



**ΑΛΕΞΑΝΔΡΕΙΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ
ΙΔΡΥΜΑ ΘΕΣΣΑΛΟΝΙΚΗΣ**

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

**ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ – WEB INTELLIGENCE**

**Συναισθηματικοί Πράκτορες Διασύνδεσης σε
Πολυπρακτορικά Συστήματα**

(Interface Emotional Agents in Multi-Agent Systems)

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΗΛΙΑ ΚΑΛΠΑΞΙΔΗ

Επιβλέπων : Δημοσθένης Σταμάτης
Καθηγητής, ΑΤΕΙΘ

Θεσσαλονίκη, Μάιος 2015



ΑΛΕΞΑΝΔΡΕΙΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ
ΘΕΣΣΑΛΟΝΙΚΗΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ – WEB INTELLIGENCE

Συναισθηματικοί Πράκτορες Διασύνδεσης σε Πολυπρακτορικά Συστήματα

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΗΛΙΑ ΚΑΛΠΑΞΙΔΗ

Επιβλέπων : Δημοσθένης Σταμάτης
Καθηγητής, ΑΤΕΙΘ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή στις 1 Ιουλίου 2015.

Σταμάτης Δημοσθένης
Καθηγητής

Αντώνης Σιδηρόπουλος
Καθηγητής Εφαρμογών / Λέκτορας

Ευκλείδης Κεραμόπουλος
Επίκουρος Καθηγητής

Θεσσαλονίκη, 1 Ιουλίου 2015

.....
Ηλίας Καπαζίδης
Μηχανικός Πληροφορικής
© 2015– Allrightsreserved

Περίληψη

Η τρέχουσα έρευνα στην Τεχνητή Νοημοσύνη έχει δείξει ένα πολύ μεγάλο ενδιαφέρον για τους πράκτορες με συναισθήματα. Η αλήθεια είναι ότι έχουν εφαρμογή σε μία ευρεία ποικιλία περιοχών. Ωστόσο δεν είναι ακόμη ξεκάθαρο πότε αυτοί που καλούνται πράκτορες με συναισθήματα αξίζουν πάντα τις ιδιότητές τους. Με λίγα λόγια τίθεται στην καλύτερη περίπτωση ένα ανοικτό ερώτημα, αν και πότε αυτοί οι πράκτορες έχουν πραγματικά συναισθήματα. Σκοπός της παρούσας διπλωματικής εργασίας είναι να "ταιριάξει" τους Πράκτορες Διασύνδεσης (Interface Agents) με τους Πράκτορες Συναισθημάτων (Emotional Agents) ορίζοντας ένα μοντέλο το οποίο να βασίζεται σε πρωτόκολλα διαπραγμάτευσης πρακτόρων καθώς και στην ιστορία της επικοινωνίας τους, στο πλαίσιο ενός Πολυπρακτορικού Συστήματος. Οι πράκτορες αυτοί ονομάζονται εφεξής Συναισθηματικοί Πράκτορες Διασύνδεσης (ΣΠΔ)

Η μεθοδολογία που χρησιμοποιήθηκε για την ανάπτυξη του συγκεκριμένου μοντέλου των ΣΠΔ στο επίπεδο της απεικόνισης των συναισθημάτων συνδυάζεται επίσης με τα δυαδικά δέντρα αναζήτησης και τα δέντρα αποφάσεων λόγω του ότι υποστηρίζουν τις διεργασίες της αναζήτησης και της εισαγωγής σε γρήγορα σχετικά χρόνο. Για την υλοποίηση της παραπάνω μεθοδολογίας χρησιμοποιήθηκε η τεχνολογία της JAVA και η JADE η οποία είναι μία ελεύθερη πλατφόρμα αντικειμενοστραφούς ανάπτυξης λογισμικού πρακτόρων, και ουσιαστικά αποτελεί ένα σύνολο βιβλιοθηκών της JAVA για αυτό το σκοπό.

Ως μελέτη περίπτωσης για τη δοκιμή του μοντέλου που αναπτύξαμε χρησιμοποιήθηκε ένα πολυπρακτορικό σύστημα για την κατάστρωση ωρολογίου προγράμματος μαθημάτων (Πεπές, 2015). Οι ΣΠΑ στην περίπτωση αυτή χρησιμοποιούνται για την υποβοήθηση της επικοινωνίας του συστήματος με τους χρήστες (καθηγητές και φοιτητές). Τα αποτελέσματα της εφαρμογής φαίνονται να είναι τουλάχιστον ελπιδοφόρα. Έχουμε φτάσει στον πρώτο μας στόχο που ήταν να παρατηρήσουμε διαφορετικές συμπεριφορές σε διάφορες ενέργειες των πρακτόρων. Για παράδειγμα, ένας θυμωμένος και αναστατωμένος πράκτορας θα εκφραστεί νευρικά και χωρίς τον απαραίτητο σεβασμό, ενώ ένας χαρούμενος και συγκεντρωμένος πράκτορας θα προσπαθήσει ήρεμα μέσω διαβουλεύσεων να ικανοποιήσει τις απαιτήσεις των καθηγητών. Επιπλέον, παρατηρήσαμε σε ευρύτερη κλίμακα του συνολικού συστήματος, ότι υπάρχουν διαφορές τόσο στην αποδοτικότητα της δουλειάς όσο

και στη «ψυχολογική ευημερία» των πρακτόρων και αυτό έχει άμεση σχέση με τη σύνθεση των ομάδων τους. Η αρχική προσομοίωση έχει οδηγήσει ήδη σε τροποποιήσεις που ευνοούν την ολοκλήρωση των διαπραγματεύσεων της κατάστρωσης του ωρολογίου προγράμματος σε ένα «αγχωτικό» περιβάλλον.

Λέξεις Κλειδιά: <<Πράκτορες Διασύνδεσης, Πράκτορες με Συναισθήματα, Συναισθηματικοί Πράκτορες Διασύνδεσης, Πρωτόκολλα Διαπραγμάτευσης, Επικοινωνία Πρακτόρων>>

Abstract

Current research in Artificial Intelligence has shown a great interest on emotioanal agents. The truth is that they have implementations to a wide variety of areas. However it is not yet clear when agents that they are called emotional always deserve their properties. In short it is at best an open question whether and when these agents express real feelings. The purpose of this thesis is to "match" the Interface Agents with Emotional Agents in a model based on agent negotiation protocols and in the history of their interactions in the context of a Multi-Agent System. These agents will be referred here after as Emotional Interface Agents (EIA).

The methodology used for the development of the model of EIAs regarding the mapping of emotions to theme is also based on binary search trees and decision trees because they support the processes of search and insertion effectively. To implement the above methodology, JAVA technology was used and JADE, which is a open source object-oriented software development platform for developing agents, actually a set of JAVA libraries for the specific purpose.

As a case study to apply the model developed we used a Multi-Agent System for solving the Timetabling Teaching Program. The results of the application appear to be at least promising. We have reached our first goal, which was to observe different behaviors in different actions of the agents. For example, an angry and upset agent will be expressed nervously and without due respect, while a happy and focused agent will try calmly through consultation to satisfy the teachers' demands. Furthermore, we observed in larger scale of the overall system that there are differences in both the efficiency of work and the "psychological welfare" of agents and this is directly related to the composition of their teams. The simulation has already led to changes that favor the completion of negotiations in solving the timetabling problem in such a "stressful" environment.

Keywords: "Interface Agents, Emotional Agents, Negotiation Protocols, Agents' communication"

Ευχαριστίες

Θέλω αρχικά να ευχαριστήσω τον επιβλέποντα Καθηγητή Σταμάτη Δημοσθένη για τη δυνατότητα που μου έδωσε να ασχοληθώ με ένα σύγχρονο θέμα που παρουσιάζει εξαιρετικό ενδιαφέρον και την εξεταστική επιτροπή των καθηγητών που θα μελετήσουν και θα αξιολογήσουν αυτή τη διπλωματική εργασία.

Ιδιαίτερες ευχαριστίες επιβάλλεται να δώσω στον κ. Πεπέ Σωκράτη, ο οποίος μελέτησε με προσοχή τη διπλωματική μου και βοήθησε τα μέγιστα στη σωστή διαμόρφωσή της, με πολύτιμες παρατηρήσεις και υποστήριξη μέχρι την τελευταία στιγμή. Ήταν (και θα κλέψω αυτό που λέει) "ο πολύ καλός άνθρωπος" που χρειάστηκε για την περάτωση της όλης εργασίας μου.

Θα ήθελα στη συνέχεια να ευχαριστήσω θερμά τους συναδέλφους-συμφοιτητές και μερικούς ήδη πολύ καλούς φίλους (με αλφαβητική σειρά): Γρηγορίου Ελισάβετ, Μάγγου Αικατερίνη, για το ενδιαφέρον και την υποστήριξη από την αρχή των σπουδών μας μέχρι σήμερα, που οδήγησαν στην επίτευξη της διπλωματικής εργασίας. Το ενδιαφέρον και η όποια υποστήριξη "παιδιά" είναι και από μέρους μου περισσότερο από δεδομένη!

Τέλος, ίσως το σημαντικότερο, θα ήθελα να ευχαριστήσω την οικογένειά μου. Τους γονείς μου, στους οποίους οφείλω "το ζην και το ευ ζην", τους ευχαριστώ για όλες τους τις θυσίες και τις ανησυχίες τους και τα όσα μου έχουν προσφέρει για τα όμορφα φοιτητικά χρόνια ως μεταπτυχιακός πλέον φοιτητής. Εύχομαι να τους προσφέρω χαρές και ευτυχία. Ευχαριστώ πολύ και την αδελφή μου που υπέφερε συχνά από την συγκατοίκηση μαζί μου και έριξε μία τελική ματιά στο κείμενο αυτό καθώς της έχω τυφλή εμπιστοσύνη.

Η διπλωματική εργασία είναι εξαιρετικά αφιερωμένη στην οικογένειά μου και σε όσους παιδεύονται να κάνουν την Τεχνητή Νοημοσύνη λιγότερο "τεχνητή" και περισσότερο "νοημοσύνη"!

Πίνακας Περιεχομένων

Πίνακας περιεχομένων

Περίληψη	6
Abstract	8
Ευχαριστίες	10
Πίνακας Περιεχομένων	- 12 -
Πίνακας εικόνων	- 15 -
Πίνακας Πινάκων	- 17 -
1. Εισαγωγή	- 18 -
1.1 Χειρισμός των συναισθημάτων από τους πράκτορες διασύνδεσης και εφαρμογή τους σε πολυπρακτορικά υπολογιστικά συστήματα	- 18 -
1.2 Αντικείμενο της διπλωματικής	- 19 -
1.2.1 Συνεισφορά.....	- 20 -
1.3 Οργάνωση του κειμένου	- 20 -
2. Σχετικές εργασίες.....	- 22 -
2.1 Αρχιτεκτονική Μοντέλου Συναισθηματικής Λογικής.....	- 22 -
2.2 Αρχιτεκτονική Μοντέλου Συναισθημάτων.....	- 23 -
2.3 Αρχιτεκτονική Ασαφούς Λογικής Προσαρμοστικού Μοντέλου Συναισθημάτων.....	- 24 -
2.4 Κλιμακωτή Υβριδική Αρχιτεκτονική Μίμησης Συναισθημάτων	- 25 -
2.5 Αρχιτεκτονική Μοντέλου Συναισθημάτων και Προσαρμογής.....	- 26 -
2.6 Συμβουλευτική Αρχιτεκτονική πρακτόρων με συναισθήματα	- 26 -
2.7 Αρχιτεκτονική Αλληλεπίδρασης.....	- 28 -
2.8 Αρχιτεκτονική για Πολυπρακτορικά συστήματα Ηλεκτρονικής Μάθησης	- 29 -
2.9 EBDI Αρχιτεκτονική.....	- 30 -
2.10 Αρχιτεκτονική στήριξης αποφάσεων.....	- 31 -
2.11 Αρχιτεκτονική πράκτορα με συναισθήματα που παρατηρεί τις κινήσεις ενός χορευτή. Ορθολογική και Αντιδραστική	- 32 -

2.11.1	<i>Το συστατικό εισόδου (Input component)</i>	- 36 -
2.11.2	<i>Το συστατικό εξόδου (Output component)</i>	- 37 -
2.11.3	<i>Το συστατικό αντίδρασης (Reactive component)</i>	- 37 -
2.11.4	<i>Το ορθολογικό συστατικό (Rational component)</i>	- 38 -
2.11.5	<i>Το συναισθηματικό συστατικό (Emotional component)</i>	- 38 -
2.12	Αρχιτεκτονική Γενικού Πράκτορα Συναισθημάτων.....	- 39 -
3.	Συναισθηματικοί Πράκτορες Διασύνδεσης: Θεωρητικό υπόβαθρο	- 42 -
3.1	Τεχνολογία του Πράκτορα.....	- 43 -
3.2	Χαρακτηριστικά Πρακτόρων.....	- 44 -
3.3	Απαιτήσεις και Συστατικά για το σχεδιασμό ενός Πράκτορα	- 46 -
3.4	Αρχιτεκτονικές Πρακτόρων.....	- 46 -
3.5	Ιστορία των Πρακτόρων στο πλαίσιο της Τεχνητής Νοημοσύνης	- 47 -
3.6	Πολυπρακτορικά Συστήματα.....	- 49 -
3.7	Συναισθήματα στα υπολογιστικά συστήματα.....	- 52 -
3.7.1	<i>Γιατί μελετάμε τα συναισθήματα</i>	- 54 -
3.7.2	<i>Πώς ερμηνεύονται τα συναισθήματα</i>	- 54 -
3.8	Πράκτορες Διασύνδεσης.....	- 56 -
3.8.1	<i>Χαρακτηριστικά Πρακτόρων Διασύνδεσης</i>	- 59 -
3.8.2	<i>Αρχιτεκτονικές και Αλγόριθμοι</i>	- 60 -
3.8.3	<i>Αξιολόγηση των Πρακτόρων Διασύνδεσης</i>	- 61 -
3.8.4	<i>Ταξινόμηση των Πρακτόρων Διασύνδεσης</i>	- 63 -
3.9	Πράκτορες με Συναισθήματα	- 66 -
3.9.1	<i>Γιατί Πράκτορες με Συναισθήματα</i>	- 67 -
3.9.2	<i>Ο ρόλος των συναισθημάτων στους πράκτορες</i>	- 69 -
3.9.3	<i>Ιστορία των Πρακτόρων με Συναισθήματα</i>	- 71 -
3.9.4	<i>Προβλήματα των Πρακτόρων με Συναισθήματα</i>	- 72 -
3.10	Περιγραφή προβλήματος ωρολογίου προγράμματος.....	- 73 -
4.	Συναισθηματικός Πράκτορας Διασύνδεσης	- 77 -
4.1	Ανάλυση Απαιτήσεων	- 77 -
4.2	Εισαγωγή Συναισθηματικού Πράκτορα Διασύνδεσης σε Πολυπρακτορικό Σύστημα ωρών και μαθημάτων.....	- 79 -

4.3	Ανάπτυξη του Συναισθηματικού Πράκτορα Διασύνδεσης.....	- 80 -
4.4	Υλοποίηση του Συναισθηματικού Πράκτορα Διασύνδεσης.....	84
5	Συμπεράσματα – Μελλοντικές επεκτάσεις	120
	Βιβλιογραφία.....	126

Πίνακας εικόνων

Εικόνα 1 Αρχιτεκτονική Μοντέλου Συναισθημάτων (Reilly, 1996)	- 24 -
Εικόνα 2 Αρχιτεκτονική FLAME (El-Nasr, 2000).	- 25 -
Εικόνα 3 General emotional agent conceptual model. (Zitar, 2007).....	- 27 -
Εικόνα 4 Αρχιτεκτονική μοντέλου αλληλεπίδρασης πρακτόρων (Koutsabasis, 1997)	- 29 -
Εικόνα 5 Πολυπρακτορικό Σύστημα Ηλεκτρονικής Μάθησης (Ammar, 2007).....	- 30 -
Εικόνα 6 EBDI Αρχιτεκτονική (Jiang & Vidal, 2007).....	- 31 -
Εικόνα 7 Αρχιτεκτονική Μοντέλου Στήριξης Αποφάσεων (Daviet & Desmier, 2005).....	- 32 -
Εικόνα 8 Παρατήρηση κινήσεων και χειρονομιών του χορευτή (Camurri, 1999).....	- 33 -
Εικόνα 9 Δομή ενός πράκτορα με συναισθήματα (Camurri & Ferrentino, 1999)	- 35 -
Εικόνα 10 Παραδείγματα δύο συναισθηματικών χώρων (Camurri & Ferrentino, 1999) ...	- 39 -
Εικόνα 11 Αρχιτεκτονική του GEMA (Kazemifard, 2011)	- 40 -
Εικόνα 12 OCC Model (Zitar, 2007).....	- 41 -
Εικόνα 13 Πράκτορας Λογισμικού σε Πολυπρακτορικό Σύστημα.....	- 50 -
Εικόνα 14 Πολυπρακτορικό Σύστημα.....	- 51 -
Εικόνα 15 Συναισθήματα στα υπολογιστικά συστήματα	- 52 -
Εικόνα 16 Συναισθήματα στα υπολογιστικά συστήματα	- 53 -
Εικόνα 17 Πράκτορες Διασύνδεσης.....	- 57 -
Εικόνα 18 Συναισθήματα στους πράκτορες	- 67 -
Εικόνα 19 Πράκτορας Διασύνδεσης (Ammar, 2007).....	- 69 -
Εικόνα 20 Ρόλος των συναισθημάτων στους Πράκτορες	- 71 -
Εικόνα 21 Πολυπρακτορικό μοντέλο για ωρολόγιο πρόγραμμα Πανεπιστημίου	- 76 -
Εικόνα 22 Δυαδικό Δένδρο Απόφασης.....	- 82 -
Εικόνα 23 Μοντέλο ΣΠΔ με Χρήση Δυαδικού Δένδρου Αναζήτησης.....	- 83 -
Εικόνα 24 Περιβάλλον ανάπτυξης Eclipse	86
Εικόνα 25 Eclipse_Workspace.....	86
Εικόνα 26 Άνοιγμα του Eclipse.....	87
Εικόνα 27 Κεντρικό παράθυρο περιβάλλοντος Eclipse	87
Εικόνα 28 Διαμορφώσεις εκτέλεσης.....	88
Εικόνα 29 Εισαγωγή ονόματος του project.....	88
Εικόνα 30 Ορισμός κύριας κλάσης	89

Εικόνα 31 Ορισμός Ιδιοτήτων προγράμματος	90
Εικόνα 32 Προσθήκη βιβλιοθηκών προγράμματος.....	91
Εικόνα 33 Μέθοδος Preorder Traversal	101
Εικόνα 34 Επεξεργασία εικόνων με mywebface.com.....	108
Εικόνα 35 Συναισθηματικός πράκτορας διασύνδεσης (ΣΠΔ).....	110
Εικόνα 36 Μελλοντικές προεκτάσεις στους BDI Πράκτορες (Vidal, 2007).....	123
Εικόνα 37 Αναγνώριση συναισθημάτων σε Κινητά τηλέφωνα	124
Εικόνα 38 Μελλοντικές προεκτάσεις σε Πολυπρακτορικά συστήματα.....	125

Πίνακας Πινάκων

Πίνακας 1 Σχέσεις μεταξύ των συστατικών (Camurri & Ferrentino, 1999) - 36 -

Πίνακας 2 Δεδομένα που μεταφέρουν οι πράκτορες (Camurri & Ferrentino, 1999)..... - 37 -

1. Εισαγωγή

1.1 Χειρισμός των συναισθημάτων από τους πράκτορες διασύνδεσης και εφαρμογή τους σε πολυπρακτορικά υπολογιστικά συστήματα

Ο όρος συναισθήματα μπορεί να παρερμηνευτεί στα υπολογιστικά συστήματα. Είναι προφανές ότι οι υπολογιστές δεν μπορούν να αισθανθούν όπως εμείς, οι άνθρωποι. Στην πραγματικότητα οι υπολογιστές δεν αισθάνονται. Άλλα όταν αλληλεπιδρούμε με αυτούς, εκφράζουμε συναισθήματα ακόμη και όταν ξέρουμε ότι δεν βρίσκεται κανείς δίπλα μας για να δει ή να νιώσει τις αντιδράσεις μας. Στη συγκεκριμένη εργασία απομονώνονται και παρουσιάζονται μερικοί από τους λόγους για τους οποίους οι πράκτορες με συναισθήματα είναι επιθυμητοί και επισημαίνεται η δυσκολία ακριβούς κατανόησης των συναισθημάτων.

Υπάρχουν πολλές θεωρίες συναισθηματισμού. Ο Sloman το 1987 (Sloman, 1987) πρότεινε ένα αρχιτεκτονικό πλαίσιο για την κατανόηση των συναισθηματικών καταστάσεων, το οποίο στηριζόταν σε μερική απώλεια του ελέγχου της προσοχής. Το 1994 ο Beaudoin (Beaudoin, 1994) σχεδίασε μία αρχιτεκτονική πράκτορα που όταν εφαρμοστεί θα πρέπει να επεκταθούν οι συναισθηματικές καταστάσεις. Το 1986 ο Smith (Smith, 1986) σχεδίασε ένα αυτό-αναφορικό σύστημα που βασίζεται στη θεωρία του εαυτού και εφαρμόστηκε στον πραγματικό κόσμο. Ως εκ τούτου, θα εφαρμοστεί μία πρωτότυπη αρχιτεκτονική πράκτορα που θα είναι αυτό-αναφορική και θα μπορεί να ανιχνεύει συναισθηματικές καταστάσεις και να προτείνει μία λύση για το πρόβλημα. Τα αποτελέσματα της έρευνας αυτής θα συμβάλουν στο σχεδιασμό ενός αυτόνομου πράκτορα, στη συναισθηματικότητα και στην εσωτερική αντίληψη.

Σε δυναμικούς και απρόβλεπτους τομείς όπως είναι ο πραγματικός κόσμος, οι πράκτορες έρχονται συνέχεια αντιμέτωποι με νέες απαιτήσεις και περιορισμούς σχετικά με την ποιότητα και τις λύσεις που παράγουν. Οποιοσδήποτε σχεδιασμός πράκτορα θα περιορίζεται πάντα με κάποιο τρόπο. Τέτοιες σκέψεις επισημαίνουν την ανάγκη για αυτό-αναφορικό μηχανισμό, δηλαδή πράκτορες με την ικανότητα να εξετάζουν την εσωτερική τους διαδικασία έτσι ώστε να ελέγχουν και να βελτιώνουν τη λειτουργικότητά τους. Μερικές συναισθηματικές καταστάσεις όπως η εξουθένωση ή η θλίψη θα μπορούσαν να θεωρηθούν ως «διασπαστικές»

απαντήσεις στις νέες συνθήκες.. Η έρευνα αυτή επιθυμεί να διερευνήσει την παραγωγή τέτοιων συναισθηματικών καταστάσεων, τον εντοπισμό τους μέσω των μηχανισμών που προαναφέραμε και πιθανές στρατηγικές αντιμετώπισης, πρόληψης ή λύσης στο πλαίσιο μίας ευρείας αρχιτεκτονικής πράκτορα.

1.2 Αντικείμενο της διπλωματικής

Η σημασία του συναισθήματος έχει ταυτοποιηθεί πρόσφατα με την ανθρώπινη ευφυΐα. Για να βελτιωθεί η ικανότητα ενός πράκτορα να διαχειρίζεται συναισθήματα, προτείνεται μία αρχιτεκτονική που παντρεύει τους πράκτορες διασύνδεσης με τους συναισθηματικούς πράκτορες, οι οποίοι επιτελούν ρόλο στο πλαίσιο ενός πολυπρακτορικού συστήματος. Το μοντέλο των εν λόγω Συναισθηματικών Πρακτόρων Διασύνδεσης (ΣΠΔ) είναι βασισμένο στα πρωτόκολλα διαπραγμάτευσης καθώς και στην ιστορία της επικοινωνίας των πρακτόρων του πολυπρακτορικού συστήματος. Αρχικά καθορίζεται μία νέα βάση γνώσεων για την έκφραση συναισθημάτων, στη συνέχεια εκτελείται ένας αλγόριθμος συναισθηματικής λογικής και τέλος εφαρμόζεται η «θεραπεία» συναισθημάτων. Τα αποτελέσματα από πειράματα που έγιναν πάνω σε έναν πράκτορα βασισμένο στη συγκεκριμένη αρχιτεκτονική, έδειξαν ότι είναι περισσότερο αποτελεσματικό να υπάρχουν απλά συναισθήματα μεταξύ των πρακτόρων συναισθημάτων. Στο παρελθόν τα συναισθήματα απορρίφθηκαν γιατί θεωρήθηκε ότι αποσπούν την προσοχή σε κάποιον που προσπαθεί να σκεφτεί λογικά και επιστημονικά κατά τη διάρκεια της επικοινωνίας του με ένα σύστημα που χρησιμοποιεί. Πιο πρόσφατα ωστόσο, η σημασία των συναισθημάτων στην ανθρώπινη νοημοσύνη έχει αναγνωριστεί (Turing, 1950). Η έρευνα πάνω στους πράκτορες συναισθημάτων βοηθάει ώστε να αποδειχθούν τα έξυπνα χαρακτηριστικά των πρακτόρων (McCarthy, 1960). Στη ψυχολογία, το συναίσθημα εξηγείται ως οι αξιολογήσεις των πρακτόρων ξεχωριστά ή της σχέσης των πρακτόρων με το περιβάλλον. Η συναισθηματική αντίδραση των ανθρώπων προέρχεται από τη διέγερση του εξωτερικού περιβάλλοντος και επηρεάζεται από το μηχανισμό προσφοράς και ζήτησης που έχουν στον οργανισμό τους. Από την οπτική γωνία του πράκτορα, τα συναισθήματα είναι αξιολογήσεις και αντιδράσεις για την τρέχουσα εσωτερική του κατάσταση. Ως εκ τούτου, γίνεται προσπάθεια να βρεθεί κάποια αρχιτεκτονική που θα βοηθάει τους πράκτορες να χειρίζονται τα συναισθήματά τους. Αρχικά λοιπόν μελετήσαμε τους συναισθηματικούς πράκτορες, τα πεδία εφαρμογής τους καθώς και τις αρχιτεκτονικές των πρακτόρων που προσπαθούν να ανιχνεύσουν συναισθηματικές καταστάσεις και αποφασίσαμε να αναφερθούμε στα συναισθήματα της χαράς, της θλίψης, της απορίας, του ενθουσιασμού, του εκνευρισμού και της απογοήτευσης σε διάφορα επίπεδα. Επικεντρωθήκαμε στα μοντέλα έκφρασης και καταγραφής του ιστορικού των

συναισθημάτων και πειραματιστήκαμε με ένα συγκεκριμένο πολυπρακτορικό σύστημα ωρών και μαθημάτων καθώς και με το πώς θα εισαχθεί ο πράκτορας με τα συναισθήματα στο συγκεκριμένο σύστημα ως πράκτορας διασύνδεσης.

1.2.1 Συνεισφορά

Η συνεισφορά της διπλωματικής έρευνας συνοψίζεται ως εξής:

1. Μελετήσαμε τους πράκτορες διασύνδεσης και τους πράκτορες με συναισθήματα.
2. Επικεντρωθήκαμε στο μοντέλο έκφρασης και καταγραφής του ιστορικού των συναισθημάτων.
3. Υλοποιήσαμε έναν αλγόριθμο που βασίστηκε στα πρωτόκολλα διαπραγμάτευσης πολυπρακτορικών συστημάτων καθώς και στα δέντρα αποφάσεων για την απεικόνιση συναισθημάτων του πράκτορα.
4. Αξιολογήσαμε την επίδοση του αλγορίθμου και καταλήξαμε στο συμπέρασμα ότι ανάλογα με το βάρος του κριτηρίου που δέχεται ο αλγόριθμος, εμφανίζει και την αντίστοιχη εικόνα έκφρασης συναισθήματος.
5. Πειραματιστήκαμε με το ρόλο του ΣΠΔ στο πλαίσιο ενός πολυπρακτορικού συστήματος που χρησιμοποιείται για την κατάστρωση προγράμματος ωρών και μαθημάτων.
6. Ενσωματώσαμε τον αλγόριθμο και τον πράκτορα με συναισθήματα στο συγκεκριμένο πολυπρακτορικό σύστημα.

1.3 Οργάνωση του κειμένου

Η ανάπτυξη πρακτόρων διασύνδεσης με συναισθήματα είναι ο τομέας της Τεχνητής Νοημοσύνης στον οποίο επικεντρώνεται η παρούσα διπλωματική εργασία. Αρχικά γίνεται μία εποπτεία στις έννοιες των πρακτόρων, την τεχνολογία που χρησιμοποιούν, τις αρχιτεκτονικές και τα χαρακτηριστικά τους. Στη συνέχεια μελετάμε τα συναισθήματα στα υπολογιστικά συστήματα, για ποιο λόγο είναι χρήσιμα και πώς ερμηνεύονται. Επιπλέον, γίνεται μία πιο αναλυτική περιγραφή των εννοιών των πρακτόρων διασύνδεσης και των συναισθηματικών πρακτόρων, μία αναφορά στην ταξινόμηση και στην αξιολόγησή τους, ποιοι είναι οι βασικοί λόγοι που τους χρησιμοποιούμε καθώς και μερικά από τα προβλήματα που παρουσιάζουν. Στη συνέχεια, γίνεται μία επισκόπηση του πεδίου εφαρμογών πρακτόρων με συναισθήματα σε περιβάλλοντα μάθησης. Ακολουθεί η υλοποίηση ενός πράκτορα με συναισθήματα ο οποίος συνεργάζεται με έναν παιδαγωγικό πράκτορα και σύμφωνα με τις εισόδους που δέχεται αντιδράει ανάλογα και εμφανίζει τις κατάλληλες εξόδους που είναι

ουσιαστικά συναισθήματα που νιώθει. Τέλος, παρουσιάζεται η υλοποίηση της αρχιτεκτονικής με τη χρήση της JADE πλατφόρμας και της γλώσσας προγραμματισμού JAVA, όπου και περιέχονται οι λειτουργίες, τα σενάρια χρήσης και οι ανάλογες αντιδράσεις του πράκτορα.

2. Σχετικές εργασίες

Στην Τεχνητή Νοημοσύνη, η λέξη πράκτορας χρησιμοποιείται ετερόκλητα για να αναφερθεί σε όλα τα είδη των οντοτήτων, όπως επίμονες διεργασίες, αντιδραστικούς ελέγχους, προγράμματα που ελέγχουν μέρη του πραγματικού ή του κόσμου προσομοίωσης και απλούς προσαρμοστικούς αλγορίθμους. Υπάρχουν το λιγότερο δύο χρήσεις της λέξης αρχιτεκτονική. Η μία αναφέρεται στο σχεδιασμό και η άλλη στο να κάνει πιο συμπαγείς τις περιπτώσεις του σχεδιασμού. Η αρχιτεκτονική ενός πολύπλοκου συστήματος μπορεί να εξηγήσει πώς οι δυνατότητες και η συμπεριφορά του προκύπτουν από τις δυνατότητες, τη συμπεριφορά, τις σχέσεις και τις αλληλεπιδράσεις των συστατικών του. Μία αρχιτεκτονική μπορεί να καθοριστεί σε διάφορα επίπεδα πληροφοριών. Η αρχιτεκτονική ενός πράκτορα ωστόσο, είναι μία περιγραφή των μηχανισμών επεξεργασίας πληροφοριών ενός συστήματος ελέγχου σε ένα συγκεκριμένο επίπεδο. Υπάρχουν πολλά παραδείγματα αρχιτεκτονικών πρακτόρων στη βιβλιογραφία. Σε αυτή τη διατριβή θα εξετάσουμε τις απαιτήσεις των αυτόνομων πρακτόρων που έχουν συναισθήματα.

Φυσιολογικά, οι σχεδιαστές των πρακτόρων δεν περιμένουν να ικανοποιήσουν όλες τις απαιτήσεις των χρηστών αλλά ένα υποσύνολό τους. Η ποικιλία των απαιτήσεων βασίζεται στους στόχους των ερευνών. Για παράδειγμα, οι σχεδιαστές των αρχιτεκτονικών αντίδρασης προσπαθούν να ικανοποιήσουν απαιτήσεις για γρήγορες και άμεσες απαντήσεις στα απρόβλεπτα του περιβάλλοντος. Οι σχεδιαστές των συμβουλευτικών αρχιτεκτονικών προσπαθούν να ικανοποιήσουν απαιτήσεις επίλυσης προβλημάτων και σχεδιασμού ικανοτήτων και οι σχεδιαστές του συνδυασμού των δύο παραπάνω αρχιτεκτονικών προσπαθούν να ενσωματώσουν τις ικανότητες και των δύο σε ένα απλό πλαίσιο ελέγχου. Παρακάτω θα αναφερθούν ορισμένα από τα μοντέλα και οι αρχιτεκτονικές που προσομοιώνουν τη διαδικασία των συναισθημάτων ανάλογα και με την χρονική περίοδο που εμφανίστηκαν.

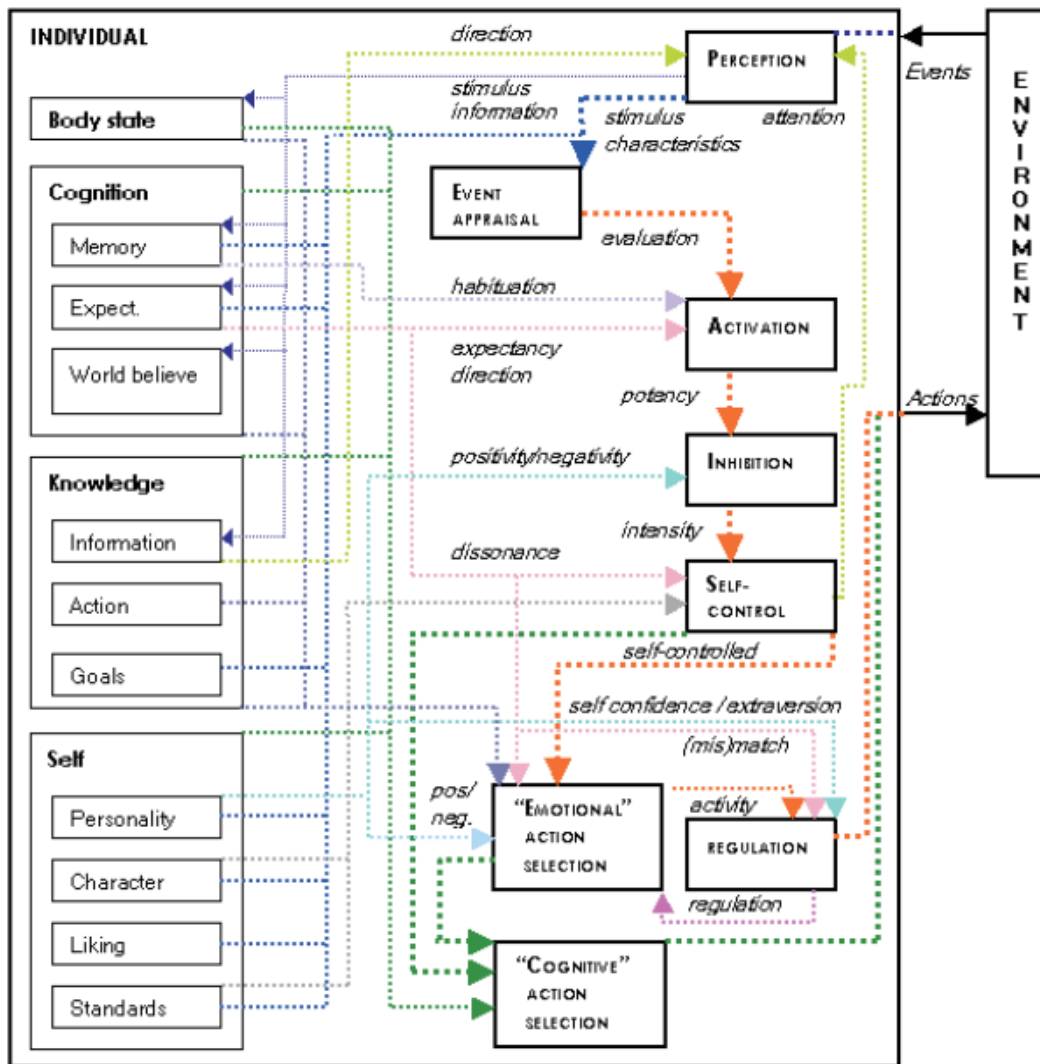
2.1 Αρχιτεκτονική Μοντέλου Συναισθηματικής Λογικής

Τα μοντέλα συναισθηματικής λογικής (Affective Reasoner, AR) μοντελοποιούν ένα κόσμο πολλαπλών πρακτόρων και παρέχουν μία απλή συναισθηματική ζωή στους πράκτορες με τη μορφή των βασικών συναισθημάτων, των συναισθημάτων που προκαλούνται από ενέργειες

και των βασικών προσωπικοτήτων. Οι πράκτορες είναι σε θέση να βιώσουν 14 τύπους συναισθημάτων και 1400 ενέργειες που προκαλούνται από αυτά τα συναισθήματα. Οι συνθήκες κάτω από τις οποίες προκλήθηκαν αυτά τα συναισθήματα είναι κάθε φορά διαφορετικές. Κάθε συνθήκη έχει μία διακριτή τιμή η οποία αντιπροσωπεύει το αν είναι αληθής ή ψευδής. Για παράδειγμα, η χαρά ενεργοποιείται όταν πραγματοποιηθεί ένα επιθυμητό γεγονός. Το τμήμα της διάθεσης που επηρεάζει την αξιολόγηση του πράκτορα, υλοποιείται στην εργασιακή μνήμη των πρακτόρων. Όλοι οι πράκτορες έχουν προσωπικότητες που χωρίζονται σε δύο μέρη. Το ένα είναι το ερμηνευτικό τους κομμάτι που ερμηνεύει τον κόσμο και οδηγεί σε διαφορετικές συναισθηματικές αντιδράσεις σε διάφορες καταστάσεις που προκύπτουν και το άλλο είναι το εκφραστικό τους κομμάτι που ελέγχει πως εκφράζονται τα συναισθήματα. Αυτό το μοντέλο απαιτεί έναν αριθμό από ειδικούς κανόνες για την εκτίμηση των γεγονότων (Elliott, 1992) .

2.2 Αρχιτεκτονική Μοντέλου Συναισθημάτων

Η αρχιτεκτονική συναισθημάτων (EM Architecture) χρησιμοποιείται σε διάφορα σχέδια για τα συναισθήματα και την κοινωνική συμπεριφορά. Σκοπός του μοντέλου αυτού (Εικόνα 1) ήταν να παρέχει στους χρήστες την εμπειρία του να ζεις σε δραματικά ενδιαφέρουσες κοινωνίες που περιλαμβάνουν αρμόδιους πράκτορες συναισθημάτων. Κάθε πράκτορας έχει αρχικά προκαθορισμένη στάση απέναντι στα αντικείμενα του περιβάλλοντος. Επιπλέον, κάθε πράκτορας έχει κάποιους αρχικούς στόχους και μία σειρά από στρατηγικές που μπορεί να χρησιμοποιήσει για την επίτευξη ενός συγκεκριμένου στόχου. Ο πράκτορας αισθάνεται ή αντιλαμβάνεται ένα γεγονός στο περιβάλλον. Το πόσο επιθυμητό είναι αυτό το γεγονός, συνδέεται άμεσα με την πιθανότητα επίτευξης του στόχου σε ένα αντιδραστικό σχεδιαστικό πλαίσιο (Gratch & Marsella, 2004). Αυτό το μοντέλο βασίζεται πάνω σε ειδικούς κανόνες για να πετύχει τους στόχους του σύμφωνα με τις προσδοκίες του (Reilly, 1996).



Εικόνα 1 Αρχιτεκτονική Μοντέλου Συναισθημάτων (Reilly, 1996)

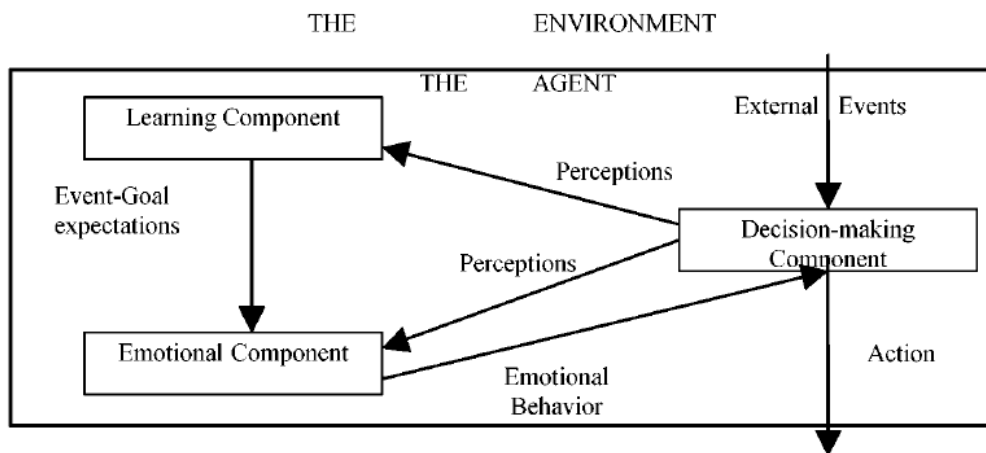
2.3 Αρχιτεκτονική Ασαφούς Λογικής Προσαρμοστικού Μοντέλου Συναισθημάτων

Το ασαφούς λογικής προσαρμοστικό μοντέλο συναισθημάτων FLAME (Εικόνα 2) είναι ένα υπολογιστικό μοντέλο που βασίζεται στην εκτίμηση των γεγονότων. Ενσωματώνει ορισμένα στοιχεία μάθησης για να προσαρμόσει πιο εύκολα τα συναισθήματα. Προτείνει μία ασαφή λογική για να χειριστεί τις ακόλουθες έννοιες:

- Ο ασαφής στόχος εισάγει ένα βαθμό επιτυχίας και αποτυχίας έτσι ώστε να επιτύχει συγκεκριμένους στόχους.
- Σύμφωνα με τα ασαφή μοντέλα μελών, πόσο ταιριάζει ένα γεγονός σε έναν στόχο;

- Η ασαφής "χαρτογράφηση" καθορίζεται ανάλογα με το βαθμό με τον οποίο επηρεάζει ένα γεγονός ορισμένους στόχους καθώς και τη σημασία των στόχων αυτών.

Το μοντέλο χρησιμοποιεί μία μέθοδο εκμάθησης για τα πρότυπα συμπεριφοράς του χρήστη έτσι ώστε να μπορεί να υπολογιστεί η αναμενόμενη έκβαση ενός γεγονότος. Το FLAME μοντέλο χρησιμοποιεί διαδικασίες απόφασης Markov (Markov-decision processes, MDPs) για να υπολογίσει τη σκοπιμότητα των ενεργειών και των δράσεων. Ένα μικρό πρόβλημα που παρουσιάζεται είναι ότι το συγκεκριμένο μοντέλο μπορεί να εκπροσωπήσει ένα μικρό αριθμό καταστάσεων και να αναλάβει καθορισμένους στόχους. Με άλλα λόγια, αν ο τομέας έχει έναν μεγάλο αριθμό στόχων και ενεργειών, τότε χρειάζεται ένας αντίστοιχα μεγάλος αριθμός ασαφών κανόνων. Αυτό μπορεί να αποτελέσει ένα εμπόδιο στη διαμόρφωση του μοντέλου (El-Nasr, 2000).



Εικόνα 2 Αρχιτεκτονική FLAME (El-Nasr, 2000).

2.4 Κλιμακωτή Υβριδική Αρχιτεκτονική Μίμησης Συναισθημάτων

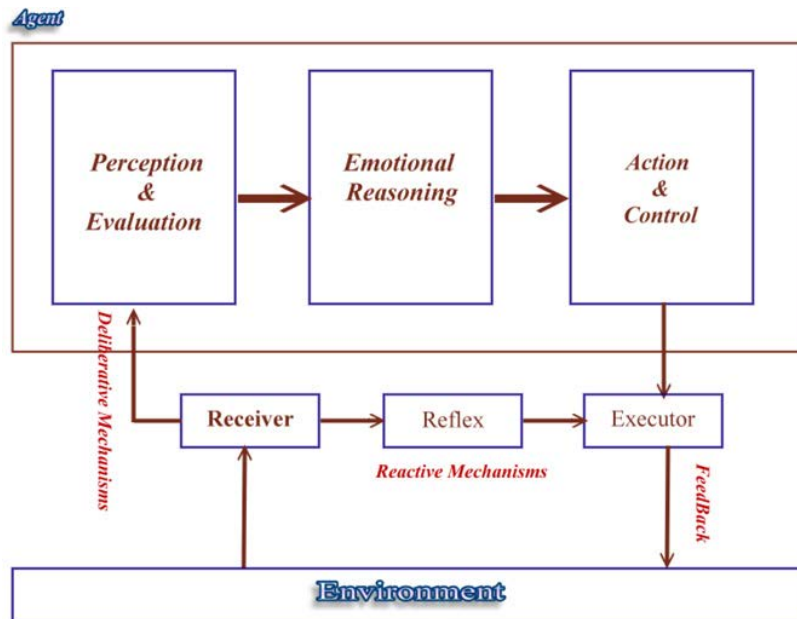
Η κλιμακωτή υβριδική αρχιτεκτονική μίμησης συναισθημάτων SHAME είναι ένα προσεγγιστικό μοντέλο αξιολόγησης των γεγονότων που βασίζεται στα νευρωνικά δίκτυα. Τα δίκτυα αυτά εκπαιδεύονται έτσι ώστε να καταλαβαίνουν τα συναισθήματα που προκαλούνται από αυτά τα γεγονότα. Κάθε αξιολογητής εκπαιδεύεται για συγκεκριμένο τομέα και εξαρτάται αποκλειστικά από αυτόν. Η παροχή δεδομένων από το περιβάλλον δεν είναι εύκολη. Η "ένταση" του κάθε συναισθήματος (μεταξύ 0 και 100) προσδιορίζεται με βάση τις μεταβλητές του τομέα και δε χρησιμοποιεί γενικούς κανόνες για τον υπολογισμό. Για παράδειγμα, όταν ένας πράκτορας με SHAME αρχιτεκτονική παίρνει ένα μήλο, η ποσότητα του συναισθήματος της χαράς ισούται με 100 μείον την ποσότητα των τροφίμων (Kazemifard, 2011).

2.5 Αρχιτεκτονική Μοντέλου Συναισθημάτων και Προσαρμογής

Η αρχιτεκτονική συναισθημάτων και προσαρμογής (Emotion and Adaption Architecture/EMA) είναι ουσιαστικά μία επέκταση του μοντέλου της συναισθηματικής λογικής (Gratch, 2003). Η EMA αρχιτεκτονική έχει αναπτύξει γενικούς και ανεξάρτητους από το πεδίο αλγορίθμους για να υποστηρίξει την εκτίμηση των γεγονότων. Για τον υπολογισμό της έντασης των συναισθημάτων χρησιμοποιούνται δύο παράμετροι, η πιθανότητα επίτευξης των στόχων και η σημασία τους. Η κύρια συνεισφορά του μοντέλου είναι να δείξει πως οι τεχνικές σχεδιασμού θεωρητικών αποφάσεων μπορούν να υπολογίσουν την επιρροή των γεγονότων στους στόχους και στις αξιολογήσεις. Παρέχει επίσης ένα μοντέλο αντιγραφής των διαδικασιών. Η EMA αρχιτεκτονική ενημερώνεται και αναπτύσσεται σύμφωνα με τις πρόσφατες εξελίξεις του μοντέλου. Το μοντέλο αυτό περιλαμβάνει γρήγορες και αργές συναισθηματικές αντιδράσεις και προσομοιώνει μόνο έξι συναισθήματα, την χαρά, την αγωνία, την ελπίδα, το φόβο, το θυμό και την ενοχή (Gratch, 2003).

2.6 Συμβουλευτική Αρχιτεκτονική πρακτόρων με συναισθήματα

Οι σύντομες και συμβουλευτικές αρχιτεκτονικές πρακτόρων είναι προγράμματα που εφαρμόζουν τον επαγωγικό συλλογισμό για να επιλύσουν προβλήματα όπως τα κλασικά συστήματα σχεδιασμού (π.χ. STRIPS, NOAH, NONLIN). Τέτοια συστήματα μπορούν εύκολα να θεωρηθούν πράκτορες στην προσπάθειά τους να ελέγξουν μέρη του πραγματικού κόσμου. Αυτές οι αρχιτεκτονικές εργάζονται σε μία διεργασία τη φορά, δεν συνδυάζουν τη συμβουλευτική με τη δράση και διεξάγουν έρευνα για να βρουν λύση στα διάφορα προβλήματα που προκύπτουν. Μία τέτοια αρχιτεκτονική είναι γνωστή ως εννοιολογικό μοντέλο τριπλού πύργου. Ο εκτελεστής είναι υπεύθυνος για την τελική ενέργεια ή συμπεριφορά μέσω ανατροφοδότησης για το περιβάλλον. Ο δέκτης και ο εκτελεστής εξαρτώνται από το πεδίο έτσι ώστε ο σχεδιαστής του πράκτορα να είναι ο βασικός υπεύθυνος για την περιγραφή τους και την εφαρμογή τους. Το συγκεκριμένο μοντέλο αποτελείται από αντιδραστικούς και συμβουλευτικούς μηχανισμούς. Το μονοπάτι που δίνει την πληροφορία για τα συναισθήματα φαίνεται στο παρακάτω σχήμα (Εικόνα 3) (Zitar, 2007).



Εικόνα 3 General emotional agent conceptual model. (Zitar, 2007)

Οι μηχανισμοί αντίδρασης που αποτελούν το πιο χαμηλό επίπεδο πληροφοριών συνδέουν άμεσα τους δέκτες με τους εκτελεστές. Αντίστοιχα, οι μηχανισμοί διαβούλευσης που αποτελούν το υψηλότερο επίπεδο πληροφοριών, εκτελούν δύο διαδικασίες: τη γνωστική και τη συναισθηματική διεργασία. Η γνωστική διεργασία είναι υπεύθυνη για την αντίληψη, την αξιολόγηση, τη λογική και τη λήψη αποφάσεων. Η συναισθηματική διεργασία δημιουργεί σήματα συναισθημάτων ανάλογα με τα αποτελέσματα των αξιολογήσεων της πρώτης. Η εκτίμηση των πρακτόρων για τα συναισθήματα προκαλεί κάποιες μεταβλητές που εφαρμόζονται σε συναισθήματα που βασίζονται σε γεγονότα, σε συναισθήματα απόδοσης και σε συναισθήματα έλξης. Η γνωστική και η συναισθηματική διεργασία αλληλεπιδρούν μεταξύ τους. Οι συμπεριφορές μας δημιουργούνται στο πλαίσιο αντίστοιχων μηχανισμών λήψης αποφάσεων στην περίπτωση της πραγματικής ζωής.

Οι μηχανισμοί αντίδρασης απαιτούνται για να αποφευχθεί ένα ξαφνικό εμπόδιο και για να υποστηρίξουν την επιβίωση του πράκτορα. Οι μηχανισμοί λήψης αποφάσεων αξιολογούν μέσω της γνωστικής διαδικασίας και παρέχουν μία πιο ακριβή αναπαράσταση του πράκτορα. Ωστόσο, ένας τέτοιος μηχανισμός παίρνει λίγο περισσότερο χρόνο για να αποφασίσει ποια είναι η σωστή συμπεριφορά που πρέπει να ληφθεί από τον πράκτορα. Από την άλλη μεριά, ένας μηχανισμός αντίδρασης παρέχει το γρηγορότερο μονοπάτι για άμεσες αντιδράσεις του πράκτορα. Το εννοιολογικό μοντέλο ενός πράκτορα συναισθημάτων αποτελείται από δύο τύπους περιβάλλοντος:

- **Γενετικό περιβάλλον.** Ο πράκτορας λαμβάνει αρχικές καταστάσεις της μνήμης του και κάποιες άλλες αρχικές παραμέτρους για τη συμπεριφορά του.

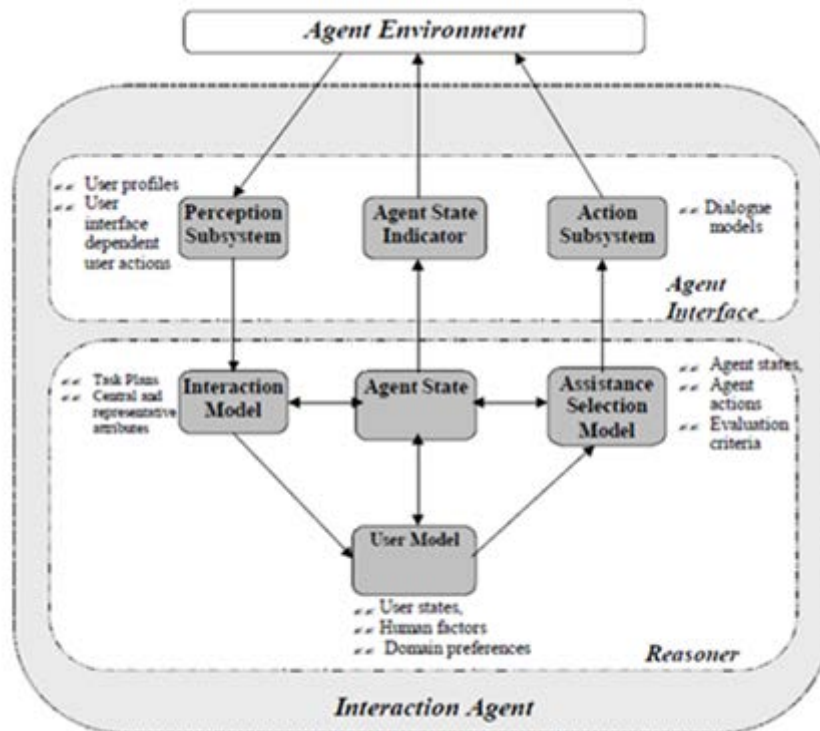
-
- **Περιβάλλον συμπεριφοράς.** Μπορεί να αλλάξει λόγω της συμπεριφοράς του πράκτορα ή λόγω κάποιων άλλων επιρροών όπως προσομοίωση ή αλληλεπίδραση του χρήστη.

Για παράδειγμα, ένα τέτοιο σύστημα σχεδιασμού μπορεί να φτιάξει ένα πλάνο για να πετύχει κάποιους στόχους ερευνώντας για τη λίστα των πρώτων ενεργειών που μετατρέπει ένα αρχικό στάδιο σε τελικό στάδιο επίτευξης (Zitar, 2007).

2.7 Αρχιτεκτονική Αλληλεπίδρασης

Τα βασικά μοντέλα πάνω στα οποία βασίζεται η συγκεκριμένη αρχιτεκτονική είναι το μοντέλο αναπαράστασης της γνώσης για την αλληλεπίδραση, το μοντέλο του χρήστη και το μοντέλο της επιλογής χρήσιμης βοήθειας. Αυτά τα τρία συστατικά είναι εσωτερικά για τον χρήστη. Υπάρχουν βέβαια πρόσθετα στοιχεία όπως η διεπαφή του χρήστη τα οποία πρέπει να εισαχθούν στο περιβάλλον του λογισμικού για να μπορέσει ο πράκτορας να τα λάβει αλλά και να τα επηρεάσει. Έμφαση δίνεται στη διερεύνηση των μηχανισμών με τους οποίους θα μπορέσει να ολοκληρωθεί η διεπαφή του χρήστη σύμφωνα με τη λειτουργικότητα του πράκτορα. Η συγκεκριμένη αρχιτεκτονική αντιπροσωπεύει τις σχέσεις των τριών βασικών μοντέλων καθώς και την εκπροσώπηση των στοιχείων της διεπαφής του χρήστη. Μία γραφική απεικόνιση της αρχιτεκτονικής παρουσιάζεται στο παρακάτω σχήμα (Εικόνα 4).

Ο πράκτορας προσπαθεί να προσαρμόζεται συνεχώς στα χαρακτηριστικά του χρήστη που παρατηρούνται κυρίως στο περιβάλλον διεπαφής του. Επίσης το υποκείμενο σύστημα διαμεσολάβησης καθορίζεται από μία στατική εφαρμογή του προγράμματος διεπαφής, το οποίο χρησιμοποιείται μόνο σε περιπτώσεις αυτόνομων λειτουργιών πράκτορα. Το να μαθαίνει κάποιος από την εμπειρία αποτελεί ένα βασικό χαρακτηριστικό ενός προσαρμοστικού και ευφυούς συστήματος. Τα συστατικά της αρχιτεκτονικής του πράκτορα αλληλεπίδρασης είναι σχεδιασμένα κατά τέτοιο τρόπο που να στηρίζουν την εκμάθηση και την προσαρμογή. Ωστόσο, υπάρχουν και χαρακτηριστικά των μοντέλων αυτών που εξαρτώνται από τον τομέα και τα οποία είναι στατικά. Αυτή είναι η «κατάρα» του κάθε έξυπνου συστήματος. Το αντιπροσωπευτικό μοντέλο πρέπει να ενσωματώσει κάποιες μορφές της διεργασίας (Koutsabasis, 1997).



Εικόνα 4 Αρχιτεκτονική μοντέλου αλληλεπίδρασης πρακτόρων (Koutsabasis, 1997)

2.8 Αρχιτεκτονική για Πολυπρακτορικά συστήματα Ηλεκτρονικής Μάθησης

Σύμφωνα με τον Ben Ammar στην κατασκευή των ενσωματωμένων πρακτόρων που ήταν ικανοί για εκφραστικές και επικοινωνιακές συμπεριφορές σε περιβάλλον διαδικτυακής μάθησης, σημαντικό ρόλο έπαιξε η ικανότητα να παράσχει λεπτομέρειες και ομιλητικές εκφράσεις του προσώπου σε σύνθετα πρόσωπα. Για παράδειγμα, ένας κινούμενος πράκτορας διασύνδεσης χρησιμοποιείται τώρα σε ένα ευρύ φάσμα εφαρμογών όπως οι προσωπικοί βοηθοί, το ηλεκτρονικό εμπόριο, η ψυχαγωγία και η ηλεκτρονική μάθηση. Τα συναισθήματα είναι ιδιαίτερα σημαντικό κομμάτι για έναν τέτοιο πράκτορα τη στιγμή που καλύπτουν μεγάλο μέρος της ομιλίας. Ο William James αντιλαμβάνεται τα συναισθήματα ως μία άμεση απάντηση στην αντίληψη ενός γεγονότος που συμβάλλουν στην επιβίωση του ατόμου και επιμένει στις αλλαγές που προκαλούνται από τη συμπεριφορά του σώματος στο άτομο. Το σώμα απαντά αρχικά με έναν προγραμματισμένο τρόπο σε αυτές τις αλλαγές και αυτό αποκαλείται συναίσθημα σύμφωνα με την προτεινόμενη αρχιτεκτονική (Ammar, 2007).



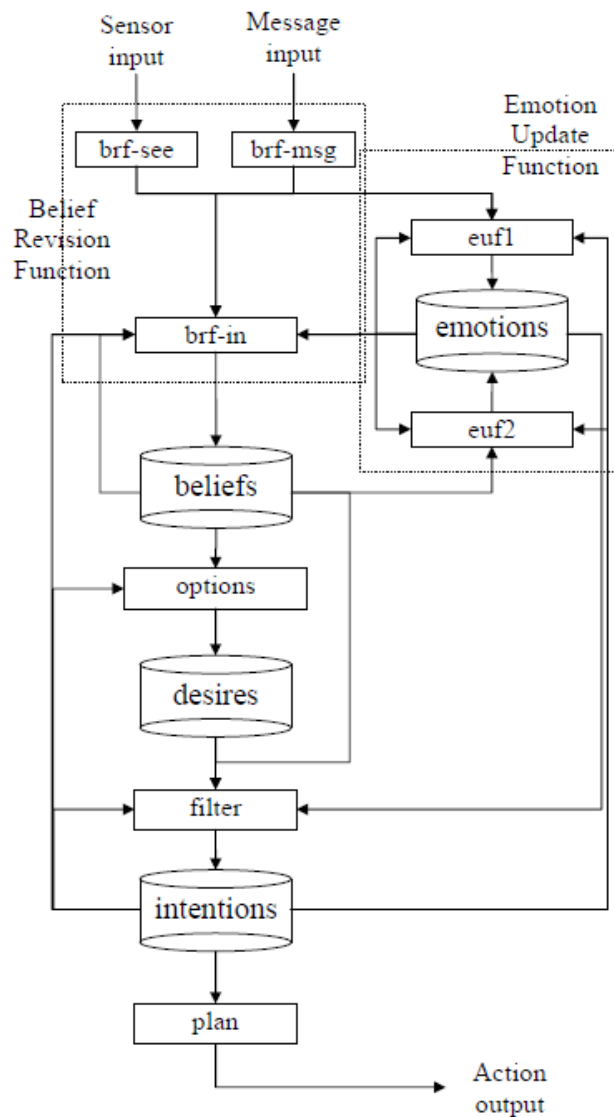
Εικόνα 5 Πολυπρακτορικό Σύστημα Ηλεκτρονικής Μάθησης (Ammar, 2007)

2.9 EBDI Αρχιτεκτονική

Σύμφωνα με τους Hong Jiang και Jose M. Vidal για να ενσωματωθούν τα συναισθήματα στους πράκτορες χρειάζεται να λυθούν τρία προβλήματα:

1. Πώς να μετρηθούν ή να παρουσιαστούν τα συναισθήματα;
2. Πώς τα συναισθήματα επηρεάζουν τη διαδικασία λήψης αποφάσεων;
3. Πώς πρέπει να γίνεται η ενημέρωση της κατάστασης των συναισθημάτων;

Τις περισσότερες φορές, οι λεπτομέρειες των λύσεων εξαρτώνται από συγκεκριμένες εφαρμογές. Η αρχιτεκτονική EBDI (Emotional-Belief-Desire-Intention) συνδυάζει αυτές τις τρεις ανησυχίες σε μία BDI αρχιτεκτονική που βασίζεται στην πρακτική διαδικασία σκέψης του ανθρώπου, ενώ αφήνει ανοιχτές τις λεπτομέρειες για τους διάφορους σχεδιαστές. Μπορούμε να συνοψίσουμε τον κύκλο εκτέλεσης ως εξής: Όταν υπάρχουν νέες πληροφορίες από το περιβάλλον μέσω μηνυμάτων αισθητήρα ή μηνυμάτων επικοινωνίας, ο πράκτορας EBDI δημιουργεί υποψήφια πιστεύω. Αυτά τα υποψήφια πιστεύω μαζί με τις τρέχουσες προθέσεις ανανέωσης συναισθημάτων, επιτρέπουν στον πράκτορα να αποκτήσει μία πρώτη αίσθηση σχετικά με τις πληροφορίες που δέχεται. Με βάση τις νέες πληροφορίες και τα νέα συναισθήματα που έχει δεχτεί, έχοντας ως οδηγό τις νέες προθέσεις, ο πράκτορας επαναξιολογεί τις πεποιθήσεις του (Jiang & Vidal, 2007).

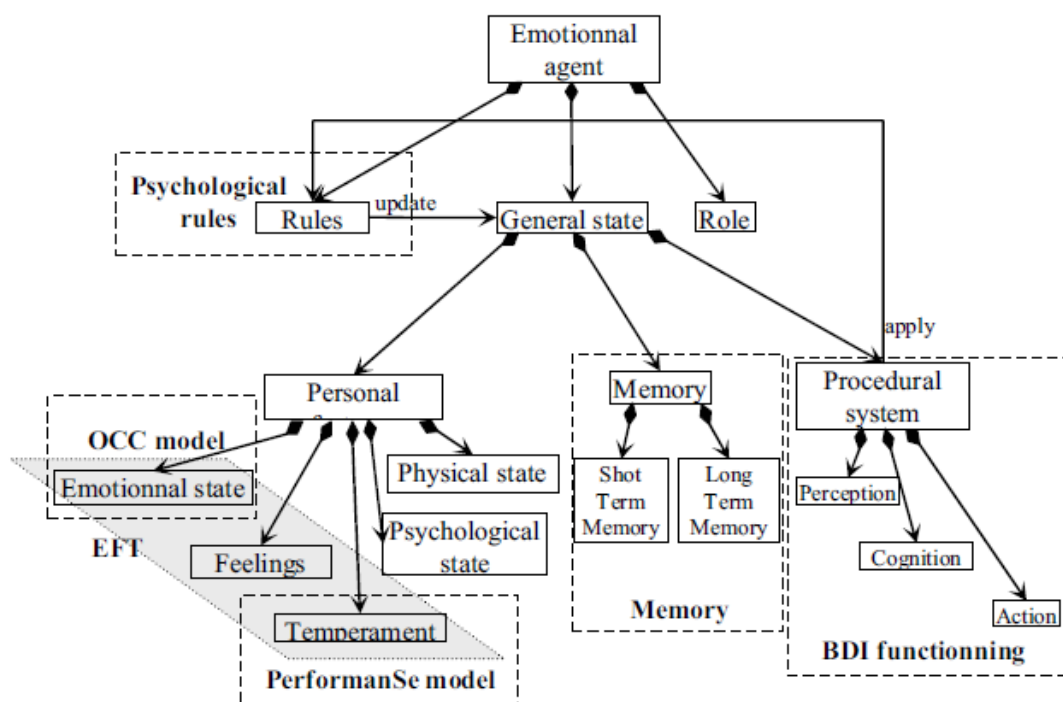


Εικόνα 6 EBDI Αρχιτεκτονική (Jiang & Vidal, 2007)

2.10 Αρχιτεκτονική στήριξης αποφάσεων

Ο συγκεκριμένος πράκτορας συναισθημάτων και υποστήριξης αποφάσεων είναι βασισμένος πάνω στη BDI (Belief-Desire-Intention) αρχιτεκτονική. Τα προσωπικά του στοιχεία ξεχωρίζουν για το σύνολο των εσωτερικών χαρακτηριστικών της συμπεριφοράς του. Πρώτον, η πρωτοτυπία του μοντέλου είναι ότι η προσωπικότητα του πράκτορα χωρίζεται σε τρία επίπεδα: συναισθήματα, αισθήματα και ιδιοσυγκρασία. Αυτός ο διαχωρισμός γίνεται με βάση την ποικιλομορφία του κάθε επιπέδου. Τα συναισθήματα είναι εφήμερα και εκπροσωπούν άμεσα τα αισθήματα. Για παράδειγμα, τις περισσότερες φορές αισθάνεστε φοβισμένοι μόνο για ένα σύντομο χρονικό διάστημα. Όσον αφορά την ιδιοσυγκρασία, αντιπροσωπεύει τη φύση του πράκτορα και περιμένει κάποιος να είναι σταθερή στο πέρασμα του χρόνου. Πράγματι, αν λάβουμε υπόψη την εξωστρέφεια, υποθέτουμε ότι αυτό το χαρακτηριστικό δεν

εξελίσσεται. Απλά κρατώντας αυτά τα δύο μέρη δεν επιτρέπουμε στον πράκτορά μας να θυμάται τις αλληλεπιδράσεις του ή τι ακριβώς αισθάνθηκε. Για παράδειγμα, ένας πράκτορας είναι πιο πιθανό να αποκτήσει φιλικές σχέσεις με έναν άλλο πράκτορα που του δίνει χαρά και ευτυχία παρά με κάποιον άλλο. Αυτός είναι ο λόγος για τον οποίο έγινε η εισαγωγή των συναισθημάτων. Τα αισθήματα εξελίσσονται σε συνάρτηση με τα συναισθήματα και την ιδιοσυγκρασία του πράκτορα (Daviet & Desmier, 2005).



Εικόνα 7 Αρχιτεκτονική Μοντέλου Στήριξης Αποφάσεων (Daviet & Desmier, 2005)

2.11 Αρχιτεκτονική πράκτορα με συναισθήματα που παρατηρεί τις κινήσεις ενός χορευτή. Ορθολογική και Αντιδραστική

Φανταστείτε ένα χώρο, για παράδειγμα ένα μουσείο ή ένα θεατρικό στάδιο, στο οποίο μπορείτε να σταθείτε περιβαλλόμενος από μουσική και να ελέγξετε τη ροή, το ρυθμό και την ενορχήστρωση, από τη δική σας συνολική κίνηση του σώματος, χωρίς να χρειάζεται να χρησιμοποιήσετε κάποιο άλλο σύστημα ελέγχου. Επιπλέον, φανταστείτε ότι ο χώρος αυτός περιλαμβάνει κινητούς πράκτορες οι οποίοι μπορούν να αλληλεπιδράσουν, να σας τραγουδήσουν, να σας μιλήσουν και να σας ψυχαγωγήσουν σαν να ήταν ηθοποιοί πάνω σε μία σκηνή. Αυτό το είδος των πολυμορφικών συστημάτων απαιτεί μεταξύ άλλων ευφρείς

διεπαφές και συμπεριφορές προσαρμογής. Το παρακάτω σχήμα (Εικόνα 8) δείχνει ένα παράδειγμα από αυτό το σενάριο.



Εικόνα 8 Παρατήρηση κινήσεων και χειρονομιών του χορευτή (Camurri, 1999)

Πιστεύουμε ότι συμπεριλαμβάνοντας τα τεχνητά συναισθήματα ως ένα από τα βασικά χαρακτηριστικά αυτών των συστημάτων, θα επιτευχθεί μία πιο αποτελεσματική, ενθαρρυντική και φυσική αλληλεπίδραση με τους ανθρώπους. Φανταστείτε για παράδειγμα ένα πράκτορα που θα κινείται στη σκηνή και μετά από επανειλημμένες φορές που θα βρει έναν ηθοποιό στο δρόμο του, θα ανακοινώσει με ένα ψηφιακό δείγμα φωνής: «Σταμάτα να μπλοκάρεις το δρόμο μου, είμαι θυμωμένος» καθώς αναβοσβήνουν τα μάτια του ή «Με μισεί, είμαι τόσο λυπημένος» καθώς κινείται αργά προς μία γωνία της σκηνής. Σαφώς, η ποιότητα και η αποτελεσματικότητα της αλληλεπίδρασης του πράκτορα με τον ηθοποιό, ενισχύεται σε μεγάλο βαθμό από αυτού του είδους τις συμπεριφορές, διότι αφορά, τουλάχιστον εξωτερικά, ανθρώπινες συμπεριφορές που σχετίζονται με συναισθήματα. Στην πραγματικότητα ο πράκτορας αποδεικνύεται περισσότερο πειστικός. Για να εισάγουμε αυτές τις συμπεριφορές στο λογισμικό ελέγχου του πράκτορα, μπορούμε να κωδικοποιήσουμε μία κατάσταση συναισθημάτων που εξελίσσεται με την πάροδο του χρόνου, επηρεάζεται από εξωτερικά γεγονότα και επηρεάζει τις αντιδράσεις του πράκτορα (Camurri & Ferrentino, 1999).

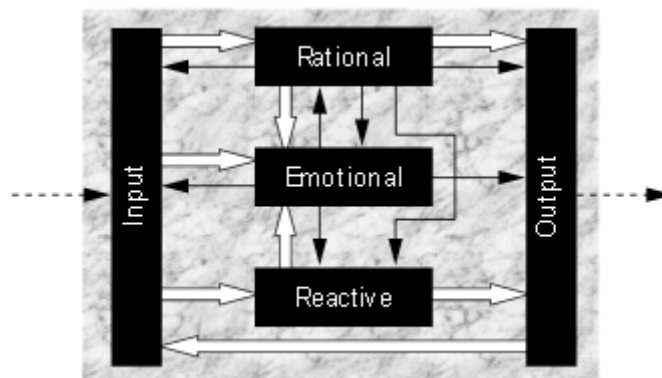
Σκεφτείτε, ως ένα άλλο παράδειγμα, την ευρέως αποδεκτή πεποίθηση, ότι η μουσική και ο χορός έχουν κυρίως συναισθηματικό περιεχόμενο, ενσωματωμένο στην αλληλεπίδραση και στην έκφραση των ερμηνευτών. Ως εκ τούτου, ένας ενεργός χώρος που παράγει μουσική από τη συνολική κίνηση του σώματος, μπορεί να συλλάβει χαρακτηριστικά κίνησης με επίκεντρο την αλληλεπίδραση και τα συναισθήματα που σχετίζονται με την ερμηνεία και να ενσωματώσει το συναισθηματικό περιεχόμενο στη μουσική που θα παραχθεί. Αυτό μπορεί να

υλοποιηθεί κωδικοποιώντας μία συναισθηματική κατάσταση η οποία επηρεάζεται από την κίνηση και με τη σειρά της επηρεάζει τη μουσική. Σε αυτό το άρθρο προτείνεται μία γενική αρχιτεκτονική δημιουργίας πρακτόρων που θα έχουν συναισθήματα. Ένας πράκτορας συναισθημάτων αλληλεπιδρά με τον εξωτερικό κόσμο λαμβάνοντας εισόδους και στέλνοντας εξόδους πιθανότατα σε πραγματικό χρόνο. Το βασικό ερώτημα που γεννιέται είναι αν η συναισθηματική κατάσταση του πράκτορα εξελίσσεται με την πάροδο του χρόνου. Επηρεάζει τις εξόδους και η εξέλιξή της ανταποκρίνεται στις εισόδους. Ο πράκτορας διαθέτει επίσης ορθολογική γνώση η οποία εξελίσσεται με την πάροδο του χρόνου. Η εξέλιξη αυτή εξαρτάται σε μεγάλο βαθμό από τα ερεθίσματα που δέχεται ο πράκτορας και την ανταπόκριση που έχει σε αυτά. Επιπλέον, ορισμένα από τα αποτελέσματα παράγονται αντιδραστικά από τις εισόδους με έναν πραγματικά άμεσο και γρήγορο τρόπο. Αυτοί οι συναισθηματικοί, ορθολογικοί και αντιδραστικοί υπολογισμοί δεν λειτουργούν μεμονωμένα αλλά επηρεάζει ο ένας τον άλλο. Για παράδειγμα, η συναισθηματική κατάσταση μπορεί να διαμορφώνει τις αντιδράσεις του πράκτορα. Η αρχιτεκτονική που προτείνεται προσπαθεί να δομήσει έναν υγιή και βολικό πράκτορα που θα είναι ικανός να λειτουργεί και να φέρεται σαν άνθρωπος. Τα κίνητρα της αρχιτεκτονικής προκύπτουν από τις συμπεριφορές που σκοπεύουμε να εφαρμόσουμε στον πράκτορά μας και δεν προέρχονται από τις θεωρίες της ανθρώπινης συμπεριφοράς. Το κυριότερο κριτήριο αξιολόγησης της αρχιτεκτονικής είναι η δυνατότητα δόμησης πρακτόρων των οποίων οι συμπεριφορές θα μεταφέρουν συναισθήματα στα άτομα που εκτίθενται στο σύστημα. Το λογισμικό για τέτοιες εφαρμογές μπορεί να γίνει κατανοητό ως ένας πληθυσμός από πράκτορες με συναισθήματα. Για παράδειγμα, θα μπορούσε να υπάρχει ένας πράκτορας υπεύθυνος για την πλοήγηση και ένας ή περισσότεροι πράκτορες υπεύθυνοι για το χώρο στον οποίο κινούνται οι άνθρωποι. Ουσιαστικά, οι εισοδοί για αυτούς τους πράκτορες περιλαμβάνουν δεδομένα από την ανθρώπινη κίνηση και τους αισθητήρες των ρομπότ καθώς και μηνύματα από άλλους πράκτορες. Οι έξοδοι περιλαμβάνουν εντολές για συσκευές ήχου καθώς επίσης και μηνύματα προς άλλους πράκτορες. Η συναισθηματική κατάσταση αυτών των πρακτόρων μπορεί να τροποποιείται ανάλογα με τις κινήσεις των ανθρώπων. Οι γρήγορες και γεμάτες ενέργεια κινήσεις μπορεί να οδηγήσουν τον πράκτορα σε μία ευχάριστη διάθεση ενώ αντίθετα η άρνηση της κίνησής του, μπορεί να οδηγήσει σε μία οργισμένη διάθεση. Μία σειρά από πολλά υποσχόμενες εφαρμογές για τους πράκτορες με συναισθήματα έχουν προκύψει όπως:

- Διαδραστικής ψυχαγωγίας όπως μία ντισκοτέκ όπου οι χορευτές μπορούν να επηρεάζουν και να αλλάζουν τη μουσική καθώς και παιχνίδια όπως το «χορευτικό» караоке όπου όσο καλύτερα χορεύει κάποιος, τόσο καλύτερη μουσική ακούει.
- Διαδραστικό θέατρο στο σπίτι.
- Διαδραστικά εργαλεία για αερόμπικ και γυμναστική.

- Εργαλεία αποκατάστασης όπως θεραπείες για τη ψυχική αναπηρία που υποστηρίζουν οι πράκτορες.
- Εργαλεία για διδασκαλία μέσα από το παιχνίδι σε προσομοιωμένα περιβάλλοντα.
- Εργαλεία για τη βελτίωση της επικοινωνίας σχετικά με νέα προϊόντα και ιδέες σε συνέδρια όπως είναι οι επιδείξεις μόδας.
- Πολιτιστικές εφαρμογές που χρησιμοποιούν αυτόνομους πράκτορες για να προσελκύσουν και να ψυχαγωγήσουν τον κόσμο.
- Εφαρμογές επαυξημένης πραγματικότητας.

Το σχήμα της εικόνας 10 δείχνει τη δομή ενός πράκτορα με συναισθήματα. Τα πέντε ορθογώνια αντιπροσωπεύουν τα δραστικά συστατικά της δομής. Ένα παχύ λευκό βέλος από το ένα ενεργό συστατικό προς το άλλο, αναλογεί σε μία μεταξύ τους συνεργασία όπου το ένα έχει το ρόλο του παραγωγού και το άλλο το ρόλο του καταναλωτή. Ο πίνακας 1 αριθμεί τις σχέσεις μεταξύ των ενεργών συστατικών. Ένα λεπτό μαύρο βέλος από ένα συστατικό προς ένα άλλο αντιπροσωπεύει ένα δοχείο δεδομένων πάνω στο οποίο το πρώτο συστατικό έχει δικαίωμα να διαβάσει και να επεξεργαστεί δεδομένα και το άλλο έχει δικαίωμα μόνο να διαβάσει. Τέλος, τα δύο διακεκομμένα βέλη αντιπροσωπεύουν ροές πληροφοριών από τον εξωτερικό κόσμο προς την είσοδο και από την έξοδο προς τον εξωτερικό κόσμο (Bates, 2004). Στη συνέχεια θα περιγραφούν τα συστατικά με περισσότερες λεπτομέρειες καθώς και τα δεδομένα μέσω των οποίων αυτά αλληλεπιδρούν μεταξύ τους και με τον έξω κόσμο.



Εικόνα 9 Δομή ενός πράκτορα με συναισθήματα (Camurri & Ferrentino, 1999)

From	To	Contains
Input	Reactive	Reactive inputs
Reactive	Output	Reactive outputs
Input	Rational	Rational inputs
Rational	Output	Rational outputs
Input	Emotional	Emotional stimuli
Rational	Emotional	Emotional stimuli
Reactive	Emotional	Emotional stimuli
Output	Input	Internal feedbacks

Πίνακας 1 Σχέσεις μεταξύ των συστατικών (Camurri & Ferrentino, 1999)

2.11.1 Το συστατικό εισόδου (*Input component*)

Το συστατικό εισόδου λαμβάνει εισροές από τον εξωτερικό κόσμο, όπως δεδομένα από τους αισθητήρες κίνησης καθώς και μηνύματα από άλλους πράκτορες, σύμφωνα με το μοντέλο ζήτησης και παραλαβής. Το συστατικό εισόδου επεξεργάζεται αυτά τα δεδομένα για να παράγει αντιδραστικά, ορθολογικά και συναισθηματικά ερεθίσματα. Η επεξεργασία μπορεί να είναι περισσότερο ή λιγότερο πολύπλοκη και εξαρτάται από κάποια εσωτερική κατάσταση της εισόδου. Για παράδειγμα, πληροφορίες που παράγονται από την ανάλυση των δεδομένων από αισθητήρες πλήρους κίνησης του ανθρώπινου σώματος μπορούν να περιλαμβάνουν δυναμικές ποσότητες, όπως θέσεις, ταχύτητες, ενέργειες, καθώς και συμβολικές χειρονομίες, όπως ο βαθμός στον οποίο οι χορευτές παραμένουν στο σωστό τέμπο ή πώς καταλαμβάνουν το χώρο της σκηνής. Συνήθως αποκτούμε αυτές τις πληροφορίες με την ενσωμάτωση διαφόρων αισθητήρων δεδομένων. Από τη στιγμή που η είσοδος μπορεί να διαβάσει τη συναισθηματική παράμετρο, η επεξεργασία της πληροφορίας εξαρτάται από την τρέχουσα συναισθηματική κατάσταση του πράκτορα. Όπως θα δούμε παρακάτω, το συστατικό συναισθημάτων είναι υπεύθυνο για τη συναισθηματική κατάσταση του πράκτορα και κρατάει πληροφορίες για αυτήν στην παράμετρο εισόδου που προαναφέραμε. Με αυτό τον τρόπο, το συστατικό εισόδου μπορεί να κάνει ελαφρώς διαφορετικές ενέργειες σύμφωνα με την τιμή της παραμέτρου. Για παράδειγμα, σκεφτείτε ένα ρομπότ που πλοηγείται σε μία σκηνή και κατά μήκος της διαδρομής του συναντά έναν άνθρωπο. Όταν το συστατικό εισόδου αναγνωρίζει αυτή την κατάσταση από τους αισθητήρες δεδομένων του ρομπότ, αποκτά μία συγκεκριμένη άποψη για το αν το ρομπότ είναι λυπημένο ή δυσαρεστημένο ή αντίθετα, αν είναι χαρούμενο.

From	To	Contains
Emotional	Input	Emotional-input parameter
Emotional	Output	Emotional-output parameter
Emotional	Reactive	Emotional-reactive parameter
Emotional	Rational	Emotional-rational parameter
Rational	Input	Rational-input parameter
Rational	Output	Rational-output parameter
Rational	Reactive	Rational-reactive parameter
Rational	Emotional	Rational-emotional parameter

Πίνακας 2 Δεδομένα που μεταφέρουν οι πράκτορες (Camurri & Ferrentino, 1999)

2.11.2 Το συστατικό εξόδου (*Output component*)

Το συστατικό εξόδου στέλνει εξόδους, όπως μεταφέρονται τα δεδομένα ήχου και άλλα μηνύματα προς άλλους πράκτορες. Η χρήση εντολών σε έναν οδηγό συσκευής είναι μία τέτοια περίπτωση. Αυτό το συστατικό παράγει τις εξόδους με την επεξεργασία των αντιδραστικών και ορθολογικών εξόδων καθώς και με τις συναισθηματικές και ορθολογικές παραμέτρους. Παραδείγματα αντιδραστικών και ορθολογικών εξόδων είναι οι οδηγοί για συσκευές ήχου, από τους οποίους προκύπτουν πληροφορίες για το πώς να παίζεις απλές νότες, μουσικά κονσέρτα και ψηφιοποιημένο λόγο. Οι ορθολογικοί και συναισθηματικοί παράγοντες περιλαμβάνουν πληροφορίες για την τρέχουσα ορθολογική και συναισθηματική κατάσταση του πράκτορα. Έτσι, η επεξεργασία της εξόδου μπορεί να επηρεάζεται και από τις δύο καταστάσεις. Για παράδειγμα, το ίδιο μουσικό απόσπασμα μπορεί να παιχτεί με διαφορετικά ηχοχρώματα σε διαφορετικές συναισθηματικές καταστάσεις, μία φωτεινή χροιά για την ευτυχία και μία πιο σκοτεινή για τη θλίψη. Το συστατικό εξόδου μπορεί να παράγει διάφορες εξόδους για τον εξωτερικό κόσμο και συνήθως αυτό εξαρτάται από τις εισόδους που δέχεται.

2.11.3 Το συστατικό αντίδρασης (*Reactive component*)

Το συστατικό αντίδρασης επεξεργάζεται τις αντιδραστικές εισόδους που δέχεται από το συστατικό εισόδου και παράγει αντιδραστικές εξόδους για το συστατικό εξόδου και συναισθηματικά ερεθίσματα για το συστατικό συναισθημάτων. Το συστατικό αντίδρασης επεξεργάζεται τις πληροφορίες σχετικά γρήγορα και όταν ο πράκτορας ανταλλάσσει τα δεδομένα σε πραγματικό χρόνο, τότε συνειδητοποιεί και την πραγματική του συμπεριφορά. Μπορεί να εκτελέσει μία ποικιλία από υπολογισμούς γιατί εξαρτάται από τις ορθολογικές και συναισθηματικές καταστάσεις μέσω των ορθολογικών και συναισθηματικών παραμέτρων. Ένα ενδιαφέρον παράδειγμα είναι αυτό με τα εικονικά μουσικά όργανα, όπου μία

επανελημμένη νευρική και ρυθμική χειρονομία του ανθρώπου μετασχηματίζει σε ορισμένα σημεία της σκηνής, τους ουδέτερους ήχους σε ήχους τυμπάνου.

2.11.4 Το ορθολογικό συστατικό (*Rational component*)

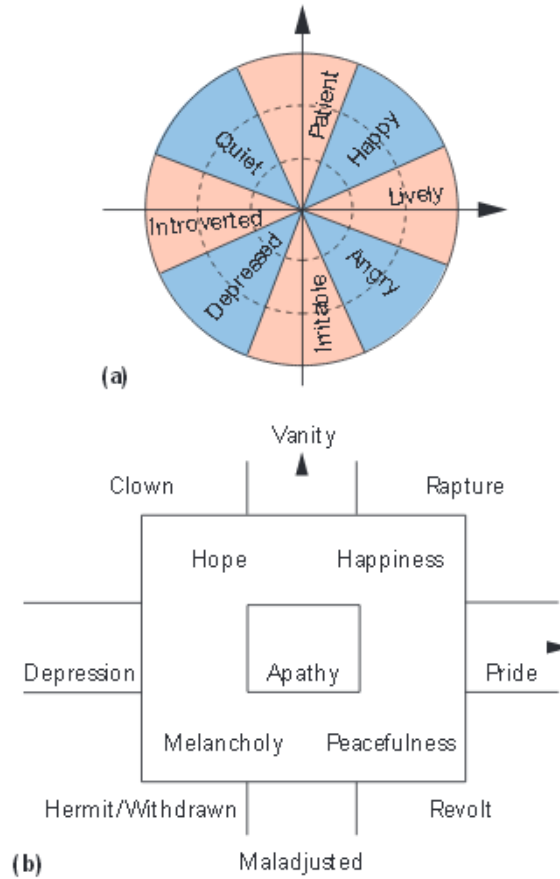
Το ορθολογικό συστατικό είναι υπεύθυνο για την ορθολογική κατάσταση του πράκτορα η οποία αποτελείται από δύο μέρη: α) γνώση σχετικά με τον εξωτερικό κόσμο όπως, πώς κινούνται οι άνθρωποι, ποια εικονικά όργανα δημιουργήθηκαν, κ.ά. και β) τον ίδιο τον πράκτορα, όπως τους στόχους που έχει θέσει. Αυτή η γνώση προκύπτει από μία μηχανή εξαγωγής συμπερασμάτων η εξέλιξη της οποίας εξαρτάται από την επεξεργασία των ορθολογικών εισόδων από το συστατικό εισόδου και παράγει ορθολογικές εξόδους για το συστατικό εξόδου. Η ορθολογική γνώση και η μηχανή συμπερασμού μπορεί να είναι περισσότερο ή λιγότερο πολύπλοκες, από απλούς ισχυρισμούς και αποδείξεις θεωρημάτων μέχρι μία μίξη διαφορετικών πραγμάτων όπως τα υβριδικά μοντέλα. Σε αντίθεση με το συστατικό αντίδρασης, το ορθολογικό συστατικό δεν έχει αυστηρούς χρονικούς περιορισμούς και ως εκ τούτου, οι υπολογισμοί του είναι αρκετά πολύπλοκοι. Η ορθολογική γνώση μπορεί επίσης να βασίζεται στην τρέχουσα συναισθηματική κατάσταση του πράκτορα. Αυτή η εξάρτηση μπορεί να πραγματοποιηθεί με τους δύο ακόλουθους τρόπους:

1. Η παράμετρος κωδικοποιεί τη γνώση σχετικά με την τρέχουσα συναισθηματική κατάσταση του πράκτορα, έτσι ώστε η μηχανή συμπερασμού να λάβει ακριβώς το ίδιο μήνυμα χωρίς να τροποποιήσει κάτι.
2. Η παράμετρος επηρεάζει τη μηχανή συμπερασμού, έτσι ώστε η ίδια γνώση να εξελίσσεται και να μεταφέρεται διαφορετικά σε διάφορες συναισθηματικές καταστάσεις.

2.11.5 Το συναισθηματικό συστατικό (*Emotional component*)

Το συναισθηματικό συστατικό είναι υπεύθυνο για τη συναισθηματική κατάσταση του πράκτορα της οποίας η χρονική εξέλιξη διέπεται από συναισθηματικά ερεθίσματα που δέχεται από τα συστατικά εισόδου, αντίδρασης και ορθολογισμού. Μία ενδιαφέρουσα υπολογιστική διαπίστωση μίας συναισθηματικής κατάστασης, μεταξύ πολλών πιθανών, είναι οι συντεταγμένες ενός σημείου σε ένα «συναισθηματικό» χώρο. Οι συντεταγμένες μπορούν να αλλάξουν βήμα προς βήμα σύμφωνα με κάποια συναισθηματικά ερεθίσματα. Ο συναισθηματικός χώρος είναι συνήθως χωρισμένος σε ζώνες οι οποίες χαρακτηρίζονται από συμβολικά ονόματα όπως ευτυχία, θλίψη, ενθουσιασμός. Έτσι, διαφορετικά συναισθηματικά ερεθίσματα τείνουν να κινούν τη μάζα ανάμεσα στις διάφορες ζώνες. Επιπλέον, εκτός από τη λήψη συναισθηματικών ερεθισμάτων από το συστατικό ορθολογισμού, το συναισθηματικό

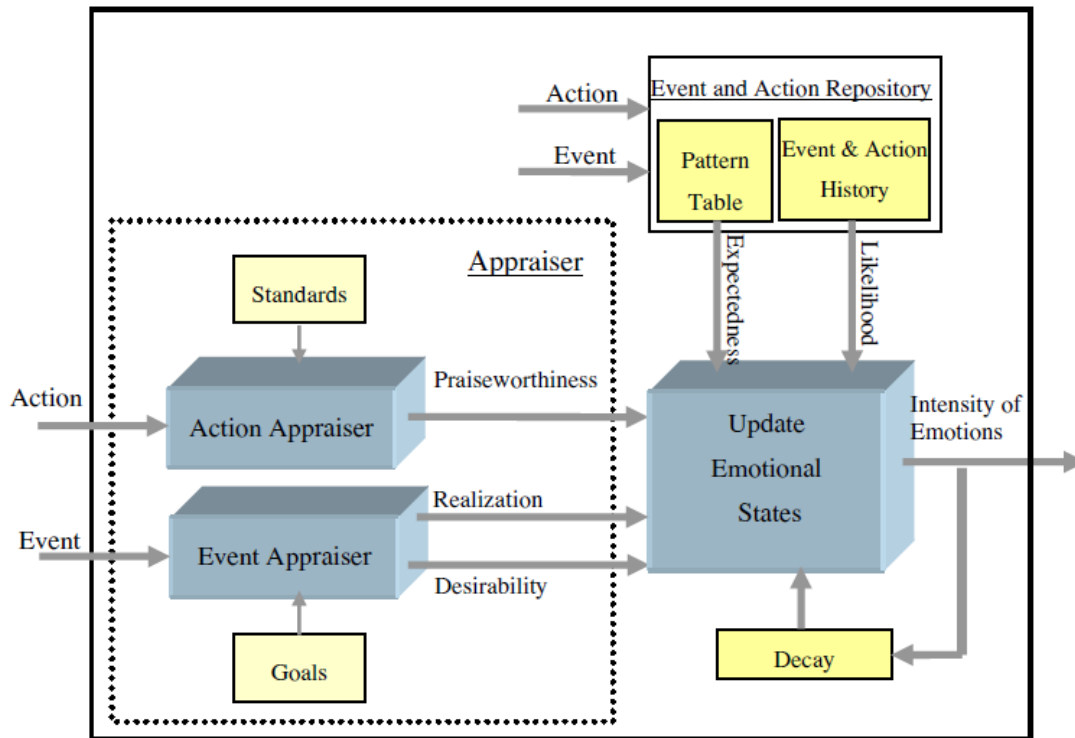
συστατικό μπορεί να επηρεαστεί από την ορθολογική κατάσταση μέσω των ορθολογικών και συναισθηματικών παραγόντων. Για παράδειγμα, η ταχύτερη δυναμική των αλλαγών της τρέχουσας συναισθηματικής κατάστασης ενός ρομπότ, αποδεικνύεται χρήσιμη σε καταστάσεις άμεσης αλληλεπίδρασης με τους ανθρώπους (Camurri, 2007).



Εικόνα 10 Παραδείγματα δύο συναισθηματικών χώρων (Camurri & Ferrentino, 1999)

2.12 Αρχιτεκτονική Γενικού Πράκτορα Συναισθημάτων

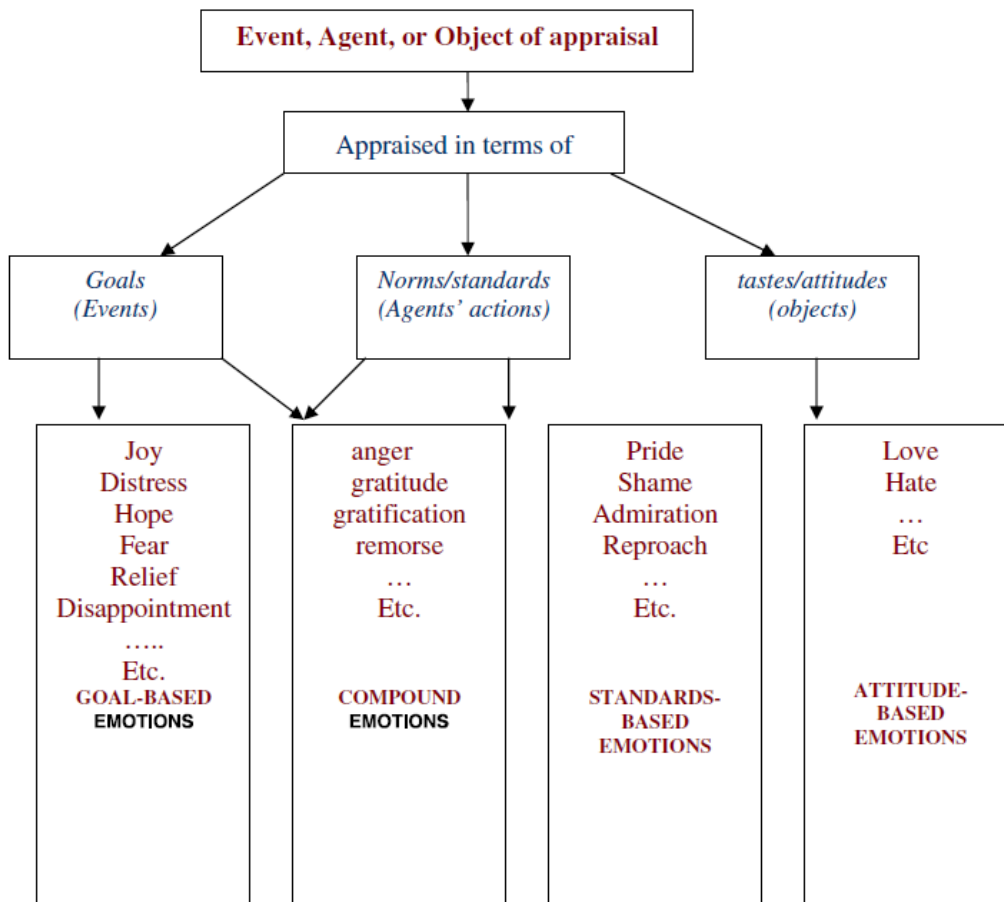
Η συνολική αρχιτεκτονική του γενικού πράκτορα συναισθημάτων (Generic Emotional Agent, GEMA) παρουσιάζεται στο παρακάτω σχήμα (Εικόνα 11).



Εικόνα 11 Αρχιτεκτονική του GEMA (Kazemifard, 2011)

Ο εκτιμητής (Appraiser) περιλαμβάνει δύο στοιχεία, την εκτίμηση του γεγονότος και την εκτίμηση της δράσης, τα οποία αξιολογούν τις ενέργειες και τις δράσεις του πράκτορα καθώς και άλλων πρακτόρων με βάση τους στόχους και τα πρότυπα ενός εξ ορισμού πράκτορα. Η αρχιτεκτονική βασίζεται στο μοντέλο OCC (Ortony, Clore and Collins's) με βάση το οποίο χρησιμοποιούνται ολοκληρωμένες τοπικές αλλά και καθολικές μεταβλητές για να υπολογιστεί η ένταση των συναισθημάτων καθώς και μεθόδους για την εκτίμηση των γεγονότων και των δράσεων. Ως εξόδους, ο εκτιμητής έχει βαθμούς ικανοποίησης, συνειδητοποίησης και επαίνου. Η ενημέρωση των συναισθηματικών καταστάσεων καθορίζει την ένταση των 16 συναισθημάτων που χρησιμοποιεί το μοντέλο για τον υπολογισμό της οποίας εφαρμόζει διάφορους κανόνες. Για τον υπολογισμό της έντασης του συναισθήματος χρησιμοποιούνται μία καθολική και τέσσερις τοπικές μεταβλητές. Οι τιμές αυτών των μεταβλητών προέρχονται από διάφορα στοιχεία του μοντέλου. Οι τοπικές μεταβλητές είναι η ικανοποίηση, η συνειδητοποίηση, ο έπαινος και η πιθανότητα. Η καθολική μεταβλητή είναι η αναμενόμενη έκβαση. Οι τομείς των ενεργειών και των δράσεων λειτουργούν με δύο στοιχεία, τον πίνακα προτύπων και την ιστορία των ενεργειών. Τα πρότυπα χρησιμοποιούνται για να υπολογίσουν την αναμενόμενη έκβαση. Αυτό το στοιχείο χρησιμοποιεί μία τροποποιημένη μέθοδο της FLAME αρχιτεκτονικής. Η ιστορία των ενεργειών και δράσεων αποθηκεύει το πόσες φορές εμφανίζεται μία ενέργεια έτσι ώστε να υπολογίσει την πιθανότητα εμφάνισης ενός γεγονότος. Η πιθανότητα αυτή εμφανίζεται με συναίσθημα

ελπίδας για τα επιθυμητά γεγονότα και με συναίσθημα φόβου για τα ανεπιθύμητα γεγονότα. Η αποσύνθεση (decay) μειώνει τα συναισθήματα σε κάθε κύκλο (Kazemifard, 2011).



Εικόνα 12 OCC Model (Zitar, 2007)

3. Συναισθηματικοί Πράκτορες

Διασύνδεσης: Θεωρητικό υπόβαθρο

Υπάρχουν πολλοί άνθρωποι που εργάζονται στον τομέα των πρακτόρων και παρόλο που όλοι έχουν μία συγκεκριμένη άποψη για το τι είναι και από τι αποτελείται ένας πράκτορας, δεν υπάρχει μία καθολική συμφωνία για τον ορισμό της έννοιας ενός πράκτορα. Φαίνεται ότι υπάρχουν τόσοι ορισμοί όσοι είναι και οι ερευνητές καθώς κάθε ορισμός προκύπτει από το πεδίο και την περιοχή της έρευνας. Η μη-επιστημονική ιδέα της ύπαρξης ενός πράκτορα σε ένα επιστημονικό περιβάλλον, οδήγησε σε μία θολή και ασαφή περιγραφή του μοντέλου που ονομάζεται τεχνολογία του πράκτορα. Τα κυριότερα συστατικά στο πεδίο των πρακτόρων είναι τέσσερις παράγοντες που πρέπει το σύστημα να πληροί. Πρόκειται για την ακρίβεια, τη λογική, την κάλυψη και τη λογοδοσία (accountability) (Cohen, 1992).

Η λεξικογραφική ερμηνεία της αγγλικής λέξης "agent" είναι ένα άτομο το οποίο ενεργεί εκ μέρους ενός άλλου ατόμου, δηλαδή ο αντιπρόσωπος, ο πράκτορας ή ο μεσίτης. Η ελληνική μετάφραση της λέξης "agent" όταν χρησιμοποιείται σε ορολογία σχετική με υπολογιστές είναι "πράκτορας" σύμφωνα με τον Ελληνικό Οργανισμό Τυποποίησης (ΕΛΟΤ).

Είναι γνωστή σε όλους η έννοια του πράκτορα, όπως αυτή χρησιμοποιείται στην καθημερινή ζωή. Για παράδειγμα, ο ρόλος που παίζει ένας ταξιδιωτικός πράκτορας ή ένας κτηματομεσίτης. Και οι δύο έχουν αντιπροσωπευτικό ρόλο, δηλαδή ενεργούν εκ μέρους κάποιου άλλου. Στην περίπτωση του ταξιδιωτικού πράκτορα εκ μέρους των ξενοδοχείων και των αεροπορικών εταιρειών, ενώ στην περίπτωση του κτηματομεσίτη εκ μέρους του ιδιοκτήτη. Επιπλέον, εκτός από το χαρακτήρα του μεσολαβητή που εμφανίζουν οι παραπάνω πράκτορες, έχουν ένα εξίσου σημαντικό χαρακτηριστικό: εμφανίζουν διαφορετικό βαθμό αυτονομίας. Για παράδειγμα ο κτηματομεσίτης αναλαμβάνει να κλείσει ένα ραντεβού για επίδειξη ενός σπιτιού σε κάποιον πιθανό αγοραστή, χωρίς να εμπλέκει στη διαδικασία αυτή τους ιδιοκτήτες, ενώ ίσως έχει και την εξουσιοδότηση να διαπραγματευτεί και την τιμή πώλησης. Ένα τρίτο σημαντικό χαρακτηριστικό του κάθε πράκτορα είναι ο βαθμός της αντίδρασης και της προνοητικότητας που υπάρχει στη συμπεριφορά του. Για παράδειγμα, ένας κτηματομεσίτης που απλά τοποθετεί μία πινακίδα με το σήμα "Πωλείται" έξω από μία ιδιοκτησία και περιμένει τους πελάτες να εμφανιστούν στο γραφείο του, βασίζεται

περισσότερο στην αντίδραση. Αντίθετα, ένας κτηματομεσίτης ο οποίος διαφημίζει γενικά την προσφορά ιδιοκτησίας στον τοπικό τύπο ενεργεί προνοητικά προκαλώντας το ενδιαφέρον πιθανών πελατών ακόμα και πριν να έχει έτοιμη για διάθεση κάποια ιδιοκτησία. Οι έννοιες της αντίδρασης (reactiveness) και της προνοητικότητας (proactiveness) δεν είναι αντίθετες. Ο ίδιος πράκτορας μπορεί να εμφανίσει υψηλούς βαθμούς αντίδρασης και προνοητικότητας σε διαφορετικές χρονικές στιγμές. Τέλος, οι προαναφερθέντες πράκτορες εμφανίζουν και άλλα χαρακτηριστικά, όπως η δυνατότητα μάθησης, η συνεργατικότητα και η κινητικότητα.

Αντίστοιχα δεδομένα ισχύουν και για τους υπολογιστικούς πράκτορες (computational agents). Ο όρος πράκτορας χρησιμοποιείται ευρύτατα στο χώρο της πληροφορικής για να περιγράψει ένα μεγάλο εύρος από υπολογιστικές οντότητες όπως λογισμικά συστήματα και αυτόνομα ρομπότ. Ο όρος "πράκτορας" στο χώρο των υπολογιστών έχει τη βάση του στο χώρο της τεχνητής νοημοσύνης όπου οι ερευνητές προσπαθούν να κατασκευάσουν τεχνητά όντα τα οποία να μιμούνται τις ανθρώπινες ικανότητες.

Ως πράκτορα μπορούμε να θεωρήσουμε ένα σύστημα το οποίο προσπαθεί να ικανοποιήσει ένα σύνολο στόχων μέσα σε ένα πολύπλοκο, δυναμικό περιβάλλον. Ο πράκτορας βρίσκεται μέσα στο περιβάλλον, μπορεί να αντιληφθεί το περιβάλλον του μέσω διάφορων αισθητήρων (sensors) και να ενεργήσει στο περιβάλλον μέσω διαφόρων οργάνων δράσης (actuators). Ένας ανθρώπινος πράκτορας έχει μάτια, αυτιά και άλλα όργανα-αισθητήρες και έχει πόδια, στόμα και άλλα μέλη του σώματος για την εξάσκηση δράσης. Ένας μηχανικός πράκτορας υποκαθιστά με κάμερες και υπέρυθρους ανιχνευτές πεδίου τους αισθητήρες ενώ επιτυγχάνει δράση με διάφορα μοτέρ. Ένας λογισμικός πράκτορας έχει κωδικοποιημένες σειρές χαρακτήρων ως όργανα αίσθησης και δράσης. Οι στόχοι που έχει ο κάθε πράκτορας μπορούν να πάρουν πολλές διαφορετικές μορφές. Μπορεί να είναι τελικοί στόχοι ή καταστάσεις τις οποίες ο πράκτορας προσπαθεί να επιτύχει, μία αμοιβή την οποία ο πράκτορας προσπαθεί να μεγιστοποιήσει, εσωτερικές ανάγκες ή κίνητρα τα οποία ο πράκτορας πρέπει να κρατήσει μέσα σε ορισμένα πλαίσια και όρια.

3.1 Τεχνολογία του Πράκτορα

Το 1998, ο Wooldrige πρότεινε έναν ευρύτερα αποδεκτό ορισμό, «Ο πράκτορας είναι κάποιο είδος υπολογιστικού υλικού ή λογισμικού συστήματος, με τα χαρακτηριστικά της αυτονομίας, της κοινωνικότητας, της αντιδραστικότητας και της προνοητικότητας (Wooldrige, 1986). Ουσιαστικά δηλαδή ο πράκτορας είναι κάτι πολύ ευφυές. Τι είναι όμως ευφυές και πώς αξιολογείται η ευφυΐα είναι δύο θέματα τα οποία δεν έχουν επιλυθεί ακόμη και για αυτό το λόγο ταυτίζουμε, στη συγκεκριμένη περίπτωση, την αυτονομία με την ευφυΐα. Αυτό γίνεται γιατί η αυτονομία είναι ο πυρήνας της έννοιας του πράκτορα και

σημαίνει ότι ένας πράκτορας δεν μπορεί να ελεγχθεί άμεσα από τους ανθρώπους τη στιγμή που το κυρίως σώμα του εκτελεί κάποιες διαδικασίες με ρεαλιστικούς στόχους από μόνο του (Maes, 1994). Την ίδια χρονική στιγμή, με κάποιο τρόπο έχει δικαίωμα ελέγχου της εσωτερικής του κατάστασης. Θεωρούμε δηλαδή ότι ένας ευφυής πράκτορας είναι κάποιο είδος λογισμικού που μπορεί να ολοκληρώσει με επιτυχία τα καθήκοντά του, να πάρει πρωτοβουλίες μετά από οδηγίες των ανθρώπων και να μην χρειάζεται έλεγχο και επιτήρηση κατά τη διάρκεια των εργασιών του (Wooldridge, 1995).

3.2 Χαρακτηριστικά Πρακτόρων

Πολλοί ερευνητές έχουν δώσει κατά καιρούς διάφορους ορισμούς για την έννοια του πράκτορα, ελπίζοντας να εξωτερικεύσουν το δικό τους τρόπο χρήσης των πρακτόρων. Μία συνοπτική παρουσίαση διαφορετικών ορισμών δίνεται από τους Jennings, Sycara και Wooldridge. Παρά την εκτεταμένη χρήση του όρου "πράκτορας" στην πληροφορική, δεν υπάρχει ακόμη ένας σαφής και ξεκάθαρος ορισμός του. Αντίθετα, η έκταση της χρήσης του όρου τείνει να θολώσει τις διαφορές μεταξύ εντελώς διαφορετικών κατευθύνσεων στη χρήση πρακτόρων. Για παράδειγμα, κάποιοι πράκτορες είναι σχεδιασμένοι να δουλεύουν μόνοι τους, άλλοι συνεργάζονται μέσα σε κοινωνίες πρακτόρων, μερικοί είναι κινητοί, άλλοι είναι στατικοί, άλλοι επικοινωνούν μεταξύ τους μέσω μηνυμάτων, άλλοι μαθαίνουν και αλλάζουν τρόπο συμπεριφοράς και άλλοι όχι.

Το βασικό κοινό χαρακτηριστικό που κάνει ένα κομμάτι υλικού ή λογισμικού να ονομάζεται πράκτορας, είναι το γεγονός ότι ο δημιουργός του επέλεξε να το αναλύσει, να το σχεδιάσει και να το ελέγξει χρησιμοποιώντας νοητικούς και ανθρωπομορφικούς όρους. Έτσι, πράκτορας είναι μία οντότητα της οποίας την κατάσταση μπορούμε να την περιγράψουμε με ανθρωπομορφικές έννοιες όπως γνώσεις, ικανότητες, επιλογές και υποχρεώσεις. Αντί λοιπόν αυστηρού ορισμού, η ανασκόπηση αυτή παραθέτει ένα σύνολο από χαρακτηριστικά που διακρίνουν τους πράκτορες λογισμικού:

- **Αυτονομία:** Οι πράκτορες δρουν εξ ορισμού αυτόνομα, μεσολαβώντας για ένα χρήστη, χωρίς την απευθείας παρεμβολή του χρήστη ή άλλων και έχουν σε κάποιο βαθμό τον έλεγχο πάνω στις πράξεις τους και την εσωτερική τους κατάσταση.
- **Νοημοσύνη:** Σαν αποτέλεσμα της αυτονομίας οι πράκτορες παρουσιάζουν κάποιο επίπεδο νοημοσύνης, το οποίο συνήθως αναδεικνύεται μέσα από το σύνολο της συμπεριφοράς τους και ενισχύεται από μηχανισμούς για την προσαρμογή σε αλλαγές του περιβάλλοντος και την αυτό-εκμάθηση.

-
- Αντίδραση-Προνοητικότητα: Η νοημοσύνη αυτή επιτρέπει στους πράκτορες να αντιλαμβάνονται το περιβάλλον τους και σαν αποτέλεσμα να μπορούν να δρουν όχι μόνο αντιδραστικά (reactively) αλλά και προνοητικά (proactively).
 - Κοινωνική ένταξη: Οι πράκτορες είναι κοινωνικά ενταγμένοι και ενεργοποιημένοι, δηλαδή μπορούν να επικοινωνούν με το χρήστη και χρειάζονται ειδικούς πόρους για κάτι τέτοιο. Πιο προχωρημένοι πράκτορες πιθανά να συνεργάζονται και με άλλους πράκτορες για να πετύχουν στόχους πέρα των δυνατοτήτων τους, χρησιμοποιώντας κάποια ειδική γλώσσα για την επικοινωνία μεταξύ τους. Το περιβάλλον και η κοινωνία των πρακτόρων δηλαδή, μπορεί να αποτελείται είτε μόνο από το χρήστη, είτε από μία ποικιλία χρηστών και άλλων πρακτόρων.
 - Ευρωστία: Ένας πράκτορας πρέπει να είναι έτοιμος να μάθει και να ανασυνταχθεί από την αποτυχία. Έτσι, επικεντρώνει την προσοχή του στους ελέγχους που έγιναν, στους στόχους που επιτεύχθηκαν, στις αποφάσεις που πάρθηκαν, προσδιορίζει τι πήγε λάθος, μαθαίνει από τα λάθη του και ενδεχομένως σχεδιάζει πάλι το πλάνο του από την αρχή.

Εκτός από τα παραπάνω βασικά χαρακτηριστικά των πρακτόρων, υπάρχει και ένα σύνολο από δευτερεύουσες ιδιότητες που συχνά τους χαρακτηρίζουν:

- Ειλικρίνεια: Οι πράκτορες δε θα δώσουν ποτέ εν γνώση τους λάθος πληροφορία όταν τους ζητηθεί.
- Καλή προαίρεση: Οι πράκτορες δεν έχουν συγκρουόμενους στόχους, δηλαδή κάθε πράκτορας προσπαθεί να κάνει αυτό που του ζητήθηκε.
- Λογική: Ένας πράκτορας ενεργεί πάντα με τρόπο που να προσπαθεί να πετύχει τους στόχους του, και ποτέ δε θα ενεργήσει με τρόπο που να απαγορεύει την επίτευξη των στόχων του.
- Κινητικότητα: Τελικά, είτε σαν κινητά ή ενεργά αντικείμενα, μπορούν να μεταφέρονται μεταξύ συστημάτων για να χρησιμοποιήσουν απομακρυσμένους πόρους ή να συναντηθούν και να συνεργαστούν με άλλους πράκτορες.

Η χρήση εννοιών από το χώρο των πρακτόρων μπορεί πολλές φορές να βοηθήσει στη μοντελοποίηση και κατανόηση ενός προβλήματος. Η ερώτηση "τι είναι πράκτορας" μετατρέπεται στην ερώτηση "ποια μέρη λογισμικού μπορούν να περιγραφούν με ανθρωπομορφικές έννοιες". Έτσι έχει δημιουργηθεί μεθοδολογία προγραμματισμού με επίκεντρο την έννοια του "πράκτορα", η οποία στη διεθνή βιβλιογραφία είναι γνωστή με την ονομασία Agent Oriented Programming (AOP). Πρέπει να σημειωθεί ότι ο τρόπος αυτός προγραμματισμού μπορεί να θεωρηθεί ειδικευση του αντικειμενοστραφούς μοντέλου προγραμματισμού (Object Oriented Programming, OOP).

3.3 Απαιτήσεις και Συστατικά για το σχεδιασμό ενός Πράκτορα

Ένα δυναμικό, απρόβλεπτο και πραγματικό περιβάλλον συνεπάγεται και συγκεκριμένες απαιτήσεις για να λειτουργήσει σωστά ένας αυτόνομος πράκτορας. Για παράδειγμα, θα υπάρξουν πάρα πολλές πληροφορίες ταυτόχρονα, εξωτερικά από το περιβάλλον και εσωτερικά από τον πράκτορα, σε κάθε τρέχουσα κατάσταση. Ο πράκτορας θα πρέπει να είναι συγκεντρωμένος και προσεκτικός. Οι πληροφορίες θα διανεμηθούν ευρέως τόσο σε χρόνο όσο και σε χώρο, απαιτώντας από τον πράκτορα να αναζητήσει για σχετικές πληροφορίες στο παρελθόν. Η ποιότητα των πληροφοριών μέσω των αισθητήρων θα διαφέρει. Ο πράκτορας όμως θα πρέπει να είναι σε θέση να «απομακρύνει» τις εν μέρει ακριβείς ή τις ανακριβείς πληροφορίες. Με την πάροδο του χρόνου το περιβάλλον θα τοποθετεί και άλλες απαιτήσεις για τη λειτουργία του πράκτορα. Μπορεί από τη μία στιγμή ο πράκτορας να έχει πολύ λίγα πράγματα να κάνει και του δίνεται η ευκαιρία να διαβουλευτεί και από την άλλη να πρέπει να ολοκληρώσει πολλές δουλειές μαζί γρήγορα, ποιοτικά και επαρκώς. Η αδυναμία πρόβλεψης του περιβάλλοντος καθιστά αδύνατο τον πλήρη σχεδιασμό πριν από τη δράση. Αντί αυτού, οι ευκαιρίες και οι απειλές θα πρέπει να βρίσκονται υπό συνεχή παρακολούθηση. Επίσης, κάποιος σχεδιασμός θα πρέπει να είναι σχετικά πρόχειρος και οι ακριβείς πληροφορίες να επιλέγονται κατά την εκτέλεση. Ως εκ τούτου, ένας αυτόνομος πράκτορας θα πρέπει να είναι ισχυρός, ευέλικτος και να έχει την ικανότητα να αντιμετωπίζει τα πολλαπλά προβλήματα που προκύπτουν σε ένα περιβάλλον πραγματικού κόσμου.

3.4 Αρχιτεκτονικές Πρακτόρων

Οι αρχιτεκτονικές των πρακτόρων συνήθως αναφέρονται σε μηχανισμούς για συνεκτική και αποτελεσματική συμπεριφορά σε σύνθετα, απρόβλεπτα και δυναμικά περιβάλλοντα όπως είναι ο πραγματικός κόσμος. Για να επιτευχθεί μία τέτοια συμπεριφορά, είναι απαραίτητες κάποιες απαιτήσεις καθώς και κάποιιοι περιορισμοί. Οι αρχιτεκτονικές των πρακτόρων ταξινομούνται σύμφωνα με την πτυχή των απαιτήσεων της αυτονομίας τις οποίες έχουν σχεδιαστεί να διευθύνουν. Για παράδειγμα, οι αντιδραστικές αρχιτεκτονικές σχετίζονται με άμεσες απαντήσεις για την τρέχουσα κατάσταση. Οι διαβουλευτικές αρχιτεκτονικές ασχολούνται με το σχεδιασμό της οριοθέτησης των στόχων και οι υβριδικές αρχιτεκτονικές ασχολούνται με το συνδυασμό της αντίδρασης και της διαβούλευσης. Είναι αναμενόμενο ότι κάθε αυτόνομος πράκτορας θα πρέπει να συνδυάσει τόσο την αντίδραση σε απρόβλεπτα γεγονότα όσο και το μακροπρόθεσμο σχεδιασμό για μία πιο αποτελεσματική λειτουργία.

Ο προγραμματισμός με βάση τους πράκτορες, όπως αναφέρθηκε, προτείνει τη μοντελοποίηση και επίλυση ενός προβλήματος χρησιμοποιώντας διάφορα αντικείμενα-πράκτορες, τα οποία αλληλεπιδρούν και συνεργάζονται μεταξύ τους, χωρίς όμως να προτείνει τον τρόπο με τον οποίο γίνεται κάτι τέτοιο. Η μεθοδολογία με την οποία αναπτύσσονται οι πράκτορες ορίζεται από την αρχιτεκτονική τους, η οποία καθορίζει τον τρόπο με τον οποίο ο πράκτορας μπορεί να αναλυθεί σε διάφορα κομμάτια και πώς αυτά τα κομμάτια αλληλεπιδρούν μεταξύ τους. Το σύνολο των κομματιών και οι αλληλεπιδράσεις τους καθορίζουν τις ενέργειες του πράκτορα και τη μελλοντική κατάστασή του σε σχέση με τα δεδομένα των αισθητήριων και σε συνδυασμό με την εσωτερική κατάσταση του πράκτορα. Σύμφωνα με τη βιβλιογραφική ανασκόπηση που διεξήχθη, οι αρχιτεκτονικές πρακτόρων μπορούν σε γενικές γραμμές να χωριστούν στις παρακάτω κατηγορίες:

- **Αρχιτεκτονικές προκαθορισμού (deliberative):** Ο πράκτορας διαθέτει ένα απόλυτα καθορισμένο συμβολικό μοντέλο του κόσμου και οι αποφάσεις δράσης παίρνονται με λογική αιτιολόγηση, η οποία βασίζεται σε ταίριασμα προτύπων και δεδομένων αισθητήρων και συμβόλων.
- **Αρχιτεκτονικές αντίδρασης (reactive):** Εδώ δεν υπάρχει κεντρικό συμβολικό μοντέλο του κόσμου και οι αποφάσεις δράσης δεν απαιτούν πολύπλοκο συλλογισμό με χρήση συμβόλων. Αντίθετα, η δράση του πράκτορα είναι αποτέλεσμα απλής αντίδρασης στη λήψη συγκεκριμένων δεδομένων στους αισθητήρες. Ανάλογο από το βιολογικό κόσμο μπορεί να θεωρηθεί η φυσιολογία των αντανακλαστικών.
- **Υβριδικές αρχιτεκτονικές:** Ένα μεγάλο μέρος των αρχιτεκτονικών πρακτόρων έχει στοιχεία και από τις δύο παραπάνω αρχιτεκτονικές.

3.5 Ιστορία των Πρακτόρων στο πλαίσιο της Τεχνητής Νοημοσύνης

Τη δεκαετία του 1990 είδαμε την ανάπτυξη ενός νέου προτύπου στην επιστήμη της πληροφορικής, τους πράκτορες λογισμικού. Πολλοί ερευνητές που δραστηριοποιούνται σήμερα σε αυτό το πεδίο, αντλούν αρκετά στοιχεία από έρευνες που έχουν γίνει στο πλαίσιο της Τεχνητής Νοημοσύνης (Artificial Intelligence) και της ανθρώπινης υπολογιστικής αλληλεπίδρασης. Οι πράκτορες διεπαφής (Interface Agents) είναι μία πτυχή των συστημάτων λογισμικού που έφερε επανάσταση στον τομέα της πληροφορικής. Αυτό γιατί μας επιτρέπει να προχωρήσουμε από την άμεση χειραγώγηση των συστημάτων στην έμμεση αλληλεπίδραση με τους πράκτορες. Ελευθερώνει τους ανθρώπους από την απαίτηση να ασχοληθούν με τις ελάχιστες λεπτομέρειες που τους παρέχει ένα έργο και ολοκληρώνουν οι ίδιοι τους στόχους.

Ο Alan Turing έθεσε αρχικά το ερώτημα: «Μπορούν οι υπολογιστικές μηχανές να σκεφτούν;» (Turing, 1950). Η δοκιμή του, όπου ένα πρόσωπο επικοινωνεί με ένα άλλο πρόσωπο ή με μία μηχανή, έγινε γνωστή ως δοκιμή Turing. Η δοκιμή Turing απαιτεί έναν «ομιλητικό» υπολογιστή ικανό να «κοροϊδεύει» τον άνθρωπο. Η γέννηση της Τεχνητής Νοημοσύνης χρονολογείται στην εποχή της δοκιμής Turing. Ο Marvin Minsky μετά από κάποια έρευνα που έκανε με τα νευρωνικά δίκτυα, συνεργάστηκε με τον John McCarthy για να εργαστούν σε συμβολικά συστήματα με βάση την αναζήτηση. Ταυτόχρονα, στο Carnegie-Mellon, ο Allen Newell και ο Herbert Simon, κατάφεραν επιτυχώς να αποδείξουν τα λογικά θεωρήματα με τη βοήθεια της ευρετικής αναζήτησης. Έτσι, οι συμβολικές αναπαραστάσεις έγιναν η κυρίαρχη προσέγγιση για την Τεχνητή Νοημοσύνη.

Το 1960 υπήρξε μεγάλη πρόοδος στον τομέα των πρακτόρων λογισμικού. Στο πανεπιστήμιο του Στάνφορντ, ο McCarthy (McCarthy, 1960) προσπάθησε να εκπροσωπήσει τον κόσμο με σύμβολα χρησιμοποιώντας τη λογική για την επίλυση των προβλημάτων (McCarthy, 1969). Ταυτόχρονα ο Newell δημιούργησε μία γενική μέθοδο επίλυσης των προβλημάτων, η οποία με την κατάλληλη παρουσίαση θα μπορούσε να λύσει οποιοδήποτε πρόβλημα. Τα προβλήματα που λύνονται είναι απλά και χωρίς πολλά σύμβολα έτσι ώστε να μπορούν να γενικεύσουν και να δώσουν λύση σε παρόμοια προβλήματα του πραγματικού κόσμου. Οι ερευνητές δεν ανησυχούν για τη διατήρηση του υπολογισμού σε ανθρώπινη χρονική κλίμακα χρησιμοποιώντας τις αυξήσεις στην απόδοση του υλικού έτσι ώστε να αυξάνεται συνεχώς ο πιθανός χώρος αναζήτησης, επιλύοντας έτσι όλο και περισσότερα προβλήματα.

Κατά τη διάρκεια της δεκαετίας του 1970 η αναζήτηση έγινε απόλυτα κατανοητή (Nilsson, 1971). Τα συμβολικά συστήματα εξακολουθούν να κυριαρχούν με συνεχείς βελτιώσεις του υλικού που επιτρέπουν σταθερή και επιτυχημένη πρόοδο. Δημιουργούνται τα ρομπότ, όπως ο Shakey (Nilsson, 1984), που έζησε σε έναν ειδικό κόσμο και μπορούσε να περιηγηθεί σε έναν χώρο λογικά. Τέτοιοι απλουστευμένοι κόσμοι αποφεύγουν την πολυπλοκότητα των προβλημάτων του πραγματικού κόσμου.

Το 1980 εξειδικευμένα συστήματα δημιουργήθηκαν για να επιλύσουν πραγματικά προβλήματα. Ο McCarthy συνειδητοποίησε ότι η κοινή λογική (McCarthy, 1993) απαιτείται να εξειδικευτεί περισσότερο και να μπορέσει να λύσει προβλήματα του μικρόκοσμου. Ένα υπό-πεδίο της Τεχνητής Νοημοσύνης, η αναπαράσταση γνώσης τέθηκε σε λειτουργία, για να εξετάσει τρόπους προσέγγισης της καθημερινότητας. Δυστυχώς, η ιδέα της κοινής λογικής δεν είχε τα αναμενόμενα αποτελέσματα και τα συστήματα που βασίζονται στη γνώση απέτυχαν, σε πολλές περιπτώσεις, να λύσουν προβλήματα του πραγματικού κόσμου. Παράλληλα, ο αλγόριθμος ανάστροφης διάδοσης (Rumelhart, 1986) προκάλεσε αναζωπύρωση του ενδιαφέροντος για προσεγγίσεις που παλαιότερα είχαν θεωρηθεί αποτυχημένες. Ο Minsky εξέτασε μία προσέγγιση βασισμένη σε πράκτορες για τη νοημοσύνη

(Minsky, 1986). Τα τέλη της δεκαετίας του 1980 και αρχές της δεκαετίας του 1990 είδαμε τις πρώτες αναζητήσεις βασισμένες σε συμβολικές προσεγγίσεις. Ο Brooks πειστικά αμφισβήτησε τις βασικές παραδοχές των συμβολικών προσεγγίσεων και μάλιστα δημιούργησε και ενσωματωμένα συστήματα χρησιμοποιώντας το ρητό «ο κόσμος όπως το δικό του καλύτερο μοντέλο» (Brooks, 1989). Αυτή η προσέγγιση είχε αρχικά κάποιες επιτυχίες αλλά στη συνέχεια απέτυχε να αντιμετωπίσει περίπλοκα προβλήματα.

Λόγω του ότι η γνώση της μηχανικής ήταν κάτι δαπανηρό και δύσκολο να επαναχρησιμοποιηθεί, αντικαταστάθηκε από τεχνικές μηχανικής μάθησης που υπάρχουν στην Τεχνητή Νοημοσύνη. Προς το τέλος της δεκαετίας του 1990, τα πρότυπα μάθησης αλγορίθμων μπορούσαν κατάλληλα να ταξινομήσουν τα πεδία της γνώσης, όπως ειδήσεις και γραπτά, με την ίδια ακρίβεια της χειροκίνητης ταξινόμησης. Στα μέσα της δεκαετίας του 1990 κυριάρχησαν τα πειράματα με πράκτορες διασύνδεσης που προσπαθούν με την αλληλεπίδραση να επιτύχουν τους στόχους τους. Τέτοια συστήματα πρακτόρων είναι δομημένα στον πραγματικό κόσμο και χρησιμοποιούν αποδεδειγμένες τεχνικές Τεχνητής Νοημοσύνης για να επιτευχθούν συγκεκριμένα αποτελέσματα (Maes, 1994 & Wooldridge, 1995). Η μοντελοποίηση του χρήστη άλλαξε τη δεκαετία του 1990 και αυτό οδήγησε στην μετάβαση σε μοντέλα βασισμένα στη δυναμική γνώση. Οι τεχνικές μηχανικής μάθησης αποδείχθηκαν ιδιαίτερα ικανές να αναγνωρίζουν μοτίβα στη συμπεριφορά των χρηστών.

3.6 Πολυπρακτορικά Συστήματα

Ένας πράκτορας είναι μία δηλωτική οντότητα που ορίζεται ως ένα υπολογιστικό σύστημα ικανό να δρα αυτόνομα και ευέλικτα σε κάποιο περιβάλλον προκειμένου να ανταποκριθεί στον αντικειμενικό στόχο σχεδίασης του (Wooldridge, 1995). Η δηλωτική φύση των πρακτόρων μπορεί να χαρακτηριστεί από την αυτονομία, την ευελιξία και την καταλληλότητα (Wooldridge, 1996). Η αυτονομία των πρακτόρων αντικατοπτρίζει την κοινωνική και αποκεντρωτική τους φύση. Ο ευέλικτος τρόπος με τον οποίο οι πράκτορες επιτυγχάνουν τους στόχους τους είναι ιδιαίτερα κατάλληλος για συστήματα τα οποία αναμένεται να λειτουργήσουν σε ένα διαρκώς μεταβαλλόμενο περιβάλλον. Οι πράκτορες είναι περισσότερο χρήσιμοι ως μέρος ενός πολυπρακτορικού συστήματος παρά ως ατομικές οντότητες ξεχωριστά. Η παγκόσμια συμπεριφορά ενός πολυπρακτορικού συστήματος προέρχεται από την αλληλεπίδραση μεταξύ των πρακτόρων σύστασης. Αυτοί συνεργάζονται, συντονίζουν τις διάφορες λειτουργίες και διαπραγματεύονται ο ένας με τον άλλο. Ένα πολυπρακτορικό σύστημα είναι μία κοινωνία από αυτόνομους και συνεργατικούς πράκτορες, όπως μία κοινωνική οργάνωση. Οι πράκτορες συμπεριφέρονται έξυπνα και συνεργάζονται έτσι ώστε να εκπληρώσουν τους στόχους τους. Ο κάθε ρόλος που ένας πράκτορας έχει την

ικανότητα να παίξει έχει ένα καλά καθορισμένο σύνολο στόχων, που επιτυγχάνεται είτε με τις δικές του ικανότητες είτε με την αλληλεπίδραση με τους άλλους πράκτορες.



Εικόνα 13 Πράκτορας Λογισμικού σε Πολυπρακτορικό Σύστημα

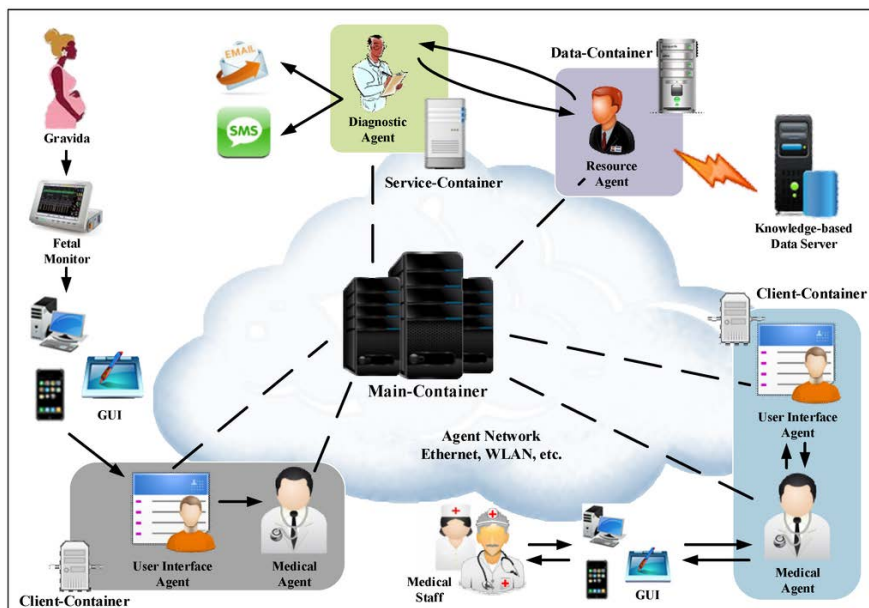
Επιπλέον, μία άλλη ταξινόμηση των συστημάτων πρακτόρων αφορά τον αριθμό των πρακτόρων που περιέχουν. Διακρίνουμε έτσι συστήματα ενός πράκτορα (Single Agent Systems, SAS) και συστήματα πολλαπλών πρακτόρων (Multi Agent Systems, MAS). Στα συστήματα SAS ο πράκτορας ενεργεί σαν μεσολαβητής ενός χρήστη ή μιας διεργασίας. Ενώ εκτελεί την εργασία του ο πράκτορας μπορεί να επικοινωνήσει με το χρήστη αλλά σε καμία περίπτωση με κάποιον άλλο πράκτορα. Αντιθέτως, στα συστήματα MAS οι πράκτορες δεν επικοινωνούν μόνο με το χρήστη και τους διάφορους πόρους του συστήματος αλλά επικοινωνούν και συνεργάζονται ενεργά και με άλλους πράκτορες, δίνοντας λύση σε προβλήματα που ξεπερνάνε τις δυνατότητες του καθένα ξεχωριστά. Στα συστήματα MAS πολύ συχνά εμφανίζονται προβλήματα συντονισμού και συνεργασίας της κοινότητας των πρακτόρων. Για την επίλυσή τους εφαρμόζονται μέθοδοι κατανεμημένης λύσης προβλημάτων. Πολλές φορές χρειάζονται διαπραγματεύσεις για την επίλυση πιθανών συγκρούσεων. Τέτοιες συγκρούσεις προκαλούνται όταν οι πράκτορες ανταγωνίζονται μεταξύ τους ή όταν προσπαθούν να συνεργαστούν στην κατασκευή ενός κοινού σχεδίου για την επίλυση ενός προβλήματος. Έτσι έχουμε δύο είδη πρακτόρων που συμμετέχουν σε MAS συστήματα:

- **Ανταγωνιστικοί πράκτορες:** Ο στόχος του κάθε πράκτορα είναι η μεγιστοποίηση των κερδών του, ενώ προσπαθεί να φτάσει σε συμφωνία με τους υπόλοιπους. Παραδείγματα αποτελούν πράκτορες που πουλάνε ή αγοράζουν προϊόντα, όπου οι

διαβουλεύσεις για την επίλυση των συγκρούσεων γίνονται σε ανταγωνιστικό επίπεδο. Κάθε ένας προσπαθεί να πετύχει την υψηλότερη ή χαμηλότερη δυνατή τιμή για το δικό του όφελος, και όχι για το καλό της εμπορικής κοινότητας σαν σύνολο. Έτσι αυτοί οι πράκτορες εργάζονται για κάποιο συγκεκριμένο χρήστη και όχι για κάποια ενοποιημένη κοινότητα χρηστών.

- **Συνεργατικοί πράκτορες:** Σε αντίθεση με τους προηγούμενους, οι συνεργατικοί πράκτορες μοιράζονται τη γνώση τους και τα πιστεύω τους και προσπαθούν να μεγιστοποιήσουν το όφελος της κοινότητας στο σύνολό της. Ένα σενάριο προσομοίωσης ελέγχου της εναέριας κυκλοφορίας είναι χαρακτηριστικό. Κάθε πράκτορας αντιπροσωπεύει έναν πιλότο ή αεροπλάνο και με τη βοήθεια των πρακτόρων που αντιπροσωπεύουν το κέντρο ελέγχου της εναέριας κυκλοφορίας, όλοι μαζί, λαμβάνουν μέρος σε μία συνεργατική διεργασία προγραμματισμού του καλύτερου σχεδίου για την προσέγγιση και απομάκρυνση των αεροπλάνων από το αεροδρόμιο. Στο παράδειγμα αυτό η διαπραγμάτευση γίνεται σε συνεργατικό επίπεδο, με τους πράκτορες να προσπαθούν να λύσουν τις συγκρούσεις τους μέσω συνεργατικών συζητήσεων και όχι μέσω ανταγωνιστικών δημοπρασιών.

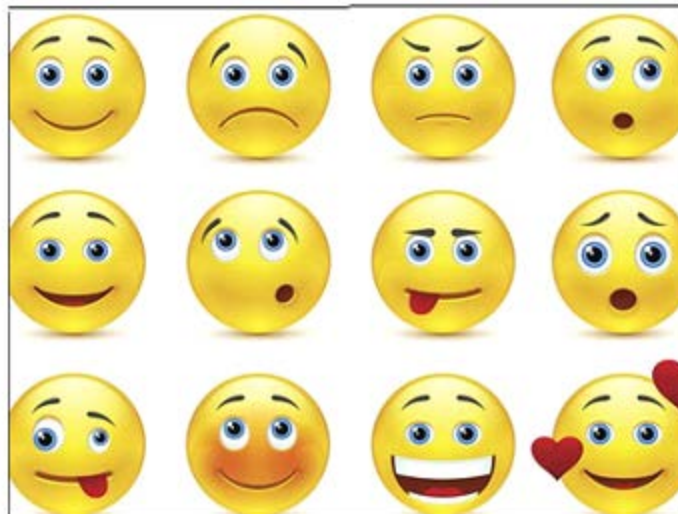
Ένα από τα σημαντικότερα ζητήματα που πρέπει να αντιμετωπιστούν στα συστήματα MAS είναι αυτό της επικοινωνίας μεταξύ των πρακτόρων, με δύο γενικές κατηγορίες λύσεων να προτείνονται: την απευθείας επικοινωνία μεταξύ τους, ή την επικοινωνία μέσω μεσολαβητή.



Εικόνα 14 Πολυπρακτορικό Σύστημα

3.7 Συναισθήματα στα υπολογιστικά συστήματα

Μερικοί άνθρωποι μιλάνε στα υπολογιστικά τους συστήματα, άλλες φορές τα χτυπάνε ή τα φιλάνε ή συμπεριφέρονται με τρόπους που προτείνουν μία κοινωνική σχέση μεταξύ χρηστών και μηχανών. Οι χρήστες έχουν συναισθήματα όταν αλληλεπιδρούν με τους υπολογιστές αλλά τα υπολογιστικά συστήματα δεν ανταποκρίνονται ακριβώς στα συναισθήματα του χρήστη από τη στιγμή που αυτές οι εφαρμογές είναι σπάνια σχεδιασμένες με τέτοιες δυνατότητες. Για να λυθεί αυτό το θέμα και να προστεθεί το συναισθηματικό στοιχείο της επικοινωνίας μεταξύ χρηστών και συστημάτων, οι ερευνητές πρότειναν ένα ανθρωπομορφικό χαρακτήρα ο οποίος θα ενσωματώνει το συναίσθημα και θα δρα σαν ενδιάμεσος μεταξύ του χρήστη και της μηχανής. Μέχρι τότε, οι πράκτορες με συναισθήματα χρησιμοποιούνταν ευρέως σε διάφορες εφαρμογές και για διαφορετικούς σκοπούς. Κάποιες φορές σε απλά συστήματα πρακτόρων, κάποιες άλλες σε πολυπρακτορικά συστήματα, έπαιζαν πάντα το ρόλο του προσωπικού βοηθού του χρήστη. Μπορεί να έχουν ανθρωπομορφικά χαρακτηριστικά, μπορούν να κινηθούν, να μιλήσουν και ουσιαστικά δεν εκφράζουν μόνο συναισθήματα αλλά είναι εξοπλισμένοι με συναισθηματική ευφυΐα που τους επιτρέπει να κατανοούν τη συμπεριφορά του χρήστη.



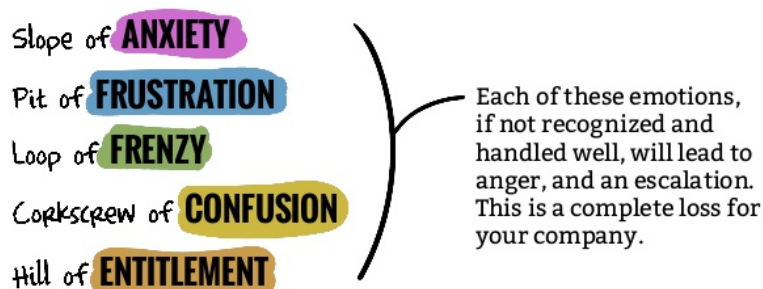
Εικόνα 15 Συναισθήματα στα υπολογιστικά συστήματα

Οι αρχιτεκτονικές των πρακτόρων αφορούν συνήθως μηχανές που έχουν σχεδιαστεί για να ενεργούν σε δυναμικά περιβάλλοντα. Οι περισσότεροι πράκτορες έχουν πολλαπλά κίνητρα για να αποκτήσουν πληροφορίες σχετικά με την κατάσταση του περιβάλλοντός τους μέσω των αισθήσεων. Ένας αυτόνομος πράκτορας πρέπει να συντηρείται με την πάροδο του χρόνου και να είναι η πηγή του αυτό-ελέγχου του. Αυτόνομη συμπεριφορά συνεπάγεται ότι

ένας πράκτορας έχει τη δική του «ατζέντα» μέσα σε έναν κόσμο. Για παράδειγμα, σκεφτείτε ένα πρόγραμμα επεξεργασίας κειμένου που δέχεται παθητικά εντολές από ένα χρήστη και μόνο περιστασιακά προσπαθεί να ελέγξει τι δέχεται σαν είσοδο πετώντας ένα μήνυμα λάθους ή απορρίπτοντας το περιεχόμενο. Σε αντίθεση, ένας αυτόνομος πράκτορας θα έχει μία σειρά από ανησυχίες τις οποίες θα πρέπει να ικανοποιήσει μέσα από τη δική του δράση. Συνεπώς, ένας ενεργός πράκτορας δεν χρειάζεται έναν επιπλέον χρήστη για να ελέγχει τις ενέργειές του.

How can we **HELP** ourselves and our **AGENTS**
MANAGE them more **SUCCESSFULLY**?

THE BIG 5 (that cause the most issues)



Εικόνα 16 Συναισθήματα στα υπολογιστικά συστήματα

Οι άνθρωποι και τα περισσότερα ζώα ταυτίζονται με τον ορισμό των αυτόνομων πρακτόρων επειδή έχουν όλοι συναισθηματικότητα. Δύο ερωτήματα προκύπτουν άμεσα, α) τι είναι συναισθηματικότητα και β) γιατί είναι μία τόσο σημαντική πτυχή των περισσότερων αυτόνομων πρακτόρων, ειδικά στους ανθρώπους. Οι απαντήσεις σε αυτά τα ερωτήματα είναι αλληλένδετες. Η εξέταση των απαιτήσεων για την αυτονομία και την ανάλυση του τι συνιστά συναισθηματικό φαινόμενο, συγκλίνουν σε μία αρχιτεκτονική απάντηση. Ένα ουσιαστικό χαρακτηριστικό πολλών συναισθηματικών καταστάσεων είναι ότι προκαλούν μερική απώλεια του ελέγχου της προσοχής. Για παράδειγμα, μία μητέρα η οποία θρηνεί για την αναχώρηση της κόρης της, δεν είναι σε θέση να σκεφτεί τίποτα άλλο από το γεγονός ότι έχει κάποια όνειρα και στόχους να εκπληρώσει.

Σε δυναμικούς και απρόβλεπτους τομείς όπως είναι ο πραγματικός κόσμος, οι πράκτορες έρχονται συνέχεια αντιμέτωποι με νέες απαιτήσεις και περιορισμούς σχετικά με την ποιότητα και τις λύσεις που παράγουν. Οποιοσδήποτε σχεδιασμός πράκτορα θα περιορίζεται πάντα με κάποιο τρόπο. Ένα άμεσο πρόβλημα είναι το τι συνιστά έναν «εαυτό». Αν δεν απαντήσουμε σε αυτό το ερώτημα, δεν θα είμαστε σε θέση να κάνουμε διαχωρισμό μεταξύ των αυτό-αναφορικών και άλλου τύπου μηχανισμών. Ο «εαυτός» είναι μία εμφανώς αφηρημένη έννοια

και έχει διάφορες ερμηνείες σε διαφορετικά πεδία έρευνας. Στο πεδίο των πρακτόρων, «εαυτός» θεωρείται ένας αυτό-αναφορικός μηχανισμός που σχεδιάζεται σε συγκεκριμένο χρόνο καθώς και τα αρχιτεκτονικά στοιχεία που αναφέρονται σε αυτόν το μηχανισμό. Οι αρχιτεκτονικές των πρακτόρων είναι δυναμικές και εξελίσσονται συνεχώς. Μπορούμε τώρα να συνδέσουμε την αυτονομία με τον αυτό-έλεγχο. Οι απαιτήσεις που οριοθετούνται από μία αυτονομία που βασίζεται σε πηγές θέτουν κάποια προβλήματα σχεδιασμού. Μία ευρέως αποδεκτή λύση σχεδιασμού είναι ένας μηχανισμός με διάφορα φίλτρα, ο οποίος προστατεύει την επεξεργασία με περιορισμένους πόρους. Το πρόβλημα που προκύπτει όμως εδώ είναι ότι μία περίπλοκη αρχιτεκτονική μπορεί να δημιουργήσει καταστάσεις που είναι δύσκολο να προσαρμοστούν. Τη λύση στο συγκεκριμένο πρόβλημα θα δώσει ένας αυτό-αναφορικός μηχανισμός, ένας μηχανισμός αυτό-ελέγχου δηλαδή που μπορεί να ανιχνεύσει προβληματικές καταστάσεις και προσπαθεί να τις διορθώσει.

3.7.1 Γιατί μελετάμε τα συναισθήματα

Καθώς οι άνθρωποι χρησιμοποιούν συναισθηματικές λέξεις για να περιγράψουν διάφορες εσωτερικές καταστάσεις αλλά και ορατές συμπεριφορές, μπορεί να φαίνεται ότι ήδη γνωρίζουμε τι είναι τα συναισθήματα. Για παράδειγμα, οι παρατηρητές ενός άντρα που κλωτσάει ένα αυτοκίνητο, θα τον περιέγραφαν ως θυμωμένο. Αν ρωτηθεί ο άντρας πώς ένιωθε, είναι πολύ πιθανό να απαντήσει ακριβώς το ίδιο. Αυτό όμως οδηγεί σε λάθος συμπεράσματα πολλές φορές. Η ικανότητα να ξέρει κάποιος ακριβώς ποιες συνθήκες οδήγησαν σε μία συγκεκριμένη συναισθηματική κατάσταση και να αναγνωρίζει τη συναισθηματική κατάσταση ενός ή πολλών ανθρώπων, απαιτεί ένα σετ από πολύ χρήσιμες δεξιότητες που συνήθως χρειάζονται αρκετά χρόνια για να αποκτηθούν.

Πέρα από το ενδιαφέρον για να εξηγηθεί ένα φυσικό φαινόμενο, η μελέτη των συναισθημάτων μπορεί να βρει πρακτική εφαρμογή σε πολλές επιστημονικές περιοχές. Για παράδειγμα, η καλύτερη κατανόηση των συναισθημάτων μπορεί να οδηγήσει σε καλύτερες κλινικές πρακτικές σε σχέση με τις συναισθηματικές διαταραχές, όπως η κατάθλιψη. Πληροφορίες που προκύπτουν από την επεξεργασία της θεωρίας των συναισθημάτων, μπορούν να βοηθήσουν τους επιστήμονες να κατανοήσουν καλύτερα τις διάφορες περιοχές του εγκεφάλου. Η ανάπτυξη θεωριών κατανόησης των μηχανισμών των συναισθημάτων μπορούν να "ταιριάξουν" με το πεδίο της Τεχνητής Νοημοσύνης.

3.7.2 Πώς ερμηνεύονται τα συναισθήματα

Τα συναισθήματα μπορούν να εξηγηθούν με διαφορετικούς τρόπους ανάλογα με τις ερωτήσεις που γίνονται. Για παράδειγμα, αν κάποιος ρωτήσει πώς προκύπτει ένα

συναίσθημα, πρέπει να πάρει μία απάντηση για το μηχανισμό παραγωγής συναισθημάτων. Εναλλακτικά, κάποιος μπορεί να ρωτήσει γιατί υπάρχουν τα συναισθήματα. Επομένως, πρέπει να πάρει μία απάντηση σχετικά με τους λόγους εξέλιξης των συναισθηματικών καταστάσεων και συμπεριφορών. Κάποιος άλλος μπορεί να ρωτήσει για την ανάπτυξη των συναισθημάτων ατομικά, τότε δηλαδή τα συναισθήματα είναι έμφυτα ή μαθαίνονται ή και τα δύο και τι είδους συναισθηματικές αλλαγές μπορούν να γίνουν κατά τη διάρκεια του κύκλου ζωής. Τέλος, κάποιος άλλος μπορεί να ρωτήσει το λόγο ύπαρξης των συναισθημάτων, το οποίο απαιτεί την εξήγηση της λειτουργίας των συναισθημάτων. Οι διαφορετικές ερωτήσεις οδηγούν τους ερευνητές στο να επικεντρωθούν σε διαφορετικές πτυχές των συναισθηματικών φαινομένων. Αυτή η εργασία επικεντρώνεται κυρίως σε ερωτήσεις "πώς" και "τι", πώς δηλαδή ένας πράκτορας μπορεί να αντιδράσει γρήγορα και άμεσα σε πραγματικά γεγονότα και ταυτόχρονα να σχεδιάσει τις μελλοντικές του κινήσεις. Η σημασία του συναισθήματος έχει ταυτοποιηθεί πρόσφατα με την ανθρώπινη ευφυΐα. Για να βελτιωθεί η ικανότητα ενός πράκτορα να διαχειρίζεται συναισθήματα, προτείνεται μία αρχιτεκτονική βασισμένη στο παραδοσιακό μοντέλο πρακτόρων. Αρχικά καθορίζεται μία νέα βάση γνώσεων για την έκφραση συναισθημάτων, στη συνέχεια εκτελείται ένας αλγόριθμος συναισθηματικής λογικής και τέλος εφαρμόζεται η «θεραπεία» συναισθημάτων. Τα αποτελέσματα από πειράματα που έγιναν πάνω σε έναν πράκτορα βασισμένο στη συγκεκριμένη αρχιτεκτονική, έδειξαν ότι είναι περισσότερο αποτελεσματικό να υπάρχουν απλά συναισθήματα μεταξύ των πρακτόρων. Στο παρελθόν τα συναισθήματα απορρίφθηκαν γιατί θεωρήθηκε ότι αποσπούν την προσοχή σε κάποιον που προσπαθεί να σκεφτεί λογικά και επιστημονικά. Πιο πρόσφατα ωστόσο, η σημασία των συναισθημάτων στην ανθρώπινη νοημοσύνη έχει αναγνωριστεί (Turing, 1950). Η έρευνα πάνω στους πράκτορες συναισθημάτων βοηθάει ώστε να αποδειχθούν τα έξυπνα χαρακτηριστικά των πρακτόρων (McCarthy, 1960). Στη ψυχολογία, το συναίσθημα εξηγείται ως οι αξιολογήσεις των πρακτόρων ξεχωριστά ή της σχέσης των πρακτόρων με το περιβάλλον. Η συναισθηματική αντίδραση των ανθρώπων προέρχεται από τη διέγερση του εξωτερικού περιβάλλοντος και επηρεάζεται από το μηχανισμό προσφοράς και ζήτησης που έχουν στον οργανισμό τους. Από την οπτική γωνία του πράκτορα, τα συναισθήματα είναι αξιολογήσεις και αντιδράσεις για την τρέχουσα εσωτερική του κατάσταση. Ως εκ τούτου, γίνεται προσπάθεια να βρεθεί κάποια αρχιτεκτονική που θα βοηθάει τους πράκτορες να χειρίζονται τα συναισθήματά τους. Μία τέτοια αρχιτεκτονική, αρχικά καθιερώνει μία νέα βάση γνώσης για την έκφραση των συναισθημάτων, στη συνέχεια εκτελεί έναν αλγόριθμο συναισθηματικής λογικής και τέλος, εφαρμόζει τη θεραπεία συναισθημάτων.

3.8 Πράκτορες Διασύνδεσης

Ένας Πράκτορας Διασύνδεσης (Π.Δ.) ή Πράκτορας Διεπαφής μπορεί να οριστεί ως μια οντότητα λογισμικού, η οποία έχει σαν σκοπό, να ενεργεί για λογαριασμό του χρήστη σε ένα εικονικό περιβάλλον. Ο Π.Δ. μαθαίνει από την συμπεριφορά του χρήστη και εκπαιδεύεται, προκειμένου να τον βοηθήσει ενεργά στις εργασίες που εκτελεί σε διάφορες εφαρμογές. Ο επικρατέστερος ορισμός για τους πράκτορες διεπαφής δόθηκε από τον Maes και είναι ο εξής: «Αντί για την αλληλεπίδραση του χρήστη με άλλους χρήστες ή με το σύστημα, μέσω εντολών και την άμεση χειραγώγησή του. Ο χρήστης εμπλέκεται σε μία συνεργατική διαδικασία στην οποία άνθρωποι και πράκτορες επικοινωνούν, παρακολουθούν γεγονότα και εκτελούν καθήκοντα (Maes, 2007). Η παραγωγικότητα και η ευκολία πρόσβασης σε πληροφορίες αποτελούν κύρια συστατικά της αποδοτικής χρήσης υπολογιστικών συστημάτων. Οι Πράκτορες Διασύνδεσης (Π.Δ.) ή Προσωπικοί Βοηθοί, είναι αυτόνομες εφαρμογές λογισμικού που χρησιμοποιούνται ευρέως για την παροχή εξατομικευμένης υποστήριξης σε χρήστες εφαρμογών υπολογιστών. Ως αυτόνομες οντότητες μεσολαβούν μεταξύ του χρήστη και του υπόλοιπου πληροφοριακού συστήματος. Ένα από τα κύρια χαρακτηριστικά τους είναι η δυνατότητα να μαθαίνουν από τις προτιμήσεις του χρήστη, και να του παρέχουν ενεργητική υποστήριξη, με σκοπό την αύξηση της παραγωγικότητας. Παρέχονται έτσι σημαντικά οφέλη επιτρέποντας τη μετάβαση από τον άμεσο χειρισμό των επιμέρους τελικών εφαρμογών, στην έμμεση αλληλεπίδραση με Προσωπικούς Βοηθούς. Η κεντρική ιδέα πίσω από τους πράκτορες διεπαφής είναι να επιτρέψει στο χρήστη να αναθέσει χρονοβόρες και περίπλοκες εργασίες σε ένα βοηθητικό μέσο με το οποίο θα συνεργάζεται στο ίδιο περιβάλλον εργασίας. Για παράδειγμα, ένας πράκτορας προγραμματισμού αναθέτει σε έναν βοηθητικό πράκτορα τον προγραμματισμό των συναντήσεων, το φιλτράρισμα των ηλεκτρονικών μηνυμάτων και την επιλογή των καλών βιβλιοθηκών. Στόχος είναι να μειωθεί ο φόρτος εργασίας των χρηστών με τη δημιουργία εξειδικευμένων βοηθητικών πρακτόρων στους οποίους θα ανατίθεται προσωπική εργασία. Οι Π.Δ. είναι προγράμματα υπολογιστών που χρησιμοποιούν τεχνικές της τεχνητής νοημοσύνης για να βοηθήσουν ενεργά τους χρήστες με τις εργασίες που εκτελούν σε διάφορες εφαρμογές (Maes, 2007). Ο στόχος ενός έξυπνου Π.Δ. είναι να βοηθά τον τελικό χρήστη να εκτελέσει τις εργασίες του στον υπολογιστή σε ένα πιο φιλικό περιβάλλον, ώστε να μειωθούν οι εργασίες που έχει να εκτελέσει ο χρήστης (Aboulenien and P. De Wilde, 1998). Ένας πράκτορας μπορεί να αλλάξει την εμπειρία που έχει ο χρήστης από μια εφαρμογή δρώντας σαν ένας προσωπικός βοηθός. Ο Π.Δ. δεν ενεργεί ως μεσάζων μεταξύ του χρήστη και της εφαρμογής, αλλά συμπεριφέρεται ως ένας προσωπικός βοηθός που συνεργάζεται με το χρήστη, όπου φυσικά ο χρήστης έχει τη δυνατότητα να παρακάμψει τον πράκτορα.



Εικόνα 17 Πράκτορες Διασύνδεσης

Οι Π.Δ. αποκτούν τις ικανότητές τους μαθαίνοντας από το χρήστη, καθώς και από πράκτορες που βοηθούν άλλους χρήστες. Οι πράκτορες έχουν τη δυνατότητα να μαθαίνουν από τις ενέργειες που εκτελεί ο ίδιος ο χρήστης και να συγκεντρώνουν την εμπειρία παρατηρώντας και αλληλεπιδρώντας μαζί του. Επίσης οι πράκτορες μπορούν να αποκτήσουν εμπειρία ζητώντας από άλλους πράκτορες να τους στείλουν τις δικές τους εμπειρίες που έχουν συγκεντρώσει από τους δικούς τους χρήστες. Υπάρχουν πολλά συστήματα πρακτόρων διεπαφής που βρίσκουν εφαρμογή σε πολλά πεδία. Το κοινό με αυτά τα συστήματα είναι ότι υπάρχουν τρία ζητήματα τα οποία πρέπει να αντιμετωπιστούν πριν από την επιτυχημένη συνεργασία του χρήστη με έναν πράκτορα:

- Γνωριμία με το χρήστη
- Αλληλεπίδραση με το χρήστη
- Ανταγωνισμός βοηθώντας το χρήστη

Η γνωριμία με το χρήστη περιλαμβάνει την εκμάθηση των προτιμήσεών του και των εργασιακών του συνηθειών. Αν ένας βοηθητικός πράκτορας θέλει να προσφέρει βοήθεια στο σωστό χρόνο και τη σωστή στιγμή, θα πρέπει να μάθει πως προτιμά να εργάζεται ο χρήστης. Ένας βοηθός που πάντα διακόπτει με άσχετες πληροφορίες, μόνο θα ενοχλήσει και θα αυξήσει το συνολικό φόρτο εργασίας. Οι ακόλουθες προκλήσεις προκύπτουν από την προσπάθεια των συστημάτων να μάθουν για τους χρήστες:

- Εξαγωγή στόχων και προθέσεων των χρηστών από τις παρατηρήσεις και τα σχόλια.
- Προσαρμογή στην αλλαγή των στόχων του χρήστη.
- Μείωση του αρχικού χρόνου εκπαίδευσης.

Σε κάθε δεδομένη στιγμή, ένας πράκτορας διεπαφής πρέπει να γνωρίζει τι προσπαθεί ο χρήστης να επιτύχει, προκειμένου να είναι σε θέση να προσφέρει αποτελεσματική βοήθεια. Εκτός από τη γνώση των προθέσεων του χρήστη, πρέπει να υπάρχουν και επαρκείς σχετικές πληροφορίες για να αποφευχθεί βοήθεια που δεν χρειάζεται. Οι τεχνικές μηχανικής μάθησης μπορούν να βοηθήσουν στη συγκεκριμένη περίπτωση. Ένα άλλο πρόβλημα είναι ότι οι τακτικοί χρήστες έχουν συνήθως να εκτελέσουν πολλές εργασίες ταυτόχρονα. Αν ένας πράκτορας θέλει να είναι χρήσιμος, πρέπει να είναι σε θέση να καταλάβει πότε ένας χρήστης έχει σταματήσει να ασχολείται με μία εργασία και έχει προχωρήσει στην επόμενη. Αλλά ποιος είναι ο καλύτερος τρόπος για να το αντιμετωπίσει; Οι χρήστες είναι γενικά απρόθυμοι να επενδύσουν πολύ χρόνο και προσπάθεια για να εκπαιδεύσουν τα συστήματα λογισμικού. Αυτό σημαίνει ότι οι πράκτορες διεπαφής πρέπει να περιορίσουν την αρχική περίοδο κατά την οποία ο πράκτορας μαθαίνει αρκετά πράγματα για το χρήστη. Τι αντίκτυπο έχει όμως αυτό στην ικανότητα εκμάθησης του πράκτορα; Οι χρήστες πρέπει να αισθάνονται ότι ελέγχουν την κατάσταση, οι προσδοκίες δεν θα πρέπει να είναι πολύ «φουσκωμένες» και τα λάθη δεν θα πρέπει να τιμωρούνται (Norman, 1994). Από την αλληλεπίδραση με το χρήστη προκύπτουν οι ακόλουθες προκλήσεις:

- Αποφασίζεται πόσο έλεγχος πρέπει να δοθεί τον πράκτορα.
- Χτίζεται η εμπιστοσύνη με τον πράκτορα.
- Επιλέγεται ο τρόπος αλληλεπίδρασης με τον πράκτορα.
- Δημιουργία απλών συστημάτων που θα χρησιμοποιούν και οι αρχάριοι.

Δύο βασικοί λόγοι δικαιολογούν την αυτόματη παραγωγή των διεπαφών χρήστη από την εργασία, τον τομέα και τα μοντέλα πλαισίων. Ο πρώτος είναι η ομοιότητα αυτών των μοντέλων. Προκειμένου να αναπτυχθεί ένα μοντέλο, ο προγραμματιστής ξεκινάει συνήθως «χτίζοντας» το έργο, τον τομέα και τα μοντέλα πλαισίων έτσι ώστε να είναι βολικό να δημιουργήσει τη διεπαφή του χρήστη με βάση αυτούς τους πόρους. Ο δεύτερος είναι η φύση των μοντέλων από μόνη της. Πράγματι, υπάρχουν τμήματα του σχεδιασμού της διεπαφής του χρήστη περισσότερο ακριβή:

- 1 Το μοντέλο του έργου εκφράζει πως ένας τελικός χρήστης μπορεί να θέλει να αλληλεπιδράσει με ένα σύστημα προκειμένου να επιτευχθεί ένας συγκεκριμένος στόχος. Αυτή η έκφραση είναι ανεξάρτητη από οποιαδήποτε συγκεκριμένη εφαρμογή ή τεχνολογία. Αυτό εξηγεί γιατί ένα ίδιο σετ μοντέλων μπορεί να εξυπηρετήσει διαφορετικές διεπαφές χρήστη. Το μοντέλο του έργου χρησιμοποιείται για να καθορίσει μία γενική διεπαφή χρήστη.
- 2 Το μοντέλο του τομέα καθορίζει τις πτυχές της εφαρμογής που μπορούν να προσαρμοστούν ή απαιτούνται για τη λειτουργία του συστήματος. Το μοντέλο του

τομέα χρησιμοποιείται για να καθορίσει τον έλεγχο της διεπαφής χρήστη. Σε αυτό το επίπεδο ο χρήστης διεπαφής καθορίζεται με περισσότερες λεπτομέρειες.

- 3 Το μοντέλο πλαισίου περιγράφει τις ικανότητες του χρήστη, το περιβάλλον στο οποίο ο χρήστης εργάζεται και την πλατφόρμα που μπορεί να χρησιμοποιηθεί για να αλληλεπιδράσει με το σύστημα. Το συγκεκριμένο μοντέλο χρησιμοποιείται για να επηρεάσει το σχεδιασμό και να επιλέξει μεταξύ εναλλακτικών λύσεων στο χώρο του σχεδιασμού.

Είναι γνωστό από διεπαφές άμεσης χειραγώγησης ότι οι χρήστες θέλουν να έχουν πλήρη έλεγχο της χρήσης των εργαλείων τους. Λόγω της φύσης ενός αυτόνομου πράκτορα διασύνδεσης, μερικός έλεγχος έχει ανατεθεί σε αυτόν για να μπορέσει να ολοκληρώσει το έργο του. Το ερώτημα είναι πώς χτίζουμε την εμπιστοσύνη των χρηστών και όταν καθιερωθεί ένα επίπεδο εμπιστοσύνης, πόσο έλεγχο δίνουμε στους πράκτορες. Ο Shneiderman τάσσεται υπέρ ενός συνδυασμού άμεσης και έμμεσης χειραγώγησης που προάγει την κατανόηση των πρακτόρων και την ικανότητα των χρηστών να ελέγχουν τη συμπεριφορά τους άμεσα. Οι μεταφορές διασύνδεσης όπως είναι η επιφάνεια εργασίας καθοδηγούν τους χρήστες στο σχηματισμό ενός σωστού εννοιολογικά μοντέλου. Νέες μεταφορές θα χρειαστούν για την έμμεση χειραγώγηση, παρουσιάζοντας τους πράκτορες κατά τέτοιο τρόπο, εύκολα κατανοητό από τους νέους χρήστες στο σύστημα. Στην ιδανική περίπτωση, οι πράκτορες διασύνδεσης πρέπει να είναι τόσο απλοί στη χρήση που η ανάθεση καθηκόντων γίνεται ένας φυσικός τρόπος εργασίας ακόμα και για τον αρχάριο χρήστη. Τέλος υπάρχει το ζήτημα της αρμοδιότητας. Μόλις ο πράκτορας μάθει τι κάνει ο χρήστης και έχει ένα καλό στυλ αλληλεπίδρασης, θα πρέπει να εφαρμόσει ένα σχέδιο δράσης που βοηθάει και δεν εμποδίζει το χρήστη στην ολοκλήρωση των εργασιών. Οι προκλήσεις που προκύπτουν είναι οι εξής:

- Να γνωρίζει ο πράκτορας πότε και αν πρέπει να διακόψει το χρήστη.
- Να εκτελεί ο πράκτορας τα καθήκοντα αυτόνομα με τρόπο που προτιμάει ο χρήστης.
- Εύρεση στρατηγικών από τον πράκτορα για την μερική αυτοματοποίηση των εργασιών.

Αν ένας πράκτορας δεν καταφέρει να μειώσει το φόρτο εργασίας ενός πραγματικού χρήστη σε ένα πραγματικό περιβάλλον, τότε είναι κάτι λιγότερο από χρήσιμος.

3.8.1 Χαρακτηριστικά Πρακτόρων Διασύνδεσης

Οι πράκτορες χρησιμοποιούνται ευρέως σήμερα σε ένα μεγάλο πλήθος εφαρμογών. Όλοι οι πράκτορες παρουσιάζουν κάποια κοινά χαρακτηριστικά, όπως (Nwana, 1996):

-
- **Αυτονομία:** Είναι η δυνατότητα λειτουργίας τους χωρίς την άμεση παρέμβαση του ανθρώπου και το να έχουν κάποιο έλεγχο στις ενέργειές τους.
 - **Συνεργασία:** Η αλληλεπίδραση με άλλους πράκτορες ή και ανθρώπους μέσω ενός κοινού τρόπου επικοινωνίας.
 - **Μάθηση:** Οι πράκτορες πρέπει να μπορούν να αντιλαμβάνονται το περιβάλλον τους και να ανταποκρίνονται στις αλλαγές που συμβαίνουν σε αυτό. Η ικανότητα ενός πράκτορα να μπορεί να πάρει τις αποφάσεις με βάση την εμπειρία που έχει αποκτήσει είναι η μάθηση.

Στα χαρακτηριστικά των Π.Δ. εμφανίζονται όλα τα προαναφερθέντα. Οι Π.Δ. επικεντρώνονται περισσότερο στην αυτονομία και στη μάθηση, ενώ το χαρακτηριστικό της συνεργασίας με άλλους πράκτορες θα εμφανιζόταν σε ένα σχήμα όπου πολλοί πράκτορες θα συνεργαζόταν μεταξύ τους.

Σε ένα πραγματικό περιβάλλον εργασίας, ο Π.Δ. συνεργάζεται με το χρήστη σε μια εφαρμογή. Ο βοηθός σταδιακά γίνεται πιο αποτελεσματικός, δεδομένου ότι μαθαίνει τα ενδιαφέροντα του χρήστη, τις συνήθειες και τις προτιμήσεις του (Norman, 1994).

- Οι Π.Δ. μπορούν να βοηθήσουν τους χρήστες με μια σειρά από διαφορετικούς τρόπους:
- Εκτελούν εργασίες για λογαριασμό του χρήστη.
- Μπορούν να εκπαιδεύσουν ή να διδάξουν το χρήστη.
- Βοηθούν τους διαφορετικούς χρήστες να συνεργάζονται μεταξύ τους.
- Παρακολουθούν τα γεγονότα και τις διαδικασίες.

Οι εργασίες και οι εφαρμογές που μπορεί να έχει ένας Π.Δ. είναι σχεδόν απεριόριστες: Φιλτράρισμα και ανάκτηση πληροφοριών, διαχείριση mail, προγραμματισμός συναντήσεων, διαχείριση συλλογών (βιβλία, ταινίες, μουσική κτλ.).

3.8.2 Αρχιτεκτονικές και Αλγόριθμοι

Όσον αφορά το μοντέλο αλληλεπίδρασης με τον χρήστη, πολύ διαδεδομένη είναι η αρχιτεκτονική MI²A (Model-Based Intelligent Interface Agent Architecture). Σε αυτήν υπάρχει ένας πράκτορας παρατηρητής (Observer Agent), ο οποίος αναλαμβάνει να παρακολουθεί και να συλλέγει τις ενέργειες των χρηστών, να επιλέγει τις πιο σημαντικές και να τις στέλνει στον πράκτορα διαχείρισης του μοντέλου (User Model Manager Agent). Αυτός δημιουργεί και αποθηκεύει το μοντέλο αλληλεπιδράσεων το οποίο θα χρειαστεί για να κατευθύνει την δράση του στο μέλλον.

- **Action Fragmentation Algorithm**

Ο αλγόριθμος τμηματοποίησης δράσης βασίζει τις ενέργειες του στο μοντέλο εργασίας που κατασκευάστηκε με τον προηγούμενο τρόπο και χαρτογραφεί τις ενέργειες του σε έναν πίνακα καθηκόντων που εξαρτάται από τις ενέργειες του χρήστη, προκειμένου να δημιουργήσει μια αλυσίδα ενεργειών.

Άλλοι ενδιαφέροντες αλγόριθμοι για την παραπάνω αρχιτεκτονική είναι ο action pair selection algorithm, ο interaction graph construction algorithm και ο probability calculation algorithm (Koutsabasis, 1999).

- **WOS learning algorithm**

Στόχος του αλγορίθμου είναι η εκμάθηση των πλαισίων που προτιμά ο χρήστης για την επίλυση ενός προβλήματος, ή για την αντιμετώπιση κάποιας κατάστασης. Επιτρέπει σε έναν Π.Δ. να επιλέξει την πιο αποδεκτή ενέργεια για τον χρήστη στη δεδομένη κατάσταση. Η είσοδος στον αλγόριθμο είναι μια σειρά από εμπειρίες των πρακτόρων του χρήστη (Schiaffino and Amandi, 2006).

- **ATIA Algorithm**

Ο ATIA (Algorithm teaching interface agents) είναι ένας προσαρμοστικός και δυναμικός αλγόριθμος που διδάσκει αλγορίθμους, δρώντας ως μεσολαβητής ανάμεσα στον μαθητή και το περιβάλλον. Η γνώση του ATIA περιέχει πληροφορίες που είναι αποθηκευμένες στη μνήμη του σχετικά με το περιβάλλον και τον αναμενόμενο κόσμο. Περιλαμβάνει το μοντέλο μαθητή και το μοντέλο γνώσης του συστήματος και αποτελείται από τρία συστατικά. Ένα για την επικοινωνία, ένα για την επίλυση προβλημάτων και ένα για την αναβάθμιση γνώσης (Shakshuk, 2004).

3.8.3 Αξιολόγηση των Πρακτόρων Διασύνδεσης

Η αξιολόγηση, αλλιώς ονομάζεται και εκτίμηση, παίζει πολύ σημαντικό ρόλο στο σχεδιασμό των διεπαφών χρήστη που βασίζονται στους πράκτορες, τόσο για την ανάπτυξη του συστήματος όσο και για τους υποψήφιους χρήστες. Σύμφωνα με τους Hirschman και Thompson, τρεις είναι οι βασικοί τύποι αξιολόγησης:

- Αξιολόγηση της επάρκειας.
- Αξιολόγηση των διαγνωστικών.
- Αξιολόγηση των επιδόσεων.

Συνήθως, για έναν υποψήφιο χρήστη, η αξιολόγηση της επάρκειας είναι ο προσδιορισμός του πόσο καλά ένα σύστημα ταιριάζει με ένα σκοπό και κατά πόσο ανταποκρίνεται στις πραγματικές ανάγκες και προσδοκίες των χρηστών. Η διαγνωστική αξιολόγηση χρησιμοποιείται συνήθως από τους προγραμματιστές ενός συστήματος και ουσιαστικά είναι η

δημιουργία ενός προφίλ επιδόσεων για την ανίχνευση και την ανάλυση του σχεδιασμού και των σφαλμάτων εφαρμογής σε σχέση με κάποια κενά διαστήματα (Hirschmann and Thompson, 2004). Η αξιολόγηση των επιδόσεων συνήθως δημιουργείται για τους προγραμματιστές και είναι η μέτρηση της απόδοσης του συστήματος σε έναν ή περισσότερους συγκεκριμένους τομείς. Οι μεθοδολογίες αξιολόγησης για τα πολυπρακτορικά συστήματα μπορεί να είναι ποσοτικές ή ποιοτικές, αντικειμενικές ή υποκειμενικές. Η ποσοτική αξιολόγηση συνίσταται στην ποσοτικοποίηση κάποιας παραμέτρου μέσω ενός ανεξάρτητου αριθμού χωρίς ιδιαίτερο νόημα, όπως είναι ο χρόνος για να ολοκληρωθεί το έργο, ο αριθμός αλληλεπιδράσεων ανά έργο και το ποσοστό επιτυχούς ολοκλήρωσης της εργασίας, η οποία επιτρέπει τη σύγκριση των συστημάτων (Dybkjær and Bernsen, 2001). Η ποιοτική αξιολόγηση συνίσταται στην εκτίμηση ή κρίση κάποιων παραμέτρων όπως το επίπεδο δυσκολίας, η φιλικότητα προς το χρήστη, η διαίσθηση, η γενική εντύπωση σε σχέση με κανόνες και πρότυπα ειδικών. Σε μία υποκειμενική αξιολόγηση, τα ανθρώπινα θέματα δοκιμής αλληλεπιδρούν με το σύστημα και οι μετρικές μπορεί να είναι ποσοτικές ή ποιοτικές. Η αντικειμενική αξιολόγηση είναι ανεξάρτητη από τα υποκείμενα, οπότε μειώνεται ο ρόλος των θεμάτων δοκιμής και η συμπεριφορά του συστήματος είναι ελεγχόμενη. Μία δοκιμή μπορεί να επαναληφθεί υπό διαφορετικές συνθήκες και σε διαφορετικές χρονικές περιόδους. Για παράδειγμα, τα σήματα ομιλίας θα καταγράφονται και μπορούν να χρησιμοποιηθούν πολλές φορές προκειμένου να ελεγχθούν διαφορετικά συστήματα κάτω από τις ίδιες συνθήκες (Leeuwen, 1998 & Gibbon, 1997).

Για την αξιολόγηση και τη σύγκριση της απόδοσης διαφορετικών πρακτόρων διαλόγου, έχει προταθεί ένα πλαίσιο, γνωστό ως PARADISE. Το PARADISE στηρίζει τις συγκρίσεις μεταξύ των στρατηγικών διαλόγου παρέχοντας μία αναπαράσταση έργου και μία θεωρητική απόφαση πλαισίου για να καθορίσει τη σχετική συμβολή των διαφόρων παραγόντων στη συνολική απόδοση ενός πράκτορα. Η απόδοση μοντελοποιείται ως λειτουργία και τα βάρη υπολογίζονται συσχετίζοντας την ικανοποίηση των χρηστών με την απόδοση του συστήματος (Walker, 1997). Επίσης, το πλαίσιο PARADISE υιοθετεί την άποψη ότι η απόδοση έχει άμεση σχέση με ένα ουσιαστικό εξωτερικό κριτήριο όπως είναι η χρηστικότητα. Ωστόσο, οι Dybkjær και Bernsen (Dybkjær and Bernsen, 2001) υποστηρίζουν ότι πολλά σημαντικά θέματα χρηστικότητας δεν μπορούν να υποβληθούν σε ποσοτικοποίηση. Έτσι, προτείνουν ένα σύνολο από κριτήρια αξιολόγησης της ευχρηστίας. Ισχυρίζονται ότι η ποιότητα της χρηστικότητας της προφορικής γλώσσας σε συστήματα διαλόγων μπορεί να συνεχιστεί με την εστίαση σε ένα ολοκληρωμένο σύνολο 15 θεμάτων ευχρηστίας όπως η καταλληλότητα τροποποίησης, η επάρκεια της αναγνώρισης εισόδου, η φυσικότητα της ομιλίας του χρήστη, η ποιότητα φωνής της παραγωγής, η φυσικότητα του διαλόγου, η επάρκεια

ανατροφοδότησης, η επάρκεια αλληλεπίδρασης, οι δυνατότητες συλλογισμών, ο χειρισμός σφαιμάτων, η ικανοποίηση των χρηστών κ.ά.

3.8.4 Ταξινόμηση των Πρακτόρων Διασύνδεσης

Μια ταξινόμηση της λειτουργίας των Π.Δ. είναι: (Middleton, 2002)

- **Πράκτορες που βασίζονται σε χαρακτήρες (Character-based agents)**
Χρησιμοποιούνται προηγμένα Interfaces, τα οποία βασίζονται σε χαρακτήρες, χρησιμοποιώντας πραγματικού κόσμου χαρακτήρες, όπως π.χ. ένα σκύλο, ή ένα άνθρωπο βοηθό.
- **Κοινωνικοί πράκτορες (Social Agents)**
Μιλούν με άλλους πράκτορες, συνήθως του ίδιου τύπου, με στόχο την ανταλλαγή πληροφοριών. Υποκατηγορία τους είναι τα Recommender Systems, τα οποία ονομάζονται και φίλτρα συνεργασίας, τα οποία είναι ένας ειδικός τύπος κοινωνικού πράκτορα για εύρεση στοιχείων τύπων με βάση τις συστάσεις άλλων πρακτόρων.
- **Πράκτορες που μαθαίνουν για τον χρήστη**

Πρόκειται για πράκτορες που παρακολουθούν τη συμπεριφορά του χρήστη και εκπαιδεύονται. Υπάρχουν τρεις διαφορετικές υποκατηγορίες:

- [1] Πράκτορες προγραμματιζόμενοι από τον χρήστη. Είναι οι πιο εύκολοι στην υλοποίηση, αλλά ταυτόχρονα δεν είναι πάντα και η πιο ενδεδειγμένη λύση, επειδή η λειτουργία τους δεν συνεπάγεται και νοημοσύνη. Ο χρήστης είναι αυτός που παρέχει απευθείας τους κανόνες, το ευρετήριο λειτουργίας και προγραμματίζει τη λειτουργία του προκειμένου να βοηθηθεί στην αλληλεπίδραση του με την εφαρμογή. Το κυριότερο πρόβλημα τους είναι η εξάρτησή τους από την ικανότητα του χρήστη στον προγραμματισμό.
- [2] Πράκτορες που λαμβάνουν ανατροφοδότηση από τον χρήστη. Αυτού του είδους οι πράκτορες συνήθως δημιουργούνται με τη χρήση συστημάτων που βασίζονται στη γνώση. Μπορούν να λάβουν είτε ρητή ανατροφοδότηση, είτε μόνο μια αρχική εκπαίδευση. Η ανάπτυξη τους είναι αρκετά πολύπλοκη, αλλά παρουσιάζουν ευφείς συμπεριφορές. Ο πράκτορας προγραμματίζεται από κάποιον ειδικό και χρησιμοποιώντας τη γνώση που λαμβάνει, βελτιώνει την αλληλεπίδραση του χρήστη με την εφαρμογή. Σε περίπτωση που ο πράκτορας λαμβάνει μόνο μια αρχική κατάσταση, δεν μπορεί να προσαρμοστεί στις συνήθειες και στις προτιμήσεις του χρήστη.
- [3] Πράκτορες που παρακολουθούν τη συμπεριφορά του χρήστη. Πρόκειται για πράκτορες που χρειάζονται κάποια γνώση του περιβάλλοντος προκειμένου

μάθουν μια συμπεριφορά και να εξάγουν μοτίβα από αυτή, λαμβάνοντας υπόψη τις ενέργειες του χρήστη ή άλλων χρηστών. Τόσο ο πράκτορας, όσο και ο χρήστης αλληλεπιδρούν με την εφαρμογή. Ωστόσο, η βασική διαφορά σε σχέση με τις προηγούμενες προσεγγίσεις, είναι ότι ο πράκτορας παρακολουθεί την αλληλεπίδραση μεταξύ χρήστη και εφαρμογής και μαθαίνει από αυτήν. Οι πράκτορες αποκτούν τις αρμοδιότητες τους από τέσσερις πηγές: 1) την παρακολούθηση του χρήστη ενώ αυτός αλληλεπιδρά με την εφαρμογή, 2) την άμεση ή έμμεση αλληλεπίδραση με τον χρήστη, 3) παραδείγματα που παρέχονται από τον χρήστη και 4) ρωτώντας άλλους πράκτορες που βοηθούν άλλους χρήστες και έχουν τις ίδιες εργασίες. Οι πράκτορες που παρακολουθούν τη συμπεριφορά των χρηστών γίνονται όλο και πιο ικανοί, καθώς μαθαίνουν τις προτιμήσεις του χρήστη και αναπτύσσουν σταδιακά τις ικανότητες τους (Aznar and Sempere, 2009).

- **Παράγοντες με μοντέλα χρήστη**

Εμφανίζονται σε δυο διαφορετικές μορφές. Συμπεριφοράς, οι οποίοι είναι γενικά το αποτέλεσμα της παρακολούθησης του χρήστη κατά τη διάρκεια μιας δραστηριότητας, και μάθησης, οι οποίοι είναι το αποτέλεσμα ερωτηματολογίων και μελετών των χρηστών. Είναι δυνατόν να εφαρμοστούν στερεότυπα και στις δυο περιπτώσεις, ταξινομώντας τους χρήστες σε ομάδες (ή στερεότυπα) με στόχο την εφαρμογή γενικεύσεων στα μέλη μιας ομάδας (Middleton, 2002).

Έχουν προταθεί διάφορες ταξινομήσεις για τους πράκτορες λογισμικού και συνήθως οι πράκτορες διασύνδεσης είναι μία ξεχωριστή κλάση από μόνοι τους. Με την ωρίμανση του πεδίου των πρακτόρων και τον αυξανόμενο αριθμό των πρακτόρων διεπαφής, μία πιο λεπτομερής ανάλυση είναι δικαιολογημένη. Ο Mladenic προσπάθησε για να επιτύχει αυτή την απαίτηση υιοθετώντας μία άποψη μηχανικής μάθησης για τους πράκτορες διεπαφής (Maes, 2007). Οι πράκτορες διασύνδεσης μπορούν να ταξινομηθούν ανάλογα με το ρόλο που επιτελούν, την τεχνολογία που χρησιμοποιούν ή την περιοχή που ανήκουν. Η βασική τεχνολογία πίσω από τους πράκτορες πάντως δεν αλλάζει ριζικά, οπότε παρέχεται μία πιο σταθερή βάση πάνω στην οποία θα βασιστεί η ταξινόμηση των πρακτόρων διασύνδεσης. Πέρα από αυτά που προαναφέρθηκαν, μία περαιτέρω ταξινόμηση των τεχνολογιών που τα ειδικά συστήματα πρακτόρων υποστηρίζουν είναι:

1. Ρητή ανατροφοδότηση
2. Αρχικό σετ εκπαίδευσης
 - Προγραμματισμός από το χρήστη
3. Πράκτορες με μοντέλα χρηστών

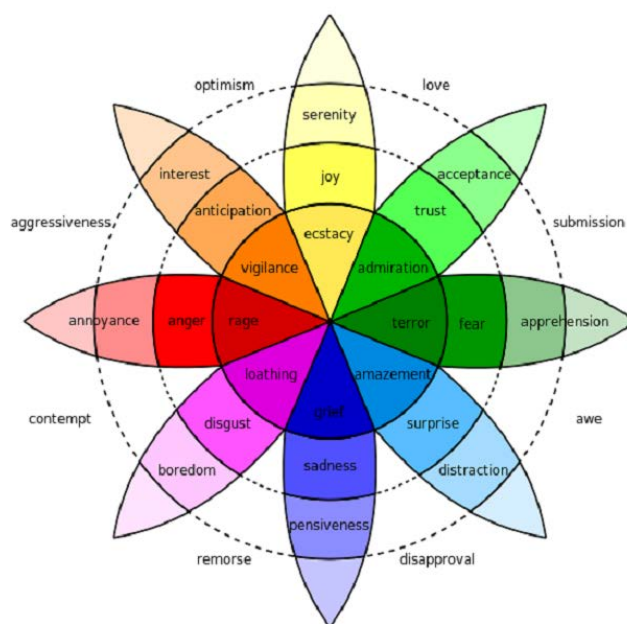
-
- Μοντέλο συμπεριφοράς
 - Μοντέλο που βασίζεται στη γνώση
 - Στερεότυπα

Οι πράκτορες που βασίζονται σε χαρακτήρες χρησιμοποιούν προηγμένες αρχιτεκτονικές διασυνδέσεων που αντιπροσωπεύουν χαρακτήρες του πραγματικού κόσμου. Τέτοιοι πράκτορες χρησιμοποιούνται στα πρωτόκολλα του πραγματικού κόσμου για να διευκολύνουν την αλληλεπίδραση μεταξύ των χρηστών (Maes, 1995). Υπάρχουν επίσης εφαρμογές στον τομέα της ψυχαγωγίας όπως δημιουργία υπερσύγχρονων εικονικών κόσμων που κατοικούνται από πιστευτούς πράκτορες. Οι κοινωνικοί πράκτορες μιλούν με άλλους πράκτορες, συνήθως με πράκτορες διασύνδεσης του ίδιου τύπου, με σκοπό την ανταλλαγή πληροφοριών. Η τεχνική αυτή χρησιμοποιείται συχνά για το ξεκίνημα των νέων άπειρων πρακτόρων διασύνδεσης. Τα συστήματα εισηγήσεων είναι ένας ειδικός τύπος κοινωνικών πρακτόρων. Αναφέρονται επίσης και ως φίλτρα συνεργασίας και στόχος τους είναι η εύρεση σχετικών στοιχείων που προτείνονται από άλλους πράκτορες. Συνήθως, η αυτό-αξιολόγηση ενός χρήστη χρησιμοποιείται για να βρεθούν παρόμοιοι χρήστες με στόχο το μοίρασμα των συστάσεων σε κοινούς τομείς ενδιαφέροντος (Kobsa, 1993). Οι πράκτορες που χρησιμοποιούν τεχνολογία εκμάθησης κατατάσσονται ανάλογα με τον τύπο των πληροφοριών που απαιτούνται για την τεχνική εκμάθησης και τον τρόπο που το μοντέλο χρήστη εκπροσωπείται. Οι αλγόριθμοι που απαιτούν ένα ξεκάθαρο εκπαιδευτικό σετ χρησιμοποιούν μάθηση με επίβλεψη ενώ οι υπόλοιποι χρησιμοποιούν μάθηση χωρίς επίβλεψη (Mitchell, 1997). Η παρακολούθηση της συμπεριφοράς του χρήστη παράγει δεδομένα χωρίς ετικέτα, κατάλληλα για ανεξέλεγκτες τεχνικές μάθησης. Αυτός είναι ο πιο δύσκολος τρόπος για να μάθει ο χρήστης αλλά είναι παράλληλα και ο λιγότερο παρεμβατικός. Ζητώντας από το χρήστη για την ανατροφοδότηση, είτε πρόκειται για την κάθε περίπτωση ξεχωριστά ή για ολόκληρο το σετ εκπαίδευσης, παράγει δεδομένα με ετικέτα. Το μειονέκτημα είναι ότι για να παρέχεται η ανατροφοδότηση απαιτείται μία επένδυση της προσπάθειας στον πράκτορα από το χρήστη. Ο προγραμματισμός του χρήστη περιλαμβάνει σημαντικές αλλαγές για τον πράκτορα. Ο προγραμματισμός μπορεί να πραγματοποιηθεί με ποικίλους τρόπους, από σύνθετες γλώσσες μέχρι απλά γραφήματα και απαιτεί σημαντική προσπάθεια από το χρήστη. Η μοντελοποίηση των χρηστών γίνεται με δύο τρόπους, με βάση τη συμπεριφορά και με βάση τη γνώση (Kobsa, 1993). Αυτή που βασίζεται στη γνώση είναι συνήθως το αποτέλεσμα των ερωτηματολογίων και των μελετών που γίνονται από τους χρήστες. Αυτή που βασίζεται στη συμπεριφορά είναι αποτέλεσμα της παρακολούθησης του χρήστη κατά τη διάρκεια της δραστηριότητας. Τα στερεότυπα (Rich, 1979) μπορούν να εφαρμοστούν και στις δύο περιπτώσεις, ταξινομώντας τους χρήστες σε ομάδες με στόχο την εφαρμογή γενικεύσεων στους ανθρώπους σε αυτές τις ομάδες. Ειδικοί

παράγοντες διασύνδεσης εφαρμόζουν πολλά από τα παραπάνω είδη τεχνολογίας, οπότε εμφανίζονται σε διάφορες κατηγορίες. Ένα κοινό παράδειγμα είναι ένας πράκτορας που μαθαίνει για το χρήστη και επίσης υποστηρίζει ένα μοντέλο χρήστη.

3.9 Πράκτορες με Συναισθήματα

Οι ευφυείς πράκτορες έχουν ήδη βρει το δρόμο τους προς το κοινό και κατά τη διάρκεια των τελευταίων δεκαετιών έχουν δείξει ότι είναι αποτελεσματικοί στην επίλυση συγκεκριμένων καθηκόντων. Κάνοντας τη μετάβαση όμως από ευφυείς σε αυτόνομους πράκτορες απαιτείται μία ικανότητα λήψης αποφάσεων που θα τους επιτρέψει να αποφασίσουν τι πρέπει να γίνει με βάση την εσωτερική τους ή και ψυχική τους κατάσταση. Τα συναισθήματα επιτυγχάνουν λειτουργίες με τις οποίες μεταφέρονται σε ανθρώπους, επιτρέποντας έτσι περισσότερο αποτελεσματικές και φυσικές αλληλεπιδράσεις μεταξύ ανθρώπων και πρακτόρων. Ένας πράκτορας με συναισθήματα κατέχει ορθολογική γνώση, έχει ικανότητες αντίδρασης και αλληλεπιδρά με τον έξω κόσμο συμπεριλαμβάνοντας και άλλους πράκτορες. Η τρέχουσα έρευνα στην Τεχνητή Νοημοσύνη έχει δείξει ένα αυξανόμενο ενδιαφέρον για τους πράκτορες με συναισθήματα. Από τους πιστευτούς πράκτορες στη βιομηχανία της ψυχαγωγίας περάσαμε στην αλληλεπίδραση ανθρώπου-υπολογιστή και ανθρώπου-ρομπότ. Οι πράκτορες με συναισθήματα μπορούν να εφαρμοστούν σε ένα ευρύ φάσμα περιοχών είτε για εκπαιδευτικό είτε για ψυχαγωγικό σκοπό. Ωστόσο, είναι ακόμη σχετικά ασαφές αν οι ιδιότητες που αποδίδονται στους πράκτορες από τους χρήστες είναι σωστές και αν τα συναισθήματα που έχουν είναι πραγματικά. Στη συγκεκριμένη ενότητα θα παρουσιάσουμε μία γενική εικόνα από τις έρευνες που έχουν γίνει στον τομέα των πρακτόρων με συναισθήματα και θα επιχειρήσουμε να απομονώσουμε μερικούς από τους λόγους για τους οποίους ένας τέτοιος πράκτορας είναι επιθυμητός καθώς και τις δυσκολίες λήψης ενός ακριβούς συναισθήματος (Griffiths, 1997). Στη συνέχεια απαριθμούνται και κάποια προβλήματα που συνδέονται με τους "δήθεν" συναισθηματικούς πράκτορες, πότε δηλαδή ένας πράκτορας εφαρμόζει πραγματικά ή όχι συναισθήματα. Υπάρχουν αρκετοί λόγοι για να χρησιμοποιήσει κάποιος πράκτορες με συναισθήματα, μερικοί από τους οποίους θα αναφερθούν παρακάτω.



Εικόνα 18 Συναισθήματα στους πράκτορες

3.9.1 Γιατί Πράκτορες με Συναισθήματα

Η έρευνα στην τεχνητή νοημοσύνη επικεντρώθηκε αρχικά στους παραστατικούς μηχανισμούς λήψης αποφάσεων και διαδικασιών, όπως ο προγραμματισμός, η αναζήτηση, η συλλογιστική, για να ελέγξει τους διάφορους τύπους των πρακτόρων. Οι μηχανισμοί αυτοί έχουν χρησιμοποιηθεί σε πολλούς τομείς με μεγάλη επιτυχία. Γιατί λοιπόν να ενδιαφερθούμε για τους συναισθηματικούς πράκτορες τη στιγμή που όπως φαίνεται από την ανθρώπινη εμπειρία τουλάχιστον, τα συναισθήματα δεν ταιριάζουν με τον ορθολογισμό.

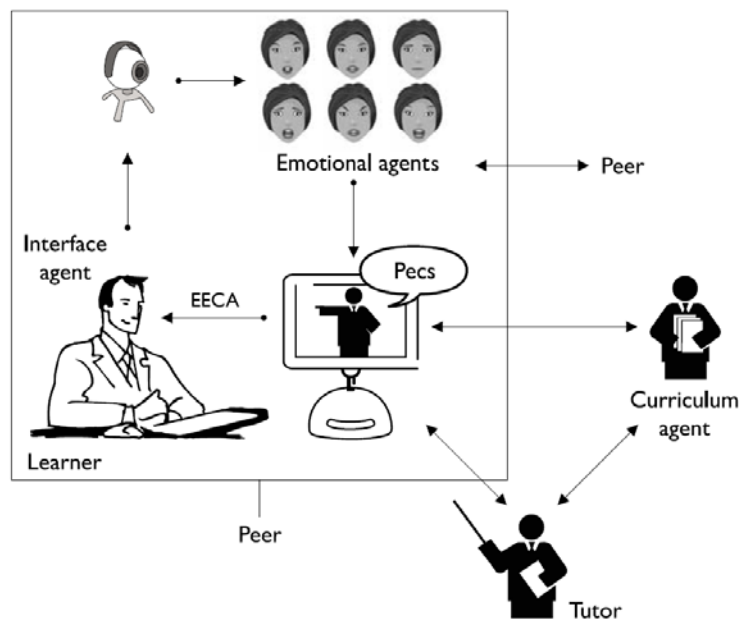
Κάποιος μπορεί να ενδιαφέρεται να μιμηθεί την ανθρώπινη συμπεριφορά με ένα πιο φυσικό τρόπο. Αυτό είναι κατά πρώτο λόγο η προοπτική ενός γνωστικού επιστήμονα αλλά και επιστήμονες της πληροφορικής που ασχολούνται με την κατασκευή παιχνιδιών για τον υπολογιστή, βρίσκουν πολύ ενδιαφέρουσα μία τέτοια εφαρμογή (Tao and Tan, 2005). Το ενδιαφέρον για τους πράκτορες με συναισθήματα έγινε πιο έντονο από τη στιγμή που τα συναισθήματα αποτελούν σημαντικά κομμάτια που μπορούν να χρησιμοποιηθούν για να καθορίσουν μία αποτελεσματική και αποδοτική διαδικασία λήψης αποφάσεων (Sloman, 1990). Ένας ακόμη λόγος είναι ότι σε πολλούς τομείς εφαρμογής, υπάρχουν διάφοροι περιορισμοί που δεν επιτρέπουν την ύπαρξη συμβουλευτικών μεθόδων. Για παράδειγμα, σε ορισμένους τομείς ή περιβάλλοντα, υπάρχουν περιορισμοί στο χρόνο επεξεργασίας ή περιορισμοί σχετικά με τη διαθέσιμη μνήμη και τους υπολογιστικούς πόρους ενός πράκτορα. Σε αυτά τα περιβάλλοντα περιορισμένων πηγών, οι συμβουλευτικές μέθοδοι δεν είναι ο καλύτερος τρόπος για τον έλεγχο των πρακτόρων. Αντίθετα, κάποιες μη αντιπροσωπευτικές

και αντιδραστικές μέθοδοι ελέγχου των πρακτόρων, παρόμοιες με αυτές που χρησιμοποιούνται στους βιολογικούς οργανισμούς, μπορεί να αποδειχθούν περισσότερο κατάλληλες και αποτελεσματικές. Ειδικότερα, οι λεγόμενες συναισθηματικές καταστάσεις, μπορούν να φανούν πολύ χρήσιμες ως αποτελεσματικές σε αδόμητα, ανταγωνιστικά και περιορισμένα σε πόρους περιβάλλοντα. Ένας ακόμη λόγος είναι ότι πρέπει να εφαρμόσουμε τα μοντέλα των συναισθημάτων για να επαληθεύσουμε τις θεωρητικές αναλύσεις και τις προβλέψεις μας, έτσι ώστε να κατανοήσουμε τον πλούτο των ανθρώπινων συναισθημάτων. Είναι ωστόσο πολύ πιθανόν να δημιουργηθούν πράκτορες για τη μέτρηση και αξιολόγηση των ανθρώπινων συναισθηματικών καταστάσεων χωρίς την ανάγκη να έχουν οι ίδιοι πραγματικά συναισθήματα. Ένας ακόμη λόγος μπορεί να είναι ότι ορισμένα συναισθήματα αποτελούν υποπροϊόν της επεξεργασίας κάποιων μηχανισμών. Για παράδειγμα, τυπικά ανθρώπινα συναισθήματα, όπως η χαρά, η λύπη, η ενοχή, φαίνεται να οδηγούν σε διακοπές των συμβουλευτικών διαδικασιών. Μερικά συναισθήματα μπορούν να ερμηνευτούν ως απώλεια του ελέγχου ορισμένων διαδικασιών αντανάκλασης, οι οποίες είναι υπεύθυνες για την ισορροπία των συμβουλευτικών διαδικασιών (Oatley and Jenkins, 2001). Τέτοια συναισθήματα δεν αποτελούν μέρος της αρχιτεκτονικής ενός πράκτορα αλλά προκύπτουν ως αποτέλεσμα των αλληλεπιδράσεων των διαφόρων συνιστωσών του συστήματος ελέγχου του πράκτορα. αν η ανάλυση αυτών των συναισθημάτων είναι σωστή, ένα ρομπότ το οποίο είναι ικανό για διαδικασίες αντανάκλασης παρόμοιες με εκείνες των ανθρώπων, θα είναι επίσης ικανό να έχει συγκινήσεις παρόμοιες με αυτές που προκύπτουν από την αλληλεπίδραση αυτών των διαδικασιών.

Τι είναι όμως στην πραγματικότητα τα συναισθήματα στα υπολογιστικά συστήματα;

- Παραγωγή συναισθηματικής αντίδρασης.
- Χρήση υπολογιστή που "σχετίζεται με", "προκύπτει από" ή "επιρεάζει σκόπιμα τα συναισθήματα".
- Η ικανότητα ενός ηλεκτρονικού υπολογιστικού συστήματος να αναγνωρίζει και να εκφράζει τα συναισθήματα όπως ακριβώς και οι άνθρωποι.
- Η ικανότητα αναγνώρισης και έκφρασης δε σημαίνει ότι έχουν και στην πραγματικότητα συναισθήματα.

Στην παρακάτω εικόνα παρατηρούμε τη διαδικασία συνεργασίας του Πράκτορα Διασύνδεσης με τον Πράκτορα συναισθημάτων σε ένα πολυπρακτορικό σύστημα έτσι ώστε να εμφανιστεί η ανάλογη έκφραση του χρήστη.



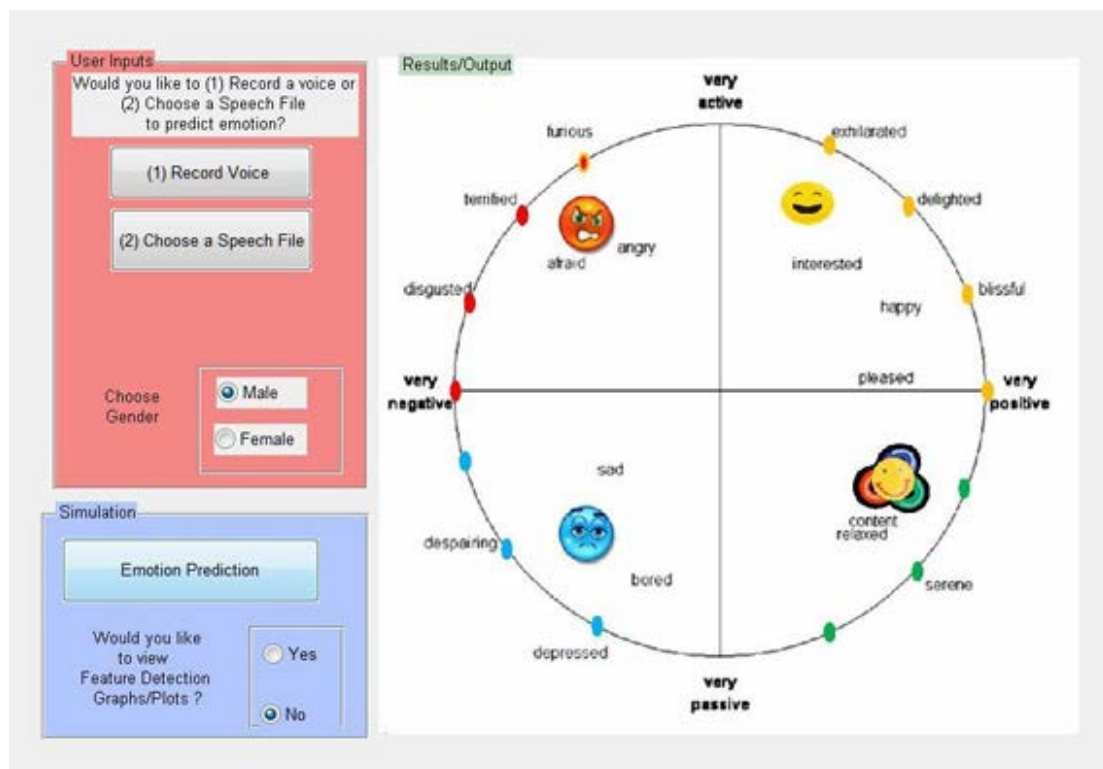
Εικόνα 19 Πράκτορας Διασύνδεσης (Ammar, 2007)

Ωστόσο, οι κύριοι λόγοι ένταξης των πρακτόρων με συναισθήματα στην τεχνητή νοημοσύνη δεν φαίνεται να συμπίπτουν με καμία από τις παραπάνω κατηγορίες. Αντίθετα, φαίνεται να βασίζεται σε μία πιο επιφανειακή ανάλυση του πώς θα μπορούσε να βελτιωθεί η αλληλεπίδραση ανθρώπου-υπολογιστή και ανθρώπου-υπολογιστικού συστήματος. Η ανάγκη για πράκτορες με συναισθήματα συνήθως βασίζεται στον ισχυρισμό ότι τέτοιου είδους πράκτορες επικοινωνούν καλύτερα και με πιο φυσικό τρόπο με τους ανθρώπους και το αντίστροφο. Είναι βέβαιο ότι οι πράκτορες που λαμβάνουν υπόψη τους τον ανθρώπινο παράγοντα στις απαντήσεις τους, εμφανίζονται περισσότερο πειστικοί, ρεαλιστικοί, αξιόπιστοι και είναι πολύ χρήσιμοι στη διδασκαλία που βασίζεται στον υπολογιστή (Reilly and Bates, 2002). Βέβαια αυτό δεν σημαίνει ότι οι πράκτορες αυτοί πρέπει να είναι σε θέση να έχουν οι ίδιοι συναισθήματα, ούτε ότι η εφαρμογή τους παράγει πραγματικά συναισθήματα. Θεωρώντας ως δεδομένο ότι οι πράκτορες αυτοί έχουν ανθρώπινα συναισθήματα αλλά είναι τελείως διαφορετικοί από τους ανθρώπους, προκύπτουν κάποιες δυσκολίες που συνδέονται με τη φύση των συναισθημάτων που θα αναλυθούν παρακάτω.

3.9.2 Ο ρόλος των συναισθημάτων στους πράκτορες

Με βάση τους λειτουργικούς ρόλους των συναισθημάτων που προτείνονται από τους ερευνητές των φυσικών συστημάτων, θα ήταν χρήσιμο να αναρωτηθούμε αν τα συναισθήματα μπορούν να φανούν το ίδιο λειτουργικά και στα τεχνητά συστήματα. Συγκεκριμένα, μπορούμε να καθορίσουμε 12 δυναμικούς ρόλους των συναισθημάτων στα τεχνητά συστήματα:

-
- Επιλογή Δράσης-->Ποιο είναι το επόμενο βήμα με βάση την τρέχουσα συναισθηματική κατάσταση.
 - Προσαρμογή-->Βραχυπρόθεσμες ή μακροπρόθεσμες αλλαγές στη συμπεριφορά λόγω των συναισθηματικών καταστάσεων.
 - Κοινωνική ρύθμιση-->Επικοινωνία ή ανταλλαγή πληροφοριών μέσω συναισθηματικών εκφράσεων.
 - Ολοκλήρωση των Αισθήσεων-->Συναισθηματικό φιλτράρισμα των δεδομένων.
 - Μηχανισμοί Συναγερμού-->Γρήγορα αντανακλαστικά όπως αντιδράσεις σε κρίσιμες καταστάσεις που διακόπτουν άλλες διεργασίες.
 - Κίνητρο-->Δημιουργία κινήτρων ως μέρος ενός συναισθηματικού μηχανισμού αντιμετώπισης.
 - Διαχείριση Στόχων-->Δημιουργία νέων στόχων ή επαναπροσδιορισμός των υπαρχόντων.
 - Μάθηση-->Συναισθηματικές αξιολογήσεις στην ενισχυτική μάθηση.
 - Συγκέντρωση-->Η επιλογή των προς επεξεργασία δεδομένων βασίζεται σε συναισθηματική αξιολόγηση.
 - Έλεγχος της μνήμης-->Συναισθηματική προκατάληψη σχετικά με την πρόσβαση μνήμης και ανάκτησης καθώς και ο ρυθμός αποσύνθεσης των στοιχείων μνήμης.
 - Στρατηγική Επεξεργασίας-->Επιλογή των διαφορετικών στρατηγικών αναζήτησης που βασίζονται στη συνολική συναισθηματική κατάσταση.
 - Αυτό-μοντέλο-->Συναισθήματα όπως παραστάσεις του τι είναι μία κατάσταση για έναν πράκτορα.



Εικόνα 20 Ρόλος των συναισθημάτων στους Πράκτορες

3.9.3 Ιστορία των Πρακτόρων με Συναισθήματα

Τις τελευταίες δύο δεκαετίες, η ενσωμάτωση των συναισθημάτων στη συμπεριφορά των πρακτόρων έχει προσελκύσει ιδιαίτερη προσοχή. Αρκετοί ερευνητές ανέπτυξαν μοντέλα για να αποδείξουν την επικοινωνία μεταξύ χρηστών και υπολογιστών. Εικονικοί χαρακτήρες με συναισθηματική συμπεριφορά προς τους χρήστες αναπτύχθηκαν, όπως το πρότυπο Oz του 1992 και τη δουλειά του Maldonado και των συμφοιτητών του που δημιούργησαν έναν εικονικό πράκτορα-συμμαθητή. Σε ένα πιο πρόσφατο πείραμα που έγινε από τον Yen Yang και την ομάδα του, εξετάστηκε ένα ολοκληρωμένο υπολογιστικό μοντέλο από συναισθήματα με σκοπό να ρυθμίσει τη συμπεριφορά του πράκτορα. Επίσης, το μοντέλο των Conati και Maclaren που αναπτύχθηκε την ίδια χρονιά, αντιπροσωπεύει τις πιθανολογικές σχέσεις μεταξύ αιτιών, επιδράσεων και συναισθηματικών καταστάσεων των χρηστών. Οι ερευνητές διεξάγουν πειράματα με σκοπό να απαντήσουν σε κρίσιμες επιστημονικές ερωτήσεις που αφορούν τη συμπεριφορά του πράκτορα, τους τύπους των συναισθημάτων που εκπροσωπούνται, την εμπιστευτικότητα και την αλληλεπίδραση μεταξύ συναισθηματικών αντιπροσώπων. Ωστόσο, η επιρροή που έχει η προσωπικότητα του πράκτορα στους χρήστες δεν έχει εξεταστεί ακόμη αρκετά. Μερικές μόνο λεπτομερείς αξιολογήσεις πάνω στους πράκτορες με συναισθήματα έχουν δημοσιευθεί. Όλες προσπαθούν να φτάσουν σε

συμπεράσματα μέσω των θετικών ή αρνητικών επιδράσεων που έχουν τα συστήματα που ενσωματώνουν ανθρωπομορφικά χαρακτηριστικά στους χρήστες. Μία από τις τελευταίες προσπάθειες δημοσιεύθηκε το 2009 και περιλαμβάνει μία αντιπροσωπευτική και εκτενή ανάλυση στα προηγούμενα συστήματα από πολλούς διαφορετικούς τομείς όπως υγεία, μάθηση, εκπαίδευση, εμπόριο, στρατιωτικές υπηρεσίες). Ο Beale και ο Creed, αφού εξέτασαν πάνω από 100 άρθρα, αξιολόγησαν εκτενώς μερικά πειράματα που συγκεντρώνουν τους ίδιους παράγοντες, με σκοπό να τους ταξινομήσουν και να συγκρίνουν τις τιμές τους. Ο Yee, ο Bailenson και ο Rickertsen το 2007 διεξήγαν μία μετά-ανάλυση στην οποία ισχυρίστηκαν ότι η σχέση μεταξύ πρακτόρων και χρηστών αποδεικνύει την επικοινωνία και προσφέρει ένα πιο φυσικό τρόπο αλληλεπίδρασης. Ο Dehn και ο Van Mulken το 2000 αξιολόγησαν 25 αναφορές και άρθρα σχετικά με την αλληλεπίδραση της συναισθηματικής συμπεριφοράς των πρακτόρων με τους χρήστες.

3.9.4 Προβλήματα των Πρακτόρων με Συναισθήματα

Ένα βασικό πρόβλημα είναι ότι ενώ μπορούμε να πούμε τι είναι περισσότερο ή λιγότερο τυπικό από ένα συναίσθημα δεν μπορούμε να παρέχουμε μία σειρά από αναγκαίες και επαρκείς προϋποθέσεις οι οποίες θα καθορίζουν την έννοια του συναίσθηματος. Η δυσκολία στον ορισμό προκύπτει από τους πολυάριθμους και ποικίλους χαρακτηρισμούς. Είναι δύσκολο να πει κάποιος τι ακριβώς είναι το συναίσθημα. Με τους διάφορους ορισμούς τονίζονται κάποιες από τις πτυχές των συναισθημάτων όπως οι εγκεφαλικές διεργασίες, οι περιφερειακές εσωτερικές διαδικασίες, τα πρότυπα συμπεριφοράς, οι λειτουργικοί ρόλοι, οι εσωστρεφείς ιδιότητες κ.ά. Οι ορισμοί των συναισθημάτων διαφέρουν επίσης και στο πεδίο της εφαρμογής. Κάποιοι για παράδειγμα θεωρούν το κίνητρο ή την επιθυμία για κάτι ως συναίσθημα ενώ άλλοι όχι. Επίσης, κάποιοι θεωρούν την έκπληξη ως συναίσθημα ενώ άλλοι τη θεωρούν ως μία γνωστική κατάσταση στην οποία έχει βρεθεί μία πεποίθηση, η οποία με τη σειρά της μπορεί ή δεν μπορεί να παράγει μία συναισθηματική αντίδραση. Ένα ακόμη πρόβλημα που προκύπτει είναι ότι ενώ οι κοινωνικοί επιστήμονες έχουν την τάση να οριστεί το συναίσθημα έτσι ώστε να επικεντρωθούν σε κοινωνικά φαινόμενα όπως η αμηχανία, η ενοχή ή η υπερηφάνεια, οι επιστήμονες του εγκεφάλου προτιμούν να το καθορίσουν με τέτοιο τρόπο έτσι ώστε να αναφέρεται στις λειτουργίες του εγκεφάλου και στις εκτεταμένες συμπεριφορές των ανθρώπων. Όλη αυτή η ποικιλομορφία παρέχει ένα ισχυρό αποδεικτικό στοιχείο ότι τα συναισθήματα είναι ένα σύμπλεγμα εννοιών και ότι είναι σχεδόν απίθανο να μπορέσουμε να βρούμε έναν ορισμό που να ισχύει εξίσου για όλες τις διαφορετικές πτυχές των συναισθημάτων. Ένας ακόμη λόγος που δυσκολεύει την ύπαρξη ενός ακριβούς και λειτουργικού ορισμού των συναισθημάτων είναι ότι οι περισσότερες έννοιες έχουν δημιουργηθεί με βάση την αρχιτεκτονική του (Sloman, 2000). Έννοιες όπως "ξεχνάω",

"ονειρεύομαι", "φαντάζομαι", "αισθάνομαι ένοχος", εξαρτώνται έμμεσα από μία συγκεκριμένη αρχιτεκτονική.

3.10 Περιγραφή προβλήματος ωρολογίου προγράμματος

Το πρόβλημα της δημιουργίας ωρολογίων προγραμμάτων απασχολεί εδώ και πάρα πολλά χρόνια τους επιστήμονες σε διάφορους τομείς όπως νοσοκομεία, τρένα, σχολεία, πανεπιστήμια και πολλά άλλα. Το πρόβλημα δημιουργίας ωρολογίου προγράμματος μαθημάτων για Πανεπιστημιακά Ιδρύματα όμως είναι αυτό το οποίο έχει μελετηθεί ευρέως και από θεωρητική και από πρακτική σκοπιά. Είναι μία από τις πιο σημαντικές και χρονοβόρες διαδικασίες οι οποίες απασχολούν ανά τακτά χρονικά διαστήματα (Τρίμηνο, Εξάμηνο, Έτος) όλα τα Πανεπιστημιακά Ιδρύματα στον κόσμο. Η ποιότητα του ωρολογίου προγράμματος έχει μεγάλη επίπτωση σε ένα ευρύ φάσμα δραστηριοτήτων όλων των εμπλεκόμενων πλευρών, συμπεριλαμβανομένου και των καθηγητών, των μαθητών και των διοικητικών υπαλλήλων (Aarts, 2005).

Το ωρολόγιο πρόγραμμα μαθημάτων είναι μία ιδιαίτερη περίπτωση του γενικότερου προβλήματος δημιουργίας προγραμμάτων. Στην απλούστερη μορφή του είναι ένα πρόβλημα προγραμματισμού μίας σειράς εκδηλώσεων (διαλέξεις, σεμινάρια, εργαστήρια) σε ένα σύνολο αιθουσών διδασκαλίας και σε σύνολο χρονοθυρίδων, τα οποία διδάσκονται από ένα σύνολο καθηγητών και τα παρακολουθεί ένα σύνολο φοιτητών. Οι εκδηλώσεις θα πρέπει να είναι οργανωμένες με τέτοιο τρόπο ώστε κανένας φοιτητής ή καθηγητής να μην είναι σε περισσότερες από μία αίθουσες ταυτόχρονα και επίσης να υπάρχει αρκετός χώρος σε κάθε αίθουσα για τον αριθμό των φοιτητών που παρακολουθούν. Επίσης θα πρέπει να μην συμβαίνουν δυο γεγονότα στην ίδια αίθουσα ταυτόχρονα (Malim, 2005).

Η δομή των προσφερόμενων μαθημάτων ενδέχεται να είναι αυτή που εμφανίζει τα περισσότερα προβλήματα. Παρόλο που το πρόβλημα που αντιμετωπίζουμε ονομάζεται «Δημιουργία ωρολογίου προγράμματος μαθημάτων», στην πραγματικότητα οι επιμέρους συναντήσεις (μαθήματα) είναι αυτές που δημιουργούν το μάθημα. Ένα μάθημα είναι ένα σύνολο από επιμέρους συναντήσεις μίας ομάδας φοιτητών, οι οποίοι έχουν εγγραφεί στο μάθημα με έναν καθηγητή, και οι οποίες πραγματοποιούνται σε μία τάξη σε προκαθορισμένα χρονικά διαστήματα μέσα σε μία εβδομάδα. Τα μαθήματα συνήθως αποτελούνται από έναν αριθμό εβδομαδιαίων συναντήσεων, είτε διαλέξεων, είτε εργαστηρίων, είτε σεμιναρίων. Οι μαθητές εγγράφονται σε μία σειρά μαθημάτων σύμφωνα με τις ανάγκες του ακαδημαϊκού τους προγράμματος.

Η δημιουργία του προγράμματος γίνεται δημιουργώντας μία λογική δομή ανάμεσα στα δεδομένα τα οποία αντιπροσωπεύουν όλα τα επιμέρους κομμάτια του μαθήματος (καθηγητές,

μαθητές, τάξεις, χρονικά διαστήματα) καθώς και τις σχέσεις μεταξύ αυτών και του υπόλοιπου συνόλου των μαθημάτων, λαμβάνοντας υπόψη τους περιορισμούς που τίθενται. Πολλά μαθήματα αποτελούνται από ορισμένα υπομήματα, όπως σεμινάρια ή εργαστήρια, τα οποία συνδέονται με το μάθημα. Το ίδιο μάθημα μπορεί να αποτελείται από διαφορετικά υπομήματα ανάλογα με τον καθηγητή. Για παράδειγμα κάποιος καθηγητής επιθυμεί κάποιες ώρες στο εργαστήριο, ενώ κάποιος άλλος όχι.

Υπάρχουν δυο είδη ισχυρών περιορισμών που πρέπει να υλοποιηθούν. Ισχυροί περιορισμοί πόρων, ένας καθηγητής δεν μπορεί να είναι σε δυο τάξεις ταυτόχρονα, δεν μπορούν να γίνουν στην ίδια τάξη δυο μαθήματα την ίδια χρονική στιγμή και ισχυροί περιορισμοί διανομής οι οποίοι εκφράζουν απαιτούμενες ή απαγορευμένες σχέσεις μεταξύ πολλών τάξεων, δυο διαλέξεις του ίδιου μαθήματος δεν μπορεί να γίνουν την ίδια χρονική στιγμή, τα μαθήματα σε μία τάξη πρέπει να γίνονται διαδοχικά. Υπάρχουν τρεις τύποι χαλαρών περιορισμών, ο πρώτος αφορά χρόνους και αίθουσες, ο δεύτερος απαιτήσεις των φοιτητών και των καθηγητών και ο τρίτος αφορά τους περιορισμούς διανομής, οι οποίοι εκφράζουν προτιμήσεις σε τάξεις και μαθήματα (McCullum, 2004).

Προκειμένου να βρεθεί η καλύτερη δυνατή λύση, θα πρέπει να ληφθεί υπόψη το σύνολο περιορισμών που τίθενται για το πρόβλημα και οι οποίοι θα πρέπει να τηρηθούν. Οι περιορισμοί αυτοί ταξινομούνται σε δυο κατηγορίες. Η πρώτη είναι οι ισχυροί περιορισμοί (hard constraints) και η δεύτερη οι χαλαροί περιορισμοί (soft constraints).

Ισχυροί Περιορισμοί (Hard Constraints). Οι ισχυροί περιορισμοί δεν επιτρέπεται να παραβιαστούν κάτω από οποιοσδήποτε συνθήκες (κυρίως λόγω των φυσικών περιορισμών). Θα πρέπει σε κάθε περίπτωση να ικανοποιηθεί το σύνολο τους, γιατί η παραβίαση τους θα οδηγήσει σε μη αποδεκτή λύση (Burke and Marecek, 2010). Οι κυριότεροι και πιο συνηθισμένοι ισχυροί περιορισμοί για τη δημιουργία του ωρολόγιου προγράμματος είναι:

- Ένας καθηγητής ή ένας μαθητής μπορεί να βρίσκεται μόνο σε ένα μάθημα την ίδια χρονική στιγμή. Οι διαλέξεις οι οποίες γίνονται από έναν καθηγητή δεν μπορούν να πραγματοποιηθούν ταυτόχρονα
- Ένας καθηγητής δεν μπορεί να καλύψει ένα γεγονός σε χρονοθυρίδες (timeslots) κατά τις οποίες αυτός δεν είναι διαθέσιμος.
- Ο αριθμός των χρονοθυρίδων που ανατίθενται σε κάθε μάθημα για μία εβδομάδα πρέπει να είναι ίσος με τον εβδομαδιαίο αριθμό ωρών του μαθήματος.
- Δυο γεγονότα του ίδιου μαθήματος δεν μπορούν να γίνουν ταυτόχρονα.
- Ο αριθμός των γεγονότων που συμβαίνουν ταυτόχρονα δεν θα πρέπει να υπερβαίνει τον συνολικό αριθμό των αιθουσών.

-
- Μία αίθουσα μπορεί να φιλοξενήσει μόνο ένα γεγονός σε μία χρονοθυρίδα. Δυο διαλέξεις δεν μπορούν να πραγματοποιηθούν στην ίδια τάξη και την ίδια χρονική στιγμή (Malim, 2005).
 - Μία διάλεξη ομάδας δεν μπορεί να γίνεται την ίδια περίοδο με μία άλλη η οποία δεν είναι διάλεξη ομάδας και ανήκουν και οι δυο στο ίδιο επίπεδο σπουδών.
 - Ο αριθμός των φοιτητών που έχουν εγγραφεί σε ένα μάθημα θα πρέπει να είναι μικρότερος ή ίσος με τη χωρητικότητα της αίθουσας στην οποία πραγματοποιείται (Burke and Marecek, 2010).

Το κάθε Πανεπιστήμιο μπορεί να θέσει τους δικούς του ιδιαίτερους ισχυρούς περιορισμούς οι οποίοι θα βασίζονται στις δικές του ανάγκες και απαιτήσεις. Κάθε ωρολόγιο πρόγραμμα που αποτυγχάνει να ικανοποιήσει αυτούς τους περιορισμούς θεωρείται μη αποδεκτό (Malim, 2005).

Χαλαροί Περιορισμοί (Soft Constrains). Οι χαλαροί περιορισμοί είναι ιδιαίτερες επιθυμίες και προτιμήσεις των εμπλεκόμενων μερών, οι οποίες όμως δεν είναι τόσο κρίσιμες για τη δημιουργία του ωρολόγιου προγράμματος. Στην πράξη είναι σχεδόν αδύνατο να βρεθούν εφικτές λύσεις, οι οποίες θα ικανοποιούν όλους τους χαλαρούς περιορισμούς. Αυτοί ποικίλουν και συχνά έρχονται σε αντίθεση μεταξύ τους και είναι περισσότεροι σε αριθμό και σε γενικές γραμμές πιο πολύπλοκοι από τους ισχυρούς περιορισμούς. Κάποιοι συνηθισμένοι χαλαροί περιορισμοί είναι:

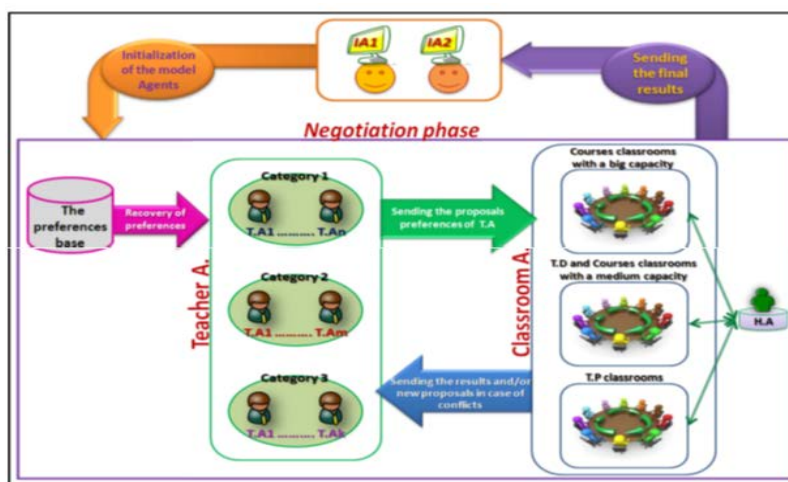
- Ο κάθε καθηγητής να αναλαμβάνει τα μαθήματα που προτιμά.
- Μαθήματα της ίδιας ομάδας φοιτητών θα πρέπει να ανατίθενται σε διαδοχικές χρονοθυρίδες.
- Σε μαθήματα, τα οποία περιλαμβάνουν και θεωρητικό και εργαστηριακό μέρος, το θεωρητικό μέρος θα πρέπει να προηγείται στην εβδομάδα από το εργαστηριακό.
- Ένα μάθημα θα πρέπει ή δεν θα πρέπει να πραγματοποιείται σε κάποια συγκεκριμένη χρονοθυρίδα.
- Όλα τα μαθήματα θα πρέπει να ανατίθενται σε αίθουσες με τέτοιο τρόπο, ώστε να μεγιστοποιηθεί η χρησιμοποίηση της αίθουσας (Malim, 2005).
- Κάποιος καθηγητής προτιμά πρωινά από βραδινά μαθήματα.
- Ο καθηγητής επιθυμεί τα μαθήματα του να είναι συνεχόμενα και να μην έχει κενά διαστήματα ανάμεσα σε αυτά.

Πρόκειται στην ουσία για τους περιορισμούς, οι οποίοι μας επιτρέπουν να συγκρίνουμε το πόσο καλή είναι μία δοσμένη λύση σε σχέση με άλλες παρόμοιες λύσεις. Αυτό γίνεται ελέγχοντας πόσοι χαλαροί περιορισμοί έχουν ικανοποιηθεί με την κάθε λύση (Burke and Marecek, 2010).

Συνήθως στη δημιουργία ενός ωρολόγιου προγράμματος δεν λαμβάνονται υπόψη οι ατομικές προτιμήσεις γιατί είναι δύσκολο να συλληφθούν και να μοντελοποιηθούν. Απλές προτιμήσεις του στυλ "Προτιμώ πρωινά μαθήματα και όχι βραδινά" είναι εύκολο να μοντελοποιηθούν. Αντιθέτως προτάσεις του στυλ "Προτιμώ βραδινά μαθήματα παρά τρία πρωινά στη σειρά" παρουσιάζουν δυσκολίες στη μοντελοποίηση (Oprea, 2007).

Ο μεγάλος αριθμός και η ποικιλία των περιορισμών που προκύπτουν τόσο στη δομή του μαθήματος, όσο και διάφορες ειδικές απαιτήσεις, αυξάνουν τη δυσκολία δημιουργίας ενός προγράμματος το οποίο θα ικανοποιεί όλα τα εμπλεκόμενα μέρη. Ο κάθε δημιουργός του προγράμματος μπορεί να ορίσει τους δικούς του ισχυρούς και χαλαρούς περιορισμούς (McCollum, 2004).

Το πρόβλημα δημιουργίας ωρολόγιου προγράμματος μαθημάτων μπορεί να θεωρηθεί ότι αποτελείται από τρία υποπροβλήματα. Την ανάθεση μαθημάτων σε καθηγητές, την ανάθεση γεγονότων σε χρονοθυρίδες και την ανάθεση αιθουσών σε γεγονότα. Οι καθηγητές αναλαμβάνουν μία σειρά εκδηλώσεων για κάθε μάθημα. Ένα μάθημα μπορεί να το διδάσκουν περισσότεροι από ένας καθηγητές (McCollum, 2006). Στη συνέχεια γίνεται ανάθεση προκαθορισμένων χρονοθυρίδων σε εβδομαδιαία βάση για το σύνολο των εκδηλώσεων του μαθήματος (διαλέξεις και εργαστήρια). Έπεται η εκχώρηση αιθουσών σε όλα τα γεγονότα του μαθήματος. Και τέλος οι μαθητές επιλέγουν τα μαθήματα, τα οποία θα παρακολουθήσουν. Ως εκ τούτου, το πρόβλημα δημιουργίας ωρολόγιου προγράμματος μαθημάτων είναι ένα πρόβλημα εκχώρησης τιμών σε μία προκαθορισμένη τετράδα (a,b,c,d), όπου a ανήκει στα γεγονότα (Events), b ανήκει στις χρονοθυρίδες (timeslots), c ανήκει στις αίθουσες (Rooms) και d ανήκει στους καθηγητές (Professors). Ένα γεγονός a ξεκινά στη χρονοθυρίδα b και διδάσκεται στην αίθουσα c από τον καθηγητή d (Burke and Marecek, 2010). Το πρόβλημα έγκειται στην αντιστοίχιση χρόνου και πηγών για την δημιουργία συναντήσεων και την ταυτόχρονη ικανοποίηση όσο το δυνατόν περισσότερων περιορισμών (Oprea, 2007).



Εικόνα 21 Πολυπρακτορικό μοντέλο για ωρολόγιο πρόγραμμα Πανεπιστημίου

4. Συναισθηματικός Πράκτορας

Διασύνδεσης

4.1 Ανάλυση Απαιτήσεων

Στα πλαίσια αυτής της εργασίας αποφασίστηκε να αναπτυχθεί ένας συναισθηματικός πράκτορας διασύνδεσης για τη Μελέτη Περίπτωσης του ωρολογίου προγράμματος μαθημάτων για το προπτυχιακό τμήμα σπουδών του Τμήματος Μηχανικών Πληροφορικής του Αλεξάνδρειου Τεχνολογικού Εκπαιδευτικού Ιδρύματος Θεσσαλονίκης. Στόχος της συγκεκριμένης διπλωματικής εργασίας είναι να "παντρέψει" τους πράκτορες διασύνδεσης με τους πράκτορες συναισθημάτων και να κάνει πιο κατανοητούς τους λόγους για τους οποίους χρησιμοποιούμε τα συναισθήματα στα πολυπρακτορικά συστήματα όπως είναι και το ωρολόγιο πρόγραμμα. Το πρόβλημα της δημιουργίας ωρολογίου προγράμματος για Ανώτατα Εκπαιδευτικά Ιδρύματα είναι ένα σύνθετο πρόβλημα, στο οποίο θα πρέπει να γίνει μία σωστή κατανομή των διαθέσιμων πόρων, όπως επίσης να υπάρχει και μία αποτελεσματική χρήση των περιορισμών που απαιτούνται για τη δημιουργία του προγράμματος.

Κάθε εξάμηνο ή κάθε χρόνο σε ένα Ανώτατο Πανεπιστημιακό Ίδρυμα, κάθε επιμέρους τμήμα του, θα πρέπει να σχεδιάσει ένα νέο χρονοδιάγραμμα για τα μαθήματα του. Η επίλυση αυτού του προβλήματος συνίσταται στην τοποθέτηση αυτών των μαθημάτων (modules), τα οποία θα πρέπει να αντιστοιχηθούν με τους διαθέσιμους πόρους του Ιδρύματος. Τέτοιοι πόροι είναι οι καθηγητές, οι αίθουσες διδασκαλίας και οι ημέρες και ώρες στις οποίες θα πραγματοποιηθούν τα μαθήματα. Αυτή συνήθως είναι μία χρονοβόρα και επίπονη διαδικασία, η οποία δημιουργεί διάφορες εντάσεις εντός του Ιδρύματος. Αποφασίσαμε να κινηθούμε στα πρότυπα της GEMA(Generic Emotional Agent) αρχιτεκτονικής με κάποιες τροποποιήσεις γιατί μας δείχνει πώς εφαρμόζονται τα συναισθήματα στη διαδικασία της μάθησης αλλά και της διδασκαλίας μέσω των μαθητών αλλά και των καθηγητών. Στη παρούσα διπλωματική εργασία, λαμβάνουμε υπόψη το ρόλο των συναισθημάτων μόνο σε σχέση με τη δημιουργία του ωρολογίου προγράμματος. Πολλές άλλες μεταβλητές αξιολόγησης, όπως ο έλεγχος και η κυριαρχία πρέπει να εξεταστούν σε ένα ολοκληρωμένο σύστημα μάθησης. Σε αυτό το σύστημα, υπάρχει ένας πράκτορας συναισθημάτων που χρησιμοποιεί την αρχιτεκτονική μας για να προσομοιώσει τα συναισθήματα του πράκτορα

που δημιουργεί το πρόγραμμα. Η επιλογή των συναισθημάτων έγινε με βάση το μοντέλο OCC και ο λόγος είναι ότι χρησιμοποιεί ολοκληρωμένες τοπικές αλλά και καθολικές μεταβλητές για να υπολογίσει την ένταση των συναισθημάτων καθώς και μεθόδους για την εκτίμηση των γεγονότων και των δράσεων. Ως εξόδους, ο εκτιμητής έχει βαθμούς ικανοποίησης, συνειδητοποίησης και επαίνου. Η ενημέρωση των συναισθηματικών καταστάσεων καθορίζει την ένταση των 16 συναισθημάτων που χρησιμοποιεί το μοντέλο για τον υπολογισμό της οποίας εφαρμόζει διάφορους κανόνες. Εμείς στη συγκεκριμένη διπλωματική εργασία αποφασίσαμε να χρησιμοποιήσουμε 8 από τα 16 συναισθήματα του μοντέλου τα οποία αφενός:

- (α) Θεωρούνται τα πιο διαδεδομένα σύμφωνα με τις αρχιτεκτονικές και τα μοντέλα που έχουν αναπτυχθεί, μελετήθηκαν και παρουσιάστηκαν στο προηγούμενο κεφάλαιο και αφετέρου
- (β) Κατά τη διαδικασία ανάλυσης των απαιτήσεων του συγκεκριμένου πολυπρακτορικού συστήματος για τη δημιουργία ωρολογίου προγράμματος μαθημάτων κρίθηκαν συμβατά και απαραίτητα.

Σε αυτό το σημείο απλά θα τα αναφέρουμε και παρακάτω θα γίνει μία περαιτέρω ανάλυση μέσω του μοντέλου έκφρασης που χρησιμοποιεί ο πράκτοράς μας. Τα οκτώ λοιπόν συναισθήματα που χρησιμοποιήθηκαν είναι:

- Ηρεμία
- Χαρά
- Ευτυχία
- Ανακούφιση
- Προβληματισμός
- Θυμός
- Εξόργιση
- Αγανάκτηση

Για τον υπολογισμό της έντασης του συναισθήματος χρησιμοποιούνται μία καθολική και τέσσερις τοπικές μεταβλητές. Οι τιμές αυτών των μεταβλητών προέρχονται από διάφορα στοιχεία του μοντέλου. Οι τοπικές μεταβλητές είναι η ικανοποίηση, η συνειδητοποίηση, ο έπαινος και η πιθανότητα. Η καθολική μεταβλητή είναι η αναμενόμενη έκβαση. Οι τομείς των ενεργειών και των δράσεων λειτουργούν με δύο στοιχεία, τον πίνακα προτύπων και την ιστορία των ενεργειών. Τα πρότυπα χρησιμοποιούνται για να υπολογίσουν την αναμενόμενη έκβαση. Η ιστορία των ενεργειών και δράσεων αποθηκεύει το πόσες φορές εμφανίζεται μία ενέργεια έτσι ώστε να υπολογίσει την πιθανότητα εμφάνισης ενός γεγονότος. Η πιθανότητα

αυτή εμφανίζεται με συναίσθημα ελπίδας για τα επιθυμητά γεγονότα και με συναίσθημα φόβου για τα ανεπιθύμητα γεγονότα.

Στη συνολική λοιπόν φάση της δημιουργίας του ωρολογίου προγράμματος εισάγονται και τα συναισθήματα μέσω του πράκτορά μας. Ο πράκτορας συναισθημάτων στη συγκεκριμένη περίπτωση, ταυτίζεται με τον πράκτορα που δημιουργεί το πρόγραμμα (Scheduler Agent) και εκφράζει τα αντίστοιχα συναισθήματα προς τον πράκτορα-καθηγητή (Teacher Agent). Στη διαδικασία αυτή προσπαθεί η έκφραση αυτών των συναισθημάτων προς τον τελικό χρήστη (καθηγητή) να βοηθάει στην καλύτερη επικοινωνία και διαδικασία διαπραγμάτευσης.

4.2 Εισαγωγή Συναισθηματικού Πράκτορα Διασύνδεσης σε Πολυπρακτορικό Σύστημα ωρών και μαθημάτων

Στο τελευταίο κομμάτι της δημιουργίας του ωρολογίου προγράμματος εμφανίζεται και ο πράκτορας με συναισθήματα. Ουσιαστικά ο πράκτοράς μας παίρνει κάποια δεδομένα σαν είσοδο και εμφανίζει σαν έξοδο την αντίστοιχη εικόνα. Οι εικόνες, που είναι διάφορες εκφράσεις του προσώπου ανάλογα με τα συναισθήματα που θέλει να εμφανίσει, είναι αποθηκευμένες σε μία βάση δεδομένων και κάθε φορά, ανάλογα με την περίπτωση, παρουσιάζει και μία διαφορετική. Το βασικό στοιχείο που πρέπει να αναφέρουμε σε αυτή τη φάση είναι ότι η εμφάνιση της αντίστοιχης εικόνας καθορίζεται από τους περιορισμούς που θα επιστραφούν από τους πράκτορες καθηγητές. Αν, παραδείγματος χάριν, οι περιορισμοί που θα επιστραφούν είναι ισχυροί, το οποίο σημαίνει ότι δε μπορούν να παραβιαστούν και πρέπει να ικανοποιηθούν έτσι ώστε να είναι αποδεκτή η τελική λύση, θα επικρατήσει το συναίσθημα του προβληματισμού, της στεναχώριας και της απογοήτευσης. Ανάλογα βέβαια με το πόσοι αλλά και ποιοι περιορισμοί στάλθηκαν, επιστρέφεται και η ανάλογη εικόνα. Τα κριτήρια που επιλέξαμε να χρησιμοποιήσουμε για να ρυθμίζουμε το ποια εικόνα θα εμφανιστεί είναι πέντε:

- Η βαθμίδα του καθηγητή.
- Εάν υπάρχουν ή όχι περιορισμοί (conflicts).
- Πόσες προτιμήσεις δεν έγινε δυνατόν να ικανοποιηθούν και επεστράφησαν.
- Τι είδους είναι οι προτιμήσεις που επεστράφησαν.
- Η φάση της διαπραγμάτευσης με βάση το πρωτόκολλο αλληλεπίδρασης.

Οι φάσεις της διαπραγμάτευσης που θα μας απασχολήσουν είναι η πρώτη (1η), η δεύτερη (2η) και η Τρίτη (3η). Αντίστοιχα, οι βαθμίδες των καθηγητών είναι ο επίκουρος καθηγητής/καθηγητής εφαρμογών και ο τακτικός καθηγητής/αναπληρωτής καθηγητής. Εννοείται ότι κάθε ένα από τα παραπάνω κριτήρια έχουν ξεχωριστά βάρη και καθορίζουν

ποια εικόνα θα εμφανιστεί. Επομένως, αν βρισκόμαστε στην πρώτη φάση της διαπραγμάτευσης, ο περιορισμός είναι λιγότερο ισχυρός, συνεπώς ο πράκτορας δημιουργίας του προγράμματος θα είναι πιο επιεικής, πιο χαλαρός και πιο ευδιάθετος γιατί μπορεί να ικανοποιήσει τα περισσότερα από αυτά που ζητούν οι καθηγητές. Βασικό επίσης ρόλο στα συναισθήματα που θα εμφανιστούν παίζει και η βαθμίδα του καθηγητή. Είναι διαφορετικό να προσπαθεί να εξυπηρετήσει τον τακτικό καθηγητή και διαφορετικό τον καθηγητή εφαρμογών. Προτεραιότητα θα δοθεί σαφώς στον πρώτο γιατί, όπως προκύπτει από την εμπειρία κατάστρωσης ωρολογίων προγραμμάτων στην πράξη, έχει μεγαλύτερο "βάρος". Πολύ σημαντικό επίσης είναι και ο αριθμός των conflicts που επέστρεψε ο ProgramScheduler στους καθηγητές καθώς και το τι είδους ήταν αυτές. Είναι λογικό ότι αν επιστραφούν στην πρώτη φάση λίγοι σχετικά περιορισμοί από τον ProgramScheduler στον αναπληρωτή καθηγητή, θα επικρατήσει το συναίσθημα της ικανοποίησης και της χαράς γιατί ο πράκτορας κατάφερε να τους εξυπηρετήσει σχεδόν όλους. Αντίθετα, αν επιστραφούν στη δεύτερη φάση τέσσερις περιορισμοί από τον καθηγητή εφαρμογών, θα επικρατήσει το συναίσθημα της λύπης γιατί δεν μπόρεσε να τους εξυπηρετήσει και του εκνευρισμού γιατί δεν προσπαθεί να τροποποιήσει κάπως το πρόγραμμά του και επιμένει στις αρχικές του προτιμήσεις. Τέλος, είναι σαφώς πιο δύσκολο να ικανοποιηθεί ένας καθηγητής που ζήτησε να έχει μάθημα κάθε μέρα μετά τις 14:00 ή πριν τις 10:00 από κάποιον άλλο που επιθυμεί μαθήματα Δευτέρα με Πέμπτη από τις 10:00 μέχρι τις 16:00. Είναι δηλαδή πολύ σημαντικό και παίζει καθοριστικό ρόλο στα συναισθήματα που θα δημιουργηθούν, το ποιες προτιμήσεις θα επιστραφούν από τους καθηγητές.

4.3 Ανάπτυξη του Συναισθηματικού Πράκτορα Διασύνδεσης

Σε αυτή τη φάση θα αναφερθούν, κατ' αρχήν τα βήματα δημιουργίας του ωρολογίου προγράμματος από τους αντίστοιχους πράκτορες για να γίνει περισσότερο κατανοητή η διαδικασία που ακολουθείται και σε ποιες περιπτώσεις μπορεί να εμπλακεί ο Συναισθηματικός Πράκτορας Διασύνδεσης:

- ***Βήμα 1ο:*** Αρχικά στέλνει ο Scheduler ένα μήνυμα εισαγωγής (call_for_proposal) στους πράκτορες-καθηγητές (Teacher Agents) και στους πράκτορες-μαθητές (Student Agents) ότι ξεκινάει η διαδικασία και μπορούν να στείλουν τις προτιμήσεις (conflicts) τους.
- ***Βήμα 2ο:*** Οι πράκτορες-καθηγητές και οι πράκτορες-μαθητές στέλνουν τις προτιμήσεις τους.
- ***Βήμα 3ο:*** Παίρνει ο Scheduler τις προτιμήσεις και προσπαθεί να τις ταιριάξει στο πρόγραμμα.

-
- **Βήμα 4ο/Σενάριο 1ο:** Αν ο Scheduler ικανοποιήσει κάποιον πράκτορα-καθηγητή, στέλνει μήνυμα στον ίδιο ότι έχει τελειώσει καθώς και το πρόγραμμά του με τα μαθήματα στις ώρες που ζήτησε.
 - **Βήμα 4ο/Σενάριο2ο:** Αν υπάρχουν conflicts, ο Scheduler στέλνει πίσω το πρόγραμμα με τις προτιμήσεις που μπόρεσε να ικανοποιήσει καθώς και αυτές που δεν ικανοποιούνται και ζητάει από τους πράκτορες-καθηγητές να στείλουν πάλι τις προτιμήσεις τους.
 - **Βήμα 5ο:** Οι πράκτορες-καθηγητές βλέπουν το πρόγραμμα που έχει δημιουργηθεί και στέλνουν πάλι νέες προτιμήσεις οι οποίες βέβαια μπορεί να είναι και ίδιες με τις αρχικές.
 - **Βήμα 6ο:** Όπως και στο βήμα 3, ο Scheduler παίρνει τις προτιμήσεις των καθηγητών και προσπαθεί να τις βάλει στο πρόγραμμα.
 - **Βήμα 7ο:** Όπως και στο βήμα 4, αν ο Scheduler ικανοποιήσει κάποιον πράκτορα-καθηγητή, στέλνει μήνυμα στον ίδιο ότι έχει τελειώσει καθώς και το πρόγραμμά του με τα μαθήματα στις ώρες που ζήτησε.
 - **Βήμα 8ο:** Αν ο Scheduler δεν καταφέρει πάλι να τους ικανοποιήσει, στέλνει τα conflicts, το υπάρχον πρόγραμμα μέχρι εκείνη τη στιγμή καθώς και προτάσεις για να διαλέξουν οι καθηγητές που θα τοποθετήσουν τα μαθήματα που δεν έχουν μπει ακόμη στο πρόγραμμα.
 - **Βήμα 9ο:** Οι πράκτορες-καθηγητές στέλνουν τις προτιμήσεις τους σχετικά με τις προτάσεις που έλαβαν από τον Scheduler.
 - **Βήμα 10ο:** Παίρνει ο Scheduler τις προτιμήσεις τους, ελέγχει αν μπορεί να τις ικανοποιήσει και αν δεν μπορέσει, γεμίζει το πρόγραμμα με τα μαθήματα των καθηγητών αλλά με τυχαίο αλγόριθμο.
 - **Βήμα 11ο:** Ο Scheduler στέλνει πίσω το τελικό πρόγραμμα στους πράκτορες-καθηγητές.

Για την ανάπτυξη και την υλοποίηση του Πράκτορα Συναισθημάτων αποφασίσαμε να χρησιμοποιήσουμε τα Δυαδικά Δένδρα Αναζήτησης (Binary Search Trees) για να μπορέσουμε να δημιουργήσουμε ένα δένδρο πέντε επιπέδων, όπου κάθε επίπεδο θα αποτελεί και ένα κριτήριο διαπραγμάτευσης του πράκτορα καθηγητή (Teacher Agent) με τον πράκτορα που φτιάχνει το πρόγραμμα (Program Agent). Σύμφωνα λοιπόν με τις προτιμήσεις που θα επιστρέφονται από τους καθηγητές θα διασχίζουμε τους κόμβους του δένδρου μας και όταν φτάνουμε σε κάποιο κόμβο που δε θα έχει υποδέντρα, δηλαδή θα είναι φύλλο του δένδρου, τότε θα επιστρέφεται η αντίστοιχη εικόνα από τον πράκτορά μας ανάλογα με την

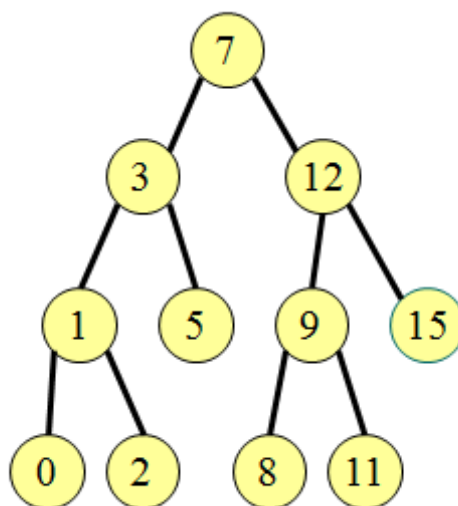
τιμή που θα έχει ο συγκεκριμένος κόμβος. Λίγα λόγια για τα δυαδικά δένδρα αναζήτησης θα αναφέρουμε παρακάτω:

Δυαδικά Δένδρα Αναζήτησης: Στην επιστήμη των υπολογιστών, το δυαδικό δέντρο είναι μία δομή δεδομένων σε ένα δέντρο, στο οποίο κάθε κόμβος έχει το πολύ δύο παιδιά, τα οποία αναφέρονται ως το αριστερό παιδί και το δεξί παιδί. Ουσιαστικά είναι ένας τρόπος αναπαράστασης μίας ταξινομημένης λίστας αντικειμένων. Υποστηρίζει τις παρακάτω διεργασίες σε γρήγορο σχετικά χρόνο:

- Αναζήτηση ενός υπάρχοντος αντικειμένου.
- Εισαγωγή ενός καινούριου αντικειμένου, αν δεν είναι ήδη παρόν.

Τα αντικείμενα αποθηκεύονται στους κόμβους του δέντρου και το δέντρο είναι οργανωμένο κατά τέτοιο τρόπο ώστε να ισχύει πάντα το παρακάτω:

- Το κλειδί κάθε κόμβου στο αριστερό υποδέντρο του x είναι μικρότερο του κλειδιού του x .
- Το κλειδί κάθε κόμβου στο δεξιό υποδέντρο του x είναι μεγαλύτερο του κλειδιού του x .



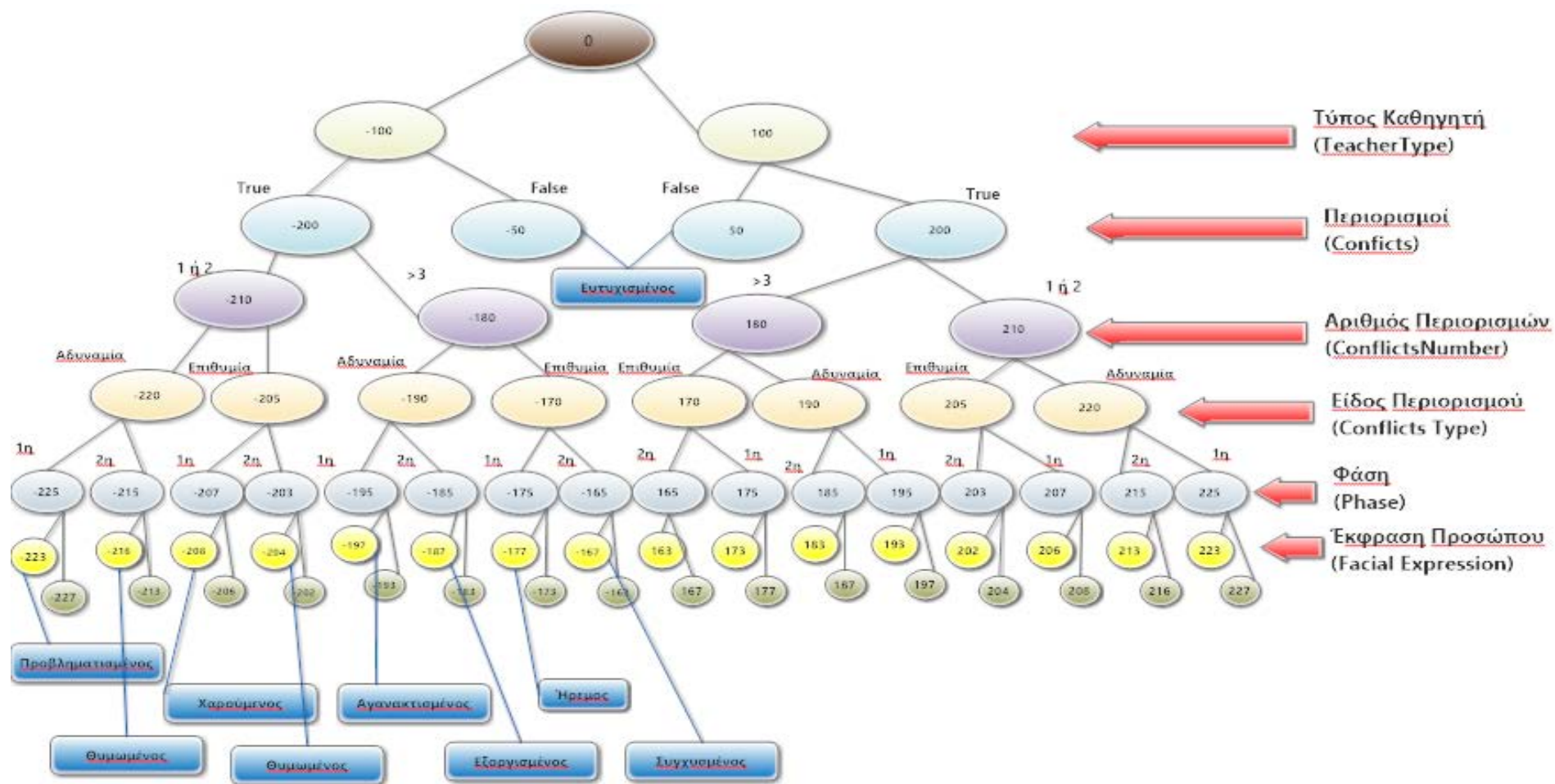
Εικόνα 22 Δυαδικό Δένδρο Απόφασης

Ένα δυαδικό δέντρο αναζήτησης είναι μια ειδική κατηγορία ενός δέντρου απόφασης. Ένα **δέντρο απόφασης** (decision tree) αναπαριστά μια *διαδικασία λήψης απόφασης*.

Κάθε πιθανό “σημείο απόφασης” ή κατάσταση αναπαριστάται από έναν κόμβο.

Κάθε πιθανή επιλογή που μπορεί να γίνει σε ένα σημείο απόφασης αναπαριστάται με μία ακμή προς έναν κόμβο-παιδί.

Σε εκτεταμένα δέντρα απόφασης που χρησιμοποιούνται στην *ανάλυση αποφάσεων*, συμπεριλαμβάνονται κόμβοι που αναπαριστούν τυχαία γεγονότα και τα αποτελέσματά τους.



Εικόνα 23 Μοντέλο ΣΠΔ με Χρήση Δυαδικού Δένδρου Αναζήτησης

4.4 Υλοποίηση του Συναισθηματικού Πράκτορα Διασύνδεσης

Το πολυπρακτορικό σύστημα και συνεπακόλουθα και του ΣΠΔ αποφασίστηκε να υλοποιηθεί στο περιβάλλον του eclipse με τη χρήση του Java Agent Development Framework (JADE), το οποίο περιέχει ενσωματωμένο το Contract Net Protocol, το οποίο θα χρησιμοποιηθεί για την ανάπτυξη της εφαρμογής. Σε αυτό το σημείο απλά θα αναφερθούμε επιγραμματικά στα εργαλεία που χρησιμοποιήσαμε για να υλοποιήσουμε τον πράκτορα συναισθημάτων μας:

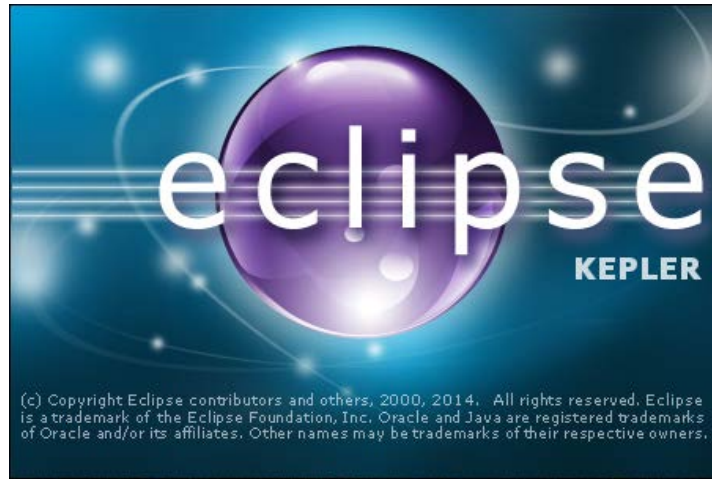
- Eclipse. Στον προγραμματισμό ηλεκτρονικών υπολογιστών, το Eclipse είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης. Περιέχει ένα βασικό χώρο που γράφεται ο κώδικας και ένα σύνολο προσθέτων (plug-ins) που είναι επεκτάσιμα έτσι ώστε να προσαρμόζεται το περιβάλλον. Το Eclipse, γραμμένο κυρίως σε Java, μπορεί να χρησιμοποιηθεί για την ανάπτυξη εφαρμογών. Με τη βοήθεια των διαφόρων plug-ins, το Eclipse μπορεί να χρησιμοποιηθεί και για την ανάπτυξη εφαρμογών σε άλλες γλώσσες όπως COMBOL, JavaScript, Fortran, Python, Perl, PHP, Prolog, Ruby κ.ά. Χρησιμοποιείται ακόμη αρκετά για την ανάπτυξη πακέτων του λογισμικού Mathematica. Η αρχική βάση κώδικα προέρχεται από την IBM VisualAge. Το λογισμικό Eclipse το οποίο περιλαμβάνει τα εργαλεία ανάπτυξης της Java, προορίζεται κυρίως για προγραμματιστές Java. Οι χρήστες μπορούν να επεκτείνουν τις ικανότητές τους με την εγκατάσταση plug-ins για την πλατφόρμα Eclipse όπως είναι η ανάπτυξη εργαλείων για άλλες γλώσσες προγραμματισμού. Εκδίδεται σύμφωνα με τους όρους της Eclipse Public License και είναι ελεύθερο και ανοικτού κώδικα λογισμικό.
- JAVA. Ένα από τα βασικά πλεονεκτήματα της Java έναντι των περισσότερων άλλων γλωσσών είναι η ανεξαρτησία του λειτουργικού συστήματος και πλατφόρμας. Τα προγράμματα που είναι γραμμένα σε Java τρέχουν ακριβώς το ίδιο σε Windows, Linux, Unix και Macintosh (σύντομα θα τρέχουν και σε PlayStation καθώς και σε άλλες κονσόλες παιχνιδιών) χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση (compiling) ή να αλλάξει ο πηγαίος κώδικας για κάθε διαφορετικό λειτουργικό σύστημα. Για να επιτευχθεί όμως αυτό χρειαζόταν κάποιος τρόπος έτσι ώστε τα προγράμματα γραμμένα σε Java να μπορούν να είναι «κατανοητά» από κάθε υπολογιστή ανεξάρτητα του είδους επεξεργαστή αλλά και λειτουργικού συστήματος. Ο λόγος είναι ότι κάθε κεντρική μονάδα επεξεργασίας κατανοεί διαφορετικό κώδικα μηχανής. Ο συμβολικός κώδικας (*assembly*) που μεταφράζεται και εκτελείται σε Windows είναι διαφορετικός από αυτόν που μεταφράζεται και εκτελείται σε έναν υπολογιστή Macintosh. Η λύση δόθηκε με την ανάπτυξη της *Εικονικής Μηχανής* (*Virtual Machine* ή VM ή EM στα ελληνικά).

-
- **JAVA Swing.** Το Swing είναι ένα προηγμένο εργαλείο της Java το οποίο παρέχει μια γραφική διεπαφή χρήστη για τους προγραμματιστές της. Αναπτύχθηκε για να παρέχει ένα πιο εξελιγμένο σύνολο στοιχείων GUI σε σχέση με το προηγούμενο παρόμοιο εργαλείο. Παρέχει μία φυσική αίσθηση, είναι φιλικό προς το χρήστη και τα στοιχεία του είναι πιο ευέλικτα και δυναμικά. Εκτός από τα γνωστά στοιχεία όπως κουμπιά, πίνακες ελέγχου και ετικέτες, το Swing περιέχει και καινούρια στοιχεία όπως καρτέλες πάνελ, παράθυρα κύλισης, δένδρα, πίνακες και λίστες. Τα στοιχεία του Swing δεν εφαρμόζονται από ειδικό κώδικα πλατφόρμας αλλά είναι γραμμένα εξ' ολοκλήρου σε Java και αυτό το στοιχείο το κάνει αρκετά ελαφρύ σαν πρόγραμμα.
 - **JADE.** Η JADE είναι μία ιδιόκτητη πλατφόρμα αντικειμενοστραφούς ανάπτυξης λογισμικού και προϊόντων που έχει ως βάση τη Νέα Ζηλανδία και κυκλοφόρησε για πρώτη φορά το 1996. Αποτελείται από τη γλώσσα προγραμματισμού JADE, μεταγλωττιστή, ενσωματωμένο server εφαρμογής και ένα σύστημα διαχείρισης των αντικειμένων μίας βάσης δεδομένων. Είναι σχεδιασμένο ως περιβάλλον ανάπτυξης end-to-end για να επιτρέπει στα συστήματα να κωδικοποιούνται σε μία γλώσσα από το διακομιστή της βάσης δεδομένων μέχρι τους πελάτες. Επιπλέον, παρέχει APIs για άλλες γλώσσες συμπεριλαμβανομένων των .NET, Framework, Java, C/C++ και τις υπηρεσίες WEB. Ως γλώσσα προγραμματισμού, κύριοι ανταγωνιστές της είναι η Java και η C και ως βάση δεδομένων ανταγωνίζεται με αντικειμενοστραφείς και μετά-σχεσιακές βάσεις καθώς και με πακέτα λογισμικού βάσεων δεδομένων όπως η Oracle και η Microsoft SQL Server. Ουσιαστικά, η JADE είναι κάποιες βιβλιοθήκες της Java και κυκλοφορεί μία περιορισμένη άδεια χρήσης για την ανάπτυξη λογισμικού.

Προκειμένου να λειτουργήσει σωστά η εφαρμογή μας χρειάστηκε να γίνουν τα ακόλουθα βήματα:

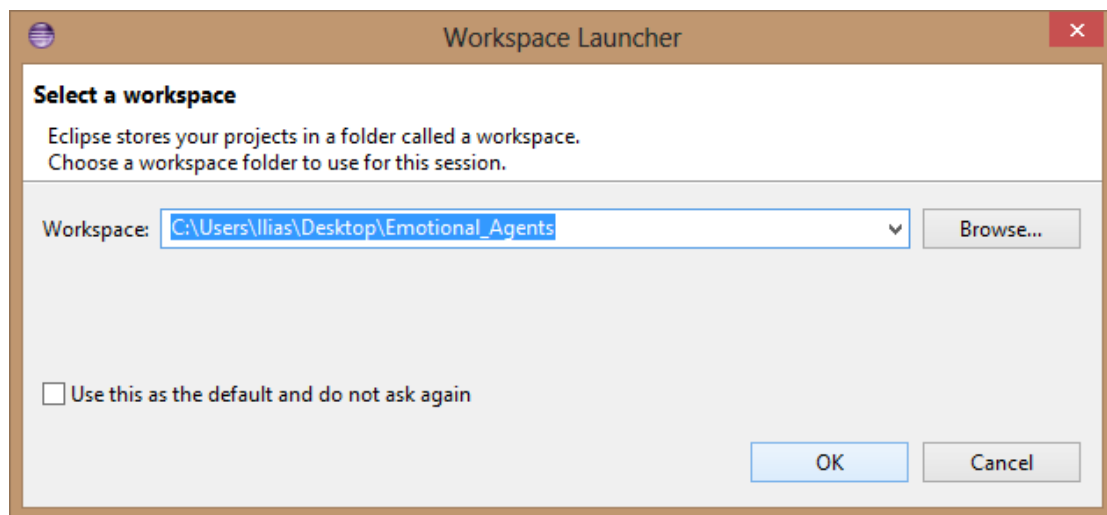
1. Τοποθέτηση των αρχείων της εφαρμογής στο workspace του eclipse.
2. Τοποθέτηση των αρχείων της JADE στο workspace του eclipse.
3. Δεξί κλικ στο Project-Properties-Java Build Path-Add External JARs. Εκεί τοποθετούμε τα jade.jar και commons-codec-1.3.jar.
4. Run Configurations-Java Application. Στο main - main class επιλέγουμε jade.Boot και στο Arguments -gui και τη λίστα των πρακτόρων που θέλουμε να εκκινήσουμε.

Παρακάτω, αναφέρονται λίγο πιο αναλυτικά τα βήματα που απαιτούνται για να είναι πιο εύκολη η εγκατάσταση και η προσαρμογή με το Eclipse και τη JADE. Αρχικά, αφού εγκαταστήσουμε το Eclipse, ανοίγουμε το περιβάλλον ανάπτυξης.



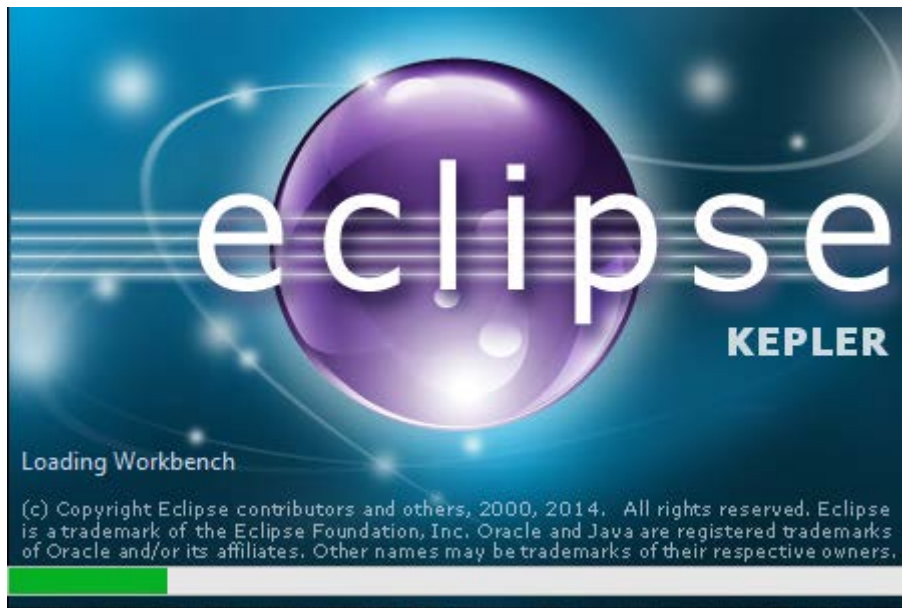
Εικόνα 24 Περιβάλλον ανάπτυξης Eclipse

Στη συνέχεια δηλώνεται ο χώρος εργασίας (Workspace) από τον οποίο αντλεί τις πληροφορίες το ανάλογο project όπως φαίνεται στην παρακάτω εικόνα.

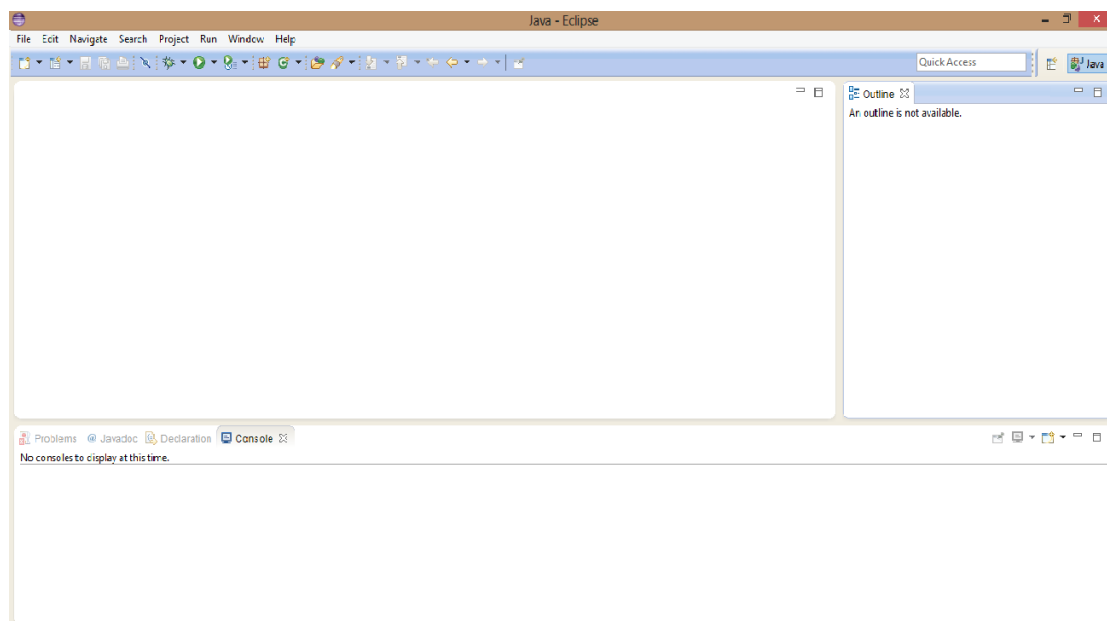


Εικόνα 25 Eclipse_Workspace

Αφού φορτώσει το Eclipse όπως φαίνεται παρακάτω, ανοίγει το κεντρικό παράθυρο της εφαρμογής με τις διάφορες επιλογές που υπάρχουν.

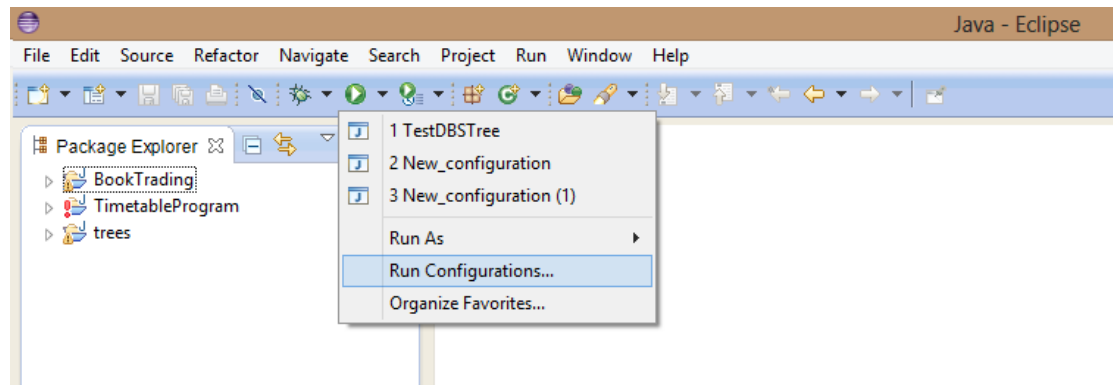


Εικόνα 26 Άνοιγμα του Eclipse

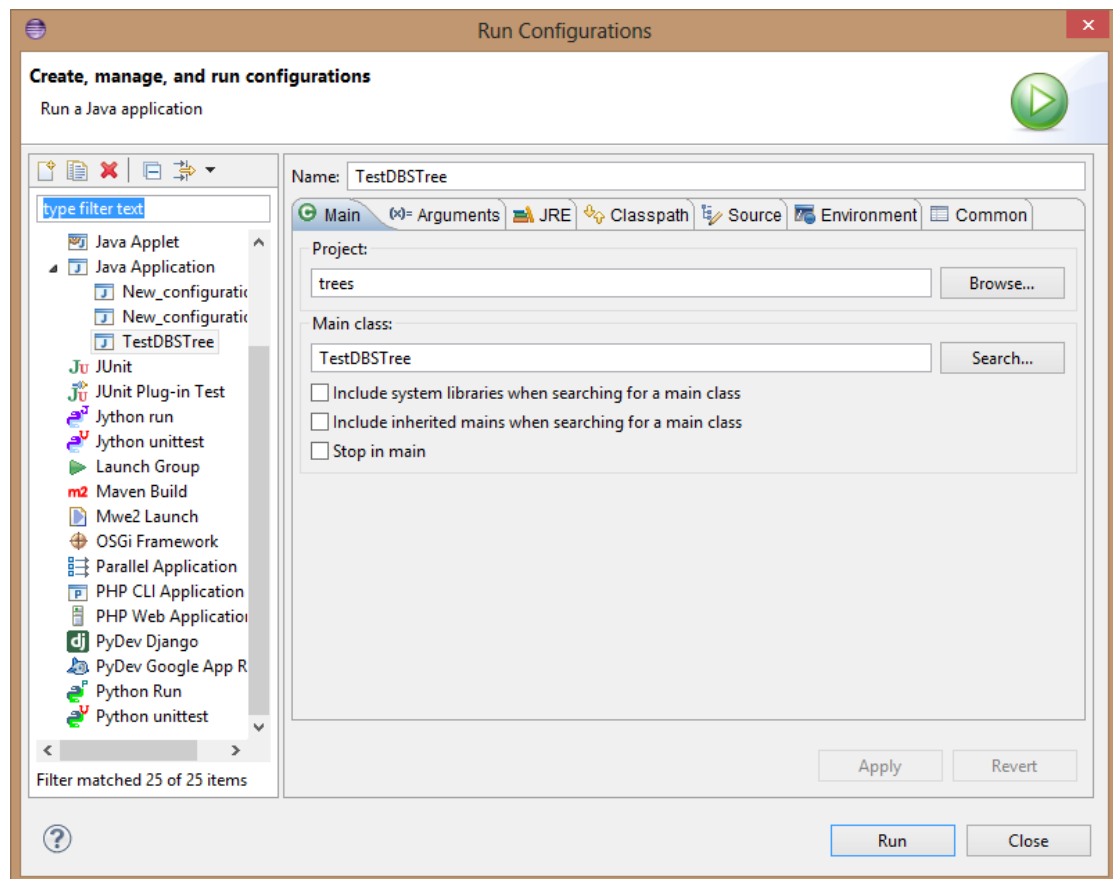


Εικόνα 27 Κεντρικό παράθυρο περιβάλλοντος Eclipse

Πατώντας στο κουμπάκι εκτέλεσης (Run) της εφαρμογής ανοίγει ένα παράθυρο που φαίνεται παρακάτω στο οποίο διαμορφώνουμε την εκτέλεση του προγράμματος.

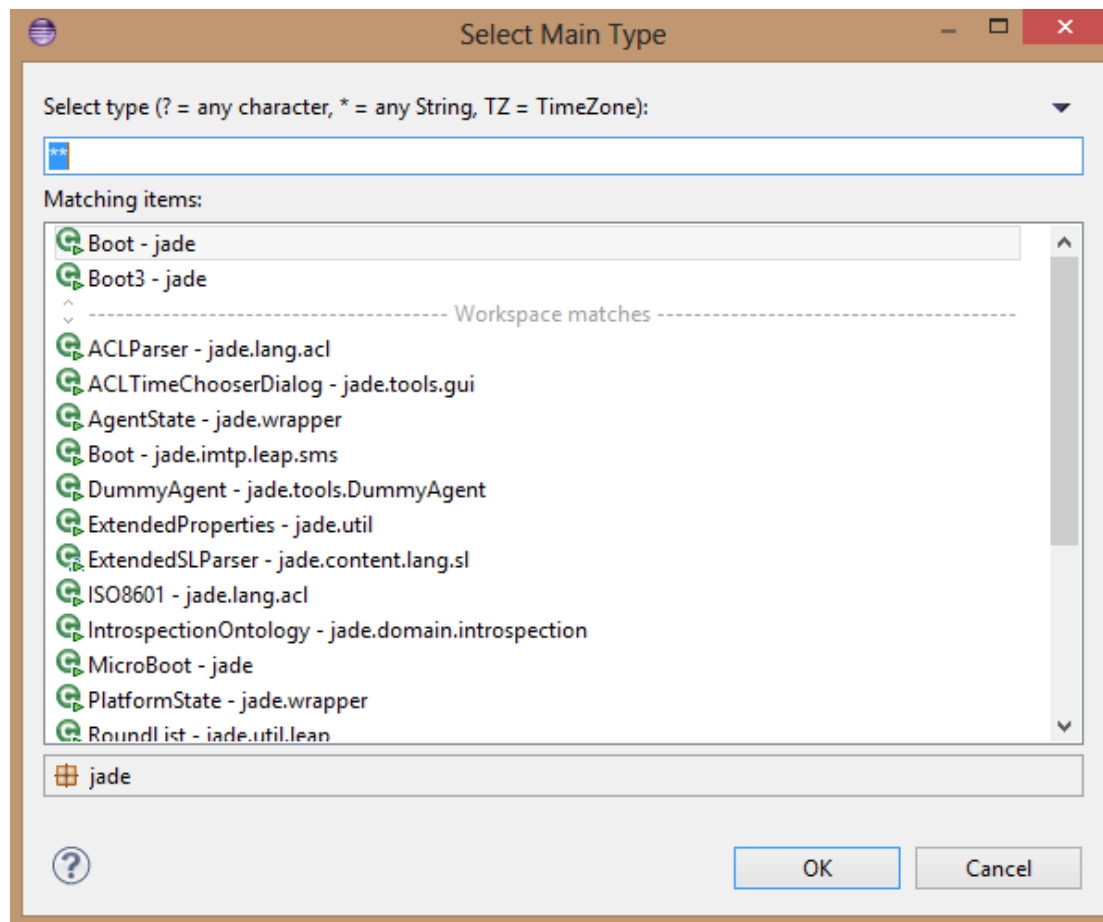
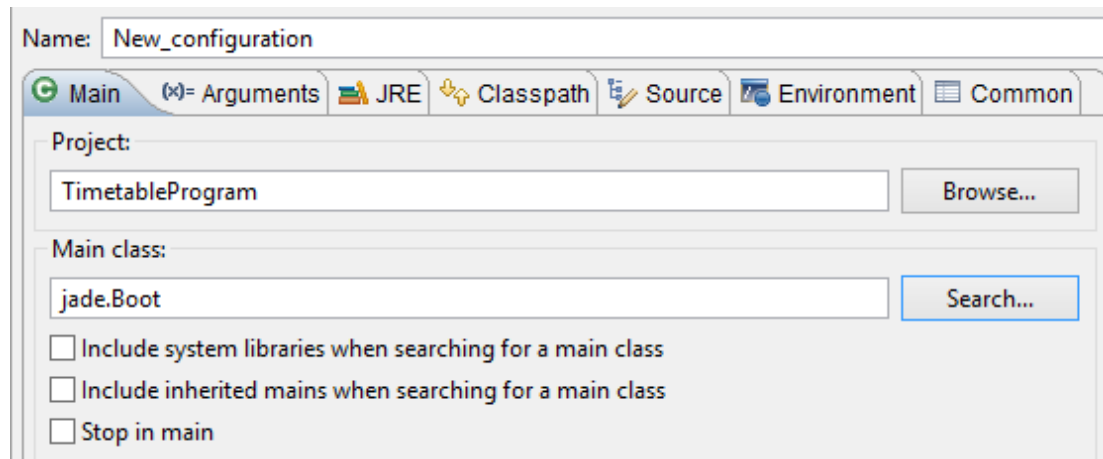


Εικόνα 28 Διαμορφώσεις εκτέλεσης



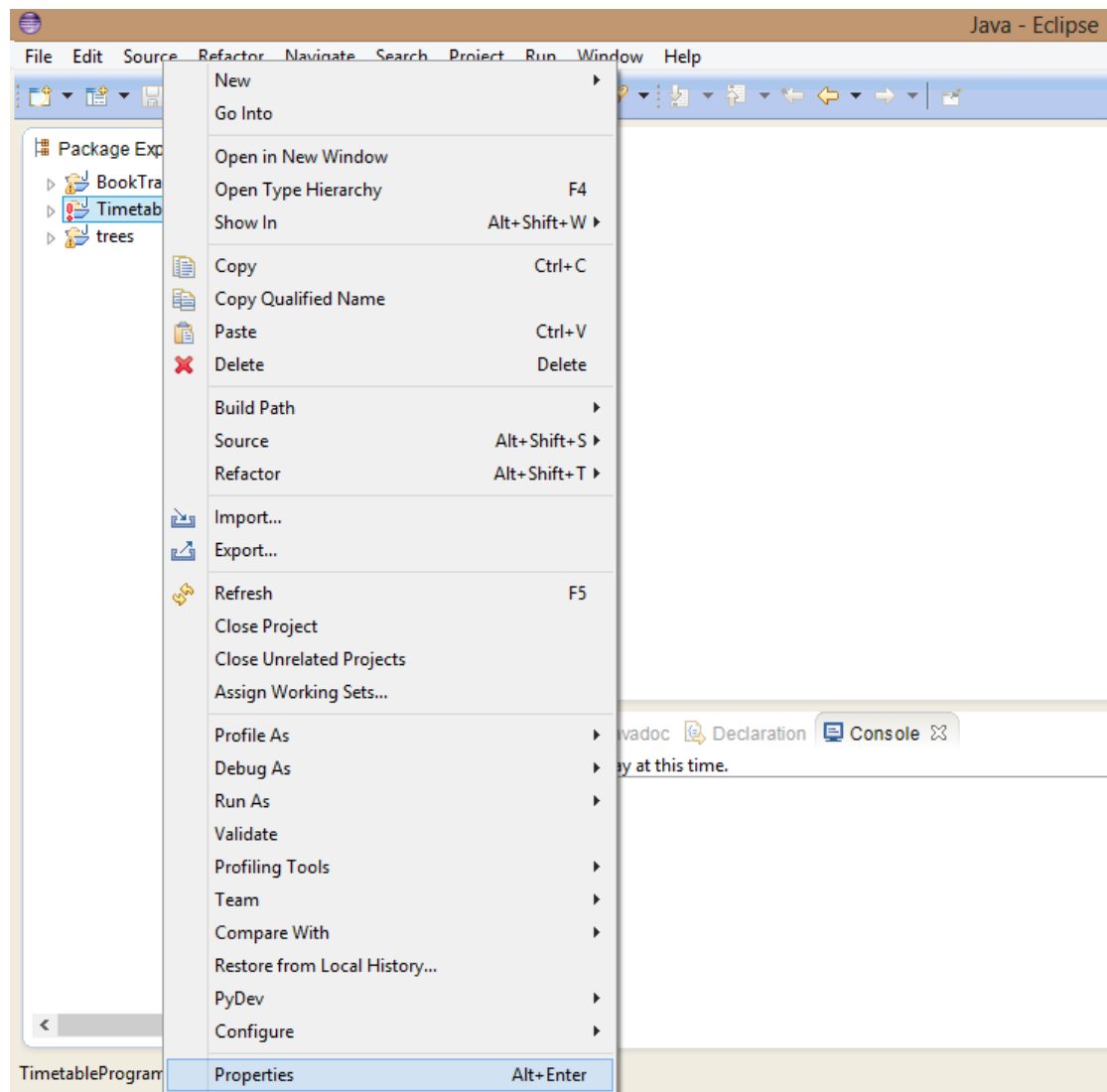
Εικόνα 29 Εισαγωγή ονόματος του project

Στη συνέχεια ορίζουμε την κύρια κλάση του προγράμματος που πρέπει να είναι πάντα η `JadeBoot` για να μπορεί να τρέχει σωστά η εφαρμογή. Ο τρόπος επιλογής της κύριας κλάσης φαίνεται στην παρακάτω εικόνα.

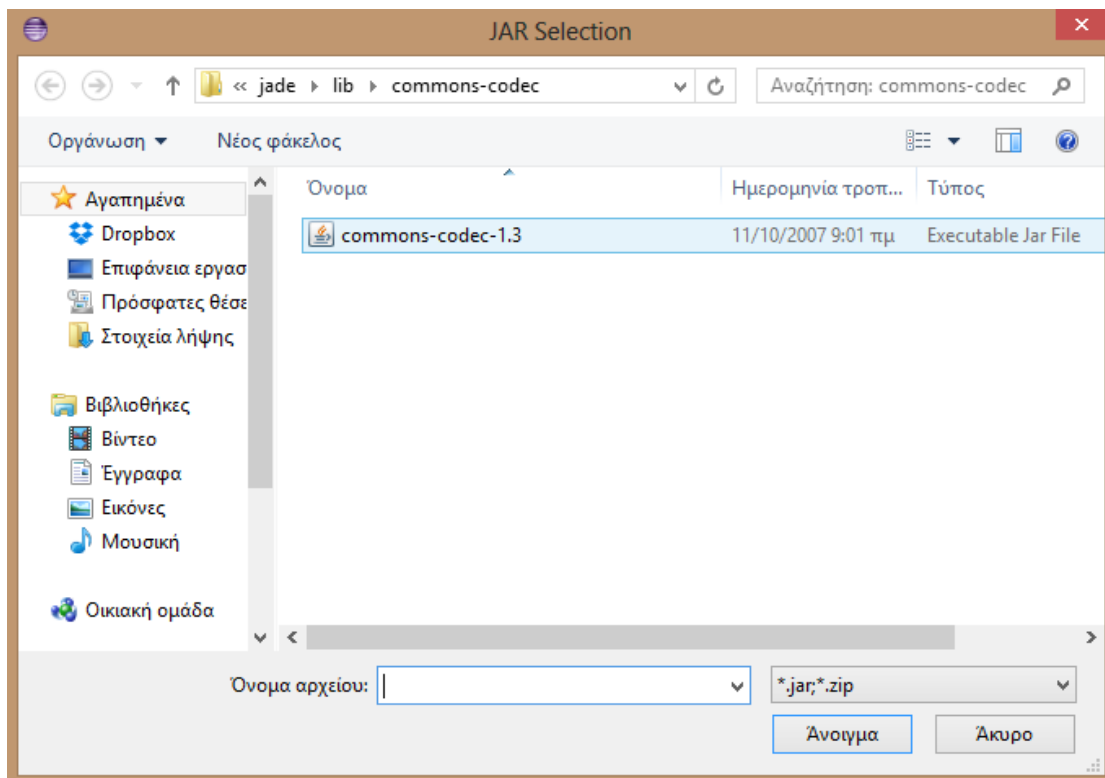
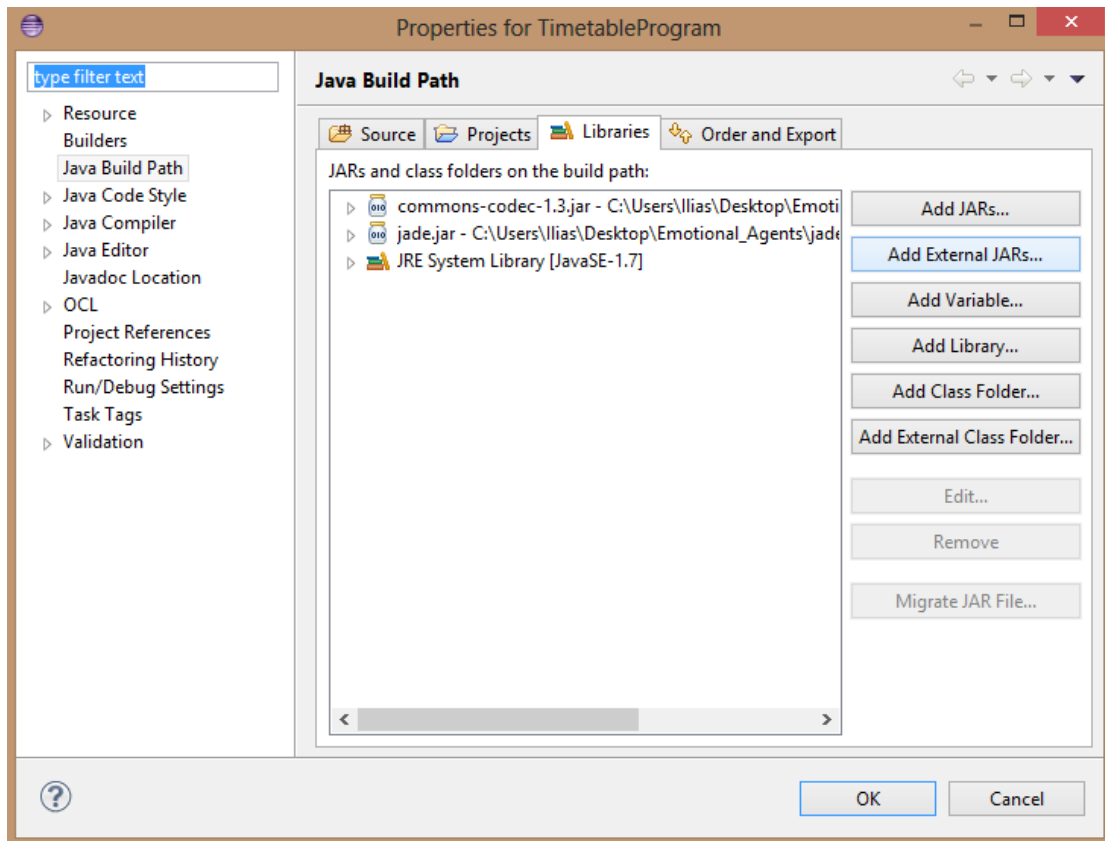


Εικόνα 30 Ορισμός κύριας κλάσης

Επιπλέον, είναι απαραίτητο να ρυθμιστούν κάποιες ιδιότητες για το project που θα τρέξει και να προστεθούν οι κατάλληλες βιβλιοθήκες που περιέχουν τις σωστές πληροφορίες. Ο τρόπος ρύθμισης των ιδιοτήτων και προσθήκης των βιβλιοθηκών του προγράμματος παρουσιάζεται στις παρακάτω τρεις εικόνες.



Εικόνα 31 Ορισμός Ιδιοτήτων προγράμματος



Εικόνα 32 Προσθήκη βιβλιοθηκών προγράμματος

Έχοντας εκτελέσει όλες τις παραπάνω ενέργειες, το περιβάλλον ανάπτυξης του Eclipse, έχει πάρει την κύρια κλάση και έχει ενσωματωθεί με τις κατάλληλες βιβλιοθήκες έτσι ώστε να μπορεί να λειτουργήσει σωστά η πλατφόρμα της Jade.

Για την υλοποίηση του μοντέλου του ΣΠΔ με τη βοήθεια δυαδικού δέντρου αναζήτησης αρχικά δημιουργήσαμε την κλάση `TreeNode` που ουσιαστικά καλείται από την κεντρική μας κλάση για να γεμίσει τους κόμβους του δένδρου μας με τις τιμές που δέχεται. Ενδεικτικά ένα κομμάτι κώδικα που την περιγράφει είναι το παρακάτω:

```
public TreeNode(Comparable obj)
```

```
{  
    item = obj;  
    left = right = null;  
}
```

Returns the data that the node contains

```
public Comparable getNodeItem()
```

```
{  
    return item;  
}
```

Return left tree node

```
public TreeNode getLeftNode()
```

```
{  
    return left;  
}
```

Return right tree node

```
public TreeNode getRightNode()
```

```
{  
    return right;  
}
```

Return true if the node is leaf

```
public boolean isLeaf()
```

```
{  
    return ( (left == null) && (right==null));  
}
```

Στη συνέχεια, στην υπάρχουσα κλάση του πράκτορα καθηγητή (TeacherAgent) δημιουργούμε τη μέθοδο createTreeParams η οποία δέχεται ως παραμέτρους τον αριθμό των προτιμήσεων που στέλνουν οι καθηγητές καθώς και τη φάση της διαπραγμάτευσης στην οποία βρισκόμαστε. Μέσα στη συγκεκριμένη μέθοδο υπάρχουν και άλλες τρεις μέθοδοι, οι int numConf, int typeConf και String grade που μας επιστρέφουν τον αριθμό των προτιμήσεων, το είδος των προτιμήσεων και τον τύπο του καθηγητή αντίστοιχα. Ο κώδικας με τον οποίο πραγματοποιείται αυτό που προαναφέρθηκε φαίνεται παρακάτω:

```
private void createTreeParams(String conflicts, int stage)
{
    int numConf = conflictsNum(conflicts);
    int typeConf = conflictsType(conflicts);
    String grade = "";
    try {
        grade = getTeacherType();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

Η μέθοδος η οποία δημιουργήθηκε για να μας επιστρέφει τον τύπο του καθηγητή, αν δηλαδή είναι Καθηγητής/Αναπληρωτής Καθηγητής ή Καθηγητής Εφαρμογών/Επίκουρος Καθηγητής φαίνεται στο παρακάτω τμήμα κώδικα:

```
public String getTeacherType() throws IOException{
    String[] l;
    BufferedReader teachers_file = new BufferedReader(new
    FileReader("data/teachers.txt"));
    String teachers_line;
    LineNumberReader lnr = new LineNumberReader(teachers_file);
    int linenumber = 0;
    while (lnr.readLine() != null)
    {
        linenumber++;
    }
}
```

```

}

teachers_file.close();

Inr.close();

teachers_file = new BufferedReader(new FileReader("data/teachers.txt"));

String [][] teacher = new String[linenumber][2];

linenumber = 0;

String delimiter = " ";

while((teachers_line = teachers_file.readLine()) != null)

{

    l = teachers_line.split(delimiter);

    teacher[linenumber][0] = l[0];

    teacher[linenumber][1] = l[1];

    linenumber++;

}

teachers_file.close();

String teacherName = getAID().getName().split("@")[0];

String grade = "";

for (int i = 0; i < teacher.length; i++) {

    if (teacher[i][0].equals(teacherName)) {

        grade = teacher[i][1]; }

}

return grade;

}

```

Η μέθοδος που δημιουργήθηκε για να μας επιστρέφει τον αριθμό των προτιμήσεων, όταν δηλαδή οι καθηγητές επιστρέψουν conflicts και δεν ικανοποιηθεί το πρόγραμμά τους στην πρώτη φάση, είναι η conflictsNum και φαίνεται στο παρακάτω τμήμα

κώδικα:

```

public int conflictsNum(String conflicts)
{
    String delimiter = "#";
    prefNegotiation = conflicts.split(delimiter);
    int num = prefNegotiation.length/6;
    System.out.println("num="+num);
    return num;
}

```

Αντίστοιχα, η μέθοδος που δημιουργήθηκε για να μας επιστρέφει το είδος των προτιμήσεων όπως και στην παραπάνω κατάσταση, αν δηλαδή ένας καθηγητής επιλέξει Επιθυμώ/Δεν Επιθυμώ ή Δεν Μπορώ, φαίνεται στο παρακάτω τμήμα κώδικα:

```

public int conflictsType(String conflicts)
{
    int num = 0;
    String delimiter = "#";
    prefNegotiation = conflicts.split(delimiter);
    for (int i = 0; i < prefNegotiation.length; i = i + 6) {
        if (prefNegotiation[i+3].equals("true")) {
            num = 1;
        }else if (prefNegotiation[i+4].equals("true")) {
        }else if (prefNegotiation[i+5].equals("true")) {
            num = 3;
        }
        System.out.println("numreturn="+num);
        return num;
    }
}

```

Για να μπορέσουμε να καθορίσουμε ποια εικόνα θα εμφανιστεί κάθε φορά, δημιουργήσαμε ένα πίνακα τύπου Object που είναι 5 θέσεων και δέχεται ως παραμέτρους τα πέντε διαφορετικά κριτήρια της διαπραγμάτευσης τα οποία είναι:

1. Ο τύπος του καθηγητή.
2. Το αν έχουν επιστραφεί ή όχι conflicts.
3. Τον αριθμό των conflicts, αν δηλαδή είναι 1 ή 2 ή >3.
4. Το είδος των conflicts, αν δηλαδή έχει επιστραφεί Επιθυμώ/Δεν Επιθυμώ ή Δεν Μπορώ.
5. Τη φάση στην οποία βρίσκεται η διαπραγμάτευση, την πρώτη, τη δεύτερη ή την τρίτη.

Η υλοποίηση που αναφέρεται παραπάνω φαίνεται στο παρακάτω τμήμα κώδικα:

```
Object [] params = new Object[5];
String[] parts = returnMsg.split("%");
if (parts.length == 2) {
    conflicts = CreateMessage(parts[0]);
    stage = 1;
    createTreeParams(parts[0], stage);
    if (!parts[1].equals("#")) {
        teacherPr = CreateProgram(parts[1]);
        myGui.userProgram(teacherPr);
    }
    Object picture = TestDBSTree.main(params);
    System.out.println("picture="+picture);
    myGui.notifyUser(conflicts);
} else if (parts.length == 3) {
    stage = 2;
    if (!parts[1].equals("#")) {
        teacherPr = CreateProgram(parts[1]);
        myGui.userProgram(teacherPr);
    }
    createTreeParams(parts[0], stage);
```

```
Object picture = TestDBSTree.main(params);
```

```
System.out.println("picture="+picture);
```

Το πώς λειτουργεί συνολικά η μέθοδος createTreeParams φαίνεται στο παρακάτω τμήμα του κώδικα:

```
private void createTreeParams(String conflicts, int stage){  
    int numConf = conflictsNum(conflicts);  
    int typeConf = conflictsType(conflicts);  
    String grade = "";  
    try {  
        grade = getTeacherType();  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
    int x = 0;  
    if (grade.equals("Professor") || grade.equals("AssistantProfessor")) {  
        params[0] = -100;  
        x = -1;  
    }else{  
        params[0] = 100;  
        x = 1; }  
    if (numConf == 0) {  
        params[1] = 50*x;  
    }else{  
        params[1] = 200*x;  
    }  
    if (numConf <= 2) {  
        params[2] = 210*x;  
        if (typeConf == 1 || typeConf == 2) {  
            params[3] = 205*x;  
            if (stage == 1) {  
                params[4] = 207*x;  
            }else{  
                params[4] = 203*x; }  
            else{  
                params[3] = 220*x;  
            }  
        }  
    }  
}
```

```

if (stage == 1) {
    params[4] = 225*x;
    else{
        params[4] = 215*x; }
    else{
        params[2] = 180*x;
        if (typeConf == 1 || typeConf == 2) {
            params[3] = 170*x;
            if (stage == 1) {
                params[4] = 175*x;
            else{
                params[4] = 165*x; }
            else{
                params[3] = 190*x;
                if (stage == 1) {
                    params[4] = 195*x;
                else{
                    params[4] = 185*x; }}} }

```

Στην κλάση Program στη συνέχεια υπάρχει ένα πίνακας program 3 διαστάσεων. Η πρώτη διάσταση είναι 5 θέσεων για κάθε ημέρα, η δεύτερη διάσταση 13 θέσεων για κάθε ώρα της ημέρας που γίνονται μαθήματα και η Τρίτη διάσταση 7 θέσεων που αντιπροσωπεύουν τα εξάμηνα σπουδών.

Επίσης υπάρχουν τα HashMap conflicts, teacherProgram και semesterProgram, στα οποία κρατούνται οι συγκρούσεις που εμφανίζονται, το πρόγραμμα του κάθε καθηγητή και το πρόγραμμα του κάθε εξαμήνου αντίστοιχα.

Στη μέθοδο createProgram πραγματοποιούνται τα ακόλουθα βήματα:

- Δεσμεύονται θέσεις για το διάστημα που έχει επιλεγεί να πραγματοποιούνται οι αναπληρώσεις του προγράμματος. Επιλέξαμε να γίνονται αναπληρώσεις την Πέμπτη από τις 18:00 μέχρι τις 21:00.
- Γίνεται η τοποθέτηση των προτιμήσεων των σπουδαστών.
- Για κάθε καθηγητή σύμφωνα με τις προτεραιότητες που έχουμε δημιουργήσει προηγουμένως, λαμβάνονται τα μαθήματα του από το αρχείο lessons.txt και ξεκινάει η διαδικασία δημιουργίας του προγράμματος του.

- Αρχικά γίνεται προσπάθεια να ικανοποιήσουμε όλες τις προτιμήσεις που μας έχει αποστείλει, ελέγχοντας ένας πίνακα preference έχουμε δημιουργήσει.
- Αφού ολοκληρωθεί η διαδικασία με την ικανοποίηση των προτιμήσεων, μέσω της μεθόδου putRestLessons, γίνεται προσπάθεια να τοποθετηθούν τα εναπομείναντα μαθήματα.
- Αν δεν τοποθετηθούν όλα τα μαθήματα στο πρόγραμμα, τότε καλείται η μέθοδος putLessonsOnStudentPrefs μέσω της οποίας τοποθετούνται τα μαθήματα στις προτιμήσεις των σπουδαστών.
- Στη συνέχεια καλείται η μέθοδος η putDenErithumwLessons, μέσω της οποίας τοποθετούνται τα εναπομείναντα μαθήματα εκεί που ο καθηγητής δήλωσε ότι δεν επιθυμεί να τοποθετηθεί μάθημα.
- Αν και τώρα δεν τοποθετηθούν μαθήματα, τότε μέσω της CheckPreferencedenMporw τοποθετούνται στο conflicts οι προτιμήσεις του καθηγητή που περιέχουν τον περιορισμό «Δεν μπορώ» και οι όποιες μπλοκάρουν το πρόγραμμα.
- Τοποθετείται στο teacherProgram το πρόγραμμα του καθηγητή.
- Η διαδικασία συνεχίζεται με τον ίδιο τρόπο και για τους υπόλοιπους καθηγητές και τα βήματα 3 έως 9 επαναλαμβάνονται.

Το κομμάτι του κώδικα που εξηγεί την παραπάνω μέθοδο φαίνεται παρακάτω:

```
private String CreateProgram(String returnProg){
String teacherPr = "";
String delimiter = "#";
String [] program = returnProg.split(delimiter);
for (int day = 0; day < 5; day++) {
    String dayName = transformDay(Integer.toString(day));
    teacherPr = teacherPr + dayName;
    for (int i = 0; i < program.length; i=i+4) {
        if (day == 0) {
            program[i+1] = transformDay(program[i+1]);
            program[i+2] = transformHour(program[i+2]); }
        if (dayName.equals(program[i+1])) {
```

```

        teacherPr = teacherPr + " " + program[i] + " " +
program[i+2] + "(" + program[i+3] + ")";}

    }

    teacherPr = teacherPr + "\n";}

    return teacherPr; }

```

Αν δεν ικανοποιηθούν οι προτιμήσεις κάποιου καθηγητή στέλνεται αντίστοιχο μήνυμα το οποίο φαίνεται στο παρακάτω τμήμα κώδικα:

```

String[] parts = returnMsg.split("%");

if (parts.length == 2) {

    conflicts = CreateMessage(parts[0]);

    stage = 1;

    createTreeParams(parts[0], stage);

    if (!parts[1].equals("#")) {

        teacherPr = CreateProgram(parts[1]);

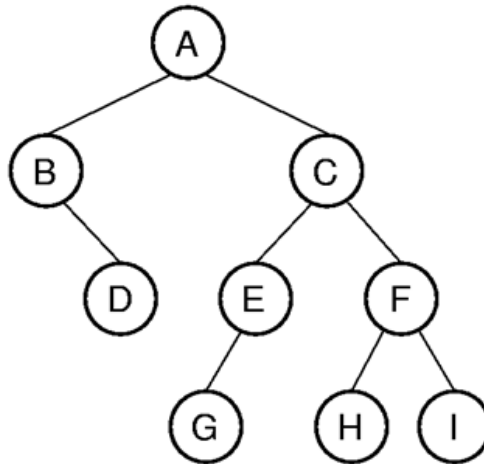
        myGui.userProgram(teacherPr);

    }
}

```

Για την εμφάνιση της αντίστοιχης εικόνας από τον πράκτορα συναισθημάτων (Emotional Agent) ανάλογα με το τι έχει δεχτεί ως παραμέτρους, επιλέξαμε να χρησιμοποιήσουμε ένα δυαδικό δένδρο αναζήτησης πέντε επιπέδων που το κάθε επίπεδο θα αποτελεί και ένα από τα κριτήρια διαπραγμάτευσης. Το πρώτο επίπεδο θα αφορά τον τύπο του καθηγητή, το δεύτερο επίπεδο θα αφορά το αν επέστρεψε ή όχι προτιμήσεις (conflicts) ο καθηγητής, το τρίτο επίπεδο θα αφορά τον αριθμό των προτιμήσεων του καθηγητή, το τέταρτο επίπεδο θα αφορά το τι προτιμήσεις επέστρεψε ο κάθε καθηγητής και το πέμπτο και τελευταίο επίπεδο θα αφορά τη φάση της διαπραγμάτευσης στην οποία βρισκόμαστε. Τα δυαδικά δένδρα αναζήτησης εξηγούνται λίγο πιο αναλυτικά στο θεωρητικό υπόβαθρο αυτής της διπλωματικής εργασίας. Λόγω του ότι θέλουμε να επισκεφθούμε όλους τους κόμβους του δένδρου μας ξεχωριστά, επιλέξαμε να το διασχίσουμε χρησιμοποιώντας τη μέθοδο της Προθεματικής Διάσχισσης (Preorder Traversal) σύμφωνα με την οποία, για κάθε κόμβο του δένδρου, επισκεπτόμαστε πρώτα τον ίδιο τον κόμβο, έπειτα τους κόμβους του αριστερού του υποδένδρου και στη συνέχεια τους κόμβους του δεξιού του υποδένδρου. Εδώ θα αναφερθούμε επιγραμματικά στη συγκεκριμένη μέθοδο.

Προθεματική Διάσχιση (Preorder Traversal): Επισκεπτόμαστε(εκτυπώνουμε) πρώτα κάποιο κόμβο και μετά τα παιδιά του ή διαφορετικά για κάθε κόμβο, επισκεπτόμαστε πρώτα τον ίδιο τον κόμβο, έπειτα τους κόμβους του αριστερού του υποδένδρου και στη συνέχεια τους κόμβους του δεξιού του υποδένδρου.



Εικόνα 33 Μέθοδος Preorder Traversal

➤ Προδιατεταγμένη:
A, B, D, C, E, G, F, H, I

Παρακάτω αναφέρεται ο κώδικας για τη συγκεκριμένη μέθοδο:

```
void PreOrder(BinaryTreeNode<T> *t)  
{// Preorder traversal of *t.  
if (t) {Visit(t); PreOrder(t->LeftChild);  
PreOrder(t->RightChild); }}
```

Παρακάτω φαίνεται το τμήμα του κώδικα που περιγράφει τη μέθοδο Preorder:

```
public void preOrderTraversal() {  
System.out.println("PREORDER TRAVERSAL");  
preOrder(root);  
System.out.println();}
```

Στην κλάση `SentimentAgent` δημιουργείται το δένδρο μας το οποίο έχει 67 κόμβους και το διανύουμε μέσω της μεθόδου `Preorder Traversal` που αναφέραμε παραπάνω. Ο κώδικας με τον οποίο σχηματίζεται το δένδρο φαίνεται στο παρακάτω κομμάτι:

```
class TestDBSTree {  
  
    public static Object main(Object obj[]){  
  
        for (int i = 0; i < obj.length; i++) {  
            System.out.println("obj[i]="+obj[i]);  
        }  
  
        DBSTree emotionTree = new DBSTree();  
  
        int j=67;  
  
        TreeNode a[]=new TreeNode[j];  
  
        for(int i=0; i<j; i++){  
            a[i]=new TreeNode(i);  
            emotionTree.add(a[i].getNodeItem());  
        }  
  
        emotionTree.preOrderTraversal();  
  
        Object face = emotionTree.treeSearch(obj);  
  
        return face;  
  
    }  
}
```

Εδώ ουσιαστικά αφού δημιουργηθεί το δένδρο, το διασχίζουμε μέσω της μεθόδου `Preorder Traversal` και ψάχνουμε στους κόμβους του με τη μέθοδο `getNodeItem` για το αντικείμενο `Object`. Αυτό γίνεται με τη μέθοδο `treeSearch` η οποία παίρνει ως παράμετρο το `Object` και θα εξηγηθεί παρακάτω. Αφού λοιπόν εντοπίσει το `Object` στους κόμβους του δένδρου, το επιστρέφει στη μεταβλητή `face` που είναι και αυτή τύπου `Object`. Η μέθοδος `treeSearch` που προαναφέρθηκε παίρνει ως παράμετρο ένα `Object` και διασχίζει τους κόμβους του δένδρου από τα αριστερά προς τα δεξιά. Φτάνοντας σε κάθε κόμβο, συγκρίνει την τιμή του μέσω της μεθόδου `Comparable` που παίρνει και αυτή ως παράμετρο ένα `Object`. Αν η τιμή που θα βρει είναι `Object`, τότε κάνει το συγκεκριμένο κόμβο ρίζα έτσι ώστε την επόμενη φορά να συνεχίσει να ψάχνει από εκείνο το σημείο. Αν φτάσει σε κάποιο κόμβο ο οποίος δεν έχει υποκόμβους, είναι δηλαδή φύλλο του δένδρου, τότε επιστρέφεται στη μεταβλητή `face` μία συγκεκριμένη εικόνα ανάλογα με τις παραμέτρους και τα κριτήρια που έχουμε θέσει. Το πώς λειτουργεί η μέθοδος `treeSearch` φαίνεται στο παρακάτω τμήμα κώδικα:

```

public Object treeSearch(Object obj[]) {
    Object face = "";
    TreeNode x = root;
    boolean flag = false;

    for(int i=1; i < obj.length; i++){
        if ( !(obj[i] instanceof Comparable) )
            throw new IllegalArgumentException("Argument is not
instance of Comparable.");

        x = search(x,(Comparable)obj[i] );
        if (x.getLeftNode().isLeaf()){ // node is leaf
            System.out.println("aaaaaaa="+x.getRightNode());
            face = x.getRightNode();
        }
        System.out.println("x="+x);
    }
    return face;
}

```

Σε αυτό το σημείο απλά θα αναφέρουμε και τις δύο boolean μεθόδους που χρησιμοποιήθηκαν για την υλοποίηση του δυαδικού δένδρου αναζήτησης, τη μέθοδο contains και τη μέθοδο equals. Τα κομμάτια του κώδικα που υλοποιούν τις δύο παραπάνω μεθόδους, φαίνονται παρακάτω:

```

public boolean contains(Object obj) {
    if ( !(obj instanceof Comparable) )
        //throw new IllegalArgumentException("Argument is not instance of
Comparable.");
    return (search(root,(Comparable)obj) != null); }

public void clear() {
    root=null; }

public boolean equals(DataStructure otherStructure) {
    if (otherStructure.getClass() != DBSTree.class) return false;

```

```

if (otherStructure.size() != size()) return false;

if (size() == 0) return true;

return contains(root, (DBSTree)otherStructure);
}

```

Στη συνέχεια θα αναφερθούν οι δύο void μέθοδοι που χρησιμοποιήσαμε για να προσθέτουμε και να αφαιρούμε στοιχεία από τους κόμβους του δένδρου μας, οι οποίες παίρνουν ως παράμετρο ένα Comparable item και δεν επιστρέφουν κάτι ως έξοδο. Τα κομμάτια κώδικα που υλοποιούν τις δύο μεθόδους φαίνονται παρακάτω:

```

public void add(Comparable item) throws StructureFullException {

    if (isFull()) throw new StructureFullException("The Tree is Full");

    if (isEmpty())

        root = new TreeNode(item);

    else

        place(root,item);}

public void remove(Comparable item) throws StructureEmptyException {

    if (isEmpty()) throw new StructureEmptyException("The Tree is Empty");

    delete(root, null, item);

}

```

Σε αυτό το σημείο θα περιγράψουμε την αναδρομική (recursive) μέθοδο place η οποία αφού προστεθεί κάποιο στοιχείο στο δένδρο, το τοποθετεί στο σωστό κόμβο. Όταν προστεθεί ένα καινούριο στοιχείο στο δένδρο, η place ελέγχει εάν είναι θετικό ή αρνητικό. Αν είναι θετικό, κινείται προς την αριστερή μεριά του δένδρου και ελέγχει τον αριστερό κόμβο. Αν η τιμή του κόμβου είναι null, τότε τοποθετεί το στοιχείο στον κόμβο, αλλιώς αν ο κόμβος έχει κάποια τιμή, ξανακαλεί την place για να κάνει τον ίδιο έλεγχο κάτω από την υπάρχουσα τιμή του αριστερού κόμβου. Με τον ίδιο ακριβώς τρόπο λειτουργεί και για τη δεξιά μεριά του δένδρου με τη διαφορά ότι κινείται προς τα εκεί όταν το καινούριο στοιχείο που είναι να προστεθεί στο δένδρο είναι αρνητικό. Ο κώδικας που περιγράφει την αναδρομή και τον τρόπο λειτουργίας της μεθόδου place φαίνεται παρακάτω:

```

// Recursive place the item to the correct node

private void place(TreeNode node,Comparable item) {

    if (item.compareTo(node.getNodeItem())> 0) //move left

    {

        if (node.getLeftNode()==null)

```



```

        node.setLeftNode(new TreeNode(item));
    }
    else
        place(node.getLeftNode(),item);
}
else // move right
{
    if (node.getRightNode()==null)
        node.setRightNode(new TreeNode(item));
    else
        place(node.getRightNode(),item);
}
}
}

```

Αφού λοιπόν τοποθετηθούν τα στοιχεία στους κόμβους του δένδρου, καλείται η αναδρομική μέθοδος `search` η οποία ελέγχει αρχικά την τιμή του στοιχείου και αν είναι ίδια με την τιμή του κόμβου, τότε επιστρέφει τον κόμβο. Αν η τιμή του στοιχείου είναι αρνητική, τότε κινείται στη δεξιά μεριά του δένδρου και ψάχνει μέσω της `search` τον κόμβο που έχει τη συγκεκριμένη τιμή για να τον επιστρέψει. Σε διαφορετική περίπτωση, όταν δηλαδή η τιμή του στοιχείου είναι θετική, κινείται προς την αριστερή μεριά του κόμβου και ξανακαλεί τη `search` για να επαναλάβει τη διαδικασία εντοπισμού του επιθυμητού κόμβου. Ο κώδικας που ακολουθεί περιγράφει τη διαδικασία που περιγράψαμε παραπάνω:

```

// Recursive search that returns the node containing the item
private TreeNode search(TreeNode node, Comparable item) {
    if (node==null)
        return null;
    if (node.getNodeItem().equals(item))
        return node;
    System.out.println("item="+item);
    System.out.println("node.getNodeItem()="+node.getNodeItem());
    System.out.println("node.getLeftNode()="+node.getLeftNode());
    if (node.getLeftNode().isLeaf()) // node is leaf
        System.out.println("image="+node.getLeftNode());
    if (item.compareTo(node.getNodeItem()) < 0)
        return search(node.getRightNode(), item);
}
}

```

```
        else
            return search(node.getLeftNode(), item);
    }
```

Τέλος, δημιουργήσαμε και την κλάση `TeacherGui` που είναι ουσιαστικά το γραφικό του καθηγητή και μέσα σε αυτήν φτιάχτηκε η μέθοδος με την οποία ο πράκτοράς μας εμφανίζει την κατάλληλη εικόνα. Στην κλάση αυτή δηλώνονται που ακριβώς θα εμφανιστεί το πρόγραμμα του κάθε καθηγητή, οι περιορισμοί που υπάρχουν στις προτιμήσεις του καθώς και η αντίστοιχη έκφραση προσώπου του πράκτορα που ταυτίζεται με τον καθηγητή. Στο κομμάτι του κώδικα που ακολουθεί φαίνονται τα παραπάνω:

```
class TeacherGui extends JFrame {
    private TeacherAgent myAgent;
    JTextArea returnText;
    JTextArea prefText;
    JPanel p = new JPanel();

    JTextArea programText;
    private Choice daychoice, prefchoice, hourchoice;
    private JRadioButton epithymo;
    private JRadioButton denmporw;
    private JRadioButton denepithymo;
    private JPanel p_1;
    private JLabel addedPref;
    private JLabel teacherprogram;
    final JButton addButton;
    final JButton sendButton;
    String showPref = "";
    private ImageIcon image1;
    private JLabel imageLabel;
    JPanel p_0 = new JPanel();
}
```

Η μέθοδος `showImage` δεν επιστρέφει κάτι, απλά εμφανίζει την αντίστοιχη εικόνα στο πλαίσιο που έχουμε θέσει. Μπορούμε να αλλάξουμε τα όρια του πλαισίου ανάλογα με το

σημείο στο οποίο θέλουμε να εμφανίζεται η εικόνα. Η υλοποίηση της μεθόδου φαίνεται στο παρακάτω κομμάτι κώδικα:

```
public void showImage(){
    JLabel lblNewLabel = new JLabel("");
    ImageIcon icon = new ImageIcon("c:\\image.jpg");
    lblNewLabel.setIcon(icon);
    lblNewLabel.setIcon(new
ImageIcon(TeacherGui.class.getResource("image.jpg")));
    lblNewLabel.setBounds(2, 2, 44, 83);
    p_0.add(lblNewLabel);
}
```

Επειδή βέβαια έχουμε διαφορετικές εικόνες που θέλουμε να εμφανίζουμε ανάλογα με τις παραμέτρους που δέχεται ο πράκτοράς μας, επιλέξαμε να μετατρέπουμε αρχικά το νούμερο που μας επιστρέφεται από τον κόμβο-φύλλο του δένδρου σε θετικό ακέραιο. Στη συνέχεια, μετατρέπουμε αυτόν τον ακέραιο αριθμό σε String και το στέλνουμε ως παράμετρο στη μέθοδο showImage της teacherGui. Το κομμάτι του κώδικα παρακάτω περιγράφει την μετατροπή του ακεραίου αριθμού σε string και την είσοδό του ως παράμετρο στη μέθοδο showImage:

```
Object picture = SentimentAgent.main(params);
String teacherImage = String.valueOf(picture);
int pic = Integer.parseInt(teacherImage);
if(pic < 0){
    pic = -pic;
}
teacherImage = Integer.toString(pic);
myGui.showImage(teacherImage);
```

Σε αυτή τη φάση θεωρούμε απαραίτητο να αναφέρουμε τα προγράμματα που χρησιμοποιήθηκαν για την επεξεργασία των εικόνων που εμφανίζει ο πράκτοράς μας στο τελικό στάδιο. Αρχικά, χρησιμοποιήσαμε το Adobe Photoshop για να επεξεργαστούμε τα γραφικά της εφαρμογής.

Adobe Photoshop: Το Adobe Photoshop, ή απλά Photoshop, είναι ένα πρόγραμμα επεξεργασίας γραφικών που αναπτύχθηκε και κυκλοφόρησε από την Adobe Systems. Αυτή τη στιγμή αποτελεί ηγέτη της αγοράς (*market leader*) των προγραμμάτων επεξεργασίας εικόνων, και είναι το προϊόν - σήμα κατατεθέν της Adobe Systems. Χαρακτηρίζεται ως

"απαραίτητο εργαλείο για τους επαγγελματίες γραφίστες" και θεωρείται πως προώθησε τις αγορές των Macintosh, και στη συνέχεια των Windows. Η πιο πρόσφατη έκδοση του Adobe Photoshop είναι η Adobe Photoshop CS6 (13.0), που κυκλοφόρησε τον Μάιο του 2012. Διατίθεται στις εκδόσεις Standard και Extended. Η τελευταία διαφοροποιείται έναντι της απλής έκδοσης χάρη στα εργαλεία επεξεργασίας τρισδιάστατων αντικειμένων και ανάλυσης ποσοτικών δεδομένων εικόνας.

Τέλος, για να επεξεργαστούμε τις εικόνες μας και να δημιουργήσουμε cartoons και μερικά gifs χρησιμοποιήσαμε δύο online εργαλεία που φαίνονται παρακάτω:

- mywebface (<http://download.mywebface.com>)



Εικόνα 34 Επεξεργασία εικόνων με mywebface.com

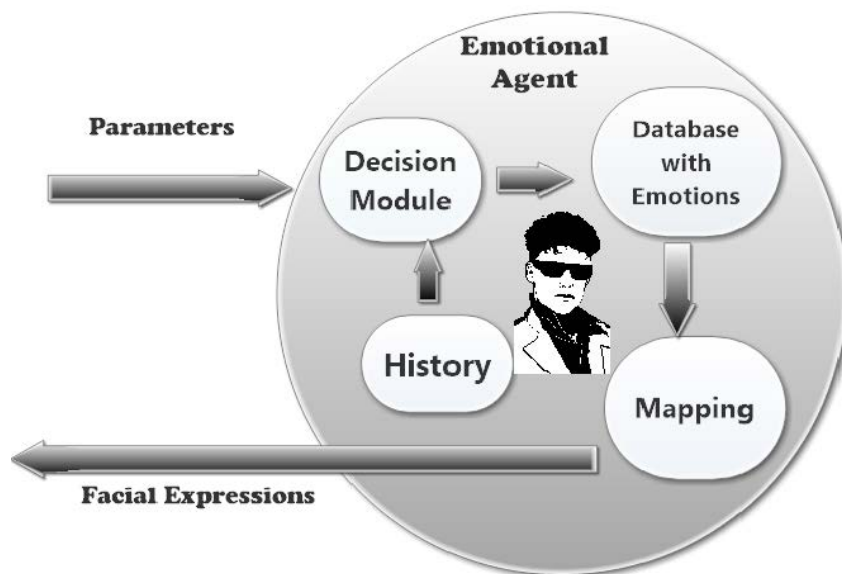
- Animated GIF Maker (<http://ezgif.com/maker>)

Το Animated GIF Maker είναι ένα πολύ εύχρηστο εργαλείο που μας επιτρέπει να δημιουργήσουμε εύκολα και γρήγορα κινούμενα gifs συνδυάζοντας ξεχωριστά αρχεία εικόνων. Εκτός από αρχεία gif, μπορούμε να φορτώσουμε και αρχεία jpg ή png ή bmp τα οποία μετατρέπονται αυτόματα. Μετά τη φόρτωση των εικόνων (καρέ), μπορούμε να ρυθμίσουμε την ταχύτητα κίνησης, τα όρια εμφάνισης του κινούμενου σχεδίου καθώς και πολλά άλλα πραγματάκια όπως η αλλαγή μεγέθους (resize), το κόψιμο (crop) κ.ά.

Ένας πράκτορας συναισθημάτων όπως προαναφέρθηκε και στις αρχιτεκτονικές που είδαμε παραπάνω, δέχεται κάποια ερεθίσματα ως είσοδο στον εσωτερικό του κόσμο, αντιδράει σε αυτά ανάλογα με το μοντέλο διαπραγμάτευσης που έχει, και ανάλογα με την απόφαση που πήρε, εμφανίζει το αντίστοιχο συναίσθημα προς τον πραγματικό κόσμο ως έξοδο. Σε αυτό το σημείο θα προσπαθήσουμε να εξηγήσουμε με τη βοήθεια ενός σχήματος το συνολικό τρόπο λειτουργίας του πράκτορά μας, το τι δέχεται δηλαδή ως είσοδο (input), ποιος είναι ο μηχανισμός αντίδρασής του και το τι εμφανίζει ως έξοδο (output). Συγκεκριμένα λοιπόν, ο πράκτοράς μας μπορεί να δεχτεί ως είσοδο πέντε διαφορετικές παραμέτρους που είναι ουσιαστικά και τα 5 κριτήρια της διαπραγμάτευσης με τους άλλους πράκτορες. Τα κριτήρια αυτά αφορούν:

1. Βαθμίδα καθηγητή, αν δηλαδή είναι Καθηγητής / Επίκουρος Καθηγητής ή αν είναι Καθηγητής Εφαρμογών / Αναπληρωτής Καθηγητής.
2. Επιστροφή προτιμήσεων, αν έχουν επιστραφεί δηλαδή conflicts ή όχι.
3. Σύνολο προτιμήσεων που έχουν επιστραφεί, αν δηλαδή ο αριθμός των conflicts είναι 1 / 2 ή παραπάνω από 3.
4. Είδος προτιμήσεων που έχουν επιστραφεί, αν δηλαδή έχει επιστραφεί Επιθυμώ / Δεν Επιθυμώ ή Δεν μπορώ.
5. Φάση διαπραγμάτευσης, αν δηλαδή βρισκόμαστε στην πρώτη(1η) ή στη δεύτερη(2η) ή στην τρίτη(3η) φάση.

Τις παραμέτρους εισόδου είναι πολύ σημαντικό να τις αναφέρουμε γιατί έχουν ξεχωριστά βάρη η καθεμία και παίζουν καθοριστικό ρόλο στο ποια θα είναι η έξοδος κάθε φορά. Για παράδειγμα, είναι τελείως διαφορετικό να επιστραφούν 1 ή 2 περιορισμοί / προτιμήσεις από έναν Επίκουρο Καθηγητή στην πρώτη φάση της διαπραγμάτευσης από το να επιστραφούν περισσότερες από 3 προτιμήσεις στη δεύτερη φάση της διαπραγμάτευσης από έναν Αναπληρωτή Καθηγητή. Στην πρώτη περίπτωση, ο συναισθηματικός πράκτορας θα είναι σχετικά πιο ήρεμος και χαρούμενος σε αντίθεση με τη δεύτερη περίπτωση που θα εμφανιστεί σαφώς πιο θυμωμένος και αγανακτισμένος. Σε αυτό το σημείο καλό θα ήταν να αναφέρουμε ότι η βαρύτητα των παραμέτρων είναι κλιμακωτή, κάτι που σημαίνει ότι ένας Καθηγητής έχει σαφώς μεγαλύτερη βαρύτητα από έναν Αναπληρωτή Καθηγητή, η επιστροφή προτίμησης Δεν Μπορώ από κάποιον καθηγητή έχει σαφώς μεγαλύτερη βαρύτητα από την προτίμηση Επιθυμώ και τέλος, αν βρισκόμαστε στην τελευταία φάση της διαπραγμάτευσης έχει σαφώς μεγαλύτερη βαρύτητα από το να βρισκόμαστε στην αρχική.



Εικόνα 35 Συναισθηματικός πράκτορας διασύνδεσης (ΣΠΑ)

Σε συνέχεια λοιπόν αυτής της κατάστασης, ο συναισθηματικός πράκτορας παίρνει τις παραμέτρους που δέχθηκε και τις εισάγει σε ένα μοντέλο απόφασης (decision module) που έχει στο εσωτερικό του για να μπορέσει να τις επεξεργαστεί και να καταλήξει σε ένα τελικό συμπέρασμα. Το μοντέλο απόφασης του πράκτορα συναισθημάτων στη συγκεκριμένη διπλωματική εργασία είναι ένα Δυαδικό Δένδρο Αποφάσεων (Binary Decision Tree). Τα πέντε κριτήρια που δέχεται ως παραμέτρους ο πράκτορας είναι ουσιαστικά και τα πέντε επίπεδα του δένδρου. Ανάλογα με το ποια παράμετρο δέχεται κάθε φορά, ο πράκτορας μετακινείται στους κόμβους του δένδρου διασχίζοντάς το με την προδιατεταγμένη μέθοδο (Preorder Traversal) που αναφέραμε παραπάνω, μέχρι να καταλήξει σε ένα κόμβο του δένδρου που είναι φύλλο, δηλαδή δεν έχει υποκόμβους, οπότε και θα εμφανίσει την αντίστοιχη εικόνα. Πολύ σημαντικό σε αυτό το σημείο είναι να αναφέρουμε πως το μοντέλο απόφασης του πράκτορά μας λειτουργεί σε συνάρτηση με το ιστορικό των διαπραγματεύσεων και των αποφάσεων που έχουν παρθεί στο παρελθόν. Αυτό γίνεται μέσω της μεθόδου search που είναι τύπου Comparable και έχει αναφερθεί στην υλοποίηση. Η συγκεκριμένη μέθοδος κάνει διάφορες συγκρίσεις καθώς διασχίζεται το δένδρο και όταν φτάσει σε κάποιο τερματικό κόμβο ο οποίος είναι φύλλο του δένδρου εμφανίζει μία εικόνα. Την επόμενη φορά όμως που θα επιχειρήσει να διασχίσει το δένδρο, θα ξεκινήσει από τον κόμβο που σταμάτησε τον οποίο θα τον έχει κάνει ρίζα. Με τον τρόπο αυτό, ο πράκτορας συναισθημάτων καταγράφει το ιστορικό των διαπραγματεύσεων και έχει τη δυνατότητα να βελτιώσει το μοντέλο των αποφάσεων του.

Στο τελικό στάδιο του μοντέλου διαπραγμάτευσης, αφού έχει καταλήξει ο πράκτοράς μας σε ένα συμπέρασμα, πρέπει να εμφανίσει και την αντίστοιχη εικόνα που συνδέεται με αυτή την

απόφαση. Η διαδικασία που ακολουθείται για να εμφανιστεί η κατάλληλη εικόνα θα περιγραφεί παρακάτω. Στο εσωτερικό του πράκτορα υπάρχει μία βάση με διάφορα συναισθήματα (database with emotions) τα οποία αποτελούν και τις πιθανές εξόδους του μοντέλου έκφρασης. Με τη βοήθεια μίας μεθόδου αντιστοίχισης (map method), τα συναισθήματα αναλύονται και αντιστοιχίζονται σε εκφράσεις του προσώπου (facial expressions). Έτσι, ο πράκτορας συναισθημάτων μετά από όλα τα στάδια διαπραγμάτευσης που έχει περάσει, εμφανίζει την κατάλληλη έκφραση του προσώπου μέσα από μία εικόνα, που είναι ουσιαστικά και η επιθυμητή έξοδος του μοντέλου.

Τέλος, αν θέλαμε να περιγράψουμε τον πράκτορά μας και τον τρόπο λειτουργίας του με μία πρόταση, θα λέγαμε ότι είναι ένας πράκτορας διασύνδεσης με συναισθήματα, το μοντέλο του οποίου βασίζεται σε πρωτόκολλα διαπραγμάτευσης πρακτόρων και στην ιστορία της επικοινωνίας τους.

Έχοντας εξηγήσει παραπάνω τα 5 επίπεδα του δυαδικού μας δένδρου, τα κριτήρια που λαμβάνουμε ως παραμέτρους και τον τρόπο με τον οποίο διασχίζουμε τους κόμβους του μέχρι να φτάσουμε σε ένα φύλλο του δένδρου και να εκφράσουμε ένα συναίσθημα μέσω της αντίστοιχης εικόνας, θεωρούμε απαραίτητο να περιγράψουμε και τις 9 διαδρομές που υλοποιήθηκαν στην εικόνα 24 για να γίνει πιο κατανοητός ο τρόπος λειτουργίας του πράκτορά μας.

1. Διαδρομή 1η

- Τύπος Καθηγητή (Teacher Type): Καθηγητής ή Αναπληρωτής Καθηγητής
- Περιορισμοί (Conflicts): Όχι
- Φάση (Phase): Πρώτη
- Συναίσθημα (emotion): Ευτυχισμένος
- Εικόνα που εμφανίζεται (image):



2. Διαδρομή 2η

- Τύπος Καθηγητή (Teacher Type): Καθηγητής ή Αναπληρωτής Καθηγητής
- Περιορισμοί (Conflicts): Ναι
- Αριθμός Περιορισμών (Conflicts Number): 1 ή 2
- Είδος Περιορισμών (Conflicts Type): Αδυναμία (Δεν μπορώ)
- Φάση (Phase): Πρώτη
- Συναίσθημα (emotion): Προβληματισμένος
- Εικόνα που εμφανίζεται (image):



3. Διαδρομή 3η

- Τύπος Καθηγητή (Teacher Type): Καθηγητής ή Αναπληρωτής Καθηγητής
- Περιορισμοί (Conflicts): Ναι
- Αριθμός Περιορισμών (Conflicts Number): 1 ή 2
- Είδος Περιορισμών (Conflicts Type): Αδυναμία (Δεν μπορώ)
- Φάση (Phase): Δεύτερη
- Συναίσθημα (emotion): Θυμωμένος
- Εικόνα που εμφανίζεται (image):



4. Διαδρομή 4η

- Τύπος Καθηγητή (Teacher Type): Καθηγητής ή Αναπληρωτής Καθηγητής
- Περιορισμοί (Conflicts): Ναι
- Αριθμός Περιορισμών (Conflicts Number): 1 ή 2
- Είδος Περιορισμών (Conflicts Type): Επιθυμία (Επιθυμώ/Δεν Επιθυμώ)
- Φάση (Phase): Πρώτη
- Συναίσθημα (emotion): Χαρούμενος
- Εικόνα που εμφανίζεται (image):



5. Διαδρομή 5η

- Τύπος Καθηγητή (Teacher Type): Καθηγητής ή Αναπληρωτής Καθηγητής
- Περιορισμοί (Conflicts): Ναι
- Αριθμός Περιορισμών (Conflicts Number): 1 ή 2
- Είδος Περιορισμών (Conflicts Type): Επιθυμία (Επιθυμώ/Δεν Επιθυμώ)
- Φάση (Phase): Δεύτερη
- Συναίσθημα (emotion): Θυμωμένος
- Εικόνα που εμφανίζεται (image):



6. Διαδρομή 6η

- Τύπος Καθηγητή (Teacher Type): Καθηγητής ή Αναπληρωτής Καθηγητής
- Περιορισμοί (Conflicts): Ναι
- Αριθμός Περιορισμών (Conflicts Number): >3
- Είδος Περιορισμών (Conflicts Type): Αδυναμία (Δεν μπορώ)
- Φάση (Phase): Πρώτη
- Συναίσθημα (emotion): Αγανακτισμένος
- Εικόνα που εμφανίζεται (image):



7. Διαδρομή 7η

- Τύπος Καθηγητή (Teacher Type): Καθηγητής ή Αναπληρωτής Καθηγητής
- Περιορισμοί (Conflicts): Ναι
- Αριθμός Περιορισμών (Conflicts Number): >3
- Είδος Περιορισμών (Conflicts Type): Αδυναμία (Δεν μπορώ)
- Φάση (Phase): Δεύτερη
- Συναίσθημα (emotion): Εξοργισμένος
- Εικόνα που εμφανίζεται (image):



8. Διαδρομή 8η

- Τύπος Καθηγητή (Teacher Type): Καθηγητής ή Αναπληρωτής Καθηγητής
- Περιορισμοί (Conflicts): Ναι
- Αριθμός Περιορισμών (Conflicts Number): >3
- Είδος Περιορισμών (Conflicts Type): Επιθυμία (Επιθυμώ/Δεν Επιθυμώ)
- Φάση (Phase): Πρώτη
- Συναίσθημα (emotion): Ήρεμος
- Εικόνα που εμφανίζεται (image):



9. Διαδρομή 9η

- Τύπος Καθηγητή (Teacher Type): Καθηγητής ή Αναπληρωτής Καθηγητής
- Περιορισμοί (Conflicts): Ναι
- Αριθμός Περιορισμών (Conflicts Number): >3
- Είδος Περιορισμών (Conflicts Type): Επιθυμία (Επιθυμώ/Δεν Επιθυμώ)
- Φάση (Phase): Δεύτερη
- Συναίσθημα (emotion): Συγχυσμένος
- Εικόνα που εμφανίζεται (image):



5 Συμπεράσματα – Μελλοντικές επεκτάσεις

Σε αυτή τη διπλωματική εργασία, προτείνεται ένα νέο μοντέλο Συναισθηματικού Πράκτορα Διασύνδεσης. Κύριος στόχος του μοντέλου ήταν να εισάγει τα συναισθήματα στη διαδικασία λήψης αποφάσεων από τους πράκτορες, έτσι ώστε ένας πράκτορας να μπορεί να παρουσιάζει ανθρωπομορφικά χαρακτηριστικά συμπεριφοράς. Επίσης, προσφέρεται μία νέα υπολογιστική μέθοδος η οποία βασίζεται στην αξιολόγηση των γεγονότων και των δράσεων. Όσον αφορά τους στόχους και τα πρότυπα του πράκτορα, οι υπολογιστικοί κανόνες χρησιμοποιήθηκαν για να χαρτογραφήσουν την επιρροή των γεγονότων και των δράσεων στις συναισθηματικές καταστάσεις. Αυτή η μέθοδος είναι ευέλικτη και προσαρμόσιμη σε διάφορα περιβάλλοντα και μπορεί να χρησιμοποιηθεί ως ξεχωριστή μονάδα σε προηγμένες εφαρμογές πολυπρακτορικών συστημάτων. Οι πηγές της αρχιτεκτονικής είναι διαθέσιμες και οι χρήστες μπορούν να προσαρμόσουν τις παραμέτρους στις ιδιαίτερες ανάγκες τους. Το γενικό μοντέλο συναισθημάτων υλοποιήθηκε στον τομέα της μάθησης για να δείξει προσαρμοστικότητα.

Τα αποτελέσματα της εφαρμογής φαίνονται να είναι τουλάχιστον ελπιδοφόρα. Έχουμε φτάσει στον πρώτο μας στόχο που ήταν να παρατηρήσουμε διαφορετικές συμπεριφορές σε διάφορες ενέργειες των πρακτόρων. Για παράδειγμα, ένας νευρικός και αναστατωμένος πράκτορας θα πακετάρει και μη επιτρεπόμενα πακέτα, ενώ ένας ήρεμος και συγκεντρωμένος πράκτορας θα τα «πετάξει» στον κάδο. Επιπλέον, παρατηρήσαμε σε ευρύτερη κλίμακα του συνολικού συστήματος, ότι υπάρχουν διαφορές τόσο στην αποδοτικότητα της δουλειάς όσο και στη ψυχολογική ευημερία των πρακτόρων και αυτό έχει άμεση σχέση με τη σύνθεση των ομάδων τους. Η προσομοίωση έχει οδηγήσει ήδη σε τροποποιήσεις που ευνοούν την ολοκλήρωση των εργασιών σε ένα ιδιαίτερα αγχωτικό περιβάλλον.

Σε μελλοντική εργασία μας, έχουμε ως στόχο να εκτελεστούν πειράματα σε περισσότερο περίπλοκες καταστάσεις τα οποία θα περιλαμβάνουν την προσωπικότητα καθώς και τα κίνητρα και τη διάθεση για υπολογισμούς. Η έλλειψη σημαντικών στοιχείων, η παρανόηση από τους χρήστες σχετικά με τους ορισμούς των συναισθημάτων καθώς και η άγνοια των χρηστών σχετικά με τα συναισθήματά τους, είναι μερικά από τα εμπόδια στη δουλειά μας. Η συγκεκριμένη αρχιτεκτονική μπορεί να ελεγχθεί με περισσότερα δεδομένα και πιο ακριβείς μεθόδους στο μέλλον. Η εύρεση μίας λειτουργίας επίτευξης του κατάλληλου στόχου για τους κόμβους στην ιεραρχία των στόχων είναι ένα ζήτημα σε κάθε πρόγραμμα που χρησιμοποιεί αυτή την αρχιτεκτονική. Η εύρεση μίας μεθόδου που να μπορεί να επιλέγει την κατάλληλη λειτουργία από ένα γενικό σύνολο είναι ένα ακόμη μελλοντικό σχέδιο. Στην

παρούσα διπλωματική εργασία, επικεντρωθήκαμε μόνο στα 8 από τα 16 συναισθήματα του OCC μοντέλου. Χρησιμοποιήσαμε μία σειρά από το group των συναισθημάτων που βασίζονται στο στόχο και μία σειρά από το group των συναισθημάτων που βασίζονται στο πρότυπο. Τέλος, μελλοντικά σκοπεύουμε να εισάγουμε και να αξιολογήσουμε και τα υπόλοιπα συναισθήματα του προτύπου καθώς και να διερευνήσουμε τις επιδράσεις που μπορεί να έχει η προσωπικότητα και η ηθική στη λήψη αποφάσεων.

Το πρώτο πράγμα που οραματιζόμαστε είναι να πάρει περισσότερη «ψυχολογική» αξιολόγηση από εμπειρογνώμονες η εφαρμογή και στη συνέχεια να προστεθούν νέες πιθανές αλληλεπιδράσεις μεταξύ των πρακτόρων και του περιβάλλοντος. Η κύρια δυσκολία εδώ είναι ότι ο αριθμός των συναισθηματικών κανόνων για τη συλλογή αυξάνεται με τον αριθμό των πιθανών ενεργειών των πρακτόρων. Αυτό συμβαίνει γιατί κάθε δράση πρέπει να έχει τους δικούς της κανόνες ενεργοποίησης. Σε αυτό το σημείο θα μπορούσαμε να οραματιστούμε δύο πιθανές λύσεις: α) τον προσδιορισμό κάποιων μετά-κανόνων για τις μετά-τάξεις των δράσεων ή β) η δημιουργία ενός λιγότερου ντετερμινιστικού συστήματος που δεν χρειάζεται αυτού του είδους τους κανόνες. Σε όλες τις περιπτώσεις είναι απαραίτητο να γίνει μία προσομοίωση πιο εξαντλητική και κατά συνέπεια πιο αποτελεσματική και αξιόπιστη για αυτούς που θα αποφασίσουν. Θα πρέπει να συνεργαστούμε μαζί τους για να αξιολογήσουμε και να ικανοποιήσουμε τις νέες τους απαιτήσεις, που θα εμφανιστούν μαζί με μία βελτίωση της προσομοίωσης.

Η τεχνολογία των ευφυών πρακτόρων βρίσκεται σε ένα ενδιαφέρον σημείο όσον αφορά την ανάπτυξή της. Η εμπορική δύναμη των εφαρμογών στους πράκτορες αναπτύσσεται όλο και περισσότερο και σε διαφορετικούς τομείς όπως η μετεωρολογία, οι κατασκευαστικές εταιρείες, τα παιχνίδια στον υπολογιστή, η αξιολόγηση των ικανοτήτων και η διαχείριση των εργασιών. Επιπλέον, αρκετά περιβάλλοντα ανάπτυξης που υποστηρίζονται εμπορικά είναι διαθέσιμα όπως και μεθοδολογίες σχεδιασμού, αρχιτεκτονικές αναφορές και πρότυπα αρχίζουν να εμφανίζονται. Αυτά όλα είναι ισχυρές ενδείξεις μίας ώριμης τεχνολογίας. Ωστόσο, η ανάπτυξη αυτής της τεχνολογίας δεν είναι τόσο ταχεία ή τόσο διάχυτη όπως περίμεναν οι υποστηρικτές της. Ο βασικός στόχος είναι να γίνει σε κάποια φάση η τεχνολογία των ευφυών πρακτόρων πρότυπο επιλογής για την ανάπτυξη των σύνθετων καταναμημένων συστημάτων και φυσική εξέλιξη του αντικειμενοστραφούς προγραμματισμού. Υπάρχουν δύο ερωτήματα τα οποία ίσως πρέπει να απαντηθούν για να μπορέσει η συγκεκριμένη τεχνολογία να αναπτυχθεί περισσότερο:

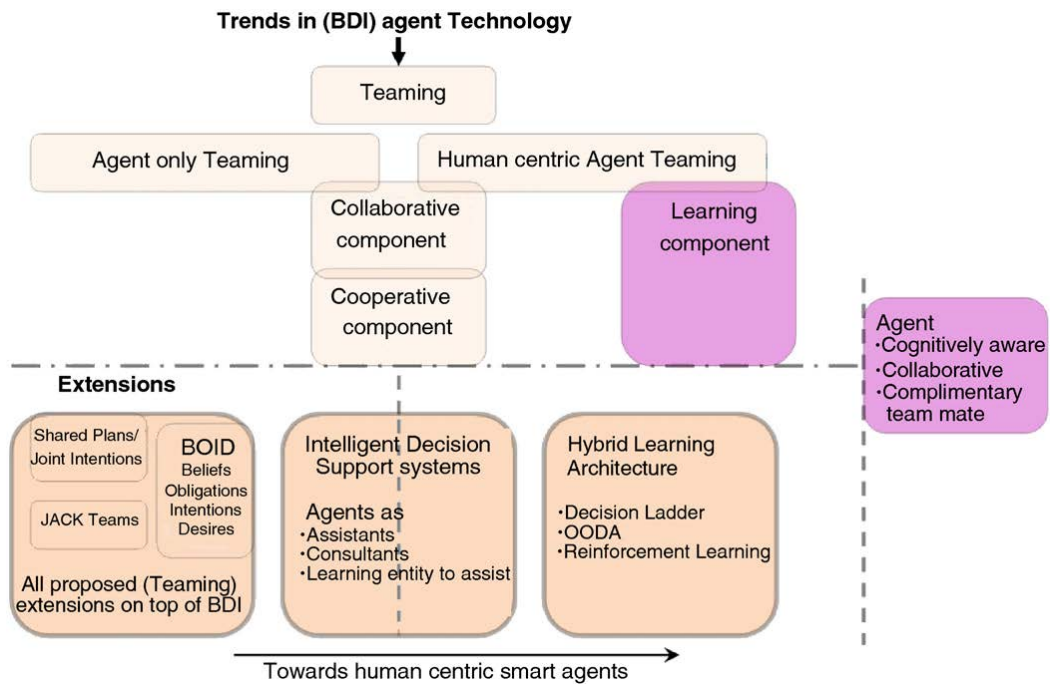
- Είναι θεμελιώδεις οι λόγοι για τους οποίους μία τέτοια τεχνολογία που υπόσχεται τόσα πολλά δεν έχει προσφέρει τα αναμενόμενα;
- Είναι πραγματικά τόσο αναγκαία όσο φαίνεται και τι επιφυλάσσει το μέλλον για αυτήν την τεχνολογία;

Από την οπτική γωνία ενός μηχανικού λογισμικού, θα προσδοκούσε να αποκτήσει σημαντικά οφέλη από την τεχνολογία των ευφυών πρακτόρων μέσω της ανάπτυξής της σε πολύπλοκες καταναμημένες εφαρμογές όπως η διαχείριση της εικονικής επιχείρησης και η διαχείριση των δικτύων αισθητήρων. Ωστόσο, ενώ το πρότυπο του πράκτορα προσφέρει την υπόσχεση να παρέχει ένα καλύτερο πλαίσιο για τη σύλληψη εννοιών και την εφαρμογή αυτών των τύπων του συστήματος, πρέπει να υποστηριχθεί με νέες ιδέες, μεθοδολογίες σχεδιασμού και αρχιτεκτονικές αναφορές έτσι ώστε να μπορούν αναπτυχθούν αποτελεσματικά οι σχετικές τεχνολογίες. Το θέμα δεν είναι όλα αυτά απλά να εμφανιστούν αλλά η κοινότητα λογισμικού να εκπαιδεύεται συνεχώς για να αποκτήσει περισσότερη εμπειρία όσον αφορά τη χρήση τους. Δεδομένου της φύσης αυτών των εφαρμογών, αναμένεται να δούμε μία σταδιακή μετάβαση από το αντικειμενοστραφές στο πρότυπο που βασίζεται στους πράκτορες καθώς το πλαίσιο υποστήριξης ωριμάζει. Οι αρχικές θεωρίες της αρχιτεκτονικής των πρακτόρων χρονολογούνται κοντά στη δεκαετία του 1980 και αυτό δείχνει ότι η μακροζωία είναι βασικό θεμέλιο για την εξέλιξη μίας υγιούς και επεκτάσιμης τεχνολογίας.

Αν και είναι φανερό ότι οι υπάρχουσες θεωρίες είναι αρκετά ευέλικτες για να φιλοξενήσουν νέες γνωστικές έννοιες, γίνεται αντιληπτό ότι υπάρχει η ανάγκη να αναπτυχθούν νέα μοντέλα συλλογισμού. Στην περίπτωση της εφαρμογής JACK του BDI, ένα νέο ανεπτυγμένο μοντέλο συλλογιστικής ομάδας αναμένεται να είναι διαθέσιμο σε σύντομο χρονικό διάστημα πέρα από το αρχικό. Αυτό το μοντέλο έχει τη δυνατότητα να επηρεάζει την μνήμη του προηγούμενου με γνωστικό και ρεαλιστικό τρόπο μέσω εξωτερικών εποπτών συμπεριφοράς όπως η καφεΐνη ή η κόπωση. Έτσι, υπάρχει η δυνατότητα μέσω της παροχής μοντέλων συλλογισμού "υψηλότερου επιπέδου" που θα ενσωματωθούν στις μεθοδολογίες σχεδιασμού, να επηρεάσουν σημαντικά την παραγωγικότητα και επομένως να διεισδύσουν στην αγορά.

Η ανάπτυξη των εφαρμογών των ευφυών πρακτόρων χρησιμοποιώντας τις τρέχουσες γενιές πρακτόρων δεν είναι ακόμα ρουτίνα. Για να αποφευχθεί κάτι τέτοιο, πρέπει να χρησιμοποιηθεί ένα πιο διαισθητικό μοντέλο λογικής και καλύτερα πλαίσια στήριξης. Το κυριότερο εμπόδιο βέβαια για την ανάπτυξη αυτών των εφαρμογών παραμένει η συμπεριφορά απόκτησης και αυτό προσπαθούν οι υπεύθυνοι λογισμικού να βελτιώσουν. Το διακριτικό χαρακτηριστικό του μοντέλου που έχει προταθεί είναι ότι ο έξυπνος πράκτορας θα έχει τη δυνατότητα να καθορίσει πώς πρέπει να ανταποκριθεί στα διάφορα αιτήματα για τις παρεχόμενες υπηρεσίες. Ανάλογα με την εφαρμογή, αποκτώντας τις συμπεριφορές που είναι απαραίτητες για να επιτευχθεί ο απαιτούμενος βαθμός της αυτόνομης λειτουργίας, θα μπορούσε να είναι ένα ακόμη στάδιο βελτίωσης. Εν ολίγοις, αναμένεται ότι οι ευφυείς πράκτορες θα διατηρήσουν τα αρχιτεκτονικά τους θεμέλια αλλά οι πιο κατάλληλες και καλύτερα σχεδιασμένες μεθοδολογίες θα χρησιμοποιούνται όλο και περισσότερο στην ανάπτυξη λογισμικού για πράκτορες με συναισθήματα. Επιπλέον, θα παρέχεται καλύτερη

υποστήριξη για τις ομάδες ανθρώπων-πρακτόρων έτσι ώστε να αναπτυχθεί μία νέα κατηγορία έξυπνων εφαρμογών υποστήριξης αποφάσεων.



Εικόνα 36 Μελλοντικές προεκτάσεις στους BDI Πράκτορες (Vidal, 2007)

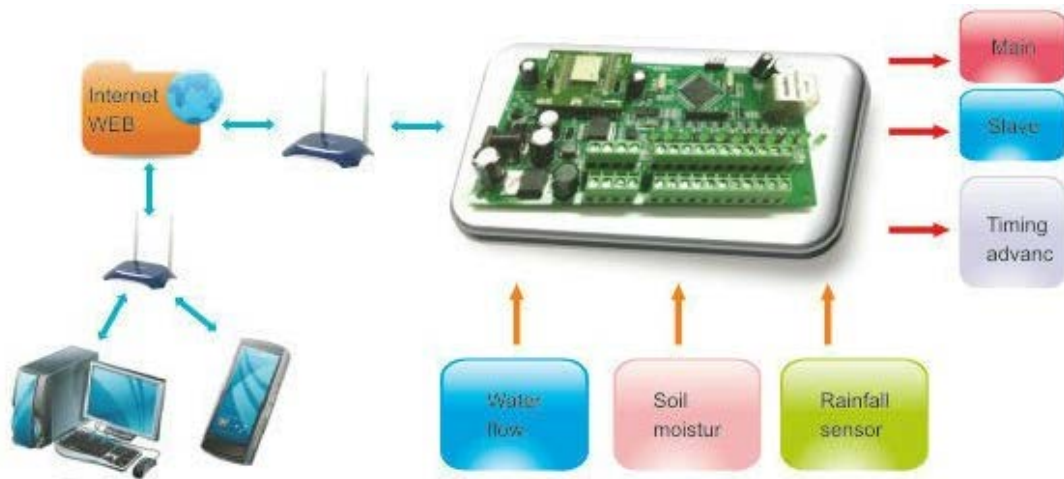
Τα χαρακτηριστικά φωνής και έκφρασης συναισθήματος του ανθρώπινου λόγου είναι ανιχνεύσιμα σε δεδομένα ήχου. Υπάρχει μία επείγουσα ανάγκη για αυτού του είδους τις πληροφορίες που προκύπτουν σε διάφορες πτυχές των τηλεφωνικών υπηρεσιών. Οι τηλεφωνικές εταιρείες πρέπει να αναπτύξουν μία ισχυρή αίσθηση εξυπηρέτησης πελατών με το πιο σημαντικό σημείο επαφής με τον πελάτη να είναι στον τομέα της διαχείρισης παραπόνων. Προκειμένου να βελτιωθεί η εμπειρία του πελάτη, δύο σημαντικά προβλήματα πρέπει να αντιμετωπιστούν, τα συναισθήματα του πελάτη που έχουν "καθυστερήσει" και είναι δύσκολο να ανιχνευθούν και η έλλειψη μίας αποτελεσματικής στρατηγικής επίλυσης προβλημάτων για περιπτώσεις πελατών με διαφορετικές συναισθηματικές αντιδράσεις. Το πρώτο πρόβλημα μπορεί να αντιμετωπιστεί με τεχνικές ανίχνευσης συναισθημάτων που λειτουργούν με δεδομένα ήχου. Προκειμένου να εντοπιστούν συναισθήματα όπως η χαρά, η εμπιστοσύνη, ο φόβος, η έκπληξη, η λύπη, ο ενθουσιασμός, η οργή, η αναμονή καθώς και διαφορετικές εκδηλώσεις του καθενός, τα μοντέλα ταξινόμησης είναι δομημένα πάνω σε ποικίλα χαρακτηριστικά γνωρίσματα που εξάγονται από δεδομένα ήχου και ανήκουν σε διαφορετικούς τύπους (Koolagudi, 2012).



Εικόνα 37 Αναγνώριση συναισθημάτων σε Κινητά τηλέφωνα

Ένας ακόμη λόγος που προβλέπεται να χρησιμοποιηθεί μελλοντικά η τεχνολογία των πρακτόρων είναι η ανίχνευση συναισθημάτων σε online έγγραφα, όπως blogs, φόρουμ συζητήσεων, κοινωνικά δίκτυα και ειδήσεις που αναφέρουν γεγονότα έκτακτης ανάγκης. Δύο τύποι πρακτόρων θα αναπτυχθούν, αυτός που θα "θερίζει" τα δεδομένα και θα βρίσκει αυτά που θέλει και αυτός που στη συνέχεια θα τα αναλύει. Οι πρώτοι θα έχουν το ρόλο των αντιολισθητικών αλυσίδων στο διαδίκτυο. Θα σαρώνουν το διαδίκτυο και θα αναζητούν έγγραφα που πληρούν τα κριτήρια που έχουμε δώσει και στη συνέχεια θα τα αποθηκεύει σε μία τοπική βάση δεδομένων. Οι πράκτορες-αναλυτές θα αναλύουν τα αποτελέσματα σε κατανεμημένο περιβάλλον. Η λειτουργία τους θα πρέπει να βασίζεται στις τρέχουσες προσεγγίσεις και στους αλγορίθμους ανίχνευσης συναισθημάτων. Η διαδικασία αυτή θα γίνεται σε ένα ολοκληρωμένο δίκτυο υπολογιστών για να επιτευχθεί καλύτερη απόδοση [Mitrovic, 2014]. Αρχικά, η έμφαση θα δοθεί στα αγγλικά έγγραφα με στόχο την επέκταση σε άλλες μεγάλες ευρωπαϊκές γλώσσες όπως Γερμανικά, Γαλλικά, Ιταλικά. Για το σκοπό αυτό θα χρησιμοποιηθούν αρκετά ελεύθερα διαθέσιμα λεξικά συναισθημάτων.

Τέλος, ελπίζουμε σε κάποια φάση, να ενσωματωθεί η πλατφόρμα δικτύου που βασίζεται στην peer-to-peer τεχνολογία σε ένα σύστημα πλέγματος (grid system). Είναι κάτι που θα αποτελέσει επαναστατικό αντίκτυπο για το μέλλον της κοινωνίας μας.



Εικόνα 38 Μελλοντικές προεκτάσεις σε Πολυπρακτορικά συστήματα

Βιβλιογραφία

Βλαχάβας, Ι., Κεφαλάς Π., Βασιλειάδης Ν., Κόκκορας Φ. και Σακελλαρίου Η. (2011). Τεχνητή Νοημοσύνη - Γ' Έκδοση, ISBN: 978-960-8396-64-7, Έκδοση/Διάθεση: [Εκδόσεις Πανεπιστημίου Μακεδονίας](#).

Πεπές, Σ., (2015). Ευφυείς Πράκτορες ως Διαπραγματευτές για την κατάσταση Προγράμματος Ωρών και Μαθημάτων, Διπλωματική Εργασία, Πρόγραμμα Μεταπτυχιακών Σπουδών «Ευφυείς Τεχνολογίες Διαδικτύου», ΑΤΕΙΘ

Aarts, E.H.L., Korst, J., and Michiels, W. (2005). Simulated annealing. E.K. Burke and G. Kendall (eds.). *Introductory Tutorials in Optimisation, Decision Support and Search Methodology*, Springer, vol. 7, p.p. 187-211.

Aboulenien, H. A., De Wilde, P. (1998). A Simple Interface Agent, *Proceedings of the Joint Conference on Information Sciences*, vol. 3, pp. 190–192.

Ammar, M. B., Neji, M., Alimi, A.M., Gouardères, G. (2007). Emotional agents for collaborative e-learning, *International Research Journal on Digital Future. (FormaMente)*, vol. 2, n. 3-4, pp. 33-64.

Aube, M., Senteni, A. (1996). *What are emotions for*, The MIT Press, pp 264-271.

Aznar, F., Sempere, M., Pujol, M., Rizo, R. (2007). On intelligent interface agents for human based computation, *Intelligent Data Engineering and Automated Learning-IDEAL*, Springer, pp. 930–939.

Bates, J. (1994). The role of emotion in believable agents, *Communications of the ACM*, issue 37, vol. 7, pp 122-125.

Bates, J. Loyall, A. B., & Reilly, W. S. (1992). An architecture for action, emotion, and social behavior (Technical Report No. CMU-CS-92-144). School of Computer Science, Pittsburgh, PA: Carnegie Mellon University.

Brooks, R.A. (2001). Intelligence Without Reason, *Proceedings of the 1991 International Joint Conference on Artificial Intelligence*, pp.569-695.

Burghouts, G. J., Heylen, D., Poel, M., op den Akker, R., & Nijholt, A. (2003, May). An Action Selection Architecture for an Emotional Agent. In *FLAIRS Conference*, pp. 293-297.

Burke, E.K., Marecek, J., Parkes, A.J., Rudová, H. (2010). A supernodal formulation of vertex colouring with applications in course timetabling, *Ann Oper Res*, vol. 179, pp. 105-130.

Calhoun, C., Solomon, R.C. (1984). *What is an emotion?*, Oxford University Press.

Camurri, A. (1997). Toward Kansei Information Processing in Music/Dance Interactive Multimodal Environments, Proceedings. Italian Assoc. for Musical Informatics (AIMI) Intl. Workshop Kansei: The Technology of Emotion, AIMI and DIST (University of Genoa), Genoa, Italy, pp. 74-78.

Camurri, A., Ferrentino, P. (1999). Interactive Environments for Music and Multimedia, *Multimedia Systems* (special issue on audio and multimedia), ACM and Springer-Verlag, Berlin.

Cohen, W. W., Borgida, A., & Hirsh, H. (1992, July). Computing least common subsumers in description logics. *AAAI*, vol. 1992, pp. 754-760.

Daviet, S., Desmier, H., Briand, H., Guillet, F., Philippe, V. (2005). A system of emotional agents for decision-support, *IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pp. 711–717.

Dybkjær, L., Bernsen, N.O. (2001). Usability evaluation in spoken language dialogue systems, Proceedings of the workshop on Evaluation for Language and Dialogue Systems (Annual Meeting of the ACL), vol. 9, pp. 1-10.

El-Nasr, M. S., Yen, J., & Ioerger, T. R. (2000). Flame—fuzzy logic adaptive model of emotions. *Autonomous Agents and Multi-agent systems*, issue 3, vol. 3, pp. 219-257.

Elliot, C., (1992). The affective reasoner: A process model of emotions in a multi-agent system, PhD thesis, Institute for the Learning Sciences, Evanston, IL: Northwestern University.

Failenschmid, K., Williams, D., Dybkjær, L., Bernsen, N.O. (1999). Draft proposal on best practice methods and procedures in human factors, DISC Deliverable D3.6.

Gibbon, D., Moore, R., Winski, R. (1997). *Handbook of Standards and Resources for Spoken Language Systems*, Mouton de Gruyter, Berlin, New York.

Gratch, J., & Marsella, S. (2004). A domain-independent framework for modeling emotion. *Cognitive Systems Research*, issue 5, vol. 4, pp. 269–306.

Gratch, J., & Marsella, S. (2003). Fight the way you train: The role and limits of emotions in training for combat. *Brown Journal of World Affairs*, issue 10, vol. 1, pp. 63–76.

-
- Green, S., Hurst, L., Nangle, B., & Cunningham, P. (1997). *Software agents: A review*. Trinity College Dublin, Department of Computer Science.
- Griffiths, P. E. (1997). *What emotions really are: The problem of psychological categories*, Chicago: University of Chicago Press, pp. 114.
- Hirschman, L., & Thompson, H. S. (1997). Overview of evaluation in speech and natural language processing.
- Jiang, H., Vidal, J. M., & Huhns, M. N. (2007, May). EBDI: an architecture for emotional agents, *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pp. 11.
- Kazemifard, M., Ghasem-Aghaee, N., & Ören, T. I. (2011). Design and implementation of GEmA: A generic emotional agent, *Expert systems with applications*, issue 38, vol. 3, pp. 2640-2652.
- Kobsa, A. (1993). User modeling: Recent work, prospects and hazards, *Human Factors in Information Technology*, issue 10, pp. 111-111.
- Koolagudi, S. G., & Rao, K. S. (2012). Emotion recognition from speech: a review. *International journal of speech technology*, issue 15, vol. 2, pp. 99-117.
- Koutsabasis, P., Darzentas, J. S., Spyrou, T., & Darzentas, J. (1999). Modeling Agents for Providing User Interface Assistance: The Design of the GAIA Interaction Agent. *Computers and Communications Journal*.
- Leeuwen, D. V. (1998). EAGLES recommendations on products and service evaluation, Handbook of Standards and Resources for Spoken Language Systems - Supplement Volume, (<http://www.spectrum.uni-bielefeld.de/EAGLES/SLWG/archive/supplement/chcots.pdf>).
- Maes, P. (1995). Artificial life meets entertainment: lifelike autonomous agents. *Communications of the ACM*, issue 38, vol. 11, pp. 108-114.
- Maes, P. (1994). Agents that reduce work and information overload. *Communications of the ACM*, issue 37, vol. 7, pp. 30-40.
- Maes, P., & Kozierok, R. (1993, July). Learning interface agents. In *AAAI* , vol. 93, pp. 459-465.
- Malim, M. R., Khader, A. T., & Mustafa, A. (2005). University Course Timetabling: A General Model.
- Maria, K. A., & Zitar, R. A. (2007). Emotional agents: A modeling and an application. *Information and Software Technology*, issue 49, vol. 7, pp. 695-716.
- McCarthy, J. (1960). Recursive functions of symbolic expressions and their computation by machine, Part I. *Communications of the ACM*, issue 3, vol. 4, pp.184-195.

-
- McCarthy, J. (1984). Some expert systems need common sense. *Annals of the New York Academy of Sciences*, issue 426, vol. 1, pp. 129-137.
- McCarthy, J., & Hayes, P. J. (1969). Some philosophical problems from the standpoint of artificial intelligence. *Readings in artificial intelligence*, pp. 431-450.
- McCollum, B., & Ireland, N. (2006). University timetabling: Bridging the gap between research and practice. *E Burke, HR, ed.: PATAT*, pp. 15-35.
- Middleton, S. E. (2002). Interface agents: A review of the field. *arXiv preprint cs/0203012*.
- Minsky, M. (1988). *Society of mind*. Simon and Schuster.
- Mitchell, T.M., (1997). *Machine Learning*, McGraw-Hill.
- Mitrovic, D., Ivanović, M., & Geler, Z. (2014). Agent-based distributed computing for dynamic networks. *Information Technology And Control*, issue 43, vol. 1, pp. 88-97.
- Mladenic, D. (1999). Text-learning and related intelligent agents: a survey. *IEEE Intelligent Systems*, issue 4, pp. 44-54.
- Ben Ammar, M., & Neji, M. (2007). EMASPEL (emotional multi-agents system for peer to peer e-learning). In *The 1st international conference on information and communication technology and accessibility, Hammamet, Tunisia, April 12*, Vol. 14.
- Killsen, N. (1971). Problem-solving methods in artificial intelligence.
- Nilsson, N. J. (1984). *Shakey the Robot, SRI A.I. Center Technical Note 323, April*.
- Norman, D. A. (1994). How might people interact with agents. *Communications of the ACM*, issue 37, vol. 7, pp. 68-71.
- Nwana, H. S. (1996). Software agents: An overview. *The knowledge engineering review*, issue 11, vol. 03, pp. 205-244.
- Oatley, K., Jenkins, J. (2001). *Understanding Emotions*, Oxford: Blackwell.
- Oprea, M. (2007). MAS_UP-UCT: A multi-agent system for university course timetable scheduling. *International Journal of Computers, Communications & Control*, issue 2, vol. 1, pp. 94-102.
- Payne, T. R., & Edwards, P. (1997). Interface agents that learn an investigation of learning issues in a mail agent interface. *Applied artificial intelligence*, issue 11, vol. 1, pp. 1-32.
- W. Reilly, J. Bates, (2002), Building emotional agents. Technical report, CMU-CS-92-143.
- Resnick, P., & Varian, H. R. (1997). Recommender systems. *Communications of the ACM*, issue 40, vol. 3, pp. 56-58.

-
- Rich, E. (1979). User modeling via stereotypes*. *Cognitive science*, issue 3, vol. 4, pp. 329-354.
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). *Learning internal representations by error propagation* (No. ICS-8506). CALIFORNIA UNIV SAN DIEGO LA JOLLA INST FOR COGNITIVE SCIENCE.
- Schiaffino, S., & Amandi, A. (2006). Personalizing user-agent interaction. *Knowledge-Based Systems*, issue 19, vol. 1, pp. 43-49.
- Schleiffer, R. (2002). *A generalized intelligent agent model and its application in the domain of transportation research*. Deutsches Zentrum für Luft-und Raumfahrt.
- Seo, Y. W., & Zhang, B. T. (2000, January). A reinforcement learning agent for personalized information filtering. In *Proceedings of the 5th international conference on Intelligent user interfaces* (pp. 248-251).
- Shakshuk, E., Müldner, T., & Haughn, B. (2004). ATIA: algorithm teaching interface agent. In *Information Technology Based Higher Education and Training, 2004. ITHET 2004. Proceedings of the Fifth International Conference on* (pp. 539-544).
- Shneiderman, B., & Maes, P. (1997). Direct manipulation vs. interface agents. *interactions*, issue 4, vol. 6, pp. 42-61.
- Sloman, A. (1987). Motives, mechanisms, and emotions. *Cognition and Emotion*, issue 1, vol. 3, pp. 217-233.
- Luo, J. (2012). *Affective computing and intelligent interaction*, Springer Science & Business Media, vol. 137.
- Trunk, G. R. (1998), Computer models of emotions and their meaning for emotion-psychological research, November.
- Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, pp. 433-460.
- Walker, M. A., Litman, D. J., Kamm, C. A., & Abella, A. (1997, July). PARADISE: A framework for evaluating spoken dialogue agents. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, pp. 271-280
- Wooldridge, M., & Jennings, N. R. (1995). Intelligent agents: Theory and practice. *The knowledge engineering review*, issue 10, vol. 02, pp. 115-152.
- Wooldridge, M., & Jennings, N. R. (1996). Special issue on intelligent agents and multi-agent systems. *Applied Artificial Intelligence Journal*, issue 9, vol. 4, pp. 74-86.

Αναφορές για εικόνες:

<http://www.intechopen.com/source/html/6704/media/image2.png> , Ανακτήθηκε 18 Μαΐου 2015.

<http://image.slidesharecdn.com/affectivecomputing-130701032350-phpapp01/95/affective-computing-4-638.jpg?cb=1372649126>, Ανακτήθηκε 18 Μαΐου 2015.

http://comps.canstockphoto.com/can-stock-photo_csp11749697.jpg , Ανακτήθηκε 18 Μαΐου 2015.

https://encrypted-tbn2.gstatic.com/images?q=tbn:ANd9GcSBvKVad_X2NV5rrLCHJhGjHKEafIgztn47Znx8i7xqO-q8Kh2jnQ , Ανακτήθηκε 18 Μαΐου 2015.

http://hz01.i.aliimg.com/img/pb/250/221/528/528221250_871.jpg , Ανακτήθηκε 18 Μαΐου 2015.

<http://blogs-images.forbes.com/amymorin/files/2014/07/emotionsforbes3.jpg> , Ανακτήθηκε 18 Μαΐου 2015.

<http://i100.independent.co.uk/image/25946-1ri0w8h.jpg> , Ανακτήθηκε 18 Μαΐου 2015.

<http://www.riec.tohoku.ac.jp/lab/kinoshita/map3.jpg> , Ανακτήθηκε 18 Μαΐου 2015.

http://en.wikipedia.org/wiki/Binary_search_tree

http://www.electromagnetics.unisalento.it/wp-content/uploads/2014/09/SA_ENG-300x273.png Ανακτήθηκε 20 Μαΐου 2015

MyWebFace (<http://download.mywebface.com>)

Animated GIF Maker (<http://ezgif.com/maker>)