

**ΤΕΧΝΟΛΟΓΙΚΟ ΙΔΡΥΜΑ ΛΑΡΙΣΑΣ**  
**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ**  
**ΤΜΗΜΑ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

**ΕΥΦΥΗ ΣΥΣΤΗΜΑΤΑ ΛΟΓΙΣΜΙΚΟΥ**  
**(ΘΕΩΡΙΑ)**

**ΣΩΜΑΡΑΣ ΧΡΗΣΤΟΣ**

**ΙΑΝΟΥΑΡΙΟΣ 2004**

## **ΠΡΟΛΟΓΟΣ**

«ΕΝ ΟΙΔΑ ΟΤΙ ΟΥΔΕΝ ΟΙΔΑ»

Σωκράτης, Απολογία Πλάτωνος

Πάντοτε αναζητούσα τη καινούργια και πραγματική γνώση σε πράγματα τα οποία είναι ξένα και δύσκολα για πολύ κόσμο. Δε γνωρίζω αν έως τώρα πέτυχα να την αποκτήσω. Πάντως, ελπίζω το παρόν πόνημα να δώσει ερεθίσματα και να αποτελέσει βάση για επιστημονική αναζήτηση στον κόσμο της Κατανεμημένης Τεχνητής Νοημοσύνης και των Πρακτόρων.

Εγώ κρατώ τη χαρά της συγγραφής αυτών των σημειώσεων.

*Αφιερωμένο σε μια εκπαίδευση  
με ουσία και περιεχόμενο*

ΣΩΜΑΡΑΣ ΧΡΗΣΤΟΣ

4/1/2004

## **ΚΕΦΑΛΑΙΟ Α**

### **ΚΑΤΗΓΟΡΙΕΣ ΠΡΑΚΤΟΡΩΝ**

<b>i. ΓΕΝΙΚΑ ΣΤΟΙΧΕΙΑ ΓΙΑ ΠΡΑΚΤΟΡΕΣ.....</b>	<b>4</b>
<b>ii. ΠΡΑΚΤΟΡΕΣ ΛΟΓΙΣΜΙΚΟΥ .....</b>	<b>13</b>
<b>iii. ΝΟΗΜΟΝΕΣ ΠΡΑΚΤΟΡΕΣ .....</b>	<b>17</b>

## 1. Εισαγωγή

### Περιστατικό 1ο

Έστω ότι η Υπηρεσία Ελέγχου της εναέριας κυκλοφορίας του Ελ. Βενιζέλος καταρρέει κάποια στιγμή λόγω κακών καιρικών συνθηκών. Για να αντιμετωπισθεί η επικοινωνία και ο έλεγχος των αεροπλάνων, ώστε αυτά να προσγειωθούν ασφαλώς, γίνεται το εξής:

Οι πύργοι ελέγχου των γειτονικών αεροδρομίων της χώρας, μέσω ενός «έξυπνου συστήματος», επικοινωνούν και διαπραγματεύονται μεταξύ τους, αναλαμβάνοντας τελικά ο καθένας, τον έλεγχο και την προσγείωση των πιο κοντινών σε αυτούς αεροσκαφών.

### Περιστατικό 2ο

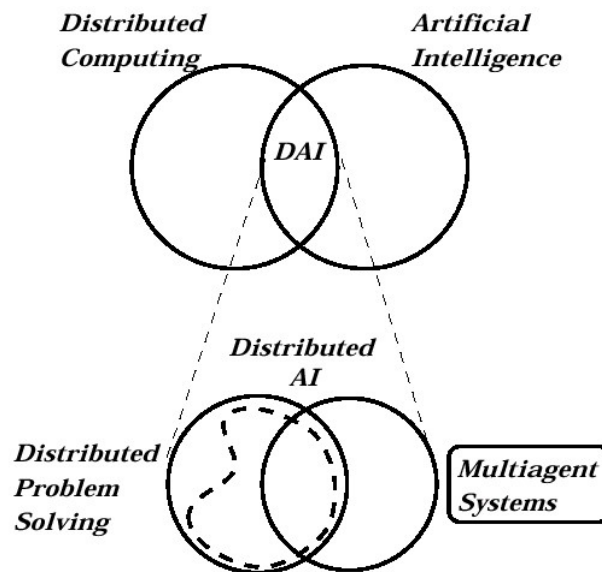
Καθώς βρίσκεστε μπροστά στον Η/Υ λαμβάνετε ένα σημαντικό email. Ανοίγοντάς το, ενημερώνεστε ότι έχει γίνει αποδεκτή η συμμετοχή σας σε ένα συνέδριο, για να παρουσιάσετε μια εργασία σας και αυτόματα σας προτείνονται οι οικονομικά καλύτερες λύσεις τόσο για τη μετακίνηση σας, όσο και για τη διαμονή στον τόπο του συνεδρίου.

### Περιστατικό 3ο

Ανοίγοντας τον Η/Υ λαμβάνετε ένα email, το οποίο σας ενημερώνει για ένα άρθρο παρόμοιο με αυτό που γράφετε και ταυτόχρονα σας παρέχει μια λίστα με πιθανές παραπομπές, για τις έννοιες που συμπεριλαμβάνονται στη συγγραφική σας δουλειά.

Όλα τα παραπάνω είναι πραγματικότητα εξαιτίας της έρευνας στον τομέα της Τεχνητής Νοημοσύνης (ειδικότερα στη **Κατανεμημένη Τεχνητή Νοημοσύνη - KTN**). Η βασική έννοια της συγκεκριμένης έρευνας είναι ο **πράκτορας**. Η **Κατανεμημένη Τεχνητή Νοημοσύνη**, είναι τμήμα της Επιστήμης των Υπολογιστών και έχει ως στόχο τη δημιουργία αντικειμένων που διαθέτουν νοημοσύνη στη συμπεριφορά τους (πχ οι πράκτορες).

**Σχήμα 1: Τοποθέτηση του αντικειμένου έρευνας της Κ Τ Ν**

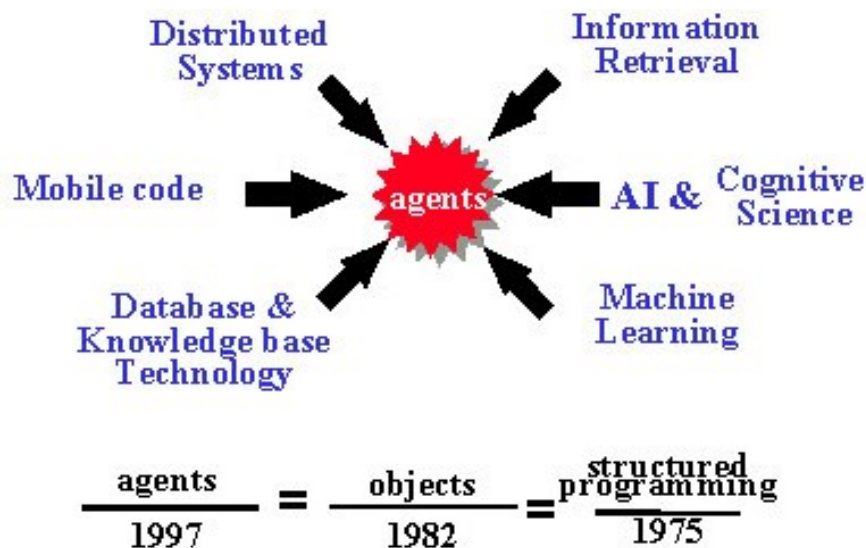


Πηγή: Stone P., Veloso M., 1997

Η έρευνα στους πράκτορες άρχισε να ανθίζει μετά το 1980. Σήμερα, υπάρχει μελέτη και ενσωμάτωση τους για την αντιμετώπιση προβλημάτων σε διάφορους τομείς, όπως:

- ❖ στην παραδοσιακή επιστήμη των υπολογιστών
- ❖ στις επικοινωνίες δεδομένων
- ❖ στη ρομποτική
- ❖ στον σχεδιασμό ΓΠΔ (γραφικού περιβάλλοντος διεπαφής του Η/Υ με το χρήστη)
- ❖ στα παράλληλα συστήματα

**Σχήμα 2: Έρευνα και πράκτορες**

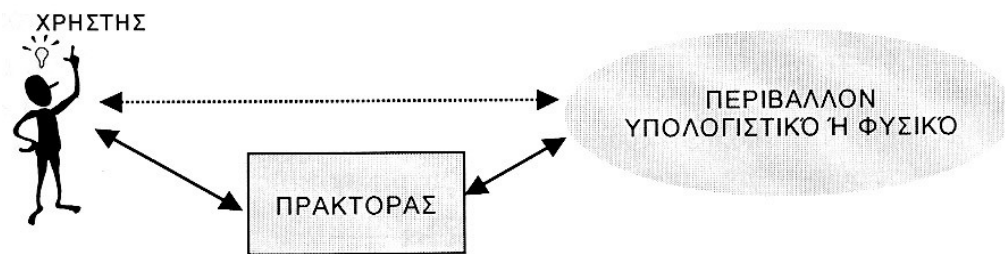


Πηγή: <http://www.csee.umbc.edu/~finin/talks/sisce97/sld001.htm>

Η τεχνολογία των πρακτόρων και ο προγραμματισμός με τη χρήση αυτών, είναι δύο θέματα που απασχολούν τη βιομηχανία και τους ερευνητές από το 1992. Πολλοί θεωρούν ότι αποτελούν το μέλλον στα υπολογιστικά συστήματα.

Το ενδιαφέρον για αυτούς είναι συνεχώς αυξανόμενο καθώς η τεχνολογία τους θα αλλάξει τη μορφή της διασύνδεσης χρήστη-λογισμικού. Ο χρήστης δε θα επικοινωνεί απευθείας με κάποια εφαρμογή αλλά θα χρησιμοποιεί έναν πράκτορα. Αυτός θα τον διευκολύνει σε χρονοβόρες διαδικασίες ρουτίνας ή διαδικασίες που χρειάζονται κάποια ικανότητα, που ο χρήστης δε διαθέτει.

**Σχήμα 3: Επικοινωνία χρήστη και πράκτορα**



Πηγή: Vlachavas I. κ.α., 2002

Οι πράκτορες χρησιμοποιούνται σε πλήθος εφαρμογών, όπως η παροχή έξυπνων υπηρεσιών βοήθειας, η αναζήτηση και το φιλτράρισμα πληροφοριών στο διαδίκτυο, η οργάνωση καθημερινού προγράμματος, το ηλεκτρονικό εμπόριο κλπ. Μελέτη και έρευνα που σχετίζεται με τους πράκτορες υπάρχει στην Κατανεμημένη Τεχνητή Νοημοσύνη, στη Ρομποτική, στη Τεχνητή Ζωή, στα κατανεμημένα συστήματα, στη διεπαφή ανθρώπου-μηχανής, στην ανάκτηση δεδομένων και απόκτηση γνώσης και αλλού.

Οι πράκτορες στη σύγχρονη μορφή τους είναι αρκετά ικανοί να διεκπεραιώσουν παρόμοιες με τις προαναφερθείσες εργασίες. Ταυτόχρονα, το Web έχει γίνει ο ιδανικός χώρος δραστηριοποίησής για τους λεγόμενους «εμπορικούς» πράκτορες. Αυτοί παρέχουν ενδιαφέρουσες υπηρεσίες όπως κυνήγι ευκαιριών, παρακολούθηση μετοχών, ανταλλαγές, μεσιτείες κ.α.

Πιο συγκεκριμένα, οι προσπάθειες του εμπορικού και επιχειρηματικού κόσμου στο τομέα του ηλεκτρονικού εμπορίου (e-commerce), είναι επικεντρωμένες, στη μετατροπή του Web σε μια βάση για αυτό και στην επίτευξη υψηλών στάνταρ ασφαλείας στις συναλλαγές και τις ηλεκτρονικές πληρωμές.

Οι περισσότερες εφαρμογές e-commerce απαιτούν την απόδειξη γνησιότητας των δύο πλευρών που συναλλάσσονται, καθώς και την ασφαλή ανταλλαγή κρίσιμων στοιχείων.

## 2. Ορισμός της έννοιας Πράκτορας (Agent)

### 2.1. Γενικά

Οι **Πράκτορες** για πολλά χρόνια (από το 1970 και μετά) αποτελούν αντικείμενο συζητήσεων και έρευνας. Όμως, παρόλη την επιστημονική προσπάθεια όλων αυτών των χρόνων, αποτελούν πλούσιο, ποικίλο και ατελές πεδίο έρευνας. Είναι μια δυναμική και πανταχού παρούσα αφηρημένη έννοια για την οποία ακόμη υφίσταται το ερώτημα: *τι ακριβώς είναι ένας Πράκτορας ;*

Ο *John McCarthy* στα μέσα της δεκαετίας του `50, ήταν ο πρώτος που εισήγαγε την ιδέα του πράκτορα και μαζί με τον *Oliver C. Selfridge* την καθιέρωσαν λίγα χρόνια αργότερα. Με την έννοια «πράκτορας» θέλανε να περιγράψουν ένα σύστημα, που όταν δέχονταν ένα στόχο, μπορούσε να επεξεργαστεί τα δεδομένα και βάσει του αποτελέσματος των υπολογισμών, να ζητά και να λαμβάνει συμβουλές, σε όρους φυσικής ανθρώπινης γλώσσας.

Ο *Hewitt* (1970) πρότεινε τη ιδέα ενός αυτοπροσδιοριζόμενου, διαδραστικού και συγχρόνως εκτελέσιμου αντικειμένου του **Ηθοποιού**, σε ένα μοντέλο από τέτοια αντικείμενα. Ο κάθε ηθοποιός είχε ενθουλακωμένη μια εσωτερική κατάσταση με τρόπους συμπεριφοράς και μπορούσε να επικοινωνεί μέσω μηνυμάτων με τα άλλα αντικείμενα. Το σημαντικό ήταν ότι πολλά αντικείμενα μπορούσαν να επικοινωνούν μεταξύ τους και να συμπεριφέρονται ποικιλοτρόπως αλλά ταυτόχρονα. Αυτή η ιδέα του Ηθοποιού ήταν ο προπομπός των πρακτόρων.

Αν το θέσουμε απλά, θα ορίσουμε τους πράκτορες ως προσωπικούς βοηθούς σε επίπεδο λογισμικού, με «εξουσιοδότηση» από τους χρήστες τους.

Ο *Negreonte* (1970, 1989) και *Alan Kay* (1984) ήταν από τους πρώτους που αναγνώρισαν την αξία των προσωπικών βοηθών σε επίπεδο λογισμικού. Είχαν αναπτύξει την ιδέα της ανάθεσης πρακτόρων για την εκτέλεση συγκεκριμένων computer-based εργασιών, στο περιβάλλον επικοινωνίας (interface). Πρόσφατα, αρκετοί κατασκευαστές υπολογιστών έχουν υιοθετήσει αυτήν την ιδέα, σαν την ιδανική περιγραφή του περιβάλλοντος εργασίας του μέλλοντος. Παλιό παράδειγμα, τα videos της Apple (1988).

Σύμφωνα με τα λόγια του Ted Selker (1994) από το κέντρο ερευνών της IBM, «οι πράκτορες είναι προγράμματα υπολογιστών που προσομοιώνουν μια ανθρώπινη σχέση, ενεργώντας οι ίδιοι αντί των χρηστών που εξυπηρετούν.»

Στο λεξικό Webster's New World Dictionary στο λήμμα Agent αναφέρεται ότι: «πράκτορας είναι ένα άτομο ή μια οντότητα, η οποία πράττει ή είναι ικανή να πράξει ή είναι εξουσιοδοτημένη να πράξει εκ μέρους κάποιου άλλου».

Σ` αυτή την ερμηνεία τονίζεται ότι ένας πράκτορας:

- ❖ αποτελεί μια ξεχωριστή οντότητα, με αρκετή έως απόλυτη αυτονομία
- ❖ κάνει κάποιες ενέργειες
- ❖ ενεργεί εκ' μέρους κάποιου άλλου

Στον τομέα της πληροφορικής σαν πράκτορας ορίζεται «Μια υπολογιστική μονάδα (τμήμα λογισμικού ή αυτόνομη εφαρμογή), η οποία αναλαμβάνει κάποια εξειδικευμένα καθήκοντα αυτόνομα.»

Ο N. Negreponte, διευθυντής του AI Lab του MIT, είχε αναφέρει ότι κάποτε θα υπάρξει ένας ηλεκτρονικός βοηθός, ο οποίος θα γνωρίζει πολύ καλά εμάς και κάποια συγκεκριμένη θεματική ενότητα. Έτσι, αυτός θα μας δίνει απαντήσεις σε διάφορα προβλήματα ή θέματα μας ενδιαφέρουν και σχετίζονται με τις γνώσεις του. Αυτό στους επιστημονικούς κύκλους αναφέρεται σαν την «ηλεκτρονική πεθερά» του Negreponte και αποτελεί ένα από τα πιο επίκαιρα αντικείμενα επιστημονικής μελέτης.

Σύμφωνα με τον Fah-Chun Cheong, ο πράκτορας είναι: «Λογισμικό που λειτουργεί ως προσωπικός βοηθός με ξεχωριστές αρμοδιότητες από τον χρήστη του.»

Στον παγκόσμιο ιστό (WWW) ως πράκτορας αναφέρεται: «το πρόγραμμα που συγκεντρώνει πληροφορίες ή εκτελεί κάποιες άλλες λειτουργίες, χωρίς την άμεση επίβλεψη του χρήστη.»

Γενικά, οι πράκτορες είναι θεωρητικά «έξυπνα» προγράμματα, που βοηθούν το χρήστη να ολοκληρώσει κάποιες ιδιαίτερα χρονοβόρες διεργασίες. Οι εργασίες τους σχετίζονται κυρίως με την αναζήτηση, την εύρεση και το φιλτράρισμα πληροφοριών. Ο καθένας μπορεί να αναθέσει μία αποστολή-εργασία σε κάποιο πράκτορα.



*Οι ρόλοι που παίζουν οι πράκτορες και θεωρίες που βασίζονται, οι οποίες έχουν αντληθεί από τα θέματα των αυτόνομων πρακτόρων και της Τεχνητής νοημοσύνης, συνδυάζονται και δημιουργούν ένα περιβάλλον αλληλεπίδρασης, άλλοτε συνεργασίας και άλλοτε ανταγωνισμού.*

## **2.2. Θεωρήσεις για τους πράκτορες**

*Χαλαρή θεώρηση (weak notion of agency):* Από τη γενικότερη πλευρά της Επιστήμης Υπολογιστών, οι πράκτορες πρέπει να είναι αυτοπροσδιοριζόμενοι (selfcontained) και να εκτελούνται ταυτόχρονα με κάποιους άλλους (concurrently executing). Επίσης, να μπορούν να έχουν ενθουλακωμένη μια εσωτερική κατάσταση που θα τους καθορίζει τη συμπεριφορά και να επικοινωνούν με άλλους ομοϊδέατες τους μέσω μηνυμάτων. Η άποψη αυτή θεωρεί ότι οι πράκτορες δεν πρέπει απαραίτητα να είναι "ευφυείς".

*Ισχυρή θεώρηση (strong notion of agency):* Από την πλευρά της ΤΝ, οι πράκτορες πρέπει να έχουν γνώση (knowledge), πεποιθήσεις (beliefs), επιθυμίες (desires), προθέσεις (intentions) και υποχρεώσεις (obligations). Η άποψη αυτή προϋποθέτει ότι: **α.** οι πράκτορες κατέχουν ένα αυστηρά καθορισμένο μοντέλο του κόσμου μέσα στον οποίο λαμβάνουν αποφάσεις **β.** χρησιμοποιούν συλλογιστική πριν καταλήξουν σε απόφαση και **γ.** αντιδρούν μεταξύ τους και με άλλες οντότητες του κόσμου που ζουν, με τρόπους που προσεγγίζουν την ανθρώπινη συμπεριφορά (δηλαδή, συχνά αποκτούν συναισθήματα).

## **2.3. Διαφορές από άλλες κατηγορίες λογισμικού**

Ένα ερώτημα που τίθεται είναι σε τι διαφέρουν οι πράκτορες (κυρίως οι λογισμικοί) από τα συμβατικά προγράμματα. Η απάντηση δεν είναι απλή, γιατί τα όρια είναι πολλές φορές δυσδιάκριτα και ο όρος πράκτορας περιλαμβάνει ένα πλήθος συστημάτων με διαφορές τόσο στην πολυπλοκότητα, όσο και στα επιμέρους χαρακτηριστικά. Έτσι, οι πράκτορες διαφέρουν από τα τυπικά προγράμματα στο ότι είναι εξατομικευμένοι, "τρέχουν" συνεχώς και είναι σχετικά αυτοδύναμοι.

Για παράδειγμα, ένα πρόγραμμα μισθοδοσίας, παρόλο που μπορεί να θεωρηθεί ότι αντιλαμβάνεται το περιβάλλον του μέσω του input που του δίνεται και ενεργεί σ' αυτό μέσω του output που αυτό δίνει, δεν είναι πράκτορας, γιατί το output που προσφέρει δεν θα έχει επίδραση σ' αυτό που θα αντιλαμβάνεται στο μέλλον. Επίσης,

δε "εκτελείται" συνεχώς, αλλά "εκτελείται" μια φορά και μετά σταματά, περιμένοντας να ξαναεκτελεστεί όταν αυτό επιλέξει ο χρήστης.

Για τους ίδιους λόγους, ένα πρόγραμμα ελέγχου ορθογραφίας που είναι ενσωματωμένο μέσα σ' έναν κειμενογράφο, δεν είναι πράκτορας. Ωστόσο, εάν το πρόγραμμα αυτό παρακολουθούσε το χρήστη που γράφει και τον διόρθωνε αυτόματα συνεχώς (on the fly), θα μπορούσε να χαρακτηριστεί πράκτορας. Έτσι, καταλήγουμε στο συμπέρασμα ότι όλοι οι πράκτορες είναι προγράμματα, αλλά όλα τα προγράμματα δεν είναι πράκτορες.

Τα «συστατικά στοιχεία» σε άλλα προγράμματα και στην κατανεμημένη επεξεργασία δεν είναι έξυπνα, αυτοδύναμα και δεν ανταλλάσσουν μεταξύ τους μηνύματα υψηλού επιπέδου, όπως κάνουν οι πράκτορες. Επίσης, δε λειτουργούν σε επίπεδο γνώσης αλλά κυρίως σε συμβολικό.

Τέλος, σχετικά με τις διαφορές των πρακτόρων με τις οντότητες του οντοκεντρικού προγραμματισμού αναφέρεται ότι, μια οντότητα ενθυλακώνει κάποια κατάσταση και την ελέγχει μόνο μέσω μεθόδων που η ίδια η οντότητα (αντικείμενο) παρέχει. Από την άλλη οι πράκτορες, πέρα από την κατάσταση που έχουν ενσωματωμένη, διατηρούν μέσα τους και τη συμπεριφορά τους, δηλαδή έχουν έλεγχο στις μεθόδους που ορίζουν τη συμπεριφορά τους.

#### **2.4. Είδη Πρακτόρων και συνώνυμα αυτών**

Ως πράκτορες μπορούν να αναφερθούν οι:

- ❖ Daemons (πράκτορες οι οποίοι παρακολουθούν ένα λογισμικό περιβάλλον και εκτελούν πράξεις για την αλλαγή αυτού, καθόσον αλλάζουν οι συνθήκες. Τέτοιοι πρακτόρες υπάρχουν στην υπηρεσία ftp και στο Unix, όπου πχ το πρόγραμμα xbiff παρακολουθεί το mail ενός χρήστη και τον ενημερώνει για κάθε νέο μήνυμα που έρχεται)
- ❖ Πελάτες ΓΠΔ διαφόρων εφαρμογών με το χρήστη (πχ πράκτορες αλληλογραφίας)
- ❖ Φυσικοί πράκτορες (Ρομποτική)
- ❖ Αληθοφανείς πράκτορες (Εικονική πραγματικότητα)
- ❖ Ευφυή συστήματα λογισμικού - Intelligent software agents

Οι πράκτορες ανάλογα με το που κινούνται ονομάζονται:

- ❖ **Knowbots** – ρομπότ με λειτουργία βασισμένη στη γνώση, τα οποία στόχο έχουν την ανάκτηση πληροφοριών (τέτοια είναι τα MetaCrawler, SavvySearch, Shopbot, Bargain Finder, Fido κ.α.).
- ❖ **Softbots** – ρομπότ λογισμικού που αντιδρούν σε μηνύματα και στέλνουν μηνύματα στο προγραμματιστικό περιβάλλον που κινούνται. Ένα softbot αποτελείται από εντολές, που παίζουν το ρόλο αισθητήρων και το βοηθούν να αλλάζει τη κατάσταση του περιβάλλοντος του και να αποκτά πληροφόρηση
- ❖ **Taskbots** – ρομπότ με συγκεκριμένη λειτουργία
- ❖ **Userbots** – ρομπότ που υπακούουν στις διαθέσεις του χρήστη
- ❖ **Robots** – που κινούνται στο φυσικό κόσμο
- ❖ **Personal agents** και **Personal assistants**
- ❖ **Autonomous agents** – που κινούνται σε δυναμικά και αβέβαια περιβάλλοντα

Ανάλογα την κύρια λειτουργία που επιτελούν οι διάφοροι πράκτορες τους ονομάζουμε:

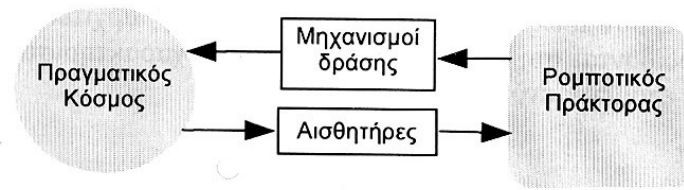
- ❖ αναζήτησης
- ❖ αναφοράς
- ❖ παρουσίασης
- ❖ προσανατολισμού
- ❖ διαχείρισης, κλπ.

## 2.5. Αρχική τυπολογία πρακτόρων

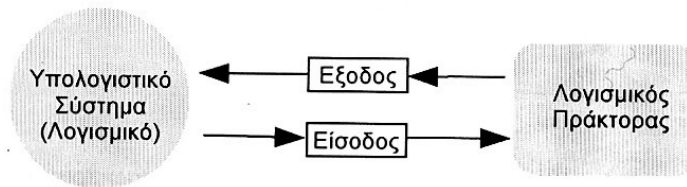
Οι πράκτορες αρχικά μπορούν να διαχωριστούν σε:

- ❖ **βιολογικούς**, που χρησιμοποιούν τις αισθήσεις, τις γνώσεις και τα μέρη του σώματος τους για να αντιληφθούν το γύρω κόσμο, να βγάλουν συμπεράσματα για αυτόν και να εφαρμόσουν τις ενέργειες που προκύπτουν από τη συλλογιστική τους.
- ❖ **τεχνητούς** (*artificial agents*) λειτουργούν παρόμοια με τους βιολογικούς και χωρίζονται στους:
  - α. ρομποτικούς** (*robotic agents* ή *robots*), οι οποίοι έχουν σαν αισθητήρες και μηχανισμούς δράσης μηχανικά ή ηλεκτρονικά μέρη και δρουν στον πραγματικό κόσμο
  - β. λογισμικούς** (*software agents* ή *softbots*), οι οποίοι είναι προγράμματα και δρουν σε ένα υπολογιστικό σύστημα

**Σχήμα 4: Αρχική τυπολογία πρακτόρων**



(α) Ρομποτικός πράκτορας



(β) Λογισμικός πράκτορας

Πηγή: Vlachavas I. κ.α., 2002

Και τα δύο είδη πρακτόρων εμπεριέχουν μία συλλογιστική διαδικασία (reasoning), μέσω της οποίας επεξεργάζονται τα ερεθίσματα του περιβάλλοντος τους, και εφαρμόζουν τα αποτελέσματα της συλλογιστικής σε αυτό, αλλάζοντας έτσι την κατάσταση του.

Λόγω του ότι οι πράκτορες αποτελούν πεδίο ενδιαφέροντος πολλών περιοχών της επιστήμης των υπολογιστών και μπορούν να εξεταστούν από διαφορετικές σκοπιές, υπάρχουν πολλές θεωρήσεις για αυτούς.

**Έτσι για:**

**είναι:**

τη Τεχνητή Νοημοσύνη	ευφυείς οντότητες
τον αντικειμενοστραφή προγραμματισμό	οντότητες (αντικείμενα) που επικοινωνούν μεταξύ τους
το παράλληλο προγραμματισμό	οντότητες που εκτελούνται παράλληλα
το πεδίο διασύνδεσης ανθρώπου-μηχανής	οντότητες που παίρνουν πρωτοβουλία σε συνεργασία με το χρήστη για να επιτελέσουν τα καθήκοντά τους

Στο σημείο αυτό αξίζει να αναφερθεί η θέση του Hyacinth S. Nwana ο οποίος χωρίζει την έρευνα γύρω από τους πράκτορες σε δύο περιόδους. Η πρώτη περίοδος είναι από το 1977 – σήμερα, και σχετίζεται με τα είδη συλλογιστικών πρακτόρων με εσωτερικά μοντέλα συμβολικής αναπαράστασης του περιβάλλοντος τους. Η έρευνα προς αυτή τη κατεύθυνση καλύπτει τους τρόπους διεπαφής και επικοινωνίας, την αποδόμηση και κατανομή των εργασιών, τον προσανατολισμό και τη συνεργασία, το συναγωνισμό ή ανταγωνισμό μεταξύ των πρακτόρων κ.α.

Η δεύτερη περίοδος ξεκινά το 1980 και φτάνει μέχρι σήμερα. Είναι προσανατολισμένη προς την πλήρη αποσαφήνιση και το διαχωρισμό των τύπων πρακτόρων με την ανάδειξη των ειδοποιών διαφορών. Στόχος είναι η ανάδειξη μιας κοινά αποδεκτής τυπολογίας πρακτόρων με ισχυρή επιχειρηματολογία, από τους πιο απλούς χωρίς μηχανισμούς ευφυΐας, μέχρι τους ευφυείς πράκτορες.

Έτσι, όπως θα δούμε παρακάτω σήμερα υπάρχουν αρκετές σημαντικές τυπολογίες πρακτόρων. Αυτό που πρέπει να σημειωθεί είναι, ότι οι πράκτορες περιλαμβάνονται και βοηθούν συγκεκριμένα συστήματα που εκτελούν συγκεκριμένες λειτουργίες. Ένα τέτοιο σύστημα, βασισμένο σε πράκτορες, μπορεί να αποτελείται από μόνο έναν (single agent system) ή από πολλούς περισσότερους (multi agent system). Η ανάλυση που ακολουθεί αναφέρεται στα είδη των πρακτόρων και των συστημάτων πρακτόρων.

### **3. Βασικά χαρακτηριστικά πρακτόρων – πράκτορες λογισμικού (Software agents)**

Τον τελευταίο καιρό η έρευνα οδηγείται περισσότερο στην υλοποίηση και μελέτη πρακτόρων λογισμικού παρά υλικού. Οι πράκτορες λογισμικού:

- ❖ είναι διαδικασίες αυτόνομες ή καθοδηγούμενες από τον στόχο υλοποίησης τους, που δε χρειάζονται επίβλεψη
- ❖ είναι τοποθετημένοι σε συγκεκριμένη θέση, έχουν πλήρη επίγνωση του περιβάλλοντος τους και αντιδρούν με αυτό
- ❖ συνεργάζονται με άλλους πράκτορες για να εκπληρώσουν τις εντολές τους (δηλ. το στόχο τους)
  
- ❖ είναι προσωπικοί βοηθοί σε επίπεδο software, με «εξουσιοδότηση» από τους χρήστες τους
  
- ❖ βοηθούν στην αυτοματοποίηση πολλών εργασιών

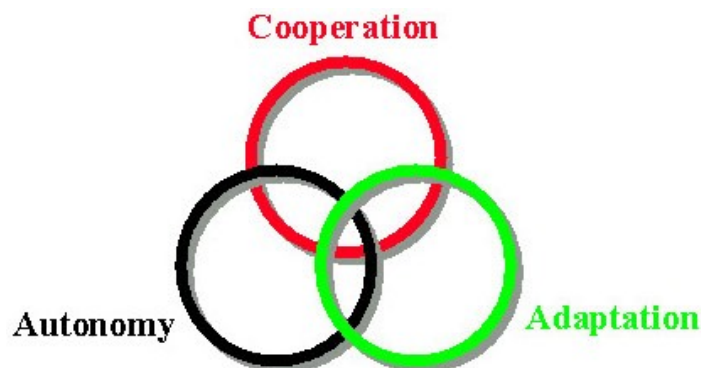
Τα βασικά χαρακτηριστικά ενός πράκτορα λογισμικού είναι:

- ❖ η **αυτονομία** (autonomy), ο πράκτορας λειτουργεί ανεξάρτητα από το χρήστη και ευθύνεται ο ίδιος για τον έλεγχο των πράξεών του. Ένας αυτοδύναμος πράκτορας είναι ένα σύστημα που τοποθετείται μέσα σ' ένα περιβάλλον και ταυτόχρονα είναι κομμάτι αυτού. Αυτός αντιλαμβάνεται το περιβάλλον και

ενεργεί σ' αυτό κατά περιόδους, προσπαθώντας να εκπληρώσει τις εργασίες-σκοπούς του με τέτοιο τρόπο, ώστε να έχουν επίδραση σε ότι θα αντιλαμβάνεται στο μέλλον.

- ❖ η **δυνατότητα αντίδρασης και προσαρμοστικότητας** (reactive and adaptive), ο πράκτορας αισθάνεται συνεχώς το εξωτερικό του περιβάλλον και προσαρμόζει συνεχώς τη συμπεριφορά του σύμφωνα με τις αλλαγές που γίνονται σ' αυτό (μηχανική ανακάλυψη γνώσης, ανταλλαγή metadata, μοντέλα χρήστη). Δηλαδή, ο πράκτορας αντιδρά και προσαρμόζεται, με την έννοια ότι αντιλαμβάνεται και μετά ενεργεί, απαντά δηλαδή στις αλλαγές που γίνονται στο περιβάλλον.
- ❖ η **δυνατότητα επικοινωνίας** (communication-cooperation-social ability), ο πράκτορας μπορεί να συνεργάζεται ή να ανταγωνίζεται με άλλες οντότητες, είτε αυτές είναι κάποιοι χρήστες, είτε είναι άλλοι Agents, είτε κάποια αντικείμενα, βάσεις δεδομένων κ.α, μέσω γλωσσών και πρωτοκόλλων επικοινωνίας.

**Σχήμα 5: Βασικά χαρακτηριστικά των πρακτόρων**



Πηγή: <http://www.csee.umbc.edu/~finin/talks/sisce97/sld001.htm>

Ένας πράκτορας μπορεί επίσης να:

- ❖ είναι κατανομημένος, ετερογενής, υπερενεργητικός, εξελίξιμος, κατανοητός, μοντελοποιήσιμος, εύστροφος
- ❖ είναι χρονικά συνεχής, με την έννοια ότι είναι μια διαδικασία που συνεχώς "τρέχει"
- ❖ είναι προσανατολισμένος σ' ένα σκοπό, δηλαδή, δεν ενεργεί απλά σαν απάντηση στο περιβάλλον

- ❖ είναι κινητός, δηλαδή, μπορεί να μετακινήσει τον εαυτό του από μια μηχανή σε άλλη
- ❖ είναι εύκαμπτος, με την έννοια ότι οι ενέργειες του δεν είναι από πριν προκαθορισμένες, ως ένα ακριβές πλάνο δράσης (μια σκεπτόμενη προσωπικότητα)
- ❖ διαθέτει κλιμακούμενη κληρονομικότητα και συναισθήματα
- ❖ παίζει συγκεκριμένο ρόλο
- ❖ συμπεριφέρεται με ειλικρίνεια – καλοσύνη – εμπιστοσύνη
- ❖ μαθαίνει, με την έννοια ότι αλλάζει την συμπεριφορά του σύμφωνα με την προηγούμενη εμπειρία του

Οι πράκτορες λογισμικού ανάλογα:

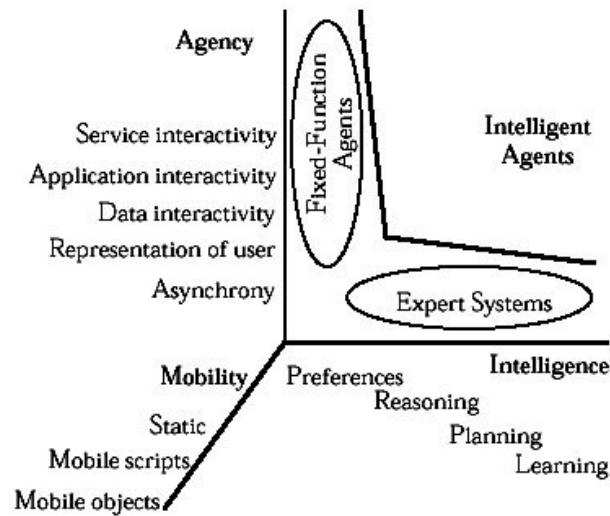
- ❖ με τη θέση που κινούνται, είναι στατικοί ή κινητικοί
- ❖ με το μηχανισμό δράσης που διαθέτουν, είναι συγκεκριμένου μοντέλου λογικής ή αντιδραστικοί ή διαδραστικοί
- ❖ με το είδος της προσαρμοστικότητας που έχουν, είναι συνεργατικοί ή ανταγωνιστικοί

Αν προσθέσουμε στα παραπάνω χαρακτηριστικά του πράκτορα λογισμικού, τις ικανότητες να θέτει στόχους, να διατηρεί μοντέλα πεποιθήσεων, να επεξεργάζεται και να κρίνει τις πράξεις του καθώς και τις πράξεις άλλων πρακτόρων (συμπεριλαμβανομένων και των χρηστών), και τέλος να μαθαίνει από τη συμπεριφορά του και να βελτιώνεται, τότε έχουμε έναν νοήμονα πράκτορα (Intelligent Agent).

#### **4. Σύγχρονες τυπολογίες πρακτόρων**

Ο Gilbert το 1995, έκανε μια προσπάθεια ταξινόμησης των πρακτόρων σε σχέση με το πόσο κινητικοί, πόσο έξυπνοι, τι υπηρεσίες προσφέρουν και με τι τρόπο το κάνουν αυτό (το είδος της συνεργασίας).

**Σχήμα 6: Τυπολογία πρακτόρων κατά Gilbert**



Πηγή: Jeffrey M. Bradshaw

*Συνεργασία* (agency), νοείται ο βαθμός αυτονομίας και εξουσίας του πράκτορα να διαχειρίζεται την επικοινωνία και διεπαφή του με άλλες οντότητες. Έτσι, οι πράκτορες μπορούν να είναι ασύγχρονοι, να απαιτούν τη παρουσία και τον έλεγχο του χρήστη, να έρχονται σε επαφή με τα δεδομένα, τις εφαρμογές και τελικά τις υπηρεσίες του περιβάλλοντος που βρίσκονται και να διαπραγματεύονται με αυτά.

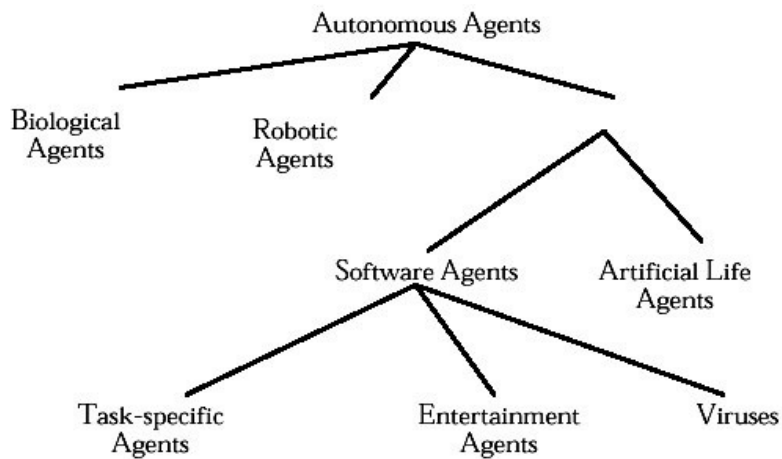
*Ευφυΐα - Νοημοσύνη* (Intelligence), νοείται ο βαθμός της λογικής, της συμπερασματολογίας και συμπεριφοράς του πράκτορα σε κάθε τι που λαμβάνει από το χρήστη ή από άλλες οντότητες. Έτσι, οι πιο απλοί πράκτορες δηλώνουν απλά κάποιες επιλογές τους, ενώ οι πιο σύνθετοι διαθέτουν μηχανισμούς λογικής στην συμπεριφορά τους, σχεδιάζουν τον τρόπο δράσης τους και τέλος μαθαίνουν το περιβάλλον τους και καθορίζουν βάσει των βιωμάτων τους μελλοντικές συμπεριφορές.

*Κινητικότητα* (Mobility), νοείται ο βαθμός κίνησης των πρακτόρων σε κάποιο δίκτυο. Έτσι, κάποιοι είναι στατικοί σε συγκεκριμένα υπολογιστικά συστήματα, κάποιοι άλλοι εκτελούνται από απόσταση και άλλοι έχουν μεταναστεύσει.

Οι Franklin και Graeser, το 1996, ήρθαν να αλλάξουν λίγο την αρχική τυπολογία, ενσωματώνοντας πλέον ως χαρακτηριστικά στην κατηγοριοποίηση, τις δομές ελέγχου και τη γλώσσα προγραμματισμού των πρακτόρων. Επίσης, ενέταξαν στο μηχανισμό κατηγοριοποίησης τα περιβάλλοντα κίνησης και τις εφαρμογές στα οποία είναι τμήμα και συμμετέχουν οι πράκτορες.



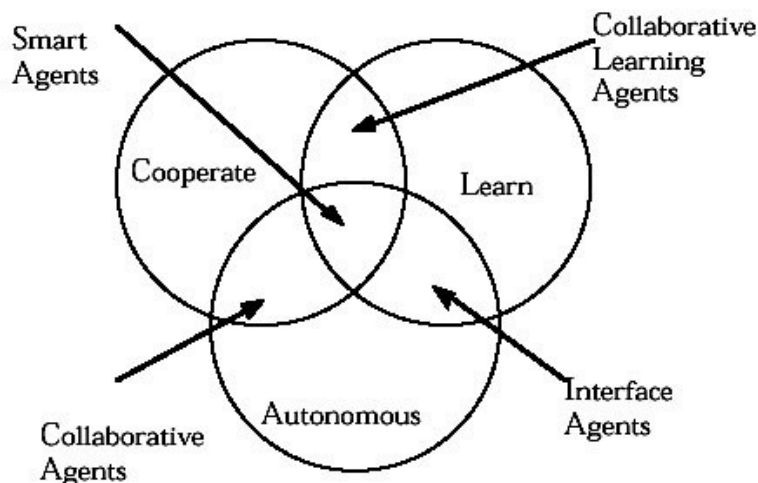
**Σχήμα 7: Τυπολογία πρακτόρων κατά Franklin & Graeser**



Πηγή: Jeffrey M. Bradshaw

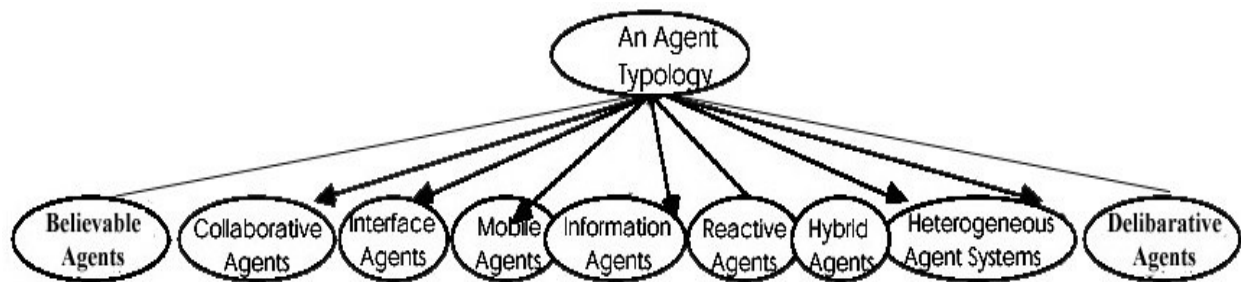
Ο Hyacinth S. Nwana το 1996, ταξινόμησε τους πράκτορες σύμφωνα με το πόσο αυτοί διαθέτουν περισσότερο ή λιγότερο κάποιο από τα βασικά τους χαρακτηριστικά.

**Σχήμα 8: Τυπολογία πρακτόρων κατά Nwana**



Πηγή: Jeffrey M. Bradshaw

Η παραπάνω μεθοδολογία ενσωματώνοντας πλέον και άλλες συνιστώσες, οδήγησε στη μετεξέλιξη της σε μια νέα τυπολογία. Ο καθορισμός της ομάδας κάποιου εξεταζόμενου πράκτορα, γίνονταν βάσει των παραπάνω χαρακτηριστικών και επιπλέον της κινητικότητάς του, της παρουσίας συμβολικού μοντέλου ή αντιδραστικής συμπεριφοράς, του ρόλου και των στόχων του, την υιοθέτηση σε αυτόν υβριδικής συμπεριφοράς, τη συμπαρουσία του μαζί με άλλους διαφορετικού τύπου κ.α. Στη παρακάτω εικόνα, για τους πράκτορες, φαίνεται η πλέον αντιπροσωπευτική κατηγοριοποίηση-τυπολογία.

**Σχήμα 9: Προτεινόμενη τυπολογία πρακτόρων**

Πηγή: Ιδία Επεξεργασία

## 5. Νοήμονες πράκτορες (Intelligent Agents)

### 5.1. Γενικά

Οι νοήμονες πράκτορες (*intelligent agents*) είναι ένας από τους πιο πρόσφατους και με μεγαλύτερο ενδιαφέρον κλάδους της ΤΝ. Στους νοήμονες πράκτορες, ο χρήστης ορίζει έναν στόχο και ο πράκτορας χρησιμοποιώντας ρουτίνες νοημοσύνης, αναλαμβάνει την ολοκληρωμένη υλοποίησή του, με τρόπο και σε χρόνο που επιλέγει ο ίδιος κάθε φορά. Αυτοί αποτελούν ίσως την πλέον "ανθρωπόμορφη" αρχιτεκτονική λογισμικού, στην οποία αποδίδονται συχνά και ανθρωποκεντρικά χαρακτηριστικά.

Ο ευφυής ή νοήμων πράκτορας:

- ❖ αντιλαμβάνεται το περιβάλλον μέσα στο οποίο βρίσκεται με τη βοήθεια αισθητήρων (*sensors*)
- ❖ είναι μέρος του περιβάλλοντος
- ❖ κάνει συλλογισμούς για το περιβάλλον και μαθαίνει από αυτό
- ❖ δρα πάνω σε αυτό με τη βοήθεια μηχανισμών δράσης (*effectors*), για την επίτευξη των στόχων που έχει

### 5.2. Διάφοροι ορισμοί

Οι *Russel & Norvig* υποστήριξαν ότι "Πράκτορας είναι οτιδήποτε μπορεί να αντιληφθεί το περιβάλλον του μέσω αισθητήρων και να αντιδράσει πάνω στο περιβάλλον μέσω μηχανισμών δράσης". Οι ίδιοι ανέδειξαν την αλληλεπίδραση των πρακτόρων με το περιβάλλον τους.

Ο *Virdhagrishwaran* υποστηρίζει ότι "Ο όρος πράκτορας χρησιμοποιείται για να αναπαραστήσει δύο ορθογώνιες έννοιες: την αυτόνομη εκτέλεση και τη συλλογιστική πάνω σε κάποιο συγκεκριμένο πεδίο".

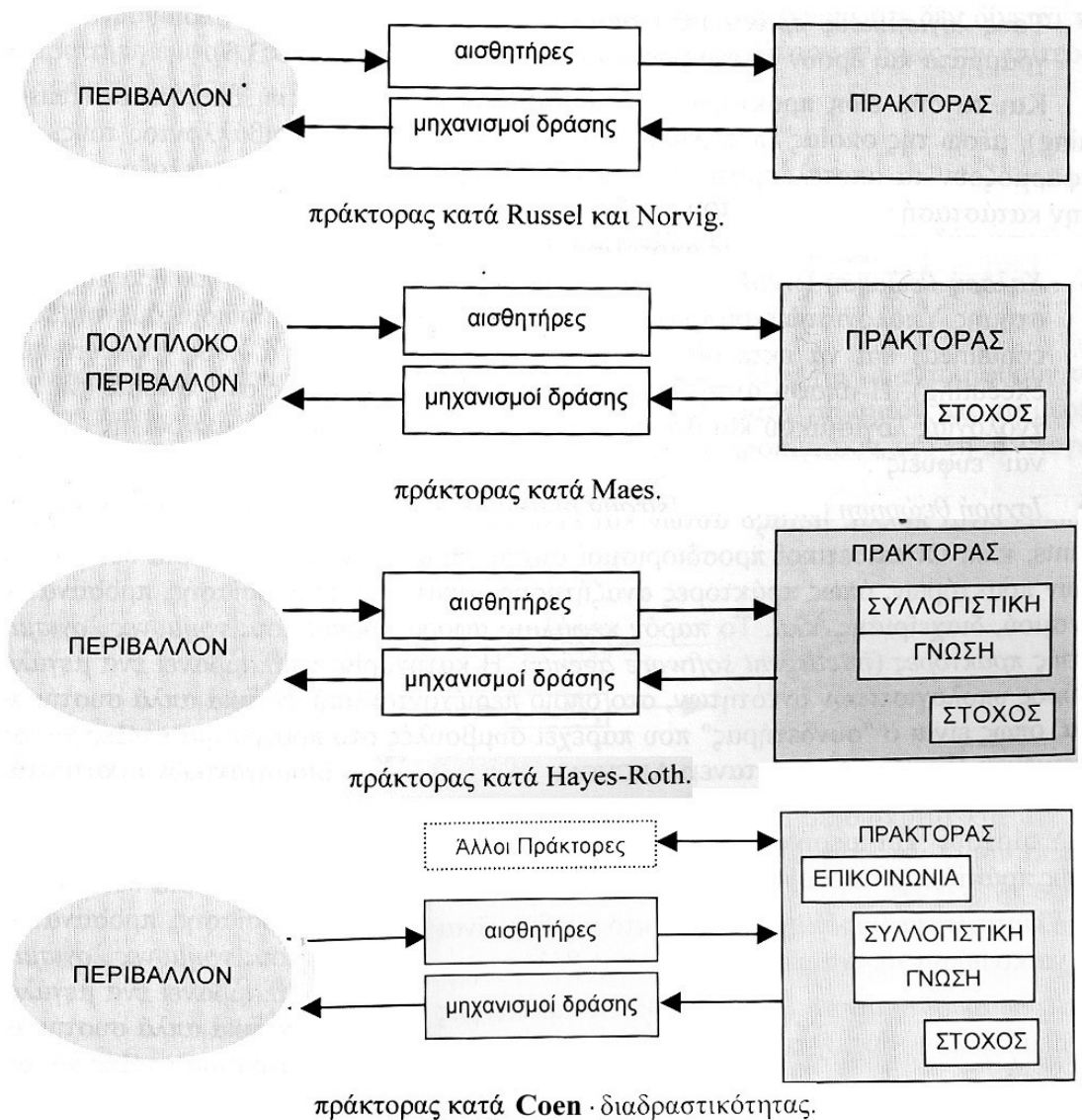
Η *Maes*, τόνισε ότι "Οι πράκτορες είναι υπολογιστικά συστήματα που δρουν σε ένα πολύπλοκο και δυναμικό περιβάλλον, αντιλαμβάνονται και δρουν αυτόνομα πάνω σε αυτό, πετυχαίνοντας έτσι ένα σύνολο από στόχους, για τους οποίους έχουν κατασκευαστεί".

Κατά τη *Hayes-Roth* "Οι ευφυείς πράκτορες:

- (α) αντιλαμβάνονται τις δυναμικές συνθήκες του περιβάλλοντος
- (β) δρουν πάνω στο περιβάλλον ώστε να το αλλάξουν
- (γ) συλλογίζονται ώστε να ερμηνεύσουν αυτά που αντιλαμβάνονται, να λύσουν προβλήματα, να συμπεράνουν και να καθορίσουν τη δράση τους"

Τέλος, ο *Coen* αναφέρει ότι οι "νοήμονες πράκτορες είναι προγράμματα που διενεργούν διάλογο, διαπραγματεύονται και συντονίζουν τη ροή πληροφοριών"

**Σχήμα 10: Ορισμοί νοημόνων πρακτόρων**



Πηγή: Vlachavas I. κ.α., 2002

**5.3. Περιβάλλοντα κίνησης**

Κάθε πράκτορας κινείται σε συγκεκριμένο περιβάλλον. Αυτά μπορούν να διαχωριστούν:

1. ανάλογα με το εάν υπάρχει διαθέσιμη πλήρης, ακριβής, και ανανεωμένη πληροφορία, σε **Προσβάσιμα** (Accessible) ή **Μη Προσβάσιμα** (Inaccessible)
2. αναφορικά με το εάν μία συγκεκριμένη ενέργεια έχει πάντα συγκεκριμένα αποτελέσματα, σε **Αιτιοκρατικά** (Deterministic) ή **Μη Αιτιοκρατικά** (Non-Deterministic)
3. αναφορικά με το αν το περιβάλλον χωρίζεται ή όχι σε διακριτά και ανεξάρτητα επεισόδια, σε **Επεισοδιακά** (Episodic) ή **Μη Επεισοδιακά** (Non-Episodic). Σε ένα επεισοδιακό περιβάλλον δεν υπάρχει ανάγκη να εξετάζει ο πράκτορας την

επίδραση που θα έχουν οι τρέχουσες ενέργειες σε μελλοντικές καταστάσεις. Δηλαδή, οι ενέργειες του πράκτορα σε ένα επεισόδιο δεν έχουν επίδραση στα υπόλοιπα επεισόδια.

4. σύμφωνα με το εάν εμφανίζονται αλλαγές χωρίς την παρέμβαση του πράκτορα, σε **Δυναμικά** (Dynamic) ή **Στατικά** (Static)
5. ως προς το εάν υπάρχει ή όχι ένας πεπερασμένος αριθμός ενεργειών και δεδομένων στο μηχανισμό αντίληψης του πράκτορα, σε **Διακριτά** (Discrete) ή **Συνεχή** (Continuous)

#### **5.4. Χαρακτηριστικά**

Τα χαρακτηριστικά που συνθέτουν την «προσωπικότητα» ενός νοήμονα πράκτορα είναι:.

##### **α. Αυτονομία (autonomy)**

Οι πράκτορες λειτουργούν χωρίς την άμεση παρέμβαση των χρηστών ή άλλων πρακτόρων και έχουν αυτοέλεγχο. Ελέγχουν δηλαδή, οι ίδιοι την εσωτερική τους κατάσταση και αυτενεργούν. Έχουν τη δυνατότητα να επιδιώκουν τους στόχους τους, χωρίς να δέχονται συνεχώς εντολές από το χρήστη ή κάποια άλλη εξωτερική πηγή. Ο χρήστης προσδιορίζει μόνο το γενικό στόχο και δεν απασχολείται με τη λήψη των επιμέρους αποφάσεων. Τα παραπάνω συμβαίνουν στο βαθμό που καθορίζει ο ίδιος ο χρήστης.

Η αυτονομία μπορεί να διαφέρει από πράκτορα σε πράκτορα. Για παράδειγμα, υπάρχουν πράκτορες που αυτόνομα δημιουργούν ένα απλό βραδινό backup και άλλοι που διαπραγματεύονται μόνοι τους τις τιμές αγοράς προϊόντων από on-line κέντρα πωλήσεων.

##### **β. Αφοσίωση στο χρήστη (Delegation):**

Η εξουσία του πράκτορα περιορίζεται από τον χρήστη του γιατί αυτός θέτει πάντα τους τελικούς στόχους. Ένας πράκτορας λειτουργεί πάντα για λογαριασμό του χρήστη του ή ενός άλλου πράκτορα και οι ενέργειές του θα πρέπει να συμφωνούν με τις ενέργειες που θα έκανε ο ίδιος ο χρήστης ή ο άλλος πράκτορας.

##### **γ. Κοινωνικότητα και Δυνατότητες επικοινωνίας (social ability - Communication skills)**

Ο πράκτορας πρέπει να αλληλεπιδρά με το χρήστη για να παίρνει οδηγίες από αυτόν καθώς και να τον ενημερώνει για την πορεία των εργασιών του.

Επίσης, μπορεί να αλληλεπιδρά με άλλους πράκτορες για την επίτευξη ενός κοινού στόχου. Η επικοινωνία επιτυγχάνεται μέσω μίας κοινά κατανοητής γλώσσας. Έτσι, οι πράκτορες μπορούν να συνεργαστούν για την επίτευξη των στόχων τους. Είτε γιατί οι στόχοι είναι ανεξάρτητοι και ξεχωριστοί για τον κάθε πράκτορα, είτε γιατί είναι κοινοί για πολλούς από τους πράκτορες.

#### **δ. Νοημοσύνη (*Intelligence*)**

Ένας πράκτορας πρέπει να «καταλαβαίνει» τα συμβάντα που παρακολουθεί και να παίρνει αποφάσεις για να λειτουργήσει αυτόνομα. Η θεωρία της Τεχνητής Νοημοσύνης, των Νευρωνικών Δικτύων και της Ασαφούς Λογικής (*Fussy Logic*) συναντάται συνεχώς και στο σύνολο των νοημόνων πρακτόρων.

#### **ε. Λογικότητα (*rationality*)**

Οι πράκτορες δρουν για να πετύχουν τους στόχους τους. Δεν κάνουν αναίτιες ενέργειες και δεν λειτουργούν εναντίον της επίτευξης των στόχων τους.

#### **στ. Αντιδραστικότητα (*reactiveness*)**

Οι πράκτορες αντιλαμβάνονται το περιβάλλον τους και αντιδρούν μέσα σε συγκεκριμένα χρονικά πλαίσια, στις αλλαγές που επέρχονται σε αυτό.

#### **ζ. Προνοητικότητα (*Pro-activeness*)**

Οι πράκτορες είναι ικανοί να επιδείξουν συμπεριφορά που βασίζεται στους στόχους τους και στις συνθήκες, οι οποίες εμφανίζονται στο περιβάλλον τους.

Τόσο η προνοητικότητα, όσο και η αντιδραστικότητα απαιτούν σε κάποιο βαθμό, τη δυνατότητα συλλογισμού και λογικής από τον πράκτορα και την ύπαρξη νοημοσύνης σε αυτόν.

#### **η. Δυνατότητα μεταφοράς - Κινητικότητα (*mobility*)**

Κάποιοι πράκτορες δεν είναι πάντα στατικοί, αλλά μπορούν να κινηθούν σε ένα υπολογιστικό περιβάλλον. Έχουν δηλαδή, τη δυνατότητα να μεταφέρονται από ένα υπολογιστικό σύστημα σε ένα άλλο, χρησιμοποιώντας στο μέγιστο τις δυνατότητες των δικτύων και του Internet.

Αυτό το χαρακτηριστικό το συναντάμε πολύ συχνά στους λεγόμενους «Internet Agents», «Web Agents», «Spiders», «WebCrawlers» ή «Worms».

**θ. Παρακολούθηση (Monitoring)**

Ένας πράκτορας χρειάζεται να μπορεί να παρακολουθεί το περιβάλλον του για να λειτουργήσει αυτόνομα.

**ι. Προσαρμοστικότητα (adaptivity)**

Οι πράκτορες προσαρμόζονται αυτόματα στις προτιμήσεις των χρηστών τους λαμβάνοντας υπ' όψη προηγούμενες εμπειρίες. Επίσης, μπορούν και προσαρμόζονται αυτόματα στο περιβάλλον τους μιας και το παρακολουθούν συνεχώς. Έχουν δηλαδή ικανότητα για μάθηση.

**κ. Ειλικρίνεια (veracity)**

Οι πράκτορες δε δίνουν εσκεμμένα λάθος πληροφορίες.

**λ. Αγαθή προαίρεση (benevolence)**

Οι πράκτορες προσπαθούν να επιτύχουν πάντα τους στόχους που τους έχουν ανατεθεί.

**μ. Χαρακτήρας (Character)**

Μερικοί πράκτορες έχουν μια καλά-καθορισμένη «προσωπικότητα» και ψυχολογική κατάσταση.

**ν. Δυναμικότητα (Temporal continuity)**

Ένας πράκτορας χρειάζεται να επεξεργάζεται συνεχώς δεδομένα σε αντιδιαστολή με τα απλά προγράμματα. Ένα απλό πρόγραμμα, κατά την εφαρμογή του, συγκεντρώνει κάποια στοιχεία προς επεξεργασία, τα επεξεργάζεται και εμφανίζει στον χρήστη τα αποτελέσματα.

*«Τρέχει» δηλαδή μια φορά και δίνει «στατικά» αποτελέσματα .*

Αντίθετα, για παράδειγμα, ένας πράκτορας που παρακολουθεί τις τιμές μιας μετοχής στο χρηματιστήριο και ερευνά το αν θα πρέπει να επενδύσουμε σε αυτή ή όχι, χρειάζεται συνεχή ενημέρωση από το περιβάλλον του.

*Δηλαδή «τρέχει» συνεχώς και δίνει διαφορετικά αποτελέσματα σε διαφορετικές χρονικές στιγμές.*

Κάθε νοήμων πράκτορας έχει ορισμένα από τα παραπάνω χαρακτηριστικά. Όλοι τους όμως έχουν την ίδια φιλοσοφία σχεδιασμού και τους ίδιους στόχους, οπότε και ανήκουν στην ίδια κατηγορία προγραμμάτων.

### **5.5. Πλεονεκτήματα χρήσης Νοημόνων Πρακτόρων**

Τα χαρακτηριστικά των νοημόνων πρακτόρων δίνουν σημαντικές λύσεις στην επίλυση σύνθετων προβλημάτων, κάτι που δεν συμβαίνει με τα άλλα προγράμματα. Η χρήση νοημόνων πρακτόρων προσφέρει:

***–απλοποίηση της κατανεμημένης επεξεργασίας και επίτευξη της διασυνδεσιμότητας διαφορετικών πληροφοριακών συστημάτων***

Οι πράκτορες θα υφίστανται ως οντότητες παρακολούθησης και διαχειριστές των διαθεσίμων πηγών.

***–υπερπήδηση των προβλημάτων διεπαφής και επικοινωνίας του χρήστη με τον Η/Υ και τις εφαρμογές***

Οι πράκτορες θα υφίστανται ως οντότητες ενημέρωσης, υποβοήθησης του χρήστη με άμεσες ή σε λογικό χρόνο αποκρίσεις

***–αντιμετώπιση της πολυπλοκότητας των προβλημάτων***

Οι νοήμονες πράκτορες θα χρησιμοποιούν τη νοημοσύνη τους για να συνδυάζουν πολλαπλές μεταβλητές, με στόχο την επίλυση των προβλημάτων που συναντούν ή που τους ανατίθενται. Γνωρίζουμε ότι για τον άνθρωπο, πιο δύσκολα στην επίλυσή τους, είναι τα πολυδιάστατα προβλήματα. Όσο αυξάνει ο αριθμός των μεταβλητών, τόσο πιο σύνθετο γίνεται και το πρόβλημα.

Τα συστήματα στήριξης αποφάσεων είναι σχεδιασμένα για να βοηθούν τους ανθρώπους στις αποφάσεις τους. Όμως ένας νοήμων πράκτορας θα δώσει καλύτερες λύσεις γιατί λαμβάνει υπ' όψη του τις προσωπικές συνήθειες του χρήστη.

Για παράδειγμα, ένας πράκτορας υπεύθυνος για αυτόματες κρατήσεις δωματίων σε ξενοδοχεία, θα υπολογίζει τις ιδιαιτερότητες του χρήστη του, που επιθυμεί το δωμάτιο να βρίσκεται κοντά στο αεροδρόμιο και κοντά στην έκθεση ή την εταιρία που θα επισκεφθεί, να είναι Α κατηγορίας, με χρέωση για πρωινό κλπ.

***–μείωση του κόστους επίλυσης προβλημάτων***

Με τη χρήση ενός νοήμων πράκτορα θα μειωθούν τα έξοδα για αναζήτηση πληροφοριών. Αν οι πληροφορίες που χρειαζόμαστε είναι δυσεύρετες και απαιτούν πολύ χρόνο για τη συγκέντρωση και την επεξεργασία τους, τότε η χρήση νοημόνων πρακτόρων είναι αναγκαία.

Πχ. μια απλή αναζήτηση σε κάποια βάση δεδομένων δε θεωρούμε ότι λειτουργεί με «νοημοσύνη». Όμως, ένας νοήμων πράκτορας, αναζητά πληροφορίες σε πολλαπλές



βάσεις δεδομένων και τις συνδυάζει κατάλληλα για να τις επεξεργαστεί και να μας δώσει το σωστό αποτέλεσμα.

### ***–αυτόματη εκμάθηση και τυποποίηση της γνώσης για μελλοντικές αναζητήσεις***

Ένα απλό πρόγραμμα αναζήτησης χρησιμοποιεί τυποποιημένες «Βάσεις Γνώσεων» (Knowledge Bases) για να αναπαραστήσει τις πληροφορίες. Αντίθετα, ένας νοήμων πράκτορας χρησιμοποιεί ρουτίνες Τεχνητής Νοημοσύνης για να «μαθαίνει», ανάλογα με την προηγούμενη συμπεριφορά του χρήστη.

### ***–αναζήτηση σε περισσότερους και πιο πολύπλοκους μηχανισμούς και συνεργασία με αυτούς για το τελικό αποτέλεσμα***

Ο αριθμός των συμβαλλόμενων στοιχείων ενός προβλήματος σχετίζεται άμεσα με την πολυπλοκότητα του προβλήματος αυτού.

π.χ. Ένα Infobot (απλό πρόγραμμα χωρίς νοημοσύνη που συλλέγει πληροφορίες από το Internet) μπορεί να συναλλάσσεται με διαφορετικούς φορείς που «μιλάνε» την ίδια γλώσσα. Ένας Intelligent Agent μπορεί να επικοινωνεί με τελείως διαφορετικούς φορείς (π.χ ξενοδοχεία, αεροπορικές εταιρίες, θέατρα, κλπ) για να πετύχει τον στόχο του.

### ***–επίτευξη συμπεριφοράς βάσει χρονικού προγράμματος***

Οι intelligent agents παρακολουθούν ανά πάσα στιγμή το περιβάλλον τους για να ανταποκρίνονται στα ερεθίσματα. Για παράδειγμα ένας νοήμων πράκτορας που ενημερώνει τον χρήστη του για τις καθημερινές ειδήσεις από πηγές του Internet, πρέπει να επεξεργάζεται τα δεδομένα του κάθε μέρα.

### ***–επίτευξη απομακρυσμένης λειτουργίας***

Ένα απλό πρόγραμμα λειτουργεί συνήθως «τοπικά», στον υπολογιστή του κάθε χρήστη. Ένας νοήμων πράκτωρ μπορεί να χρησιμοποιήσει τεχνολογίες δικτύων (όπως το Internet) και να βρίσκεται μακριά από τον χρήστη του, απαλλάσσοντάς τον από χρόνο και υπολογιστική ισχύ.

## **5.6. Μειονεκτήματα χρήσης Νοημόνων Πρακτόρων**

Οι νοήμονες πράκτορες είναι προγράμματα σχεδιασμένα από ανθρώπους, όπως και όλα τα άλλα κατασκευάσματα της Τεχνητής Νοημοσύνης. Είναι πιθανό λοιπόν, να υπάρχουν λάθη στο σχεδιασμό και στην επιλογή των παραμέτρων τους. Έτσι, μπορούν να παρουσιαστούν προβλήματα στην επεξεργασία και γενικότερα διαχείριση

των «γνώσεων» από ένα πράκτορα, με αποτέλεσμα να δώσει λανθασμένα αποτελέσματα και να μη συμβουλέψει σωστά το χρήστη.

Η χρησιμοποίηση νοημών πρακτόρων με αυξημένη αυτονομία, που δεν επιτρέπουν στο χρήστη να παρεμβαίνει στις αποφάσεις τους, οδηγεί σε εσφαλμένα συμπεράσματα μιας και οι πράκτορες λόγω της κατάστασης τους χαρακτηρίζονται από υπερβολή στις κινήσεις τους.

Επιπρόσθετα, η χρήση νοημών πρακτόρων προϋποθέτει μεγάλη υπολογιστική ισχύ για να έχουμε τα επιθυμητά αποτελέσματα στο χρόνο τους. Έτσι η τοπική χρήση προηγμένων πρακτόρων είναι σχεδόν απαγορευτική σε προσωπικούς υπολογιστές.

Η τεχνολογία τους δεν είναι διαδεδομένη, παρά μόνο σε στρατιωτικούς οργανισμούς και σε πανεπιστημιακά εργαστήρια. Στον επιχειρηματικό τομέα, πέρα από τον κλάδο των ηλεκτρονικών υπολογιστών, δύσκολα συναντάμε νοήμονες πράκτορες.

Ένα ακόμη μειονέκτημα των νοημών πρακτόρων, ιδιαίτερα των κινητών (mobile), είναι η ασφάλεια. Πολλοί Web Administrators δε θέλουν να βρίσκονται στο σύστημά τους διάφοροι κινητοί πράκτορες και να έχουν διάφορες εξουσίες για λόγους ασφαλείας.

Μερικοί πράκτορες που χρησιμοποιούν το Internet ως πηγή γνώσης, μπορεί να έχουν περιορισμένη πρόσβαση σε συγκεκριμένες πηγές. Ακόμη, αυτοί απαιτούν υψηλές ταχύτητες και μεγάλο bandwidth. Για παράδειγμα, ένας μικρός λιανοπωλητής που διατηρεί on-line βιβλιοπωλείο στο Internet, μπορεί να μη θέλει να «εκθέσει» τις τιμές των βιβλίων του σε έναν νοήμονα πράκτορα, για να αποφύγει τον ανταγωνισμό από τα μεγαλύτερα βιβλιοπωλεία, τα οποία έχουν καλύτερες τιμές και θα προτιμηθούν από τον πράκτορα.

## **ΚΕΦΑΛΑΙΟ Β**

### **ΑΡΧΙΤΕΚΤΟΝΙΚΕΣ – ΤΥΠΟΙ ΠΡΑΚΤΟΡΩΝ**

i.	ΕΣΩΤΕΡΙΚΗΣ ΚΑΤΑΣΤΑΣΗΣ .....	28
	BDI AGENTS.....	31
ii.	ΣΥΝΕΡΓΑΣΙΑΚΟΙ .....	32
iii.	ΑΝΤΙΔΡΑΣΤΙΚΟΙ .....	34
	ΠΡΑΚΤΟΡΕΣ ΥΠΙΑΓΩΓΗΣ.....	35
iv.	ΥΒΡΙΔΙΚΟΙ .....	38
v.	ΚΙΝΗΤΟΙ .....	43
vi.	ΔΙΕΠΑΦΗΣ .....	47
vii.	ΠΛΗΡΟΦΟΡΙΑΚΟΙ .....	49
viii.	.....ΕΤΕΡΟΓΕΝΗ ΣΥΣΤΗΜΑΤΑ	
	59	
	ΠΟΛΥΠΡΑΚΤΟΡΙΚΑ ΣΥΣΤΗΜΑΤΑ ...	61
ix.	ΑΛΗΘΟΦΑΝΕΙΣ.....	65

## 1. Εισαγωγή

**Αρχιτεκτονική πρακτόρων** είναι μια συγκεκριμένη μεθοδολογία δόμησης πρακτόρων. Αυτή καθορίζει τα μοντέλα που αποτελούν τον πράκτορα και τον τρόπο που αυτά αλληλεπιδρούν μεταξύ τους ή με άλλους πράκτορες και αντικείμενα. Η αλληλεπίδραση αυτών των μοντέλων δίνει απάντηση, στο πως τελικά συνδυάζονται, τα εξωτερικά δεδομένα που παίρνει ο πράκτορας, από τις περιβάλλουσες οντότητες του, με την εσωτερική κατάσταση που διαθέτει και καθορίζουν τελικά τις ενέργειες του και τη μελλοντική του εσωτερική κατάσταση. Στην ουσία η αρχιτεκτονική περικλείει τις τεχνικές και τους αλγόριθμους που υπάρχουν σε ένα πράκτορα, για να επικοινωνεί με το περιβάλλον του και τελικά να αποφασίζει πως θα κινείται σε αυτό. (Maes, 1991)

Ο Kaebbling την ίδια χρονιά αναφέρει ότι αρχιτεκτονική αποτελεί μια συγκεκριμένη συλλογή μοντέλων λογισμικού ή υλικού, που αποτυπώνονται ως κουτιά. Μεταξύ τους εμφανίζονται ως βέλη οι ροές ελέγχου και δεδομένων. Τέλος, αρχιτεκτονική αποτελεί η γενική μεθοδολογία της σχεδιαστικής ανάλυσης των μοντέλων που περιγράφουν συγκεκριμένες εντολές-ενέργειες.

Οι αρχιτεκτονικές που υπάρχουν στους πράκτορες αρχικά διαχωρίζονταν σε δύο μεγάλες κατηγορίες:

Η πρώτη αποτελεί την κλασσική προσέγγιση του θέματος και τους θεωρεί συστήματα γνώσης. Βασικό χαρακτηριστικό της πρώτης κατηγορίας αρχιτεκτονικών, είναι η ύπαρξη εσωτερικής συμβολικής αναπαράστασης του κόσμου στον οποίο δρα ο πράκτορας. Οι **πράκτορες με εσωτερική κατάσταση** (*deliberative agents*) έχουν μια εσωτερική συμβολική αναπαράσταση του περιβάλλοντος στο οποίο βρίσκονται και ένα σύνολο κανόνων βάσει των οποίων καθορίζουν την επόμενη ενέργεια τους. Κάθε φορά συγκρατούν στη μνήμη τους, την κατάσταση που αναπαριστά την εξέλιξη του περιβάλλοντος και την τρέχουσα λειτουργία την οποία εκτελούν, ώστε να αποφασίσουν για την επόμενη ενέργεια.

Πολλές φορές οι πράκτορες με εσωτερική κατάσταση εμφανίζουν προνοητική συμπεριφορά. Σε αυτές τις περιπτώσεις ένα σύνολο από στόχους καθορίζουν την επόμενη ενέργεια τους.

Η δεύτερη αρχιτεκτονική αποτελεί αποτέλεσμα της σκέψης της λεγόμενης εναλλακτικής σχολής. Οι πράκτορες αυτής της κατηγορίας δεν έχουν εσωτερική αναπαράσταση του κόσμου στον οποίο δρουν και αντιδρούν, μέσω ενός συνόλου

κανόνων, στα ερεθίσματα του περιβάλλοντος τους. Ονομάζονται **αντιδραστικοί πράκτορες** (*reactive agents*). Επιπλέον, συμπεριφέρονται με μια φιλοσοφία ερεθίσματος/αντίδρασης (*stimulus/response type behavior*), στην κάθε τρέχουσα κατάσταση του περιβάλλοντος που βρίσκονται.

Οι αντιδραστικοί πράκτορες παίρνουν δεδομένα από το περιβάλλον (αντίληψη) και βάσει των κανόνων λειτουργίας τους αντιδρούν. Δεν έχουν μνήμη και έτσι δεν υπολογίζουν τις επόμενες ενέργειες τους βάσει παλαιότερων καταστάσεων του κόσμου.

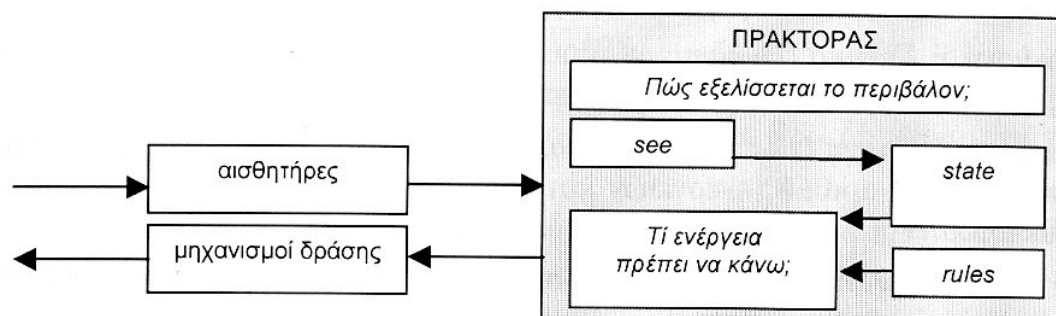
Παρακάτω αναλύεται διεξοδικά η τυπολογία πρακτόρων, βάσει της ενσωματωμένης αρχιτεκτονικής που έχουν. Γενικά, υπάρχουν οι απλοί πράκτορες που λειτουργούν αυτόνομα, τα συστήματα πολλαπλών πρακτόρων που το τελικό αποτέλεσμα προέρχεται από τη συνεργασία των πρακτόρων (*swarm, markets*), και η σύνδεση πρακτόρων με σχέση πελάτη/εξυπηρετητή.

## 2. Πράκτορες με εσωτερική κατάσταση (Deliberative Agents)

Η θεμελιώδης αρχή της αρχιτεκτονικής αυτού του είδους πρακτόρων είναι η υπόθεση του **φυσικού-συμβολικού συστήματος** (Newell & Simon, 1976). Αυτό αποτελείται από φυσικά αντιληπτές οντότητες (σύμβολα) οι οποίες υλοποιούν δομές. Επιπλέον, χαρακτηρίζεται από εκτελέσιμες διαδικασίες (κωδικοποιημένες ομάδες εντολών) που αφορούν και σχετίζονται με τις φυσικές οντότητες. Το παραπάνω σύστημα απέχει πολύ λίγο από την έννοια του πράκτορα με εσωτερική κατάσταση, στον οποίο πρώτος ο McCarthy αναφέρθηκε.

Οι πράκτορες με εσωτερική κατάσταση είναι επιγέννημα της κλασικής σχολής της ΤΝ, η οποία υποστηρίζει ότι «τα ευφυή συστήματα επεξεργάζονται και κάνουν συλλογισμούς χρησιμοποιώντας σύμβολα (*physical symbol system hypothesis*)».

**Σχήμα 11: Αρχιτεκτονική πράκτορα εσωτερικής κατάστασης**



Πηγή: Vlachavas I. κ.α., 2002

Ο πράκτορας με τέτοια αρχιτεκτονική διατηρεί εσωτερική συμβολική αναπαράσταση του περιβάλλοντος του. Επιτυγχάνει τους στόχους του και λαμβάνει τις αποφάσεις του χρησιμοποιώντας κλασσικές συλλογιστικές της Λογικής καθώς και τεχνικές ταυτοποίησης προτύπων και χειρισμού συμβόλων.

Χαρακτηριστικό των πρακτόρων εσωτερικής κατάστασης είναι η βάση γνώσης. Αυτή τους βοηθά στην αντίληψή του περιβάλλοντος κόσμου με μορφή λογικών προτάσεων. Άλλο χαρακτηριστικό αποτελεί το σύνολο κανόνων που αναπαριστούν τις πιθανές εκτελέσιμες ενέργειες του πράκτορα. Τέλος, πολλοί πράκτορες καταστρώνουν πλάνα για την επίτευξη των στόχων τους ή για να κάνουν λογικούς συμπερασμούς.

Το θετικό σε αυτή την αρχιτεκτονική είναι ότι χρησιμοποιούνται μέθοδοι και τεχνικές με σαφώς καθορισμένη και απλή σημασιολογία. Το δύσκολο είναι ότι οι αποφάσεις λαμβάνονται μέσα από την ενεργοποίηση των περιορισμών του μοντέλου με τους συμβολισμούς.

Τα αρνητικά είναι:

- ❖ η αδυναμία ακριβούς και ικανοποιητικής συμβολικής περιγραφής για τον πραγματικό κόσμο
- ❖ η σύντομη και ικανοποιητική εξαγωγή κατάλληλων συμπερασμάτων
- ❖ το υπολογιστικό κόστος που εμφανίζουν οι τεχνικές που βασίζονται στη λογική
- ❖ η θεώρηση της στατικότητας του κόσμου κατά το χρόνο που ο πράκτορας εξάγει συμπεράσματα
- ❖ η δυσκολία αναπαράστασης δυναμικών περιβαλλόντων
- ❖ η αναπαράσταση της έννοιας του χρόνου
- ❖ η δυσκολία απάντησης στην ερώτηση "τι πρέπει να γίνει" κάθε φορά

### **2.1. Παραδείγματα πρακτόρων εσωτερικής κατάστασης**

- το **Strips** των Fikes & Nilson (1971): αυτό έπαιρνε μια συμβολική περιγραφή του κόσμου και των επιθυμητών στόχων, και μια περιγραφή των διαφόρων ενεργειών που μπορούσε να κάνει, οι οποίες συσχετιζόνταν με διάφορες προ και μετά – καταστάσεις του κόσμου. Η επίτευξη του στόχου καθοριζόνταν από την επιλογή εκείνων των ενεργειών, που οδηγούσαν καλύτερα στις μετά-καταστάσεις που ήταν τα βήματα προς την επίτευξη του στόχου.

- τα **IPEM** (Integrated, Planning, Execution and Monitoring), **AUTODRIVE** (προσομοίωση κυκλοφοριακού φόρτου), **PHEONIX** (προσομοίωση αντιπυρικού

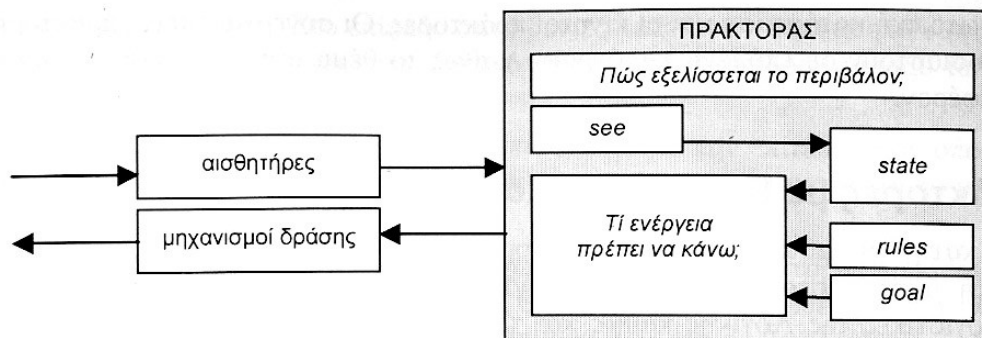
σχεδιασμού) τα οποία κινούνταν σε πιο πολύπλοκα περιβάλλοντα και βασιζόνταν σε ένα μηχανισμό υλοποίησης πλάνων ενεργειών, κάτι που το Strips δεν είχε.

- το **IRMA** των Bratman, Israel και Pollack (Intelligent Resource-bounded Machine Architecture - 1988): το οποίο είχε τέσσερις συμβολικές δομές δεδομένων (βιβλιοθήκη σχεδίων, πεποιθήσεων, επιθυμιών και προθέσεων).

Έτσι την IRMA, υπήρχαν:

- ❖ συμπερασματικός μηχανισμός που μετέφερε στον πράκτορα τα γεγονότα του έξω κόσμου
- ❖ αναλυτής προθέσεων - αναθεωρητής των πεποιθήσεων, που οδηγούσε στην επιλογή του κατάλληλου σχεδίου για την ικανοποίηση των προθέσεων του πράκτορα
- ❖ αναλυτής ευκαιριών - διαθέσιμων επιλογών ο οποίος παρακολουθούσε κάθε τι νέο στο εξωτερικό περιβάλλον
- ❖ μια διαδικασία φιλτραρίσματος των πληροφοριών για την επιλογή της κατάλληλης κάθε φοράς ενέργειας
- ❖ μια διαδικασία σύγκρισης των δεδομένων με την εσωτερική κατάσταση του πράκτορα και επιλογής κατάλληλης ενέργειας

### Σχήμα 12: Αρχιτεκτονική πράκτορα εσωτερικής κατάστασης με στόχους



Πηγή: Vlachavas I. κ.α., 2002

- το **HOMER** των Vere & Bickmore (1990): το οποίο ήταν ένας πρωτότυπος αυτόνομος πράκτορας με ικανότητα κατανόησης φυσικής γλώσσας, σχεδιασμό ενεργειών, και ικανότητες βηματικής κίνησης. Αυτό έπαιρνε εντολές σε μορφή φυσικής γλώσσας και επέλεγε κινήσεις για διάφορες οντότητες σε ένα καναββο που αντιπροσώπευε κάποιο θαλάσσιο κόσμο.

- το **GRATE** του Jennings (1993): το οποίο διαθέτει αρχιτεκτονική με επίπεδα καθορισμού της συμπεριφοράς. Στην πιο απλή μορφή το πρώτο επίπεδο, επίλυσε

προβλήματα που αφορούσαν τον πράκτορα. Ενώ το δεύτερο, επίλυε θέματα επικοινωνίας, συνεργασίας και συντονισμού του πράκτορα με το περιβάλλον και τους άλλους πράκτορες.

## 2.2. Πράκτορες με πεποιθήσεις-επιθυμίες-προθέσεις (BDI Agents)

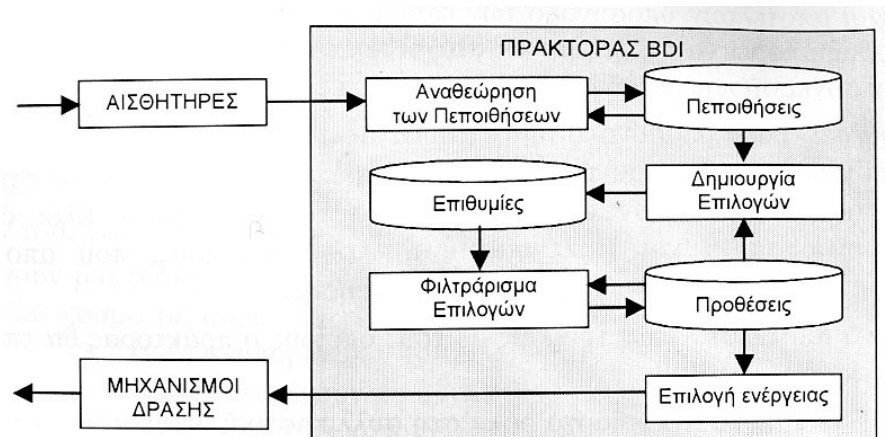
Οι **πράκτορες με πεποιθήσεις - επιθυμίες - προθέσεις** (*belief-desire-intention* ή *BDI agents*) έχουν μια πολυπλοκότερη αναπαράσταση του περιβάλλοντος και σχεδιάζουν (planning) την επίτευξη των στόχων τους.

Η εσωτερική κατάσταση των πρακτόρων αυτών αποτελείται από:

- ❖ *πεποιθήσεις (beliefs)*: αποτελούν την άποψη (view) και τη γνώση που έχει ο πράκτορας για το περιβάλλον του
- ❖ *επιθυμίες (desires)*: αφορούν την κρίση του πράκτορα για τις μελλοντικές καταστάσεις του περιβάλλοντός του, οι οποίες όμως μπορεί να μην είναι τελικά οι επιθυμητές
- ❖ *στόχους (goals)*: αποτελούν υποσύνολο των επιθυμιών, για την επίτευξη των οποίων ο πράκτορας μπορεί να ενεργήσει. Οι στόχοι πρέπει να είναι εφικτοί και να μην περιέχουν συγκρούσεις μεταξύ τους.
- ❖ *προθέσεις (intentions)*: είναι υποσύνολο των στόχων, τις οποίες ο πράκτορας προσπαθεί να επιτύχει σε μια δεδομένη χρονική στιγμή. Επειδή, δεν είναι δυνατό να επιτευχθούν όλοι οι στόχοι ταυτόχρονα, επιλέγεται ένα υποσύνολο τους (προθέσεις), βάσει ορισμένων κριτηρίων ιεράρχησης.
- ❖ *πλάνα (plans)*: αποτελούν τους τρόπους με τους οποίους ο πράκτορας θα επιτύχει τις προθέσεις του.

Το IRMA που αναφέρθηκε πιο πριν εμπεριέχει BDI πράκτορες.

**Σχήμα 13: Αρχιτεκτονική πράκτορα BDI**





Οι προθέσεις παίζουν κυρίαρχο ρόλο στη συλλογιστική διαδικασία ενός πράκτορα BDI. Όταν αυτός διαμορφώσει μια πρόθεση θα πρέπει ταυτόχρονα να αποφασίσει πώς θα την υλοποιήσει. Αυτό έχει ως αποτέλεσμα να δεσμεύεται για τις μελλοντικές ενέργειες του και να αποκλείει την επίτευξη προθέσεων που είναι ασύμβατες με την αρχική.

Οι πεποιθήσεις ενός πράκτορα επηρεάζονται από τις προθέσεις. Αυτός ενεργεί λογικά βάσει αυτών. Ακόμη, το περιβάλλον ζωής του, τον οδηγεί μέχρι εκεί που μπορεί να επιδιώκει την επίτευξη των προθέσεων του. Έτσι, σε στατικά περιβάλλοντα οι τολμηροί πράκτορες έχουν καλύτερη απόδοση. Ενώ, σε ένα ταχέως μεταβαλλόμενο περιβάλλον, οι προσεκτικοί πράκτορες, που επανεξετάζουν τις προθέσεις τους, συχνά ανταποκρίνονται καλύτερα.

Η παρούσα αρχιτεκτονική είναι αποδεκτή μιας και ταιριάζει στην κοινή αντίληψη λειτουργίας αυτόνομων οντοτήτων και έχει ξεκάθαρη αντιστοίχιση των στοιχείων της με λειτουργικά μέρη.

### **3. Συνεργασιακοί Πράκτορες (Collaborative Agents)**

Σε αυτούς του πράκτορες τα βασικά χαρακτηριστικά είναι η αυτονομία και η συνεργασία, για την επίτευξη των στόχων τους. Η κάθε κίνηση ή ενέργειά τους έρχεται ύστερα από διαπραγματεύσεις με άλλους πράκτορες. Άλλα δευτερεύοντα χαρακτηριστικά τους είναι η κοινωνικότητα και επικοινωνία, η ανταπόκριση και η στατικότητα. Επιτυγχάνουν επίσης, εκμάθηση συγκεκριμένων επαναλαμβανόμενων ενεργειών.

Τα συστήματα τα οποία ενσωματώνουν τέτοιους πράκτορες έχουν ως στόχο την εκμετάλλευση των συνολικών δυνατοτήτων όλων των μεμονωμένων πρακτόρων. Τέτοια συστήματα βοηθούν στην επίλυση προβλημάτων:

- ❖ τα οποία απαιτούν αποκεντρωμένη επίλυση (κατανεμημένη επίλυση)
- ❖ όπου τα συνολικά δεδομένα τα οποία θα οδηγήσουν στη βέλτιστη λύση είναι διάσπαρτα
- ❖ που συμμετέχουν πολλά είδη συστημάτων (έμπειρα, συστήματα υποστήριξης αποφάσεων κ.α)

Οι συνεργασιακοί πράκτορες αποτελούν μικρή διαφοροποίηση των πρακτόρων εσωτερικής κατάστασης

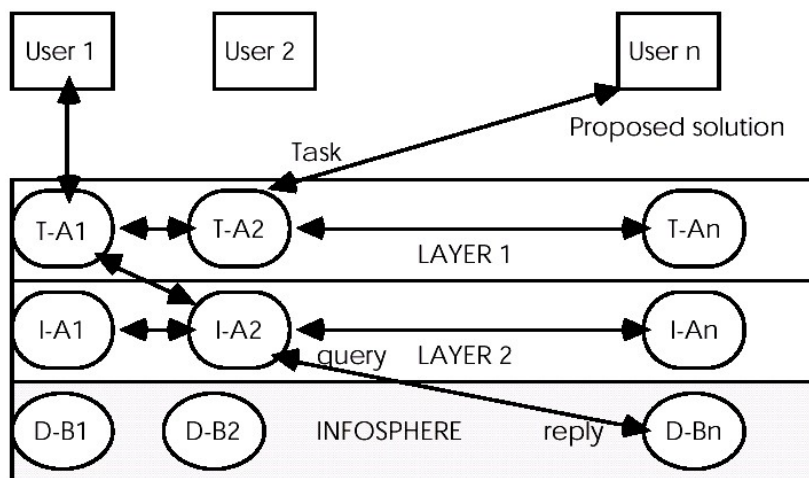
#### **3.1. Το σύστημα Pleiades της CMU**

Το συγκεκριμένο σύστημα εμπνεύστηκαν οι Mitchell & Sycara το 1995, και προσπαθεί με τη χρήση συνεργασιακών πρακτόρων, να αντιμετωπίσει θέματα λήψης αποφάσεων για οργανωσιακούς σκοπούς, με άντληση πληροφοριών από πηγές ετερογενών συστημάτων (infosphere).

Αποτελείται από δύο επίπεδα:

- ❖ το επίπεδο εξειδίκευσης στις εργασίες: περιλαμβάνει τους task-assistants (T-A), που ο καθένας εκτελεί συγκεκριμένη εργασία. Οι T-A συνεργάζονται για να διευθετήσουν πιθανές αντίθετες καταστάσεις. Προσπαθούν να ενοποιήσουν-ολοκληρώσουν τη πληροφορία, που θέλει τελικά το σύστημα να αποδώσει στο χρήστη και έτσι να επιτελέσουν τις εργασίες τους.
- ❖ το επίπεδο εξειδίκευσης στη πληροφορία: περιλαμβάνει τους information-assistants (I-A), οι οποίοι συνεργάζονται μεταξύ τους για να αποδώσουν την πληροφορία που ζητούν οι task-assistants ώστε να επιτελέσουν την εργασία τους. Πηγές των πληροφοριών είναι οι βάσεις δεδομένων (D-B).

**Σχήμα 14: Αρχιτεκτονική του συστήματος Pleiades**



Πηγή: Nwana, 1996

Οι task-assistants κατέχουν γνώση βασισμένη σε ένα μοντέλο, σχετική με:

- ❖ το πεδίο αναφοράς της εργασίας που πρέπει να επιτελέσουν
- ❖ τον τρόπο εκτέλεσης της εργασίας
- ❖ τον τρόπο συλλογής της πληροφορίας
- ❖ το πότε, το πώς και που θα συνεργασθούν με τους άλλους πράκτορες είτε αυτοί είναι task-assistants, είτε information-assistants (δηλ. το τι στρατηγική πρέπει να ακολουθήσουν για την ανάκτηση και απόκτηση της πληροφορίας που χρειάζονται)

Οι task-assistants συχνά μαθαίνουν πώς να εκτελούν εργασίες που τις έχουν ξανακάνει.

Οι information-assistants κατέχουν γνώση βασισμένη σε μοντέλο, σχετική με:

- ❖ τις υπάρχουσες βάσεις δεδομένων
- ❖ τον τρόπο πρόσβαση στις βάσεις δεδομένων
- ❖ τη στρατηγική ανάκτησης της πληροφορίας μέσω της συνεργασίας
- ❖ την επικοινωνία με άλλους πράκτορες

Οι information-assistants αντιδρούν άμεσα, σε συχνά συναντόμενα αιτήματα για συγκεκριμένες πληροφορίες, μιας και έχουν ένα απλοποιημένο μηχανισμό μάθησης.

Κάθε πράκτορας αποτελείται από ένα μοντέλο σχεδιασμού κινήσεων με πεποιθήσεις και γεγονότα. Έχει επίσης ένα προγραμματιστή ενεργειών, ένα μηχανισμό συνεργασίας και μια τεχνική ή σύνολο τεχνικών παρακολούθησης των κινήσεων του. Έτσι, ο κάθε πράκτορας μπορεί να σχεδιάζει και να οργανώνει πλάνα συνεργασίας με άλλους πράκτορες και να παρακολουθεί τις κινήσεις του.

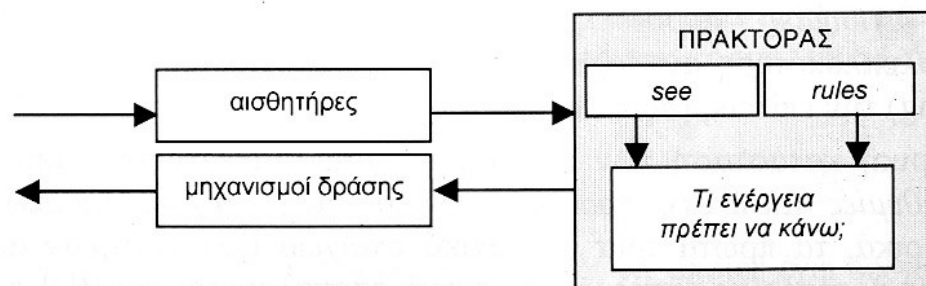
Το συγκεκριμένο σύστημα χρησιμοποιείται κυρίως για την επιτυχή αντιμετώπιση της ανάκτησης, του φιλτραρίσματος και του συνδυασμού δεδομένων-πληροφοριών από καταναμεημένα και ετερογενή συστήματα.

#### 4. Αντιδραστικοί Πράκτορες (Reactive Agents)

Η αρχιτεκτονική των αντιδραστικών πρακτόρων χαρακτηρίζεται από την απουσία εσωτερικής συμβολικής αναπαράστασης του περιβάλλοντος που κινούνται. Οι πράκτορες αντιδρούν στα ερεθίσματα που λαμβάνουν (stimulus/response behavior).

Αποτελούν αποτέλεσμα σκέψης μιας εναλλακτικής σχολής της TN, σύμφωνα με την οποία «η ευφυής συμπεριφορά των τεχνητών συστημάτων θα επιτευχθεί με το συνδυασμό και την αλληλεπίδραση απλών σχετικά μερών (modules) - (physical grounding hypothesis)».

**Σχήμα 15: Αρχιτεκτονική αντιδραστικού πράκτορα**



Πηγή: Vlachavas I. κ.α., 2002

Η συγκεκριμένη αρχιτεκτονική είναι αρκετά απλή. Αυτό σημαίνει ότι οι πράκτορες μπορούν να αλληλεπιδρούν μεταξύ τους και με το περιβάλλον πολύ απλά. Τρία είναι τα βασικά τους χαρακτηριστικά σύμφωνα με τη Maes (1991):

- ❖ η άμεση λειτουργικότητα – με την έννοια της άμεσης αντίδρασης στο ερέθισμα
- ❖ η ανάλυση ενεργειών – δηλαδή, το γεγονός ότι ένας πράκτορας αποτελείται από πολλά αυτόνομα μοντέλα, τα οποία υφίστανται για την εκπλήρωση συγκεκριμένων ενεργειών (αίσθηση, υπολογισμοί, έλεγχος κίνησης).
- ❖ η λειτουργία στο επίπεδο των δεδομένων – δηλαδή, η επικοινωνία των μοντέλων των πρακτόρων σε χαμηλό επίπεδο, βάσει των πληροφοριών που λαμβάνονται από τους αισθητήρες, χωρίς την επαφή με κάποια εσωτερική βάση γνώσης

#### 4.1. Πράκτορες υπαγωγής

Η *αρχιτεκτονική υπαγωγής (subsumption architecture)* θεμελιώθηκε από τον Brooks το 1985 και είχε ως στόχο την ανάπτυξη μηχανισμών αυτόνομων κινητών ρομπότ, με τη χρήση συμπεριφορικής γλώσσας προγραμματισμού.

Για τη δημιουργία των πρακτόρων υπαγωγής ο Brooks θεώρησε ότι:

1. η ευφυής συμπεριφορά επιτυγχάνεται χωρίς την ύπαρξη συμβολισμών στο εσωτερικό του πράκτορα
2. η ευφυής συμπεριφορά επιτυγχάνεται χωρίς την ύπαρξη μοντέλου λογικής
3. η νόηση είναι αναπτυσσόμενη ιδιότητα των πολύπλοκων συστημάτων

Οι βασικές ιδέες δημιουργίας τους είναι:

1. η τοποθεσία και η ενσωμάτωση, δηλαδή ότι η ευφυής συμπεριφορά υφίσταται στον πραγματικό κόσμο και όχι σε συστήματα τα οποία ενσωματώνονται στο μηχανισμό λειτουργίας ενός πράκτορα ή ενός συστήματος
2. η ευφυΐα και η διεύρυνση, δηλαδή ότι η ευφυής συμπεριφορά προκύπτει σαν αποτέλεσμα της αλληλεπίδρασης του πράκτορα με το περιβάλλον του. Η νόηση δεν είναι απομονωμένη ιδιότητα.

Σύμφωνα με την αρχιτεκτονική υπαγωγής, το σύστημα αποτελείται από μια ιεραρχία συμπεριφορών επίτευξης συγκεκριμένης ενέργειας. Κάθε συμπεριφορά

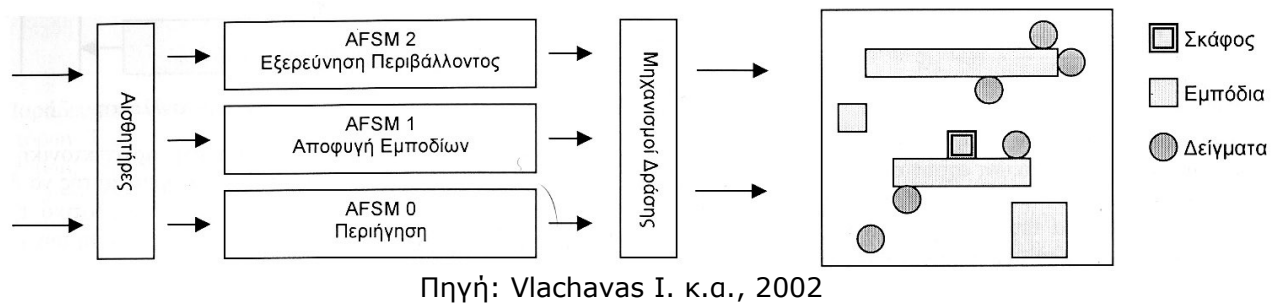
ανταγωνίζεται την άλλη για την επίτευξη του ελέγχου του πράκτορα. Τα χαμηλά επίπεδα αντιπροσωπεύουν απλές συμπεριφορές ενώ πιο σύνθετες επιτυγχάνονται σε ανώτερα επίπεδα.

Πιο συγκεκριμένα, ένας πράκτορας υπαγωγής μπορεί να αποτελείται από *επαυξημένες μηχανές πεπερασμένων καταστάσεων* (Augmented Finite State Machines - **AFSM**). Κάθε μια είναι υπεύθυνη για μια ενέργεια / συμπεριφορά (αποφυγή εμποδίων, κίνηση, κ.α). Κάθε AFSM ενεργοποιείται από τους αισθητήρες του πράκτορα μέσω των οποίων αντιλαμβάνεται το περιβάλλον.

Οι AFSM είναι τοποθετημένες σε επίπεδα και αυτές που ανήκουν σε ανώτερα αναστέλλουν πολλές φορές τη λειτουργία των κατώτερων. Με την αλληλεπίδραση των AFSM, ο πράκτορας αντιδρά γρήγορα στα ερεθίσματα και επιδεικνύει "ευφυή" συμπεριφορά.

Η συμπεριφορά των πρακτόρων υπαγωγής καθορίζεται από την αλληλεπίδραση των μερών τους (modules). Έτσι, οι ενέργειες τους είναι κωδικοποιημένες στα μέρη αυτά εκ των προτέρων. Δηλαδή, η δημιουργία των πλάνων επίτευξης των στόχων τους καθορίζεται από τον προγραμματιστή. Το πρόβλημα που αναδεικνύεται, είναι η δυσκολία στην επέκταση και αποσφαλμάτωση των πρακτόρων, όταν απαιτηθεί αύξηση των αλληλεπιδρώντων επιπέδων. Μια ακόμη δυσκολία, είναι το γεγονός ότι η κατανόηση της συμπεριφορά των πρακτόρων αυτού του είδους, γίνεται μέσα από μεθόδους δοκιμής και λάθους (trial and error).

Παράδειγμα, πρακτόρων υπαγωγής είναι τα ρομπότ που χρησιμοποιούνται για την εξερεύνηση πλανητών. Αυτά έχουν αισθητήρες που ανακαλύπτουν τα εμπόδια και έτσι τα αποφεύγουν. Δε γνωρίζουν τίποτα εκ των προτέρων για το ανάγλυφο του πλανήτη. Επιπλέον, έχουν μηχανισμούς οι οποίοι αποθηκεύουν τα δείγματα πετρωμάτων που βρίσκουν. Τελικά, τα δείγματα επιστρέφονται στο διαστημικό όχημα, το οποίο έστειλε τα ρομπότ στην επιφάνεια του πλανήτη. Αυτό επιτυγχάνεται μέσω ενός σήματος που εκπέμπει το διαστημικό όχημα και ενημερώνει τα ρομπότ για τη θέση στην οποία βρίσκεται. Η παραπάνω συλλογιστική περιγράφεται με την εξής αρχιτεκτονική.

**Σχήμα 16: Αρχιτεκτονική πράκτορα υπαγωγής**

#### 4.2. Άλλα παραδείγματα αντιδραστικών πρακτόρων

- το **PENGI** των Agre & Charman (1986): αυτό ήταν ένα παιχνίδι για Ηλεκτρονικό Υπολογιστή του οποίου ο βασικός χαρακτήρας ελέγχονταν και ενεργούσε βάσει των τρεχόντων γεγονότων. Δηλαδή, οι αποφάσεις που έπαιρνε για καθημερινά γεγονότα που του συνέβαιναν, αποτελούσαν ρουτίνα και αντιμετωπίζονταν σε χαμηλό επίπεδο. Απλά απαιτούνταν περιοδική ενημέρωση για την αντιμετώπιση νέων γεγονότων.

- τα **αυτόματα κατάστασης** (RULER & GAPPS) των Rosenschein & Kaelbling (1991): όπου ο κάθε πράκτορας καθορίζονταν και εξειδικεύονταν με δηλωτικούς όρους. Μια ψηφιακή μηχανή επεξεργάζονταν τις δηλώσεις και αντιλαμβάνονταν τις καταστάσεις των αυτόματων και του κόσμου. Για κάθε πράκτορα-αυτόματο ορίζονταν η αντιληπτική του ικανότητα (Ruler) και ο τρόπος ενέργειάς του (Gapps). Το κομμάτι RULER του κάθε πράκτορα λάμβανε **α)** τις εισόδους του περιβάλλοντος προς τον πράκτορα (όταν το bit είναι 1 τότε On και πχ τότε βρέχει), **β)** μια ομάδα στατικών γεγονότων (πχ όταν βρέχει το χρώμα είναι υγρό) και **γ)** μια αναφορά των αλλαγών των καταστάσεων του κόσμου (πχ αν το χρώμα είναι υγρό αυτό θα αλλάξει μόνο αν βγει ήλιος)

Το κομμάτι GAPPS του κάθε πράκτορα αφού λάμβανε τις εισόδους, έθετε έναν υψηλού επιπέδου στόχο, τους κανόνες αναγωγής στο στόχο (επίτευξη του στόχου) και ενεργοποιούσε το αυτόματο για να τα κατανοήσει όλα τα παραπάνω. Ο μηχανισμός μετάφρασης των παραπάνω δεδομένων εμπεριείχε διαχείριση συμβόλων μόνο κατά τη μετάφραση. Κανένας από τους μηχανισμούς του πράκτορα δε βασιζόταν σε εκφράσεις συμβολικής αναπαράστασης του οτιδήποτε.

- η **δικτυακή αρχιτεκτονική** της Maes (1991): όπου ο κάθε πράκτορας ορίζονταν από διάφορα συναγωνιστικά μοντέλα. Αυτή θυμίζει αρκετά τους πράκτορες υπαγωγής σε συνδυασμό με το STRIPS. Κάθε μοντέλο καθορίζεται από: **i.** τις προ και μετά καταστάσεις, **ii.** ένα επίπεδο ενεργοποίησης, που δίνει το δείκτη

σπουδαιότητας χρήσης του μοντέλου και **iii.** τον πράκτορα σε μια συγκεκριμένη κατάσταση.

Όσο πιο μεγάλος ο δείκτης σπουδαιότητας ενός μοντέλου τόσο πιο κοντά είναι αυτό στο να επηρεάσει τη συμπεριφορά του πράκτορα. Αυτά τα μοντέλα αποτελούν ένα δίκτυο και συνδέονται ανάλογα με τις προ και μετά κοινές καταστάσεις. Καθώς ένας πράκτορας εκτελείται, το αποτέλεσμα μπορεί να είναι ένας μηχανισμός δράσης ή η αύξηση του επιπέδου ενεργοποίησης ενός μοντέλου. Η συγκεκριμένη αρχιτεκτονική θυμίζει πολύ τα νευρωνικά δίκτυα.

## 5. Υβριδικοί Πράκτορες (Hybrid Agents)

Ο συνδυασμός των πρακτόρων εσωτερικής κατάστασης και των αντιδραστικών αποτέλεσε τη δημιουργία μιας τρίτης αρχιτεκτονικής με βασικούς εκπροσώπους της, τους *υβριδικούς πράκτορες (hybrid agents)*. Αυτοί συνδυάζουν τα πλεονεκτήματα των δύο προηγούμενων κατηγοριών. Αποτελούν μια οντότητα στην οποία υπάρχουν τουλάχιστον δύο υποσυστήματα.

Το πρώτο περιλαμβάνει ένα μοντέλο συμβολικής απεικόνισης του περιβάλλοντα χώρου κίνησης του πράκτορα, αναπτύσσει πλάνα και καταλήγει σε αποφάσεις.

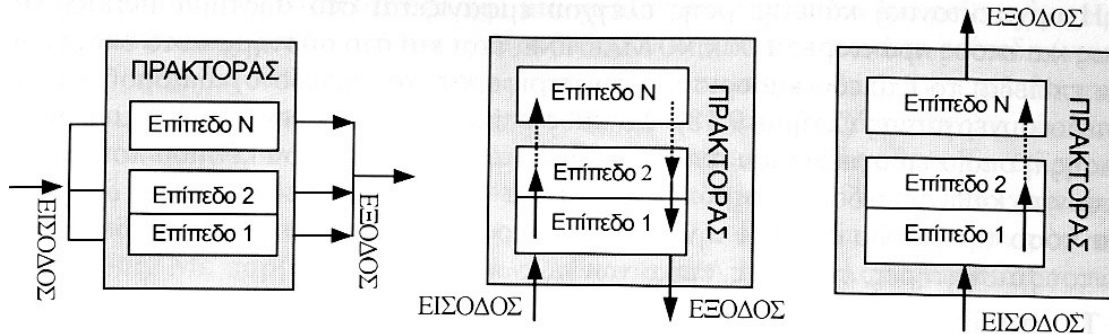
Το δεύτερο είναι υπεύθυνο για την αντιδραστική συμπεριφορά του πράκτορα στα γεγονότα που τυχαίνουν, χωρίς αυτή η αντίδραση να απαιτεί πολύπλοκη λογική.

Συνήθως, το αντιδραστικό υποσύστημα κυριαρχεί του υποσυστήματος εσωτερικής κατάστασης, με αποτέλεσμα την άμεση αντίδραση του πράκτορα στις αλλαγές. Για τον έλεγχο της λειτουργίας του πράκτορα υπάρχουν επίπεδα ελέγχου λειτουργίας τα οποία οργανώνονται ιεραρχικά. Τα υψηλότερα σε ιεραρχία επίπεδα αντιμετωπίζουν την επίτευξη των μακροπρόθεσμων στόχων του πράκτορα. Τα χαμηλότερα επίπεδα στην ιεραρχία επιλέγουν άμεσα μηχανισμούς δράσης στα γεγονότα που συναντούν.

Ο έλεγχος ροής των δεδομένων που δέχεται ο πράκτορας μπορεί να είναι:

- ❖ οριζόντιος, όπου όλα τα επίπεδα συνδέονται στους αισθητήρες εισόδου και στους μηχανισμούς δράσης
- ❖ κάθετος, όπου ένα μόνο επίπεδο συνδέεται στους αισθητήρες και ένα στους μηχανισμούς δράσης

Σχήμα 17: Αρχιτεκτονικές υβριδικών πρακτόρων



Πηγή: Vlachavas I. κ.α., 2002

### 5.1. Σύστημα διαδικαστικής λογικής (Procedural Reasoning System)

Αυτό το είδος υβριδικής αρχιτεκτονικής το ανέπτυξαν οι Georgeff & Lansky το 1987. Είναι μια μετεξέλιξη του IRMA. Το PRS αποτελείται από μια βιβλιοθήκη πλάνων, πεποιθήσεων, επιθυμιών και προθέσεων.

- ❖ οι πεποιθήσεις είναι γεγονότα του περιβάλλοντος ή της κατάστασης του πράκτορα
- ❖ οι επιθυμίες είναι συμπεριφορές
- ❖ τα πλάνα είναι οι γνωστικές περιοχές του συστήματος που ενεργοποιούνται ανάλογα την κατάσταση του πράκτορα (δηλαδή τότε τα έχει αυτός ανάγκη να ενεργοποιηθούν- πχ για την επίτευξη ενός στόχου ή για την αντίδραση σε δεδομένα που αυτός λαμβάνει). Επιπλέον, τα πλάνα μπορούν να είναι άμεσα αντιδραστικά προς τις προσλαμβάνουσες από το περιβάλλον
- ❖ οι προθέσεις είναι κάθε φορά το σύνολο των ενεργοποιημένων πλάνων

Το συνολικό έλεγχο τον κάνει ένας διερμηνέας που ενημερώνει μετά από κάθε ενέργεια του πράκτορα τις πεποιθήσεις, τα πλάνα και τις εκτελέσιμες ενέργειες αυτού. Τέτοιοι πράκτορες χρησιμοποιήθηκαν στον αγώνα κατάκτησης του διαστήματος.

### 5.2. TOURING MACHINE

Οι μηχανές Touring ήταν αποτέλεσμα της σκέψης του Ferguson το 1992. Προτάθηκαν για την καθοδήγηση αυτόνομων οχημάτων μεταξύ σημείων σε περιβάλλον που υπάρχουν δρόμοι που κινούνται παρόμοιοι πράκτορες. Είναι υβριδική αρχιτεκτονική οριζόντιας ροής. Περιέχει N διαφορετικά αλληλοσυμπληρούμενα επίπεδα συμπεριφοράς και ένα μεσολαβητή (mediator) ή



σημείο κεντρικού ελέγχου, που καθορίζει ποιο επίπεδο κάθε φορά ελέγχει τον πράκτορα.

Τρία είναι τα βασικά επίπεδα στην πιο απλή μορφή μιας τέτοιας μηχανής. Αυτά δέχονται ταυτόχρονα δεδομένα εισόδου από τους αισθητήρες και προτείνουν ενέργειες που πρέπει να κάνει ο πράκτορας.

Το **αντιδραστικό επίπεδο** (*reactive layer*) παρακολουθεί τις αλλαγές που γίνονται στο περιβάλλον και προτείνει άμεσες ενέργειες. Η υλοποίησή του ακολουθεί εκείνη της αρχιτεκτονικής υπαγωγής.

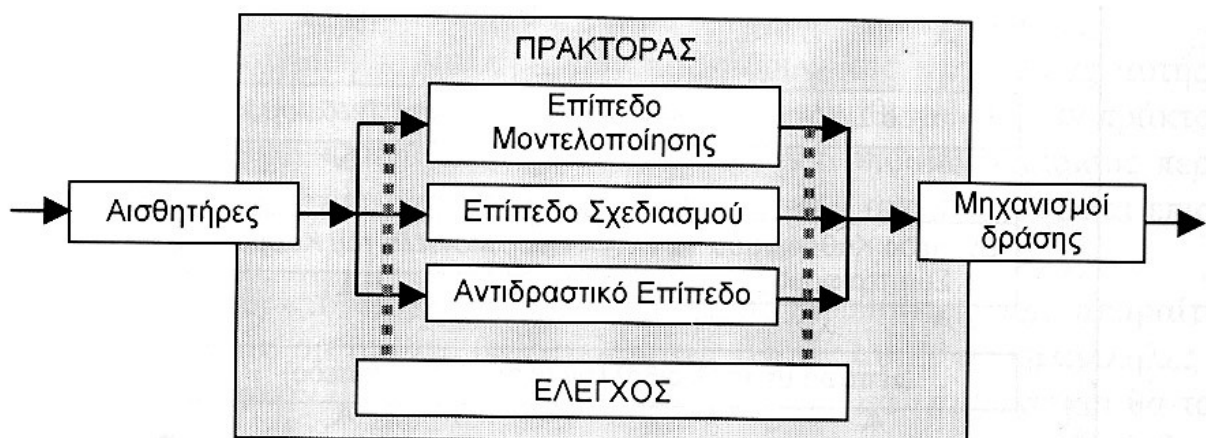
Το **επίπεδο σχεδιασμού** (*planning layer*) συντάσσει πλάνα δράσης και επιλέγει ενέργειες ώστε να επιτύχει ο πράκτορας τους στόχους του. Αυτό αποτελείται από δύο συστατικά: τον σχεδιαστή, και το μηχανισμό εστίασης-προσήλωσης.

Ο σχεδιαστής προσαρμόζει τα πλάνα δράσης και επιλέγει τον τρόπο εκτέλεσης τους. Αναζητά και βρίσκει σε μια βιβλιοθήκη από ήδη υπάρχοντα πλάνα κάποιο το οποίο το προσαρμόζει στο περιβάλλοντα χώρο (τοπολογία, καταστάσεις, συνθήκες) με σκοπό την επίτευξη του στόχου του πράκτορα.

Ο μηχανισμός εστίασης-προσήλωσης περιορίζει τον όγκο της πληροφορίας με την οποία θα ασχοληθεί ο σχεδιαστής, για να επιτυγχάνεται ευελιξία και γρήγορη απόκριση του πράκτορα.

Το **επίπεδο μοντελοποίησης** (*modeling layer*) είναι υπεύθυνο για την αναπαράσταση όλων των αντικειμένων στο περιβάλλον του πράκτορα. Επίσης, προβλέπει τις πιθανές συγκρούσεις συμφερόντων με άλλους πράκτορες και προτείνει τους στόχους που θα τεθούν για την αποφυγή των συγκρούσεων.

**Σχήμα 18: Αρχιτεκτονική μηχανής TOURING**



Πηγή: Vlachavas I. κ.α., 2002

Ο πράκτορας τελικά ενεργεί μέσω του συστήματος ελέγχου που είναι υπεύθυνο για την επιλογή της κατάλληλης ενέργειας ανάμεσα στις προτεινόμενες από τα τρία επίπεδα. Πολλές φορές το σύστημα ελέγχου αποκρύπτει κάποια δεδομένα εισόδου από ένα επίπεδο, αφήνοντας κάποιο άλλο να προτείνει ενέργεια για τη συγκεκριμένη κατάσταση.

### 5.3. COSY

Η αρχιτεκτονική COSY είναι μια υβριδική BDI αρχιτεκτονική που εμπεριέχει στοιχεία από το PRS και από το IRMA και αναπτύχθηκε-εξελίχθηκε από τους Burmeister, Sundermeyer και Haddadi (1992-1994). Το σύστημα που ενσωμάτωσε αυτή την αρχιτεκτονική είναι το DASEDIS.

Τα τμήματα από τα οποία αποτελείται είναι:

- ❖ αισθητήρες, που δέχονται τις εισόδους από το περιβάλλον
- ❖ κινητήρες, που επιτρέπουν τη κίνηση του πράκτορα
- ❖ επικοινωνίες, που επιτρέπουν στον πράκτορα να αποστέλλει και να δέχεται μηνύματα
- ❖ νόηση, που περιλαμβάνει μακροπρόθεσμους στόχους, συμπεριφορές, ευθύνες και στοιχεία ελέγχου, τα οποία λαμβάνουν μέρος στη λογική συμπερασματολογία και στην απόφαση για κάθε τι.
- ❖ πρόθεση, που είναι υπεύθυνη για την μεσολάβηση μεταξύ των προθέσεων και των πεποιθήσεων του πράκτορα, ώστε να επιλεγεί η κατάλληλη ενέργεια από αυτόν. Μέσα σε αυτό το τμήμα βρίσκονται η βάση γνώσης του πράκτορα, οι τρόποι εκτέλεσης διαφόρων scripts, πρωτοκόλλων και ο τρόπος κατάληξης σε συγκεκριμένη λογική, απόφαση και αντίδραση. Για την εκτέλεση των παραπάνω υπάρχει ένα διάγραμμα παρουσίας τους και ένα φίλτρο το οποίο καθορίζει τι θα εκτελεστεί (αποτέλεσμα του συναγωνισμού των παραπάνω).

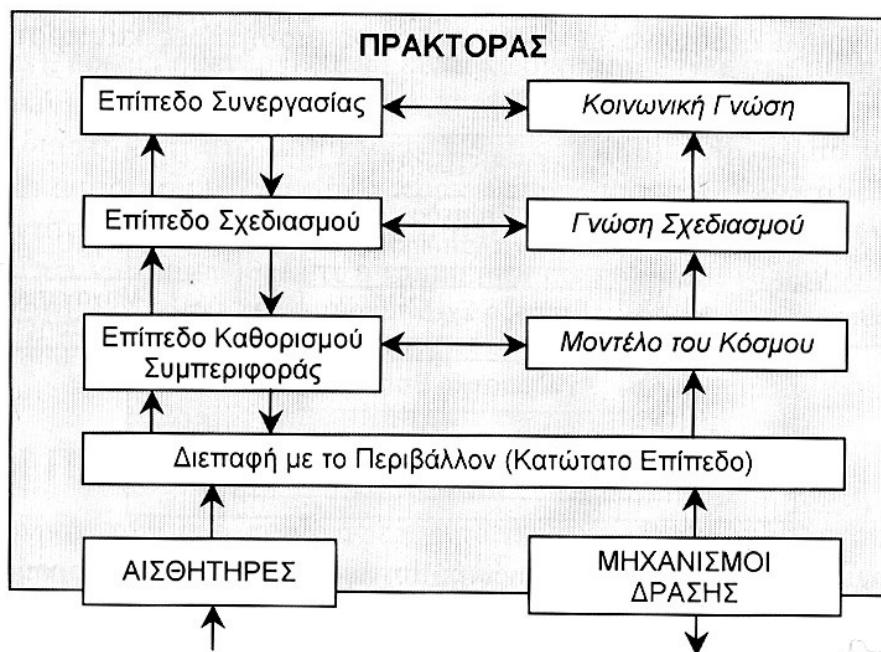
### 5.4. INTERRAP

Το INTERRAP ήταν αποτέλεσμα σκέψης του Muller & Pischel, το 1994. Είναι υβριδική αρχιτεκτονική κάθετης ροής ελέγχου. Στο σύστημα αυτό υπάρχουν τρία επίπεδα. Σε καθένα από αυτά είναι προσαρτημένη μια βάση γνώσης, με αναπαράσταση των στοιχείων του κόσμου που το ενδιαφέρουν. Ένα κατώτατο επίπεδο διαχειρίζεται τα δεδομένα που εισέρχονται και εξέρχονται από/προς τον

πράκτορα προς/από το περιβάλλον (δεδομένα από τους αισθητήρες, ενέργειες, επικοινωνία, κλπ). Οι πράκτορες λειτουργούν ως εξής:

- ❖ για τις χαμηλού επιπέδου αντιδραστικές ενέργειες του πράκτορα, υπεύθυνο είναι το επίπεδο καθορισμού συμπεριφοράς (*behaviour layer*). Η αντίστοιχη βάση γνώσης περιέχει χαμηλού επιπέδου πληροφορίες για τον κόσμο.
- ❖ για τη δημιουργία πλάνων για την επίτευξη των στόχων του πράκτορα, υπεύθυνο είναι το επίπεδο σχεδιασμού (*planning layer*). Η αντίστοιχη βάση γνώσης περιέχει μια βιβλιοθήκη πλάνων.
- ❖ για τη δημιουργία πλάνων συνεργασίας με άλλους πράκτορες, υπεύθυνο είναι το επίπεδο συνεργασίας (*cooperation layer*.) Η βάση γνώσης του περιέχει πληροφορίες για τους άλλους πράκτορες.

**Σχήμα 19: Αρχιτεκτονική *interrap***



Πηγή: Vlachavas I. κ.α., 2002

Η ροή ελέγχου κινείται από κάτω προς τα πάνω και στη συνέχεια κατεβαίνει προς τα κάτω. Αυτή βασίζεται:

- ❖ στην ενεργοποίηση από κάτω προς τα πάνω (*bottom-up activation*), η οποία δηλώνει ότι μια κατάσταση αντιμετωπίζεται με ενεργοποίηση των επιπέδων. Η αντιμετώπιση γίνεται, ξεκινώντας από το κατώτερο επίπεδο και συνεχίζει στο αμέσως ανώτερο, μέχρι κάποιο από αυτά να είναι ικανό να διαχειριστεί τη συγκεκριμένη κατάσταση. Μόλις συμβεί αυτό η ενεργοποίηση σταματά.

- ❖ στην εκτέλεση από πάνω προς τα κάτω (top-down execution), η οποία δηλώνει ότι η υλοποίηση των στόχων ενός επιπέδου, γίνεται με τη χρήση των δυνατοτήτων που προσφέρει το αμέσως κατώτερο.

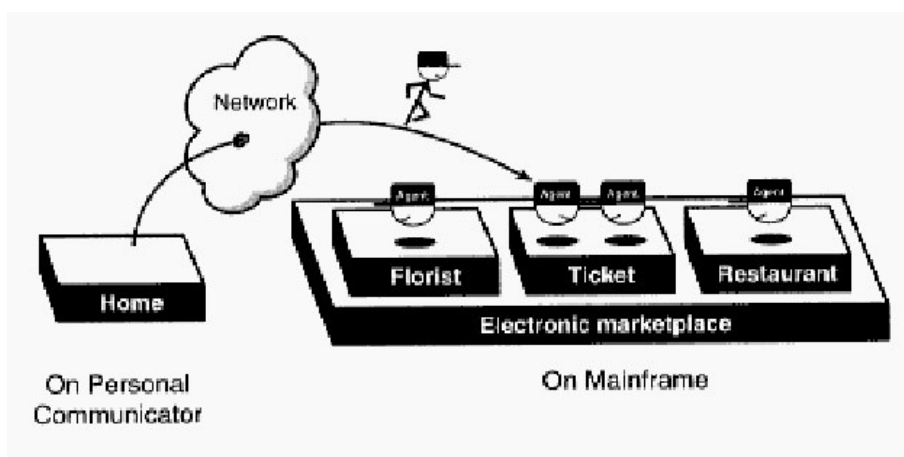
## 6. Κινητοί Πράκτορες (Mobile Agents)

Οι κινητοί πράκτορες μετακινούνται σε δικτυακό περιβάλλον με σκοπό να πετύχουν τους στόχους τους. Αποτελούν διεργασίες (software processes), οι οποίες εκτελούνται μεταφερόμενες στους υπολογιστές ενός δικτύου (LAN ή WAN). Έχουν πάντοτε τη δυνατότητα επικοινωνίας με άλλους πράκτορες ή με άλλες οντότητες παροχής πληροφοριών ή υπηρεσιών. Πολλές φορές υιοθετούν την αρχιτεκτονική πελάτη - εξυπηρετητή, πράγμα που δεν αποκλείει το γεγονός, ότι κάποιος πράκτορας μπορεί να είναι πελάτης σε κάποιον άλλο και εξυπηρετητής σε ένα τρίτο.

Έστω, ένας κινητός πράκτορας ο οποίος είναι υπεύθυνος για την αναζήτηση και εύρεση κάποιων τίτλων CD, συγκεκριμένης θεματολογίας και με περιορισμούς στην τιμή αγοράς:

**βήμα 1<sup>ο</sup>:** Αρχικά αυτός θα έφευγε από τον Η/Υ του χρήστη που θέλει να βρει τα CD και θα μεταφέρονταν στον πρώτο υπολογιστή που θα έβρισκε κάποια βάση δεδομένων.

**Σχήμα 20: Λειτουργία κινητού πράκτορα**



Πηγή: [www-2.cs.cmu.edu/~softagents/](http://www-2.cs.cmu.edu/~softagents/)

**βήμα 2<sup>ο</sup>:** Εκεί θα έκανε τοπικά όλες τις αναζητήσεις και φιλτραρίσματα βάσει των αρχικών περιορισμών-κριτηρίων.

**βήμα 3°:** Έπειτα θα μεταφέρονταν σε κάποια επόμενη βάση δεδομένων επαναλαμβάνοντας τα προηγούμενα κ.ο.θ.

**τελευταίο βήμα:** Στο τέλος θα επέστρεφε στον υπολογιστή του χρήστη με τα αποτελέσματα της αναζήτησης.

Στην παραπάνω περίπτωση απαιτείται σύνδεση του Η/Υ του χρήστη με το δίκτυο μόνο κατά την "αναχώρηση" και την "άφιξη" του πράκτορα.

Αν ο πράκτορας ήταν στατικός:

**βήμα 1°:** θα πήγαινε στη πρώτη βάση δεδομένων και εκεί θα έκανε τις αναζητήσεις και φιλτραρίσματα. Όταν τελειώνει θα επέστρεφε τα αποτελέσματα στο χρήστη.

**βήμα 2°:** Έπειτα θα έκανε το ίδιο για τη δεύτερη κ.ο.θ.

Το σημαντικό στην πιο πάνω περίπτωση είναι ότι ο Η/Υ του χρήστη, πρέπει να είναι συνδεδεμένος στο δίκτυο όλη την ώρα, κατά την οποία γίνονται οι εργασίες της αναζήτησης και του φιλτραρίσματος στη κάθε βάση δεδομένων.

Οι κινητοί πράκτορες είναι πολύ χρήσιμοι γιατί:

- ❖ μειώνουν το κόστος επικοινωνίας και τον όγκο των δεδομένων που μεταφέρονται μέσω του δικτύου
- ❖ η λειτουργία του συνολικού συστήματος ανεξαρτητοποιείται εν μέρει από τη διαθεσιμότητα του δικτύου
- ❖ υπάρχει ασύγχρονη εκτέλεση του κινητού πράκτορα και των υπολοίπων εφαρμογών του χρήστη (μετά την αποστολή του πράκτορα, το σύστημα του χρήστη μπορεί να συνεχίσει την εκτέλεση άλλων λειτουργιών)

Η υλοποίηση και χρησιμοποίηση κινητών πρακτόρων παρουσιάζει:

- ❖ δυσκολίες στην ανάπτυξη τους, γιατί τίθενται ζητήματα ασφάλειας του κάθε πράκτορα και του συστήματος που θα τον φιλοξενεί
- ❖ αυξημένο κόστος στη μεταφορά τους
- ❖ δυσκολία στον έλεγχο της εκτέλεσης τους καθώς δεν εντοπίζεται εύκολα η θέση και η τρέχουσα κατάσταση του καθενός

Για την υλοποίηση των κινητών πρακτόρων μπορούν να χρησιμοποιηθούν διάφορες γλώσσες προγραμματισμού όπως οι:

- ❖ TELESCRIPT

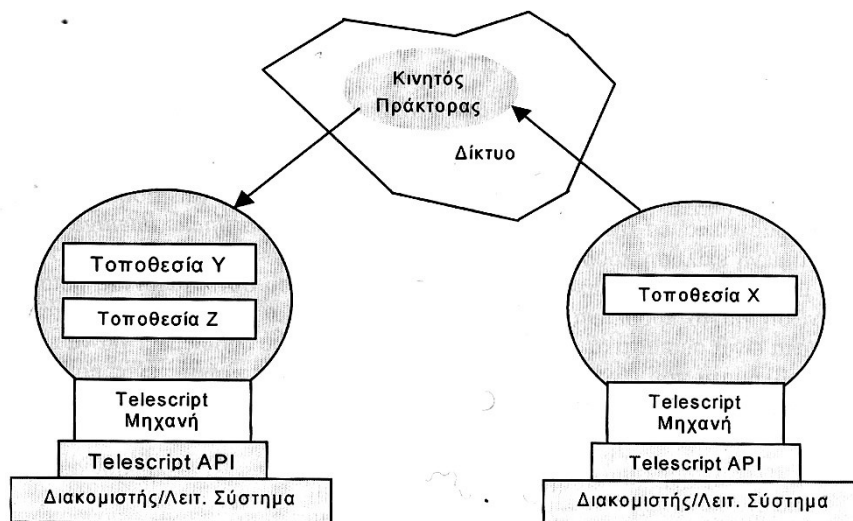
- ❖ JAVA
- ❖ AGENT-TCL
- ❖ SAFE-TCL
- ❖ C ή C++

### 6.1. Οι πράκτορες Telescript

Η General Magic, που βρίσκεται στη Silicon Valley, έχει αναπτύξει μια γλώσσα προγραμματισμού κινητών πρακτόρων, την οποία έχει ονομάσει Telescript. Οι πράκτορες αποτελούν κομμάτια κώδικα, που στέλνονται σε όλα τα μήκη και πλάτη του δικτύου (White 1994). Η βασική φιλοσοφία της ανάπτυξης τέτοιων προγραμμάτων πρακτόρων, είναι ότι αυτοί αποτελούν ηλεκτρονικές προεκτάσεις των χρηστών. Οι πράκτορες είναι ικανοί να μαζεύουν επιλεκτικά πληροφορίες, να κάνουν διαπραγματεύσεις και να εκτελούν οικονομικές συναλλαγές εκ μέρους των χρηστών τους. Παραμετροποιούνται καταλλήλως και έχουν δόση «ευφυΐας», με την έννοια ότι είναι δυνατό να προσαρμόζονται στις συνθήκες του περιβάλλοντος και να ενεργούν ανάλογα με τα «μέρη» στα οποία βρίσκονται. (π.χ. αλλάζοντας από ένα κύριο πλάνο δράσης στο εναλλακτικό του)

Η TELESCRIPT είναι αντικειμενοστραφής γλώσσα προγραμματισμού που επιτρέπει την ανάπτυξη κατανεμημένων εφαρμογών. Κάθε εφαρμογή που αναπτύσσεται στη συγκεκριμένη γλώσσα αποτελείται από **πράκτορες** (TELESCRIPT *agents*), που κινούνται μέσω **διαδικασιών μετακίνησης**, σε έναν κόσμο που περιέχει **TELESCRIPT μηχανές** (TELESCRIPT *engines*) και **τοποθεσίες** (*places*).

**Σχήμα 21: Αρχιτεκτονική πρακτόρων TELESCRIPT**



οι πράκτορες είναι διεργασίες οι οποίες δεν περιέχουν άλλες διεργασίες αλλά μετακινούνται από τη μια τοποθεσία στην άλλη

οι TELESCRIPT μηχανές είναι διερμηνευτές (interpreters) που φιλοξενούν διεργασίες

οι τοποθεσίες είναι διεργασίες που προσφέρουν υπηρεσίες (functionalities) και μπορούν να περιέχουν και άλλες διεργασίες

η διαδικασία μετακίνησης (go) απαιτεί την ύπαρξη τοποθεσίας στόχου του πράκτορα. Περιλαμβάνει:

1. το «πακετάρισμα» του **κώδικα** του πράκτορα, των **δεδομένων** του, της **στοίβας** και του **δείκτη εντολών** από τη μηχανή TELESCRIPT.
2. την αποστολή του πράκτορα στην τοποθεσία στόχο
3. το «ξεπακετάρισμα» και τον έλεγχο της αυθεντικότητάς του από τη μηχανή TELESCRIPT της τοποθεσίας στόχου
4. τη συνέχεια της εκτέλεσης του από το σημείο διακοπής

Για να μπορέσει ένας πράκτορας να χρησιμοποιήσει τις υπηρεσίες μιας τοποθεσίας, θα πρέπει να βρίσκεται σε αυτή, ενώ μπορεί να επικοινωνήσει μόνο με πράκτορες που βρίσκονται στην ίδια τοποθεσία.

Παραδείγματα τοποθεσιών είναι τα διάφορα «ηλεκτρονικά γραμματοκιβώτια», ημερολόγια, ηλεκτρονικά μαγαζιά, σημεία συνάντησης στο δίκτυο, κλπ.

## 6.2. Τα AGLETS

Άλλος τρόπος υλοποίησης κινητών πρακτόρων είναι ο προγραμματισμός τους σε JAVA. Ένα πρόγραμμα σε JAVA μεταφράζεται σε ενδιάμεσο διερμηνευμένο κώδικα (bytecode), ο οποίος μεταφέρεται και εκτελείται από μια εικονική μηχανή που συνδέεται σε ένα υπολογιστικό περιβάλλον (JVM). Χαρακτηριστικό εργαλείο ανάπτυξης κινητών πρακτόρων που βασίζονται στην JAVA είναι το περιβάλλον ανάπτυξης των AGLETS, της IBM Research.

Το περιβάλλον αυτό προσφέρει δυνατότητες υλοποίησης:

- ❖ της διαδικασίας μετακίνησης πρακτόρων
- ❖ της επικοινωνίας μεταξύ τους
- ❖ των διαδικασιών ασφάλειας

Οι AGLETS είναι κινητοί πράκτορες και αποτελούνται από μεθόδους και κλάσεις που υλοποιούν τα παραπάνω. Μαζί με τον κώδικά τους μεταφέρεται η κατάσταση

του προγράμματος (πράκτορα) και συνεχίζεται η εκτέλεση, από το σημείο που είχε σταματήσει πριν τη μετακίνηση. Οι AGLETS κινούνται σε ένα περιβάλλον που αποτελείται από εικονικές μηχανές (AGLET Host) και δρουν για την επίτευξη των στόχων τους. Υπάρχουν δύο είδη εικονικών μηχανών το AGLET VIEWER (με μικρή ισχύ) και το AGLET SERVER (ισχυρή μηχανή προσανατολισμένη στην παροχή υπηρεσιών)

## 7. Πράκτορες Διεπαφής (Interface Agents)

Οι **πράκτορες διεπαφής** (interface agents) αποτελούν προσωπικούς βοηθούς (personal assistants) του χρήστη. Η Maes που καθιέρωσε αυτή την αρχιτεκτονική πρακτόρων, υποστηρίζει ότι βασικό της χαρακτηριστικό είναι η συνεργασία των πρακτόρων αποκλειστικά και μόνο με το χρήστη. Η επικοινωνία και η συνεργασία με αυτόν δεν απαιτεί συγκεκριμένη και καθορισμένη γλώσσα επικοινωνίας, όπως πρέπει να γίνεται για την επικοινωνία μεταξύ των πρακτόρων. Πολύ γνωστό παράδειγμα αυτής της αρχιτεκτονικής είναι ο βοηθός ή οι μάγοι των εφαρμογών του Microsoft Office.

Η βασική ιδέα δημιουργίας των πρακτόρων διεπαφής προήλθε από τη παρατήρηση των ήδη υπάρχουσών εφαρμογών, που για να λειτουργήσουν και να αναγκάσουν τον Η/Υ να εκτελέσει κάποια επεξεργασία, απαιτείται η παρέμβαση του χρήστη. Στην ουσία οι πράκτορες διεπαφής δημιουργούν μια διαίσθηση, η οποία τους κάνει να προβλέπουν το τι θέλει ο χρήστης να κάνει μέσα από την εφαρμογή. Η μελλοντική εξέλιξή τους θα έχει ως αποτέλεσμα την από κοινού με το χρήστη παρακολούθηση, έλεγχο και εκτέλεση εντολών μέσα από μια εφαρμογή. Όλα τα παραπάνω θα γίνονται με σκοπό την υλοποίηση των επιθυμιών του χρήστη.

Οι πράκτορες διεπαφής βοηθούν το χρήστη να χρησιμοποιεί και να εκμεταλλεύεται στο μέγιστο της δυνατότητες μιας εφαρμογής. Είναι τμήματα του προγράμματος και συνδέονται με το κομμάτι της διεπαφής αυτού με το χρήστη (το ΓΠΔ). Αυτοί μαθαίνουν:

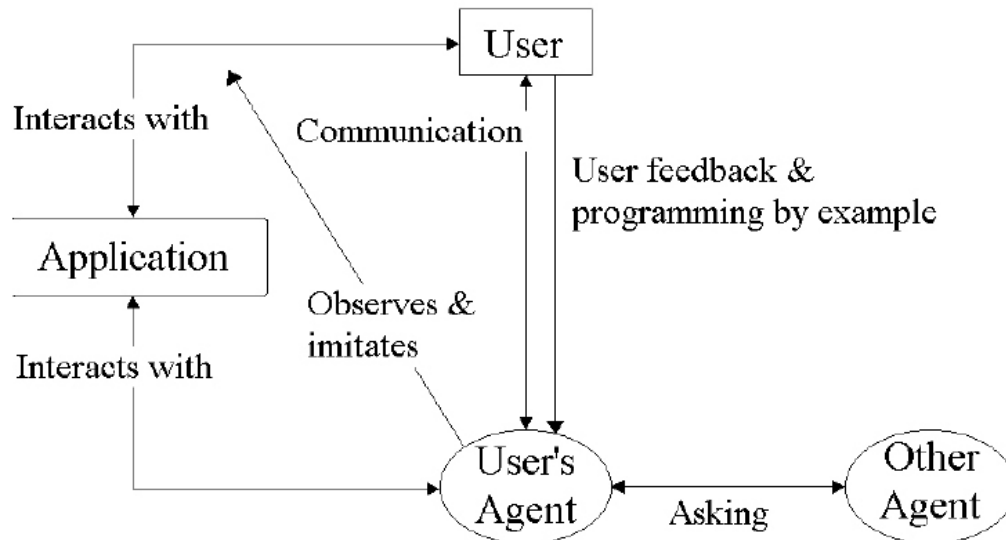
- ❖ παρακολουθώντας και μιμούμενοι το χρήστη
- ❖ όντας ευαίσθητοι και λαμβάνοντας τις ιδιαίτερες προτιμήσεις του χρήστη, κατά την περιήγηση του στο πρόγραμμα
- ❖ δεχόμενοι εντολές από το χρήστη
- ❖ συμβουλευόμενοι άλλους πράκτορες

Οι συγκεκριμένοι πράκτορες δε συνεργάζονται με άλλους πράκτορες για την



επίτευξη κοινού στόχου. Χρησιμοποιούν μηχανική μάθηση για να παρουσιάσουν μια ψευδοέξυπνη συμπεριφορά, βοηθώντας το χρήστη, σε κάθε χρονική στιγμή να επιτυγχάνει τους στόχους του.

**Σχήμα 22: Λειτουργία πρακτόρων Διεπαφής**



Πηγή: Nwana, 1996

Τέτοιοι πράκτορες υπάρχουν:

- ❖ σε εφαρμογές βοηθών και οδηγών εκμάθησης προγραμμάτων
- ❖ σε εφαρμογές εύρεσης πληροφοριών όπως η **Letizia** (Liebermann, 1995). Αυτή είναι μια εφαρμογή που λειτουργεί παρόμοια με τις μηχανές αναζήτησης του διαδικτύου. Η αναζήτηση πληροφοριών που επιτυγχάνει, είναι Breadth-first και όχι depth-first, όπως κάνουν οι μηχανές αναζήτησης. Επιπλέον, μαντεύει τις επιθυμίες του χρήστη, από τη προηγούμενη συμπεριφορά αυτού κατά τις άλλες παρελθούσες αναζητήσεις που εκτέλεσε, αφού παρακολουθεί ανελλιπώς το φυλλομετρητή του. Έτσι, στα αποτελέσματα που βρίσκει, ενσωματώνει την εμπειρία της από τις παρελθούσες αναζητήσεις του χρήστη και φιλτράρει την τελική πληροφορία, εμφανίζοντας εκείνη, που είναι πιο κοντά στις επιθυμίες και επιλογές του χρήστη.
- ❖ σε εφαρμογές που βοηθούν τους χρήστες να κανονίσουν το ημερήσιο πρόγραμμα τους (CALENDAR AGENT, CALENDAR APRENTICE) μαθαίνοντας με τη πάροδο του χρόνου τις προτιμήσεις τους
- ❖ σε εφαρμογές στην παρακολούθηση ασθενών
- ❖ σε παιχνίδια και εφαρμογές προσομοίωσης δραστηριοτήτων
- ❖ στο ηλεκτρονικό εμπόριο, με εύρεση προϊόντων που θα αρέσουν στο χρήστη ή δημιουργίας αγορών (KASBAH – Chavez & Maes, 1996)

- ❖ στην ηλεκτρονική αλληλογραφία, για έλεγχο και απόρριψη εκείνης που είναι αδιάφορη ή επικίνδυνη, ή για αναζήτηση συγκεκριμένης θεματολογίας σε αυτή (renembrance agent- Rhodes & Starner, 1996)

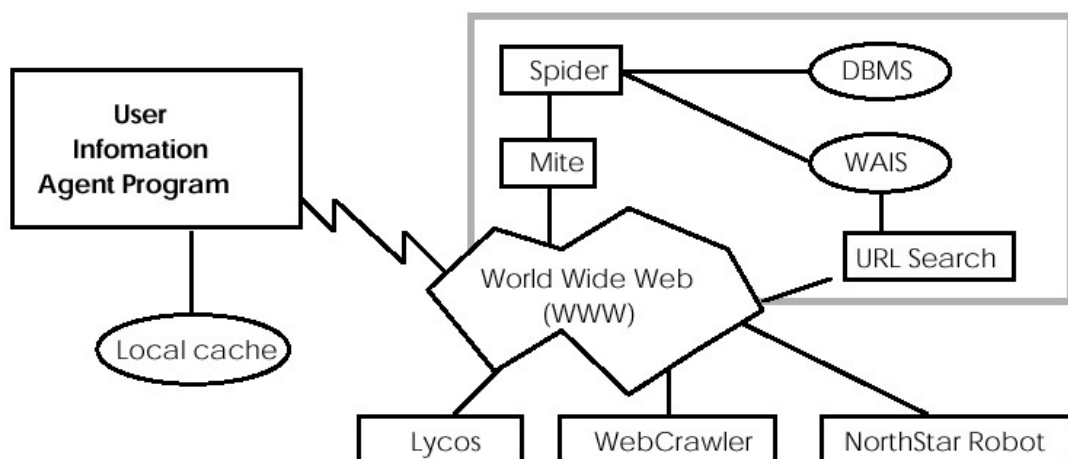
## 8. Πληροφοριακοί Πράκτορες – (Internet ή Information Agents)

Ένας πληροφοριακός πράκτορας έχει πρόσβαση σε κάποιες πληροφοριακές πηγές (βάσεις δεδομένων, άλλους πληροφοριακούς πράκτορες-πχ. Meta-search engines) και είναι ικανός να αντιπαραβάλλει και να διαχειρίζεται αυτές τις πληροφορίες, με σκοπό να δίνει απαντήσεις σε ερωτήματα και σε αιτήματα χρηστών ή άλλων πρακτόρων.

Οι πληροφοριακοί πράκτορες κινούνται πολλές φορές σε πηγές που εδράζονται σε δίκτυα υπολογιστών. Δύο σημαντικές προσπάθειες τέτοιων πρακτόρων είναι το IRA (Information Retrieval Agent – Levy, Gruber, 1991-1994) και το CARNOT (Huhns, 1992).

Οι πληροφοριακοί πράκτορες μπορούν να είναι στατικοί ή κινητοί, κοινωνικοί ή μη, και να μαθαίνουν ή όχι. Όταν ένας πράκτορας είναι στατικός έχει τη παρακάτω αρχιτεκτονική. Αυτός λειτουργεί εν μέσω ενός φυλλομετρητή και χρησιμοποιεί τα spiders και τις μηχανές αναζήτησης, με σκοπό να συλλέξει πληροφορίες. Να βρει δηλαδή, τα URL στα οποία υπάρχει πληροφορία που ενδιαφέρει το αίτημα του χρήστη.

**Σχήμα 23: Λειτουργία πληροφοριακών πρακτόρων**



Πηγή: Nwana, 1996

### 8.1. Πράκτορες παγκόσμιου διαδικτύου πληροφοριών

Ο μεγάλος όγκος πληροφοριών που βρίσκεται στο Internet, αντιμετωπίζεται

σήμερα με εφαρμογές πρακτόρων, που διευκολύνουν το χρήστη στην ανεύρεση πληροφορίας. Αυτοί οι πράκτορες φιλτράρουν την εισερχόμενη μέσω newsgroups ή mailling lists πληροφορία ή αναζητούν στο διαδίκτυο πληροφορίες που αφορούν το χρήστη.

Κύριο χαρακτηριστικό των διαδικτυακών πρακτόρων, είναι ότι μαθαίνουν τις προτιμήσεις και τα ενδιαφέροντα του κάθε χρήστη, είτε:

- ❖ χρησιμοποιώντας μεθόδους μηχανικής μάθησης
- ❖ παρατηρώντας τις επιλογές του (MAXIMS)
- ❖ μέσω παραδειγμάτων εκπαίδευσης του πράκτορα (NEWT)

Οι Entzioni & Weld το 1994, εισήγαγαν την έννοια του softbot, το οποίο περιγράφει τον απόλυτο πληροφοριακό πράκτορα. Σε αυτό οι χρήστες εισάγουν υψηλού επιπέδου αιτήματα (σε φυσική γλώσσα) και περιμένουν αποτελέσματα από τους δικτυακούς τόπους του παγκόσμιου διαδικτύου. Το softbot περιγράφεται σαν ένα ρομπότ, του οποίου οι μηχανισμοί δράσης ήταν εντολές διαχείρισης mv ή compress (UNIX) και συνδέονταν με κάποιες υπηρεσίες του www όπως ftp, telnet, mail κ.α. Έτσι, οι αισθητήρες του περιείχαν υπηρεσίες όπως τα archie, gopher, netfind και εντολές του unix όπως οι προαναφερθείσες κ.α.

Το softbot:

- ❖ παρέχει ένα ολοκληρωμένο και εκφραστικό περιβάλλον διεπαφής με το χρήστη, για την έκφραση των ερωτημάτων
- ❖ επιλέγει δυναμικά ποιες υπηρεσίες θα ενεργοποιήσει και σε τι βαθμό, και τον τρόπο που θα βρει, φιλτράρει και παρουσιάσει την πληροφορία
- ❖ μπορεί να βρει διαφορετικές οδούς πρόσβασης στην πληροφορία, όταν σε ένα μονοπάτι πάει κάτι λάθος

### **8.1.1. Μηχανές αναζήτησης (search engines)**

Η **μηχανή αναζήτησης** είναι μια υπηρεσία που εξυπηρετεί την εύρεση πληροφοριών, βάσει κάποιων κριτηρίων. Κάθε μηχανή αναζήτησης περιέχει μια βάση δεδομένων, με στοιχεία για τα περιεχόμενα κάποιων κόμβων στον Παγκόσμιο Ιστό (Internet ), πχ:

- ❖ τα URL (Universal Resource Locator) των σελίδων που περιέχονται σε έναν κόμβο
- ❖ τους τίτλους των σελίδων που περιέχονται σε έναν κόμβο

- ❖ την επικεφαλίδα της κάθε ιστοσελίδας
- ❖ μικρές περιγραφές των περιεχομένων του κόμβου
- ❖ αντιπροσωπευτικές λέξεις-κλειδιά για κάθε σελίδα
- ❖ τις πρώτες 100 με 200 λέξεις της κάθε σελίδας

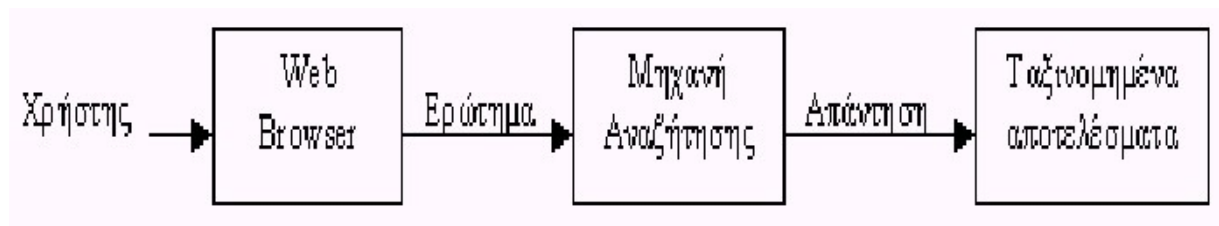
Οι ισχυρές μηχανές αναζήτησης διατηρούν στη βάση δεδομένων τους όλο το περιεχόμενο των σελίδων ενός κόμβου (χωρίς βέβαια τις εικόνες). Η βάση αυτή ενημερώνεται και εμπλουτίζεται καθημερινά, με τη συνδρομή των διαχειριστών των εγγεγραμμένων κόμβων και με τη χρήση των λεγόμενων **web robots**, τα οποία θα μελετηθούν αργότερα.

Ο χρήστης για να βρει την πληροφορία που τον ενδιαφέρει πρέπει:

- ❖ να συνδεθεί με μια μηχανή αναζήτησης, μέσω του φυλλομετρητή του (Web browser)
- ❖ να στείλει μια ή και περισσότερες λέξεις-κλειδιά, που περιγράφουν στη μηχανή αναζήτησης, την πληροφορία που ψάχνει να βρει

Έπειτα, η μηχανή αναζητεί τις συγκεκριμένες λέξεις-κλειδιά στη βάση δεδομένων της και επιστρέφει στο χρήστη τα αποτελέσματα (τα URLs των διευθύνσεων που περιέχουν τις λέξεις-κλειδιά).

**Σχήμα 24: Τρόπος διεπαφής χρήστη και μηχανής αναζήτησης**

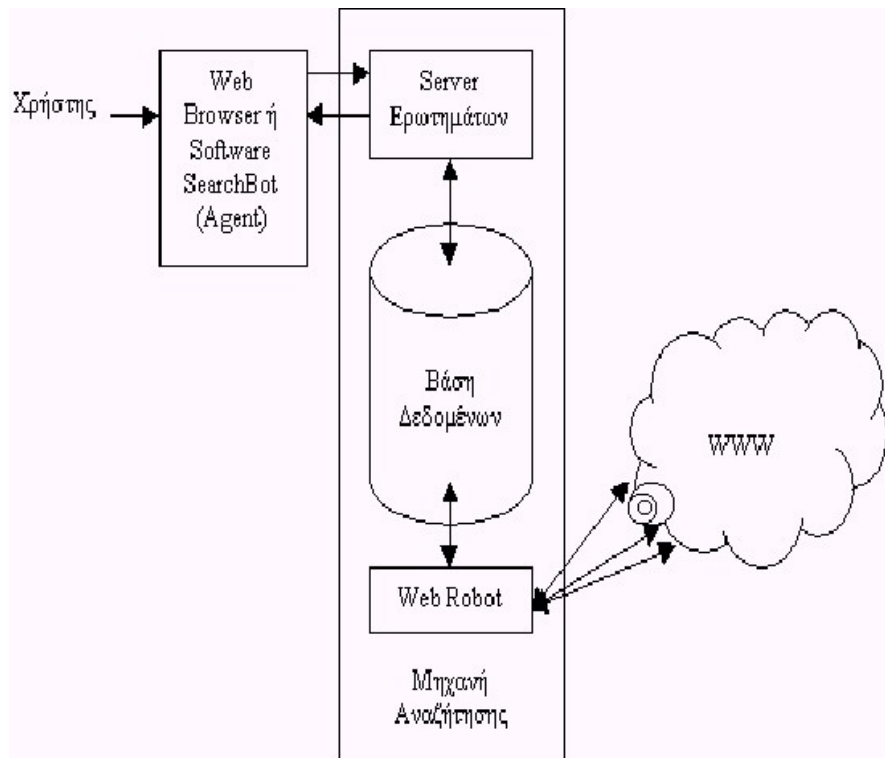


Πηγή: Ίδια Επεξεργασία

Μια μηχανή αναζήτησης αποτελείται από:

- ❖ τον server ερωτημάτων, που είναι υπεύθυνος για την αλληλεπίδραση με το χρήστη. Δέχεται από αυτόν τις λέξεις-κλειδιά με τη μορφή κάποιου ερωτήματος (query) και στέλνει πίσω τα αποτελέσματα
- ❖ τη βάση δεδομένων, που βρίσκονται αποθηκευμένες όλες οι πληροφορίες που ζητάει ο χρήστης, για το που βρίσκονται οι html σελίδες που τον ενδιαφέρουν
- ❖ κάποιο Web Robot, που είναι υπεύθυνο για την ενημέρωση της βάσης δεδομένων και την αλληλεπίδραση με το WWW

Σχήμα 25: Λειτουργία μηχανής αναζήτησης



Πηγή: Ϊδια Επεξεργασία

Γενικά, η ισχύς και η επιτυχία μιας μηχανής αναζήτησης κρίνεται από την ακρίβεια (precision) και την ανάκληση (recall).

**Ακρίβεια:** το ποσοστό των σελίδων που ενδιέφεραν όντος τον χρήστη προς το σύνολο των σελίδων που επέστρεψε η μηχανή αναζήτησης.

**Ανάκληση:** το ποσοστό των σελίδων που επέστρεψε η μηχανή αναζήτησης προς το ποσοστό όλων των σχετικών σελίδων στον Ιστό. Η ανάκληση συνδέεται άμεσα με την πληρότητά της (coverage). Δηλαδή, με το ποσοστό των εγγεγραμμένων html σελίδων στη βάση δεδομένων της μηχανής αναζήτησης, προς το ποσοστό όλων των html σελίδων που βρίσκονται στον Ιστό

Οι μηχανές αναζήτησης παρουσιάζουν τα αποτελέσματά τους (URLs σχετικών html σελίδων) κατά δεκάδες, ταξινομημένα βάσει κάποιου αλγορίθμου. Σύμφωνα με στατιστικές μελέτες: «αν ένα Web site δεν περιέχεται στις πρώτες τρεις σελίδες αποτελεσμάτων, δηλαδή στα πρώτα 30 προτεινόμενα URLs, υπάρχει μόνο 28% πιθανότητα να επιλεγεί από έναν χρήστη. Αν δεν περιέχεται στα πρώτα 70 προτεινόμενα URLs, τότε το ποσοστό αυτό πέφτει στο 12%».

Οι μηχανές αναζήτησης αποτελούν ένα από τα πιο διαδεδομένα μέσα αναζήτησης πληροφοριών στο Internet μαζί με τους καταλόγους.

Έτσι αποτελούν ισχυρό μέσο, το οποίο κρίνει την επιτυχία ή αποτυχία ενός website. Συνεπώς, γίνεται κατανοητή η ανάγκη ενός site, για μια καλύτερη θέση μέσα στα αποτελέσματα μιας μηχανής αναζήτησης.

Η θέση αυτή ονομάζεται ranking. Σχετικά με το ranking αξίζουν να αναφερθούν τα εξής:

Η WebCrawler, χρησιμοποιεί το link popularity check για να προωθήσει κάποιες σελίδες. Ελέγχει πόσες φορές παρουσιάζονται σε άλλες html σελίδες, links τα οποία στέλνουν στη σελίδα που πρέπει να βαθμολογήσει. Όσο πιο πολλά links βρει, τόσο πιο αξιοπρόσεκτη και αποδεκτή θεωρεί πως είναι η σελίδα που μελετάει, από την κοινότητα του Internet. Έτσι, αυξάνει το ranking της.

Η Hotbot, ελέγχει επίσης, το μέγεθος της html σελίδας που μελετάει. Οι μικρές html σελίδες που έχουν τον ίδιο αριθμό λέξεων-κλειδιών με κάποιες μεγαλύτερες θα έχουν καλύτερο ranking.

Η Infoseek Ultra, χρησιμοποιεί βάρη σε μερικές λέξεις-κλειδιά. Οι ασυνήθιστες λέξεις, που παρουσιάζονται σπάνια στη βάση δεδομένων της Infoseek, έχουν μεγαλύτερο βάρος από τις συνηθισμένες. Η λογική αυτή βοηθάει τη μηχανή αναζήτησης στην επίτευξη καλύτερου αποτελέσματος, όταν σαν κριτήριο αναζήτησης δίνονται περισσότερες από μία λέξεις-κλειδιά.

### 8.1.2. Web Robots

Μια μηχανή αναζήτησης βασίζει τη λειτουργία της και αυτά που επιτελεί στα Web Robots (ή αλλιώς Spiders, Crawlers), τα οποία είναι πράκτορες. Τα Web Robots «ζουν και εργάζονται ασταμάτητα» στο WWW από το 1994. Σε αυτό το χρονικό διάστημα, προσφέρουν υπηρεσίες χρήσιμες για την εύρεση και την ταξινόμησή πληροφοριών βάσει ερωτημάτων, και είναι χρήσιμα για στατιστικά κάθε είδους στο Internet.

Ένα **Spider** ή **Crawler**, είναι πρόγραμμα που επισκέπτεται Websites. Σε αυτά διαβάζει τις σελίδες τους και άλλες πληροφορίες και δημιουργεί ή ενημερώνει εγγραφές στον κατάλογο κάποιας μηχανής αναζήτησης.

Για να «συρθεί» ένα Spider μέσα σε ένα website υπάρχουν πολλοί τρόποι. Για παράδειγμα, θα μπορούσε να ακολουθήσει τους συνδέσμους του hypertext της κεντρικής σελίδας και να συνεχίσει με τις επόμενες, μέχρι να διαβάσει όλες τις

διαθέσιμες σελίδες που περιέχει το website ή μέχρι να εκπληρωθεί ένας στόχος. Τα προγράμματα αυτά μπορούν να δραστηριοποιηθούν σε ολόκληρο το WWW, αλλά συνήθως κάνουν επιλεκτικές επισκέψεις σε κόμβους, οι οποίοι έχουν δηλωθεί ως «νέοι» ή «προσφάτως ανακαινισμένοι» από τους διαχειριστές τους.

Κάθε μηχανή αναζήτησης στο WWW έχει δικά της spiders. Αυτά είναι υπεύθυνα για την ενημέρωση της βάσης δεδομένων της μηχανής αναζήτησης και την αλληλεπίδραση με το WWW.

Τα γνωστότερα Web Robots που συναντάμε στο Internet ανήκουν στις γνωστότερες μηχανές αναζήτησης.

Έτσι, για τη μηχανή αναζήτησης υπάρχει το web robot

**Alta Vista**

**Excite**

**HotBot**

**InfoSeek**

**Lycos**

**Northern Light**

**Web Crawler**

**Scooter**

**Crawlx.atext**

**Slurp the Web**

**Sidewinder**

**T-Rex**

**Gulliver**

**Spidey**

Ολοκληρωμένη λίστα με Web robots διατηρείται στην ιστοσελίδα:

<http://info.webcrawler.com/mak/projects/robots/active/html/type.html>

Τα web robots έχουν υψηλές απαιτήσεις σε ταχύτητα και εύρος δικτύου (bandwidth). Η χρήση των Web robots ή των Spiders και η συνεχόμενη σφυρηλάτηση ενός Web Server με Requests, θα μπορούσε να μειώσει αισθητά την απόδοσή του και να τον οδηγήσει ακόμα και σε κατάρρευση.

Όπως ξέρουμε, η ικανότητα ενός Server να εξυπηρετεί εξαρτάται από:

- ❖ το εύρος του δικτύου (bandwidth)
- ❖ την ταχύτητα του
- ❖ την ποσότητα των δεδομένων που μεταφέρονται σε μια χρονική στιγμή, η οποία εξαρτάται άμεσα από τον αριθμό των απαιτήσεων (Requests) των χρηστών

Έτσι, η λογική των Web robots αποτελεί αμφιλεγόμενο θέμα ανάμεσα στους Web Administrators, κυρίως επειδή ανεύθυνοι προγραμματιστές έχουν δημιουργήσει τέτοια που τρέχουν πολύ γρήγορα, κάνοντας πολλαπλά Requests σε πολύ μικρό χρονικό διάστημα (αυτό ονομάζεται rapid fire).

Αν αναλογιστούμε και την αυξανόμενη χρήση των φορμών και των CGI scripts στις σελίδες του Internet (τα οποία ένας Spider δεν μπορεί να επεξεργαστεί), βλέπουμε την ανάγκη δημιουργίας ενός προτύπου, το οποίο θα βοηθάει τον Web Administrator να ελέγχει την είσοδο των διαφόρων Spiders στον server του.

### 8.1.3. Delivery Agents

Άλλο είδος πληροφοριακών πρακτόρων, είναι οι **delivery agents** που αντιμετωπίζουν το πρόβλημα της ταχύτητας μετάδοσης δεδομένων προς το χρήστη. Το μεγαλύτερο μέρος των χρηστών συνδέεται με το Internet μέσω μιας Dial-up σύνδεσης, με ταχύτητες που φτάνουν δύσκολα τα 56 Kbps. Έτσι, η μετάδοση της πληροφορίας μπορεί να γίνει πολύ χρονοβόρα.

#### Background Ftp

Μια από της υπηρεσίες που προσφέρει η Forthnet, είναι η υπηρεσία Background FTP. Ο χρήστης που είναι συνδρομητής στη Forthnet, ορίζει την ηλεκτρονική διεύθυνση στην οποία βρίσκεται ένα αρχείο που τον ενδιαφέρει. Το **background FTP** αναλαμβάνει τότε να κατεβάσει (download) αυτό το αρχείο στον server της Forthnet, από τον οποίο μπορεί τελικά ο χρήστης να το κατεβάσει στον Η/Υ του. Το κέρδος για τον χρήστη είναι ότι, η ταχύτητα μετάδοσης του αρχείου από τον server της Forthnet στον υπολογιστή του, είναι ενδεχομένως πολύ μεγαλύτερη, από αυτή που θα είχε επιτευχθεί με μια απ' ευθείας σύνδεση με τον server, στον οποίο ανήκε αρχικά το αρχείο.

#### Mirroring Programs

Πολλές φορές συναντάμε στο Internet τον όρο Site Mirror (mirror = καθρέφτης). Ένα mirroring program είναι πρόγραμμα υπεύθυνο για την αντιγραφή πολλών αρχείων από έναν server του Internet σε έναν άλλο. Επίσης, μπορεί να είναι πρόγραμμα υπεύθυνο για την αντιγραφή των περιεχομένων ενός ολόκληρου server σε έναν άλλον. Αυτό τρέχει μια φορά την ημέρα και έχει ρουτίνες παρόμοιες με αυτές των Web Robots ή Spiders, για να καταλαβαίνει τη δομή του site και να αντιγράφει όλες τις πληροφορίες που περιέχονται σ' αυτό.

### 8.1.4. Internet notification agents

Οι **Notification Agents** είναι προγράμματα που ειδοποιούν το χρήστη (ή άλλους Agents), όταν συμβαίνουν περιστατικά τα οποία τον ενδιαφέρουν. Στηρίζουν συνεπώς τη λειτουργία τους στην ανάγκη ενός πράκτορα για αυξημένη



προσαρμοστικότητα στις αλλαγές του περιβάλλοντός του. Οι πράκτορες αυτοί λειτουργούν σε συνεργασία με άλλους ή απευθείας με το χρήστη για να βοηθήσουν στην επίτευξη ενός συλλογικού στόχου. Στόχος μπορεί να είναι οποιαδήποτε υπολογιστική λειτουργία πχ:

- ❖ η δημιουργία ενός απλού backup, όταν το περιστατικό που συμβαίνει είναι ένας απλός χτύπος του ρολογιού του Η/Υ
- ❖ η on-line αγοροπωλησία μετοχών, όταν η τιμή τους φτάσει σε ένα συγκεκριμένο κατώτατο ή ανώτατο επίπεδο

### WWW Monitoring Programs

Τα συγκεκριμένα προγράμματα είναι υπεύθυνα για την προσαρμογή χρηστών ή άλλων Agents σε περιστατικά τα οποία αφορούν την κοινότητα του Internet. Ως περιστατικό, μπορεί να είναι η αλλαγή των πληροφοριών μιας html σελίδας η οποία ενδιαφέρει το χρήστη.

Μερικές σελίδες στο WWW δίνουν τη δυνατότητα στο χρήστη να επιλέξει εάν θέλει να ειδοποιηθεί με e-mail σε περίπτωση που τα περιεχόμενα τους αλλάξουν. Ο χρήστης δίνει το e-mail του και μπαίνει σε μια λίστα με άλλους ενδιαφερόμενους χρήστες. Αν αλλάξουν τα περιεχόμενα της σελίδας αυτής, ο υπεύθυνος (webmaster) θα τρέξει έναν Notification Agent για να στείλει με email ένα τυποποιημένο μήνυμα μαζικά στα μέλη της λίστας. Το μήνυμα θα έχει την εξής περίπου μορφή:

«Σας ενημερώνουμε ότι τα περιεχόμενα της σελίδας [www.site.com/page.html](http://www.site.com/page.html) έχουν αλλάξει».

Στην περίπτωση που ο webmaster μιας σελίδας δεν έχει προβλέψει για την ενημέρωση των χρηστών που την παρακολουθούν, υπάρχουν προγράμματα τα οποία κάνουν ακριβώς αυτή τη δουλειά για οποιαδήποτε html σελίδα. Αυτά επισκέπτονται συγκεκριμένες html σελίδες στο WWW και ελέγχουν αν έχουν αλλάξει τα περιεχόμενά τους.

Τα προγράμματα αυτά μπορούν να τρέχουν τοπικά, από το Desktop ενός απλού χρήστη. Επειδή, όμως κάτι τέτοιο θα σπαταλούσε τον πολύτιμο on-line χρόνο του χρήστη σε συνεχείς ελέγχους html σελίδων, έχουν δημιουργηθεί sites που αναλαμβάνουν τη δουλειά αυτή εκ μέρους του. Η χρησιμότητα των υπηρεσιών αυτών διαφέρει από χρήστη σε χρήστη. Τέλος, υπάρχουν Monitoring Agents που βοηθούν στην καταγραφή (NETMind).

### Link Validation Programs

Είναι παρόμοια με τα WWW Monitoring Programs. Αποτελούν εφαρμογές, που ελέγχουν μια ή και περισσότερες html σελίδες για «σπασμένα» links. Μπορούν να διατρέξουν ένα ολόκληρο site και να ενημερώσουν για το ποια links δε λειτουργούν. Δηλαδή, ποιες από τις διευθύνσεις που υπάρχουν στο site δεν στέλνουν σε έγκυρες html σελίδες.

Τα Link Validation Programs χρησιμοποιούνται κυρίως από Web Administrators, για τον έλεγχο των html σελίδων τους και από μηχανές αναζήτησης, για να ενημερώνουν κατάλληλα τη βάση δεδομένων τους (σε περίπτωση που ένα site πάψει να υπάρχει).

#### **8.1.5. Off – Line Browsers**

Οι **off-line browsers** μοιάζουν με τα Mirroring Programs, με τη διαφορά ότι έχουν ως στόχο την αντιγραφή ολόκληρων κόμβων (ή ενός μέρους τους) τοπικά, στο σκληρό δίσκο του χρήστη και όχι σε άλλον κόμβο του Internet. Έτσι, ο χρήστης μπορεί off-line να επισκεφθεί αργότερα έναν κόμβο που τον ενδιαφέρει (χωρίς να είναι συνδεδεμένος).

Οι ρουτίνες που χρησιμοποιεί ένας Off-Line Browser είναι παρόμοιες με αυτές ενός Mirroring Program ή ενός Spider. Βασίζονται δηλαδή στο φιλτράρισμα των html σελίδων. Η διαδικασία έχει ως εξής:

1. ο χρήστης δίνει συνήθως την κεντρική σελίδα ενός Website
2. το πρόγραμμα τη διαβάσει, την αντιγράφει στο σκληρό δίσκο και βρίσκει τα links τα οποία υπάρχουν στη σελίδα αυτή
3. μετά διαβάσει ένα-ένα τα links, τα αντιγράφει και βρίσκει τα δικά τους links. Η λειτουργία αυτή επαναλαμβάνεται μέχρι να εξαντληθούν όλες οι σελίδες του Website ή να εκπληρωθεί ένας άλλος στόχος του προγράμματος.

Ένας off-line browser μπορεί:

- ❖ να αντιγράψει τα περιεχόμενα ενός ολόκληρου Website στο σκληρό δίσκο του χρήστη
- ❖ να αντιγράψει τα περιεχόμενα ενός ολόκληρου Website σε έναν άλλο κόμβο στο Internet (Mirroring)
- ❖ να ψάξει όλες τις διευθύνσεις email που αναφέρονται σε ένα ολόκληρο site
- ❖ να δημιουργήσει σχηματικά τη δομή ενός Website
- ❖ να ψάξει ένα Website για συγκεκριμένα αρχεία και να τα αντιγράψει τοπικά

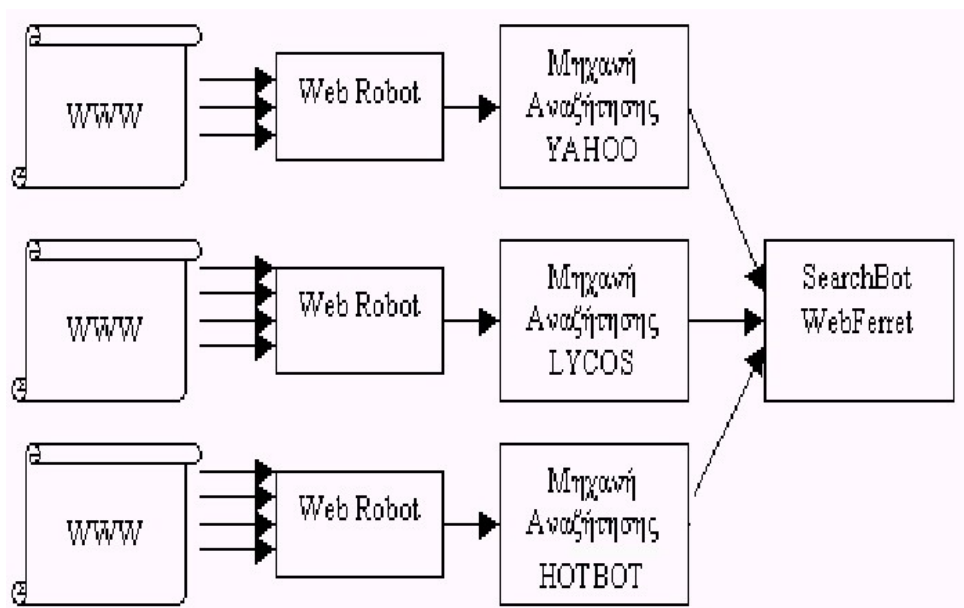
### 8.1.6. Searchbots

Τα **searchbots** είναι πληροφοριακοί πράκτορες που αναλαμβάνουν την αυτοματοποίηση της επικοινωνίας του Web Browser με τη μηχανή αναζήτησης. Ένα searchbot είναι ένα πιο σύνθετο webrobot. Δηλαδή, το SearchBot μπορεί να ρωτάει πολλές μηχανές αναζήτησης ταυτόχρονα, ενώ το webrobot σχετίζεται μόνο με μια.

Τα SearchBots είναι μικρά προγράμματα:

- ❖ που εγκαθίστανται και τρέχουν τοπικά, στο σκληρό δίσκο του χρήστη
- ❖ έχουν δικό τους user interface και μπορούν να λειτουργήσουν αυτόνομα ή σε συνεργασία με άλλες εφαρμογές
- ❖ μπορούν να συνεργαστούν με τον default Web Browser
- ❖ κάνουν παράλληλες αναζητήσεις (requests) σε διάφορες μηχανές αναζήτησης και εμφανίζουν τα αποτελέσματα συγκεντρωμένα και ταξινομημένα.

**Σχήμα 26: Λειτουργία του Searchbot**



Πηγή: Ίδια Επεξεργασία

Ένας χρήστης μπορεί πιο εύκολα να διατυπώσει το ερώτημά του σε ένα SearchBot από το να ανοίξει τον Web Browser του, να συνδεθεί με τη σελίδα της κάθε μηχανής αναζήτησης και να προσπαθήσει να αναζητήσει από εκεί τα αποτελέσματα της.

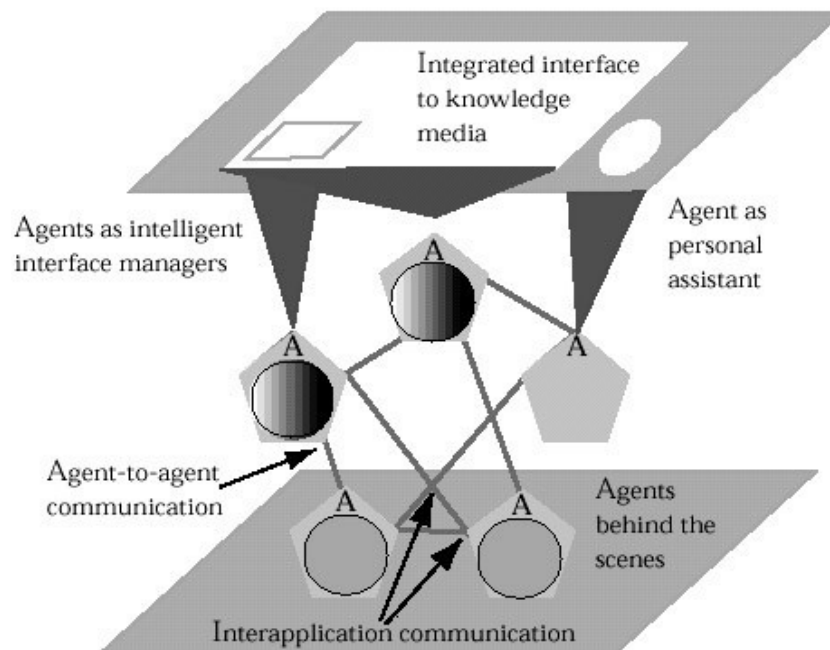
## 9. Ετερογενή Συστήματα Πρακτόρων (Heterogeneous Agent Systems)

Τα ετερογενή συστήματα πρακτόρων αποτελούνται από περισσότερους του ενός τύπου πράκτορες, που κινούνται στο ίδιο περιβάλλον. Η δημιουργία τέτοιου είδους πρακτόρων ήρθε στο μυαλό των Genesereth & Ketchpel το 1994, μιας και αυτοί ήθελαν να περιγράψουν με αυτούς την ποικιλομορφία των συμπεριφορών στην πραγματική ζωή.

Τέτοιοι πράκτορες μπορούν να προσομοιάσουν τις λειτουργίες και διαδικασίες συστημάτων της πραγματικότητας. Η παρουσία διαφορετικών ειδών και τύπων πρακτόρων επιτυγχάνει το παραπάνω. Έτσι, σε ένα τέτοιο σύστημα διαφορετικά είδη προγραμμάτων τα οποία παρέχουν διαφορετικές υπηρεσίες και εξυπηρετούν διαφορετικά πεδία ενεργειών ως μεμονωμένες οντότητες, διασυνδέονται και αλληλεπιδρούν αποτελώντας μια ολοκληρωμένη εικόνα ενός κόσμου από διαφοροποιήσεις.

Αυτά τα συστήματα που είναι βασισμένα στους πράκτορες αποδίδονται ως agent-based systems (ABS), agent-based simulations, agent-based software engineering. Το βασικό χαρακτηριστικό τους είναι η ύπαρξη μιας γλώσσας επικοινωνίας μεταξύ των διαφορετικών ειδών πρακτόρων.

**Σχήμα 27: Σύνδεση και επικοινωνία πρακτόρων σε ένα ετερογενές σύστημα**



Πηγή: Jeffrey M. Bradshaw

Τα οφέλη από τη χρήση τέτοιων συστημάτων είναι:

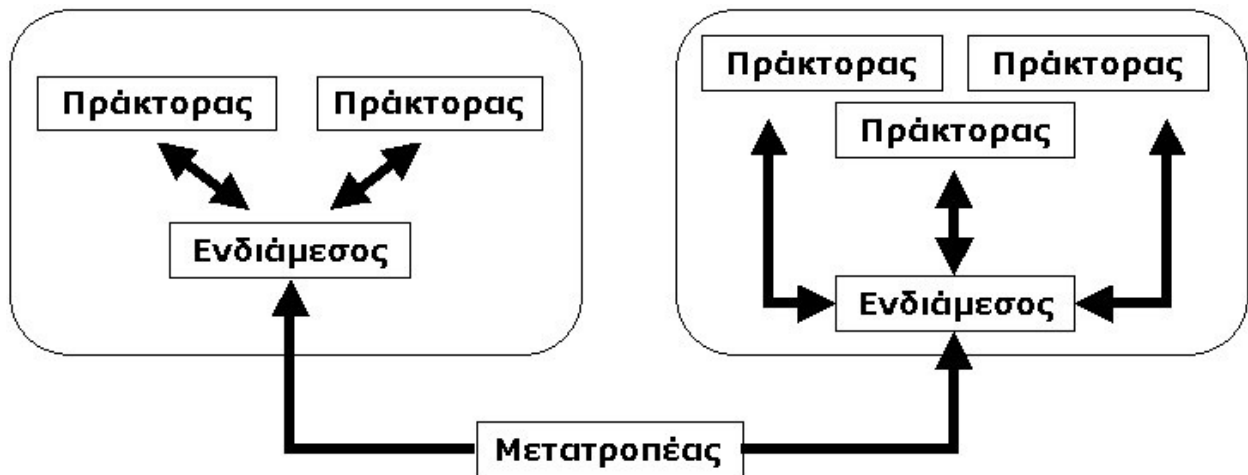
- ❖ η δημιουργία υπηρεσιών προστιθέμενης αξίας από πολυπρακτορικές εφαρμογές
- ❖ η μείωση του χρόνου, του κόπου και του κόστους κατά την υλοποίηση νέων εφαρμογών, όταν στις πιο παλιές εκδόσεις αυτών είχαν βρεθεί λάθη
- ❖ οι νέες προοπτικές στον προγραμματισμό και στη διασυνδεσιμότητα των εφαρμογών

Κατά το σκεπτικό των Genesereth & Ketchpel, ο πράκτορας σε ένα ετερογενές σύστημα πρακτόρων είναι όπως ένα αντικείμενο στον αντικειμενοστραφή προγραμματισμό. Αυτό παρέχει ένα περιβάλλον διεπαφής με μηνύματα μεταξύ του χρήστη, των εσωτερικών δομών δεδομένων που βασίζεται και των αλγορίθμων του. Η διαφορά που υπάρχει έγκειται στο γεγονός ότι στα ABS υπάρχει κοινή γλώσσα επικοινωνίας των πρακτόρων, ενώ στον αντικειμενοστραφή προγραμματισμό, το περιεχόμενο ενός μηνύματος το αντιλαμβάνεται διαφορετικά το κάθε αντικείμενο (πολυμορφισμός).

Η αρχιτεκτονική τέτοιων συστημάτων αποτελείται από:

- ❖ τους *κάθε τύπου πράκτορες*
- ❖ το *μετατροπέα*, που βοηθά τους πράκτορες να λαμβάνουν τα διάφορα μηνύματα στη γλώσσα που έχουν δημιουργηθεί. Επίσης, αυτός μεταφέρει τις ενέργειές τους προς τα έξω. Ο μετατροπέας εξυπηρετεί την ύπαρξη της γλώσσας επικοινωνίας μεταξύ των πρακτόρων, μεταφράζοντας αυτά που υπάρχουν στον εξωτερικό κόσμο, για να γίνουν κατανοητά σε αυτούς και το αντίστροφο. Έτσι, όλοι μπορούν να καταλάβουν όλους.
- ❖ τον *ενδιάμεσο*, ο οποίος καθορίζει τον προσανατολισμό και τη συνεργασία ομάδων πρακτόρων με άλλους πράκτορες. Ο ενδιάμεσος στερεί την αυτονομία των μεμονομένων πρακτόρων και αποφασίζει για αυτούς σχετικά με τη θέση που θα βρίσκονται, τις υπηρεσίες που θα δέχονται, τη σύνδεση τους με άλλα περιβάλλοντα και τη συνομιλία τους με άλλους πράκτορες.

Σχήμα 28: Ετερογενή συστήματα κατά Genesereth & Ketchpel

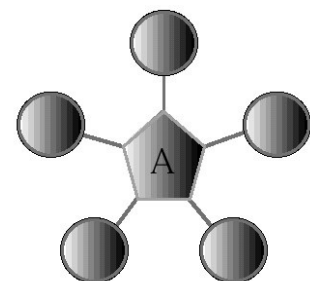


Πηγή: Ιδία Επεξεργασία

Παραδείγματα ετερογενών συστημάτων πρακτόρων είναι:

- το **ARCHON** (Wittig, 1992)
- το **ICIS** (Intelligent and Cooperative Information Systems, 1992), που ενοποιεί διάφορα πληροφοριακά συστήματα, αρχιτεκτονικές λογισμικού, βάσεις δεδομένων και εφαρμογές τεχνητής νοημοσύνης. Σε αυτό κάθε πράκτορας είναι προσκολλημένος σε κάποια από τις προηγούμενες οντότητες και συμπεριφέρεται εξυπηρετώντας τα ζητούμενα της οντότητας που συνδέεται.
- το **PACT** (Palo Alto Collaborative Testbed - Cutkosky το 1993), που περιγράφει την πραγματική ζωή. Στην ουσία εμπεριέχει τεχνολογικά και κοινωνιολογικά θέματα που σχετίζονται με την υλοποίηση μεγάλης κλίμακας καταμεμημένων παράλληλων συστημάτων, για το προηγούμενο σκοπό. Περιελάμβανε 31 ABS και 15 Η/Υ. Οι πράκτορες οργανώθηκαν υπό την καθοδήγηση διαφόρων μεσαζόντων. Το PACT αποτέλεσε τμήμα του DARPA.

Τέλος, αξίζει να αναφερθεί είναι ότι στα ετερογενή συστήματα κάθε πράκτορας μπορεί τελικά να αποτελείται από περισσότερους πράκτορες.



### 9.1. Πολυπρακτορικά Συστήματα

Ένα ετερογενές σύστημα πρακτόρων όταν αποτελείται από πολλούς πράκτορες είναι **πολυπρακτορικό σύστημα** (*multi-agent system*). Κάθε πολυπρακτορικό

σύστημα σχεδιάστηκε και υλοποιήθηκε ως σύνολο πρακτόρων που συνεργάζονται, συνεννοούνται και διαπραγματεύονται.

Τα πολυπρακτορικά συστήματα αποτελούν δίκτυο από πράκτορες που αλληλεπιδρούν για να επιλύσουν προβλήματα, τα οποία είναι πέρα της ικανότητας και της γνώσης ενός μόνο πράκτορα. Σε αυτά δεν υπάρχει κεντρικός σχεδιασμός του περιβάλλοντος και παρέχεται συγκεκριμένη γλώσσα επικοινωνίας.

Οι πράκτορες σε ένα πολυπρακτορικό σύστημα:

εργάζονται αυτόνομα ανταλλάσσοντας πληροφορίες προσπαθώντας να επιτύχουν τους δικούς τους ανεξάρτητους στόχους

ή

συνεργάζονται επιλύοντας υποπροβλήματα έτσι ώστε ο συνδυασμός των επιμέρους λύσεων που θα προκύψουν να αποτελέσει την τελική λύση.

Ένα πολυπρακτορικό σύστημα χαρακτηρίζεται από τα παρακάτω γεγονότα:

- ❖ οι πράκτορες *συνεργάζονται / διαπραγματεύονται* (*cooperation/ negotiation*) μέσω κάποιας γλώσσας επικοινωνίας. Φτάνουν έτσι πολύ συχνά σε κοινά αποδεκτές συμφωνίες και επιλύουν ενδεχόμενες συγκρούσεις που προκύπτουν από την επίτευξη των επιμέρους στόχων τους
- ❖ κανένας πράκτορας δεν έχει πλήρη πληροφορία
- ❖ δεν υπάρχει κεντρικός έλεγχος στο σύστημα
- ❖ τα δεδομένα είναι κατανεμημένα
- ❖ οι υπολογισμοί γίνονται με ασύγχρονο τρόπο

Τα πολυπρακτορικά συστήματα συμβάλλουν:

- ❖ στην επίλυση ενός συγκεκριμένου προβλήματος. Οι πράκτορες εκεί είναι συνεργαζόμενοι, μοιράζοντας γνώση για το πρόβλημα.
- ❖ στην επίλυση κατανεμημένων προβλημάτων. Τέτοια μπορεί να είναι:
  - η συλλογή πληροφοριών από κατανεμημένες πηγές (δίκτυα αισθητήρων)
  - η συλλογή πληροφοριών από κατανεμημένες βάσεις πληροφοριών
  - ο έλεγχος εναέριας κυκλοφορίας
- ❖ στην επίλυση προβλημάτων που είναι πολύπλοκα και δύσκολα αντιμετωπίσιμα από ένα μόνο πράκτορα
- ❖ στη διασύνδεση και λειτουργία ήδη υπαρχόντων συστημάτων (legacy systems) για να είναι εύκολη η εκμετάλλευσή τους χωρίς τροποποιήσεις (mediator systems)
- ❖ στη καλύτερη αντιμετώπιση των εργασιών που τους έχουν ανατεθεί

Γεγονός όμως είναι ότι πολλές φορές, η επικοινωνία και συνεργασία μεταξύ των πρακτόρων, λόγω του παραλληλισμού των ενεργειών, παρουσιάζει προβλήματα.

Τέτοια είναι:

- ❖ η σχεδίαση, η υλοποίηση και η διαχείριση των περιορισμένων πόρων ενός τέτοιου συστήματος
- ❖ ο τρόπος διαμοιρασμού των συνιστωσών του προβλήματος και της σύνθεσης της τελικής λύσης
- ❖ τα είδη των πρακτόρων που επικοινωνούν και το επιπλέον κόστος που αυτή η επικοινωνία απαιτεί
- ❖ οι γλώσσες και τα πρωτόκολλα που θα χρησιμοποιηθούν για να μπορούν οι πράκτορες να επικοινωνούν και να αλληλεπιδρούν. Δηλαδή, πως κάθε πράκτορας θα αναπαραστήσει και θα συλλογιστεί τις ενέργειες, τα πλάνα και τη γνώση των άλλων πρακτόρων.
- ❖ ο συμβιβασμός των διαφορετικών απόψεων των διαφορετικών πρακτόρων και η αντιμετώπιση των συγκρουόμενων προθέσεων τους
- ❖ ο χρόνος επικοινωνίας και το είδος της πληροφορίας που ανταλλάσσουν οι πράκτορες

## 9.2. Αρχιτεκτονική και επικοινωνία

Η επικοινωνία μεταξύ των πρακτόρων επιτυγχάνεται από την οργάνωση του συστήματος σε τρία επίπεδα.

### A. Το επίπεδο διασύνδεσης (method of interconnection)

Υπάρχουν δύο μοντέλα διασύνδεσης και συνεργασίας ανάμεσα στους πράκτορες:

**1. τα συστήματα μαυροπίνακα** ❖ **2. τα συστήματα ανταλλαγής μηνυμάτων** (message passing systems), όπου οι πράκτορες ανταλλάσσουν πληροφορία και συνεργάζονται μέσω μηνυμάτων. Τα μηνύματα τα αποστέλλουν ο ένας στον άλλο βάσει συγκεκριμένων γλωσσών υψηλού επιπέδου. Τα συστήματα αυτά επιτρέπουν την υλοποίηση πολύπλοκων μοντέλων συνεργασίας μεταξύ των πρακτόρων

(blackboard systems) όπου υπάρχει κοινός χώρος εργασίας (blackboard) για όλους τους πράκτορες του συστήματος. Μέσω του κοινού χώρου ανταλλάσσουν αποτελέσματα (result sharing) ή μοιράζονται εργασίες (task sharing).

Τονίζεται ότι από τη στιγμή που κάτι αποθηκεύεται στην κοινή αυτή περιοχή είναι προσπελάσιμο από όλους τους

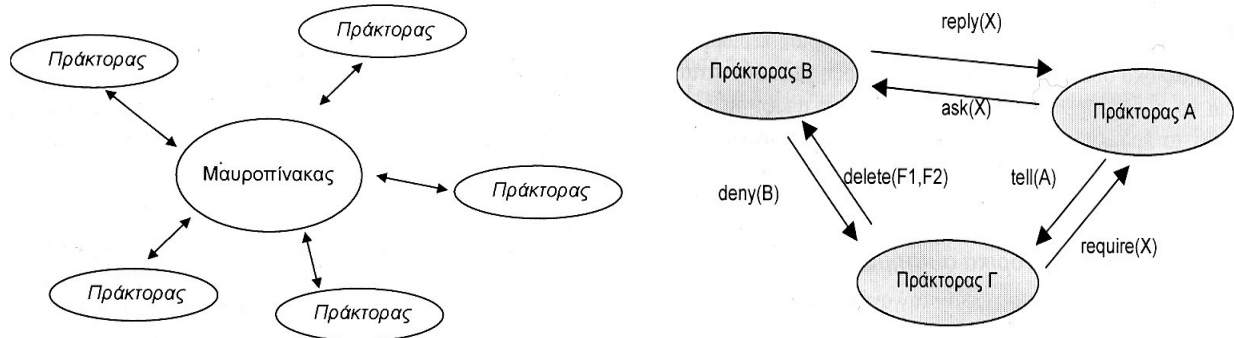
πρακτόρες.



πράκτορες που συμμετέχουν στο σύστημα.

και γενικά προσφέρουν μεγαλύτερη ευελιξία στην ανταλλαγή πληροφοριών από τα συστήματα μαυροπίνακα.

**Σχήμα 29: Σύστημα μαυροπίνακα και Σύστημα ανταλλαγής μηνυμάτων**



Πηγή: Vlachavas I. κ.α., 2002

Κάθε πράκτορας που θέτει μια ερώτηση είτε διακόπτει τη λειτουργία του, μέχρι να πάρει μια απάντηση (σύγχρονος τύπος επικοινωνίας), είτε η απάντηση έρχεται οποιαδήποτε στιγμή μετά την υποβολή της, χωρίς να διακόπτεται η λειτουργία του πράκτορα (ασύγχρονος τύπος επικοινωνίας).

Επιπλέον, αυτός μπορεί να επικοινωνήσει (να ανταλλάξει πληροφορίες) ταυτόχρονα με έναν ή με περισσότερους άλλους πράκτορες. Τέλος, πολλοί πράκτορες μπορούν να επικοινωνούν με άλλους πολλούς. (βαθμός επικοινωνίας 1-1, 1-N, N-N)

B. Το επίπεδο σύνταξης και μορφής των μηνυμάτων

Γ. Το επίπεδο σημασιολογίας

Ο κοινός τρόπος επικοινωνίας στους πράκτορες καθορίζεται από την ύπαρξη δύο πρωτοκόλλων. Αυτά εμπεριέχουν τα δύο προηγούμενα επίπεδα στην οργάνωσή τους.

Έτσι, το πρώτο, το **πρωτόκολλο επικοινωνίας** (*communication protocol*) καθορίζει τη μορφή των μηνυμάτων (μήκος, επιτρεπτά σύμβολα) και τη σημασία τους. Επίσης, είναι υπεύθυνο για να κάνει γνωστή και "κατανοητή" τη σημασία του κάθε μηνύματος, σε όλους τους πράκτορες που συμμετέχουν στο σύστημα. Δηλαδή, καθορίζει τον τύπο των μηνυμάτων για τις:

- προτάσεις - αντιπροτάσεις
- αποδοχές - απορρίψεις
- αποσύρσεις
- διαφωνίες
- ενέργειες

μεταξύ των πρακτόρων.

Το δεύτερο, το **πρωτόκολλο αλληλεπίδρασης** (*interaction protocol*) καθορίζει τον τρόπο συζήτησης (conversations) μεταξύ των πρακτόρων, δηλαδή τις ακολουθίες ανταλλαγής μηνυμάτων. Σε καταστάσεις διαπραγματεύσεων μεταξύ πρακτόρων, αυτό ορίζει το πώς θα απαντήσει κάποιος πράκτορας (με μήνυμα αποδοχής, απόρριψης, διαφωνίας, κλπ) σε κάποιον που του προτείνει μια σειρά ενεργειών μέσω της συζήτησης μεταξύ τους.

## 10. Αληθοφανείς Πράκτορες (Believable Agents)

Ο συγκεκριμένος τύπος πρακτόρων σχετίζεται με τις τεχνολογίες του κινηματογράφου, των βιντεοπαιχνιδιών και της εικονικής πραγματικότητας. Το **Oz** είναι ένα ερευνητικό πρόγραμμα και περιλαμβάνει πράκτορες της συγκεκριμένης κατηγορίας. Αυτοί είναι αρκετά ενδιαφέροντες καλλιτεχνικά, απόλυτα διαδραστικοί με το χρήστη και προσομοιώνουν καταστάσεις του πραγματικού κόσμου. Εκφραστής της ιδέας των αληθοφανών πρακτόρων, ήταν ο Bates το 1992.

Το βασικό χαρακτηριστικό αυτών είναι το συναίσθημα. Δηλαδή, η ενέργεια και η αντίδραση αυτού περιβάλλον με τρόπο ανθρώπινο, σε οποιαδήποτε προσλαμβάνουσα από το εξωτερικό. Έτσι, δε μπορεί ένας ήρωας παιχνιδιού που βασίζεται σε αληθοφανείς πράκτορες να είναι ατρόμητος, ανίκητος κ.τ.λ. Απλά, έχει ανθρώπινη διάσταση.

## **ΚΕΦΑΛΑΙΟ Γ**

### **ΓΛΩΣΣΕΣ – ΠΡΩΤΟΚΟΛΛΑ ΠΡΑΚΤΟΡΩΝ**

i. ΓΛΩΣΣΕΣ ΕΠΙΚΟΙΝΩΝΙΑΣ ΠΡΑΚΤΟΡΩΝ .....	68
ii. ΕΠΙΚΟΙΝΩΝΙΑ ΜΕ ΔΙΑΜΟΙΡΑΣΜΟ ΓΝΩΣΗΣ ..	69
KIF .....	72
ΟΝΤΟΛΟΓΙΕΣ.....	73
KQML .....	74
iii. ΑΛΛΕΣ ΓΛΩΣΣΕΣ (PLACA, ABLE, AGENTO, TRELESCRIPT, κ.α.).....	83
iv. ΠΡΩΤΟΚΟΛΛΑ ΑΛΛΗΛΕΠΙΔΡΑΣΗΣ .....	87

## 1. Εισαγωγή

Ένας φυσικός κατανοεί διαφορετικά την έννοια της ταχύτητας από ότι ένας λάτρης των γρήγορων αυτοκινήτων. Επομένως, όταν οι δύο βρεθούν σε συζήτηση για γρήγορα αυτοκίνητα, πρέπει να βρεθεί ένας κοινός κώδικας επικοινωνίας. Κατά αντιστοιχία σε ένα περιβάλλον που κινούνται πράκτορες, θα πρέπει να υπάρχει μια κοινή γλώσσα επικοινωνίας, ώστε να μπορούν αυτοί να συνεννοούνται, να έρχονται σε επαφή και να λειτουργούν για την επίτευξη των στόχων τους (συνεργαζόμενοι ή μη). Για την επίτευξη της επικοινωνίας μεταξύ των πρακτόρων απαιτούνται:

- μια κοινά κατανοητή και αποδεκτή γλώσσα
- ένας μηχανισμός κατανόησης της γνώσης και των πληροφοριών που ανταλλάσσονται μεταξύ των πρακτόρων
- η ικανότητα ανταλλαγής των παραπάνω μεταξύ των πρακτόρων

Εισερχόμενοι στον όρο της επικοινωνίας μεταξύ των πρακτόρων, θα λέγαμε, ότι αυτοί ενεργούν σε υψηλό επίπεδο για να επικοινωνήσουν. Δηλαδή, χρησιμοποιούν συμβολική απεικόνιση της πραγματικότητας, λειτουργούν με στόχο την απόκτηση γνώσης, την οποία και αποθηκεύουν ως πεποιθήσεις και διατηρούν πάντα μια εσωτερική κατάσταση για οτιδήποτε τους αφορά. Έτσι, το μέσο που θα τους βοηθήσει να πετύχουν την επικοινωνία μεταξύ τους, θα πρέπει να ενσωματώνει τα παραπάνω και να εξυπηρετεί το υψηλό επίπεδο στο οποίο αυτοί επικοινωνούν.

*Με τον όρο γλώσσα επικοινωνίας πρακτόρων εννοούμε ένα σύστημα που δίνει τη δυνατότητα σε κάποιον προγραμματιστή να προγραμματίζει συστήματα υλικού ή λογισμικού κατά τους όρους και συνιστώσες κάποιας θεωρίας πρακτόρων.*

Επιπλέον, η γλώσσα επικοινωνίας μεταξύ των πρακτόρων είναι ένα σύστημα που περιέχει πέρα από τον τρόπο αλληλεπίδρασης του χρήστη με τους πράκτορες και άλλα νοητικά χαρακτηριστικά, που εξυπηρετούν το υψηλό επίπεδο στο οποίο επικοινωνούν οι πράκτορες.

Μια γλώσσα επικοινωνίας χρησιμοποιεί διάφορα πρωτόκολλα που μπορεί να εξυπηρετούν τον πράκτορα ως μηχανισμοί μετακίνησης. Μια γλώσσα επίσης μπορεί να χρησιμοποιείται από πρωτόκολλα (αλληλεπίδρασης) για να εξυπηρετείται η αλληλεπίδραση και συνομιλία μεταξύ των πρακτόρων. Τέλος, μπορεί να χρησιμοποιεί πρωτόκολλα καθορισμού των όρων της συνομιλίας μεταξύ των πρακτόρων. Η ίδια δεν αποτελεί πρωτόκολλο.

Ένα πρωτόκολλο μπορεί να είναι:

- **μετακίνησης**, όπως τα http, smtp, ftp, κ.α
- **αλληλεπίδρασης**, όπως διαπραγμάτευσης, σχεδιασμού κ.α.
- **περιορισμών και καθορισμού των όρων της επικοινωνίας**, όπως η υλοποίηση χρονικής σειράς στα αιτήματα και στις απαντήσεις ή η περάτωση ενός θέματος και η ύστερη έναρξη καινούργιου κτλ.

Μετά τα μέσα της δεκαετίας του '80, άρχισε να υπάρχει μεγάλη αύξηση στον αριθμό των γλωσσών επικοινωνίας μεταξύ των πρακτόρων.

## 2. Γλώσσες επικοινωνίας πρακτόρων και οντότητες αυτών

Μια γλώσσα επικοινωνίας πρακτόρων διαθέτει:

**i. μορφή (form):** αυτό σημαίνει ότι πρέπει να είναι δηλωτική, συνοπτική, απλή στην ανάλυση και στην αναπαραγωγή εννοιών και τέλος κατανοητή από τους προγραμματιστές. Επιπλέον, τα μηνύματα μεταξύ των πρακτόρων θα πρέπει να κωδικοποιούνται σε δυαδική μορφή και να μεταφέρονται από το μηχανισμό μεταφοράς (πρωτόκολλο μεταφοράς). Άλλα χαρακτηριστικά της μορφής της πρέπει να είναι η γραμμικότητα και η επεκτασιμότητα.

**ii. περιεχόμενο (content):** πράγμα που σημαίνει ότι είναι διαχωρισμένη σε επίπεδα για να συνδυάζεται με διάφορα άλλα συστήματα και να αποδίδει στον πράκτορα την ικανότητα διασυνδεσιμότητας - διαλειτουργικότητας. Δηλαδή, πάντα η επικοινωνία ενός πράκτορα θα εξυπηρετείται από το κομμάτι της γλώσσας που θα αναφέρεται στις πράξεις του (επίπεδο I) και το άλλο κομμάτι που θα αναφέρεται στο περιεχόμενο της επικοινωνίας με τους άλλους πράκτορες (επίπεδο II). Τέλος, απαραίτητος είναι ο ξεκάθαρος καθορισμός των όρων των πράξεων της επικοινωνίας και η σύνδεση των δύο επιπέδων της γλώσσας για επέκταση των ικανοτήτων και των ενεργειών του πράκτορα.

**iii. σημασιολογία (semantics):** δηλαδή, να είναι ξεκαθαρισμένη η περιγραφή των εννοιών της γλώσσας (αρχές και πρωτόκολλα) για καλύτερη κατανόησή της από τους εν δυνάμει προγραμματιστές. Η γλώσσα θα πρέπει να βασίζεται σε συγκεκριμένη θεωρία και να είναι σαφής. Να έχει κανονική μορφή (η ομοιότητα στις έννοιες να ισχύει και στην παρουσίαση αυτών στην πραγματικότητα) και να ορίζεται χρονικά και χωρικά. Τέλος, θα πρέπει να δημιουργεί και να παρέχει ένα μοντέλο επικοινωνίας.

**iv. υλοποίηση (implementation):** πράγμα που σημαίνει ότι κάθε τι που αφορά στην μεταφορά των εννοιών από το θεωρητικό επίπεδο στη λογισμική αποτύπωση τους, θα γίνεται με τρόπο ευέλικτο, σε σχέση με την ιδανικότερη ταχύτητα μεταφοράς των εννοιών και τη καλύτερη χρήση του εύρους των διαύλων επικοινωνίας. Πρέπει να υπάρχει ικανοποιητική ταύτιση των εννοιών της γλώσσας με την τεχνολογία λογισμικού. Η υλοποίηση αφορά το περιβάλλον διεπαφής, το οποίο πρέπει να είναι εύχρηστο και απλό και όλες οι εκφράσεις των πράξεων επικοινωνίας μεταξύ των πρακτόρων, να είναι κρυφές και ανεπηρέαστες από το χρήστη. Επιπρόσθετα, θα πρέπει να μπορεί να γίνεται μερική υλοποίηση ενεργειών από κάποιες υπαρκτές συνολικές ομάδες, για να εξυπηρετείται το γεγονός της παρουσίας διαφόρων ειδών πρακτόρων.

**v. δικτύωση (networking):** αυτό σημαίνει ότι θα πρέπει να ταιριάζει και να υιοθετεί τις δυνατότητες της δικτυακής τεχνολογίας. Θα πρέπει επιπλέον να υποστηρίζει όλες τις βασικές συνδέσεις (point-to-point, multicast, broadcast, synchronous, asynchronous κ.α). Επιπλέον, απαιτείται να περιέχει όρους υποστήριξης ενεργειών υψηλών επιπέδων και δημιουργίας πρωτοκόλλων αλληλεπίδρασης.

**vi. περιβάλλον (environment):** το οποίο πρέπει να είναι κατανοητό, ετερογενές και δυναμικό από κάθε άποψη. Πρέπει να υποστηρίζει διαδικασίες υλοποίησης της διασυνδεσιμότητας του πράκτορα με άλλες γλώσσες και πρωτόκολλα. Επιπλέον, θα πρέπει να προσφέρει ανακάλυψη γνώσης και δυνατότητα δικτυακής μετακίνησης.

**vii. αξιοπιστία (reliability):** να προσφέρει δηλαδή ασφάλεια στην επικοινωνία μεταξύ των πρακτόρων. Επιπρόσθετα, η αξιοπιστία ως τμήμα μιας γλώσσας επικοινωνίας πρακτόρων, προάγει την αυθεντικότητα των πρακτόρων. Ταυτόχρονα, προστατεύει τους πράκτορες από ακατάλληλα και κακώς μορφοποιημένα μηνύματα. Το τελευταίο επιτυγχάνεται με μηχανισμούς εύρεσης και εντοπισμού των κακώς κείμενων, με προειδοποιήσεις και με μηνύματα ανάδειξης των λαθών.

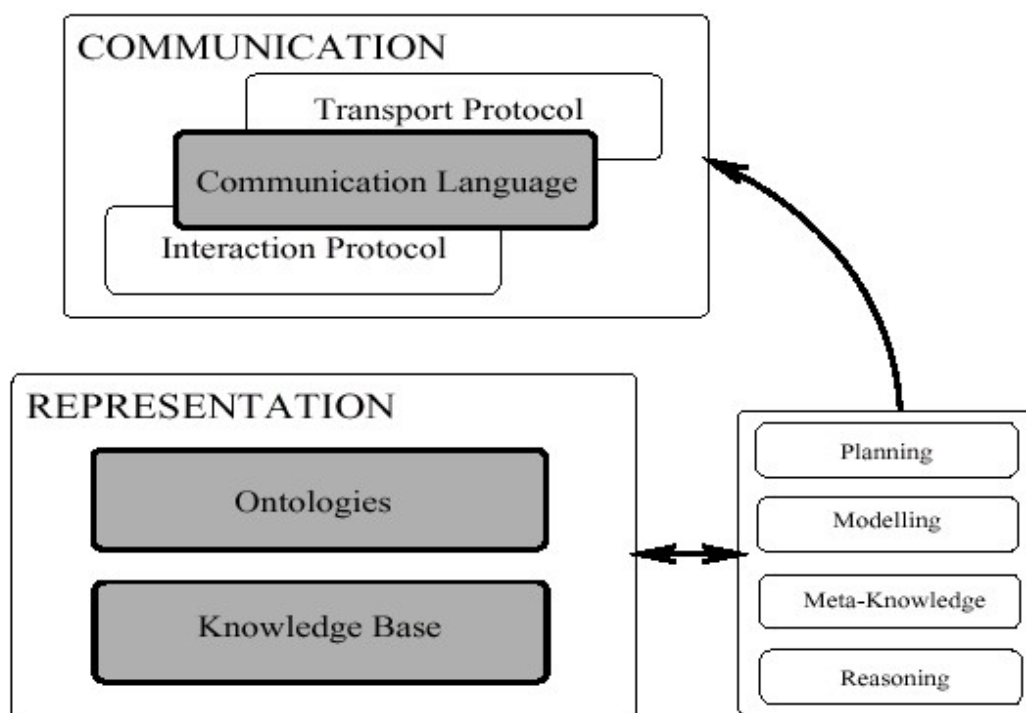
### **3. Επικοινωνία με διαμοιρασμό γνώσης (Knowledge Sharing Effort)**

Υπενθυμίζεται ότι κάθε πράκτορας έχει τη δική του απεικόνιση της πραγματικότητας. Έτσι, ένας πράκτορας πρέπει να αποκτά την κατάλληλη πληροφόρηση, έχοντας μηχανισμούς που μεταφράζουν οτιδήποτε είναι ερέθισμα για

αυτόν (μετάφραση από την μια γλώσσα απεικόνισης στην άλλη). Επιπλέον, σε μια εφαρμογή-πράκτορα που έρχεται σε επαφή με άλλες εφαρμογές-πράκτορες ή τμήματα τους, είναι απαραίτητη η προσφορά και ο διαμοιρασμός του σημασιολογικού και πραγματικού περιεχομένου της γνώσης, που η κάθε συμμετέχουσα εφαρμογή-πράκτορας προσφέρει. Έτσι, θα μπορούν όλοι οι πράκτορες να καταλαβαίνουν τους υπόλοιπους και τελικά οι εφαρμογές να διασυνδέονται μεταξύ τους.

Για την κατανόηση του τρόπου με τον οποίο επιτυγχάνεται η επικοινωνία μεταξύ των πρακτόρων, είναι χρήσιμο το παρακάτω σχήμα.

**Σχήμα 30: Μοντέλο διαλειτουργικότητας και επικοινωνίας πρακτόρων**



Πηγή: Finin, Labrou, Mayfield

Η επικοινωνία μεταξύ των πρακτόρων είναι ένα τριμερές πρόβλημα.

**Επίπεδο 1:** Αρχικά περιλαμβάνει την ύπαρξη ενός πρωτοκόλλου αλληλεπίδρασης. Αυτό αναφέρεται στις διαδικασίες στρατηγικής υψηλού επιπέδου, οι οποίες καθορίζουν το είδος της αλληλεπίδρασης του πράκτορα με τις περιβάλλουσες αυτού οντότητες. Τέτοια είναι τα πρωτόκολλα διαπραγμάτευσης, τα πρωτόκολλα θεωρίας παιγνίων και άλλα πιο απλά, που αντιμετωπίζουν το θέμα της συμπεριφοράς ενός πράκτορα με τη λογική: «αν δε ξέρεις κάτι, βρες κάποιον που ξέρει και ρώτησέ τον».

**Επίπεδο 2:** Δευτερευόντως, υφίσταται η γλώσσα επικοινωνίας πρακτόρων. Επιγραμματικά, η γλώσσα επικοινωνίας πρακτόρων είναι το μέσο από το

οποίο περνούν οι συμπεριφορές του πράκτορα, σε σχέση με το περιεχόμενο που υπάρχει κατά την αλληλεπίδρασή του με άλλες οντότητες. Η γλώσσα επικοινωνίας καθορίζει το πότε η συμπεριφορά του πράκτορα θα περιγράφεται, από κάποιο αίτημα, ερώτημα ή τοποθέτηση προς τις οντότητες με τις οποίες αλληλεπιδρά.

**Επίπεδο 3:** Τρίτη διάσταση για την επικοινωνία είναι η ύπαρξη του πρωτοκόλλου μεταφοράς το οποίο αποτελεί το μηχανισμό μεταφοράς των πληροφοριών μεταξύ των αλληλεπιδρόντων οντοτήτων (τέτοια είναι τα TCP, HTTP, SMTP κ.α.).

Όπως, γίνεται κατανοητό από το σχήμα 30, για να υπάρξει η επικοινωνία θα πρέπει ο κάθε πράκτορας να διαθέτει μια αναπαράσταση της πραγματικότητας (βάση γνώσης και οντολογία). Αυτή θα μετατρέπεται σε θέσεις του πράκτορα κατά την επικοινωνία του για την εξυπηρέτηση των αλληλεπιδράσεων του, μέσω μηχανισμών λογικής, σχεδιασμού, μοντελοποίησης και μετα-γνώσης. Σε επόμενο στάδιο η επικοινωνία θα επιτυγχάνεται μέσω των τριών επιπέδων που αναλύθηκαν παραπάνω.

Τη συγκεκριμένη προβληματική γύρω από την επικοινωνία ήρθε να καλύψει και να δώσει σε αυτή ικανοποιητικά αποτελέσματα η **KSE** (Knowledge Sharing Effort). Βασικοί χορηγοί της προσπάθειας αυτής είναι το ARPA (Advanced Research Projects Agency), το AFOSR (Air Force Office of Scientific Research), το NRI (National Research Initiative) και το NSF (National Science Foundation). Τρεις είναι οι ομάδες στο πλαίσιο της KSE, που ασχολούνται με τον καθορισμό και την οργάνωση υλικού, για την επίτευξη της επικοινωνίας μεταξύ των πρακτόρων και του διαμοιρασμού της γνώσης μεταξύ των συστημάτων.

**1.** Το Interlingua Group, το οποίο ανέπτυξε μια γλώσσα περιγραφής και παρουσίασης του περιεχομένου μιας βάσης γνώσης. Προϊόν του είναι η **KIF** (Knowledge Interchange Format), η οποία χρησιμοποιείται για την υποστήριξη της μετάφρασης μεταξύ δύο διαφορετικών γλωσσών ή σαν μια γλώσσα μεταξύ δύο πρακτόρων που διαθέτουν διαφορετική δομή και γλώσσα αναπαράστασης της πραγματικότητας. Αντίστοιχη με την KIF είναι η Knowledge Representation Specification Language (KRSL), η οποία ορίζει κοινές δομές σε ομάδες γλωσσών αναπαράστασης.

**2.** Το SRKB Group, (Shared Reusable Knowledge Bases) το οποίο εργάζεται πάνω στο περιεχόμενο των διαμοιραζομένων βάσεων γνώσεων και πιο συγκεκριμένα στο κομμάτι της ανάδειξης και καθορισμού της ορολογίας συγκεκριμένων θεματικών περιοχών (οντολογιών) με ανάπτυξη εργαλείων και μεθοδολογιών. Προϊόν του είναι



η **Ontolingua**.

**3.** Το External Interfaces Group, το οποίο εξετάζει την αλληλεπίδραση μεταξύ των συστημάτων γνώσης και άλλων οντοτήτων (άλλα συστήματα γνώσης, συστήματα διαχείρισης βάσεων δεδομένων κ.α.) σε ένα εξελισσόμενο περιβάλλον. Προϊόν του συγκεκριμένου γκρουπ είναι το πρωτόκολλο-γλώσσα **KQML**.

### 3.1. KIF (Knowledge Interchange Format)

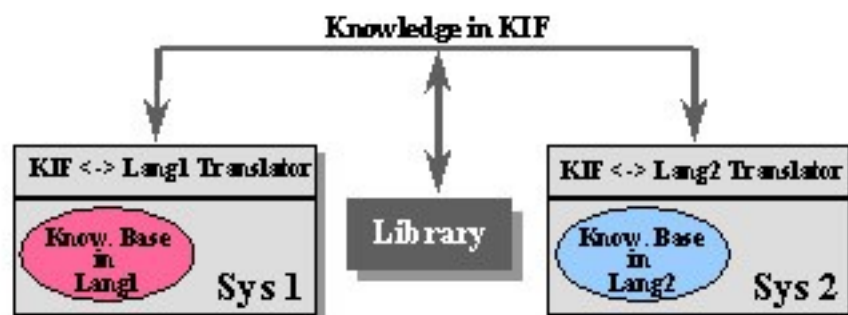
Η γλώσσα KIF επιτρέπει την ανταλλαγή γνώσης μεταξύ των πρακτόρων για οποιαδήποτε αντικείμενα του κόσμου. Η σύνταξη της βασίζεται στη λογική και υποστηρίζει σχεδόν όλα τα συνηθισμένα συνδεδετικά αυτής.

Η KIF:

- ❖ είναι βασισμένη στη λογική
- ❖ έχει αυστηρά καθορισμένη σημασιολογία
- ❖ είναι ικανοποιητικά εκφραστική
- ❖ εκφράζει τη γνώση και μετα-γνώση που ανταλλάσσεται μεταξύ των πρακτόρων
- ❖ επιτρέπει την ημιαυτόματη μετάφραση από και προς γνωστές γλώσσες αναπαράστασης
- ❖ διαθέτει σημασιολογία παρόμοια με αυτή της λογικής πρώτης τάξης

Η ανάπτυξη της KIF έρχεται να εξυπηρετήσει την ανάπτυξη ευφυών εφαρμογών και να τους δώσει τη δυνατότητα να διασυνδέονται. Επίσης, βοηθά στη δημιουργία μιας κοινά αποδεκτής μορφή διασύνδεσης που θα επιτυγχάνει τη μετάφραση μιας γλώσσας A στην KIF και από την KIF σε μια γλώσσα B.

**Σχήμα 31: Μοντέλο διαλειτουργικότητας και επικοινωνίας πρακτόρων**



Πηγή: Ιδία Επεξεργασία

Η γλώσσα περιλαμβάνει προδιαγραφές τόσο για τη σύνταξη της όσο και για τη

σημασιολογία της.

**A.** αποδίδει με κατανοητό τρόπο την έκφραση απλών δεδομένων

πχ. *Ο μισθός υπαλλήλου με ID που εργάζεται στο τμήμα εργασίας είναι το ποσό*

(salary 015-46-3946 widgets 72000)

(salary 026-40-9152 grommets 36000)

(salary 415-32-4707 fidgets 42000)

**B.** αποδίδει με πιο πολύπλοκους όρους πιο πολύπλοκες πληροφορίες

πχ. *Το τσιπ1 είναι μεγαλύτερο από το τσιπ2*

(> (\* (width chip1) (length chip1)) (\* (width chip2) (length chip 2)))

**Γ.** περιλαμβάνει ποικιλία λογικών τελεστών που βοηθούν την κωδικοποίηση λογικής πληροφορίας (άρνηση, διαίρεση, κανόνες, ποσοτικές συναρτήσεις κ.α.)

πχ. *Ο αριθμός που προκύπτει ανυψώνοντας ένα αριθμό σε ζυγή δύναμη είναι θετικός*

(=> (and (real-number ?x) (even-number ?n)) (> (expt ?x ?n) 0))

**Δ.** χρησιμοποιεί ειδικούς χαρακτήρες και λεξιλόγιο για να εκφράσει διαφορετικές μορφές γνώσης.

πχ. *Ο πράκτορας Joe ενδιαφέρεται στο να λαμβάνει τριάδες μισθών*

(interested joe `(salary ,?x ,?y ,?z))

εάν η πρόταση ήταν (interested joe `(salary ?x ?y ?z)) τότε ο joe θα ενδιαφερότανε για την πρόταση (salary ?x ?y ?z) παρά για τις περιπτώσεις των μισθών όπως συνέβαινε πριν.

**Ε.** μπορεί να περιγράψει διαδικασίες και να δημιουργήσει προγράμματα ή σκριπτάκια για πράκτορες κ.α.

πχ. *Δημιουργείστε ένα πρόγραμμα το οποίο αφήνει αρχικά μια γραμμή κενή, εκτυπώνει στη δεύτερη γραμμή το Hello και προσθέτει μια κενή γραμμή*

(progn (fresh-line t) (print "Hello!") (fresh-line t))

### 3.2. Οντολογίες

Κάθε σύστημα συγκεκριμένης γνώσης βασίζεται στην αντίληψη του κόσμου (αντικείμενα, οντότητες, συσχετίσεις, διαφοροποιήσεις που συνδέονται για την εκτέλεση μιας ενέργειας) και στην αναπαράσταση αυτών σαν έννοιες, διαφοροποιήσεις κ.α.. Μια οντολογία αναφέρεται στην αποκλειστική εξειδίκευση των σχέσεων που υπάρχουν και περιγράφουν την ύπαρξη και την ουσία των όντων. Όταν αυτή υφίσταται για να περιγράψει τις δεσμεύσεις που υπάρχουν μεταξύ

αλληλεπιδρώντων προγραμμάτων, αναφέρεται στις έννοιες και συσχετίσεις (σημασιολογικές ιδιότητες και σχέσεις), που χρησιμοποιούν τα προγράμματα για να αλληλεπιδράσουν μεταξύ τους, με άλλες βάσεις γνώσης και με τους χρήστες. Επιπλέον, σε μια οντολογία περιλαμβάνονται διάφορα αξιώματα (υπάρχοντες περιορισμοί και κανόνες), που διασφαλίζουν την πληρότητα των συσχετίσεων και των εννοιών αλλά και ένα συγκεκριμένο λεξιλόγιο (λεξικό). Όταν οι δράσεις ενός πράκτορα δεσμεύονται και διέπονται από τα συστατικά μιας συγκεκριμένης οντολογίας, αυτή τον περιγράφει.

Πολλές οντολογίες γράφτηκαν σε KIF και χρησιμοποιούν το λεξιλόγιο της Ontolingua. Κάθε οντολογία που δημιουργείται για να υλοποιηθούν λογισμικά συστήματα γνώσης, περιγράφει συγκεκριμένο πεδίο και θεματική. Μια οντολογία ορίζει ομάδα κλάσεων, λειτουργιών και σταθερών για τη συγκεκριμένη θεματική. Επιπλέον, περιλαμβάνει πλήθος αξιωμάτων, για την εκτέλεση της μετάφρασης του κώδικα σε εκτελέσιμες ενέργειες. Αυτές τις εκτελέσιμες ενέργειες αναλαμβάνει η γλώσσα επικοινωνίας να τις μεταφέρει, για να υπάρξει συνομιλία και αλληλεπίδραση μεταξύ των πρακτόρων.

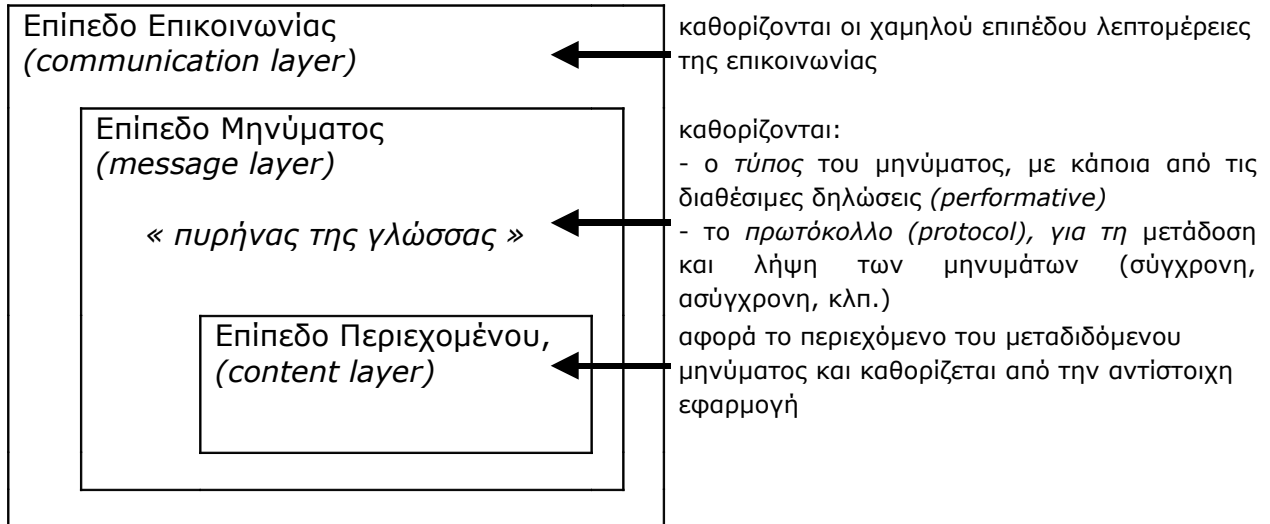
### 3.3. Το πρωτόκολλο επικοινωνίας KQML

Το πρωτόκολλο **KQML** (Knowledge and Query Manipulation Language) τα τελευταία χρόνια βρίσκεται στο επίκεντρο της επιστημονικής έρευνας. Αυτό βασίζεται στη **θεωρία πράξεων λόγου** (Austin, 1962, speech acts).

Το KQML ως πρωτόκολλο επικοινωνίας, για τα μηνύματα που ανταλλάσσονται μεταξύ των πρακτόρων (δηλαδή τις πράξεις / ενέργειες), καθορίζει:

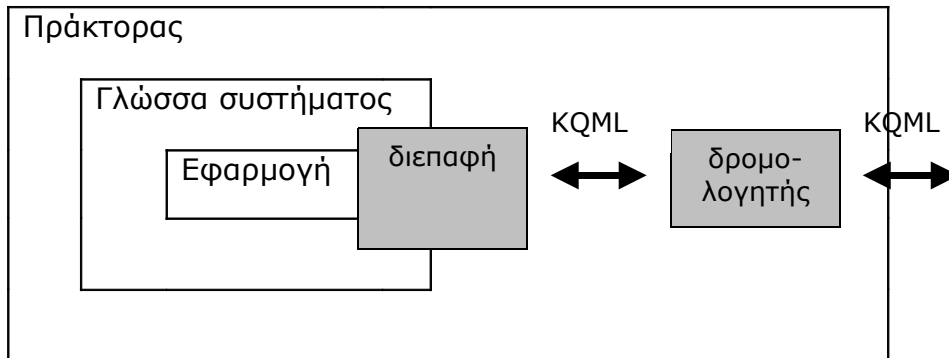
- ❖ τον τύπο-πρόθεση, δηλ. ενημέρωση, προειδοποίηση ή αίτημα (illocution). Οι τύποι των μηνυμάτων (illocutionary force) κατηγοριοποιούνται χρησιμοποιώντας δηλώσεις
- ❖ τη μορφή ή σύνταξη (locution)
- ❖ τη σημασιολογία, δηλαδή το αποτέλεσμα του μηνύματος στον παραλήπτη (perlocution). Με αυτή περιγράφονται τα αντικείμενα των πεδίων, οι διάφορες κατηγορίες των αντικειμένων καθώς και τις σχέσεις μεταξύ τους. Έτσι, όλοι οι πράκτορες που συμμετέχουν στο σύστημα, αποδίδουν στις λέξεις την ίδια σημασία.
- ❖ τον τρόπο διασύνδεσης

Η λειτουργία του KQML χωρίζεται σε τρία επίπεδα:



Η αρχιτεκτονική των KQML πρακτόρων απαιτεί την ύπαρξη μιας διεπαφής μεταξύ της KQML και της γλώσσας του συστήματος και ένα δρομολογητή που θα αναλαμβάνει την επικοινωνία σε χαμηλό επίπεδο (σύνδεση των ενεργειών των πρακτόρων με φυσικές διευθύνσεις που δίνονται στο μήνυμα με συμβολικά ονόματα, εγγραφή σε βάσεις δεδομένων ή υπηρεσίες και προώθηση ή επέκταση της αναζήτησης κα.).

**Σχήμα 32: Αρχιτεκτονική KQML**



Πηγή: Vlachavas I. κ.α., 2002

Ένα μήνυμα στην KQML είναι:

Μορφή	Επεξήγηση	Επίπεδο
( <performative>	δήλωση	
: sender	ο αποστολέας	επικοινωνίας
: receiver	ο παραλήπτης	επικοινωνίας
: language	η γλώσσα κωδικοποίησης (PROLOG, LISP, SQL ή άλλη γλώσσα επικοινωνίας πρακτόρων)	μηνύματος
: ontology	το λεξιλόγιο για τη κατανόηση του μηνύματος (σημασιολογία)	μηνύματος
:content	το περιεχόμενο του μηνύματος	περιεχομένου
)		

Οι δηλώσεις μπορούν να ομαδοποιηθούν στις παρακάτω κατηγορίες:

- ❖ Assertives - Δηλώσεις κάποιου γεγονότος
- ❖ Directives - Εντολές
- ❖ Commisives - Δεσμεύσεις
- ❖ Declaratives - Δηλώσεις γεγονότων
- ❖ Expressives - Εκδηλώσεις συναισθημάτων

Βάσει αυτών χαρακτηρίζονται τα μηνύματα έτσι έχουμε τους παρακάτω τύπους.

Τύποι μηνυμάτων	Διαθέσιμες Δηλώσεις
Βασική Ερώτηση (Basic Query):	evaluate, ask-if, ask-about, ask-one, ask-all
Ερώτηση πολλαπλών απαντήσεων (Multi-response query):	stream-about, stream-all, eos
Απάντηση (Response):	reply, sorry
Γενική Πληροφόρηση (Generic Informational):	tell, achieve, cancel, untell, unachieved
Γεννήτρια (Generator):	standby, ready, next, rest, discard, generator
Ορισμός Ικανοτήτων (Capability Definition):	advertise, subscribe, monitor, import, export
Δικτύωση (Networking):	register, unregister, forward, broadcast, route

Πηγή: Vlachavas I. κ.α., 2002

### 3.4. Παραδείγματα μηνυμάτων και επικοινωνίας με την KQML

**Παράδειγμα 1:** Αυτό περιγράφει μήνυμα ερώτησης, για την τιμή της μετοχής «ΣΤΥΛΟ ΑΕ», στη χρονική στιγμή που αποστέλλεται το μήνυμα. Ο πράκτορας που ρωτά, λέγεται «επενδυτής», εκείνος που απαντά «χρηματιστής», η γλώσσα κωδικοποίησης του μηνύματος είναι η «PROLOG», και το λεξιλόγιο (οντολογία) που θα χρησιμοποιηθεί αφορά το χρηματιστήριο Αθηνών, «ΧΑΑ». Έτσι:

```
(ask
  : sender επενδυτής
  : receiver χρηματιστής
  : language standard Prolog
  : ontology XAA
  : content price ("ΣΤΥΛΟ ΑΕ",Price))
```

**Παράδειγμα 2:** Παρακάτω φαίνεται μήνυμα ερώτησης για την τιμή της μετοχής της «IBM» (τη στιγμή που γίνεται η ερώτηση), στο χρηματιστήριο της Νέας Υόρκης. Ο πράκτορας που ρωτά, λέγεται «joe», εκείνος που απαντά ονομάζεται «stock-server», η γλώσσα κωδικοποίησης του μηνύματος είναι η «LPROLOG», και το λεξιλόγιο (οντολογία) που θα χρησιμοποιηθεί αφορά το χρηματιστήριο της Νέας Υόρκης - «NYSE - TICKS».

Επιπρόσθετα, ο «stock-server» απαντά στον «joe» και του αναφέρει την τιμή της

μετοχής.

Έτσι, η ερώτηση του joe προς τον stock-server είναι:

```
(ask-one
: sender joe
: content (PRICE IBM ?price)
: receiver stock-server
: reply-with ibm-stock
: language LPROLOG
: ontology NYSE-TICKS)
```

Και η απάντηση του stock-server στον joe, για την τιμή της IBM, τη συγκεκριμένη χρονική στιγμή, είναι:

```
(tell
: sender stock-server
: content (PRICE IBM 14)
: receiver joe
: in reply-to ibm-stock
: language LPROLOG
: ontology NYSE-TICKS)
```

**Παράδειγμα 3:** Αν θέλει ο joe όλες τις τιμές της IBM, για να δει την πορεία της, τότε θα ρωτήσει τον «stock-server» ως εξής:

```
(ask-all
: sender joe
: content "price(IBM, [?price, ?time])"
: receiver stock-server
: language standard PROLOG
: ontology NYSE-TICKS)
```

Η γλώσσα κωδικοποίησης του μηνύματος σε αυτή την περίπτωση είναι η «PROLOG». Οι τιμές της μετοχής, για κάθε χρονική στιγμή, που θα λάβει ο joe, θα είναι ομαδοποιημένες σε ένα μήνυμα.

**Παράδειγμα 4:** Αν θέλει ο joe όλες τις τιμές όλων των μετοχών, για να δει την πορεία τους σε διαφορετικά μηνύματα (ένα για κάθε μετοχή), θα ρωτήσει τον «stock-server»:

```
(stream-all
: sender joe
;;?VL is a large set of symbols
: content (PRICE ?VL ?price)
: receiver stock-server
: language standard PROLOG
: ontology NYSE-TICKS)
```

**Παράδειγμα 5:** Αν θέλει ο joe όλες τις τιμές όλων των μετοχών, για να δει την πορεία τους σε διαφορετικά μηνύματα (ένα για κάθε μετοχή) και με παραλαβή αυτών σταδιακά (στέλνοντας αποδοχή για αυτά), θα ρωτήσει τον «stock-server»:

```
(standby
: sender joe
: content (stream-all
           : content (PRICE ?VL ?price))
: receiver stock-server
: language standard PROLOG
: ontology NYSE-TICKS)
```

**Παράδειγμα 6:** Αν θέλει ο joe όλες τις τιμές της IBM, για να δει την πορεία της, αλλά και να εγγραφεί για μελλοντική του ενημέρωση, σε πιθανές αλλαγές της τιμής της, τότε θα ρωτήσει τον «stock-server» ως εξής:

```
(subscribe
: sender joe
: content (stream-all
           : content (PRICE IBM ?price))
: receiver stock-server
: language standard PROLOG
: ontology NYSE-TICKS)
```

**Παράδειγμα 7:** Κάποιος πράκτορας, μπορεί επίσης να ανακοινώσει τα είδη των μηνυμάτων (δηλώσεων) που δύναται να διαχειριστεί (advertise), ή να βρει πράκτορες, ώστε να επικοινωνήσει μαζί τους για συγκεκριμένα είδη μηνυμάτων (recruit).

Έτσι, για παράδειγμα αν κάποιος πράκτορας είναι αντίστοιχος του «stock-server» και θέλει να ενημερώσει τους άλλους, ότι μπορεί να τους εξυπηρετήσει για να βρουν τιμές μετοχών, τότε ενεργεί ως εξής:

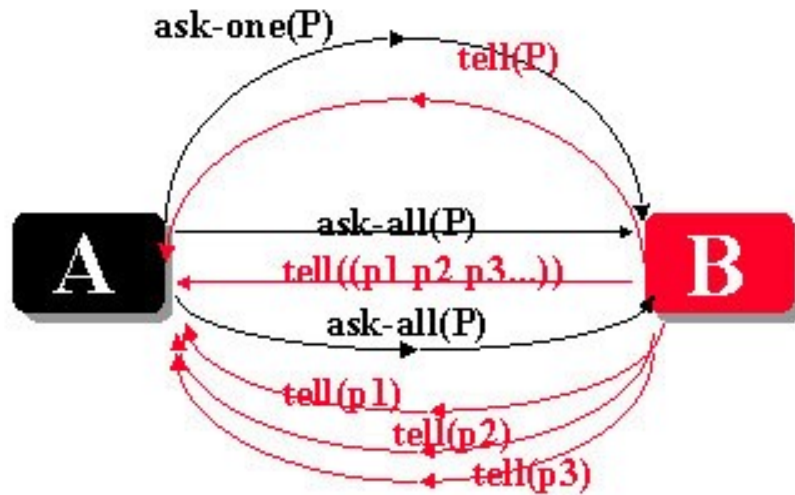
```
(advertise
: ontology NYSE-TICKS
: language LPROLOG
: content (stream-all
           : content (PRICE ?x ?y))
```

### 3.5. Πρωτόκολλα ανταλλαγής πληροφορίας στην KQML

Οι προαναφερθείσες δηλώσεις ή τύποι μηνυμάτων, αποτελούν η καθεμιά ένα πρωτόκολλο ανταλλαγής πληροφορίας στην KQML. Παρακάτω γίνεται αναλυτική αναφορά στις πιο σημαντικές από αυτές τις δηλώσεις.

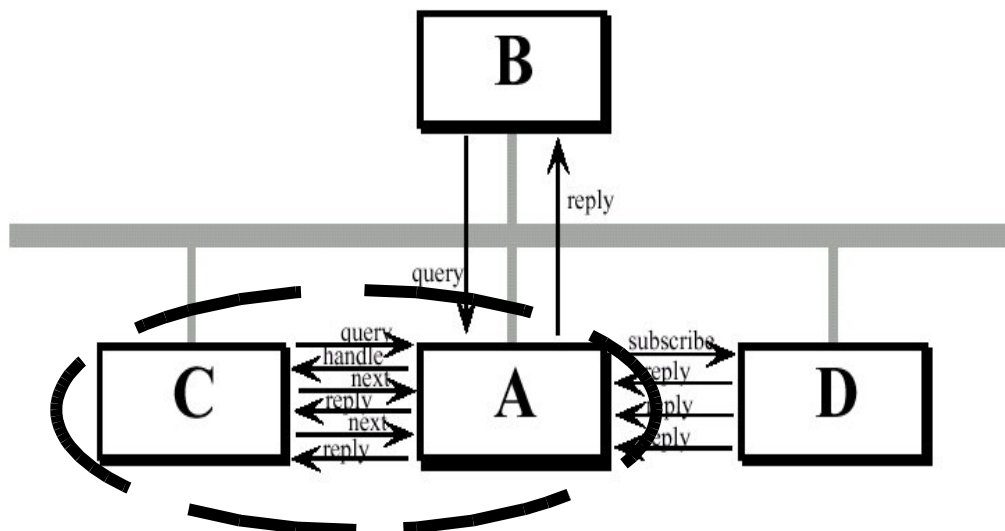
Όταν ένας πράκτορας (πελάτης), ερωτά ένα άλλο για να του εξυπηρετήσει κάποιο αίτημα (ask), τότε ο άλλος παίζει το ρόλο του εξυπηρετητή και απαντά με απλή απάντηση ή με ομάδα απαντήσεων. (Point to point protocol)

**Σχήμα 33: Δηλώσεις ask-one, ask-all, ask-if, ask-about, ask**



Όταν ένας πράκτορας (πελάτης) ερωτά έναν άλλο για κάτι και θέλει να πάρει σειρά απαντήσεων, καθορίζοντας ο ίδιος τη χρονική ακολουθία των απαντήσεων (stream-all), τότε ο άλλος (εξυπηρετητής) αποστέλλει τις απαντήσεις, ανάλογα αν δέχεται την αποδοχή του πελάτη ή όχι.

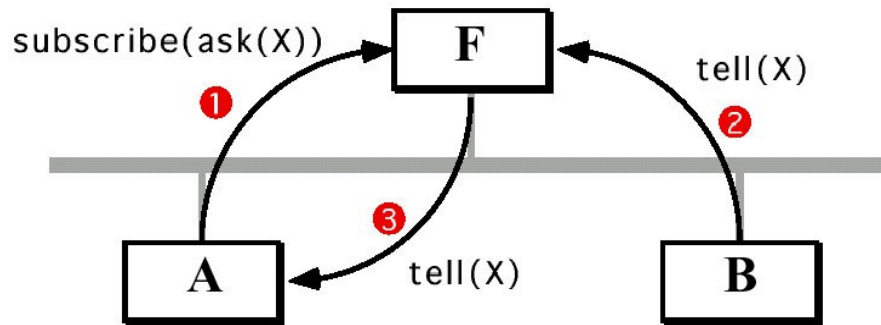
**Σχήμα 34: Δηλώσεις stream-about, stream-all**



Επιπλέον, μπορεί ο πελάτης να απαιτήσει από τον εξυπηρετητή, να τον ενημερώνει κάθε στιγμή που συμβαίνει κάτι καινούργιο, σχετικό με το αρχικό του αίτημα. (subscribe - ασύγχρονη επικοινωνία)

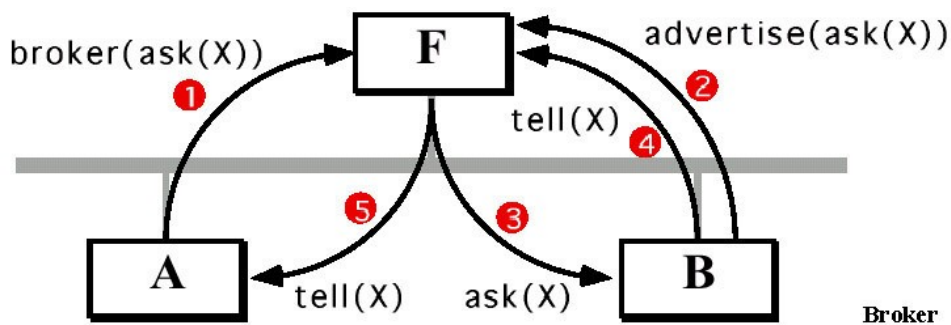
**Σχήμα 35: Δήλωση subscribe**





Πολλές φορές κάποιος πράκτορας (πελάτης), ζητά από έναν εξυπηρετητή να βρει έναν άλλο πράκτορα (με τη δήλωση broker), ο οποίος μπορεί να παράσχει πληροφορία στα αιτήματα του. Ο εξυπηρετητής όταν βρίσκει κάποιον πράκτορα που έχει απάντηση στα αιτήματα του πελάτη του (πρώτος πράκτορας), λαμβάνει από αυτόν τις κατάλληλες πληροφορίες (απαντήσεις) και τα προωθεί στον πελάτη του. Ο εξυπηρετητής βρίσκει πράκτορες που μπορούν να δώσουν απαντήσεις σε αιτήματα πελατών του, αφού πρώτα αυτοί τον ενημερώνουν για τη διαθεσιμότητά τους (κάνουν δήλωση advertise).

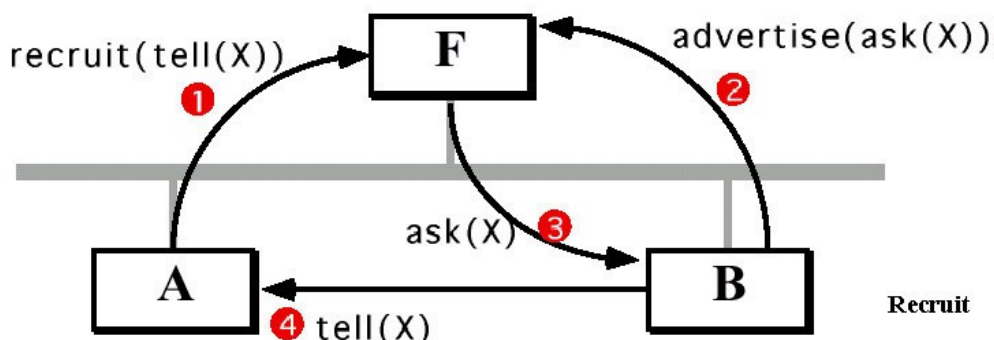
Σχήμα 36: Δηλώσεις broker και advertise



Broker

Στην προηγούμενη περίπτωση εάν ο πελάτης δηλώσει το αίτημά του με recruit, τότε ο πράκτορας που θα βρεθεί από τον εξυπηρετητή, για να παράσχει πληροφορία στον πρώτο (πελάτη), θα την προωθήσει κατευθείαν και όχι μέσω του εξυπηρετητή.

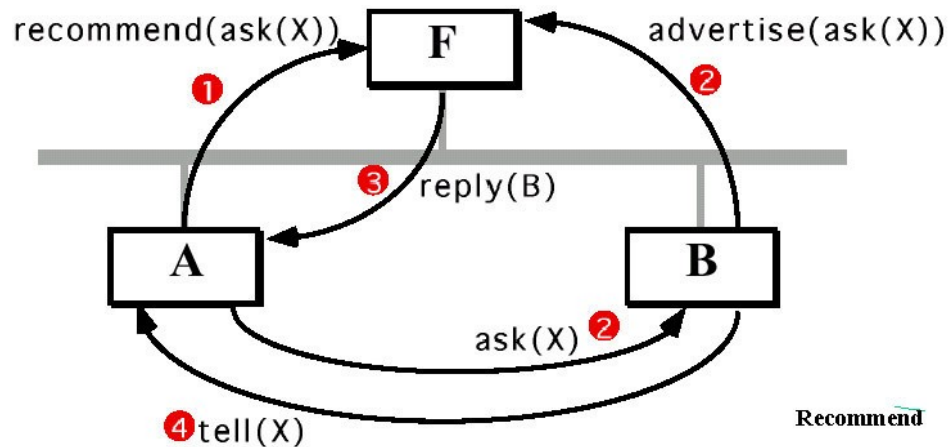
Σχήμα 37: Δήλωση recruit



Recruit

Συχνά συμβαίνει κάποιος πράκτορας να ζητά από ένα εξυπηρετητή να του βρει ένα άλλο πράκτορα, για να του απαντήσει στα αιτήματά του και αφού αυτός βρεθεί, ο πρώτος αποστέλλει αίτημα στο δεύτερο και επικοινωνούν πλέον οι δυο τους (ο πρώτος κάνει δήλωση recommend).

Σχήμα 38: Δήλωση recommend



Για την επικοινωνία μεταξύ διαφορετικών πρακτόρων-εφαρμογών, πρέπει ο καθένας από αυτούς να δηλώνεται ότι υφίσταται στους μετατροπείς (πχ ANS). Η δήλωση γίνεται με την χρήση της εντολής register, από τη πλευρά του κάθε πράκτορα (πχ πράκτορας B) και οι μετατροπείς αναλαμβάνουν την ενημέρωση των πρακτόρων για την ύπαρξη των υπολοίπων.

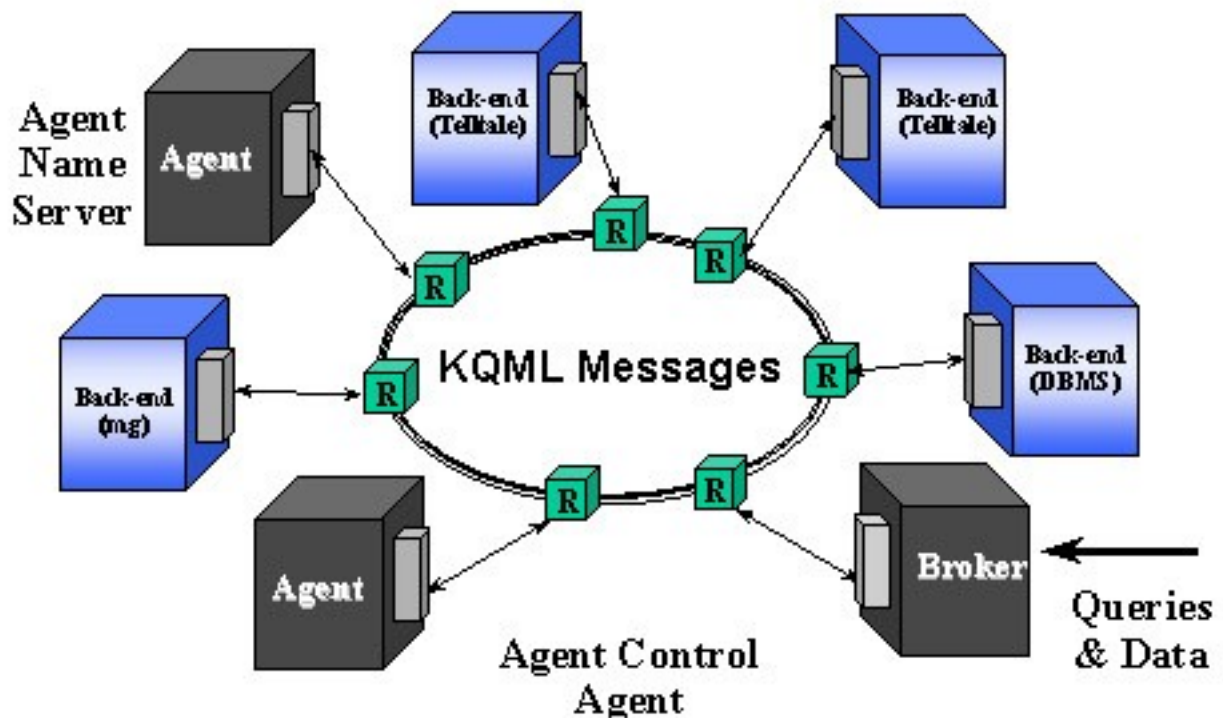
Σχήμα 39: Δήλωση register



Παρακάτω εμφανίζεται ο τρόπος με τον οποίο διάφορες εφαρμογές, πράκτορες και άλλες οντότητες επικοινωνούν μεταξύ τους με μηνύματα σε KQML. Αυτό που πρέπει να σημειωθεί είναι, ότι κάθε πράκτορας, εφαρμογή ή άλλη οντότητα έχει έναν μετατροπέα, ο οποίος στην ουσία αναγνωρίζει και ελέγχει την κάθε οντότητα που

συμμετέχει και της παρέχει τη δυνατότητα επικοινωνίας με τις άλλες (βλ. σχήμα 28, σελ 61).

**Σχήμα 40: Επικοινωνία εφαρμογών, πρακτόρων και άλλων οντοτήτων που βρίσκονται σε δίκτυο, με μηνύματα σε KQML**



### 3.6. Εφαρμογές με τη χρήση της KQML

Σαν πρώτη εφαρμογή μπορεί να αναφερθεί, αυτή της δημιουργίας ενός ολοκληρωμένου συστήματος σχεδιασμού και προγραμματισμού, για τη διαχείριση των logistics των στρατιωτικών μετακινήσεων. Σε αυτή τη εφαρμογή, που ως γλώσσα επικοινωνίας χρησιμοποιήθηκε η KQML, υλοποιήθηκε ένας πράκτορας που είχε ικανότητες σχεδιασμού (SIPE), βασιζόμενος σε συγκεκριμένο χρονικό πρόγραμμα δραστηριοτήτων (προγραμματισμένο σε Lisp), διέθετε μια βάση γνώσης (σε LOOM) και κατέληγε σε λογικές ενέργειες μέσω ενός εργαλείου σε Lisp. Όλα τα παραπάνω τμήματα του πράκτορα προϋπήρχαν και η KQML τους προσέφερε τον τρόπο για να διασυνδέονται και να συνεργάζονται, ως τμήματα ενός κατακεντρωμένου περιβάλλοντος.

Άλλη εφαρμογή είναι αυτή ενός πληροφοριακού πράκτορα, που διέθετε: **α)** συγκεκριμένο συνεργασιακό τρόπο επικοινωνίας (CoBASE), **β)** έναν ενδιάμεσο για σχεδιασμό της πρόσβασης στην πληροφορία (SIMS) και **γ)** έναν άλλο ενδιάμεσο για μετάφραση των σχεσιακών δεδομένων σε δομές γνώσεις (LIM).

Το CoBASE κομμάτι επεξεργαζότανε κάποιο ερώτημα και στην περίπτωση που δε βρίσκονταν απαντήσεις το ξαναεκτελούσε πιο απλοποιημένο. Ταυτόχρονα, ενεργοποιούνταν αναζήτηση στο SIMS, το οποίο διέθετε γνώση για διάφορες πληροφοριακές πηγές και διαμοίραζε σε ορισμένες από αυτές το ερώτημα για να βρεθεί απάντηση. Το LIM κομμάτι, το οποίο ήταν εξυπηρετητής γνώσης, απαντούσε στο ερώτημα φέρνοντας τα αποτελέσματα από την αναζήτηση στις διάφορες βάσεις δεδομένων.

Άλλο παράδειγμα χρήσης της KQML είναι αυτό του Agent Based Software Integration (Stanford University), στο οποίο η KQML αποτελούσε το ολοκληρωμένο πλαίσιο ενοποίησης διαφόρων συστημάτων λογισμικού. Σε αυτό το παράδειγμα υπήρχε ένας δρομολογητής - πράκτορας, που επεξεργάζονταν και ανάλυε το περιεχόμενο των KQML μηνυμάτων και υποστήριζε την ενοποίηση, την αλληλεπίδραση και την ταυτόχρονη λειτουργία διαφόρων συστημάτων λογισμικού.

Στο Stanford University δημιουργήθηκε επίσης η **ACL**, που είναι γλώσσα επικοινωνίας πρακτόρων (Agent Communication Language) και αποτελεί υλοποίηση της KQML με δέσμευση στην KIF - ως γλώσσα περιεχομένου. Αυτή χρησιμοποιείται σε μεγάλης κλίμακας συλλειτουργώντας και διασυνδεδεμένα συστήματα λογισμικού και τα αποτελέσματα είναι ικανοποιητικά. Αποτελεί το μέλλον για τη λειτουργία και υλοποίηση ετερογενών συστημάτων λογισμικού.

Τέλος, πολλές εφαρμογές αναζήτησης και ανάκτησης δεδομένων σε κατανεμημένα ή δικτυακά περιβάλλοντα χρησιμοποιούν την KQML.

## **4. Άλλες γλώσσες επικοινωνίας**

### **4.1. Σύγχρονες οντοκεντρικές γλώσσες (Concurrent object languages)**

Αυτές αποτελούν τους προγόνους των πρακτορικών γλωσσών. Η βασική τους ιδέα, είναι αυτή ενός αυτόνομου εκτελέσιμου αντικειμένου με εσωτερική κατάσταση, το οποίο επικοινωνεί με μηνύματα με άλλα αντικείμενα. Οι συγκεκριμένες γλώσσες είναι πολύ κοντά στην ιδέα της ύπαρξης των πρακτόρων. Παραδείγματα είναι το μοντέλο ηθοποιών του Hewitt (1977), η ABSL (Yonezawa, 1990) κ.α.

### **4.2. Agent oriented programming - AOP (Πρακτοροστρεφής προγραμματισμός)**

Ο πρακτοροστρεφής προγραμματισμός είναι ιδέα του Shoham (**Agent0**, 1990-1993), και βασίζεται σε μια πιο εκκοσμηκευμένη άποψη προγραμματισμού και υπολογισμών. Ο άνθρωπος χρησιμοποιεί μια εκ προθέσεως στάση σύνθεσης, για την

αντίληψη και παρουσίαση των ιδιοτήτων ενός πολύπλοκου συστήματος. Ο πρακτοροστρεφής προγραμματισμός βασίζεται σε αυτή τη λογική. Τα συστατικά του είναι:

- ❖ ένα λογικό σύστημα που ορίζει τη νοητική κατάσταση των πρακτόρων το οποίο περιέγραφε και καθόριζε τις πεποιθήσεις, τη δέσμευση και την ικανότητα των πρακτόρων
- ❖ μια γλώσσα προγραμματισμού
- ❖ μια διαδικασία πρακτοροποίησης, η οποία θα μετατρέπει τα προγράμματα των πρακτόρων σε χαμηλού επιπέδου εκτελέσιμα συστήματα (αίσθηση/αντίδραση). πχ το παράδειγμα των αυτόματων του Kaelbling & Rosenschein

Τα προϊόντα του πρακτοροστρεφή προγραμματισμού, είναι συνήθως πράκτορες που επικοινωνούν και εξελίσσονται. Η κατάστασή τους που περιγράφει αυτούς και το περιβάλλον που ζουν, αποτελείται από πεποιθήσεις, ικανότητες, επιλογές και δεσμεύσεις.

Πιο συγκεκριμένα, παράδειγμα τέτοιας γλώσσας προγραμματισμού είναι η γλώσσα Agent0. Αυτή διαθέτει συγκεκριμένη σημασιολογία που περιγράφει τις πιθανές ενέργειες κάθε πράκτορα, και παρέχει συγκεκριμένους όρους για την αλληλεπίδραση των πρακτόρων.

Κάθε προγραμματισμένος πράκτορας στην Agent0 έχει:

- ❖ μια ομάδα από καθορισμένες ικανότητες να κάνει πράγματα
- ❖ μια ομάδα πεποιθήσεων και δεσμεύσεων
- ❖ μια ομάδα κανόνων συμπεριφοράς του. Κάθε κανόνας περιέχει μια κατάσταση μηνύματος, μια νοητική κατάσταση και μια ενέργεια.

Για να πυροδοτηθεί ένα κανόνας συμπεριφοράς του πράκτορα, πρέπει:

- α. τα μηνύματα που έχει πάρει ο πράκτορας να ταυτίζονται με τη κατάσταση μηνυμάτων που τον καθορίζουν, και
- β. οι πεποιθήσεις του πράκτορα να ταυτίζονται με τη νοητική κατάσταση του κανόνα.

Αν πυροδοτηθεί ένας συγκεκριμένος κανόνας, τότε ο πράκτορας ενεργεί σύμφωνα με το πως ο κανόνας ορίζει. Οι συμπεριφορές που ορίζονται από κάθε κανόνα είναι είτε ιδιωτικές (μεταβάλλει την εσωτερική κατάσταση του πράκτορα, είτε επικοινωνιακές (αποστολή μηνυμάτων σε άλλους πράκτορες).

Τα μηνύματα που στέλνει ένας πράκτορας έχουν τη μορφή:

- ❖ αιτημάτων ή αναίρεσης αιτημάτων και επηρεάζουν τις δεσμεύσεις του πράκτορα
- ❖ πληροφοριών και επηρεάζουν τις πεποιθήσεις του πράκτορα

Μετεξέλιξη του AOP και της Agent0 είναι η Agent K, η οποία χρησιμοποιεί την KQML για την επικοινωνία των πρακτόρων. Το σημαντικό είναι ότι υπάρχει πρόβλημα στην επικοινωνία των πρακτόρων, που είναι προγραμματισμένοι με φιλοσοφία AOP, με κάποιους άλλους που ακολουθούν άλλη φιλοσοφία σχεδιασμού.

### **4.3. PLACA (PLA nning C ommunication A gents)**

Η συγκεκριμένη γλώσσα βασίζεται πολύ στην Agent0 και την υλοποίησε η Thomas το 1993. Πλέον, οι πράκτορες μπορούν να σχεδιάζουν και να ανταλλάσσουν αιτήματα για συγκεκριμένη συμπεριφορά, τα οποία εξυπηρετούν στόχους ενός υψηλότερου επιπέδου. Το λογικό της μοντέλο είναι ίδιο με αυτό της Agent0 και επιπλέον περιέχει τελεστές για σχεδιασμό ενεργειών και επίτευξης στόχων. Το όλο ερευνητικό πρόγραμμα όμως εξακολουθεί να βρίσκεται σε πειραματικό στάδιο.

### **4.4. METATEM**

Η METATEM αναπτύχθηκε από τον Fischer το 1994. Ένα σύστημα δομημένο στη METATEM περιλαμβάνει πολλούς ταυτόχρονα εκτελέσιμους πράκτορες, ο καθένας από τους οποίους μπορεί να επικοινωνεί με τους υπολοίπους μέσω ασύγχρονης αποστολής μηνυμάτων. Κάθε πράκτορας συμπεριφέρεται σύμφωνα με μια χρονικά προσδιορισμένη λογική. Αντιδρά σε ότι δέχεται και αυτή του η αντίδραση χτίζει ταυτόχρονα ένα λογικό μοντέλο με το τι βιώνει αυτός εκείνη τη στιγμή. Βάσει αυτού μπορεί να συνεχίσει τη ζωή του.

### **4.5. APRIL & MAIL**

Αυτές είναι δύο γλώσσες οι οποίες αναπτύχθηκαν στο πλαίσιο του ερευνητικού προγράμματος IMAGINE (1994) από τους Haugeneder και McCabe & Clark.

Η APRIL σχεδιάστηκε για να μπορεί ένας πράκτορας να καταλαβαίνει τις αρχιτεκτονικές άλλων πρακτόρων ή συστημάτων. Προσφέρει σε ένα πράκτορα πολυεπεξεργασία, επικοινωνία, κατανόηση του περιβάλλοντος και ικανότητες συμβολικής επεξεργασίας.

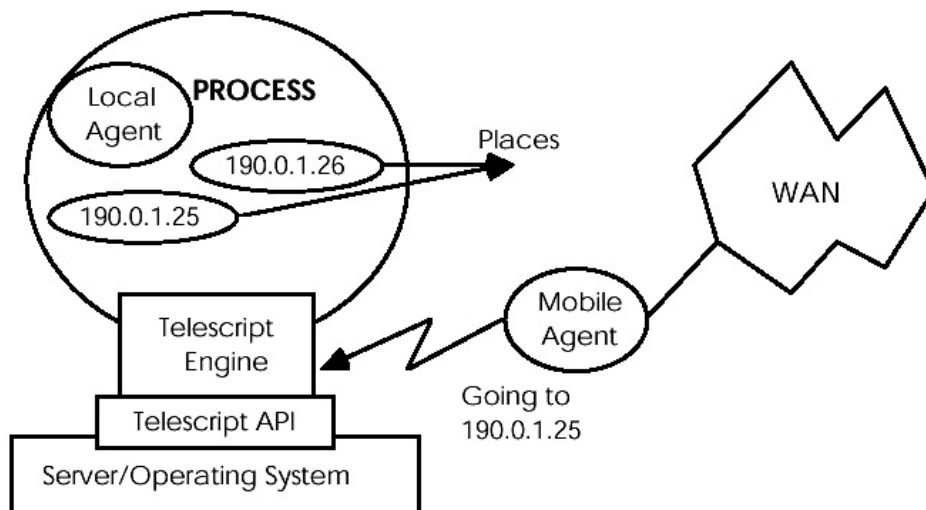
Η MAIL προσφέρει στον πράκτορα μια πλούσια συλλογή από προκαθορισμένες συνθέσεις σχεδίων. Χρησιμοποιείται περισσότερο σε πολυπρακτορικά συστήματα.

Πολυπρακτορικό σύστημα που βασίζεται στη MAIL, υπάρχει για τον έλεγχο και τη διαχείριση του κυκλοφοριακού φόρτου μιας αστικής περιοχής.

#### 4.6. TELESCRIPT

Η TELESCRIPT είναι αντικειμενοστραφής γλώσσα προγραμματισμού που επιτρέπει την ανάπτυξη καταμεμημένων εφαρμογών. Κάθε εφαρμογή που αναπτύσσεται στη συγκεκριμένη γλώσσα αποτελείται από **πράκτορες** (TELESCRIPT *agents*), που κινούνται μέσω **διαδικασιών μετακίνησης**, σε έναν κόσμο που περιέχει **TELESCRIPT μηχανές** (TELESCRIPT *engines*) και **τοποθεσίες** (TELESCRIPT *places*).

Σχήμα 41 : Μηχανή TELESCRIPT



Για την TELESCRIPT έχει υπάρξει αναφορά στην παράγραφο 6.1. σελ. 45.

Για την υποστήριξη της TELESCRIPT αναπτύχθηκαν τα παρακάτω:

- ❖ η TELESCRIPT γλώσσα, η οποία βοηθά τους πράκτορες στην πλοήγησή, μεταφορά, αυτονομία και πρόσβαση
- ❖ η TELESCRIPT μηχανή (ο μεταφραστής της γλώσσας), που διατηρεί τις τοποθεσίες, προγραμματίζει την εκτέλεση των ενεργειών των πρακτόρων, διαχειρίζεται την επικοινωνία και μεταφορά των πρακτόρων και προσφέρει ένα περιβάλλον διεπαφής με άλλες εφαρμογές
- ❖ το TELESCRIPT πρωτόκολλο, που κωδικοποιεί και αποκωδικοποιεί τις ενέργειες και τους ίδιους τους πράκτορες και υποστηρίζει τη μεταφορά τους
- ❖ τα TELESCRIPT εργαλεία ανάπτυξης εφαρμογών σε TELESCRIPT

#### 4.7. ABLE (Agent Behaviour Language)

Η ABLE είναι γλώσσα που υποστηρίζει αντιδραστικούς πράκτορες, ενώ οι προηγούμενες αναφέρονταν σε πράκτορες εσωτερικής κατάστασης. Το 1992 οι Connah και Wavish την ανέπτυξαν, με σκοπό οι πράκτορες να προγραμματίζονται με απλούς όρους και κανόνες-άδειες.

Οι κανόνες ή άδειες του κάθε πράκτορα εμπεριέχουν τη διάσταση του χρόνου. Δηλαδή, όταν ενεργεί με κάποιο τρόπο ο πράκτορας σημασία έχει ο χρόνος που συμβαίνει η ενέργεια. Ακολουθεί τη φιλοσοφία των αυτομάτων και αναλόγως τα ερεθίσματα, επιλέγεται ο κατάλληλος κανόνας συμπεριφοράς.

### 5. Πρωτόκολλα Αλληλεπίδρασης

Με αυτά οι πράκτορες ανταλλάσσουν ακολουθίες μηνυμάτων και υλοποιούν συζητήσεις (conversations), για την επίτευξη της συνεργασίας μεταξύ του. Μέσω της συνεργασίας επιτυγχάνεται η λειτουργία των πολυπρακτορικών συστημάτων.

Έτσι:

- ❖ αποφεύγεται το χάος των συγκρουόμενων ενεργειών των πρακτόρων
- ❖ ικανοποιούνται οι περιορισμοί που έχουν τεθεί σε διαδικασίες, όπως το κόστος της σχεδίασης (design) ενός συστήματος
- ❖ συνδυάζεται το σύνολο των ικανοτήτων των πρακτόρων, κατά την επίλυση του προβλήματος, κάτι που παραπέμπει στην χρησιμοποίηση κατανεμημένης εμπειρογνωμοσύνης
- ❖ αυξάνεται η αποδοτικότητα και μειώνεται ο χρόνος επίλυσης του προβλήματος

#### 5.1. Τεχνικές βασισμένες στην οργάνωση (organizational structure)

Οι πράκτορες διαθέτουν συγκεκριμένη οργανωτική δομή. Έτσι καθένας διαθέτει συγκεκριμένο ρόλο, υπευθυνότητες και ικανότητες. Με τις συγκεκριμένες τεχνικές, η κατανομή της εργασίας μεταξύ των πρακτόρων μπορεί να γίνεται με δύο τρόπους.

**1. master-slave αρχιτεκτονική**, όπου ένας κεντρικός πράκτορας αναλαμβάνει το διαχωρισμό του προβλήματος σε υποπροβλήματα. Αυτά διαμοιράζονται στους διάφορους άλλους πράκτορες (slaves). Αυτοί συνήθως επικοινωνούν μεταξύ τους και επιστρέφουν πάντοτε τα αποτελέσματα στον κεντρικό πράκτορα.

**2. αρχιτεκτονική μαυροπίνακα**, όπου σε συγκεκριμένο χώρο αποθηκεύονται τα υποπροβλήματα που χρήζουν επίλυσης και οι λύσεις-αποτελέσματα. Όλοι οι πράκτορες επισκέπτονται αυτόν το συγκεκριμένο χώρο (μαυροπίνακα), είτε αφού ελεγχθούνε και συντονισθούν από ένα τρίτο (scheduling agent),



είτε αφού εκ των προτέρων έχει υπάρξει καθορισμός των αρμοδιοτήτων του καθενός.

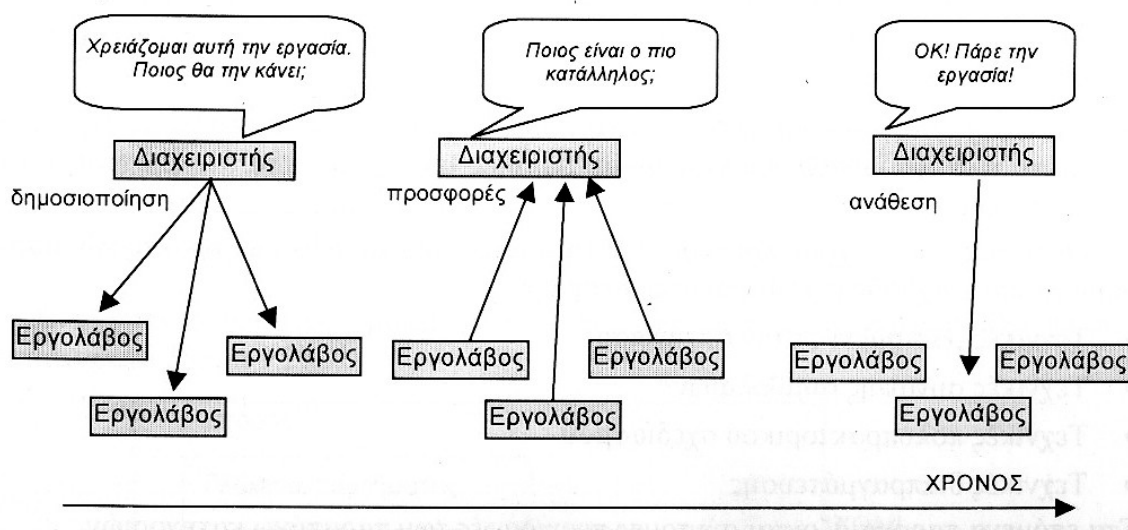
## 5.2. Τεχνικές σύναψης συμβολαίων (contracting net protocol)

Ένας πράκτορας αρχικά μετατρέπεται σε διαχειριστή (manager) και χωρίζει το κεντρικό πρόβλημα σε υποπροβλήματα, αναλαμβάνοντας να τα αναθέσει σε άλλους, τους εργολάβους (contractors). Επιπλέον, επιβλέπει την πορεία της λύσης και το ρόλο του κάθε εργολάβου (contractor), στην επίλυση του υποπροβλήματος που του έχει ανατεθεί. Ταυτόχρονα, κάποιος από τους εργολάβους μπορεί να χωρίσει το υποπρόβλημα που του έχει ανατεθεί σε περισσότερα και να τα αναθέσει σε άλλους πράκτορες. Έτσι, πλέον αυτός μετατρέπεται σε εργολάβο του πρώτου πράκτορα και διαχειριστή των τελευταίων.

Η διαδικασία ανάθεσης των υποπροβλημάτων έχει ως εξής:

1. δημοσιοποίηση των υποπροβλημάτων (announcement), από το διαχειριστή σε όλους ή σε μερικούς από τους πράκτορες
2. αξιολόγηση από τους αποδέκτες, κάθε υποπροβλήματος και ταυτόχρονη εξέταση του αν είναι ικανοί να αναλάβουν τη λύση του υποπροβλήματος, και αποστολή προσφοράς (bids) ή απορρίψης (declination) προς το διαχειριστή
3. συλλογή από το διαχειριστή των προσφορών (bids), αξιολόγησή τους και ανάθεση (awards) του ρόλου του εργολάβου στον καταλληλότερο πράκτορα

**Σχήμα 42: Λειτουργία της τεχνικής σύναψης συμβολαίων**



Πηγή: Vlachavas I. κ.α., 2002

Από την παραπάνω διαδικασία δημιουργείται ένα δίκτυο με συμβόλαια μεταξύ

των πρακτόρων που συμμετέχουν (contract net). Το σημαντικό είναι ότι κάθε πράκτορας μπορεί να είναι ταυτόχρονα διαχειριστής και εργολάβος.

### **5.3. Πολυπρακτορικός σχεδιασμός (multi-agent planning)**

Σε τέτοια μοντέλα ο κάθε πράκτορας συντάσσει πλάνο ενεργειών για την επίλυση του προβλήματος που είναι υπεύθυνος. Κατά τη διάρκεια της εκτέλεσης το πλάνο αναθεωρείται σύμφωνα με τα νέα στοιχεία και αποτελέσματα.

Στον κεντρικό πολυπρακτορικό σχεδιασμό (centralized multi-agent planning), ένας κεντρικός πράκτορας αναλαμβάνει τη συλλογή των επιμέρους πλάνων (partial or local plans) από τους υπόλοιπους, τα συνδυάζει σε ένα πλάνο και επιλύει τυχόν συγκρούσεις που θα εμφανιστούν σε αυτό.

Στον κατανεμημένο πολυπρακτορικό σχεδιασμό (distributed multi-agent planning), οι πράκτορες επικοινωνούν μεταξύ τους για τη δημιουργία των πλάνων τους και την επίλυση τυχόν συγκρούσεων που προκύπτουν.

### **5.4. Διαπραγμάτευση (negotiation)**

Η διαπραγμάτευση είναι ο τρόπος επίτευξης συμφωνίας μεταξύ δύο πρακτόρων ώστε να ικανοποιήσουν και οι δύο τους στόχους τους αλλά και τους υψηλότερους στόχους που η συνεργασία τους μπορεί μόνο να υλοποιήσει. Η διαπραγμάτευση, σε όλα τα μοντέλα συνεργασίας, αποτελεί το βασικό θέμα. Για την υλοποίηση αυτής της συνθήκης έχουν προταθεί μέθοδοι που βασίζονται:

- ❖ στη θεωρία παιχνιδιών (game theory based negotiation)
- ❖ στη δημιουργία πλάνων
- ❖ στην κατανεμημένη ικανοποίηση περιορισμών (distributed constraint satisfaction)

## ΚΕΦΑΛΑΙΟ Δ

### ΕΦΑΡΜΟΓΕΣ ΠΡΑΚΤΟΡΩΝ

i. ΕΦΑΡΜΟΓΕΣ – ΤΥΠΟΙ ΣΥΣΤΗΜΑΤΩΝ .....	91
ii. ΠΕΔΙΑ ΕΦΑΡΜΟΓΗΣ .....	93
Βιομηχανικές εφαρμογές .....	93
Εμπορικές εφαρμογές .....	95
Ιατρικές εφαρμογές .....	97
Ψυχαγωγικές εφαρμογές .....	98

## 1. Εισαγωγή

Οι εφαρμογές με τη χρήση πρακτόρων υιοθετούνται για το λόγο ότι βοηθούν στην επίλυση:

- ❖ προβλημάτων που θεωρούνταν πιο πριν αρκετά δύσκολο να επιλυθούν
- ❖ προβλημάτων για τα οποία υπήρχε λύση, που όμως με τη χρήση των πρακτόρων, ο τρόπος επίλυσής τους γίνεται φθηνότερος, ευκολότερος, φυσικότερος, περισσότερο επιτυχημένος και γρήγορος

Αξίζει να σημειωθεί ότι οι εφαρμογές αυτές είναι συστήματα που αποτελούνται από έναν ή περισσότερους πράκτορες όμοιους ή ετερογενείς μεταξύ τους.

Η επίλυση προβλημάτων που θεωρούνταν αρκετά δύσκολο να επιλυθούν πιο πριν με τη χρήση πρακτόρων, οδήγησε στη δημιουργία των **ανοικτών συστημάτων**, των **πολύπλοκων συστημάτων** και των **πλήρως διαδραστικών συστημάτων με συνεχή βοήθεια προς το χρήστη** (συστήματα βοηθοί).

Τα ανοικτά συστήματα είναι εκείνα των οποίων το περιβάλλον και η δομή τους είναι δυναμικά μεταβαλλόμενα. Τα τμήματα που τα αποτελούν δεν είναι γνωστά εξ αρχής, υπάρχει ετερογένεια μεταξύ τους και είναι χρονικά μεταβαλλόμενα. Τέτοια, είναι διάφορα υπολογιστικά συστήματα, που συνδέονται άμεσα με το Internet, μιας και αυτό αποτελεί τη βασική τους πληροφοριακή πηγή. Έτσι, κάθε τέτοιο σύστημα πρέπει να λειτουργεί, χωρίς καθοδήγηση από τους χρήστες και να συνεργάζεται με πολλούς διαφορετικούς οργανισμούς, πληροφοριακά συστήματα και βάσεις δεδομένων. Για την επίτευξη των στόχων τους, τα αυτά συστήματα, διαθέτουν τεχνικές διαπραγμάτευσης και συνεργασίας μεταξύ των μερών τους (συστήματα πολλαπλών πρακτόρων).

Τα πολύπλοκα συστήματα είναι γραμμικά και συνθετικά. Αυτό σημαίνει ότι για την επίλυση ενός προβλήματος υψηλού επιπέδου, υπάρχει αρχικά διαμοιρασμός του στις συνισταμένες του. Κάθε τμήμα ανατίθεται σε συγκεκριμένο κομμάτι του συστήματος, με σκοπό την επίλυσή του. Οι επιμέρους λύσεις συντίθεται από τους μηχανισμούς του πολύπλοκου συστήματος και έτσι παρουσιάζεται η τελική λύση. Κάθε τέτοιο σύστημα αποτελεί μια κοινωνία συνεργαζόμενων αυτόνομων μηχανισμών επίλυσης προβλημάτων.

Τα συστήματα βοηθοί ενεργούν σαν έμπειροι βοηθοί για κάποια συγκεκριμένα προβλήματα. Τα συστήματα αυτά, όντας γνώστες των χαρακτηριστικών των εφαρμογών που συνεργάζονται και των χρηστών που τις χρησιμοποιούν, αποτελούν

ικανούς συνεργάτες των τελευταίων στην επίτευξη των στόχων τους, μέσα από τις εφαρμογές. Τα χαρακτηριστικά των τμημάτων τέτοιων εφαρμογών-συστημάτων είναι: η αυτονομία στη δράση, η προνοητικότητα και ικανότητα πρόβλεψης, η άμεση ανταπόκριση στα αιτήματα και η προσαρμοστικότητα.

Η επίλυση παλιών προβλημάτων με την υλοποίηση εφαρμογών που βασίζονται στους πράκτορες, βελτιστοποιήθηκε και οδήγησε:

- ❖ στη καλύτερη διαχείριση, έλεγχο και καλύτερη ανάκτηση δεδομένων καθώς και στην υιοθέτηση στις εφαρμογές κατανεμημένων πληροφοριακών πηγών. Η κατανομή της επεξεργασίας συνοδεύεται από την ταυτόχρονη κατανομή των πληροφοριών στους διάφορους εμπλεκόμενους
- ❖ στο φιλτράρισμα της πληροφορίας, στην άμεση και εύκολη επικοινωνία των εφαρμογών με άλλες εφαρμογές και στην εξειδίκευση στις απαιτήσεις του χρήστη
- ❖ στη συνεχή ενημέρωση των πληροφοριακών συστημάτων και στη διατήρηση επίκαιρου χαρακτήρα σε αυτά, με τα οποία συνδέονται οι εφαρμογές

Παρόλα, τα θετικά που προσδίδει η ενσωμάτωση των τεχνολογιών των πρακτόρων σε νέες και παλιές εφαρμογές, υπάρχουν οι εξής προβληματισμοί.

Πολλές εφαρμογές δε μπορούν να υιοθετήσουν πράκτορες. Αυτό συμβαίνει όταν οι εφαρμογές σχετίζονται με συστήματα που:

- α) απαιτούν καθολικές παραμέτρους, άμεση και on real time απάντηση
- β) πρέπει να μένουν ανεπηρέαστα από αδιέξοδες καταστάσεις

Έτσι, οι πράκτορες δε μπορούν να χρησιμοποιηθούν σε όλες τις εφαρμογές. Βασική αιτία αποτελεί το γεγονός ότι δεν έχουν συνολική αντίληψη των πραγμάτων.

Πιο συγκεκριμένα, οι ενέργειες ενός πράκτορα εξαρτώνται από την εσωτερική του κατάσταση. Έτσι δε διαθέτει πάντα, τη δυνατότητα να επιτυγχάνει καθολικά αποδεκτές λύσεις. Το τελευταίο οφείλεται στο γεγονός, ότι γνωρίζει μόνο αυτά που τον ενδιαφέρουν από το γύρω του περιβάλλον και όχι τη συνολική πληροφορία που το περιβάλλον περιγράφει.

Τέλος, οι χρήστες, οι εταιρίες και όσοι είναι το target group των εφαρμογών πρακτορικών συστημάτων, πρέπει να αποκτήσουν εμπιστοσύνη απέναντι σε αυτούς. Από την άλλη, οι πράκτορες πρέπει να πείσουν τους προηγούμενους, για την ασφάλεια που προσφέρουν στις εργασίες που αναλαμβάνουν να φέρουν εις πέρας,

μέσω των νέων εφαρμογών.

## 2. Πεδία εφαρμογής συστημάτων πρακτόρων

Οι εφαρμογές πρακτόρων χωρίζονται αρχικά σύμφωνα με:

- ❖ τη συμπεριφορά των πρακτόρων που τις αποτελούν, οπότε μπορεί να είναι:
  - απλά σύνολα εντολών βασισμένα σε προκαθορισμένες υποθέσεις και κανόνες
  - παροχείς υπηρεσιών με λειτουργίες υψηλού επιπέδου, οι οποίες βασίζονται σε καλά ορισμένες και καθορισμένες εντολές
  - συστήματα με αυτόνομη συμπεριφορά
- ❖ το ρόλο των πρακτόρων, δηλαδή τι τελικά καλούνται να επιτελέσουν, οπότε είναι:
  - βοηθοί για υποστήριξη της διαδικασίας αποφάσεων
  - αυτόνομα συστήματα επίλυσης προβλημάτων
- ❖ τον τρόπο επίλυσης των προβλημάτων, οπότε υπάρχουν οι εφαρμογές:
  - πολλαπλών πρακτόρων
  - απλών πρακτόρων

### 2.1. Βιομηχανικές εφαρμογές

Οι πρώτες τέτοιου είδους εφαρμογές δημιουργήθηκαν το 1987, με πρώτη αυτή του Panagak, που ήταν σχετική με τον έλεγχο της παραγωγής.

Οι εφαρμογές που αντιμετωπίζουν το πρόβλημα του ελέγχου των διαδικασιών της γραμμής παραγωγής εντάσσονται στις βιομηχανικές. Τέτοιες εφαρμογές είναι κυρίως πολυπρακτορικές, μιας και κάθε διαδικασία της γραμμής παραγωγής αποτελεί ένα αυτόνομο αντιδραστικό σύστημα. Αρκετά γνωστή είναι η **ARCHON**, η οποία διαθέτει δυνατότητες υλοποίησης πολυπρακτορικών συστημάτων και συγκεκριμενοποιημένη μεθοδολογία (Jennings 1995). Με τη χρήση της ARCHON, έχουν υλοποιηθεί αρκετές εφαρμογές ελέγχου της γραμμής διαδικασιών (παραγωγή και μεταφορά ηλεκτρισμού, παραγωγή και προώθηση μηχανικών τμημάτων σε εργοστάσιο αυτοκινήτων, κ.α).

Οι πράκτορες στην ARCHON είναι βαριά υπολογιστικά συστήματα με τέσσερα βασικά συστατικά:

- το module επικοινωνίας σε υψηλό επίπεδο (High Level Communication

- Module), που διαχειρίζεται την επικοινωνία και αλληλεπίδραση των πρακτόρων
- το module σχεδιασμού και προσανατολισμού (Planning Coordination Module), που είναι υπεύθυνο για το τι τελικά θα πράξει ο πράκτορας
  - το module διαχείρισης της πληροφορίας (Agent Information Management Module), το οποίο είναι υπεύθυνο για τη διατήρηση και ενημέρωση του μοντέλου του κόσμου, στην εσωτερική κατάσταση του κάθε πράκτορα
  - και τέλος, το σύστημα ευφυΐας (Intelligent System), που περιγράφει την εξειδίκευση και ικανότητα έξυπνης συμπεριφοράς του πράκτορα σε αυτά που δέχεται

Ο Panurak το 1987 περιέγραψε μια εφαρμογή με την ονομασία **YAMS** (Yet Another Manufacturing System), η οποία παρουσιάζει το Πρωτόκολλο συμφωνίας δικτύου (Contract Net Protocol – Smith, 1980) και είναι χρήσιμη στον έλεγχο της παραγωγής. Το πρωτόκολλο συμφωνίας δικτύου δημιουργεί και περιγράφει τις επιμέρους γραμμές παραγωγής και την ιεράρχησή τους.

Τέτοιες εφαρμογές αποτελούν τη δεύτερη κατηγορία πρακτορικών εφαρμογών στο πεδίο της βιομηχανίας. Κάθε εταιρία που παράγει κάποιο προϊόν μοντελοποιείται σε ένα σύνολο εργασιών (πχ άλεση, σοβαντισμα, τόννευση, βάψιμο κ.α.). Αυτά τα σύνολα εργασιών διαιρούνται σε ευέλικτα συστήματα παραγωγής (Flexible Manufacturing Systems), όπου καθένα επιτελεί εργασίες όπως: συναρμολόγηση, βάψιμο, ιεράρχηση προϊόντων κ.α. Σύνολο των FMS αποτελούν ένα εργοστάσιο. Μια εταιρία μπορεί να έχει πολλά εργοστάσια τα οποία συνεργάζονται για να παράξουν το σύνολο της απαιτούμενης παραγωγής. Το YAMS προσπαθεί να συντονίσει και να διαχειριστεί τη διαδικασία της παραγωγής με στόχο την έγκαιρη και χωρίς μεγάλο κόστος δημιουργία και παράδοση των τελικών προϊόντων. Στο YAMS που αποτελεί πολυπρακτορικό σύστημα, κάθε εργοστάσιο και τμήμα του εργοστασίου είναι πράκτορας. Κάθε πράκτορας έχει δικά του πλάνα δράσης, και διαπραγματεύεται με τους άλλους, για την εύρεση της βέλτιστης λύσης στη συμπεριφορά του και στην επίτευξη του κοινού στόχου, που είναι η βελτιστοποίηση των διαδικασιών της παραγωγής.

Ο Kinny το 1996, περιέγραψε το σύστημα OASIS. Αυτό αποτελεί την τρίτη κατηγορία βιομηχανικών πολυπρακτορικών συστημάτων. Είναι ένα σύστημα ελέγχου της εναέριας κυκλοφορίας του αεροδρομίου του Σίδνευ της Αυστραλίας. Στο **OASIS**

τα αεροσκάφη και τα διάφορα συστήματα ελέγχου αντιπροσωπεύονται από πράκτορες. Οι πράκτορες του αναπτύχθηκαν βάσει του μοντέλου BDI και το σύστημα δοκιμάζεται σε πραγματικές συνθήκες.

Σε κάθε αεροσκάφος που μπαίνει στην επιχειρησιακή ζώνη του αεροδρομίου, ανατίθεται ένας πράκτορας, που αποκτά τους στόχους και γνωρίζει όλες τις πληροφορίες που αφορούν το αεροσκάφος, όπως για παράδειγμα σε ποιο αεροδρόμιο σκοπεύει να προσγειωθεί, ποιος είναι ο τύπος του, κλπ.

Οι πράκτορες που αντιστοιχούν στα συστήματα ελέγχου εναέριας κυκλοφορίας είναι υπεύθυνοι για το συντονισμό και τη διαχείριση ολόκληρου του συστήματος.

## 2.2. Εμπορικές εφαρμογές

Τέτοιου είδους εφαρμογές έρχονται να καλύψουν αρχικά την ανάγκη για καλύτερη διαχείριση και διάθεσή της υπάρχουσας πληροφορίας. Στις μέρες μας, γεγονός είναι η υπερφόρτωση των υπολογιστικών συστημάτων και του παγκόσμιου ιστού με πάρα πολλές πληροφορίες, τις οποίες μπορούμε πολύ δύσκολα να διαχειριστούμε. Σε αυτό συμβάλλει πολλές φορές η αδιαφορία του χρήστη κατά την αναζήτηση και η λανθασμένη οργανωτική δομή των ιστοσελίδων. Έτσι, δε γίνεται ικανοποιητικά το φιλτράρισμα και η συλλογή της πληροφορίας, που είναι χρήσιμη κατά τη διαδικασία της αναζήτησης. Όλα τα παραπάνω, θα αντιμετωπιζόνταν εάν στον έλεγχο και την οργάνωση μιας αναζήτησης, πρωταρχικό ρόλο έπαιζε κάποιος πράκτορας, που θα συνεργάζονταν με το χρήστη και όχι ο ίδιος ο χρήστης.

Η Maes το 1994, εισήγαγε ένα πράκτορα φιλτραρίσματος στα μηνύματα ηλεκτρονικού ταχυδρομείου, τον οποίο ονόμασε **Maxims**. Το συγκεκριμένο πρόγραμμα μάθαινε και επιτύγχανε ιεράρχηση, διαγραφή, προώθηση, ταξινόμηση και αρχειοθέτηση των μηνυμάτων του ηλεκτρονικού ταχυδρομείου ενός χρήστη. Στην ουσία το Maxims παρακολουθεί το χρήστη και τι ενέργειες κάνει σε κάθε μήνυμα και προσπαθεί να προβλέψει τη μελλοντική του συμπεριφορά (του χρήστη) απέναντι σε κάποιο άλλο μήνυμα. Όταν η πρόβλεψη του Maxims είναι λάθος, αυτή κατακρατείται στη μνήμη του. Μόλις, όμως καταφέρνει να προβλέπει τις μελλοντικές ενέργειες του χρήστη, τότε του κάνει προτάσεις πριν ακόμη αυτός σκεφτεί πως θα ενεργήσει.

Την ίδια χρονιά επίσης, η ίδια ερευνήτρια, περιγράφει ένα πρόγραμμα φιλτραρίσματος νέων (news) στο Internet που ονομάζεται **Newt**. Στην ουσία αυτό το πρόγραμμα που αναπτύχθηκε σε πλατφόρμα Unix και σε C++, παίρνει σαν είσοδο ένα κείμενο από την υπηρεσία Usenet και δίδει σαν έξοδο κομμάτια αυτού που



χρίζουν προσοχής από το χρήστη. Ο χρήστης εκ των προτέρων, ανάλογα με τι πληροφορία θέλει ο ίδιος να ασχοληθεί, δίδει παραδείγματα άρθρων που τον ενδιαφέρουν ή όχι στο Newt και ύστερα αυτό κάνει φιλτράρισμα στο Usenet. Επιπλέον, η αναζήτηση και το φιλτράρισμα μπορούν να λειτουργήσουν, με την απόδοση στο σύστημα από τον χρήστη, κανόνων που περιγράφουν τις επιθυμίες του σε πληροφορία.

Ο Zuno το 1997, οργάνωσε μια **ψηφιακή βιβλιοθήκη** με οργανωμένα και διαχειριζόμενα δεδομένα και υπηρεσίες που βοηθούν τους χρήστες. Αυτό είναι ένα πολυπρακτορικό σύστημα που οργανώνει διασκορπισμένα δεδομένα, ανάλογα το αίτημα του χρήστη. Οι πράκτορες χωρίζονται:

- στους *καταναλωτές*, οι οποίοι αντιπροσωπεύουν τους χρήστες και τα αιτήματα τους για πληροφορία (διατηρούν τα μοντέλα που δίνουν οι χρηστές, και τα χρησιμοποιούν για να βρискουν την πληροφορία που χρειάζονται)
- στους *παραγωγούς*, οι οποίοι αντιπροσωπεύουν τους παροχείς περιεχομένου, γνώσης και πληροφοριών
- στους *διαμεσολαβητές*, οι οποίοι οργανώνουν την επικοινωνία μεταξύ καταναλωτών και παραγωγών

Άλλες εμπορικές εφαρμογές είναι αυτές που σχετίζονται με το ηλεκτρονικό εμπόριο. Έτσι, κάποιες επιλογές εμπορικής φύσης μπορούν να υλοποιούνται από πράκτορες, αφού πρωτίστως οι χρήστες έχουν καθορίσει σε αυτούς τις επιθυμίες τους. Εφαρμογή του παραπάνω κανόνα αποτελούν πολλές από τις διαπραγματεύσεις στην ηλεκτρονική ανταλλαγή μεριδίων και μετοχών. Επιπρόσθετα, σήμερα υπάρχουν πολλές ιστοσελίδες με δημοπρασίες, όπου ο χρήστης μπορεί να θέτει τις επιθυμίες και τα αιτήματά του και τελικά ο πράκτορας, που τα αναλαμβάνει, να του αγοράζει διάφορα προϊόντα που πληρούν τις αρχικές προϋποθέσεις. Οι Chavez και Maes το 1996, δημιούργησαν μια αγορά από πράκτορες που πουλούσαν και αγόραζαν, κατά την αλληλεπίδρασή τους μέσα σε αυτή. Η εφαρμογή ονομάστηκε **Kasbah**.

Τέλος, το τρίτο είδος εμπορικών εφαρμογών με τη χρήση πρακτόρων είναι αυτές που ασχολούνται με τη διοίκηση επιχειρήσεων. Η λήψη επιχειρηματικών αποφάσεων και η επιτυχής επιχειρηματική δραστηριότητα, εξαρτάται από την ανάκτηση πλήθους πληροφοριών, που πρέπει να είναι σωστά οργανωμένες. Η ανάκτηση σταθερών, επικαιροποιημένων και ορθών πληροφοριών από τη δομή μιας επιχείρησης, είναι

πολύ δύσκολο πράγμα. Γι' αυτό αναπτύχθηκαν συστήματα τεχνολογίας πληροφορικής που υποβοηθούν την ανάκτηση πληροφορίας και τη σωστότερη διαχείριση των επιχειρηματικών διαδικασιών. Ο Jennings το 1996 υλοποίησε το ADEPT, το οποίο βλέπει κάθε επιχειρηματική διαδικασία ως μια κοινότητα από πράκτορες, που διαπραγματεύονται και παρέχουν υπηρεσίες. Κάθε πράκτορας παίζει συγκεκριμένο ρόλο στο κάθε τμήμα της επιχείρησης και παρέχει συγκεκριμένες και καθορισμένες υπηρεσίες.

πχ. Πρόβλημα: Δημιουργία ενός τηλεπικοινωνιακού δικτύου

	<b>1<sup>ος</sup> πράκτορας</b>	<b>2<sup>ος</sup> πράκτορας</b>	<b>3<sup>ος</sup> πράκτορας</b>
<i>ρόλος-υπηρεσία</i>	η σχεδίαση ενός τηλεπικοινωνιακού δικτύου	ο έλεγχος του δικτύου αν πληροί τα νόμιμα	ο υπολογισμός του προϋπολογισμού του δικτύου
<i>ανήκει</i>	τμήμα σχεδίασης	νομικό τμήμα	τμήμα μάρκετινγκ

Κάθε πράκτορας, που απαιτεί υπηρεσίες από κάποιον άλλο πράκτορα, διαπραγματεύεται για αυτές, προσπαθώντας να επιτύχει αποδεκτή τιμή, μικρό χρόνο και υψηλή ποιότητα, για την κάθε υπηρεσία. Κάθε διαπραγμάτευση όταν είναι επιτυχημένη καταλήγει σε συμφωνία.

### 2.3. Ιατρικές εφαρμογές

Τον τελευταίο καιρό έχουν υπάρξει εφαρμογές, που διαθέτουν πράκτορες και έχουν στόχο την καλύτερη παρακολούθηση των ασθενών και διαχείριση των ιατρικών δεδομένων.

Οι Hayes-Roth το 1989, ανέπτυξαν το **Guardian**, ένα σύστημα που σκοπό είχε τη διαχείριση της φροντίδας-παρακολούθησης των ασθενών στη μονάδα εντατικής θεραπείας χειρουργικής κλινικής. Η φροντίδα των ασθενών είναι υπόθεση πολλών ατόμων (γιατρών, νοσοκόμων κ.α.), οι οποίοι χρειάζεται να έχουν όλη τη διαθέσιμη-χρήσιμη πληροφόρηση για τους ασθενείς, ώστε να τους παρέχουν υψηλής ποιότητας υπηρεσίες. Έτσι, αναπτύχθηκαν τρία είδη πρακτόρων:

- οι *αντιληπτικοί-ενεργητικοί*, με ευκολία στη διαχείριση των αιτημάτων των χρηστών
- οι *λογικοί*, οι οποίοι τελικά κατάστρωναν τη συμπεριφορά του συστήματος και οδηγούσαν στη λήψη αποφάσεων από αυτό
- και οι *πράκτορες ελέγχου*

Όλοι οι πράκτορες οργανώθηκαν σε ιεραρχίες και κινούνταν σε ένα σύστημα μαύρου πίνακα, συνεργαζόμενοι και ανταλλάσσοντας πληροφορίες.

Ο Huang το 1996 πρότεινε ένα πρακτορικό καταναμημένο σύστημα, που θα επιτύγχανε την ενοποίηση των διαδικασιών και των δεδομένων που σχετίζονται με ένα ασθενή. Αυτό δέχεται την διάγνωση ενός γιατρού και σχεδιάζει πολιτική αντιμετώπισης της ασθένειας. Οι πράκτορες περιλαμβάνουν:

- μια βάση γνώσης
- ένα μηχανισμό διαχείρισης της επικοινωνίας και των μηνυμάτων άλλων πρακτόρων
- ικανότητα σύνθεσης και λογικής επεξεργασίας των πληροφοριών
- ένα ολοκληρωμένο περιβάλλον διεπαφής με το χρήστη, που επιτρέπει στον τελευταίο να προσθέτει και να διαγράφει στοιχεία καθώς και να παρακολουθεί τους στόχους του συστήματος

#### **2.4. Ψυχαγωγικές εφαρμογές**

Η βιομηχανία της ψυχαγωγίας δε λαμβάνεται συχνά υπόψη από την κοινωνία της επιστήμης των υπολογιστών. Έτσι, εφαρμογές παιχνιδιών, διαδραστικού κινηματογράφου και θεάτρου, θεωρούνται περιφερειακές όσον αφορά την έρευνα. Παρόλα αυτά, οι πράκτορες εισέρχονται και αποτελούν σημαντικά κομμάτια πολλών τέτοιων ψυχαγωγικών εφαρμογών, που βασίζονται σε ημιαυτόνομους χαρακτήρες με λογική περίπου ίδια με την ανθρώπινη.

Η βιομηχανία βίντεο παιχνιδιών είναι η πρώτη που υιοθέτησε πράκτορες. Ο Wavish το 1996 δημιούργησε ένα Tetris όπου ένας πράκτορας αναλάμβανε το ρόλο του παίκτη και προσπαθούσε να ταιριάξει τα κομμάτια, έτσι ώστε να πετύχει το μεγαλύτερο δυνατό σκορ και να φτάσει στο υψηλότερο δυνατό επίπεδο (πίστα). Το πρόγραμμα δεν περιελάμβανε παραδοσιακές συμβολικές τεχνικές της Τεχνητής Νοημοσύνης με τη χρήση κανόνων, αλλά ένα αντιδραστικό μοντέλο το οποίο ονομάζονταν **Real Time ABLE**. Στο προγραμματιστικό κομμάτι του πράκτορα, το βασικό συστατικό ήταν ο καθορισμός συνόλου συμπεριφορών που ανάλογα το ερέθισμα που είχε κάποιος, ενεργοποιούνταν χωρίς να απαιτείται πολύπλοκη συμβολική λογική.

Επιπρόσθετα, σε πολλά σύγχρονα παιχνίδια τύπου **Role Playing Games** χρησιμοποιούνται πράκτορες. Οι χαρακτήρες ανάλογα με τις επιλογές που κάνει ο χρήστης, αντιμετωπίζουν συχνά διαφορετικές καταστάσεις στο ίδιο κομμάτι της πίστας. Έτσι, το ίδιο περιβάλλον σε δύο διαφορετικές χρονικές στιγμές αποτελείται και διέπεται από διαφορετικές συνθήκες. Δηλαδή, οι συνθήκες κάθε φορά στο περιβάλλον είναι αποτέλεσμα της προηγούμενης συμπεριφοράς των χαρακτήρων του

παιχνιδιού.

Στην ανάπτυξη διαδραστικού κινηματογράφου και θεάτρου ο ρόλος των πρακτόρων είναι σημαντικός και πρωτεύοντας. Ένα σύστημα διαδραστικού κινηματογράφου και θεάτρου επιτρέπει στο χρήστη, να παίζει ρόλο ανάλογο με τους ρόλους των ανθρώπινων οντοτήτων στην εξέλιξη της ιστορίας, σε ένα φιλμ ή μια θεατρική παράσταση. Ο χρήστης μπορεί ηλεκτρονικά να αναλαμβάνει το ρόλο κάποιου ηθοποιού και ανάλογα με τη συμπεριφορά του να καθορίζεται η πλοκή και η εξέλιξη του σεναρίου.

## 2.5. Άλλες εφαρμογές

Διάφορες άλλες εφαρμογές και περιβάλλοντα ανάπτυξης πρακτορικών εφαρμογών έχουν προταθεί από διάφορους ερευνητές όπως:

**Georgeff & Rao:** που αναφέρθηκαν στους πράκτορες εσωτερικής κατάστασης - BDI agents

**Burmeister και άλλοι:** οι οποίοι αναφέρθηκαν στην υλοποίηση τεχνικών για τη διαχείριση της πληροφορίας που σχετίζεται με τη βιομηχανική παραγωγή. Τέτοια παραδείγματα είναι: **α.** το **COSY**, του οποίου η αρχιτεκτονική βασίζεται σε πράκτορες με ιδέες, συμπεριφορές, προθέσεις και πηγές πληροφόρησης και **β.** το **DASEDIS**.

**Haugender & Steiner:** οι οποίοι δημιούργησαν τη MAIL, μια γλώσσα ανάπτυξης και επικοινωνίας σε πολυπρακτορικά συστήματα. Αυτή υλοποιεί απλούς πράκτορες που συνεργάζονται και αλληλεπιδρούν μεταξύ τους

**Weihmayer & Velthuijsen:** οι οποίοι ασχολήθηκαν με την ενσωμάτωση των πρακτόρων στις τηλεπικοινωνίες (έλεγχος δικτύων, διαχείριση υπηρεσιών, μεταγωγή, σχεδιασμός και διαχείριση δικτύων κ.α.)

**Huhns & Singh:** που πρώτοι περιέγραψαν τον τρόπο, ώστε μια ομάδα αυτόνομων πρακτόρων, να μπορεί να επιτελέσει σωστά τη διαχείριση διαφόρων ουρών συναλλαγής. Εστίασαν αρκετά στη δυνατότητα διασυνδεσιμότητας, μεταξύ των ετερογενών πρακτορικών συστημάτων και την εξήγησαν με την ύπαρξη σε κάθε πράκτορα μοντέλων πληροφοριών και πηγών πληροφορίας.

**Plu:** ο οποίος παρουσίασε πρακτορικά συστήματα οργανισμών, που μπορούν να σχεδιάζουν και να υλοποιούν υπηρεσίες παροχής πληροφοριών και εμπορικών συναλλαγών σε πολλούς χρήστες. Είναι ο πρώτος που αναφέρθηκε και σχεδίασε εφαρμογές με κινητούς πράκτορες.

**Sycara και άλλοι:** οι οποίοι ασχολήθηκαν με τη σχεδίαση και υλοποίηση

πρακτόρων, που επιτελούν ενέργειες φιλτραρίσματος, αξιοποίησης, βαθμολόγησης και ενοποίησης της πληροφορίας που βρίσκεται στο Internet. Εφαρμογές τέτοιου περιεχομένου υπάρχουν πολλές κυρίως στο τομέα της διαχείριση χαρτοφυλακίου μετοχών.

**Goldberg & Senator:** αυτοί ανέπτυξαν το **FAIS**, το οποίο εξέταζε και έκανε ανάλυση σε μεγάλο πλήθος οικονομικών συναλλαγών για τον εντοπισμό «ξεπλύματος» χρήματος. Το σύστημα εμπεριείχε πράκτορες λογισμικού και ανακάλυπτε γνώση αφού έκανε ανάλυση πληροφοριών σε πολλούς και μεγάλους χώρους δεδομένων. Η συνεργασία μεταξύ των πρακτόρων έφερνε τα απαραίτητα αποτελέσματα.

**Wenger & Probst:** αυτοί ασχολήθηκαν με πρακτορικές εφαρμογές για την παροχή οικονομικών υπηρεσιών όπως: η διαχείριση χαρτοφυλακίου, οι πωλήσεις συμβολαίων, η συνεργασιακή οικονομική διαχείριση, η αξιολόγηση επενδύσεων κ.α.

## **ΒΙΒΛΙΟΓΡΑΦΙΑ**

Bradshaw J. M., *Software Agents*

Finin, Labrou, Mayfield, KQML as an agent communication language

Jennings N. R. and Wooldridge M., *Applications of Intelligent Agents*

Nwana H. S., *Software Agents: An Overview*, Knowledge Engineering Review, Vol. 11, No 3, pp. 205-244, October/November 1996.

Rao S. and Georgeff M. P., 1995, *Bdi agents: From theory to practice*. In Proceedings of the First International Conference on Multi-Agent Systems

Stone P., Veloso M., 1997, *Multiagent Systems: A survey from a Machine Learning Perspective*

Vlachavas I., Kefalas P., Bassiliades N., Refanidis I, Kokkoras F., Sakellariou H., 2002, *Artificial Intelligence*, Gartaganis Publs, Thessaloniki

Vozalis E., Nicolaou A., Margaritis K.G., *Intelligent Techniques for Web Applications: Review and Educational Application*

Wooldridge M., Jennings N. R., *Intelligent Agents: Theory and Practice*, Knowledge Engineering Review, January 1995.

## **Διαδικτυακές διευθύνσεις**

[agents.umbc.edu](http://agents.umbc.edu)

[www.agent-software.com/shared/home](http://www.agent-software.com/shared/home)

<http://www.csee.umbc.edu/~finin/talks/sisce97/sld001.htm>

[www-2.cs.cmu.edu/~softagents/](http://www-2.cs.cmu.edu/~softagents/)

**ΤΕΧΝΟΛΟΓΙΚΟ ΙΔΡΥΜΑ ΛΑΡΙΣΑΣ**  
**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ**  
**ΤΜΗΜΑ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

**ΕΥΦΥΗ ΣΥΣΤΗΜΑΤΑ ΛΟΓΙΣΜΙΚΟΥ**  
**(Εργαστήριο)**

**ΣΩΜΑΡΑΣ ΧΡΗΣΤΟΣ**

**ΙΟΥΝΙΟΣ 2004**

## **ΚΕΦΑΛΑΙΟ Ε**

### **ΛΟΓΙΣΜΙΚΟ ΥΛΟΠΟΙΗΣΗΣ ΠΡΑΚΤΟΡΙΚΩΝ ΕΦΑΡΜΟΓΩΝ**

**Το SWARM του SANTA FE INSTITUTE**

**Το JACK**

**Το ZEUS**



**Διαδικτυακές διευθύνσεις**

[www.agent-software.com/shared/home](http://www.agent-software.com/shared/home)

[www.cyber-robotics.com](http://www.cyber-robotics.com)

[www.swarm.org](http://www.swarm.org)

[www.agentland.com](http://www.agentland.com)

[www.compinfo-center.com/tpagnt-t.htm](http://www.compinfo-center.com/tpagnt-t.htm)