

Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Ευφυείς Πράκτορες και Εφαρμογές τους
Όνοματεπώνυμο Φοιτητή	Μαλακός Κωνσταντίνος του Βασιλείου
Αριθμός Μητρώου	ΜΠΣΠ/07063
Κατεύθυνση	Συστήματα Υποστήριξης Αποφάσεων
Επιβλέπων	Δημήτριος Δεσπότης, Καθηγητής

Ημερομηνία Παράδοσης **Απρίλιος 2011**

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΔΑΛΗ

Τριμελής Εξεταστική Επιτροπή

Δημήτρης Δεσπότης
Καθηγητής

Ευάγγελος Φούντας
Καθηγητής

Δημήτρης Αποστόλου
Επικ. Καθηγητής

Περιεχόμενα

Περίληψη	4
Abstract	5
Εισαγωγή	6
Κεφάλαιο 1- Ευφυή Πληροφοριακά Συστήματα	8
1.1 - Η ανάγκη για Ευφυή Πληροφοριακά Συστήματα	8
1.2 - Ευφυή Συστήματα Προσομοίωσης	8
1.3 - Ευφυή Συστήματα Πληροφοριών	9
1.4 - Ευφυή Συστήματα Υποστήριξης.....	10
1.5 – Ευφυής Μηχανές Αυτόματου ελέγχου (Robots)	11
1.6 – Ανάγκη για Υποδομή.....	11
1.7 – Ευκολία Χρήσης.....	11
1.8 – Ευέλικτες και Προσαρμόσιμες Υπηρεσίες Υποδομής.....	12
1.9 - Εργαλεία Ανάπτυξης και Υποστήριξης	13
1.10 - Εφαρμοσμένες Τεχνολογίες.....	13
Κεφάλαιο 2 – Ευφυείς Πράκτορες	17
2.1 - Η έννοια του πράκτορα.....	17
2.2 Επιπλέον ορισμοί.....	18
2.3 - Παραδείγματα πρακτόρων	19
2.4 - Χαρακτηριστικά πρακτόρων.....	19
2.5 - Πώς πρέπει να λειτουργούν οι πράκτορες	22
2.6 - Ευφυείς Πράκτορες	23
2.7 - Αλγόριθμοι Ευφυούς πρακτόρων.....	25
Κεφάλαιο 3 – Προβλήματα Πρακτόρων	27
3.1 - Προβλήματα τα οποία πρέπει να λύσουν οι πράκτορες	27
3.2 - Προβλήματα κατά τη χρήση των πρακτόρων	27
3.3 - Λύνοντας νέους τύπους προβλημάτων	28
3.4 Ανοιχτά συστήματα	28
3.5 Πολύπλοκα συστήματα.....	28
3.6 –Συστήματα Υπολογιστών.....	29
3.7 - Βελτίωση της αποτελεσματικότητας στην ανάπτυξη λογισμικού	29
3.8 - Κατανομή πόρων, ελέγχου τεχνογνωσίας, και πηγών.....	30
3.9 - Φυσική μεταφορά.....	30
3.10 - Κληρονομικά συστήματα.....	31
3.11 - Οι περιορισμοί στις λύσεις πρακτόρων	31

3.12 Προβλήματα στην ανάπτυξη του πράκτορα.....	32
3.13 Προδιαγραφές.....	32
3.14 - Σχεδιασμός συστήματος	33
3.15 - Εφαρμογές συστήματος	33
3.16 - Έλεγχος συστήματος, διόρθωση σφαλμάτων και επαλήθευση.....	34
Κεφάλαιο 4 – Εφαρμογές Πρακτόρων	35
4.1 Χαρακτηριστικά εφαρμογών πρακτόρων	35
4.2 Βιομηχανικές εφαρμογές.....	35
4.3 Διαδικασία ελέγχου	36
4.4 Έλεγχος εναέριας κυκλοφορίας.....	36
4.5 JADE (Java Agent DEvelopment Framework)	40
4.6 - Εμπορικές εφαρμογές	50
4.7 - Ηλεκτρονικό εμπόριο	54
4.8 - Διαδικασία διοίκησης επιχειρήσεων	54
4.9 - Ιατρικές εφαρμογές.....	54
4.10 - Ψυχαγωγία.....	57
4.11 – Στοιχεία των εφαρμογών πρακτόρων	59
Κεφάλαιο 5 – Συστήματα Πολλαπλών πρακτόρων	61
5.1 Επικοινωνία μεταξύ πρακτόρων.....	61
5.2 - Πρωτόκολλα Επικοινωνίας μεταξύ πρακτόρων.....	63
5.3 Το πρωτόκολλο Contract Net.....	63
5.4 Το πρωτόκολλο και γλώσσα KQML	64
5.5 Πρότυπα Πρωτόκολλου FIFA.....	65
5.6 – Συστήματα συνεργαζόμενων Ευφών Πρακτόρων	67
5.7 - Συστήματα συνεργαζόμενων Ευφών Πρακτόρων.....	68
5.8 – Ομοσπονδιακοί αρχιτέκτονες.....	70
Συμπεράσματα	73
Βιβλιογραφία.....	74

Περίληψη

Σύμφωνα, λοιπόν, με τους M. Woolbribe και N. Jennings: «Ένας πράκτορας είναι ένα υπολογιστικό σύστημα (hardware και Software) το οποίο είναι εγκατεστημένο σε ένα περιβάλλον και έχει την ικανότητα αυτόνομης δράσης στο συγκεκριμένο περιβάλλον προκειμένου να ανταποκριθεί στους σκοπούς σχεδίασης του.». Οι ευφυείς πράκτορες λοιπόν είναι ένα νέο πρότυπο για την ανάπτυξη εφαρμογών λογισμικού. Ακόμα περισσότερο οι των ευφυείς πράκτορες έχουν χαρακτηριστεί με φράσεις όπως «η επόμενη σημαντική εξέλιξη στην ανάπτυξη λογισμικού», και «η νέα επανάσταση στον τομέα του λογισμικού». Σήμερα οι ευφυείς πράκτορες είναι το επίκεντρο του έντονου ενδιαφέροντος σε πολλούς τομείς της επιστήμης των υπολογιστών και της τεχνητής νοημοσύνης. Οι πράκτορες χρησιμοποιούνται σε όλο και πιο μεγάλη ποικιλία εφαρμογών, που κυμαίνεται από συγκριτικά μικρά συστήματα όπως το ηλεκτρονικό ταχυδρομείο μέχρι και σε μεγάλα συστήματα όπως ο έλεγχος της εναέριας κυκλοφορίας. Εκ πρώτης όψεως, μπορεί να φαίνεται ότι αυτή οι εξαιρετικά διαφορετικοί τύποι συστημάτων μπορεί να έχουν ελάχιστα κοινά στοιχεία όμως το κλειδί που συνδέει τις δύο αυτές εφαρμογές είναι η χρήση των ευφυών πρακτόρων.

Στην εργασία αυτή παρουσιάζεται η θεωρία των ευφυών πρακτόρων, καθώς και οι τομείς εφαρμογών τους. Πιο συγκεκριμένα εισάγουμε και ορίζουμε την έννοια του πράκτορα και των συστημάτων ευφυών πρακτόρων, δίνουμε τα χαρακτηριστικά ενός πράκτορα και των συστημάτων πρακτόρων, εισάγουμε τα κύρια πεδία εφαρμογής τους, και τεχνολογίες πρακτόρων που έχουν επιτυχώς αναπτυχθεί μέχρι σήμερα, εντοπίζουμε τις δυσκολίες που βρίσκονται στο δρόμο υλοποίησης ενός συστήματος πρακτόρων και τέλος, εισάγουμε τα πολλαπλά συστήματα πρακτόρων και αναφερόμαστε στις μεθόδους που επικοινωνούν μεταξύ τους οι πράκτορες καθώς και με το πως αντιδρά ένα σύστημα πολλαπλών πρακτόρων με το περιβάλλον.

Καταλήγουμε στο συμπέρασμα ότι η τεχνολογία συστημάτων πρακτόρων είναι ζωντανή και ταχέως εξελισσόμενη επιστήμη τόσο στην ακαδημαϊκή έρευνα όσο και στις επιχειρηματικές εφαρμογές. Το νέο πρότυπο παρέχει τον σχεδιασμό και την εφαρμογή συστημάτων για ένα πολύπλοκο, δυναμικό περιβάλλον όπου τα γνωρίσματα είναι η διαπραγμάτευση, τα απρογραμμάτιστα γεγονότα και δράση και πρέπει να αντιμετωπισθεί με τρόπο που είναι επωφελής για τη συνολική απόδοση του συστήματος. Η τεχνολογία των ευφυών πρακτόρων συνήθως λειτουργεί είναι σε μεγάλο βαθμό σαν πανάκεια για τον τρόπο σχεδιασμού ενός ευφυούς συστήματος και την ανάπτυξη του, αλλά ο κατασκευαστής πρέπει να γνωρίζει και τις παγίδες που συμβαίνουν σε αυτό καθώς και σε κάθε νέα και μη δοκιμασμένη τεχνολογία. Οι δυνατότητες της τεχνολογίας ευφυών πρακτόρων είναι εκεί, αλλά το πλήρες όφελος δεν έχει ακόμα επιτευχθεί. Η τεχνολογία ευφυών πρακτόρων μπορεί να επιτύχει τις πραγματικές δυνατότητές της μόνο εφόσον οι χρήστες και προγραμματιστές αντιληφθούν την αξία των δραστηριοτήτων της.

Abstract

As Wooldridge and Jennings state, an agent is a hardware and/or software based computer system displaying the properties of autonomy, social adeptness, reactivity, and proactivity. So, intelligent agents are a new paradigm for developing software applications. More than this, agent-based computing has been hailed as 'the next significant breakthrough in software development', and 'the new revolution in software'. Currently, agents are the focus of intense interest on the part of many sub-fields of computer science and artificial intelligence. Agents are being used in an increasingly wide variety of applications, ranging from comparatively small systems such as email filters to large, open, complex, mission critical systems such as air traffic control. At first sight, it may appear that such extremely different types of system can have little in common. And yet this is not the case: in both, the key abstraction used is that of an agent.

In this document we present the theory of intelligent agents and the fields of their applications. More specific we introduce to the concept of an agent and agent-based systems, we show the domain characteristics of an agent-based system, we introduce the main application areas in which agent technology has been successfully deployed to date, we identify the main obstacles that lie in the way of the agent system developer, and finally we introduce the multiple agents systems and the methods that they communicate each other and with their environment.

We conclude that Agent-based systems technology is a vibrant and rapidly expanding field of academic research and business applications. By providing a new paradigm for designing and implementing systems for a complex, dynamic, and distributed environment where the common currency is negotiation, unplanned for events can be managed in a way that is beneficial to the overall system performance. Agent technology is greatly hyped as a panacea for the current ills of system design and development, but the developer is cautioned to be aware of the pitfalls inherent in any new and untested technology. The potential is there but the full benefit is yet to be realized. Agent technology will achieve its true potential only if users understand its business value.

Εισαγωγή

Οι ευφυείς πράκτορες (intelligent agents) και ο προσανατολισμένος στους πράκτορες προγραμματισμός (agent oriented programming) είναι ίσως η πιο επαναστατική μέθοδος αντίληψης, σχεδίασης και ανάπτυξης λογισμικού των τελευταίων χρόνων. Ένας ευφυής πράκτορας είναι ένα αυτόνομο πρόγραμμα που είναι συνεχώς ενεργό, παρατηρεί το περιβάλλον όπου ενεργεί και αντιδρά στα γεγονότα που το αφορούν χωρίς την παρέμβαση του χρήστη. Επιπλέον, ενεργεί σύμφωνα με κάποιους στόχους που έχει, οι οποίοι μπορεί να είναι μέρος ενός απώτερου στόχου. Προκειμένου να επιτευχθούν αυτοί οι στόχοι μπορεί να επικοινωνήσει και να συνεργαστεί με άλλους πράκτορες. [58]

Η παρούσα εργασία στόχο έχει την ανάλυση της θεωρίας και των εφαρμογών των ευφύων πρακτόρων. Χωρίζεται σε πέντε κεφάλαια. Ορίζονται τα Ευφυείς Συστήματα Πληροφοριών και στη συνέχεια οι ευφυείς πράκτορες. Περιγράφονται προβλήματα που αφορούν τους πράκτορες, δίνονται εφαρμογές τους σε διάφορους τομείς της επιστήμης και των επιχειρήσεων και στην συνέχεια ορίζονται και περιγράφονται τα συστήματα πολλαπλών πρακτόρων καθώς και πως σε αυτά τα συστήματα οι πράκτορες αλληλεπιδρούν μεταξύ τους και πως με το περιβάλλον.

Πιο συγκεκριμένα στο πρώτο κεφάλαιο παρουσιάζεται η ανάγκη χρήσης Ευφύων Συστημάτων Πληροφορικής, θέματα υποδομής για τη βέλτιστη απόδοση αυτών των συστημάτων, ενώ επίσης περιγράφονται τομείς της Τεχνητής Νοημοσύνης που μπορούν να συμβάλουν στην ανάπτυξη τέτοιων συστημάτων. Ο λόγος που αναφέρονται αυτά τα συστήματα είναι ότι σε αυτά τα συστήματα κυρίως βρίσκουν εφαρμογές οι πράκτορες ή μέσα από αυτά δραστηριοποιούνται. Στην ουσία ένας ευφυής πράκτορας είναι ένα ευφύες σύστημα από μόνο του και έτσι είναι αναγκαίο να παρουσιάσουμε την δομή και λειτουργία τους.

Στο δεύτερο κεφάλαιο παρουσιάζονται εισαγωγικά θέματα σχετικά με τους ευφυείς πράκτορες, τα οποία περιλαμβάνουν ορισμούς του όρου προκειμένου να τεθεί ένα συγκεκριμένο πλαίσιο αναφοράς, αρχιτεκτονικές σύμφωνα με τις οποίες κατασκευάζονται οι ευφυείς πράκτορες, καθώς και συστήματα που είτε λειτουργούν σε ένα ευρύ φάσμα εφαρμογών είτε βρίσκονται σε ερευνητικό επίπεδο. Αρχικά δίνονται διάφορες απόψεις της επιστημονικής κοινότητας σχετικά με το τι είναι ένας ευφυής πράκτορας και στη συνέχεια διατυπώνεται ένας αυστηρός ορισμός σύμφωνα με τα διεθνή πρότυπα. Επίσης γίνεται παρουσίαση των μεθόδων που χρησιμοποιούνται για την περιγραφή της συμπεριφοράς των ευφύων πρακτόρων, παρουσιάζονται οι αρχιτεκτονικές με τις οποίες κατασκευάζονται, οι αρχές της Τεχνητής Νοημοσύνης, που θεωρεί τους πράκτορες είτε ως συστήματα συμβόλων και κάνει χρήση μηχανισμών διαχείρισης αυτών είτε σαν αρχιτέκτονες και θεωρεί πως έξυπνη συμπεριφορά προκύπτει από το συνδυασμό πολλών στοιχειωδών ενεργειών, είτε χρησιμοποιεί στοιχεία των και των δύο προηγούμενων τεχνικών προκειμένου να εκμεταλλευτεί τα πλεονεκτήματα αυτών και να βελτιώσει την απόδοση του συστήματος.

Στο τρίτο κεφάλαιο αναφερόμαστε στα προβλήματα τα οποία πρέπει να λύσουν οι πράκτορες, του τύπου προβλημάτων που προκύπτουν, ορίζουμε τα ανοιχτά και πολύπλοκα, τα συστήματα υπολογιστών και πως εφαρμόζονται οι πράκτορες σε αυτά καθώς και στην βελτίωση της αποτελεσματικότητας της ανάπτυξη λογισμικού με χρήση των πρακτόρων

Στο τέταρτο κεφάλαιο γίνεται αναφορά σε εφαρμογές των ευφύων πρακτόρων σε εμπορικές εφαρμογές, βιομηχανικές εφαρμογές, ψυχαγωγικές εφαρμογές, παιχνίδια στο Internet κ.α. Αναλύονται τεχνικές που έχουν εφαρμοστεί σε κάθε περίπτωση και έτσι γίνεται κατανοητό πως οι ευφυείς πράκτορες μπορούν να εφαρμοστούν σε κάθε πεδίο και με ποιόν τρόπο.

Στο πέμπτο κεφάλαιο παρουσιάζονται βασικά θέματα συστημάτων πολλαπλών πρακτόρων, όπως μέθοδοι και πρωτόκολλα επικοινωνίας, αρχιτεκτονικές, τρόποι διασύνδεσης και διαλειτουργικότητας. Αρχικά περιγράφονται οι τρόποι που έχουν χρησιμοποιηθεί κατά καιρούς για την επικοινωνία ευφύων πρακτόρων και αναλύεται ο γενικότερα οι πιο αποδεκτοί και χρησιμοποιούμενοι τρόποι. Στην συνέχεια, ορίζονται και περιγράφονται τα πρωτόκολλα επικοινωνίας ευφύων πρακτόρων. Πρώτα γίνεται παρουσίαση σε δύο γνωστά και εφαρμοσμένα εφαρμογές πρωτόκολλα επικοινωνίας, αφετέρου παρουσιάζονται διεθνή πρότυπα που επικεντρώνονται κυρίως σε τυπικούς τρόπους αλληλεπίδρασης. Τέλος, αναλύονται θέματα συνεργασίας ευφύων πρακτόρων που περιλαμβάνουν παρουσίαση της

ανάγκης για συνεργασία καθώς και περιγραφή αρχιτεκτονικών συστημάτων συνεργαζόμενων πρακτόρων. Επίσης, περιγράφεται η ανάγκη για διαλειτουργικότητα μεταξύ ετερογενών συστημάτων.

Στο τέλος στα συμπεράσματα καταλήγουμε ότι η χρήση των ευφυών πρακτόρων αλλά και των πολλαπλών ευφυών πρακτόρων είναι πλέον επιβεβλημένη και έντονα εφαρμόσιμη σε πολλά πεδία της επιστήμης αλλά και της σύγχρονης ανάπτυξης λογισμικού για την καθημερινότητα. Η χρήση των ευφυών πρακτόρων δίνει λύσεις σε πεδία όπως στην εκπαίδευση, στην αναζήτηση πληροφοριών, στη Ιατρική, στην ανάπτυξη έξυπνων μηχανών καθημερινής χρήσης κ.α. Έτσι η γνώση τους για ανάπτυξη ευφυών πληροφοριακών συστημάτων καθώς και η σωστή και ορθή εφαρμογή τους είναι απαραίτητη και απαιτείται τελικά σε κάθε τομέα της πληροφορικής.

Κεφάλαιο 1- Ευφυή Πληροφοριακά Συστήματα

1.1 - Η ανάγκη για Ευφυή Πληροφοριακά Συστήματα

Τα τελευταία χρόνια τα υπολογιστικά συστήματα έχουν εισχωρήσει στη ζωή μας, τόσο στον επαγγελματικό τομέα όσο και σε προσωπικό επίπεδο. Το φαινόμενο αυτό θα έχει ακόμα πιο μεγάλη ένταση τα επόμενα χρόνια. Ήδη σχεδόν όλες οι ιδιωτικές επιχειρήσεων και οι δημόσιους οργανισμούς, οι τράπεζες καθώς και τομείς όπως η διαχείριση και κατανομή ηλεκτρικής ενέργειας, οι τηλεπικοινωνίες, ο στρατιωτικός τομέας, ο τομέας υγείας, η εκπαίδευση κ.ά. χρησιμοποιούν πληροφοριακά συστήματα μικρού ή μεγάλου μεγέθους. Ωστόσο η συσσώρευση ολοένα και μεγαλύτερου όγκου πληροφορίας, οι συνεχείς και πολλές φορές απρόβλεπτες αλλαγές του περιβάλλοντος λειτουργίας, οι συνεχώς αυξανόμενες απαιτήσεις και διαφορετικές ανάγκες των χρηστών αλλά και πλήθος άλλων παραγόντων διαφορετικών μεταξύ τους και ξεχωριστών για κάθε τομέα εφαρμογής, καθιστούν αναγκαία τη δυνατότητα προσαρμογής των υπολογιστικών συστημάτων στις νέες συνθήκες λειτουργίας και στις νέες ανάγκες που προκύπτουν. Έτσι γίνεται αναγκαία η ανάπτυξη υπολογιστικών συστημάτων με ευφυή συμπεριφορά.

Η **Τεχνητή Νοημοσύνη (Artificial Intelligence)** είναι ο κλάδος της επιστήμης των υπολογιστών, ο οποίος μελετά φαινόμενα ευφυούς συμπεριφοράς και αναπτύσσει μεθόδους και τεχνικές τεχνικής ευφυΐας για υπολογιστικά συστήματα χρησιμοποιώντας θεωρητικά και πειραματικά εργαλεία. Ο όρος «ευφυΐα» έχει αποτελέσει αντικείμενο συζήτησης πολλών φιλοσόφων και ερευνητών παρόλα αυτά η Τεχνητή νοημοσύνη χρησιμοποιεί τον όρο αυτό ορίζοντας μαθηματικά μοντέλα και αλγόριθμους ώστε τα συστήματα να συμπεριφέρονται με αυτονομία και προσαρμοστικότητα στα ανάλογα προβλήματα που καλούνται να λύσουν. Ωστόσο, ένας γενικός ορισμός σχετικά με το τι είναι ευφυΐα είναι αυτός του J. McCarthy σύμφωνα με τον οποίο «ευφυΐα είναι το υπολογιστικό μέρος της ικανότητας για την επίτευξη στόχων στον κόσμο» [35].

Τα τελευταία χρόνια η Τεχνητή Νοημοσύνη έχει δώσει πλήθος μεθόδων και τεχνικών τόσο σε θεωρητικό όσο και πειραματικό επίπεδο που βοηθούν στην κατανόηση των μηχανισμών εκδήλωσης ευφυούς συμπεριφοράς για συγκεκριμένα έργα. Έτσι έχει γίνει δυνατή η ανάπτυξη ευφυών συστημάτων σε ένα ευρύ φάσμα εφαρμογών, από ελεγκτές εναέριας κυκλοφορίας έως οικιακές συσκευές. Έτσι στην παγκόσμια βιβλιογραφία, γίνεται λόγος για τους τρόπους με τους οποίους η τεχνολογία που έχει αναπτυχθεί στην Τεχνητή Νοημοσύνη μπορεί να δώσει νέες και σημαντικές δυνατότητες σε πλήθος συστημάτων σε όλους τους τομείς της δημόσιας και ιδιωτικής δραστηριότητας, δυνατότητες οι οποίες θα έχουν επίδραση στη βελτίωση των προσφερομένων υπηρεσιών, την αύξηση της παραγωγικότητας και την αναβάθμιση της ποιότητας ζωής.

Έτσι με κριτήριο τους τομείς που μπορεί να εφαρμοστεί η Τεχνητή Νοημοσύνη έχουμε τις παρακάτω κατηγορίες Ευφυών Συστημάτων:

- Ευφυή Συστήματα Προσομοίωσης
- Ευφυή Συστήματα Πληροφοριών
- Ευφυή Συστήματα Υποστήριξης
- Ευφυή Robots

1.2 - Ευφυή Συστήματα Προσομοίωσης

Τα συστήματα προσομοίωσης είναι συστήματα που αναπαριστούν επιλεγμένα χαρακτηριστικά συμπεριφοράς ενός άλλου φυσικού ή αφηρημένου συστήματος, όπως για η προσομοίωση μιας κατάστασης ανάγκης σε μία αεροπορική πτήση. Αν και έχουν αναπτυχθεί αρκετά συστήματα προσομοίωσης μέχρι σήμερα, αυτά παρουσιάζουν περιορισμένες δυνατότητες στη ρεαλιστική αναπαράσταση των φυσικών αντικειμένων και έλλειψη δυνατοτήτων στη ρεαλιστική αναπαράσταση ανθρώπων. Η χρήση των τεχνικών της Τεχνητής Νοημοσύνης μπορεί να υποστηρίξει την ανάπτυξη

προγραμμάτων για τη μοντελοποίηση σύνθετων καταστάσεων και να δώσει ακριβείς αναπαραστάσεις στη συμπεριφορά των μερών που συμμετέχουν σε μία προσομοίωση μεγάλης κλίμακας. Τα διάφορα μέρη που συμμετέχουν μπορούν να σχεδιάζουν και να ενεργούν αυτόνομα, να συνεργάζονται μεταξύ τους, να συντονίζουν τις ενέργειές τους προκειμένου να επιτύχουν κάποιο κοινό στόχο, να αποκτούν εμπειρία από προηγούμενες καταστάσεις και να προσαρμόζουν τη συμπεριφορά τους και τις ενέργειές τους ανάλογα με τις συνθήκες του περιβάλλοντος όπου λειτουργούν. Η δυνατότητα των ευφυών συστημάτων προσομοίωσης να παράγουν ακριβείς και ρεαλιστικούς κόσμους τα καθιστά πολύτιμα σε εφαρμογές κατάρτισης και εκπαίδευσης ομάδων ατόμων σε πραγματικές καταστάσεις, όπως και σε βιομηχανικές και στρατιωτικές εφαρμογές.[60]

Στον τομέα της κατάρτισης και εκπαίδευσης, οι επιχειρήσεις μπορούν να εκπαιδεύσουν σε καταστάσεις λήψης αποφάσεων, καταστάσεις κατά τις οποίες μία στρατηγική απόφασή τους θα μπορούσε να αποβεί καταστροφική οικονομικά για την επιχείρηση. Στην ιατρική οι γιατροί μπορούν να εκπαιδευτούν στην περίθαλψη ασθενών με σπάνιες ή περίπλοκες ασθένειες και να δουν τα αποτελέσματα των διαφόρων θεραπευτικών αγωγών που ακολουθήθηκαν. Χειρουργικές ομάδες μπορούν να εκπαιδευτούν και να δουν μία δύσκολη εγχείριση κατά την οποία είναι πιθανόν να συμβούν διάφορες περιπλοκές. Χειριστές πολύπλοκων και ακριβών συσκευών και μηχανημάτων μπορούν να εκπαιδευτούν στη χρήση τους σε πραγματικές και απρόβλεπτες καταστάσεις. Στρατιωτικοί μπορούν να εκπαιδευτούν σε συνθήκες μάχης, να δοκιμάσουν νέες τακτικές, να βελτιώσουν την απόδοση των κινήσεων ανάλογα με τις συνθήκες του περιβάλλοντος των επιχειρήσεων και να παρακολουθήσουν τα αποτελέσματα των αποφάσεών τους. Στον τομέα της εκπαίδευσης, οι μαθητές μπορούν να μεταφερθούν σε διάφορες ιστορικές εποχές και να χρησιμοποιήσουν τα υπάρχοντα μέσα της εποχής για να κατασκευάσουν γεωργικά εργαλεία, ή να εξασκηθούν σε μία ξένη γλώσσα, για παράδειγμα, σε ένα περιβάλλον που προσομοιώνει ένα εμπορικό κέντρο μιας πόλης της αντίστοιχης χώρας στο οποίο πρέπει να έρθουν σε επαφή με πολίτες, εμπόρους, υπαλλήλους, οδηγούς ταξί, κ.ά.

Στον τομέα των βιομηχανικών και στρατιωτικών εφαρμογών, ένα προϊόν (π.χ. ένα αεροπλάνο, ένα αυτοκίνητο, μία ιατρική συσκευή κτλ.), αφού σχεδιαστεί, μπορεί να δοκιμαστεί σε ένα προσομοιωμένο περιβάλλον ως προς την απόδοσή του, τη συμπεριφορά του, την αξιοπιστία του, την ευχρηστία του, τη συντηρησιμότητά του και να υποστεί τροποποιήσεις και βελτιώσεις, πριν ακόμα κατασκευαστεί, αποφεύγοντας έτσι ένα τεράστιο κόστος ανάπτυξης και δοκιμής πρωτοτύπων.

1.3 - Ευφυή Συστήματα Πληροφοριών

Ο όγκος πληροφοριών που συσσωρεύεται στις βάσεις δεδομένων, στις ηλεκτρονικές βιβλιοθήκες ή βρίσκεται στο Διαδίκτυο (Internet) είναι τεράστιος και αυξάνεται εκθετικά. Καθημερινά όλο και περισσότεροι χρήστες έχουν πρόσβαση στο Διαδίκτυο και αναζητούν πληροφορίες στο από αυτό τον τεράστιο όγκο. Το Internet σήμερα έχει καθιερωθεί σαν ένα από τα βασικότερα μέσα επικοινωνίας καθώς μπορεί οποιοσδήποτε και πολύ εύκολα να αποκτήσει πρόσβαση χωρίς υψηλό κόστος. Ταυτόχρονα όσο αυξάνεται η ανάγκη για αποδοτικότερες επιχειρησιακές πρακτικές, τόσο επιτακτικότερη γίνεται και η εξεύρεση νέων καναλιών, μέσω των οποίων θα επεκταθούν οι σημερινές επιχειρήσεις, θα μεγιστοποιήσουν τα κέρδη τους και ταυτόχρονα θα ικανοποιήσουν τον καταναλωτή. [60]

Το Διαδίκτυο αποτελεί πρόκληση και ζητούμενος μοχλός ανάπτυξης για όλες τις επιχειρήσεις που έχουν δημιουργήσει δικτυακούς τόπους, με σκοπό την προβολή τους, την παροχή πληροφοριών στους χρήστες για τη γεωγραφική θέση των φυσικών καταστημάτων τους, τους καταλόγους των προϊόντων τους, για θέσεις εργασίας κ.ά. Επίσης, το Internet παίζει το ρόλο και εφαρμογής συναλλαγών μεταξύ επιχειρήσεων (Business-to-Business, B2B) ή μεταξύ επιχείρησης και καταναλωτή (Business-to-Consumer, B2C), κάτι που παρουσιάζει θεαματική ανάπτυξη τα τελευταία χρόνια. Ωστόσο, ο κάθε χρήστης που αναζητά πληροφορίες για κάποιο ορισμένο σκοπό θα πρέπει να γνωρίζει που θα τις βρει και με ποιο τρόπο να τις ψάξει. Αν και υπάρχουν αρκετά προγράμματα αναζήτησης, τις περισσότερες φορές τα αποτελέσματα δεν εξυπηρετούν το στόχο ενώ επιπλέον οι χρήστες θα πρέπει να γνωρίζουν τη σύνταξη κάποιων γλώσσας ερωτημάτων (query language) για τα συγκεκριμένα προγράμματα. Ένα άλλο πρόβλημα, που προκύπτει, είναι ο τρόπος με τον οποίο οι

χρήστες θα ενημερώνονται για νέες πληροφορίες ή υπηρεσίες που είναι διαθέσιμες και εμπίπτουν στα ενδιαφέροντά τους.[30]

Έτσι τα ευφυή συστήματα πληροφοριών καλούνται να καλύψουν τα παραπάνω προβλήματα υποστηρίζοντας με αποτελεσματικό τρόπο τη διαχείριση αυτού του τεράστιου όγκου πληροφοριών, σύμφωνα με τις ανάγκες των χρηστών. Τέτοια συστήματα θα μπορούν να προσαρμόζονται στις ανάγκες και τις ικανότητες του καθενός, να αναζητούν τις κατάλληλες πληροφορίες για κάθε στόχο ή να αναζητούν πληροφορίες που εμπίπτουν στα ενδιαφέροντα του χρήστη και να επικοινωνούν μαζί του με απλό και φυσικό τρόπο προκειμένου να υποστηριχθούν οι χρήστες με περιορισμένες γνώσεις υπολογιστών. Επιπλέον, θα μπορούν να λαμβάνουν πρωτοβουλία και να εκτελούν απλές συνηθισμένες καθημερινές ενέργειες, θα μπορούν να ανακαλύπτουν νέες προσφερόμενες υπηρεσίες χρήσιμες για τον αντίστοιχο χρήστη και να ανακαλύπτουν άτομα με κοινά ενδιαφέροντα. Τέλος, θα μπορούν να αναγνωρίζουν το περιεχόμενο των πληροφοριών ανεξάρτητα από τη μορφή που αυτές είναι αποθηκευμένες (π.χ. κείμενο, εικόνα, video).

Έτσι, ένα ευφύες σύστημα διαχείρισης ηλεκτρονικών εγγράφων ή ηλεκτρονικού ταχυδρομείου θα μπορεί παρακολουθώντας και μαθαίνοντας από τις συνήθειες του χρήστη να παραλαμβάνει, ταξινομεί και παρουσιάζει τα εισερχόμενα ή εξερχόμενα έγγραφα ή μηνύματα ανάλογα με τη σπουδαιότητά τους το που απευθύνονται ή σε ποια κατηγορία ανήκουν. Επίσης ένα ευφύες σύστημα πλοήγησης στο διαδίκτυο θα ελέγχει τους διαδικτυακούς τόπους που επισκέπτεται πιο συχνά ο χρήστης, εξετάζοντας το αν έχουν ανανεωθεί ή να διακρίνει site που ο χρήστης πιθανόν ενδιαφέρεται για τις πληροφορίες που διαθέτουν. Επίσης, ένα ευφύες σύστημα αναζήτησης θα μπορεί να δέχεται ένα ευρύ φάσμα ασαφών ερωτημάτων σε διάφορες μορφές και θα διακρίνει τις πληροφορίες με βάση άλλες πιθανές αναζητήσεις παρουσιάζοντας μετά το αποτελέσματα ομαδοποιημένα με ένα συνεκτικό τρόπο, σύμφωνα με το περιεχόμενό τους, σχολιάζοντάς τα και προτείνοντας εναλλακτικές λύσεις.[60]

1.4 - Ευφυή Συστήματα Υποστήριξης

Σαν ευφυή συστήματα υποστήριξης θεωρούμε προγράμματα που βοηθούν στην ανάπτυξη και λειτουργία σύνθετων συστημάτων. Ο ρόλος τους είναι

- η υποστήριξη ταχείας προτυποποίησης (rapid prototyping),
- η χρήση τους ως συνθετικά περιβάλλοντα πειραματικού ελέγχου (synthetic environment testbeds) [14]
- η παροχή βοήθειας για τη διαχείριση έργων (project management aid).

Ένα ευφύες σύστημα υποστήριξης ταχείας προτυποποίησης βοηθά καταλυτικά στον καθορισμό των προδιαγραφών και στο σχεδιασμό ενός συστήματος, κάνοντας χρήση τεχνικών αναπαράστασης γνώσης και εννοιών υψηλού επιπέδου. Επίσης, είναι πολύτιμο στον έλεγχο (testing) και στην επικύρωση (validation) των προδιαγραφών και του σχεδιασμού, αφού μπορεί να ανακαλύψει ασάφειες και ασυνέπειες χρησιμοποιώντας μηχανισμούς απόδειξης θεωρημάτων και αλγόριθμους λογικών συμπερασμάτων. Η ταχεία ανάπτυξη του πρωτοτύπου μπορεί να γίνει με τη χρήση τεχνικών αυτόματης ή ημιαυτόματης δημιουργίας κώδικα και με την επαναχρησιμοποίηση δομικών στοιχείων (modules) που ήδη ικανοποιούν προδιαγραφές παρόμοιες με το υπό ανάπτυξη σύστημα.[60]

Η χρήση ενός ευφυούς συστήματος υποστήριξης ως συνθετικό περιβάλλον πειραματικού ελέγχου το κάνει πολύτιμο εργαλείο στον έλεγχο της λειτουργικότητας και της απόδοσης του υπό ανάπτυξη προϊόντος, στην ανακάλυψη, διάγνωση και πρόληψη αστοχιών, ενώ μπορεί να λειτουργήσει ταυτόχρονα ως σύστημα προσομοίωσης. Τέλος, στη διαχείριση έργων, ένα ευφύες σύστημα υποστήριξης μπορεί να βοηθήσει στην καταγραφή και χρησιμοποίηση της γνώσης που αποκτάται σε όλη τη διάρκεια του έργου. Για παράδειγμα, μπορεί να καταγράψει τις αλλαγές στις διάφορες εκδόσεις των προδιαγραφών, του σχεδιασμού και της υλοποίησης ενός συστήματος, σε όλη τη διάρκεια του κύκλου ζωής του, και κατόπιν να εξηγήσει και να αιτιολογήσει τους λόγους για οποίους ελήφθησαν οι διάφορες αποφάσεις βοηθώντας στη διατήρηση της γνώσης και στη διαρκή παροχή της σε μία ομάδα

ανάπτυξης που πιθανόν αλλάζει κατά διαστήματα. Κατά τον ίδιο τρόπο μπορεί να υποστηρίξει και μελλοντικά παρόμοια έργα, προσφέροντας έτοιμες λύσεις ή εμπειρία σε παρόμοια προβλήματα.[22]

1.5 – Ευφυής Μηχανές Αυτόματου ελέγχου (Robots)

Οι ευφυής Μηχανές αυτόματου ελέγχου (robots) είναι μηχανές που, αφού προγραμματιστούν, εκτελούν ενέργειες (χειρισμού, κίνησης κτλ.) υπό συνθήκες αυτομάτου ελέγχου. Για να είναι όμως ένα robot ευφυές πρέπει να έχει ικανότητες μεγαλύτερες και διαφορετικές από αυτές των απλών robots. Έτσι, θα πρέπει να μπορεί να ενεργεί με ασφαλή και αποτελεσματικό τρόπο στο φυσικό περιβάλλον των ανθρώπων. Αυτό σημαίνει ότι αν για παράδειγμα ένα ευφυές robot μετακινείται θα μπορεί να αναγνωρίζει αντικείμενα, ποια από αυτά είναι σταθερά ή ποια μπορούν να μετακινηθούν, ποια είναι επικίνδυνα ή ποια είναι εύθραυστα, να σχεδιάζει τις ενέργειές του, να αποφασίζει ποιες ενέργειες είναι πιο αποτελεσματικές ανάλογα με το τι θέλει να επιτύχει, να αντιλαμβάνεται ήχους, χειρονομίες και φυσική γλώσσα, να μπορεί να επικοινωνήσει με τον άνθρωπο ή με άλλα robots.

Επιπλέον, ένα σημαντικό είναι η δυνατότητα συνεργασίας με άλλα ευφυή robots. Ενώ μεμονωμένα robots μπορεί να έχουν συγκεκριμένες και περιορισμένες ικανότητες, ενεργώντας ως ομάδες μπορεί να έχουν την δυνατότητα και ικανότητα εκτέλεσης σύνθετων έργων και να δίνουν λύσεις σε προβλήματα όπου απαιτείται συνδυασμός γνώσης, εμπειρίας και μηχανικής ισχύος.[54]

Επίσης, τα ευφυή robots μπορούν να χρησιμοποιηθούν σε ενέργειες που είναι επικίνδυνες ή αδύνατες για τον άνθρωπο όπως υπόγειες, υποβρύχιες ή διαστημικές αποστολές, εργασίες σε περιβάλλοντα σε τοξικής ή ραδιενεργούς μόλυνσης, κατάσβεση πυρκαγιών, ή καταστροφή ναρκών, βομβών και πυρομαχικών, καθώς και να χρησιμοποιηθούν για δουλειές οι οποίες, ενώ είναι απλές, καθημερινές και τετριμμένες για τον άνθρωπο, ωστόσο απαιτούν τη χρήση αισθήσεων, σχεδιασμό ενεργειών και τη δυνατότητα χειρισμού και κίνησης. Έτσι, μπορεί να ικανοποιηθεί πλήθος αναγκών στην βιομηχανία, στα εργαστήρια, ακόμα και στην καθημερινότητα σε κάθε νοικοκυριό.

1.6 – Ανάγκη για Υποδομή

Τα ευφυή πληροφοριακά συστήματα μπορούν να προσφέρουν νέες δυνατότητες, ευκολίες και υπηρεσίες και να συμβάλλουν με πολλαπλούς τρόπους σε όποιον τομέα της ανθρώπινης δραστηριότητας εφαρμοστούν. Ωστόσο, για να πραγματοποιηθούν και να αποδώσουν τα όσα αναφέρθηκαν στην προηγούμενη ενότητα απαιτείται να ξεπεραστούν διάφορα τεχνικά προβλήματα τα οποία έχουν ομαδοποιηθεί και είναι τα παρακάτω[60]:

- στην ευκολία χρήσης και πιο συγκεκριμένα στους τρόπους αλληλεπίδρασης ανθρώπου μηχανής,
- στις ευέλικτες και προσαρμόσιμες υπηρεσίες υποδομής, που αφορούν κυρίως νέους τρόπους αλληλεπίδρασης μεταξύ των προγραμμάτων,
- στα εργαλεία ανάπτυξης και υποστήριξης των νέων τεχνολογιών.

Και σε αυτή την περίπτωση, οι τεχνικές της Τεχνητής Νοημοσύνης και κατ' επέκταση τα ήδη υπάρχοντα ευφυή πληροφοριακά συστήματα μπορούν να παίξουν σημαντικό ρόλο για την πραγματοποίηση των μακρόπνοων στόχων και των αναμενόμενων αποτελεσμάτων.

1.7 – Ευκολία Χρήσης

Όσο οι εφαρμογές γίνονται πιο σύνθετες και ο όγκος των διαθέσιμων πληροφοριών και υπηρεσιών αυξάνεται και όσο όλο περισσότεροι χρήστες έμπειροι ή όχι χρησιμοποιούν πληροφοριακά συστήματα τόσο γίνεται πιο απαιτητική η ανάγκη για μεγαλύτερη ευκολία και πιο ευέλικτη χρήση και επικοινωνία με τους υπολογιστές. Το μέσο που είναι υπεύθυνο για την αλληλεπίδραση του ανθρώπου με τον υπολογιστή είναι το σύστημα διεπαφής χρήστη (user interface). Η σημερινή κυρίαρχη μορφή του είναι η γραφική παραθυρική (window-based graphical interface), η οποία έχει απλοποιήσει

σημαντικά τη χρήση του υπολογιστή προσφέροντας εύκολους, πολλαπλούς και πολυμεσικούς τρόπους επικοινωνίας.

Οι νέες ανάγκες όμως καθιστούν πολλές φορές ανεπαρκή αυτόν μόνο το γραφικό τρόπο αλληλεπίδρασης. Σήμερα ο μέσος χρήστης που έχει πρόσβαση στο Διαδίκτυο, έχει στη διάθεσή του ένα τεράστιο όγκο πηγών πληροφοριών και ηλεκτρονικών υπηρεσιών όπως ψυχαγωγικά, επιστημονικά ή δημοσιογραφικά άρθρα, φωτογραφίες, ήχους ή videos σχετικά με την εργασία ή τα χόμπι του, ταινίες, παιχνίδια, εφαρμογές πολυμέσων, υπηρεσίες επικοινωνίας με άλλους χρήστες, υπηρεσίες εμπορικών συναλλαγών ή ηλεκτρονικού εμπορίου κ.ά. Αυτό που χρειάζεται είναι νέοι τρόποι αλληλεπίδρασης, που θα καθιστούν εφικτή και αποτελεσματική τη χρήση των νέων προσφερόμενων αγαθών. [58]

Τις νέες αυτές αυξημένες ανάγκες καλούνται να καλύψουν τα ευφυή συστήματα διεπαφής χρήστη (intelligent user interfaces), όπου η αλληλεπίδραση με το χρήστη γίνεται σε ένα ανώτερο και πιο φυσικό επίπεδο. Τα intelligent user interfaces θα μπορούν να λειτουργήσουν ως προσωπικοί βοηθοί που θα προσαρμόζονται στις ανάγκες και στις ιδιαιτερότητες του κάθε χρήστη και θα μαθαίνουν τις συνήθειές του, παρατηρώντας τις ενέργειές του, θα κατανοούν τις ερωτήσεις, τις οδηγίες, τις εντολές ή τις αιτήσεις του χρήστη, οι οποίες θα μπορούν να τεθούν σε μία ποικιλία μορφών, όπως κείμενο, λόγος, χειρονομίες.

Σε περίπτωση που η διατύπωση είναι λανθασμένη ή ασαφής αντί να παρουσιάζουν τα γνωστά παθητικά μηνύματα λάθους, θα πρέπει να συνεργάζονται μαζί του προκειμένου να διατυπωθεί αυτό που επιθυμεί ο χρήστης με κατανοητό, αποδεκτό και εύκολο τρόπο. Οι προσωπικοί βοηθοί θα καθορίζουν μόνοι τους τις ενέργειες [58]. που πρέπει να εκτελεστούν για την επίτευξη ενός στόχου χωρίς τη συνεχή παρέμβαση του χρήστη. Αυτό που θα πρέπει τελικά να είναι απλά το ζητούμενο είναι ο χρήστης να περιγράφει αυτά που θέλει να επιτύχει με τον πιο εύκολο γι αυτόν τρόπο.

1.8 – Ευέλικτες και Προσαρμόσιμες Υπηρεσίες Υποδομής

Η ανάγκη επικοινωνίας και νέων τρόπων διαχείρισης δεδομένων και αλληλεπίδρασης δεν περιορίζεται μόνο μεταξύ ανθρώπου και υπολογιστή αλλά επεκτείνεται και μεταξύ των προγραμμάτων και συσκευών. Κάτι τέτοιο είναι φυσικό και αναμενόμενο διότι, πρώτον, για να παρασχεθούν νέες ευκολίες απαιτούνται νέα μέσα και νέες τεχνολογίες, και δεύτερον, εφόσον ο άνθρωπος απαλλάσσεται από το βάρος κάποιων ενεργειών θα πρέπει κάποιος άλλος να το επωμιστεί. Η επικοινωνία μεταξύ των προγραμμάτων αφορά τον τρόπο που αυτά θα επικοινωνούν μεταξύ τους για να ανταλλάξουν πληροφορίες, να ζητήσουν υπηρεσίες ή να συνεργαστούν για να επιτύχουν κάποιο κοινό στόχο. Ωστόσο, για να είναι αποτελεσματική μία τέτοια επικοινωνία απαιτούνται υπηρεσίες υποδομής για: [41]

- τη διαχείριση των δεδομένων και της γνώσης έτσι ώστε να επιτρέπεται η γρήγορη αναζήτηση και ανεύρεση των σχετικών πληροφοριών και πηγών μέσα από ένα τεράστιο όγκο ετερογενών και κατακεκομμένων δεδομένων,
- τη μετάφραση/μετατροπή και ενοποίηση πληροφοριών διαφόρων μορφών και μέσων σύμφωνα με τη σημασιολογία τους.

Ευφυή συστήματα αναζήτησης πληροφοριών ειδικευμένα σε συγκεκριμένα αντικείμενα και εφοδιασμένα με νέους, ισχυρούς και αποτελεσματικούς αλγόριθμους αναζήτησης αλλά και ευρετικές μεθόδους [20] μπορούν να λειτουργήσουν ως πράκτορες μεταξύ του χρήστη και των πηγών πληροφορίας. Επιπλέον, μπορούν να επικοινωνούν με άλλα αντίστοιχα συστήματα για την ανεύρεση θεμάτων σχετικών με το αντικείμενο αναζήτησης ή νεοεκδιδόμενων πληροφοριών και υπηρεσιών. Ωστόσο, η αποτελεσματικότητα τέτοιων συστημάτων απαιτεί νέες μορφές αναπαράστασης της γνώσης και υποστήριξη πολλαπλών ταξινομήσεων και σχημάτων κατηγοριοποίησης και μεθόδους αντίληψης του περιεχομένου της ανταλλασσόμενης πληροφορίας.[21]

1.9 - Εργαλεία Ανάπτυξης και Υποστήριξης

Η ανάπτυξη και η υποστήριξη των συστημάτων που θα επιτρέπουν και θα χρησιμοποιούν τις νέες μορφές αλληλεπίδρασης, απαιτούν τα αντίστοιχα εργαλεία και τις τεχνικές, εκείνες οι οποίες θα καθιστούν εφικτή την πραγματοποίηση αυτών των στόχων. Τέτοια εργαλεία είναι απαραίτητα στην ανάπτυξη προδιαγραφών για την περιγραφή του τρόπου σκέψης και συμπεριφοράς του συστήματος, στην υλοποίηση αρχιτεκτονικών συστημάτων που δρουν αυτόνομα και θα περιέχουν μηχανισμούς παρατήρησης, μάθησης και προσαρμογής στο περιβάλλον λειτουργίας, στην ανάπτυξη δομών διαχείρισης γνώσης και σχεδιασμού και επιλογής ενεργειών, καθώς και στην ανάπτυξη δομών επικοινωνίας και συνεργασίας.[60]

Επιπλέον, η υποστήριξη τέτοιων συστημάτων απαιτεί τη χρήση ανοιχτών αρχιτεκτονικών δυναμικής δραστηριοποίησης πολλαπλών αυτόνομων συστημάτων, δηλαδή αρχιτεκτονικών που επιτρέπουν την προσθήκη ή την απομάκρυνση στοιχείων τους χωρίς να μεταβάλλεται η καλή λειτουργία του όλο συστήματος. Επίσης, απαιτεί την ανάπτυξη πρωτοκόλλων και γλωσσών επικοινωνίας που θα επιτρέπουν την ανταλλαγή πληροφοριών και γενικότερα τη διαπραγμάτευση και συνεννόηση μεταξύ των συστημάτων και τέλος τη δυνατότητα επαναχρησιμοποίησης των κατάλληλων εργαλείων για την επίτευξη ενός στόχου.

1.10 - Εφαρμοσμένες Τεχνολογίες

Οι τεχνολογίες της Τεχνητής Νοημοσύνης παρέχουν μία βάση αποτελεσματικών τεχνικών και εργαλείων αλλά η μεμονωμένη χρήση και εφαρμογή τους δεν μπορεί να καλύψει τις αυξημένες ανάγκες ούτε των ευφυών υπολογιστικών συστημάτων, ούτε της απαιτούμενης υποδομής. Χρειάζεται ο συνδυασμός αυτών μέσα από μία καινοτόμο αντίληψη ανάπτυξης λογισμικού και κατ' επέκταση ευφυών συστημάτων. Στη συνέχεια θα παρουσιαστούν οι τομείς έρευνας στο πεδίο της Τεχνητής Νοημοσύνης, οι οποίοι συνδυαζόμενοι μπορούν να αποδώσουν τα αναμενόμενα αποτελέσματα.

Οι ευφυείς πράκτορες (intelligent agents) και ο προσανατολισμένος προγραμματισμός στους πράκτορες (agent oriented programming) θεωρείται τα τελευταία χρόνια η πιο επαναστατική μέθοδος αντίληψης, σχεδίασης και ανάπτυξης λογισμικού. Ο ευφυής πράκτορας είναι ένα αυτόνομο πρόγραμμα που είναι συνεχώς ενεργό, παρατηρεί και αλληλεπιδρά με το περιβάλλον όπου ενεργεί και αντιδρά στα γεγονότα που το αφορούν χωρίς την παρέμβαση του χρήστη. Ενεργεί σύμφωνα με κάποιους στόχους που έχει, οι οποίοι μπορεί να είναι μέρος ενός τελικού απώτερου στόχου. Για να επιτευχθούν αυτοί οι στόχοι μπορεί να επικοινωνεί και να συνεργάζεται με άλλους πράκτορες ή να εργάζεται σαν ομάδα μαζί τους[32].

Τα περισσότερα ευφυή συστήματα, είναι δυνατό να υλοποιηθούν ως συστήματα ενός ή περισσότερων πρακτόρων ως δομικά τους στοιχεία. Για παράδειγμα, ένα ευφές σύστημα προσομοίωσης μπορεί να αποτελείται από διάφορους πράκτορες, οι οποίοι παίζουν το ρόλο των διαφόρων οντοτήτων που συμμετέχουν στην προσομοίωση.[31]

Ο προσωπικός βοηθός (personal assistant) μπορεί να είναι ένας πράκτορας που βρίσκεται συνέχεια στο πλάι του χρήστη και τον βοηθά όπου χρειάζεται ή αναλαμβάνει να εκτελέσει εργασίες εκ μέρους του. Πράκτορες μπορούν να είναι τα ευφυή συστήματα αναζήτησης πληροφορίας (intelligent information brokers) τα οποία, ενώ βρίσκονται απομακρυσμένα σε διάφορους τόπους στο Διαδίκτυο, συνεργάζονται μεταξύ τους, για να ανευρεθεί η αναζητούμενη πληροφορία.

Οι πράκτορες μπορούν να έχουν από πολλές και σύνθετες ικανότητες έως απλές και περιορισμένες δυνατότητες και μέσα από τη συνεργασία με άλλους πράκτορες να πραγματοποιούν πολύπλοκα έργα. Το γεγονός αυτό επιτρέπει την ενσωμάτωση, σύνθεση και ολοκλήρωση διαφόρων τεχνολογιών της Τεχνητής Νοημοσύνης για την υλοποίηση ενός πράκτορα ή ενός συστήματος με πολλαπλούς πράκτορες (multi agent system). Ανάλογα λοιπόν με την εφαρμογή, μπορούν να χρησιμοποιηθούν τεχνικές αναπαράστασης γνώσης, μέθοδοι ανάκτησης πληροφορίας και ευρετικά, σχεδιασμός και επιλογή ενεργειών, τεχνικές εκμάθησης και προσαρμογής, επεξεργασία εικόνων και φυσικής γλώσσας κ.ά.

Στα συστήματα πολλαπλών πρακτόρων καθοριστικό ρόλο παίζει η οργάνωση των πρακτόρων (agent society), καθώς και η γνώση που έχει ο καθένας για τις ικανότητες των υπολοίπων[59]. Η επικοινωνία, ο συντονισμός των ενεργειών και η συνεργασία των διαφόρων πρακτόρων εξαρτάται σε μεγάλο βαθμό από αυτούς τους δύο παράγοντες. Έτσι αν θεωρήσουμε ένα εκπαιδευτικό σύστημα πολλαπλών πρακτόρων με αντικείμενο τη διδασκαλία ενός θέματος, ο εκπαιδευόμενος αλληλεπιδρά με τον κατάλληλο πράκτορα, ο οποίος γνωρίζει πώς διδάσκεται το αντικείμενο και έχει παιδαγωγικές αρχές και εκπαιδευτικούς στόχους. Τα ειδικά τεχνικά ζητήματα του αντικειμένου επιλύονται και παρέχονται σε αυτόν από άλλους πράκτορες που είναι αρμόδιοι και ειδικοί στα συγκεκριμένα θέματα, ενώ επιπλέον, αυτοί μπορούν να επικοινωνήσουν και να συνεργαστούν όταν υπάρχει επικαλυπτόμενη γνώση. Σε ένα αντίστοιχο σύστημα αναζήτησης πληροφοριών, ο πράκτορας που επικοινωνεί με το χρήστη μπορεί να μη γνωρίζει πού βρίσκεται η ζητούμενη πληροφορία, μπορεί ωστόσο να γνωρίζει άλλους πράκτορες, που έχουν τη δυνατότητα να τον πληροφορήσουν σχετικά ή να αναλάβουν οι ίδιοι την αναζήτηση.

Ένας επιπλέον παράγοντας, που σχετίζεται με την αλληλεπίδραση των συστημάτων πολλαπλών πρακτόρων και γενικότερα των ευφυών συστημάτων, είναι η κατανόηση του νοήματος της ανταλλασσόμενης πληροφορίας. Για να επιτευχθεί αυτό, το σύστημα πρέπει να έχει γνώση για τον τομέα όπου εφαρμόζεται. Η γνώση αυτή αναφέρεται ως οντολογία (ontology) του τομέα (π.χ. Οικονομία, Αρχιτεκτονική, Πληροφορική, κτλ.) και σκοπό έχει να περιγράψει το νόημα και τις σχέσεις των οντοτήτων που ανήκουν σε αυτόν τον τομέα. Οι οντολογίες αποτελούν πολύ χρήσιμα εργαλεία διότι αφενός καλύπτουν πλήθος φαινομένων και καταστάσεων και αφετέρου είναι επαναχρησιμοποιήσιμες από διάφορα συστήματα, όποτε αυτό είναι απαραίτητο. Έτσι, σχετικά με το παραπάνω παράδειγμα, το εκπαιδευτικό σύστημα χρησιμοποιώντας τις ίδιες εκπαιδευτικές στρατηγικές και τις αντίστοιχες οντολογίες, θα μπορούσε να διδάξει δερματολογία, καρδιολογία, ορθοπαιδική, κτλ. Επιπλέον, ένα ευφυές σύστημα διαπαφής χρήστη ή ένα ευφυές robot θα μπορούσε να κατανοήσει και να επικοινωνήσει σε φυσικό επίπεδο με τον άνθρωπο.

Η πλειονότητα των ευφυή πληροφοριακών συστημάτων χρησιμοποιούν και διαχειρίζονται κάποιο είδος γνώσης (συμβολικό ή αριθμητικό). Έτσι οι τεχνικές αναπαράστασης γνώσης και ο τρόπος σκέψης (λογισμός) βασισμένος στη γνώση αποτελούν βασικούς τομείς έρευνας και εφαρμογής για την Τεχνητή Νοημοσύνη. Ο στόχος είναι η ανάπτυξη μεθόδων και μορφών που αναπαριστούν ένα εύρος από πληροφορίες, από απλές και καθημερινές έως ειδικές και τεχνικές, και επιτρέπουν την κατανόηση και ταχεία παραγωγή συμπερασμάτων. Όταν η αναπαράσταση είναι σε χαμηλό επίπεδο, δηλαδή πιο κοντά στο επίπεδο της μηχανής (π.χ. βάσεις δεδομένων), τότε η ταχύτητα ανάκλησης είναι μεγάλη και επιτρέπεται η γρήγορη παραγωγή συμπερασμάτων. Ωστόσο, μόνο περιορισμένα και συγκεκριμένα κομμάτια γνώσης μπορούν να κωδικοποιηθούν με αυτή τη μορφή, ενώ παρουσιάζεται έλλειψη εκφραστικότητας και τα συμπεράσματα είναι περιορισμένα. Αντίθετα, όταν η αναπαράσταση είναι σε υψηλό επίπεδο, δηλαδή πιο κοντά στον άνθρωπο (π.χ. κείμενο φυσικής γλώσσας), η εκφραστικότητα μεγιστοποιείται, υπάρχει η δυνατότητα κωδικοποίησης μεγάλης ποσότητας κοινής γνώσης (commonsense knowledge) και τα συμπεράσματα που παράγονται είναι πιο λεπτομερή και αναλυτικά. Ωστόσο, η ανάκληση είναι κατά κανόνα πιο αργή, η διαχείρισή της πιο δύσκολη, ενώ τα συμπεράσματα πιθανόν να είναι ασαφή ή ελλιπή.

Ο συνδυασμός των δύο άκρων στην αναπαράσταση γνώσης είναι απαραίτητος, καθώς οι ευφυείς πράκτορες και γενικότερα τα ευφυή πληροφοριακά συστήματα αλληλεπιδρούν τόσο με ανθρώπους όσο και μεταξύ τους ή με άλλα μηχανικά μέρη. Σε πραγματικά συστήματα, η γνώση είναι συνήθως περιορισμένη ή μη πλήρης, ενώ το περιβάλλον και οι συνθήκες λειτουργίας πολλές φορές απαιτούν απόκριση σε περιορισμένο χρόνο και οι πληροφορίες που παρέχονται μπορεί να είναι ασαφείς, ελλιπείς ή λανθασμένες. Στην κάλυψη αυτών των αναγκών επικεντρώνεται η ανάπτυξη λογισμών με απώτερο στόχο την αντιμετώπιση του προβλήματος της αναπαράστασης, της κατανόησης και του ελέγχου της συμπεριφοράς των ευφυών συστημάτων. Ενδεικτικά αναφέρονται ο μη μονοτονικός λογισμός (non monotonic reasoning) όπου μπορούν να παραχθούν συμπεράσματα από ελλιπείς πληροφορίες και στη συνέχεια να αναιρεθούν ή να συμπληρωθούν όταν νέες πληροφορίες γίνουν διαθέσιμες, ο χρονικός λογισμός (temporal reasoning) όπου η παραγωγή συμπερασμάτων γίνεται με παραμέτρους χρόνου, ο λογισμός περιπτώσεων (case based reasoning) όπου η επίλυση προβλημάτων βασίζεται σε μεθόδους επίλυσης παλαιότερων και παρόμοιων προβλημάτων, ή τέλος τεχνικές που χρησιμοποιούν αρχές της Ασαφούς Λογικής (Fuzzy Logic) και της Θεωρίας Πιθανοτήτων (π.χ. Bayesian Networks). Όμως, προκειμένου να είναι δυνατή η συνεργασία και ο συντονισμός των

διαφόρων πρακτόρων σε ένα κοινό περιβάλλον λειτουργίας, πρέπει οι λογισμοί που χρησιμοποιούνται για τη διαχείριση της γνώσης και της συμπεριφοράς να επεκταθούν και να συμπληρωθούν με δυνατότητες επικοινωνίας και τρόπους σκέψης σχετικά με την νοητική κατάσταση τόσο του ίδιου (introspective reasoning) όσο και των υπολοίπων πρακτόρων (social reasoning).

Ο μηχανισμός σκέψης (λογισμός) ενός διαμεσολαβητή έχει άμεση σχέση με τον τρόπο που αυτός σχεδιάζει τις ενέργειές του προκειμένου να πετύχει κάποιους στόχους. Η δυνατότητα σχεδιασμού και καθορισμού ενεργειών, η εκτέλεσή τους και η παρακολούθηση των αποτελεσμάτων τους είναι απαραίτητη σε κάθε ευφυές σύστημα ή διαμεσολαβητή και απαιτεί εξελιγμένες ικανότητες αναπαράστασης και συλλογισμού σχετικά με τις ενέργειες, το χρόνο, την αντίληψη και τη νοητική κατάσταση άλλων πρακτόρων που πιθανόν συμμετέχουν στην εκπλήρωση κάποιων κοινών στόχων. Για παράδειγμα, ένα ευφυές σύστημα αναζήτησης πληροφοριών, χρειάζεται να σχεδιάσει τον καλύτερο τρόπο για να αποκτήσει μία πληροφορία εξισορροπώντας ταυτόχρονα το χρόνο που απαιτείται για την ανεύρεση, το κόστος προσπέλασης των διαφόρων βάσεων δεδομένων και την επιτυχία της αναζήτησης. Αντίστοιχα, σε ένα σύστημα προσομοίωσης, όπου συμμετέχουν διάφοροι πράκτορες και αλληλεπιδρούν τόσο μεταξύ τους όσο και με ανθρώπους, πρέπει να αντιμετωπιστούν σύνθετες καταστάσεις, που προκύπτουν από το συνδυασμό των ενεργειών αυτών όσων συμμετέχουν στην προσομοίωση, δημιουργώντας ένα δυναμικό περιβάλλον. Παρόμοιες πολύπλοκες καταστάσεις προκύπτουν στο φυσικό περιβάλλον, οι οποίες θα κληθούν να λειτουργήσουν ευφυή robots όπου ο τυχαίος και απρόβλεπτος παράγοντας παίζει πολύ σημαντικό ρόλο.

Το μεγαλύτερο και δυσκολότερο θέμα που προκύπτει είναι ότι το σύστημα πρέπει να επιλέξει το κατάλληλο για την περίπτωση πλάνο προκειμένου να πετύχει τα καλύτερα αποτελέσματα σε συγκεκριμένο χρόνο με δεδομένες τις τρέχουσες συνθήκες του περιβάλλοντος. Γι' αυτό μεγάλη ερευνητική προσπάθεια επικεντρώνεται σε τρόπους παραγωγής και επιλογής πλάνων ανάλογα με το στόχο που πρέπει να επιτευχθεί. Αυτό απαιτεί συνδυασμό τεχνικών της Τεχνητής Νοημοσύνης, της Θεωρίας Αποφάσεων και της Θεωρίας Παιγνίων.

Ένα ευφυές σύστημα μπορεί να έχει αυξημένες επιδόσεις και μεγαλύτερη ευελιξία, αν έχει δυνατότητες μάθησης και προσαρμογής σε νέες καταστάσεις. Η μηχανική μάθηση (machine learning) επικεντρώνεται κυρίως στην ανάπτυξη λογισμικού που βελτιώνεται αυτόματα μέσω εμπειρίας και στην εξαγωγή κανόνων από ένα σύνολο συγκεκριμένων δεδομένων [58]. Οι μέθοδοι μηχανικής μάθησης μπορούν να εφαρμοστούν στις περισσότερες κατηγορίες συστημάτων, που έχουν αναφερθεί παραπάνω, όπως σε προσωπικούς βοηθούς, σε συστήματα αναζήτησης πληροφοριών, σε συστήματα εκπαίδευσης και κατάρτισης, σε συστήματα υποστήριξης, σε ευφυή robots κτλ. Ήδη αρκετές μέθοδοι, όπως επαγωγή από δέντρα αποφάσεων, νευρωνικά δίκτυα, γενετικοί αλγόριθμοι κ.ά., έχουν εφαρμοστεί σε ένα ευρύ φάσμα πραγματικών προβλημάτων με αξιοσημείωτα αποτελέσματα. Ωστόσο, είναι απαραίτητες νέες μέθοδοι που θα μπορούν να εκμεταλλευτούν τις δυνατότητες που παρέχει το Διαδίκτυο, συνδυάζοντας τις πολλαπλές πληροφορίες των διαφόρων βάσεων δεδομένων και τις πληροφορίες που παρέχονται από τους χρήστες. Επιπλέον, ο συνδυασμός και η ενοποίηση των μεθόδων μηχανικής μάθησης, των μεθόδων παραγωγής και επιλογής πλάνων, καθώς και των μεθόδων επεξεργασίας φυσικής γλώσσας και εικόνας που περιγράφονται στη συνέχεια, θα αποτελέσει έναν ισχυρό μηχανισμό για την συμπεριφορά των ευφυών συστημάτων και την αλληλεπίδρασή τους με τους ανθρώπους.

Σε πολλές από τις περιπτώσεις που έχουν περιγραφεί παραπάνω, τα ευφυή πληροφοριακά συστήματα και οι ευφυείς πράκτορες είναι απαραίτητο να έχουν αυξημένες και ιδιαίτερες ικανότητες επικοινωνίας με τους χρήστες και αντίληψης του περιβάλλοντος. Για παράδειγμα, τα ευφυή robots για να λειτουργήσουν ρεαλιστικά χρειάζονται όραση, χρήση και κατανόηση φυσικής γλώσσας, ίσως και αφή. Αντίστοιχα οι ευφυείς προσωπικοί βοηθοί θα πρέπει να μπορούν να χρησιμοποιούν και να καταλαβαίνουν τη φυσική γλώσσα ή ακόμα να αντιλαμβάνονται χειρονομίες ή εκφράσεις του προσώπου. Τα συστήματα προσομοίωσης, τα συστήματα κατάρτισης και εκπαίδευσης, τα συστήματα βοήθειας για τη διαχείριση έργων θα πρέπει να έχουν δυνατότητες επεξεργασίας και σύνθεσης λόγου και εικόνας. Ο συνδυασμός της Τεχνητής Νοημοσύνης με αρχές σχετικές με τους τρόπους αντίληψης και επικοινωνίας που αφορούν τον άνθρωπο έχουν δώσει αρκετά και ικανοποιητικά αποτελέσματα και έχουν δημιουργήσει ένα υπόβαθρο τόσο στην επεξεργασία λόγου και γλώσσας όσο και στην επεξεργασία εικόνας. Ωστόσο, τα περισσότερα από τα ήδη υπάρχοντα συστήματα δίνουν τα επιθυμητά αποτελέσματα συνήθως κάτω από συγκεκριμένες και πολλές φορές ιδανικές συνθήκες

λειτουργίας. Προκειμένου να υπάρξουν συστήματα που να μπορούν να αντιλαμβάνονται και να επικοινωνούν με ρεαλιστικό τρόπο χρειάζεται περισσότερη έρευνα, βελτίωση των μεθόδων που έχουν προταθεί και ενίσχυσή τους με τη χρήση μεθόδων μηχανικής μάθησης.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΑΙΑ

Κεφάλαιο 2 – Ευφυείς Πράκτορες

2.1 - Η έννοια του πράκτορα

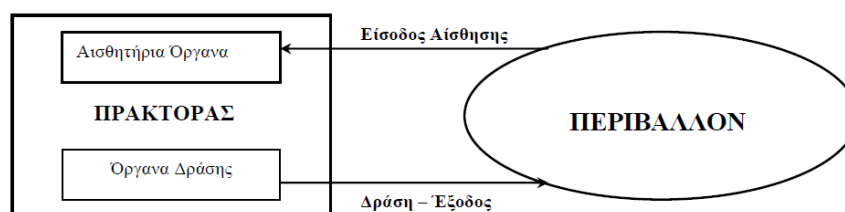
Όπως αναφέραμε και στο προηγούμενο κεφάλαιο η έννοια του πράκτορα δημιουργήθηκε κατά την δεκαετία του 1970 στο πλαίσιο ερευνών για την Κατανεμημένη Τεχνητή Νοημοσύνη [41]. Το 1977 εισήχθη από τον Carl Hewitt που πρότεινε την έννοια του “actor” ως έναν υπολογιστικό πράκτορα ο οποίος χαρακτηρίζεται από την διεύθυνσή του και την συμπεριφορά του, ενώ έχει μία εσωτερική κατάσταση και μπορεί να απαντά σε μηνύματα άλλων παρόμοιων αντικειμένων [53].

Στις δεκαετίες που ακολούθησαν υπήρξε αρκετή έρευνα σχετικά με τους συνεργατικούς και πολλαπλούς πράκτορες με έμφαση στην αλληλεπίδραση και επικοινωνία μεταξύ πρακτόρων με σκοπό τον καταμερισμό και συντονισμό των εργασιών. Την δεκαετία του 1990 και μετά η ιδέα του πράκτορα λογισμικού πήρε ευρύτερη έννοια καθώς πολλοί οργανισμοί, Πανεπιστήμια και εταιρείες ασχολούνται με την τεχνολογία αυτή και είναι δύσκολο να δοθεί ένας σαφής ορισμός όσον αφορά το τι είναι πράκτορας λογισμικού σε αντίθεση με ένα απλό πρόγραμμα. Ο λόγος ο οποίος καθιστά δύσκολη την εύρεση ενός ορισμού για το τι είναι πράκτορας είναι τα χαρακτηριστικά τα οποία συσχετίζονται με αυτόν και την λειτουργία του, τα οποία δεν έχουν την ίδια ακριβώς σημασία σε όλους τους επιστημονικούς τομείς. Έτσι, για παράδειγμα, η ικανότητα μάθησης ενός πράκτορα, σύμφωνα με τις εμπειρίες του έχει μεγάλη σημασία για κάποιες εφαρμογές, ενώ για κάποιες άλλες η συγκεκριμένη ικανότητα είναι ασήμαντη έως ανεπιθύμητη.

Σύμφωνα, λοιπόν, με τους M. Woolbribge και N. Jennings: «Ένας πράκτορας είναι ένα υπολογιστικό σύστημα το οποίο είναι εγκατεστημένο σε ένα περιβάλλον και έχει την ικανότητα αυτόνομης δράσης στο συγκεκριμένο περιβάλλον προκειμένου να ανταποκριθεί στους σκοπούς σχεδίασης του.»[58]

Είναι πολύ σημαντικό να αναφέρουμε ότι ο συγκεκριμένος ορισμός αναφέρεται σε πράκτορες και όχι σε ευφυείς πράκτορες και ότι επίσης δεν έχει οριστεί ούτε η έννοια της αυτονομίας αλλά ούτε και η αποσαφήνιση του τύπου περιβάλλοντος.

Ένας όρος ο οποίος είναι συνδεδεμένος με την έννοια του Πράκτορα, είναι αυτός της αυτονομίας. Αυτονομία είναι η ικανότητα του πράκτορα να δρα χωρίς την παρέμβαση ανθρώπων ή άλλων συστημάτων, δηλαδή η ικανότητα του πράκτορα να γνωρίζει και να ελέγχει την κατάσταση του και εσωτερικά και εξωτερικά.



Το παραπάνω σχήμα δείχνει πως επιδρά ένας πράκτορας στο περιβάλλον του. Στα περισσότερα πεδία λογικής πολυπλοκότητας, ένας πράκτορας δεν έχει τον απόλυτο έλεγχο του περιβάλλοντός του. Απλά υπάρχουν περιπτώσεις όπου θα έχει μερικό έλεγχο, δηλαδή θα μπορεί να το επηρεάζει απόλυτα.

Αυτό για τον πράκτορα σημαίνει ότι αν επαναλάβει την ίδια κίνηση δεύτερη φορά κάτω από τις ίδιες συνθήκες, είναι πολύ πιθανόν να έχει εντελώς διαφορετικό αποτέλεσμα το οποίο είναι πολύ πιθανόν

να μην είναι και το επιθυμητό. Αυτό σημαίνει πως ένας πράκτορας πρέπει να είναι προετοιμασμένος για την πιθανότητα της αποτυχίας.

Συνήθως ένας πράκτορας για κάθε χρονική στιγμή έχει στην διάθεσή του ένα δεδομένο σύνολο κινήσεων δράσης. Το εύρος αυτών των κινήσεων αναπαριστά την ικανότητα του πράκτορα να επιδρά, δηλαδή την ικανότητα του να αλλάζει το περιβάλλον του. Εδώ μπορεί να σημειωθεί ότι δεν είναι πάντα εφικτές όλες οι κινήσεις ενός πράκτορα. Πχ η πράξη βάλει 50 λίτρα βενζίνη από ένα βενζινάδικο θα είναι εφικτή μόνο αν υπάρχουν τα συγκεκριμένα λίτρα στη συγκεκριμένο βενζινάδικο.

Το βασικό πρόβλημα αντιμετωπίζοντας έναν πράκτορα είναι να αποφασιστεί ποια από τις ενέργειες του πρέπει να πραγματοποιηθούν, για να μπορέσουν να πετύχουν του σχεδιαστικούς του στόχους. Οι αρχιτεκτονικές των πρακτόρων είναι αρχιτεκτονικές λογισμικού για συστήματα λήψης αποφάσεων τα οποία είναι ενσωματωμένα σ' ένα περιβάλλον. Η πολυπλοκότητα της διαδικασίας της λήψης αποφάσεων μπορεί να επηρεαστεί από ένα σύνολο περιβαλλοντικών ιδιοτήτων. Ο Russel και ο Norvig πρότειναν τις ακόλουθες ταξινομήσεις στις ιδιότητες του περιβάλλοντος:[50]

- **Προσβάσιμες εναντίον μη προσβάσιμες:** Ένα προσβάσιμο περιβάλλον θεωρείται αυτό στο οποίο ο πράκτορας μπορεί να αποκτήσει ακριβές και σύγχρονες πληροφορίες για την κατάσταση του περιβάλλοντος του. Πολλά μέτρια και πολύπλοκα περιβάλλοντα είναι συνήθως μη προσβάσιμα. Ένα παράδειγμα τέτοιου περιβάλλοντος είναι το Internet. Το πιο προσβάσιμο είναι συνήθως το περιβάλλον όπου οι πράκτορες είναι εύκολο να δημιουργηθούν έτσι ώστε να λειτουργούν γι' αυτό.
- **Ντετερμινιστικό εναντίον μη Ντετερμινιστικό:** Ντετερμινιστικό θεωρείται το περιβάλλον στο οποίο κάθε ενέργεια έχει ένα μοναδικό εγγυημένο αποτέλεσμα, και δεν υπάρχει αμφιβολία για την κατάσταση η οποία προκύπτει από την διαδικασία συγκεκριμένης ενέργειας. Το φυσικό περιβάλλον συνήθως θεωρείται μη ντετερμινιστικό. Το παρόν των μη ντετερμινιστικών περιβαλλόντων δημιουργεί προβλήματα στο σχεδιασμό των πρακτόρων.
- **Κινητικό εναντίον μη κινητικό:** Σ' ένα κινητικό περιβάλλον οι επιδόσεις των πρακτόρων εξαρτώνται από τον αριθμό διακριτών κινήσεων, χωρίς να υπάρχει σχέση μεταξύ των επιδόσεων των πρακτόρων σε άλλα σενάρια. Παράδειγμα τέτοιου περιβάλλοντος είναι ένα σύστημα mail. Τα συγκεκριμένα είναι πιο απλά σε σχέση με τις προοπτικές του δημιουργού του πράκτορα, γιατί ο πράκτορας αποφασίζει για τις ενέργειες τις οποίες θα εκτελέσει βασισμένος στη τωρινή κίνηση και δεν χρειάζεται λόγο για την αλληλεπίδραση μεταξύ αυτών και των μελλοντικών κινήσεων.
- **Στατικά εναντίον δυναμικών:** Το στατικό περιβάλλον είναι αυτό το οποίο υποτίθεται ότι μπορεί να μείνει αμετάβλητο ανεξάρτητα από τις επιδόσεις των ενεργειών ενός πράκτορα. Δυναμικό περιβάλλον είναι αυτό το οποίο αποτελείται από άλλες διεργασίες που λειτουργούν σε αυτό, και το οποίο εμποδίζει τις αλλαγές οι οποίες αφορούν τον έλεγχο των πρακτόρων. Το φυσικό περιβάλλον είναι αρκετά δυναμικό.
- **Διακριτά εναντίον συνεχόμενων:** Ένα παράδειγμα διακριτών συστημάτων είναι ένα παιχνίδι σκάκι και συνεχόμενων η οδήγηση ταξί. Συνεχόμενο θεωρείται το περιβάλλον που αποτελείται από συνεχόμενο αριθμό πράξεων.

2.2 Επιπλέον ορισμοί

Κάποιοι επιπλέον ορισμοί των ευφυών πρακτόρων είναι:

- «Ένας Ευφυής Πράκτορας είναι λογισμικό που μπορεί να κάνει πράγματα που πιθανότατα θα έκανε ο κάθε χρήστης, αν είχε το χρόνο» [19]
- «Ένα κομμάτι λογισμικού, το οποίο προκειμένου να κινηθεί κατάλληλα για την επιτυχή διεκπεραίωση μιας δοσμένης εργασίας, χρησιμοποιεί πληροφορίες που έχει αποκτήσει από το περιβάλλον του.» [26]

- «Οι Νοήμονες Πράκτορες είναι υπολογιστικά συστήματα, που κατοικοεδρεύουν σε πολύπλοκα, δυναμικά περιβάλλοντα, έχοντας δυνατότητες αίσθησης κι αυτόνομης δράσης μέσα σ' αυτά και έτσι αντιλαμβάνονται ένα σύνολο στόχων ή εργασιών, για την επίτευξη των οποίων είναι σχεδιασμένοι. Οι πράκτορες αυτοί μπορούν να πάρουν διάφορες μορφές ανάλογα με τη φύση του περιβάλλοντος κατοικίας τους (Αυτόνομα Ρομπότ - Πράκτορες Λογισμικού, Knowbots - Πράκτορες Διεπιφανειών Εργασίας). » [10]

2.3 - Παραδείγματα πρακτόρων

Σαν πράκτορα μπορούμε να θεωρήσουμε κάθε σύστημα ελέγχου. Ένα απλό παράδειγμα πράκτορα είναι το σύστημα με το οποίο λειτουργούν τα φώτα από τα καινούργια αμάξια. Το συγκεκριμένο σύστημα έχει ένα αισθητήρα για τον εντοπισμό της φωτεινότητας του περιβάλλοντος. Αυτός ο αισθητήρας είναι άμεσα ενσωματωμένος στο περιβάλλον και παράγει σαν έξοδο δύο σήματα: το πρώτο σήμα μας υποδεικνύει ότι το φως είναι λίγο κι ένα που υποδεικνύει ότι το φως είναι αρκετό. Το συγκεκριμένο σύστημα παράγει δύο εξόδους: «άνοιγμα φώτων» ή «κλείσιμο φώτων». Το υπεραπλοστευμένο σύστημα αποφάσεων του θερμοστάτη υλοποιείται με τους ακόλουθους κανόνες:

- Χαμηλή φως -> Άνοιγμα φώτων
- Φως αρκετό -> Κλείσιμο φώτων

Συνοψίζοντας, οι πράκτορες είναι υπολογιστικά συστήματα, ικανά να δρουν αυτόνομα σε ένα συγκεκριμένο περιβάλλον προσπαθώντας να φέρουν σε πέρας τους σχεδιαστικούς τους σκοπούς. Γενικά ένας πράκτορας αντιλαμβάνεται το περιβάλλον του με φυσικούς αισθητήρες, αν πρόκειται για πράκτορα σε κάποιο κομμάτι φυσικού κόσμου, ή με αισθητήρες λογισμικού, αν πρόκειται για πράκτορα λογισμικού. Έτσι έχει διαθέσιμες κινήσεις για κάθε περίπτωση, η οποία θα τον βοηθήσει ώστε να μπορέσει να διαφοροποιήσει το περιβάλλον, το οποίο πιθανότατα απαντά μη-ντετερμινιστικά στην εκτέλεση των κινήσεων αυτών.

2.4 - Χαρακτηριστικά πρακτόρων

Οι πράκτορες φαντάζουν ως ένα μοντέλο μιας νέας γενιάς λογισμικού που αφορά την πολύπλοκα και «διαστρεβλωμένα» συστήματα. Οι πράκτορες επίσης χρησιμοποιούνται ως ένα πλαίσιο το οποίο θα φέρει κοντά το συστατικό Τεχνικής Νοημοσύνης το οποίο είναι απαραίτητο για να σχεδιαστούν και να κατασκευαστούν έξυπνες οντότητες.

Ένας πράκτορας είναι ένα υπολογιστικό σύστημα το οποίο είναι τοποθετημένο μέσα σε ένα περιβάλλον το οποίο είναι ικανό και ευέλικτο, έτσι ώστε με την βοήθεια αυτόνομων ενεργειών στο συγκεκριμένο περιβάλλον να εκπληρώνονται οι σχεδιαστικοί στόχοι του συστήματος.

Οι πράκτορες θα χρειαστεί να αλληλεπιδρούν μεταξύ τους, είτε για να επιτύχουν τους αυτόνομους στόχους τους, είτε για να διαχειριστούν τις εξαρτήσεις που ακολουθούν μετά την τοποθέτησή τους σε ένα σύνθετο περιβάλλον. Αυτές οι αλληλεπιδράσεις μπορούν να είναι είτε απλές συναλλαγές, είτε ερωτήσεις για συγκεκριμένες ενέργειες οι οποίες θα μπορούν να διενεργηθούν και στο συντονισμό και στη συνεργασία.

Εντούτοις, πιθανώς ο πιο θεμελιώδης και σημαντικός μηχανισμός έτσι ώστε να διαχειριστούν οι εξαρτήσεις μεταξύ των πρακτόρων στο χρόνο εκτέλεσης είναι οι διαπραγματεύσεις, η διαδικασία κατά την οποία μια ομάδα πρακτόρων έρχεται σε αμοιβαία συμφωνία. Οι διαπραγματεύσεις υποστηρίζουν τις προσπάθειες για συντονισμό και συνεργασία, δύο στοιχεία τα οποία χρειάζονται οι πράκτορες έτσι ώστε να μπορούν να αλληλεπιδρούν μεταξύ τους.

Τα χαρακτηριστικά τα οποία ξεχωρίζουν τους ευφυείς πράκτορες από άλλου τύπου εφαρμογές λογισμικού:[19]

- **Αυτονομία:** Οι ευφυείς πράκτορες πρέπει να έχουν την ικανότητα να ενεργούν με σκοπό την ολοκλήρωση διαφόρων εργασιών ή αντικειμενικών σκοπών χωρίς ενεργοποίηση ή ώθηση από το χρήστη. Πρέπει να υπάρχει ένα στοιχείο αυτονομίας στον πράκτορα. Όπως ακριβώς συμβαίνει και με τους ανθρώπους πράκτορες οι οποίοι παίρνουν τις κατάλληλες κατευθύνσεις, τα ενδιαφέροντα και επιθυμίες μας ως είσοδο και κατόπιν παίρνουν τις κατάλληλες αποφάσεις για να εκτελέσουν τις αναμενόμενες διεργασίες.
- **Ικανότητα επικοινωνίας:** Για έναν ευφυή πράκτορα βασικός του στόχος πρέπει να είναι η προσπάθεια για να πετύχει τους αντικειμενικούς σκοπούς του, να αποκτά πρόσβαση σε πληροφορίες από πηγές τρίτων μερών, σχετικά με την παρούσα κατάσταση του περιβάλλοντος. Αυτό απαιτεί δυνατότητα επικοινωνίας με τα αποθηκευτικά μέσα αυτής της πληροφορίας. Αυτά μπορεί να είναι άλλοι πράκτορες ή φύλακες των χώρων αποθήκευσης.
- **Ικανότητα συνεργασίας:** Ένας ευφυής πράκτορας πρέπει να έχει πνεύμα συνεργασίας για να επιβιώσει και να επιτύχει σε αυτό που ονομάζεται 'agent-oriented συστήματα'. Το όραμα για τους ευφυείς πράκτορες είναι να εργάζονται μαζί έτσι ώστε να μπορούν να εκτελούν αμοιβαία και επικερδή άλλες σύνθετες διεργασίες. Αυτό είναι το σημείο κατά το οποίο η έρευνα για τους ευφυείς πράκτορες διασταυρώνεται με την τεχνητή νοημοσύνη.
- **Ικανότητα συλλογισμού:** Η ικανότητα συλλογισμού αποτελεί ένα από τα σημαντικά κομμάτια έτσι ώστε να μπορούμε να διαχωρίζουμε τους ευφυείς πράκτορες από άλλους πιο 'robotic' πράκτορες. Η ικανότητα του συλλογισμού μας δείχνει ότι ένας πράκτορας σύμφωνα με συγκεκριμένες γνώσεις που έχει θα μπορεί να συμπεραίνει, με έναν ορθολογιστικό και αναπαράξιμο τρόπο
- **Προσαρμοστική συμπεριφορά:** Για να διατηρεί ένας ευφυής πράκτορας τις ικανότητες αυτονομίας και συλλογισμού πρέπει να είναι εφοδιασμένος με κάποιο μηχανισμό κατά τον οποίο θα μπορεί να εκτίμη της παρούσας κατάστασης του εξωτερικού περιβάλλοντος δίνοντας μια πληροφορία στον πράκτορα, την οποία θα ενσωματώνει στις αποφάσεις του για μελλοντικές πράξεις. Ο ευφυής πράκτορας πρέπει να έχει την ικανότητα να εξετάζει το εξωτερικό περιβάλλον καθώς και το αποτέλεσμα το οποίο είχαν προηγούμενες ενέργειες οι οποίες έγιναν κάτω από παρόμοιες συνθήκες και σαν συνέπεια αυτού να μπορεί να προσαρμόζει τις ενέργειες του με σκοπό την αύξηση του ποσοστού για την επίτευξη των στόχων του.
- **Αξιοπιστία:** Βασικό στοιχείο στην αποδοχή των ευφυών πρακτόρων δυνατότητα κατά την οποία μπορούν να αντιπροσωπεύουν τον χρήστη τους. Με άλλα λόγια, ο χρήστης θα πρέπει να είναι σίγουρος πως ο πράκτορας θα ενεργήσει και θα δώσει αναφορά με ειλικρίνεια, και θα ενεργήσει για το καλό του χρήστη.
- **Συνεργατική Συμπεριφορά:** Το συγκεκριμένο χαρακτηριστικό δημιουργήθηκε με βάση την ιδέα της ικανότητας για επικοινωνία που προαναφέρθηκε. Οι ερευνητές ασχολούνται με σύνολα από agents ή multi-agents (MAS). Κάθε agent δίνει αποδίδει ξεχωριστό έργο. Θα υπάρξουν φορές βέβαια που θα λειτουργούν παράλληλα, όπως όταν βρουν τις ίδιες πληροφορίες από διαφορετικές πηγές. Βασικό είναι να μπορούν να συνεργάζονται ώστε να καθορίζουν ποιος agent θα εκτελέσει το κάθε έργο και πως θα συγχωνεύσουν τις πληροφορίες που συλλέξανε για παρουσίαση στον χρήστη. Οι πράκτορες υποχρεούνται να λειτουργούν σε σχέδιο με τους άλλους πράκτορες, πιθανόν μέσω μιας γλωσσάς επικοινωνίας προκειμένου να επιτύχουν ένα κοινό στόχο. Επίσης οι agents είναι δυνατόν να ανταλλάσσουν γνώσεις και εμπειρίες κατά τη διάρκεια της συνεργασίας τους.
- **Ευκινησία:** Η ευκινησία είναι απαραίτητη για agents του δικτύου. Αυτή η ιδέα αναφέρεται στην ικανότητα του agent να μεταναστεύει με έναν αυτοελεγχόμενο τρόπο από το ένα στο άλλο τμήμα του δικτύου με την προϋπόθεση να εκτελεί τα προσδιορισμένα καθήκοντα του. Στα

καθήκοντα του μπορεί να περιλαμβάνονται η συνάθροιση πληροφοριών ή η κίνηση στο δίκτυο. Γενικά μπορεί να θεωρηθεί σαν μια προέκταση της ιδέας για αυτονομία.

Οι αυτοματοποιημένες έρευνες διαπραγματεύσεις μπορούν να εξετάσουν τη σχέση με τρία ευρέως θέματα:[19]

- **Πρωτόκολλα διαπραγμάτευση:** Είναι το σύνολο των κανόνων που κυβερνούν τις διαπραγματεύσεις. Αυτό καλύπτει τους επιτρεπτούς τύπους συμμετεχόντων, το status των διαπραγματεύσεων(π. χ. Αποδοχή πρότασης, κλείσιμο διαπραγματεύσεων κ.λπ.), τα γεγονότα τα οποία οδηγούν στην αλλαγή του status των διαπραγματεύσεων και τις έγκυρες ενέργειες των συμμετεχόντων σε μία συγκεκριμένη κατάσταση διαπραγμάτευσης.
- **Αντικείμενα Διαπραγμάτευσης:** Το όριο των θεμάτων κατά το οποίο μπορεί να επιτευχθεί μια συμφωνία. Μπορεί ένα αντικείμενο να έχει μόνο έναν τομέα π. χ. τιμή , αλλά μπορεί να έχει και πολλούς τομείς. Στην πιο απλή περίπτωση, η κατασκευή και τα στοιχεία της συμφωνίας είναι φτιαγμένα και οι συμμετέχοντες μπορούν είτε να δεχθούν είτε να απορρίψουν τη συμφωνία. Στο επόμενο επίπεδο, οι συμμετέχοντες έχουν την ευελιξία να αλλάξουν τις τιμές των τομέων στο αντικείμενο διαπραγματεύσεων. Τέλος οι συμμετέχοντες έχουν τη δυνατότητα να αλλάξουν δυναμικά την κατασκευή του αντικειμένου το οποίο διαπραγματεύονται.
- **Μοντέλο απόφασης Πρακτόρων:** Η συσκευή απόφασης κάνει του συμμετέχοντες να ενεργούν σύμφωνα με το πρωτόκολλο διαπραγμάτευσης έτσι ώστε να επιτύχουν τους στόχους τους. Η εκλέπτυνση του μοντέλου, όπως και το όριο των αποφάσεων που έχουν να παρθούν επηρεάζονται από το πρωτόκολλο το οποίο είναι σε ισχύει, από τη φύση του αντικειμένου το οποίο διαπραγματευόμαστε και από το όριο των διενεργειών που μπορούν να εκτελεστούν σ' αυτό.

Η σημασία των τριών αυτών τομέων ποικίλει ανάλογα τις διαπραγματεύσεις και το αντίστοιχο περιβάλλον. Έτσι σε κάποιες περιπτώσεις το πρωτόκολλο διαπραγμάτευσης είναι η κυρίαρχη έννοια.

Οι διαπραγματεύσεις μπορούν να φανούν σαν διανεμημένη έρευνα μέσα σένα χώρο αμφιβόλων συμφωνιών. Η διστακτικότητα και η τοπολογία του συγκεκριμένου χώρου αποφασίζεται από την κατασκευή του αντικειμένου διαπραγμάτευσης. Έτσι όταν προστίθεται ή αφαιρείται κατά την διάρκεια της διαπραγμάτευσης, τότε προστίθενται ή αφαιρούνται νέες διαπραγματεύσεις και οι πόντοι συμφωνίας μπορούν να αυξηθούν η να μειωθούν ανάλογα. Απλά, όταν ένας πράκτορας αλλάζει μία από τις τιμές ενός από τα χαρακτηριστικά του κατά τη διάρκεια μιας προσφοράς, μεταφέρεται από τον ένα σημείο της συμφωνίας στο άλλο.

Στην αρχή μία διαδικασίας κάθε πράκτορας έχει μια αναλογία στο χώρο στον οποίο θέλει να γίνει η συμφωνία. Κατά τη διάρκεια της διαδικασίας ο χώρος συμφωνίας των συμμετεχόντων μπορεί να αλλάξει γιατί το περιβάλλον αλλάζει ή γιατί είναι πεπεισμένοι να αλλάξουν τις απόψεις τους. Η έρευνα τελειώνει όταν ένας απαιτούμενος αριθμός συμμετεχόντων βρει ένα αμοιβαίο σημείο συμφωνίας ή όταν το πρωτόκολλο αποφασίσει πώς η συμφωνία πρέπει να τερματιστεί χωρίς να επιτευχθεί συμφωνία.

Οι ελάχιστες απαιτήσεις για έναν πράκτορα που έχει σχέση με τις διαπραγματεύσεις είναι να μπορεί να δημιουργεί και να απαντά σε προτάσεις. Εάν ένας πράκτορας μπορεί μόνο να απορρίπτει ή να δέχεται προτάσεις, τότε η διαπραγμάτευση μπορεί να είναι πολύ καιρό καταναλωτική και ανεπαρκής εφόσον αυτός που την προτείνει δεν έχει κανένα μέσο γιατί η πρόταση είναι μη αποδεκτή, ούτε όταν ο πράκτορας είναι κοντά σε συμφωνία, ούτε σε οποιαδήποτε διάσταση του χώρου συμφωνίας θα μπορεί να πάει μετά.

Για να βελτιώσουμε την ικανότητα τις διαπραγματευτικής διαδικασίας, ο παραλήπτης πρέπει να παρέχει μια χρήσιμη ανατροφοδότηση στις προτάσεις τις οποίες δέχεται. Αυτή η ανατροφοδότηση παίρνει τη μορφή κριτικής ή μιας αντιπρότασης. Από αυτή την ανατροφοδότηση, αυτός που προτείνει θα μπορεί να είναι στη θέση να γεννά μια πρόταση η οποία είναι πολύ πιθανό να επιφέρει συμφωνία.

2.5 - Πώς πρέπει να λειτουργούν οι πράκτορες

Ένας λογικός πράκτορας είναι αυτός ο οποίος κάνει τη σωστή κίνηση. Προφανώς, αυτό αποτελεί καλύτερη κίνηση από το να κάνει κάτι λάθος. Σαν πρώτη προσέγγιση, θα πούμε ότι η σωστή δράση είναι αυτή που θα αναγκάσει τον πράκτορα για να είναι πιο σωστός. Αυτός μας δημιουργεί το πρόβλημα της αποφασιστικότητας, δηλαδή το να μπορούμε να αποφασίσουμε πότε ένας πράκτορας είναι επιτυχής.

Χρησιμοποιούμε ένα μέτρο απόδοσης για το πώς, αλλά και κριτήρια τα οποία καθορίζουν κατά πόσο είναι επιτυχής ένας πράκτορας. Προφανώς βέβαια δεν υπάρχει ένα σταθερό μέτρο για να μπορούμε να αξιολογήσουμε όλους τους πράκτορες. Θα μπορούσαμε να ρωτήσουμε έναν πράκτορα κατά πόσο είναι ικανοποιημένος με την απόδοσή του, αλλά επειδή μερικοί πράκτορες θα ήταν ανίκανοι να απαντήσουν, και άλλοι θα εξαπατούνταν. (Οι ανθρώπινοι πράκτορες ειδικότερα είναι πασίγνωστοι για τα «ξινά σταφύλια», δηλαδή δεν ζητάνε τίποτα από τη στιγμή που κρίθηκαν ανεπιτυχής) επομένως θα επιμείνουμε στην εύρεση ενός κοινού και αντικειμενικού μέτρου απόδοσης το οποίο θα στηρίζεται σε μια συγκεκριμένη αρχή. Με άλλα λόγια, ως εξωτερικοί παρατηρητές δημιουργούμε πρότυπα τα οποία μας οδηγούν στο αποτέλεσμα ότι ένας πράκτορας είναι επιτυχής και θα μπορούν να χρησιμοποιηθούν για να μετρήσουν την απόδοση των πρακτόρων[19].

Για παράδειγμα, ας εξετάσουμε την περίπτωση ενός πράκτορα ο οποίος πρέπει να σκουπίσει ένα βρώμικο πάτωμα με ηλεκτρική σκούπα. Ένα εύλογο μέτρο απόδοσης θα ήταν το ποσό ρύπου καθώς και το σύνολο το οποίο καθαρίστηκε μέσα σε οκτώ ώρες. Ένα περιπλοκότερο μέτρο απόδοσης είναι το ποσό ηλεκτρικής ενέργειας που καταναλώθηκε και το ποσό θορύβου που παρήχθη επίσης. Μια τρίτη δύναμη μέτρου απόδοσης η οποία δίνει υψηλή απόδοση στον πράκτορα που θα μπορούσε να είναι ο ήσυχος και αποτελεσματικός καθαρισμός, αλλά και να δημιουργείται ελεύθερος χρόνος το Σαββατοκύριακο.

Το πότε θα αξιολογηθεί η απόδοση ενός πράκτορα αποτελεί κάτι εξίσου σημαντικό. Εάν μετρήσαμε το πόσο του ρύπου αλλά και το τι έχει καθαρίσει ένας πράκτορας την πρώτη ώρα της ημέρας, θα μπορούσαμε να ανταμείψουμε εκείνους τους πράκτορες που αρχίζουν γρήγορα (ακόμα κι αν μετά την μια ώρα κάνουν ελάχιστη εργασία ή ακόμα και τίποτα), και θα τιμωρούσαμε εκείνοι που λειτουργούν με συνέπεια και αργούν γενικά. Κατά συνέπεια, θέλουμε να μετρήσουμε την απόδοση μακροπρόθεσμα, είτε πρόκειται για μια οκτώ ωρών μετατόπιση ή μια διάρκεια ζωής.

Πρέπει να είμαστε προσεκτικοί για να μπορούμε να διακρίνουμε τη διαφορά μεταξύ της ορθολογιστικής ικανότητας και της παντογνωσίας. Ένας πάνσοφος πράκτορας ξέρει την πραγματική έκβαση των ενεργειών του, και μπορεί να ενεργήσει αναλόγως αλλά η παντογνωσία είναι αδύνατη στην πραγματικότητα. Για να γίνει καλύτερη ανάλυση υπάρχει το ακόλουθο παράδειγμα: Περπατώ κατά μήκος της Κηφισίας μια ημέρα και βλέπω έναν παλιό φίλο από την άλλη μεριά του δρόμου. Δεν έχει κίνηση ο δρόμος και όντας λογικός, αρχίζω να διασχίζω τον δρόμο. Εν τω μεταξύ, σε 33.000 πόδια, ένα φορτίο πέφτει από ένα επιβατικό αεροσκάφος καθώς περνάω το δρόμο και προτού να περάσω την άλλη πλευρά του δρόμου πέφτει πάνω μου. Ήταν παράλογο να περάσω το δρόμο; Είναι απίθανο ότι η νεκρολογία μου θα διάβαζε «ηλίθιε προσπάθησες να διασχιστεί τον δρόμο.»

Μάλλον, αυτό μας οδηγεί στο συμπέρασμα ότι η ορθολογιστική ικανότητα ενδιαφέρεται για την αναμενόμενη επιτυχία και όχι για το τι γίνεται αντιληπτό. Το πέρασμα της οδού ήταν λογικό επειδή τις περισσότερες φορές το πέρασμα θα ήταν επιτυχές, και δεν υπήρξε κανένας τρόπος που θα μπορούσα να έχω προβλέψει φορτίο. Σημειώστε ότι ένας άλλος πράκτορας που εξοπλίστηκε με το ραντάρ για την ανίχνευση των φορτίων και την απόκρουση τους θα ήταν επιτυχέστερο, αλλά δεν θα ήταν άλλο λογικό.

Με άλλα λόγια, δεν μπορούμε να κατηγορήσουμε έναν πράκτορα για την αποτυχία να λάβει υπόψη κάτι που δεν θα μπορούσε να αντιληφθεί, ή για την αποτυχία να λάβει μέτρα (όπως η απόκρουση της πόρτας φορτίου) ότι είναι ανίκανος. Αλλά η χαλάρωση στο γεγονός να μην απαιτείται η τελειότητα δεν αποτελεί μια απλή ερώτηση η οποία θα ήταν δίκαιη στους πράκτορες. Το θέμα είναι πως αν θέσουμε ως βασικό παράγοντα πως ένας ένας ευφυής πράκτορας πρέπει πάντα να κάνει αυτό που θεωρείται σωστό, αυτό θα μας οδηγήσει στο συμπέρασμα στο ότι είναι αδύνατο να σχεδιαστεί ένας πράκτορας

για να εκπληρώσει αυτή την προδιαγραφή εκτός αν βελτιώσουμε και τις άλλες παραμέτρους. Εν περιλήψει, τι είναι λογικός οποιαδήποτε στιγμή εξαρτάται από τέσσερα πράγματα[19]:

- Το μέτρο απόδοσης το οποίο εκφράζει το μέτρο της επιτυχίας
- Ότι έχει αντιληφθεί ο πράκτορας μέχρι τώρα ,θα μπορέσουμε να το θεωρήσουμε ιστορικό ποσοστό συνέχειας.
- Ότι γνωρίζουν οι πράκτορες για το περιβάλλον
- Οι ενέργειες τις οποίες μπορεί να εκτελέσει ένας πράκτορας

Αυτό οδηγεί σε δημιουργία ενός ιδανικού λογικού πράκτορα: Για κάθε πιθανή ακολουθία percept, ένας ιδανικός λογικός πράκτορας πρέπει να κάνει οποιαδήποτε δράση χρειαστεί για να μεγιστοποιήσει το μέτρο απόδοσής της, βάσει των στοιχείων που παρέχονται από την ακολουθία percept και οποιαδήποτε ενσωματωμένη γνώση έχει ένας πράκτορας. Ο καθορισμός του συγκεκριμένου πράκτορα πρέπει να γίνει με αρκετή προσοχή. Καταρχάς σαν πρώτη ματιά φαίνεται να επιτρέπεται στον πράκτορα να εκτελεί μερικές μη έξυπνες δραστηριότητες. Παραδείγματος χάριν, εάν ένας πράκτορας δεν λάβει υπόψη του και τις δύο παραμέτρους πριν διασχίζει έναν πολυάσχολο δρόμο, τότε η ακολουθία percept δεν θα μπορεί να τον ενημερώσει εάν πλησιάζει ένα μεγάλο φορτηγό με υψηλή ταχύτητα. Ο καθορισμός φαίνεται να λέει ότι θα ήταν εντάξει για τον για να διασχίσει το δρόμο. Στην πραγματικότητα, αυτή η ερμηνεία κάνει λάθος σε δύο αριθμήσεις. Κατ' αρχάς, δεν θα ήταν λογικό να διασχιστεί ο δρόμος: ο κίνδυνος χωρίς κοίταγμα είναι πάρα πολύ μεγάλος. Δεύτερον, ένας ιδανικός λογικός πράκτορας θα είχε επιλέξει «να φανεί» δράση πριν περπατήσει τον δρόμο, επειδή το κοίταγμα βοηθάει στο να μεγιστοποιηθεί η αναμενόμενη απόδοση.

Η έννοια ενός πράκτορα προορίζεται να είναι ένα εργαλείο για τα συστήματα, όχι ένας απόλυτος χαρακτηρισμός που διαιρεί τον κόσμο σε πράκτορες και μη-πράκτορες. Εξετάστε ένα ρολόι. Μπορεί να θεωρηθεί ως ένα άψυχο αντικείμενο, ή μπορεί να θεωρηθεί ως ένας απλός πράκτορας. Σαν πράκτορα, τα περισσότερα ρολόγια κάνουν πάντα τη σωστή δράση: κίνηση των χεριών τους (ή επίδειξη των ψηφίων) ανάλογα τη μόδα. Τα ρολόγια είναι ένα είδος εκφυλισμένου πράκτορα δεδομένου ότι η ακολουθία percept τους είναι κενή οτιδήποτε και να συμβαίνει εξωτερικά, η δράση του ρολογιού δεν πρέπει να επηρεάζεται.

Το συγκεκριμένο δεν είναι απόλυτα αληθές. Εάν το ρολόι και ο ιδιοκτήτης ταξιδέψουν από Καλιφόρνια στην Αυστραλία, η σωστή κίνηση για το συγκεκριμένο ρολόι θα ήταν να γυρίσει έξι ώρες πίσω. Δεν είμαστε δυσαρεστημένοι επειδή τα ρολόγια μας αρνούνται να κάνουν τη συγκεκριμένη κίνηση επειδή αντιλαμβανόμαστε ότι δρουν λογικά έχοντας κάποιες φορές έλλειψη εξοπλισμού.

2.6 - Ευφυείς Πράκτορες

Στο μέλλον ο μόνος τρόπος για να φάξει κανείς το Internet, θα είναι μέσω έξυπνων μηχανών, γιατί άσχετα με το πόσο καλύτερα οργανωμένο θα είναι, δεν θα μπορεί να συμβαδίσει με την αύξηση των πληροφοριών [30]. Η συγκεκριμένη προφητική δήλωση όπως αποδείχθηκε αναφέρεται φυσικά στην τεχνολογία των ευφυών πρακτόρων. Με το πέρασμα του χρόνου και συνεχή αύξηση των πληροφοριών το Internet και το World Wide Web συνεχίζουν να μεγαλώνουν με πρωτοφανή ρυθμό. Το Web έχει εκατομμύρια θαυμαστές και εξελίσσεται σε ένα μοντέλο για γρήγορη και άμεση πρόσβαση πληροφοριών. Η πρωτοφανής ανάπτυξη των πηγών πληροφοριών στο Web προκάλεσε σε πολλούς χρήστες του υπερφόρτωση πληροφοριών.

Εξαιτίας της αύξησης των υπολογιστικών συστημάτων δημιουργήθηκε ο όρος των ευφυιών πρακτόρων. Ένας ευφυής πράκτορας αποτελεί ένα λογισμικό το οποίο ενεργεί ανθρώπους και ενέργειες για λογαριασμό τους. Οι ευφυείς πράκτορες δουλεύουν επιτρέποντας σε άτομα να αντιπροσωπεύουν δουλειά η οποία πρέπει να έχει γίνει, στο λογισμικό των πρακτόρων. Οι χρήστες βλέπουν τη δυνατότητα των πρακτόρων να εκτελέσουν πολλές εργασίες, από απλές και τετριμμένες μέχρι και σύνθετες. Οι ευφυείς πράκτορες αποτελούν ένα νέο πρότυπο για την ανάπτυξη εφαρμογών λογισμικού. Περισσότερο και από αυτό, οι πράκτορες οι οποίοι βασίζονται σε υπολογιστές έχουν

χαρακτηριστεί ως η επόμενη πιο σημαντική ανακάλυψη λογισμικού » σύμφωνα με τον Sargent το 1992, καθώς και ως την νέα επανάσταση στο λογισμικό σύμφωνα με τον Onum το 1994 [56]. Επί του παρόντος, οι πράκτορες αποτελούν το επίκεντρο αλλά και το κέντρο του ενδιαφέροντος για πολλούς τομείς της επιστήμης των υπολογιστών και της τεχνητής νοημοσύνης. Οι πράκτορες χρησιμοποιούνται σε ένα μεγάλο εύρος εφαρμογών, που μπορούν να είναι μικρές εφαρμογές όπως το ηλεκτρονικό ταχυδρομείο, αλλά και σε μεγάλες εφαρμογές όπως ο έλεγχος της εναέριας κυκλοφορίας.

Εκ πρώτης όψεως, βλέπουμε πως τα δύο αυτά συστήματα έχουν μεγάλες διαφορές μεταξύ τους. Παρόλα αυτά δεν είναι αυτό το θέμα: και στα δύο συστήματα, το κλειδί που χρησιμοποιείται για την λειτουργία τους είναι ότι η χρήση ενός πράκτορα.

Όπως είπαμε ένας πράκτορας είναι ένα σύστημα ηλεκτρονικού υπολογιστή που βρίσκεται σε ένα συγκεκριμένο περιβάλλον, και αυτό είναι ικανό να δρα αυτόνομα στο περιβάλλον, προκειμένου να πετυχαίνει τους στόχους του. Η αυτονομία είναι μια δύσκολη έννοια για να καθοριστεί με ακρίβεια, εμείς απλά εννοούμε ότι το σύστημα θα πρέπει να είναι σε θέση να δρα χωρίς την άμεση παρέμβαση των ανθρώπων (ή άλλων πρακτόρων), και θα πρέπει να έχει τον έλεγχο των δικών του κινήσεων αλλά και της εσωτερικής του κατάστασης. Θα μπορούσε να είναι χρήσιμο για να μπορέσουμε να αποδείξουμε την ύπαρξη αναλογίας μεταξύ της έννοιας αυτονομίας όσον αφορά στους πράκτορες και της αποθήκευσης σε σχέση με τα αντικειμενοστραφή συστήματα. Σ' ένα αντικείμενο μπορούν να ενσωματωθούν συγκεκριμένες κινήσεις, καθώς έχει μερικό έλεγχο της κατάστασης του, εννοώντας ότι μπορεί κανείς να έχει πρόσβαση ή να μπορεί να κάνει τροποποιήσεις με την βοήθεια των μεθόδων που παρέχει το συγκεκριμένο αντικείμενο.

Οι πράκτορες μπορούν να αποθηκεύουν μια συγκεκριμένη κατάσταση μ' ένα συγκεκριμένο τρόπο, αλλά ταυτόχρονα να αποθηκεύουν και μια συγκεκριμένη συμπεριφορά. Από την άλλη μεριά ένα αντικείμενο δεν μπορεί να ενσωματώνει συγκεκριμένη συμπεριφορά.: ο λόγος είναι ότι δεν έχει τον έλεγχο κατά την εκτέλεση συγκεκριμένων μεθόδων, παραδείγματος χάρη εάν ένα αντικείμενο x επικαλείται μια μέθοδο μ σε ένα αντικείμενο y , τότε το αντικείμενο y δεν μπορεί να ελέγξει αν το μ εκτελέστηκε ή όχι. Υπό την έννοια αυτή, το y δεν είναι αυτόνομο, καθώς δεν έχει τον έλεγχο των πράξεων του.

Αντίθετα, πιστεύουμε ότι έναν πράκτορα έχει αυτονομία καθώς έχει τον έλεγχο όλων των ενεργειών που εκτελεί. Εξαιτίας αυτής της διάκρισης έχουμε την τάση να σκεφτούμε τους πράκτορες επικαλούμενοι ενέργειες που μπορούν να γίνουν. Η απόφαση για το αν θα πραγματοποιηθεί η συγκεκριμένη ενέργεια ή όχι εξαρτάται καθαρά από τον παραλήπτη.

Φυσικά τα αυτόματα συστήματα πληροφορικής δεν αποτελούν κάτι καινούργιο. Υπάρχουν παραδείγματα τέτοιων συστημάτων που λειτουργούν είδη [58]:

- Οποιαδήποτε διαδικασία ελέγχου συστήματος, η οποία παρακολουθεί ένα πραγματικό περιβάλλον και πρέπει να δρα σύμφωνα με τις αλλαγές του περιβάλλοντος και πάντα σε πραγματικό χρόνο. Τέτοια συστήματα μπορεί να είναι πολύ απλά όπως ένας θερμοστάτης έως και πολύ περίπλοκα, όπως πυρηνικά συστήματα ελέγχου.
- Προγράμματα λογισμικού, τα οποία παρακολουθούν ένα συγκεκριμένο λογισμικό και εκτελούν διάφορες πράξεις για την αλλαγή του περιβάλλοντος ανάλογα με την αλλαγή των συνθηκών. Τέτοια παραδείγματα θεωρούμε τα προγράμματα Unix και xbuff τα οποία παρακολουθούν τα mail του χρήστη και του εφιστούν την προσοχή τοποθετώντας ένα εικονίδιο με την ταμπέλα new εφόσον έχει έρθει κάτι καινούργιο.

Είναι παράξενο ότι χρησιμοποιούμε τέτοια συστήματα πρακτόρων αλλά αυτά δεν είναι ευφυείς πράκτορες. Ευφυείς πράκτορες είναι ένα υπολογιστικό σύστημα το οποίο έχει την δυνατότητα να δρα ευέλικτα και αυτόνομα έτσι ώστε να πετυχαίνει του σχεδιαστικού του στόχους. Με τον όρο ευέλικτα εννοούμε: [58]

- Αντιδραστικότητα (reactivity): η ικανότητα των ευφυών πρακτόρων να αντιλαμβάνονται το περιβάλλον και να ανταποκρίνονται με έγκαιρες αλλαγές, για να μπορούν να αντεπεξέλθουν στις απαιτήσεις της αρχικής σχεδίασης.
- Προνοητικότητα (pro-activeness): οι ευφείς πράκτορες είναι ικανοί να επιδεικνύουν μια συγκεκριμένη συμπεριφορά ώστε να μπορούν να εκπληρώνουν τους σκοπούς τους σύμφωνα με δικές τους πρωτοβουλίες.
- Κοινωνική Συμπεριφορά (social ability): οι ευφείς πράκτορες είναι ικανοί να διεπικοινωνούν με άλλους πράκτορες (και πιθανόν ανθρώπους), προκειμένου να εκπληρώσουν στους στόχους τους.

Άλλοι ερευνητές τονίζουν άλλες πτυχές των πρακτόρων όπως κινητικότητα η προσαρμοστικότητα. Εννοείται βέβαια πως κάποιοι ευφείς πράκτορες έχουν επιπρόσθετα χαρακτηριστικά, συγκεκριμένες εφαρμογές αλλά και ιδιότητες η οποίες μπορούν να είναι πιο σημαντικές από άλλες. Παρόλα αυτά πιστεύουμε πως αυτή είναι η παρουσία των τεσσάρων ιδιοτήτων σε μια μοναδική λογισμική οντότητα, η οποία παρέχει τη δύναμη ενός παραδείγματος πρακτόρων το οποίο μπορεί να ξεχωρίσει συστήματα πρακτόρων από απλά συστήματα λογισμικού όπως αντικειμενοστραφή συστήματα παραδείγματος χάρη.

Όταν αναφερόμαστε σ' ένα σύστημα το οποίο βασίζεται σε ευφείς πράκτορες, εννοούμε το σύστημα που έχει ως βασικό του κλειδί τον πράκτορα. Κατ' αρχήν τέτοια συστήματα λειτουργούν με τους όρους του πράκτορα, και εννοείτε μπαίνουν σε εφαρμογή χωρίς συστήματα λογισμικού τα οποία να σχετίζονται με πράκτορες. Κάνοντας έναν παραλληλισμό με αντικειμενοστραφή αντικείμενα, βλέπουμε ότι είναι πολύ πιθανό και λογικό να δημιουργήσουμε σύστημα σύμφωνα με τους όρους του αντικειμένου, χωρίς όμως τη χρήση ενός αντικειμενοστραφούς λογισμικού. Αυτό όμως θα ήταν ασυνήθιστο αλλά και αντιπαραγωγικό. Το πιο λογικό είναι να δημιουργηθεί ένα σύστημα το οποίο να σχεδιαστεί και να λειτουργεί σύμφωνα με τους όρους του πράκτορα.

Ακόμα, καλό είναι να σημειώσουμε ότι ένα σύστημα το οποίο βασίζεται σε πράκτορες μπορεί να περιέχει ένα μη μηδενικό αριθμό πρακτόρων. Η περίπτωση πολλαπλών πρακτόρων, δηλαδή ένα σύστημα το οποίο κατασκευάζεται και τίθεται σε εφαρμογή σαν αποτέλεσμα αλληλεπιδράσεις πρακτόρων, και σίγουρα είναι πιο πολύπλοκο από την περίπτωση του ενός πράκτορα. Ένα παράδειγμα το οποίο θα δούμε και παρακάτω είναι τα συστήματα γνωστά και ως εμπειρογνώμονες, όπου ο πράκτορας λειτουργεί σαν βοηθός του χρήστη οποίος χρησιμοποιεί τον υπολογιστή για συγκεκριμένες εργασίες.

2.7 - Αλγόριθμοι Ευφούς πρακτόρων

Μέθοδοι με βάση την μνήμη

Οι μέθοδοι με βάση την μνήμη επιδρούν στο σύνολο του πίνακα χρηστών αντικειμένων, για την παραγωγή των προβλέψεων τους, ενώ η πλειοψηφία τους ανήκει στην κατηγορία των συστημάτων Συνεργατικής διήθησης βάσει Χρηστών. Τα συγκεκριμένα συστήματα βασίζονται κυρίως σε υπολογισμούς εγγύτητας μεταξύ χρηστών, με τελικό στόχο την κατασκευή των λεγόμενων γειτονιών χρηστών. [58]

Τα κεντρικά βήματα της συγκεκριμένης διαδικασίας είναι:

Αναπαράσταση: Η αρχική αναπαράσταση ενός συστήματος όπου χρησιμοποιεί ευφείς πράκτορες, τα δεδομένα εισόδου λαμβάνονται ως μία συλλογή αριθμητικών στοιχείων από m χρήστες προς n αντικείμενα, η οποία οργανώνεται για να πάρει μορφή ενός πίνακα χρηστών αντικειμένων διαστάσεων $m \times n$. Ο συγκεκριμένος πίνακας είναι συνήθως είναι «αραιός» και όπως μπορεί να δημιουργηθούν προβλήματα σποραδικότητας. Για να αντιμετωπιστούν τα συγκεκριμένα προβλήματα υπάρχουν συγκεκριμένες τεχνικές όπως:

- Εξορισμού ψήφιση
- Επεξεργασία με μέσους όρους Χρηστών Αντικειμένων και συνδυαστικές μεθόδους
- Χρήση Filterbots
- Χρήση Τεχνικών Μείωσης Διαστάσεων

Διαμόρφωση γειτονιάς: Η συγκεκριμένη μέθοδος ξεκινάει με τον υπολογισμό της εγγύτητας μεταξύ όλων των χρηστών από τον πίνακα χρηστών αντικειμένων. Απομονώνουμε τους χρήστες που σύμφωνα με τα αποτελέσματα των υπολογισμών εγγύτητας εμφανίζονται να είναι συναφείς με τον ενεργό χρήστη, και με την βοήθεια τους, κατασκευάζουμε την γειτονιά ενεργού χρήστη. Η συγκεκριμένη γειτονιά θα χρησιμοποιηθεί στο επόμενο βήμα της αλγοριθμικής διαδικασίας, για την παραγωγή τελικών προβλέψεων. Τα βήματα της συγκεκριμένης διαδικασίας είναι τα εξής:

- Εγγύτητα Χρηστών
- Παραγωγή γειτονιάς

Παραγωγή συστάσεων η προβλέψεων: Στη συγκεκριμένη διαδικασία αντιστοιχεί είτε η παραγωγή μιας πρόβλεψης είτε μια σύσταση. Και στις δύο περιπτώσεις το αποτέλεσμα βασίζεται στη γειτονιά των χρηστών, η οποία έχει ήδη κατασκευαστεί.

Κάποιες εναλλακτικές της συγκεκριμένης μεθόδου είναι:

- Διήθηση Βάσει περιεχομένου
- Συστήματα Συστάσεων που χρησιμοποιούν Υβριδικές Μεθόδους
- Συστήματα Συστάσεων που χρησιμοποιούν SVD
- Συστήματα Συστάσεων Βάσει μοντέλων

Κεφάλαιο 3 – Προβλήματα Πρακτόρων

3.1 - Προβλήματα τα οποία πρέπει να λύσουν οι πράκτορες

Τα προβλήματα που καλούνται να επιλύσουν οι πράκτορες είναι άπειρα και όπως έχουμε αναφέρει καλύπτουν σχεδόν όλο το φάσμα των επιστημών και δραστηριοτήτων του ανθρώπου. Αρχικά θα αναλύσουμε απλά πρακτικά προβλήματα τα οποία καλούνται να επιλύσουν οι ευφυείς πράκτορες. Η λίστα από τα συγκεκριμένα προβλήματα είναι απλά ενδεικτική. Ένα απλό πρόβλημα είναι εξεύρεση μιας σωστής πληροφορία και η διαδικασία να την βρούμε με όσο το δυνατό πιο γρήγορα αλλά και αποτελεσματικό τρόπο. Αυτό μπορεί να συμβεί αν ψάχνουμε σε εταιρικά δεδομένα ή στο internet. Ένας ευφυής πράκτορας μπορεί να βοηθήσει βρίσκοντας την πληροφορία και φιλτράροντας τη μέσα από σειρά δεδομένων που ταιριάζουν με το ερώτημα του χρήστη.

Ένα άλλο πρόβλημα είναι να εντυφύσουμε και να διεξάγουμε συμπεράσματα από τις συγκεκριμένες πληροφορίες. Το πρόβλημα αυτό είναι ένα από τα πιο δύσκολα που έχουν να λύσουν οι πράκτορες και γίνεται μεγαλύτερο όταν ο όγκος πληροφοριών μεγαλώνει. Οι πράκτορες μπορούν να λύσουν το συγκεκριμένο πρόβλημα προσαρμόζοντας τις πληροφορίες σύμφωνα με τις προτιμήσεις του χρήστη, να διακρίνουν αρχικά τις πληροφορίες χωρίς πρόσθετα κριτήρια και έτσι να επιλύσουν το πρόβλημα σε ένα μικρότερο σύνολο στοιχείων. Έτσι εξοικονομείτε χρόνος και γίνεται πιο αποτελεσματική η απόδοση του αλγόριθμου λύσης του προβλήματος.

Άλλο ένα πρόβλημα είναι πως από πολλά γεγονότα που συμβαίνουν σε ένα περιβάλλον και χρειάζονται δράση κανείς τελικά δεν λαμβάνει δράση. Για παράδειγμα σε μια έκτακτη ανάγκη να βγει η κατάλληλη ενημέρωση ή ανακοίνωση προς το ανάλογο κοινό, ή σε ένα ανταγωνιστικό περιβάλλον να εντοπιστεί και να ανακοινωθεί μια ανταγωνιστική ενημέρωση, ή ακόμα και δημιουργία ενός νέου εγγράφου του υπολογιστή μας για την εξυπηρέτηση μιας άμεσης εργασίας. Ο πράκτορας μπορεί να βοηθήσει αυτοματοποιώντας διαδικασίες που ο άνθρωπος μπορεί να ξεχάσει σχετικά εύκολα ή να μην προσέξει. Όλα αυτά τα προβλήματα μπορούν να λυθούν με διαφορετικούς τρόπους. Ο πράκτορας έρχεται να λύσει το πρόβλημα της προσοχής ενός ατόμου ή εργαζομένου που είτε ο φόρτος εργασίας τον κάνει αδύνατο να λύσει ένα πρόβλημα είτε η ανθρώπινη περιορισμένη φύση τον κάνει αδύναμο να προσέξει κάποια σημαντικά στοιχεία ή γεγονότα. Η επιχειρηματική αξία της χρήσης των ευφυών πρακτόρων είναι ότι παίρνουμε την πληροφορία που χρειαζόμαστε την σωστή στιγμή αποφεύγοντας αδυναμίες των μέχρι τώρα παραδοσιακών τεχνικών.

3.2 - Προβλήματα κατά τη χρήση των πρακτόρων

Κάθε αλγόριθμος ο οποίος προτείνεται για εφαρμογή από ένα συγκεκριμένο σύστημα όπου χρησιμοποιεί ευφυείς πράκτορες, οφείλει να ενσωματώνει λύσεις για την αντιμετώπιση κάποιων από τα προβλήματα που ακολουθούν. [53]

- **Σποραδικότητα:** Δηλαδή η δημιουργία αραιών πινάκων χρηστών-αντικειμένων, και σαν συνέπεια αυτού η μη εύρεση χαρακτηριστικών γειτόνων και τελικά παραγωγή αποτυχημένων συστάσεων
- **Επεκτασιμότητα:** Τα συγκεκριμένα συστήματα απαιτούν υπολογισμούς ,το πλήθος των οποίων αυξάνεται με τον αριθμό των χρηστών. Δηλαδή ένα αλγόριθμος που είναι πιο αποδοτικός όταν εφαρμόζεται σε περιορισμένο αριθμό δεδομένων εισόδου μπορεί να αποδειχθεί ανεπαρκής με την αύξηση των δεδομένων. Εδώ βοηθάνε η αλγόριθμοι διήθησης η οποίοι έχουν τη δυνατότητα να επεκτείνονται.
- **Συνωνυμία:** Τα συγκεκριμένα συστήματα δυσκολεύονται στον εντοπισμό λανθασμένων σχέσεων μεταξύ αντικειμένων τα οποία μπορεί να έχουν διαφορετική επωνυμία αλλά να αναφέρονται σε κοινά η παρόμοια προϊόντα. Αυτό αφορά κυρίως τη χρήση των συγκεκριμένων συστημάτων στο ηλεκτρονικό επιχειρών.

3.3 - Λύνοντας νέους τύπους προβλημάτων

Ορισμένοι τύποι λογισμικού είναι από την αρχή δύσκολο να σχεδιαστούν αλλά και να εφαρμοστούν. Τέτοια συστήματα είναι τα λειτουργικά συστήματα. Τα λειτουργικά συστήματα λαμβάνουν μια συγκεκριμένη είσοδο, και υπολογίζουν την λειτουργία τους και δίνουν την συγκεκριμένη παραγωγή σαν αποτέλεσμα. Οι μεταφραστές επίσης αποτελούν παράδειγμα τέτοιων συστημάτων. Επίσης έχουμε τα λεγόμενα αντιδραστικά συστήματα, τα οποία διατηρούν επαφή με το περιβάλλον και τα οποία είναι δύσκολο να σχεδιαστούν αλλά εύκολο να εφαρμοστούν. Η διαδικασία ελέγχου συστημάτων, τα λειτουργικά συστήματα υπολογιστών και τα συστήματα διαχείρισης δικτύου και αλλά παραδείγματα αντιδραστικών συστημάτων. Όλα τα προηγούμενα παραδείγματα τα συστήματα ηλεκτρονικού υπολογιστή λειτουργούν ανεξάρτητα για μεγάλο χρονικό διάστημα και έχουν θεωρηθεί ως τα πιο πολύπλοκα στην σχεδίαση και στην εφαρμογή. Έχει γίνει μεγάλη προσπάθεια για την ανάπτυξη λογισμικού και γλωσσών προγραμματισμού για να καταπολεμηθεί η συγκεκριμένη πολυπλοκότητα με περιορισμένη επιτυχία.

Τα συστήματα αυτά μπορούμε να τα χωρίσουμε σε τρεις κατηγορίες [41]:

- Ανοιχτά συστήματα
- Πολύπλοκα συστήματα
- Υπολογιστικά Συστήματα

3.4 Ανοιχτά συστήματα

Ανοιχτό σύστημα θεωρείται το σύστημα του οποίου η κατασκευή είναι ικανή για δυναμικές αλλαγές. Τα χαρακτηριστικά του συγκεκριμένου συστήματος είναι ότι τα εξαρτήματα του είναι γνωστά από την αρχή, μπορούν να αλλάζουν κατά την πάροδο του χρόνου, και μπορεί να είναι άκρως ετερογενή (με την έννοια ότι μπορούν να εφαρμόζονται από διαφορετικούς ανθρώπους σε διαφορετικές χρονικές στιγμές, με διαφορετικά εργαλεία λογισμικού και διαφορετικές τεχνικές).

Πλέον οι εφαρμογές στον τομέα της πληροφορικής απαιτούν όλο και πιο πολύ από τους χρήστες να λειτουργούν σε τέτοια πεδία. Ένα από τα πιο γνωστά παραδείγματα ανοιχτού περιβάλλοντος αποτελεί το internet, πρόκειται για ένα συνδεδεμένο δίκτυο υπολογιστών το οποίο μεγαλώνει συνεχώς και γίνεται όλο και πιο πολύπλοκο. Ο σχεδιασμός και η κατασκευή των εργαλείων λογισμικού τα οποία αξιοποιούν τις τεράστιες δυνατότητες του internet αλλά και οι συναφείς τεχνολογίες είναι από τις μεγαλύτερες προκλήσεις που αντιμετώπισαν οι επιστήμονες την δεκαετία του 90, και γι' αυτό το λόγο μπορεί να χρησιμοποιηθεί σαν μελέτη περίπτωσης.

Το internet μπορεί να θεωρηθεί σαν μια πηγή διανομής πληροφοριών, με κόμβους σχεδιασμένους στο διαδίκτυο, και εφαρμόζονται από διάφορους οργανισμούς και ατομικά με διαφορετικές ατζέντες. Κάθε υπολογιστικό σύστημα το οποίο λειτουργεί μέσω διαδικτύου πρέπει να μάθει να λειτουργεί σύμφωνα με αυτά χωρίς την συνεχή καθοδήγηση από τους χρήστες(μέσα σε καθορισμένα όρια πάντα). Αυτή η λειτουργία χρειάζεται τεχνικές που έχουν σχέση με συνεργασία και διαπραγμάτευση, τα οποία βρίσκονται πάρα πολύ στον τομέα των πολυπρακτόρων.

3.5 Πολύπλοκα συστήματα

Τα πιο ισχυρά εργαλεία για να αντιμετωπίσεις ένα πολύπλοκο σύστημα είναι ο αφαιρετισμός αλλά και η μεθοδικότητα. Οι πράκτορες αποτελούν ένα ισχυρό όπλο για να μεθοδεύσουν ένα πολύπλοκο σύστημα. Αν υπάρχει πρόβλημα σε έναν τομέα το οποίο είναι πολύπλοκο, η μεγάλο, η απρόβλεπτο τότε η μονή λύση για να αντιμετωπισθεί είναι να αναπτυχθεί μια σειρά από σπονδυλωτά χαρακτηριστικά τα οποία είναι εξειδικευμένα για να λύνουν τέτοια προβλήματα.

Σε τέτοιες περιπτώσεις όταν προκύπτουν αλληλοεξαρτώμενα προβλήματα, οι πράκτορες των συστημάτων πρέπει να συνεργάζονται έτσι ώστε να εξασφαλίσουν πώς οι αλληλεξαρτήσεις διαχειρίστηκαν σωστά. Σε τέτοιες περιπτώσεις μια προσέγγιση βασισμένη σε πράκτορες σημαίνει ότι

το συνολικό πρόβλημα μπορεί να σπάσει σε μικρότερα τμήματα και σε απλούστερα συστατικά, κάτι το οποίο σημαίνει πώς το πρόβλημα είναι πιο εύκολο να αναπτυχθεί αλλά και να διατηρηθεί, και τα οποία είναι εξειδικευμένα για την λύση τέτοιων προβλημάτων. Αυτή η διάσπαση του προβλήματος επιτρέπει στο κάθε πράκτορα να χρησιμοποιεί τα πλέον κατάλληλα παραδείγματα για την επίλυση του συγκεκριμένου προβλήματος, και όχι να υιοθετηθεί μια κοινή λογική η οποία θα οδηγήσει σε ένα συμβιβασμό για το σύνολο του συστήματος, αλλά θα αποτελέσει την βέλτιστη λύση για κάθε τμήμα του προβλήματος. Η έννοια του αυτόνομου πράκτορα περιέχει τη χρήσιμη έννοια του αφαιρετισμού με τον ίδιο τρόπο όπως οι διαδικασίες, οι τύποι δεδομένων, και τα αντικείμενα που λειτουργούν με την ίδια έννοια. Επιτρέπει στην ανάπτυξη του λογισμικού να αντιλαμβάνεται το πολύπλοκο λογισμικό ως μια κοινωνία συνεργαζόμενων και αυτόνομων λυτών προβλημάτων. Για πολλές εφαρμογές οι οποίες λειτουργούν σε υψηλό επίπεδο η συγκεκριμένη εφαρμογή προτιμάται.

3.6 –Συστήματα Υπολογιστών

Παρά τις καινοτομίες που έγιναν στο σχεδιασμό της επαφής μεταξύ ανθρώπου-υπολογιστή τις δυο προηγούμενες δεκαετίες, καθώς και η ευρεία διαθεσιμότητα των διεπαφών χρήστη η οποίες βασίζονται σε windows, οι χρήστες οι οποίοι είναι αφελής στη χρήση υπολογιστών συνεχίζουν να βρίσκουν τη χρήση συγκεκριμένων λογισμικών δύσκολη. Ένα λόγος για αυτό είναι ότι ο χρήστης ενός παραγόμενου λογισμικού πρέπει τυπικά να περιγράψει με λεπτομέρεια κάθε βήμα που πρέπει να εκτελεστεί για να λυθεί το πρόβλημα.

Εάν η δύναμη των σημερινών εφαρμογών λογισμικού πρόκειται να χρησιμοποιηθεί από του συγκεκριμένους χρήστες, τότε πρέπει να ξανασκεφτούμε για την αλληλεπίδραση χρηστών και υπολογιστών. Πρέπει να υπάρχει μια ισότιμη σχέση, το μηχάνημα δεν πρέπει να λειτουργεί σαν ένας απλός υποδοχέας εργασιών αλλά πρέπει να συνεργάζεται με τον χρήστη για την επίτευξη των στόχων τους. Για να υπάρχει αυτή η λειτουργικότητα οι εφαρμογές λογισμικού θα πρέπει να είναι:

- **Αυτόνομες:** Πρέπει να καθοριστεί ποιος είναι ο καλύτερος τρόπος για να λυθεί το πρόβλημα και να λυθεί χωρίς καμία καθοδήγηση από το χρήστη.
- **Διορατικές:** Δεν θα πρέπει να τους πουν τι να κάνουν μετά, αλλά πρέπει να κάνουν αυτές προτάσεις στο χρήστη.
- **Να ανταποκρίνονται:** Θα πρέπει να έχουν υπόψη τους πιθανές αλλαγές στα ζητούμενα του χρήστη αλλά και στο περιβάλλον.
- **Προσαρμοστικές:** Θα πρέπει να γνωρίζει τις προτιμήσεις του χρήστη και να προσαρμόζεται σε αυτές.

Με αλλά λόγια, πρέπει να λειτουργεί σαν ευφυείς πράκτορας. Αυτές οι σκέψεις έχουν οδηγήσει στην ιδέα ενός πράκτορα που λειτουργεί ως βοηθός εμπειρογνώμονα σε σχέση με κάποιες εφαρμογές, γνωστές τόσο για τις ίδιες της εφαρμογές αλλά και για τον χρήστη, και να είναι σε θέση να λειτουργούν μαζί με το χρήστη για την επίτευξη των στόχων τους.

3.7 - Βελτίωση της αποτελεσματικότητας στην ανάπτυξη λογισμικού

Οι τεχνολογία των πρακτόρων μας δίνει τα εργαλεία με τα οποία μπορούμε να δημιουργήσουμε εφαρμογές που παλιότερα δεν μπορούσαμε. Αλλά μπορεί επίσης να παρέχει μια καλύτερη θεώρηση της συγκεκριμένης εφαρμογής. Εδώ αναφέρονται τρεις σπουδαίοι τομείς χαρακτηριστικών που συχνά αναφέρονται ως λογικοί για την υιοθέτηση της τεχνολογίας πρακτόρων [41]:

- Τα δεδομένα, τον έλεγχο, την τεχνογνωσία, οι πηγές που διανέμονται εγγενώς

- Το σύστημα φυσικά θεωρείται σαν μια κοινωνία αυτόνομων συνεργαζόμενων στοιχείων.
- Το σύστημα περιέχει κληρονομικά στοιχεία, το οποίο έχουν φτιαχτεί ώστε να αλληλεπιδρούν με άλλα, πιθανώς νέα αντικείμενα λογισμικού.

3.8 - Κατανομή πόρων, ελέγχου τεχνογνωσίας, και πηγών

Ένα ολοκληρωμένο σύστημα επίλυσης ενός προβλήματος περιέχει ένα σύνολο αυτοτελών οντοτήτων για την λύση προβλημάτων, οι οποίες διανέμονται φυσικά ή λογικά (όσον αφορά δεδομένα, έλεγχο κ.λ.π), και οι οποίες θέλουν να αλληλεπιδρούν μεταξύ τους έτσι ώστε να μπορούν να λύνουν προβλήματα. Οι πράκτορες μπορούν να παρέχουν μια αποτελεσματική λύση για την καλύτερη επικοινωνία και έλεγχο μεταξύ των οντοτήτων αυτών. Για παράδειγμα για ιατρική περίθαλψη πρέπει να συνεργαστούν όλοι (γιατροί, νοσοκόμες κ.λ.π) έτσι ώστε να παρέχουμε την κατάλληλη φροντίδα στον ασθενή. Σ' αυτόν τον τομέα υπάρχει [43]:

- Διανομή δεδομένων: Για παράδειγμα ο γιατρός έχει πληροφορίες για τον ασθενή διαφορετικές από την νοσοκόμα και ας πρόκειται και για το ίδιο άτομο.
- Διανομή ελέγχου: Κάθε άτομο είναι υπεύθυνο για ένα σύνολο εργασιών.
- Διανομή τεχνογνωσίας: Η γνώση των ειδικών γιατρών πχ είναι πολύ διαφορετική από αυτή της νοσοκόμας ή ενός απλού γιατρού.
- Διανομή πόρων: Ο ειδικός είναι υπεύθυνος για τα κρεβάτια που απαιτούνται για τους ασθενείς του, ενώ αυτός που κάνει πρακτική πχ και δεν είναι ειδικός για να πληρώνει τους λογαριασμούς του νοσοκομείου κ.λ.π.

Σε περιπτώσεις όπως αυτή, οι πράκτορες παρέχουν μια φυσική μορφή για την μοντελοποίηση του προβλήματος: οι οντότητες του πραγματικού κόσμου και οι αλληλεπιδράσεις τους μπορούν να θεωρηθούν ως αυτόνομοι πράκτορες για τη λύση προβλημάτων με τις δικές τους πηγές και τη δική τους τεχνογνωσία, και οι οποίοι είναι σε θέση να αλληλεπιδρούν μεταξύ τους για μπορούν να εκτελούν κάθε διεργασία.

Επίσης στην περίπτωση της διανομής πηγών δεδομένων, η χρήση πρακτόρων σημαίνει την ότι μπορεί να πραγματοποιηθεί μεγάλη ποσότητα διαδικασιών στην πηγή δεδομένων, με μόνο υψηλού επιπέδου πληροφορίες να ανταλλάσσονται. Αυτό μειώνει την πιθανότητα να σταλούν μεγάλες ποσότητες μη επεξεργασμένων δεδομένων σ' έναν μακρινό κεντρικό επεξεργαστή, καθιστώντας πιο αποτελεσματική την χρήση επικοινωνιών εύρος ζώνης.

3.9 - Φυσική μεταφορά

Η έννοια του αυτόνομου πράκτορα είναι συχνά η πιο κατάλληλη μεταφορά για την παρουσίαση μιας δεδομένης λειτουργικότητας λογισμικού. Για παράδειγμα:

- Ένα πρόγραμμα που φιλτράρει το ηλεκτρονικό ταχυδρομείο μπορεί να παρουσιαστεί στον κάθε χρήστη μέσω της μεταφοράς ενός ψηφιακού προσωπικού βοηθού.
- Η σύσκεψη για την παρουσίαση ενός λογισμικού μπορεί να παρουσιαστεί φυσικά σαν μια εξουσία αυτόνομους, κοινωνικούς πράκτορες που μπορούν να αλληλεπιδρούν με άλλα στοιχεία σχετικά με τον λογαριασμό του χρήστη.

Στις συγκεκριμένες λειτουργίες, το γεγονός ότι αυτές οι λειτουργίες υλοποιούνται μέσω μιας σειράς τοπικών πρακτόρων, το οποίο σημαίνει ότι μπορούν να εξατομικευτούν για να αντικατοπτριστούν οι

προτιμήσεις του χρήστη. Τέλος, σε ηλεκτρονικά παιχνίδια και σε συστήματα εικονικής πραγματικότητας οι χαρακτήρες μπορούν φυσικά να αναπαριστούνται από πράκτορες.

3.10 - Κληρονομικά συστήματα

Αρκετοί οργανισμοί έχουν αρκετές εφαρμογές λογισμικού οι οποίες εκτελούν σημαντικές οργανωτικές λειτουργίες. Για να μπορεί να συμβαδίσει με τις συνεχόμενες αλλαγές των επιχειρήσεων τα συστήματα αυτά θα πρέπει να ενημερώνονται περιοδικά. Παρόλα αυτά το να τροποποιήσεις τέτοια συστήματα είναι γενικά πολύ δύσκολο γιατί η δομή του συστήματος και η εσωτερική του λειτουργία είναι πολύ δύσκολο να αλλαχθεί, τα δεδομένα και αποτελέσματα αλλάζουν μορφή οπότε τα δομικά στοιχεία είσοδος και έξοδος διαφοροποιούνται με το πέρασμα του χρόνου, και οι ειδικοί στην επίγνωση του συγκεκριμένου λογισμικού αλλάζουν θέσεις και το πιο πιθανόν είναι να μην υπάρχουν στην ανάλογη θέση εργασίας.

Ξεγράφοντας το συγκεκριμένο σύστημα και αντικαθιστώντας τα ανάλογα λογισμικά το σύστημα γίνεται απαγορευτικό και ακριβό, και συχνά αδύνατο να βρεθούν νέα προσαρμοσμένα λογισμικά. Ως εκ τούτου σε μακροχρόνια βάση, ο μόνος τρόπος να διατηρηθούν τέτοια χρήσιμα συστήματα είναι να ενσωματωθούν σε μια ευρύτερη κοινωνία συνεργασίας, όπου μπορούν να αξιοποιηθούν από άλλα κομμάτια λογισμικού.

Τα λογισμικά αυτά όμως πρέπει να είναι έτσι σχεδιασμένα ώστε να μπορούν να ενσωματωθούν σε οποιοδήποτε σύστημα. Τα λογισμικά αυτά και γενικότερα τα συστήματα αυτά λέγονται κληρονομικά συστήματα γιατί σχεδιάζονται έτσι ώστε να μπορούν οι ιδιότητες τους και οι μέθοδοι τους να κληρονομηθούν ανάλογα.

3.11 - Οι περιορισμοί στις λύσεις πρακτόρων

Παρότι η τεχνολογία των πρακτόρων παίζει ένα πολύ σημαντικό ρόλο στην ανάπτυξη των εφαρμογών πληροφορικής πρέπει να χρησιμοποιείται με σύνεση και όπου αυτή είναι απολύτως απαραίτητη. Πολλές από τις σημερινές εφαρμογές που χρησιμοποιούν τεχνολογίες πρακτόρων θα μπορούσαν να κατασκευαστούν και χωρίς αυτές. Έτσι το απλό γεγονός ότι ένα πρόβλημα έχει διανεμημένες πηγές ή κληροδοτημένα συστήματα δεν συνεπάγεται ότι ένα σύστημα πρακτόρων είναι το πιο κατάλληλο ή έστω το πιο εφικτό.

Όπως με όλα τα συστήματα που σχεδιάζονται, η τελική επιλογή των μεθόδων και διαδικασιών που πρέπει να χρησιμοποιήσουμε εξαρτάται από πολλούς παράγοντες. Έτσι πρέπει να προσδιορίσει κανείς τους τύπους των καταστάσεων των οποίων οι λύσεις θα είναι βασισμένες στους πράκτορες, και ποιες από αυτές είναι απαραίτητο να αναπτυχθούν. Επιπλέον, θα πρέπει να σημειωθεί πως η επιλογή χρήσης πρακτόρων σε ένα πρόβλημα οδηγεί σε πολλές δυσκολίες και σε νέα προβλήματα συνήθως κοινά σε όλες τις εφαρμογές που βασίζονται σε πράκτορες. Τα πιο συχνά από αυτά είναι: [58]

- Μη συνολικός έλεγχος συστήματος. Μια λύση η οποία είναι βασισμένη σε πράκτορες δεν είναι κατάλληλη για τομείς όπου η παγκόσμιοι περιορισμοί πρέπει να διατηρηθούν και που η απάντηση σε πραγματικό χρόνο πρέπει να είναι εγγυημένη ή τα αδιέξοδα πρέπει να αποφεύγονται.
- Μη παγκόσμια προοπτική. Η ενέργεια ενός πράκτορα εξορισμού καθορίζεται από τη θέση του τοπικού πράκτορα. Ωστόσο σε κάθε ρεαλιστικό σύστημα πρακτόρων, δεν είναι πιθανό να ολοκληρωθεί η παγκόσμια γνώση, αυτό μπορεί να σημαίνει πως οι πράκτορες μπορούν να παίρνουν απλά βέλτιστες αποφάσεις. Το θέμα της λήψης αποφάσεων με βάση την τοπική

γνώση και την επιθυμία για επίτευξη της βέλτιστης απόδοσης αποτελεί ένα σημαντικό θέμα στην έρευνα πολλαπλών πρακτόρων.

- Εμπιστοσύνη και αντιπροσωπία. Για να αισθάνονται τα άτομα άνετα με την ιδέα της μεταβίβασης των καθηκόντων τους στους πράκτορες, θα πρέπει πρώτα να τους εμπιστευτούν. Και τα άτομα και οι οργανισμοί θα πρέπει να έχουν μεγαλύτερη αυτοπεποίθηση και να είναι συνηθισμένοι στην έννοια των αυτόνομων τμημάτων λογισμικού, εάν πρόκειται να χρησιμοποιηθούν ευρέως. Οι χρήστες πρέπει να κερδίσουν την εμπιστοσύνη των πρακτόρων που εργάζονται για αυτούς, και αυτή η διαδικασία θα πάρει καιρό. Κατά τη διάρκεια αυτής της περιόδου, οι πράκτορες πρέπει να πετύχουν ισορροπία μεταξύ συνεχούς καθοδήγησης και της μη καθοδήγησης. Παρόλα αυτά ένας πράκτορας πρέπει να γνωρίζει τα όρια του.

3.12 Προβλήματα στην ανάπτυξη του πράκτορα.

Κατά τη διάρκεια του υλοποίησης ενός συστήματος πρακτόρων βλέπουμε ότι η εμπειρία στο σχεδιασμό και στην εφαρμογή εφαρμογών για πράκτορες είναι ακόμα μικρή. Υπάρχει λίγη εμπειρία για την παραγωγή και υποστήριξη λογισμικού για εφαρμογές πρακτόρων, και ακόμα λιγότερη κατανόηση για θέματα που πρέπει να αντιμετωπιστούν κατά τη διάρκεια δημιουργίας του συστήματος. Ακόμα περισσότερο τα σημερινά προγράμματα που χρησιμοποιούν πράκτορες δημιουργούνται από το μηδέν, χρησιμοποιώντας κατά παραγγελία τεχνικές και εργαλεία που δεν μπορούν να εφαρμοστούν σε άλλα συστήματα.

Επίσης διαπιστώνεται ότι η περισσότερη από την υποδομή που χρειάζεται για την υλοποίηση ενός συστήματος δεν έχει πλήρως καθοριστεί ή υλοποιηθεί με έτοιμα εργαλεία ή εφαρμογές και έτσι οι προγραμματιστές δημιουργούν όλες τις καταστάσεις από την αρχή, κάτι που φυσικά δεν είναι παραγωγικό και βιώσιμο. Ωστόσο υπάρχει μια αυξανόμενη τάση στο ότι αυτό πρέπει να αλλάξει, εάν οι πράκτορες θέλουν να αποτελέσουν τον βασικό κορμό στην ανάπτυξη ενός λογισμικού [58].

Η υποστήριξη είναι απαραίτητη σε όλη τη διαδικασία ανάπτυξης συστήματος που στηρίζεται σε πράκτορες. Αν χρησιμοποιήσουμε ένα παραδοσιακό μοντέλο ανάπτυξης λογισμικού, ως βάση για να δομήσουμε τη ανάλυση μας τότε πάντα ξεκινάμε με την διερεύνηση των απαιτήσεων και προδιαγραφών του απαιτούμενου συστήματος. Δεδομένο ότι αυτό το στάδιο προηγείται, η ανάλυση των απαιτήσεων δεν διαφέρει ουσιαστικά από τους άλλους τύπους ανάπτυξης του λογισμικού. Η ανάλυση των απαιτήσεων συνήθως καταλήγει στις απαιτήσεις του χρήστη η οποίες απεικονίζονται σ' ένα έγγραφο, μια δήλωση για τις ιδιότητες όπου ο προγραμματιστής πιστεύει πως θέλει ο χρήστης για το νέο σύστημα. Το έγγραφο αυτό συνήθως γράφεται σε μη τεχνική γλώσσα, έτσι ώστε να μπορεί να γίνει κατανοητό από τον πελάτη και από τον προγραμματιστή.

3.13 Προδιαγραφές

Μόλις κατανοηθούν οι ανάγκες του πελάτη απαιτείται μια απάντηση για το τι σκοπεύουν να φτιάξουν οι προγραμματιστές. Το συγκεκριμένο έγγραφο με τις ανάγκες του πελάτη πρέπει να γραφτεί όσο πιο τυπικά γίνεται, και αυτό αποσκοπεί στο να είναι σαφής η απάντηση για τις ιδιότητες του συστήματος το οποίο θα φτιάξει ο προγραμματιστής.

Ένας αριθμός προσεγγίσεων έχουν αναπτυχθεί σύμφωνα με τις προδιαγραφές των συστημάτων που βασίζονται σε πράκτορες, και κυρίως σε επίσημες προσεγγίσεις. Οι λεπτομέρειες αυτών των τυποποιήσεων τείνουν να είναι πολύπλοκες, αλλά σε γενικές γραμμές, αλλά χαρακτηρίζουν τους πράκτορες ως ορθολογικούς λήπτες αποφάσεων. Με αυτό εννοούμε συστήματα των οποίων η εσωτερική κατάσταση μπορεί να εκφραστεί με όρους κατασκευών όπως η πεποίθηση, η επιθυμία και η πρόθεση.

Η λήψη αποφάσεων ενός πράκτορα μπορεί να χαρακτηριστεί με τους όρους της εν λόγω κατασκευής. Θα μπορούσε να είναι δύσκολο να οριστεί ένα σύστημα υπολογιστών με αυτό τον τρόπο, αλλά η ιδέα είναι πραγματικά πολύ απλή. Ένας δημιουργεί μια προδιαγραφή με την βοήθεια μιας σειράς κανόνων κάπως έτσι:

Εάν ο πρώτος πράκτορας πιστεύει πως ο δεύτερος πράκτορας πιστεύει πως ο πρώτος διάδρομος είναι ανοιχτός, τότε ο πρώτος πράκτορας πιστεύει πως ο δεύτερος έχει εσφαλμένες πληροφορίες, και ο πρώτος πράκτορας θα πληροφορήσει το δεύτερο για το λάθος του.

Κάποιος θα διερωτάται γιατί χρησιμοποιούνται οι όροι όπως το πιστεύω και η πρόθεση. Οι πεποιθήσεις συνήθως χρησιμοποιούνται για να αναφέρουν πληροφορίες τι οποίες έχουν οι πράκτορες για το περιβάλλον τους. Αυτές οι πληροφορίες μπορεί να είναι λανθασμένες κατά τον ίδιο τρόπο κατά τον οποίο η πληροφορία που έχουμε εμείς για το περιβάλλον είναι λάθος. Ο όρος που χρησιμοποιείται για να αναφερθεί στο στόχο που πρέπει να κυνηγήσει ο πράκτορας μέχρι να τον πετύχει είτε θα τον παραλείψει εντελώς, και φυσικά ο πράκτορας παραδέχεται την πιθανότητα της αποτυχίας, μια ρεαλιστική δυνατότητα σε πολλά πολύπλοκα συστήματα, ένα από τα οποία είναι ο παραδοσιακός τυποποιησμός τον οποίο αγνοεί. Το κεφάλαιο από τον Georgeff και τον Rao δίνουν μια σαφέστερη άποψη.

3.14 - Σχεδιασμός συστήματος

Ο σχεδιασμός των ισχυρών, και αποτελεσματικών συστημάτων είναι από τα πιο σημαντικά θέματα που πρέπει να λύσουν οι μηχανικοί λογισμικού, και έχει γίνει μια μεγάλη προσπάθεια για τον μεθοδικό σχεδιασμό του συστήματος, μια πολύ αυστηρή διαδικασία. Η ανάλυση, ο σχεδιασμός και η υλοποίηση ενός συστήματος έχουν μελετηθεί σχετικά λίγο στην κοινωνία της έρευνας των πρακτόρων ενώ ταυτόχρονα έχουν εξεταστεί και προταθεί ελάχιστες μεθοδολογίες σχεδιασμού που θα μπορούσαν να εφαρμοστούν σε συστήματα πρακτόρων. Η κατάσταση αυτή απαιτείται να αλλάξει και ώστε τα συστήματα πρακτόρων να γίνουν ευρέως γνωστά. Ειδικότερα οι ακόλουθες βασικές πολιτικές πρέπει που πρέπει να ακολουθηθούν είναι :

- Υιοθέτηση προσέγγισης η οποία βασίζεται σε πράκτορες. Σε ένα μεγάλο μέρος μιας υλοποίησης βασισμένη στους πράκτορες, το σκεπτικό για την επιλογή μιας προσέγγισης είναι ασαφές. Η ανάλυση των ιδιοτήτων και μεθόδων στον συγκεκριμένου τομέα αποτελεί ένα ανάγκη ώστε να γίνει πιο εύκολη η προσέγγιση αυτή.
- Κατασκευές μακρό συστημάτων. Μόλις δοθεί μια αφηρημένη, υψηλού επιπέδου προδιαγραφή συστήματος, πρέπει να μετατρέψουμε αυτή την προδιαγραφή σε πράκτορες, και συσχετίσουμε τους πράκτορες μεταξύ τους σε μία δομή συστήματος. Αυτή η διαδικασία είναι γνωστή και ως refinement. Κατά τον ίδιο τρόπο υπάρχει η αντικειμενοστραφής μεθοδολογία σχεδιασμού, η οποία δίνει τις κατευθυντήριες γραμμές σχετικά με τον προσδιορισμό των αντικειμένων και των αλληλεξαρτήσεων τους. Έτσι μπορούμε σύντομα να αναμένουμε έναν πράκτορα προσανατολισμένο σε μεθοδολογίες σχεδιασμού, οι οποίες προσφέρουν την καθοδήγηση για το πώς να αποσυνθέσουμε έναν πρόβλημα σε πράκτορες, και τι επιπτώσεις θα υπάρχουν στις επιδώσεις του συστήματος. Τα προβλήματα συνήθως επιδεινώνονται επειδή τα κληρονομικά συστήματα πρέπει να ενσωματωθούν σε συστήματα πρακτόρων. Αυτός ο σχεδιασμός πρέπει να γίνει τόσο από πάνω προς τα κάτω όσο και από κάτω προς τα πάνω, και οι δύο οι προοπτικές πρέπει να προσαρμοστούν.

3.15 - Εφαρμογές συστήματος

Ο σχεδιασμός ενός συστήματος το οποίο βασίζεται σε πράκτορες περιγράφει αφενός τους διαφορετικούς ρόλους που υπάρχουν σε ένα σύστημα και αφετέρου χαρακτηρίζει τις σχέσεις που υπάρχουν μεταξύ των ρόλων αυτών. Ωστόσο ο συγκεκριμένος σχεδιασμός δεν προϋποθέτει και την υλοποίηση των συγκεκριμένων πρακτόρων. Έτσι έχοντας προσδιορίσει του διάφορους ρόλους των πρακτόρων στο σύστημα, το επόμενο βήμα είναι να βρούμε πως μπορούν να πραγματοποιούντα καλύτερα. Μία αρχιτεκτονική πράκτορα χρειάζεται να υιοθετηθεί για κάθε ρόλο, και η οποία θα παρέχει τα απαιτούμενα λειτουργικά και μη λειτουργικά χαρακτηριστικά του κάθε ρόλου.

Έχουν αναπτυχθεί πολλές τεχνικές πράκτορα από την κοινότητα των ευφυών πρακτόρων, με πολλές διαφορετικές ιδιότητες. Υπάρχουν συστήματα ευφυών πρακτόρων που επιτρέπουν στους χρήστες να χτίσουν πράκτορες σαν συστήματα βασισμένα στη γνώση ή σαν απόδοση λογικού θεωρήματος. Για να δημιουργήσουμε πράκτορες οι οποίοι θα χρησιμοποιήσουν τέτοια συστήματα, ένας πρέπει να περνάει από μια πρότυπη διαδικασία συστήματος βασισμένο στη εκμαίευση γνώσης και στην εκπροσώπηση, κωδικοποιώντας την συμπεριφορά των πρακτόρων υπό τους όρους κανόνων, πλαισίων και σημασιολογικών δικτύων.

Στο άλλο άκρο, υπάρχουν πολλά πλαίσια πρακτόρων τα οποία προσφέρουν ενισχυμένες εκδόσεις γλώσσα προγραμματισμού Java τα οποία δεν περιέχουν καθόλου τεχνικές ευφυών πρακτόρων. Κανένα από αυτά τα άκρα είναι αυστηρά σωστά ή λάθος: έχουν και τα δυο πλεονεκτήματα και μειονεκτήματα. Σε γενικές γραμμές η πιο απλή λύση είναι και αυτή που θα λύσει το πρόβλημα. Υπάρχει νόημα στην χρησιμοποίηση λειτουργικών συστημάτων όπου θα χρησιμοποιηθούν απλοί πράκτορες Java. Προφανώς, χρειάζεται πιο λεπτομερή καθοδήγηση για να βοηθηθεί η διαδικασία λήψης αποφάσεων.

3.16 - Έλεγχος συστήματος, διόρθωση σφαλμάτων και επαλήθευση

Στα σχόλια που έκανε Gasser το 1987 [14] υποστηρίζει ότι ο συναγωνισμός και η αβεβαιότητα του τομέα αυτού έχουν καταστήσει δύσκολο να κατανοηθεί η δραστηριότητα ενός έξυπνου συστήματος, επίσης χρειάζονται γραφικά με τις δραστηριότητες του συστήματος η οποίες να συνδέονται με έξυπνα εργαλεία τα οποία βοηθάνε για την ανάπτυξη μια αναμενόμενης και παρατηρούμενης συμπεριφοράς.

Δυστυχώς ο έλεγχος του συστήματος η διόρθωση σφαλμάτων και η επαλήθευση είναι παραμελημένες περιοχές στην διαδικασία ανάπτυξης του πράκτορα. Οι προγραμματιστές χρειάζονται βοήθεια για να δούνε τη γίνεται, και χρειάζονται εγκαταστάσεις επαλήθευσης λαθών για να κάνουν βήματα για την εκτέλεση και την τροποποίηση της συμπεριφοράς όπου χρειάζεται.

Η οπτικοποίηση είναι πολύ σημαντική, για τις αποφάσεις στο τι συμβαίνει σε ασύγχρονα και παράλληλα συστήματα είναι εξαιρετικά δύσκολο έργο. Αυτό ισχύει τόσο στο εσωτερικό του πράκτορα και ακόμα περισσότερο μεταξύ των πρακτόρων. Οι παραδοσιακές δυνατότητες εντοπισμού σφαλμάτων επιτρέπουν την διακοπή μίας εκτέλεσης, την εξέταση των εσωτερικών καταστάσεων, την αλλαγή των καταστάσεων, την προσομοίωση των εισερχόμενων μηνυμάτων, πράγματα τα οποία είναι πολύ απαραίτητα. Στα περισσότερα συστήματα αυτά τα χαρακτηριστικά και τα εργαλεία συνήθως αναπτύσσονται από το μηδέν, γεγονός που κάνει αρκετά δύσκολη την ανάπτυξη τους και αποτελεί την πιο ανεπιθύμητη κατάσταση στην ανάπτυξη λογισμικού πρακτόρων.

Κεφάλαιο 4 – Εφαρμογές Πρακτόρων

4.1 Χαρακτηριστικά εφαρμογών πρακτόρων

Υπάρχουν πολλοί τρόποι με τους οποίους μπορούν να ταξινομηθούν οι εφαρμογές πρακτόρων. Μπορούν να ταξινομηθούν με βάση τον τύπο του πράκτορα, σύμφωνα με την τεχνολογία που εφαρμόζεται στις εφαρμογές ενός πράκτορα, ή από τον ίδιο τον τομέα της εφαρμογής. Επιλέγουμε να χρησιμοποιήσουμε τον τύπο του τομέα αφού αυτός είναι ο πιο διαδεδομένος τρόπος ταξινόμησης των εφαρμογών πρακτόρων.

Σε κάθε νέα τεχνολογία που θεωρείται χρήσιμη στην επιστήμη των υπολογιστών θα πρέπει να προσφέρονται δύο συγκεκριμένα πράγματα:

- Η ικανότητα επίλυσης προβλημάτων πέρα από τον αυτοματισμό και αυτό γίνεται για δυο λόγους:
 1. Είτε γιατί η υπάρχουσα τεχνολογία δεν μπορεί να χρησιμοποιηθεί για την λύση ενός συγκεκριμένου προβλήματος.
 2. Είτε γιατί θεωρήθηκε πολύ ακριβό να γίνει έρευνα για νέα τεχνολογία για το αντίστοιχο πρόβλημα.
- Η δυνατότητα να επιλυθούν προβλήματα πολύ πιο αποτελεσματικά(φτηνότερα, γρηγορότερα κ.λ.π) από ότι λύνονται είδη.

4.2 Βιομηχανικές εφαρμογές

Οι βιομηχανικές εφαρμογές στην τεχνολογία των πρακτόρων είναι μεταξύ των πρώτων που αναπτύχθηκαν: ήδη από το 1987, η αναφοράς του Parunak [44] αποκτούν σημασία εφαρμόζοντας το δικτυακό πρωτόκολλο κατανομής σε ένα περιβάλλον παραγωγής. Σήμερα οι πράκτορες εφαρμόζονται σε ένα εύρη φάσμα βιομηχανικών εφαρμογών.

Ο Parunak το 1987 [44] δημιούργησε τα YAMS ως άλλα συστήματα βιομηχανίας, τα οποία εφαρμόζουν το γνωστό πρωτόκολλο του Smith το οποίο δημιουργήθηκε το 1980 για την δημιουργία ελέγχου. Το βασικό πρόβλημα αναλύεται ως εξής. Η κατασκευή μιας επιχείρησης μοντελοποιείται ως μια ιεραρχία εργασιών κελιών. Για παράδειγμα θα υπάρξουν τέτοια κελιά για την άλεση, το σανίδωμα, ζωγραφική κ.λ.π. Τα συγκεκριμένα κελιά ομαδοποιούνται σε πιο ευέλικτα συστήματα παραγωγής (FMS), καθένα από το οποία θα παράγει συγκεκριμένες λειτουργίες όπως η συναρμολόγηση, η φόρτωση προϊόντων κ.λ.π. Η συλλογή των συγκεκριμένων συστημάτων είναι χωρισμένη σε ένα εργοστάσιο.

Μια απλή επιχείρηση ή ένας οργανισμός μπορούν να έχουν διαφορετικά εργοστάσια, και αυτά επίσης μπορούν επίσης να έχουν διπλή λειτουργικότητα ή δυνατότητες. Ο στόχος αυτών των συστημάτων είναι να διαχειριστούν αποτελεσματικά την παραγωγική διαδικασία των συγκεκριμένων μονάδων. Αυτή η διαδικασία καθορίζεται από κάποιες σταθερά μεταβαλλόμενες παραμέτρους, όπως είναι τα προϊόντα τα οποία κατασκευάζονται, διαθέσιμους πόρους, χρονικούς περιορισμούς και ούτω κάθε εξής.

Προκειμένου να επιτευχθεί αυτό το πολύπλοκο έργο, τα YAMS υιοθετούν μια προσέγγιση πολυπρακτόρων, όπου κάθε εργοστάσιο ή τα συστατικά τους παρουσιάζονται ως ένας πράκτορας. Το πρωτόκολλο σύμβασης δικτύου επιτρέπει τα καθήκοντα όπως εντολές παραγωγής κ.λ.π που πρέπει να ανατίθενται σε μεμονωμένα εργοστάσια, καθώς και από επιμέρους εργοστάσια μέχρι τα FMS, και στη συνέχεια σε μεμονωμένα εργασιακά κελιά. Το πρωτόκολλο σύμβασης βασίζεται στην ιδέα της διαπραγμάτευσης, και ως εκ τούτου τα YAMS βλέπουν το πρόβλημα της επιλογής, δηλαδή το πόσο

καλύτερα πρέπει να επεξεργάζονται την παραγωγή απαιτήσεων μιας εταιρίας σαν πρόβλημα διαπραγμάτευσης.

4.3 Διαδικασία ελέγχου

Η διαδικασία ελέγχου είναι μια φυσική εφαρμογή για ευφυείς πράκτορες και για συστήματα πολυπρακτόρων, δεδομένου ότι οι ελεγκτές της διαδικασίας είναι αυτόνομα και αντιδραστικά συστήματα. Αυτό δεν είναι τυχαίο, ο αριθμός των πρακτόρων που βασίζονται σε εφαρμογές ελέγχου θα πρέπει να αναπτυχθεί. Η πιο γνωστή είναι η ARCHON, μια πλατφόρμα λογισμικού για την ανάπτυξη συστημάτων πολυπρακτόρων, και μια σχετική μεθοδολογία για την ανάπτυξη της συγκεκριμένης πλατφόρμας. [60]

Το ARCHON έχει εφαρμοστεί σε συγκεκριμένες διαδικασίες ελέγχου εφαρμογών, συμπεριλαμβανομένης της διαχείρισης της μεταφοράς ηλεκτρικής ενέργειας, και τον έλεγχο επιτάχυνσης σωματιδίων. Επίσης έχει την διάκριση στο να είναι ένα από τα πρώτα δοκιμασμένα συστήματα πολυπρακτόρων στον κόσμο. Οι πράκτορες στο ARCHON είναι αρκετά βαριά υπολογιστικά συστήματα, με τέσσερα κύρια στοιχεία:

- Υψηλού επιπέδου επικοινωνία η οποία διαχειρίζεται την επικοινωνία μεταξύ των πρακτόρων.
- Προγραμματισμός και ενότητα συντονισμού, το οποίο στην ουσία είναι υπεύθυνο για το τι πρέπει να κάνει ο πράκτορας.
- Ένα ενημερωτικός πράκτορας μονάδων διαχείρισης, ο οποίος είναι υπεύθυνος για να διατηρήσει το μοντέλο του πράκτορα στο κόσμο.
- Ένα υποκειμενικά έξυπνο σύστημα, το οποίο αντιπροσωπεύει την εξειδίκευση του πράκτορα.

Τα τέσσερα αυτά στοιχεία μαζί μπορούν να θεωρηθούν σαν ένας πράκτορας wrapper ο οποίος μπορεί να χρησιμοποιηθεί για να συμπεριλάβει ένα υπάρχον ευφύες σύστημα και να το μετατρέψει σε ένα πράκτορα.

4.4 Έλεγχος εναέριας κυκλοφορίας

Για τα συστήματα εναέριας κυκλοφορίας γενικά έχουν παρουσιάσει έναν εξελιγμένο πράκτορα ο οποίος αντιλαμβάνεται το σύστημα έλεγχου της εναέριας κυκλοφορίας γνωστό και ως OASIS [60]. Το συγκεκριμένο σύστημα το οποίο βρίσκεται σε πρώιμο στάδιο στο αεροδρόμιο του Σύνδνεϋ, οι πράκτορες χρησιμοποιούνται για να αντιπροσωπεύουν τόσο τα αεροσκάφη αλλά και τα διάφορα συστήματα εναέριας κυκλοφορίας που είναι σε λειτουργία. Οι μεταφορά του πράκτορα παρέχει έναν φυσικό και χρήσιμο τρόπο μοντελοποίησης των πραγματικών αυτόνομων συστατικών.

Όταν ένα αεροσκάφος μπαίνει στο αεροδρόμιο του Σύνδνεϋ, ένας πράκτορας διατίθεται γι' αυτό, και επίσης αποκτά τις πληροφορίες και του στόχους που αντιστοιχούν στο συγκεκριμένο αεροσκάφος. Για παράδειγμα ένα αεροσκάφος έχει στόχο να προσγειωθεί σ'ένα συγκεκριμένο διάδρομο σε συγκεκριμένο χρόνο. Τα συστήματα εναέριας κυκλοφορίας είναι υπεύθυνα για την διαχείριση του συστήματος. Το OASIS υλοποιείται χρησιμοποιώντας τα AII'S το οποίο περιέχει Dmars συστήματα. Το σύστημα επιτρέπει στον πράκτορα να υλοποιείται με τη χρήση μοντέλου πεποίθησης-επιθυμίας, μια από τις πιο δημοφιλείς προσεγγίσεις για την αιτιολογία σχετικά με τους πράκτορες σε θεωρητικά συστήματα πολύ πρακτόρων.

Κατανεμημένη εφαρμογή για τον έλεγχο κυκλοφορίας με τη χρήση ευφυών πρακτόρων

Η βελτιστοποίηση της αστικής κυκλοφορίας είναι από τα πιο σημαντικά προβλήματα τόσο σε παγκόσμια κλίμακα αλλά και τόσο σε τοπικό επίπεδο, αλλά κυρίως στις μητροπολιτικές περιοχές. Το συγκεκριμένο πρόβλημα δημιουργήθηκε λόγω της αύξησης των οχημάτων, του πληθυσμού και τις οικονομικής ανάπτυξης. Η αύξηση αυτή μαζί με την έλλειψη προγραμματισμού της κυκλοφορίας μπορεί να οδηγήσει σε οικονομικές απώλειες. Παραδείγματος χάρη οι ηνωμένες πολιτείες της Αμερικής το 2007 έχασαν 82.7 δισεκατομμύρια δολάρια.

Τα συστήματα πολλαπλών πρακτόρων αναδεικνύονται σαν παραδείγματα ανάπτυξης λογισμικού. Εξαιτίας των χαρακτηριστικών των πρακτόρων όπως είναι η αυτονομία, αλλά και των ευφυών πρακτόρων όπως οι κοινωνικές ικανότητες και η αντιδραστικότητα οδηγούμαστε στο συμπέρασμα πως τα συστήματα πολλαπλών πρακτόρων είναι τα κατάλληλα για προβλήματα με διανεμημένη ή συνεργατική κληρονομική φύση. Η βελτιστοποίηση της κυκλοφορίας έχει όλα τα χαρακτηριστικά που απαιτεί η χρήση πρακτόρων.

Το Jade είναι από τα γνωστά πλαίσια για την ανάπτυξη τέτοιων συστημάτων, το οποίο χειρίζεται τη δημιουργία, τη επικοινωνία αλλά και την μετανάστευση των πρακτόρων με έναν ευέλικτο τρόπο, κάτι το οποίο επιτρέπει στους προγραμματιστές να συγκεντρωθούν. Για την συγκεκριμένη περίπτωση προτάθηκε ένα σύστημα το οποίο έχει δύο βασικούς παράγοντες:

- I TRAFFIC
- Traffic watcher

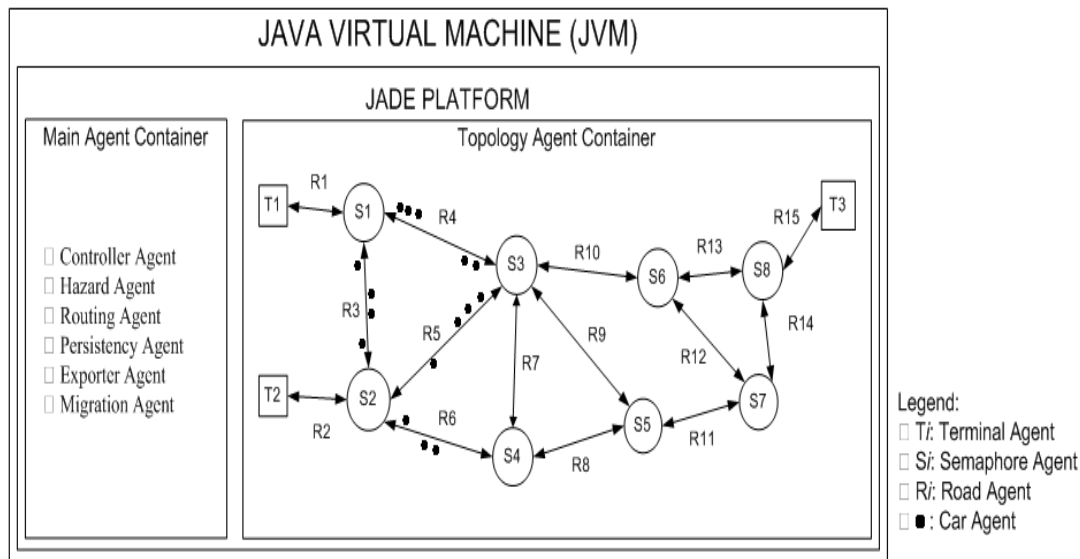
Λόγω της έλλειψης πρόσβασης σε αισθητήρες κίνησης, το traffic watcher αντιπροσωπεύει ένα περιβάλλον προσομοίωσης που χρησιμοποιεί συστήματα πολυπρακτόρων για να αντιπροσωπεύει την πραγματική τοπολογία. Η συγκεκριμένη εφαρμογή μπορεί να χρησιμοποιηθεί σε πραγματικές συνθήκες χωρίς πολλές αλλαγές, αλλάζοντας απλά τις τυχάιες γεννήτριες για τα μοτίβα κίνησης, με πραγματικά δεδομένα τα οποία προέρχονται από αισθητήρες οι οποίοι τοποθετούνται στο δρόμο. Το I traffic χρησιμοποιεί τον έλεγχο της κυκλοφορίας, την ανάλυση, την βελτιστοποίηση και τη διανομή του φόρτου εργασίας μέσα στο σύστημα. Η συγκεκριμένη λύση μπορεί να χρησιμοποιηθεί ακόμα και για offline συστήματα, εάν βέβαια οι παράμετροι κυκλοφορίας κινούνται σε φυσιολογικά επίπεδα, καθώς και σε online έλεγχο αν η κατάσταση είναι πιο περίπλοκη.

Καταρχήν η απόδοση ενός αστικού κυκλοφοριακού συστήματος εξαρτάται από την ικανότητα του συστήματος να ανταποκρίνεται σε νέες κυκλοφοριακές καταστάσεις και σε απροσδόκητα γεγονότα. Οι παράγοντες που επηρεάζουν τέτοια συστήματα είναι τα ατυχήματα, οι αλλαγές σε σχέδια κυκλοφορίας, η συμπεριφορά των οδηγών και οι κυκλοφοριακές αλλαγές.

Ένα έξυπνο σύστημα ελέγχου αστικής κυκλοφορίας, καταρχήν βασίζεται σε μια σειρά σηματοφόρων πρακτόρων οι οποίοι τοποθετούνται στις διασταυρώσεις, και από κυρίαρχους πράκτορες που επιβλέπουν την κυκλοφορία σε τοπικό αλλά και σε παγκόσμιο επίπεδο. Το συγκεκριμένο σύστημα βασίζει της αποφάσεις του σε στόχους, δυνατότητες, στην αντίληψη του περιβάλλοντος και σε τρέχοντα δεδομένα συστήματος. Το δυναμικό μοτίβο κυκλοφορίας, οι μεταβλητοί κανόνες και οι διαφορετικοί τρόποι δράσης για το κάθε γεγονός που βασίζονται σε προηγούμενη εμπειρία υποδηλώνει πως το σύστημα πρέπει να έχει παραπάνω έλεγχο, τόσο σε τοπικό αλλά και σε παγκόσμιο επίπεδο.

Τεχνολογία I watcher και i-traffic

Το i-watcher επιτρέπει στους χρήστες να προσομοιώσουν μια κανονική οδική κυκλοφορία με συγκεκριμένα οδικά χαρακτηριστικά. Γι'αυτό το λόγο αναπτύχθηκε ένα υβριδικό σύστημα πολυπρακτόρων, το οποίο συνδυάζει τα χαρακτηριστικά του Jade και του Jadex τα οποία ταιριάζουν σ'ένα Jmη χρόνο εκτέλεσης. Το διάγραμμα της συγκεκριμένης εφαρμογής φαίνεται στο ακόλουθο σχήμα.



Η πλατφόρμα Jade αποτελεί την στήριξη κατά τη διάρκεια όπου εκτελείται το συγκεκριμένο σύστημα. Επιτρέπει την άμεση επικοινωνία με τους εσωτερικούς πράκτορες αλλά και με πλατφόρμες εκτός. Ανάλογα με την τεχνολογία που χρησιμοποιείται αλλά και το ρόλο που έχουν σε κάθε λογική εφαρμογή, οι πράκτορες οι οποίοι σχεδιάστηκαν σε μια συγκεκριμένη πλατφόρμα ομαδοποιούνται σε δυο διαφορετικές δεξαμενές παραγόντων.

Η συγκεκριμένη τοπολογία αποτελείται από λογικούς Jadex πράκτορες οι οποίοι ακολουθούν το μοντέλο BDI, το οποίο μιμείται κύριες οντότητες της οδικής κυκλοφορίας όπως σηματοδότες κ.λ.π., αλλά επίσης και jade πράκτορες. Το traffic watcher στην ουσία προσομοιώνει ένα οδικό περιβάλλον κυκλοφορίας το οποίο θα αποθηκεύει διάφορα στατιστικά στοιχεία, τα οποία θα αναλύσει το i-traffic ώστε να μπορέσει να δώσει λύσεις για θέματα όπως η αποσυμφόρηση.

Η εφαρμογή είναι σχεδιασμένη να λειτουργεί με οποιαδήποτε τοπολογία με τη μορφή ενός γραφήματος το οποίο ορίζεται από ένα xml αρχείο και το οποίο επικυρώνεται από ένα προκαθορισμένο σχήμα XSD. Η αρχική τοπολογία χωρίζεται μεταξύ ενός αριθμού ενεργών πλατφορμών οι οποίες είναι λογικά συνδεδεμένες. Η προσομοίωση μπορεί να προβληθεί οποιαδήποτε στιγμή λόγω της υπηρεσίας εξαγωγής για κάθε πράκτορα της πλατφόρμας, το οποίο οδηγεί σ'ένα xml αρχείο, στο οποίο κάθε εφαρμογή διασύνδεσης μπορεί να αντιπροσωπεύει της πληροφορίες που μπορούν να χρησιμοποιηθούν.

Από την άλλη μεριά το i-traffic είναι μια διανεμημένη έξυπνη πλατφόρμα η οποία βελτιστοποιεί την αστική κυκλοφορία και προσφέρει λύσεις στα προβλήματα τα οποία έχουν τα UTC συστήματα. Αποτελείται από έξι ενότητες και χρησιμοποιεί το traffic watcher σαν την βάση του συστήματος πρακτόρων ώστε να δουλεύει με τις οντότητες που συμμετέχουν στην προσωμοιοτική διαδικασία:

1. Δεδομένα ανάκτησης και εξαγωγής
2. Διαχείριση και έλεγχος
3. Διανομή ενότητας
4. Ενότητα βελτιστοποίησης
5. Στατιστική ενότητα

6. Ενότητα μετανάστευσης

Η βάση του i-traffic είναι η προσομοίωση της αστικής κυκλοφορίας σε σχέση με την προηγούμενη και καθορισμένη υποδομή δρόμων, η οποία μπορεί να υποβληθεί σε επεξεργασία και εξαγωγή δεδομένων, μετατρέποντας τα δεδομένα από ένα τυποποιημένο έντυπο σε μια i-traffic οντότητα. Τα φιλτραρισμένα δεδομένα παρουσιάζονται στην επιφάνεια εργασίας για επεξεργασία και έλεγχο μέσω της μονάδας διαχείρισης ελέγχου. Το i-traffic επιτρέπει στο χρήστη να χωρίσει μια γεωγραφική περιοχή σε υπομήματα, έτσι ώστε να μπορέσει να εξυπηρετήσει ένα κομμάτι της αρχικής ζώνης(BLOCK). Το κάθε block αντιπροσωπεύεται στο σύστημα UTC μέσω μιας πλατφόρμας πράκτορα.

Πειράματα της συγκεκριμένης εφαρμογής

Τα πειράματα τα οποία έγιναν είχαν ως στόχο την σταθερότητα και την κατανάλωση των πόρων. Τα πειράματα τα οποία έγιναν κατά την διάρκεια της ανάπτυξης, εκτέλεσαν 250 με 300 πράκτορες αυτοκινήτων την ίδια στιγμή μέσα στο block κυκλοφορίας, ενώ η εφαρμογή συνεχίζει να συμπεριφέρεται κανονικά.

Δεδομένου ότι το traffic watcher είναι φτιαγμένο για να δουλεύει μέσα σ'ένα κέντρο δεδομένων, το οποίο μπορεί να παρέχει επαρκής επεξεργασμένους πόρους για πολύπλοκες προσομοίωσης, ο αριθμός αυτός μπορεί βέβαια να αυξηθεί σε 800 με 900 ενεργά αμάξια. Οι περιορισμοί μπαίνουν από της αρχιτεκτονικές τον JADE και JADEX. Τα πειράματα βασίζονται σε δύο διαφορετικά τμήματα του οδικού δικτύου που έχουν πραγματική ανταπόκριση, ως εξής:

1. Τοπολογία πρώτη: Μια λωρίδα κυκλοφορίας που αποτελείται από περίπου 100 φορείς και ένα μέσο όρο από 250 αμάξια κάθε στιγμή. Λαμβάνοντας υπόψη πως κάθε πράκτορας τρέχει σε μια ξεχωριστή λωρίδα, παρόμοια με τα σχέδια δράσης των φορέων σημάτων, τα οποία τρέχουν σε μια δευτερογενή λωρίδα, η περιγραφόμενη διαμόρφωση αναμένεται να φτάσει 600 με 700 λωρίδες ταυτόχρονα.
2. Τοπολογία δεύτερη: Μια λωρίδα κυκλοφορίας με περίπου 60 οντότητες όπως τερματικού σταθμούς, τμήματα οδικού δικτύου κ.λ.π, οι οποίες τρέχουν κανονικά 150 πράκτορες αυτοκινήτων κάθε χρονική στιγμή, με εκτίμηση ότι τρέχουν 400 με 500 ταυτόχρονα.

Γι'αυτές τις δοκιμές χωρίστηκαν τα block σε 4 και αναπτύχθηκαν σε 4 διαφορετικούς υπολογιστές με την ίδια σύνθεση. Δεδομένου ότι η παραγωγή των αυτοκινήτων γίνεται σε τυχαία χρονικά διαστήματα, μπορούμε να δούμε πως τα αμάξια στο σύστημα φτάνουν ταχύτερα στον προορισμό τους, έτσι στο προηγούμενο σχήμα είδαμε την μείωση του συνολικού αριθμού των αυτοκινήτων.

Στο προηγούμενο σχήμα το ποσοστό του μέσου όρου βελτιστοποίησης για το σύνολο των block όσον αφορά το χρόνο που σπαταλάνε τα αυτοκίνητα στο σύστημα περιμένοντας σ'έναν σηματοφορέα είναι 38,13%. Τέλος η συγκεκριμένη προσέγγιση μας δίνει τη δυνατότητα να συνδέεται με άλλες εφαρμογές που εφαρμόζουν ευφυείς πράκτορες επίσης. Επίσης μας δίνει το πλεονέκτημα να την χρησιμοποιούμε σε συγκεντρωτική προσέγγιση. Το μόνο πρόβλημα που παραμένει είναι η μεγάλη δύναμη υπολογιστών που χρειάζεται.

4.5 JADE (Java Agent DEvelopment Framework)

Η JADE (Java Agent DEvelopment Framework) είναι ένα πλαίσιο λογισμικού με στόχο την ανάπτυξη συστημάτων και εφαρμογών πολλαπλών πρακτόρων που προσαρμόζονται στα πρότυπα FIPA (Foundation for Intelligent Physical Agents) που αφορούν ευφυείς πράκτορες. Χρησιμοποιείται για την ανάπτυξη και την εκτέλεση των peer-to-peer εφαρμογών που είναι βασισμένες στους πράκτορες λογισμικού και που μπορούν αφανώς να εργαστούν και να επικοινωνήσουν τόσο σε ενσύρματο όσο και σε ασύρματο περιβάλλον [61].

Η JADE είναι πλήρως υλοποιημένη στη γλώσσα Java και αποτελείται από διάφορα πακέτα Java, που δίνουν στους προγραμματιστές εφαρμογών έτοιμα κομμάτια λειτουργικότητας και αφηρημένες (abstract) διεπαφές για συνηθισμένες, εξαρτώμενες από την εφαρμογή εργασίες. Η γλώσσα προγραμματισμού Java επιλέχθηκε λόγω των πολλών ελκυστικών χαρακτηριστικών της, που συνδέονται ιδιαίτερα με τον αντικειμενοστραφή προγραμματισμό στα κατανεμημένα ετερογενή περιβάλλοντα. Μερικά από αυτά τα χαρακτηριστικά είναι: Object Serialization, Reflection API and Remote Method Invocation (RMI) [61].

Η JADE περιλαμβάνει δύο κύρια προϊόντα: μια πλατφόρμα πρακτόρων λογισμικού σύμφωνη με τις προδιαγραφές της FIPA και ένα πακέτο για την ανάπτυξη Java πρακτόρων λογισμικού. Απλοποιεί την υλοποίηση συστημάτων πολλαπλών πρακτόρων μέσω ενός υλικο-λογισμικού που υποστηρίζει ότι συμμορφώνεται με τις προδιαγραφές FIPA και μέσω ενός συνόλου εργαλείων που υποστηρίζει τις φάσεις ανίχνευσης και διόρθωσης λαθών και ανάπτυξης εφαρμογών. Η πλατφόρμα πρακτόρων μπορεί να διανεμηθεί στις μηχανές (που δεν είναι απαραίτητο να έχουν το ίδιο λειτουργικό σύστημα) και η διαμόρφωση της μπορεί να ελεγχθεί μέσω μιας απομακρυσμένης γραφικής διεπαφής χρήστη (Graphical User Interface – GUI). Η διαμόρφωση μπορεί να αλλαχθεί ακόμη και στο χρόνο εκτέλεσης με τη μετακίνηση των πρακτόρων λογισμικού από μία μηχανή σε άλλη, όταν και όπως ζητηθεί. Η μόνη απαίτηση του συστήματος είναι να χρησιμοποιείται έκδοση Java Run Time 1.4 ή επόμενη [61].

Στόχος της JADE είναι να απλοποιηθεί η ανάπτυξη των συστημάτων πολλαπλών πρακτόρων εξασφαλίζοντας συμμόρφωση στις προδιαγραφές FIPA μέσω ενός περιεκτικού συνόλου υπηρεσιών συστήματος και πρακτόρων λογισμικού: υπηρεσία ονομάτων, υπηρεσία κίτρινων-σελίδων, μεταφορά μηνυμάτων, υπηρεσία parsing και βιβλιοθήκη πρωτοκόλλων αλληλεπίδρασης FIPA έτοιμων να χρησιμοποιηθούν. Όλη η επικοινωνία πρακτόρων λογισμικού εκτελείται μέσω μηνυμάτων, όπου η γλώσσα στα μηνύματα είναι FIPA ACL (Agent Communication Language).

Η αρχιτεκτονική επικοινωνίας προσφέρει έναν ευέλικτο και αποδοτικό τρόπο ανταλλαγής μηνυμάτων, όπου η JADE δημιουργεί και διαχειρίζεται μια ουρά εισερχόμενων ACL μηνυμάτων, ξεχωριστά σε κάθε πράκτορα. Οι πράκτορες μπορούν να έχουν πρόσβαση στην ουρά τους μέσω συνδυασμού διάφορων τρόπων όπως το μπλοκάρισμα, η σταθμοσκόπηση, η λήξη εγκυρότητας μιας ενέργειας και το ταίριασμα μοτίβων (blocking, polling, timeout and pattern matching).

Το πλήρες πρότυπο επικοινωνίας FIPA έχει υλοποιηθεί και τα συστατικά του έχουν σαφώς διαχωριστεί και ενσωματωθεί πλήρως, όπως τα πρωτόκολλα αλληλεπίδρασης, τα ACL μηνύματα, οι γλώσσες περιεχομένου, τα σχήματα κωδικοποίησης, η ανάπτυξη οντολογιών και τα πρωτόκολλα μεταφοράς μηνυμάτων.

Ο μηχανισμός μεταφοράς μηνυμάτων, ειδικότερα, προσαρμόζεται σε κάθε κατάσταση, επιλέγοντας το καταλληλότερο διαθέσιμο πρωτόκολλο. Σήμερα, χρησιμοποιούνται τα πρωτόκολλα Java RMI, event-notification, και IIOP, ωστόσο περισσότερα πρωτόκολλα μπορούν να προστεθούν εύκολα και η ολοκλήρωση του HTTP έχει επιτευχθεί ήδη. Τα περισσότερα από τα πρωτόκολλα αλληλεπίδρασης που καθορίζονται από τη FIPA είναι ήδη διαθέσιμα και μπορούν να υλοποιηθούν, έπειτα από τον καθορισμό της εξαρτημένης από την εφαρμογή συμπεριφοράς για την κάθε κατάσταση του πρωτοκόλλου.

Η SL και η οντολογία διαχείρισης πρακτόρων έχει ήδη υλοποιηθεί, καθώς επίσης και η υποστήριξη για τις καθοριζόμενες από το χρήστη γλώσσες περιεχομένου και τις οντολογίες που μπορούν να

υλοποιηθούν, να καταχωρηθούν με τους πράκτορες, και να χρησιμοποιηθούν αυτόματα από το πλαίσιο.

Στη JADE οι πράκτορες υλοποιούνται ως νήματα με ένα νήμα να αντιστοιχεί σε κάθε πράκτορα, αλλά πρέπει συχνά να εκτελέσουν παράλληλες εργασίες. Πέρα από τη λύση των πολύ-νημάτων, που προσφέρεται άμεσα από τη γλώσσα Java, η JADE υποστηρίζει επίσης τον προγραμματισμό συνεργαζόμενων συμπεριφορών, όπου προγραμματίζονται οι εργασίες αυτές με έναν εύκολο και αποτελεσματικό τρόπο. Το περιβάλλον αυτό περιλαμβάνει επίσης μερικές έτοιμες να χρησιμοποιηθούν συμπεριφορές για τις πιο κοινές εργασίες στον προγραμματισμό πρακτόρων, όπως πρωτόκολλα αλληλεπίδρασης FIPA, ενεργοποίηση κατά την εκπλήρωση μιας συνθήκης και κατασκευή σύνθετων εργασιών ως συναθροίσεις απλούστερων. Μεταξύ άλλων, η JADE προσφέρει επίσης την αποκαλούμενη JessBehaviour που επιτρέπει την πλήρη ολοκλήρωση με τη JESS, όπου η JADE παρέχει το κέλυφος του πράκτορα και εγγυάται (όπου είναι δυνατόν) τη συμμόρφωση FIPA, ενώ η JESS είναι η μηχανή του πράκτορα που εκτελεί όλο τον απαραίτητο συλλογισμό.

Η μεγάλη δύναμη της JADE είναι η φορητότητά της σε διαφορετικά περιβάλλοντα εκτέλεσης: χάρη στη συμβολή του ευρωπαϊκού IST προγράμματος αποκαλούμενου LEAP, η JADE έχει ενδολειτουργικές εκδόσεις για κάθε προφίλ μηχανών Java, από J2EE για τους Internet servers μέχρι J2ME MIDP για τα κινητά τερματικά. Έχει εξεταστεί ήδη η συμβατότητά της με διαφορετικά κινητά τερματικά όπου μπορεί να εγκατασταθεί λόγω μειωμένου memory footprint (100 - 50KB) μεταξύ των οποίων: Nokia 3650, Motorola Accompli008, Siemens SX45, PalmVx, Compaq iPaq, Psion5MX, HP Jornada 560.

Προκειμένου να ικανοποιήσει τους περιορισμούς μνήμης και υπολογιστικής ισχύος των κινητών συσκευών και τα χαρακτηριστικά των ασύρματων δικτύων (ιδιαίτερα GPRS) από την άποψη του εύρους ζώνης, της καθυστέρησης μεταφοράς (latency), της διαλείπουσας συνδετικότητας και της μεταβλητότητας IP διευθύνσεων, και προκειμένου να είναι συγχρόνως αποδοτική όταν εκτελείται σε σταθερούς hosts δικτύων, η JADE μπορεί να διαμορφωθεί ώστε να προσαρμοστεί στα χαρακτηριστικά του περιβάλλοντος ανάπτυξης. Η JADE αρχιτεκτονική είναι εντελώς σπονδυλωτή και, με την ενεργοποίηση ορισμένων τμημάτων (modules) αντί άλλων, είναι δυνατό να καλυφθούν οι διαφορετικές απαιτήσεις από την άποψη της συνδετικότητας, μνήμης και υπολογιστικής ισχύος. Λεπτομερέστερα, ένα τμήμα (module) αποκαλούμενο LEAP επιτρέπει τη βελτιστοποίηση όλων των μηχανισμών επικοινωνίας για συσκευές με περιορισμένους πόρους και συνδεδεμένες μέσω ασύρματων δικτύων.

Η JADE-S είναι ένα πρόσθετο τμήμα (add-on) της JADE. Επιτρέπει την προστασία ενός βασισμένου στη JADE συστήματος πολλαπλών πρακτόρων από τις επιθέσεις ασφάλειας. Λεπτομερέστερα, όλα τα συστατικά (πράκτορες, και containers) σε μια πλατφόρμα ανήκουν σε εξακριβωμένους χρήστες, οι οποίοι, στη συνέχεια, εξουσιοδοτούνται από τον διαχειριστή πλατφόρμας να εκτελέσουν μόνο ορισμένες προνομιούχες ενέργειες. Κάθε πράκτορας έχει στην κατοχή του ένα δημόσιο και ιδιωτικό κλειδί με τη βοήθεια του οποίου μπορεί να υπογράψει και να κρυπτογραφήσει μηνύματα.

Η JADE είναι εξαιρετικά ευπροσάρμοστη και επομένως, όχι μόνο προσαρμόζεται στους περιορισμούς περιβαλλόντων με περιορισμένους πόρους, αλλά έχει ήδη ενσωματωθεί σε σύνθετες αρχιτεκτονικές όπως .NET ή J2EE όπου και γίνεται υπηρεσία εκτέλεσης πολυμερών δυναμικών εφαρμογών.

Οι τομείς εφαρμογών όπου τα συστήματα JADE μπορούν να σχεδιαστούν αποτελεσματικά και να επεκταθούν είναι πολλοί και εκτείνονται από τις υπηρεσίες του διαδικτύου, που απευθύνονται τόσο στον καταναλωτή όσο και στις εταιρίες, ως το κινητό περιβάλλον, με τα βέλτιστα αποτελέσματα. Επίσης, χρησιμοποιείται στον τομέα των εφαρμογών machine-to-machine, όπου το peers proactivity και ο καταναλωμένος συντονισμός εργασιών είναι λειτουργίες που εξυπηρετούν τέλεια τις χαρακτηριστικές αυτοματοποιημένες απαιτήσεις των δικτύων. Όλα αυτά τα περιβάλλοντα μπορούν να μοιραστούν το ίδιο βασικό σύνολο λειτουργιών που απαιτούνται για να υποστηρίξουν πρωτόκολλα επικοινωνίας, αυτόνομες συμπεριφορές και ομότιμη διαχείριση.

Η JADE είναι ελεύθερο λογισμικό και διανέμεται από την TILAB, κάτοχο πνευματικών δικαιωμάτων, με άδεια Open Source υπό τους όρους του LGPL (Lesser General Public License Version 2). Από τον Μάιο του 2003 έχει δημιουργηθεί η JADE Board, η οποία επιτηρεί τη διαχείριση του προγράμματος JADE. Αυτήν την περίοδο η JADE Board αριθμεί 5 μέλη: TILAB, Motorola, Whitestein Technologies AG, Profactor GmbH, και E&A της France Telecom.

Η JADE χρησιμοποιείται από διάφορες επιχειρήσεις και ακαδημαϊκές ομάδες, μέλη και μη μέλη της FIPA, όπως BT, CNET, NHK, Imperial College, IRST, KPN, το πανεπιστήμιο Helsinki, INRIA, ATOS και πολλές άλλες.

Η JADE είναι ένα εμπορικό σήμα CSELT. Έχει αναπτυχθεί από κοινού από τη CSELT και την ομάδα Εφαρμοσμένης Μηχανικής Υπολογιστών του Πανεπιστημίου της Πάρμας .

Χαρακτηριστικά της JADE

Η JADE προσφέρει στον προγραμματιστή πρακτόρων τα εξής χαρακτηριστικά[61]:

- Κατανεμημένη πλατφόρμα πρακτόρων λογισμικού. Η πλατφόρμα πρακτόρων μπορεί να εκτείνεται σε πολλούς διαφορετικούς hosts (δεδομένου ότι μπορούν να συνδεθούν μέσω RMI). Μόνο μία εφαρμογή της Java, και επομένως μόνο μία Java Virtual Machine, εκτελείται σε κάθε host. Οι πράκτορες υλοποιούνται ως νήματα της Java και ζουν μέσα σε Agent Containers που παρέχουν την υποστήριξη χρόνου εκτέλεσης στους πράκτορες.
- Γραφική διεπαφή χρήστη (GUI) για την απομακρυσμένη διαχείριση, παρακολούθηση και έλεγχο της κατάστασης των πρακτόρων λογισμικού, επιτρέποντας, παραδείγματος χάριν, να σταματήσει και να επανεκκινήσει τους πράκτορες. Το GUI επιτρέπει επίσης τη δημιουργία και την εκκίνηση της εκτέλεσης ενός πράκτορα σε έναν απομακρυσμένο host υπό τον όρο ότι ένα κιβώτιο (container) πρακτόρων τρέχει ήδη καθώς και τον έλεγχο άλλων συμμορφωμένων με τη FIPA πλατφόρμων πρακτόρων.
- Εργαλεία debugging που βοηθούν στην ανάπτυξη εφαρμογών πολλαπλών πρακτόρων βασισμένων στη JADE.
- Ενδοπλατφορμική κινητικότητα πρακτόρων, συμπεριλαμβανομένης της μεταφοράς τόσο της κατάστασης όσο και του κώδικα (όταν είναι απαραίτητο) του πράκτορα.
- Υποστήριξη στην εκτέλεση πολλαπλών, παράλληλων και ταυτόχρονων δραστηριοτήτων πρακτόρων, μέσω του μοντέλου συμπεριφοράς. Η JADE προγραμματίζει χρονικά τις συμπεριφορές των πρακτόρων non-preemptively.
- Πλατφόρμα πρακτόρων που συμμορφώνεται με τη FIPA. Η πλατφόρμα περιλαμβάνει το AMS (Agent Management System - σύστημα διαχείρισης πρακτόρων), τον DF (Directory Facilitator - βοηθός καταλόγου), και το ACC (Agent Communication Channel - δίαυλο επικοινωνίας πρακτόρων). Αυτά τα τρία συστατικά ενεργοποιούνται αυτόματα κατά την εκκίνηση της πλατφόρμας πρακτόρων.
- Πολλοί σύμφωνοι με τις προδιαγραφές της FIPA DFs μπορούν να εκκινηθούν στο χρόνο εκτέλεσης προκειμένου να υλοποιηθούν εφαρμογές πολλαπλών περιοχών, όπου μια περιοχή είναι ένα λογικό σύνολο πρακτόρων, των οποίων οι υπηρεσίες διαφημίζονται από έναν κοινό βοηθό (DF). Κάθε DF κληρονομεί ένα GUI και όλες τις τυποποιημένες ικανότητες που καθορίζονται από τη FIPA (δηλ. ικανότητα εγγραφής, διαγραφής, τροποποίησης και αναζήτησης περιγραφών πρακτόρων και ικανότητα ένωσης σε συνομοσπονδία μέσα σε ένα δίκτυο DF's).
- Αποδοτική μεταφορά μηνυμάτων ACL μέσα στην ίδια πλατφόρμα πρακτόρων. Στην πραγματικότητα, τα μηνύματα μεταφέρονται κωδικοποιημένα ως Java αντικείμενα, παρά ως

συμβολοσειρές, προκειμένου να αποφευχθούν marshalling και unmarshalling διαδικασίες. Όταν ο αποστολέας ή ο παραλήπτης δεν ανήκουν στην ίδια πλατφόρμα, το μήνυμα αυτόματα μετατρέπεται σε / από σύνταξη, κωδικοποίηση και πρωτόκολλο μεταφοράς, που είναι σύμφωνα με τη FIPA. Αυτή η μετατροπή δεν είναι εμφανής στους υπεύθυνους υλοποίησης πρακτόρων λογισμικού, που πρέπει μόνο να ασχοληθούν με Java αντικείμενα.

- Βιβλιοθήκη FIPA πρωτοκόλλων αλληλεπίδρασης έτοιμων να χρησιμοποιηθούν.
- Αυτόματη εγγραφή και διαγραφή πρακτόρων με το AMS.
- Υπηρεσία ονομάτων σύμφωνη με τη FIPA: στην εκκίνηση οι πράκτορες λαμβάνουν το GUID (Globally Unique Identifier) τους από την πλατφόρμα.
- Υποστήριξη για καθοριζόμενες στην εφαρμογή γλώσσες περιεχομένου και οντολογίες.
- Διεπαφή InProcess για να επιτρέψει στις εξωτερικές εφαρμογές να εκκινούν αυτόνομους πράκτορες.

Εργαλεία της JADE

Η JADE περιλαμβάνει μερικά εργαλεία που απλοποιούν τη διαχείριση πλατφόρμων και την ανάπτυξη εφαρμογών. Τα παρακάτω εργαλεία είναι διαθέσιμα, αυτή την περίοδο [61]:

- Ο Remote Management Agent (RMA), που ενεργεί ως γραφική κονσόλα για τη διαχείριση και τον έλεγχο της πλατφόρμας. Ένα πρώτο στιγμιότυπο ενός RMA μπορεί να αρχίσει με την επιλογή γραμμής εντολών ("- gui "), αλλά έπειτα περισσότερα από ένα GUI μπορούν να ενεργοποιηθούν. Η JADE διατηρεί τη συνοχή μεταξύ πολλαπλών RMAs, μέσω multicasting γεγονότων σε όλους τους RMAs. Επιπλέον, η κονσόλα RMA είναι σε θέση να εκκινήσει άλλα JADE εργαλεία.
- Ο Dummy Agent είναι ένα εργαλείο ελέγχου, ανίχνευσης και διόρθωσης λαθών, αποτελούμενο από ένα GUI και έναν JADE πράκτορα. Χρησιμοποιώντας το GUI είναι δυνατό να συνθέσει μηνύματα ACL και να τα στείλει σε άλλους πράκτορες. Είναι επίσης δυνατό να επιδειχθεί η λίστα όλων των ACL μηνυμάτων που στέλνονται ή λαμβάνονται, συμπληρωμένη με πληροφορίες χρονικής σφραγίδας (timestamp) ώστε να επιτραπεί η καταγραφή και η ακρόαση συνομιλιών πρακτόρων.
- Ο Sniffer είναι πράκτορας που μπορεί να συλλάβει τα μηνύματα ACL ενώ μεταφέρονται, και να τα εκθέτει γραφικά χρησιμοποιώντας μια σημειογραφία παρόμοια με τα διαγράμματα ακολουθίας UML. Είναι χρήσιμος για την ανίχνευση και διόρθωση των λαθών στις κοινωνίες πρακτόρων λογισμικού, παρατηρώντας πώς αυτές ανταλλάσσουν ACL μηνύματα.
- Ο Introspector είναι ένας πράκτορας που επιτρέπει την παρακολούθηση του κύκλου ζωής ενός πράκτορα, των ανταλλασσόμενων ACL μηνυμάτων και των συμπεριφορών στην εκτέλεση.
- Το DF GUI είναι ένα πλήρες GUI που χρησιμοποιείται από τον προκαθορισμένο Directory Facilitator (DF) της JADE και μπορεί επίσης να χρησιμοποιηθεί από κάθε άλλο DF που ο χρήστης μπορεί να χρειαστεί. Με αυτό τον τρόπο, ο χρήστης μπορεί να δημιουργήσει ένα σύνθετο δίκτυο περιοχών και υπο-περιοχών κίτρινων σελίδων καταλόγου. Αυτό το GUI επιτρέπει απλά και διαισθητικά τον έλεγχο της βάσης των γνώσεων ενός DF, τη συνένωση σε

ομοσπονδία ενός DF με άλλους DFs, και τον απομακρυσμένο έλεγχο (εγγραφή/ διαγραφή/ τροποποίηση/ αναζήτηση) της βάσης γνώσεων του γονέα DF και επίσης των παιδιών DF (υλοποιώντας το δίκτυο των περιοχών και υποπεριοχών).

- Ο LogManagerAgent είναι ένας πράκτορας που επιτρέπει την ανάθεση σε χρόνο εκτέλεσης πληροφοριών καταχώρησης (logging) τόσο για τις κλάσεις JADE όσο και για τις συγκεκριμένες κλάσεις εφαρμογής που χρησιμοποιούν Java Logging.
- Ο SocketProxyAgent είναι ένας απλός πράκτορας, που ενεργεί ως αμφίδρομη πύλη μεταξύ μιας πλατφόρμας JADE και μιας συνηθισμένης TCP/IP σύνδεσης. Τα μηνύματα ACL, που ταξιδεύουν μέσω της JADE ιδιόκτητης υπηρεσίας μεταφορών, μετατρέπονται σε απλές συμβολοσειρές ASCII και στέλνονται μέσω μιας σύνδεσης υποδοχών. Αντίστροφα, τα μηνύματα ACL μπορούν να μεταφερθούν μέσω αυτής της TCP/IP σύνδεσης στην πλατφόρμα JADE. Αυτός ο πράκτορας είναι χρήσιμος, π.χ. για να χειριστεί τα firewalls του δικτύου ή για να παρέχει αλληλεπιδράσεις πλατφόρμων με Java applets μέσα σε έναν web browser.

Αρχιτεκτονική της JADE

Κάθε τρέχον στιγμιότυπο του περιβάλλοντος εκτέλεσης της JADE ονομάζεται Container καθώς μπορεί να περιέχει αρκετούς πράκτορες. Το σύνολο των ενεργών containers ονομάζεται πλατφόρμα (Platform) και παρέχει ένα ομοιογενές στρώμα που κρύβει στους πράκτορες (και στους υπεύθυνους για την ανάπτυξη εφαρμογής επίσης) την πολυπλοκότητα και την ποικιλομορφία των βασικών συστατικών (hardware, λειτουργικά συστήματα, τύποι δικτύων, JVM).

Κάθε πράκτορας προσδιορίζεται από ένα "αναγνωριστικό πρακτόρων" (agent identifier - AID). Ένα αναγνωριστικό πρακτόρων περιλαμβάνει ένα παγκόσμιο μοναδικό όνομα συν διάφορες διευθύνσεις. Το όνομα στη JADE έχει τη μορφή <agentname>@<platform-name> και οι διευθύνσεις είναι οι διευθύνσεις της πλατφόρμας όπου ζουν οι πράκτορες. Η σύνταξη αυτών των διευθύνσεων είναι μια ακολουθία URI και αυτές χρησιμοποιούνται μόνο όταν πρέπει ένας πράκτορας να επικοινωνήσει με έναν άλλο πράκτορα που ζει σε μια διαφορετική πλατφόρμα.

Η κατανεμημένη αρχιτεκτονική της JADE στηρίζεται σε έναν ειδικό κόμβο, ονομαζόμενο Main container, για να συγχρονίσει όλους τους άλλους κόμβους και για να κρατά ενιαία όλη την πλατφόρμα.

Ένας τουλάχιστον Main container πρέπει να είναι ενεργός στην πλατφόρμα και όλοι οι άλλοι containers εγγράφονται σε αυτόν μόλις εκκινούν. Συμπερασματικά, ο πρώτος container που εκκινεί σε μια πλατφόρμα πρέπει να είναι Main container. Οι πρόσθετοι containers πρακτόρων πρέπει να είναι "κανονικοί" (δηλ. μη-κύριοι) containers και πρέπει να τους δοθεί πού θα βρουν (host και θύρα) τον Main container τους (δηλ. τον Main container που εγγράφονται). Αυτοί οι κανονικοί containers μπορούν να εκκινήσουν έπειτα στον ίδιο host, ή σε απομακρυσμένους hosts, οι οποίοι συνδέονται με τον Main Container της πλατφόρμας πρακτόρων, με συνέπεια ένα κατανεμημένο σύστημα που φαίνεται από έξω ως μια ενιαία πλατφόρμα πρακτόρων.

Παρόλο που οι περισσότερες λειτουργίες της JADE είναι αποκεντρωμένες, υπάρχουν κάποια απαραίτητα χαρακτηριστικά που υποστηρίζονται μόνο από τον Main Container.

Αυτά τα χαρακτηριστικά είναι:

- Διαχείριση του Container Table (δηλ. του σύνολο όλων των κόμβων που συνθέτουν την κατανεμημένη πλατφόρμα).

- Διαχείριση του Global Agent Descriptor Table (δηλ. του σύνολο όλων των πρακτόρων λογισμικού που φιλοξενούνται από την κατανεμημένη πλατφόρμα, μαζί με την τρέχουσα θέση τους).
- Διαχείριση του πίνακα MTP (δηλ. του σύνολο όλων των παραταγμένων σημείων τέλους (endpoints) MTP, μαζί με τη θέση παράταξης).
- Φιλοξενία του πράκτορα AMS της πλατφόρμας.
- Φιλοξενία του πράκτορα Default DF της πλατφόρμας.

Εάν ένα άλλο Main container εκκινήσει κάπου στο δίκτυο αποτελεί μια διαφορετική πλατφόρμα στην οποία οι νέοι κανονικοί containers μπορούν ενδεχομένως να εγγραφούν.

Οι JADE πράκτορες προσδιορίζονται από ένα μοναδικό όνομα και, υπό τον όρο ότι ξέρουν ο ένας το όνομα του άλλου, μπορούν να επικοινωνήσουν διαφανώς ανεξάρτητα από την πραγματική θέση τους: ίδιος container, διαφορετικοί containers στην ίδια πλατφόρμα ή διαφορετικές πλατφόρμες.

Η πλατφόρμα πρακτόρων JADE περιλαμβάνει όλους εκείνους τους υποχρεωτικούς πράκτορες που διαχειρίζονται την πλατφόρμα, οι οποίοι είναι ο ACC, ο AMS, και ο DF. Όλη η επικοινωνία πρακτόρων λογισμικού εκτελείται μέσω της διαβίβασης μηνυμάτων, όπου η γλώσσα των μηνυμάτων είναι η FIPA ACL.

Ο πράκτορας ACC

Το σύστημα μεταφορών μηνυμάτων, αποκαλούμενο επίσης διάυλος επικοινωνίας πρακτόρων (Agent Communication Channel - ACC), είναι ο πράκτορας που παρέχει την πορεία για τη βασική επαφή μεταξύ των πρακτόρων λογισμικού μέσα και έξω από την πλατφόρμα. Είναι η προεπιλεγμένη μέθοδος επικοινωνίας που προσφέρει μια αξιόπιστη, τακτική και ακριβή στερεότυπη υπηρεσία μηνυμάτων. Αυτός ο πράκτορας [8] είναι ένα CORBA IIOP server αντικείμενο και δέχεται τις απομακρυσμένες αιτήσεις. Κάθε φορά που λαμβάνει ένα ACL μήνυμα που κωδικοποιείται ως string (συνήθως από τους non-JADE πράκτορες), αναλύει το μήνυμα και το μετατρέπει σε ένα Java ACLMessage αντικείμενο που χρησιμοποιείται από όλους τους JADE πράκτορες. Μπορεί επίσης να εκτελέσει την αντίστροφη μετατροπή όταν στέλνει ένας JADE πράκτορας ένα μήνυμα σε έναν non-JADE πράκτορα.

Ο πράκτορας AMS

Ο AMS (Agent Management System - Σύστημα διαχείρισης πρακτόρων) είναι ο πράκτορας που ασκεί τον εποπτικό έλεγχο της πρόσβασης και της χρήσης της πλατφόρμας πρακτόρων. Μόνο ένας AMS υπάρχει σε μια ενιαία πλατφόρμα. Ο AMS παρέχει τις υπηρεσίες “λευκών σελίδων” (υπηρεσία ονομάτων, δηλ. εξασφαλίζει ότι κάθε πράκτορας στην πλατφόρμα έχει ένα μοναδικό όνομα) και κύκλου ζωής, διατηρώντας έναν κατάλογο των αναγνωριστικών (AID) και της κατάστασης των πρακτόρων. Κάθε πράκτορας πρέπει να καταχωρηθεί με έναν AMS προκειμένου να αποκτήσει ένα έγκυρο AID. Ο AMS αντιπροσωπεύει την αρχή στην πλατφόρμα (παραδείγματος χάριν είναι δυνατό να δημιουργηθούν / σκοτωθούν πράκτορες σε απομακρυσμένα containers με μια αίτηση αυτού στον AMS).

Ο πράκτορας DF

Μια υπηρεσία “κίτρινων σελίδων” επιτρέπει στους πράκτορες να δημοσιεύσουν μια ή περισσότερες υπηρεσίες που παρέχουν έτσι ώστε άλλοι πράκτορες να μπορούν να τις βρουν και να τις εκμεταλλευτούν προκειμένου να επιτευχθούν οι στόχοι τους.

Η υπηρεσία κίτρινων σελίδων στη JADE (σύμφωνα με την προδιαγραφή FIPA) παρέχεται από έναν πράκτορα αποκαλούμενο DF (Directory Facilitator - βοηθός καταλόγου). Κάθε πλατφόρμα σύμφωνη με τις προδιαγραφές της FIPA φιλοξενεί έναν προεπιλεγμένο (default) πράκτορα DF (του οποίου τοπικό όνομα είναι "df"). Άλλοι DF πράκτορες μπορούν να ενεργοποιηθούν και διάφοροι DF πράκτορες (συμπεριλαμβανομένου και του προεπιλεγμένου) μπορούν να συνενωθούν σε ομοσπονδία έτσι ώστε να παρέχουν έναν ενιαίο καταναμεμημένο κατάλογο κίτρινων σελίδων.

Εφόσον ο DF είναι ένας πράκτορας, είναι δυνατή η αλληλεπίδραση με αυτόν όπως συνήθως, με την ανταλλαγή ACL μηνυμάτων που χρησιμοποιούν μια κατάλληλη γλώσσα περιεχομένου (η SLO γλώσσα) και μια κατάλληλη οντολογία (η FIPA-agent-management ontology) σύμφωνα με την προδιαγραφή FIPA. Προκειμένου να απλοποιηθούν αυτές οι αλληλεπιδράσεις, ωστόσο, η JADE παρέχει την κλάση `jade.domain.DFService` με τη βοήθεια της οποίας είναι δυνατό να δημοσιευθούν και αναζητηθούν υπηρεσίες μέσω κλήσεων των μεθόδων της κλάσης.

Ένας πράκτορας που επιθυμεί να δημοσιεύσει μια ή περισσότερες υπηρεσίες πρέπει να παρέχει στον DF μια περιγραφή συμπεριλαμβανομένου του AID του, ενδεχομένως τη λίστα γλωσσών και οντολογιών που οι άλλοι πράκτορες πρέπει να ξέρουν για να αλληλεπιδράσουν με αυτόν και τη λίστα δημοσιευόμενων υπηρεσιών. Για κάθε δημοσιευμένη υπηρεσία παρέχεται μια περιγραφή που περιλαμβάνει τον τύπο υπηρεσιών, το όνομα υπηρεσιών, τις γλώσσες και τις οντολογίες που απαιτούνται για να εκμεταλλευτούν εκείνη την υπηρεσία και διάφορες συγκεκριμένες ιδιότητες της υπηρεσίας.

Η αρχιτεκτονική λογισμικού της JADE είναι βασισμένη στη συνύπαρξη διάφορων εικονικών μηχανών Java (Java Virtual Machines - JVMs) και η επικοινωνία στηρίζεται στην Java RMI (Remote Method Invocation) μεταξύ διαφορετικών JVMs και στη σηματοδότηση γεγονότων (event signaling) μέσα σε μία ενιαία JVM. Η πλατφόρμα πρακτόρων μπορεί να καταναμεμηθεί σε διάφορους hosts. Κάθε JVM είναι ένα βασικό container πρακτόρων που παρέχει ένα πλήρες περιβάλλον εκτέλεσης για τους πράκτορες και επιτρέπει σε διάφορους πράκτορες να εκτελεστούν ταυτόχρονα στον ίδιο host. Σε γενικές γραμμές, η αρχιτεκτονική επιτρέπει επίσης σε διάφορες JVMs να εκτελεστούν στον ίδιο host. Εντούτοις, αυτό αποθαρρύνεται λόγω της αύξησης του overhead και της έλλειψης οποιουδήποτε οφέλους.

Κάθε container πρακτόρων είναι ένα πολύ-νηματικό (multithreaded) περιβάλλον εκτέλεσης που αποτελείται από ένα νήμα για κάθε πράκτορα συν τα νήματα συστημάτων που παράγονται από το RMI σύστημα χρόνου εκτέλεσης για την αποστολή μηνυμάτων. Ένα ειδικό container, ο Main container που προαναφέρθηκε, παίζει το ρόλο της διαπροσωπείας, με την εκτέλεση των πρακτόρων διαχείρισης και την αντιπροσώπευση ολόκληρης της πλατφόρμας στον εξωτερικό κόσμο. Μια πλήρης πλατφόρμα (AP) πρακτόρων αποτελείται, έπειτα, από διάφορα containers πρακτόρων όπως φαίνεται στο σχήμα που ακολουθεί. Η διανομή των containers σε ένα δίκτυο υπολογιστών επιτρέπεται, υπό τον όρο ότι η επικοινωνία RMI μεταξύ των hosts τους διατηρείται.

Κάθε container πρακτόρων είναι ένα RMI server αντικείμενο, που διαχειρίζεται τοπικά ένα σύνολο πρακτόρων λογισμικού. Ελέγχει τον κύκλο ζωής των πρακτόρων δημιουργώντας τους, αναστέλλοντας την εκτέλεσή τους, επανεκκινώντας και τερματίζοντάς τους. Εκτός αυτού, διαχειρίζεται όλες τις πτυχές της επικοινωνίας, αποστέλλοντας εισερχόμενα ACL μηνύματα, δρομολογώντας τα σύμφωνα με τον προορισμό τους (receiver) και τοποθετώντας τα σε ιδιωτικές ουρές μηνυμάτων των πρακτόρων. Για τα εξερχόμενα μηνύματα, ο container πρακτόρων διατηρεί αρκετές πληροφορίες για να αναζητήσει τη θέση πρακτόρων - δεκτών και να επιλέξει έναν κατάλληλο τρόπο μεταφοράς για να διαβιβάσει το ACL μήνυμα.

Η πλατφόρμα πρακτόρων παρέχει μία γραφική διεπαφή χρήστη (GUI) για την απομακρυσμένη διαχείριση, παρακολούθηση και έλεγχο της κατάστασης των πρακτόρων, που επιτρέπει, παραδείγματος χάριν, το σταμάτημα και την εκκίνηση πρακτόρων. Το GUI επιτρέπει επίσης να δημιουργηθεί και να αρχίσει η εκτέλεση ενός πράκτορα σε έναν απομακρυσμένο host, υπό τον όρο ότι ένα container πρακτόρων τρέχει ήδη. Το ίδιο το GUI έχει υλοποιηθεί ως πράκτορας, αποκαλούμενος RMA (Remote Monitoring Agent).

Όταν εκκινεί μια JADE πλατφόρμα, ο AMS και DF δημιουργούνται αμέσως και το ACC τίθεται ώστε να επιτρέψει την επικοινωνία μηνυμάτων. Ο Main container, ή front-end, είναι ο container πρακτόρων όπου ο AMS και ο DF ζουν και όπου το RMI registry, που χρησιμοποιείται εσωτερικά από τη JADE, δημιουργείται. Τα άλλα containers πρακτόρων συνδέονται με το main container και παρέχουν ένα πλήρες περιβάλλον εκτέλεσης για την εκτέλεση οποιουδήποτε συνόλου πρακτόρων JADE.

Σύμφωνα με τις προδιαγραφές FIPA, ο DF κι ο AMS πράκτορας επικοινωνούν χρησιμοποιώντας την FIPA-SLO γλώσσα περιεχομένου, την οντολογία fipa-agent-management, και το πρωτόκολλο αλληλεπίδρασης FIPA request.

Ο Main Container είναι ένας container μετωπικού άκρου (front-end) και διατηρεί εσωτερικά ένα RMI registry, που χρησιμοποιείται από άλλους containers πρακτόρων στο χρόνο εκκίνησης για να εγγράφονται με το μετωπικό άκρο (front-end), κι έτσι να γίνονται μέλη της πλατφόρμας πρακτόρων. Ο Main Container διατηρεί έναν πίνακα όλων των containers (Agent Container Table) μαζί με την RMI αναφορά αντικειμένου τους. Επίσης, ένας γενικός πίνακας περιγραφών πρακτόρων (Agent Global Descriptor Table) κρατείται, ο οποίος συσχετίζει κάθε όνομα πράκτορα με τα AMS δεδομένα του και με την RMI αναφορά αντικειμένου του container του.

Όταν ένα νέο front-end container αρχίζει να εκτελείται, δημιουργεί ένα εσωτερικό RMI registry στον τρέχοντα host που ακούει σε μία διευκρινισμένη από το χρήστη TCP/IP θύρα. Κατόπιν, εκκινεί τους πράκτορες συστημάτων FIPA (ACC, AMS και DF).

Όταν ένα νέο container αρχίζει να εκτελείται, αναζητά το RMI registry που είναι ενσωματωμένο μέσα στο εκτελούμενο front-end container της πλατφόρμας στην οποία θέλει να συνδεθεί. Έπειτα, ανακτά από αυτό την RMI αναφορά αντικειμένου του front-end της πλατφόρμας. Ο νέος container πρακτόρων καταχωρείται στο front-end και προστίθεται στον Agent Container Table. Κατόπιν, ενημερώνει το front-end του όποτε ένας πράκτορας δημιουργείται ή τερματίζει, προκειμένου να κρατηθεί ο Agent Global Descriptor Table συνεπής.

Προκειμένου να αυξηθεί η απόδοση, κάθε container διατηρεί στη γρήγορη μνήμη την αναφορά αντικειμένου άλλων containers, μόλις στέλνεται ένα μήνυμα σε αυτούς. Έτσι αποφεύγεται η αναζήτηση στον Agent Global Descriptor Table κάθε φορά που πρέπει να σταλεί ένα μήνυμα. Εντούτοις, προκειμένου να αντιμετωπιστούν δυναμικές καταστάσεις, όπου οι containers πρακτόρων εμφανίζονται και εξαφανίζονται, εάν μια απομακρυσμένη μέθοδος προκαλέσει μια εξαίρεση, ο Agent Global Descriptor Table χρησιμοποιείται πάλι.

Το περιβάλλον χρόνου εκτέλεσης της JADE διατηρεί τους κατάλληλους πίνακες πρακτόρων και είναι έτσι ικανός να επιλέξει τον αποδοτικότερο μηχανισμό μηνύματος, σύμφωνα με τη θέση των πρακτόρων-δεκτών.

Η πλατφόρμα παρουσιάζει μια ενιαία διεπαφή στον εξωτερικό κόσμο χρησιμοποιώντας τον τυποποιημένο πράκτορα ACC που προαναφέρθηκε. Ο container πρακτόρων επιτρέπει στη JADE να προσαρμοστεί στην κύρια προϋπόθεση ότι οι μηχανισμοί επικοινωνίας πρέπει να είναι άγνωστοι σε όλους τους πράκτορες. Στην πραγματικότητα, κάθε νήμα πρακτόρων είναι μια απολύτως ανεξάρτητη κλάση, η οποία δεν γνωρίζει τις λεπτομέρειες των μηχανισμών επικοινωνίας και της αρχιτεκτονικής της πλατφόρμας πρακτόρων.

Επικοινωνία JADE πρακτόρων

Ένα από τα σημαντικότερα χαρακτηριστικά γνωρίσματα που παρέχουν οι JADE πράκτορες είναι η δυνατότητα να επικοινωνήσουν. Υιοθετείται η ασύγχρονη διαβίβαση μηνυμάτων. Κάθε πράκτορας έχει μια ουρά μηνυμάτων πρακτόρων όπου κατά τον χρόνο εκτέλεσης εισάγονται τα μηνύματα που στέλνονται από άλλους πράκτορες. Όποτε ένα μήνυμα εισάγεται στην ουρά μηνυμάτων ειδοποιείται ο λαμβάνων πράκτορας. Εάν και πότε παίρνει πραγματικά ο πράκτορας το μήνυμα από την ουρά μηνυμάτων για να το επεξεργαστεί εξαρτάται από τον προγραμματιστή.

Τα μηνύματα που ανταλλάσσονται από τους JADE πράκτορες έχουν μια μορφή που καθορίζεται από τη γλώσσα ACL ορισμένη από τα διεθνή πρότυπα FIPA, για τη διαλειτουργικότητα των πρακτόρων λογισμικού. Αυτή η μορφή περιλαμβάνει διάφορα στοιχεία και ειδικότερα:

- Τον αποστολέα(sender) του μηνύματος
- Τη λίστα των αποδεκτών(receivers)
- Την επικοινωνιακή πρόθεση (επίσης αποκαλούμενη "performative") που προσδιορίζει τι ο αποστολέας σκοπεύει να επιτύχει με την αποστολή του μηνύματος. Το performative μπορεί να είναι REQUEST, εάν ο αποστολέας θέλει ο δέκτης να εκτελέσει μια ενέργεια, INFORM, εάν ο αποστολέας θέλει να πληροφορηθεί το δέκτη για ένα γεγονός, QUERY_IF, εάν ο αποστολέας θέλει να ξέρει εάν μια δεδομένη συνθήκη ισχύει ή όχι, CFP (call for proposal), PROPOSE, ACCEPT_PROPOSAL, REJECT_PROPOSAL, εάν ο αποστολέας και ο δέκτης συμμετέχουν σε μια διαπραγμάτευση, κι άλλα.

Το περιεχόμενο που αποτελεί τις πραγματικές πληροφορίες που περιλαμβάνονται στο μήνυμα (δηλ. η ενέργεια που εκτελείται σε ένα μήνυμα REQUEST, το γεγονός που ο αποστολέας θέλει να αποκαλύψει σε ένα INFORM μήνυμα).

Τη γλώσσα περιεχομένου που ορίζει τη σύνταξη που χρησιμοποιείται για να εκφραστεί το περιεχόμενο (και ο αποστολέας και ο δέκτης πρέπει να είναι σε θέση να κωδικοποιήσουν / αναλύσουν τις εκφράσεις που είναι σύμφωνες με αυτήν την σύνταξη για να είναι αποτελεσματική η επικοινωνία).

Την οντολογία που παρέχει το λεξιλόγιο των συμβόλων που χρησιμοποιούνται στο περιεχόμενο και την έννοιά τους (και ο αποστολέας και ο δέκτης πρέπει να αποδώσουν το ίδιο νόημα στα σύμβολα για να είναι αποτελεσματική η επικοινωνία).

Μερικά πεδία που χρησιμοποιούνται για να ελέγξουν διάφορες ταυτόχρονες συνομιλίες και για να διευκρινίσουν προθεσμίες για τη λήψη μιας απάντησης όπως conversation-id, reply-with, in-reply-to, reply-by.

Όταν ένας JADE πράκτορας στέλνει ένα μήνυμα, οι ακόλουθες διαφορετικές περιπτώσεις είναι δυνατές:

Εάν ο πράκτορας-δέκτης ζει στο ίδιο container πρακτόρων, το Java αντικείμενο που αντιπροσωπεύει το ACL μήνυμα περνά στο δέκτη χρησιμοποιώντας ένα αντικείμενο γεγονός, χωρίς οποιαδήποτε μετάφραση μηνυμάτων.

Εάν ο πράκτορας-δέκτης ζει στην ίδια JADE πλατφόρμα αλλά μέσα σε ένα διαφορετικό container, το ACL μήνυμα στέλνεται χρησιμοποιώντας το Java πλαίσιο Remote Method Invocation. Εκτός από την απόδοση, ο πράκτορας λαμβάνει ένα Java αντικείμενο, ακριβώς όπως το μήνυμα μεταξύ containers.

Εάν ο δέκτης βρίσκεται σε μια διαφορετική πλατφόρμα πρακτόρων, το τυποποιημένο πρωτόκολλο IIOP και η διεπαφή OMG IDL χρησιμοποιούνται, σύμφωνα με τα πρότυπα FIPA. Αυτό περιλαμβάνει τη μετάφραση του αντικειμένου ACL μηνύματος σε string και έπειτα την εκτέλεση μιας απομακρυσμένης κλήσης χρησιμοποιώντας IIOP ως πρωτόκολλο υλικολογισμικού. Από την πλευρά των δεκτών, το IIOP θα χρησιμοποιηθεί, παράγοντας ένα Java string, το οποίο θα αναλυθεί σε ένα αντικείμενο

ACLMessage. Τελικά, το Java αντικείμενο θα αποσταλεί στον πράκτορα-δέκτη (μέσω Java γεγονότων ή κλήσεων RMI).

Γλώσσες Περιεχομένου και Οντολογίες στη JADE

Μέσα στο ACL μήνυμα, οι πληροφορίες αναπαρίστανται ως έκφραση περιεχομένου σύμφωνη με μια κατάλληλη γλώσσα περιεχομένου, π.χ. SL, και κωδικοποιούνται σε μία κατάλληλη μορφή, π.χ. string. Οι πράκτορες έχουν το δικό τους (ενδεχομένως διαφορετικό) τρόπο να αναπαριστούν εσωτερικά τις πληροφορίες αυτές. Ο τρόπος αυτός πρέπει να επιτρέπει τον εύκολο χειρισμό των πληροφοριών, οπότε και είναι αρκετά σαφές ότι η αναπαράσταση που χρησιμοποιείται σε μια ACL έκφραση περιεχομένου δεν είναι κατάλληλη για το εσωτερικό ενός πράκτορα. Επίσης, η αποθήκευση αυτών των πληροφοριών μέσα σε έναν πράκτορα απλά ως μεταβλητή string δεν είναι κατάλληλη για τον χειρισμό των πληροφοριών. Στους πράκτορες λογισμικού που γράφονται σε Java (όπως οι πράκτορες JADE), οι πληροφορίες μπορούν βολικά να αναπαρασταθούν μέσα σε έναν πράκτορα ως Java αντικείμενα.

Η JADE υποστηρίζει τις γλώσσες περιεχομένου και τις οντολογίες προκειμένου οι μετατροπές από τη μία αναπαράσταση στην άλλη και οι λειτουργίες ελέγχου των πληροφοριών να εκτελούνται αυτόματα, επιτρέποντας στους υπεύθυνους ανάπτυξης εφαρμογών το χειρισμό των πληροφοριών μέσα στους πράκτορες ως Java αντικείμενα. Πιο συγκεκριμένα, η οντολογία επικυρώνει σημασιολογικά τις πληροφορίες, ενώ ο codec (codec για μια γλώσσα περιεχομένου είναι ένα Java αντικείμενο ικανό να διαχειριστεί εκφράσεις περιεχομένου γραμμένες στην γλώσσα αυτή) εκτελεί τη μετάφραση σε strings (ή ακολουθίες bytes) σύμφωνα με τους συντακτικούς κανόνες της σχετικής γλώσσας περιεχομένου. Ο χρήστης δεν χρειάζεται να φροντίσει για αυτές τις διαδικασίες, εκτός αν πρέπει να χρησιμοποιήσει μερικά προηγμένα χαρακτηριστικά, όπως να θέσει ερωτήσεις.

Για να εκτελέσει η JADE τους κατάλληλους σημασιολογικούς ελέγχους σε μια δεδομένη έκφραση περιεχομένου είναι απαραίτητο να ταξινομηθούν, σύμφωνα με τα γενικά σημασιολογικά χαρακτηριστικά τους, όλα τα πιθανά στοιχεία που μπορούν να εμφανιστούν μέσα σε μια έγκυρη πρόταση που στέλνεται από έναν πράκτορα σε έναν άλλον, ως περιεχόμενο ενός ACL μηνύματος. Λεπτομερέστερα στο πρώτο επίπεδο κάνουμε τη διάκριση μεταξύ των predicates (κατηγορημάτων) και των terms (όρων).

Predicates: Είναι εκφράσεις που λένε κάτι για την κατάσταση του κόσμου και μπορούν να είναι true ή false.

Terms: Είναι εκφράσεις που προσδιορίζουν τις οντότητες (αφηρημένες ή συγκεκριμένες) που “υπάρχουν” στον κόσμο και για τις οποίες οι πράκτορες μιλούν και εξάγουν συλλογισμούς. Είναι περαιτέρω ταξινομημένοι σε:

Concepts: Εκφράσεις που υποδεικνύουν οντότητες με μια σύνθετη δομή που μπορεί να καθορισθεί σε σχέση με τα slots. Τα concepts (έννοιες) συνήθως δεν έχουν κανένα νόημα εάν χρησιμοποιούνται άμεσα ως περιεχόμενο ενός ACL μηνύματος. Γενικά αναφέρονται μέσα σε κατηγορήματα και σε άλλα concepts.

Agent actions: Ειδικές έννοιες που υποδεικνύουν ενέργειες που μπορούν να εκτελεσθούν από κάποιους agents. Οι επικοινωνιακές πράξεις (δηλ. μηνύματα ACL) είναι κι αυτές ενέργειες πρακτόρων.

Primitives: Εκφράσεις που υποδεικνύουν ατομικές οντότητες όπως strings και ακέραιοι αριθμοί

Aggregates: Εκφράσεις που υποδεικνύουν οντότητες που είναι ομάδες άλλων οντοτήτων

Identifying Referential Expressions (IRE): Εκφράσεις που προσδιορίζουν την οντότητα (ή τις οντότητες) για την οποία ισχύει ένα δεδομένο κατηγορήμα

Variables: Εκφράσεις (που χρησιμοποιούνται συνήθως στις ερωτήσεις) που υποδεικνύουν ένα γενικό στοιχείο άγνωστο αρχικά.

Μια πλήρως εκφραστική γλώσσα περιεχομένου πρέπει να είναι σε θέση να αναπαραστήσει και να διακρίνει τα στοιχεία όλων των ανωτέρω τύπων. Η JADE προσφέρει codecs για δύο γλώσσες περιεχομένου (την SL και την LEAP γλώσσα).

Η γλώσσα SL είναι μια γλώσσα περιεχομένου κατανοήσιμη από τον άνθρωπο, κωδικοποιημένη σε string. Είναι πιθανώς η περισσότερο διαδεδομένη γλώσσα περιεχομένου στην επιστημονική κοινότητα που ασχολείται με τους ευφυείς πράκτορες.

Η γλώσσα LEAP είναι μια γλώσσα περιεχομένου μη κατανοήσιμη από τον άνθρωπο, κωδικοποιημένη σε bytes που έχει οριστεί ως πρόσθετο της JADE μέσα στο πρόγραμμα LEAP. Είναι επομένως σαφές ότι μόνο οι JADE πράκτορες θα είναι σε θέση να επικοινωνήσουν με τη γλώσσα LEAP. Όταν υπάρχουν ισχυροί περιορισμοί μνήμης η γλώσσα αυτή είναι προτιμητέα.

Μια οντολογία για μια δεδομένη θεματική περιοχή είναι ένα σύνολο σχημάτων που καθορίζουν τη δομή των κατηγορημάτων, των ενεργειών πρακτόρων και των εννοιών (βασικά τα ονόματά τους και τα slots τους) που σχετίζονται με εκείνη την περιοχή.

Κινητοί Πράκτορες στη JADE

Χρησιμοποιώντας τη JADE, οι υπεύθυνοι για την ανάπτυξη εφαρμογών μπορούν να δημιουργήσουν κινητούς πράκτορες, οι οποίοι είναι σε θέση να μεταναστεύσουν ή να δημιουργήσουν αντίγραφο τους σε πολλαπλούς hosts δικτύων. Στην τρέχουσα έκδοση της JADE, μόνο ενδοπλατφορμική κινητικότητα υποστηρίζεται, που σημαίνει ότι ένας JADE κινητός πράκτορας μπορεί να μετακινηθεί σε διαφορετικά containers πρακτόρων αλλά είναι περιορισμένος σε μια ενιαία JADE πλατφόρμα.

Οι κινητοί πράκτορες πρέπει να είναι ενήμεροι θέσεων προκειμένου να αποφασίσουν πότε και πού να κινηθούν. Επομένως, η JADE παρέχει μια οντολογία, που ονομάζεται jade-mobility-ontology, που περιέχει απαραίτητες έννοιες και ενέργειες.

Η κίνηση ενός πράκτορα περιλαμβάνει την αποστολή του κώδικα και της κατάστασής του μέσω ενός καναλιού δικτύου. Έτσι οι καθορισμένοι από το χρήστη κινητοί πράκτορες πρέπει να διαχειριστούν τη serialization και unserialization διαδικασία. Μερικοί από τους πόρους που χρησιμοποιούνται από τον κινητό πράκτορα θα κινηθούν, ενώ μερικοί άλλοι θα αποσυνδεθούν πριν τη μετακίνηση και θα επανασυνδεθούν στον προορισμό [10].

4.6 - Εμπορικές εφαρμογές

Διαχείριση πληροφοριών

Η αύξηση του όγκου και της ποικιλίας της πληροφορίας την σημερινή εποχή μας έχει οδηγήσει στην ανάγκη να μπορούμε να διαχειριστούμε αυτές τις πληροφορίες. Η έλλειψη αποτελεσματικών εργαλείων διαχείρισης της πληροφορίας έχει οδηγήσει στο πρόβλημα της υπερφόρτωσης της πληροφορίας. Ο μεγάλος όγκος πληροφοριών που προέρχεται από το internet και από το world wide web αποτελεί ένα σημαντικό πρόβλημα.

Το δυναμικό αυτής της πηγής είναι άμεσα εμφανής σε οποιονδήποτε είναι γνώστης του WWW. Το θέμα είναι πως η πραγματικότητα είναι συχνά απογοητευτική και υπάρχουν πολλοί λόγοι για αυτό. Τόσο οι ανθρώπινοι παράγοντες όσο και οι οργανωτικοί συνωμοτούν εναντίον των χρηστών που

προσπαθούν να χρησιμοποιήσουν τους πόρους με συστηματικό τρόπο. Μπορούμε να χαρακτηρίσουμε το πρόβλημα της υπερφόρτωσης των πληροφοριών με δύο τρόπους:

- Πληροφορίες φιλτραρίσματος: Κάθε μέρα παρουσιάζονται μεγάλος όγκος πληροφοριών όπως email κ.λ.π., μόνο ένα ποσοστό εκ των οποίων είναι σχετικό ή σημαντικό. Πρέπει να μπορούμε να ξεχωρίζουμε τις πληροφορίες και να επιλέγουμε αυτές που είναι σημαντικές για εμάς.
- Συλλογή πληροφοριών: Ο όγκος των πληροφοριών μας βοηθά να βρούμε τις πληροφορίες που χρειαζόμαστε για να απαντήσουμε σε συγκεκριμένες ερωτήσεις. Πρέπει να είμαστε ικανοί να βρίσκουμε πληροφορίες που ικανοποιούν τις απαιτήσεις μας, ακόμα και αν αυτές μπορούν να συλλέγονται από διαφορετικά site.

Ένας σημαντικός παράγοντας που οδηγεί στην υπερφόρτωση των πληροφοριών είναι το γεγονός πως ο τελικός χρήστης πρέπει να κατευθύνει την διαδικασία διαχείρισης. Αλλά δεν υπάρχει λόγος για τον οποίον οι έρευνες δεν πρέπει να διενεργούνται από πράκτορες, οι οποίοι θα ενεργούν αυτόνομα για εύρεση πληροφοριών στο WEB για λογαριασμό των χρηστών. Η ιδέα είναι τόσο αναγκαστική που πολλά έργα λειτουργούν είδη έτσι. Παρακάτω περιγράφονται τρία από αυτά [22]:

- Maxims(1994): Περιγράφει την δημιουργία ενός πράκτορα ο οποίος μπορεί και φιλτράρει ένα ηλεκτρονικό ταχυδρομείο. Το πρόγραμμα μπορεί να θέτει προτεραιότητες, να διαγράφει, προωθεί, και να ταξινομεί τα μηνύματα σε αρχείο για λογαριασμό του χρήστη. Λειτουργεί σαν βοηθός του χρήστη και χρησιμοποιεί ενέργειες του χρήστη σαν μάθημα. Το πρόγραμμα κάνει εσωτερικές προβλέψεις για το τι μπορεί να κάνει ένας χρήστης μ' ένα μήνυμα. Εάν αυτές οι προβλέψεις βγουν ανακριβείς τότε πρόγραμμα τις κρατά για το ίδιο. Εάν όμως βγουν επιτυχείς τότε αρχίζουν να κάνουν προτάσεις στο χρήστη.
- Newt(1994): Το συγκεκριμένο πρόγραμμα περιγράφει το φιλτράρισμα των νέων του internet. Το συγκεκριμένο πρόγραμμα υλοποιείται σε C++ πάνω σε πλατφόρμα UNIX. Παίρνει σαν είσοδο νέα και σαν έξοδο δίνει τις πληροφορίες που συνίσταται να διαβάσει ο χρήστης. Ο πράκτορας agent είναι προγραμματισμένος μέσω εκπαιδευτικών παραδειγμάτων: Ο χρήστης δίνει στον πράκτορα κάποια παραδείγματα άρθρων που πρέπει να διαβαστούν και κάποια που δεν πρέπει, και αυτή η ανατροφοδοτήσει χρησιμοποιείται για να τροποποιήσει το μοντέλο που θα ενδιαφέρει το χρήστη. Ο πράκτορας κάνει ανάλυση του κειμένου και επιλέγει τις λέξεις τις οποίες ενδιαφέρουν το χρήστη.
- Η ψηφιακή βιβλιοθήκη Zuno: Μια ψηφιακή πληροφορία είναι μια οργανωμένη και διαχειριζόμενη συλλογή δεδομένων, σε συνδυασμό με υπηρεσίες οι οποίες θα βοηθήσουν το χρήστη να μπορέσει να διαχειριστεί τα συγκεκριμένα δεδομένα. Το σύστημα της συγκεκριμένης βιβλιοθήκης είναι ένα σύστημα πολυπρακτόρων το οποίο επιτρέπει στο χρήστη να έχει μια μοναδική άποψη για τα αναδιοργανωτά δεδομένα όπως το WWW, τα ίδια τα δεδομένα του χρήστη, συλλογές από άρθρα κ.λ.π. Οι πράκτορες στο συγκεκριμένο σύστημα μπορούν να παίζουν τρεις ρόλους:
 - Των καταναλωτών: που αντιπροσωπεύει τους τελικούς χρήστες του συστήματος, οι οποίοι μπορούν να θεωρηθούν πως καταναλώνουν τις πληροφορίες.
 - Των παραγωγών: Αυτών στην ουσία που παρέχουν τις πληροφορίες
 - Της διευκόλυνσης: Αφορά την χαρτογράφηση μεταξύ καταναλωτών και παραγωγών.

Οι πράκτορες των καταναλωτών στο συγκεκριμένο σύστημα είναι υπεύθυνοι για να εκπροσωπούν τα συμφέροντα του χρήστη. Διατηρούν τα μοντέλα των χρηστών, και τα χρησιμοποιούν για να τους βοηθούν, παρέχοντας πληροφορίες που χρειάζονται και τους προστατεύουν από πληροφορίες που δεν χρειάζονται, τέλος μπορεί να λειτουργεί σαν φίλτρο ή συλλέκτη πληροφοριών.

Semantic Web

Το Semantic Web είναι ένα πρόγραμμα που σκοπεύει να δημιουργήσει ένα καθολικό μέσο για την ανταλλαγή πληροφοριών δίνοντας νόημα (σημασιολογία) στο περιεχόμενο των εγγράφων στο Web, κατά τρόπο κατανοητό από τους υπολογιστές. Υπό την καθοδήγηση του δημιουργού του Web, Tim Berners-Lee του World Wide Web Consortium (W3C), το Semantic Web επεκτείνει τις δυνατότητες του World Wide Web μέσω της χρήσης προτύπων, γλωσσών επισήμανσης (mark-up) και σχετικών εργαλείων επεξεργασίας [36].

Ο σημασιολογικός Ιστός παρέχει ένα κοινό πλαίσιο που επιτρέπει στα δεδομένα να διαμοιράζονται και να επαναχρησιμοποιούνται πέρα από τα όρια μιας εφαρμογής, μιας επιχείρησης ή μίας κοινότητας. Είναι μια συλλογική προσπάθεια της οποίας ηγείται το W3C με τη συμμετοχή ενός μεγάλου αριθμού ερευνητών και βιομηχανικών συνεργατών. Είναι βασισμένο στο Resource Description Framework (RDF), το οποίο ενσωματώνει ποικίλες εφαρμογές χρησιμοποιώντας XML για τη σύνταξη και URIs για την ονοματοδοσία [36].

Το 2001 ο Tim Berners-Lee έγραψε:

"Το Semantic Web είναι μια επέκταση του τρέχοντος Web, στον οποίο δίνεται ένα καθορισμένο με σαφήνεια νόημα στις πληροφορίες, δίνοντας τη δυνατότητα σε υπολογιστές και ανθρώπους να συνεργάζονται καλύτερα." [36]

Το μεγαλύτερο μέρος του περιεχομένου του Ιστού σήμερα σχεδιάζεται για να διαβαστεί από ανθρώπους, όχι για να το χειριστούν εννοιολογικά τα προγράμματα υπολογιστών. Βασίζεται κυρίως σε έγγραφα γλώσσας HTML η οποία χρησιμεύει στην περιγραφή ενός δομημένου κειμένου, με έμφαση στην οπτική απεικόνιση. Η HTML έχει περιορισμένες ικανότητες όσον αφορά την ταξινόμηση των τμημάτων κειμένου μιας σελίδας. Οι υπολογιστές μπορούν αποδοτικά να αναλύσουν ιστοσελίδες ως προς την εμφάνιση και τις ρουτίνες, αλλά γενικά, οι υπολογιστές δεν έχουν συνήθως κάποιον αξιόπιστο τρόπο να επεξεργαστούν τη σημασιολογία (semantics) που περιέχεται στη σελίδα αυτή.

Για παράδειγμα, οι μηχανές αναζήτησης εκτελούν απλές λεξιλογικές αναλύσεις που βασίζονται σε συμβολοσειρές αναζήτησης και παράγουν λίστες αποτελεσμάτων που απαιτούν την εξαγωγή συμπερασμάτων από τους ανθρώπινους χρήστες, όσον αφορά το ποια από τα δεδομένα είναι τα κατάλληλα για την θεματική περιοχή που ερευνάται. Χρειάζεται δηλαδή η λογική ανάλυση εκ μέρους του χρήστη για να βρεθούν τα συναφή συμπεράσματα. Αυτό συμβαίνει διότι οι ιστοσελίδες δεν περιέχουν αρκετή πληροφορία σχετικά με τη σημασιολογία των περιεχομένων τους οπότε το λογισμικό του χρήστη δεν μπορεί να βγάλει συμπεράσματα από το περιεχόμενο της σελίδας. Για να αντιμετωπιστεί αυτό το πρόβλημα, οι τεχνολογίες του Semantic Web δίνουν τη δυνατότητα σε αυτοματοποιημένα εργαλεία να εξάγουν αυτά τα συμπεράσματα.

Το Semantic Web επιχειρεί να δομήσει το σημασιολογικό περιεχόμενο των ιστοσελίδων, δημιουργώντας ένα περιβάλλον όπου οι πράκτορες λογισμικού που περιπλανώνται από σελίδα σε σελίδα μπορούν εύκολα να εκτελέσουν τους περίπλοκους στόχους των χρηστών τους. Έτσι, αποζητείται ο μετασχηματισμός του Web από ένα σύνολο στατικών σελίδων σε ένα δίκτυο δυναμικών παροχών υπηρεσιών (Web services) που ανακαλύπτουν αυτόματα τις αναζητούμενες πληροφορίες, διαπραγματεύονται για αγαθά που ο χρήστης σκοπεύει να αγοράσει ή συγκεντρώνουν πληροφορίες από διαφορετικές πηγές και τις συνενώνουν σε ομοιογενείς μορφές [29].

Για να λειτουργήσει το Semantic Web, πρέπει οι υπολογιστές να έχουν πρόσβαση σε δομημένες συλλογές πληροφοριών και σύνολα λογικών κανόνων που μπορούν να χρησιμοποιήσουν για να διεξάγουν την αυτοματοποιημένη εξαγωγή συμπερασμάτων. Οι ερευνητές της Τεχνητής Νοημοσύνης έχουν μελετήσει τέτοια συστήματα πολλά χρόνια πριν. Η Αναπαράσταση Γνώσης (Knowledge Representation), όπως καλείται αυτή η τεχνολογία, δεν έχει ακόμα χρησιμοποιηθεί επαρκώς ώστε να επιδείξει πλήρως τις δυνατότητές της. Για να γίνει αυτό πρέπει να υλοποιηθεί σε ένα μοναδικό παγκόσμιο σύστημα [36].

Οι τεχνολογίες που χρησιμοποιεί το Semantic Web είναι οι εξής [36]:

- τα URIs (Universal Resource Identifier): συμβολοσειρές που ταυτοποιούν μοναδικά μία οντότητα (ένα Web site, μία ιδιότητα, έναν άνθρωπο, ένα πράγμα κλπ)
- γλώσσα επιστημών XML: επιτρέπει στους χρήστες να προσθέτουν αυθαίρετη δομή στα έγγραφα τους, χωρίς να καθορίζει την σημασιολογία αυτής της δομής
- τεχνολογία RDF: χρησιμοποιείται για την αναπαράσταση δεδομένων και ανταλλαγή γνώσης στο διαδίκτυο
- τεχνολογία OWL: χρησιμοποιείται για τη δημιουργία και διανομή οντολογιών, υποστηρίζοντας προχωρημένη αναζήτηση στο διαδίκτυο, πράκτορες λογισμικού και διαχείριση γνώσης.

Κάθε μία από τις παραπάνω τεχνολογίες βασίζεται σε εκείνες που αναφέρονται πριν από αυτή. Έτσι π.χ. η RDF βασίζεται στην XML και χρησιμοποιεί URIs. Οι τεχνολογίες αυτές συνδυάζονται ώστε να παρέχουν περιγραφές που συμπληρώνουν ή υποκαθιστούν το περιεχόμενο εγγράφων στο Web, όπως περιγράφεται παρακάτω. Αυτές οι machine-readable περιγραφές επιτρέπουν την προσθήκη νοήματος στο περιεχόμενο, διευκολύνοντας έτσι την αυτόματη αναζήτηση πληροφοριών από υπολογιστές.

Το νόημα του περιεχομένου εκφράζεται μέσω της RDF, η οποία κωδικοποιεί σύνολα από triples (τριάδες) που αντιπροσωπεύουν περίπου το υποκείμενο, το ρήμα και το αντικείμενο μιας πρότασης. Αυτές οι τριάδες μπορούν να γραφούν με XML. Ένα έγγραφο της RDF δηλώνει ότι κάποιες οντότητες (ιστοσελίδες, άνθρωποι ή πράγματα) έχουν κάποιες ιδιότητες με συγκεκριμένες τιμές (π.χ. ο X είναι συγγραφέας του βιβλίου Y). Αυτός ο τρόπος αναπαράστασης μπορεί να εκφράσει τα περισσότερα δεδομένα που κατανοούν οι υπολογιστές. Το υποκείμενο, το ρήμα και το αντικείμενο προσδιορίζονται από ένα URI (Universal Resource Identifier), όπως ακριβώς γίνεται με τις ιστοσελίδες. Έτσι μπορεί ο καθένας να προσθέσει μία νέα οντότητα ή μία νέα ιδιότητα.

Ένα βασικό συστατικό του Semantic Web είναι οι οντολογίες. Οι οντολογίες ορίζουν με τυπικό τρόπο τις έννοιες και τις συσχετίσεις των εννοιών για κάποιο πεδίο. Περιέχουν ορισμούς κλάσεων αντικειμένων και σχέσεις ανάμεσα στις κλάσεις, καθώς και κανόνες εξαγωγής λογικών συμπερασμάτων. Μέσω των οντολογιών γίνεται δυνατόν να συνδυαστούν δεδομένα από διαφορετικές πηγές, οι οποίες μοιράζονται την ίδια οντολογία. Ακόμη, λύνονται προβλήματα ορολογίας καθώς η σημασία των όρων που εμφανίζονται σε μία σελίδα μπορεί να οριστεί με δείκτες προς την οντολογία. Οι οντολογίες μπορούν να βελτιώσουν την λειτουργία του Web αυξάνοντας την ακρίβεια των αναζητήσεων πληροφοριών, αφού αναζητούνται οι πληροφορίες που αναφέρονται μόνο σε μία συγκεκριμένη έννοια και όχι σε κάποια λέξη κλειδί. Μπορούν επίσης να χρησιμοποιηθούν για την συσχέτιση των πληροφοριών ενός site με αντίστοιχες δομές γνώσης και λογικούς κανόνες. Οι κοινές οντολογίες βοηθούν στην ανταλλαγή δεδομένων και νοημάτων μεταξύ διαφορετικών web-based υπηρεσιών.

Υπάρχουν πολλές αυτοματοποιημένες υπηρεσίες βασισμένες στο διαδίκτυο (web services) που δεν χρησιμοποιούν τη σημασιολογία, όμως άλλα προγράμματα (όπως είναι οι πράκτορες λογισμικού) δεν μπορούν να εντοπίσουν μόνα τους μία τέτοια υπηρεσία για κάποια συγκεκριμένη λειτουργία. Αυτό μπορεί να συμβεί μόνο όταν υπάρχει μία κοινή γλώσσα περιγραφής παρεχόμενων υπηρεσιών σε ένα δίκτυο ούτως ώστε οι διάφοροι πράκτορες να διαφημίζουν τις υπηρεσίες τους σε μία υπηρεσία κίτρινων σελίδων.

Αφού εντοπιστεί ο επιθυμητός πράκτορας, οι δύο επικοινωνούντες πράκτορες λογισμικού μπορούν να κατανοήσουν ο ένας τον άλλο ανταλλάσσοντας οντολογίες. Το Semantic Web παρέχει αυτή την ευελιξία. Ανακαλύπτοντας νέες οντολογίες οι πράκτορες λογισμικού αποκτούν νέες δυνατότητες εξαγωγής λογικών συμπερασμάτων.

Η ενοποιητική λογική γλώσσα του Semantic Web δίνει τη δυνατότητα να ενωθούν οι έννοιες που μπορεί να ορίσει ο καθένας (μέσω ενός URI) σε ένα καθολικό Web. Έτσι οι πράκτορες λογισμικού θα μπορούν να αναλύουν σημασιολογικά τη γνώση των ανθρώπων παρέχοντας μία νέα μορφή χρήσιμων εργαλείων.

4.7 - Ηλεκτρονικό εμπόριο

Σήμερα, το εμπόριο κινείται με ανθρώπινες επιδράσεις μέχρι τώρα, οι άνθρωποι αποφασίζουν για το πότε πρέπει να αγοράζουν τα εμπορεύματα, πόσο θέλουν να πληρώσουν κ.λ.π. Καταρχήν, δεν υπάρχει λόγος για τον οποίον δεν αυτοματοποιείται το εμπόριο. Αυτό σημαίνει πως η λήψη κάποιων εμπορικών αποφάσεων μπορούν να αφηθούν στα χέρια πρακτόρων. Ο αναγνώστης υποθέτει πως είναι πρωτότυπο, και ότι ένας μη λογικός εμπορικός οργανισμός μπορεί να θέσει τις αποφάσεις τις οποίες παίρνονται ως βασική ευθύνη του προγράμματος, και πρέπει να θυμόμαστε πως αυτό είναι ακριβώς που γίνεται στις σημερινές ηλεκτρονικές συναλλαγές. Το ευρέως ηλεκτρονικό εμπόριο θα έχει απόσταση από αυτό που θα είναι στο μέλλον. Βραχυπρόθεσμα οι εφαρμογές ηλεκτρονικών συναλλαγών είναι πιθανόν να είναι πιο μικρής κλίμακας.

4.8 - Διαδικασία διοίκησης επιχειρήσεων

Οι manager των επιχειρήσεων παίρνουν ενημερωμένες αποφάσεις βασισμένες στον συνδυασμό της κρίσης και πληροφορίες που έρχονται από άλλα τμήματα. Ιδανικά όλες οι σχετικές πληροφορίες πρέπει να συγκεντρωθούν πριν τους ασκηθεί κρίση. Παρόλα αυτά οι συγκέντρωση των πληροφοριών ειδικά σε μια μεγάλη επιχείρηση είναι μια αρκετά χρονοβόρα διαδικασία. Για αυτό το λόγο, οι επιχειρήσεις έχουν προσπαθήσει να αναπτύξει ένα σύστημα έξυπνων συστημάτων για να βοηθηθούν για την διαχείριση πολλών διαδικασιών.

Το έργο ADEPT αντιμετωπίζει αυτό το πρόβλημα με την προβολή μια επιχειρηματικής διαδικασίας ως ενότητα διαπραγμάτευσης, και πρακτόρων παροχής υπηρεσιών. Κάθε πράκτορας αντιπροσωπεύει ένα συγκεκριμένο ρόλο ή ένα συγκεκριμένο κομμάτι της επιχείρησης και είναι ικανό να παρέχει μια η παραπάνω υπηρεσίες. Για παράδειγμα ο σχεδιασμός ενός τμήματος μπορεί να παρέχει την υπηρεσία του σχεδιασμού ενός δικτύου τηλεπικοινωνιών, ένα νομικό τμήμα μπορεί να παρέχει την υπηρεσία του ελέγχου εάν ο σχεδιασμός είναι νομικός, τέλος το τμήμα marketing την υπηρεσία ελέγχου κόστους.

Οι πράκτορες οι οποίοι απαιτούν μια υπηρεσία από έναν άλλο πράκτορα, ο οποίος μπαίνει στη διαδικασία διαπραγμάτευσης για αυτή την διαδικασία έτσι ώστε να μπορεί να επιτύχει μια αμοιβαία αποδεκτή τιμή, χρόνο και βαθμό ποιότητας. Οι επιτυχείς διαπραγματεύσεις οδηγούν σε δεσμευτικές συμφωνίες μεταξύ πρακτόρων. Αυτή η βασισμένη σε πράκτορες προσέγγιση προσφέρει πολλά πλεονεκτήματα σε σχέση με τυπικές λύσεις για το συγκεκριμένο πρόβλημα.

Η δυναμική φύση των πρακτόρων σημαίνει πως οι υπηρεσίες μπορούν να προγραμματιστούν εκείνη την στιγμή και όχι να είναι προκαθορισμένες, επίσης οι εξαιρέσεις της υπηρεσίας μπορούν να ανιχνευτούν και να διαχειριστούν με πολύ ευέλικτο τρόπο. Το συγκεκριμένο σύστημα δοκιμαστικό σε μια μεγάλη αγγλική εταιρία τηλεπικοινωνιών η οποία περιείχε διακόσιες δραστηριότητες και εννέα τμήματα και υπήρχαν σχέδια για να εφαρμοστεί σε πλήρη κλίμακα.

4.9 - Ιατρικές εφαρμογές

Οι πληροφορική στην ιατρική είναι μια σημαντική ανάπτυξη στην επιστήμη των υπολογιστών: πολλές εφαρμογές βρίσκονται κάθε μέρα στον τομέα της υγείας. Ως εκ τούτου δεν είναι περίεργο ότι οι πράκτορες θα πρέπει να εφαρμόζονται σε αυτό το τομέα. Δύο από τις πρώτες εφαρμογές που εφαρμόστηκαν είναι στον τομέα της υγειονομικής περίθαλψης και της παρακολούθησης ασθενών.

Παρακολούθηση ασθενών

Το σύστημα Guardian το οποίο περιγράφηκε το 1989 έχει ως στόχο να βοηθήσει στην διαχείριση ασθενών στην χειρουργική εντατική μονάδα θεραπείας. Το σύστημα υποκινείται από δύο ακριβώς πράγματα [41]:

- Το μοντέλο περίθαλψης ασθενών είναι σημαντικό να δουλεύει σαν ομάδα, στην οποία υπάρχει μια συλλογή από εμπειρογνώμονες με διαφορετικούς τομείς δράσης οι οποίοι συνεργάζονται για να οργανώσουν το συγκεκριμένο μοντέλο.
- Ένας από τους πιο σημαντικούς παράγοντες είναι η κατάλληλη κατανομή πληροφοριών μεταξύ των μελών της ομάδας.

Ειδικότερα, οι ειδικοί τείνουν να έχουν λιγότερες πιθανότητες για να παρακολουθούν λεπτό προς λεπτό την κατάσταση ενός ασθενή. Αυτό το έργο τείνει να πέσει στις πλάτες των νοσηλευτών, οι οποίοι τις περισσότερες φορές δεν έχουν την τεχνογνωσία να ερμηνεύουν της πληροφορίες που παίρνουν, όπως θα έκανε ένας ειδικός πχ. Το συγκεκριμένο σύστημα διανέμει την παρακολούθηση ασθενών μεταξύ τριών τύπων πρακτόρων:

- Πράκτορες που δρουν και αντιλαμβάνονται. Είναι υπεύθυνη για την διεπαφή μεταξύ του συστήματος Guardian και του υπόλοιπου κόσμου, και χαρτογραφούν μια σειρά από αισθητήρες εισόδου σε μια χρήσιμη συμβολική μορφή, και μεταφράζουν αιτήματα δράσης από το Guardian σε μια σειρά εντολών ελέγχου.
- Αιτιολογικοί πράκτορες. Υπεύθυνοι για την οργάνωση του συστήματος λήψης αποφάσεων.
- Πράκτορες ελέγχου, από τους οποίους θα υπάρχει μόνο ένας ο οποίος θα βρίσκεται στην ανώτερη θέση ελέγχου του συστήματος.

Οι συγκεκριμένοι πράκτορες οργανώνονται ιεραρχικά και συνολικά το σύστημα σε στενά βασισμένα συστήματα ελέγχου, όπου διάφοροι παράγοντες συνεργάζονται σαν μια κοινή δομή δεδομένων.

Ιατρική φροντίδα

Ο Huang το 1996 παρουσιάζει ένα πρωτότυπο και βασισμένο σε πράκτορες διανεμημένο σύστημα ιατρική περίθαλψης. Το σύστημα είναι σχεδιασμένο ώστε να ενσωματώνει την διαδικασία διαχείριση του ασθενή, το οποίο περιέχει πληροφορίες από πολλά άτομα. Για παράδειγμα ένας γενικός γιατρός μπορεί να υποθέσει πώς ένας ασθενής έχει καρκίνο του μαστού, αλλά αυτό δεν μπορεί ούτε να επιβεβαιωθεί ούτε να απορριφτεί χωρίς την συμβουλή ενός ειδικού. Όταν το επιβεβαιώσει ο ειδικός τότε πρέπει να βρεθεί πρόγραμμα το οποίο θα θεραπεύσει τον ασθενή, το οποίο θα αφορά πηγές από άλλα άτομα.[58]

Το πρωτότυπο πρόγραμμα επιτρέπει μια φυσική αναπαράσταση της διαδικασίας, με τους πράκτορες να αντιστοιχούν σε άτομα και τους οργανισμούς να συμμετέχουν στη διαδικασία φροντίδας του ασθενή. Οι πράκτορες στο πρωτότυπο περιέχουν ένα έξυπνο σύστημα, το οποίο περιέχει τον τομέα της ειδικότητας του πράκτορα, την διεπαφή ανθρώπου υπολογιστή, επιτρέποντας το χρήστη να προσθέτει να αφαιρεί και να βλέπει του στόχους του συστήματος, την διαχείριση επικοινωνιών, η οποία αναγνωρίζει το μήνυμα που περνά για την λειτουργικότητα των πρακτόρων. Το στοιχείο του έξυπνου συστήματος βασίζεται στην ειδικότητα του μοντέλου KAD και ολόκληρη η αρχιτεκτονική του πράκτορα υλοποιείται σε prolog. Το μήνυμα το οποίο περνάει αναγνωρίζεται μέσω της επέκτασης του ηλεκτρονικού ταχυδρομείου.

Η χρήση των ευφυών πρακτόρων στη διάγνωση καρδιακών διαταραχών

Το βασικό παράδειγμα της προσέγγισης των ευφυών πρακτόρων είναι η χρήση πρωτόκολλων επικοινωνίας που επιτρέπουν σε διαφορετικά έξυπνα συστατικά να συνεργάζονται. Οι ευφυείς πράκτορες έχουν τις ρίζες τους στην διανεμημένη τεχνητή νοημοσύνη και έχουν χρησιμοποιηθεί σε διάφορες εφαρμογές. Κάθε πράκτορας είναι μια ανεξάρτητη μεθοδολογία με ικανότητες συλλογισμού, και οι οποίοι δουλεύουν για ένα προκαθορισμένο έργο. Ο στόχος του συστήματος είναι να δημιουργήσει ένα συνεργατικό περιβάλλον στο οποίο δύο ή περισσότεροι πράκτορες μπορούν να συνδυαστούν για να λύσουν προβλήματα με τη χρήση ενός διαμεσολαβητή.

Η χρήση των ευφυών πρακτόρων σε βιομετρικά συστήματα ήταν περιορισμένη, με μεγάλη προσοχή στα αποτελέσματα της υγειονομικής περίθαλψης, αν και ορισμένα πολύπλοκα ιατρικά προβλήματα έχουν αντιμετωπιστεί. Η προκύπτουσα δομή είναι μια επέκταση της υβριδικής προσέγγισης, για την ανάπτυξη των ολοκληρωμένων ιατρικών λήψεων αποφάσεων με τη χρήση υπολογιστή, τα οποία μπορούν να χρησιμοποιούν διαφορετικές μεθοδολογίες.

Το σύστημα στην συγκεκριμένη εφαρμογή περιλαμβάνει πέντε πράκτορες εκτός από τον γιατρό ο οποίος αποτελεί τον αρχικό πράκτορα.

Ο πρώτος πράκτορας είναι ένα σύστημα βασισμένο στη γνώση και το οποίο αναπτύσσεται από τους δημιουργούς του για την ανάλυση του πόνου του στήθους. Χρησιμοποιεί έναν τροποποιημένο κανόνα παραγωγής, ο οποίος ενσωματώνει προσεγγιστικές και συλλογιστικές τεχνικές με μερική τεκμηρίωση και στάθμιση των προηγούμενων. Η αρχική βάση του emerge επικεντρώθηκε στην αξιολόγηση του πόνου του στήθους στα επείγοντα περιστατικά και εκ τότε χρησιμοποιείται και σε άλλες εφαρμογές. Νέες μέθοδοι οι οποίες έχουν ευθυγραμμιστεί με αποδείξεις βασισμένες σε ιατρικές τεχνικές, χειρίζονται από τον δεύτερο πράκτορα και χρησιμοποιούνται για να ανανεώνουν την βάση.

Η βάση γνώσεων του emerge επεκτάθηκε μέσα από την εισροή εμπειρογνομώνων και τεκμηριώσεις βασισμένες σε ιατρικές τεχνικές έτσι ώστε να διασφαλιστούν όλες οι τελευταίες διαθέσιμες πληροφορίες. Η ιατρικές αποδείξεις επικεντρώνονται στη χρήση σχετικής ιατρική βιβλιογραφίας έτσι ώστε να παίρνετε η κατάλληλη κλινική απόφαση. Η πρόθεση του EBM είναι η παροχή πληροφοριών στο γιατρό για να τον βοηθήσει στην λήψη αποφάσεων, όχι όμως με την βοήθεια πληροφοριών που προέρχονται από κλινικά υπολογιστικά συστήματα. Παρόλα αυτά αυτές οι πληροφορίες μπορούν να βοηθήσουν στην αύξηση της βάσης δεδομένων αλλά και στα βοηθήματα αυτοματοποιημένων λήψεων αποφάσεων.

Ο τρίτος πράκτορας βασίζεται στο Hypernet, ένα μοντέλο νευρωνικού δικτύου το οποίο χρησιμοποιείται σαν εργαλείο ταξινόμησης για μια σειρά από ιατρικές εφαρμογές, συμπεριλαμβανομένης και τις ανάλυσης που γίνεται σε test ασκήσεων διαδρόμου. Το Hypernet χρησιμοποιεί μια διευρυμένη δυνητική συνάρτηση σ'έναν ελεγχόμενο αλγόριθμο μάθησης σε μια γραμμική δομή δικτύου με τρία η παραπάνω στρώματα. Η είσοδος των παραμέτρων μπορεί να είναι κάθε είδους δεδομένο.

Επίσης η ανάλυση του σήματος των δεδομένων είναι πολύ σημαντική σε πολλούς ιατρικούς τομείς, αλλά όχι τόσο σημαντικοί όσο το ηλεκτροκαρδιογράφημα στην καρδιολογία. Ο πράκτορας ανάλυσης σήματος χρησιμοποιεί μια μεθοδολογία βασισμένη σε μη γραμμική δυναμική. Το πακέτο συμβολίζεται από cats για χαστική ανάλυση χρονοσειρών

Η απεικόνιση όλων των τύπων είναι προσβάσιμη στη χρήση εικόνων, αρχειοθέτησης και αποθηκευτικών συστημάτων. Ενώ πολλοί αλγόριθμοι αναγνώρισης έχουν αναπτυχθεί για αυτοποιημένη αναγνώριση εικόνων, αυτές είναι γενικά συμπληρωματικές για την ερμηνεία ενός ακτινολόγου. Για την συγκεκριμένη υλοποίηση χρησιμοποιούνται φυσικές περιγραφές γλωσσών μη κανονικές. Σ'αυτή την περίπτωση ο ακτινολόγος μπορεί να θεωρηθεί σαν πράκτορας. Τα κύρια συστατικά της δομής ελέγχου είναι το έργο του διαχειριστή και η διεπαφή επικοινωνιών. Ο διαχειριστής σπάει το πρόβλημα σε δευτερεύοντες συνεργασίες και αυτές απευθύνονται στους ανάλογους πράκτορες.

Επίσης συνδυάζει αποτελέσματα από άλλους πράκτορες συμπεριλαμβανομένων των πελατών, για την συνολική αντιμετώπιση του προβλήματος. Ο τομέας που επικοινωνεί με τους πράκτορες δέχεται είσοδο από κάθε πράκτορα σε κατανοητή μορφή, και επίσης ερμηνεύει εξόδους από κάθε πράκτορα, έτσι ώστε να μπορούν να τους κατανοήσουν άλλοι πράκτορες. Τα κύρια συστατικά είναι ο συμβολικό αριθμητικός μετατροπέας, και η κοινή γλώσσα που ερμηνεύει εξόδους έτσι ώστε να μπορούν να τις ερμηνεύουν άλλοι πράκτορες.

Το μοντέλο έχει χρησιμοποιηθεί ως βοήθημα για μια σειρά από εφαρμογές, συμπεριλαμβανομένης μια διάγνωσης για καρδιακές διαταραχές Η προσέγγιση των ευφυών πρακτόρων προσφέρει το πλεονέκτημα της ελεύθερης επικοινωνίας μεταξύ διαφορετικών μεθοδολογιών συμπεριλαμβανομένης της χρήσης εξόδου από έναν πράκτορα ως εισροή σ'έναν άλλον. Για παράδειγμα, η έξοδος από τον

χαοτικό αναλυτή μπορεί να χρησιμοποιηθεί σαν είσοδο είτε σε μοντέλο νευρωνικού δικτύου είτε σε μοντέλα που βασίζονται στην γνώση.

Το μοντέλο περιλαμβάνει τόσο νέους τομείς γνώσεις, καθώς επίσης και νέες μεθοδολογίες χωρίς διορθωτικές αλλαγές στο συνολικού συστήματος. Σε σύνθετα διαγνωστικά συστήματα πολλοί τύποι δεδομένων για την πλήρη αξιολόγηση των ασθενών, με αποτέλεσμα την ανάγκη για διαφορετικές μεθοδολογίες για να μπορέσει να αλληλεπιδράσει με ομοιογενή τρόπο. Η ευφυείς πράκτορες είναι ιδιαίτερα σημαντικοί για την καρδιακή διάγνωση, και σε όλους τους τύπους που βρέθηκαν στην ιατρική και είναι σημαντικοί για την εμπειριστατωμένη αξιολόγηση. Οι μελλοντικές εργασίες θα περιλαμβάνουν δοκιμές σε μεγαλύτερους πληθυσμούς σ'ένα ευρύτερο φάσμα καρδιακών παθήσεων.

4.10 - Ψυχαγωγία

Η βιομηχανία ελεύθερου χρόνου συχνά δεν λαμβάνεται υπόψη από τη επιστήμη υπολογιστών. Οι εφαρμογές ψυχαγωγίας συχνά θεωρούνται σημαντικές στις εφαρμογές υπολογιστών. Οι εφαρμογές ελεύθερου χρόνου όπως τα βιντεοπαιχνίδια μπορεί να είναι πολύ προσοδοφόρα. Οι πράκτορες παίζουν αρκετά σημαντικό ρόλο στα βιντεοπαιχνίδια, στο θέατρο αλλά και σε εφαρμογές εικονικής πραγματικότητας, τέτοια συστήματα τείνουν να έχουν μη αυτόνομους κινούμενους χαρακτήρες, οι οποίοι μπορούν να εφαρμοστούν σε πράκτορες.

Διαλογικό θέατρο και Σινεμά

Λέγοντας διαλογικό θέατρο και σινεμά, εννοούμε ένα σύστημα που επιτρέπει στο χρήστη να παίζει ένα ρόλο ανάλογο με αυτό που παίζει στην πραγματικότητα, ηθοποιοί σε θεατρικά έργα ή ταινίες, τα οποία αλληλεπιδρούν με τεχνητούς, υπολογιστικούς χαρακτήρες που υιοθετούν συμπεριφορές πραγματικών ανθρώπων. Οι πράκτορες που παίζουν το ρόλο των ανθρώπων σε αμφιθεατρικές εφαρμογές είναι γνωστοί ως πιστευτοί πράκτορες, ένας όρος ο οποίος επινοήθηκε από τον Joe Bates [1].

Ορισμένα έργα έχουν συσταθεί για την διεύρυνση της ανάπτυξης των εν λόγω πρακτόρων. Για παράδειγμα ένα κατευθυνόμενο σύστημα αυτοσχεδιασμού, όπου οι ανθρώπινοι παράγοντες παίζουν δυναμικό ρόλο στην δημιουργία ενός νέου χρήστη κατευθυνόμενου έργου.

Παιχνίδια

Έχουν περιγράψει πολλές εφαρμογές της θεωρίας των πρακτόρων σε παιχνίδια υπολογιστών. Για παράδειγμα, έχουν αναπτύξει μια εφαρμογή του γνωστού Tetris, όπου ο χρήστης προσπαθεί να κάνει έναν τοίχο έξω από ένα ακανόνιστο σχήμα τα οποία υπάγονται σε block. Ο πράκτορας στο παιχνίδι παίζει το ρόλο του χρήστη, ο οποίος πρέπει να ελέγχει που πρέπει να πέφτουν τα block.

Προσπαθώντας να προγραμματίσουμε αυτό τον πράκτορα ο οποίος χρησιμοποιεί έξυπνες τεχνικές απαιτείται να περάσουμε από ένα στάδιο εκμείευσης της γνώσης, αντιπροσωπεύοντας γνώση για το παιχνίδι και ο ρόλος του χρήστη υπό τους όρους συμβολικής κατασκευής δεδομένων όπως κανόνες κ.λ.π. Αυτή η προσέγγιση είναι μη ρεαλιστική για ένα παιχνίδι όπως το Tetris, το οποίο έχει σκληρούς περιορισμούς. Υπάρχουν επίσης αρκετές εναλλακτικές λύσεις όπως το αντιδραστικό μοντέλο γνωστό και ως RTA(Real time able)

Σε αυτή την προσέγγιση, οι πράκτορες είναι προγραμματισμένοι σύμφωνα με την συμπεριφορά τους. Οι συγκεκριμένες συμπεριφορές είναι απλές δομές οι οποίες μοιάζουν με τους κανόνες αλλά δεν έχουν τόσο πολύπλοκη λογική.

Το γεγονός ότι τα ηλεκτρονικά παιχνίδια γίνονται πιο σύνθετα και οι καταναλωτές ζητούν πιο εξελιγμένους υπολογιστικούς αντιπάλους, αναγκάζει τους προγραμματιστές παιχνιδιών να δώσουν

έμφαση στην τεχνητή νοημοσύνη των παιχνιδιών. Από έρευνες που έγιναν πάνω στην ανάπτυξη των ευφυών πρακτόρων όπως στην αρχιτεκτονική των ευφυών πρακτόρων, την γνώση της εκπροσώπησης, την γνώση επαναχρησιμοποίησης προτάθηκαν κάποιοι τομείς των ευφυών πρακτόρων που θα μπορούν να εφαρμοστούν στα παιχνίδια. Παράδειγμα τέτοιων παιχνιδιών που χρησιμοποιήθηκαν οι παραπάνω τεχνικές είναι τα παιχνίδια Quake 2 και το Descent 3.

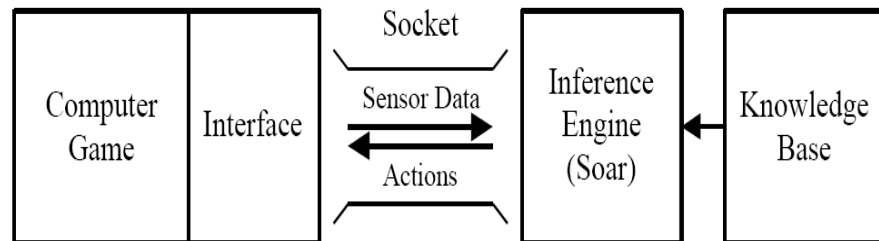
Από τα συγκεκριμένα υπάρχουν παιχνίδια υπάρχουν κάποιοι στόχοι όσον αναφορά την έρευνα και την αναπτυξιακή προοπτική των παιχνιδιών. Για την κοινότητα της συγκεκριμένης έρευνας παρέχονται περιβάλλοντα για την δοκιμή των αποτελεσμάτων σε περιοχές όπως η μηχανική μάθηση, της ευφυείς αρχιτεκτονικής κ.λ.π. Επίσης τα δύσκολα θέματα της συγκεκριμένης έρευνας έχουν προτείνει νέα ερευνητικά προβλήματα σχετικά με την εκπροσώπηση γνώσεων, καθοδήγηση πράκτορα και αλληλεπίδραση ανθρώπου υπολογιστή. Βασικός στόχος της συγκεκριμένης έρευνας είναι να κάνει τα παιχνίδια πιο διασκεδαστικά κάνοντας τους πράκτορες στο παιχνίδι πιο έξυπνους. Εάν γίνει σωστά το να παίζει με ή κόντρα στους ευφυείς πράκτορες θα είναι πιο εύκολο να παίξουν άνθρωποι διαδικτυακά όπως γίνεται σήμερα κατά κόρων.

Μια ευέλικτη ευφυείς αρχιτεκτονική όπως η Soar θα κάνει την ανάπτυξη των ευφυών πρακτόρων στα παιχνίδια ευκολότερη με την παραγωγή ενός μηχανισμού εξαγωγής συμπερασμάτων, και την επαναχρησιμοποίηση μιας βάσης γνώσεων που μπορούν να εφαρμοστούν εύκολα σε πολλά παιχνίδια. Τα δύο παιχνίδια που αναφέρθηκαν πιο πριν είναι από τα πιο γνωστά σε first person παιχνίδια, τα οποία περιλαμβάνουν το λογισμικό που επιτρέπει στους προγραμματιστές να γράφουν σε κώδικα c, κάτι το οποίο τους επιτρέπει να έχουν πρόσβαση στις εσωτερικές δομές του παιχνιδιού και στον έλεγχο του πράκτορα.

Τα προηγούμενα έχουν επιτρέψει στους ερευνητές να εξάγουν από τα παιχνίδια συμβολικές πληροφορίες, χωρίς τα δύσκολα προβλήματα που συνδέονται με τα μηνύματα που εμφανίζονται στην οθόνη του υπολογιστή. Μια κοινή προσέγγιση για να αναπτυχθούν οι ευφυείς πράκτορες στα ηλεκτρονικά παιχνίδια είναι να χρησιμοποιηθεί κώδικας C και προγραμματιστικούς ελεγκτές έτσι ώστε να μπορούν να ελέγξουν τους πράκτορες μέσω ενός αριθμού ενθétων. Όσο οι πράκτορες γίνονται πιο πολύπλοκοι, τόσο πιο δύσκολο είναι για τα on κώδικα c ο οποίος τους περιέχει τους περιέχει να κάνει debug, να διατηρηθεί και να βελτιωθεί .

Το να χρησιμοποιήσουμε την αρχιτεκτονική Soar πάρα να αναπτύξουμε μια νέα αρχιτεκτονική επωφελούμαστε από μια αρχιτεκτονική 15 χρόνων. Η αρχιτεκτονική soar λειτουργεί ως μηχανή συμπερασματολογίας. Στόχος της συγκεκριμένης μηχανής είναι να χρησιμοποιεί τις γνώσεις της υπάρχουσας κατάστασης για το τι λειτουργίες εξωτερικές και εσωτερικές πρέπει να γίνουν. Η τρέχουσα κατάσταση δεδομένων αντιπροσωπεύεται από δομές δεδομένων που εκπροσωπούνται από τομείς προσομοιωμένων αισθητήρων που υλοποιούνται στο περιβάλλον εργασίας, και οι πληροφορίες αποθηκεύονται στην εσωτερική μνήμη του Soar.

Η αρχιτεκτονική soar επιτρέπει την διάσπαση των ενεργειών μέσω μια ιεράρχησης φορέων. Οι φορείς εκμετάλλευσης στα υψηλότερα επίπεδα της ιεραρχίας αντιπροσωπεύουν το στόχο των πρακτόρων, ενώ το χαμηλότερο επίπεδο τα επιμέρους βήματα τα οποία πρέπει να γίνουν για να επιτευχθούν οι στόχοι. Οι αντιπροσωπευτικοί στόχοι στην εσωτερική μνήμη ενθαρρύνει την ανάπτυξη του πράκτορα για την δημιουργία κατευθυνόμενου στόχου. Η συγκεκριμένη τεχνική επιλέγει και εκτελεί πράξεις που σχετίζονται με την τρέχουσα κατάσταση και καθορίζονται από τις εξωτερικές ενέργειες, τις κινήσεις του πράκτορα στο παιχνίδι και εσωτερικές δράσεις όπως οι αλλαγές στους εσωτερικούς στόχους του πράκτορα.



Οι συγκεκριμένη αρχιτεκτονική αποτελείται από τρεις κινήσεις:

- Αποδοχή πληροφοριών από τον αισθητήρα του παιχνιδιού
- Επιλογή και πράξη σχετικών γνώσεων
- Εκτέλεση εσωτερικών και εξωτερικών δράσεων

Το σημαντικό του συγκεκριμένου σχεδίου είναι διασύνδεση μεταξύ της παραγωγής συμπεράσματος και του περιβάλλοντος προσομοίωσης. Η διεπαφή εξάγει τις απαραίτητες πληροφορίες τις οποίες κωδικοποιεί στη μορφή που απαιτεί η soar. Κάθε παιχνίδι απαιτεί προσαρμοσμένη διεπαφή επειδή οι λεπτομέρειες της διεπαφής και το περιεχόμενο της γνώσης ποικίλουν από παιχνίδι σε παιχνίδι. Επίσης το γεγονός κάθε διασύνδεση στηρίζεται σ'ένα κοινό πρότυπο μας επιτρέπει να χρησιμοποιήσουμε τη γνώση από ένα παιχνίδι και σε άλλα.

Οι φορείς εκμετάλλευσης στο ανώτατο επίπεδο ιεραρχίας αντιπροσωπεύουν τους γενικούς στόχους του πράκτορα ή τους τρόπους συμπεριφοράς τους. Τα χαμηλότερα επίπεδα ιεραρχίας αντιπροσωπεύουν διαδοχικά πιο ειδικές αναπαραστάσεις στη συμπεριφορά του πράκτορα. Οι φορείς εκμετάλλευσης των χαμηλών στρωμάτων είναι ατομικά μέτρα και δράσεις που υλοποιούν τα ανώτερα στρώματα.

Βασικός στόχος είναι να επωφεληθούμε από τη δημιουργία ενός ανεξάρτητου παιχνιδιού με μια βάση γνώσεων που προέρχεται και από τα δύο παιχνίδια που προαναφέραμε πιο πριν. Ένα κομμάτι του παιχνιδιού αποτελείτε από έξι σταθμούς εργασίας, τρεις για το ένα παιχνίδι και τρεις για το άλλο. Για κάθε παιχνίδι ένας σταθμός εργασίας διατρέχει τον διακομιστή των παιχνιδιών αλλά και το σύστημα ευφυών πρακτόρων, ένας δεύτερος σταθμός εμφανίζει το τρέχον παιχνίδι από την προοπτική του πράκτορα και τα μέλη του κοινού τα οποία μπορούν να παίξουν το παιχνίδι κόντρα στον ευφυή πράκτορα στον τρίτο σταθμό. Τέλος το κοινό θα είναι σε θέση να αξιολογεί την αποτελεσματικότητα στο να παίζεις ένα παιχνίδι

4.11 – Στοιχεία των εφαρμογών πρακτόρων

Σε ένα συγκεκριμένο επίπεδο αφαίρεσης, πολλές από τις παραπάνω εφαρμογές μοιράζονται κοινά χαρακτηριστικά. Οι μεμονωμένοι πράκτορες είναι σχεδιασμένοι και κατασκευασμένοι για να θεσπίζουν ιδιαίτερους ρόλους. Αυτοί οι πράκτορες είναι αυτόνομοι, ο στόχος τους στρέφεται στις οντότητες, οι οποίες ανταποκρίνονται στο περιβάλλον τους. Τυπικά πρέπει να αλληλεπιδρούν με άλλους πράκτορες προκειμένου να διενεργούν το ρόλο τους.

Αυτές οι αλληλεπιδράσεις είναι μια φυσική συνέπεια της αναπόφευκτης αλληλεξάρτησης που υπάρχει μεταξύ των πρακτόρων, το περιβάλλον τους, και τους σχεδιαστικούς του στόχους. Δεδομένου ότι οι πράκτορες είναι αυτόνομοι, οι αλληλεπιδράσεις είναι συνήθως πολύ σοφιστικέ συμπεριλαμβανομένης

της συνεργασίας, του συντονισμού και της διαπραγμάτευσης. Μετά από αυτή την άποψη μπορούν να γίνουν δύο σημαντικές παρατηρήσεις σχετικά με εφαρμογές που στηρίζονται στους πράκτορες:

- Οι λεπτομερείς ενέργειες επίλυσης του προβλήματος του πράκτορα μπορούν να καθοριστούν κατά την εκτέλεση του χρόνου. Η ατομική συμπεριφορά ρυθμίζεται από την σύνθετη αλληλεπίδραση μεταξύ της εσωτερικής κατάστασης των πρακτόρων και της εξωτερικής τους κατάστασης. Έτσι η ακριβή τροχιά που μπορεί να ακολουθήσει ένας πράκτορας μπορεί να ανακαλυφθεί μόνο αν τρέξουμε τον πράκτορα στο περιβάλλον του.
- Επειδή η συμπεριφορά των μεμονωμένων πρακτόρων δεν είναι μοναδικά αποφασισμένοι σε σχεδιασμένο χρόνο, η συμπεριφορά του συνολικού συστήματος μπορεί να εμφανιστεί μόνο σε πραγματικό χρόνο.

Αυτά τα σημεία έχουν κάποιες προφανείς συνέπειες από τη χρήση τέτοιων προσεγγίσεων οι οποίες βασίζονται σε πράκτορες με κρίσιμα ασφαλής εφαρμογές, όπως ο έλεγχος εναέριας κυκλοφορίας, όπου είναι σημαντικό ότι το σύστημα πληρεί τις προδιαγραφές του. Το συγκεκριμένο θέμα αναλύεται στο κεφάλαιο 2.2. Η από πάνω εφαρμογές επίσης δείχνουν πολλές διαφορετικές διαστάσεις κατά μήκος του οποίου μπορούμε να αναλύσουμε συστήματα που βασίζονται σε πράκτορες:

- **Πολυπλοκότητα πρακτόρων.** Οι πράκτορες μπορούν να εκτελεστούν πραγματοποιώντας τρεις τύπους συμπεριφοράς. Στο απλούστερο επίπεδο, είναι ο πράκτορας *gopher* ο οποίος εκτελεί συγκριτικά απλές διεργασίες βασισμένα σε καθορισμένους, προκαθορισμένους κανόνες και εφαρμογές. Το επόμενο επίπεδο πολυπλοκότητας περιλαμβάνει υπηρεσίες που ασκούν οι πράκτορες, οι οποίοι είναι υψηλού επιπέδου, αλλά ακόμα σαφώς καθορισμένη αποστολή έπεται από αίτηση του χρήστη(Π.χ κανονίζετε μια συνάντηση για να βρεθεί η κατάλληλη πτήση). Τέλος υπάρχουν προβλεπόμενοι και ενεργητικοί πράκτορες, ικανοί για ευέλικτη και αυτόνομη συμπεριφορά με τον τρόπο που σχολιάσαμε παραπάνω. Απλά παραδείγματα είναι πράκτορες που προσφέρουν πληροφορίες και υπηρεσίες σε ένα χρήστη, χωρίς να ερωτηθούν όποτε αυτό κρίνεται σκόπιμο.
- **Ο ρόλος των πρακτόρων.** Οι πράκτορες παίζουν πολλούς ρόλους. Για παράδειγμα σε πολλές από τις εμπορικές και βιομηχανικές εφαρμογές που αναφέρθηκαν από πάνω, ο ρόλος του πράκτορα του συστήματος είναι να παρέχει μια λειτουργία λήψης αποφάσεων. Ο πράκτορας λειτουργεί αυτόνομα και εκ των προτέρων για να συλλέξει πληροφορίες και να προβεί στις απαραίτητες συστάσεις, αλλά απαιτείται ανθρώπινος παράγοντας για τις τελικές αποφάσεις. Σε αντίθεση στον τομέα της ψυχαγωγίας, ο πράκτορας αυτοματοποιεί το ρόλο επίλυσης του προβλήματος, κατά συνέπεια είναι μια συγκεκριμένη δραστηριότητα η οποία είναι εξολοκλήρου υπεύθυνη για την υλοποίηση του.
- **Διακριτικότητα της Άποψης.** Σε κάποιες από τις εφαρμογές που αναλύσαμε παραπάνω, η σημαντική μονάδα ανάλυσης και σχεδιασμού είναι ο αυτόνομος πράκτορας, ενώ σε άλλες εφαρμογές το κλειδί είναι κοινωνία των πρακτόρων. Το αν θα υιοθετηθεί η προσέγγιση του ενός πράκτορα ή των πολυπρακτόρων εξαρτάται από τον τομέα, και είναι παρόμοιας φύσης με τις αποφάσεις σχετικά με το αν είναι μονολιθικό, κεντρικές λύσεις ή διανεμημένες, οι αποκεντρωμένες λύσεις είναι οι κατάλληλες. Τα συστήματα που χρησιμοποιούν ένα πράκτορα κατά μία έννοια είναι πιο απλά από τα συστήματα πολυπρακτόρων, δεδομένου ότι δεν απαιτούν από το σχεδιαστή να ασχοληθεί με ζητήματα όπως η συνεργασία ή διαπραγμάτευση και ούτω κάθε εξής.

Κεφάλαιο 5 – Συστήματα Πολλαπλών πρακτόρων

5.1 Επικοινωνία μεταξύ πρακτόρων

Ένας πράκτορας που δραστηριοποιείται σε ένα περιβάλλον όπου βρίσκονται και άλλοι πράκτορες χρειάζεται να αλληλεπιδρά μαζί τους προκειμένου να ζητήσει ή να παράσχει κάποιες πληροφορίες ή υπηρεσίες. Αυτό συμβαίνει διότι, αντίθετα με τα παλαιότερα συστήματα Τεχνητής Νοημοσύνης, όπως τα έμπειρα συστήματα, οι πράκτορες δεν είναι απομονωμένες οντότητες νοημοσύνης ούτε έχουν γενικά πλήρη και ορθή γνώση του περιβάλλοντος. Αντίθετα, πράκτορες που εφαρμόζονται στην πράξη, έχουν συγκεκριμένες ικανότητες για την επίτευξη των στόχων τους, περιορισμένους μηχανισμούς για την παραγωγή συμπερασμάτων και διατηρούν μερική γνώση του περιβάλλοντος. Έτσι, είναι αναγκαία η επικοινωνία με άλλους πράκτορες για την ανταλλαγή πληροφοριών, την παροχή υπηρεσιών και τον συντονισμό των ενεργειών τους. Κατά συνέπεια, ένας τέτοιος πράκτορας είναι εξ ορισμού κοινωνικός (social) και η ικανότητα επικοινωνίας απαραίτητη. Στο γενικότερο πλαίσιο της Κατανεμημένης Τεχνητής Νοημοσύνης, όπου εντάσσονται οι πράκτορες, έχουν χρησιμοποιηθεί πέντε είδη επικοινωνίας.

Το πρώτο είδος είναι ότι δεν υπάρχει καθόλου επικοινωνία, αλλά οι πράκτορες συμπεραίνουν τα πλάνα των άλλων χωρίς να επικοινωνούν μεταξύ τους ή δραστηριοποιούνται σε ένα κατάλληλα δομημένο περιβάλλον που εξασφαλίζει την ανάγκη επικοινωνίας ή ακόμα ενεργούν σύμφωνα με ένα προκαθορισμένο σχήμα, όπου δεν υπάρχει η ανάγκη ανταλλαγής πληροφοριών. Μία τέτοια προσέγγιση δεν είναι γενική και η ανάπτυξη τέτοιων συστημάτων εφαρμόζεται μόνο σε συγκεκριμένες και περιορισμένες περιπτώσεις. Το ίδιο ισχύει και για το δεύτερο είδος, την επικοινωνία μέσω σημάτων (signals) όπου η ανταλλαγή (περιορισμένων) μηνυμάτων και ο συγχρονισμός γίνεται με έναν τρόπο ανάλογο των σημάτων. Δύο άλλα είδη επικοινωνίας αφορούν αφενός τη θεωρία που έχει προταθεί από τον Hewitt και έχει επεκταθεί από τον Agha και βασίζεται στην ανταλλαγή μηνυμάτων (message passing), αφετέρου την ανταλλαγή πλάνων (plan passing). Πολλά συστήματα που έχουν αναπτυχθεί χρησιμοποιούν τέτοιες μορφές επικοινωνίας. [60]

Ωστόσο, η επικοινωνία βασισμένη σε ανταλλαγή μηνυμάτων απαιτεί μεγάλο αριθμό συγκεκριμένων πρακτόρων, ενώ η επικοινωνία βασισμένη σε ανταλλαγή πλάνων είναι μία αρκετά συγκεκριμένη τεχνική. Η πιο γενική και περισσότερο αποδεκτή μορφή επικοινωνίας είναι αυτή που βασίζεται στη θεωρία ενεργειών λόγου (speech acts theory), η οποία αρχικά αναπτύχθηκε για τη γλωσσική ανάλυση της ανθρώπινης επικοινωνίας. Η θεωρία ενεργειών λόγου είναι σύμφωνη με τα όσα αναφέρθηκαν στο Κεφάλαιο 2, δηλαδή ότι οι πράκτορες είναι συστήματα προθέσεων. Η γενικότητα της θεωρίας ενεργειών λόγου και η άμεση εφαρμογή της στους πράκτορες την καθιστούν ως τον πιο κατάλληλο τρόπο επικοινωνίας για τους πράκτορες και για αυτό εξετάζεται στη συνέχεια με περισσότερη λεπτομέρεια.

Σύμφωνα με τη θεωρία ενεργειών λόγου, οι εκφράσεις λόγου (utterances) του ανθρώπου αποτελούν ενέργειες που γίνονται με κάποια πρόθεση και είναι ανάλογες των φυσικών ενεργειών. Αυτές οι εκφράσεις λόγου γίνονται σε ένα συγκεκριμένο πλαίσιο (context) και επιδρούν σε αυτό όπως επιδρούν και όλες οι υπόλοιπες φυσικές ενέργειες. Ο Austin (που αρχικά εισήγαγε τη θεωρία ενεργειών λόγου) ονομάζει τέτοιες εκφράσεις performatives. Η σύνταξη ενός performative εκφράζεται με τη μορφή «I hereby request ...» ή «I hereby declare ...». Ουσιαστικά, performative είναι κάποιο ρήμα που το νόημά του σημαίνει και εκτέλεση της ενέργειας η οποία όμως μπορεί και να αποτύχει. Έτσι, δεν είναι όλα τα ρήματα performatives, διότι για παράδειγμα η έκφραση «I hereby solve this equation» δε σημαίνει ότι στην πραγματικότητα λύνω την εξίσωση. Γενικά, οι ενέργειες που εκφράζονται με κάποιο performative ρήμα ονομάζονται illocutionary acts και ο όρος ενέργειες λόγου τείνει να είναι συνώνυμος των illocutionary acts. Ο Austin ανέφερε άλλα δύο είδη ενεργειών, τις locutionary acts που αναφέρονται στη διατύπωση μίας έκφρασης λόγου και τις perlocutionary acts που αναφέρονται στην πρόκληση ενός αποτελέσματος στον αποδέκτη της έκφρασης λόγου. Συνεχίζοντας τη θεωρία του Austin ο Searl προσπάθησε να τυποποιήσει κατά κάποιο τρόπο τη δομή της θεωρίας ενεργειών λόγου δίνοντας ικανές και αναγκαίες συνθήκες για την επιτυχή έκβαση των ενεργειών. Επίσης, για να αποφευχθεί η σύγχυση συνωνύμων ρημάτων, όρισε συνθήκες που τα κατηγοριοποιούν ανάλογα με τα αποτελέσματα των ενεργειών. Έτσι, οι ενέργειες μπορεί να είναι οι εξής:

- Αντιπροσωπευτικές (representatives), όπου ο ομιλητής δεσμεύεται ότι η έκφραση είναι αληθής, π.χ. assert.
- Οδηγίες (directives), όπου ο ομιλητής ζητάει κάτι από τον αποδέκτη, π.χ. request.
- Δεσμευτικές (commissives), όπου ο ομιλητής δεσμεύεται για την ενέργεια, π.χ. promise.
- Εκφραστικές (expressives), όπου ο ομιλητής εκφράζει κάποια κατάσταση (συνήθως ψυχολογική), π.χ. thank.
- Δηλώσεις (declarations), όπου ο ομιλητής δηλώνει μία κατάσταση (πολλές φορές επιβάλλοντάς τη) π.χ. declare.

Σε μία προσπάθεια να ενταχθεί η θεωρία ενεργειών λόγου στην Τεχνητή Νοημοσύνη, οι Cohen και Perrault [9] αναδιατύπωσαν τις ικανές και αναγκαίες συνθήκες του Searl με μεθόδους που χρησιμοποιούνται για τα πλάνα (plans) με αποτέλεσμα τη λεγόμενη Plan-Based Theory of Speech Acts. Έτσι, πρότειναν ότι οι ενέργειες λόγου μπορούν να χρησιμοποιηθούν ως τελεστές στα πλάνα με τον ίδιο τρόπο που χρησιμοποιούνται και οι φυσικές ενέργειες. Επίσης, το γενικότερο πλαίσιο (context) όπου ενεργούν οι ενέργειες λόγου θεωρείται ότι είναι η νοητική κατάσταση των πρακτόρων που συμμετέχουν και αναπαριστάται από τις πεποιθήσεις και τις επιθυμίες. Με αυτόν τον τρόπο υπάρχει απόλυτη συνέπεια με τη θεώρηση των πρακτόρων ως συστημάτων προθέσεων.

Δύο βασικές ενέργειες αυτής της θεωρίας είναι οι Request και Inform όπως φαίνονται και στον παρακάτω πίνακα:

Για να εκτελεστούν αυτές οι ενέργειες πρέπει να ικανοποιούνται κάποιες συνθήκες (PRECONDITIONS: CANDO.PR, WANT.PR), ενώ μετά το πέρας της εκτέλεσής τους παράγονται κάποια αποτελέσματα (EFFECTS). Οι προσυνθήκες CANDO.PR και WANT.PR εκφράζουν τη δυνατότητα και θέληση ενός πράκτορα να εκτελέσει μία ενέργεια.

Έτσι, για την επιτυχή πραγματοποίηση της Request, ο πράκτορας που κάνει την αίτηση (speaker ή S) πρέπει να πιστεύει ότι ο πράκτορας στον οποίον απευθύνεται (hearer ή H) είναι ικανός να εκτελέσει την ζητούμενη ενέργεια και, επίσης, πρέπει να πιστεύει (ο S) ότι ο H πιστεύει και αυτός ότι έχει την ικανότητα να εκτελέσει τη ζητούμενη ενέργεια (προσυνθήκη CANDO.PR).

Επιπλέον, ο S θα πρέπει να πιστεύει ότι πραγματικά θέλει να εκτελεστεί η Request (προσυνθήκη WANT.PR). Το αποτέλεσμα της πραγματοποίησης της Request είναι να πιστεύει ο H ότι ο S πιστεύει ότι θέλει να εκτελεστεί κάποια ενέργεια (EFFECT). Αντίστοιχα για την Inform θα πρέπει ο S να πιστεύει ότι η φ είναι αληθής και ως αποτέλεσμα ο H πιστεύει ότι ο S πιστεύει τη φ. Ωστόσο, οι Request και Inform αναφέρονται στην αίτηση για εκτέλεση ενεργειών ή στη γνωστοποίηση πληροφοριών αντίστοιχα (illocutionary acts) αλλά δεν εγγυώνται ότι οι ζητούμενες ενέργειες θα εκτελεστούν ή οι πληροφορίες θα υιοθετηθούν τελικά.

Προκειμένου να καλυφθεί αυτή η αδυναμία, οι Cohen και Perrault [9] περιγράφουν τέτοιου είδους ενέργειες (perlocutionary acts) όπως για παράδειγμα οι Cause_to_Want και Convince. Έτσι, η Cause_to_Want έχει ως αποτέλεσμα ένας πράκτορας να πιστεύει ότι θέλει να κάνει κάτι, με την προϋπόθεση ότι κάποιος άλλος πράκτορας πιστεύει ότι θέλει να το κάνει, ενώ η Convince έχει ως αποτέλεσμα ένας πράκτορας να πιστεύει μία πρόταση φ, αν πιστεύει ότι ένας άλλος πράκτορας πιστεύει τη φ.

Τόσο η Cause_to_Want όσο και η Convince έχουν άμεση σχέση με τη Request και την Inform αντίστοιχα. Κάποια προβλήματα αυτής της θεωρίας οδήγησαν τους Cohen και Levesque [3] να διατυπώσουν μία θεωρία λογικής ενέργειας που βασίζεται περισσότερο στη λογική ανάλυση της θεωρίας ενεργειών λόγου. Το βασικό στοιχείο αυτής της θεωρίας είναι ότι οι illocutionary acts δεν είναι βασικές και θεμελιώδεις αλλά είναι σύνθετοι τύποι δεδομένων.

Δε θα αναπτυχθεί περαιτέρω ο formalismός της θεωρίας αυτής, ωστό- σο αναφέρεται ότι οι Cohen και Levesque απέδειξαν ότι πολλές από τις συνθήκες που είχε εκθέσει ο Searl μπορούν να παραχθούν από τη θεωρία τους.

5.2 - Πρωτόκολλα Επικοινωνίας μεταξύ πρακτόρων

Τις περισσότερες φορές, οι συνήθεις διάλογοι μεταξύ των πρακτόρων ακολουθούν μία συγκεκριμένη ακολουθία ενεργειών. Σε τέτοιες περιπτώσεις, όπως επίσης και σε κάθε σημείο του διαλόγου αναμένονται συγκεκριμένες ακολουθίες μηνυμάτων. Τέτοια τυπικά πρότυπα ανταλλαγής μηνυμάτων καλούνται πρωτόκολλα επικοινωνίας (communication protocols). Στη συνέχεια παρουσιάζονται τέτοια πρωτόκολλα επικοινωνίας όπως έχουν προταθεί και εφαρμοστεί σε εφαρμογές συστημάτων πολλαπλών πρακτόρων.

5.3 Το πρωτόκολλο Contract Net

Το Contract Net είναι ένα πρωτόκολλο υψηλού επιπέδου για την αποτελεσματική συνεργατική δράση σε δίκτυα συστημάτων επίλυσης προβλημάτων (problem solvers). Η βασική ιδέα είναι ότι η διαπραγμάτευση (negotiation) αποτελεί το βασικό συστατικό για την συνεργατική επίλυση προβλημάτων. Έτσι, στην ουσία, το Contract Net αποτελεί ένα πλαίσιο για την διαχείριση των διαπραγματεύσεων. Κάθε πράκτορας καλείται κόμβος (node) στην ορολογία του πρωτοκόλλου και είναι ένα αυτόνομο σύστημα επίλυσης προβλημάτων και ταυτόχρονα έχει τη δυνατότητα να επικοινωνεί με άλλους πράκτορες με ανταλλαγή μηνυμάτων. Όταν ένας κόμβος χρειαστεί κάποια ενέργεια που δεν μπορεί να εκτελέσει ο ίδιος, αναλαμβάνει το ρόλο του manager και κάνει μία σχετική ανακοίνωση (καλείται task announcement). [12]

Αν δεν έχει πληροφορίες για τις ικανότητες των υπολοίπων κόμβων (που έχουν το ρόλο του contractor), τότε η ανακοίνωση είναι γενική (general broadcast). Σε αντίθετη περίπτωση η ανακοίνωση μπορεί να είναι περιορισμένη (limited broadcast) ή ακόμα και προσωπική (peer-to-peer announcement). Στη συνέχεια η αλληλεπίδραση συνεχίζεται όπως έχει περιγραφεί στην προηγούμενη υποενότητα, δηλαδή ο κόμβος που έχει κάνει την αίτηση δέχεται προσφορές, τις αξιολογεί και ειδοποιεί ποιες από αυτές έχει αποδεχτεί και ποιες έχει απορρίψει. Αρκετά συστήματα έχουν αναπτυχθεί βασισμένα στο πρωτόκολλο Contract Net, μεταξύ αυτών το YAMS (Yet Another Manufacturing System), ένα σύστημα εφαρμοσμένο στη βιομηχανία και συγκεκριμένα στον έλεγχο των κατασκευών.

Ένα σύνολο συσκευών που συμμετέχουν για τη δημιουργία μιας κατασκευής μοντελοποιούνται ως μία ιεραρχία κελιών εργασίας (workcells). Για παράδειγμα, τέτοια κελιά εργασίας θα μπορούσαν να είναι φρέζες, τόρνοι, συσκευές λείανσης, βαφής κτλ. Επιπλέον, τα κελιά εργασίας ομαδοποιούνται περαιτέρω σε ευέλικτα συστήματα κατασκευών (flexible manufacturing systems – FMS), τα οποία προσφέρουν κάποια λειτουργικότητα, όπως συναρμολόγηση, βαφή κτλ. Μία συλλογή από FMS ομαδοποιείται σε ένα εργοστάσιο και μία εταιρεία μπορεί να έχει διάφορα τέτοια εργοστάσια. Η διαδικασία παραγωγής ορίζεται από τη συνεχή αλλαγή παραμέτρων, όπως προϊόντα που πρέπει να κατασκευαστούν, διαθέσιμοι πόροι, χρονικοί περιορισμοί κτλ.

Ο σκοπός του YAMS είναι να διαχειριστεί την πολύπλοκη διαδικασία παραγωγής και υιοθετεί μία προσέγγιση πολλαπλών πρακτόρων, όπου κάθε εργοστάσιο και κάθε συστατικό του εργοστασίου αναπαριστάται ως πράκτορας. Εφόσον η επικοινωνία των πρακτόρων βασίζεται στο πρωτόκολλο Contract Net, το πρόβλημα της απόφασης της καλύτερης διαδικασίας για την κατασκευή ενός προϊόντος σύμφωνα με τους περιορισμούς που προκύπτουν θεωρείται πρόβλημα διαπραγμάτευσης μεταξύ των πρακτόρων.

5.4 Το πρωτόκολλο και γλώσσα KQML

Μία πολύ σημαντική προσπάθεια υλοποίησης μίας κοινής γλώσσας και ταυτοχρόνως ενός πρωτοκόλλου επικοινωνίας είναι η Knowledge Query and Manipulation Language (KQML). Η KQML στηρίζεται στις αρχές της θεωρίας ενεργειών λόγου και υποστηρίζει ένα ευρύ σύνολο από στοιχειώδεις ενέργειες (καλούνται και εδώ *performatives*), οι οποίες ορίζουν τις επιτρεπόμενες λειτουργίες των πρακτόρων. Επιπλέον, αυτό το σύνολο είναι ανοικτό, επιτρέποντας έτσι την προσθήκη άλλων ή την αφαίρεση υπαρχόντων *performatives* ανάλογα με τις ανάγκες της εφαρμογής.[12]

Τα μηνύματα της KQML είναι αδιαφανή (*opaque*) ως προς το περιεχόμενο, με την έννοια ότι δεν είναι προτάσεις κάποιας συγκεκριμένης γλώσσας αλλά εκφράζουν μία θέση ως προς το περιεχόμενο, όπως αίτηση, ερώτηση, πληροφόρηση, κτλ. Ωστόσο, η KQML επιτρέπει ως ορίσματά της άλλες εσωτερικές γλώσσες με τις οποίες μπορούν να επικοινωνούν συγκεκριμένοι πράκτορες (π.χ. Prolog, Lisp). Η γλώσσα KQML αποτελείται από τρία επίπεδα:

- Το επίπεδο περιεχομένου (*content layer*), στο οποίο περιέχεται το αυθεντικό μήνυμα στη γλώσσα αναπαράστασης των πρακτόρων που επικοινωνούν και μπορούν να το κατανοήσουν ως προς τη σύνταξή του.
- Το επίπεδο επικοινωνίας (*communication layer*), στο οποίο κωδικοποιούνται τα χαρακτηριστικά του μηνύματος που περιγράφουν παραμέτρους όπως η ταυτότητα του αποστολέα, του παραλήπτη, κτλ.
- Το επίπεδο μηνύματος (*message layer*), το οποίο αποτελεί τον πυρήνα της γλώσσας και καθορίζει τα είδη των αλληλεπιδράσεων μεταξύ των πρακτόρων. Μία βασική ενέργεια που καθορίζεται στο επίπεδο μηνύματος είναι ο καθορισμός του πρωτοκόλλου μεταφοράς και η ενέργεια λόγου που συνοδεύει το περιεχόμενο του μηνύματος.

Επιπλέον, περιλαμβάνονται προαιρετικά χαρακτηριστικά που περιγράφουν τη γλώσσα που είναι γραμμένο το μήνυμα και η οντολογία που το συνοδεύει. Για παράδειγμα, ένα KQML μήνυμα από το πράκτορα *joe* στον πράκτορα *stock-server*, το οποίο αποτελεί ερώτηση σχετικά με την μετοχή της IBM και διατυπώνεται σε γλώσσα Prolog και οντολογία NYSE-TICKS, έχει την ακόλουθη μορφή:

(ask-one

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

Στο παραπάνω μήνυμα, η ενέργεια λόγου ή *performative* καθορίζεται από το *ask-one* και τα υπόλοιπα στοιχεία του μηνύματος αποτελούν τιμές των πεδίων (καλούνται λέξεις κλειδιά – *keywords*) *sender*, *content* κτλ. Η τιμή της λέξης κλειδί “:content” αποτελεί το επίπεδο περιεχομένου, ενώ οι τιμές των λέξεων κλειδιών “:sender”, “:receiver”, “:reply-with” αποτελούν το επίπεδο επικοινωνίας. Το όνομα του *performative ask-one* μαζί τις τιμές των λέξεων κλειδιών “:language” και “:ontology” αποτελούν το επίπεδο μηνύματος. Η απάντηση του πράκτορα *stock-server* έχει την ακόλουθη μορφή:

(tell

```
:sender stock-server
```

```

:content "price(ibm, (14, 'now'))"

:receiver joe

:in-reply-to ibm-stock

:language standard_prolog

:ontology NYSE-TICKS)

```

5.5 Πρότυπα Πρωτόκολλου FIFA

Καθώς, σύμφωνα με τα όσα έχουν περιγραφεί μέχρι τώρα, οι πράκτορες είναι κοινωνικοί, ο σχεδιαστής ενός συστήματος πρακτόρων πρέπει να σχεδιάσει τους πράκτορες ώστε να είναι ενήμεροι για τα νοήματα των μηνυμάτων καθώς και τους στόχους, τις προτιμήσεις και τα άλλα νοητικά στοιχεία που τους χαρακτηρίζουν. Αυτό έχει ως αποτέλεσμα ο μηχανισμός σχεδιασμού ενεργειών τους να προκαλεί την ανάκληση τέτοιων πρωτοκόλλων αυθόρμητα χωρίς την κατανάλωση πόρων και χρόνου για τη σύνθεσή τους. Σε αυτή την κατεύθυνση, η FIPA (Foundation for Intelligent Physical Agents) [7] έχει διατυπώσει μία σειρά προτύπων πρωτοκόλλων επικοινωνίας μεταξύ των ευφυών πρακτόρων βασισμένων στη θεωρία ενεργειών λόγου. Έτσι, ακόμα και απλοί σε υλοποίηση πράκτορες θα μπορούν να εμπλακούν σε συζητήσεις με νόημα με άλλους πράκτορες, απλά ακολουθώντας το κατάλληλο πρωτόκολλο.[13]

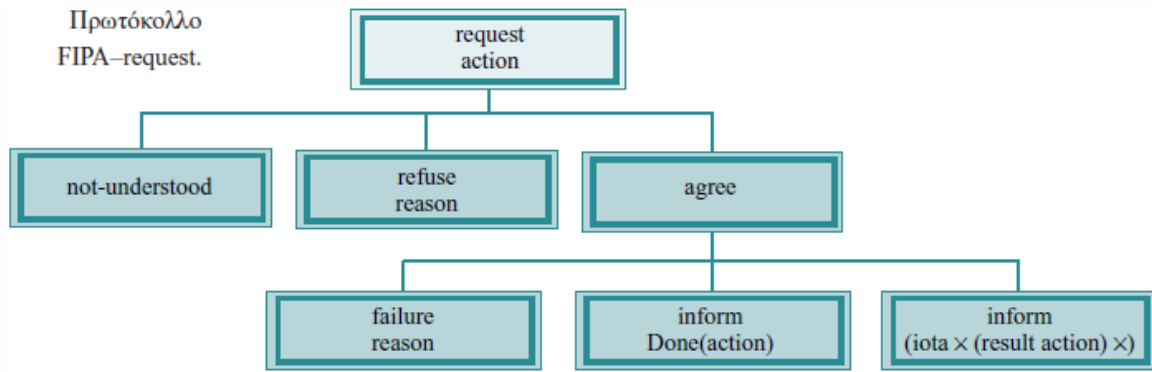
Τα πρωτόκολλα αναφέρονται με συγκεκριμένα ονόματα ανάλογα με τις υπηρεσίες που εξυπηρετούν όπως FIPA-request, FIPA-request-when, FIPA-query κ.ά. Στη συνέχεια θα δοθεί μία σύντομη περιγραφή των παραπάνω στοιχειωδών πρωτοκόλλων, καθώς επίσης και μία παραλλαγή του ευρέως γνωστού Contract Net Protocol. Σημειώνεται ότι η περιγραφή αφορά την άποψη του εξωτερικού παρατηρητή, καθώς η αναπαράσταση των εσωτερικών ενεργειών των πρακτόρων για την εκτέλεση των πρωτοκόλλων θα προκαλούσε περιορισμούς στην υλοποίηση των πρακτόρων.

Για την περιγραφή της αλληλεπίδρασης χρησιμοποιείται ο παρακάτω συμβολισμός:

- Πλαίσια με διπλή γραμμή αναπαριστούν ενέργειες επικοινωνίας.
 - Λευκά πλαίσια αναπαριστούν ενέργειες του αποστολέα.
 - Σκιασμένα πλαίσια αναπαριστούν ενέργειες του παραλήπτη.
- Πλαίσια με διακεκομμένη γραμμή αναπαριστούν στιγμιότυπα του χρόνου στα
- οποία κάποια συνθήκη είναι αληθής.
- Κείμενο με πλάγια γράμματα (*italics*) χωρίς πλαίσιο αποτελούν σχόλια.

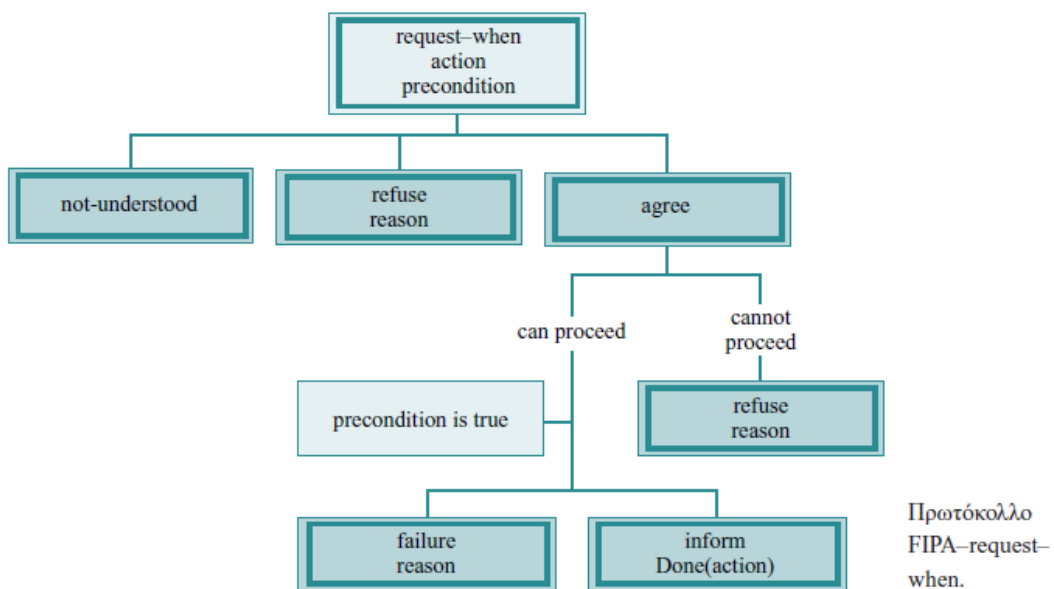
Πρωτόκολλο FIPA-request

Σύμφωνα με το πρωτόκολλο FIPA-request, ένας πράκτορας ζητάει μία ενέργεια από κάποιον άλλο πράκτορα και ο δεύτερος είτε αρνείται να εκτελέσει τη ζητούμενη ενέργεια και αιτιολογεί την άρνησή του είτε αποδέχεται να εκτελέσει την ενέργεια και μετά το πέρας της ενέργειας ενημερώνει για τον τερματισμό της και τα αποτελέσματά της. Επίσης, περιλαμβάνεται και η περίπτωση που για οποιοδήποτε λόγο το μήνυμα δεν είναι κατανοητό (όπως για παράδειγμα δεν έχει παραληφθεί ολόκληρο).



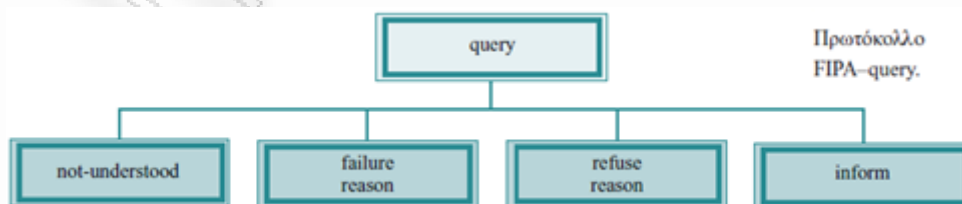
Πρωτόκολλο FIPA-request-when

Αυτό το πρωτόκολλο χρησιμοποιείται, όταν ο αποστολέας ζητάει κάποια ενέργεια από τον παραλήπτη, η οποία πραγματοποιείται κάποια στιγμή στο μέλλον, όταν μία δεδομένη συνθήκη γίνεται αληθής. Αν ο παραλήπτης καταλάβει το μήνυμα και δεν αρνηθεί την ζητούμενη ενέργεια, τότε περιμένει την πραγματοποίηση της συνθήκης για να εκτελέσει την ενέργεια και κατόπιν ειδοποιεί τον αποστολέα για την επιτυχία ή όχι της ενέργειας.



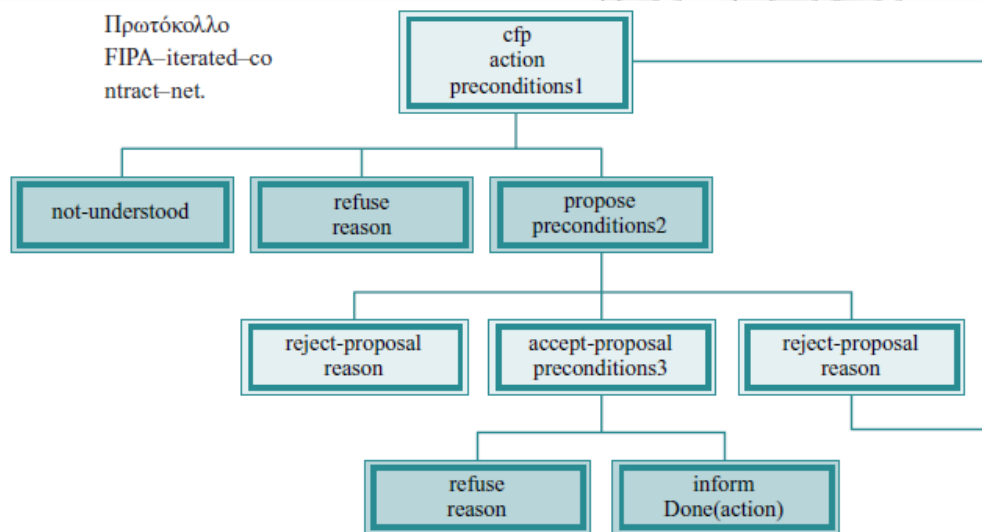
Πρωτόκολλο FIPA-query

Με το πρωτόκολλο FIPA-query, ο αποστολέας ζητάει από τον παραλήπτη να εκτελέσει μία ενέργεια πληροφόρησης. Ουσιαστικά, του ζητάει κάποια στοιχεία που του χρειάζονται και κάτι τέτοιο πραγματοποιείται με κάποιες εσωτερικές ενέργειες του παραλήπτη, όπως ανάκληση από τη βάση δεδομένων, τη βάση γνώσης ή τη νοητική του κατάσταση.



Πρωτόκολλο FIPA–contract–net

Το πρωτόκολλο FIPA–contract net αποτελεί μία μικρή τροποποίηση του γνωστού πρωτοκόλλου Contract Net. Ο αποστολέας, που στην ορολογία του πρωτοκόλλου καλείται manager, χρειάζεται μία υπηρεσία από κάποιον ή κάποιους πράκτορες και γι' αυτό κάνει μία ανακοίνωση για προτάσεις. Η ενέργεια αυτή καλείται call for proposals (cfp) και περιέχει κάποιες προϋποθέσεις για την προσφορά της υπηρεσίας, όπως για παράδειγμα κάποιο χρονικό όριο. Οι αποδέκτες του μηνύματος (καλούνται contractors) απαντούν στέλνοντας τις προτάσεις τους με τη μορφή προσφοράς (bid) ή εναλλακτικά αρνούνται την πρόταση. Όταν ο αποστολέας (manager) λάβει τις προσφορές, τις εκτιμά και επιλέγει αυτόν ή αυτούς που θα εκτελέσουν την ενέργεια. Επιπλέον, ειδοποιεί όσους έχουν απορριφθεί. Μετά το πέρας της ζητούμενης ενέργειας οι αποδέκτες ειδοποιούν τον αποστολέα για τα αποτελέσματα. Για να αποφευχθεί η περίπτωση ο αποστολέας να περιμένει απεριόριστα για προσφορές, η αίτηση για ενέργεια περιέχει και ένα χρονικό όριο για την αποστολή προσφορών. Όσες προσφορές φτάσουν μετά από αυτό το χρονικό όριο, απορρίπτονται ως εκπρόθεσμες.



5.6 – Συστήματα συνεργαζόμενων Ευφυών Πρακτόρων

Η δυνατότητα συνεργασίας (cooperation) μεταξύ των ευφυών πρακτόρων είναι ένα βασικό χαρακτηριστικό που πηγάζει από τους βασικούς στόχους της Κατανεμημένης Τεχνητής Νοημοσύνης (Distributed Artificial Intelligence) και μπορεί να συνοψισθεί ως «η δημιουργία ενός συστήματος που διασυνδέει ξεχωριστά αναπτυγμένους συνεργαζόμενους πράκτορες, καθιστώντας έτσι το σύνολο ικανό να λειτουργεί πέρα από τις ικανότητες του καθενός από τα μέλη του». Επιπλέον, συνεργαζόμενοι πράκτορες σκοπεύουν στα εξής:

Επίλυση προβλημάτων που είναι πολύ μεγάλα για έναν απλό, κεντρικό πράκτορα λόγω των περιορισμένων του πόρων.

- Διασύνδεση και διαλειτουργικότητα (interoperability) των υπαρχόντων συστημάτων (legacy systems), όπως έμπειρα συστήματα, συστήματα στήριξης αποφάσεων, συμβατικά προγράμματα.
- Παροχή λύσεων σε εγγενή κατανεμημένα προβλήματα όπως κατανεμημένες πηγές πληροφοριών, παροχή ιατρικής φροντίδας, έλεγχος εναέριας κυκλοφορίας, κτλ.

Στη συνέχεια παρουσιάζονται αντιπροσωπευτικές αρχιτεκτονικές τέτοιων συστημάτων και ένα γενικά αποδεκτό πλαίσιο διαλειτουργίας ευφυών πρακτόρων.

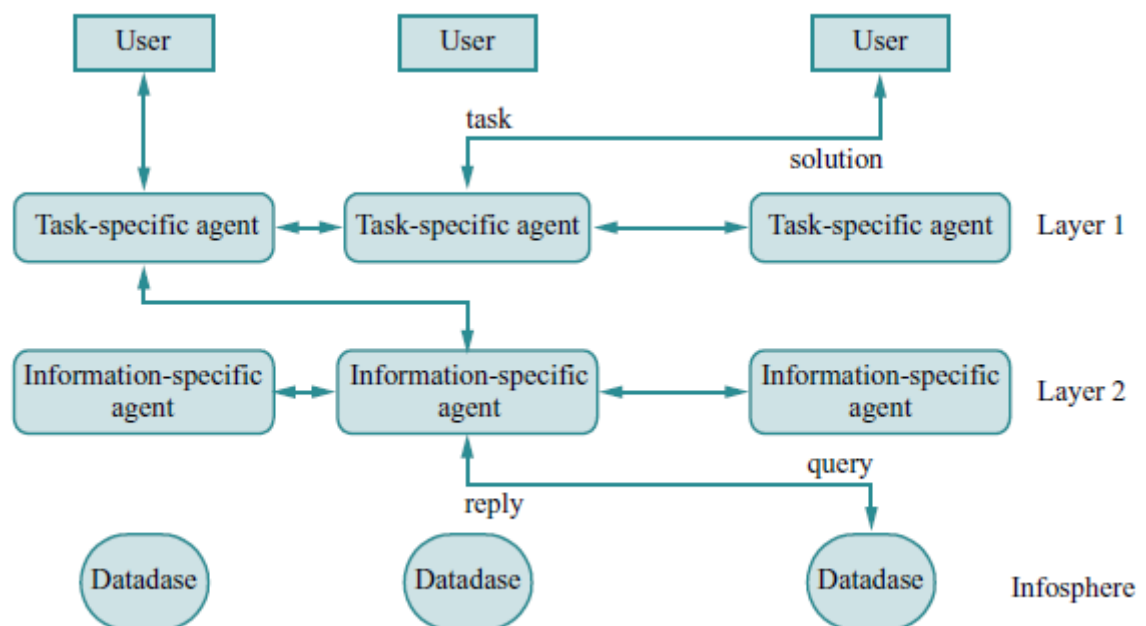
5.7 - Συστήματα συνεργαζόμενων Ευφυών Πρακτόρων

Ένα από τα πρώτα συστήματα Καταμεμημένης Τεχνητής Νοημοσύνης, που περιλάμβανε συνεργαζόμενους ευφείς πράκτορες, είναι το Multi-Agent Computing Environment (MACE) [6]. Ένα σύστημα MACE αποτελείται από:

- Ένα σύνολο από πράκτορες εφαρμογής (application agents), οι οποίοι είναι οι βασικές υπολογιστικές μονάδες σε ένα σύστημα MACE.
- Ένα σύνολο από προκαθορισμένους πράκτορες συστήματος (system agents), οι οποίοι παρέχουν υπηρεσίες στους χρήστες π.χ. user interface.
- Ένα σύνολο εργαλείων διαθέσιμων σε όλους τους πράκτορες, π.χ. pattern matcher.
- Μία βάση περιγραφών (description database), η οποία διατηρεί τις περιγραφές των πρακτόρων και παράγει εκτελέσιμους πράκτορες από αυτές τις περιγραφές.
- Ένα σύνολο από πυρήνες (kernels), έναν για κάθε φυσική μηχανή, που διαχειρίζονται την επικοινωνία, την κατανομή των μηνυμάτων, κτλ.

Οι πράκτορες στο MACE έχουν τρία χαρακτηριστικά: α) περιέχουν γνώση, η οποία αφενός είναι ειδική και τοπική και αφετέρου είναι γνώση σχετική με τους άλλους πράκτορες (acquaintance knowledge), β) αντιλαμβάνονται το περιβάλλον και γ) εκτελούν ενέργειες. Οι πράκτορες στο MACE έχουν ένα όνομα (name) και είναι οργανωμένοι σε δομημένες κλάσεις (classes). Μέσα στην κλάση κάθε πράκτορας έχει συγκεκριμένο ρόλο (role) και χαρακτηρίζεται από κάποιες ικανότητες (skills). Επιπλέον, έχει στόχους (goals), δηλαδή γνωρίζει τι θέλει να πετύχει, και πλάνα (plans), δηλαδή τρόπους για να πετύχει τους στόχους του. Η γνώση που έχει κάθε πράκτορας για τους υπόλοιπους περιλαμβάνει τα παραπάνω στοιχεία. Όλοι οι πράκτορες επικοινωνούν με μηνύματα που στέλνουν είτε σε κάποιον άλλο πράκτορα είτε σε μία ομάδα είτε σε όλους. Με το MACE έχουν υλοποιηθεί πολλά πειραματικά συστήματα καθώς και ένα σύστημα βασισμένο στο πρωτόκολλο Contract Net.

Στο project Pleiades εφαρμόζεται μία αρχιτεκτονική καταμεμημένων συνεργαζόμενων πρακτόρων προκειμένου να δημιουργηθεί και να γίνει διαχείριση του προγράμματος των επισκεπτών του Πανεπιστημίου Carnegie Mellon [60]. Οι πράκτορες είναι καταμεμημένοι σε δύο επίπεδα



Στο πρώτο επίπεδο βρίσκονται οι πράκτορες έργου (task-specific agents), οι οποίοι εκτελούν συγκεκριμένες εργασίες για τους χρήστες όπως για παράδειγμα να κανονίζουν τα ραντεβού. Αυτοί οι πράκτορες περιέχουν ένα μοντέλο της περιοχής του έργου που έχουν να εκτελέσουν, καθώς και τη γνώση για να το εκτελέσουν. Επιπλέον, περιέχουν ένα μοντέλο άντλησης πληροφοριών σχετικά με τις ικανότητες των υπολοίπων πρακτόρων και μηχανισμούς μάθησης. Οι πράκτορες έργου συνεργάζονται μεταξύ τους για να βρουν μία κοινά αποδεκτή λύση αναλύοντας πιθανές συγκρούσεις του προγράμματος και συνθέτοντας τις πληροφορίες που έχουν.

Τις απαιτούμενες πληροφορίες τις ζητούν από τους πράκτορες πληροφορίας (information specific agents) του δεύτερου επιπέδου. Οι πράκτορες πληροφορίας έχουν πρόσβαση σε βάσεις δεδομένων (καλούνται infosphere), σε γνώση σχετική με τις πηγές πληροφοριών και σε ένα μοντέλο για τις ικανότητες των υπολοίπων πρακτόρων του ίδιου επιπέδου. Επίσης, συνεργάζονται μεταξύ τους για να συνθέσουν την πληροφορία που τους ζητείται από τους πράκτορες του πρώτου επιπέδου. Χαρακτηριστικό είναι ότι στην αρχιτεκτονική Pleiades δεν εφαρμόζεται πουθενά κεντρικός έλεγχος και τα αποτελέσματα (δηλαδή το πρόγραμμα του επισκέπτη) παράγονται αποκλειστικά μέσα από τη συνεργασία των πρακτόρων των δύο επιπέδων. Αρχιτεκτονική συνεργαζόμενων πρακτόρων, όμως, με μερικό κεντρικό έλεγχο εφαρμόζεται και στο σύστημα Dermatology Tutor [60].

Ο Dermatology Tutor, είναι ευφυές εκπαιδευτικό σύστημα (Intelligent Tutoring System– ITS), με σκοπό τη διδασκαλία θεμάτων Δερματολογίας. Χρησιμοποιεί κοινωνίες από αυτόνομους πράκτορες, που ο καθένας έχει διαφορετικές ικανότητες και ρόλους και όλοι μαζί συνδιαλέγονται και συνεργάζονται προκειμένου να διδαχθεί ένα θέμα. Μία κοινωνία αποτελείται από τον εκπαιδευτικό πράκτορα (tutor agent) που εφαρμόζει μία γενικά αποδεκτή εκπαιδευτική στρατηγική και συντονίζει τη διαδικασία διδασκαλίας, τους ιατρικούς πράκτορες (medical agents) που είναι ειδικοί σε κάποιο ιατρικό πεδίο, και τους πράκτορες πληροφορίας (information agents), που χρησιμοποιούνται για εύρεση και ανάκληση πληροφοριών. Ανάλογα με το θέμα που πρόκειται να διδαχθεί, ο εκπαιδευτικός πράκτορας συνθέτει ομάδες από ιατρικούς πράκτορες οι οποίοι με τη σειρά τους ζητούν συγκεκριμένες πληροφορίες από τους πράκτορες πληροφορίας.

Η συγκεκριμένη αρχιτεκτονική είναι ανοιχτή. Αν και όπως παρουσιάζεται, έχει υλοποιηθεί για διδασκαλία θεμάτων Δερματολογίας, μπορεί να εφαρμοστεί και σε άλλα πεδία χωρίς δομικές αλλαγές παρά μόνο με την εισαγωγή ή την αφαίρεση των κατάλληλων πρακτόρων. Οι πράκτορες που περιέχει το σύστημα αρχικά δεν είναι οργανωμένοι σε ομάδες. Αντίθετα, οι ομάδες συνθέτονται όταν πρόκειται να επιλυθεί κάποιο πρόβλημα, δηλαδή να παρουσιαστεί κάποιο θέμα προς διδασκαλία. Κάθε πράκτορας έχει γνώση των ικανοτήτων του και κάποια γνώση των υπολοίπων. Ανάλογα με τις ικανότητες που έχει ο καθένας γίνεται η σύνθεση ομάδων και οι ανάθεση ρόλων. Συγκεκριμένα τώρα, το σύστημα Dermatology Tutor αποτελείται από εννέα πράκτορες τοποθετημένους σε δύο επίπεδα.

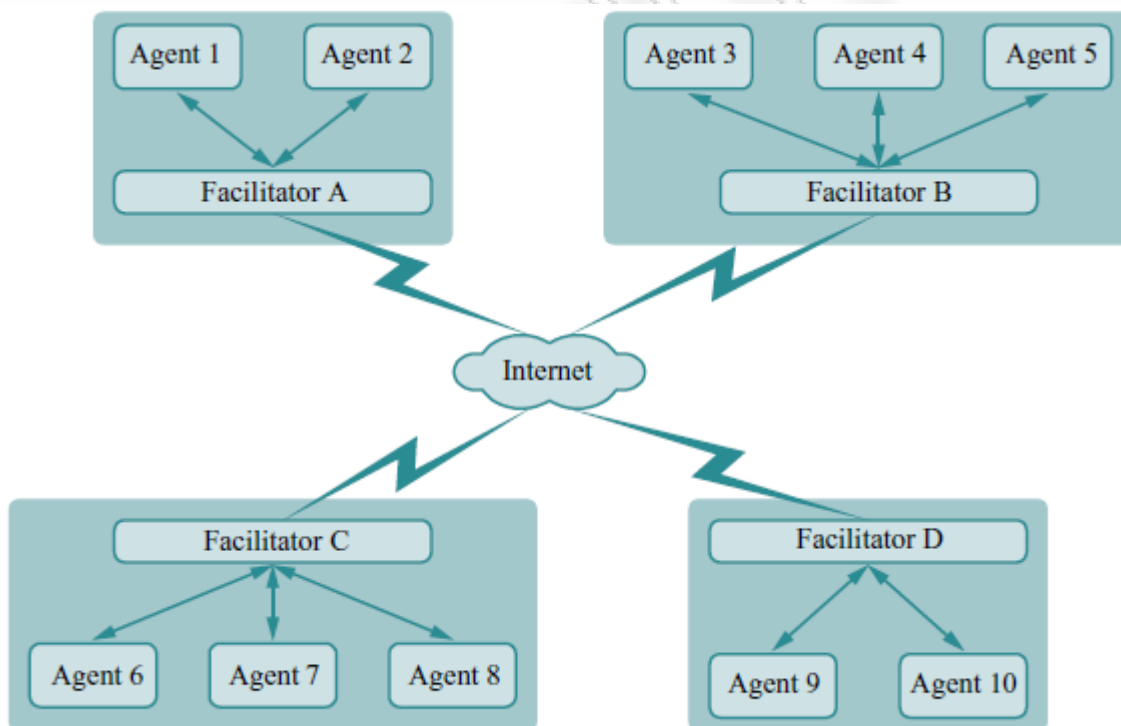
Ο χρήστης αλληλεπιδρά με το σύστημα μέσω του συστήματος διεπαφής (user interface) το οποίο μεταφέρει όλες τις αιτήσεις ενεργειών ή υπηρεσιών του περιβάλλοντος εκτός του συστήματος. Ανάλογα, ενεργοποιείται ο Dermatology Agent ή ο Help Agent, οι οποίοι βρίσκονται στο πρώτο επίπεδο. Αν διαπιστώσουν πως χρειάζεται ομαδική δράση για την ικανοποίηση της αίτησης, τότε συνθέτουν ομάδες με τους ιατρικούς πράκτορες που βρίσκονται στο δεύτερο επίπεδο. Ωστόσο, μόνο ο Dermatology Agent ενεργεί ως εκπαιδευτής και αυτός έχει γνώση των εκπαιδευτικών στρατηγικών και του τρόπου διεξαγωγής τους.

Οι ιατρικοί πράκτορες μπορούν να είναι μέλη των ομάδων που συνθέτει ο Dermatology Agent και μπορούν επίσης να επικοινωνούν μεταξύ τους. Οι επτά ιατρικοί πράκτορες Histology, Physiology, Biology, Biochemistry, Microbiology, Radiology και Immunology Agent είναι ειδικοί στα αντίστοιχα ιατρικά θέματα και μπορούν να παρέχουν τη σχετική γνώση και πληροφορία. Επιπλέον, μπορούν να ζητήσουν πληροφορίες από άλλους πράκτορες (πράκτορες πληροφορίας), οι οποίοι μπορούν να είναι τοπικοί ή απομακρυσμένοι και ο ρόλος τους είναι η αναζήτηση και ανάκληση πληροφοριών ή να αναζητήσουν μόνοι τους πληροφορίες σε γνωστές τοπικές βάσεις δεδομένων.

5.8 – Ομοσπονδιακοί αρχιτέκτονες

Η αλληλεπίδραση και διαλειτουργικότητα είναι ένα σημαντικό θέμα για την λειτουργία και συνεργασία μεταξύ ετερογενών συστημάτων. Το βασικό πρόβλημα είναι η ανομοιογένεια στον τρόπο υλοποίησης και στις γλώσσες επικοινωνίας, τόσο ως προς τη σύνταξη όσο και ως προς τη σημασιολογία. Βέβαια, έχουν γίνει προσπάθειες για κοινά πρωτόκολλα και γλώσσες επικοινωνίας, όπως περιγράφονται στην προηγούμενη ενότητα, ωστόσο υπάρχει η ανάγκη να καλυφθούν και τα ήδη υπάρχοντα συστήματα καθώς και πιθανόν νέα. Σε αυτήν την κατεύθυνση, είναι απαραίτητες και οι κατάλληλες αρχιτεκτονικές, που σε συνδυασμό με τα κοινά πρωτόκολλα και γλώσσες επικοινωνίας, καθιστούν τα συστήματα σε ευρεία κλίμακα ικανά να κατανοούν την ανταλλασσόμενη πληροφορία και να μπορούν να συνεργαστούν μεταξύ τους.

Οι πιο γνωστές αρχιτεκτονικές για αυτόν τον σκοπό, που αποτελούν ήδη ένα διεθνές πρότυπο, είναι οι ομοσπονδιακές αρχιτεκτονικές (federated architectures). Ένα σύστημα αποτελείται από πράκτορες και διευκολυντές (facilitators). Οι πράκτορες δεν επικοινωνούν απευθείας μεταξύ τους αλλά μέσω των διευκολυντών. Οι διευκολυντές είναι πράκτορες που γνωρίζουν την ύπαρξη και τις ικανότητες των υπολοίπων πρακτόρων, συντονίζουν τις ενέργειες αυτών και κρατούν ενήμερους τους άλλους διευκολυντές του δικτύου σχετικά με ποιοι πράκτορες είναι συνδεδεμένοι. Η επικοινωνία σε ένα τέτοιο σύστημα γίνεται μέσω των διευκολυντών.



Σύμφωνα με τον Genesereth, που έχει προτείνει την ομοσπονδιακή αρχιτεκτονική, οι διευκολυντές παρέχουν ένα σύνολο υπηρεσιών όπως οι επόμενες:

Λευκές Σελίδες (White Pages), όπου παρέχεται η ανεύρεση της ταυτότητας των πρακτόρων σύμφωνα με το όνομα, δηλαδή «ποιοι πράκτορες είναι συνδεδεμένοι;» ή «είναι ο τάδε πράκτορας συνδεδεμένος;».

Κίτρινες Σελίδες (Yellow Pages), όπου παρέχεται η ανεύρεση της ταυτότητας των πρακτόρων σύμφωνα με τις ικανότητες να εκτελέσουν ένα έργο, π.χ. «ποιοι πράκτορες μπορούν να απαντήσουν στην ερώτηση x;».

Δρομολόγηση σύμφωνα με το Περιεχόμενο (Content-based Routing), όπου ο διευκολυντής έχει την ευθύνη να διαχειριστεί μία αίτηση βασιζόμενος στα χαρακτηριστικά και τις υπόλοιπες πληροφορίες που του έχουν δοθεί από τους πράκτορες σχετικά με τις ικανότητές τους.

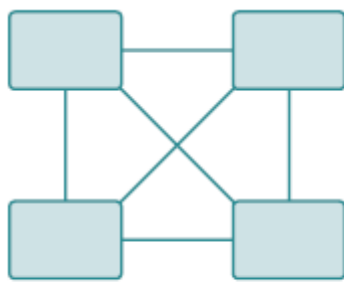
Απευθείας Επικοινωνία (Direct Communication), με αποστολή μηνύματος σε ένα συγκεκριμένο πράκτορα.

Μετάφραση (Translation), καθώς οι πράκτορες μπορεί να χρησιμοποιούν διαφορετικό λεξιλόγιο. Έτσι, ο διευκολυντής ίσως χρειαστεί να μεταφράσει το μήνυμα του αποστολέα σε γλώσσα κατανοητή από τον αποδέκτη.

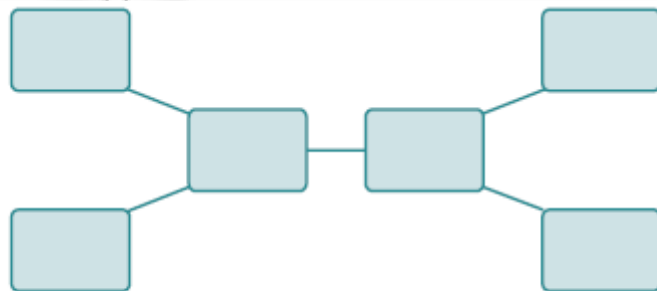
Αποσύνθεση Προβλήματος (Problem Decomposition), όπου μία σύνθετη αίτηση μπορεί να αποσυντεθεί σε υπο-προβλήματα, να παραληφθούν οι απαντήσεις των υποπροβλημάτων και κατόπιν να γίνει η σύνθεση της απάντησης στην αρχική αίτηση. Όπως στη Δρομολόγηση σύμφωνα με το Περιεχόμενο, έτσι και εδώ ο διευκολυντής βασίζεται στις πληροφορίες που του έχουν δοθεί.

Παρακολούθηση (Monitoring), με την έννοια ότι, αν ο διευκολυντής δεχτεί μία αίτηση για κάποια ανάγκη ενός πράκτορα, τότε παρακολουθεί ή εξετάζει τη γνώση του για να διαπιστώσει αν η συγκεκριμένη ανάγκη μπορεί να ικανοποιηθεί, π.χ αν κάποιος πράκτορας του ζητήσει «Ενδιαφέρομαι για στοιχεία σχετικά με τη θέση των chips στο τάδε σχέδιο», ο διευκολυντής εξετάζει αν μπορεί του δώσει αυτά τα στοιχεία.

Όπως έχει περιγραφεί παραπάνω, οι πράκτορες συνδέονται με τους διευκολυντές και οι διευκολυντές μεταξύ τους. Για πρακτικούς λόγους, στην ομοσπονδιακή αρχιτεκτονική, θεωρείται ότι κάθε πράκτορας συνδέεται με έναν μόνο διευκολυντή. Ωστόσο, η σύνδεση των διευκολυντών ποικίλει. Στο παρακάτω σχήμα παρουσιάζονται δύο βασικοί τρόποι διασύνδεσης διευκολυντών.



(α) Πλήρης διασύνδεση.



(β) Διασύνδεση spanning tree.

Στην πλήρη διασύνδεση (full interconnection), κάθε διευκολυντής συνδέεται, σε λογικό επίπεδο και όχι σε φυσικό, με όλους τους υπόλοιπους. Μία τέτοια σύνδεση είναι εφικτή, ωστόσο έχει το μειονέκτημα του κόστους, με την έννοια ότι σε ένα δίκτυο μεγάλης κλίμακας όπως το Διαδίκτυο, ο αριθμός των διευκολυντών είναι πολύ μεγάλος και θα πρέπει ο καθένας να γνωρίζει σχετικά για όλους τους υπόλοιπους.

Μία εναλλακτική αρχιτεκτονική διασύνδεσης διευκολυντών είναι η αρχιτεκτονική spanning tree όπου οι διευκολυντές συνδέονται μεταξύ τους με τέτοιο τρόπο ώστε να υπάρχει μονοπάτι (path) από κάθε

διευκολυντή σε κάθε άλλο διευκολυντή, χωρίς όμως να υπάρχουν βρόχοι (loops). Αυτή η προσέγγιση είναι καλή από την άποψη ότι αποφεύγει το κόστος του μεγάλου αριθμού συνδέσεων.

Έχει όμως το μειονέκτημα να είναι ευπαθής σε αποτυχίες (failures), καθώς αν κάποιος ή κάποιοι διευκολυντές βγουν εκτός λειτουργίας, τότε μεγάλο μέρος του δικτύου αποσυνδέεται. Τέλος, μία ενδιάμεση προσέγγιση είναι η αρχιτεκτονική γενικής διασύνδεσης (general connectivity architecture), όπου κάθε διευκολυντής συνδέεται τουλάχιστον έμμεσα με κάθε άλλο διευκολυντή, όπως στην αρχιτεκτονική spanning tree, αλλά δεν υπάρχει ο περιορισμός της μη ύπαρξης βρόχων. Η συγκεκριμένη αρχιτεκτονική έχει τα πλεονεκτήματα των άλλων δύο, δηλαδή είναι οικονομική στο κόστος διασύνδεσης και ανθεκτική σε αποτυχίες κάποιων διευκολυντών. Ωστόσο έχει το μειονέκτημα πιθανών βρόχων.

Για παράδειγμα, μπορεί ένας διευκολυντής έστω ο F1 να στείλει ένα μήνυμα στον F2, αυτός με τη σειρά του να το στείλει σε κάποιον διευκολυντή F3 και τέλος ο F3 να το στείλει πίσω στον F1. Βέβαια, υπάρχουν τρόποι να ξεπεραστεί αυτό το μειονέκτημα χρησιμοποιώντας πρόσθετη πληροφορία στο μήνυμα σχετικά με τον αποστολέα. Αυτό έχει αποτέλεσμα λίγο υψηλότερο προγραμματιστικό κόστος, όμως η προσέγγιση είναι επαρκής και αξιόπιστη.

Συμπεράσματα

Η συνεχώς αυξανόμενη χρήση του όρου διαμεσολαβητής έχει προκαλέσει μεγάλη συζήτηση στην επιστημονική κοινότητα σχετικά με τον ορισμό του. Ένας ασθενής αλλά ικανοποιητικός ορισμός που περιλαμβάνει τις γενικά αποδεκτές ιδιότητες που πρέπει να έχει ένας διαμεσολαβητής, τον χαρακτηρίζει ως ένα σύστημα που πρέπει να παρουσιάζει αυτονομία, κοινωνικότητα, αποκρισμότητα και ανάληψη πρωτοβουλιών. Ωστόσο, ένας ορισμός δεν αρκεί για να περιγραφεί η συμπεριφορά ενός διαμεσολαβητή. Σε αυτήν την κατεύθυνση έχουν χρησιμοποιηθεί μέθοδοι που βασίζονται σε φιλοσοφικές θέσεις και χρησιμοποιούν έννοιες που αποδίδονται κυρίως σε ανθρώπους όπως πεποιθήσεις, επιθυμίες, προθέσεις δεσμεύσεις, ικανότητες, επιλογές, συναισθήματα κ.ά. Μία τέτοια προσέγγιση είναι αποδεκτή και θεωρείται θεμιτή και χρήσιμη εφόσον τηρούνται κάποιες προϋποθέσεις.

Οι αρχιτεκτονικές στις οποίες στηρίζεται η κατασκευή των ευφυών πρακτόρων χωρίζονται σε δύο κατηγορίες. Η πρώτη κατηγορία (αρχιτεκτονικές νοημοσύνης) χρησιμοποιεί συμβολική αναπαράσταση και ο λογισμός, η επίλυση προβλημάτων, ο σχεδιασμός ενεργειών, η μάθηση, κτλ. πραγματοποιούνται με μηχανισμούς διαχείρισης συμβόλων. Όμως, επειδή μέχρι πρόσφατα αυτά τα θέματα αντιμετωπιζόταν μεμονωμένα και δεν απέδωσαν τα αποτελέσματα των μακρόπνων σχεδίων της Τεχνητής Νοημοσύνης, δημιουργήθηκε μία νέα κατηγορία αρχιτεκτονικών (αρχιτεκτονικές αντίδρασης) που δεν κάνει χρήση συμβολικής αναπαράστασης, αλλά βασίζεται στη σύνθεση πολλών στοιχειωδών συμπεριφορών. Έτσι, θεωρείται ότι η ευφυία είναι αποτέλεσμα της αλληλεπίδρασης των στοιχειωδών συμπεριφορών. Τα καλύτερα χαρακτηριστικά των δύο παραπάνω κατηγοριών προσπαθεί να εκμεταλλευτεί μία Τρίτη κατηγορία (υβριδικές αρχιτεκτονικές), όπου σε ένα πολυεπίπεδο σχήμα συνδυάζει στοιχειώδεις συμπεριφορές και διαχείριση συμβολικής αναπαράστασης. Οι ευφείς πράκτορες έχουν να επιδείξουν μία μεγάλη κατηγορία εφαρμογών που συνεχώς αυξάνεται, καθώς και μία έντονη ερευνητική δραστηριότητα. Πολλά εμπορικά συστήματα πρακτόρων βρίσκουν εφαρμογή στο Διαδίκτυο και επικεντρώνονται κυρίως στη διαχείριση πληροφοριών (συλλογή, φιλτράρισμα, κτλ.) και το ηλεκτρονικό εμπόριο. Επίσης, συστήματα πρακτόρων λειτουργούν σε εφαρμογές προσωπικών υπολογιστών ως προσωπικοί βοηθοί. Στη βιομηχανία έχουν εφαρμοστεί στις τηλεπικοινωνίες, σε συστήματα ελέγχου, στη διαχείριση πόρων, στη ρομποτική κ.ά. Οι εφαρμογές επεκτείνονται στον Ιατρικό τομέα, στην εκπαίδευση, ακόμα και στο χώρο του θεάματος και των παιχνιδιών.

Καταλήγουμε στο συμπέρασμα ότι η τεχνολογία συστημάτων πρακτόρων είναι ζωντανή και ταχέως εξελισσόμενη επιστήμη τόσο στην ακαδημαϊκή έρευνα όσο και στις επιχειρηματικές εφαρμογές. Το νέο πρότυπο παρέχει τον σχεδιασμό και την εφαρμογή συστημάτων για ένα πολύπλοκο, δυναμικό, περιβάλλον όπου τα γνωρίσματα είναι η διαπραγμάτευση, τα απρογραμμάτιστα γεγονότα και δράση και πρέπει να αντιμετωπισθεί με τρόπο που είναι επωφελής για τη συνολική απόδοση του συστήματος. Η τεχνολογία των ευφυών πρακτόρων συνήθως λειτουργεί είναι σε μεγάλο βαθμό σαν πανάκεια για τον τρόπο σχεδιασμού ενός ευφυούς συστήματος και την ανάπτυξη του, αλλά ο κατασκευαστής πρέπει να γνωρίζει και τις παγίδες που είναι συμβαίνουν σε κάθε νέα και μη δοκιμασμένη τεχνολογία. Οι δυνατότητες της τεχνολογίας ευφυών πρακτόρων είναι εκεί, αλλά το πλήρες όφελος δεν έχει ακόμα επιτευχθεί. Η τεχνολογία ευφυών πρακτόρων μπορεί να επιτύχει τις πραγματικές δυνατότητές της μόνο εφόσον οι χρήστες και προγραμματιστές αντιληφθούν την αξία των δραστηριοτήτων της.

Βιβλιογραφία

1. Bates, J. (1994). The role of emotion in believable agents. *Communications of the ACM*, 37(7): 122–125.
2. Baujard, O., Garbay, C. (1993). KISS, a Multi-Agent Segmentation System. Special Issue From Numeric to Symbolic Image Processing: Systems and Applications, *SPIE Optical Engineering*, 32(6): 1235–1249.
3. Brazier F. M. T., Dunin–Keplicz B. M., Jennings N. R., and Treur J.(1996): DESIRE: Modelling Multi-Agent Systems in a Compositional Formal Framework. *International Journal of Cooperative Information Systems*, 6(1): 67–94.
4. Brooks, R. A. (1986). A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23.
5. Brooks, R. A. (1991). Intelligence without representation. *Artificial Intelligence*, 47:139–159.
6. Cheikes, B. A. (1995). GIA: An Agent–Based Architecture for Intelligent Tutoring Systems. In *Proceedings of the CIKM'95 Workshop on Intelligent Information Agents*.
7. Cohen, P. R. and Levesque, H. J. (1990). Intention is Choice with Commitment. *Artificial Intelligence*, 42:213–261.
8. Cohen, P. R. and Levesque, H. J. (1990). Rational Interaction as the Basis for Communication. In Cohen, P. R., Morgan, J., and Pollack, M. E., editors, *Intentions in Communication*, pp. 221–256. The MIT Press: Cambridge, MA.
9. Cohen, P. R. and Perrault, C. R. (1979). Elements of a Plan Based Theory of Speech Acts. *Cognitive Science*, 3:177–212
10. Dennett, D. C. (1987). *The Intentional Stance*. The MIT Press: Cambridge, MA.
11. Ferguson, I. A. (1992). *Touring Machines: An Architecture for Dynamic, Rational, Mobile Agents*. Ph.D. Thesis, Clare Hall, University of Cambridge, UK
12. Finin, T., Labrou, Y., and Mayfield, J. (1997). KQML as an Agent Communication Language. In J. M. Bradshaw (ed), *Software Agents*, AAAI Press, Menlo Park, CA.
13. FIPA (1997). *Agent Communication Language. FIPA '97 Specification: Part 2, Version 1.0*, 10 October 1997.
14. Gasser, L., Braganza, C., and Hermann, N. (1987). MACE: A Flexible Testbed for Distributed AI Research. In M. Huhns, (ed.) *Distributed Artificial Intelligence*, pp. 119–152, Pitman Publishing, London. 7. FIPA (1997). *Agent Communication Language. FIPA '97 Specification: Part 2, Version 1.0*, 10 October 1997.
15. Genesereth, M. R. (1997). An Agent–Based Framework for Interoperability. In J. M. Bradshaw (ed.) *Software Agents*, AAAI Press, Menlo Park, CA.
16. Genesereth, M. R. and Ketchpel, S. P. (1994). Software agents. *Communications of the ACM*, 37(7): 48–53.

17. Georgeff, M. P., and Lansky, A. L. (1987). Reactive Reasoning and Planning. In Proceedings of the Sixth National Conference on Artificial Intelligence
18. Grosz, B., and Davis, R. (1994). A report to ARPA on twenty-first century intelligent systems, *AI Magazine* 15(3):10–20.
19. Guilfoyle, C. (1998). Vendors of Intelligent Agent Technologies: A Market Overview. In N. Jennings and M. Wooldridge (Eds.), *Agent Technology: Foundations, Applications, and Markets*, pp. 91–104, Springer.
20. Haddadi, A. (1994). A Hybrid Architecture for Multi-agent Systems. In Deen, S. M., editor, *Proceedings of the 1993 Workshop on Cooperating Knowledge Based Systems (CKBS-93)*, pp. 13–26, DAKE Centre, University of Keele, UK
21. Huang Yan, Jian Huang, Lance Hester, Anthony Allen, Oleg Andric, Priscilla Chen, Bob O’Dea, "OPNET Simulation Of A Multi-hop Self-organizing Wireless Sensor Network",.
22. Huhns, M. N., Singh, M. P. (1994). *Distributed Artificial Intelligence for Information Systems*, CKBS-94 Tutorial, University of Keele, UK.
23. Ikeda, M., and Mizoguchi, R., (1994). FITS: A Framework for ITS – a Computational Model of Tutoring. *Journal of Artificial Intelligence in Education*, 5(3), 319–348.
24. Jennings, N. R. (1993). Specification and Implementation of a Belief Desire Joint-Intention Architecture for Collaborative Problem Solving. *Journal of Intelligent and Cooperative Information Systems*, 2(3):289–318.
25. Kaelbling, L. P. (1991). A Situated Automata Approach to the Design of Embedded Agents. *SIGART Bulletin*, 2(4):85–88.
26. Karjoth, G., Lange, D. B., and Oshima, M. (1997). A Security Model for Aglets. *IEEE Internet Computing*, 1(4): 68–77.
27. Kiniry, J., and Zimmerman, D. (1997). A Hands-on Look at Java Mobile Agents. *IEEE Internet Computing*, 1(4): 21–30.
28. Konolige, K., Myers, K., Ruspini E. (1997). The Saphira Architecture: A Design for Autonomy. *Journal of Experimental and Theoretical Artificial Intelligence*, (special issue on Architectures for Physical Agents), 9: 215–235.
29. Ljungberg, M., and Lucas, A. (1992). The OASIS Air Traffic Management System. Tech. Rep. 28, Australian Artificial Intelligence Institute, Melbourne, Australia.
30. Maes P. (1997). Humanizing the Global Computer, *IEEE Internet Computing*, 1(4): 10–19.
31. Maes, P. (1991). *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back*. MIT Press, London.
32. Maes, P. (1991). The Agent Network Architecture (ANA). *SIGART Bulletin*, 2(4): 115–120.
33. Maes, P. (1994). Agents that Reduce Work and Information Overload. *Communications of the ACM*, 37 (7): 31–40.
34. McCarthy, J. (1979). Ascribing Mental Qualities to Machines. Technical Report, Memo 326, Stanford University AI Lab., Stanford, CA.

35. McCarthy, J. (1998). What is Artificial Intelligence?, Computer Science Department, Stanford University, Stanford CA 94305, (available in <http://www-formal.stanford.edu/jmc/>).
36. Grigoris Antoniou, Frank van Harmelen (March 31, 2008). A Semantic Web Primer, 2nd Edition. The MIT Press. ISBN 0-262-01242-1..
37. Minsky, M. (1985). The Society of Mind. Simon and Shuster, NY.
38. Minsky, M. (1994). A Conversation with Marvin Minsky about Agents. Communications of the ACM, 37(7): 23–29.
39. Mitchell, T., Caruana, R., Freitag, D., McDermott, J., and Zabowski, D. (1994). Experience with a Learning Personal Assistant. Communications of the ACM, 37 (7): 81–91.
40. Müller, J. P. and Pischel, M. and Theil, M. (1995). Modeling reactive Behavior in Vertically Layered Agent Architectures. In Wooldridge, M. and Jennings, N. R. (eds) Intelligent Agents: Theories, Architectures and Languages (LNAI Volume 890), pp. 261–276, Springer-Verlag: Heidelberg, Germany.
41. 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.
42. Ohsuga, A., Nagai, Y., Irie, Y., Hattori, M., and Honiden, S. (1997). PLANAGENT: An Approach to Making Mobile Agents Intelligent. IEEE Internet Computing, 1(4): 50–57.
43. Ovalle, A., Garbay, C. (1991). KIDS: a Distributed Expert System for Biomedical Image Interpretation. In Information Processing in Medical Imaging, pp. 419–433, (A.C.F. Colchester and D. J. Hawkes, eds.).
44. Parunak, H. V. D. (1987). Manufacturing Experience with the Contract Net. In M. N. Huhns (Ed.) Distributed AI, Morgan Kaufmann.
45. Petrie, C., J. (1997). What's an Agent and What's so Intelligent About it? IEEE Internet Computing, 1 (4): 4–5.
46. Searle, J. R. (1969). Speech Acts: An Essay in the Philosophy of Language. Cambridge University Press: Cambridge, UK.
47. Selker, T. (1994). COACH: A Teaching Agent that Learns. Communications of the ACM, 37(7): 92–99.
48. Shardlow, N. (1990). Action and Agency in Cognitive Science. Master's thesis, Department of Psychology, University of Manchester, UK.
49. Shoham, Y. (1989). Time for Action: On the Relation between Time, Knowledge and Action. In Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI-89), pp. 954–959, Detroit, MI.
50. Shoham, Y. (1993). Agent-oriented programming. Artificial Intelligence, 60(1): 51–92.
51. Smith, R. G. (1980). The Contract Net Protocol. IEEE Transactions on Computers, C-29 (12).
52. Steels, L. (1990). Cooperation between Distributed Agents through Self Organization. In Demazeau, Y. and Müller, J. P., editors, Decentralized AI – Proceedings of the First European Workshop on Modelling Autonomous Agents in Multi-Agent Worlds (MAAMAW-89), pp. 175–196. Elsevier Science Publishers B.V.: Amsterdam, The Netherlands.

53. Sycara, K. (1995). Intelligent Agents and the Information Revolution. UNICOM Seminar on Intelligent Agents and their Business Applications, pp. 143–159, 8–9 November, London.
54. Tidhar, G., Selvestrel, M., and Heinze, C. (1995). Modelling Teams and Team Tactics in Whole Air Mission Modelling. In the Proceedings of the Eighth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, (Gordon and Breach Science Publishers), Melbourne, Australia.
55. Weld, D. (ed.) (1997). The Role of Intelligent Systems in the National Information Infrastructure, The American Association for Artificial Intelligence, (available in <http://www.aaai.org/Policy/Papers/nii.html>).
56. Werner, E. (1989). Cooperating Agents: A Unified Theory of Communication and Social Structure. In Gasser, L. and Huhns, M., editors, Distributed Artificial Intelligence Volume II, pp. 3–36, Morgan Kaufmann Publishers Inc., San Mateo, CA.
57. Wieners, B. (1998), Shopbots: Three Degrees of Automation, *Wired*, 6(12): 72.
58. Wooldridge, M. J., and Jennings N. R. (1995). Intelligent Agents: Theory and Practice. *The Knowledge Engineering Review* 10 (2): 115–152.
59. Zaharakis I. D., Kameas A. D. and Nikiforidis G. C. (1998). A Multi-agent Architecture for Teaching Dermatology, *Medical Informatics*, 23 (4): 289–307.
60. Ζαχαράκης Ι. Ευφυή Πληροφοριακά Συστήματα και Ευρετικές Μέθοδοι. Ελληνικό Ανοικτό Πανεπιστήμιο 2001.
61. Bellifemine F., Poggi A., Rimassa G., “JADE – A FIPA-compliant agent framework”, <http://sharon.csel.it/projects/jade/papers/PAAM.pdf>