



Αριστοτέλειο
Πανεπιστήμιο
Θεσσαλονίκης

Τεχνητή Νοημοσύνη

Συστήματα Πρακτόρων

Ιώαννης Βλαχάβας

Τμήμα Πληροφορικής ΑΠΘ



Άδειες Χρήσης

Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons. Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα. Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.



Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.





Συστήματα Πρακτόρων (Agents)

Ευφυείς Πράκτορες (Intelligent Agents)

❖ Είναι ένας από τους πιο πρόσφατους και με μεγαλύτερο ενδιαφέρον κλάδους της ΤΝ.

Πράκτορας (agent): Μία οντότητα που αντιλαμβάνεται το περιβάλλον μέσα στο οποίο βρίσκεται με τη βοήθεια αισθητήρων (sensors), είναι μέρος του περιβάλλοντος αυτού, κάνει συλλογισμούς για το περιβάλλον και δρα πάνω σε αυτό με τη βοήθεια μηχανισμών δράσης (effectors), για την επίτευξη κάποιων στόχων.

❑ Αυτονομία

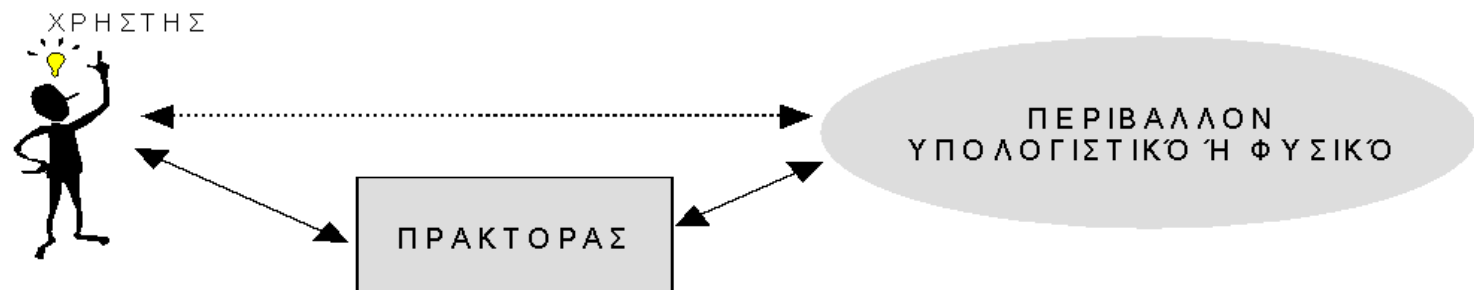
- Ο πιο κοινός παρονομαστής όλων των ειδών πρακτόρων.
- Υποχρεώνει την ύπαρξη "νοημοσύνης", τουλάχιστον σε κάποιο βαθμό (intelligent agents).

❖ Με την τεχνολογία αυτή αλλάζει η μορφή της διασύνδεσης χρήστη-λογισμικού.

- Ο χρήστης δεν επικοινωνεί απευθείας με κάποια εφαρμογή αλλά χρησιμοποιεί έναν πράκτορα ο οποίος τον διευκολύνει σε χρονοβόρες διαδικασίες, διαδικασίες ρουτίνας ή διαδικασίες που χρειάζονται κάποια ικανότητα που ο χρήστης δεν έχει αποκτήσει ακόμη (π.χ. σύνταξη επιστολής, αποστολή email, κλπ).

❖ Εφαρμογές

- ❑ Παροχή έξυπνων υπηρεσιών βοήθειας, αναζήτηση πληροφοριών στο διαδίκτυο, οργάνωση καθημερινού προγράμματος, κλπ.





Ευφυείς Πράκτορες ***(Intelligent Agents)***

Ταξινόμηση Πρακτόρων

❖ Βιολογικοί Πράκτορες

- ❑ Χρησιμοποιούν τις **αισθήσεις** τους για να αντιληφθούν το γύρω κόσμο, τις **γνώσεις** τους για να βγάλουν συμπεράσματα για αυτόν και τα **μέρη του σώματος** τους για να εφαρμόσουν τις ενέργειες που προκύπτουν από τη συλλογιστική τους

❖ Τεχνητοί Πράκτορες

❑ Ρομποτικοί πράκτορες (robotic agents ή **robots**)

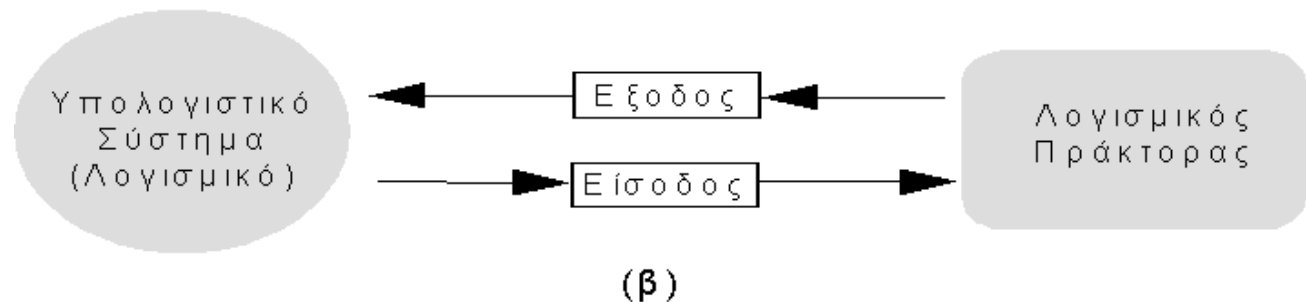
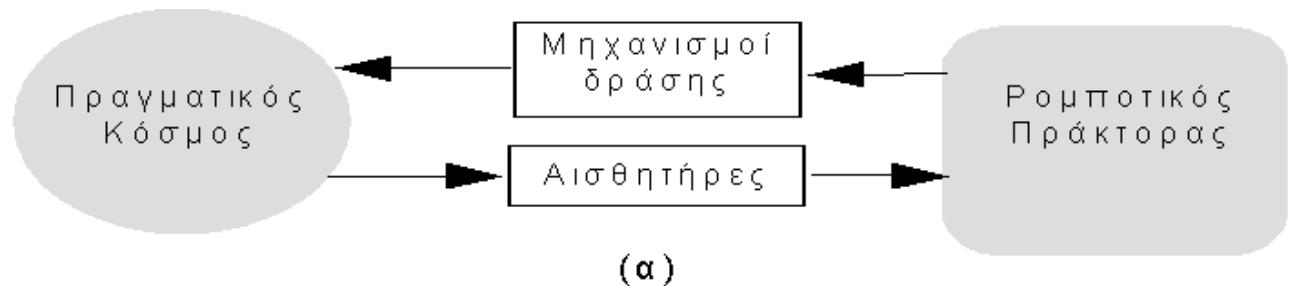
- Έχουν σαν αισθητήρες και μηχανισμούς δράσης μηχανικά ή ηλεκτρονικά μέρη και δρουν στον πραγματικό κόσμο

❑ Λογισμικοί πράκτορες (software agents ή **softbots**)

- Προγράμματα που δρουν σε ένα υπολογιστικό σύστημα

❖ Συλλογιστική διαδικασία πρακτόρων (reasoning)

- ❑ Επεξεργάζονται τα ερεθίσματα του περιβάλλοντος τους, και εφαρμόζουν τα αποτελέσματα της συλλογιστικής στο περιβάλλον.



Σύγχρονες Θεωρήσεις Πρακτόρων

- ❖ Οι πράκτορες αποτελούν κοινό πεδίο ενδιαφέροντος για πολλές περιοχές της επιστήμης υπολογιστών. Μπορούν να εξεταστούν από διαφορετικές σκοπιές:
 - ❑ Στην TN θεωρούνται ευφυείς οντότητες,
 - ❑ Στον αντικειμενοστραφή προγραμματισμό θεωρούνται οντότητες (αντικείμενα) που επικοινωνούν μεταξύ τους.
 - ❑ Στον παράλληλο προγραμματισμό είναι οντότητες που εκτελούνται παράλληλα και
 - ❑ Στο πεδίο διασύνδεσης ανθρώπου-μηχανής είναι οντότητες που παίρνουν πρωτοβουλία σε συνεργασία με το χρήστη για να επιτελέσουν τα καθήκοντά τους.
- ❖ Υπάρχουν δύο σύγχρονες θεωρήσεις για τους πράκτορες:
 - ❑ **Χαλαρή θεώρηση** (weak notion of agency): Οι πράκτορες πρέπει να είναι αυτοπροσδιοριζόμενοι (self-contained) και να εκτελούνται ταυτόχρονα με κάποιους άλλους (concurrently executing).
 - Η άποψη αυτή διευρύνει την έννοια πράκτορας θεωρώντας ότι οι πράκτορες δεν πρέπει απαραίτητα να είναι "ευφυείς".
 - ❑ **Ισχυρή θεώρηση** (strong notion of agency): Οι πράκτορες πρέπει να έχουν **γνώση** (knowledge), **πεποιθήσεις** (beliefs), **επιθυμίες** (desires), **προθέσεις** (intentions) και **υποχρεώσεις** (obligations). (BDI agents)
- ❖ Η λέξη "πράκτορας" χρησιμοποιείται πολλές φορές αδικαιολόγητα για πολλά από τα συστήματα λογισμικού που αναπτύσσονται.

Προσεγγίσεις Στην Περιγραφή Συστημάτων

- ❖ Γνώση (knowledge), πεποιθήσεις (beliefs), επιθυμίες (desires), προθέσεις (intentions) και υποχρεώσεις (obligations) για συστήματα Λογισμικού?
- ❖ Τρεις προσεγγίσεις για την περιγραφή ενός συστήματος (Daniel Dennet):
 - ❑ η φυσική προσέγγιση (physical stance), η οποία βασίζεται στη γνώση νόμων των θετικών επιστημών,
 - ❑ η σχεδιαστική προσέγγιση (design stance) όπου η περιγραφή γίνεται βάσει του σκοπού για τον οποίο σχεδιάστηκε το σύστημα,
 - ❑ η προθεσιαρχική προσέγγιση (intentional stance) που προβλέπει τη συμπεριφορά του συστήματος θεωρώντας το σαν ένα έλλογο πράκτορα (rational agent) ο οποίος δρα βάσει των "νοητικών" του καταστάσεων (mental states), δηλαδή πεποιθήσεων, επιθυμιών, προθέσεων κλπ.
 - Η τελευταία βρήκε μεγάλη απήχηση στην περιγραφή συστημάτων πρακτόρων, όπως θα δούμε στη συνέχεια.
- ❖ Η ακρίβεια της περιγραφής των προσεγγίσεων μειώνεται καθώς πάμε από την φυσική περιγραφή στην προθεσιαρχική, ενώ η περιγραφική ικανότητα αυξάνεται.
 - ❑ Πχ. η φυσική περιγραφή είναι ακριβέστερη όλων αλλά παρουσιάζει σημαντικές δυσκολίες στην περιγραφή πολύπλοκων συστημάτων.



Με τι θα ασχοληθούμε:

- ❖ Η παρουσίαση αφορά τους **νοήμονες λογισμικούς πράκτορες** (*intelligent software agents*).
 - Η κατηγορία περιλαμβάνει ένα μεγάλο εύρος υπολογιστικών οντοτήτων, στο οποίο περιέχονται από
 - σχετικά απλά συστήματα, όπως είναι ο "συνδετήρας" που παρέχει συμβουλές στο πρόγραμμα επεξεργασίας κειμένου WORD,
 - μέχρι κατανεμημένα συστήματα ελέγχου βιομηχανικών εγκαταστάσεων, όπως είναι το ARCHON.

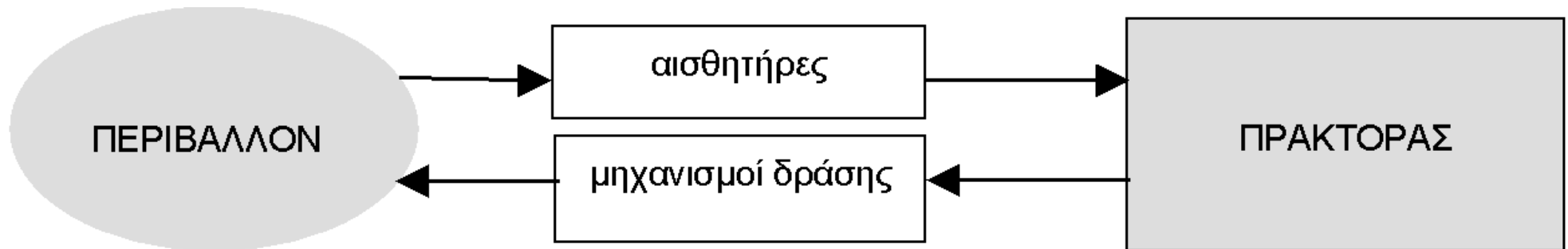
Ορισμοί Πρακτόρων

- ❖ Ανυπαρξία ενιαίου ορισμού
 - ❑ Διαφορετικοί ορισμοί προκύπτουν ανάλογα με τα χαρακτηριστικά στα οποία δίνεται έμφαση.

1) Βασικός Πράκτορας κατά Russel και Norvig

- ❖ Έμφαση στην αλληλεπίδραση πρακτόρων με το περιβάλλον τους
- ❖ Αφήνεται να εννοηθεί ότι κάθε πρόγραμμα είναι και πράκτορας

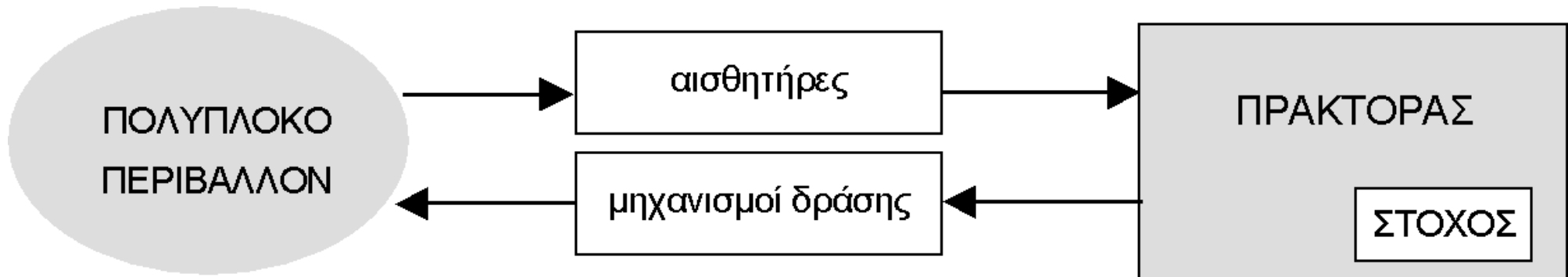
"Πράκτορας είναι οτιδήποτε μπορεί να αντιληφθεί το περιβάλλον του μέσω αισθητήρων και να αντιδράσει πάνω στο περιβάλλον μέσω μηχανισμών δράσης".



2) Βασικός Πράκτορας κατά Maes.

- ❖ Έμφαση δίνεται στην αυτονομία και στο πολύπλοκο και δυναμικό περιβάλλον

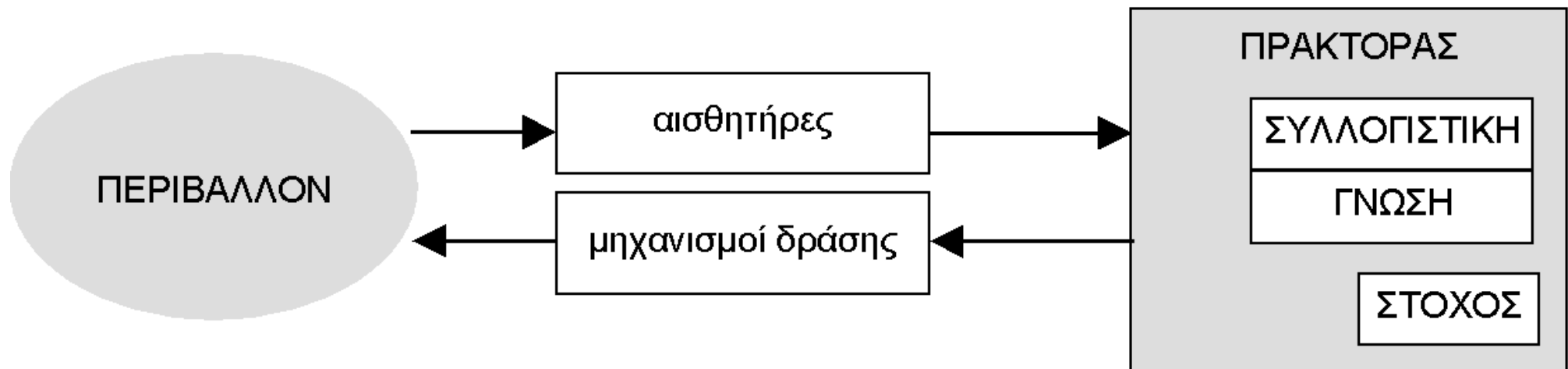
"Οι πράκτορες είναι υπολογιστικά συστήματα που δρουν σε ένα πολύπλοκο περιβάλλον, αντιλαμβάνονται και δρουν **αυτόνομα** πάνω σε αυτό, πετυχαίνοντας έτσι ένα σύνολο από **στόχους** για τους οποίους έχουν κατασκευαστεί".



3) Βασικός Πράκτορας κατά Hayes-Roth

❖ Έμφαση στη συλλογιστική

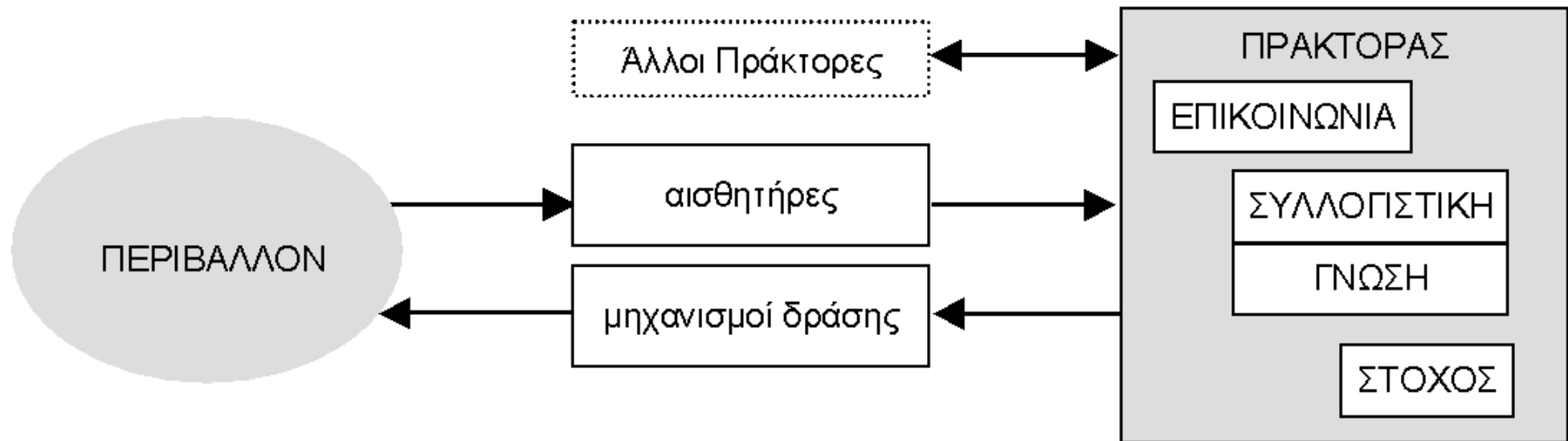
"Οι ευφυείς πράκτορες κάνουν συνεχώς τις εξής τρεις λειτουργίες: (α) αντιλαμβάνονται τις δυναμικές συνθήκες του περιβάλλοντος, (β) δρουν πάνω στο περιβάλλον ώστε να το αλλάξουν και (γ) **συλλογίζονται** ώστε να ερμηνεύσουν αυτά που αντιλαμβάνονται, να λύσουν προβλήματα, να συμπεράνουν και να καθορίσουν τη δράση τους".



4) Βασικός Πράκτορας κατά Coen

❖ Έμφαση στη διαδραστικότητα (interactivity)

"Λογισμικοί πράκτορες είναι προγράμματα που διενεργούν **διάλογο**, διαπραγματεύονται και συντονίζουν τη ροή πληροφοριών".



Κύρια Χαρακτηριστικά Πρακτόρων

- ❖ Σε τι διαφέρουν οι (λογισμικοί) πράκτορες από τα συμβατικά προγράμματα?
 - ❑ Τα όρια είναι πολλές φορές δυσδιάκριτα.

Βασικές διαφορές μεταξύ πρακτόρων και προγραμμάτων

- ❖ Σύμφωνα με τους Wooldridge και Jennings: "Πράκτορας είναι ένα σύστημα υλικού ή λογισμικού που έχει τις εξής ιδιότητες:
 - ❑ Αυτονομία (autonomy)
 - Οι πράκτορες λειτουργούν χωρίς την άμεση παρέμβαση των χρηστών ή άλλων πρακτόρων και έχουν αυτο-έλεγχο, δηλαδή έλεγχο της εσωτερικής τους κατάστασης και **αυτενέργεια**.
 - ❑ Κοινωνικότητα (social ability)
 - Οι πράκτορες επικοινωνούν με άλλους πράκτορες και χρήστες μέσω μιας κοινά κατανοητής **γλώσσας**. (π.χ. KQML, FIPA ACL)
 - ❑ Αντιδραστικότητα (reactiveness)
 - Οι πράκτορες αντιλαμβάνονται το περιβάλλον τους και αντιδρούν μέσα σε συγκεκριμένα χρονικά πλαίσια στις αλλαγές που επέρχονται σε αυτό.
 - ❑ Προνοητικότητα (pro-activeness)
 - Οι πράκτορες δεν αντιδρούν απλά στο περιβάλλον αλλά είναι ικανοί να επιδείξουν και **συμπεριφορά που βασίζεται σε στόχους**, λαμβάνοντας ουσιαστικά κάποια πρωτοβουλία ανάλογα με τις συνθήκες οι οποίες εμφανίζονται στο περιβάλλον τους.
- ❖ Τόσο η προνοητικότητα όσο και η αντιδραστικότητα απαιτούν σε κάποιο βαθμό τη δυνατότητα συλλογισμού από τον πράκτορα.

Δευτερεύοντα Χαρακτηριστικά Πρακτόρων

❖ Δεν εμφανίζονται σε όλες τις κατηγορίες πρακτόρων

❑ Κινητικότητα (mobility)

- Οι πράκτορες δεν είναι πάντα στατικοί, αλλά μπορούν να κινηθούν σε ένα υπολογιστικό περιβάλλον.

❑ Προσαρμοστικότητα (adaptivity)

- Οι πράκτορες προσαρμόζονται διαρκώς στο περιβάλλον τους ή τις απαιτήσεις του χρήστη, έχουν δηλαδή ικανότητα για μάθηση.

❑ Ειλικρίνεια (veracity)

- Οι πράκτορες δε δίνουν εσκεμμένα λάθος πληροφορίες.

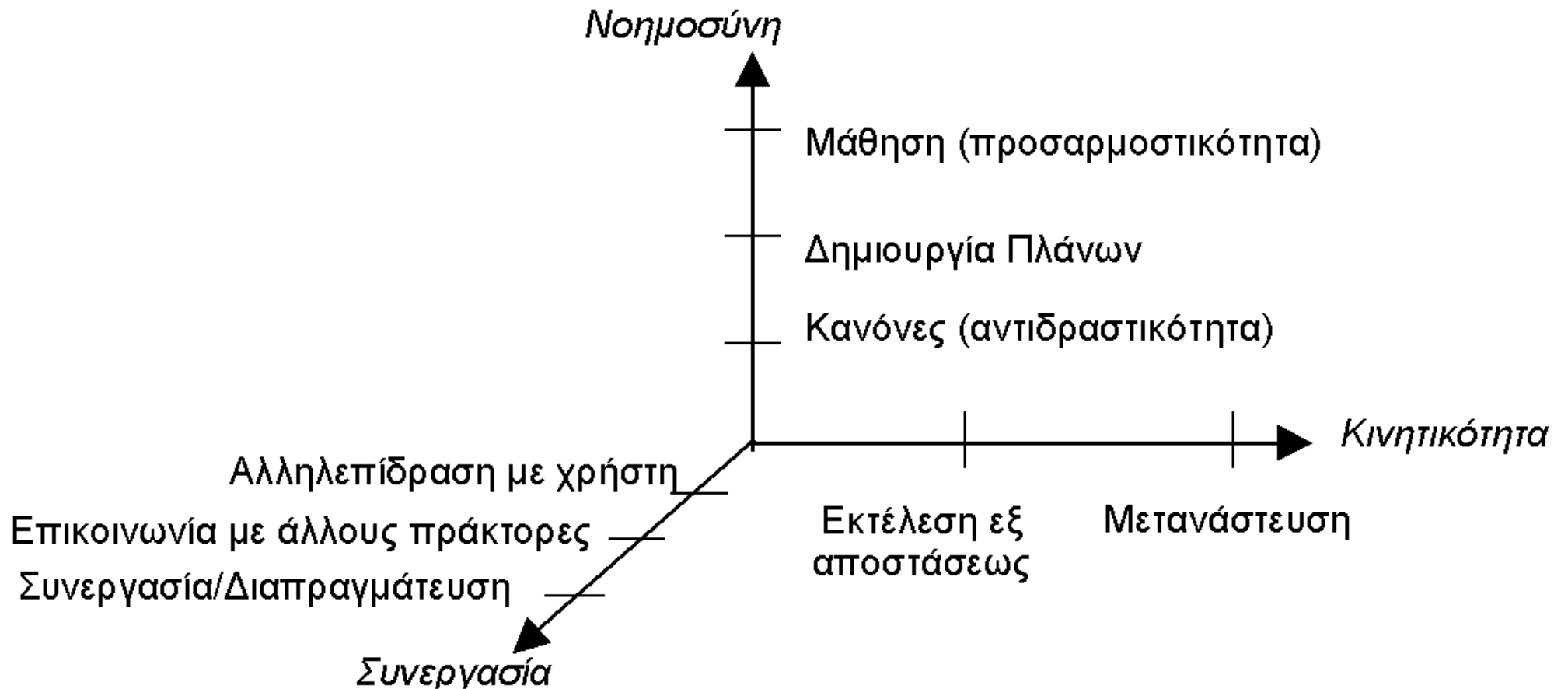
❑ Αγαθή προαίρεση (benevolence)

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

❑ Λογικότητα (rationality)

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

Άξονες Προσδιορισμού των Χαρακτηριστικών των Πρακτόρων



Μοντέλα Πρακτόρων

Θέματα που αντιμετωπίζονται:

- Είδη Περιβαλλόντων
- Αναπαράσταση (περιβάλλοντος και πράκτορα)
- Γενικές Αρχιτεκτονικές Πρακτόρων

Είδη Περιβαλλόντων

- ❖ Προσβάσιμα ή Μη Προσβάσιμα (Accessible vs. Inaccessible)
 - ❑ ανάλογα με το εάν υπάρχει διαθέσιμη πλήρης, ακριβής, και ανανεωμένη πληροφορία.
- ❖ Αιτιοκρατικά ή Μη Αιτιοκρατικά (Deterministic vs. Non-Deterministic)
 - ❑ αναφορικά με το εάν μία συγκεκριμένη ενέργεια έχει πάντα συγκεκριμένα αποτελέσματα.
- ❖ Επεισοδιακά ή Μη Επεισοδιακά (Episodic vs. Non-Episodic)
 - ❑ αναφορικά με το αν το περιβάλλον χωρίζεται ή όχι σε διακριτά και ανεξάρτητα επεισόδια.
 - ❑ Οι ενέργειες του πράκτορα σε ένα επεισόδιο δεν έχουν επίδραση στα υπόλοιπα.
 - ❑ Σε ένα επεισοδιακό περιβάλλον δεν υπάρχει ανάγκη να εξετάζει ο πράκτορας την επίδραση που θα έχουν οι τρέχουσες ενέργειες σε μελλοντικές καταστάσεις.
- ❖ Δυναμικά ή Στατικά (Dynamic vs. Static)
 - ❑ σε σχέση με το εάν εμφανίζονται αλλαγές χωρίς την παρέμβαση του πράκτορα.
- ❖ Διακριτά ή Συνεχή (Discrete vs. Continuous)
 - ❑ ως προς το εάν υπάρχει ή όχι ένας πεπερασμένος αριθμός ενεργειών και δεδομένων στο μηχανισμό αντίληψης του πράκτορα.

Αναπαράσταση Περιβάλλοντος και Πράκτορα

- ❖ Περιβάλλον (σύνολο καταστάσεων S) $S = \{s_1, s_2, \dots, s_n\}$
- ❖ Ικανότητες ενός πράκτορα (σύνολο ενεργειών A) $A = \{a_1, a_2, \dots, a_n\}$
- ❖ Πράκτορας (συνάρτηση agent)
 - ❑ πεδίο ορισμού την ακολουθία seq S των στιγμιότυπων του περιβάλλοντος
 - ❑ πεδίο τιμών τις ενέργειες A $\text{agent: seq } S \rightarrow A$
- ❖ Αντίληψη ενός πράκτορα (συνάρτηση see) $\text{see: } S \rightarrow P$
 - ❑ πεδίο ορισμού τις καταστάσεις S
 - ❑ πεδίο τιμών τα δεδομένα αντίληψης P (percepts)
- ❖ Εσωτερικές καταστάσεις I $I = \{I_1, I_2, \dots, I_n\}$
- ❖ Ορισμός εσωτερικής κατάστασης (συνάρτηση update) $\text{update: } I \times P \rightarrow I$

Γενικές Αρχιτεκτονικές Πρακτόρων

❖ **Νοήμονες Πράκτορες με Εσωτερική Κατάσταση**

- ❑ Πράκτορες με εσωτερική συμβολική αναπαράσταση του περιβάλλοντος στο οποίο βρίσκονται και ένα σύνολο κανόνων βάσει των οποίων καθορίζουν την επόμενη ενέργεια τους.
- ❑ Στόχους.

❖ **Αντιδραστικοί Πράκτορες**

A) Πράκτορες με Εσωτερική Κατάσταση

(α) Γενική Αρχιτεκτονική Πρακτόρων με Εσωτερική Κατάσταση

❖ Έχουν:

- ❑ Εσωτερική συμβολική αναπαράσταση του περιβάλλοντος
- ❑ Σύνολο κανόνων βάσει των οποίων καθορίζουν την επόμενη ενέργεια τους.

❖ Η συνάρτηση η οποία αναπαριστά τη λειτουργία τους:

Function `AgentWithState(environmentstate)` returns action

Static: variable `internalstate`, `rules`

begin

`percept` ← `see(environmentstate)`

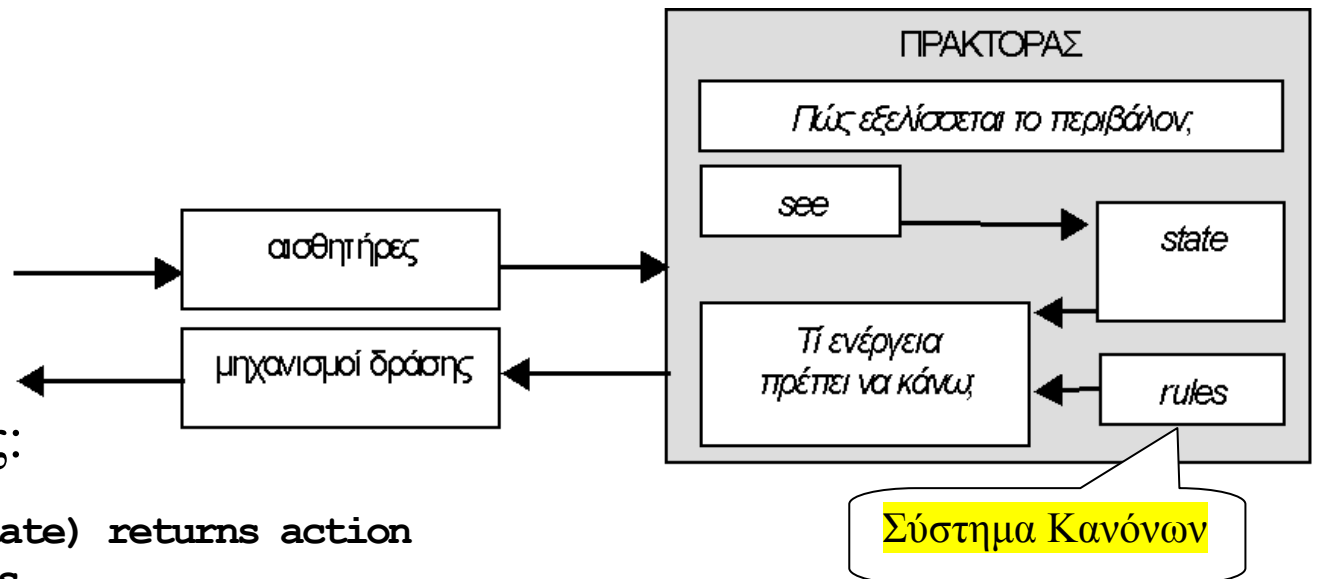
`internalstate` ← `update(internalstate, percept)`

`rule` ← `match(internalstate, rules)`

`action` ← `apply(rule)`

return action

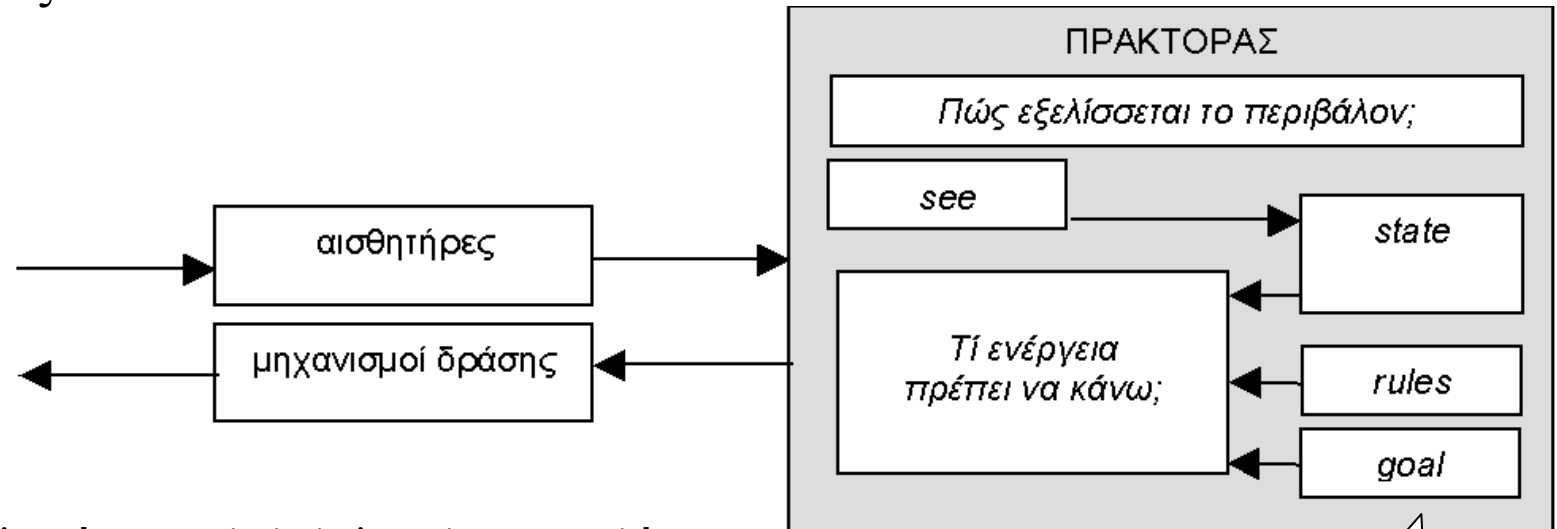
end



(β) Γενική Αρχιτεκτονική Πρακτόρων με Εσωτερική Κατάσταση και Στόχους (Προνοητική Συμπεριφορά)

- ❖ Είναι δυνατόν οι πράκτορες με εσωτερική κατάσταση να εμφανίζουν και προνοητική συμπεριφορά, να έχουν δηλαδή ένα σύνολο από στόχους που καθορίζουν την επόμενη ενέργεια τους.

- ❖ Η συνάρτηση που αναπαριστά τη λειτουργία τους:



Function GoalBasedAgents (environmentstate) returns action

Static: goal, internalstate, rules

begin

```
percept ← see(environmentstate)
```

```
internalstate ← update(internalstate, percept)
```

```
rule ← match(goal, internalstate, rules)
```

```
action ← apply(rule)
```

```
return action
```

end

Σύστημα σχεδιασμού
(Planning System)

Πράκτορες με Εσωτερική Κατάσταση

❖ Χαρακτηριστικά

- Ύπαρξη βάσης γνώσης, που περιέχει:
 - Την αντίληψή τους για τον πραγματικό κόσμο με μορφή λογικών προτάσεων (logic formulae).
 - Ένα σύνολο από κανόνες, οι οποίοι αναπαριστούν τις ενέργειες που μπορούν να εκτελέσουν.
- Διενέργεια Λογικών Συμπερασμών
- Κατάστρωση πλάνων για την επίτευξη των στόχων τους.

❖ Πλεονεκτήματα

- Οι μέθοδοι και οι τεχνικές που χρησιμοποιούνται έχουν σαφώς καθορισμένη και απλή σημασιολογία.

❖ Μειονεκτήματα

- Πιθανή αδυναμία εύρεσης μιας ακριβούς και ικανοποιητικής συμβολικής περιγραφής.
- Πιθανή αδυναμία εξαγωγής συμπερασμάτων σε ικανοποιητικό χρόνο λόγω του υψηλού υπολογιστικού κόστους των περισσότερων τεχνικών βασισμένων στη λογική.
- Δυσκολία αναπαράστασης δυναμικών περιβαλλόντων.
- Ύπαρξη αδυναμιών στην αναπαράσταση και στη συλλογιστική της έννοιας του χρόνου
- Αδυναμία στην αναπαράσταση της διαδικαστικής γνώσης.

Πράκτορες με Πεποιθήσεις - Επιθυμίες-Προθέσεις (BDI agents)

❖ Πράκτορες με Εσωτερική Κατάσταση, που αποτελούνται από:

❑ Πεποιθήσεις (Beliefs)

- αποτελούν την άποψη (view) και τη γνώση που έχει ο πράκτορας για το περιβάλλον του, η οποία ενδέχεται να είναι εσφαλμένη.

❑ Επιθυμίες (Desires)

- Αφορούν την κρίση του πράκτορα για τις μελλοντικές καταστάσεις του περιβάλλοντός του.
✓ Για παράδειγμα, μια μελλοντική κατάσταση μπορεί να είναι ή όχι επιθυμητή.
- Στο επίπεδο αυτό δεν εξετάζεται αν κάποια επιθυμητή κατάσταση είναι εφικτή, ενώ είναι δυνατό να υπάρχει σύγκρουση μεταξύ των επιθυμητών καταστάσεων.

❑ Στόχους (Goals)

- Αποτελούν υποσύνολο των επιθυμιών, για την επίτευξη των οποίων ο πράκτορας μπορεί να ενεργήσει.

❑ Προθέσεις (Intentions)

- Υποσύνολο των στόχων, που ο πράκτορας προσπαθεί να επιτύχει μια δεδομένη χρονική στιγμή.
- Θα πρέπει να σημειωθεί ότι στην πλειοψηφία των περιπτώσεων δεν είναι δυνατό να επιτευχθούν όλοι οι στόχοι ταυτόχρονα, οπότε και επιλέγεται ένα υποσύνολο τους, που αποτελεί τις προθέσεις, βάσει ορισμένων κριτηρίων ιεράρχησης.
- Οι στόχοι πρέπει να είναι εφικτοί και να μην περιέχουν συγκρούσεις μεταξύ τους.

❑ Πλάνα (Plans).

- Τα πλάνα αποτελούν τους τρόπους με τους οποίους ο πράκτορας θα επιτύχει τις προθέσεις του.

Ρόλος των Προθέσεων στην Συλλογιστική

- ❖ Οι προθέσεις παίζουν κυρίαρχο ρόλο στη συλλογιστική διαδικασία ενός πράκτορα
 - ❑ Από τη στιγμή που ο πράκτορας διαμορφώσει μια πρόθεση θα πρέπει να αποφασίσει και πώς θα την υλοποιήσει.
 - ❑ Δημιουργούν δεσμεύσεις στις μελλοντικές ενέργειες του πράκτορα.
 - ❑ Η προσπάθεια για την υλοποίηση μιας πρόθεσης αποκλείει την προσπάθεια για την επίτευξη προθέσεων που είναι ασύμβατες με αυτή.
 - ❑ Επηρεάζουν τις πεποιθήσεις του πράκτορα πάνω στις οποίες βασίζεται η μελλοντική συλλογιστική του
 - Ο πράκτορας πρέπει να πιστεύει ότι οι προθέσεις του είναι υλοποιήσιμες, αλλιώς θα ενεργεί παράλογα.
- ❖ Ο πράκτορας εμμένει στις προθέσεις του, δεν εγκαταλείπει την επιδίωξη των προθέσεων του χωρίς να συντρέχουν σημαντικοί λόγοι:
 - ❑ Να έχει επιτευχθεί η πρόθεση
 - ❑ Να πιστεύει ότι η επίτευξη της πρόθεσης είναι πλέον αδύνατη
 - ❑ Να έχει χαθεί το αρχικό κίνητρο το οποίο τον οδήγησε στην υιοθέτηση της.
- ❖ Έτσι οι πράκτορες θα πρέπει να επανεξετάσουν τις προθέσεις τους, απορρίπτοντας πιθανά προθέσεις για τις οποίες συντρέχουν οι παραπάνω λόγοι.
 - ❑ Σημαντικό υπολογιστικό κόστος
 - ❑ Συχνότητα επανεξέτασης επηρεάζει την απόδοση του πράκτορα.

Τολμηροί και Προσεκτικοί Πράκτορες

- ❖ Η απάντηση διαμορφώνεται από το περιβάλλον στο οποίο ενεργεί ο πράκτορας.
 - ❑ Σε ένα περιβάλλον το οποίο δεν μεταβάλλεται με ταχείς ρυθμούς, οι *τολμηροί* (*bold*) πράκτορες που δεν σταματούν για να επανεξετάσουν τις προθέσεις τους έχουν καλύτερη απόδοση.
 - ❑ Σε ένα ταχέως μεταβαλλόμενο περιβάλλον, οι *προσεκτικοί* (*cautious*) πράκτορες που επανεξετάζουν τις προθέσεις τους συχνά ανταποκρίνονται καλύτερα.

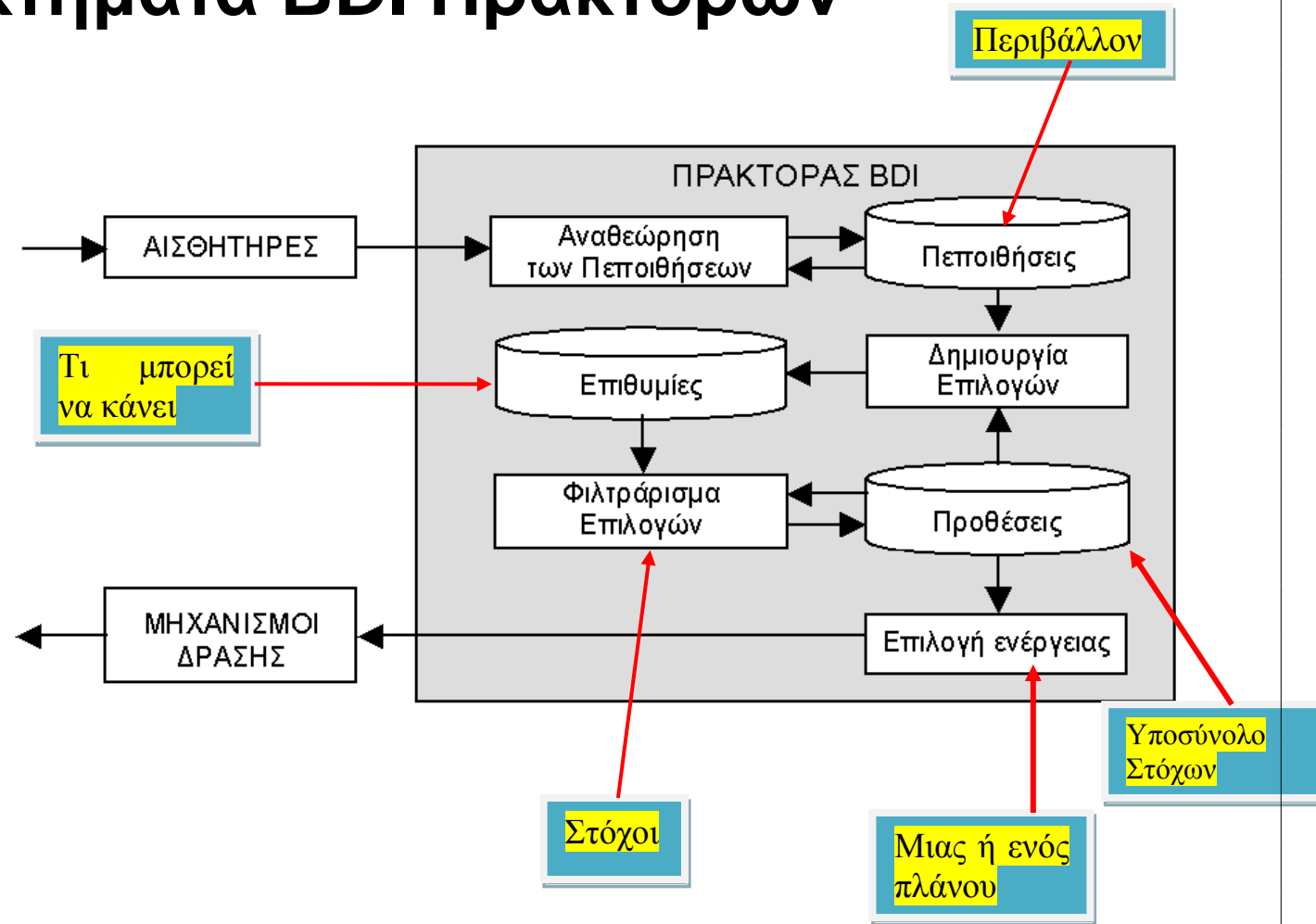
Αρχιτεκτονική BDI

- ❖ Η απλούστερη και ιστορικά πρώτη αρχιτεκτονική των πρακτόρων της κατηγορίας αυτής (BDI) αποτελείται από τα εξής μέρη (στα παρακάτω, P είναι οι προσλαμβάνουσες αντιλήψεις από τον κόσμο, και A το σύνολο των ενεργειών):
 - ❑ Ένα σύνολο B από πεποιθήσεις.
 - ❑ Ένα σύνολο D από επιθυμίες.
 - ❑ Ένα σύνολο I από προθέσεις.
 - ❑ Μια συνάρτηση αναθεώρησης των πεποιθήσεων: $Powerset(B) \times P \rightarrow Powerset(B)$
 - ❑ Μια συνάρτηση παραγωγής των διαθέσιμων επιλογών:
$$Powerset(B) \times Powerset(I) \rightarrow Powerset(D)$$
 - ❑ Μια συνάρτηση φιλτραρίσματος (deliberation):
$$Powerset(B) \times Powerset(D) \times Powerset(I) \rightarrow Powerset(I)$$
 - ❑ Μια συνάρτηση επιλογή κατάλληλης ενέργειας: $Powerset(I) \rightarrow A$

Πλεονεκτήματα BDI Πρακτόρων

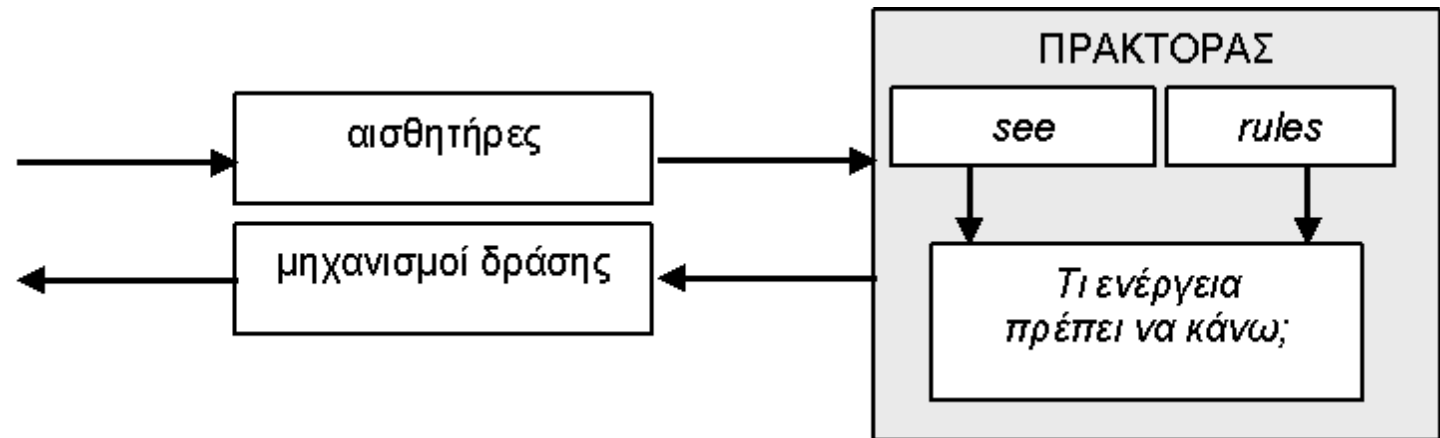
❖ Πλεονεκτήματα

- ❑ Διαισθητικά αποδεκτή αρχιτεκτονική
- ❑ Ξεκάθαρη αντιστοίχιση των επιμέρους στοιχείων της αρχιτεκτονικής σε λειτουργικά μέρη (functional decomposition).



B) Αντιδραστικοί Πράκτορες: Γενική Αρχιτεκτονική

- ❖ Ανυπαρξία εσωτερικής αναπαράστασης του κόσμου.
- ❖ Συμπεριφορά βασισμένη σε μια φιλοσοφία ερεθίσματος/αντίδρασης (**stimulus/response**).
 - ❑ Παίρνουν δεδομένα από το περιβάλλον (αντίληψη) και σύμφωνα με τους κανόνες λειτουργίας τους αποφασίζουν ποια θα είναι η ενέργεια στην οποία πρέπει να προβούν ως **αντίδραση** στην αντίληψή τους.
 - ❑ **Δεν έχουν μνήμη**, δηλαδή δεν υπολογίζουν τις επόμενες ενέργειες τους βάσει παλαιότερων καταστάσεων του κόσμου.
 - ❑ Συνάρτηση περιγραφής της λειτουργίας τους:



Function

```
SimpleReflexAgent (environments  
state) returns action
```

```
Static: rules
```

```
begin
```

```
percept ← see (environmentstate)
```

```
rule ← match (percept, rules)
```

```
action ← apply (rule)
```

```
return action
```

```
end
```



Αντιδραστικοί Πράκτορες:

- ❖ Αντιπροσωπεύουν μια εναλλακτική σχολή της TN:
 - ❑ Η ευφυής συμπεριφορά των συστημάτων θα προκύψει από συνδυασμό απλών σχετικά μερών (modules) που αλληλεπιδρούν μεταξύ τους.
- ❖ Κλασική Σχολή
 - ❑ Τα ευφυή συστήματα επεξεργάζονται και κάνουν συλλογισμούς χρησιμοποιώντας σύμβολα.

Αρχιτεκτονική Υπαγωγής

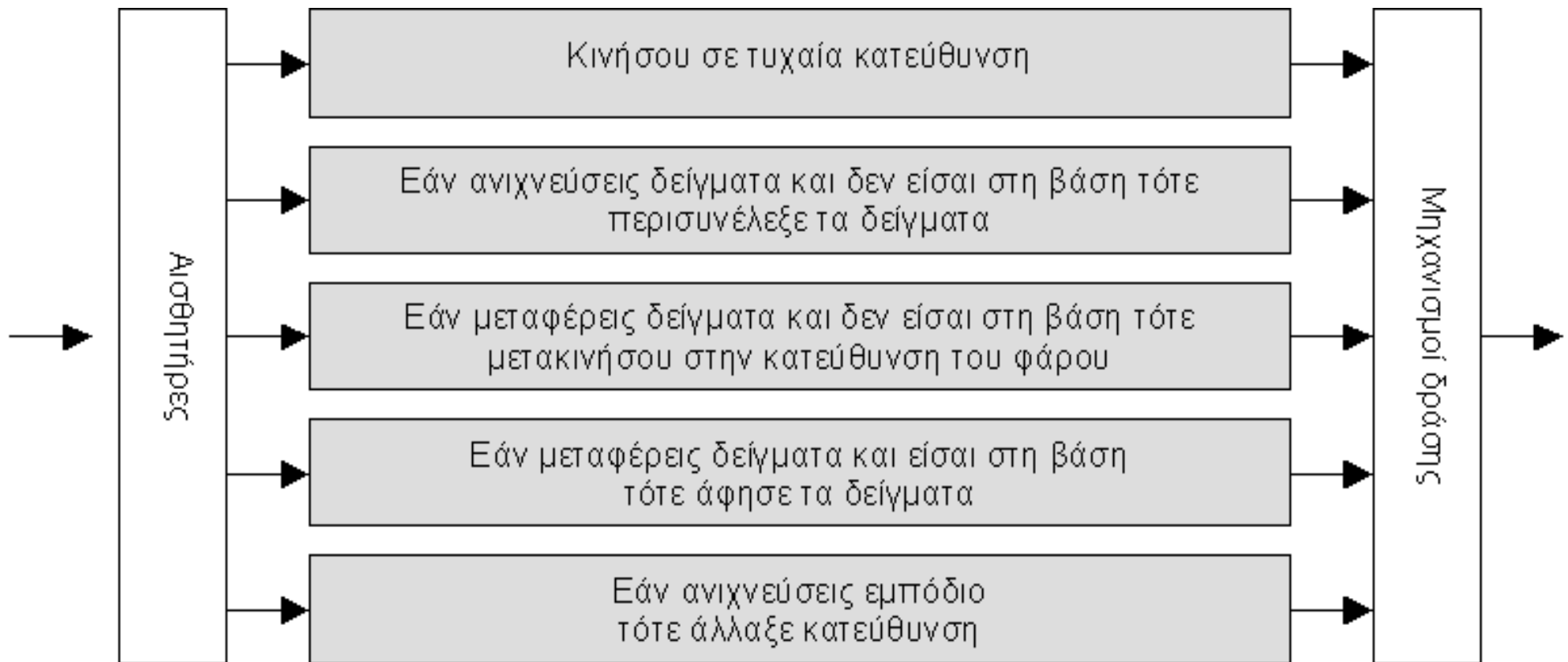
- ❖ Χαρακτηριστικό παράδειγμα αρχιτεκτονικής αντιδραστικού πράκτορα
- ❖ Αναπτύχθηκε από τον R. Brooks και εφαρμόστηκε σε ρομποτικούς πράκτορες (MIT)
- ❖ Το σύστημα αποτελείται από επαυξημένες μηχανές πεπερασμένων καταστάσεων (Augmented Finite State Machines - AFSM)
- ❖ Κάθε AFSM
 - ❑ Αναλαμβάνει μια ενέργεια/συμπεριφορά
 - ❑ Ενεργοποιείται βάσει των τιμών των αισθητήρων, μέσω των οποίων ο πράκτορας αντιλαμβάνεται το περιβάλλον.
- ❖ Οι AFSM είναι τοποθετημένες σε επίπεδα
 - ❑ AFSM ανώτερου επιπέδου να μπορούν να αναστείλουν τη λειτουργία των κατώτερων.
- ❖ Έξυπνη συμπεριφορά από την αλληλεπίδραση των μηχανών αυτών.



Παράδειγμα Πράκτορα Υπαγωγής (1/2)

- ❖ Υπάρχει ένα διαστημικό όχημα, το οποίο έχει προσηλαφιστεί σε έναν απομακρυσμένο πλανήτη και ένας αριθμός από πράκτορες, οι οποίοι πρέπει να συλλέξουν δείγματα και να τα μεταφέρουν στο σκάφος, αποφεύγοντας ένα πλήθος εμποδίων.
- ❖ Οι πράκτορες:
 - ❑ Χρησιμοποιούν ένα σύνολο από **αντιδραστικούς κανόνες (reactive rules)**, καθένας από τους οποίους αντιπροσωπεύει ουσιαστικά ένα επίπεδο σε μια αρχιτεκτονική υπαγωγής.
 - ❑ Δεν έχουν καμία απολύτως πληροφορία για το ανάγλυφο του εδάφους και αναγνωρίζουν τα εμπόδια μόνο από μικρή απόσταση.
 - ❑ Από το σκάφος εκπέμπεται ένα σήμα (φάρος προσανατολισμού) το οποίο είναι ικανό να λειτουργήσει σαν σημείο αναφοράς

Παράδειγμα Πράκτορα Υπαγωγής (2/2)



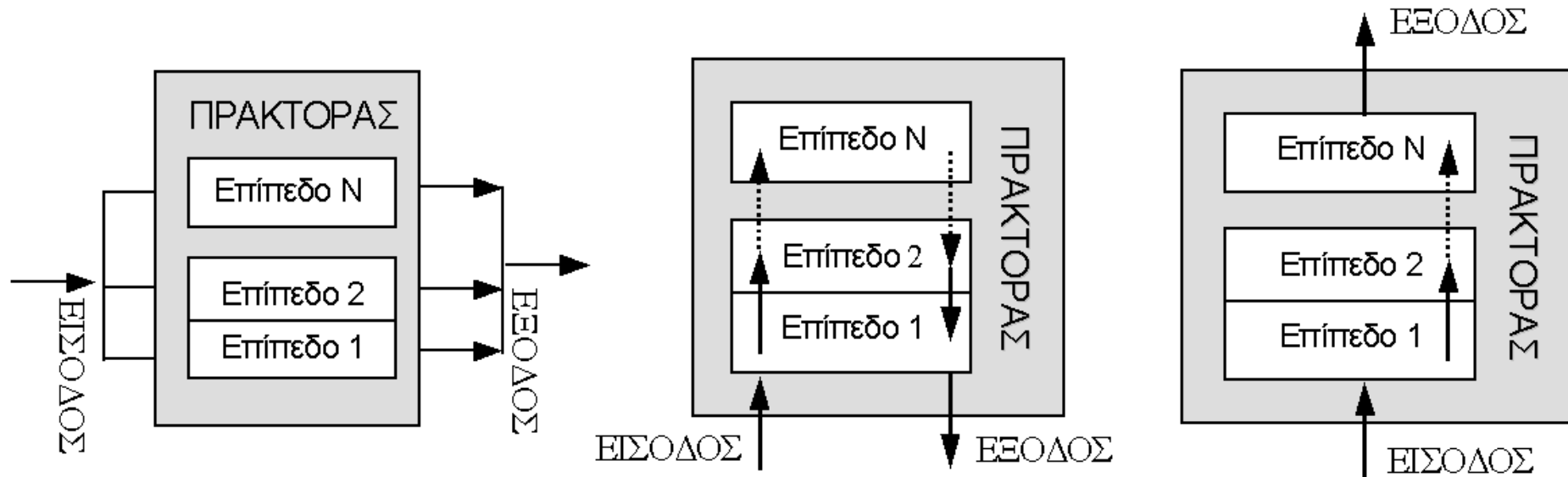
Αντιδραστικοί Πράκτορες

Μειονεκτήματα

- ❖ Είναι ανοικτό το θέμα του κατά πόσο είναι δυνατό οι πράκτορες της κατηγορίας αυτής να έχουν ένα σύνολο στόχων, το οποίο να μπορούν να διαχειριστούν με αποδοτικό τρόπο.
- ❖ Επειδή η συμπεριφορά τους καθορίζεται από την αλληλεπίδραση των μερών τους (modules), πρέπει οι ενέργειες να είναι κωδικοποιημένες στα μέρη αυτά, πράγμα που ουσιαστικά μεταφέρει το βάρος δημιουργίας πλάνων επίτευξης των στόχων στον άνθρωπο-σχεδιαστή αντί στον πράκτορα.
- ❖ Παραμένουν άλυτα τα προβλήματα επέκτασης και αποσφαλμάτωσης των πρακτόρων της κατηγορίας αυτής που παρουσιάστηκαν έντονα καθώς αυξάνονταν ο αριθμός των αλληλεπιδρώντων επιπέδων.
- ❖ Η συμπεριφορά που εμφανίζουν πολλές φορές τα συστήματα αυτά δεν είναι κατανοητή παρά μόνο μέσα από μεθόδους δοκιμής και λάθους (trial and error).

Γ) Υβριδικοί Πράκτορες

- ❖ Συνδυασμός των δύο προηγούμενων αρχιτεκτονικών
- ❖ Ύπαρξη τουλάχιστον δύο επίπεδων
 - ❑ Ένα επίπεδο για την αντιδραστική συμπεριφορά του πράκτορα
 - ❑ Ένα επίπεδο για τη συμπεριφορά με εσωτερική κατάσταση.
- ❖ Η ροή ελέγχου μπορεί να είναι:
 - ❑ **Οριζόντια:** Όλα τα επίπεδα να είναι συνδεδεμένα στους αισθητήρες εισόδου και στους μηχανισμούς δράσης. Σημείο ελέγχου ή μεσολαβητής καθορίζει ποιο επίπεδο αναλαμβάνει τον έλεγχο του πράκτορα.
 - ❑ **Κάθετη:** Ένα επίπεδο συνδεδεμένο στους αισθητήρες και ένα στους μηχανισμούς δράσης.



Αρχιτεκτονική Οριζόντιας Ροής Ελέγχου

❖ Χαρακτηριστικό παράδειγμα η Αρχιτεκτονική των πρακτόρων TOURING MACHINE

❑ Καθοδήγηση αυτόνομων οχημάτων

❖ Τρία επίπεδα:

❑ Το αντιδραστικό επίπεδο (*reactive layer*)

- Προτείνει άμεσες ενέργειες που πρέπει να γίνουν λόγω αλλαγών στο περιβάλλον
- Η υλοποίησή του ακολουθεί εκείνη της αρχιτεκτονικής υπαγωγής.

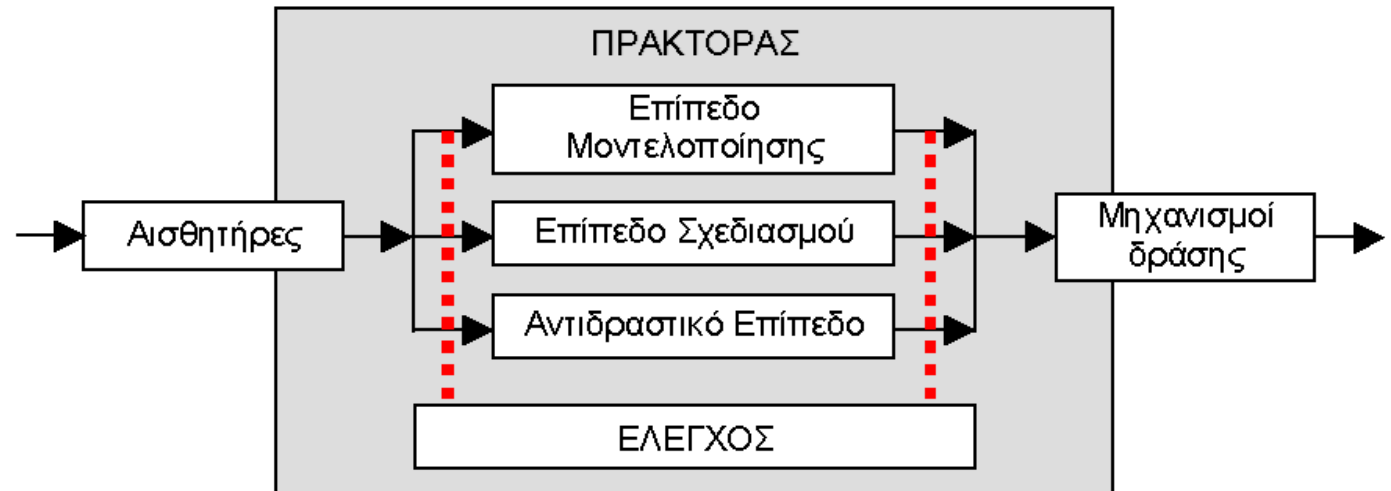
❑ Το επίπεδο σχεδιασμού (*planning layer*)

- Προτείνει τι θα κάνει ο πράκτορας κάτω από κανονικές συνθήκες και πώς θα επιτύχει τους στόχους του

❑ Το επίπεδο μοντελοποίησης (*modeling layer*)

- Είναι υπεύθυνο για την αναπαράσταση όλων των αντικειμένων στο περιβάλλον του πράκτορα,
- καθώς και για την πρόβλεψη πιθανών συγκρούσεων συμφερόντων με άλλους πράκτορες και
- προτείνει τους στόχους που θα πρέπει να τεθούν για την αποφυγή των συγκρούσεων.

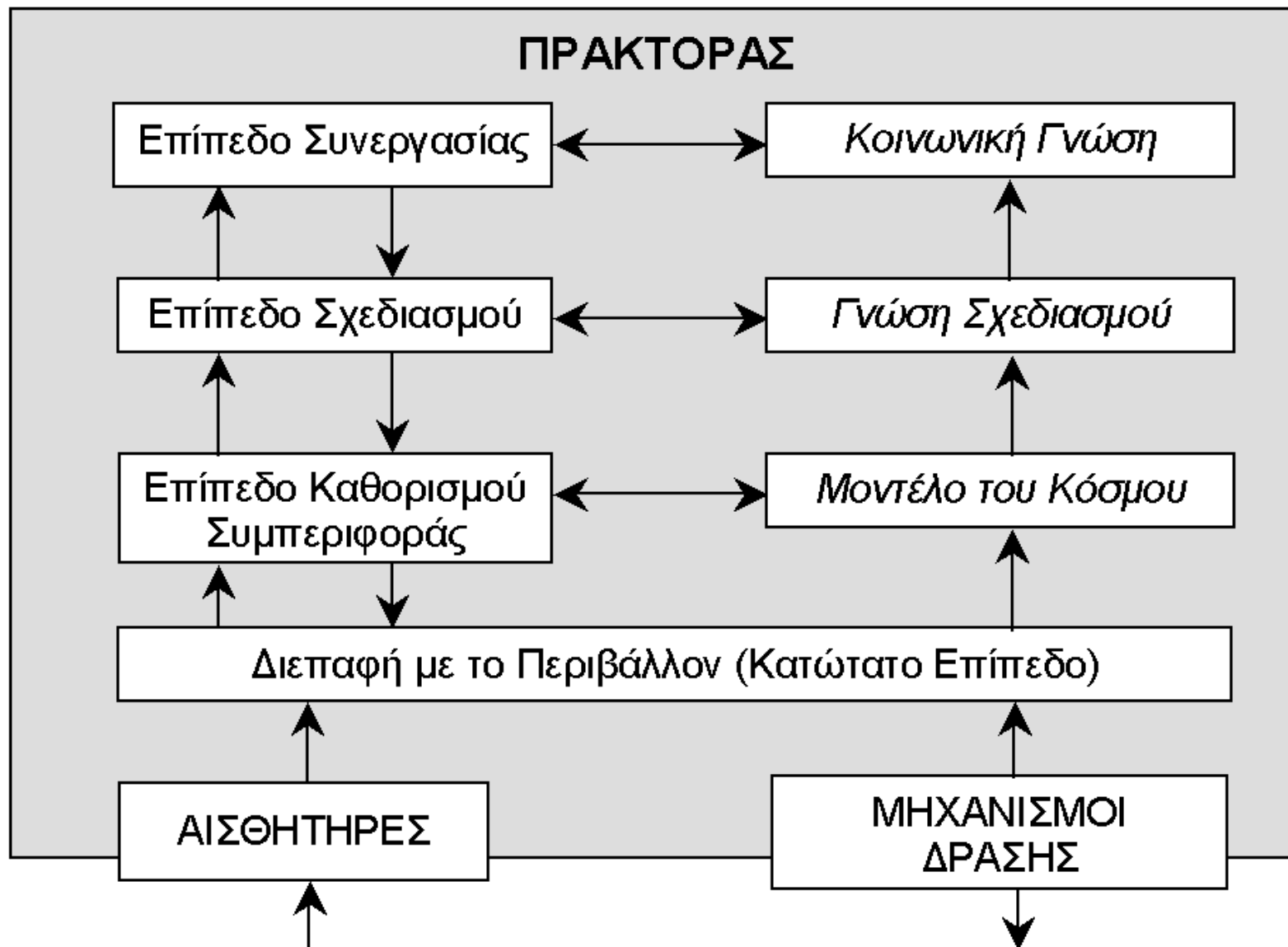
❖ Το σύστημα ελέγχου (control) είναι υπεύθυνο για την επιλογή της κατάλληλης ενέργειας ανάμεσα στις προτεινόμενες από τα τρία επίπεδα.



Αρχιτεκτονική Κάθετης Ροής Ελέγχου (1/2)

- ❖ Χαρακτηριστικό παράδειγμα η αρχιτεκτονική στο σύστημα INTERRAP.
- ❖ Τρία επίπεδα με μια βάση γνώσης στο καθένα:
 - ❑ Το επίπεδο καθορισμού συμπεριφοράς (behaviour layer)
 - Υπεύθυνο για τις χαμηλού επιπέδου αντιδραστικές ενέργειες του πράκτορα.
 - Η βάση γνώσης του περιέχει χαμηλού επιπέδου πληροφορίες για τον κόσμο.
 - ❑ Το επίπεδο σχεδιασμού (planning layer)
 - Υπεύθυνο για τη δημιουργία πλάνων για την επίτευξη των στόχων του πράκτορα.
 - Η βάση γνώσης του περιέχει μια βιβλιοθήκη πλάνων.
 - ❑ Το επίπεδο συνεργασίας (cooperation layer)
 - Υπεύθυνο για τη δημιουργία πλάνων συνεργασίας με άλλους πράκτορες.
 - Η βάση γνώσης του περιέχει πληροφορίες για τους άλλους πράκτορες.
- ❖ Ένα κατώτατο επίπεδο που διαχειρίζεται την είσοδο και την έξοδο του πράκτορα με το περιβάλλον
- ❖ Η ροή ελέγχου βασίζεται σε δύο αρχές:
 - ❑ Ενεργοποίηση από κάτω προς τα πάνω (bottom-up activation)
 - ενεργοποίηση των επιπέδων γίνεται ξεκινώντας πρώτα από τα κατώτερα επίπεδα.
 - ❑ Εκτέλεση από πάνω προς τα κάτω (top-down execution)
 - υλοποίηση των στόχων ενός επιπέδου γίνεται με τις δυνατότητες που του προσφέρει το αμέσως κατώτερο επίπεδο.

Αρχιτεκτονική Κάθετης Ροής Ελέγχου (2/2)



Δ) Κινητοί Πράκτορες

❖ Χαρακτηριστικά

- ❑ "Μετακίνηση" μέσα σε ένα δικτυακό περιβάλλον (τοπικό δίκτυο, Internet) για να επιτύχουν τους στόχους τους.
- ❑ Είναι διεργασίες (software processes), οι οποίες κατά τη διάρκεια της εκτέλεσης τους μεταφέρονται στους υπολογιστές που συμμετέχουν στο δίκτυο-περιβάλλον.
- ❑ Δυνατότητα να επικοινωνίας με άλλους πράκτορες και με διάφορες πηγές πληροφορίας ή παροχής συγκεκριμένων υπηρεσιών.

Παράδειγμα

❖ Αναζήτηση κατάλληλων πτήσεων για κάποιο προορισμό.

❖ Ένας στατικός πράκτορας:

- ❑ Θα έβρισκε πρώτα τις διευθύνσεις των απαραίτητων βάσεων δεδομένων στο δίκτυο.
- ❑ Θα έστελνε στις βάσεις αυτές τις κατάλληλες ερωτήσεις (queries).
- ❑ Θα φιλτράριζε τα δεδομένα που θα λάμβανε και θα τα παρουσίαζε στο χρήστη.

❖ Ένας κινητός πράκτορας:

- ❑ Θα μεταφέρονταν στον υπολογιστή όπου βρίσκεται η βάση δεδομένων,
- ❑ Θα έκανε όλες τις αναζητήσεις και τα απαραίτητα φιλτραρίσματα τοπικά,
- ❑ Θα μεταφέρονταν στην επόμενη βάση δεδομένων επαναλαμβάνοντας τη διαδικασία και
- ❑ Θα επέστρεφε στον υπολογιστή του χρήστη και θα του παρουσίαζε τα αποτελέσματα της αναζήτησης.

Σύγκριση των Δύο Προσεγγίσεων

- ❖ Ο σταθερός πράκτορας
 - ❑ Απαιτεί συνεχή σύνδεση με το δίκτυο
 - ❑ Έχει υψηλό επικοινωνιακό κόστος καθώς μεγάλος όγκος δεδομένων διακινείται μέσω του δικτύου

- ❖ Ο κινητός πράκτορας
 - ❑ Μειώνει στο ελάχιστο το κόστος επικοινωνίας
 - Δεν απαιτείται συνεχής σύνδεση του Η/Υ του χρήστη με το δίκτυο, παρά μόνο κατά την "αναχώρηση" και την "άφιξη" του πράκτορα,
 - ❑ Όλες οι συναλλαγές μεταξύ των βάσεων δεδομένων και του πράκτορα γίνονται σε τοπικό επίπεδο
 - Μειώνεται στο ελάχιστο ο όγκος των δεδομένων που μεταφέρονται μέσω του δικτύου.
 - ❑ Παρουσιάζει μεγαλύτερη αξιοπιστία
 - Η λειτουργία του ανεξαρτητοποιείται εν μέρει από τη διαθεσιμότητα του δικτύου.
 - ❑ Υπάρχει ασύγχρονη εκτέλεση του κινητού πράκτορα και των άλλων εφαρμογών του χρήστη.



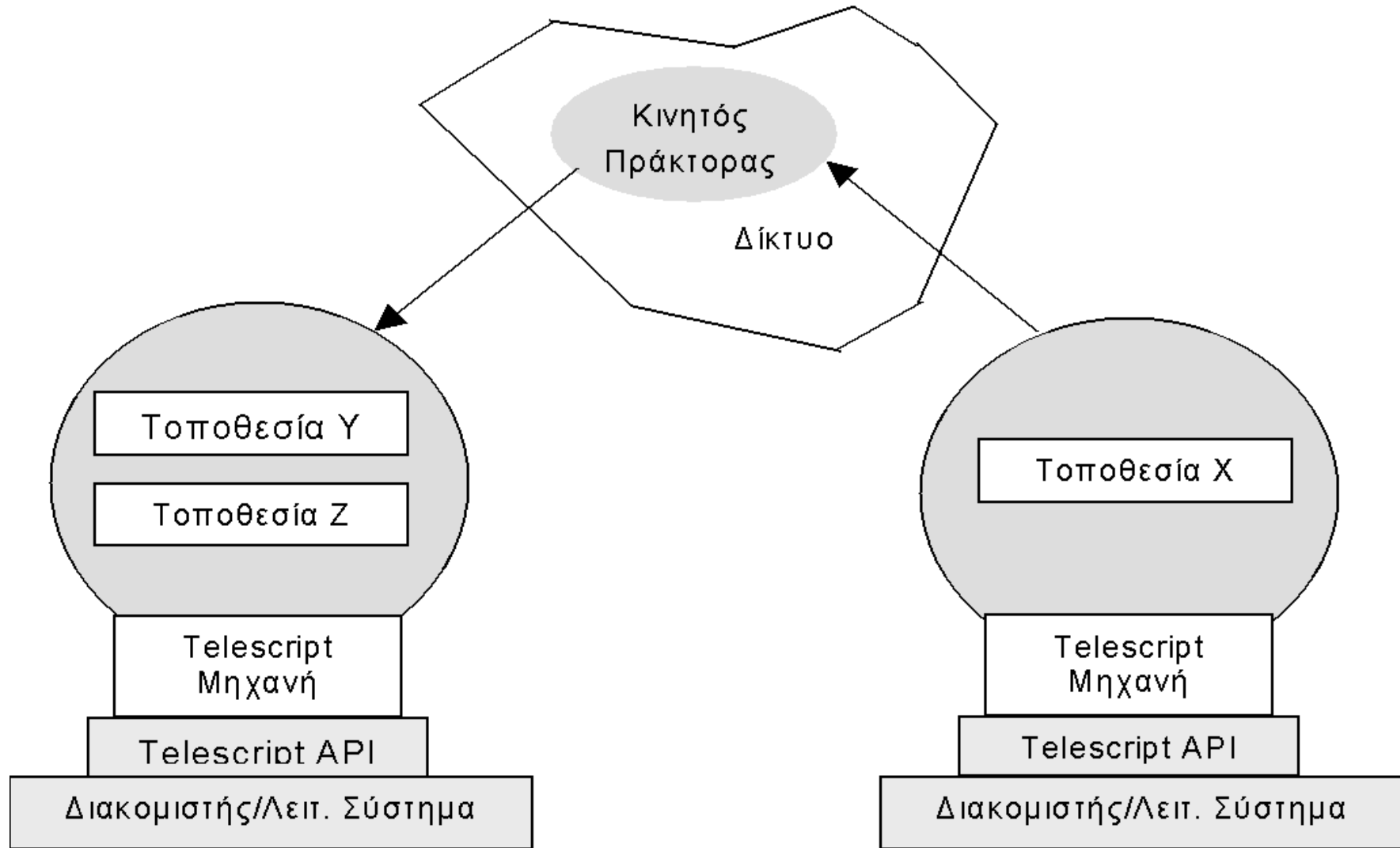
Προβλήματα στην Ανάπτυξη Κινητών Πρακτόρων

- ❖ Ασφάλεια του πράκτορα και του συστήματος που θα τον φιλοξενήσει.
 - ❑ Εξασφάλιση ότι ο κώδικας του πράκτορα δε θα αλλοιωθεί κατά την εκτέλεση του και κατά τη μεταφορά του.
 - ❑ Οχύρωση του συστήματος φιλοξενίας από πιθανές "επιθέσεις" που είναι πιθανό να δεχθεί από κινητούς πράκτορες με μη-αγαθή προαίρεση.
- ❖ Δυσκολία στον έλεγχο της εκτέλεσης των κινητών πρακτόρων
 - ❑ Δεν είναι εύκολο πάντα να καθοριστεί που βρίσκεται ο πράκτορας και ποια είναι η τρέχουσα κατάσταση του.
- ❖ Πιθανότητα το κόστος της μεταφοράς του πράκτορα να είναι συγκρίσιμο με εκείνο της αλληλεπίδρασης από απόσταση.
- ❖ Η ανάπτυξη κινητών πρακτόρων απαιτεί εξειδικευμένα εργαλεία.
 - ❑ Π.χ. η γλώσσα προγραμματισμού TELESCRIPT, JAVA, C++, κλπ.

Η Γλώσσα Προγραμματισμού TELESCRIPT (1/2)

- ❖ Αντικειμενοστραφής γλώσσα προγραμματισμού.
- ❖ Σχεδιάστηκε ειδικά για υλοποίηση κινητών πρακτόρων
- ❖ Οι εφαρμογές που αναπτύσσονται στη συγκεκριμένη γλώσσα αποτελούνται από:
 - ❑ TELESCRIPT Μηχανές (engines)
 - Διερμηνευτές (interpreters) που μπορούν να φιλοξενήσουν πολλές διεργασίες.
 - ❑ Τοποθεσίες (places)
 - Διεργασίες που προσφέρουν υπηρεσίες και μπορούν να περιέχουν και άλλες διεργασίες.
 - ❑ Πράκτορες (TELESCRIPT agents)
 - Διεργασίες οι οποίες μπορούν να μετακινηθούν από τη μια τοποθεσία στην άλλη.
- ❖ Ένας πράκτορας:
 - ❑ χρησιμοποιεί τις υπηρεσίες μιας τοποθεσίας όταν βρίσκεται σε αυτή,
 - ❑ επικοινωνεί μόνο με πράκτορες που βρίσκονται στην ίδια τοποθεσία.
- ❖ Η διαδικασία μετακίνησης (go) περιλαμβάνει:
 - ❑ Το "πακετάρισμα" του κώδικα του πράκτορα μαζί με όλα τα δεδομένα του, τη στοίβα και το δείκτη εντολών από τη μηχανή TELESCRIPT.
 - ❑ Αποστολή του πράκτορα στην τοποθεσία στόχο,
 - ❑ Το "ξεπακετάρισμα" του πράκτορα από την μηχανή, έλεγχος αυθεντικότητας και να συνέχιση της εκτέλεσης από το σημείο διακοπής.

Η Γλώσσα Προγραμματισμού TELESRIPT (2/2)



Υλοποίηση Κινητών Πρακτόρων με Java

- ❖ Η Java σχεδιάστηκε με σκοπό να μπορεί να υποστηρίξει τη μεταφορά και εκτέλεση κώδικα σε μηχανές συνδεδεμένες σε ένα υπολογιστικό περιβάλλον.
 - ❑ Αποτελεί ίσως την πλέον διαδεδομένη γλώσσα προγραμματισμού στο διαδίκτυο.
 - ❑ Ένα πρόγραμμα στη συγκεκριμένη γλώσσα μεταφράζεται σε ενδιάμεσο διερμηνευόμενο κώδικα (bytecode), ο οποίος εκτελείται από μια εικονική μηχανή.
- ❖ **Πλεονεκτήματα:**
 - ❑ Μεγάλη διάδοση
 - ❑ Μεταφέρσιμη σε πολλές υπολογιστικές πλατφόρμες
 - ❑ Ενσωματωμένος μηχανισμός ασφάλειας.
- ❖ **Μειονεκτήματα:**
 - ❑ Δυσκολία υλοποίησης πρακτόρων με νοήμονα συμπεριφορά, σε σχέση με τις κλασσικές γλώσσες δηλωτικού προγραμματισμού (Prolog).
 - ❑ Μειωμένη απόδοση των προγραμμάτων.
- ❖ Πλήθος εργαλείων ανάπτυξης κινητών πρακτόρων που βασίζονται στην Java
 - ❑ Για παράδειγμα το περιβάλλον ανάπτυξης των **Aglets**, της IBM Research.
 - ❑ Προσφέρει δυνατότητες εύκολης υλοποίησης της διαδικασίας μετακίνησης πρακτόρων, της επικοινωνίας μεταξύ αυτών και των διαδικασιών ασφάλειας.

Υλοποίηση Κινητών Πρακτόρων με Aglets

- ❖ Οι Aglets είναι κινητοί πράκτορες οι οποίοι είναι υλοποιημένοι σε JAVA
 - ❑ Σύνολο από μεθόδους και κλάσεις (υλοποιήσεις των βασικών λειτουργιών)
- ❖ **Διαφορά από τα Applets της JAVA.**
 - ❑ Μαζί με τον κώδικα μεταφέρεται η κατάσταση του προγράμματος (πράκτορα) και συνεχίζεται η εκτέλεση από το σημείο όπου είχε σταματήσει πριν τη μετακίνηση.
- ❖ Απαιτούν για την εκτέλεση τους την ύπαρξη μιας εικονικής μηχανής (Aglet Host).
 - ❑ Aglet Viewers: εικονικές μηχανές με σχετικά μικρή ισχύ.
 - ❑ Aglet Servers: ισχυρές μηχανές, προσανατολισμένες στην παροχή υπηρεσιών.
- ❖ Οι Aglets κινούνται σε ένα περιβάλλον που αποτελείται από τέτοιες μηχανές.
- ❖ Πλεονεκτήματα στην ανάπτυξη κινητών πρακτόρων:
 - ❑ Πλήρη υποστήριξη ανάπτυξης πρακτόρων βασισμένων στη Java
 - ❑ Ευκολία στην εκμάθηση και χρήση των προσφερόμενων μεθόδων και κλάσεων
 - ❑ Ύπαρξη πλήρους περιβάλλοντος ανάπτυξης.
 - ❑

Άλλες Γλώσσες (πλατφόρμες) Υλοποίησης

- ❖ JADE (Java Agent DEvelopment Framework)
- ❖ GrassHopper (JAVA)
- ❖ Agent-Tcl, Safe-Tcl
- ❖ C/C++.



Java Agent DEvelopment Framework
is an Open Source platform
for peer-to-peer agent based applications

JADE (Java Agent DEvelopment Framework) is a software Framework fully implemented in Java language. It simplifies the implementation of multi-agent systems through a middle-ware that complies with the **FIPA specifications** and through a set of **graphical tools** that supports the debugging and deployment phases. The agent platform can be distributed across machines (which not even need to share the same OS) and the configuration can be controlled via a **remote GUI**. The configuration can be even changed at run-time by moving agents from one machine to another one, as and when required. JADE is completely implemented in Java language and the minimal system requirement is the version 1.4 of JAVA (the run time environment or the JDK). The synergy between the JADE platform and the LEAP libraries allows to obtain a FIPA-compliant agent platform with reduced footprint and compatibility with mobile Java environments down to J2ME-CLDC MIDP 1.0. The LEAP libraries have been developed with the collaboration of the LEAP project and can be downloaded as an **add-on** of JADE from this same Web site.

JADE is free software and is distributed by **Telecom Italia**, the copyright holder, in open source software under the terms of the LGPL (Lesser General Public License Version 2). Since May 2003, a **JADE Board** has been created that supervisions the management of the JADE Project. Currently the JADE Board lists 5members: **Telecom Italia, Motorola, Whitestein Technologies AG, Profactor GmbH, and France Telecom R&D.**

The latest version of JADE is JADE 4.0 released on 20/04/2010.



NetLogo



- Home
- Download
- Mailing Lists
- Models
- Community Models
- Extensions
- FAQ
- Resources
- Contact Us

- User Manuals:
- Web
- Printable
- Chinese
- Czech

Donate

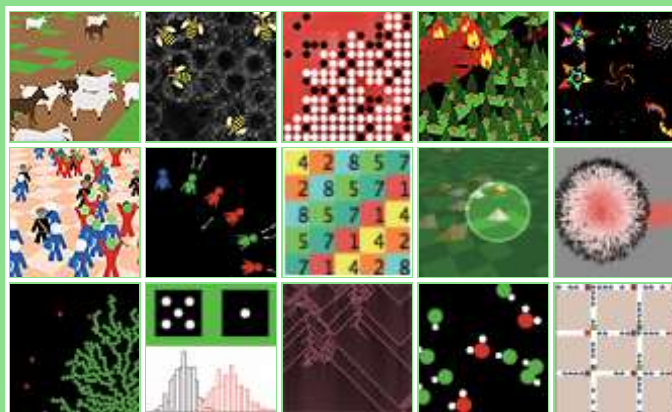
NetLogo is a multi-agent programmable modeling environment. It is used by tens of thousands of students, teachers and researchers worldwide. It also powers **HubNet** participatory simulations. It is authored by **Uri Wilensky** and developed at the **CCL**. You can download it free of charge.

What can you do with NetLogo? Read more [here](#).

Join mailing lists [here](#).

Download

NetLogo comes with a large library of sample models. Click on some examples below.



© 1999-2010 Uri Wilensky ([details & terms of use](#))





Πολυπρακτορικά Συστήματα

*"There is no such thing as a single agent system".
[Woodridge 2002]*

Πολυπρακτορικά Συστήματα

- ❖ Δίκτυο από πράκτορες που δρουν μαζί για να επιλύσουν προβλήματα που είναι πέρα των δυνατοτήτων και της γνώσης ενός μόνο πράκτορα.
- ❖ Αποτελούν βασικό τομέα της Κατανεμημένης ΤΝ
- ❖ **Ένα τέτοιο σύστημα στοχεύει στην:**
 - ❑ Επίλυση προβλημάτων που είναι πολύ πολύπλοκα για να επιλυθούν αποδοτικά από ένα μόνο πράκτορα
 - ❑ Επίλυση προβλημάτων τα οποία είναι από τη φύση τους κατανεμημένα
 - ❑ Διασύνδεση και λειτουργία ήδη υπαρχόντων συστημάτων (legacy systems) έτσι ώστε να είναι εύκολη η εκμετάλλευση τους χωρίς σημαντικές τροποποιήσεις (mediator systems).
- ❖ Κύριο χαρακτηριστικό των συνεργαζόμενων πρακτόρων: **Συντονισμός (Coordination)**

"Ο συντονισμός είναι η ιδιότητα ενός συστήματος πρακτόρων να φέρουν εις πέρας ενέργειες μέσα σε ένα κοινό περιβάλλον"

- ❖ Οι πράκτορες είτε:
 - ❑ **Εργάζονται αυτόνομα** ανταλλάσσοντας πληροφορίες και προσπαθούν να επιτύχουν τους δικούς τους ανεξάρτητους στόχους. (**Διαπραγμάτευση-Negotiation**)
 - ❑ **Συνεργάζονται** επιλύνοντας υποπροβλήματα, ώστε ο συνδυασμός των επιμέρους λύσεων που θα προκύψουν να αποτελέσει την τελική λύση. (**Συνεργασία-Cooperation**)

Άλλα Χαρακτηριστικά

- ❖ Κανένας πράκτορας δεν έχει πλήρη πληροφορία.
- ❖ Δεν υπάρχει κεντρικός έλεγχος στο σύστημα.
- ❖ Τα δεδομένα είναι κατανεμημένα.
- ❖ Οι υπολογισμοί γίνονται με ασύγχρονο τρόπο.

Κρίσιμα σημεία στη σχεδίαση και υλοποίηση

- ❖ Επικοινωνία πρακτόρων
 - ❑ Ποιες γλώσσες και πρωτόκολλα θα χρησιμοποιηθούν.
 - ❑ Πότε αυτοί επικοινωνούν και τι πληροφορία ανταλλάσσουν.
 - ❑ Με ποιους άλλους πράκτορες επικοινωνούν και ποιο είναι το επιπλέον κόστος.
- ❖ Αλληλεπίδραση πρακτόρων
 - ❑ Τυποποίηση, περιγραφή, διαμοιρασμός του προβλήματος και σύνθεση λύσεων, σε μία ομάδα νοημόνων πρακτόρων.
 - ❑ Τρόπος συμβιβασμού διαφορετικών απόψεων από πράκτορες, αντιμετώπιση ενδεχόμενων συγκρουόμενων προθέσεων τους και τρόπος διαχείρισης περιορισμένων πόρων.



ΕΠΙΚΟΙΝΩΝΙΑ

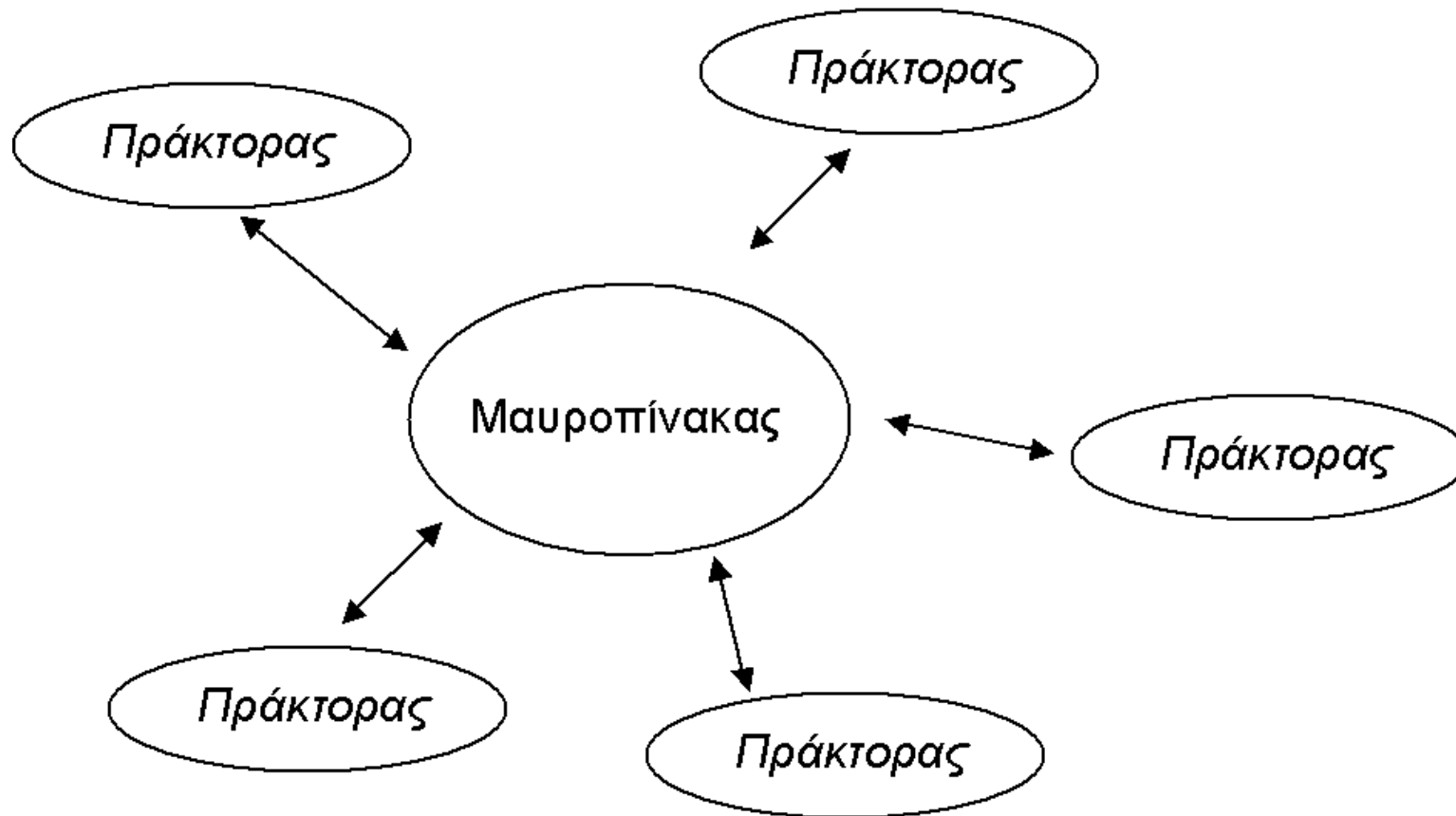
- ❖ Το σημαντικότερο και πιο κρίσιμο σημείο στην ανάπτυξη συστημάτων.
- ❖ Απαιτεί την ύπαρξη τριών διαφορετικών επιπέδων.
 - ❑ Το κατώτερο επίπεδο (τρόπος διασύνδεσης).
 - ❑ Το μεσαίο επίπεδο (σύνταξη και τη μορφή των μηνυμάτων)
 - ❑ Το ανώτερο επίπεδο (σημασιολογία)

Μοντέλα Διασύνδεσης

- ❖ Συστήματα μαυροπίνακα (blackboard systems)
- ❖ Συστήματα ανταλλαγής μηνυμάτων (message passing systems).

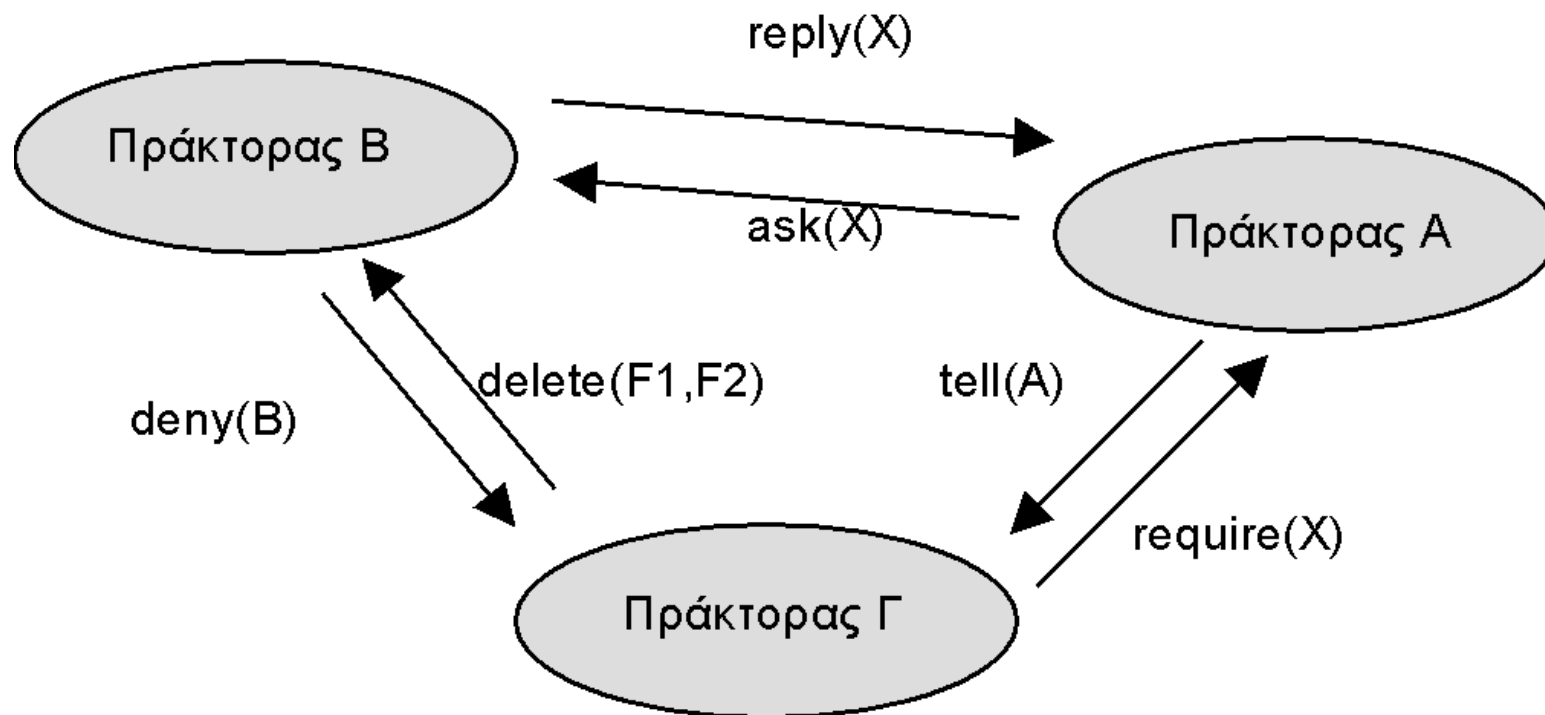
Συστήματα Μαυροπίνακα (Blackboard)

- ❖ Κοινός χώρος εργασίας για όλους τους πράκτορες του συστήματος
 - ❑ Προσπελάσιμος από όλους τους πράκτορες που συμμετέχουν στο σύστημα.
 - ❑ Ανταλλαγή αποτελεσμάτων ή διαμοιρασμός εργασιών.



Συστήματα Ανταλλαγής Μηνυμάτων (1/2)

- ❖ Ανταλλαγή πληροφορίας και συνεργασία μέσω μηνυμάτων
 - Αποστολή μηνυμάτων βάσει συγκεκριμένων γλωσσών υψηλού επιπέδου.
- ❖ Επιτρέπουν την υλοποίηση πολύπλοκων μοντέλων συνεργασίας μεταξύ των πρακτόρων
 - Προσφέρουν μεγαλύτερη ευελιξία στην ανταλλαγή πληροφοριών από ότι τα συστήματα μαυροπίνακα.



Συστήματα Ανταλλαγής Μηνυμάτων (2/2)

Τύπος Επικοινωνίας (Communication Type)

❖ Σύγχρονος

- ❑ ο πράκτορας που θέτει μια ερώτηση διακόπτει τη λειτουργία του μέχρι να πάρει μια απάντηση.

❖ Ασύγχρονος

- ❑ η απάντηση μπορεί να έρθει οποιαδήποτε στιγμή μετά το χρόνο υποβολής της ερώτησης, χωρίς να διακόπτεται η λειτουργία του πράκτορα.

Βαθμός Επικοινωνίας (Cardinality)

❖ Ο αριθμός των αποστολέων και αποδεκτών σε μια ανταλλαγή πληροφορίας.

- ❑ 1 προς 1 (μόνο δύο πράκτορες συμμετέχουν σε κάθε ανταλλαγή πληροφορίας)
- ❑ 1 προς N (ένας αποστολέας πολλοί αποδέκτες)
- ❑ N προς N (πολλοί αποστολείς και πολλοί αποδέκτες)



Πρωτόκολλα Επικοινωνίας (communication protocols)

- ❖ Όποιο και αν είναι το μοντέλο διασύνδεσης που υιοθετεί ένα πολυπρακτορικό σύστημα, απαραίτητη προϋπόθεση είναι η ύπαρξη δύο πρωτοκόλλων:
 - ❖ **Πρωτόκολλο επικοινωνίας** (*communication protocol*)
 - ❑ Καθορίζει τη μορφή των μηνυμάτων (μήκος, επιτρεπτά σύμβολα) και τη σημασία τους.
 - ❖ **Πρωτόκολλο αλληλεπίδρασης** (*interaction protocol*)
 - ❑ Δίνει τη δυνατότητα στους πράκτορες να έχουν συζητήσεις (conversations), δηλαδή ακολουθίες ανταλλαγής μηνυμάτων.

Πρωτόκολλα Επικοινωνίας

- ❖ Καθορίζουν:
 - ❑ τύπο και μορφή των μηνυμάτων
 - ❑ σημασιολογία των μηνυμάτων
 - ❑ τρόπο διασύνδεσης των μηνυμάτων

- ❖ KQML (Knowledge and Query Manipulation Language)
 - ❑ Γλώσσα επικοινωνίας πρακτόρων (Agent Communication Languages).
 - ❑ Βασίζεται στη θεωρία πράξεων λόγου (speech acts) που έχει προταθεί από το γλωσσολόγο Austin το 1962

- ❖ Η αρχική έλλειψη αυστηρά καθορισμένης σημασιολογίας έδωσε το έναυσμα για την εισαγωγή μιας εναλλακτικής γλώσσας επικοινωνίας πρακτόρων, της FIPA ACL.
 - ❑ FIPA: Foundation for Intelligent Physical Agents
 - ❑ Standards for interoperability among software agent platforms
 - ❑ Παρουσιάζει σημαντικές ομοιότητες με την KQML

- ❖ Οι διαφορές των δύο γλωσσών επικεντρώνονται στο ότι η FIPA ACL παρέχει:
 - ❑ ένα κλειστό σύνολο είκοσι δύο επικοινωνιακών πράξεων /δηλώσεων
 - ❑ οι δηλώσεις έχουν σαφή αυστηρά καθορισμένη σημασιολογία.

Θεωρία Πράξεων Λόγου (Speech Acts)

❖ Τα μηνύματα που ανταλλάσσονται:

- ❑ Αποτελούν ουσιαστικά πράξεις/ενέργειες όμοιες με εκείνες του φυσικού κόσμου
- ❑ Μπορούν να εξεταστούν από τρεις σκοπιές:
 - τη σύνταξη του μηνύματος (locution),
 - τον τύπο του μηνύματος, δηλαδή αν είναι ενημέρωση, προειδοποίηση ή αίτημα (illocution)
 - το αποτέλεσμα του μηνύματος στον παραλήπτη (perlocution).

❖ Οι τύποι των μηνυμάτων κατηγοριοποιούνται χρησιμοποιώντας δηλώσεις.

Κατηγορία	Επεξήγηση	Παραδείγματα
Assertives	Δηλώσεις κάποιου γεγονότος, στις οποίες ο ομιλητής δεσμεύεται στην αλήθεια τους.	πληροφορώ, υποθέτω, επιμένω (<i>informing, hypothesizing, insisting</i>)
Directives	Πράξεις τις οποίες απευθύνει ο ομιλητής στον ακροατή με σκοπό ο τελευταίος να κάνει μια ενέργεια.	παρακαλώ, προστάζω (<i>begging, ordering</i>)
Commissives	Δεσμεύσεις του ομιλητή για κάποια μελλοντική ενέργεια	υπόσχομαι, αναλαμβάνω (<i>promising, undertaking</i>)
Declaratives	Δηλώσεις γεγονότων που κάνουν αληθές το γεγονός	κηρύσσω πόλεμο, παντρεύομαι (<i>declaring war, marrying</i>)
Expressives	Εκδηλώσεις συναισθημάτων του ομιλητή	ευχαριστώ, καλωσορίζω (<i>thanking, wellcoming</i>)

Ρόλος των Δηλώσεων στην Σημασιολογία του Μηνύματος

- ❖ Η κατηγοριοποίηση βοηθά στο να είναι ξεκάθαρο κάθε φορά ποια είναι η πρόθεση (τύπος) του μηνύματος.
- ❖ Παράδειγμα, έστω ένα μήνυμα κωδικοποιημένο στην γλώσσα PROLOG, το οποίο έχει σαν περιεχόμενο το "**work (complete)**".
- ❖ Οι πιθανές ερμηνείες του μηνύματος βάσει του τύπου είναι:

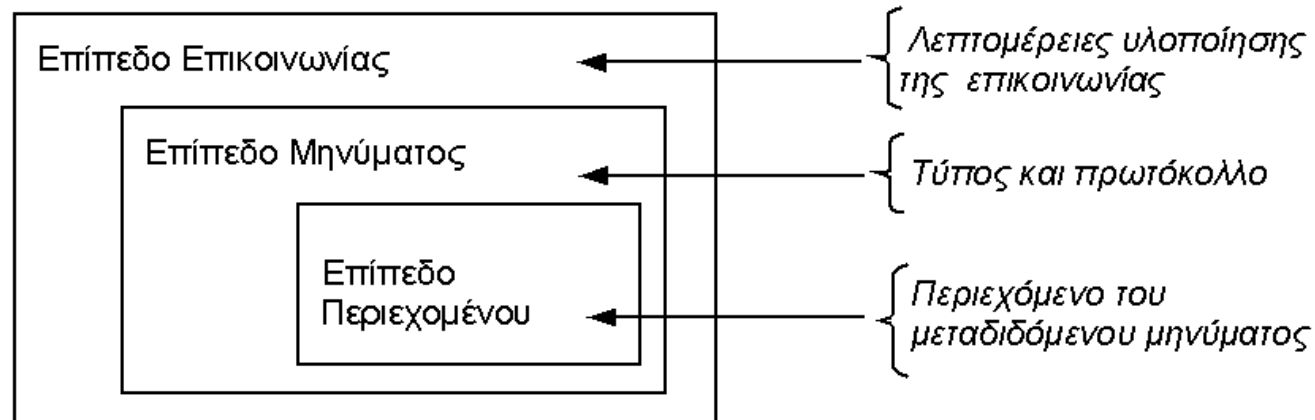
Δήλωση	Περιεχόμενο	Ερμηνεία
requesting	work(complete)	Αίτηση για την ολοκλήρωση της εργασίας
informing	work(complete)	Πληροφόρηση ότι η εργασία έχει ολοκληρωθεί
promising	work(complete)	Υπόσχεση για την ολοκλήρωση της εργασίας.

- ❖ Η ύπαρξη του τύπου
 - ❑ Περιορίζει σημαντικά το εύρος των ερμηνειών.
 - ❑ Καθορίζει επίσης σε σημαντικό βαθμό τον τρόπο με τον οποίο θα πρέπει να αντιδράσει ο πράκτορας στο μήνυμα βάσει του πρωτοκόλλου.
- ❖ Η πλήρης κατανόηση του μηνύματος απαιτεί βέβαια όχι μόνο τον τύπο του μηνύματος αλλά και ορισμό της σημασιολογίας του περιεχομένου του.
- ❖ Ακολουθώντας την θεωρία πράξεων λόγου, τόσο η KQML όσο και η FIPA ACL ορίζουν στην σύνταξη τους τον τύπο του μηνύματος.

Το Πρωτόκολλο KQML

❖ Τρία διαφορετικά επίπεδα.

- ❑ Το επίπεδο περιεχομένου (content layer), που αφορά το περιεχόμενο του μεταδιδόμενου μηνύματος.
- ❑ Το επίπεδο μηνύματος (message layer), το οποίο είναι ο πυρήνας της γλώσσας. Στο επίπεδο αυτό καθορίζεται
 - ο τύπος του μηνύματος, με κάποια από τις διαθέσιμες δηλώσεις (*performative*)
 - το πρωτόκολλο (protocol), δηλαδή ο τρόπος μετάδοσης και λήψης των μηνυμάτων, όπως για παράδειγμα σύγχρονος, ασύγχρονος, κλπ.
- ❑ Το επίπεδο επικοινωνίας (communication layer), στο οποίο καθορίζονται οι χαμηλού επιπέδου λεπτομέρειες υλοποίησης της επικοινωνίας.



Μορφή KQML Μηνυμάτων

❖ Τα μηνύματα στο KQML την ακόλουθη μορφή:

(`<performative>`

`:sender` `<ο αποστολέας>`

`:receiver` `<ο παραλήπτης>`

`:language` `<η γλώσσα στην οποία έχει κωδικοποιηθεί το περιεχόμενο>`

`:ontology` `<η οντολογία είναι το λεξιλόγιο το οποίο πρέπει να χρησιμοποιηθεί για να γίνει κατανοητό το μήνυμα>`

`:content` `<το περιεχόμενο του μηνύματος>`

)

Κατηγορίες Δηλώσεων (performative) στην KQML

- ❖ Χαρακτηρίζουν τον τύπο του μεταδιδόμενου μηνύματος

Κατηγορία Δηλώσεων	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

Τα πεδία **language** και **ontology**

- ❖ Το πεδίο **language** δηλώνει τη γλώσσα στην οποία έχει κωδικοποιηθεί το μήνυμα.
 - ❑ Για παράδειγμα PROLOG, LISP, SQL, κλπ, ή και οποιαδήποτε άλλη γλώσσα επικοινωνίας μεταξύ των πρακτόρων.
 - ❑ Υπάρχει ενδιαφέρον για ανάπτυξη μιας γλώσσας η οποία θα επιτρέψει την ανταλλαγή γνώσης μεταξύ των πρακτόρων.
 - ❑ Γλώσσα KIF (Knowledge Interchange Format)
 - Βασισμένη στη λογική
 - Εκφραστική
 - Έχει μια δομή που επιτρέπει την ημιαυτόματη μετάφραση από και προς γνωστές γλώσσες αναπαράστασης.
- ❖ Το πεδίο **ontology** καθορίζει το λεξιλόγιο που χρησιμοποιείται στο συγκεκριμένο πεδίο το οποίο αφορά το μήνυμα ώστε όλοι οι πράκτορες που συμμετέχουν στο σύστημα αποδίδουν στις λέξεις την ίδια σημασία.
 - ❑ Για παράδειγμα, η λέξη *ποντίκι* έχει διαφορετική σημασία και ιδιότητες όταν αναφερόμαστε στο πεδίο των ηλεκτρονικών υπολογιστών και στο πεδίο των προβλημάτων παλαιών κτισμάτων.

Παράδειγμα Μηνύματος KQML

- ❖ Μήνυμα-ερώτηση για την τιμή της μετοχής της εταιρίας IBM στο χρηματιστήριο.
- ❖ Ο πράκτορας που ρωτά ονομάζεται stock-client
- ❖ Ο πράκτορας που απαντά stock-server
- ❖ Η γλώσσα στην οποία είναι κωδικοποιημένο το μήνυμα είναι η Prolog,
- ❖ Το λεξιλόγιο (ontology) που θα χρησιμοποιηθεί αφορά το χρηματιστήριο της Ν. Υόρκης (NYSE-TICKS)

(ask

:sender stock-client

:receiver stock-server

:language standard Prolog

:ontology NYSE-TICKS

:content price("IBM", Price)

)

Παράδειγμα Άμεσης Επικοινωνίας

- ❖ Έστω ότι ο **AgentA** δηλώνει στον **AgentB** το γεγονός ότι υπάρχει ένα εμπόδιο στο δωμάτιο **roomA** στις συντεταγμένες (5,5).

(tell

:sender AgentA

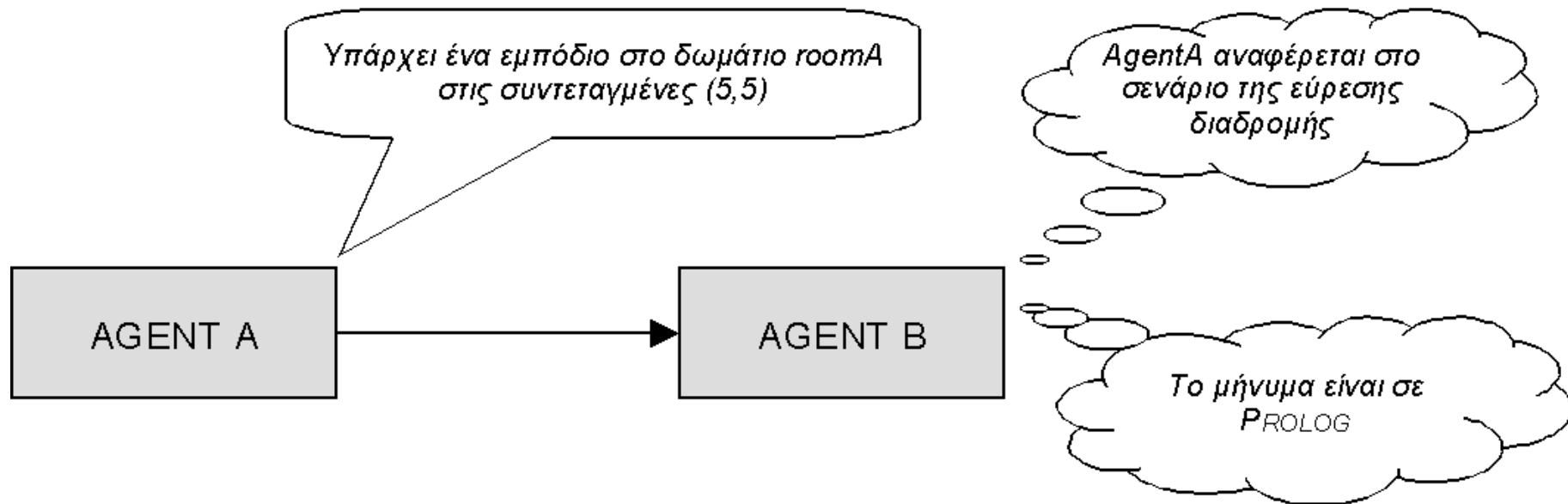
:receiver AgentB

:language Prolog

:ontology path-finding

:content obstacle_at(roomA, (5,5))

)



Παράδειγμα Έμμεσης Επικοινωνίας

- ❖ Ο **AgentA** επικοινωνεί με τον **AgentB** για να στείλει αυτός ένα μήνυμα στον πράκτορα **AgentC**, που να δηλώνει το γεγονός ότι υπάρχει ένα εμπόδιο στο δωμάτιο roomA στις συντεταγμένες (5,5).

(forward

:from AgentA

:to AgentC

:sender AgentA

:receiver AgentB

:language KQML

:ontology KQML-ontology

:content

(tell

:sender AgentA

:receiver AgentC

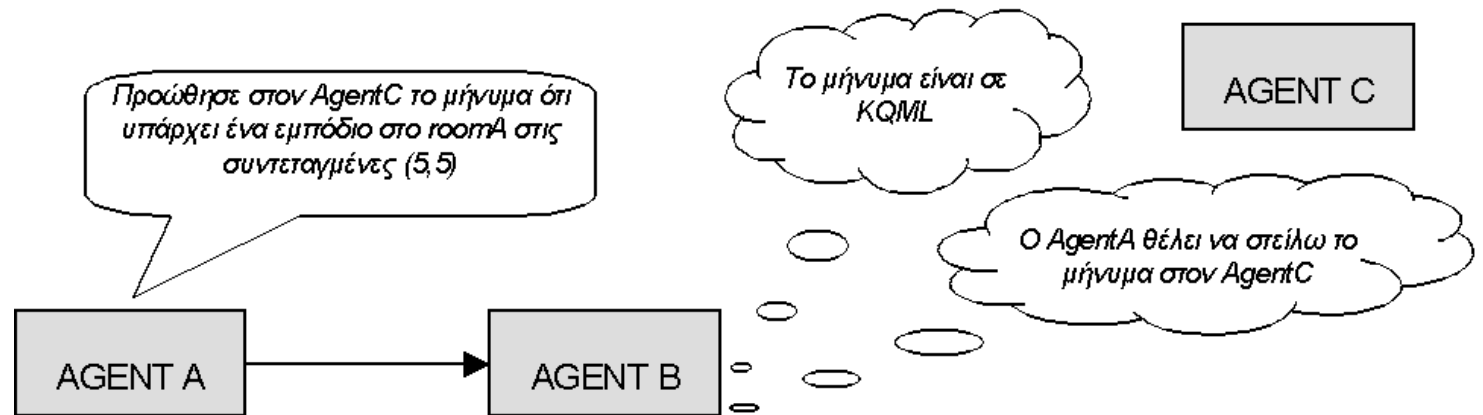
:language Prolog

:ontology path-finding

:content obstacle_at(roomA, (5,5))

)

)



Παράδειγμα Δήλωσης Ικανότητας

- ❖ Ο **AgentA** πληροφορεί τον επιβλέποντα πράκτορα (supervisor) για την ικανότητά του να κάνει αναζήτηση με τον αλγόριθμο επαναληπτικής εμφάνισης (ID) σε Prolog για την εύρεση μιας διαδρομής.

(advertise

:sender AgentA

:receiver Supervisor

:language KQML

:ontology KQML-ontology

:content

(ask-one

:sender Supervisor

:receiver AgentA

:in-reply-to message1

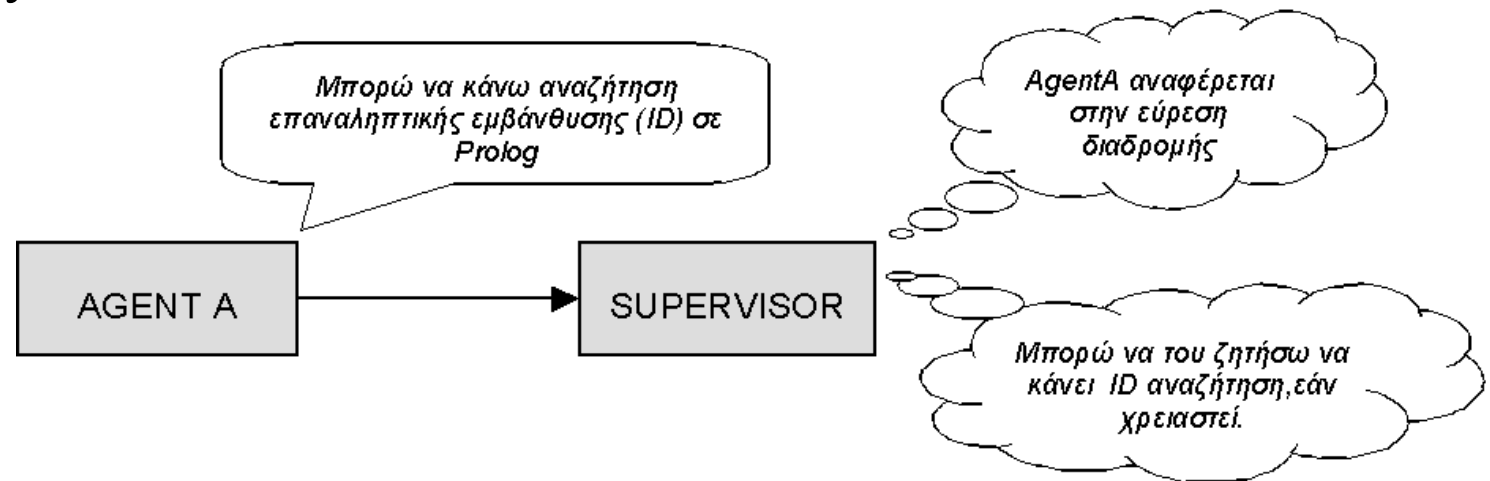
:language Prolog

:ontology path-finding

:content goid(AgentA, IS, FS, Route)

)

)



Μήνυμα για Αναζήτηση Διαδρομής

- ❖ Έτσι όταν παραστεί η ανάγκη για τέτοιου είδους αναζήτηση ο επιβλέπων στέλνει το ακόλουθο μήνυμα στο πράκτορα **AgentA**

(ask-one

:sender Supervisor

:receiver AgentA

:in-reply-to message1

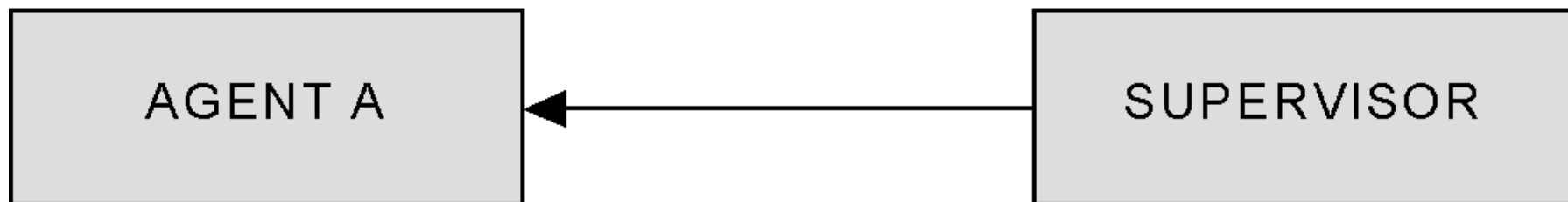
:reply-with message2

:language Prolog

:ontology path-finding

:content goid(AgentA, (3,3), (5,5), Route))

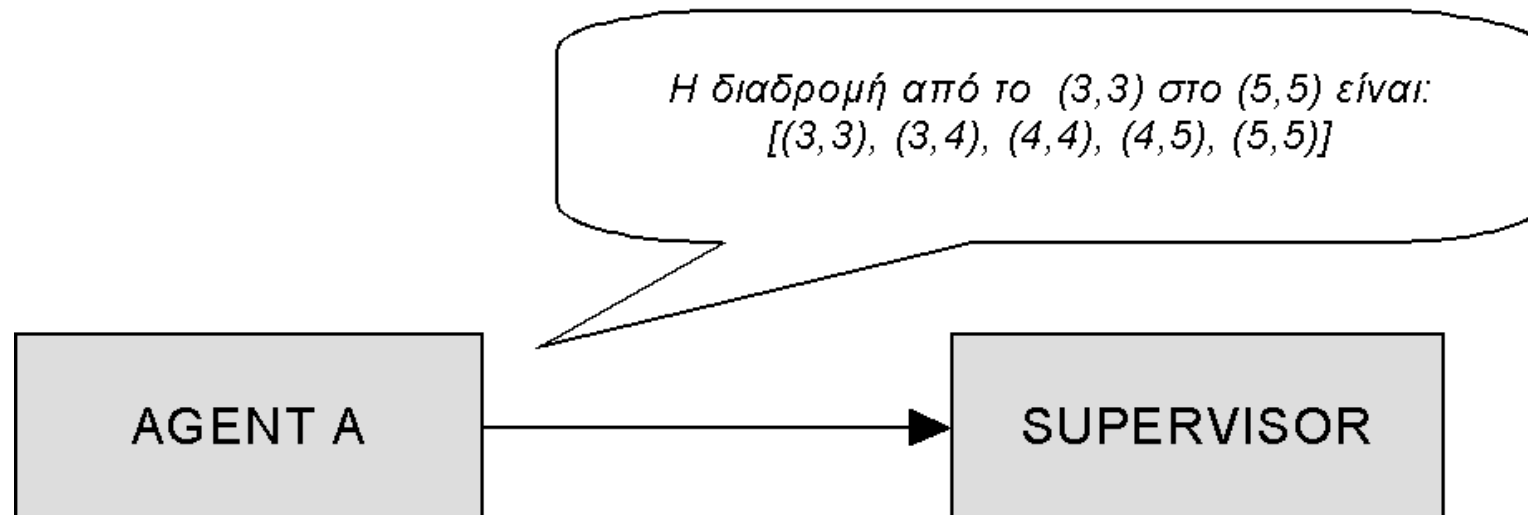
*Εφάρμοσε Iterative Deepening για να βρεις την
διαδρομή από το (3,3) στο (5,5)*



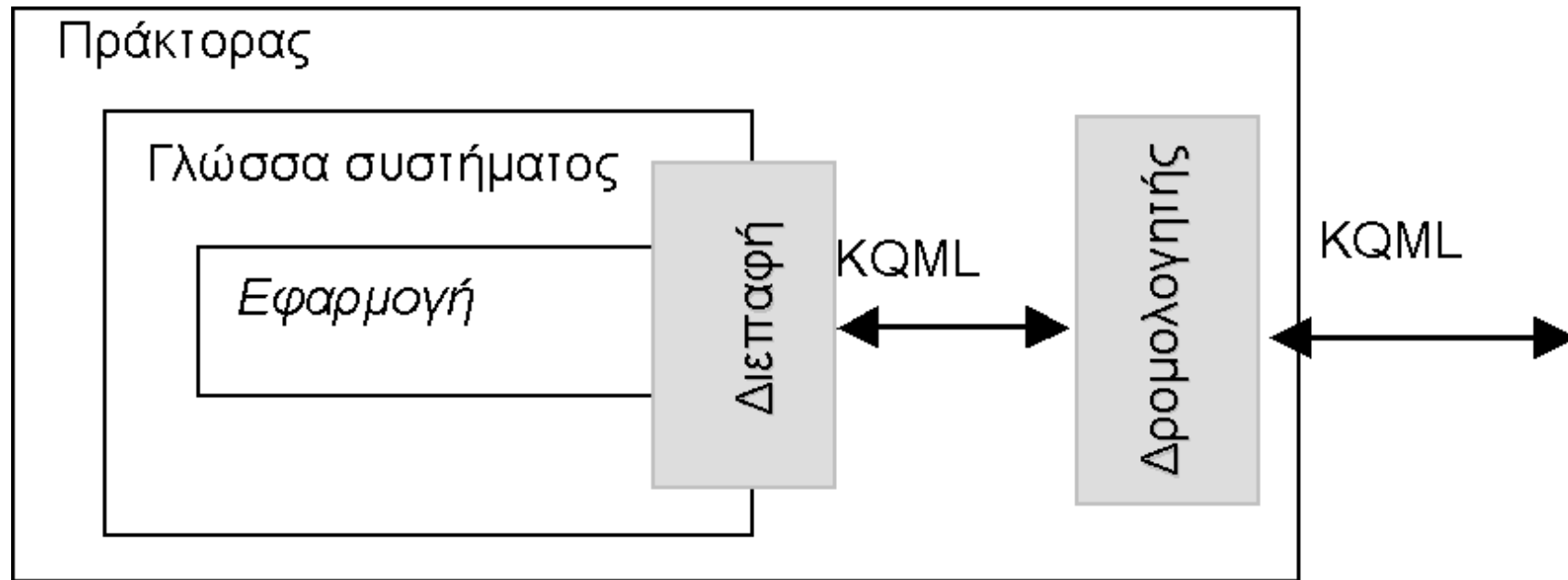
Μήνυμα Αποτελεσμάτων της Διαδρομής

- ❖ Εφόσον ο **AgentA** ολοκλήρωσε την αναζήτηση στέλνει στον επιβλέποντα το ακόλουθο μήνυμα με τα αποτελέσματα αυτής:

```
(tell  
  :sender AgentA  
  :receiver Supervisor  
  :in-reply-to message2  
  :language Prolog  
  :ontology path-finding  
  :content [(3,3), (3,4), (4,4), (4,5), (5,5)]  
)
```



Αρχιτεκτονική Πράκτορα με το πρωτόκολλο KQML



- ❖ Απαιτείται η ύπαρξη δύο στοιχείων (components):
 - ❑ Διεπαφής (interface) ανάμεσα στην KQML και στη γλώσσα συστήματος του πράκτορα
 - ❑ Δρομολογητή (router) ο οποίος αναλαμβάνει τη διαχείριση όλων των χαμηλού επιπέδου λειτουργιών επικοινωνίας.
- ❖ Ο τρόπος επικοινωνίας μεταξύ των πρακτόρων είναι ένα από τα πιο ενεργά ερευνητικά πεδία.

Επιτυχία και Κριτική της KQML

- ❖ Η γλώσσα KQML αποτέλεσε την πρώτη επιτυχημένη προσπάθεια για τον καθορισμό μιας γλώσσας επικοινωνίας των πρακτόρων
 - ❑ Έτυχε ευρείας αποδοχής.
 - ❑ Υπάρχει πληθώρα συστημάτων τα οποία χρησιμοποίησαν την KQML για την υλοποίηση της επικοινωνίας.
- ❖ Η κύρια κριτική στην KQML επικεντρώθηκε στο γεγονός ότι τόσο η γλώσσα και ιδιαίτερα
 - ❑ Οι δηλώσεις της γλώσσας (performatives) δεν είχαν σαφώς καθορισμένη σημασιολογία, αλλά η σημασιολογία τους δίδονταν από λεκτικές περιγραφές εκφρασμένες στα Αγγλικά.
 - ❑ Το σύνολο των δηλώσεων της γλώσσας είναι ανοικτό, δηλαδή μπορεί να επεκταθεί για να καλύψει πιθανές ανάγκες της εκάστοτε εφαρμογής που αναπτύσσεται.
 - Το τελευταίο αν και προσθέτει σημαντική ευελιξία, μπορεί να οδηγήσει σε καταστάσεις αδυναμίας επικοινωνίας πρακτόρων οι οποίοι υλοποιήθηκαν από διαφορετικούς φορείς.
- ❖ Το πρόβλημα ορισμού της σημασιολογίας της KQML αντιμετωπίστηκε σε ένα βαθμό από μερικούς ερευνητές
- ❖ Η αρχική έλλειψη αυστηρά καθορισμένης σημασιολογίας έδωσε το έναυσμα για την εισαγωγή μιας εναλλακτικής γλώσσας επικοινωνίας πρακτόρων, της FIPA ACL.
 - ❑ FIPA: Foundation for Intelligent Physical Agents
 - ❑ Standards for interoperability among software agent platforms

Η γλώσσα επικοινωνίας FIPA ACL

- ❖ Η FIPA ACL παρουσιάζει σημαντικές ομοιότητες με την KQML.
 - ❑ Ακολουθεί παρόμοια μορφή μηνυμάτων.
 - ❑ Χρησιμοποιεί δηλώσεις για τον καθορισμό της επικοινωνιακής πράξης του κάθε μηνύματος.
 - ❑ Υπάρχει πλήρης απουσία καθορισμού της γλώσσας στην οποία εκφράζεται το περιεχόμενο του μηνύματος.
- ❖ Οι διαφορές των δύο γλωσσών επικεντρώνονται στο ότι η FIPA ACL παρέχει:
 - ❑ ένα κλειστό σύνολο είκοσι δύο επικοινωνιακών πράξεων /δηλώσεων
 - ❑ οι δηλώσεις έχουν σαφή αυστηρά καθορισμένη σημασιολογία.

```
(inform
 :sender agentI
 :receiver agentJ
 :content "direction(west, speed(34))"
 :language Prolog)
```

Κατηγορία	Επικοινωνιακή Πράξη (communicative act)
Μετάδοση Πληροφορίας (Information passing)	confirm, disconfirm, inform, inform-if, inform-ref
Αίτηση για Πληροφορία (Requesting information)	query-if, query-ref, subscribe
Διαπραγμάτευση (Negotiation)	accept-proposal, cfp, propose, reject-proposal
Εκτέλεση Ενεργειών (Action performing)	agree, cancel, refuse, request, request-when, request-whenever
Μηνύματα Λάθους (Error handling)	failure, not-understood

Σημασιολογία FIPA ACL

- ❖ Η σημασιολογία των δηλώσεων της γλώσσας ορίζεται χρησιμοποιώντας την γλώσσα SL, που επιτρέπει την αναπαράσταση πεποιθήσεων, επιθυμιών, προθέσεων των πρακτόρων, είναι δηλαδή στενά συνδεδεμένη με το BDI μοντέλο πρακτόρων,
- ❖ Για κάθε επικοινωνιακή πράξη ορίζονται χρησιμοποιώντας τύπους της SL
 - ❑ Ποιες είναι οι προϋποθέσεις (feasibility conditions - FP) οι οποίες πρέπει να ισχύουν στον πράκτορα που πρέπει να αποστέλλει το μήνυμα
 - ❑ Ποιο είναι το αναμενόμενο αποτέλεσμα (rational effect - RE) στον πράκτορα που λαμβάνει το μήνυμα.
- ❖ Για παράδειγμα σύμφωνα με το πρότυπο της FIPA ACL για την επικοινωνιακή πράξη inform (πληροφορώ), οι προϋποθέσεις που πρέπει να ισχύουν για τον πράκτορα αποστολέα του μηνύματος με περιεχόμενο μια πρόταση (proposition) φ είναι:
 - ❑ να πιστεύει ότι η πρόταση φ είναι αληθής.
 - ❑ να έχει την πρόθεση ο παραλήπτης να αποκτήσει την πεποίθηση ότι η φ είναι αληθής.
 - ❑ να μην πιστεύει ότι ο παραλήπτης έχει κάποια γνώση για την φ . Το τελευταίο σημαίνει ότι δεν είναι "λογικό" για τον πράκτορα να υιοθετήσει την πρόθεση να επικοινωνήσει γνώση για την οποία πιστεύει ότι ο παραλήπτης έχει κάποιου είδους πεποίθηση.

Εκφράζοντας σημασιολογία σε SL

- ❖ Θεωρώντας ότι
 - ❑ i είναι ο αποστολέας του μηνύματος
 - ❑ j είναι ο παραλήπτης,
 - ❑ $B_i\phi$ και $B_j\phi$ σημαίνουν ότι οι πράκτορες i και j πιστεύουν αντίστοιχα τη ϕ ως αληθή.
 - ❑ $Bif_j\phi$ σημαίνει ότι ο πράκτορας j είτε πιστεύει την ϕ είτε την άρνηση της, δηλαδή $Bif_j\phi \equiv B_j\phi \vee B_j\neg\phi$.
 - ❑ $Uif_j\phi$ σημαίνει ότι ο πράκτορας έχει αβέβαιη γνώση για την αλήθεια της ϕ , δηλαδή $Uif_j\phi \equiv U_j\phi \vee U_j\neg\phi$.
- ❖ Ο τύπος της γλώσσα SL που εκφράζει τα παραπάνω είναι ο ακόλουθος.
 - $\langle i, \text{INFORM} (j, \phi) \rangle$
 - FP:** $B_i\phi \wedge \neg B_i(Bif_j\phi \vee Uif_j\phi)$
 - RE:** $B_j\phi$
- ❖ Το πρότυπο ορίζει με αρκετή αυστηρότητα το ποια επικοινωνιακή πράξη πρέπει να χρησιμοποιηθεί σε κάθε περίπτωση.
 - ❑ Σημαντικό πρόβλημα που ανακύπτει κατά την υλοποίηση είναι το πώς θα επαληθευτεί ότι ο πράκτορας που επικοινωνεί βάσει των προδιαγραφών της γλώσσας, όντως ακολουθεί την συγκεκριμένη σημασιολογία (conformance testing).
- ❖ Η γλώσσα SL μπορεί να χρησιμοποιηθεί και για την κωδικοποίηση του περιεχομένου του μηνύματος.

Πρωτόκολλα Αλληλεπίδρασης

- ❖ Δίνουν τη δυνατότητα στους πράκτορες να ανταλλάσσουν ακολουθίες μηνυμάτων, να έχουν δηλαδή συζητήσεις (conversations), με απώτερο σκοπό τη συνεργασία τους.

Συντονισμός

Ο συντονισμός στα πολυπρακτορικά συστήματα είναι η διαδικασία μέσω της οποίας επιτυγχάνεται η λογικά συνεπής λειτουργία ολόκληρου του συστήματος.

- ❖ Είναι απαραίτητη για:
 - Την αποφυγή καταστάσεων χάους.
 - Την ικανοποίηση καθολικών περιορισμών.
 - Την εκμετάλλευση κατανεμημένης εμπειρογνωμοσύνης.
 - Την αύξηση της αποδοτικότητας.
 -
- ❖ Ο συντονισμός μεταξύ πρακτόρων:
 - με κοινό στόχο αναφέρεται ως *συνεργασία* (cooperation),
 - με αλληλοσυγκρουόμενους προσωπικούς στόχους ως *διαπραγμάτευση* (negotiation) ή ανταγωνισμός (competition).
 - Κοινό χαρακτηριστικό: απαιτείται η εξασφάλιση της λογικά συνεπούς λειτουργίας του συστήματος.

Κεντρικός Ρόλος Δεσμεύσεων και Συμβάσεων

- ❖ Μια θεώρηση του συντονισμού περιλαμβάνει την έκφραση των σχέσεων μεταξύ των πρακτόρων ενός συστήματος σαν ένα σύνολο δεσμεύσεων (commitments) και συμβάσεων (convention).
 - ❑ Οι δεσμεύσεις αποτελούν αναλήψεις εργασιών από ένα πράκτορα ως "υπόσχεση" προς κάποιον άλλο, ή ανάληψη μιας εργασίας από μια ομάδα πρακτόρων, όποτε και ονομάζονται κοινές δεσμεύσεις (joint commitments).
 - ❑ Οι συμβάσεις αποτελούν τον τρόπο διαχείρισης και εποπτείας των δεσμεύσεων, με άλλα λόγια καθορίζουν το πως θα πρέπει να αντιδράσει ο πράκτορας σε περίπτωση που το περιβάλλον αλλάξει ή απλά η εργασία ολοκληρωθεί.
- ❖ Υπόθεση του "κεντρικού ρόλου των δεσμεύσεων και συμβάσεων" (centrality of commitments and conventions hypothesis) που διατυπώθηκε από τον N. Jennings:

"Όλοι οι μηχανισμοί συντονισμού μπορούν να αναχθούν σε (κοινές) δεσμεύσεις και τις αντίστοιχες τους (κοινωνικές) συμβάσεις".

Κατηγορίες Τεχνικών

- ❖ Κατηγορίες τεχνικών για το συντονισμό μεταξύ των πρακτόρων
 - ❑ Τεχνικές βασισμένες στην οργάνωση (organizational structure)
 - ❑ Τεχνικές σύναψης συμβολαίων (contracting net)
 - ❑ Τεχνικές πολυπρακτορικού σχεδιασμού (multiagent planning)
 - ❑ Τεχνικές διαπραγμάτευσης (negotiation)

Τεχνικές Βασισμένες στην Οργάνωση

- ❖ Προϋποθέτουν μια οργανωτική δομή των πρακτόρων.
- ❖ Κάθε πράκτορας έχει:
 - ❑ Συγκεκριμένο ρόλο
 - ❑ Υπευθυνότητες
 - ❑ Ικανότητες.
- ❖ Αρχιτεκτονική κατανομής εργασίας (master-slave) η οποία μπορεί να υλοποιηθεί με τους ακόλουθους δύο τρόπους:
 - ❑ Ένας **κεντρικός πράκτορας** αναλαμβάνει το χωρισμό του προβλήματος σε υποπροβλήματα
 - Διαμοιρασμός σε άλλους πράκτορες (slaves).
 - Επιστροφή αποτελεσμάτων στον κεντρικό πράκτορα.
 - ❑ Μέσω ενός **συστήματος μαυροπίνακα** στο οποίο αποθηκεύονται τόσο τα υποπροβλήματα που πρέπει να επιλυθούν, όσο και τα αποτελέσματα. Συντονισμός μεταξύ των πρακτόρων επιτυγχάνεται είτε:
 - μέσω ενός πράκτορα (scheduling agent) ο οποίος ελέγχει την προσπέλαση των άλλων πρακτόρων στο μαυροπίνακα,
 - ή έχοντας καθορίσει σαφώς ποιες είναι οι αρμοδιότητες του κάθε πράκτορα που συμμετέχει στο σύστημα.

Τεχνικές Σύναψης Συμβολαίων

- ❖ Βασίζονται στη δημιουργία ενός δικτύου σύναψης συμβολαίων (contract net).
- ❖ Στο μοντέλο αυτό κάθε πράκτορας μπορεί να αναλάβει δύο ρόλους:
 - ❑ Το ρόλο του διαχειριστή (manager)
 - ❑ Το ρόλο του εργολάβου (contractor)
- ❖ Ο Διαχειριστής
 - ❑ Χωρίζει το πρόβλημα σε υποπροβλήματα.
 - ❑ Αναλαμβάνει να τα αναθέσει στους εργολάβους (contractors).
 - ❑ Επιβλέπει την πορεία της λύσης
- ❖ Ο Εργολάβος
 - ❑ Αναλαμβάνει να λύσει ένα υποπρόβλημα.
 - ❑ Οι εργολάβοι μπορούν να χωρίσουν το υποπρόβλημα που έχουν αναλάβει σε περισσότερα υποπροβλήματα και με τη σειρά τους να το αναθέσουν σε άλλους πράκτορες.
- ❖ Στο μοντέλο αυτό, αντίθετα με εκείνο των τεχνικών βασισμένων στην οργάνωση, κάθε πράκτορας μπορεί να είναι ταυτόχρονα διαχειριστής και εργολάβος.



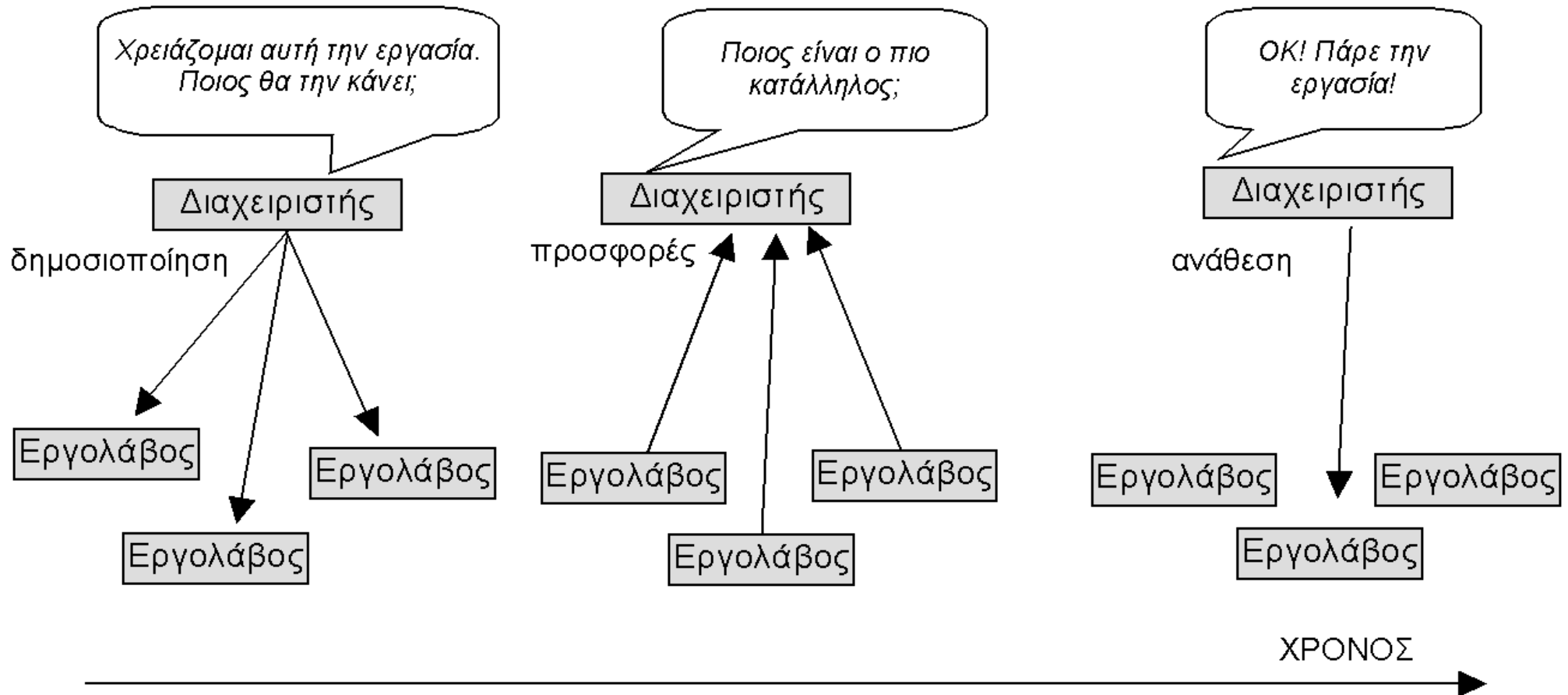
Διαδικασία Ανάθεσης

Η διαδικασία ανάθεσης των υποπροβλημάτων περιλαμβάνει:

- ❖ Τη δημοσιοποίηση τους (announcement) από το διαχειριστή σε όλους ή σε μερικούς από τους πράκτορες.
- ❖ Αξιολόγηση από τους αποδέκτες των δημοσιοποιήσεων των υποπροβλημάτων ανάλογα με την ικανότητα τους
- ❖ Αποστολή από τους αποδέκτες στο διαχειριστή
 - προσφορών (bids) ή
 - απορρίψεων (declination)
- ❖ Συλλογή από τον διαχειριστή των προσφορών (bids) από τους υποψήφιους εργολάβους
- ❖ Αξιολόγηση των προσφορών
- ❖ Κατάλληλες αναθέσεις (awards).

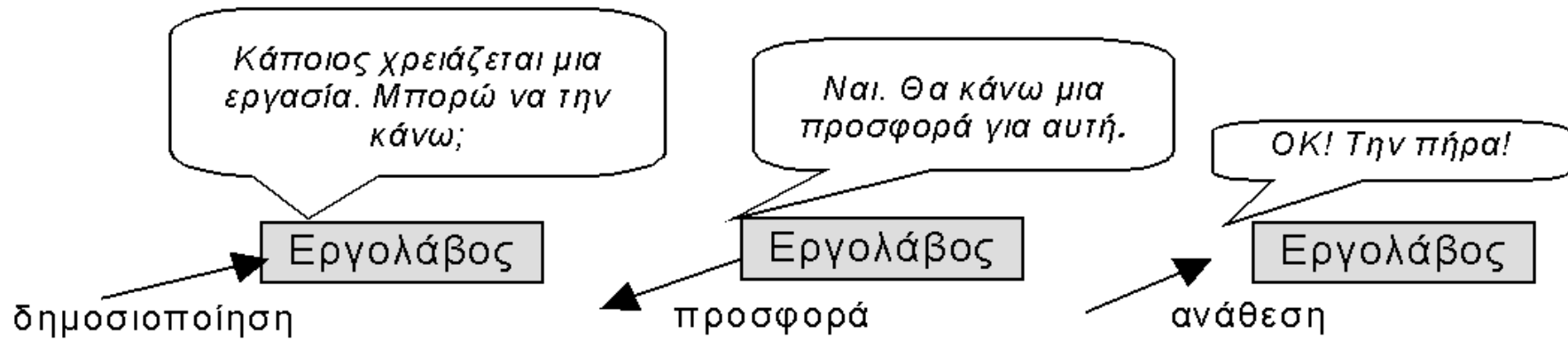
Ανάθεση προβλημάτων (1/2)

❖ Από την πλευρά του διαχειριστή



Ανάθεση προβλημάτων (2/2)

- ❖ Αποδοχή (α) και απόρριψη (β) μιας δημοσιοποίησης από ένα πράκτορα.

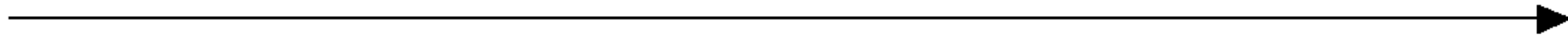


(α)



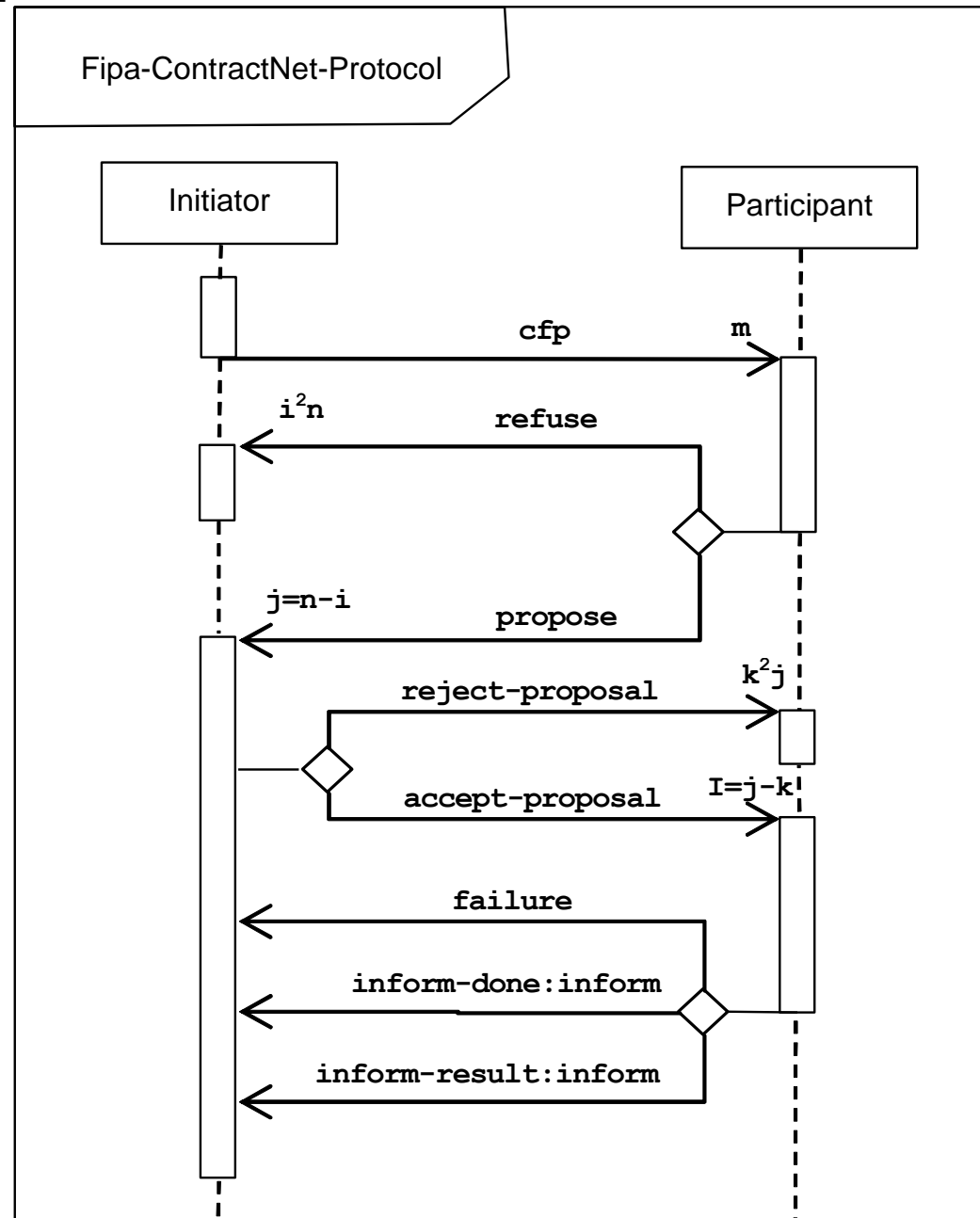
(β)

ΧΡΟΝΟΣ



Σύναψη Συμβολαίων σε FIPA ACL

- ❖ Η τεχνική ανάθεσης συμβολαίων αποτελεί ίσως το πλέον διαδομένο πρωτόκολλο αλληλεπίδρασης πρακτόρων για διαμοιρασμό προβλημάτων.
- ❖ Η μεγάλη της διάδοση της τεχνικής οδήγησε τον οργανισμό FIPA να συμπεριλάβει στα πρότυπα του προδιαγραφές του πρωτοκόλλου χρησιμοποιώντας την γλώσσα FIPA ACL.





Πολυπρακτορικός Σχεδιασμός

- ❖ Οι πράκτορες συντάσσουν ένα πλάνο ενεργειών, βάσει του οποίου θα επιλύσουν το πρόβλημα.
- ❖ Δύο τύποι πολυπρακτορικού σχεδιασμού:
 - ❑ Κεντρικός πολυπρακτορικός σχεδιασμός
 - Ένας κεντρικός πράκτορας αναλαμβάνει τη συλλογή των επιμέρους πλάνων (partial or local plans) από τους πράκτορες, τα συνδυάζει σε ένα πλάνο και επιλύει τυχόν συγκρούσεις που θα εμφανιστούν σε αυτό.
 - ❑ Κατανεμημένος πολυπρακτορικός σχεδιασμός
 - Οι πράκτορες επικοινωνούν μεταξύ τους για τη δημιουργία των πλάνων τους και την επίλυση τυχόν συγκρούσεων που προκύπτουν.

Διαπραγμάτευση

- ❖ Η διαπραγμάτευση (*negotiation*) αφορά το πώς πράκτορες με προσωπικούς, πιθανά αλληλοσυγκρουόμενους/ανταγωνιστικούς στόχους θα φτάσουν σε κοινά ωφέλιμες συμφωνίες.
- ❖ Μεγάλος αριθμός μεθόδων για διαπραγμάτευση:
 - ❑ Μέθοδοι που βασίζονται στη θεωρία παιχνιδιών (*game theory based negotiation*),
 - ❑ Μέθοδοι που βασίζονται σε δημιουργία πλάνων
 - ❑ Μέθοδοι που βασίζονται στην κατανομημένη ικανοποίηση περιορισμών (*distributed constraint satisfaction*).

Πρωτόκολλα πλειστηριασμού

- ❖ Μια από τις απλούστερες κατηγορίες πρωτοκόλλων διαπραγμάτευσης.
- ❖ Οι πλειστηριασμοί (*auctions*) αποτελούν πρωτόκολλα μέσω των οποίων διακινούνται αγαθά, όπως και στον πραγματικό κόσμο.
 - ❑ Στόχος είναι να καθοριστεί τόσο η τιμή του αγαθού προς διάθεση όσο και σε ποιον πράκτορα θα διατεθεί.

Πρωτόκολλα πλειστηριασμού (συν)

- ❖ Σε ένα πλειστηριασμό υπάρχουν δύο εμπλεκόμενα μέρη:
 - ❑ ο εκπλειστηριαστής (auctioneer) ο οποίος θέλει να διαθέσει το συγκεκριμένο αγαθό στην μεγαλύτερη δυνατή τιμή
 - ❑ οι πλειοδότες (bidders), οι οποίοι ανταγωνίζονται για την απόκτηση του αγαθού που διατίθεται, φυσικά στην χαμηλότερη δυνατή τιμή.
- ❖ Οι πλειοδότες κάνουν μια ή περισσότερες προσφορές (bids) και το αγαθό διατίθεται στον πράκτορα που έδωσε την μεγαλύτερη προσφορά.
 - ❑ Κάθε εμπλεκόμενος πράκτορας σε ένα πλειστηριασμό έχει τον προσωπικό στόχο που πρέπει να ικανοποιήσει, ο οποίος είναι ανταγωνιστικός ως προς τους στόχους των υπολοίπων πρακτόρων.
- ❖ Οι μηχανισμοί με τους οποίους καθορίζεται ο τελικός πλειοδότης διαφέρουν στα ακόλουθα σημεία:
 - ❑ Αν οι προσφορές είναι ανοικτές (open cry) ή κλειστές (sealed bid), δηλαδή αν οι πράκτορες γνωρίζουν τις προσφορές των άλλων πλειοδοτών - πρακτόρων.
 - ❑ Αν γίνεται μόνο ένας κύκλος προσφορών (one shot) ή περισσότεροι. Στην δεύτερη περίπτωση μπορεί η τιμή των προσφορών σε κάθε κύκλο να αυξάνεται (ascending) ή να μειώνεται (descending).
 - ❑ Ο τρόπος με τον οποίο καθορίζεται η τελική τιμή του αγαθού, δηλαδή αν ο πράκτορας καλείται να "καταβάλλει" την τιμή της προσφοράς του ή την δεύτερη χαμηλότερη τιμή.

Είδη Πλειστηριασμών

❖ Υπάρχουν τέσσερα είδη πλειστηριασμών:

- ❑ Πλειστηριασμοί κλειστών προσφορών ενός κύκλου (one shot sealed bid auctions),
- ❑ Πλειστηριασμοί Αγγλικού τύπου (English auctions),
- ❑ Πλειστηριασμοί Ολλανδικού τύπου (Dutch Auctions)
- ❑ Πλειστηριασμοί Vickrey (Vickrey auctions).

Πλειστηριασμοί κλειστών προσφορών ενός κύκλου.

- ❑ Αποτελούν την απλούστερη μορφή πλειστηριασμών.
- ❑ Οι πλειοδότες καταθέτουν τις κλειστές προσφορές τους στον εκπλειστηριαστή, ο οποίος είναι υπεύθυνος για τον έλεγχο των προσφορών και την ανάδειξη του νικητή.
- ❑ Το αγαθό διατίθεται στον πράκτορα με τη μέγιστη προσφορά στην τιμή που προσέφερε.

Πλειστηριασμοί Αγγλικού τύπου

- ❑ Οι προσφορές είναι ανοικτές και ο πλειστηριασμός λαμβάνει χώρα σε περισσότερους του ενός κύκλους.
- ❑ Οι προσφορές ξεκινούν από μια χαμηλή τιμή και οι πράκτορες έχουν δικαίωμα να κάνουν μια προσφορά με τιμή μεγαλύτερη της τρέχουσας.
- ❑ Μέσα από διαδοχικούς κύκλους προσφορών η τιμή ανεβαίνει έτσι ώστε κάποια στιγμή οι προσφορές σταματούν.
- ❑ Ο νικητής του πλειστηριασμού είναι εκείνος ο οποίος έχει κάνει την μεγαλύτερη (τελευταία) προσφορά και καλείται να καταβάλλει το αντίστοιχο τίμημα που προσέφερε.

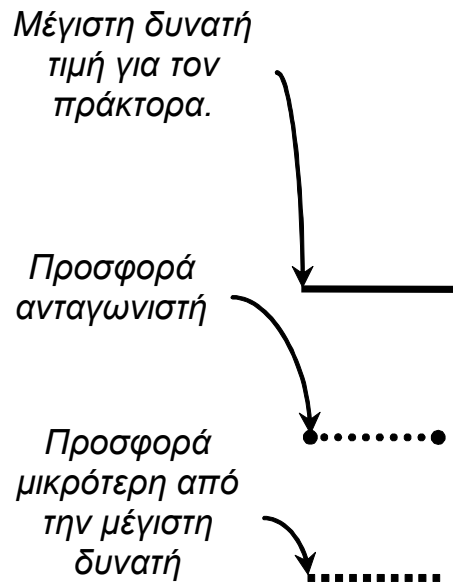
Πλειστηριασμοί Ολλανδικού τύπου

- ❑ Παρόμοιοι με τους πλειστηριασμούς Αγγλικού τύπου.
- ❑ Διαφορά: ο εκπλειστηριαστής ξεκινά την διαδικασία από μια αρκετά υψηλή τιμή την οποία ανακοινώνει στους πλειοδότες αναμένοντας προσφορά από κάποιον πλειοδότη.
- ❑ Σε περίπτωση κανένας από τους πλειοδότες δεν προχωρήσει σε προσφορά, τότε ο εκπλειστηριαστής μειώνει την τιμή και ξεκινά ένας νέος κύκλος πλειστηριασμού.
- ❑ Το αγαθό διατίθεται σε εκείνον που κάνει την πρώτη προσφορά.

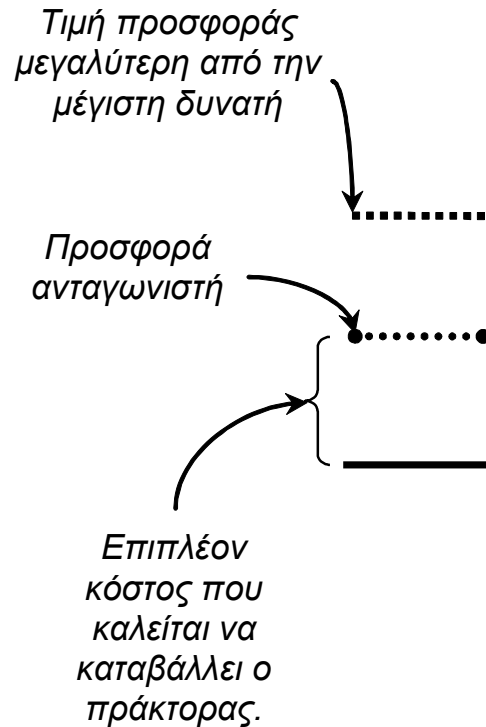
Πλειστηριασμοί Vickrey

- ❑ Αποτελούν ίσως το πλέον ενδιαφέρον (και σπάνιο) είδος πλειστηριασμών.
- ❑ Είναι πλειστηριασμοί ενός κύκλου, κλειστού τύπου όπου ο νικητής είναι εκείνος που κατέθεσε την μεγαλύτερη προσφορά αλλά καλείται να καταβάλλει το τίμημα της δεύτερης μεγαλύτερης προσφοράς.
- ❑ Εξασφαλίζουν ότι οι πράκτορες θα προσφέρουν την μέγιστη δυνατή τους τιμή για το αγαθό.

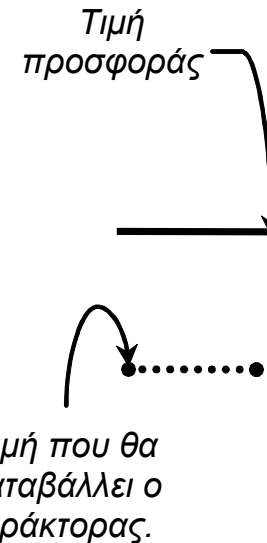
Γιατί οι πράκτορες δίνουν την πραγματική τους τιμή στους Πλειστηριασμούς Vickrey



Περίπτωση (α): Ο πράκτορας χάνει τον πλειστηριασμό ενώ μπορούσε να τον κερδίσει.



Περίπτωση (β): Ο πράκτορας καλείται να πληρώσει μεγαλύτερη τιμή από εκείνη την οποία ήταν διατεθειμένος να πληρώσει.



Περίπτωση (γ): Ο πράκτορας κερδίζει την δημοπρασία με την μικρότερη δυνατή τιμή.

Εφαρμογές Πρακτόρων

- ❖ Πλήθος βιομηχανικών και εμπορικών εφαρμογών, όπως ο έλεγχος μεταφοράς ηλεκτρικής ενέργειας, ο έλεγχος γραμμών παραγωγής.

Έλεγχος εναέριας κυκλοφορίας

- ❖ Το σύστημα OASIS είναι ένα πολυπρακτορικό σύστημα στο οποίο
 - ❑ Τα αεροσκάφη όσο και τα διάφορα συστήματα ελέγχου αναπαρίστανται από πράκτορες.
 - ❑ Σε κάθε αεροσκάφος που μπαίνει στην επιχειρησιακή ζώνη του αεροδρομίου ανατίθεται ένας πράκτορας
 - Αποκτά τους στόχους και γνωρίζει όλες τις πληροφορίες που αφορούν το αεροσκάφος, (αεροδρόμιο προορισμού, τύπος αφους, κλπ.)
 - ❑ Οι πράκτορες οι οποίοι αντιστοιχούν στα συστήματα ελέγχου εναέριας κυκλοφορίας είναι υπεύθυνοι για το συντονισμό και τη διαχείριση ολόκληρου του συστήματος.
- ❖ Οι πράκτορες του OASIS έχουν αναπτυχθεί βάσει του μοντέλου BDI
- ❖ Όλο σύστημα δοκιμάζεται σε πραγματικές συνθήκες στο αεροδρόμιο του Σίδνεϋ.



Εφαρμογές:

Πράκτορες διαδικτύου/πληροφοριών

- ❖ Διευκολύνουν το χρήστη στην ανεύρεση της χρήσιμης πληροφορίας στο διαδίκτυο.
- ❖ Στόχοι τους είναι:
 - Να φιλτράρουν την εισερχόμενη μέσω newsgroups ή mailing lists πληροφορία
 - Να αναζητούν στο διαδίκτυο πληροφορίες που αφορούν ειδικότερα ενδιαφέροντα του χρήστη.
- ❖ Μαθαίνουν τις προτιμήσεις και τα ενδιαφέροντα του κάθε χρήστη χρησιμοποιώντας μεθόδους μηχανικής μάθησης.
 - Παρατηρούν τις επιλογές του (σύστημα MAXIMS)
 - Μέσω παραδειγμάτων (σύστημα NEWT)



Εφαρμογές:

Πράκτορες διεπαφής

- ❖ Αποτελούν προσωπικούς βοηθούς (personal assistants) του χρήστη
 - ❑ Μαθαίνουν τις ιδιαίτερες προτιμήσεις του ή ακόμη και τον βοηθούν στη χρήση προγραμμάτων.
- ❖ Παραδείγματα
 - ❑ Ο γνωστός συνδετήρας του προγράμματος WORD.
 - ❑ Πράκτορες οι οποίοι βοηθούν τους χρήστες να κανονίσουν το ημερήσιο πρόγραμμα τους
 - CALENDAR AGENT, CALENDAR APRENTICE

Άλλες Εφαρμογές

- ❖ Παρακολούθηση ασθενών
- ❖ Παιχνίδια
- ❖ Προσομοίωση
- ❖ Ηλεκτρονικό εμπόριο, κλπ.

Είναι βέβαιο ότι η τεχνολογία των πρακτόρων θα επηρεάσει σε πολύ μεγάλο βαθμό όλα τα πεδία εφαρμογής της πληροφορικής στις επόμενες δεκαετίες.