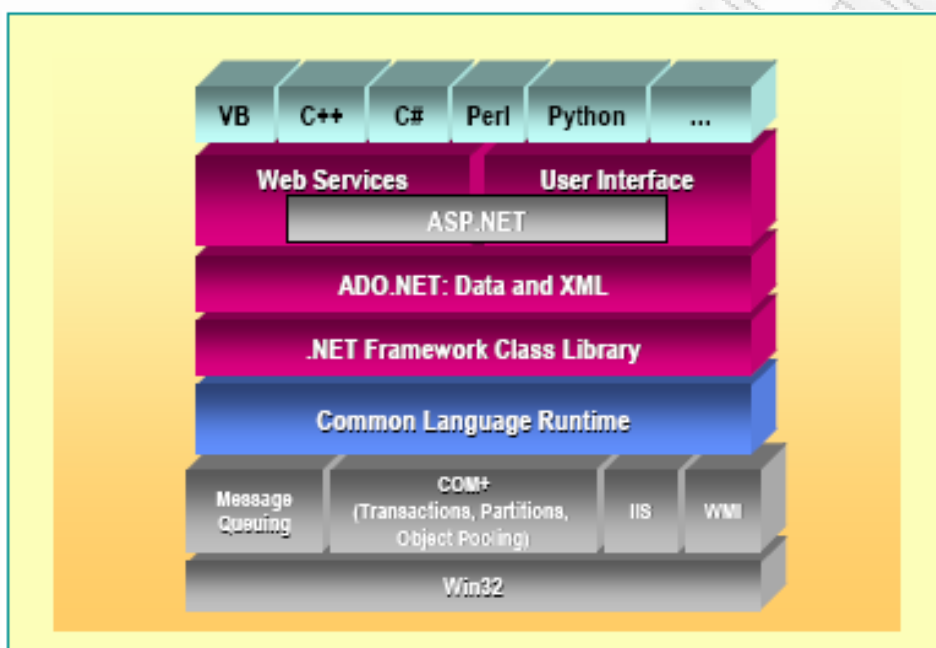
**Εικόνα 63 Movies Shop Screen Help**

## 5 ΥΛΟΠΟΙΗΣΗ

Στο κεφάλαιο αυτό παρουσιάζουμε τον τρόπο με τον οποίο υλοποιήσαμε το λογισμικό σύστημα που αναλύσαμε παραπάνω. Επιπλέον, παρουσιάζουμε τις πλατφόρμες, τα εργαλεία, τις τεχνολογίες και τα πρωτόκολλα τα οποία χρησιμοποιήσαμε κατά την υλοποίηση του συστήματος.

### 5.1 Πλατφόρμες και προγραμματιστικά εργαλεία

#### 5.1.1 Το .Net Framework

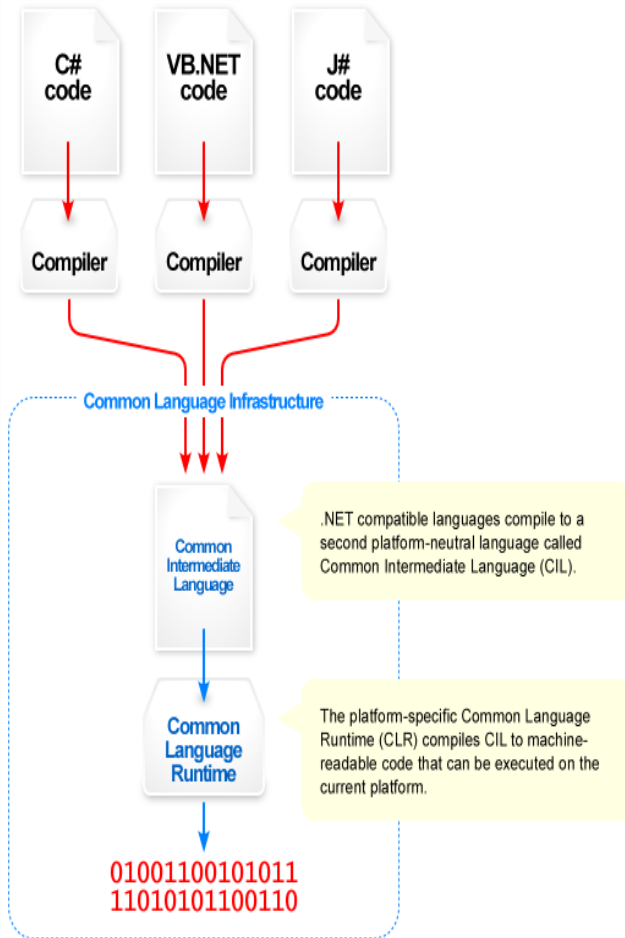


Εικόνα 64 .NET Framework

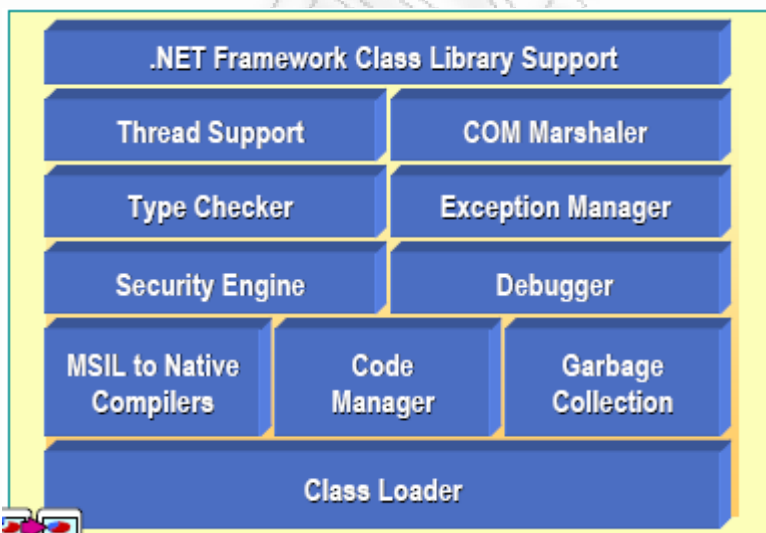
Το .Net Framework παρέχει το κατάλληλο μεταγλωττιστή για τη δημιουργία και την εκτέλεση .Net Εφαρμογών. Το .Net Framework πρέπει να τρέχει σε ένα λειτουργικό σύστημα. Συγκεκριμένα έχει δημιουργηθεί για να τρέχει σε Microsoft Win32 λειτουργικά συστήματα.

#### 5.1.2 Common Language Runtime

Η πλατφόρμα common language runtime απλοποιεί την ανάπτυξη εφαρμογών, παρέχοντας ένα ασφαλές περιβάλλον εκτέλεσης, υποστηρίζοντας πολλαπλές γλώσσες, και δυνατότητα διαχείρισης εφαρμογών. Επίσης προσφέρει ένα έτοιμο περιβάλλον διαχείρισης των εφαρμογών, στο οποίο κοινές υπηρεσίες, όπως η garbage collection κ.α, παρέχονται αυτόματα.



Τα χαρακτηριστικά της πλατφόρμας common language runtime παρουσιάζονται στη παρακάτω εικόνα.

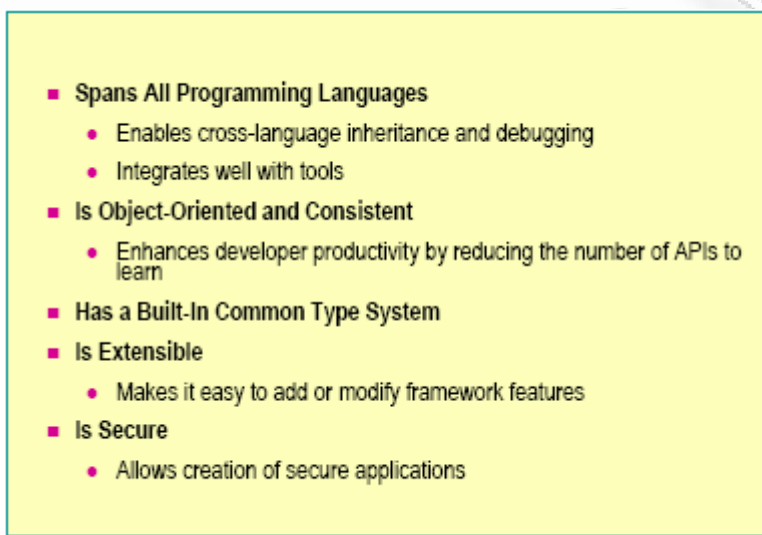


Εικόνα 65 Common language runtime

### 5.1.3 Η Βιβλιοθήκη κλάσεων του .NET Framework

Η βιβλιοθήκη κλάσεων του .NET Framework παρουσιάζει στο χρήστη χαρακτηριστικά του runtime και παρέχει άλλες υψηλού επιπέδου υπηρεσίες τις οποίες ο κάθε προγραμματιστής χρειάζεται. Οι κλάσεις απλοποιούν την ανάπτυξη .NET εφαρμογών. Οι προγραμματιστές έχουν τη δυνατότητα να χρησιμοποιήσουν τις ήδη υπάρχοντες κλάσεις να τις αναπτύξουν και να δημιουργήσουν τις δικές τους βιβλιοθήκες κλάσεων. Επειδή υπάρχουν εκατοντάδες κλάσεις στη βιβλιοθήκη κλάσεων του .NET Framework έχουν ομαδοποιηθεί σε χώρους ονομάτων. Το πρώτο μέρος του ονόματος το οποίο τοποθετείται πριν τη πιο δεξιά τέλεια είναι το όνομα του χώρου ονόματος. Το τελευταίο τμήμα του ονόματος το οποίο τοποθετείται μετά την τέλεια, είναι το όνομα της κλάσης. Για παράδειγμα, το System.Collections.ArrayList αναπαριστά την ArrayList κλάση, η οποία ανήκει στο χώρο ονομάτων System.Collections. Οι τύποι στο χώρο ονομάτων System.Collections μπορούν να χρησιμοποιηθούν για τη δημιουργία collections αντικειμένων.

Τα χαρακτηριστικά της βιβλιοθήκης του .Net Framework παρουσιάζονται στη παρακάτω εικόνα:



Εικόνα 66 .Net Framework Class Library

Η βιβλιοθήκη κλάσεων του .NET Framework είναι ανεξάρτητη των γλωσσών επιτρέποντας έτσι τη cross language ανάπτυξη και αποσφαλμάτωση.

- **Is Object-Oriented and Consistent**

Σε αντίθεση με τα APIs τα οποία είναι αρκετά και μη οργανωμένα, η βιβλιοθήκη κλάσεων του .NET Framework είναι οργανωμένη σε χώρους ονομάτων και κλάσεις. Αυτή η αντικειμενοστραφής προσέγγιση ομαδοποιεί τη σχετική λειτουργικότητα και δεδομένα μαζί και επιτρέπει στο προγραμματιστή να εργαστεί με τη βιβλιοθήκη με έναν πιο φυσικό τρόπο.
- **Has a Built-In Common Type System**

Η βιβλιοθήκη κλάσεων του .NET Framework παρέχει ασφάλεια τύπων μέσω του common type system, το οποίο είναι μέρος της πλατφόρμας common language runtime.
- **Is Extensible**

Προσφέρει τη δυνατότητα να δημιουργήσεις τις δικές σου κλάσεις και να τις μεταγλωττίσεις μέσα σε βιβλιοθήκες. Εάν η βιβλιοθήκη οργανωθεί σωστά θα είναι επίσης αντικειμενοστραφής και ανεξάρτητη γλωσσών.

➤ Is Secure

Η βιβλιοθήκη κλάσεων του .NET Framework παρέχει σημαντική ασφάλεια για τις εφαρμογές. Μπορείς να χρησιμοποιήσεις την ασφάλεια code access και τη role-based για να ρυθμίσεις τις δικές σου πολιτικές ασφαλείας. Επιπλέον υπάρχουν αρκετά εργαλεία ασφαλείας το οποίο διευκολύνουν τη δημιουργία πιστοποιητικών, την παρουσίαση των αδειών χρήσης κ.α. [12]

## 5.2 Σύστημα Διαχείρισης Βάσεων Δεδομένων

Για το Σύστημα Διαχείρισης Βάσεων Δεδομένων στο οποίο θα αναπτυχθεί η βάση δεδομένων του συστήματος δεν υπάρχει κανένα περιορισμός. Η βάση δεδομένων θα μπορεί να βρίσκεται σε ένα οποιοδήποτε ΣΔΒΔ σε υπολογιστή με οποιοδήποτε λειτουργικό σύστημα. Για την ανάπτυξη και τον έλεγχο, όμως, του συστήματος, που γίνεται σε λειτουργικό σύστημα Windows, έχει επιλεγεί ως ΣΔΒΔ ο Microsoft SQL Server 2008 R2.

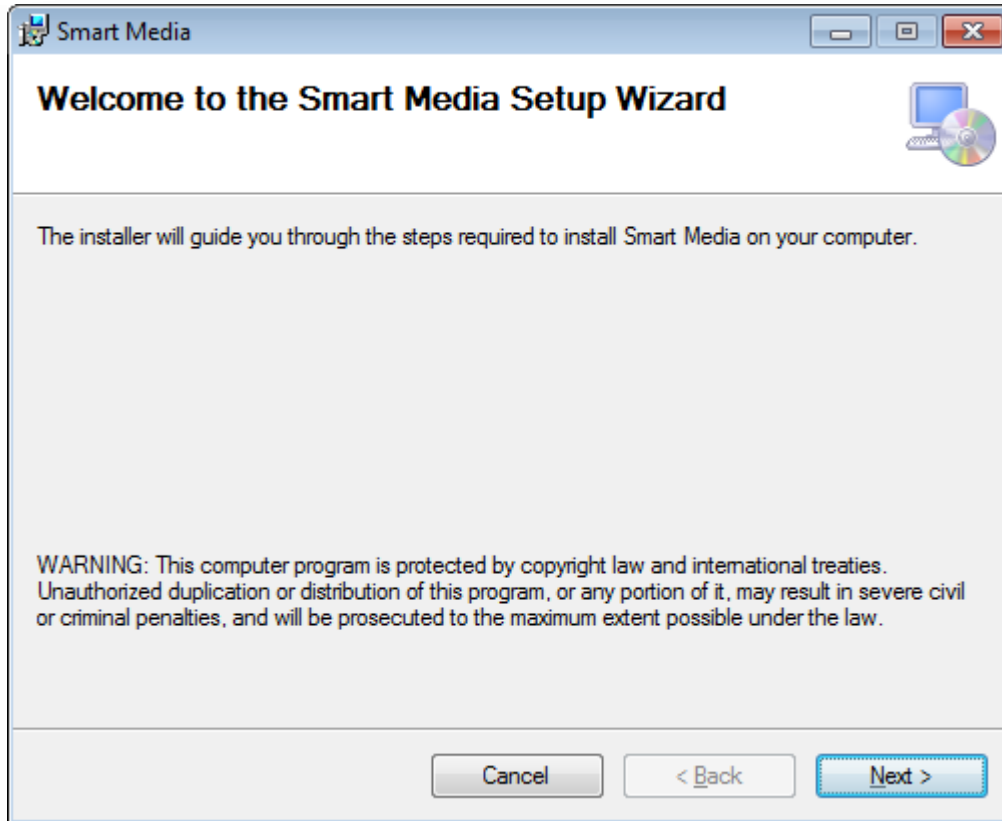
## 5.3 Οδηγός Εγκατάστασης

### 5.3.1 Απαιτήσεις Συστήματος

Για να εγκατασταθεί και να εκτελεστεί η εφαρμογή είναι απαραίτητο ο υπολογιστής να διαθέτει το .Net Framework 4 και για την εκτέλεση είναι απαραίτητη μια σύνδεση στο Internet ώστε η εφαρμογή να επικοινωνήσει με τα Web Services.

### 5.3.2 Εγκατάσταση

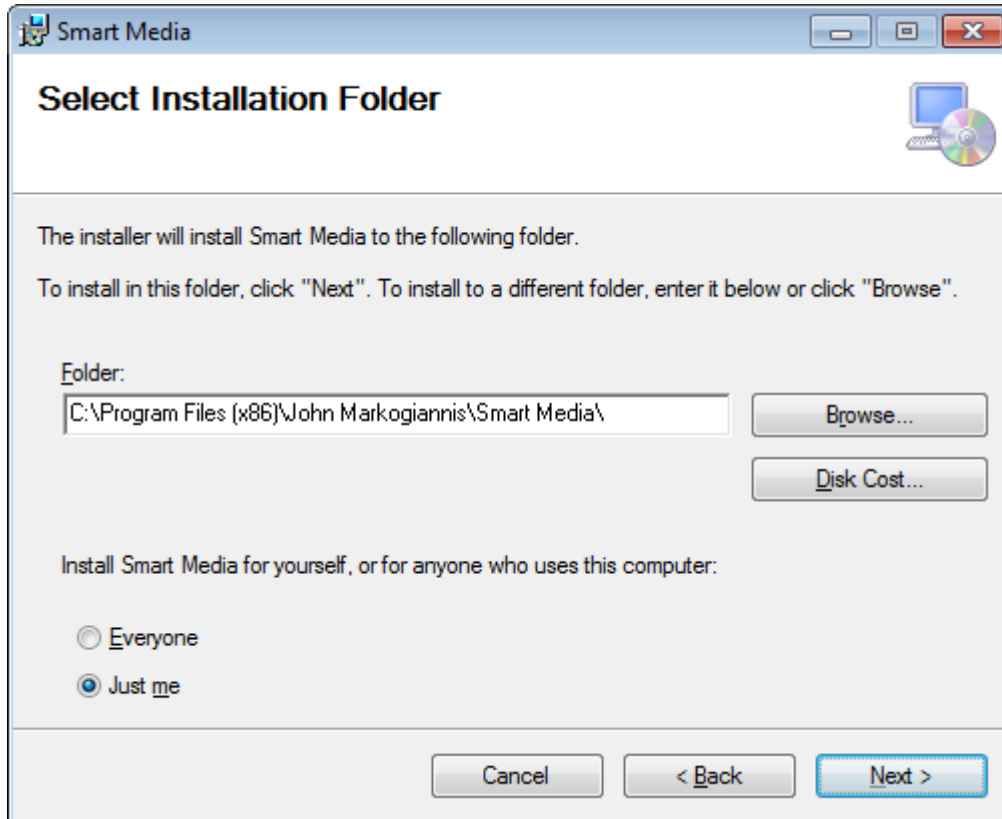
Η εγκατάσταση της εφαρμογής γίνεται με την εκτέλεση του αρχείου setup.exe. Παρακατω παραθέτω ενδεικτικές εικόνες από την διαδικασία της εγκατάστασης.



**Εικόνα 67 Εγκατάσταση βήμα 1**

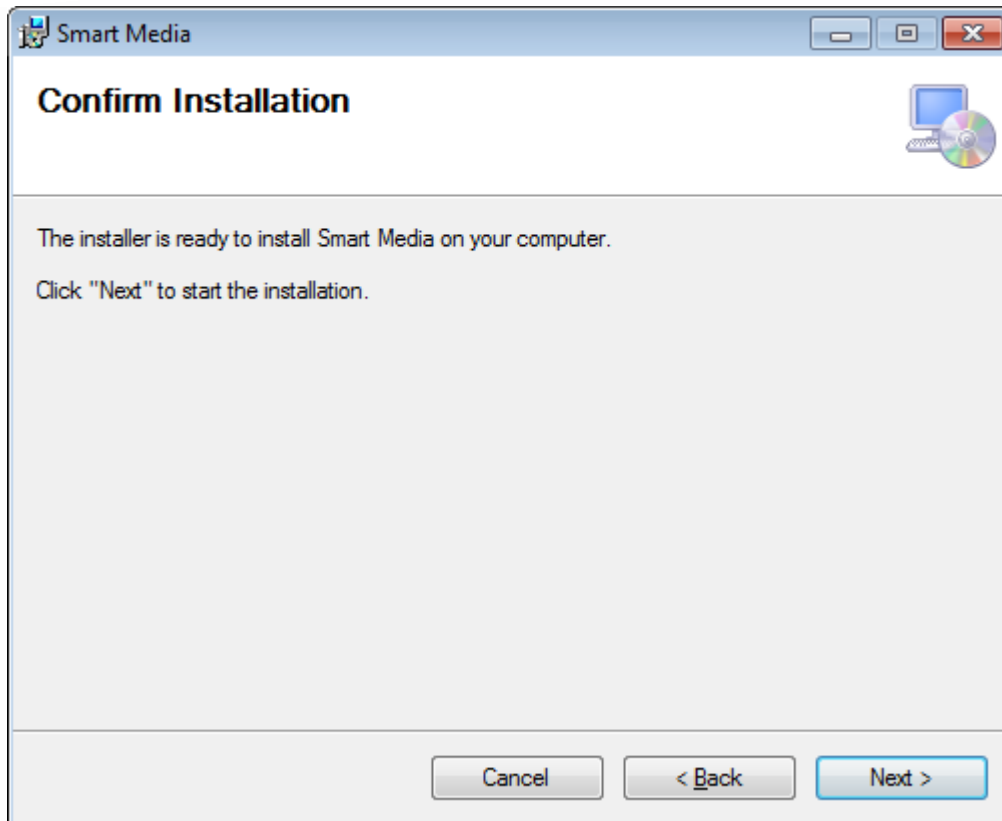
Οθόνη εισαγωγής. Επιλεγουμε Next για συνέχεια.





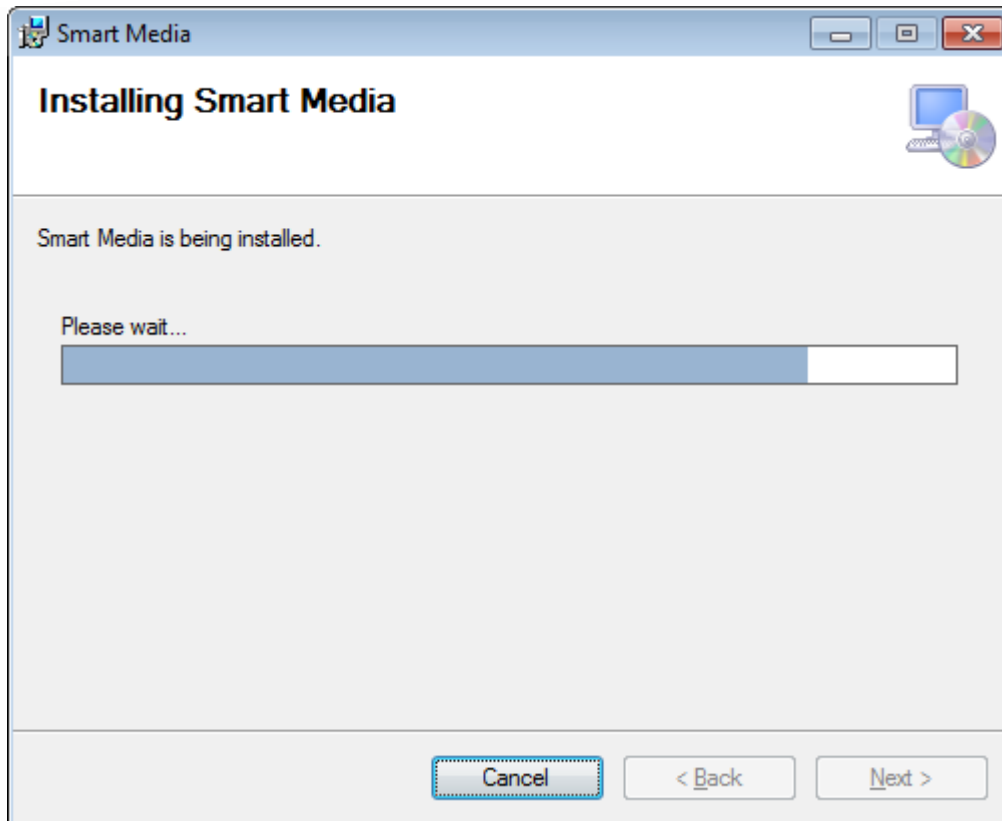
**Εικόνα 68 Εγκατάσταση βήμα 2**

Επιλεγούμε τη τοποθεσία εγκατάστασης της εφαρμογής στον υπολογιστή μας. Σας προεπιλογή υπάρχει η τοποθεσία: C:\Program Files\Chalvantzis Konstantinos\Smart Earth\ και στη συνέχεια ρυθμίζουμε τα δικαιώματα της εγκατάστασης. Έπειτα πατάμε Next.



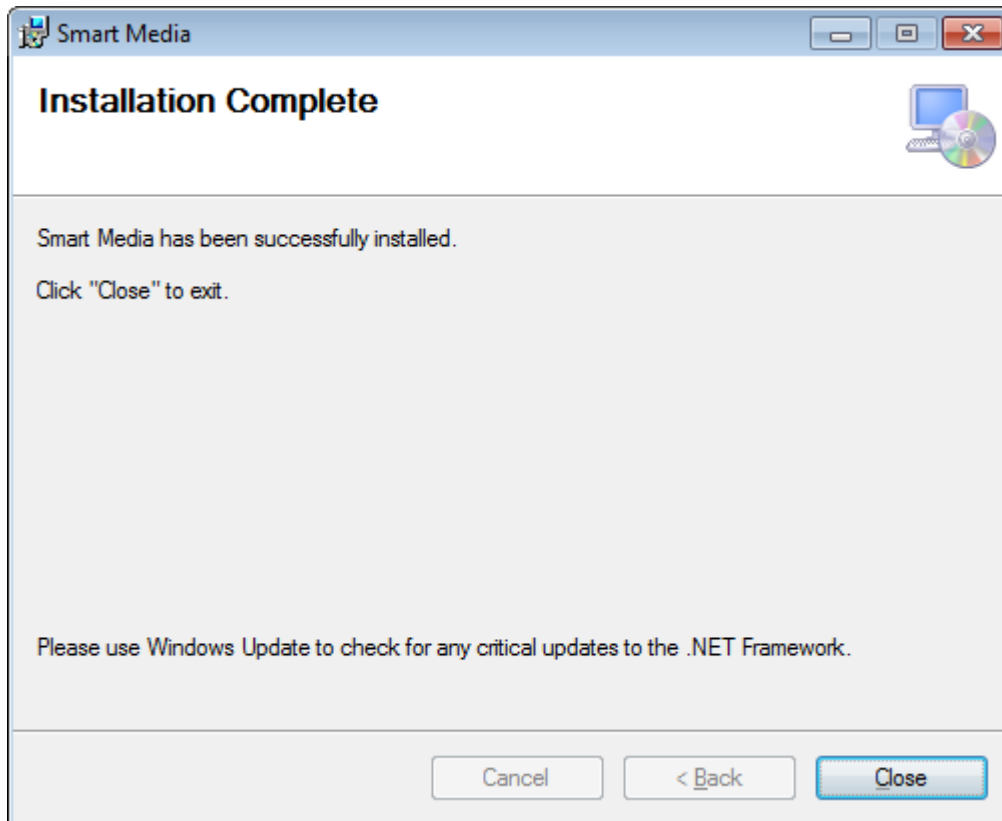
**Εικόνα 69 Εγκατάσταση βήμα 3**

Επιβεβαιώνουμε την εγκατάσταση και πατάμε Next.



Εικόνα 70 Εγκατάσταση βήμα 4

Περιμένουμε μέχρι να ολοκληρωθεί η εγκατάσταση και τέλος



**Εικόνα 71 Εγκατάσταση βήμα 5**

Πατάμε Close για έξοδο.

## 6 ΕΠΙΛΟΓΟΣ

### 6.1 Σύνοψη - Σύγκριση και Συμπεράσματα

Ανακεφαλαιώνοντας, μπορούμε να ισχυριστούμε ότι σε γενικές γραμμές ο αρχικός στόχος που είχαμε θέσει επετεύχθη. Υλοποιήθηκε γενικά το σύστημά το οποίο είχαμε σχεδιάσει. Από πλευράς υπηρεσιών υλοποιήθηκαν οι σημαντικότερες υπηρεσίες όπως είναι:

- ✓ Υλοποίηση υποσυστήματος προσαρμοστικών διεπαφών
- ✓ Υλοποίηση ευφυούς υποσυστήματος βοήθειας
- ✓ Υλοποίηση εξατομικευμένου υποσυστήματος συστάσεων
- ✓ Υλοποίηση συστήματος δυναμικής δημιουργίας μοντέλων εξατομίκευσης

Πιο συγκεκριμένα πρέπει να επισημάνουμε πως οι περισσότερες διαθέσιμες εφαρμογές που επιτυγχάνουν τις παραπάνω υπηρεσίες είτε εμπορικές είτε όχι είναι εφαρμογές συγκεκριμένου σκοπού και οι παραπάνω υπηρεσίες εφαρμόζονται στα συγκεκριμένα συστήματα αποκλείοντας την ενσωματωσή τους σε διαφορετικής φύσης λογισμικά. Στα πλαίσια της συγκεκριμένης εργασίας προσπαθήσαμε να υλοποιήσουμε όσο το δυνατόν ήταν εφικτό ένα ανεξάρτητο φυσικού προβλήματος σύστημα framework, επεκτάσιμο και δυναμικό ώστε να μπορεί να ενσωματωθεί σε οποιοδήποτε πληροφοριακό σύστημα. Αυτό είναι και το κυριότερο πλεονέκτημα του συστήματος που υλοποιήσαμε. Έτσι δυναμικά ο διαχειριστής του κάθε συστήματος έχει τη δυνατότητα να δημιουργήσει, ελέγξει και εφαρμόσει δυναμικά μέσα από το γραφικό περιβάλλον του framework δικά του μοντέλα εξόρυξης γνώσης εφαρμόζοντας διαφορετικούς αλγορίθμους κάθε φορά με διαφορετικά κριτήρια και παραμέτρους δημιουργώντας έτσι τα δικά του εξατομικευμένα συστήματα συστάσεων που συμπεριφέρονται καλύτερα κάθε φορά στο σύστημα του. Επίσης έχει τη δυνατότητα να θέσει σε λειτουργία το υποσύστημα ευφυούς βοήθειας και προσαρμοστικών διεπαφών στο δικό του πληροφοριακό σύστημα κάτω από προϋποθέσεις βεβαία για τη σωστή λειτουργία τους.

### 6.2 Μελλοντικές επεκτάσεις

Το σύστημα το οποίο υλοποιήσαμε αποτελεί μια βιβλιοθήκη – framework που έχει τη δυνατότητα να ενσωματωθεί σε οποιοδήποτε πληροφοριακό σύστημα. Βέβαια, πρέπει να γίνουν αρκετές επεκτάσεις πάνω σε αυτή την αρχική έκδοση του συστήματος έτσι ώστε το σύστημα αυτό να μπορέσει να είναι μία πλήρης βιβλιοθήκη και να έχουν εκπληρωθεί όλοι οι στόχοι που είχαν αρχικά τεθεί.

- ❖ Αρχικά θα πρέπει να γίνει δοκιμή της υπάρχουσας έκδοσης του συστήματος σε ένα διαφορετικής φύσης πληροφοριακό σύστημα από αυτό που υλοποιήσαμε για την επίδειξη του.

Από πλευράς βελτιώσεων του υπάρχοντος κώδικα, μπορούν να γίνουν οι εξής επεκτάσεις:

- ❖ Υλοποίηση διαφορετικών αλγορίθμων εξόρυξης γνώσης. (neural networks, k-means clustering, fuzzy decisions)
- ❖ Υλοποίηση αλγορίθμου δυναμικών παραμέτρων για το υποσύστημα ευφυούς βοήθειας
- ❖ Επέκταση του υποσυστήματος ευφυούς βοήθειας με δυναμικό περιεχόμενο βοήθειας
- ❖ Επέκταση του υποσυστήματος προσαρμοστικών διεπαφών με περισσότερες δυνατότητες προσαρμογής.

Αυτές είναι ίσως οι κυριότερες επεκτάσεις που μπορούν να γίνουν στο μέλλον ώστε να έχουμε μία πλήρη βιβλιοθήκη εξατομίκευσης. Βέβαια πρέπει επίσης να γίνουν πειράματα με τους χρόνους απόκρισης των αλγορίθμων και των μοντέλων εξόρυξης γνώσης, ιδίως στην περίπτωση απότομης κλιμάκωσης του συστήματος.

## 7 ΒΙΒΛΙΟΓΡΑΦΙΑ

A/A	Τίτλος	Συγγραφέας	Ημερομηνία	Πηγή
[1]	<b>Microsoft Agent Software Development Kit</b>	Microsoft Press	1999	
[2]	<b>Data Mining Tutorial</b>	Seth Paul, Jamie, MacLennan, Zhaohui Tang, Scott Oveson	2005	
[3]	<b>Relational Classification for Personalized Tag Recommendation</b>	Leandro Balby Marinho, Christine Preisach, Lars Schmidt-Thieme	2009	ECML PKDD Discovery Challenge 2009 at the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD- DC 2009), Bled, Slovenia.
[4]	<b>MyMedia: Producing an Extensible Framework for Recommendation .</b>	Paul Marrow, Rich Hanbidge, Steffen Rendle, Christian Wartena, Christoph Freudenthaler	2009	NEM summit 2009, 28--30 September 2009, Saint Malo France
[5]	<b>Data Mining with Microsoft SQL Server 2008</b>	Jamie MacLennan, Zhao Hui Tang, Bogdan Crivat	2008	