

ΑΝΩΤΑΤΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΠΕΙΡΑΙΑ ΤΕΧΝΟΛΟΓΙΚΟΥ ΤΟΜΕΑ



**«ΑΝΑΠΤΥΞΗ ΣΜΗΝΟΥΣ ΒΟΤΣ ΓΙΑ ΤΗΝ ΑΝΙΧΝΕΥΣΗ ΣΤΟ ΔΙΑΔΙΚΤΥΟ
ΣΥΓΚΕΚΡΙΜΕΝΩΝ ΠΛΗΡΟΦΟΡΙΩΝ»**

**ΠΑΤΣΕΛΑΣ ΠΑΝΑΓΙΩΤΗΣ
ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΔΡΟΣΟΣ ΧΡΗΣΤΟΣ**



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΑΥΤΟΜΑΤΙΣΜΟΥ**

ΑΙΓΑΛΕΩ, ΜΑΡΤΙΟΣ 2017

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο / η κάτωθι υπογεγραμμένος / η Πατσέλας Παπαχιώτης
του Γεωργίου με αριθμό μητρώου 44596 φοιτητής / η του
Τμήματος **Μηχανικών Αυτοματισμού Τ.Ε.** του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβω την
εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του
συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και
πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται
αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη
αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα
του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος
φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα
του έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η
Συνέλευση του Τμήματος με νέα απόφαση της, μετά από αίτηση του ενδιαφερόμενου, του
αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα
καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός
ημερολογιακού 6μήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα
προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»

Ο Δηλών



Ημερομηνία

21/3/2017

Περιεχόμενα

Περίληψη.....	σελ.6
Εισαγωγή.....	σελ.7
ΚΕΦΑΛΑΙΟ 1: Η ΣΗΜΑΣΙΑ ΤΗΣ ΠΛΗΡΟΦΟΡΙΑΣ	σελ.7
1.1 Πληροφορία και πληροφοριακά συστήματα.....	σελ.8
1.2 Μέτρα προστασίας της πληροφορίας.....	σελ.10
1.3 Η επίδραση της τεχνολογίας.....	σελ.10
1.4 Το φαινόμενο της πληροφοριακής υπερφόρτωσης.....	σελ.12
1.5 Αντιμετώπιση της πληροφοριακής υπερφόρτωσης.....	σελ.12
ΚΕΦΑΛΑΙΟ 2: ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ.....	σελ.15
2.1 Ιστορική αναδρομή και εξέλιξη της τεχνητής νοημοσύνης.....	σελ.15
2.2 Ορισμός Τεχνητής Νοημοσύνη.....	σελ.17
2.3 Η δοκιμασία TURING.....	σελ.18
ΚΕΦΑΛΑΙΟ 3: ΕΥΦΥΕΙΣ ΠΡΑΚΤΟΡΕΣ.....	σελ.20
3.1 Ορισμός πράκτορα.....	σελ.20
3.2 Κατηγορίες ευφύων πρακτόρων.....	σελ.23
3.3 Ιδιότητες πρακτόρων.....	σελ.24
3.4 Τύποι περιβάλλοντος του πράκτορα.....	σελ.27
3.5 Αρχιτεκτονική του πράκτορα.....	σελ.28
3.6 Προβλήματα λειτουργίας των πρακτόρων.....	σελ.33
3.7 Πληροφοριακή υπερφόρτωση.....	σελ.34
3.8 Ανίχνευση συγκεκριμένης πληροφορίας.....	σελ.36
3.9 Στάση χρηστών απέναντι στους πράκτορες.....	σελ.38
ΚΕΦΑΛΑΙΟ 4: ΣΥΣΤΗΜΑ ΠΟΛΛΑΠΛΩΝ ΠΡΑΚΤΟΡΩΝ.....	σελ.39
4.1 Μετα-πράκτορες.....	σελ.39
4.2 Εισαγωγή στα πολυπρακτορικά συστήματα.....	σελ.40
4.3 Έννοια των πολυπρακτορικών συστημάτων.....	σελ.40
4.4 Δομικά στοιχεία ενός ΣΠΠ.....	σελ.42
4.5 Τρόπος επικοινωνίας των ευφύων πρακτόρων.....	σελ.43

4.6 Τύποι αλληλεπίδρασης ευφρών πρακτόρων.....σελ.46	σελ.46
4.7 Πρωτόκολλο επικοινωνίας CONTRACT NET.....σελ.48	σελ.48
4.8 Το πρωτόκολλο και γλώσσα επικοινωνίας KQML.....σελ.49	σελ.49
4.8.1 Τα χαρακτηριστικά της KQML και το περιβάλλον των πρακτόρων.....σελ.49	σελ.49
4.9 Η γλώσσα επικοινωνίας FIPA ACL.....σελ.53	σελ.53
4.9.1 Η ερμηνεία της FIPA ACL.....σελ.55	σελ.55
4.9.2 Διαφορές μεταξύ της KQML και της FIPA ACL.....σελ.55	σελ.55
ΚΕΦΑΛΑΙΟ 5: ΑΝΑΠΤΥΞΗ ΣΥΣΤΗΜΑΤΩΝ ΚΑΙ ΕΦΑΡΜΟΓΩΝ ΠΡΑΚΤΟΡΩΝ...σελ.56	σελ.56
5.1 Λογισμικό JADE.....σελ.56	σελ.56
5.2 Τα χαρακτηριστικά και η αρχιτεκτονική της JADE.....σελ.58	σελ.58
5.3 Ο πράκτορας ACC, AMS και ο DF.....σελ.61	σελ.61
5.4 Ο τρόπος επικοινωνίας των JADE πρακτόρων.....σελ.64	σελ.64
5.5 Οντολογίες και γλώσσες περιεχομένου στην γλώσσα JADE.....σελ.66	σελ.66
5.6 Κινητοί πράκτορες στην JADE.....σελ.68	σελ.68
Βιβλιογραφία.....σελ.69	σελ.69

Ευχαριστίες

Θα ήθελα να ευχαριστήσω αρχικά τον επιβλέποντα καθηγητή Κύριο Δρόσο για την ανάθεση της συγκεκριμένης πτυχιακής αλλά και για την πολύτιμη βοήθεια του και για την καθοδήγησή του κατά την διάρκεια εκπόνησης της.

Τέλος θα ήθελα να ευχαριστήσω θερμά τους γονείς μου και την αδερφή μου , οι οποίοι μου στήριξαν οικονομικά όλο αυτό το διάστημα αλλά και για την ηθική συμπαράσταση που μου παρείχαν.

ΠΕΡΙΛΗΨΗ

Στα πλαίσια της παρούσας πτυχιακής εργασίας που έχει ως σκοπό να μας δείξει το προβληματισμό που δημιουργείται στους χρήστες που ψάχνουν μια συγκεκριμένη πληροφορία στο διαδίκτυο. Αυτό συμβαίνει επειδή ο όγκος πληροφορίας στο διαδίκτυο αναπτύσσεται με μεγαλειώδης ρυθμούς. Έτσι αυτό έχει ως αποτέλεσμα να δυσκολεύει τον χρήστη να επιλέξει σωστά. Λύση σε αυτό το πρόβλημα δίνουν τα σμήνη bots ή αλλιώς όπως αναφέρονται ευφυείς πράκτορες. Ο ευφυής πράκτορας αποτελεί μια εφαρμογή η οποία ενεργεί εκ μέρους του ανθρώπου αλλά δεν επιλέγει. Λόγω της νοημοσύνης που διαθέτουν μπορούν να συλλέγουν πληροφορίες από το διαδίκτυο για τον χρήστη και να τις αποθηκεύουν και αν χρειαστεί να τις ανακτήσουν. Επίσης έχουμε τα πολυπρακτορικά συστήματα που αποτελούνται από ένα σύνολο πρακτόρων, οι οποίοι συνεργάζονται για την λύση ενός προβλήματος. Υπάρχουν δυο γλώσσες επικοινωνίας ανάμεσα πράκτορες, αρχικά την KQML και μετά την FIPA ACL. Το λογισμικό το οποίο χρησιμοποιείται για την ανάπτυξη συστημάτων ή εφαρμογών πολλαπλών πρακτόρων ονομάζεται JADE.

ΚΕΦΑΛΑΙΟ 1

Η ΣΗΜΑΣΙΑ ΤΗΣ ΠΛΗΡΟΦΟΡΙΑΣ

ΕΙΣΑΓΩΓΗ

Τα δύο κύρια προβλήματα τα οποία παρακωλύουν την αποτελεσματική χρήση του παγκόσμιου ιστού είναι οι ραγδαίοι ρυθμοί με τους οποίους αυτό αναπτύσσεται και ο υπέρογκος αριθμός πληροφοριών που το κατακλύζει καθημερινά. Αυτό έχει αρνητική επίδραση στον χρήστη του διαδικτύου, επειδή του δημιουργούνται αισθήματα άγχους και υπερέντασης για το ποια πληροφορία θα είναι η καλύτερη ως απάντηση σε αυτό το πράγμα που ψάχνει. Δηλαδή δεν μπορεί να διαλέξει ανάμεσα στις τόσες πληροφορίες που υπάρχουν στο διαδίκτυο, για το ποια είναι η καταλληλότερη. Προκειμένου να αποφύγουμε όλη αυτή την πληροφοριακή υπερφόρτωση υπάρχουν τα σμήνη bots ή αλλιώς όπως θα τα αναφέρουμε ευφυείς πράκτορες (intelligent agents). Τα οποία μας βοηθούν στην αναζήτηση πληροφορίας στο διαδίκτυο αλλά όπως είπαμε πιο πάνω, μας βοηθούν και στην καταπολέμηση της πληροφοριακής υπερφόρτωσης. Κάποιες άλλες ονομασίες με τις οποίες μπορούμε να τα βρούμε είναι: ρομποτ του διαδικτύου (web robots), νοήμονες πράκτορες. Το σύνολο πολλών πρακτόρων με το οποίο σχεδιάστηκαν και υλοποιήθηκαν να συνεργάζονται, να συντονίζονται για την επίλυση πολύπλοκων προβλημάτων ονομάζονται πολυπρακτορικά συστήματα (Multi-Agent Systems). Οι εφαρμογές με τις οποίες λειτουργούν οι ευφυείς πράκτορες ανήκουν στον κλάδο της τεχνητής νοημοσύνης (TN). Επειδή έχουν την ικανότητα να δρουν αυτόνομα, συλλέγοντας πληροφορίες, εκπαιδεύοντας τον εαυτό τους και επικοινωνώντας μεταξύ τους ή με τον χρήστη. Με την γλώσσα επικοινωνίας KQML, η οποία είναι γλώσσα υψηλού επιπέδου που βοηθάει στην ανταλλαγή πληροφοριών ανάμεσα στους ευφυείς πράκτορες. Η επόμενη πιο γνωστή γλώσσα επικοινωνίας των πρακτόρων ACL (Agent-Communication-Language) είναι η FIPA ACL. Η FIPA ACL έχει κάποια κοινά στοιχεία με την KQML αλλά και διαφορετικά. Τέλος η JADE (Java-Agent-Development Framework) είναι ένα λογισμικό που έχει ως σκοπό την ανάπτυξη συστημάτων αλλά και εφαρμογών πολλαπλών πρακτόρων σε σχέση με τα πρότυπα της FIPA. Η JADE θεωρείται μια υλοποιημένη εντελώς στην γλώσσα JAVA γιατί περιέχει κάποια πακέτα στην JAVA.

1.1 ΠΛΗΡΟΦΟΡΙΑ ΚΑΙ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ

Ως πληροφορία μπορούμε να ορίσουμε κάποια δεδομένα. Όπου ένα ίδιο δεδομένο στον ίδιο άνθρωπο σε διαφορετική χρονική στιγμή στην ζωή του υπάρχει περίπτωση να έχει διαφορετική σημασία[1]. Έτσι η σημασία του δεδομένου δεν αλλάζει μόνο από άνθρωπο σε άνθρωπο, αλλά και για τον ίδιο τον άνθρωπο μέσα στο χρονικό διάστημα. Μια πληροφορία μπορεί να έχει την μορφή ενός δεδομένου όπου έχει αποθηκευτεί με την προϋπόθεση να μπορεί στο μέλλον να αποκτήσει σημασία. Όπως επίσης και το νόημα που θα δώσει ο άνθρωπος στο δεδομένο, με βάση στους κανόνες που χρησιμοποιούνται για την σωστή αναπαράσταση[2]. Λαμβάνοντας υπόψη την Θεωρία Της Πληροφορίας, μας λέει ότι η πληροφορία δημιουργείται από τα δεδομένα. Για να μεταφερθεί η πληροφορία στο κανάλι επικοινωνίας, μεταφέρεται σε μορφή σήματος. Σε περίπτωση που καθυστερήσει μια πληροφορία, τότε από χρήσιμη πληροφορία μετατρέπεται σε άχρηστα δεδομένα[1]. Η αύξηση του χώρου αποθήκευσης των αποθηκευτικών μέσων έχει βοηθήσει πολύ τους χρήστες γιατί τώρα μπορούν να αποθηκεύσουν ότι θέλουν. Ενώ πριν αυξηθεί το μέγεθος των αποθηκευτικών μέσων έπρεπε να διαλέξουν τι θα χρειάζοταν σίγουρα. Αυτό βέβαια έχει δημιουργήσει μια δυσκολία στην κατανόηση για το ποια πληροφορία είναι χρήσιμη αλλά και ποια έχει δυσκολία στην ανάκτηση της[3].

Ως πληροφοριακό σύστημα μπορούμε να ορίσουμε ένα οργανωμένο σύνολο το οποίο αποτελείται από μονάδες λογισμικού, υλικού και τηλεπικοινωνιακού δικτύου[1]. Επιπλέον μεταφέρουν οργανωμένη και επεξεργασμένη μια πληροφορία όπου την αποθηκεύουν και την επεξεργάζονται ώστε να μπορούν να την προβάλουν σε πολλά είδη[3]. Τα πληροφοριακά συστήματα πέρα από το να παρέχουν πληροφορίες βοηθούν τους χρήστες στην δουλειά τους. Δηλαδή κάνουν πιο εύκολη την επικοινωνία και τον διαχωρισμό των πληροφοριών ανάμεσα τους και έχουν εργαλεία για τον χειρισμό των δεδομένων.

Τα τελευταία χρόνια λόγω της μεγάλης ανάπτυξης του διαδικτύου, έχει δημιουργηθεί ως ένας μεγάλος αποθηκευτικός χώρος ψηφιακής πληροφορίας. Όλοι οι άνθρωποι έχουν στραφεί στο διαδίκτυο προκειμένου να ικανοποιήσουν την ανάγκη της πληροφόρησης τους[4]. Σύμφωνα με τον Denett[5] ο ρυθμός με τον οποίο αναζητούν, συγκεντρώνουν, μοιράζονται και καταναλώνουν πληροφορίες οι χρήστες δεν μπορεί να καθοριστεί από άλλους οργανισμούς. Όπως αναφέραμε και προηγουμένως τα πληροφοριακά συστήματα κάνουν προτάσεις ενώ οι άνθρωποι παίρνουν τις αποφάσεις. Τα πληροφοριακά συστήματα ακολουθούν τους κανόνες οι οποίοι είναι προγραμματισμένοι με μηχανικό τρόπο. Αυτό

όμως δεν σημαίνει λήψη αποφάσεων , η λήψη αποφάσεων μέσω της δημιουργίας κανόνων από τους ανθρώπους ορίζεται ως λήψη αποφάσεων. Για να χαρακτηρίσουμε μια πληροφορία ως χρήσιμη θα πρέπει να είναι διαθέσιμη όταν χρειαστεί, κατανοητή, αξιόπιστη, σύγχρονη και ευέλικτη στον άνθρωπο.

Όταν ψάχνουμε για μια πληροφορία και την βρούμε δεν μπορεί να χαρακτηριστεί ως ποιοτική κιόλας. Όπως είπε ο Bentley[3] για να είναι μια πληροφορία ποιοτική θα πρέπει να είναι αξιόπιστη, σχετική και εύρωστη.

- **Αξιόπιστη Πληροφορία:** Για να είναι μια πληροφορία αξιόπιστη θα πρέπει να αφορά την τωρινή κατάσταση , ώστε να μπορεί να χρησιμοποιηθεί όταν χρειαστεί. Όπως επίσης να είναι ακριβής για τον τρόπο χρήσης της αλλά και επαληθεύσιμη για την ανάλυση δεδομένων.
- **Σχετική Πληροφορία:** Μια πληροφορία είναι σχετική όταν αυτός που την δέχεται έχει πιο καλή απόδοση με την χρήση της. Ο βαθμός πάνω στον οποία δρα η πληροφορία στον χρήστη χαρακτηρίζει πόση σημαντική είναι .
- **Εύρωστη Πληροφορία:** Για να είναι εύρωστη μια πληροφορία θα πρέπει να τα καταφέρει στην σφοδρότητα του χρόνου, την έλλειψη ανθρώπινης δύναμης και στην τυχόν αποτυχία του συστήματος.

Όπως αναφέραμε στην εύρωστη πληροφορία για το θέμα του χρόνου , εννοούμε στην ανταπόκριση που θα έχει για να κρατηθεί μετά το πέρασμα του χρόνου. Επειδή μετά από καιρό θεωρούνται παλιές αυτές οι πληροφορίες. Η έλλειψη ανθρώπινης δύναμης είναι ένας παράγοντας που μπορεί να έχει επίδραση μέσω των λαθών της πληροφορία στον χειρισμό δεδομένων. Τα μέτρα που λαμβάνονται για την καταπολέμηση τέτοιων λαθών γίνονται μέσω των τακτικών ελέγχων και επαληθεύσεων προκειμένου να μην έχει διαστρεβλωθεί καμία πληροφορία. Στην αποτυχία του συστήματος μπορεί να καταστραφεί μια πληροφορία έτσι θα πρέπει να προστατευθεί από την τυχόν αποτυχία ανάλογα με το περιεχόμενο της επικαιρότητας. Σύμφωνα με τον Peter Drucker[6], οι άνθρωποι κατά κύριο λόγο λαμβάνουν μια απόφαση χωρίς να διαθέτουν τις κατάλληλες και επαρκείς γνώσεις. Όταν θέλουμε να πάρουμε σωστές και σημαντικές αποφάσεις αυτό που παίζει σημαντικό ρόλο είναι να έχουμε κατανοήσει ποια πληροφορία λείπει. Όστε να γνωρίζουμε τον βαθμό ακρίβειας αλλά και κατά πόσο έγκυρη είναι. Όπως είναι προφανές δεν είναι ότι καλύτερο να θέλεις να πάρεις μια πληροφορία η οποία δεν είναι ολόκληρη και ψευδής.

1.2 ΜΕΤΡΑ ΠΡΟΣΤΑΣΙΑΣ ΤΗΣ ΠΛΗΡΟΦΟΡΙΑΣ

Το να μαζέψει και να αποθηκεύσει κάποιος πληροφορίες προκειμένου να έχει την δυνατότητα κάποιος να της χρησιμοποιήσει απαιτεί αρκετό χρόνο και κόπο. Για αυτό τον λόγο αποκτά αξία μια πληροφορία, και θα πρέπει να προφυλαχθούν. Υπάρχουν τέσσερις τρόποι προκειμένου να προφυλαχθούν[3]:

- 1) Να έχουν πρόσβαση όσοι έχουν δικαίωμα αλλά και όσοι χρειάζονται την πληροφορία. Όπως επίσης να έχουν συγκεκριμένο αριθμό πρόσβαση και να έχουν την δυνατότητα ανάγνωσης μόνο.
- 2) Να είναι επιβεβαιωμένο ότι οι πληροφορίες είναι σωστές και ορθές διατυπωμένες. Δηλαδή να μην λείπει κανένα στοιχείο και τα στοιχεία να είναι αληθινά, υπολογισμένα σωστά και να εφαρμόζονται.
- 3) Να υπάρχει προστασία σε περίπτωση απώλειας πληροφορίας είτε επίτηδες είτε κατα λάθος. Αυτό επιτυγχάνεται με τον να έχουν πρόσβαση όσο λιγότεροι άνθρωποι στα αρχεία που φυλλάσσεται η πληροφορία. Ακόμη μπορεί να καταγράφονται όποιες τυχόν αλλαγές γίνονται στα αρχεία.
- 4) Να μην υπάρξει διαστρέβλωση της πληροφορίας μετά το πέρασμα χρονικού ορίου. Επειδή κάποιες πληροφορίες έχουν μικρό διάστημα ύπαρξης και άλλες μεγάλο. Για να εξασφαλίσουμε την καταλληλότητα της πληροφορίας θα πρέπει να υπάρχει συνεχής έλεγχος για το αν αυτή η πληροφορία είναι η καλύτερη και πιο πρόσφατη.

1.3 Η ΕΠΙΔΡΑΣΗ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ

Οι χρήστες υπάρχει περίπτωση να δεχτούν αρνητική επίδραση σε ότι αφορά τον τομέα της τεχνολογίας. Μπορεί να δημιουργηθούν αισθήματα άγχους και ανησυχίας λόγω της μη ολοκληρωμένης σχέσης ανάμεσα στον υπολογιστή και στους χρήστες. Οι λόγοι οι οποίοι οδηγούν στον φόβο που διακατέχει τους χρήστες απέναντι στην τεχνολογία είναι οι παρακάτω σύμφωνα με τον Bentley (1998)[3]. Όπως είναι λογικό όταν συναντάμε κάτι άγνωστο ή κάτι το οποίο δεν το κατανοούμε απολύτως θα κρατάμε μια στάση φόβου

απέναντι του. Ο χρήστης κατά την διάρκεια της ζωής του θα έρθει αντιμέτωπος με την αμφισβήτηση του εαυτού του λόγω κάποιων άσχημων πληροφοριών που θα του πουν. Με αποτέλεσμα όταν θα χρειαστεί να πάρει κάποιες αποφάσεις να μην είναι σε θέση να σκεφτεί σωστά για να κάνει τις απαραίτητες ενέργειες. Επειδή τον διακατέχει μια προκατάληψη σε ότι αφορά την ικανότητα του. Όταν θέλουμε να κάνουμε μια δουλεία θα πρέπει να υπάρχει ένα κίνητρο το οποίο θα μας δώσει ώθηση να την τελειώσουμε γρήγορα. Εάν το κίνητρο μας δεν είναι ισχυρό τότε θα δουλεύουμε με αργούς ρυθμούς και θα το κάνουμε απλά επειδή πρέπει και όχι για να μάθουμε. Επιπλέον υπάρχει και φόβος της κοροϊδίας ο οποίος συχνά οδηγεί να αισθάνεται απόρριψη ο χρήστης και έτσι να κλείνεται στον εαυτό του. Ακόμη μια ενδεχόμενη αποτυχία πέρα από τον φόβο της κοροϊδίας δημιουργεί τον φόβο της παρατήρησης. Για αυτό τον λόγο σε πολλούς ανθρώπους δεν τους αρέσει να δοκιμάζουν καινούργιες τεχνικές αλλά να συνεχίζουν σε αυτές που γνωρίζουν. Πολλοί άνθρωποι δεν χειρίζονται πολύ εύκολα τους υπολογιστές λόγω της λίγης γνώσης που έχουν για αυτούς. Επίσης περιμένουν να υπάρχει μια καλή σχέση ανάμεσα στον υπολογιστή και στον χρήστη. Κάποια υπολογιστικά συστήματα έχουν δημιουργηθεί ώστε να παρέχουν περισσότερη βοήθεια στον χρήστη, είτε να δίνουν πληροφορίες πάνω σε ένα πρόβλημα είτε να υποδεικνύουν ενέργειες για την λήψη αποφάσεων. Υπάρχει ένας όρος ο οποίος χαρακτηρίζει την τεχνολογία φιλική προς τον άνθρωπο. Απολύτως φιλική δεν μπορεί να είναι ακόμη αλλά μέχρι ένα βαθμό. Γίνονται πολλές ενέργειες πάνω σε αυτό το κομμάτι ώστε να μπορεί να γίνει η τεχνολογία όσο περισσότερο φιλική μπορεί. Για παράδειγμα έχουν καταφέρει να δημιουργήσουν την αναγνώριση φωνής στα πληροφοριακά συστήματα. Ομως όχι την απόλυτη επικοινωνία μεταξύ ανθρώπου και του συστήματος. Οι χρήστες έχουν μάθει να έχουν αυτοί τον απόλυτο έλεγχο σε ένα σύστημα. Για αυτό τον λόγο όταν θέλει να κάνει κάποια ενέργεια μέσα σε ένα πληροφοριακό σύστημα δεν μπορεί να ακολουθήσει τον δικό του τρόπο. Επειδή υπάρχει ήδη ένας καθορισμένος τρόπος ο οποίος έχει δημιουργηθεί από τον δημιουργό παρότι μπορεί να είναι πιο αργός αλλά και να μην έχει την σωστή δομή. Όλα αυτά σε ένα χρήστη του δημιουργούν μια ενόχληση αλλά από την στιγμή που δεν υπάρχει άλλος τρόπος αφήνει το σύστημα να τον καθοδηγεί.

1.4 ΤΟ ΦΑΙΝΟΜΕΝΟ ΤΗΣ ΠΛΗΡΟΦΟΡΙΑΚΗΣ ΥΠΕΡΦΟΡΤΩΣΗΣ

Καθημερινά κάθε άνθρωπος έρχεται αντιμέτωπος με μια μεγάλη ποσότητα πληροφοριών που θα πρέπει να διαχειριστεί για να πάρει μια συγκεκριμένη απόφαση. Επειδή υπάρχει όλος αυτός ο όγκος πληροφορίας δυσκολεύει τον χρήστη να βλέπει πιο καθαρά τις σημαντικές πληροφορίες που δέχεται από ότι τις ασήμαντες. Η πληροφοριακή υπερφόρτωση επιδρά αρνητικά στον χρήστη επειδή του ασκείται ψυχολογική και σωματική πίεση. Όλο αυτό το σύνολο από πληροφορίες που υπάρχει στο πληροφοριακό σύστημα, έχει ως αποτέλεσμα να προκαλεί άγχος στον χρήστη δυσκολεύοντας τον να πάρει την σωστή απόφαση[2].

Οι αιτίες που οδηγούν στην πληροφοριακή υπερφόρτωση είναι οι εξής[2]:

- Ο μεγάλος ρυθμός ανάπτυξης της παραγωγής νέων πληροφοριών, προϊόντων και υπηρεσιών, για λογαριασμό του πληροφοριακού συστήματος.
- Η συγκέντρωση ενός μεγάλου αριθμού πληροφοριών στη βάση δεδομένων του πληροφοριακού συστήματος.
- Η αύξηση του αριθμού υπαρχουσών καναλιών στο πληροφοριακό σύστημα πληροφοριών.
- Ο μεγάλος αριθμός μη έγκυρων πληροφοριών.
- Η εξέλιξη της τεχνολογίας που βοήθησε στην μεγιστοποίηση του όγκου, της μετάδοσης και διάδοσης των δεδομένων στα τηλεπικοινωνιακά δίκτυα του πληροφοριακού συστήματος.

1.5 ΑΝΤΙΜΕΤΩΠΙΣΗ ΤΗΣ ΠΛΗΡΟΦΟΡΙΑΚΗΣ ΥΠΕΡΦΟΡΤΩΣΗΣ

Η πληροφοριακή υπερφόρτωση που δέχεται ένας χρήστης δρα αρνητικά στον τομέα της ψυχολογικής του κατάστασης. Σύμφωνα με ένα πείραμα που έχει πραγματοποιηθεί [7] βγήκε το συμπέρασμα ότι η επιλογή των χρηστών να επιλέξουν κάποια απόφαση επηρεάζεται λόγω της ψυχολογίας τους. Οι αποφάσεις που παίρνουν δεν θα είναι τόσο σωστές και τόσο

ποιοτικές σε ότι αφορά το περιεχόμενο τους. Ένας οποιαδήποτε χρήστης έχει συγκεκριμένο όριο με το οποίο μπορεί να επεξεργαστεί αλλά και να εμπεδώσει μια πληροφορία μέσα σε ένα χρονικό όριο. Αξίζει να σημειωθεί πως καθημερινά κάθε χρήστης αφιερώνει πολύ χρόνο για να ψάξει πληροφορίες στο διαδίκτυο. Παρακάτω θα αναλυθούν κάποιοι τρόποι αντιμετώπισης της πληροφοριακής υπερφόρτωσης. Όπως είναι η εξατομίκευση, τα συστήματα φιλτραρίσματος πληροφορίας, τα συστήματα λήψης αποφάσεων, οι αποθήκες δεδομένων και η εξόρυξη δεδομένων. Στα επόμενα κεφάλαια θα ασχοληθούμε αναλυτικά με τις εφαρμογές στον τομέα της τεχνητής νοημοσύνης που αντιμετωπίζουν την πληροφοριακή υπερφόρτωση, δηλαδή για τους ευφυείς πράκτορες.

Η εξατομίκευση βοηθάει άμεσα στην λύση της πληροφοριακής υπερφόρτωσης επειδή δίνει στους χρήστες αυτό που χρειάζονται χωρίς να το αναζητήσουν ή να το ζητήσουν οι ίδιοι. Χρησιμοποιεί τεχνικές από άλλα επιστημονικά πεδία, προκειμένου να μαζέψει τα δεδομένα που χρειάζεται και να δώσει την εξατομικευμένη έξοδο για τον χρήστη. Οι τεχνικές τις οποίες χρησιμοποιεί περιλαμβάνουν την ανάκτηση πληροφοριών, τη μοντελοποίηση χρήστη, την τεχνητή νοημοσύνη και τις βάσεις δεδομένων. Με απλά λόγια η εξατομίκευση είναι μια διαδικασία συγκέντρωσης και αποθήκευσης πληροφοριών σε σχέση με τους χρήστες ενός πληροφοριακού συστήματος[4].

Η εξόρυξη δεδομένων είναι μια διαδικασία με την οποία μπορούν οι χρήστες να ανακαλύψουν γνώσεις από τις βάσεις δεδομένων.

Οι πιο ευρέως γνωστές τεχνικές εξόρυξης είναι οι εξής [8] :

- **Κατηγοροποίηση:** Η εύρεση ενός μοντέλου που περιλαμβάνει καινούργια δεδομένα σε μια ήδη υπάρχουσα κλάση.
- **Ομαδοποίηση:** Η διαπίστωση ενός συγκεκριμένου αριθμού ομάδων που χωρίζουν τα δεδομένα ανάλογα με τις ομοιότητες και τις διαφορές τους.
- **Χαρακτηρισμός και Διάκριση:** Η εύρεση μια περιγραφής η οποία είναι σε ισχύ για ένα συνολικό μέρος δεδομένων.
- **Συσχέτιση:** Η εξαγωγή συνδυασμού κανόνων που επισημαίνουν τις σχέσεις αίτιου-αποτελέσματος ανάμεσα από τα χαρακτηριστικά ενός συνόλου δεδομένων.
- **Ανάλυση Ακραίων Τιμών:** Η διαπίστωση και διαγραφή δεδομένων που δεν προσαρμόζονται σύμφωνα με τα άλλα μητρώα δεδομένων.
- **Τάση και Ανάλυση Ανάπτυξης :** Η εύρεση τάσεων, και η έρευνα της ανάπτυξης μιας κατάστασης στο πέρασμα του χρόνου.

Οι βάσεις δεδομένων προσφέρουν έναν τρόπο αποθήκευσης, διαχείρισης και πρόσβασης πληροφοριών. Όμως με τον τρόπο όπως είναι φτιαγμένες οι βάσεις δεδομένων κάνει πιο δύσκολη την αναζήτηση συγκεκριμένης πληροφορίας ανάμεσα σε τόσες άλλες. Για την λύση αυτού του προβλήματος υπάρχουν οι Αποθήκες Δεδομένων. Όπου περιέχονται μεγάλες ποσότητες δεδομένων στις οποίες μπορεί να γίνει γρήγορα προσπέλαση και ταξινόμηση. Οι αποθήκες δεδομένων δεν αναπληρώνουν τις λειτουργικές βάσεις δεδομένων, αλλά είναι ένα σημαντικό εργαλείο που παρέχει πρόσβαση σε ακριβή και λεπτομερή δεδομένα[3].

Το σύστημα υποστήριξης αποφάσεων είναι ένα πληροφοριακό σύστημα το οποίο δίνει πληροφορίες σε αυτούς που λαμβάνουν αποφάσεις ώστε να τους θέσει σε λειτουργία να πάρουν τις απαραίτητες αποφάσεις γρήγορα. Έτσι υπάρχει η δυνατότητα αποθήκευσης και ανάκτησης δεδομένων. Τα συστήματα αυτά βοηθούν τη σχεδίαση, τη μοντελοποίηση και την επίλυση προβλημάτων[3].

Το σύστημα φιλτραρίσματος πληροφορίας είναι μια μέθοδος η οποία μπορεί να διαχειριστεί μεγάλο όγκο πληροφορίας. Το καλό με αυτό το σύστημα είναι ότι δίνει στους χρήστες όσες πληροφορίες σχετίζονται με αυτούς. Δηλαδή είναι ένα σύστημα με δυνατότητα ανάκτησης πληροφορίας που μπορούν να διαχειρίζονται πολλά και διάφορα δεδομένα, τα οποία είναι βασισμένα στις προτιμήσεις των χρηστών. Οι μέθοδοι και οι τεχνικές που χρησιμοποιούν τα συστήματα αυτά προέρχονται από την ανάκτηση πληροφορίας στον τομέα της Τεχνητής Νοημοσύνης[9].

Το φιλτράρισμα πληροφορίας το χωρίζουμε σε τρεις κατηγορίες:

Απλο Φιλτράρισμα: Χωρίζει τους χρήστες σε συγκεκριμένες ομάδες, και στην συνέχεια στέλνει συγκεκριμένες σελίδες στους χρήστες κάθε κατηγορίας[4].

Φιλτράρισμα Βάσει Περιεχομένου: Αναλύει το περιεχόμενο των πληροφοριών, ώστε να μπορεί να σχηματίσει εντύπωση για τα ενδιαφέροντα του εκάστοτε χρήστη. Προτείνει στους χρήστες πληροφορίες, σύμφωνα με τις οποίες έχει επιλέξει και έχει συμφωνήσει στο παρελθόν, με αυτές δηλαδή είναι μέσα στις προτιμήσεις του. Το φιλτράρισμα βάσει περιεχομένου χρησιμοποιεί την Ευκλείδεια απόσταση για την ανάλυση τιμών προκειμένου να υπολογιστεί ποια πληροφορία έχει πιο κοντινή τιμή για να προταθεί στον χρήστη[10].

Συνεργατικό Φιλτράρισμα: Το φιλτράρισμα πληροφορίας σε αυτή την κατηγορία μαζεύει με έναν τρόπο την γνώμη των χρηστών για ένα σύνολο από αντικείμενα ώστε να δημιουργήσει διάφορες ομάδες με την ίδια γνώμη. Το σύστημα αυτό προτείνει πληροφορίες

οι οποίες έχουν βαθμολογηθεί θετικά από χρήστες , όπου οι βαθμολογίες και οι προτιμήσεις μοιάζουν με τον χρήστη που θα λάβει την πρόταση. Η Ευκλείδεια απόσταση χρησιμοποιείται πάλι εδώ για την ανάλυση τιμών και για τις πληροφορίες που προτείνουν στον χρήστη. Το φιλτράρισμα πληροφορίας αποτελεί ένα καλό μέτρο για την αντιμετώπιση της πληροφοριακής υπερφόρτωσης. Κάποιες από τις εφαρμογές της είναι τα φίλτρα των αποτελεσμάτων αναζήτησης στο διαδίκτυο ,τα φίλτρα περιηγητών που μπλοκάρουν άχρηστες πληροφορίες και το φιλτράρισμα ιστοσελίδων[11].

ΚΕΦΑΛΑΙΟ 2

ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ

2.1 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ ΚΑΙ ΕΞΕΛΙΞΗ ΤΗΣ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ

Από τα αρχαία χρόνια ο Αριστοτέλης το 384-322 π.Χ είχε θέσει κάποιους συλλογισμούς που έδιναν κάποιες εκφράσεις που κατέληγαν σε σωστά συμπεράσματα από σωστές υποθέσεις. Στα μέσα του 1940 παρουσιάστηκε η πρώτη μαθηματική περιγραφή τεχνητού νευρωνικού δικτύου, όμως με μικρή δυνατότητα λύσης αριθμητικών προβλημάτων. Αξίζει να τονίσουμε το 1950 ο μαθηματικός Alan Turing, ήταν ο δημιουργός της θεωρίας υπολογισμού και γενάρχης της τεχνητής νοημοσύνης. Ο οποίος πρότεινε την δοκιμή Turing , όπου υπήρχε η δυνατότητα να ελεγχθεί αν μια μηχανή διέθετε ευφυΐα. Θα αναλύσουμε την δοκιμή Turing παρακάτω[12]. Η ουσιαστική θεμελίωση της τεχνητής νοημοσύνης έγινε στην συνάντηση κάποιων Αμερικανών επιστημόνων το 1956 κάποιιοι από αυτούς ήταν: John McCarthy,

Marvin Minsky, Claude Shannon κ.λ.π. Εκείνη την χρονιά δημιουργήθηκε ένα πρόγραμμα το Logic Theorist, το οποίο βασιζόταν σε τυπικούς κανόνες συμπερασματικής λογικής και σε αλγόριθμους αναζήτησης για την απόδειξη μαθηματικών θεωρημάτων. Το 1958 ο John McCarthy ανέπτυξε μια γλώσσα προγραμματισμού η οποία βοήθησε στην δημιουργία εφαρμογών τεχνητής νοημοσύνης στα επόμενα χρόνια από τους Friedberg και Rosenblatt. Από το 1970 μέχρι το 1980 άρχισαν να φτιάχνονται πιο δυνατές και περισσότερες εφαρμογές στα νευρωνικά δίκτυα λόγω της νέας γλώσσας προγραμματισμού Prolog. Στα μέσα του 1990 λόγω της χρησιμοποίησης του διαδικτύου υπήρχε ανάπτυξη στους εφυείς πράκτορες. Για αυτό τον λόγο σήμερα η τεχνητή νοημοσύνη είναι μια επιστήμη η οποία ασχολείται με την υλοποίηση και την σχεδίαση εφυών πρακτόρων[13].

Παρακάτω θα δούμε επιγραμματικά την εξέλιξη της τεχνητής νοημοσύνης[14]:

1854: Ο George Boole έβαλε τις βάσεις της προτασιακής λογικής.

1879: Ο Gottlieb Frege πρότεινε ένα σύστημα αυτοματοποιημένης συλλογιστικής

1943: Ο McCulloch και ο Pitts έφτιαξαν ένα μοντέλο τεχνητών νευρώνων που μπορούσε να μάθει και να υπολογίζει κάθε υπολογίσιμη συνάρτηση.

1950: Ο Alan Turing(1913-1954) θεωρείται δημιουργός της τεχνητής νοημοσύνης, δημιούργησε ένα τεστ , για την ανίχνευση εφυών μηχανών.

1951: Ο Minsky και ο Edmonds έφτιαξαν το πρώτο νευρωνικό δίκτυο.

1956: Έγινε μια συνεδρίαση η οποία δημιουργήθηκε από τους McCarthy, Minsky, Shannon και Rochester η οποία έμελλε να είναι η αρχή του τομέα της τεχνητής νοημοσύνης.

1958: Ο McCarthy δημιούργησε μια γλώσσα προγραμματισμού LISP.

1958: Ο Friedberg έθεσε μια νέα τεχνική, τη μηχανική εξέλιξη.

Δεκαετία του 1960: Φτιάχτηκε το πρώτο robot.

1968: Ο Tom Evans έφτιαξε ένα πρόγραμμα το οποίο είχε την δυνατότητα να λύνει προβλήματα γεωμετρικής φύσης, τα οποία χρησιμοποιούνταν σε τεστ ευφυΐας.

1962: Υπήρχε μια αλλαγή στον τρόπο της μεθόδου μάθησης των νευρωνικών δικτύων του Hebb.

1965: Ο Weizenbaum έφτιαξε ένα πρόγραμμα με το όνομα ELIZA το οποίο είχε την δυνατότητα να κάνει συζήτηση για ένα θέμα που θα το ρωτούσαν. Για να τα κάνει αυτά όμως παραλλάσσοντας και χρησιμοποιώντας τις προτάσεις που έδινε ο χρήστης σαν ερώτηση.

Δεκαετία του 1970: Δημιουργήθηκαν τα πρώτα συστήματα τα οποία με γνώση ώστε να μπορούν να συμπεριφέρονται όπως οι άνθρωποι.

1975: Ο Minsky πρότεινε τα πλαίσια.

1981: Οι Ιάπωνες έφτιαξαν υπολογιστές με γλώσσα προγραμματισμού την Prolog.

Μέσα δεκαετίας του 1980: Εμφανίστηκαν πάλι τα νευρωνικά δίκτυα.

Δεκαετία του 1980: Χρησιμοποιήθηκε πάλι ο αλγόριθμος μάθησης με οπισθοδρόμηση και έδωσε λύση σε πολλά προβλήματα.

Σήμερα: Η Τεχνητή νοημοσύνη έχει αναπτυχθεί τόσο ώστε να μπορεί να ανταποκρίνεται σε συστήματα τα οποία έχουν ως κύριο στόχο την εξελικτική διαδικασία ή σε πράκτορες, πέρα από τα έμπειρα συστήματα.

2.2 ΟΡΙΣΜΟΣ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ

Τεχνητή νοημοσύνη είναι μια δημιουργία προγραμμάτων όπου ο κύριος στόχος τους είναι να δρουν έξυπνα όπως ο ανθρώπινος νους. Με τον όρο έξυπνα εννοούμε να υπάρχει η ικανότητα να λύσουν προβλήματα, να έχουν την δυνατότητα μάθησης από προηγούμενες εκτελέσεις, να έχουν δυνατότητα αντίληψης καταστάσεων. Δηλαδή τις ομοιότητες, τις διαφορές αλλά και τον βαθμό δυσκολίας μιας κατάστασης. Αν θέλουμε να δώσουμε έναν γενικό ορισμό για την τεχνητή νοημοσύνη είναι ο εξής: ορίζουμε τον τομέα της επιστήμης υπολογιστών όπου το αντικείμενο με το οποίο ασχολείται είναι η υλοποίηση και η σχεδίαση υπολογιστικών συστημάτων. Τα συστήματα αυτά είναι σχεδιασμένα με βάση τα ανθρώπινα χαρακτηριστικά δηλαδή μάθηση, προσαρμοστικότητα, επίλυση προβλημάτων, εξαγωγή συμπερασμάτων κ.λ.π[14]. Ο J. McCarthy διατύπωσε τον τομέα ως «επιστήμη και μεθοδολογία της δημιουργίας νοούντων μηχανών»[15].

Η τεχνητή νοημοσύνη αποτελεί κοινό σημείο πολλών τομέων όπως για παράδειγμα: της επιστήμης υπολογιστών, της ψυχολογίας, της φιλοσοφίας, της νευρολογίας, της γλωσσολογίας και της επιστήμης μηχανικών. Με κύριο σκοπό τον συνδυασμό ευφυούς συμπεριφοράς, με τα απαραίτητα στοιχεία μάθησης και προσαρμογής στο περιβάλλον.

Σύμφωνα με τους Russel και Norvig διαχωρίζουν τους ορισμούς της τεχνητής νοημοσύνης σε τέσσερις κατηγορίες, σύμφωνα με το αν χαρακτηρίζουν ένα σύστημα ευφυές με τα εξής κριτήρια[16]:

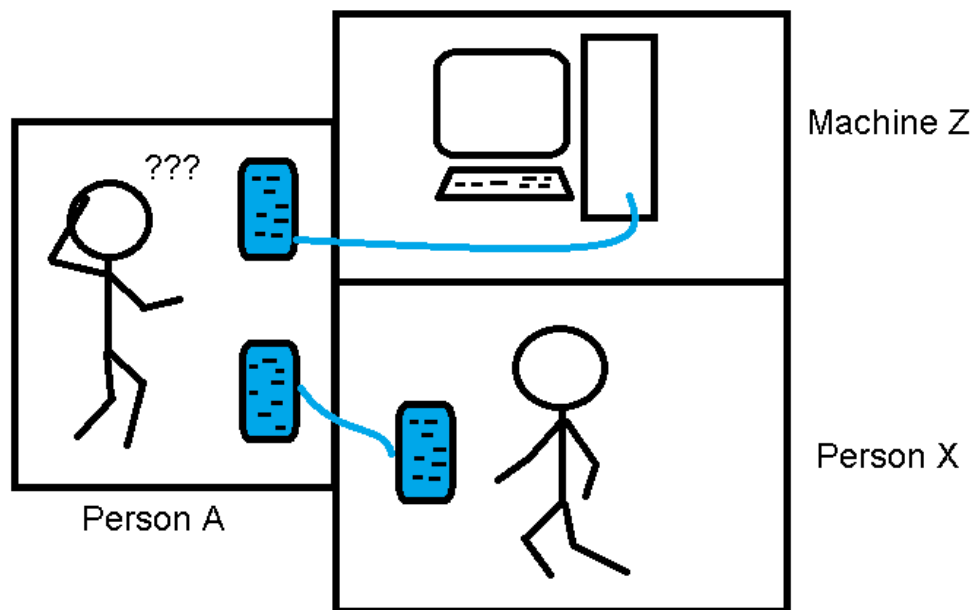
- ✓ Αν ενεργεί σαν άνθρωπος (Συμπεριφορά, Turing test)
- ✓ Αν σκέφτεται σαν άνθρωπος (Μηχανισμός)
- ✓ Αν ενεργεί ορθολογικά (Συμπεριφορά, ορθολογικοί πράκτορες)
- ✓ Αν σκέφτεται ορθολογικά (Μηχανισμός)

2.3 Η ΔΟΚΙΜΑΣΙΑ TURING

Όπως είχαμε αναφέρει και πιο πάνω το 1950 ο Alan Turing επινόησε την δοκιμασία Turing. Δηλαδή ανάπτυξη μιας θεωρίας με την οποία μπορούσαμε να καταλάβουμε αν ένας υπολογιστής διαθέτει ευφυΐα. Για να πραγματοποιηθεί αυτή η διαδικασία χρειάζονται τρία δωμάτια χωρίς να υπάρχει άμεση επικοινωνία μεταξύ τους, δυο άνθρωποι και ένας ηλεκτρονικός υπολογιστής. Τον πρώτο άνθρωπο θα τον ονομάσουμε A και τον δεύτερο άνθρωπο B. Λοιπόν ο άνθρωπος A τοποθετείται μόνος του στο πρώτο δωμάτιο πληκτρολογώντας ερωτήσεις διάφορου περιεχομένου. Όπως για παράδειγμα μπορεί να είναι ερωτήσεις συναισθηματικές, κρίσεως, εγκυκλοπαιδικών γνώσεων κ.λ.π. Οι ερωτήσεις αυτές στέλνονται στον άνθρωπο B και στον ηλεκτρονικό υπολογιστή που βρίσκονται ξεχωριστά στα άλλα δυο δωμάτια. Ο άνθρωπος B και ο ηλεκτρονικός υπολογιστής απαντούν στις ερωτήσεις που έχει θέσει ο A. Ο οποίος με την σειρά του προσπαθεί βλέποντας τις απαντήσεις που του έχουν δοθεί ποιός είναι ο B και ποιός είναι ο υπολογιστής. Όταν δεν μπορεί ο άνθρωπος A να τους ξεχωρίσει τότε ο ηλεκτρονικός υπολογιστής έχει ευφυΐα. Για να μπορέσει ο υπολογιστής να καταφέρει να περάσει αυτή την δοκιμασία με επιτυχία θα πρέπει να έχει τις εξής ικανότητες[12]:

- A) Επεξεργασία φυσικής γλώσσας
- B) Αυτοματοποιημένη συλλογιστική
- Γ) Αναπαράσταση γνώσης
- Δ) Μηχανική μάθηση
- E) Μηχανική όραση και ρομποτική

Παρακάτω βλέπουμε μια εικόνα για το πώς είναι σχηματικά το τεστ Turing. Βλέπουμε τα τρία δωμάτια τα οποία δεν έχουν επικοινωνία μεταξύ τους, δυο ανθρώπους και έναν υπολογιστή.



Σχήμα 2.1 Εικόνα από Δοκιμή Turing[17]

ΚΕΦΑΛΑΙΟ 3

ΕΥΦΥΕΙΣ ΠΡΑΚΤΟΡΕΣ

3.1 ΟΡΙΣΜΟΣ ΠΡΑΚΤΟΡΑ

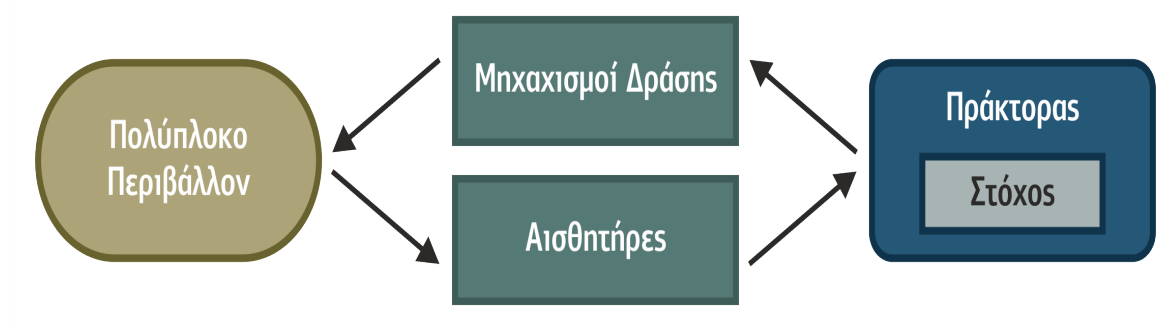
Η λέξη ευφυής πράκτορας , είναι μια έννοια όπου δεν υπάρχει ένας κοινός ορισμός από όλους τι είναι ακριβώς. Επειδή είναι δύσκολο να βρεθούν τα χαρακτηριστικά του , τα οποία σχετίζονται με αυτόν και την λειτουργία του γιατί δεν έχουν την ίδια σημασία σε διάφορους επιστημονικούς τομείς. Θα μπορούσαμε να θεωρήσουμε ότι ένας πράκτορας (agent) αντιλαμβάνεται το περιβάλλον (environment) μέσα στο οποίο υπάρχει, με την βοήθεια αισθητήρων (sensors). Δηλαδή είναι κομμάτι του περιβάλλοντος αυτού , κάνοντας συλλογισμούς για αυτό και δρώντας πάνω σε αυτό με την βοήθεια μηχανισμών δράσης (effectors), προκειμένου να επιτύχει κάποιους στόχους (goals) (Russell & Norvig, 1995). Παρακατώ βλέπουμε σε σχηματική απεικόνιση αυτό που υποστήριξαν οι δυο εκπρόσωποι.



Σχήμα 3.1 Βασικός Πράκτορας κατά Russel και Norvig.[18]

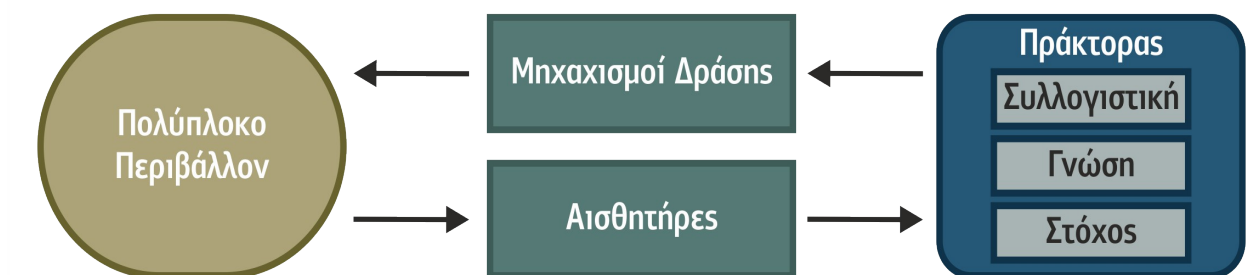
Η Pattie Maes (1995) υποστήριξε ότι η βασική ιδέα είναι κάτω από ορισμένες συνθήκες ο ευφυής πράκτορας « θα μπορεί να προγραμματίζει τον εαυτό του» αποκτώντας όμως την κατάλληλη γνώση ώστε να μπορεί να υποστηρίξει τον χρήστη. Είναι σημαντικό να πούμε ότι ο πράκτορας οι πληροφορίες που λαμβάνει είναι μηδαμινές , μαθαίνοντας όμως την σωστή συμπεριφορά είτε από τον χρήστη είτε από άλλους πράκτορες. Θέλει να δώσει έμφαση στο πολύπλοκο και δυναμικό περιβάλλον πέρα από την αυτονομία. Όπως υποστήριξε ακριβώς: «Οι πράκτορες είναι υπολογιστικά συστήματα που δρουν σε ένα πολύπλοκο περιβάλλον, αντιλαμβάνονται και δρουν

αυτόνομα πάνω σε αυτό, πετυχαίνοντας έτσι ένα σύνολο από στόχους για τους οποίους έχουν κατασκευαστεί»[19]



Σχήμα 3.2 Βασικός Πράκτορας κατά Maes [20]

Όταν θέλουμε να δώσουμε έναν ορισμό για έναν ευφυή πράκτορα ο οποίος να ανταποκρίνεται στα χαρακτηριστικά τους είναι το μοντέλο Hayes – Roth (1995), που δίνει έμφαση στη συλλογιστική: « Οι ευφυείς πράκτορες κάνουν συνεχώς τις εξής τρεις λειτουργίες : (α) αντιλαμβάνονται τις δυναμικές συνθήκες του περιβάλλοντος, (β) δρουν πάνω στο περιβάλλον ώστε να το αλλάξουν και (γ) συλλογίζονται ώστε να ερμηνεύσουν αυτά που αντιλαμβάνονται, να λύσουν προβλήματα , να συμπεράνουν και να καθορίσουν τη δράση τους.» [21]



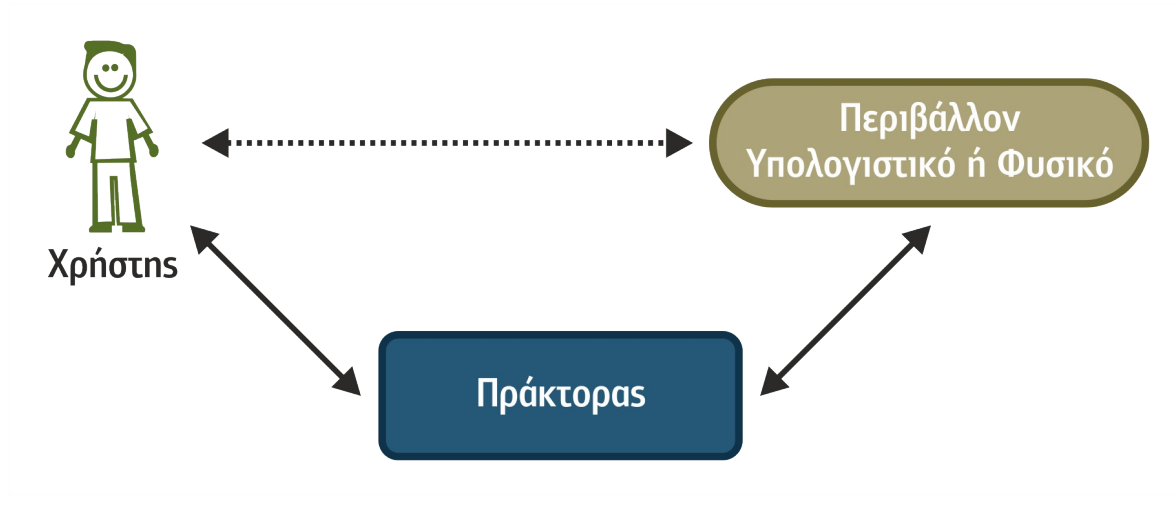
Σχήμα 3.3 Βασικός πράκτορας κατά Hayes-Roth [22]

Ο Coen δίνει έμφαση στην έννοια της διαδραστικότητας αλλά και στην επικοινωνία μεταξύ των πρακτόρων. «Λογισμικοί πράκτορες είναι προγράμματα που διενεργούν διάλογο, διαπραγματεύονται και συντονίζουν τη ροή πληροφοριών.»[23]



Σχήμα 3.4 Βασικός πράκτορας κατά Coen[24]

Η γενική χρήση των πρακτόρων οδηγεί σε μια έμμεση επικοινωνία του χρήστη με το λογισμικό όπως θα παρατηρήσουμε και στην παρακάτω εικόνα.



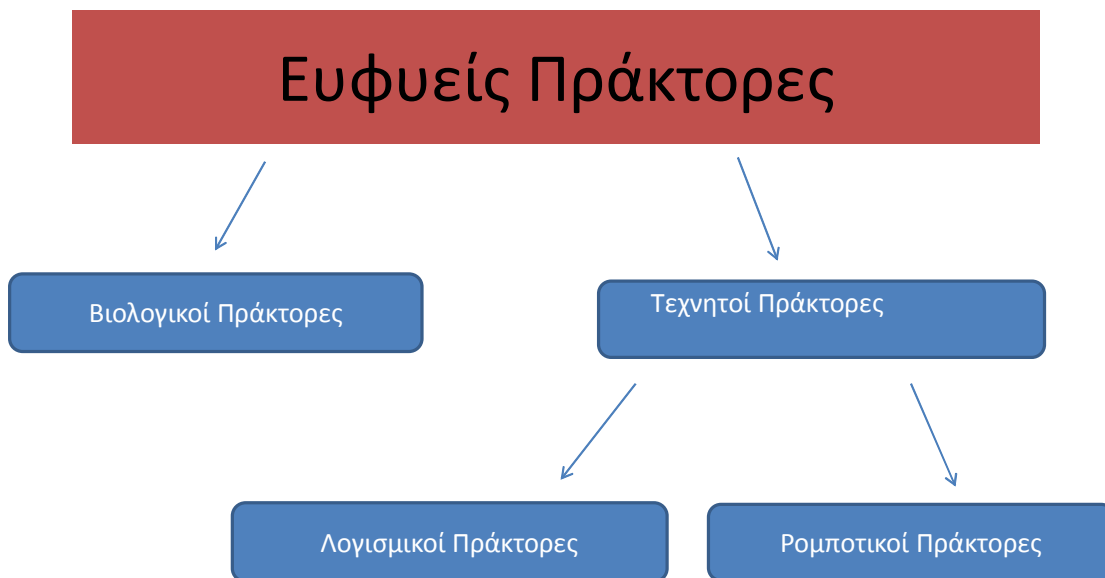
Σχήμα 3.5 Πίνακας πράκτορας [25]

Ο ευφυής πράκτορας δηλαδή αποτελεί μια εφαρμογή η οποία δρα εκ μέρους του ανθρώπου και υλοποιεί τις διεργασίες που του έχουν διαμεριστεί , μέσα σε κάποιους περιορισμούς[26]. Λόγω της αναπτυγμένης νοημοσύνη τους , μπορούν να συλλέξουν πληροφορίες από το διαδίκτυο για τον χρήστη χωρίς την παρουσία του. Όταν συλλέξουν τις πληροφορίες , τις αποθηκευούν ώστε όταν χρειαστεί να μπορούν εύκολα να τις ανακτήσουν. Είναι σημαντικό να αναφέρουμε ότι οι ευφυείς πράκτορες μπορούν να επανεκπαιδευτούν και να αλλάξουν ώστε να κάνουν την καλύτερη δυνατή επιλογή στην συλλογή δεδομένων. Επειδή υπάρχουν εκατοντάδες πληροφορίες στο διαδίκτυο, με αυτό τον τρόπο γίνονται πιο αποδοτικοί γιατί μαθαίνουν τις προτιμήσεις του χρήστη[3]. Ακόμη παρέχουν την δυνατότητα σε διαφορετικούς χρήστες να συνεργάζονται μεταξύ τους , να παρακολουθούν διαδικασίες και γεγονότα. Οι εφαρμογές που μπορούμε να έχουμε στην ουσία είναι πάρα πολλές , απεριόριστες θα λέγαμε. Περιλαμβάνουν την αναζήτηση, την ανάκτηση , το φιλτράρισμα μιας πληροφορίας, την διαχείριση ηλεκτρονικών μηνυμάτων μέχρι την επιλογή μιας μουσικής ή ενός βιβλίου[27].

3.2 ΚΑΤΗΓΟΡΙΕΣ ΕΥΦΥΩΝ ΠΡΑΚΤΟΡΩΝ

Σε ένα πιο γενικό επίπεδο για τους πράκτορες , μπορούμε να τους χωρίσουμε σε δυο μεγάλες κατηγορίες: α) στους τεχνητούς και β) στους βιολογικούς πράκτορες. Οι τεχνητοί πράκτορες χωρίζονται σε μια ακόμη υποκατηγορία στους ρομποτικούς (robotic agents ή robots) και στους λογισμικούς πράκτορες (software agents ή softbots). Οι ρομποτικοί πράκτορες χρησιμοποιούν ως αισθητήρα μηχανικά ή ηλεκτρονικά μέρη τα οποία όμως έχουν ανταπόκριση σε πραγματικό χρόνο. Αυτά τα δυο είδη περιέχουν μια συλλογιστική διαδικασία με την οποία μπορούν να επεξεργαστούν το περιβάλλον τους και αυτό έχει ως αποτέλεσμα να αλλάξουν την κατάσταση του. Οι βιολογικοί πράκτορες προκειμένου να αντιληφθούν το περιβάλλον στο οποίο βρίσκονται χρησιμοποιούν τις αισθήσεις τους[28].

Παρακάτω βλέπουμε το σχήμα με την ιεραρχική ταξινόμηση τους :



Σχήμα 3.6 Ταξινόμηση των πρακτόρων

3.3 ΙΔΙΟΤΗΤΕΣ ΠΡΑΚΤΟΡΩΝ

Όπως είχαμε αναφέρει πάλι πιο πάνω δεν υπάρχει ένας ακριβής όρος που να είναι ευρέως διαδεδομένος σε παγκόσμιο πλαίσιο, για την ακριβή έννοια του νοήμονα πράκτορα. Έτσι θα αναφέρουμε κάποιες ιδιότητες τους ώστε να μπορούμε να κατανοήσουμε τον τρόπο με τον οποίο λειτουργούν. Ένα σημαντικό πράγμα που πρέπει να αναφέρουμε, είναι ότι υπάρχουν διαφορές ανάμεσα στον πράκτορα λογισμικού και στα συμβατικά προγράμματα. Φυσικά δεν μπορούμε να απαντήσουμε εύκολα σε αυτό το πράγμα. Επειδή ο όρος του περιέχει ένα αριθμό συστημάτων με

διαφορές ανάμεσα στην πολυπλοκότητα και στα μεμονωμένα χαρακτηριστικά. Το μοντέλο των πρακτόρων θεωρείται ως νέας γενιάς λογισμικού και χρησιμοποιούνται ως στοιχείο της Τεχνητής Νοημοσύνης το οποίο είναι αναγκαίο ώστε να φτιαχτούν έξυπνες οντότητες. Ένα σημαντικό στοιχείο που θα πρέπει να υπάρχει ανάμεσα στους πράκτορες είναι οι διαπραγματεύσεις. Σε περίπτωση που δεν υφίστανται διαπραγμάτευση, όπως είναι λογικό δεν υπάρχει και συμφωνία με την εκάστοτε ομάδα πρακτόρων. Όπως επίσης η διαπραγμάτευση μεταξύ τους, προϋποθέτει την συνεργασία και το συντονισμό ώστε να μπορεί να υπάρξει αλληλεπίδραση.

Οι ιδιότητες που κάνουν τους πράκτορες να ξεχωρίζουν από τα άλλα είδη εφαρμογών λογισμικού είναι:[29]

➤ **Αυτονομία (autonomy)**: Οι πράκτορες μπορούν να λειτουργήσουν δίχως την παρεμβολή του εκάστοτε χρήστη ή άλλων πρακτόρων και κατέχουν τον αυτοέλεγχο. Δηλαδή έχουν αυτοί τον πλήρη έλεγχο της εσωτερικής τους κατάστασης και των ενεργειών τους. Έτσι με αυτόν τον τρόπο οι πράκτορες μπορούν να λειτουργήσουν χωρίς να δέχονται συχνά εντολές και έτσι απαλλάσει τον χρήστη από την έννοια της λήψης απόφασης.

➤ **Κοινωνικότητα (social ability)**: Η επικοινωνία μεταξύ των πρακτόρων αλλά και με χρήστες απαιτεί μια κοινή γλώσσα ώστε να υπάρχει κατανόηση μεταξύ τους. Έτσι θα έρθει το επιθυμητό αποτέλεσμα που δεν είναι άλλο από την συνεργασία για την κατόρθωση στόχων.

➤ **Αντιδραστικότητα (reactiveness)**: Είναι μια ιδιότητα η οποία κάνει τους πράκτορες να αντιδρούν αλλά και να αντιλαμβάνονται τις αλλαγές του περιβάλλοντος, όμως μέσα σε συγκεκριμένο χρονικό όριο.

➤ **Προνοητικότητα (pro-activeness)**: Οι πράκτορες δεν αντιδρούν απλα στο περιβάλλον αλλά έχουν την δυνατότητα να δείχνουν συμπεριφορά που κατευθύνεται από στόχους. Στην ουσία λαμβάνουν μια πρωτοβουλία ανάλογα με τις υπάρχουσες συνθήκες. Οι δύο ιδιότητες η αντιδραστικότητα και η προνοητικότητα βάζουν τον πράκτορα σε διαδικασία συλλογισμού.

Αξίζει να σημειωθεί ότι υπάρχουν και κάποιες δευτερεύοντες ιδιότητες οι οποίες είναι:

➤ **Προσαρμοστικότητα (adaptivity)**: Είναι η ικανότητα προσαρμογής που έχουν σε διάφορες συνθήκες περιβάλλοντος.

➤ **Κινητικότητα (mobility)**: Οι πράκτορες έχουν την ικανότητα να μην μένουν σταθεροί αλλά να μπορούν να κινηθούν σε διάφορα υπολογιστικά περιβάλλοντα.

➤ **Ειλικρίνεια (veracity)**: Οι πράκτορες ενεργούν και θα δώσουν αναφορά στον χρήστη με ειλικρίνεια, δηλαδή δεν θα δώσουν επίτηδες λάθος πληροφορίες πάντα με κύριο σκοπό έχουν το καλό του χρήστη.

- **Λογικότητα (rationality)**: Οι πράκτορες δεν αντιτίθενται στο αρχικό πλάνο των στόχων που τους έχουν ανατεθεί αντίθετα κάνουν τις απαραίτητες ενέργειες προκειμένου να καταφέρουν να τους υλοποιήσουν.
- **Αγαθή προαίρεση (benevolence)**: Οι πράκτορες καταβάλουν προσπάθεια ώστε να φέρουν πάντα εις πέρας τους στόχους που τους έχουν ανατεθεί.
- **Ευκινησία (agility)**: Μπορούμε να τη χαρακτηρίσουμε και ως συνέχεια από την ιδιότητα της αυτονομίας, επειδή ο πράκτορας μπορεί να αλλάζει από το ένα τμήμα του δικτύου στο άλλο με τρόπο αυτολεγχόμενο. Όλα αυτά τα κάνει με δεδομένο να μην παρεκκλίνει από τα καθήκοντα τα οποία του έχουν ανατεθεί.

Τις αυτοματοποιημένες έρευνες διαπραγματεύσεων μπορούμε να τις εξετάσουμε σε τρεις κατηγορίες ανάλογα φυσικά με την διαπραγμάτευση αλλά και το περιβάλλον που υπάρχει. Οι κατηγορίες ή όπως ονομάζονται συνήθως πρωτόκολλα διαπραγμάτευσης είναι:[29]

- α) Πρωτόκολλα Διαπραγμάτευσης (σύνολα κανόνων που κανονίζουν τις διαπραγματεύσεις),
- β) Αντικείμενα Διαπραγμάτευσης (η δυνατότητα αλλαγής κατασκευής του αντικειμένου που διαπραγματεύονται).
- γ) Μοντέλο Απόφασης Πρακτόρων (η ενέργεια που κάνουν σύμφωνα με το πρωτόκολλο διαπραγμάτευσης προκειμένου να επιτύχουν τους στόχους τους) .

Όταν ξεκινάει μια διαδικασία ο κάθε πράκτορας έχει μια ποσότητα στο χώρο όπου θέλει να γίνει η συμφωνία. Στην χρονική διάρκεια που εκτελείται η διαδικασία ο χώρος που γίνεται η διαδικασία υπάρχει πιθανότητα να αλλάξει. Αυτό γίνεται επειδή αλλάζει το περιβάλλον ή είναι αποφασισμένοι να αλλάξουν τις απόψεις τους. Το τέλος μια έρευνας έρχεται όταν ένας σημαντικός αριθμός από αυτούς που συμμετέχουν έχουν βρει ένα ίδιο σημείο συμφωνίας ή όταν το πρωτόκολλο έχει οριστεί να τερματιστεί η συμφωνία ανεξαρτήτου αν έχει πραγματοποιηθεί. Μια ενέργεια που έχουν οι πράκτορες σε σχέση με τις διαπραγματεύσεις είναι να μπορούν να δημιουργούν αλλά και να απαντάνε σε προτάσεις. Όμως σε περίπτωση που μπορούν μόνο να δέχονται προτάσεις ή να απορρίπτουν τότε η διαπραγμάτευση δεν θεωρείται επαρκής, από την στιγμή που αυτός που την προτείνει δεν έχει κανένα μέσο. Επειδή η πρόταση δεν είναι αποδεκτή, ούτε ο πράκτορας είναι κοντά να επέλθει σε συμφωνία αλλά ούτε σε οποιαδήποτε κατάσταση χώρου συμφωνίας θα μπορεί να πάει αργότερα. Το μέσο το οποίο μπορεί να μας βοηθήσει να βελτιώσουμε την διαδικασία διαπραγμάτευσης είναι η ανατροφοδότηση. Επειδή στις προτάσεις τις οποίες δέχεται η ανατροφοδότηση παίρνει το ρόλο κριτικής ή μιας αντιπρότασης. Έτσι θα μπορεί

να επιτύχει, αυτός που θα προτείνει και θα είναι σε θέση να δημιουργεί μια πρόταση όπου υπάρχει πιθανότητα να επέλθει η συμφωνία.

3.4 ΤΥΠΟΙ ΠΕΡΙΒΑΛΛΟΝΤΟΣ ΤΟΥ ΠΡΑΚΤΟΡΑ

Μια έννοια που αναφέραμε συχνά στις ιδιότητες των πρακτόρων είναι η λέξη περιβάλλον μέσα στο οποίο ενεργεί ο πράκτορας. Ανάλογα με τα χαρακτηριστικά που διακατέχει τον καθένα πράκτορα, μπορούμε να χωρίσουμε σε κατηγορίες τα περιβάλλοντα τους. Παρακάτω έχουμε τις εξής κατηγορίες (Russell and Norvig, 1995)[30]:

- **Προσβάσιμα ή Μη Προσβάσιμα (Accessible- Inaccessible):**

Για να θεωρηθεί ένα περιβάλλον προσβάσιμο θα πρέπει οι πράκτορες να έχουν κυρίαρχη και ακριβή προσπέλαση στις ενημερωμένες πληροφορίες σε σχέση με την επικρατούσα κατάσταση του περιβάλλοντος. Όταν ένα περιβάλλον είναι προσβάσιμο ο πράκτορας δεν χρειάζεται να έχει εσωτερική κατάσταση και έχει την δυνατότητα να λειτουργήσει πιο εύκολα μέσα σε αυτό. Τις περισσότερες φορές τα πολύπλοκα περιβάλλοντα είναι μη προσβάσιμα όπως για παράδειγμα το διαδίκτυο.

- **Ντετερμινιστικό ή Μη Ντετερμινιστικό (Deterministic –Nondeterministic):**

Ένα περιβάλλον για να θεωρηθεί ντετερμινιστικό θα πρέπει κάθε ενέργεια του πράκτορα να έχει μοναδική επίδραση πάνω σε αυτό. Δηλαδή να υπάρχει σαφήνεια για την κατάσταση του περιβάλλοντος σε κάθε ενέργεια του. Ένας γενικός ορισμός για το τι θεωρούμε μη ντετερμινιστικό είναι όταν ο πράκτορας εκτελεί κάποια ενέργεια και το αποτέλεσμα δεν είναι το ίδιο πάντα. Επίσης δεν είναι επιτυχής πάντα η ενέργεια που θα κάνει ένα παράδειγμα που μπορούμε να αναφέρουμε είναι ο φυσικός κόσμος.

- **Επεισοδιακό ή Μη Επεισοδιακό (Episodic- Nonepisodic) :**

Στο επεισοδιακό περιβάλλον η απόδοση του πράκτορα εξαρτάται από μία σειρά διακριτών επεισοδίων. Το σημαντικό είναι ότι ο κάθε πράκτορας έχει την δυνατότητα να επιλέξει μια δράση θέλει να εκπληρώσει πρώτη ανάλογα βέβαια με το τρέχων επεισόδιο. Αυτό έχει ως

αποτέλεσμα αυτό το περιβάλλον να θεωρείται πιο απλό και εύκολο για την ανάπτυξη των πρακτόρων.

▪ **Διακριτό ή Συνεχές (Discrete- Continuous) :**

Ένα περιβάλλον για να είναι διακριτό πρέπει να υπάρχει ένας πεπερασμένος και συγκεκριμένος αριθμός ενεργειών στον μηχανισμό αντίληψης του πράκτορα. Στις δύσκολες περιπτώσεις σε ένα περιβάλλον μπορούμε να προσθέσουμε το συνεχές εκτός πέρα από το μη προσβάσιμο, μη επεισοδιακό κ.λ.π

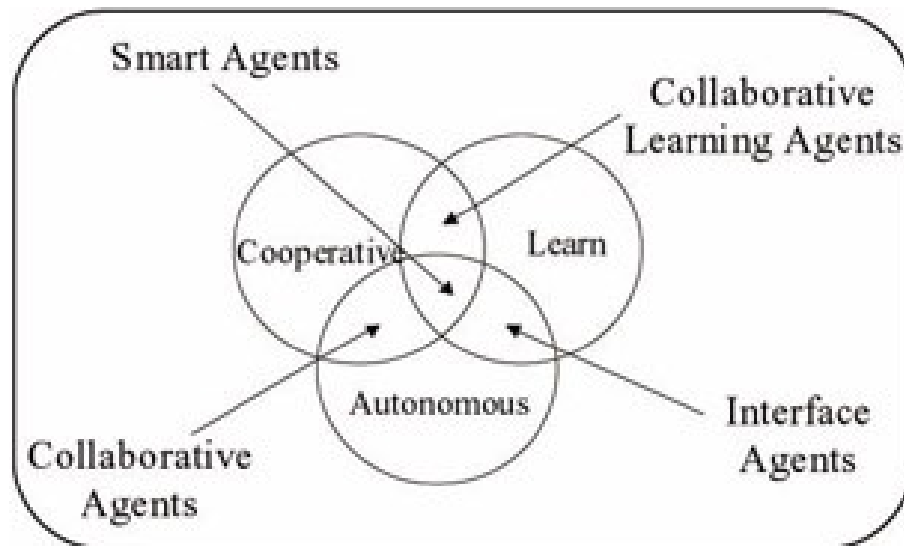
▪ **Στατικό ή Δυναμικό (Dynamic- Static) :**

Το στατικό περιβάλλον είναι όταν μένει χωρίς καμία αλλαγή όταν κανένας πράκτορας δεν εκτελεί καμία ενέργεια. Έτσι θεωρείται πιο απλουστευμένο το περιβάλλον αυτό γιατί δεν χρειάζεται ο πράκτορας πριν να εκτελέσει κάποια ενέργεια να το έχει παρατηρήσει.

3.5 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ ΠΡΑΚΤΟΡΑ

Για να μπορέσουμε να χωρίσουμε τους νοήμονες πράκτορες σε συγκεκριμένες κλάσεις, θα πρέπει να δώσουμε σημασία σε κάποιες παραμέτρους. Θα ακολουθήσουμε όπως τις αναφέρει η Nwana το 1996 [31]. Οι δύο κλάσεις που δημιουργούνται είναι στατικοί και κινητοί πράκτορες, το κριτήριο με το οποίο κάνουμε τον διαχωρισμό είναι αν μπορούν να κινηθούν μέσα σε ένα δικτυο. Ακόμη μπορούμε να τους χαρακτηρίσουμε ως αντιδραστικούς ή συλλογιστικούς. Οι αντιδραστικοί πράκτορες δεν έχουν εσωτερικό συμβολικό μοντέλο, δρουν με ένα μοντέλο απάντησης σχετικά με την τωρινή κατάσταση του περιβάλλοντος που τους έχει ανατεθεί. Αντίθετα οι συλλογιστικοί πράκτορες έχουν εσωτερικό συμβολικό μοντέλο. Ο κύριος σκοπός τους είναι ο προγραμματισμός και φυσικά η διαπραγμάτευση με άλλους πράκτορες για την υλοποίηση των στόχων τους. Μια ακόμη ένταξη που μπορούμε να κάνουμε είναι ανάλογα αν εμφανίζουν χαρακτηριστικά όπως είναι η αυτονομία, η κοινωνικότητα και η μάθηση. Βασισμένοι πάνω σε αυτά τα χαρακτηριστικά μπορεί να προκύψουν και άλλοι τύποι πρακτόρων όπως : α) συνεργατικούς πράκτορες, β) συνεργατικούς πράκτορες μάθησης, γ) πράκτορες διεπαφής και δ) νοήμονες πράκτορες.

Παρακατώ έχουμε μια συνοπτική απεικόνιση όπως τα έχει αναφέρει η Nwana:



Σχήμα 3.7 Διάκριση πρακτόρων (Nwana, 1996) [31]

Ένας άλλος τρόπος που μπορούμε να διαχωρίσουμε τους πράκτορες είναι ανάλογα με τον ρόλο που έχουν. Όπως οι πράκτορες πληροφορίας οι οποίοι δυσκολεύονται γενικά γιατί θα πρέπει να διαχειριστούν έναν τεράστιο αριθμό πληροφοριών. Όλες αυτές οι πληροφορίες είναι στην διάθεση των χρηστών στο διαδίκτυο. Επιπλέον μπορούμε να προσθέσουμε στην λίστα την κατηγορία των υβριδικών πρακτόρων, η οποία μπορεί να χωριστεί σε περισσότερες από μια διαφορετικές φιλοσοφίες αλλά σε ένα πράκτορα. Ο διαχωρισμός των πρακτόρων σε διάφορες κατηγορίες έχει την απαίτηση να υπάρχουν πολλές διαστάσεις αλλά και πάλι μια τέτοια ενέργεια δεν θα ήταν εύστοχη. Έτσι αυτό μας οδηγεί να έχουμε μια τελική λίστα από τους διαχωρισμούς των πρακτόρων την οποία θα αναλύσουμε πιο κάτω. Όπως θα παρατηρήσουμε κάποιες εφαρμογές απαιτούν την συσχέτιση πρακτόρων από διαφορετικές κατηγορίες. Αξίζει να τονίσουμε ότι οι πράκτορες δεν λειτουργούν πάντα με καλές προθέσεις, δηλαδή υπάρχει περίπτωση να υπάρχει ανταγωνισμός μεταξύ τους προκειμένου ο καθένας να

καταφέρει να επιτύχει τους στόχους του. Οι κατηγορίες στις οποίες θα τους χωρίσουμε σύμφωνα με την Nwana (1996) είναι οι εξής[31]:

Πράκτορες Συνεργασίας (Collaborative Agents):

Το χαρακτηριστικό που κάνει τους πράκτορες συνεργασίας να ξεχωρίζουν είναι η αυτονομία και η συνεργασία με άλλους πράκτορες για την υλοποίηση των στόχων που τους έχουν ανατεθεί από τους χρήστες τους. Κινούνται σε ανοιχτά περιβάλλοντα και η λειτουργία τους υπάρχει ένα ενδεχόμενο να περιέχει μια μικρή μορφή μάθησης. Για να υπάρχει μια καλή οργάνωση μεταξύ των πρακτόρων απαιτείται να υπάρχει διαπραγμάτευση. Αυτό θα τους αποφέρει σε ενδεχόμενη συμφωνία για την υλοποίηση των στόχων τους. Τα πλεονεκτήματα που έχουν οι πράκτορες συνεργασίας είναι ότι έχουν την δυνατότητα να λύσουν τα προβλήματα με ταχύτητα τα οποία είναι χρονοβόρα αλλά και για τα οποία χρειάζεται πάνω από ένας πράκτορας. Ένα άλλο πλεονέκτημα είναι ότι μπορούν να συνεργαστούν , να ενωθούν με συμβατικά προγράμματα και με έμπειρα συστήματα κ.λ.π

Πράκτορες Διεπαφής (Interface Agents):

Οι πράκτορες Διεπαφής δίνουν σημασία όπως και οι πράκτορες συνεργασίας στην αυτονομία αλλά και στην μάθηση ώστε να φέρουν εις πέρας τους στόχους που τους έχουν ανατεθεί. Η Pattie Maes (1994) επισημαίνει για αυτούς θα πρέπει να έχουν την έννοια ως « Προσωπικοί βοηθοί που συνεργάζονται με τον χρήστη στο ίδιο περιβάλλον»[27]. Η διαφορά συνεργασίας σε σχέση με τους άλλους πράκτορες είναι ότι δεν ενδείκνυται η γνώσης μιας αποδεκτής γλώσσας επικοινωνίας με ένα χρήστη. Όπως είναι εξίσου σημαντική με άλλους πράκτορες συνεργασίας . Η αρμοδιότητα που έχουν οι πράκτορες διεπαφής είναι να προσφέρουν βοήθεια στον εκάστοτε χρήστη που θέλει να μάθει να χρησιμοποιεί μια εφαρμογή. Αυτό το επιτυγχάνει με την παρακολούθηση και την καταγραφή των ενεργειών του χρήστη μαθαίνοντας τον έτσι . Με αποτέλεσμα να μπορεί να προτείνει βέλτιστες μεθόδους για διάφορες λειτουργίες. Όπως αναφέρει η Maes(1994) η μάθηση του πράκτορα μπορεί να γίνει με τους ακόλουθους τρόπους: 1) να παρατηρεί τον χρήστη, 2) να λαμβάνει θετική και αρνητική ανατροφοδότηση από τον χρήστη, 3) να λαμβάνει σαφές οδηγίες από τον χρήστη και 4) να ζητά συμβουλές από άλλους πράκτορες . Πρέπει να διευκρινίσουμε ότι η συνεργασία εδώ δεν σημαίνει διαδικασία διαπραγμάτευσης αλλά σημαίνει παροχή συμβουλών. Συνοψίζοντας οι πράκτορες διεπαφής χρησιμοποιούνται για την μάθηση άπειρων χρηστών πάνω σε μια εφαρμογή ή για την επιτέλεση καθηκόντων που του έχουν ανατεθεί. Το πλεονέκτημα εδώ είναι ότι χρειάζεται λιγότερος χρόνος και κόπος από τον σχεδιαστή μια εφαρμογής σε σχέση με τις άλλες

εφαρμογές πρακτόρων. Επιπρόσθετα ο πράκτορας έχει την ικανότητα να προσαρμόζεται στις προτιμήσεις των χρηστών όσο περνάει ένα χρονικό περιθώριο.

Πληροφοριακοί Πράκτορες (Information Agents):

Οι πληροφοριακοί πράκτορες δημιουργήθηκαν για να βοηθήνε στην συλλογή την επεξεργασία και την διαχείριση δεδομένων. Προκειμένου να μπουν σε μια σειρά οι πολλές πληροφορίες που έρχονται από το διαδίκτυο . Αυτό που κάνει τους πράκτορες πληροφορίας να ξεχωρίζουν, είναι ότι ορίστηκαν από τι κάνουν αυτοί ενώ οι άλλες κατηγορίες πρακτόρων από το τι είναι αυτοί . Μπορεί να είναι κινητοί αν και συνήθως είναι σταθεροί , προσηλωμένοι σε μια μηχανή αναζήτησης στο διαδίκτυο για να συλλέξουν πληροφορίες. Ένα παράδειγμα που μπορούμε να αναφέρουμε είναι ο πράκτορας Jasper σύμφωνα με τους Davies & Weeks (1995)[32]. Ο οποίος λειτουργεί αντί του χρήστη, έχει την δυνατότητα να αποθηκεύει και να ανακτεί πληροφορίες . Όπως ακόμη να ενημερώνει τους άλλους πράκτορες με διάφορες πληροφορίες βασισμένες όμως από το διαδίκτυο. Ακόμη έχει την ικανότητα να αντιλαμβάνεται τις προτιμήσεις του χρήστη μέσω κάποιων λέξεων κλειδιών που χρησιμοποιεί συχνά. Έτσι αυτό θα έχει ως αποτέλεσμα να μπορεί να προτείνει στον χρήστη κάποιες τυχόν ενδιαφέρουσες σελίδες στο διαδίκτυο.

Πράκτορες Αντίδρασης (Reactive Software Agents):

Οι πράκτορες αντίδρασης ανήκουν σε μια ειδική κατηγορία λόγω ότι δεν διαθέτουν εσωτερικά μοντέλα του περιβάλλοντος όπου είναι. Λειτουργώντας όμως με κύριο γνώμονα στις αντιδράσεις και στα ερεθίσματα που εισέρχονται από την τωρινή κατάσταση στην οποία βρίσκονται από το περιβάλλον. Ένα χαρακτηριστικό που τους κάνει να διαφέρουν είναι ότι είναι απλοί στην λειτουργία τους αλλά και η επικοινωνία με άλλους πράκτορες γίνεται με τους πιο απλουστευμένους τρόπους επικοινωνίας. Ακόμη ένα ιδιαίτερο χαρακτηριστικό που έχουν είναι η κατανομή των αρμοδιοτήτων. Δηλαδή αντιμετωπίζει μέσω της ένωσης πολλών τμημάτων, από τα οποία καθένα λειτουργεί αυτόνομα και έχει την ευθύνη για συγκεκριμένα καθήκοντα.

Κινητοί Πράκτορες (Mobile Agents):

Οι κινητοί πράκτορες μπορούν να κινούνται στο διαδίκτυο , έρχοντας σε επαφή με άλλους υπολογιστές εκτελώντας τα καθήκοντα τα οποία τους έχουν ανατεθεί από τον χρήστη. Μόλις εκτελέσουν όλες τις αρμοδιότητες που τους έχουν ανατεθεί αλλά και να έχουν καταφέρει να τις υλοποιήσουν τότε επιστρέφουν. Την ονομασία πράκτορες την παίρνουν επειδή έχουν

αυτονομία αλλά και συνεργάζονται με άλλους πράκτορες όμως με διαφορετικό τρόπο από ότι οι πράκτορες συνεργασίας που είχαμε αναφέρει προηγουμένως. Τα πλεονεκτήματα που έχουν είναι ότι δεν μένουν σταθεροί στο σύστημα από το οποίο ξεκίνησαν να λειτουργούν αλλά έχουν την δυνατότητα να κινούνται ελεύθερα σε άλλους υπολογιστές του δικτύου. Ενώ η δημιουργία τους αφορά ένα συγκεκριμένο περιβάλλον λειτουργίας, μπορούν να μετακινήσουν μαζί τους την κατάσταση και τον κώδικα τους σε άλλο περιβάλλον λειτουργίας. Όμως ο χρήστης δεν εμπιστεύεται πολύ τους κινητούς πράκτορες, θα πρέπει να είναι πολύ ισχυρό το κίνητρο ώστε να τους αφήσει στο λειτουργικό του σύστημα. Ο λόγος που δεν δείχνει εμπιστοσύνη είναι ότι φοβάται μην είναι κανένας ιός και προκαλέσει ζημιά στο σύστημα.

Υβριδικοί Πράκτορες (Hybrid Agents):

Ο συνδυασμός ανάμεσα των πρακτόρων με εσωτερική κατάσταση και των αντιδραστικών βοήθησε στην δημιουργία των υβριδικών πρακτόρων. Με τα πλεονεκτήματα και των δύο κατηγοριών, χωρισμένο σε δυο κατηγορίες. Η πρώτη αφορά την αντιδραστική συμπεριφορά του πράκτορα και η δεύτερη την συμπεριφορά με εσωτερική κατάσταση. Η ροή ελέγχου μπορεί να είναι με οριζόντιο τρόπο, δηλαδή όλα τα επίπεδα να είναι συνδεδεμένα με τους αισθητήρες εισόδου και με τον μηχανισμό δράσης. Ο άλλος τρόπος είναι κάθετα έχοντας μόνο ένα επίπεδο συνδεδεμένο στους αισθητήρες αλλά και στους μηχανισμούς δράσης.

Νοήμονες Πράκτορες (Intelligent Agents):

Όπως έχει αναφέρει ο Wooldridge[33] ο νοήμονας πράκτορας έχει μια αυτόνομη ενέργεια που μπορεί να ελίσσεται με σκοπό να καταφέρει να πετύχει τους στόχους δημιουργίας του. Πρέπει να τονίσουμε με την έννοια να ελλίσεται εννοούμε την προνοητικότητα, την αντιδραστικότητα και την κοινωνικότητα. Έχει τις εξής ικανότητες όπως δηλαδή να παρακολουθεί το περιβάλλον του και να κάνει αλλαγές μετά από ένα χρονικό όριο αλλά και να συμπεριφέεται ανάλογα με τους στόχους που του έχουν ανατεθεί. Όπως επίσης να υπάρχει αλληλεπίδραση με άλλους πράκτορες ώστε να επιτύχουν τους στόχους δημιουργίας τους.

3.6 ΠΡΟΒΛΗΜΑΤΑ ΛΕΙΤΟΥΡΓΙΑΣ ΤΩΝ ΠΡΑΚΤΟΡΩΝ

Οι πράκτορες όπως είναι λογικό σκεπάζουν ένα ευρύ φάσμα διάφορων δραστηριοτήτων, με αποτέλεσμα να πρέπει να έρθουν με κάποια προβλήματα αντιμέτωποι[34]. Ένα πρόβλημα για παράδειγμα είναι η διαδικασία εύρεσης μια πληροφορίας. Όμως η διαδικασία αυτή θα πρέπει να γίνει γρήγορα αλλά και αποτελεσματικά. Επιπλέον θα πρέπει να καταλαβαίνουν σε βάθος ώστε να μπορούν να βγάλουν συμπέρασμα από διάφορες πληροφορίες. Σε αυτό το πρόβλημα όμως δυσκολεύονται να βρουν λύση οι πράκτορες επειδή ο ογκός πληροφοριών που έχουν να επεξεργαστούν είναι μεγάλος κάθε φορά. Για να περιορίσουν όλο αυτό το μέγεθος που συγκεντρώνεται εφαρμόζουν τις πληροφορίες σύμφωνα με τις προτιμήσεις του χρήστη. Με αυτό τον τρόπο χωρίζουν τις πληροφορίες χωρίς κριτήρια και έτσι έχουν λιγότερο όγκο στοιχείων. Οδηγώντας τους να είναι πιο αποτελεσματικοί και να χρειάζονται λιγότερο χρόνο να επιλύσουν το πρόβλημα. Ακόμη ένα πρόβλημα είναι ότι σε ένα περιβάλλον θα χρειαστεί κάποιος να κάνει κάποια ενέργεια , όπως για παράδειγμα να βγει κάποια ενημέρωση ή ανακοίνωση στον ανάλογο χρήστη. Ο πράκτορας βοηθάει μέσω της αυτοματοποιημένης διαδικασίας όπου ο χρήστης είτε δεν θα το προσέξει είτε δεν θα το θυμηθεί. Λύνοντας έτσι το πρόβλημα που έχουν οι χρήστες με τον φόρτο εργασίας , γιατί τους εμποδίζει να λύσουν ένα πρόβλημα ή δεν έχουν την σωστή αντίληψη να δώσουν προσοχή σε σημαντικά στοιχεία. Κάθε αλγόριθμος που χρησιμοποιείται για εφαρμογή σε ένα σύστημα μέσω των ευφυών πρακτόρων θα πρέπει να λύσει ακόμη κάποια προβλήματα. Όπως για παράδειγμα την σποραδικότητα δηλαδή η δημιουργία αραιών πινάκων μεταξύ χρηστών και αντικειμένων με αποτέλεσμα τη μη επιτυχημένη σύσταση. Από την άλλη έχουμε την επεκτασιμότητα δηλαδή η απαίτηση που έχουν τα συστήματα για τον υπολογισμό των χρηστών που αυξάνεται. Αυτό συμβαίνει γιατί αν ένας αλγόριθμος έχει δημιουργηθεί για συγκεκριμένο αριθμό δεδομένων εισόδων, αν του αυξήσουμε το αριθμο τότε δεν θα μπορεί να ανταποκριθεί. Ένα άλλο πρόβλημα που έχουν να αντιμετωπίσουν είναι η συνωνυμία . Δηλαδή πολλά αντικείμενα μπορεί να έχουν την διαφορετική ονομασία αλλά να αφορούν ίδια πράγματα , δυσκολεύοντας έτσι τα συστήματα στο εντοπισμό των λανθασμένων σχέσεων τους.

Υπάρχουν όμως κάποια συστήματα, ένα από αυτά είναι τα λειτουργικά τα οποία είναι δύσκολο να τα σχεδιάσουμε αλλά και να τα εφαρμόσουμε. Δέχονται μια συγκεκριμένη είσοδο και υπολογίζουν την λειτουργία τους , με αποτέλεσμα να δώσουν συγκεκριμένη παραγωγή. Τέτοιου είδους συστήματα μπορούμε να τα διαχωρίσουμε σε τρεις κατηγορίες . Αρχικά έχουμε τα ανοιχτά συστήματα τα οποία η δομή τους έχει την δυνατότητα να κάνει δυναμικές αλλαγές.

Δηλαδή μολονότι τα τμήματα που το αποτελούν τα γνωρίζουμε από την αρχή , έχουν την δυνατότητα να αλλάζουν κατά διαστήματα και μπορούν να εφαρμοστούν σε οποιαδήποτε χρονική στιγμή. Όπως επίσης από οποιαδήποτε χρήστη με οποιοδήποτε λειτουργικό θέλει. Ένα γνωστό και πολύ χρησιμοποιούμενο παράδειγμα ανοιχτού περιβάλλοντος στην σημερινή εποχή είναι το διαδίκτυο. Η κατασκευή αλλά και ο σχεδιασμός του λογισμικού του διαδικτύου είναι πολύ πολύπλοκος. Αυτό οφείλεται γιατί το δίκτυο των υπολογιστών που είναι συνδεδεμένο αυξάνεται. Ένα άλλο σύστημα είναι τα πολύπλοκα συστήματα, τα οποία τα αντιμετωπίζουμε μέσω των πρακτόρων. Όταν έχει δημιουργηθεί ένα πρόβλημα το οποίο είναι πολύπλοκο για να το αντιμετωπίσουν οι πράκτορες τότε το χωρίζουν σε μικρότερα και πιο απλά μέρη. Επιτυγχάνοντας με αυτόν τον τρόπο να είναι πιο εύκολα να επεξεργαστούν και να τα αναπτύξουν ο κάθε πράκτορας με την κατάλληλη λογική που απαιτείται. Το τελευταίο παράδειγμα ανοιχτού περιβάλλοντος είναι τα συστήματα υπολογιστών . Δηλαδή η σχέση που υπάρχει ανάμεσα στον υπολογιστή και στον χρήστη θα πρέπει να είναι ίδια μεταξύ τους. Το μηχάνημα που χρησιμοποιούμε δεν πρέπει να υπάρχει μόνο να δέχεται εντολές από τον χρήστη αλλά να υπάρχει συνεργασία μεταξύ τους για την υλοποίηση των στόχων. Όμως οι εφαρμογές λογισμικού που θα χρησιμοποιήσουμε πρέπει να είναι αυτόνομες. Να μην χρειάζεται να δώσει εντολή ο χρήστης για το ποιά είναι η βέλτιστη λύση ενός προβλήματος αλλά να μπορεί να καταλάβει από μόνη της. Ένα άλλο σημαντικό στοιχείο είναι να έχουν διορατικότητα , να μην περιμένουν μόλις τελειώσουν μια δουλειά που τους έχει ανατεθεί τι να κάνουν μετά αλλά να προτείνουν οι ίδιες οι εφαρμογές στον χρήστη. Σημαντικό ρόλο παίζει και η προσαρμοστικότητα που θα πρέπει να έχουν. Επειδή θα πρέπει να έχουν γνώση στις προτιμήσεις του χρήστη ώστε να μπορούν να προσαρμοστούν. Όπως ακόμη να υπάρχει και αντίδραση σε περίπτωση που θα υπάρξει μια αλλαγή στο περιβάλλον να μπορεί να το διαχειριστεί[35].

3.7 ΠΛΗΡΟΦΟΡΙΑΚΗ ΥΠΕΡΦΟΡΤΩΣΗ

Οι χρήστες τους διαδικτύου όταν θέλουν να ψάξουν οτιδήποτε στην πρώτη πηγή αναζήτησης που τρέχουν είναι το διαδίκτυο. Όμως το κύριο πρόβλημα που τους κατακλύζει είναι η λεγόμενη πληροφοριακή υπερφόρτωση. Με όλο αυτό το αποτέλεσμα να οδηγεί τον χρήστη

να μην μπορεί να πάρει την σωστή απόφαση για το ποια πληροφορία του είναι χρήσιμη ή ποια είναι άσχετη με αυτό που ψάχνει. Ένας από τους λόγους που οδηγείται ο χρήστης στην αδιέξοδο της σωστής επιλογής και αποφάσης είναι ο τρόπος που παρουσιάζεται η πληροφορία. Επειδή οι πληροφορίες πολλές φορές που υπάρχουν στον παγκόσμιο ιστό, δεν έχουν καλή δόμηση αλλά και οργάνωση όπου θα βοηθούσε αρκετά στην πιο ξεκάθαρη εικόνα τους. Όπως έχουμε αναφέρει το πρόβλημα αυτό λύνεται μέσω των πρακτόρων και συγκεκριμένα των ευφυών πρακτόρων. Αξίζει να σημειωθεί πως οι πράκτορες δεν επιλέγουν αυτοί αντί του χρήστη αλλά προτείνουν στον χρήστη σύμφωνα με τις προτιμήσεις του. Ο ρόλος τους είναι βοηθητικός ώστε να ληφθεί μια απόφαση από τον χρήστη, αλλά και να μειώσουν το όγκο πληροφορίας που δέχεται. Με αποτέλεσμα να μην έχει πολύ πληροφορία να επεξεργαστεί και έτσι μπορεί να παρεί πιο ορθά αποφάσεις. Φιλτράροντας και βάζοντας σε μια πιο οργανωμένη σειρά τις πληροφορίες οι νοήμονες πράκτορες κάνουν πιο εύκολη την αναζήτηση στον χρήστη[7]. Χαρακτηριστικό είναι ότι βελτιώνεται το επίπεδο σε ότι αφορά τις αποφάσεις δηλαδή θα είναι πιο ποιοτικές αποφάσεις. Ένα πείραμα που έχει πραγματοποιηθεί έδειξε ότι οι χρήστες παρότι υπήρχε πολύ διαθέσιμη πληροφορία και τους διακατέχει το άγχος σωστής επιλογή έλεγαν πως είχαν τον έλεγχο. Αυτό γινόταν γιατί τα αποτελέσματα δεν ήταν άμεσα και δεν μπορούσαν να αντιληφθούν ότι όσες περισσότερες πληροφορίες είχαν τόσο δυσκολότερο είναι να επιλέξεις σωστά. Έτσι θεωρούσαν ότι η επιλογή τους ήταν η καταλληλότερη. Με αυτόν τον τρόπο έχουμε μείωση στην ικανοποίηση και στην εμπιστοσύνη του χρήστη. Για να αυξηθούν αυτά τα δύο θα πρέπει η πληροφοριακή υπερφόρτωση να είναι σε επίπεδα διαχειρίσιμα από τον χρήστη. Άρα η χρησιμοποίηση των ευφυών πρακτόρων εξαλείφει την όποια έλλειψη βεβαιότητας επιλογής και ενισχύει την εμπιστοσύνη[35].

3.8 ΑΝΙΧΝΕΥΣΗ ΣΥΓΚΕΚΡΙΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΑΣ

Μέσω του διαδικτύου οι χρήστες έχουν έναν τεράστιο όγκο πληροφοριών που είναι στην διαθεσή τους ανα πάσα στιγμή. Το μόνο που έχουν να κάνουν είναι να πληκτρολογήσουν σε μια μηχανή αναζήτησης αυτό που θέλουν να βρουν. Οι μηχανές αναζήτησης ψάχνουν με βάση τις λέξεις κλειδιά του περιεχομένου της πρότασης. Αυτός ο τρόπος αναζήτησης υστερεί όμως γιατί δεν μπορούν να καταλάβουν ακριβώς για το τι εννοούμε ή τι θέλουμε στην ουσία. Οι πράκτορες ανίχνευσης συγκεκριμένης πληροφορίας προσπαθούν να κατανοήσουν τι θέλουν οι χρήστες ώστε να μπορέσουν να προσφέρουν όσο καλύτερα γίνεται. Ο Etzioni (1996)[36] διατύπωσε ένα μοντέλο πρακτόρων το οποίο δίνει αξία στο υλικό πληροφορίας που υπάρχει στο διαδίκτυο. Την ονόμασε τροφική αλυσίδα των πληροφοριών. Στο σχήμα πυραμίδας όπως είναι σχεδιασμένη στο χαμηλότερο διάστρωμα είναι οι ιστοσελίδες ανθρώπων και οργανισμών. Στο επόμενο διάστρωμα συναντάμε τους δείκτες καταλόγους όπως για παράδειγμα είναι το yahoo, το google. Στο πρώτο διάστρωμα υπάρχουν οι πράκτορες λογισμικού όπως για παράδειγμα μια μηχανή μετα-αναζήτησης. Οι μηχανές αναζήτησης έχουν μεγάλες βάσεις δεδομένων, επειδή είναι αποθηκευμένο το περιεχόμενο από ιστοσελίδες και ευρετήρια. Κάποια προβλήματα που υπάρχουν είναι ότι είναι δύσκολη όλη αυτή η διαδικασία αποθήκευσης, επεξεργασίας αλλά και επαναφοράς δεδομένων μέσα σε μικρό χρονικό διάστημα. Οι μηχανές αναζήτησης πρέπει να ανανεώνουν τις ιστοσελίδες αλλά και να δημιουργούν νέα ευρετήρια για αυτές. Αυτό γινόταν με δύο τρόπους ο πρώτος ήταν με το χέρι και ο δεύτερος με πράκτορες λογισμικού. Με τον πρώτο τρόπο έπρεπε κάποιοι άνθρωποι να ψάχνουν και να ταξινομούν ιστοσελίδες με το χέρι. Έτσι πετύχαιναν βέβαια μια ταξινόμηση που θα ήταν χρήσιμη αλλά θα έβγαζε και νόημα. Από την άλλη όμως η αναζήτηση και η ταξινόμηση δεν γίνονταν αναγκαία πάντα σε βάθος. Ο δεύτερος τρόπος είναι με την χρησιμοποίηση των πρακτόρων λογισμικού, μια από τις ονομασίες τους ήταν ρομπότ του διαδικτύου. Τα ρομπότ αυτά έψαχναν τον ιστότοπο με βάση όλους του συνδέσμους και ταξινομούσαν αυτόματα το περιεχόμενο. Το θέμα ήταν ότι όλες οι μηχανές αναζήτησης δεν κάλυπταν ολόκληρο το περιεχόμενο αλλά ένα μέρος. Για να λυθεί αυτό το πρόβλημα ο Etzioni πρότεινε μια μηχανή μετα-αναζήτησης. Η λειτουργία μιας τέτοιας μηχανής δεν γινόταν συντηρώντας την ίδια βάση δεδομένων αλλά έστελνε ερωτήματα και σε άλλες μηχανές αναζήτησης. Έτσι επέστρεφαν κάποια αποτελέσματα συγκεντρωμένα που ήταν στην διάθεση του χρήστη.

Παρακάτω βλέπουμε και σχηματικά την τροφική αλυσίδα των πληροφοριών :



Σχήμα 3.8 Η τροφική αλυσίδα των πληροφοριών (Etzioni 1996)[36]

3.9 ΣΤΑΣΗ ΧΡΗΣΤΩΝ ΑΠΕΝΑΝΤΙ ΣΤΟΥΣ ΠΡΑΚΤΟΡΕΣ

Η στάση που κρατάνε οι χρήστες απέναντι στους ευφυείς πράκτορες διαμορφώνεται ανάλογα με τις προτάσεις που τους κάνουν. Δηλαδή αν ο εκάστοτε χρήστης καταλάβει ότι οι προτάσεις που του προτείνουν οι ευφυείς πράκτορες δεν ευσταθούν ή είναι απρόβλεπτες και καταλάβει ότι ο ευφυής πράκτορας δεν είναι τόσο ευφυής τότε θα έχει αρνητική στάση απέναντι τους. Αυτό θα έχει ως αποτέλεσμα να μην εμπιστεύεται να επιλέξει αυτές τις προτάσεις που θα του έχει υποδείξει. Όπως έχουμε αναφέρει πάλι η πληροφοριακή υπερφόρτωση δεν αφήνει τον χρήστη να δει με προσοχή και να επεξεργαστεί τις πληροφορίες που δέχεται. Έτσι ο χρήστης μένει επαναπαυμένος σε μια πρόταση χωρίς να την ερευνήσει περαιτέρω για να δει αν συμφωνεί ή όχι μαζί της[7].

Μια έρευνα που διεξήχθη από την Pattie Maes[31], η οποία εφαρμόστηκε σε πραγματικούς χρήστες ήταν η χρησιμοποίηση ευφύων πρακτόρων. Το αποτέλεσμα ήταν πολύ θετικό γιατί οι χρήστες δεν έφεραν αντίρρηση στην χρησιμοποίησή τους. Αυτό φυσικά συνέβαλε ότι ήταν ένας τρόπος να καταπολεμήσουν την πληροφοριακή υπερφόρτωση. Όπως και οτιδήποτε άλλος τρόπος να υπήρχε που θα μπορούσε να τους βοηθήσει να μπορούν να διαχειριστούν τον μεγάλο όγκο πληροφοριών θα το χρησιμοποιούσαν. Του είχε γίνει συνήθεια του χρήστη ο ευφυής πράκτορας να τον χρησιμοποιεί όταν μαζευόταν πολύ πληροφοριακή υπερφόρτωση. Ακόμη τον συμβουλευόταν και ένιωθε βολικά στον να τον χρησιμοποιεί. Όσον αφορά για ένα χρήστη ο οποίος θα θέλει να δαπανήσει περισσότερο χρόνο στο να ψάξει μια πληροφορία από πολλές πηγές, θα είναι δύσκολο να συμφωνήσει σε μια πρόταση που θα του προταθεί από τον πράκτορα. Αυτό συμβαίνει για τρεις λόγους[7]:

- Αρχικά οι προτάσεις των ευφύων πρακτόρων δεν είναι οι καλύτερες επιλογές που περιμένει ο χρήστης. Ο κάθε χρήστης για να δείξει στον πράκτορα τις προτιμήσεις του όπως είναι λογικό θα χρειαστεί πολύ χρόνο και κόπο. Με αποτέλεσμα από την στιγμή που ο πράκτορας δεν έχει την πλήρη εικόνα για τα τις προτιμήσεις του χρήστη, δεν θα μπορεί να προτείνει την καλύτερη επιλογή. Έτσι ο χρήστης που θέλει να ψάξει όσο το περισσότερο μπορεί τότε θα απορρίψει τις προτάσεις και θα συνεχίσει την αναζήτηση.
- Ο χρήστης μπορεί να αμφισβητήσει την φερεγγυότητα λειτουργίας ενός ευφύου πράκτορα ή για τα κίνητρα που χρησιμοποιεί λόγω της επιπλέον γνώσης που έχει.
- Η νοητική προσπάθεια ακόμη και αν μεταφέρεται στην επεξεργασία πληροφορίας δεν μπορεί να εγγωθηθεί κανείς στην καλύτερη και ακριβή επιλογή. Σε περίπτωση που οι

επιλογές του χρήστη είναι ίδιες με τις προτεινόμενες αλλά και όσο μεγαλώνει ο αριθμός τους τότε γίνεται ακόμη πιο δύσκολο να επιλέξει. Όλο αυτό του δημιουργεί μια αγχωτική στάση ως προς τι να επιλέξει.

ΚΕΦΑΛΑΙΟ 4

ΣΥΣΤΗΜΑΤΑ ΠΟΛΛΑΠΛΩΝ ΠΡΑΚΤΟΡΩΝ

4.1 ΜΕΤΑ- ΠΡΑΚΤΟΡΕΣ

Η σημασία των μετά-πρακτόρων (meta-agents) στηρίζεται στην τεχνική της μετα-συλλογιστικής. Με την τεχνική αυτή το σύστημα λειτουργεί με βάση την λογική του και βοηθάει στην ύπαρξη αλληλεπίδρασης ανάμεσα στους πράκτορες. Επιπλέον οι μετά-πράκτορες έχουν την δυνατότητα να σχεδιάζουν ενέργειες και να διατηρούν την κατάσταση πληροφορίας των πρακτόρων. Ακόμη μπορούν να ελέγχουν και να κατευθύνουν την συμπεριφορά των πρακτόρων που θα έχουν μελλοντικά. Ένα σημαντικό στοιχείο είναι ότι υπάρχει η δυνατότητα να συνεργαστούν οι ευφυείς πράκτορες με τους μετά-πράκτορες σε ένα δίκτυο. Με αυτό τον τρόπο μπορούν να ελέγξουν αλλά και να βελτιώσουν την απόδοσή τους. Όπως ακόμη να υπολογίσουν τις καλύτερες αποφάσεις που θα πάρουν , ανάλογα με την απόδοση που έχουν οι πράκτορες μέσα σε ένα περιβάλλον. Σε ένα οργανισμό οι μετά-πράκτορες λειτουργούν ως ένα σημαντικό εργαλείο , που δίνουν την δυνατότητα να συνδυαστούν διάφορες πληροφορίες αλλά και να παρθεί η καλύτερη απόφαση[38].

4.2 ΕΙΣΑΓΩΓΗ ΣΤΑ ΠΟΛΥΠΡΑΚΤΟΡΙΚΑ ΣΥΣΤΗΜΑΤΑ

Στα πολυπρακτορικά συστήματα ή αλλιώς όπως το αναφέραμε στην αρχή συστήματα πολλαπλών πρακτόρων (Multi-Agent Systems- MAS) για την εκτέλεση κάποιων διεργασιών χρησιμοποιούνται ευφυείς πράκτορες. Για παράδειγμα μια διεργασία μπορεί να είναι η ανίχνευση μιας συγκεκριμένης πληροφορίας στο διαδίκτυο. Πρέπει να τονίσουμε ότι υπάρχει ένα καινούργιο στοιχείο στην αλληλεπίδραση των πολυπρακτορικών συστημάτων. Το στοιχείο αυτό είναι ότι η αλληλεπίδραση είναι δυναμική καθορισμένη, που σημαίνει ότι το σύστημα δεν πάυει να λειτουργεί σωστά σε περιπτώσεις που δεν είχαν καθοριστεί από την αρχή. Αυτή η τεχνολογία των πρακτόρων δίνει το έναυσμα για ένα νέο είδος προγραμματισμού πολλαπλών πρακτόρων που θα αφορά τους πράκτορες. Για την δημιουργία τέτοιων συστημάτων συναντάμε κάποια προβλήματα , τα οποία οφείλονται στην δυναμική φύση των πρακτόρων και στην αυτονομία που έχουν[39].

4.3 ΕΝΝΟΙΑ ΤΩΝ ΠΟΛΥΠΡΑΚΤΟΡΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Ένα πολυπρακτορικό σύστημα αποτελείται από ένα σύνολο πρακτόρων που μεταξύ τους υπάρχει αλληλεπίδραση, συνεργασία , συντονισμός κ.λ.π. Τα πολυπρακτορικά συστήματα μαζί με την κατανομημένη επίλυση προβλημάτων είναι μέρος της κατανομημένης τεχνητής νοημοσύνης. Αυτή η επίλυση προβλημάτων έχει ως κύριο στόχο τον τρόπο με το οποίο λύνονται τέτοιου είδους προβλήματα. Δηλαδή η συνεργασία προκειμένου να μοιραστεί η γνώση για το πρόβλημα αλλά και για τις λύσεις που μπορούν να βρεθούν. Στα πολυπρακτορικά συστήματα υπάρχουν πολλοί πράκτορες που ενεργούν όλοι μαζί προκειμένου να λύσουν ένα πρόβλημα. Είναι καλύτερο να συνεργάζονται πολλοί πράκτορες για να λύσουν ένα πρόβλημα παρότι ένας, επειδή οι περισσότεροι έχουν περισσότερες λύσεις. Όπως έχουμε αναφέρει πάλι σε αυτά τα συστήματα η συνεργασία είναι δυναμική και υπάρχει αυτονομία, πράγμα που σημαίνει ότι αυτοί αποφασίζουν για τον τρόπο αλλά και τον

χρόνο που θα συνεργαστούν. Κάποιους στόχους που μπορεί να έχει ένα πολυπρακτορικό σύστημα είναι[40]:

- Την αποδοτική επίλυση πολύπλοκων προβλημάτων από έναν πράκτορα
- Την επίλυση προβλημάτων όπου υπάρχει κατανεμημένη εμπειρογνωμοσύνη.
- Την επίλυση προβλημάτων όπου εξαρχής ήταν κατανεμημένα.
- Την λειτουργία των ήδη υπαρχόντων συστημάτων ώστε να μην χρειαστεί καμία αλλαγή και να είναι εύκολη η χρησιμοποίησή τους.

Οι πράκτορες δουλεύουν αυτόνομα ανταλλάσσοντας πληροφορίες ώστε να υλοποιήσουν τους στόχους τους. Για να βρουν την κατάλληλη λύση πολλές φορές χωρίζουν το αρχικό πρόβλημα σε υποπροβλήματα , μετά ενώνουν όλες τις λύσεις που έχουν βρει και μέσω του συνδυασμού αυτού προκύπτει η τελική λύση. Για να επέλθει συμφωνία μεταξύ τους υπάρχει κάποια γλώσσα επικοινωνίας. Κάποια από τα χαρακτηριστικά ενός πολυπρακτορικού συστήματος είναι [41]:

- Κανένας από τους πράκτορες δεν έχει όλη την πληροφορία
- Τα δεδομένα που υπάρχουν τις περισσότερες φορές είναι ομαδοποιημένα
- Ο κεντρικός έλεγχος του συστήματος δεν υπάρχει
- Οι λειτουργίες γίνονται με ασύγχρονο τρόπο

Όπως θα μας φαίνεται λογικό θεωρούμε πάντα ότι όσο περισσότεροι πράκτορες υπάρχουν τότε είναι ακόμη πιο εύκολο να καταφέρουν να υλοποιήσουν την δουλειά που τους έχει ανατεθεί παρά να ήταν ένας πράκτορας μόνος του. Αυτό όμως δεν συμβαίνει πάντα επειδή όταν υπάρχουν πολλοί πράκτορες μπορεί να προκύψει πρόβλημα σε ότι αφορά την επικοινωνία μεταξύ τους. Στην σχεδίαση ενός πολυπρακτορικού συστήματος υπάρχουν κάποιες λεπτομέρειες που θέλουν προσοχή ώστε να βρεθεί ο καταλληλότερος τρόπος για να αντιμετωπιστεί. Τα προβλήματα αυτά αφορούν τον τρόπο αλληλεπίδρασης που έχουν μεταξύ τους αλλά και την επικοινωνία.

Το πρόβλημα της αλληλεπίδρασης είναι πως θα καθοριστεί, θα σχεδιαστεί, θα κατανεμηθεί και πως θα τεθούν οι λύσεις του σε μια ομάδα που αποτελείται από ευφυείς πράκτορες. Ένα ακόμη πρόβλημα που προκύπτει είναι πως θα υπάρξει μια κοινή συμφωνία ανάμεσα στις διαφορετικές απόψεις που θα έχουν οι πράκτορες μεταξύ τους. Από την στιγμή που αυτό το σύστημα είναι τόσο πολύπλοκο υπάρχει πιθανότητα να επικρατεί μια αταξία, όπου όμως θα πρέπει να υπάρξει τρόπος ώστε να μπορεί να προβλεφθεί και να αποφευχθεί.

Το πρόβλημα της επικοινωνίας που θα πρέπει να αντιμετωπιστεί έχει να κάνει πότε επικοινωνούν οι πράκτορες αλλά και τι πληροφορία ανταλλάσσουν. Επίσης θα πρέπει να οριστεί με ποιούς άλλους πράκτορες υπάρχει επικοινωνία, ποιες γλώσσες και ποια πρωτόκολλα θα χρησιμοποιηθούν [42].

4.4 ΔΟΜΙΚΑ ΣΤΟΙΧΕΙΑ ΕΝΟΣ ΣΠΠ

Παλιότερα πολλά Συστήματα Πολλαπλών Πρακτόρων έδιναν βάση για την λύση κάποιων προβλημάτων. Όπως για παράδειγμα η σωστή ανάθεση εργασιών στους ευφυείς πράκτορες, στην σωστή οργάνωση όλης της ομάδας όπως ακόμη στις επικοινωνίες και στις διαπραγματεύσεις. Όμως έτσι δεν ήταν εύκολο να προσαρμοστούν σε οποιαδήποτε νέα αλλαγή. Για αυτόν τον λόγο στην σημερινή εποχή έχει δημιουργηθεί μια σωστή υποδομή ώστε να μπορεί το σύστημα να προσαρμοστεί πιο εύκολα σε αλλαγές ακολουθώντας κάποια στοιχεία οργάνωσης. Στο περιβάλλον του πολυπρακτορικού συστήματος η σύνδεση πληροφορίας και ελέγχου ανάμεσα στους πράκτορες όπως και η ανάθεση λύσης προβλημάτων μεταξύ τους ονομάζεται δομή. Με αυτό τον τρόπο η δομή δίνει αντίληψη υψηλού επιπέδου σε κάθε πράκτορα πως μια ομάδα δίνει λύσεις στα προβλήματά της, αλλά και ποιο ρόλο έχει κάθε πράκτορας μέσα στην ομάδα. Οι πράκτορες για να βεβαιωθούν ότι ικανοποιούν τις απαραίτητες προϋποθέσεις οι οποίες είναι σημαντικές για την λύση ενός προβλήματος σύμφωνα με τον Corkill και Lesser είναι οι εξής[41]:

- **Κάλυψη:** Στις ικανότητες ενός πράκτορα πρέπει να βρίσκεται ένα κομμάτι του συνολικού προβλήματος.
- **Συνδεσιμότητα:** Πρέπει να υπάρχει αλληλεπίδραση ανάμεσα στους πράκτορες ώστε οι δραστηριότητές τους να αποτελούν μια ολοκληρωμένη λύση.
- **Ικανότητα:** Η συνδεσιμότητα και η κάλυψη πρέπει να είναι εφικτές στα υπολογιστικά και επικοινωνιακά όρια, όπως και τα λεπτομερή στοιχεία για την αξιοπιστία της ομάδας.

Στην ουσία ο στόχος της δομής είναι να καθορίζει ρόλους και σχέσεις προκειμένου να πετύχει τις προηγούμενες προϋποθέσεις που αναφέραμε πιο πάνω.

Σύμφωνα με τον Gasser (1986)[43], έδωσε τον ορισμό στην οργάνωση ως « ένα ιδιαίτερο σύνολο από ορισμένα και αόριστα ζητήματα τα οποία αφορούν τις πεποιθήσεις και τις

δράσεις μέσα από τις οποίες οι πράκτορες έχουν άποψη για τους άλλους πράκτορες». Δηλαδή σε ένα πρόβλημα μπορούν να συνεργάζονται δυο πράκτορες μεταξύ τους , όμως σε κάποιο άλλο πρόβλημα που μπορεί να ανατεθεί για να λυθεί υπάρχει μεγάλη πιθανότητα να μην είναι στην ίδια ομάδα.

Σύμφωνα με τον Malone (1990)[43], που ασχολείται με την οργάνωση ομάδας έχει αναφέρει ότι «Μια ομάδα πρακτόρων είναι οργανωμένη αν οι τελευταίοι είναι συνδεδεμένοι με κάποιο τρόπο όπου οι συνδυασμένες ενέργειές τους αποδίδουν καλύτερα από το να μην υπήρχε αυτή η σύνδεση. Η οργανωμένη ομάδα αποτελείται από: μια ομάδα πρακτόρων, ένα σύνολο ενεργειών των πρακτόρων, και ένα σύνολο από στόχους ή κριτήρια εκτίμησης με τα οποία οι συνδυασμένες ενέργειες των πρακτόρων εκτιμούνται ».

4.5 ΤΡΟΠΟΣ ΕΠΙΚΟΙΝΩΝΙΑΣ ΤΩΝ ΕΥΦΥΩΝ ΠΡΑΚΤΟΡΩΝ

Οι πράκτορες μέσω της επικοινωνίας καταφέρνουν να ανταλλάξουν πληροφορίες και να διευθύνουν την δράση τους. Τα είδη επικοινωνίας που υπερισχύουν είναι δύο , η άμεση ανταλλαγή μηνυμάτων και η συγκέντρωση δεδομένων σε ένα κοινό σημείο λεγόμενος ως μαυροπίνακας. Για οποιονδήποτε τύπο συμφωνίας στην ανάπτυξη πολυπρακτορικών συστημάτων είναι η πραγμάτωση επικοινωνίας μεταξύ των πρακτόρων. Η επικοινωνία αυτή χρειάζεται την ύπαρξη τριών διαφορετικών επιπέδων. Το πιο χαμηλό επίπεδο έχει να κάνει με τον τρόπο διασύνδεσης , το μεσαίο επίπεδο έχει να κάνει με την σύνταξη και την μορφή των μηνυμάτων. Τέλος το ανώτερο επίπεδο έχει να κάνει με την ανάλυση και την ερμηνεία της σημασίας των λέξεων. Υπάρχουν κάποιοι μέθοδοι συμφωνίας οι οποίοι καθορίζουν ελάχιστη ή καθόλου επικοινωνία μεταξύ των ευφυών πρακτόρων. Σε περίπτωση που δεν χρειάζεται να συνεργαστούν οι ευφυείς πράκτορες για να υλοποιήσουν τους στόχους τους και δεν έρχονται σε σύγκρουση τα συμφέροντα τους τότε η επικοινωνία μεταξύ τους δεν είναι αναγκαία. Υπάρχουν κάποιες φορές όπου μια μικρή επικοινωνία με μορφή ενός συγκεκριμένου ορίου συνόλου σημάτων είναι ικανό ώστε να επέλθει συμφωνία μεταξύ των πρακτόρων σύμφωνα με τον Georgeff, 1983[44]. Υπάρχουν και κάποιοι άλλοι πράκτορες που μπορούν να συντονιστούν παρακολουθώντας μόνο τις αλλαγές που προκαλούν άλλοι πράκτορες σε ένα περιβάλλον. Με αυτό τον τρόπο οι πράκτορες προσπαθούν να

καταλαμβάνουν τι κίνηση έκαναν αλλά και πως κινήθηκαν βλέποντας τις ενέργειες τους στο περιβάλλον (Rosenschein και Breese, 1989) [45].

Ο τύπος διασύνδεσης, ο οποίος έχει προταθεί για την συνεργασία μεταξύ των πρακτόρων περιέχει δυο βασικά είδη κατηγοριών. Η πρώτη είναι τα συστήματα μαυροπίνακα και τα συστήματα ανταλλαγής μηνυμάτων.

Ο μαυροπίνακας έχει την μορφή κοινής μνήμης (Nii, 1986)[46] όπου οι πράκτορες έχουν την δυνατότητα να γράψουν κάτι ή να τον διαβάσουν. Ο μαυροπίνακας χωρίζεται σε πεδία όπου οι πράκτορες αποκτούν δικαιώματα με βάση την ικανότητα που έχουν να διαβάζουν περιοχές. Αυτό θα έχει ως αποτέλεσμα κάποιοι να έχουν την ικανότητα να μπορούν να διαβάσουν και να γράψουν σε περισσότερα πεδία σε σχέση με κάποιους άλλους. Σε ένα σύστημα μαυροπίνακα υπάρχει ένας ίδιος χώρος εργασίας για τους πράκτορες που υπάρχουν στο σύστημα. Σε αυτόν τον χώρο γίνεται η ανταλλαγή αποτελεσμάτων ή καθορίζονται οι εργασίες. Ακόμη στον χώρο αυτό από την στιγμή που αποθηκεύεται κάτι υπάρχει η δυνατότητα να έχουν πρόσβαση όσοι πράκτορες συμμετέχουν στον σύστημα. Παρακάτω βλέπουμε σχηματικά την αρχιτεκτονική του μαυροπίνακα.



Σχήμα 4.1 Αρχιτεκτονική μαυροπίνακα(Nii)

Ο πιο ευρέως γνωστός τρόπος επικοινωνίας που έχουμε στα πολυπρακτορικά συστήματα είναι η ανταλλαγή μηνυμάτων. Το κάθε μήνυμα δεν έχει το ίδιο περιεχόμενο αλλά είναι διαφορετικό από σύστημα σε σύστημα. Στα μηνύματα πρέπει να διαμορφωθεί ένα πρωτόκολλο γλώσσας, σύνταξης αλλά και μεταφοράς του μηνύματος. Ακόμη πρέπει να καθοριστούν οι διαφορετικοί τύποι μηνυμάτων. Ο Rosenschein 1986 [47] ανέφερε κάποιους κινδύνους που υπάρχουν όταν στέλνεται μέσω μηνυμάτων το σχέδιο δράσης ενός πράκτορα. Ο λόγος ύπαρξης των κινδύνων αυτών υφίσταται γιατί στην ανταλλαγή μεγάλων μηνυμάτων η μετάδοσή τους καθυστερεί έτσι δημιουργείται ανάγκη για πολλούς διαύλους επικοινωνίας. Επιπλέον όταν έχουμε δυναμικό περιβάλλον και φτάσει το σχέδιο υπάρχει πιθανότητα να μην εκπροσωπεί πια τις επιθυμίες του αποστολέα. Όταν έχουμε αυτόνομους πράκτορες αλλά

και σε άλλες περιπτώσεις , δεν είναι σίγουρο αυτοί που θα λάβουν το σχέδιο δράσης ότι θα το ενστερνιστούν. Οι πράκτορες έχουν και άλλη δυνατότητα πέρα από την ανταλλαγή ενός απλού μηνύματος μπορούν να βάλουν σε ένα μήνυμα την δυνατότητα να επιδρά στις γνώσεις και την γνώμη του πράκτορα που δέχεται το μήνυμα. Σε περίπτωση που οι πράκτορες ακολουθήσουν την BDI λογική οι γνώμες τους μπορούν να αλλάξουν ανάλογα με τις αλλαγές στο περιβάλλον ή στην συμπεριφορά άλλων πρακτόρων. Έχουν δημιουργηθεί κάποιες σύγχρονες γλώσσες όπως για παράδειγμα η γλώσσα χειρισμού ερωτοαπαντήσεων (Knowledge Query Manipulation Language, KQML) ή την γλώσσα επικοινωνίας πρακτόρων (Agent Communication Language, ACL). Έτσι υπάρχει η δυνατότητα δράσης μέσω του λόγου που προτάθηκε από τον Searle το 1969 [48]. Οι πράκτορες στα συστήματα ανταλλαγής μηνυμάτων πέρα από την ανταλλαγή πληροφορίας συνεργάζονται μέσω μηνυμάτων όπου τα στέλνουν μεταξύ τους σύμφωνα με γλώσσες υψηλού επιπέδου. Επίσης στα συστήματα αυτά υπάρχει η δυνατότητα δημιουργίας περίπλοκων μοντέλων συνεργασίας ανάμεσα στους πράκτορες. Πρέπει να τονιστεί ότι υπάρχει μεγάλη ικανότητα στην ανταλλαγή πληροφοριών σε σχέση με τα συστήματα μαυροπίνακα. Επίσης τον τύπο της επικοινωνίας τον χωρίζουμε σε δύο κατηγορίες , τον σύγχρονο και τον ασύγχρονο. Στον σύγχρονο τύπο ο πράκτορας που έχει θέσει μια ερώτηση σταματάει την λειτουργία του μέχρι να πάρει κάποια απάντηση. Αντίθετα στον ασύγχρονο τύπο ο πράκτορας δεν σταματάει την λειτουργία παρόλο που η απάντηση μπορεί να έρθει οποιαδήποτε στιγμή μετά το πέρας του χρόνου υποβολής της.

4.6 ΤΥΠΟΙ ΑΛΛΗΛΕΠΙΔΡΑΣΗΣ ΕΥΦΥΩΝ ΠΡΑΚΤΟΡΩΝ

Αρχικά οι Jennings, Sycara και Wooldridge (1998) θεωρούσαν ότι ο συντονισμός, η συνεργασία και οι διαπραγματεύσεις αφορούσαν τους τύπους αλληλεπίδρασης των ευφύων πρακτόρων. Επειδή οι πράκτορες για να λειτουργήσουν σωστά και να αποφύγουν τις συγκρούσεις θα χρειαστεί να υπάρξει ο συντονισμός. Όπως ακόμη για να καταφέρουν να υλοποιήσουν ένα στόχο χρειάζεται συνεργασία μεταξύ τους , αλλά και διαπραγματεύσεις. Όστε να υπάρχει μια κοινή συμφωνία από όλους προκειμένου να υλοποιήσουν τον αρχικό τους στόχο. Στην επόμενη παράγραφο θα αναφέρουμε τους τύπους αυτούς πιο αναλυτικά.

Ο πρώτος τύπος είναι ο Συντονισμός , ο Malone το 1990 [49] υποστήριξε την ύπαρξη δυο τρόπων συντονισμού τις ιεραρχίες και τις αγορές. Οι ιεραρχίες λειτουργούν στην άμεση επιτήρηση των εργασιών. Αντίθετα οι αγορές λειτουργούν με βάση να επέλθει συμφωνία ανάμεσα στους πράκτορες , όπου όμως η πρόσβαση είναι από διαφορετικούς πόρους. Ο Jennings 1993 [26] υποστήριξε ότι οι πράκτορες επιτυγχάνουν τον συντονισμό μέσω των συμφωνιών που έχουν κάνει μεταξύ τους. Πολλές φορές ο Συντονισμός έχει ήδη υπάρξει πριν να έχουν ενεργήσει οι πράκτορες για τον σχεδιασμό της δράσης τους. Όμως σε κάποιες περιπτώσεις υπάρχει η δυνατότητα να γίνει και αργότερα. Σύμφωνα με τον Durfee (1989) [50] υποστήριξε ότι η Συνεργασία όπου ανήκει στον δεύτερο τύπο , θα πρέπει να έχει κάποιους στόχους . Οι οποίοι είναι: να υπάρχει μείωση της αλληλεπίδρασης μεταξύ των εργασιών, να υπάρχει αύξηση της ολοκλήρωσης των εργασιών με την διπλή εκτέλεσή τους, αύξηση τους εύρους των εργασιών που έχουν την δυνατότητα να εκτελεστούν με την κοινή χρήση πόρων και τέλος αύξηση του ρυθμού αποπεράτωσης των εργασιών σε σχέση με την εκτέλεση των υπο-εργασιών. Οι πράκτορες οι οποίοι λύνουν τέτοιου είδους προβλήματα είναι οι συνεργατικοί γιατί δεν έχει την δυνατότητα να τα λύσει ένας πράκτορας. Έτσι χωρίζουν το αρχικό πρόβλημα σε υποπροβλήματα και από εκεί προκύπτει η λύση τους. Ο Decker (1987)[51] σε ότι αφορά το θέμα της συνεργασίας υποστήριξε ότι οι πράκτορες μπορεί να είναι από εντελώς συνεργάσιμοι μέχρι ανταγωνιστικοί. Οι πράκτορες οι οποίοι είναι συνεργάσιμοι εκτελούν μεν τις εργασίες τους όμως υπάρχει μεγάλο επικοινωνιακό κόστος αλλά και με καθυστέρηση. Αυτό γίνεται γιατί υπάρχει περίπτωση να αλλάξουν τους στόχους τους ώστε να καλύψει τις τυχόν ανάγκες άλλων πρακτόρων. Οι ανταγωνιστικοί πράκτορες δεν είναι συνεργάσιμοι όταν δεν τους συμφέρει, και μπλοκάρουν ο ένας τους στόχους των άλλων. Ο τρίτος τύπος είναι η Διαπραγμάτευση όπου βοηθάει στον συντονισμό μια ομάδας που αποτελείται από ευφυείς πράκτορες. Αξίζει να τονίσουμε ότι οι Durfee, Lesser και Corkill (1989)[41] αναφέρουν ότι η διαπραγμάτευση είναι ένας τρόπος βελτίωσης σε μια συμφωνία . Αντίθετα οι Jennings, Sycara και Wooldridge (1998)[26] αναφέρουν ότι η διαπραγμάτευση είναι μια μέθοδος η οποία συντονίζει αλλά και λύνει τυχόν συγκρούσεις. Τα χαρακτηριστικά τα οποία είναι αναγκαία για την χρήση διαπραγμάτευσης ενός Συστήματος Πολλαπλών Πρακτόρων είναι: Σε περίπτωση που θα υπάρξει σύγκρουση θα λυθεί μέσω της διαπραγμάτευσης. Όπως ακόμη η λειτουργία κάποιων πρακτόρων οι οποίοι κοιτάνε μόνο τον εαυτό τους, που έχουν μια λειτουργία με όρια και ελάχιστη πληροφόρηση. Όταν υπάρχει ανάγκη για να αυτοματοποιηθεί η διαπραγμάτευση έχουμε τρία θέματα που θα πρέπει να καθορίσουμε:

- Αρχικά έχουμε τα **πρωτόκολλα διαπραγμάτευσης** τα οποία αποτελούνται από ένα σύνολο κανόνων που καθορίζουν ποιοι ευφείς πράκτορες θα διαπραγματευτούν .
- **Αντικείμενα διαπραγμάτευσης** είναι τα πράγματα-αντικείμενα για τα οποία θα γίνει διαπραγμάτευση.
- **Μοντέλα λήψης αποφάσεων των πρακτόρων** είναι κάποια μοντέλα που χρησιμοποιούνται μαζί με το πρωτόκολλο από τους πράκτορες για να προστατεύσουν τα συμφέροντά τους.

4.7 ΠΡΩΤΟΚΟΛΛΟ ΕΠΙΚΟΙΝΩΝΙΑΣ CONTRACT NET

Το πρωτόκολλο Contract Net μπορεί να χαρακτηριστεί ως υψηλού επιπέδου γιατί η συνεργατική του ενέργεια σε συστήματα επίλυσης προβλημάτων ήταν επιτυχής. Το σημαντικό στοιχείο για να έχουμε μια επιτυχημένη συνεργατική επίλυση προβλημάτων είναι η διαπραγμάτευση. Το Contract Net θεωρείται ότι συμβάλλει στην σωστή διαχείριση των διαπραγματεύσεων ως ένα πλαίσιο. Στην ορολογία του πρωτοκόλλου ένας πράκτορας ονομάζεται ως κόμβος και είναι ένα αυτόνομο σύστημα επίλυσης προβλημάτων και μπορεί να επικοινωνεί με άλλους πράκτορες μέσω ανταλλαγής μηνυμάτων. Όταν ένας κόμβος (πράκτορας) θελήσει να εκτελέσει κάποια ενέργεια αλλά δεν μπορεί να την εκτελέσει ο ίδιος τότε κάνει μια ανακοίνωση. Όταν δεν γνωρίζει για τις δυνατότητες των υπόλοιπων κόμβων τότε θα βγάλει μια ανακοίνωση γενική. Διαφορετικά η ανακοίνωση μπορεί να είναι περιορισμένη ή και προσωπική. Στην συνέχεια δέχεται αιτήσεις από τις οποίες άλλες τις δέχεται και άλλες τις απορρίπτει[52].

4.8 ΤΟ ΠΡΩΤΟΚΟΛΛΟ ΚΑΙ ΓΛΩΣΣΑ ΕΠΙΚΟΙΝΩΝΙΑΣ KQML

Η KQML μπορεί να χαρακτηριστεί ως υψηλού επιπέδου γλώσσα επικοινωνίας των πρακτόρων που με τον συνδυασμό πρωτοκόλλων στηρίζει την κοινή χρήση αλλά και την ανταλλαγή πληροφοριών μεταξύ των ευφυών πρακτόρων. Η KQML χρησιμεύει σε μια εφαρμογή ως μια γλώσσα που έχει την δυνατότητα να αλληλεπιδρά ανάμεσα σε πράκτορες για την επίλυση περίπλοκων προβλημάτων. Πρέπει να τονίσουμε ότι η γλώσσα KQML είναι ανεξάρτητη από την γλώσσα περιεχομένου(π.χ Prolog, Sql). Όπως επίσης είναι ανεξάρτητη από το πρωτόκολλο μεταφοράς (π.χ TCP/IP) και ανεξάρτητη από την οντολογία του περιεχομένου. Η γλώσσα αυτή έχει ένα σύνολο που μπορεί να επεκταθεί το οποίο αποτελείται από εκτελεστές. Οι οποίοι καθορίζουν ποιες λειτουργίες είναι επιτρεπτές τις οποίες μπορούν να εφαρμόσουν οι πράκτορες στην γνώση αλλά και τους στόχους άλλων πρακτόρων. Αξίζει να σημειωθεί ότι οι εκτελεστές αποτελούν την βάση πάνω στην οποία μπορούν να εξελιχθούν μοντέλα επικοινωνίας. Ακόμη η KQML έχει μια αρχιτεκτονική για κοινή χρήση γνώσης μέσω μιας ειδικής κατηγορίας πρακτόρων οι οποίοι διευθύνουν την αλληλεπίδραση άλλων πρακτόρων. Όταν αναφέρουμε ότι οι πράκτορες μιλούν KQML, εννοούμε ότι χρησιμοποιούν την βιβλιοθήκη των επικοινωνιακών ενεργειών με την ορισμένη τους ερμηνεία[53].

4.8.1 Τα χαρακτηριστικά της KQML και το περιβάλλον των πρακτόρων

Όπως έχουμε αναφέρει πάλι η KQML αποτελεί ταυτόχρονα γλώσσα και πρωτόκολλο επικοινωνίας ανάμεσα στους πράκτορες. Μπορούμε να θεωρήσουμε για αυτήν ότι περιέχει τρία διαφορετικά επίπεδα τα οποία είναι: α) το επίπεδο περιεχομένου, β) το επίπεδο μηνυμάτων και γ) το επίπεδο επικοινωνίας.

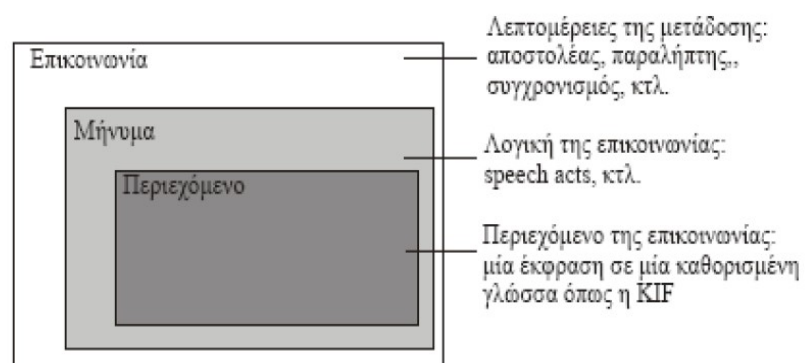
Στο επίπεδο επικοινωνίας μεταφέρονται τα στοιχεία του αποστολεά και του παραλήπτη τα οποία είναι μοναδικά στην επικοινωνία. Όπως ακόμη χαρακτηρίζεται αν η επικοινωνία είναι σύγχρονη ή ασύγχρονη.

Στο επίπεδο μηνυμάτων γίνεται η κωδικοποίηση των μηνυμάτων που μεταφέρει ένας πράκτορας σε έναν άλλο. Είναι σημαντικό μέρος της γλώσσας KQML γιατί ορίζονται ποια

είδη επικοινωνίας μπορεί να υποστηρίξει ο πράκτορας που μιλάει την γλώσσα. Υπάρχουν κάποια μηνύματα τα οποία το περιεχόμενο τους δεν είναι φανερό έτσι για αυτό τον λόγο σε αυτό το επίπεδο υπάρχουν χαρακτηριστικά που περιγράφουν την γλώσσα του περιεχομένου. Έτσι υπάρχει η δυνατότητα στις εφαρμογές που χρησιμοποιούν την KQML να αναλύουν και να δίνουν τα μηνύματα σωστά και σε περίπτωση που δεν είναι επιτρεπτό να διαβαστεί το περιεχόμενό τους.

Το επίπεδο περιεχομένου συνδέεται με το περιεχόμενο του μηνύματος που στέλνεται. Αυτό ορίζεται ανάλογα με την εφαρμογή και μπορεί να είναι οτιδήποτε.

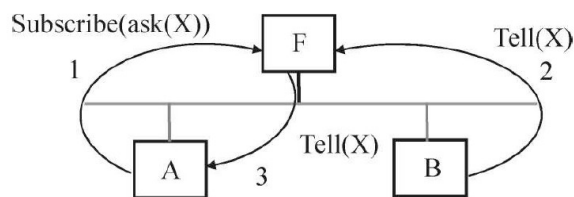
Παρακάτω βλέπουμε σχηματικά τα επίπεδα λειτουργίας:



Σχήμα 4.2 Επίπεδα του πρωτοκόλλου KQML[54]

Το περιβάλλον σχεδίασης της KQML ήταν να δημιουργηθεί μια γλώσσα που θα μπορεί να υποστηρίξει ένα μεγάλο αριθμό αρχιτεκτονικών πρακτόρων. Ο τρόπος που επιλέχθηκε ήταν η δημιουργία ενός αριθμού εκτελεστών που θα χαρακτηρίζουν τις ικανότητες και τις προδιαγραφές επικοινωνίας και η εισαγωγή μιας συγκεκριμένης

κατηγορίας πρακτόρων που ονομάζεται *facilitators*. *Facilitator* ονομάζεται ένας πράκτορας που κάνει διάφορες σημαντικές υπηρεσίες επικοινωνίας όπως για παράδειγμα η δρομολόγηση μηνυμάτων, η προώθηση μηνυμάτων στον σωστό παραλήπτη και η μετάφραση των στοιχείων της επικοινωνίας ανάμεσα στα μέλη της. Με την βοήθεια της χρήσης των *facilitators* η επικοινωνία μεταξύ των πρακτόρων δεν γίνεται άμεσα ανάμεσα τους αλλά με τους *facilitators*. Όπου εκεί διατυπώνουν τις ανάγκες τους, τις δυνατότητές τους και τις ερωτήσεις τους. Ένα παράδειγμα που μπορούμε να αναφέρουμε είναι ότι έχουμε δυο πράκτορες τον A και τον B αλλά και μια κατάσταση την οποία θα την ονομάσουμε X. Αν υποθέσουμε ότι ο πράκτορας A θέλει να μάθει μια κατάσταση X την οποία όμως κρατάει στην βάση δεδομένων του ο πράκτορας B. Σε περίπτωση που ο πράκτορας A γνωρίζει ότι ο πράκτορας B έχει την δυνατότητα να απαντήσει στην ερώτηση, τότε επικοινωνεί άμεσα μαζί του με το πρωτόκολλο που ονομάζεται *point-to-point*. Αν ο πράκτορας A δεν γνωρίζει ποιοι πράκτορες είναι διαθέσιμοι ή ποιος από αυτούς μπορεί να απαντήσει στην ερώτηση του, τότε πηγαίνει στον *facilitator* όπου αναλαμβάνει χρέη διαμεσολαβητή. Παρακάτω βλέπουμε ένα παράδειγμα σε σχήμα όπου ο πράκτορας A χρησιμοποιεί τον εκτελεστή *subscribe* ώστε να κάνει τον *facilitator* F να παρατηρεί την κατάσταση X. Όταν ο πράκτορας B πληροφορεί τον F για την κατάσταση X που επικρατεί, τότε αυτή η πληροφορία θα πάει από τον F και στον A [56].



Σχήμα 4.3 Χρήση του εκτελεστή [56]

Παρακάτω θα δούμε τα μηνύματα στην KQML την μορφή που έχουν[57]:

```
(ask-one
  :sender joe
  :content “ price(ibm,(Price, Time)) ”
  :receiver stock-server
  :reply-with ibm-stock
  :language standard_prolog
  :ontology NYSE-TICKS)
```

Το μήνυμα παραπάνω αναφέρεται σε μια ερώτηση για την τιμή της μετοχής στο χρηματιστήριο της εταιρίας IBM. Στον κώδικα ο πράκτορας που ρωτάει ονομάζεται stock-client ενώ αντίστοιχα αυτός που απαντάει stock-server και η γλώσσα διατυπώνεται σε Prolog και οντολογία NYSE-TICKS. Το ask-one καθορίζει την ενέργεια λόγου (performative), οι τιμές των πεδίων :sender, :content ονομάζονται ως λέξεις κλειδιά (keywords). Το επίπεδο περιεχομένου αποτελείται από την τιμή της λέξης κλειδί “:content”, ενώ το επίπεδο επικοινωνίας αποτελείται από τις τιμές των λέξεων κλειδιών “:sender, :receiver, :reply-with”. Τέλος το όνομα ask-one μαζί με τις τιμές των λέξεων “:language” και “:ontology” αποτελούν το επίπεδο του μηνύματος.

Παρακάτω έχουμε την απάντηση του πράκτορα stock-server:

```
(tell
  :sender stock-server
  :content “price(ibm,(14, ‘now’))”
  :receiver joe
  :in-reply-to ibm-stock
  :language standard_prolog
  :ontology NYSE-TICKS)
```

4.9 Η ΓΛΩΣΣΑ ΕΠΙΚΟΙΝΩΝΙΑΣ FIPA ACL

Η δεύτερη πιο γνωστή γλώσσα επικοινωνίας των πρακτόρων ACL (Agent-Communication-Language) μετά τη KQML είναι η FIPA ACL. Το 1995 η FIPA άρχισε να ερευνά την δημιουργία συστημάτων πρακτόρων ενώ ένα μέρος από την έρευνα αφορούσε μια γλώσσα επικοινωνίας πρακτόρων. Η γλώσσα αυτή μοιάζει σε κάποια πράγματα με την KQML, γιατί και αυτή ορίζει μια εξωτερική φόρμα για τα μηνύματα, λειτουργεί με εκτελεστές και δεν υποχρεώνει να μπουν περιορισμοί στην γλώσσα περιοχομένου. Ακόμη το συντακτικό των δυο γλωσσών επικοινωνίας μοιάζει σε κάποια σημεία. Η Semantic Language έχει οριστεί από την FIPA ACL , η οποία περιέχει τελεστές ώστε να αναφέρεται σε αυτά που χρειάζονται οι πράκτορες. Υπάρχει μια λιτή εργαλειοθήκη αλλά εξίσου ισχυρή ώστε να μπορούν να αναπαρασταθούν οι ενέργειες των πρακτόρων.

Ένα μήνυμα στην FIPA ACL περιέχει ένα ή παραπάνω στοιχεία. Τα στοιχεία τα οποία είναι σημαντικά για μια καλή επικοινωνία έχει σχέση από την περίπτωση. Το στοιχείο που θα πρέπει να υπάρχει κάθε φορά είναι ο τύπος της επικοινωνιακής ενέργειας , ενώ στα περισσότερα μηνύματα περιέχονται ακόμη τα στοιχεία του αποστολέα, του παραλήπτη και του περιεχομένου[57].

Παρακάτω έχουμε ένα σύνολο στοιχείων τα οποία αποτελούν μέρος ενός μηνύματος σε FIPA ACL[58]:

- Η επικοινωνιακή ενέργεια φανερώνει στον παραλήπτη πώς να μπορέσει να μεταφράσει το περιεχόμενο του μηνύματος.
- Ο αποστολέας (sender) του μηνύματος είναι το όνομα του πράκτορα που αρχίζει την ενέργεια της επικοινωνίας. Συχνά πολλές φορές ένα κομμάτι ενός μηνύματος μπορεί να παραμείνει άγνωστος ο αποστολέας .
- Ο παραλήπτης (receiver) του μηνύματος είναι το όνομα από τους πράκτορες στους οποίους απευθύνεται το μήνυμα. Το όνομα του παραλήπτη μπορεί να ληφθεί αν τα στοιχεία του είναι γνωστά από το περιεχόμενο του μηνύματος.
- Το στοιχείο reply- το ορίζει ότι τα επόμενα μηνύματα που ανήκουν στο ίδιο θέμα συνομιλίας υποχρεούνται να αφορούν τον πράκτορα που ορίζεται μετά το στοιχείο αυτό και όχι στον αποστολέα.

- Το στοιχείο content ορίζει το αντικείμενο και το περιεχόμενο του μηνύματος , τα περισσότερα μηνύματα σε αυτή την γλώσσα χρειάζονται το στοιχείο αυτό.
- Η γλώσσα language καθορίζει τον τύπο της γλώσσας που διατυπώνεται το περιεχόμενο. Υπάρχει μια βιβλιοθήκη στην οποία είναι αποθηκευμένες και άλλες συμβατές γλώσσες.
- Η κωδικοποίηση encoding ορίζει την προσδιορισμένη κωδικοποίηση που χρησιμοποιείται στην γλώσσα περιεχομένου. Η κωδικοποίηση του περιεχομένου μπορεί να γίνει με πολλούς τρόπους. Την κωδικοποίηση encoding την χρησιμοποιούμε για να περιγράψουμε την κωδικοποίηση που γίνεται ανάμεσα στον πράκτορα- παραλήπτη.
- Το στοιχείο ontology δίνει στα σύμβολα που υπάρχουν μέσα στις εκφράσεις του περιεχομένου.
- Το πρωτόκολλο (protocol) ορίζει το πρωτόκολλο αλληλεπίδρασης που χρησιμοποιεί ο πράκτορας αποστολέας για την αποστολή του μηνύματος. Όταν αναφέρουμε τον όρο πρωτόκολλο αλληλεπίδρασης εννοούμε μια σειρά μηνυμάτων που είναι απαραίτητο να ανταλλαχθούν σε κάποιο συγκεκριμένο σημείο της συνομιλίας. Αυτός που έχει σχεδιάσει το σύστημα μπορεί να καθορίσει τους πράκτορες ώστε να γνωρίζουν τους στόχους τους.
- Το στοιχείο conversation-id χρησιμοποιείται για να αναγνωρίζεται μια σειρά μηνυμάτων απεικονίζει μια συγκεκριμένη συνομιλία. Αυτό το στοιχείο βοηθάει τους πράκτορες να χειρίζονται τις ενεργές συνομιλίες που υπάρχουν με τους πράκτορες.
- Το στοιχείο reply-with χρησιμοποιείται σε μεγάλο αριθμό από συνομιλίες που υπάρχουν για λόγους καθαρά διαχείρισης.
- Το στοιχείο reply-by ορίζει τον χρόνο δηλαδή την ημερομηνία που θέλει ο πράκτορας αποστολέας να του έρθει μια απάντηση.

4.9.1 Η ερμηνεία της FIPA ACL

Η ερμηνεία στην FIPA ACL μπορεί να οριστεί από μια γλώσσα περιγραφής την Semantic Language(SL). Αυτή η γλώσσα περιγραφής προσδιορίζει με τελεστές οι οποίοι αναφέρουν τις προθέσεις κάθε πράκτορα. Στην FIPA ACL η ερμηνεία κάθε εκτελεστή ορίζεται σαν ένα σύνολο από φόρμες της Semantic Language που αναφέρουν την κατάσταση που έχει ο πράκτορας πριν την αποστολή του μηνύματος. Σε κάθε εκτελεστή οι pre-conditions αναλύουν ποιες είναι οι αναγκαίες συνθήκες του πράκτορα για να μπορέσει να στείλει τον εκτελεστή αυτό. Το αποτέλεσμα του εκτελεστή δείχνει τις συνέπειες που περιμένει ο πράκτορας να γίνει λόγω αυτής της επικοινωνιακής ενέργειας. Όμως ο πράκτορας παραλήπτης δεν υποχρεούται να πετύχει το αποτέλεσμα που θέλει ο πράκτορας αποστολέας γιατί αυτό μπορεί να είναι ανέφικτο για αυτόν. Για αυτό τον λόγο ο πράκτορας αποστολέας χρησιμοποιεί τη γνώση του για το αποτέλεσμα που θέλει ώστε να χρησιμοποιήσει ένα εκτελεστή αλλά δεν είναι βέβαιος ότι θα επιτύχει το αποτέλεσμα[58].

4.9.2 Διαφορές μεταξύ της KQML και της FIPA ACL

Αρχικά η KQML και η FIPA ACL είναι όμοιες στις βασικές τους αρχές και έννοιες. Επειδή η FIPA ACL ήθελε το συνακτικό της να μοιάζει με την KQML έγιναν κάποιες προσπάθειες ώστε το συνακτικό τους να είναι ίδιο. Αυτό είχε ως αποτέλεσμα η διαδικασία μετάβασης μεταξύ τους να γίνεται ευκολότερα. Αντίθετα στην δομή της ερμηνείας τους διαφέρουν αρκετά οι δύο γλώσσες. Για αυτό τον λόγο δεν μπορεί να γίνει πλήρης ταύτιση μεταξύ στους εκτελεστές των δύο γλωσσών. Ακόμη στην FIPA ACL, δεν γίνεται οι πράκτορες της να επιδρούν άμεσα στην κατάσταση ενός άλλου πράκτορα. Οι εκτελεστές της KQML δεν έχουν κανένα απολύτως νόημα στην FIPA ACL. Οι εκτελεστές στην FIPA ACL περικλείουν με μεγάλη ακρίβεια στις λειτουργίες επικοινωνίας μεταξύ των πρακτόρων. Επίσης η αρχιτεκτονική των πρακτόρων της περιέχει ένα σύστημα διαχείρισης πρακτόρων το οποίο εκτελεί τις υπηρεσίες διαχείρισης της κοινότητας των πρακτόρων. Στην KQML ένας πράκτορας οδηγεί έναν άλλο πράκτορα σε έναν ακριβή στόχο με την χρησιμοποίηση μόνο από το λεξιλόγιο της γλώσσας επικοινωνίας. Αντίθετα η FIPA ACL διαβιβάζει την αίτηση για την υλοποίηση του στόχου στην γλώσσα περιεχομένου. Τα δεδομένα θεωρούνται ότι αποτελούν μέρος της γλώσσας περιεχομένου, για αυτόν τον λόγο δεν έχει εκτελεστές για την

διαχείριση των δομών δεδομένων. Τέλος στην FIPA ACL είναι διαφορετικό ανάμεσα σε έναν πράκτορα που δεν θέλει να απαντήσει και σε έναν άλλο που δεν έχει την δυνατότητα να επεξεργαστεί την ερώτηση. Ενώ στην KQML υπάρχει ο ίδιο ο εκτελεστής[58].

ΚΕΦΑΛΑΙΟ 5

ΑΝΑΠΤΥΞΗ ΣΥΣΤΗΜΑΤΩΝ ΚΑΙ ΕΦΑΡΜΟΓΩΝ ΠΡΑΚΤΟΡΩΝ

5.1 ΛΟΓΙΣΜΙΚΟ JADE

Η JADE (Java Agent DEvelopment Framework) είναι ένα λογισμικό το οποίο έχει ως σκοπό την ανάπτυξη συστημάτων αλλά και εφαρμογών πολλαπλών πρακτόρων που ταιριάζει σύμφωνα με τα πρότυπα της FIPA. Χρησιμοποιεί στην ανάπτυξη αλλά και την εκτέλεση των peer-to-peer εφαρμογών που έχουν την αρχή τους στους πράκτορες λογισμικού. Έτσι έχουν την δυνατότητα να εργαστούν και να επικοινωνήσουν σε ενσύρματο και σε ασύρματο περιβάλλον[60]. Η JADE θεωρείται μια εντελώς υλοποιημένη γλώσσα στην JAVA καθώς έχει διάφορα πακέτα στην JAVA. Τα οποία δίνουν την δυνατότητα στους προγραμματιστές εφαρμογών κάποια έτοιμα κομμάτια λειτουργικότητας και αφηρημένες διεπαφές. Η επιλογή της γλώσσας προγραμματισμού JAVA έγινε λόγω των χαρακτηριστικών της, που υπάρχει μια σύνδεση με τον αντικειμενοστραφή προγραμματισμό στα περιβάλλοντα[59].

Η JADE περιέχει δυο κύρια προϊόντα : α) την πλατφόρμα πρακτόρων λογισμικού με τις προδιαγραφές της FIPA και β) ένα πακέτο για την ανάπτυξη JAVA πρακτόρων λογισμικού.

Μέσω ενός υλικο- λογισμικού απλουστεύει την δημιουργία συστημάτων πολλαπλών πρακτόρων που ενεργεί με τις απαιτήσεις της FIPA και ενός συνόλου εργασιών. Η πλατφόρμα πρακτόρων διανέμεται στις μηχανές και η διαμόρφωσή της ελέγχεται μέσω μιας γραφικής διεπαφής. Επίσης η διαμόρφωση έχει την δυνατότητα να επεξεργαστεί ενώ εκτελείται με την μετακίνηση των πρακτόρων λογισμικού από μια μηχανή σε μια άλλη , σε περίπτωση που χρειαστεί. Ως βασικό στόχο η JADE έχει την απλούστευση της ανάπτυξης συστημάτων πολλαπλών πρακτόρων σύμφωνα με τις απαιτήσεις της FIPA μέσω ενός συνόλου υπηρεσιών συστήματος και πρακτόρων λογισμικού. Η επικοινωνία των πρακτόρων λογισμικού γίνεται μέσω μηνυμάτων , όπου η γλώσσα είναι FIPA ACL[61].

Η αρχιτεκτονική επικοινωνίας δίνει ένα αποδοτικό και προσαρμόσιμο τρόπο στην ανταλλαγή μηνυμάτων , όπου η JADE αναλαμβάνει την δημιουργία και την διαχείριση μιας ουράς από εισερχόμενα ACL μηνύματα ξεχωριστά σε κάθε πράκτορα. Η πρόσβαση στην ουρά για τους πράκτορες μπορεί να επέλθει μέσω διάφορων τρόπων όπως το μπλοκάρισμα, η λήξη εγκυρότητας μιας ενέργειας κ.λ.π. Το πλήρες πρότυπο επικοινωνίας FIPA έχει δημιουργηθεί και τα στοιχεία που την αποτελούν έχουν χωριστεί και ενταχθεί πλήρως. Όπως για παράδειγμα είναι τα πρωτόκολλα αλληλεπίδρασης, τα ACL μηνύματα, οι γλώσσες περιεχομένου κ.λ.π. Ο μηχανισμός μεταφοράς μηνυμάτων είναι ευέλικτος σε κάθε κατάσταση , διαλέγοντας αντίστοιχα το καλύτερο εύκαιρο πρωτόκολλο. Ο μεγαλύτερος αριθμός από τα πρωτόκολλα αλληλεπίδρασης που ορίζονται από την FIPA είναι διαθέσιμα προς υλοποίηση , μετά τον καθαρισμό από την εφαρμογή κατάστασης για την κάθε κατάσταση του πρωτοκόλλου.

Οι πράκτορες στην JADE εκτελούνται ως νήματα με ένα νήμα να αναλογεί σε κάθε πράκτορα με την προϋπόθεση να εκτελούν παράλληλες εργασίες. Ακόμη η JADE υποστηρίζει τον προγραμματισμό συνεργαζόμενων συμπεριφορών , όπου οι εργασίες προγραμματίζονται με έναν εύκολο και αποτελεσματικό τρόπο. Για να ικανοποιήσει κάποιους περιορισμούς μνήμης, της υπολογιστικής ισχύος των κινητών συσκευών και τα χαρακτηριστικά των ασύρματων δικτύων , δηλαδή της εύρους ζώνης , της καθυστέρησης μεταφοράς , της μεταβλητότητας IP διευθύνσεων , ώστε να είναι συγχρόνως αποδοτική όταν εκτελείται έχει την δυνατότητα να διαμορφωθεί για να ταιριάζει στα χαρακτηριστικά του περιβάλλοντος. Με την ενεργοποίηση ορισμένων τμημάτων αντί άλλων στην JADE, δίνει την δυνατότητα να ικανοποιηθούν οι διάφορες απαιτήσεις όπως της συνδετικότητας , μνήμης και υπολογιστικής ισχύος[61].

Ένα πρόσθετο τμήμα στην JADE ονομάζεται JADE-S. Το οποίο αφήνει την προστασία ενός συστήματος πολλαπλών πρακτόρων στην JADE από τις επιθέσεις ασφάλειας. Συγκεκριμένα

όλα τα οποία υπάρχουν σε μια πλατφόρμα συνδέονται με χρήστες , οι οποίοι έχουν το δικαίωμα από τον διαχειριστή της πλατφόρμας να εκτελέσουν κάποιες ενέργειες. Κάθε πράκτορας έχει ένα κλειδί είτε δημόσιο είτε ιδιωτικό και έχει την δυνατότητα να μπορεί να υπογράψει αλλά και να κρυπτογραφήσει μηνύματα[59]. Οι τομείς εφαρμογών όπου τα συστήματα JADE έχουν την δυνατότητα να σχεδιαστούν αποτελεσματικά αλλά και να εξαπλωθούν είναι πολλοί. Επεκτείνονται από τις υπηρεσίες του διαδικτύου , όπου απευθύνονται στον καταναλωτή και στις εταιρίες ως το κινητό περιβάλλον με τα καλύτερα αποτελέσματα. Ακόμη στον τομέα των εφαρμογών χρησιμοποιείται machine-to-machine, όπου το peers proactivity και ο συντονισμός εργασιών είναι οι λειτουργίες που προσφέρουν καλά τις χαρακτηριστικές απαιτήσεις των δικτύων. Αυτά τα περιβάλλοντα έχουν την δυνατότητα να χρησιμοποιήσουν το ίδιο σύνολο λειτουργιών που ενδείκνυται ώστε να μπορούν να υποστηρίξουν τα πρωτόκολλα επικοινωνίας[60].

5.2 ΤΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΚΑΙ Η ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΗΣ JADE

Τα χαρακτηριστικά τα οποία παρέχει σε έναν προγραμματιστή η JADE είναι τα εξής[59]:

- Είναι μια πλατφόρμα πρακτόρων λογισμικού η οποία μπορεί να επεκταθεί σε πολλούς διαφορετικούς hosts. Μόνο μια εφαρμογή της JAVA μπορεί να εκτελεστεί σε κάθε host.
- Η γραφική διεπαφή χρήστη (GUI) χρησιμοποιείται για την μακρινή διαχείριση, παρακολούθηση αλλά και τον έλεγχο της κατάστασης των πρακτόρων λογισμικού. Έτσι με αυτόν τον τρόπο μπορεί να σταματάει αλλά και αν επανεκκινεί τους πράκτορες. Επιπλέον το GUI δίνει την δυνατότητα στην δημιουργία και την εκκίνηση της εκτέλεσης ενός πράκτορα σε ένα μακρινό host με την προϋπόθεση ένα κιβώτιο πρακτόρων λειτουργεί ήδη.
- Υπάρχουν κάποια εργαλεία με την ονομασία debugging που συμβάλλουν στην ανάπτυξη εφαρμογών με πολλαπλούς πράκτορες με βάση την JADE .
- Την κινητικότητα πρακτόρων , μαζί με την μεταφορά του κώδικα και της κατάστασης του πράκτορα.

- Μέσω του μοντέλου συμπεριφοράς μπορεί να υποστηρίξει την εκτέλεση πολλών, παράλληλων και ταυτόχρονων δραστηριοτήτων των πρακτόρων.
- Με την FIPA η πλατφόρμα πρακτόρων μπορεί να συμμορφωθεί. Στην πλατφόρμα εντάσσεται το AMS (Agent Management System -σύστημα διαχείρισης πρακτόρων), τον DF (Directory Facilitator- βοηθός καταλόγου) , και το ACC (Agent Communication Channel- διάυλος επικοινωνίας πρακτόρων). Αυτά τα στοιχεία ενεργοποιούνται αυτόματα κάθε φορά που εκκινείται η πλατφόρμα.
- Πολλοί μπορεί να εκκινηθούν σύμφωνα με τις προδιαγραφές της FIPA DFs στον χρόνο εκτέλεσης ώστε να πραγματοποιηθούν πολλαπλές περιοχές. Όπου μια περιοχή αντιστοιχεί σε ένα λογικό σύνολο πρακτόρων , των οποίων οι υπηρεσίες παρουσιάζονται από έναν κοινό βοηθό (DF). Σε κάθε DF μεταβιβάζεται ένα GUI και όλες τις ικανότητες που ορίζονται από την FIPA. Όπως για παράδειγμα την ικανότητα εγγραφής, διαγραφής, αναζήτησης περιγραφών πρακτόρων κ.λ.π.
- Υπάρχει αποδοτική μεταφορά μηνυμάτων ACL μέσα στην ίδια πλατφόρμα πρακτόρων. Τα μηνύματα μεταφέρονται κωδικοποιημένα ως αντικείμενα στην JAVA, παρά ως συμβολοσειρές.
- Υπάρχει βιβλιοθήκη με πρωτόκολλα αλληλεπίδρασης FIPA για την άμεση χρησιμοποίησή τους.
- Με το AMS υπάρχει η δυνατότητα να γίνει η εγγραφή και η διαγραφή πρακτόρων αυτόματα.
- Στην εκκίνησή τους οι πράκτορες παίρνουν το GUID (Globally Unique Identifier) από την πλατφόρμα σύμφωνα με την υπηρεσία ονομάτων της FIPA.
- Υποστήριξη για οριζόμενες γλώσσες περιεχομένου και οντολογίες στην εφαρμογή.
- Η διεπαφή InProcess δίνει την δυνατότητα να εκκινούν πράκτορες οι εξωτερικές εφαρμογές.

Κάθε στιγμιότυπο του περιβάλλοντος εκτέλεσης της JADE αναφέρεται ως Container επειδή έχει την δυνατότητα να περιλαμβάνει μεγάλο αριθμό πρακτόρων. Πλατφόρμα ονομάζεται το σύνολο ενεργών containers. Επίσης προσφέρει ένα στρώμα που κρύβει στους πράκτορες την πολυπλοκότητα και την εμφάνιση διάφορων μορφών των κύριων στοιχείων[62]. Κάθε πράκτορας καθορίζεται από ένα αναγνωριστικό πρακτόρων. Δηλαδή το αναγνωριστικό πρακτόρων περιέχει ένα μοναδικό παγκόσμιο όνομα μαζί με διάφορες διευθύνσεις. Το όνομα στην JADE έχει την εξής μορφή *<agentname>@<platform-name>* και οι διευθύνσεις είναι διευθύνσεις της πλατφόρμας όπου είναι οι πράκτορες. Οι διευθύνσεις αυτές συντάσσονται

από μια ακολουθία URI και χρησιμοποιούνται όταν ένας πράκτορας πρέπει να επικοινωνήσει με έναν άλλο που είναι σε μια διαφορετική πλατφόρμα[63].

Η αρχιτεκτονική της JADE βασίζεται σε έναν ειδικό κόμβο, που αναφέρεται ως Main container, ώστε να συγχρονίσει τους άλλους κόμβους και να κρατά ενωμένη όλη την πλατφόρμα. Στην πλατφόρμα ένας έστω Main container θα πρέπει να είναι ενεργός και όλοι οι άλλοι containers εγγράφονται μόλις εκκινούν. Ο πρώτος container που ξεκινάει σε μια πλατφόρμα πρέπει να είναι Main container. Οι επιπρόσθετοι containers πρακτόρων πρέπει να είναι κανονικοί containers και τους παρέχεται που θα βρουν τον Main containers που εγγράφονται. Οι κανονικοί containers έχουν την δυνατότητα να εκκινήσουν στον ίδιο ή σε μακρινούς hosts, όπου όμως είναι συνδεδεμένοι με τον Main container της πλατφόρμας των πρακτόρων. Με αποτέλεσμα ένα διαμοιρασμένο σύστημα που δίνει την εντύπωση ότι αποτελεί μια ενωμένη πλατφόρμα πρακτόρων. Υπάρχουν κάποια χαρακτηριστικά που λειτουργούν μόνο από τον Main container.

Τα χαρακτηριστικά αυτά είναι :

- Χειρισμός όλων των κόμβων που αποτελούν την πλατφόρμα (Container Table)
- Χειρισμός του συνόλου όλων των πρακτόρων λογισμικού που μαζί με την θέση στην οποία βρίσκονται φιλοξενούνται από την πλατφόρμα (Global Agent Descriptor).
- Χειρισμός όλων των σημείων τέλους (endpoints) MTP με σειρά, μαζί με την θέση τους.
- Φιλοξενία του πράκτορα AMS και του πράκτορα Default DF της πλατφόρμας.

Οι JADE πράκτορες έχουν ένα μοναδικό όνομα και αν ξέρουν ο ένας το όνομα του άλλου, έχουν την δυνατότητα να μπορούν να έρθουν σε επικοινωνία χωρίς να έχει σχέση η θέση τους. Μια πλατφόρμα πρακτόρων JADE περιέχει όλους αυτούς τους πράκτορες που χειρίζονται την πλατφόρμα οι οποίοι είναι: ο ACC, ο AMS, και DF. Η επικοινωνία των πρακτόρων λογισμικού γίνεται μέσω της μετάδοσης μηνυμάτων , όπου η γλώσσα που χρησιμοποιείται είναι η FIPA ACL.

5.3 Ο ΠΡΑΚΤΟΡΑΣ ACC, AMS ΚΑΙ Ο DF

Στον πράκτορα ACC το σύστημα μετάδοσης μηνυμάτων , γνωστό ως και διάυλος επικοινωνίας πρακτόρων , είναι ο πράκτορας που δείχνει την κατεύθυνση για την επαφή ανάμεσα στους πράκτορες λογισμικού μέσα και έξω από την πλατφόρμα. Είναι μια τεχνική επικοινωνίας που δίνει μια αξιόπιστη τακτική και σταθερή υπηρεσία μηνυμάτων. Ο πράκτορας αυτός κάθε φορά που λαμβάνει ένα ACL μήνυμα που κωδικοποιείται ως string, αναλύει το μήνυμα και το μετατρέπει σε ένα JAVA ACLMessage αντικείμενο που το χρησιμοποιούν όλοι οι πράκτορες JADE[60].

Ο AMS πράκτορας είναι αυτός ο πράκτορας που παρατηρεί την πρόσβαση και την χρήση της πλατφόρμας πρακτόρων. Σε μια ενιαία πλατφόρμα υπάρχει μόνο ένας AMS. Ο AMS προφυλάσσει ότι κάθε πράκτορας έχει δικό του μοναδικό όνομα και κύκλου ζωής , κρατώντας έναν κατάλογο των αναγνωριστικών (AID) και της κατάστασης των πρακτόρων. Η καταχώρηση κάθε πράκτορα γίνεται με έναν AMS ώστε να μπορεί να έχει ένα έγκυρο AID. Ο ρόλος του AMS ισοδυναμεί με την αρχή στην πλατφόρμα δηλαδή μπορεί να δημιουργηθούν ή να καταστραφούν πράκτορες σε μακρινά containers με μια αίτηση αυτού στον AMS[62].

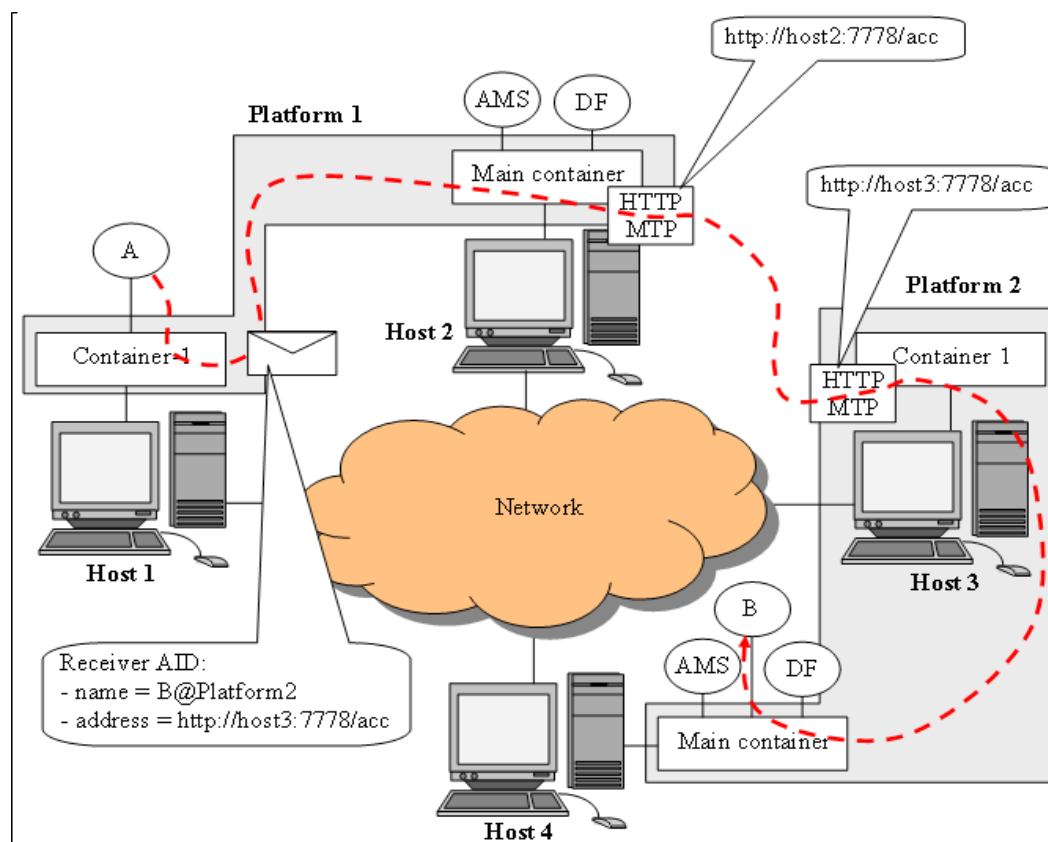
Στην JADE υπάρχει μια υπηρεσία κίτρινων σελίδων υλοποιείται από έναν πράκτορα DF. Κάθε πλατφόρμα η οποία ακολουθεί τις προδιαγραφές της FIPA έχει έναν πράκτορα DF. Άλλοι πράκτορες DF μπορούν να ενεργοποιηθούν και διάφοροι DF μπορούν να ενωθούν με αποτέλεσμα να δίνουν έναν κατάλογο κίτρινων σελίδων. Από την στιγμή που ο DF είναι πράκτορας , μπορεί να γίνει η αλληλεπίδραση με αυτόν τις περισσότερες φορές , με την ανταλλαγή ACL μηνυμάτων που χρησιμοποιούν την κατάλληλη γλώσσα περιεχομένου και μια σωστή οντολογία σύμφωνα με τις προδιαγραφές της FIPA. Όταν ένας πράκτορας θέλει να δημοσιεύσει κάποιες υπηρεσίες θα πρέπει να δίνει στον DF μια περιγραφή της οποίας θα περιλαμβάνει το AID του, την λίστα γλωσσών και οντολογιών που θα πρέπει να γνωρίζουν οι άλλοι πράκτορες για να αλληλεπιδράσουν με αυτόν και την λίστα με τις υπηρεσίες που έχουν δημοσιευθεί. Κάθε υπηρεσία που είναι δημοσιευμένη θα πρέπει να δίνεται μια κατάσταση που θα περιέχει τον τύπο υπηρεσιών, το όνομα υπηρεσιών , τις γλώσσες και τις οντολογίες που υποχρεούνται ώστε να ωφεληθούν από την υπηρεσία αλλά και τις ιδιότητες της υπηρεσίας[62].

Η αρχιτεκτονική λογισμικού της JADE βασίζεται στην ταυτόχρονη ύπαρξη κάποιων εικονικών μηχανών Java (Java Virtual Machines - JVMs) ενώ η επικοινωνία βασίζεται στην

Java RMI (Remote Method Invocation) ανάμεσα σε διαφορετικούς JVMs και στην σηματοδότηση γεγονότων μέσα σε μια JVM. Μια πλατφόρμα πρακτόρων μπορεί να διανεμηθεί σε διάφορους hosts. Κάθε JVM αποτελείται από ένα container πρακτόρων που δίνει ένα περιβάλλον εκτέλεσης για τους πράκτορες και αφήνει διάφορους πράκτορες να εκτελεστούν ταυτόχρονα στον ίδιο hosts[64].

Κάθε container πρακτόρων είναι ένα πολύ-νηματικό περιβάλλον εκτέλεσης που απαρτίζεται από ένα νήμα για κάθε πράκτορα. Όπως επίσης τα νημάτα συστημάτων που δημιουργούνται από το RMI σύστημα χρόνου εκτέλεσης για την αποστολή μηνυμάτων. Ο Main container εδώ έχει τον ρόλο της διαπροσωπείας, δηλαδή την εκτέλεση των πρακτόρων διαχείρισης και την αντιπροσώπευση όλης της πλατφόρμας στον έξω κόσμο. Όταν η επικοινωνία RMI ανάμεσα στα hosts συνεχίζεται , τότε η διανομή containers σε ένα δίκτυο υπολογιστών μπορεί να γίνει.

Όπως βλέπουμε στο σχήμα από κάτω μια πλατφόρμα πρακτόρων (AP) από διάφορα containers πρακτόρων.



Σχήμα 5.1 Container και πλατφόρμες[62]

Κάθε container πρακτόρων είναι ένα RMI server αντικείμενο που χειρίζεται ένα σύνολο από πράκτορες λογισμικού. Επιπλέον επιβλέπει τον χρόνο ζωής των πρακτόρων δημιουργώντας τους, διακόπτοντας την εκτέλεσή τους, επανεκκινώντας και τερματίζοντάς τους. Ακόμη χειρίζεται όλες τις πλευρές της επικοινωνίας, στέλνοντας ACL εισερχόμενα μηνύματα σύμφωνα με εκεί που θα πρέπει να σταλθούν αλλά και βάζοντας τα στις ουρές μηνυμάτων των πρακτόρων. Ο container πρακτόρων για τα εξερχόμενα μηνύματα, έχει πληροφορίες ώστε να μπορεί να βρει την θέση των πρακτόρων – δεκτών και να διαλέξει τον καλύτερο τρόπο μεταφοράς για να στείλει ACL μήνυμα.

Η γραφική διεπαφή χρήστη (GUI) στην πλατφόρμα πρακτόρων χρησιμοποιείται για την μακρινή διαχείριση, παρατήρηση και έλεγχο της κατάστασης των πρακτόρων, που έχει την δυνατότητα να εκκινεί ή να σταματάει τους πράκτορες. Το GUI δίνει την δυνατότητα να δημιουργηθεί και να αρχίσει η εκτέλεση ενός πράκτορα σε ένα μακρινό host, με την προϋπόθεση ότι τρέχει ήδη ένα container πρακτόρων. Μια JADE πλατφόρμα όταν εκκινείται, ο AMS και DF αμέσως ενώ το ACC τοποθετείται ώστε να έχει την δυνατότητα να επιτρέπει την επικοινωνία μηνυμάτων. Ο Main container, είναι ο container πρακτόρων που υπάρχουν ο AMS και DF, δημιουργείται το RMI registry που χρησιμοποιείται από την JADE. Τα υπόλοιπα containers πρακτόρων συνδέονται με τον main container και δίνουν ένα περιβάλλον εκτέλεσης προκειμένου να εκτελεστεί οποιοδήποτε σύνολο πρακτόρων JADE[59]. Πιο αναλυτικά ο Main Container είναι ένας container μετωπικού άκρου (front-end) όπου υπάρχει εσωτερικά ένα RMI registry, που τον χρησιμοποιούν άλλοι containers πρακτόρων στο διαστήμα εκκίνησης ώστε να μπορούν να εγγράφονται με το μετωπικό άκρο (front-end) για να γίνονται μέλη της πλατφόρμας. Ο Main Container έχει έναν πίνακα όλων των containers και την RMI αναφορά των αντικειμένων τους. Επιπλέον υπάρχει ένας γενικός πίνακας περιγραφής των πρακτόρων, ο οποίος συνδυάζει κάθε όνομα πρακτόρων με τα δεδομένα AMS και με την RMI αναφορά αντικειμένου του container του.

Όταν αρχίζει να εκτελείται ένα νέο front-end container, δημιουργείται ένα νέο εσωτερικό RMI registry στο host που καταλαβαίνει μια συγκεκριμένη θύρα TCP/IP από τον χρήστη. Στην συνέχεια εκκινεί τους πράκτορες των συστημάτων (AMS, DF και ACC). Όταν ξεκινάει ένα νέο container να εκτελείται, ψάχνει το RMI registry το οποίο είναι ενσωματωμένο μέσα στο εκτελούμενο front-end container της πλατφόρμας στην οποία θέλει να συνδεθεί. Μετά αποκτά από αυτό την RMI αναφορά αντικειμένου του front-end της πλατφόρμας. Για να είναι πιο αυξημένη η απόδοση κάθε container, έχει στην γρήγορη μνήμη την αναφορά αντικειμένου άλλων containers, κάθε φορά που στέλνεται μήνυμα σε αυτούς. Με αυτό τον τρόπο δεν χρειάζεται κάθε φορά που πρέπει να στείλει ένα μήνυμα να κάνει αναζήτηση στον

Agent Global Descriptor Table. Το περιβάλλον της JADE έχει τους καλούς πίνακες πρακτόρων ώστε να μπορεί να διαλέξει τον πιο αποδοτικό μηχανισμό μηνύματος , σε σχέση με την θέση πρακτόρων-δεκτών. Ο container πρακτόρων δίνει την δυνατότητα στην JADE να προσαρμοστεί στην προϋπόθεση ότι οι μηχανισμοί επικοινωνίας πρέπει να είναι άγνωστοι σε όλους τους πράκτορες[64].

5.4 Ο ΤΡΟΠΟΣ ΕΠΙΚΟΙΝΩΝΙΑΣ ΤΩΝ JADE ΠΡΑΚΤΟΡΩΝ

Ένα από τα πιο σημαντικά χαρακτηριστικά που έχουν οι JADE πράκτορες είναι η δυνατότητα να επικοινωνήσουν. Δηλαδή κάθε πράκτορας έχει μια ουρά από μηνύματα που κατά το διάστημα εκτέλεσης εισάγονται τα μηνύματα που στέλνουν άλλοι πράκτορες. Έτσι όταν εισάγεται ένα μήνυμα στην ουρά μηνυμάτων ενημερώνεται ο πράκτορας που το λαμβάνει. Αν ένας πράκτορας πάρει αλλά και πότε ένα μήνυμα από την ουρά μηνυμάτων , ώστε να το επεξεργαστεί έχει να κάνει με τον ρόλο του προγραμματιστή[62].

Η μορφή των μηνυμάτων που ανταλλάσσονται από τους JADE πράκτορες , ορίζεται από την γλώσσα ACL σύμφωνα με τα πρότυπα FIPA. Η μορφή αυτή περιέχει διάφορα στοιχεία και συγκεκριμένα [65]:

- ❖ Τον αποστολέα (sender) του μηνύματος
- ❖ Την λίστα των δεκτών (receivers)
- ❖ Την πρόθεση (performative) που έχει ο αποστολέας προκειμένου να ορίσει ποιος είναι ο στόχος του με την αποστολή του μηνύματος.
- ❖ Την γλώσσα περιεχομένου δηλαδή την σύνταξη που χρησιμοποιείται για να διατυπωθεί το περιεχόμενο. Επειδή ο αποστολέας και ο δέκτης πρέπει να μπορούν να κωδικοποιήσουν το συγκεκριμένο περιεχόμενο για να μπορεί να υπάρξει μια αποτελεσματική επικοινωνία.

- ❖ Το περιεχόμενο που περιλαμβάνει τις πραγματικές πληροφορίες που υπάρχουν σε ένα μήνυμα.
- ❖ Την οντολογία που δίνει το λεξιλόγιο των συμβόλων που χρησιμοποιούνται στην έννοια και στο περιεχόμενό τους. Και εδώ ο αποστολέας και ο δέκτης πρέπει να ερμηνεύσουν το ίδιο νόημα στα σύμβολα ώστε να είναι αποτελεσματική η επικοινωνία.
- ❖ Κάποια από τα πεδία που χρησιμοποιούνται για να ελέγξουν ταυτόχρονες συνομιλίες και για να ορίσουν την προθεσμία που θα δοθεί η απάντηση είναι: conversation-id, reply-with, in-reply-to, reply-by.

Κάθε φορά που στέλνει ένας JADE πράκτορας ένα μήνυμα, συμβαίνουν οι παρακάτω περιπτώσεις[64]:

- 1) Αν ο πράκτορας - δέκτης είναι μαζί στο ίδιο container πρακτόρων, τότε το αντικείμενο Java ενεργεί το ACL μήνυμα περνά στον δέκτη, χωρίς να μεταφραστεί το μήνυμα.
- 2) Αν ο πράκτορας – δέκτης είναι μαζί στην ίδια JADE πλατφόρμα αλλά σε διαφορετικό container, τότε το ACL μήνυμα στέλνεται χρησιμοποιώντας το πλαίσιο Java RMI. Ακόμη ο πράκτορας λαμβάνει ένα Java αντικείμενο, όπως είναι το μήνυμα ανάμεσα στους containers.
- 3) Σε περίπτωση που ο δέκτης είναι σε διαφορετική πλατφόρμα πρακτόρων, τότε το τυποποιημένο πρωτόκολλο IIOP αλλά και η διεπαφή OMG IDL χρησιμοποιούνται με βάση τα πρότυπα της FIPA. Περιέχει την μετάφραση του αντικειμένου ACL μηνύματος και την εκτέλεση μιας μακρινής κλήσης χρησιμοποιώντας IIOP πρωτόκολλο.

5.5 ΟΝΤΟΛΟΓΙΕΣ ΚΑΙ ΓΛΩΣΣΕΣ ΠΕΡΙΕΧΟΜΕΝΟΥ ΣΤΗ ΓΛΩΣΣΑ JADE

Οι πληροφορίες που βρίσκονται μέσα στο ACL μήνυμα, απεικονίζονται ως έκφραση περιεχομένου με βάση μια κατάλληλη γλώσσα περιεχομένου. Οι πράκτορες έχουν έναν συγκεκριμένο δικό τους τρόπο να απεικονίζουν εσωτερικά τις πληροφορίες αυτές. Αυτός ο τρόπος πρέπει να δίνει την δυνατότητα στον εύκολο χειρισμό των πληροφοριών. Όστε να είναι σαφές ότι η αναπαράσταση που χρησιμοποιείται σε μια ACL έκφραση περιεχομένου δεν είναι καλή για το εσωτερικό ενός πράκτορα. Ακόμη η αποθήκευση των πληροφοριών αυτών μέσα σε ένα πράκτορα ως μια μεταβλητή string δεν είναι η καλύτερη δυνατή για τον χειρισμό των πληροφοριών. Στον πράκτορα JADE όπου γράφεται με Java , οι πληροφορίες απεικονίζονται μέσα σε έναν πράκτορα ως Java αντικείμενο[66]. Η JADE στηρίζει τις γλώσσες περιεχομένου και τις οντολογίες ώστε οι μετατροπές από την μια αναπαράσταση στην άλλη και οι λειτουργίες ελέγχου πληροφοριών να έχουν την δυνατότητα να εκτελούνται αυτόματα. Έτσι δίνοντας την δυνατότητα στους υπεύθυνους ανάπτυξης εφαρμογών το χειρισμό των πληροφοριών μέσα στους πράκτορες ως αντικείμενα Java. Η JADE για να εκτελέσει τους κατάλληλους εννοιολογικούς ελέγχους σε μια συγκεκριμένη έκφραση περιεχομένου θα πρέπει να ταξινομηθούν , με βάση τα εννοιολογικά χαρακτηριστικά τους. Όπως και όλα τα πιθανά στοιχεία που θα μπορούν να εμφανιστούν μέσα σε μια πρόταση που στέλνεται από έναν πράκτορα σε έναν άλλο, ως περιεχόμενο ενός μηνύματος ACL. Ειδικότερα στο πρώτο επίπεδο κάνουμε τον διαχωρισμό μεταξύ των predicates (κατηγορημάτων) και terms (όρων). Οι εκφράσεις είναι οι εξής:

- **Predicates:** Είναι εκφράσεις που αναφέρονται για την κατάσταση που επικρατεί στον κόσμο και μπορεί να είναι true ή false.
- **Terms:** Είναι εκφράσεις που ορίζουν τις οντότητες που είναι στον κόσμο από τις οποίες οι πράκτορες βγάζουν συλλογισμούς. Μπορούμε να τις ταξινομήσουμε ως εξής:
 - **Concepts:** Τα concepts είναι εκφράσεις που συνήθως δεν έχουν κανένα νόημα αν χρησιμοποιηθούν άμεσα ως περιεχόμενο ACL μηνύματος. Η αναφορά που κάνουν είναι μέσα σε κατηγορήματα και σε άλλα concepts.

- Agent actions: Είναι κάποιες έννοιες που διατυπώνουν ενέργειες που μπορούν να εκτελεστούν από κάποιους agents. Οι πράξεις επικοινωνίας όπως τα μηνύματα ACL θεωρούνται και αυτές ενέργειες πρακτόρων.
- Primitives: Είναι οι εκφράσεις που διατυπώνουν ατομικές οντότητες όπως για παράδειγμα strings και ακέραιους αριθμούς.
- Aggregates: Είναι οι εκφράσεις που διατυπώνουν οντότητες που ανήκουν σε ομάδες άλλων οντοτήτων.
- Identifying Referential Expressions (IRE): Είναι οι εκφράσεις που ορίζουν την οντότητα για την οποία ισχύει ένα συγκεκριμένο κατηγορημα.
- Variables: Είναι οι εκφράσεις που ορίζουν ένα άγνωστο γενικό στοιχείο και συνήθως χρησιμοποιείται στις ερωτήσεις.

Επίσης μια εκφραστική γλώσσα περιεχομένου πρέπει να έχει την δυνατότητα να αναπαραστήσει και να διαχωρίσει όλα τα παραπάνω στοιχεία των προηγούμενων τύπων. Η Semantic Language θεωρείται μια γλώσσα περιεχομένου κατανοητή από τον άνθρωπο και κωδικοποιημένη σε string. Θεωρείται από την πιο διαδεδομένη γλώσσα περιεχομένου στον τομέα της επιστημονικής κοινότητας που το αντικείμενο ενασχόλης τους είναι οι ευφυείς πράκτορες. Αντίθετα η LEAP είναι μια γλώσσα περιεχομένου που δεν είναι κατανοητή από τον άνθρωπο, κωδικοποιημένη σε bytes που έχει προστεθεί στην JADE μέσα στο πρόγραμμα LEAP. Άρα όπως είναι δεδομένο μόνο οι JADE πράκτορες θα έχουν την δυνατότητα και την γνώση να επικοινωνήσουν με την γλώσσα LEAP. Συνήθως προτιμούν αυτή την γλώσσα όταν υπάρχουν περιορισμοί μνήμης.

Μια οντολογία για μια συγκεκριμένη θεματική περιοχή είναι ένα σύνολο από σχήματα τα οποία ορίζουν την δομή των κατηγορημάτων, των ενεργειών των πρακτόρων και εννοιών που αφορούν εκείνη την περιοχή[66].

5.6 ΚΙΝΗΤΟΙ ΠΡΑΚΤΟΡΕΣ ΣΤΗΝ JADE

Οι υπεύθυνοι που χρησιμοποιούν την JADE για την ανάπτυξη εφαρμογών έχουν την δυνατότητα να δημιουργήσουν κινητούς πράκτορες , που θα μπορούν να φύγουν ή να δημιουργήσουν αντίγραφο τους σε πολλά hosts δικτύων. Στην τωρινή έκδοση της JADE, ένας κινητός πράκτορας μπορεί να πάει σε διαφορετικά containers πρακτόρων αλλά έχει περιορισμό σε μια ενιαία JADE πλατφόρμα. Οι κινητοί πράκτορες πρέπει να είναι ενημερωμένοι για τις θέσεις ώστε να μπορούν να αποφασίσουν πότε και που θα κινηθούν. Έτσι από την JADE παρέχεται μια οντολογία, που ονομάζεται jade-mobility-ontology, που περιλαμβάνει απαραίτητες ενέργειες και έννοιες. Η κίνηση ενός πράκτορα περιέχει την αποστολή κώδικα αλλά και της κατάστασής του μέσω ενός καναλιού δικτύου. Κάποιοι από τους πόρους που χρησιμοποιούνται από τον κινητό πράκτορα θα κινηθούν, ενώ αντίθετα κάποιοι άλλοι θα τερματιστούν πριν την μετακίνηση και μετά θα επαναλειτουργήσουν πάλι στον προορισμό[59].

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Γ. Παυλίδης , Ολοκληρωμένη Τεχνολογία Πληροφορικής, Αθήνα: Gutenberg,2001
2. Γ. Παυλίδης, Εφαρμοσμένα Πληροφοριακά Συστήματα I, Πάτρα, 2013
3. T.J. Bentley, Managing information: Avoiding Overload, London: CIMA,1998.
4. Κ. Μάρκελλος, Π. Μαρκέλλου, Μ. Ρήγκου, Σ. Συρμακέσης και Α. Τσακαλίδης, e-Επιχειρηματικότητα, Αθήνα: Ελληνικά Γράμματα, 2005.
5. D. C Dennett, Consiousness Explained, Boston: Brown and Co., 1991
6. P.F. Drucker, The Practice of Management, New York: HarperCollins Publishers, 1954.
7. M. Aljukhadar, S.Senecal and C.-E. Daoust, “Using Recommendation Agents to Cope with Information Overload ”, Internation Journal of Electronic Commerce, vol.17, no.2 , pp .41-70, 2013
8. A. Symeonidis and P. Mitkas, Agent Intelligence Through Data Mining, Springer, 2005.
9. U. Hanani, B. Shapira and P. Shoval “ Information Filtering: Overview of Issues, Research and Systems” , User Modeling and User-Adapted Interaction, pp.203-259,2001.
10. J. Palme, “ Information Filtering”, Stockholm, 1998.
11. K. Juszczyszyn, P. Kazienko and K. Musial, “ Personalized Ontology-Based Recommender Systems for Multimedia Objects”, in Hakansson, A.et al.(Eds.), Agent and Multi- Agent Technology for Internet and Enterprise Systems, Berlin, Springer, 2010, pp.271-288
12. Stuart Russel & Peter Norvig «Τεχνητή νοημοσύνη , μια σύγχρονη προσέγγιση», Β' έκδοση 2005, Εκδόσεις Κλειδάριθμος, σς. 31-69.
- 13.https://el.wikipedia.org/wiki/%CE%A4%CE%B5%CF%87%CE%BD%CE%B7%CF%84%CE%AE_%CE%BD%CE%BF%CE%B7%CE%BC%CE%BF%CF%83%CF%8D%CE%BD%CE%B7
14. Τίμος Σελλάς, Τεχνητή Νοημοσύνη, Διάλεξη 1, <http://ecourse.lib.ntua.gr/NODEL/L0/20.html>
15. McCarthy J. (1998) what is artificial Intelligence? Computer science Department, Stanford University, Standford CA94305
- 16.Russel, S. and Norvig, P., Artificial Intelligent: A Modern Approach. Prentice Hall, Englewood Cliffs, USA, 1995
17. <http://filipinofreethinkers.org/wp-content/uploads/2012/06/Turing-test.png>
- 18.https://www.google.gr/search?q=%CE%B2%CE%B1%CF%83%CE%B9%CE%BA%CE%BF%CF%82+%CF%80%CF%81%CE%B1%CE%BA%CF%84%CE%BF%CF%81%CE%B1%CF%82+%CE%BA%CE%B1%CF%84%CE%B1+russell+%26+norvig&biw=1366&bih=643&source=lnms&tbn=isch&sa=X&ved=0ahUKewiB1OWz5frOAhUkJ8AKHVAhDjgQ_AUIBygC#tbn=isch&q=%CE%B2%CE%B1%CF%83%CE%B9%CE%BA%CE%BF%CF%82+%CF%80%CF%81%CE%B1%CE%BA%CF%84%CE%BF%CF%81%CE%B1%CF%82+russell+%26+norvig&imgsrc=eGMPHoOf2g8v7M%3A
19. <http://web.media.mit.edu/~pattie/CACM-94/CACM-94.p1.html>

20. http://repfiles.kallipos.gr/html_books/93/img_book/sxima_6.3.png
21. http://onlinelibrary.wiley.com/doi/10.1207/s15516709cog0304_1/epdf
22. http://repfiles.kallipos.gr/html_books/93/img_book/sxima_6.4.png
23. <ftp://publications.ai.mit.edu/ai-publications/pdf/AITR-1493.pdf>
24. http://repfiles.kallipos.gr/html_books/93/img_book/sxima_6.5.png
25. http://repfiles.kallipos.gr/html_books/93/img_book/sxima_6.1.png
26. N.R. Jennings and M.J. Wooldridge, Agent Technology, Berlin; Springer,1998
27. P. Maes, “ Agent that Reduce Work and Information Overload”,
Communication of the ACM, vol 34, no pp. 30-40,1994
28. <https://dspace.lib.uom.gr/bitstream/2159/13304/2/AthanasoglouMsc2009.pdf>
29. Guilfoyle, C.(1998). Vendors of Intelligent Agent Technologies: A Market Overview. In N. Jennings and M. Wooldridge (Eds.), Agent Technology: Foundations, Applications, and Market, pp. 91-104, Springer
30. Russel, S. and Norvig, P., Artificial Intelligent: A Modern Approach. Prentice Hall, Englewood Cliffs, USA, 1995
31. Nwana H.S., “Software Agents : An Overview,” Knowledge Engineering Review, vol. 11(3), sept. 1996, p. 1-40
32. Davies N. and Weeks R., Jasper: Communicating Information Agents, In proceedings of the 4th International Conference on the Word Wide Web, Boston , USA, December 1995
33. Wooldridge M. , On agent-based software engineering, Artificial Intelligence, Vol.117, 2000, p.277-296
34. Sycara, K. (1995). Intelligent Agents and the Information Revolution. UNICOM Seminar on Intelligent Agents and their Business Applications, pp. 143-159, 8-9 November
35. Nwana, H. S., Ndumu, D.T. (1998). A Brief Introduction to Software Agent Technology. In N. R. Jennings, M. J. Wooldridge (eds) Agent Technology : Foundations, Applications, and Markets, Springer-Verlag, Berlin.
36. O. Etzioni Moving up the information food chain: deploying softbots on the World Wide Web, 1996
37. <http://dione.lib.unipi.gr/xmlui/bitstream/handle/unipi/4207/Malakos.pdf?sequence=2>
38. M. Apelkrans and A.Hakansson, “Information Coordination Using Meta-agents in Information Logistics Processes ”, in Hakansson , A.et al.(Eds.), Springer, 2010, pp.117-134.
39. Michael Wooldridge (2008) Εισαγωγή στα Πολυπρακτορικά Συστήματα Αθήνα
40. V. Honavar, “Intelligent Agents and Multi Agent Systems ”, A Tutorial Presented at IEEE CEC 99, Washington, D.C. July 1999.
41. Lesser, V.R.; Corkill, D.D.: The Distributed Vehicle Monitoring Test-bed: A Tool for Investigating Distributed Problem Solving Networks. In: AI Magazine 4(1983) Nr.6, P.15-33.
42. R. A. Flores-Mendez, “Towards a Standardization of Multi-Agent System Frameworks”, ACM Crossroads, <http://www.acm.org/crossroads>, Summer 1999 – 5.4.
43. L. Gasser, “The Integration of Computing and Routine Work. ” ACM Transactions on office Information Systems, 4:3, July, 1986

44. M.P. Georgeff, “ Communication and Interaction in Multi-agent Planning,” Proceedings of the National Conference on Artificial Intelligence, Washington, D.C., August 1983, pp. 125-129
45. Rosenschein, J. S., and Breese, J. S. (1989). Communication-free interactions among rational agents. In Distributed Artificial Intelligence2 (L.Gasser and M. N. Huhns,eds.) pp.99-118. Morgan Kaufmann, Los Altos, CA/Pitman, London.
46. Nii, H.P., Blackboard Systems Part One, AI Magazine, Vol. 7, No. 2, pp. 38-53, 1986.
47. J.S. Rosenschein. Rational Interaction: Cooperation Among Intelligent Agents. PhD thesis, Stanford University, 1986.
48. Searle, J.R., 1969. Speech Acts: An Essay in the Philosophy of Language. Cambridge University Press, Cambridge, UK
49. Malone, T.W. and Crowston, K.(1990). What is Coordination Theory and How Can It Help Design Cooperative Work Systems. In [Halasz 1990], pp. 357-370.
50. E.H. Durfee and Victor R. Lesser Negotiating Task Decomposition and Allocation Using Partial Global Planning. In L. Gasser and M.N. Huhns, editor, Distributed Artificial Intelligence, Volume II, pages 229-244. Pitman/Morgan Kaufmann, London, 1989.
51. K.S. Decker, “ Distributed problem solving: A survey,” IEEE Transactions on Systems, Man, and Cybernetics, vol. 17, no.5, pp.729-740, 1987.
52. Finin T., Fritzson R. & McKay D.: "A Language and Protocol to Support Intelligent Agent Interoperability", In Proceedings of the CE & CALS Washington '92 Conference, June 1992
53. Finin , T ., Labrou, Y., and Mayfield, J. (1987). KQML as an Agent Communication Language In J. M. Bradshaw (ed), Software Agents, AAAI Press, Menlo Park, CA
54. Tim Finin, Don McKay, Rich Fritzson, and Robin McEntire. KQML: an information and knowledge exchange protocol. In Kazuhiro Fuchi and Toshio Yokoi, editors, Knowledge Building and Knowledge Sharing. Ohmsha and IOS Press, 1994.
55. Tim Finin, Don McKay, Rich Fritzson, and Robin McEntire. The KQML information and knowledge exchange protocol. In Third International Conference on Information and Knowledge Management, November 1994.
56. Finin T., Labrou Y. & Mayfield J.: "KQML as an agent communication language", in Bradshaw J.(Ed.), "Software Agents", AAAI Press/MIT Press, 1997.
57. FIPA(1997) , Agent Communication Language. FIPA '97 Specification: Part 2, Version 1.0, 10 October 1997.
58. <http://filipinofreethinkers.org/wp-content/uploads/2012/06/Turing-test.png>
59. F. Bellifemine, G. Caire, T. Trucco, G. Rimassa, “JADE Tutorial: JADE programmer’s guide”, <http://jade.tilab.com/doc/programmersguide.pdf>, July 2004.
60. F. Bellifemine, G. Caire, A. Poggi, G. Rimassa, “JADE A White Paper”, Telecom Italia Lab (TILAB), <http://exp.telecomitalialab.com>, exp - Volume 3 - n., September 2003.

61. The Java Agent DEvelopment Framework (JADE), Introduction, <http://jade.tilab.com/doc/html/intro.htm>
62. G. Caire, “JADE Tutorial: JADE programming for beginners”, <http://jade.tilab.com/doc/JADEProgramming-Tutorial-for-beginners.pdf>, December 2003.
63. F. Bellifemine, G. Caire, T. Trucco, G. Rimassa, R. Mungenast, “JADE Tutorial: JADE administrator’s guide”, <http://jade.tilab.com/doc/administratorsguide.pdf>, December 2003
64. Bellifemine F., Poggi A., Rimassa G., “JADE – A FIPA- compliant agent framework ”, <http://sharon.cselt.it/projects/jade/papers/PAAM.pdf>
65. The Foundation for Intelligent Physical Agents, <http://www.fipa.org/>
66. G. Caire, D. Cabanillas, “JADE Tutorial: Application-defined content languages and ontologies”, <http://jade.tilab.com/doc/CLOntoSupport.pdf>, November 2004.

