

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ**

**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ  
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ**

Διπλωματική εργασία του φοιτητή  
Στανόγια Νικόλαου

με θέμα:

“Πολυπρακτορικό πρωτότυπο για συνεργατικό  
χρονοπρογραμματισμό

“A multiagent prototype for collaborative scheduling”

**Επιβλέποντες καθηγητές**

Δασκαλοπούλου Ασπασία

Λάλης Σπυρίδων – Γεράσιμος

## Ευχαριστίες

Τελειώνοντας τη διπλωματική αυτή, φτάνει πλέον το τέλος της φοιτητικής μου ζωής. Όταν το 2007 εισήχθηκα στο Τμήμα Μηχανικών Η/Υ Τηλεπικοινωνιών και Δικτύων δε μπορούσα να φανταστώ τις έντονες εμπειρίες, τις συγκινήσεις, τις συναισθηματικές αντιθέσεις που θα αποκόμιζα κατά την διάρκεια των σπουδών μου. Γνώρισα πολλούς ανθρώπους, έζησα ποικίλες καταστάσεις, ωρίμασα. Είναι η στιγμή που θέλω να πω **ευχαριστώ** σε όσους με στήριξαν και ήταν δίπλα μου κατά τη διάρκεια των σπουδών. Αρχικά, αισθάνομαι την ανάγκη να ευχαριστήσω τους γονείς μου για την αμέριστη συμπαράσταση όλα αυτά τα χρόνια, ηθική αλλά και οικονομική. Οφείλω επίσης να ευχαριστήσω όλους τους φίλους μου που μαζί ξοδέψαμε τα καλύτερά μας χρόνια. Όσον αφορά τις σπουδές μου, θα ήθελα να ευχαριστήσω όλους τους καθηγητές του ΤΜΗΥΤΔ, οι οποίοι μου πρόσφεραν ποιοτική ακαδημαϊκή γνώση και επιστημονική κατάρτιση. Ευχαριστώ, ιδιαιτέρως την καθηγήτρια κα. Ασπασία Δασκαλοπούλου και τον καθηγητή κ. Σπύρο Λάλη για την πολύτιμη βοήθεια που πρόσφεραν για την ολοκλήρωση της διπλωματικής εργασίας. Τέλος, θα ήθελα να ευχαριστήσω και την Γραμματεία του τμήματος, για την υποστήριξή της κατά την διάρκεια των σπουδών μου.

Βόλος, Φεβρουάριος 2013

Νικόλαος Στανόγιας

# Περίληψη

Αντικείμενο της παρούσας διπλωματικής εργασίας είναι η σχεδίαση και υλοποίηση ενός πολυπρακτορικού συστήματος το οποίο επιτρέπει στους χρήστες του να διαπραγματεύονται μία κοινή συνάντηση ώστε να παρακολουθήσουν μία ταινία στον κινηματογράφο. Συγκεκριμένα, το σύστημα περιέχει μία βάση δεδομένων με διάφορες ταινίες που είτε προβάλλονται στο παρών είτε όχι. Οι χρήστες, αφού αξιολογήσουν έναν αριθμό ταινιών που έχουν δει παίρνουν συστάσεις για ταινίες που προβάλλονται από το σύστημα και στη συνέχεια μπορούν να κανονίσουν μία συνάντηση με οποιονδήποτε άλλο χρήστη που εμφανίζει τα ίδια ενδιαφέροντα. Επιπλέον θα είναι εφικτή και η αλληλεπίδραση μεταξύ των χρηστών αυτών στέλνοντας μηνύματα μεταξύ τους. Οι χρήστες αναπαρίστανται από agents (κάθε χρήστης είναι και ένας διαφορετικός agent στο σύστημα), και σαν πλατφόρμα ανάπτυξης της εφαρμογής, ορίστηκε η πλατφόρμα JADE. Για την διευκόλυνση του χειρισμού κάθε χρήστη αναπτύχθηκε ένα απλό GUI. Οι χρήστες αφού αξιολογήσουν έναν ικανοποιητικό αριθμό ταινιών, μπορούν να επικοινωνήσουν με άλλους χρήστες που παρουσιάζουν τα ίδια ενδιαφέροντα, καθώς και να βρουν εύκολα και γρήγορα ταινίες που ταιριάζουν με τις προτιμήσεις τους. Επιπλέον μπορούν να αναζητήσουν ταινίες ανάλογα με ένα συγκεκριμένο είδος. Για την υλοποίηση της εφαρμογής χρησιμοποιούνται αλγόριθμοι της τεχνικής του συνεργατικού φιλτραρίσματος.

# Abstract

The subject of the present thesis is the design and implementation of a multi-agent system that allows its users to negotiate for a common meeting in order to go and watch a film in cinema. More specifically, the system provides a database with movies that are either available at present or not. Moreover, the interaction of these users is possible by sending messages to each other. Users are represented by agents (each user is an agent with a unique id in the system), and JADE platform was defined for the built of the application. For the facilitation of the users' handling, a simple GUI was introduced. After rating an efficient number of movies, users can communicate with other users who seem to have common interests, and find quickly and readily movies that match their interests as well. In addition, they are able to search for movies according to a specific genre. For the implementation of the application, algorithms of collaborative filtering technique are used.

# Πίνακας Περιεχομένων

---

<b>1 Εισαγωγή</b> .....	7
1.1 Εισαγωγή στον χρονοπρογραμματισμό.....	7
1.2 Εισαγωγή στα συστήματα συστάσεων και το συλλογικό φιλτράρισμα.....	8
1.3 Αντικείμενο διπλωματικής .....	10
1.4 Οργάνωση κειμένου .....	10
<b>2 Σχετικές εργασίες</b> .....	11
2.1 Εφαρμογές χρονοπρογραμματισμού συναντήσεων .....	11
2.1.1 Doodle .....	11
2.1.2 Tungle .....	12
2.1.3 ScheduleOnce .....	13
2.2 Συστήματα συστάσεων ταινιών .....	13
2.2.1 Jinni .....	14
2.2.2 Netflix .....	14
2.2.3 Movielens .....	14
<b>3 Θεωρητικό υπόβαθρο</b> .....	15
3.1 Η έννοια του πράκτορα λογισμικού .....	15
3.2 Η γλώσσα ACL (Agent Communication Language) .....	16
3.3 Η συμπεριφορά ενός πράκτορα και η κλάση Behavior .....	17
3.4 Η κλάση Agent .....	19
3.5 JADE Οντολογίες .....	22
3.6 Η κλάση GuiAgent .....	26

<b>4 Σχεδίαση Συστήματος</b> .....	27
4.1 Πράκτορες του συστήματος .....	27
4.1.1 UserAgent .....	27
4.1.2 FacilitatorAgent .....	30
4.1.3 ServerAgent .....	31
4.1.4 StartAgent .....	32
4.2 Οντολογία του συστήματος .....	32
4.3 Επιλογή γλώσσας περιεχομένου .....	34
4.4 Αλληλεπιδράσεις μεταξύ πρακτόρων της εφαρμογής .....	34
<b>5 Υλοποίηση</b> .....	36
5.1 Συλλογικό Φιλτράρισμα .....	36
5.2 Συλλογή των προτιμήσεων .....	37
5.3 Αποθήκευση των βαθμολογιών σε τοπική βάση δεδομένων .....	37
5.4 Αποθήκευση των βαθμολογιών στο cloud.....	38
5.4.1 Η υπηρεσία AmazonS3.....	38
5.5 Βρίσκοντας παρόμοιους χρήστες .....	39
5.5.1 Βαθμός με βάση την Ευκλείδεια Απόσταση .....	39
5.5.2 Βαθμός με βάση τη συσχέτιση κατά Pearson .....	40
5.6 Σύσταση ταινιών .....	42
5.7 Βρίσκοντας ταινίες με βάση τον τίτλο .....	43
5.8 Επικοινωνία μεταξύ των χρηστών .....	44
5.9 Παραδειγματική εκτέλεση .....	45
5.10 Πλατφόρμες και προγραμματιστικά εργαλεία .....	54
5.10.1 Ενσωματωμένο περιβάλλον ανάπτυξης (IDE) Netbeans .....	54
5.10.2 Πλατφόρμα JADE .....	55
5.10.3 Εκκίνηση του περιβάλλοντος JADE στο Netbeans .....	56
<b>6 Επίλογος</b> .....	59
6.1 Σύνοψη και συμπεράσματα .....	59
6.2 Μελλοντικές επεκτάσεις .....	60
<b>7 Βιβλιογραφία</b> .....	61

# 1 Εισαγωγή

## 1.1 Εισαγωγή στον χρονοπρογραμματισμό

Πόσες φορές θέλετε να βγείτε για έναν καφέ με δυο φίλους και πρέπει να κάνετε 3-4 τηλεφωνήματα μέχρι να συνεννοηθείτε; Όχι ο ένας δεν μπορεί τότε, όχι ο άλλος μπορεί αλλά προτιμάει λίγο αργότερα κ.ο.κ.

Σκεφτείτε το πρόβλημα σε ακόμα υψηλότερο επίπεδο. Έχετε μια εταιρία και θέλετε να κάνετε ένα meeting με τους 15 συνεργάτες σας από διάφορες πόλεις. Πόση ώρα πιστεύετε ότι χρειάζεται να καταναλώσετε μέχρι να συμφωνήσουν όλοι με όλους;

Στην επιχειρηματική ζωή λοιπόν και όχι μόνο, πραγματοποιούνται συχνά συναντήσεις προκειμένου να μοιραστούν καινούργιες ιδέες και να διατηρηθεί ο συντονισμός. Ο χρονοπρογραμματισμός των συναντήσεων είναι συχνά μια χρονοβόρα διαδικασία που μπορεί μερικές φορές να γίνει πολύ βαρετή. Τις περισσότερες φορές, οι άνθρωποι πρέπει να προγραμματίσουν πολλές συνεδριάσεις την ίδια στιγμή, λαμβάνοντας υπόψη διάφορους περιορισμούς όπως ένα κατάλληλο χρονικό όριο, μία κατάλληλη τοποθεσία και / ή έναν ορισμένο αριθμό συμμετοχόντων. Η διαδικασία της αναζήτησης για ένα ευρέως διαθέσιμο χρόνο, τόπο κλπ, μπορεί να είναι απογοητευτική και να οδηγήσει σε υπο-βέλτιστες λύσεις με την παρουσία αρκετών συναντήσεων να προγραμματίζονται ταυτόχρονα. Κάποιος που έχει επιχειρήσει να προγραμματίσει μία συνάντηση με περισσότερα από 3 άτομα μπορεί να εκτιμήσει τη χρησιμότητα ενός είδους αυτοματοποιημένης βοήθειας. Ως εκ τούτου, η αυτοματοποίηση του προγραμματισμού συναντήσεων μπορεί να εξοικονομήσει χρόνο και προσπάθεια εκ μέρους των συμμετοχόντων και επίσης μπορεί να οδηγήσει σε πιο αποτελεσματικά χρονοδιαγράμματα.

Τα περισσότερα εμπορικά συστήματα που έχουν αναπτυχθεί μέχρι στιγμής για τη διευκόλυνση μιας συνάντησης, επιτρέπουν σε κάποιον χρήστη να περιηγηθεί σε ένα ηλεκτρονικό ημερολόγιο άλλων χρηστών προκειμένου να βρει ένα άδειο slot, και να κλείσει ένα ραντεβού στα πλαίσια του χρόνου που επιτρέπεται. Μερικά παραδείγματα αναφέρονται στο επόμενο κεφάλαιο.

Στην δική μας προσέγγιση χρησιμοποιούμε πράκτορες για την αυτοματοποίηση της διαδικασίας του χρονοπρογραμματισμού. Τα πολυπρακτορικά συστήματα παρέχουν τις

κατάλληλες λύσεις για τον χρονοπρογραμματισμό, επειδή ο χρονοπρογραμματισμός είναι ως επί το πλείστον μια επαναληπτική εργασία που μπορεί να οδηγήσει σε χάσιμο χρόνου. Η κατανεμημένη φύση του χρονοπρογραμματισμού συναντήσεων, που αφορά κυρίως την κατανεμημένη αναζήτηση ημερολογίου ταιριάζει επίσης καλά με την κατανεμημένο μοντέλο υπολογισμού βασισμένο σε έναν πράκτορα. Επιπλέον οι αυτόνομοι πράκτορες ταιριάζουν απόλυτα στη διαδικασία χρονοπρογραμματισμού μιας συνάντησης, καθώς μπορούν να αποφασίσουν πώς ο τελικός στόχος μπορεί να επιτευχθεί εκ μέρους των χρηστών τους και μπορούν να εκτελέσουν τις απαραίτητες ενέργειες για αυτό. Καθώς οι πράκτορες χειρίζονται τις αλληλεπιδράσεις με άλλους πράκτορες, οι χρήστες δε συμμετέχουν στις διαπραγματεύσεις και δε χρειάζεται να παρακολουθούν τη διαδικασία του χρονοπρογραμματισμού. Επίσης η χρήση των πρακτόρων μειώνουν το κλασικό πρόβλημα της συμφόρησης, όταν πολλά άτομα προσπαθούν να επικοινωνήσουν με ένα άτομο.

Στα πλαίσια αυτής της εργασίας ο χρονοπρογραμματισμός έγκειται στη διαπραγμάτευση μεταξύ χρηστών προκειμένου να βρουν μια κοινή ώρα και ημέρα για μία ταινία που πρόκειται να παρακολουθήσουν στον κινηματογράφο. Ο κάθε χρήστης επιλέγει ταινίες ανάλογα με τις συστάσεις που παίρνει από το σύστημα με την τεχνική του συλλογικού φιλτραρίσματος.

## 1.2 Εισαγωγή στα συστήματα συστάσεων και το συλλογικό φιλτράρισμα

Προκειμένου να τραβήξουν την προσοχή των χρηστών και να αυξήσουν την ικανοποίησή τους για την παροχή αποτελεσμάτων σε μία αναζήτηση, οι προγραμματιστές των μηχανών αναζήτησης και οι πωλητές προσπαθούν να προβλέψουν τις προτιμήσεις του χρήστη με βάση τη συμπεριφορά του. Οι συστάσεις που παρέχονται στους χρήστες είναι είτε από τις μηχανές αναζήτησης είτε από προμηθευτές. Συστήματα συστάσεων εφαρμόζονται κυρίως σε εμπορικές ιστοσελίδες, όπου οι ακριβείς προβλέψεις μπορούν να οδηγήσουν σε υψηλότερα ποσοστά πωλήσεων. Οι κύριες λειτουργίες των συστημάτων συστάσεων περιλαμβάνουν την ανάλυση των δεδομένων των χρηστών και την εξαγωγή χρήσιμων πληροφοριών για περαιτέρω προβλέψεις. Τα συστήματα συστάσεων έχουν σχεδιαστεί για να επιτρέπουν στους χρήστες να εντοπίζουν τα προτιμητέα είδη γρήγορα, και να αποφευχθεί κάθε δυνατή υπερφόρτωση πληροφορίας. Έχουν εφαρμογή σε τεχνικές εξόρυξης δεδομένων για τον προσδιορισμό της ομοιότητας ανάμεσα σε χιλιάδες ή ακόμα και εκατομμύρια δεδομένα. Μερικές ιστοσελίδες που χρησιμοποιούν μηχανές συστάσεων είναι οι εξής:

- **Amazon:** Όσοι έχετε επισκεφτεί το Amazon σίγουρα θα έχετε παρατηρήσει μια περιοχή της ιστοσελίδας που έχει την ετικέτα “Customers Who Bought This Item Also Bought”,



με μία λίστα παρόμοιων προϊόντων κάτω από την ετικέτα. Επίσης αν έχετε προμηθευτεί κάτι από το Amazon είναι σίγουρο ότι συχνά θα βλέπετε στα mail σας μηνύματα τα οποία σας προτείνουν νέες αγορές που θα ικανοποιούν τα ενδιαφέροντά σας. Το Amazon λοιπόν συλλέγει πληροφορίες σχετικά με διάφορα προϊόντα που αγοράζουν οι χρήστες, καθορίζει ποιοι από αυτούς τους χρήστες έχουν κοινά ενδιαφέροντα και ποια προϊόντα μοιάζουν με κάποια άλλα, και στη συνέχεια χρησιμοποιεί αυτές τις πληροφορίες για να εξάγει συστάσεις προς τους χρήστες ανάλογα με τις αγορές τους.

- **Youtube:** Ανάλογα με το τι είδος μουσικής ακούτε, ποιοι καλλιτέχνες σας αρέσουν και πολλά άλλα κριτήρια, το Youtube σας προτείνει και τα ανάλογα βίντεο. Σίγουρα θα έχετε παρατηρήσει ακόμα, ότι σε κάθε βίντεο που παίζεται, στο δεξί μέρος της ιστοσελίδας υπάρχει μια ετικέτα με το μήνυμα Suggestions που προβάλλει βίντεο παρόμοια με αυτό που παίζεται εκείνη τη στιγμή.
- **Reddit:** Ένα πολύ ενδιαφέρον site το οποίο επιτρέπει στο χρήστη να ψηφίζει συνδέσμους προς άλλους δικτυακούς τόπους και στη συνέχεια χρησιμοποιεί τις ψήφους αυτές για να προτείνει άλλους συνδέσμους τους οποίους ο χρήστης είναι πιθανό να βρίσκει ενδιαφέρον.
- **eHarmony:** Σε αυτούς τους καιρούς που ζούμε, οι άνθρωποι χρησιμοποιούν στρατηγικές προκειμένου να επιλέξουν ακόμα και με ποιον θα βγούν ραντεβού! Εάν έχετε κουραστεί λοιπόν να σπαταλάτε τις νύχτες σας στα μπαρ προσπαθώντας να βρείτε κάποιον/α για να βγείτε ραντεβού το eHarmony είναι μία καλή λύση. Στην πολυάσχολη ζωή μας, μπορεί να είναι δύσκολο συναντήσετε κάποιον/α, αλλά ακόμα και αν το κάνετε δεν είστε σίγουροι αν μοιράζεστε τα ίδια ενδιαφέροντα και πεποιθήσεις που θα σας οδηγήσουν σε μία μακροχρόνια και ευτυχισμένη σχέση. Με βάση λοιπόν 29 διαφορετικές διαστάσεις το eHarmony συλλέγει πληροφορίες από τους συμμετέχοντες και καθορίζει ποιος/α θα μπορούσε να είναι ένα καλό ταίρι.

Από τα παραπάνω παραδείγματα είδαμε ότι οι προτιμήσεις μπορούν να συγκεντρωθούν με πολλούς διαφορετικούς τρόπους. Μερικές φορές τα δεδομένα είναι αντικείμενα που έχουν αγοραστεί από ανθρώπους, και απόψεις σχετικά με αυτά τα αντικείμενα μπορούν να αναπαρασταθούν με θετικούς/αρνητικούς ψήφους, ή με βαθμολόγηση από το ένα μέχρι το πέντε. Στα πλαίσια αυτής της διπλωματικής θα εξετάσουμε τρόπους και αλγορίθμους που εκπροσωπούν αυτές τις περιπτώσεις, χρησιμοποιώντας κριτικές από χρήστες για διάφορες ταινίες.

### **1.3 Αντικείμενο διπλωματικής**

Στα πλαίσια της εργασίας αυτής δημιουργήσαμε μια πολυπρακτορική εφαρμογή με τη χρήση της δημοφιλούς πλατφόρμας JADE η οποία παρέχει επίσης συμβατότητα με τις προδιαγραφές της FIPA. Οι χρήστες μπορούν να βαθμολογήσουν όσες ταινίες έχουν δει από μία βάση δεδομένων με ταινίες και στη συνέχεια να ψάξουν για όσες ταινίες δεν έχουν δει και είναι πιθανόν να τους αρέσουν. Επιπλέον μπορούν να περιορίσουν την αναζήτηση ταινιών ανάλογα με συγκεκριμένες ταινίες που θα θέλουν να μοιάζουν. Στη συνέχεια οι χρήστες μπορούν να κανονίσουν μία συνάντηση μεταξύ τους για να δουν μια ταινία στους κινηματογράφους. Τέλος, η εφαρμογή λειτουργεί και σαν κοινωνικό δίκτυο για φίλους του κινηματογράφου και έτσι μπορούν να προκύψουν και φιλίες μεταξύ των χρηστών.

### **1.4 Οργάνωση κειμένου**

Το κείμενο της διπλωματικής εργασίας παρουσιάζει την ακόλουθη διάρθρωση. Στο 1<sup>ο</sup> κεφάλαιο έχουμε μια εισαγωγή στο θέμα της διπλωματικής εργασίας. Στο κεφάλαιο 2 παρουσιάζονται σχετικές εργασίες και ακολουθεί σύγκριση αυτών με τη δική μας υλοποίηση. Θέματα θεωρητικού υπόβαθρου συζητώνται στο κεφάλαιο 3. Στο κεφάλαιο 4 προχωρούμε στον σχεδιασμό των λειτουργιών και της αρχιτεκτονικής του συστήματος. Συνεχίζοντας, στο κεφάλαιο 5 προχωρούμε σε παρουσίαση του τρόπου υλοποίησης της εφαρμογής. Ο επίλογος στο κεφάλαιο 6 ολοκληρώνει τη διάρθρωση του κειμένου και τέλος, στο κεφάλαιο 7 παρουσιάζεται η βιβλιογραφία που απαιτήθηκε.

## 2 Σχετικές εργασίες

### 2.1 Εφαρμογές χρονοπρογραμματισμού συναντήσεων

Πολλές είναι οι online εφαρμογές που προσπαθούν να εξαλείψουν τον πονοκέφαλο του χρονοπρογραμματισμού συναντήσεων. Σε αντίθεση μάλιστα με το Microsoft Outlook και το iCal, είναι συμβατές με το Google Calendar και δεν απαιτούν από τους προσκεκλημένους να εγκαταστήσουν κάποιο λογισμικό ή να κάνουν εγγραφή στην ιστοσελίδα τους. Παρουσιάζουμε στη συνέχεια τρεις ελκυστικές εφαρμογές που κάνουν τον χρονοπρογραμματισμό συναντήσεων εύκολο έργο.

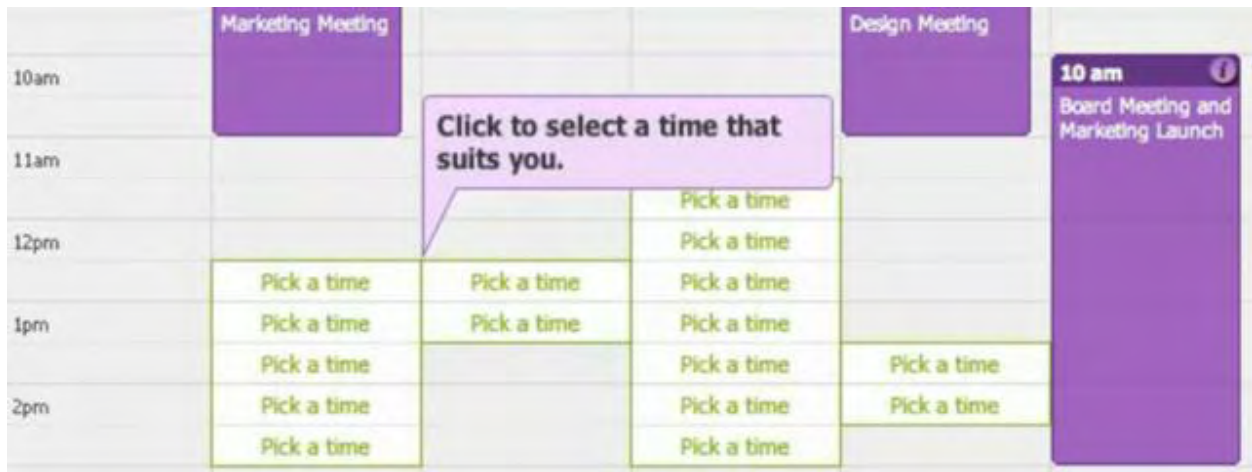
#### 2.1.1 [Doodle](#)

April 2010					
	Wed 14	Thu 15	Mon 19		
	12:00 PM - 2:00 PM	9:15 AM - 11:15 AM	2:45 PM - 4:45 PM	9:15 AM - 11:15 AM	2:45 PM - 4:45 PM
Paul		OK		OK	OK
Your name	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Count	0	1	0	1	1

Ουσιαστικά πρόκειται για μια υπηρεσία διοργάνωσης event, είτε αυτό μπορεί να είναι επαγγελματική συνάντηση, είτε γιορτή, είτε ομαδική εκδρομή. Με το doodle δημιουργείτε ένα

event και το κοινοποιείτε στα άτομα που σας ενδιαφέρει να παραβρεθούν. Αυτοί απαντούν ανάλογα αν μπορούν ή όχι και αν όχι ποιες ημερομηνίες θα τους βόλευαν.Εσείς μπορείτε να δείτε τι έχει σημειώσει ο καθένας αλλά και να κοινοποιήσετε τις σημειώσεις του ενός στον άλλον.Έτσι όλες οι διαθέσιμες ημερομηνίες μαζεύονται μαζί και με το doodle έχετε λύσει σε λίγα λεπτά ένα πρόβλημα που σας ταλαιπωρούσε για μέρες.

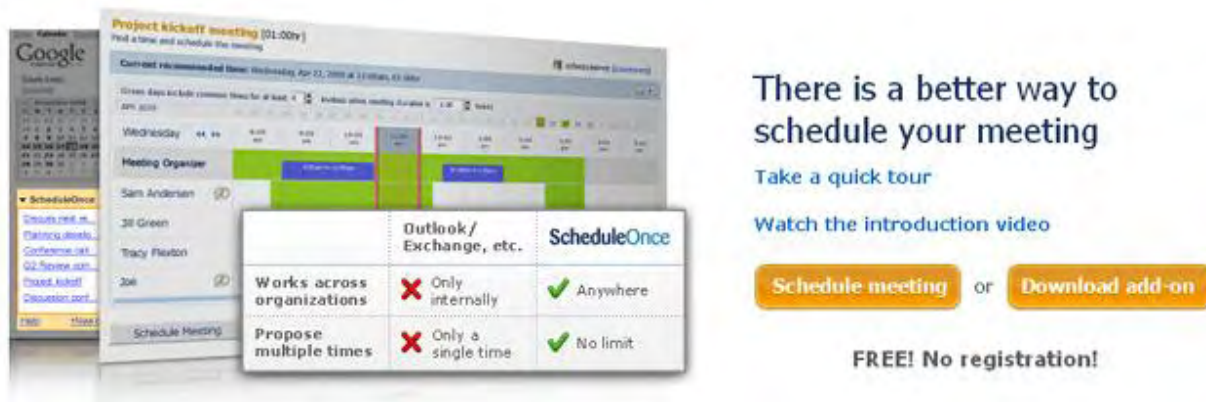
## 2.1.2 Tungle



Το Tungle είναι μια online υπηρεσία που σας δίνει την δυνατότητα να συγχρονίζετε το παρόν ημερολόγιο σας όπως το Google ή Outlook, με τον online λογαριασμό σας και επιπλέον οι πληροφορίες που περικλείονται σε αυτό μπορούν να μοιραστούν online μέσω του προσωπικού σας URL το οποίο οι άλλοι μπορούν να χρησιμοποιήσουν για να ελέγξουν την διαθεσιμότητα σας.Έχοντας δημιουργήσει ένα λογαριασμό, μπορείτε να κάνετε χρήση της εφαρμογή Tungle στο iPhone σας. Καθώς η εφαρμογή σας προσφέρει την δυνατότητα να συγχρονίζετε τα ημερολόγια σας συμπεριλαμβανομένων και αυτών σε Mac και Windows formats, είναι εύκολο να ελέγξετε την διαθεσιμότητα ενός συνεργάτη σας ανεξάρτητα από το τι υπολογιστή χρησιμοποιεί.Με την εφαρμογή αυτή είναι πιθανόν να προγραμματίσετε συναντήσεις στέλνοντας αιτήσεις προς τις επαφές σας.Μπορείτε να μοιραστείτε το ημερολόγιο σας με τους άλλους και αν έχετε πρόσβαση στα ημερολόγια τους μπορείτε να τοποθετήσετε το ημερολόγιο σας πάνω στα ημερολόγια των άλλων ώστε να υπολογίσετε αντίστοιχες στιγμές όπου και οι δύο μπορείτε για να κανονίσετε την συνάντηση.Εάν έχετε αρκετές στιγμές διαθέσιμες για μία συνάντηση, μπορείτε να συμπεριλάβετε κάθε μια από αυτές τις πιθανές στιγμές στην πρόσκληση σας.Καθώς ο προσκεκλημένος απαντάει

υποδεικνύοντας σε ποια από τις πιθανές συναντήσεις θα μπορέσει να παρευρεθεί, το Tungle μπορεί αυτόματα να καθορίσει ποια είναι η καλύτερη στιγμή για όλους ώστε να μπορέσουν να παρευρεθούν. Ακόμα και όταν ένας μικρός σχετικά αριθμός ατόμων εμπλέκονται, το Tungle είναι ένα εργαλείο που σας γλιτώνει πολύτιμο χρόνο και σας αφήνει να επικεντρωθείτε περισσότερο στην εργασία σας.

### 2.1.3. [ScheduleOnce](#)



The image shows a screenshot of the ScheduleOnce interface. On the left, there's a Google Calendar snippet. The main part shows a meeting titled "Project kickoff meeting [01:00v]" with a "Current recommended time" of Wednesday, Apr 27, 2011 at 11:00am. Below this is a calendar grid for Wednesday, April 27, 2011, with meeting slots highlighted in green. A list of meeting organizers is shown: Sam Anderson, Jill Green, Tracy Fleaton, and Job. A comparison table is overlaid on the bottom right of the screenshot:

	Outlook/ Exchange, etc.	ScheduleOnce
Works across organizations	✗ Only internally	✓ Anywhere
Propose multiple times	✗ Only a single time	✓ No limit

Below the screenshot, there is a promotional banner with the text: "There is a better way to schedule your meeting". It includes links for "Take a quick tour" and "Watch the introduction video". There are two orange buttons: "Schedule meeting" and "Download add-on". At the bottom, it says "FREE! No registration!"

Η γραφική διεπαφή χρήστη του ScheduleOnce είναι παρόμοια με αυτή του Tungle, και απαιτείται περίπου το ίδιο χρονικό διάστημα για να προγραμματιστεί μία συνάντηση. Σε αντίθεση με το Tungle, οι χρόνοι των συναντήσεων επιλέγονται σε μία οριζόντια γραμμή του χρόνου. Αυτό φαίνεται να είναι πιο έξυπνο αν ο χρόνος συνάντησης που εξετάζεται είναι μόνο για μία μέρα. Το ScheduleOnce διαθέτει ενσωματωμένο plugin για το Google Calendar και το Gmail, όπου τοποθετείται ένα κουτί (στο Gmail) ακριβώς πάνω από το παράθυρο συνομιλίας που εμφανίζει τις προσεχείς συνεδριάσεις.

## 2.2 Συστήματα συστάσεων ταινιών

Όσον αφορά τα συστήματα συστάσεων ταινιών, είναι λογικό το γεγονός ότι τα τελευταία χρόνια υπάρχει πληθώρα αντίστοιχων εγχειρημάτων στο διαδίκτυο. Οι λόγοι ύπαρξης αυτών είναι προφανής και σχετίζονται με την ανάγκη των οπαδών του κινηματογράφου για ταινίες που ταιριάζουν με τα ενδιαφέροντά τους. Υπάρχουν λοιπόν δεκάδες μηχανές συστάσεων ταινιών στο διαδίκτυο από τις οποίες κάποιες απαιτούν λίγη ή και καθόλου είσοδο δεδομένων από τον χρήστη, πριν του εμφανίσουν τίτλους ταινιών, ενώ άλλες θέλουν να μάθουν ακριβώς ποια είναι τα ενδιαφέροντά του. Ακολουθούν οι πιο αξιόλογες.

### 2.2.1 [Jinni](#)

Το Jinni είναι αυτή τη στιγμή η καλύτερη μηχανή συστάσεων ταινιών στο Διαδίκτυο. Είτε θέλετε να αναζητήσετε ταινίες στο πεδίο αναζήτησης, είτε θέλετε να βρείτε ταινίες ανάλογα με τη διάθεση σας, το διαθέσιμο χρόνο που έχετε ή τα σχόλια που έχουν γίνει, αυτή η υπηρεσία τα έχει όλα. Ψάχνω ταινίες ανάλογα με τη διάθεση μου και την πλοκή που προτιμώ, και κάθε φορά εμφανίζονται ταινίες που έχω δει και έχω αγαπήσει στο παρελθόν ή ό,τι δεν έχω δει αλλά σίγουρα θα ήθελα να δω. Αλλά ίσως το πιο συναρπαστικό χαρακτηριστικό που προσφέρει το Jinni είναι η σημασιολογική αναζήτηση του. Μπορείτε να εισάγεται όρους όπως “ταινίες που έχουν γκάνγκστερ”, ή “ταινίες που δείχνουν Jason Statham να παλεύει” και η ιστοσελίδα επιστρέφει ταινίες που ταιριάζουν στις επιλογές σας. Χρησιμοποιήστε το Jinni και δε θα το μετανιώσετε.

### 2.2.2 [Netflix](#)

Το Netflix ζητάει από το χρήστη να αξιολογήσει κάποιες ταινίες πριν καθορίσει τις κατάλληλες ταινίες για να δει στη συνέχεια. Είναι πολύ εύκολο στο χειρισμό και η λίστα με τις ταινίες που επιστρέφει δεν είναι και τόσο μεγάλη. Το 2006 το Netflix ανακοίνωσε έναν διαγωνισμό με έπαθλο 1.000.000 δολάρια σε αυτόν που θα καταφέρνε να βελτιώσει το σύστημα συστάσεων κατά 10 τοις εκατό, μαζί με βραβεία προόδου 50.000 δολλάρων στην τρέχων καλύτερη ομάδα όσο “έτρεχε” ο διαγωνισμός. Χιλιάδες ομάδες από όλον τον κόσμο μαζεύτηκαν, και τον Απρίλιο του 2007 η ηγετική ομάδα κατάφερε να βελτιώσει το σύστημα κατά 7 τοις εκατό. Χρησιμοποιώντας δεδομένα σχετικά με τις ταινίες που απολαμβάνει ο χρήστης, το Netflix μπορεί να προτίνει ταινίες, για τις οποίες δεν έχουν ακούσει ποτέ, σε άλλους χρήστες. Και ας μη ξεχνάμε ότι οποιοδήποτε τρόπος βελτίωσης του συστήματος θα αποφέρει πολλά λεφτά!

### 2.2.3 [Movielens](#)

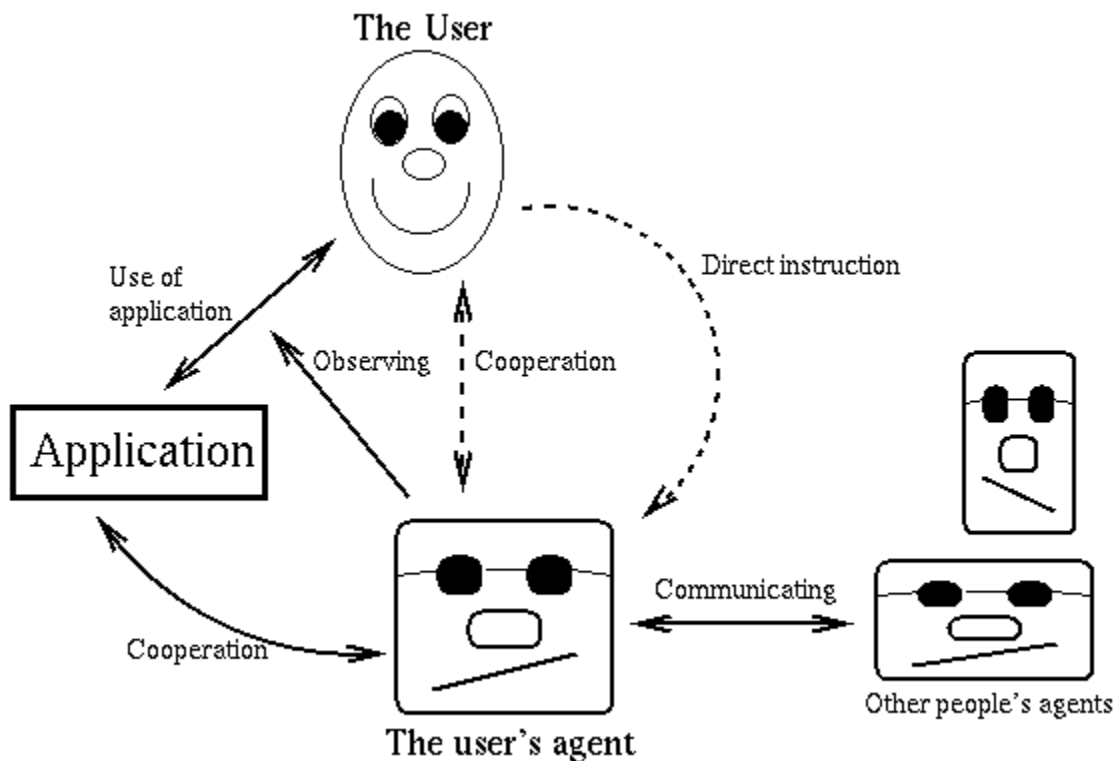
Το Movielens μπορεί να υστερεί λίγο σε ομορφιά σχεδίασης, αλλά είναι μια πανίσχυρη μηχανή συστάσεων η οποία εκτιμά τις προτιμήσεις του χρήστη ανάλογα με τις αξιολογήσεις που έχει κάνει σε ταινίες που έχει ήδη δει. Αφού αξιολογήσει ο χρήστης 15 ταινίες του επιστρέφονται ακριβείς συστάσεις που σίγουρα θα ταιριάζουν με τα ενδιαφέροντα του.

## 3 Θεωρητικό υπόβαθρο

Για την παρούσα εργασία απαιτήθηκαν γνώσεις οι οποίες προετοίμασαν το έδαφος για την υλοποίηση της εφαρμογής. Πριν ασχοληθούμε με το αλγοριθμικό κομμάτι και τη γενικότερη σχεδίαση του συστήματος, έπρεπε να κατανοήσουμε κάποια βασικά χαρακτηριστικά καθώς και κάποιες πιο προχωρημένες έννοιες της πλατοφόρμας JADE.

### 3.1 Η έννοια του πράκτορα λογισμικού

Όπως ήδη γνωρίζουμε ο πράκτορας (agent) είναι ένα κομμάτι λογισμικού που αντιλαμβάνεται το περιβάλλον του και δρα για κάποιον χρήστη ή καποιον άλλο πράκτορα εκ μέρους εκείνου. Αυτή η δράση υποκδυκνύει την εξουσιοδότηση για λήψη απόφασης για κάποια επόμενη δράση, σε περίπτωση που προβλέπεται μία.



**Εικόνα 3-1.** Η έννοια του πράκτορα λογισμικού

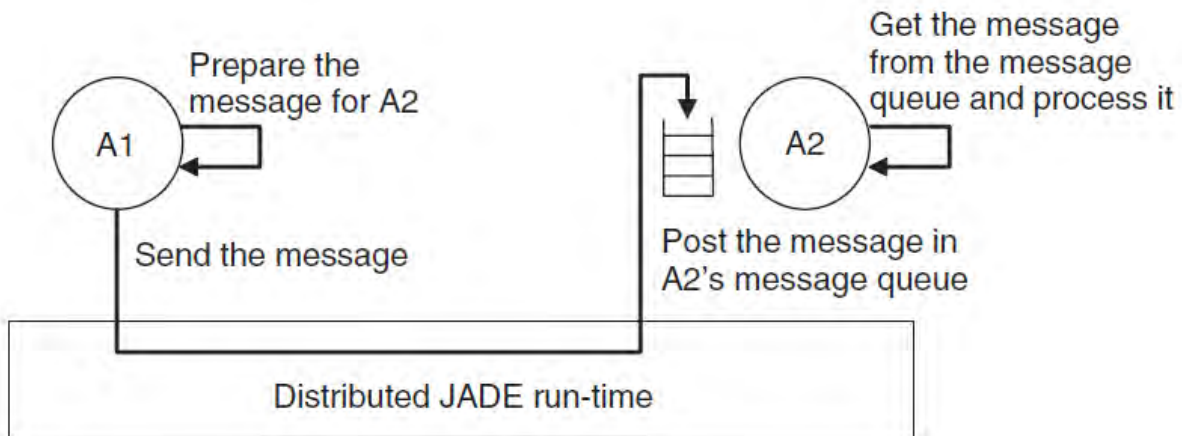
Η εφαρμογή μας όπως προαναφέρθηκε χρησιμοποιεί πράκτορες που δημιουργούνται και επικοινωνούν στα πλαίσια της πλατφόρμας εκτέλεσης και επικοινωνίας JADE. Με τη χρήση των

JADE πρακτόρων έχουμε στη διάθεσή μας επιπλέον λειτουργίες και εργαλεία που διευκολύνουν τη χρήση τους. Τα βασικότερα αυτών παρουσιάζονται ακολούθως.

### 3.2 Η γλώσσα ACL (Agent Communication Language)

Η επικοινωνία των πρακτόρων είναι πιθανόν το πιο στοιχειώδες στοιχείο της πλατφόρμας JADE και υλοποιείται με τις προδιαγραφές FIPA όπως επισημάναμε και στην εισαγωγή.

Το μοντέλο επικοινωνίας βασίζεται στην ασύγχρονη ανταλλαγή μηνυμάτων. Έτσι, κάθε πράκτορας έχει μία θυρίδα, στην οποία μπαίνουν τα εισερχόμενα μηνύματα από άλλους πράκτορες. Όταν εισέρχεται ένα μήνυμα στη θυρίδα ειδοποιείται ο πράκτορας που το έλαβε. Ωστόσο, ο τρόπος με τον οποίο ο πράκτορας θα λάβει το μήνυμα από τη θυρίδα ή αν θα το λάβει τελικά, εξαρτάται από τον προγραμματιστή. Η διαδικασία αυτή απεικονίζεται στην παρακάτω εικόνα.



Εικόνα 3-2. Απεικόνιση του τρόπου με τον τον οποίο δέχονται οι πράκτορες μηνύματα

Κάθε μήνυμα που ανταλλάσσεται μεταξύ JADE πρακτόρων περιλαμβάνει τα ακόλουθα πεδία:

- Τον αποστολέα του μηνύματος.
- Τη λίστα με τους παραλήπτες.
- Την πρόθεση της επικοινωνίας, που αναφέρεται στο λόγο για τον οποίο γίνεται η επικοινωνία. Οι λόγοι μπορεί να είναι αίτηση (**Request**), αν ο αποστολέας επιθυμεί ο παραλήπτης να εκτελέσει μία λειτουργία, ενημέρωση (**Inform**), αν ο αποστολέας επιθυμεί ο παραλήπτης να είναι ενήμερος από κάποιο γεγονός, ερώτηση (**Query if**) για να επιβεβαιώσει ο αποστολέας αν μια συνθήκη είναι αληθής, κλήση για προτάσεις (**Call For Proposal**), πρόταση (**Proposal**), αποδοχή (**Accept Proposal**) και απόρριψη (**Reject**).
- Το περιεχόμενο, το οποίο καθορίζεται από την πρόθεση επικοινωνίας.



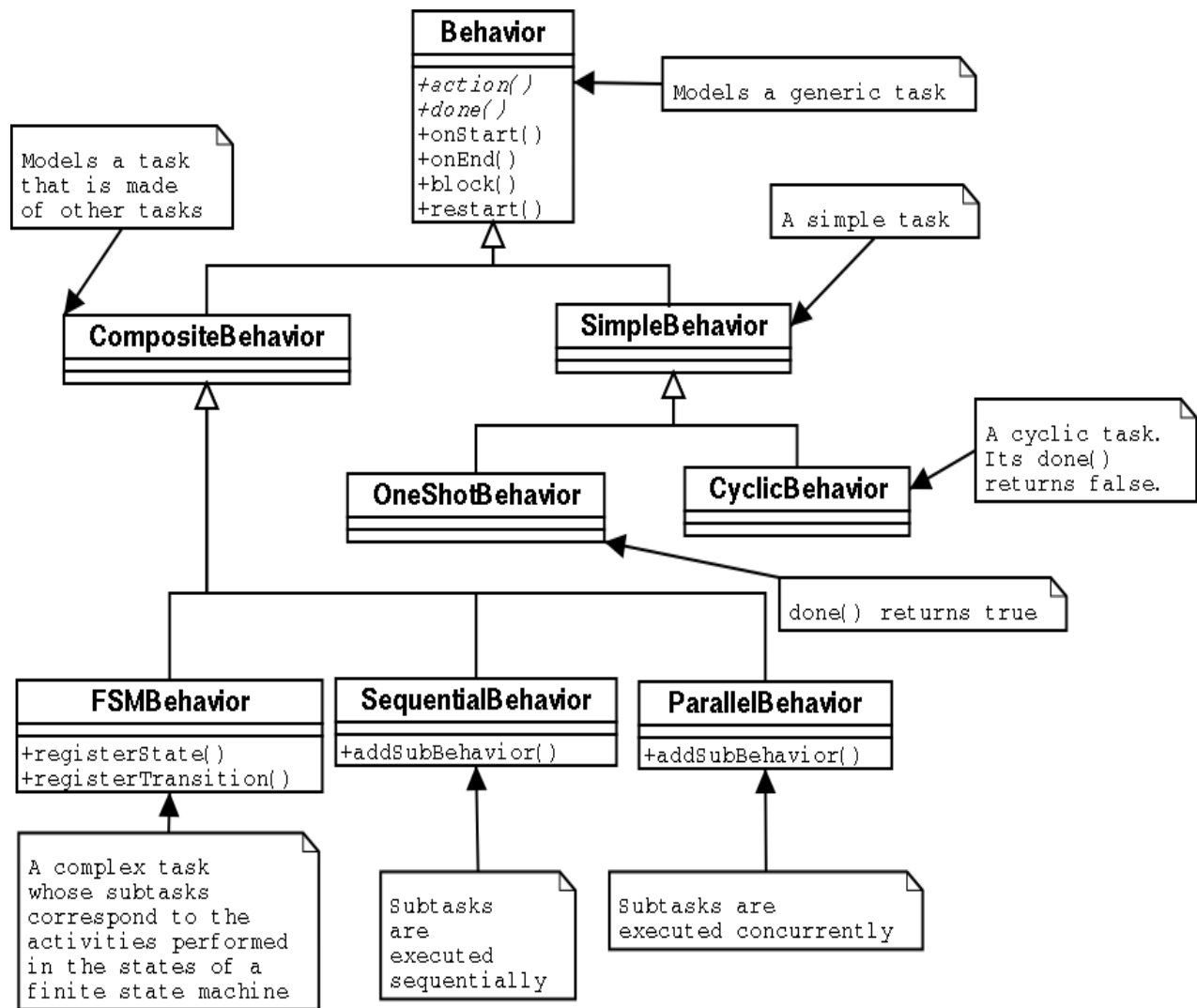
- Τη γλώσσα περιεχομένου.
- Την οντολογία που αναφέρεται στην έννοια των συμβόλων που χρησιμοποιούν αποστολείς και παραλήπτες. Εδώ πρέπει παραλήπτες και αποστολείς να αποδίδουν το ίδιο νόημα στα χρησιμοποιούμενα σύμβολα.
- Μερικά επιπρόσθετα πεδία που ρυθμίζουν συζητήσεις που γίνονται ταυτόχρονα και καθορίζουν τα χρονικά περιθώρια για τη λήψη μιας απάντησης.

### 3.3 Η συμπεριφορά ενός πράκτορα και η κλάση Behavior

Η εργασία που εκτελεί ένας πράκτορας διεξάγεται μέσω συμπεριφορών. Έτσι, ένα σύνολο συμπεριφορών προγραμματίζονται και εκτελούνται για να διεκπεραιώσουν τις υποχρεώσεις του κάθε πράκτορα. Οι πράκτορες της κλάσης agent, που θα παρουσιαστεί εκτενώς παρακάτω μπορούν να προσθέσουν ή να αφαιρέσουν συμπεριφορές με τις ακόλουθες μεθόδους:

```
addBehavior (Behavior)
removeBehavior (Behavior)
```

Ο λόγος για τον οποίο η προσθήκη / αφαίρεση συμπεριφορών είναι ουσιώδης είναι το γεγονός ότι ένας πράκτορας πρέπει να εκτελεί πολλές εργασίες και να ανταποκρίνεται σε πολλά εξωτερικά ερεθίσματα παράλληλα. Το τελευταίο επιτυγχάνεται με την εισαγωγή της κλάσης behavior, η οποία δίνει τη δυνατότητα στον πράκτορα να ανταποκρίνεται κατάλληλα στα ερεθίσματα. Η εκτέλεσή τους δε γίνεται παράλληλα, αλλά διαδοχικά. Κάθε φορά που ολοκληρώνεται ένας κύκλος λειτουργίας μιας behavior συνεχίζει η επόμενη. Το τέλος ενός κύκλου λειτουργίας μιας behavior καθορίζεται από το χρήστη. Με τον τρόπο αυτό μπορούμε να πραγματοποιήσουμε πολύπλοκες διεργασίες για τους πράκτορες. Έτσι, η κλάση behavior προσφέρει μεθόδους για την ενεργοποίηση της συμπεριφοράς (action()), την ολοκλήρωσή της (done()), και την επανεκκίνησή της (restart()).



**Εικόνα 3-3.** Δομή της κλάσης Behaviour

Η βιβλιοθήκη της JADE έχει στη διάθεσή της δύο βασικές κατηγορίες Behaviors για την καλύτερη εξυπηρέτηση του προγραμματιστή. Αυτές είναι οι:

- **SimpleBehavior:** είναι η απλούστερη συμπεριφορά ενός διαμεσολαβητή, που είτε εκτελείται μια φορά (OneShotBehavior), είτε εκτελούνται περιοδικά (CyclicBehavior).
- **CompositeBehavior:** αποτελούν τις πιο περίπλοκες συμπεριφορές, όπου αποτελούνται από πολλές υποσυμπεριφορές και οι οποίες εκτελούνται ανάλογα με τις ανάγκες της εφαρμογής. Είναι είτε μία σειρά από καθορισμένες διαδοχικές συμπεριφορές, είτε μία σειρά από συμπεριφορές που εκτελούνται ταυτόχρονα, είτε μία σειρά από

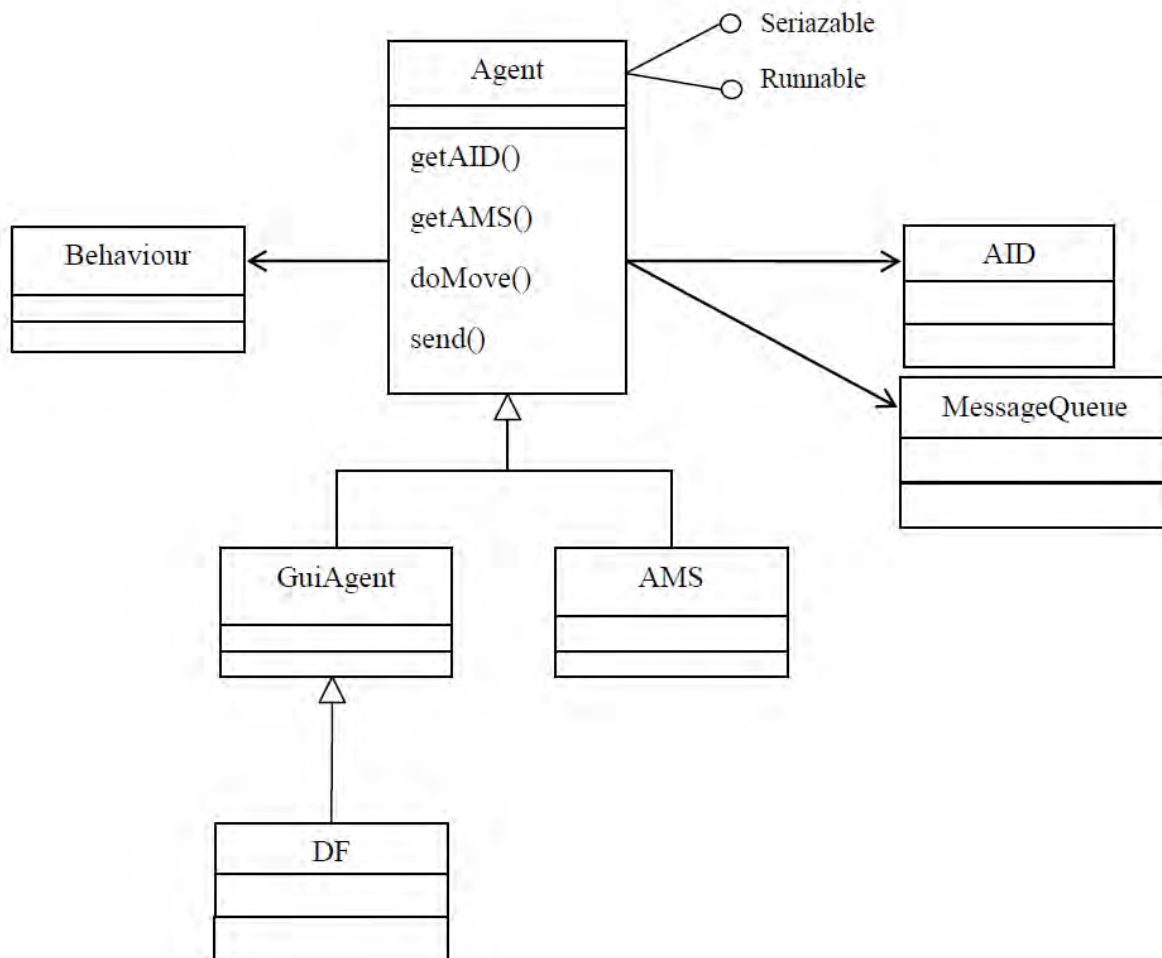
συμπεριφορές που εκτελούνται σύμφωνα με μία πεπερασμένη μηχανή καταστάσεων που έχει προκαθοριστεί από τον χρήστη.

### 3.4 Η κλάση Agent

Η κλάση Agent που χρησιμοποιείται κατά κόρον στην εφαρμογή μας περιγράφει τη λειτουργικότητα ενός πράκτορα. Όταν λοιπόν δημιουργείται ένα αντικείμενο της κλάσης αυτής μας δίνεται η δυνατότητα να:

- Περιγράψουμε έναν πράκτορα (Agent()).
- Ορίσουμε, βρούμε τη διεύθυνσή του (setAdress() / getAddress()).
- Ορίσουμε, βρούμε το όνομά του (setName() / getName()).
- Ορίσουμε, βρούμε το πρωτόκολλο επικοινωνίας με άλλους agents (setProtocol() / getProtocol()).

Πιο συγκεκριμένα οι πράκτορες JADE έχουν την ακόλουθη δομή:



**Εικόνα 3-4.** Δομή της κλάσης Agent

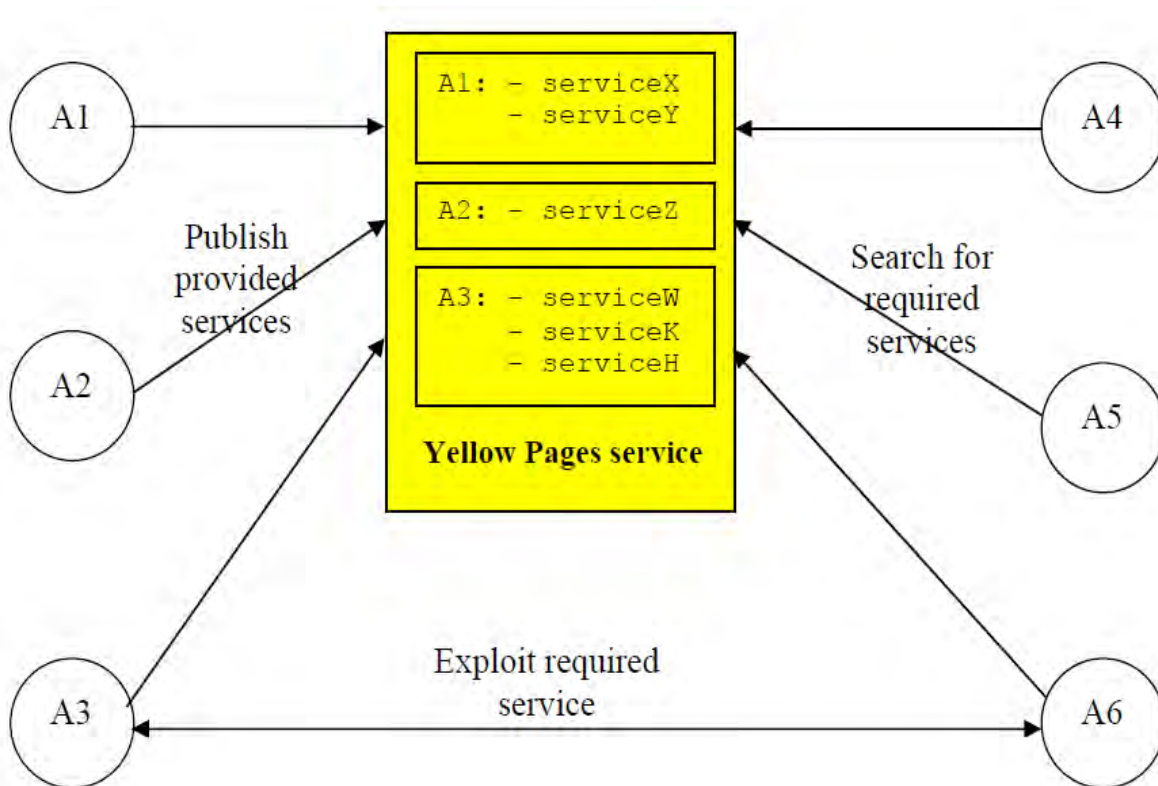
Όπως περιγράψαμε και προηγουμένως, κάθε υπηρεσία και λειτουργία του πράκτορα υλοποιείται ως συμπεριφορά (class Behavior). Ένας scheduler, ο οποίος δεν είναι φανερός στον προγραμματιστή, διαχειρίζεται και εκτελεί τις behaviors διαδοχικά.

Ένας πράκτορας μόλις εισέρχεται στην πλατφόρμα εκτέλεσης, λαμβάνει μια μοναδική ταυτότητα (AID) και με χρήση αυτής της ταυτότητας δέχεται και στέλνει μηνύματα κατά μήκος αυτής.

Ο AMS (**A**gent **M**anagement **S**ystem) παρέχει την “υπηρεσία ονόματος”, naming service, που διαβεβαιώνει ότι κάθε πράκτορας στην πλατφόρμα έχει μοναδικό όνομα και ασκεί εξουσία

στο σύστημα. Η εξουσία αυτή συνοψίζεται στο γεγονός ότι μπορεί να δημιουργεί και να τερματίζει τη λειτουργία πρακτόρων.

Ο DF (Directory Facilitator) παρέχει την υπηρεσία “κίτρινων σελίδων”, yellow pages, η οποία επιτρέπει στους πράκτορες να δημοσιεύουν περιγραφές μίας ή πολλών υπηρεσιών που παρέχουν, έτσι ώστε άλλοι πράκτορες να μπορούν να τις ανακαλύψουν και αν τις χρησιμοποιήσουν.



**Εικόνα 3-5.** Η υπηρεσία των “Κίτρινων Σελίδων”

Οποιοσδήποτε πράκτορας μπορεί να δημοσιεύει υπηρεσίες και να ψάξει για υπηρεσίες. Όλα αυτά μπορούν να πραγματοποιηθούν οποιαδήποτε στιγμή κατά τη διάρκεια ζωής ενός πράκτορα.

### 3.5 JADE Οντολογίες

Όταν ένας πράκτορας A επικοινωνεί με έναν πράκτορα B, μία ορισμένη ποσότητα πληροφορίας I μεταφέρεται από τον A στον B μέσω ενός ACL μηνύματος. Μέσα σε αυτό το μήνυμα, η πληροφορία I αναπαριστάται σαν μία έκφραση περιεχομένου σύμφωνα με μία αρμόζουσα γλώσσα επικοινωνίας (π.χ SL) και κωδικοποιείται σε μία κατάλληλη μορφή (π.χ String). Και οι δύο αυτοί πράκτορες έχουν το δικό τους (ενδεχομένως διαφορετικό) τρόπο εσωτερικής αναπαράστασης της πληροφορίας αυτής. Λαμβάνοντας υπόψη ότι ο τρόπος με τον οποίο ένας πράκτορας αναπαριστά ένα κομμάτι πληροφορίας πρέπει να επιτρέπει τον εύκολο χειρισμό αυτού του κομματιού, είναι σαφές ότι η αναπαράσταση που χρησιμοποιείται σαν μια έκφραση περιεχομένου ACL δεν είναι κατάλληλη για το εσωτερικό του πράκτορα.

Για παράδειγμα η πληροφορία *υπάρχει ένας άνθρωπος που τον λένε Νίκο και είναι 23 χρονών* σε μια ACL έκφραση περιεχομένου θα μπορούσε να αναπαρασταθεί ως ένα String { Άνθρωπος :Όνομα Νίκος :ηλικία 23}.

Αποθηκεύοντας αυτή τη πληροφορία σαν ένα string στο εσωτερικό ενός πράκτορα δεν είναι ο πιο κατάλληλος τρόπος για τον χειρισμό της πληροφορίας αυτής αφού για παράδειγμα για να πάρουμε την ηλικία του Νίκου θα χρειαζόταν κάθε φορά να αναλύσουμε το string.

Λαμβάνοντας υπόψη ότι οι πράκτορες λογισμικού είναι είναι γραμμένοι σε JAVA (όπως είναι και οι πράκτορες JADE), οι πληροφορίες μπορούν εύκολα να αναπαρίστανται ως αντικείμενα στο εσωτερικό ενός πράκτορα. Για παράδειγμα η παραπάνω πληροφορία θα μπορούσε να αναπαρασταθεί ως ένα αντικείμενο μιας συγκεκριμένης κλάσης Άνθρωπος

```
class Person {
    String name;
    int age;

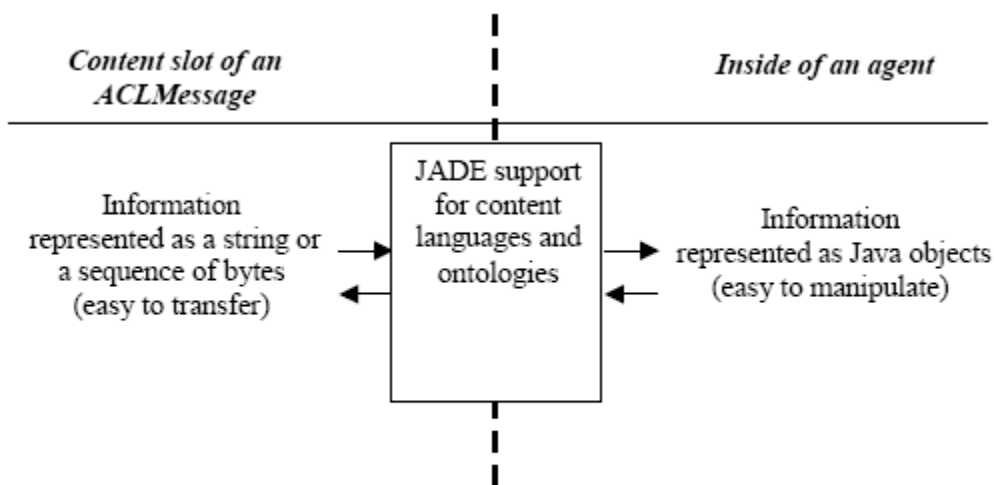
    public String getName() { return name; }
    public int getAge() { return age; }
    public void setName(String n) { name = n; }
    public void setAge(int a) { age = a; }
}
```

αρχικοποιώντας το αντικείμενο με name = "Νίκος" και age "23", κάνοντας έτσι πολύ πιο εύκολο τον χειρισμό της πληροφορίας αυτής.

Είναι σαφές ωστόσο ότι, αν και ο χειρισμός της πληροφορίας είναι πια εύκολος, κάθε φορά που ο πράκτορας A στέλνει ένα κομμάτι πληροφορίας I στον πράκτορα B,

- 1) Ο A πρέπει να μετατρέψει την εσωτερική του αναπαράσταση της πληροφορίας I στην αντίστοιχη έκφραση περιχομένου ACL, και ο B πρέπει να εκτελέσει την αντίστροφη μετατροπή.
- 2) Επιπλέον ο B θα πρέπει να εκτελέσει μια σειρά από σημασιολογικούς ελέγχους για να εξακριβώσει ότι το I είναι ένα κομμάτι πληροφορίας που έχει νόημα (για παράδειγμα ότι η ηλικία του Νίκου είναι ένας ακέραιος αριθμός), δηλαδή συμμορφώνεται με τους κανόνες της οντολογίας μέσω της οποίας και ο A και ο B, αποδίδουν ένα κατάλληλο νόημα στο I.

Η στήριξη για τις γλώσσες περιεχομένου και τις οντολογίες που παρέχει το JADE, είναι σχεδιασμένη έτσι ώστε να εκτελεί τις παραπάνω μετατροπές και τον έλεγχο των εργασιών αυτόματα όπως απεικονίζεται στην Εικόνα χ επιτρέποντας έτσι στους προγραμματιστές να χειρίζονται την πληροφορία των πρακτόρων τους ως JAVA αντικείμενα χωρίς την ανάγκη οποιασδήποτε επιπλέον εργασίας.



*(The conversion performed by the JADE support for content languages and ontologies)*

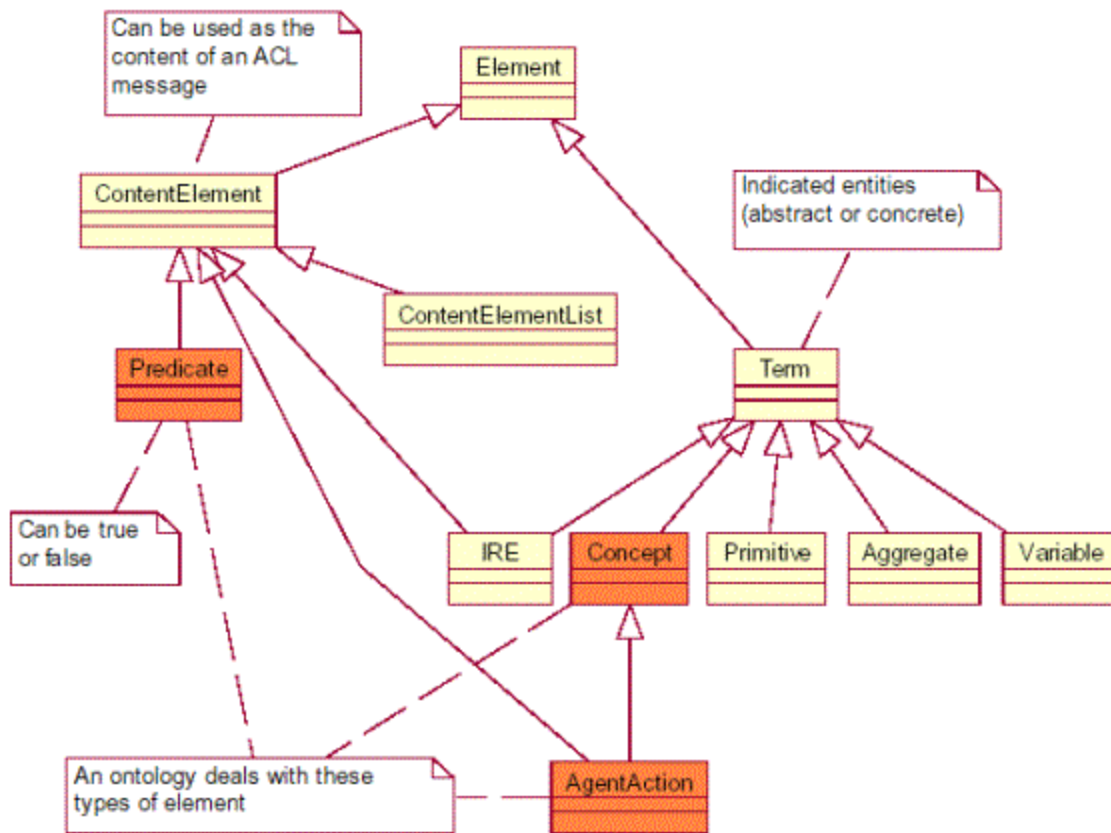
Προκειμένου το JADE να εκτελέσει τους κατάλληλους σημασιολογικούς ελέγχους σε μία συγκεκριμένη έκφραση περιεχομένου είναι αναγκαία η ταξινόμηση όλων των πιθανών στοιχείων που μπορούν να εμφανιστούν σε μία έγκυρη πρόταση που αποστέλλεται από έναν πράκτορα ως το περιεχόμενο ενός ACL μηνύματος, σε ένα γενικό μοντέλο αναφοράς

περιεχομένου. Αυτή η ταξινόμηση προέρχεται από την γλώσσα ACL που ορίζει η FIPA και μερικά από τα πιο σημαντικά και συνηθισμένα στοιχεία είναι τα εξής:

1. Κατηγορήματα (Predicates): Είναι εκφράσεις που λένε κάτι για την κατάσταση του κόσμου και μπορεί να είναι αληθείς ή ψευδείς.  
Παράδειγμα (Δουλεύει για (Άτομο όνομα: Νίκος) (Εταιρία όνομα: XXX)), δηλώνοντας ότι “το άτομο Νίκος δουλεύει για την εταιρία XXX”.  
Τα κατηγορήματα μπορούν να χρησιμοποιηθούν σαν το περιεχόμενο ενός μηνύματος (**Query-If**) ή (**Inform**), ενώ δε θα είχε νόημα να τα χρησιμοποιήσουμε σαν το περιεχόμενο ενός μηνύματος (**Request**).
2. Έννοιες (Concepts) : Είναι εκφράσεις που υποδεικνύουν οντότητες με μία σύνθετη δομή που μπορεί να οριστεί με όρους.  
Παράδειγμα (Άτομο όνομα: Νίκος ηλικία:23)
3. Ενέργειες πράκτορα (Agent actions): Είναι ειδικές έννοιες που μπορούν να εκτελεστούν από ορισμένους πράκτορες.  
Παράδειγμα (Πουλάω(Βιβλίο τίτλος: Harry Potter) (Άτομο όνομα: Γιώργος)).  
Αυτού του είδους τα στοιχεία έχουν νόημα σε ορισμένους τύπους ACL μηνυμάτων όπως μηνύματα (**Request**) .

Παρακάτω, βλέπουμε το μοντέλο αναφοράς περιεχομένου του JADE καθώς και η ταξινόμηση όλων των στοιχείων.





**Εικόνα 3-6.** Μοντέλο Αναφοράς Περιεχομένου του JADE.

## 3.6 Η κλάση GuiAgent

Προκειμένου οι χρήστες της εφαρμογής να χειρίζονται τους πράκτορες και τις λειτουργίες του με περισσότερη άνεση ενσωματώσαμε ένα γραφικό περιβάλλον για τον πράκτορα-χρήστη.

Στη γλώσσα προγραμματισμού JAVA, μία γραφική διεπαφή χρήστη (GUI) τρέχει με το δικό της νήμα, το οποίο της επιτρέπει να χειρίζεται και αντιδρά άμεσα στα γεγονότα που δημιουργούνται κάθε φορά που ο χρήστης αλληλεπιδρά με το GUI μέσω ενός στοιχείου, όπως το πάτημα ενός κουμπιού ή η αλλαγή μεγέθους του παραθύρου. Από την άλλη μεριά, ένα πρόγραμμα πράκτορα εκτελείται και αυτό με δικό του νήμα εκτέλεσης το οποίο επιτρέπει στον πράκτορα να χειριστεί τις συμπεριφορές του. Επειδή δεν είναι αποτελεσματικό ένα νήμα να καλεί άμεσα τις μεθόδους ενός άλλου νήματος, το JADE προσφέρει έναν κατάλληλο μηχανισμό για τη διαχείριση των αλληλεπιδράσεων μεταξύ δύο νημάτων κατά την ενσωμάτωση ενός GUI με έναν πράκτορα.

Ο μηχανισμός αυτός βασίζεται απλά στο πέρασμα γεγονότων. Αντί να περιγράψουμε αναλυτικά τις μεθόδους που καλούνται κατά την αλληλεπίδραση ενός gui με έναν πράκτορα και αντίστροφα, θα πούμε μόνο ότι η κλάση GuiAgent παρέχει δύο συγκεκριμένες μεθόδους, τις **postGuiEvent()** και **onGuiEvent()** με τις οποίες επιτυγχάνεται αυτή η αλληλεπίδραση.

Για περισσότερες λεπτομέρειες μπορεί κανείς να δει το API του JADE.

## 4 Σχεδίαση συστήματος

Στο συγκεκριμένο κεφάλαιο περιγράφεται ο σχεδιασμός, οι λειτουργίες και η αλληλεπιδραστικότητα που υπάρχει μεταξύ των διάφορων πρακτόρων του συστήματος.

### 4.1 Πράκτορες του συστήματος

Αυτή η ενότητα περιγράφει τους πράκτορες που αναπτύχθηκαν για τις ανάγκες της εφαρμογής. Συνολικά υπάρχουν 4 πράκτορες. Παρόλο που μπορούμε να έχουμε περισσότερους από έναν πράκτορες-χρήστη, μπορούμε να έχουμε μόνο έναν πράκτορα-εξυπηρετητή και έναν πράκτορα-διευκόλυνσης σε κάθε χρονική στιγμή. Αυτός ο περιορισμός επεβλήθη στο σύστημα προκειμένου να μειώσει την πολυπλοκότητα της εφαρμογής. Το κύριο κομμάτι του συστήματος είναι οι λειτουργίες που πραγματοποιεί ο πράκτορας-διευκόλυνσης προκειμένου να συστήνει ταινίες στον χρήστη και στη συνέχεια αφού δεχτεί τις προτιμήσεις του κάθε χρήστη σχετικά με την ώρα και τη μέρα για μια συγκεκριμένη ταινία να αναπαράγει τις κοινές συμφωνίες μεταξύ των χρηστών.

#### 4.1.1 UserAgent

Ο userAgent είναι σχεδιασμένος ούτως ώστε να εκτελεί τις λειτουργίες που φαίνονται στον παρακάτω πίνακα.

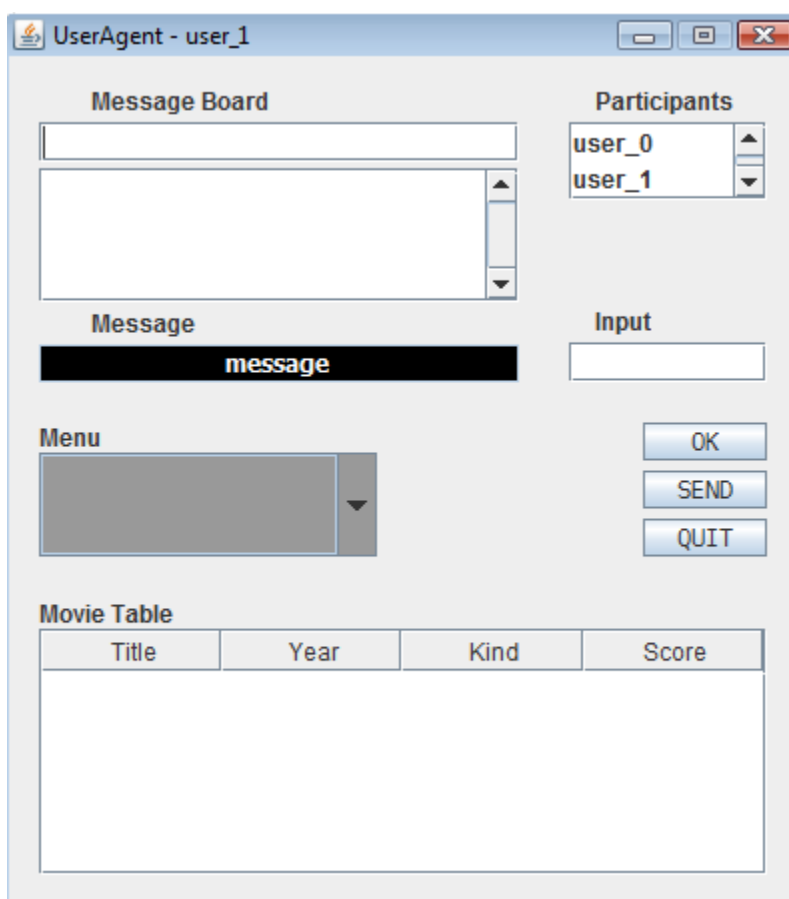
Τύπος Πράκτορα	Λειτουργίες
UserAgent	Εγγράφεται αρχικά στον Directory Facilitator ώστε να μπορούν να τον εντοπίσουν άλλοι χρήστες του συστήματος. Στέλνει τις βαθμολογίες που έχει εισάγει ο χρήστης για διάφορες ταινίες, στον FacilitatorAgent. Αποθηκεύει τις βαθμολογίες σε μια τοπιή βάση δεδομένων. Αποθηκεύει τις βαθμολογίες στο cloud. Στέλνει τον τίτλο της ταινίας για την οποία ο χρήστης ψάχνει παρόμοιες, στον FacilitatorAgent.

	<p>Ενημερώνει τον χρήστη σε περίπτωση που του έρθει μήνυμα από άλλον χρήστη του συστήματος.</p> <p>Ενημερώνει τον χρήστη σχετικά με τα αποτελέσματα που παίρνει είτε από τον FacilitatorAgent, είτε από τον ServerAgent για μία συγκεκριμένη λειτουργία.</p> <p>Επιλέγει από το διαθέσιμο πρόγραμμα ταινιών που προβάλλονται, τις προτιμήσεις του όσον αφορά ώρα και μέρα για μια συγκεκριμένη ταινία.</p>
--	--

**Πίνακας 4-1:** Λειτουργίες του πράκτορα UserAgent

Ο UserAgent χρησιμοποιεί μια κυκλική συμπεριφορά (CyclicBehaviour), για τη λήψη των αποτελεσμάτων από τον FacilitatorAgent και τον ServerAgent.

Η αλληλεπίδραση του χρήστη με τον UserAgent γίνεται μέσω ενός γραφικού περιβάλλοντος χρήστη (GUI) που επιτρέπει στον χρήστη να εισάγει εύκολα δεδομένα με τις προτιμήσεις του, να στέλνει μηνύματα σε άλλους χρήστες του συστήματος και να επιλέγει την λειτουργία που θέλει να πραγματοποιηθεί κάθε φορά. Απεικονίζεται στην ακόλουθη εικόνα.

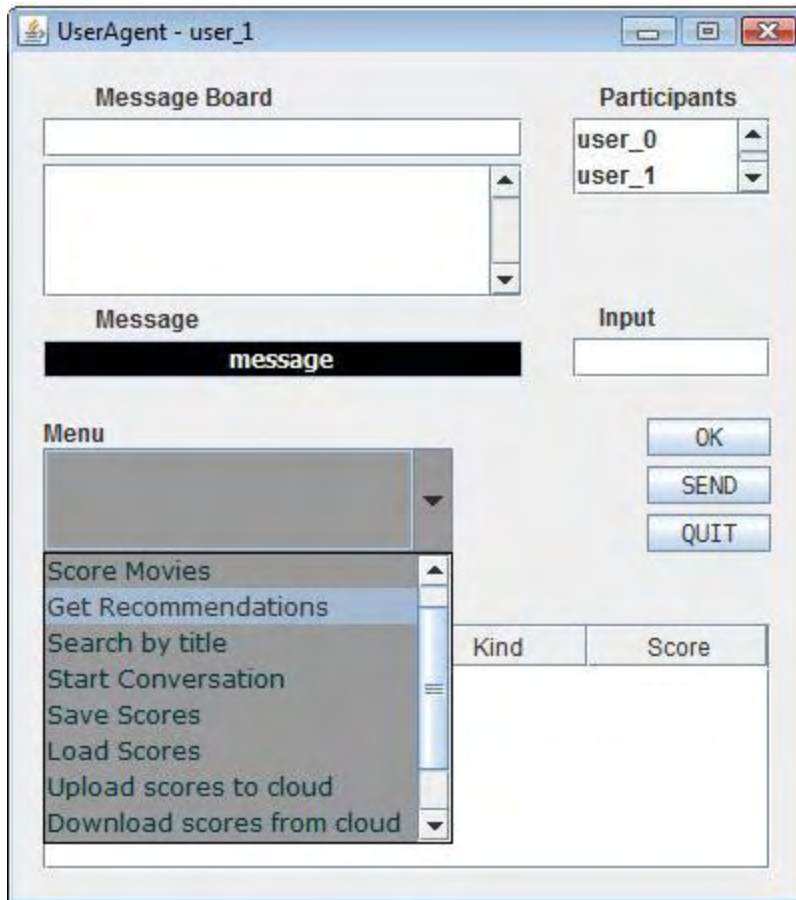


**Εικόνα 4-1.** Γραφικό περιβάλλον χρήστη

Όπως βλέπουμε, το GUI αποτελείται από:

- 1) Ένα μενού με τις επιλογές του χρήστη.
- 2) Έναν πίνακα που παρουσιάζονται οι ταινίες.
- 3) Μία είσοδο προκειμένου να εισάγει ο χρήστης τον τίτλο της ταινίας για την οποία επιθυμεί να πάρει αποτελέσματα παρόμοιων ταινιών.
- 4) Μία περιοχή όπου ο χρήστης βλέπει εισερχόμενα μηνύματα από άλλους χρήστες, καθώς και ένα πεδίο όπου μπορεί να πληκτρολογήσει την απάντηση του.
- 5) Και τέλος μία λίστα όπου ο χρήστης μπορεί να δει ποιοι είναι στο σύστημα παρών αυτή τη στιγμή και να επιλέξει τον χρήστη στον οποίο θα στείλει το μήνυμά του.

Παρακάτω φαίνονται και οι επιλογές του χρήστη οι οποίες θα αναλυθούν λεπτομερώς στο επόμενο κεφάλαιο.



Εικόνα 4-2. Οι επιλογές του χρήστη.

### 4.1.2 FacilitatorAgent

Ο FacilitatorAgent είναι αυτός που αφού λάβει δεδομένα εισόδου από τον UserAgent, τα επεξεργάζεται για να παράγει τα κατάλληλα αποτελέσματα. Οι λειτουργίες του αναγράφονται στον παρακάτω πίνακα.

Τύπος Πράκτορα	Λειτουργίες
FacilitatorAgent	<p>Εγγράφεται αρχικά στον Directory Facilitator ώστε να μπορούν να τον εντοπίσουν οι UserAgents.</p> <p>Συλλέγει τις προτιμήσεις όλων των χρηστών του συστήματος και τις αποθηκεύει σε μια δομή (HashMap).</p> <p>Μετά από αίτημα του χρήστη, στέλνει σε αυτόν αποτελέσματα με ταινίες τις οποίες συστήνει στον χρήστη.</p> <p>Μετά από αίτημα του χρήστη και ανάλογα με τον συγκεκριμένο τίτλο ταινίας που έχει λάβει από τον τελευταίο, στέλνει πίσω παρόμοιες ταινίες.</p> <p>Μετά από αίτημα του χρήστη εμφανίζει τα γκρουπ των χρηστών που έχουν διαμορφωθεί ανάλογα με τις προτιμήσεις τους.</p>

**Πίνακας 4-2.** Λειτουργίες του πράκτορα FacilitatorAgent

Ο FacilitatorAgent δέχεται αιτήματα από τον UserAgent μέσω μιας κυκλικής συμπεριφοράς (CyclicBehaviour).

### 4.1.3 ServerAgent

Ο ServerAgent είναι αυτός που έχει αποθηκευμένη τη βάση δεδομένων με τις ταινίες. Ο πίνακας λειτουργιών του είναι ο εξής:

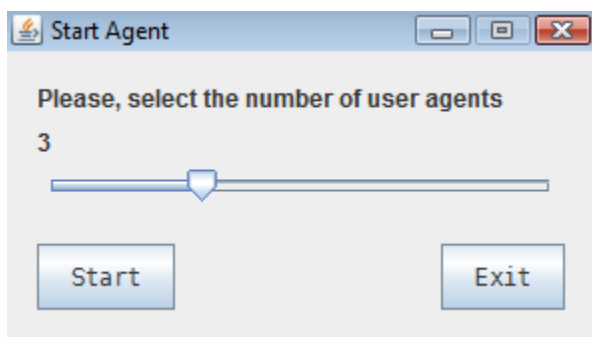
Τύπος πράκτορα	Λειτουργίες
ServerAgent	<p>Εγγράφεται αρχικά στον Directory Facilitator ώστε να μπορούν να τον εντοπίσουν οι UserAgents.</p> <p>Επιστρέφει μετά από αίτημα του χρήστη τις ταινίες που υπάρχουν στη βάση δεδομένων.</p>

**Πίνακας 4-3.** Λειτουργίες του πράκτορα ServerAgent

Όπως και ο FacilitatorAgent, έτσι και ο ServerAgent δέχεται αιτήματα από τον UserAgent μέσω μιας κυκλικής συμπεριφοράς (CyclicBehaviour).

### 4.1.4 StartAgent

Για τις ανάγκες της παρούσας διπλωματικής αναπτύξαμε τον StartAgent, ο οποίος είναι υπεύθυνος μόνο για την εκκίνηση ενός συγκεκριμένου αριθμού από UserAgents. Αφού το κάνει αυτό, στη συνέχεια τερματίζει. Φροντίζει επίσης ώστε τα ονόματα των UserAgent να είναι μοναδικά για να μην παρουσιαστούν τυχόν προβλήματα επικαλύψεων. Ο πράκτορας αυτός υλοποιήθηκε για να μην χρειάζεται κάθε φορά να κάνουμε χειροκίνητη εκκίνηση ενός UserAgent. Συνδυάζεται με ένα GUI το οποίο επιτρέπει στον χρήστη της εφαρμογής να εκκινήσει το σύστημα με τον αριθμό UserAgent πρακτόρων που αυτός επιθυμεί. Ο χρήστης μπορεί να εκκινήσει από έναν μέχρι 10 UserAgents.



Εικόνα 4-3. Γραφικό περιβάλλον του StartAgent

## 4.2 Οντολογία του συστήματος

Όταν οι πράκτορες της εφαρμογής αλληλεπιδρούν μεταξύ τους, ανταλλάσσουν πληροφορία που αναφέρεται σε οντότητες. Αυτές οι οντότητες μπορεί να είναι θεμελιώδη στοιχεία, όπως ένα αλφαριθμητικό ή ένας ακέραιος, ή μπορεί να έχουν πιο περίπλοκη δομή η οποία ορίζεται από πολλά πεδία, οι τιμές των οποίων είναι ενός συγκεκριμένου τύπου. Οι περίπλοκες δομές αυτές ονομάζονται Έννοιες (*Concepts*) και είναι ξεχωριστές κλάσεις. Στη δική μας εφαρμογή για παράδειγμα έχουμε την έννοια *Movie*.

Movie:
- title (String)



<ul style="list-style-type: none"> <li>- year (Int)</li> <li>- kind (String)</li> </ul>
---

**Εικόνα 4-4.** Concept Movie

Επιπλέον οι οντότητες αυτές σχετίζονται πολλές φορές με εκφράσεις που μπορεί να είναι αληθείς ή ψευδείς. Οι εκφράσεις αυτές είναι γνωστές ως *Κατηγορήματα (Predicates)*. Στη δική μας εφαρμογή δεν χρειάστηκε να χρησιμοποιήσουμε κατηγορήματα.

Τέλος, ένα άλλο είδος πολύπλοκης οντότητας αναπαριστάται από ενέργειες τις οποίες μπορεί να εκτελέσει ένας πράκτορας (AgentActions). Παρακάτω βλέπουμε τις ενέργειες που εκτελούν οι πράκτορες της εφαρμογής μας.

CreatePreferences:
<ul style="list-style-type: none"> <li>- name (String)</li> <li>- movieList (String)</li> </ul>

**Εικόνα 4-5.** AgentAction CreatePreferences

Information:
<ul style="list-style-type: none"> <li>- type (Int)</li> <li>- msg (String)</li> </ul>

**Εικόνα 4-6.** AgentAction Information

Η οντολογία του συστήματος λοιπόν ορίζεται ως το σύνολο των παραπάνω εννοιών και ενεργειών πράκτορα.

### 4.3 Επιλογή Γλώσσας Περιεχομένου

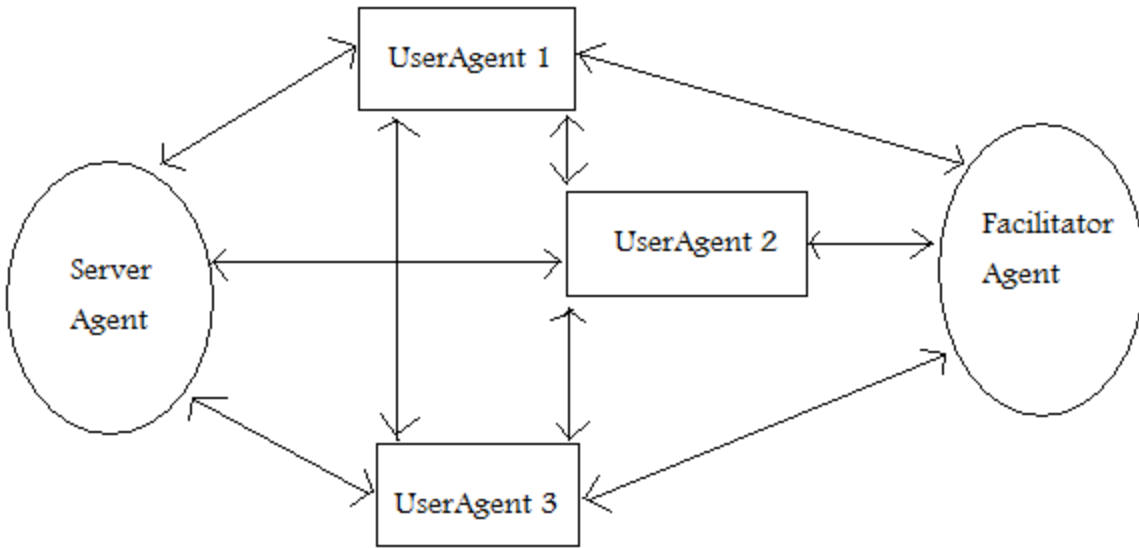
Το JADE παρέχει δύο κωδικοποιητές για 2 διαφορετικές γλώσσες περιεχομένου: Την γλώσσα SL και την γλώσσα LEAP (μέσω του πακέτου `jade.content`). Επιπλέον, ένας κωδικοποιητής μπορεί να οριστεί και από έναν προγραμματιστή αν αυτός επιθυμεί οι πράκτορες να “μιλάνε” μία διαφορετική γλώσσα περιεχομένου. Η γλώσσα SL είναι αναγνώσιμη από έναν άνθρωπο επειδή είναι κωδικοποιημένη σε αλφαριθμητικά, ενώ η LEAP δεν είναι αφού είναι κωδικοποιημένη σε μία σειρά από bytes. Μερικά κριτήρια για την επιλογή γλώσσας περιεχομένου είναι τα παρακάτω:

- Η SL είναι κατάλληλη γλώσσα για πολυπρακτορικές εφαρμογές που είναι (ή μπορεί να γίνουν) ανοικτές (δηλαδή πράκτορες από διαφορετικούς κατασκευαστές που τρέχουν σε διαφορετικές πλατφόρμες να μπορούν να επικοινωνήσουν).
- Η κλάση `LEAPCodec` είναι “ελαφρύτερη” από την κλάση `SLCodec`. Έτσι, όταν υπάρχουν περιορισμοί μνήμης, η γλώσσα LEAP είναι προτιμότερη. Για παράδειγμα, για εφαρμογές που στρίζονται σε κινητές συσκευές, όπως τα κινητά τηλέφωνα, η γλώσσα LEAP είναι αρκετά προτιμότερη. Αντίθετα, για εφαρμογές όπου εμπλέκονται υπολογιστές υψηλής χωριτικότητας και ταχύτητας, προτιμάται η γλώσσα SL.

Στη δική μας περίπτωση χρησιμοποιήσαμε τη γλώσσα SL, καθώς είναι πιο απλή προς τον χρήστη και δεν εμπλέκονταν κινητές συσκευές στην υλοποίηση.

### 4.4 Αλληλεπιδράσεις μεταξύ των πρακτόρων της εφαρμογής

Όπως γίνεται αντιληπτό και από τις λειτουργίες του κάθε πράκτορα προκύπτει το ακόλουθο σχήμα που δείχνει για κάθε πράκτορα, τους άλλους πράκτορες με τους οποίους μπορεί να ανταλλάξει μηνύματα.



**Διάγραμμα 4-1.** Διάγραμμα αλληλεπίδρασης πρακτόρων.

# 5 Υλοποίηση

Στο συγκεκριμένο κεφάλαιο θα αναλύσουμε την τεχνική του συλλογικού φιλτραρίσματος (collaborative filtering) την οποία χρησιμοποιήσαμε για την υλοποίηση της εφαρμογής μας. Θα δούμε επίσης αλγόριθμους συγκρίσεων οντοτήτων για εξακρίβωση ομοιότητας μεταξύ αυτών. Τέλος θα περιγράψουμε τον τρόπο με τον οποίο ο FacilitatorAgent ομαδοποιεί τις προτιμήσεις των χρηστών και εμφανίζει τα κατάλληλα αποτελέσματα.

## 5.1 Συλλογικό Φιλτράρισμα (Collaborative Filtering)

Ο όρος collaborative filtering χρησιμοποιήθηκε για πρώτη φορά από τον David Goldberg σε μία δημοσίευσή του με όνομα “Using collaborative filtering to weave an information tapestry”. Σχεδίασε ένα σύστημα με όνομα Tapestry το οποίο επέτρεπε στους ανθρώπους να σχολιάζουν έγγραφα ως ενδιαφέρον ή μη-ενδιαφέρον και χρησιμοποιούσε αυτή τη πληροφορία για να φιλτράρει έγγραφα για άλλους ανθρώπους.

Σήμερα, ο όρος collaborative filtering αναφέρεται στην μέθοδο με την οποία γίνονται αυτόματες προβλέψεις σχετικά με τα ενδιαφέροντα ενός ατόμου, συλλέγοντας πληροφορίες για τις προτιμήσεις ή το γούστο πολλών ατόμων (collaborative). Η βαθύτερη ιδέα της προσέγγισης του συλλογικού φιλτραρίσματος είναι η υπόθεση ότι αν ο A έχει την ίδια γνώμη με τον B πάνω σε ένα θέμα  $x$ , τότε ο A είναι πιο πιθανό να έχει την γνώμη του B πάνω σε ένα άλλο θέμα  $y$ , παρά να έχει την γνώμη κάποιου άλλου ατόμου που έχει επιλεχθεί τυχαία από ένα σύνολο.

Υπάρχουν εκατοντάδες ιστοσελίδες σήμερα που χρησιμοποιούν αλγορίθμους του συλλογικού φιλτραρίσματος για ταινίες, μουσική, βιβλία, ραντεβού, ψώνια, άλλες ιστοσελίδες, άρθρα, ακόμα και για ανέκδοτα.

## 5.2 Συλλογή των προτιμήσεων

Το πρώτο πράγμα που χρειαζόμαστε είναι ένας τρόπος να αναπαραστήσουμε τους διαφορετικούς χρήστες και τις προτιμήσεις τους. Στη Java, αυτό γίνεται με την κλάση *Map*. Κατά την ανάλυση των επιλογών του χρήστη θα χρησιμοποιήσουμε καθόλη τη διάρκεια του κεφαλαίου ένα παράδειγμα για επεξηγηματικούς σκοπούς, καθώς και για να πειραματιστούμε με τους αλγόριθμους που θα παρουσιάσουμε. Έτσι, θα γίνει και πιο εύκολο στον αναγνώστη να καταλάβει τα θεμελιώδη στοιχεία της εφαρμογής.

Ας υποθέσουμε λοιπόν ότι έχουμε τους χρήστες, τις ταινίες, και τις προτιμήσεις των χρηστών αυτών που φαίνονται στον παρακάτω πίνακα.

	Django	Life of Pi	Lincoln	The Prestige	Inception	Looper
User_0	2.5	3.5	3.0	3.5	2.5	3.0
User_1	3.0	3.5	1.5	5.0	3.5	3.0
User_2	2.5	3.0		3.5		4.0
User_3		3.5	3.0	4.0	2.5	4.5
User_4	3.0	4.0	2.0	3.0	2.0	3.0
User_5	3.0	4.0		5.0	3.5	3.0
User_6		4.5		4.0	1.0	

**Πίνακας 5-1.** Πίνακας χρηστών-ταινιών.

Η υλοποίησή μας χρησιμοποιεί σύστημα αξιολογήσεων από το 1 μέχρι το 5. Έτσι, αν κάποιος αξιολογήσει μία ταινία με 5 σημαίνει ότι την βρήκε εξαιρετική, ενώ αν τη βαθμολογήσει με 1 σημαίνει ότι δεν του άρεσε καθόλου.

## 5.3 Αποθήκευση των βαθμολογιών σε τοπική βάση δεδομένων

Σε περίπτωση αναπάντεχου τερματισμού ενός πράκτορα *UserAgent* του συστήματος, θα χανόντουσαν οι προτιμήσεις του χρήστη που εκπροσωπείται από τον συγκεκριμένο πράκτορα

και θα έπρεπε να τις εισάγει από την αρχή προκειμένου να πάρει συστάσεις από το σύστημα. Για να αποφύγουμε κάτι τέτοιο, οι χρήστες μπορούν να αποθηκεύσουν τις προτιμήσεις τους σε μία τοπική βάση δεδομένων και να ανακτήσουν την πληροφορία όποτε συνδεθούν και πάλι με το σύστημα.

## 5.4 Αποθήκευση των βαθμολογιών στο cloud.

Τα δεδομένα από την αποθήκευση των προτιμήσεων ενός χρήστη σε μία τοπική βάση δεδομένων είναι ορατά μόνο από τον ίδιο τον χρήστη. Κάποιος μπορεί να θέλει να γνωστοποιήσει τις προτιμήσεις του σε άλλους χρήστες του συστήματος και εκτός από αυτό μπορεί να μη θέλει να "σπαταλάει" χώρο από τον σκληρό του δίσκο. Για το λόγο αυτό λοιπόν είναι εφικτή η αποθήκευση και στο cloud μέσω της υπηρεσίας AmazonS3.

### 5.4.1 Η υπηρεσία AmazonS3



**Εικόνα 5-1.** Η υπηρεσία AmazonS3.

Το AmazonS3 (Simple Storage Service) είναι μια διαδικτυακή υπηρεσία αποθήκευσης δεδομένων που προσφέρεται από το Amazon Web Services. Ξεκίνησε για πρώτη φορά δημοσίως τον Μάρτιο του 2006 στις Ηνωμένες Πολιτείες και τον Νοέμβριο του 2007 στην

Ευρώπη. Κατά την ίδρυσή της, η Amazon χρέωνε 0,15 δολάρια ανά GigaByte το μήνα με επιπλέον χρεώσεις ανάλογα με το bandwidth που χρησιμοποιείται για την αποστολή και λήψη δεδομένων. Η Amazon ισχυρίζεται ότι το AmazonS3 χρησιμοποιεί την ίδια κλιμακωτή δομή αποθήκευσης που χρησιμοποιεί το [Amazon.com](http://Amazon.com) τη λειτουργία του δικού του παγκόσμιου εμπορικού δικτύου. Τον Ιούνιο του 2012 το AmazonS3 είχε αποθηκευμένα πάνω από ένα τρισεκατομμύριο αντικείμενα ενώ τον Μάρτιο του 2010 είχε 102 δισεκατομμύρια.

Το S3 αποθηκεύει αυθαίρετα αντικείμενα μέχρι 5 terabytes σε μέγεθος. Τα αντικείμενα αυτά οργανώνονται σε κάδους και σε κάθε κάδο με ένα μοναδικό κλειδί που έχει ανατεθεί από τον χρήστη.

Οι κάδοι και τα αντικείμενα μπορούν να δημιουργηθούν και να ανακτηθούν χρησιμοποιώντας μία REST-style HTTP διεπαφή ή μια διεπαφή SOAP. Επιπλέον, τα αντικείμενα μπορούν να ληφθούν χρησιμοποιώντας HTTP GET διεπαφή.

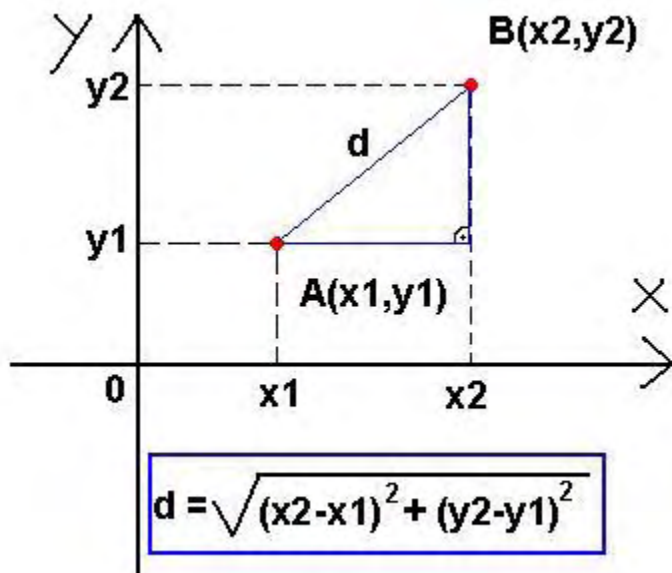
Για την μεταφορά και τη λήψη των δεδομένων στο AmazonS3 χρησιμοποιήσαμε το AWS (Amazon Web Services) SDK για την γλώσσα προγραμματισμού Java, το οποίο παρέχει τις κατάλληλες βιβλιοθήκες για τον χειρισμό των διάφορων υπηρεσιών που προσφέρει το Amazon, ανάμεσά τους και το AmazonS3.

## 5.5 Βρίσκοντας παρόμοιους χρήστες

Αφού συλλέξουμε τα δεδομένα σχετικά με το τι αρέσει στον κάθε χρήστη χρειαζόμαστε μία μέθοδο για να καθορίσουμε πώς σχετίζονται οι χρήστες μεταξύ τους ανάλογα με τις βαθμολογίες τους. Αυτό το κάνουμε συγκρίνοντας κάθε χρήστη με όλους τους υπόλοιπους, και υπολογίζοντας έναν *βαθμό ομοιότητας*. Για να υπολογιστεί αυτός ο βαθμός υπάρχουν αρκετοί τρόποι, 2 από τους οποίους παρουσιάζονται στη συνέχεια.

### 5.5.1 Βαθμός με βάση την Ευκλείδια Απόσταση

Ένας πολύ απλός αλλά αποτελεσματικός τρόπος να υπολογίσουμε τον βαθμό ομοιότητας είναι να χρησιμοποιήσουμε τον αλγόριθμο της Ευκλείδιας Απόστασης, ο οποίος παίρνει τα κοινά αντικείμενα που έχουν βαθμολογήσει οι χρήστες και τα χρησιμοποιεί σαν άξονες ενός διαγράμματος. Στην παρακάτω εικόνα για παράδειγμα οι άξονες  $x$  και  $y$  θα μπορούσαν να αντιπροσωπεύουν τίτλους ταινιών και οι  $A, B$  να αντιπροσωπεύουν 2 διαφορετικούς χρήστες.



**Εικόνα 5-2.** Προτιμήσεις του A και B για τις ταινίες x,y.

Παρατηρούμε ότι ο A έχει βαθμολογήσει την ταινία χ με  $x_1$  και την ταινία γ με  $y_1$ , ενώ ο B έχει βαθμολογήσει την ταινία χ με  $x_2$  και την γ με  $y_2$ . Όσο πιο κοντά βρίσκονται 2 χρήστες στον χώρο τόσο πιο παρόμοιες είναι οι προτιμήσεις τους. Καθώς το διάγραμμα έχει 2 μόνο διαστάσεις, μπορούμε να δούμε μόνο βαθμολογίες για 2 ταινίες κάθε φορά, αλλά η αρχή είναι η ίδια για μεγαλύτερο αριθμό ταινιών.

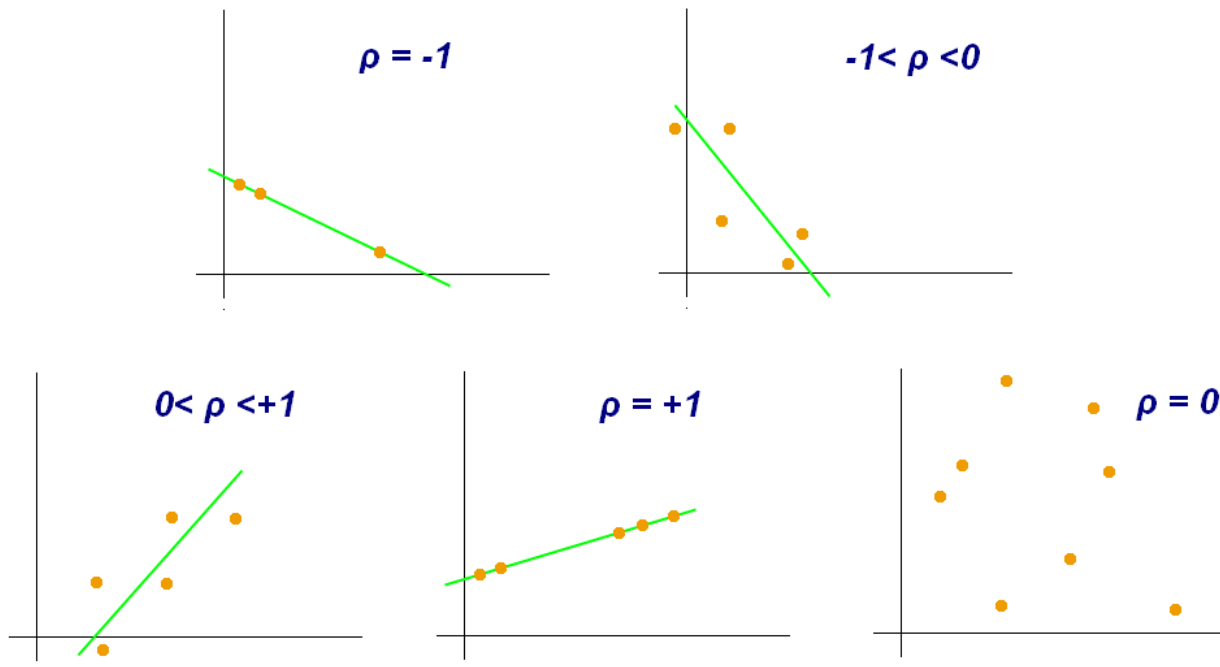
Για να υπολογίσουμε τη διαφορά μεταξύ του A και του B στο διάγραμμα εφαρμόζουμε τον τύπο φαίνεται στην εικόνα 5-1.

### 5.5.2 Βαθμός με βάση τη συσχέτιση κατά Pearson

Ο συντελεστής συσχέτισης κατά Pearson είναι ένας πιο πολύπλοκος και πιο εξελιγμένος τρόπος για να βρεθεί ο βαθμός ομοιότητας και τείνει να δίνει καλύτερα αποτελέσματα από την Ευκλείδεια Απόσταση όταν τα δεδομένα δεν είναι καλά κανονικοποιημένα – για παράδειγμα όταν οι βαθμολογίες των χρηστών είναι δριμύτερες από τον μέσο όρο. Ο αλγόριθμος παράγει μια “best fit” γραμμή μεταξύ των γνωρισμάτων δύο αντικειμένων. Όταν η γραμμή αυτή, η οποία περνάει από τα σημεία που περιγράφουν τα γνωρίσματα των



αντικειμένων, έχει θετική κλίση αυτό σημαίνει ότι τα αντικείμενα μοιάζουν μεταξύ τους. Τώρα, οι άξονες αναπαριστούν 2 διαφορετικούς χρήστες και τα σημεία αποτελούν τις ταινίες τις οποίες έχουν βαθμολογήσει από κοινού οι 2 αυτοί χρήστες. Στην υλοποίησή μας χρησιμοποιούμε αυτόν τον αλγόριθμο.



**Εικόνα 5-3.** Παραδείγματα διαγραμμάτων διασποράς με διαφορετικές τιμές συντελεστών συσχέτισης.

Αν οι χρήστες έχουν πανομοιότυπες βαθμολογίες τότε η γραμμή αυτή θα περνάει από όλα τα σημεία δίνοντας τέλει βαθμό ομοιότητας. Ο βαθμός ομοιότητας καθορίζεται από μια μεταβλητή  $\rho$ , η οποία μπορεί να έχει τιμή από  $-1$  (καθόλου ομοιότητα), μέχρι  $1$  (για κάθε ταινία οι χρήστες έχουν ακριβώς τις ίδιες βαθμολογίες). Η μεταβλητή  $\rho$  βρίσκεται από τον παρακάτω τύπο.

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}}$$

Όπου n είναι ο αριθμός των αντικειμένων, x είναι η τιμή που δίνει ο πρώτος χρήστης σε κάθε διαφορεικό αντικείμενο (στη δική μας περίπτωση τα αντικείμενα αυτά είναι ταινίες), και y είναι η τιμή που δίνει ο δεύτερος χρήστης στο αντικείμενο αυτό.

## 5.6 Σύσταση ταινιών

Για να βρούμε κατάλληλες ταινίες για κάποιον χρήστη (π.χ για τον User\_6) θα μπορούσαμε να βρούμε με ποιον από τους υπόλοιπους χρήστες του συστήματος έχει τον μεγαλύτερο βαθμό ομοιότητας, και να του προτείνουμε ταινίες που έχει δει ο τελευταίος και έχει αξιολογήσει με υψηλό σκορ. Ωστόσο αυτή η προσέγγιση θα μπορούσε να επιστρέψει χρήστες οι οποίοι δεν έχουν αξιολογήσει ταινίες, οι οποίες μπορεί να αρέσουν στον User\_6.

Για να αποφύγουμε τέτοια ζητήματα πρέπει να αξιολογήσουμε τα αντικείμενα παράγοντας ένα βεβαρυμένο σκόρ από τις κριτικές όλων των χρηστών του συστήματος που έχουν θετικό συντελεστή συσχέτισης με τον User\_6 . Στη συνέχεια πολλαπλασιάζουμε αυτούς τους συντελεστές με την βαθμολογία που έχουν δώσει σε κάθε ταινία. Η διαδικασία αυτή φαίνεται στον παρακάτω πίνακα.

Κριτής	Συντελεστής συσχέτισης	Looper	Sim.Looper	Django	Sim.Django	Lincoln	Sim.Lincoln
User_0	0.99	3.0	2.97	2.5	2.48	3.0	2.97
User_1	0.38	3.0	1.14	3.0	1.14	1.5	0.57
User_3	0.89	4.5	4.02			3.0	2.68
User_4	0.92	3.0	2.77	3.0	2.77	2.0	1.85
User_5	0.66	3.0	1.99	3.0	1.99		
Σύνολο			12.89		8.38		8.07
Σύνολο Σ.σ			3.84		2.95		3.18

Σύνολο / Σύνολο Σ.σ			3.35		2.83		2.53
---------------------------	--	--	------	--	------	--	------

**Πίνακας 5-2.** Φτιάχνοντας συστάσεις για τον User\_6.

Ο παραπάνω πίνακας δείχνει τους συντελεστές σχετικότητας και τις βαθμολογίες των χρηστών για τις ταινίες που δεν έχει δει ο User\_6. Οι στήλες που ξεκινάνε με Sim είναι το γινόμενο του συντελεστή συσχέτισης με την βαθμολογία, έτσι ώστε αν κάποιος χρήστης παρουσιάζει μεγαλύτερο βαθμό ομοιότητας με τον User\_6, να συνεισφέρει περισσότερο στο συνολικό σκορ από κάποιον με τον οποίο ο User\_6 έχει διαφορετικά ενδιαφέροντα. Η γραμμή Σύνολο περιέχει το άθροισμα αυτών των αριθμών.

Θα μπορούσαμε απλά να χρησιμοποιήσουμε το Σύνολο για να υπολογίσουμε τις βαθμολογίες αλλά τότε, μια ταινία που έχει βαθμολογηθεί από πολλά άτομα θα είχε μεγάλο πλεονέκτημα. Για το διορθώσουμε αυτό διαιρούμε το Σύνολο με το Σύνολο Σ.χ (Σύνολο συντελεστών συσχετίσεων) για όσους έχουν δει τη συγκεκριμένη ταινία. Για παράδειγμα για το Lopper που το έχουν δει όλοι οι χρήστες διαιρούμε το Σύνολο με το άθροισμα όλων των Συντελεστών Συσχετίσεων, ενώ για το Django που δεν το έχει δει ο User\_3 διαιρούμε το Σύνολο μόνο το άθροισμα των υπόλοιπων Συντελεστών Συσχετίσεων. Η τελευταία γραμμή περιέχει τα αποτελέσματα αυτών των διαιρέσεων.

## 5.7 Βρίσκοντας ταινίες με βάση τον τίτλο

Μέχρι στιγμής μπορούμε να βρούμε με ποιους χρήστες ταιριάζουμε περισσότερο, καθώς και να βρούμε ταινίες για να τις προτινουμε σε κάποιον χρήστη. Τι γίνεται όμως όταν κάποιος θέλει να δει ποιές ταινίες είναι παρόμοιες μεταξύ τους; Σίγουρα θα έχετε αντιμετωπίσει τέτοιου είδους καταστάσεις σε πολλές ιστοσελίδες στο διαδίκτυο ειδικότερα όταν οι ιστοσελίδες αυτές δεν έχουν συλλέξει πολλές πληροφορίες για εσάς. Η παρακάτω εικόνα δείχνει που εμφανίζεται στην ιστοσελίδα του Amazon δείχνει παρόμοια προϊόντα με μία συγκεκριμένη αγορά.

## Customers Who Bought This Item Also Bought



Advanced Web Metrics with Google Analytics  
> Brian Clifton  
★★★★★ (11)  
Paperback  
\$23.78

Convert!: Designing Web Sites to Increase Sales & Conversion  
> Ben Hunt  
★★★★★ (39)  
Paperback  
\$19.79

Web Analytics: An Hour a Day  
> Avinash Kaushik  
★★★★★ (81)  
Paperback  
\$17.84

Εικόνα 5-4. Παρουσίαση παρόμοιων προϊόντων με μια συγκεκριμένη αγορά από το Amazon.

Σε αυτή τη περίπτωση η ομοιότητα καθορίζεται κοιτώντας σε ποιους άρεσε ένα συγκεκριμένο προϊόν και βλέποντας στη συνέχεια τα υπόλοιπα προϊόντα που τους άρεσαν. Αυτή βασικά, είναι η ίδια μέθοδος με αυτή που χρησιμοποιήσαμε νωρίτερα για να καθορίσουμε την ομοιότητα των χρηστών του συστήματος. Το μόνο που χρειάζεται να κάνουμε είναι να κάνουμε μία ανταλλαγή μεταξύ των χρηστών και των ταινιών που έχουμε αποθηκευμένα στο HashMap αντικείμενο που χρησιμοποιούμε για την συλλογή των προτιμήσεων των χρηστών που αναφέραμε νωρίτερα σε αυτό το κεφάλαιο. Για να γίνουμε πιο κατανοητοί θα πρέπει να μετατρέψουμε το HashMap από:

```
{'User_0', {'Django': 2.5, 'Life of Pi': 3.5},  
'User_1', {'Django': 3.0, 'Life of Pi': 3.5}}
```

σε:

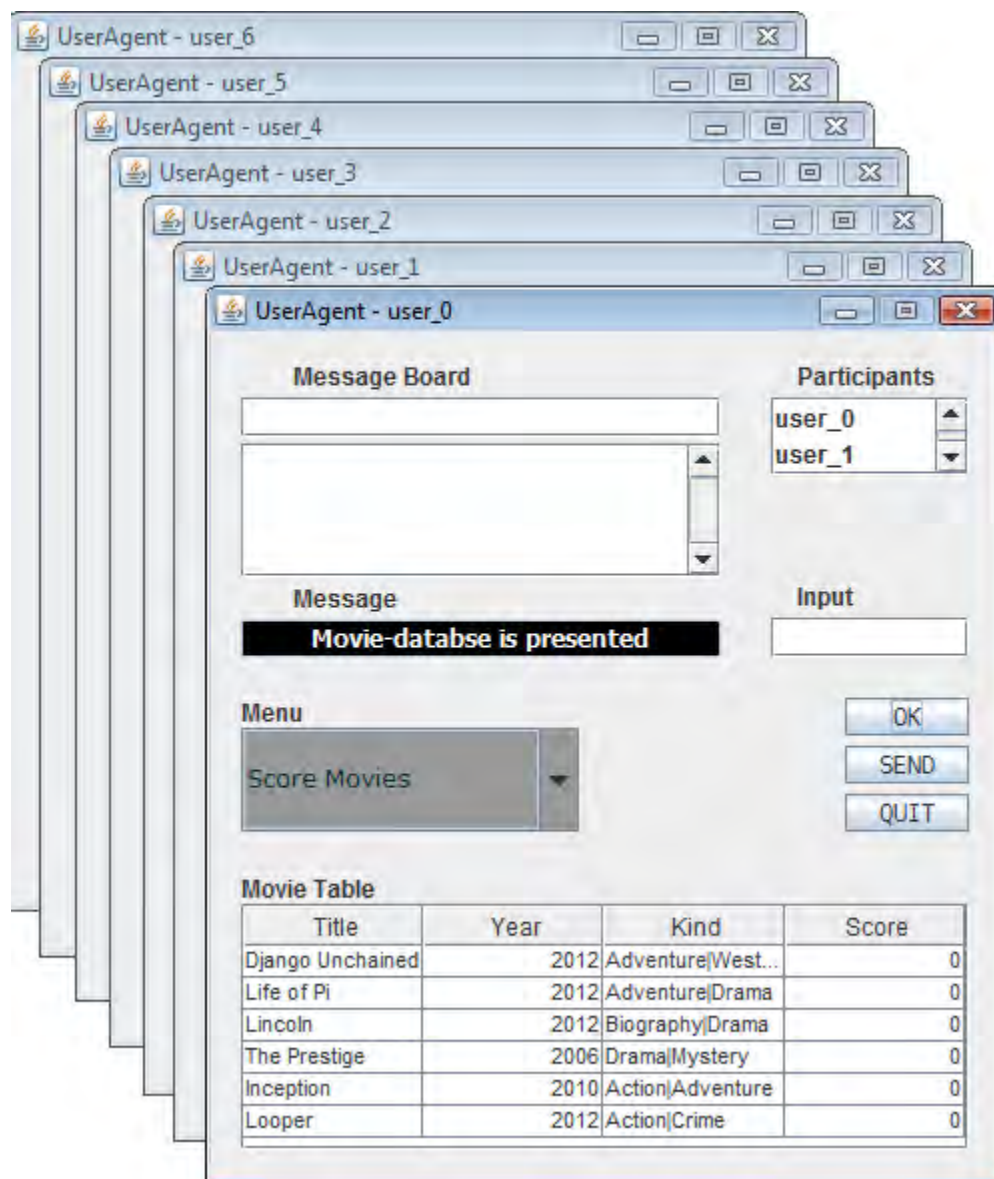
```
{'Django', {'User_0': 2.5, 'User_1': 3.0},  
'Life of Pi', {'User_0': 3.5, 'User_1': 3.5}}
```

## 5.8 Επικοινωνία μεταξύ των χρηστών

Όπως εξηγήσαμε και στο προηγούμενο κεφάλαιο θα υπάρχει επιλογή για τον χρήστη της εφαρμογής για να μπορεί να στέλνει μηνύματα σε όποιον χρήστη θέλει. Εφικτή είναι και η επικοινωνία με πολλούς χρήστες ταυτόχρονα.

## 5.9 Παραδειγματική εκτέλεση

Θα παρουσιάσουμε στη συνέχεια μερικές εικόνες από εκτελέσεις της εφαρμογής μας δίνοντας σαν είσοδο δεδομένων τον πίνακα 5-1. Ξεκινούμε λοιπόν την εφαρμογή με 7 χρήστες και αφού επιλέξουμε την επιλογή Score Movies για καθέναν από αυτούς προκύπτει η ακόλουθη εικόνα.

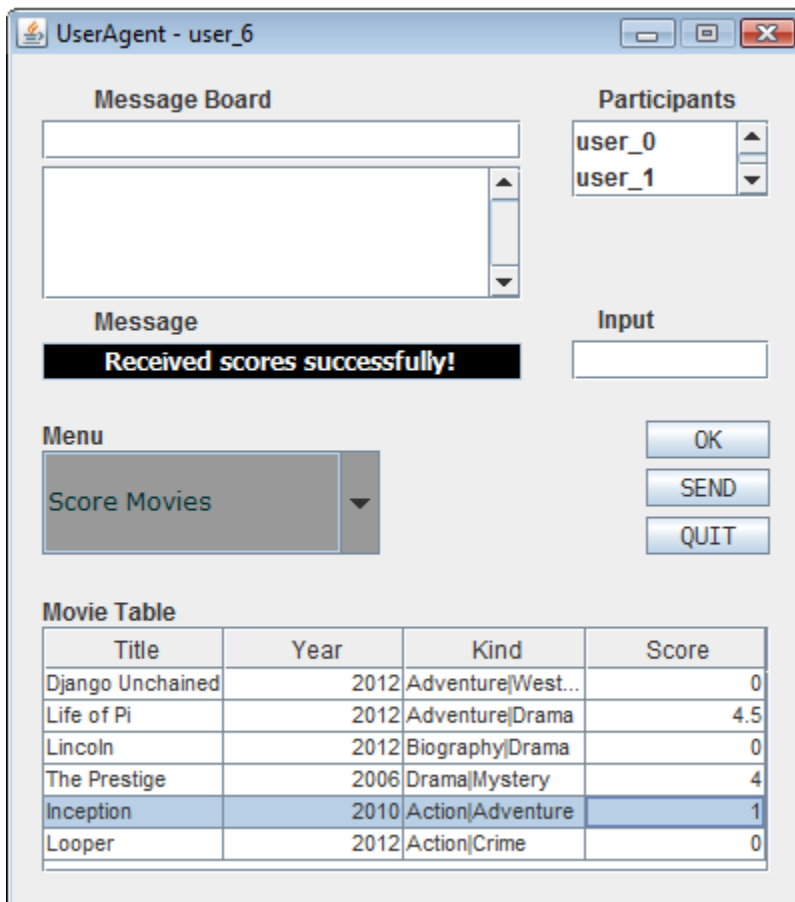


Εικόνα 5-5. Εκκίνηση της εφαρμογής.

Για επεξηγηματικούς λόγους θα δείχνουμε από εδώ και στο εξής μόνο την εικόνα του User\_6.

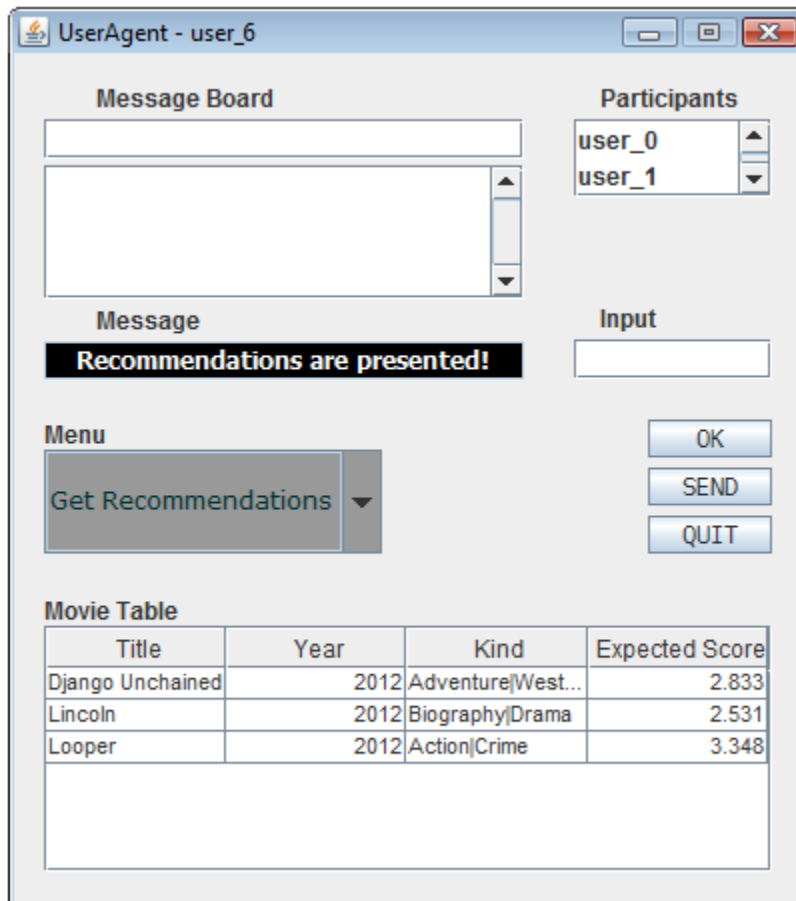
Όπως παρατηρούμε, με την επιλογή Score Movies ο ServerAgent επιστρέφει στον χρήστη τις ταινίες που υπάρχουν στη βάση δεδομένων περιμένοντας από αυτόν να αξιολογήσει αυτές που έχει δει.

Μόλις ο χρήστης User\_6 βαθμολογήσει τις ταινίες με βάση πάντα τον πίνακα 5-1 και πατήσει το κουμπί **Send** εμφανίζεται η παρακάτω εικόνα η οποία ενημερώνει τον χρήστη με ένα μήνυμα από τον FacilitatorAgent ότι έλαβε τις προτιμήσεις του πρώτου επιτυχώς.



Εικόνα 5-5. Μήνυμα ενημέρωσης επιτυχούς λήψης προτιμήσεων.

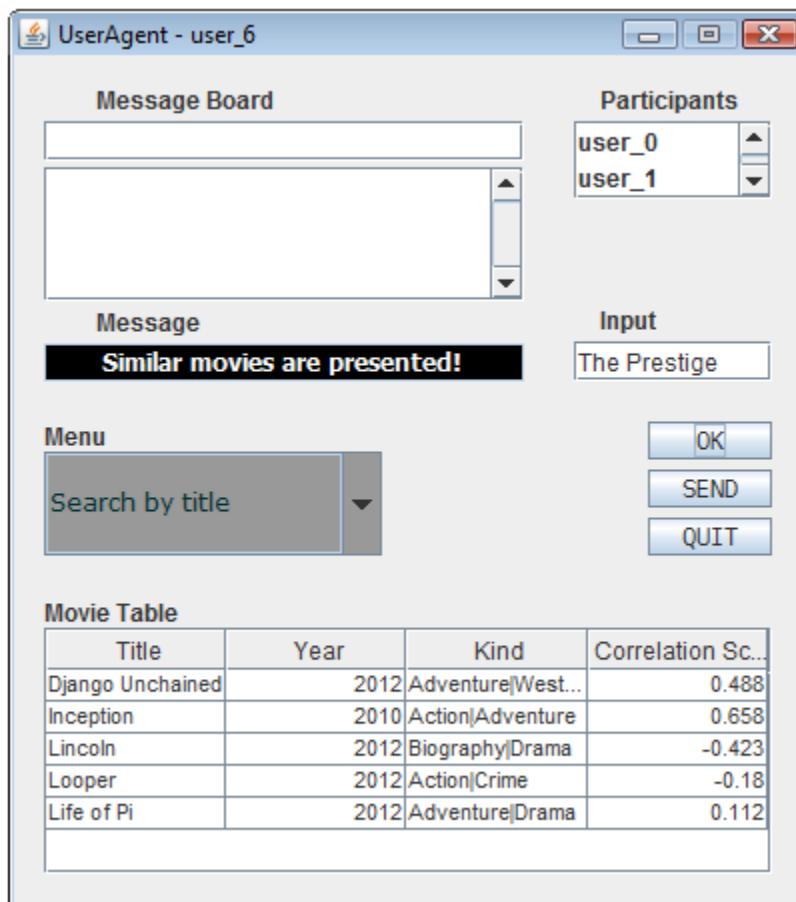
Στη συνέχεια (αφού στείλουν και οι υπόλοιποι χρήστες τις προτιμήσεις τους) ας πάμε να βρούμε προτινόμενες ταινίες για τον User\_6. Αφού επιλέξουμε λοιπόν Get Recommendations από το μενού, εμφανίζεται η παρακάτω εικόνα.



**Εικόνα 5-6.** Αποτελέσματα συστάσεων ταινιών.

Απ' ότι βλέπουμε, ο User\_6 όχι μόνο παίρνει μία λίστα με ταινίες, αλλά δέχεται επίσης μια μαντεψιά από το σύστημα για το ποια θα μπορούσε να είναι η δική του βαθμολογία για τις συγκεκριμένες ταινίες!

Πάμε τώρα να δούμε τις ταινίες που μοιάζουν με το “The Prestige”. Αφού ο χρήστης επιλέξει από το μενού Search by title και στη συνέχεια πληκτρολογήσει στο πεδίο Input τον τίτλο της ταινίας για την οποία επιθυμεί να λάβει παρόμοιες ταινίες, του εμφανίζεται η εξής εικόνα. Όσο μεγαλύτερο είναι το Correlation Score, τόσο πιο πολύ μοιάζουν οι ταινίες μεταξύ τους.

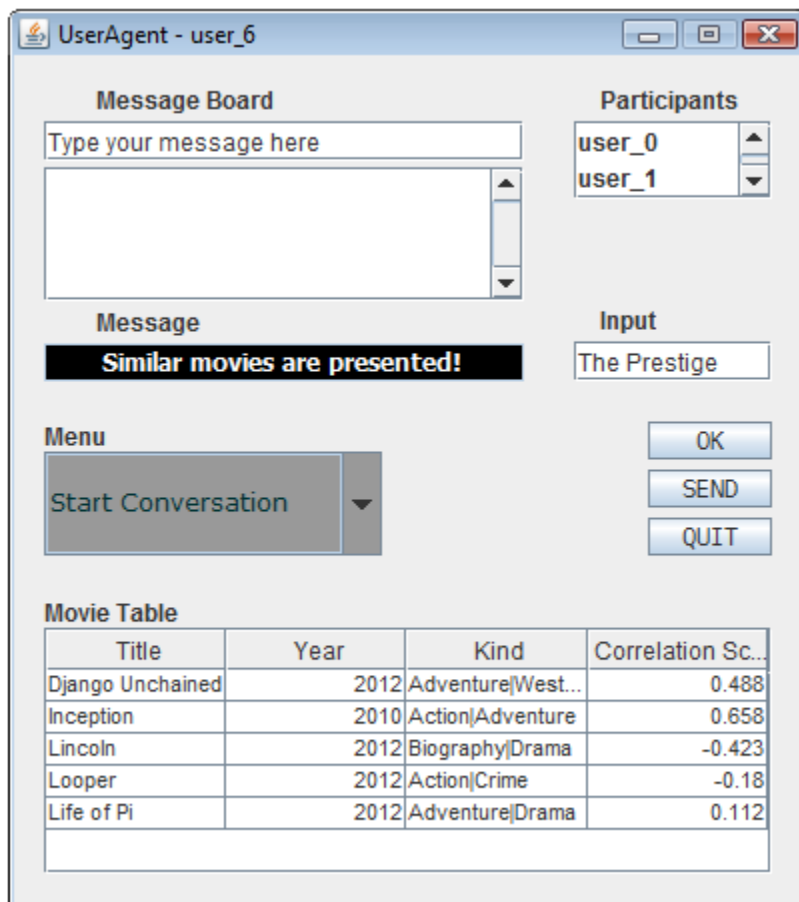


**Εικόνα 5-7.** Αποτελέσματα παρόμοιων ταινιών με το “The Prestige”

Παρατηρούμε ότι στην 4<sup>η</sup> στήλη εμφανίζονται μερικά αρνητικά αποτελέσματα. Αυτό σημαίνει ότι σε αυτούς που άρεσε το The Prestige δεν τους άρεσε και πολύ το Lincoln, όπως φαίνεται και από την εικόνα.

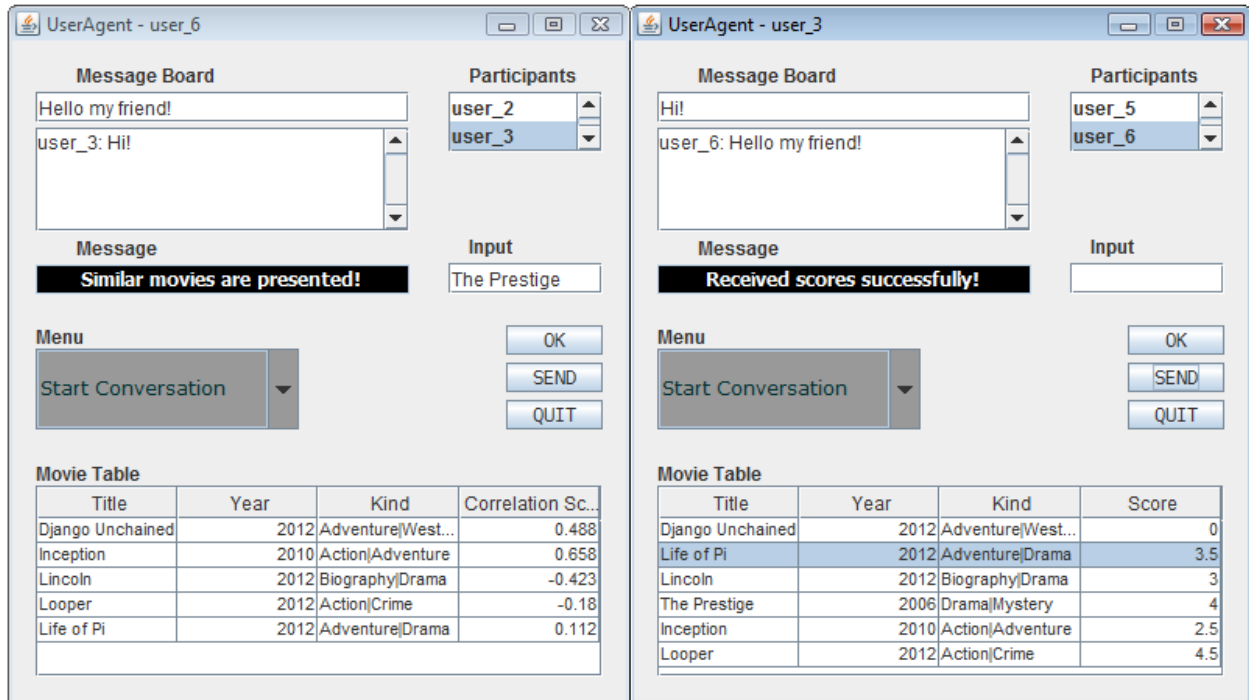


Στη συνέχεια, πάμε να δούμε τον τρόπο με τον οποίο επικοινωνούν οι χρήστες της εφαρμογής. Αφού επιλέξουν από το μενού Start Conversation, εμφανίζεται ένα μήνυμα στον “Type your message here” και ο κέρσορας μεταβαίνει στο πεδίο αυτό για να γράψει ο χρήστης το μήνυμά του.



Εικόνα 5-8. Επιλογή Συνομιλίας με άλλους χρήστες.

Όταν ο χρήστης πληκτρολογήσει το μήνυμά του, μπορεί να επιλέξει από το πεδίο “Participants” τον χρήστη στον οποίο επιθυμεί να στείλει το μήνυμά του και στη συνέχεια πατάει το κουμπί **SEND**. Αφού γίνει αυτό ο χρήστης στον οποίο στάλθηκε το μήνυμα, βλέπει το μήνυμα αυτό στην περιοχή που φαίνεται από την παρακάτω εικόνα.

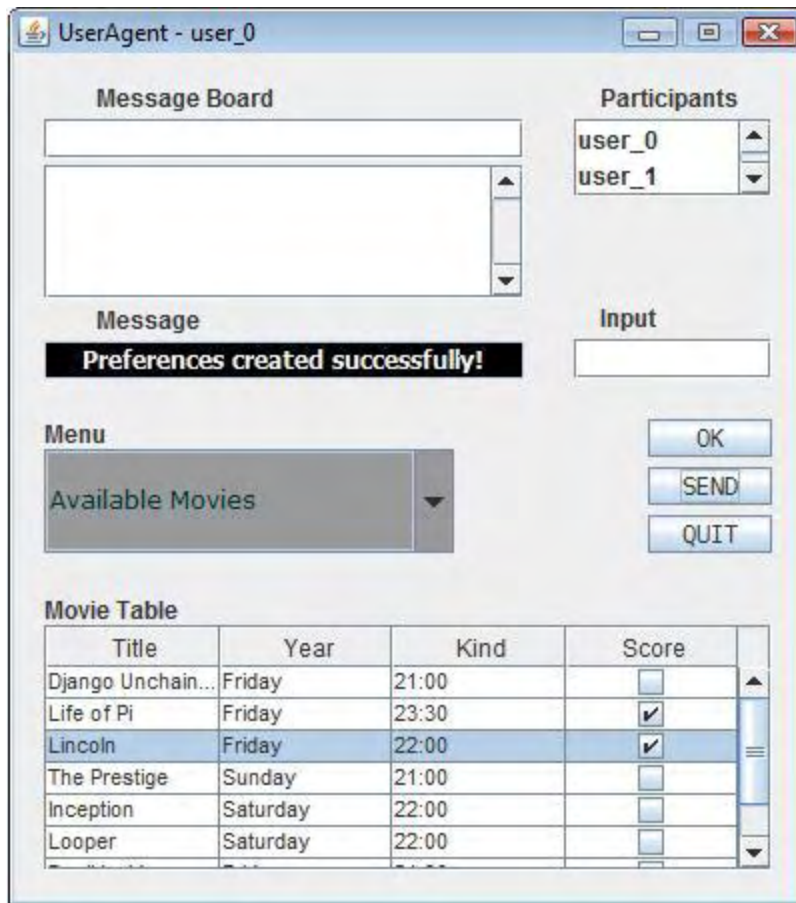


**Εικόνα 5-9.** Στιγμιότυπο επικοινωνίας μεταξύ του User\_3 και User\_6.

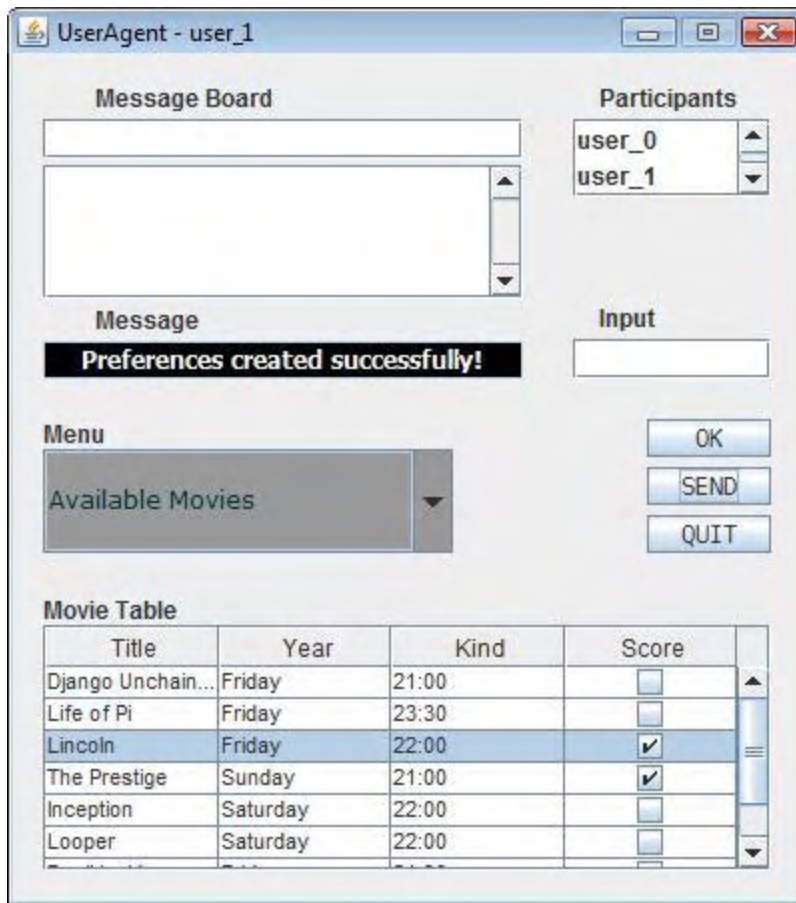
Παρατηρούμε ότι κάθε φορά που κάποιος δέχεται ένα μήνυμα, ξέρει επίσης από ποιον του ήρθε. Τα εισερχόμενα μηνύματα μπαίνουν το ένα κάτω από το άλλο. Αυτό υλοποιείται με το αντικείμενο JTextArea της κλάσης Swing. Να πούμε επίσης ότι ένας χρήστης μπορεί να δέχεται μηνύματα από πολλούς χρήστες και να επικοινωνεί με όλους αυτούς ταυτόχρονα.

Τέλος, παρουσιάζουμε τη λειτουργία του χρονοπρογραμματισμού. Αφού επιλέξουν οι χρήστες τις προτιμήσεις τους σχετικά με την ώρα και μέρα μιας συγκεκριμένης ταινίας τις στέλνουν στον FacilitatorAgent και στη συνέχεια αυτός αφού ομαδοποιήσει τις κοινές αποφάσεις μεταξύ των χρηστών, τις εμφανίζει ώστε οι τελικοί χρήστες του συστήματος να ξέρουν τα ζευγάρια που διαμορφώθηκαν και να ξέρουν ποιοι συμφωνούν με το προγράμμα τους.

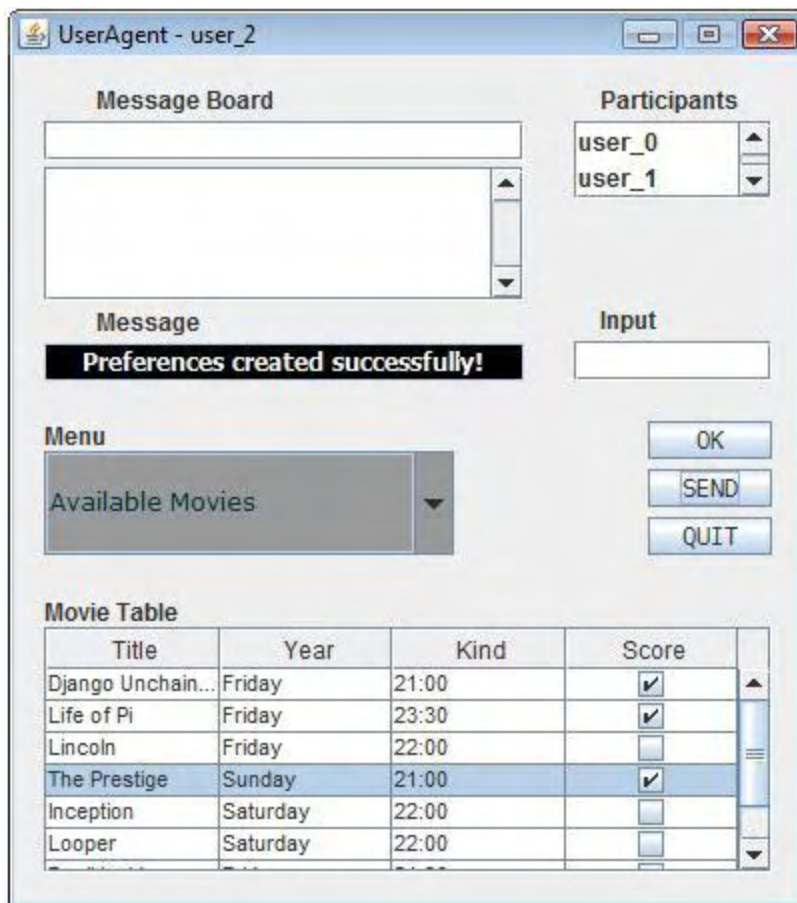
Αφού εισάγουν οι χρήστες τις προτιμήσεις που φαίνονται από τις παρακάτω εικόνες



Εικόνα 5-10. Προτιμήσεις του χρήστη user\_0

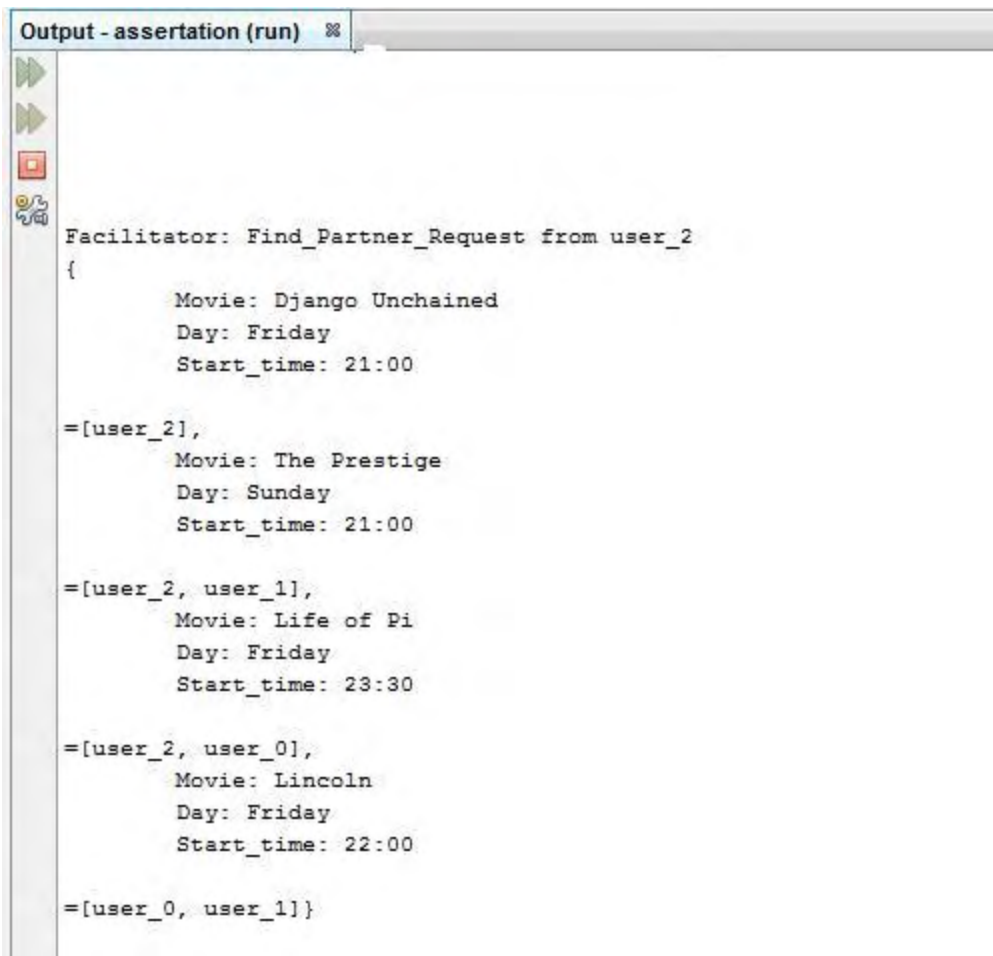


Εικόνα 5-11. Προτιμήσεις του χρήστη user\_1.



Εικόνα 5-12. Προτιμήσεις του χρήστη user\_2.

Ο FacilitatorAgent παράγει τα εξής σύνολα συμφωνιών.



```
Output - assertion (run) ✖
Facilitator: Find_Partner_Request from user_2
{
  Movie: Django Unchained
  Day: Friday
  Start_time: 21:00

=[user_2],
  Movie: The Prestige
  Day: Sunday
  Start_time: 21:00

=[user_2, user_1],
  Movie: Life of Pi
  Day: Friday
  Start_time: 23:30

=[user_2, user_0],
  Movie: Lincoln
  Day: Friday
  Start_time: 22:00

=[user_0, user_1]
```

Εικόνα 5-13. Σύνολα συμφωνιών των χρηστών.

## 5.10 Πλατφόρμες και προγραμματιστικά εργαλεία

### 5.10.1 Ενσωματωμένο περιβάλλον ανάπτυξης (IDE) Netbeans

Για την υλοποίηση της εφαρμογής μας χρησιμοποιήθηκε το περιβάλλον ανάπτυξης **Netbeans 7.2.1**. Ο λόγος που επιλέχθηκε το συγκεκριμένο περιβάλλον είναι η ευχρηστία του, καθώς και η προηγούμενη εμπειρία χρήσης του.

## 5.10.2 Πλατφόρμα JADE

Όπως αναφέραμε και στο προηγούμενο κεφάλαιο χρησιμοποιήσαμε την πλατφόρμα JADE για την ανάπτυξη της εφαρμογής μας. Το JADE είναι μία πλατφόρμα εκτέλεσης (middleware) ελεύθερου λογισμικού, που διευκολύνει την ανάπτυξη πολυπρακτορικών συστημάτων. Ανάμεσα σε άλλα περιλαμβάνει:

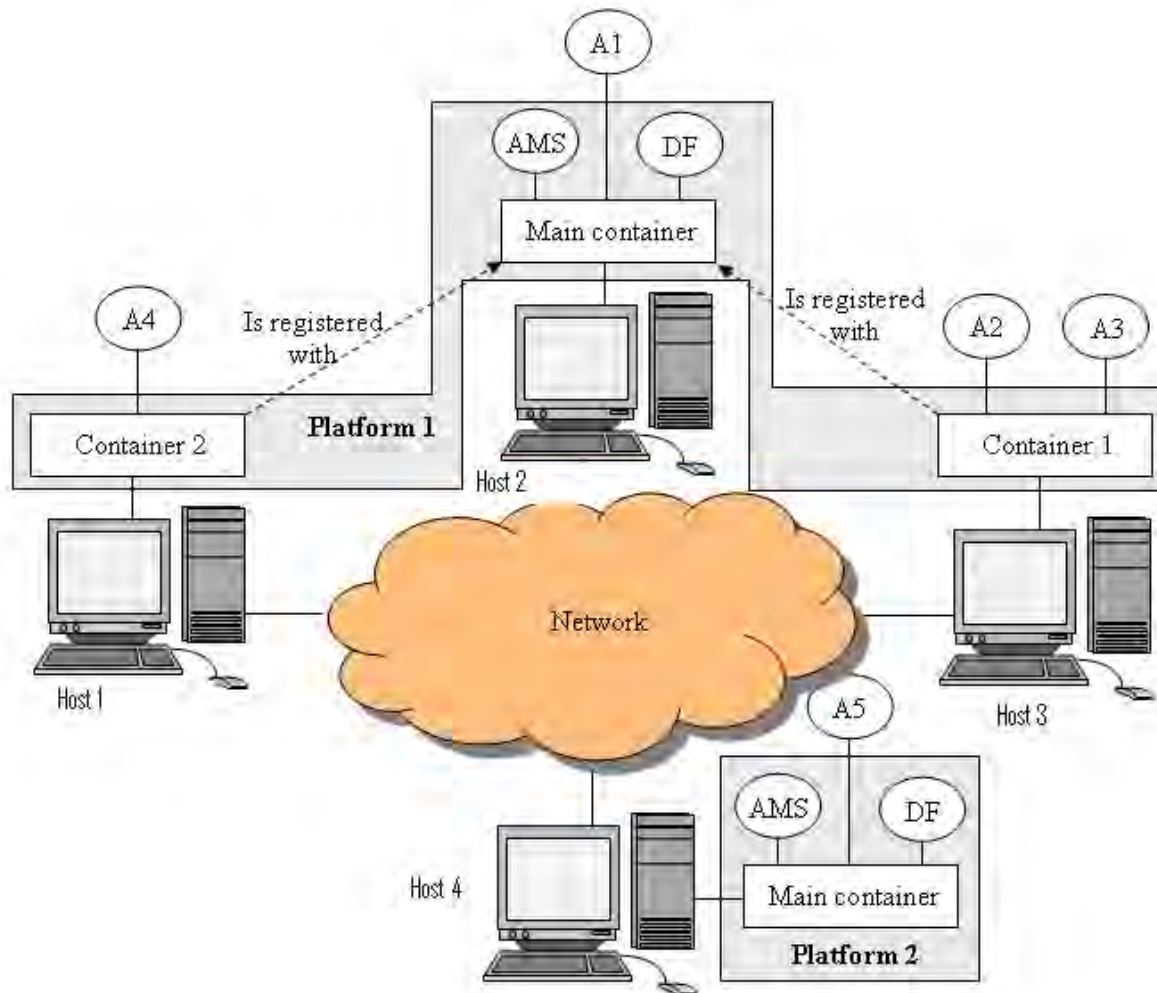
- Περιβάλλον εκτέλεσης (runtime environment) όπου ζουν οι πράκτορες.
- Βιβλιοθήκη κλάσεων που απαιτούνται για την ανάπτυξη των πρακτόρων.
- Μια σουίτα γραφικών εργαλείων, που επιτρέπει τη διαχείριση και την παρακολούθηση των δραστηριοτήτων των πρακτόρων.

Κάθε στιγμιότυπο “τρεξίματος” του περιβάλλοντος εκτέλεσης JADE λέγεται *container*, καθώς μπορεί να περιέχει πολλούς πράκτορες. Το σύνολο των ενεργών πρακτόρων λέγεται *platform*. Απαιτείται πάντα να υπάρχει ένα *main container*, όπου όλα τα υπόλοιπα container πρέπει να εγγραφούν σε αυτό για να χρησιμοποιήσουν την πλατφόρμα. Αν δημιουργηθεί κάποιο άλλο *main container*, αμέσως συνιστά ένα άλλο *platform*. Εκτός από την ικανότητα αποδοχής καταχωρίσεων από άλλα container ένα *main container* διαφέρει από τα υπόλοιπα καθώς περιλαμβάνει δύο ειδικούς πράκτορες (οι οποίοι ξεκινάνε αυτόματα όταν ξεκινήσει και το *main container*), τον **AMS** και τον **DF**, τους οποίους περιγράψαμε στο προηγούμενο κεφάλαιο.

Οι JADE πράκτορες διακρίνονται από ένα μοναδικό όνομα και, γνωρίζονται τα ονόματα μεταξύ τους μπορούν να επικοινωνήσουν ανεξάρτητα με την τοποθεσία τους:

- Στο ίδιο container (A2 και A3).
- Σε διαφορετικά containers στην ίδια πλατφόρμα (A1 και A2).
- Σε διαφορετικές πλατφόρμες.

Μπορεί κανείς να δει την αρχιτεκτονική του JADE από την παρακάτω εικόνα



**Εικόνα 5-14:** Αρχιτεκτονική του JADE

### 5.10.3 Εκκίνηση του περιβάλλοντος JADE στο Netbeans

Αφού ολοκληρωθεί η διαδικασία ορισμού της λειτουργικότητας κάθε πράκτορα, αυτοί πρέπει να ενεργοποιηθούν με σκοπό να επιτελέσουν το έργο τους.

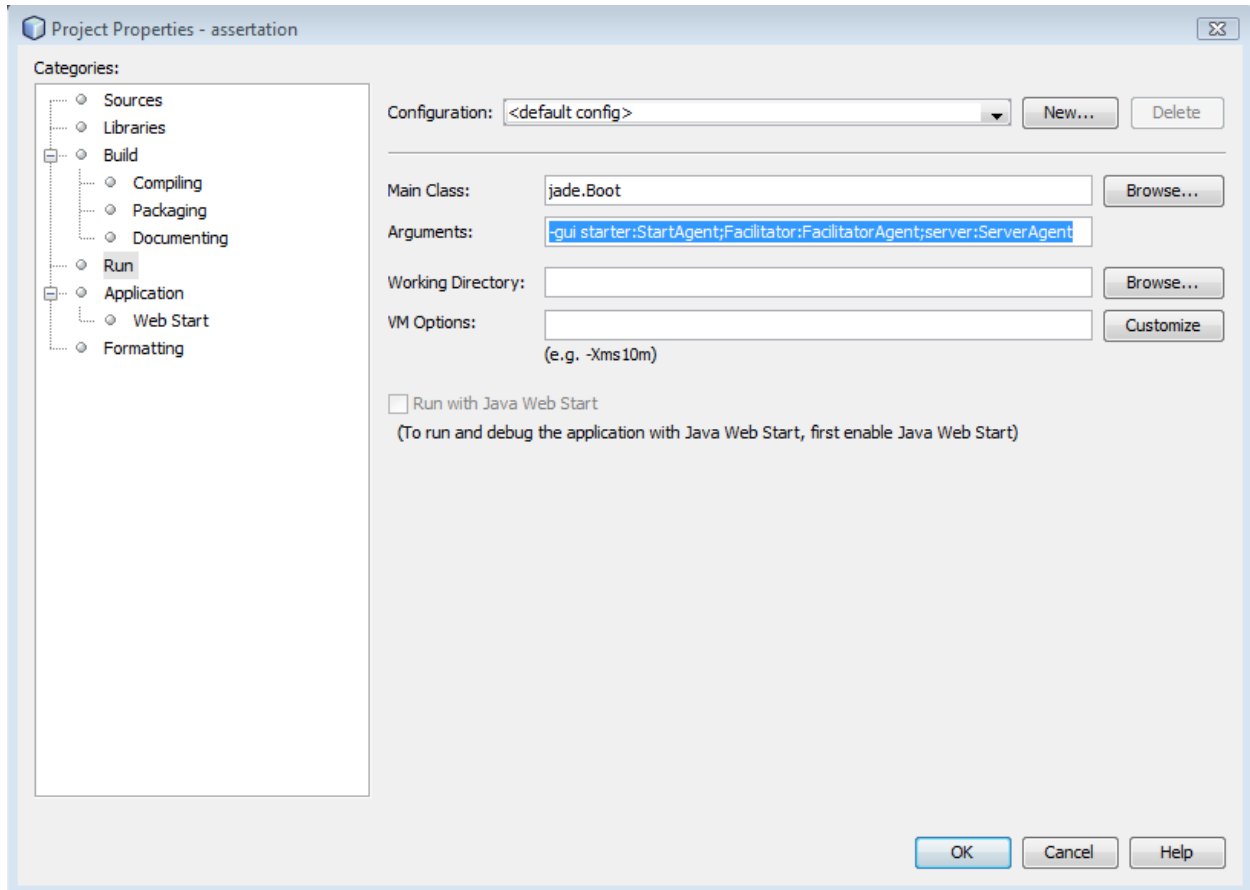
Για να εκκινήσουμε την πλατφόρμα JADE από το Netbeans πρέπει πρώτον να προσθέσουμε τις βιβλιοθήκες (jade.jar, commons-codec-1.3.jar) του JADE στο working directory της εφαρμογής. Στη συνέχεια, με την επιλογή Run -> Set Project Configuration ορίζουμε σαν Main Class την Jade.Boot από τη βιβλιοθήκη του JADE (όπως αναφέρουμε παρακάτω).



Συνεχίζοντας, στο πεδίο Arguments πληκτρολογούμε την εντολή:

```
-gui starter:StartAgent;Facilitator:FacilitatorAgent;server:ServerAgent
```

Όπως φαίνεται και στο παράθυρο:

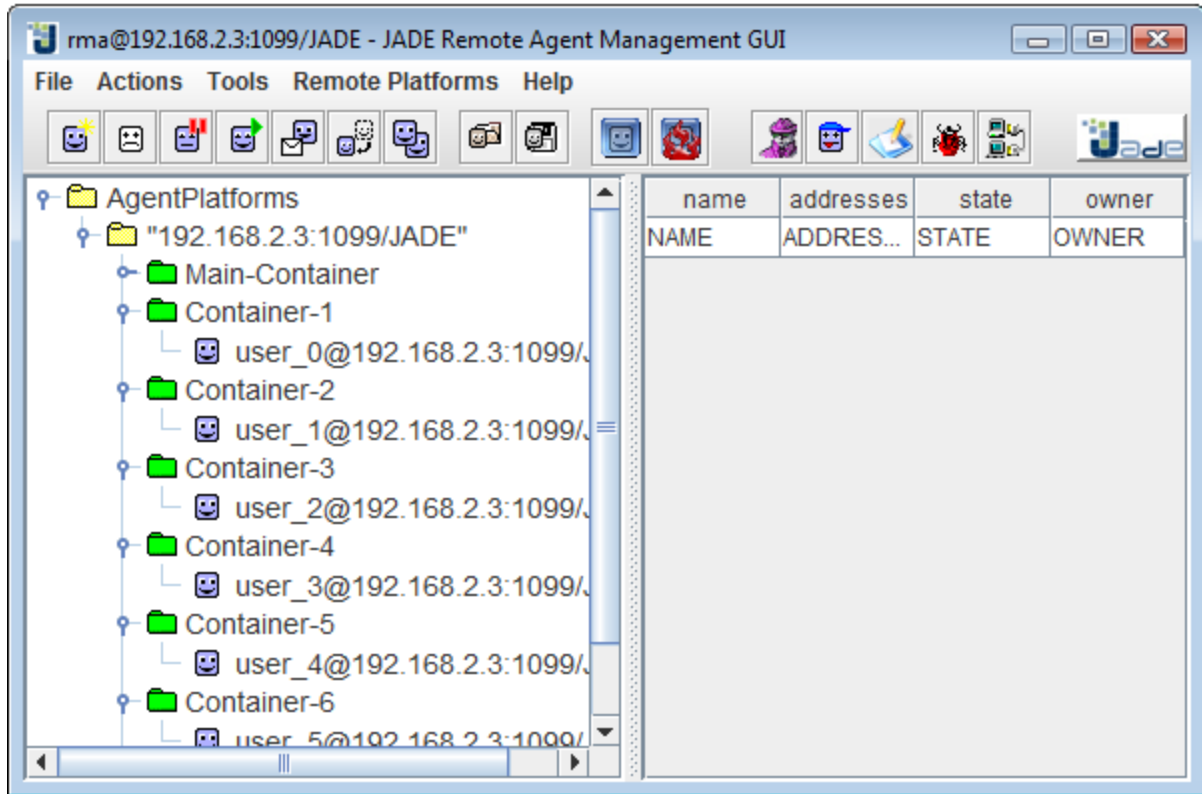


**Εικόνα 5-15:** Ενεργοποίηση πρακτόρων

Με τον τρόπο αυτόν εκκινούμε τους πράκτορες StartAgent, FacilitatorAgent, ServerAgent.

Τέλος, παραθέτουμε και ένα στιγμιότυπο του γραφικού περιβάλλοντος διεπαφής χρήστη του JADE που δείχνει τα υπάρχουσα containers και platforms καθώς και τα περιφεριακά containers που αντιπροσωπεύουν τους UserAgents και συνδέονται με το main container.

Αφού λοιπόν ο StartAgent εκκινήσει 7 UserAgents, βλέπουμε την ακόλουθη εικόνα.



Εικόνα 5-16: Γραφικό περιβάλλον JADE

# 6 Επίλογος

## 6.1 Σύνοψη και συμπεράσματα

Το πρόβλημα του χρονοπρογραμματισμού συναντήσεων είναι ένα δύσκολο πρόβλημα που δεν επιδέχεται εύκολα βελτιστοποίηση. Στην παρούσα υλοποίηση ο κάθε χρήστης έχει τη δυνατότητα να δει όλα τα κοινά σύνολα συμφωνιών και να αποφασίσει στη συνέχεια ποιό γκρούπ θέλει να ακολουθήσει. Αυτό μπορεί να είναι μία εύκολη απόφαση δεδομένου ότι μπορεί να γνωρίζει εκ των προτέρων με βάση τις συστάσεις που έχει πάρει από το σύστημα κατά πόσο θα του αρέσει μία ταινία ή όχι.

Η ποσότητα της πληροφορίας αυξάνεται πολύ γρηγορότερα από την ικανότητά μας να την επεξεργαστούμε. Όλοι έχουμε νιώσει λίγο πολύ το αίσθημα του κατακλυσμού από πληροφορίες σχετικά με τον τον αριθμό βιβλίων, τα άρθρα που αναρτώνται στο Διαδίκτυο και τις καινούργιες ταινίες που προβάλλονται κάθε χρόνο. Σε αυτές τις συνθήκες είναι δύσκολο να αποφασίσει κανείς τι είναι αυτό που θα ικανοποιήσει περισσότερο τις ανάγκες του. Τα συστήματα συστάσεων αποτελούν για αυτόν τον λόγο μία ισχυρή νέα τεχνολογία για την εξόρυξη φιλτραρισμένης πληροφορίας από τεράστιες βάσεις δεδομένων. Τα συστήματα αυτά βοηθούν τους χρήστες να βρίσκουν εύκολα και γρήγορα αντικείμενα που θέλουν να αγοράσουν από μία επιχείρηση, αλλά και την επιχείρηση την ίδια αποφέροντας περισσότερες πωλήσεις. Γίνονται ολοένα και πιο σημαντικά εργαλεία στο ηλεκτρονικό εμπόριο στο Διαδίκτυο.

Στη διπλωματική αυτή έγινε η προσπάθεια να παρουσιάσουμε μια εφαρμογή συστάσεων ταινιών για τους χρήστες που τη χρησιμοποιούν. Συγκεκριμένα, το σύστημα δέχεται τις προσωπικές προτιμήσεις του κάθε χρήστη και προβλέπει τις προτιμήσεις τους. Προτίνει με αυτόν τον τρόπο τις κατάλληλες ταινίες στους χρήστες. Επιπλέον προσθέσαμε και το στοιχείο της επικοινωνίας μεταξύ των χρηστών. Η υλοποίηση της εφαρμογής έγινε με τις πλατφόρμες JADE και Netbeans. Η πλατφόρμα JADE είναι πολύ εύκολη στη μάθηση και τη χρήση. Ακόμα συμπεράναμε ότι είναι η κατάλληλη πλατφόρμα για ανάπτυξη πολυ-πρακτορικών συστημάτων, αφού υποστηρίζει πολλά στοιχεία που χρησιμοποιούνται από αυτά, όπως η επικοινωνία, η συμπεριφορά πρακτόρων και η οντολογία.

## 6.2 Μελλοντικές επεκτάσεις

Όσον αφορά τη λειτουργία του χρονοπρογραμματισμού θα μπορούσαμε εκτός από τη μέρα και την ώρα να προσθέσουμε επιπλέον κριτήρια στα οποία διαπραγματεύονται οι χρήστες (π.χ τοποθεσία). Επίσης μπορεί να βελτιωθεί και το πρωτόκολο κοινών συμφωνιών εισάγοντας γύρους επικοινωνίας. Όσον αφορά τη λειτουργία του συλλογικού φιλτραρίσματος, από τη φύση του συστήματος δεν είναι απλό έργο να αξιολογήσουμε την απόδοση δεδομένου ότι δεν υπάρχει σωστή ή λάθος σύσταση. Είναι καθαρά θέμα άποψης. Μία μεγαλύτερη βάση δεδομένων με ταινίες, καθώς και ένας αξιοπρεπής αριθμός αξιολόγησης ταινιών από τους χρήστες, θα επιτρέψει πιο ουσιαστικά αποτελέσματα χρησιμοποιώντας την εφαρμογή μας. Ωστόσο αυτό θα επιφέρει επιπλέον κόστος όσον αφορά τον χρόνο απόκρισης αποτελεσμάτων. Υπάρχουν αρκετοί άλλοι αλγόριθμοι που μπορούμε να χρησιμοποιήσουμε για την συλλογή προτιμήσεων των χρηστών και την επεξεργασία τους. Κάποιοι είναι περισσότερο αποδοτικοί όσον αφορά την ταχύτητα με την οποία επεξεργάζονται τα δεδομένα, αλλά μπορεί να υστερούν σε ακρίβεια. Μία πιθανή επέκταση είναι να χρησιμοποιήσουμε κάποιον άλλον αλγόριθμο και να συγκρίνουμε τους χρόνους απόκρισης για μεγάλα σύνολα δεδομένων. Αφού επιλέξουμε τον πιο αποδοτικό όσον αφορά την ταχύτητα με την οποία επιστρέφονται οι ταινίες στους χρήστες και την ακρίβεια των αποτελεσμάτων, θα μπορούσαμε να εισάγουμε στην εφαρμογή μας περισσότερους από έναν Facilitator Agents και να μοιράζονται την δουλειά που κάνει ο ένας στη δική μας υλοποίηση. Κάτι άλλο που θα μπορούσαμε να κάνουμε για να μειώσουμε τον χρόνο απόκρισης είναι να επιτρέψουμε στον Facilitator Agent να επεξεργάζεται τα δεδομένα χωρίς να του ζητηθεί ρητά από τον χρήστη. Την ώρα δηλαδή που κάποιος χρήστης στέλνει τις αξιολογήσεις που έχει κάνει για διάφορες ταινίες στον Facilitator Agent, ο τελευταίος να επεξεργάζεται τα όσα δεδομένα έχει από όλους χρήστες του έχουν στείλει αξιολογήσεις μέχρι εκείνη τη στιγμή. Αυτό θα μπορούσε να βελτιώσει τον χρόνο απόκρισης του συστήματος αλλά θα επέφερε άσκοπους υπολογισμούς. Τέλος, όπως είδαμε, ο User Agent αναπαρίσταται από ένα GUI, το οποίο βοηθάει τον τελικό χρήστη να χειρίζεται τις λειτουργίες του. Θα μπορούσαμε να χειρόμαστε τον User Agent από ένα κινητό τηλέφωνο. Αυτό είναι εφικτό διότι το JADE παρέχει πρόσθετα εργαλεία (**LEAP add-on**) τα οποία επιτρέπουν την ανάπτυξη και ενεργοποίηση πρακτόρων σε κινητά τηλέφωνα που υποστηρίζουν Java.

## 7 Βιβλιογραφία

[1] *Recommender Systems: An Introduction* Dietmar Jannach, Markus Zanker, Alexander Felfernig, Gerhard Friedrich.

[2] [http://en.wikipedia.org/wiki/Collaborative\\_filtering](http://en.wikipedia.org/wiki/Collaborative_filtering)

[3] [http://en.wikipedia.org/wiki/Recommendation\\_system](http://en.wikipedia.org/wiki/Recommendation_system)

[4] [http://en.wikipedia.org/wiki/Collective\\_intelligence](http://en.wikipedia.org/wiki/Collective_intelligence)

[5] [http://en.wikipedia.org/wiki/Pearson\\_product-moment\\_correlation\\_coefficient](http://en.wikipedia.org/wiki/Pearson_product-moment_correlation_coefficient)

[6] <http://www.iro.umontreal.ca/~vaucher/Agents/Jade/JadePrimer.html>

[7] <http://jade.tilab.com/doc/tutorials/JADEProgramming-Tutorial-for-beginners.pdf>

[8] <http://jade.tilab.com/doc/programmersguide.pdf>

[9] <http://jade.tilab.com/doc/api/index.html>

[10] *Developing Multi-agent Systems with JADE* Fabio Belfemine, Giovanni Caire, Dominic Greenwood.