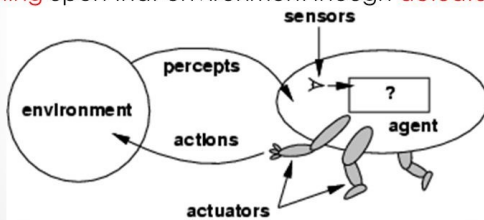


# ΜΕΡΟΣ Ζ

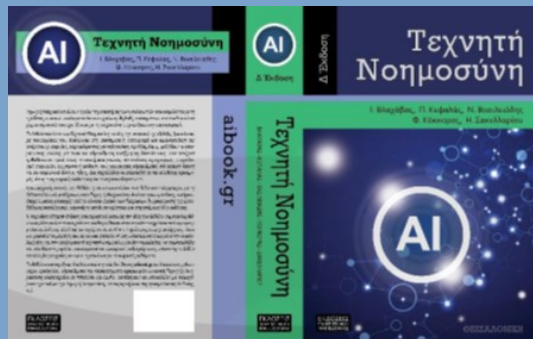
## Agents

- An AI program = An intelligent Agent
- An **agent** is anything that can be viewed as **perceiving** its **environment** through **sensors** and **acting** upon that environment through **actuators**.



- Perception-Action Cycle

## Συστήματα Πρακτόρων (Agents)



Ι. Βλαχάβας, καθηγητής  
Τμήμα Πληροφορικής, ΑΠΘ

Τεχνητή Νοημοσύνη - Δ' Έκδοση-Εκδόσεις Πανεπιστημίου Μακεδονίας  
ISBN: 978-618-5196-44-8 - <https://aibook.gr/>  
Ι. Βλαχάβας, Π. Κεφαλάς, Ν. Βασιλειάδης, Φ. Κόκκορας, Η. Σακελλαρίου.



# Εισαγωγή

- ❖ Ο πιλότος πλησιάζοντας στο Λονδίνο, το αεροσκάφος τον ενημερώνει ότι λόγω της ύπαρξης άλλων 10 αεροσκαφών που κατευθύνονταν στο ίδιο σημείο στον εναέριο χώρο, άλλαξε προσωρινά την πορεία του και θα επιστρέψει στο ίδιο ύψος πτήσης σε 10 λεπτά.
  - ❑ Καμιά επικοινωνία με τον πύργο ελέγχου".
- ❖ Επιστημονική φαντασία; Όχι εντελώς.
  - ❑ Καθώς ο αριθμός πτήσεων αυξάνεται δραματικά οι κλασικές μέθοδοι διαχείρισης της εναέριας κυκλοφορίας, μέσω κεντρικού ελέγχου (ελεγκτές εναέριας κυκλοφορίας) αδυνατεί να διαχειριστεί αποδοτικά τον όγκο των πτήσεων.
- ❖ Πειραματικό σύστημα AGENTFLY
  - ❑ Κάθε αεροσκάφος μοντελοποιείται από ένα ευφυή αυτόνομο πράκτορα ο οποίος είναι υπεύθυνος για τον καθορισμό της πορείας του. Ο πράκτορας πρέπει να:
    - ✓ α) γνωρίζει το σχέδιο πτήσης και να μπορεί να ανιχνεύσει πιθανές καταστάσεις σύγκρουσης με "γειτονικά" αεροσκάφη,
    - ✓ β) επικοινωνεί και να διαπραγματεύεται αλλαγές στην πορεία του με άλλα αεροσκάφη και πύργο ελέγχου
    - ✓ γ) επανασχεδιάζει την πτήση



# Ευφυείς Πράκτορες (Intelligent Agents)

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

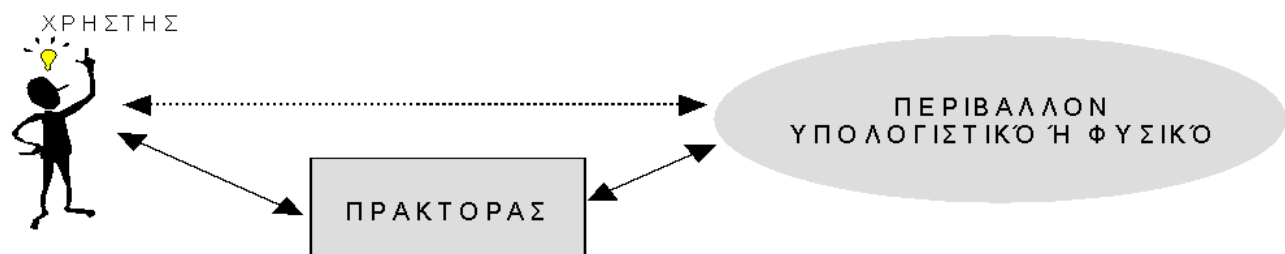
**Πράκτορας (agent): Μία οντότητα που αντιλαμβάνεται το περιβάλλον μέσα στο οποίο βρίσκεται με τη βοήθεια αισθητήρων (sensors), είναι μέρος του περιβάλλοντος αυτού, κάνει συλλογισμούς για το περιβάλλον και δρα πάνω σε αυτό με τη βοήθεια μηχανισμών δράσης (effectors), για την επίτευξη κάποιων στόχων.**

- ☐ **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, ο αυτοματοποιημένος έλεγχος μεγάλων εργοστασιακών μονάδων, κτλ.
- ❖ Τα συνώνυμα που αναφέρονται σε πράκτορες, πέρα από τους όρους softbots και robots είναι πολλά, μεταξύ αυτών και knowbots, taskbots, userbots, personal assistants.
  - ❑ Οι επιθετικοί προσδιορισμοί συχνά χαρακτηρίζουν την κύρια λειτουργία των πρακτόρων, όπως πράκτορες αναζήτησης, αναφοράς, παρουσίασης, προσανατολισμού, διαχείρισης, κτλ.
  - ❑ Ήδη συστήματα όπως το SIRI, GOOGLE NOW, CORTANA και το ECHO αποτελούν εμπορικά διαθέσιμα εργαλεία προσωπικών βοηθών (personal assistant agents).
- ❖ Με τι θα ασχοληθούμε:
- ❖ Η παρουσίαση αφορά τους **νοήμονες λογισμικούς πράκτορες** (*intelligent software agents*) η οποία περιλαμβάνει ένα μεγάλο εύρος υπολογιστικών οντοτήτων, από
  - ❑ σχετικά απλά συστήματα, όπως είναι ο "βοηθός" που παρέχει συμβουλές ορθογραφικές και σύνταξης στα προγράμματα επεξεργασίας κειμένου,
  - ❑ μέχρι κατανεμημένα συστήματα ελέγχου βιομηχανικών εγκαταστάσεων, παραγωγής ενέργειας, διαχείρισης στόλων ταξί, κλπ.



# ***Κεφάλαιο 27***

## ***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**)

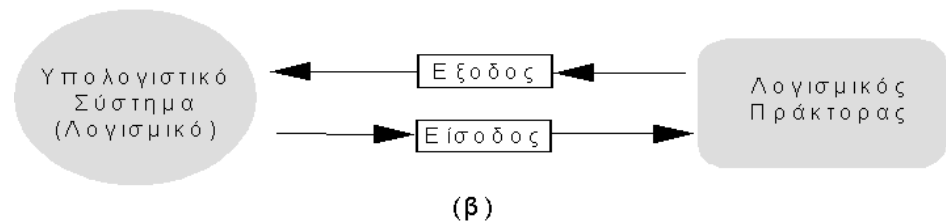
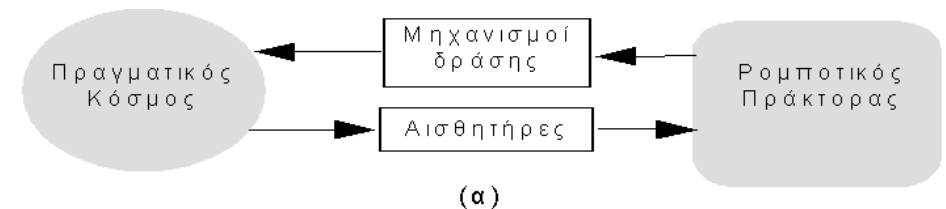
- ✓ Έχουν σαν αισθητήρες και μηχανισμούς δράσης μηχανικά ή ηλεκτρονικά μέρη και δρουν στον πραγματικό κόσμο

### ❑ 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 ΤΝ 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.





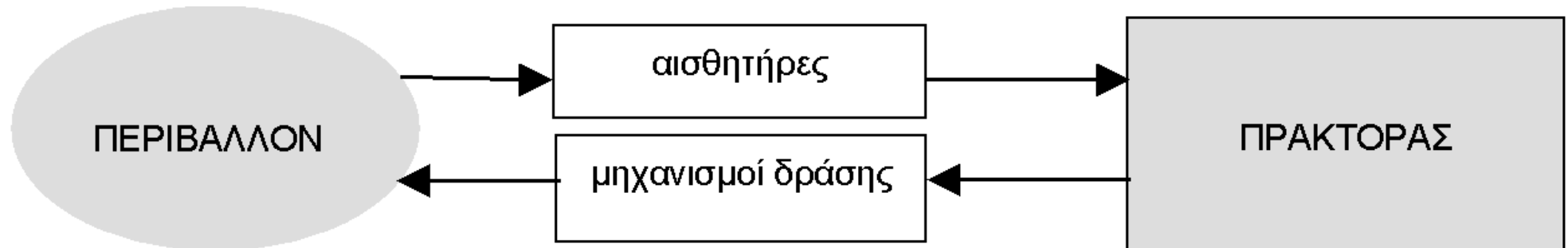
# Ορισμοί Πρακτόρων

- ❖ 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".

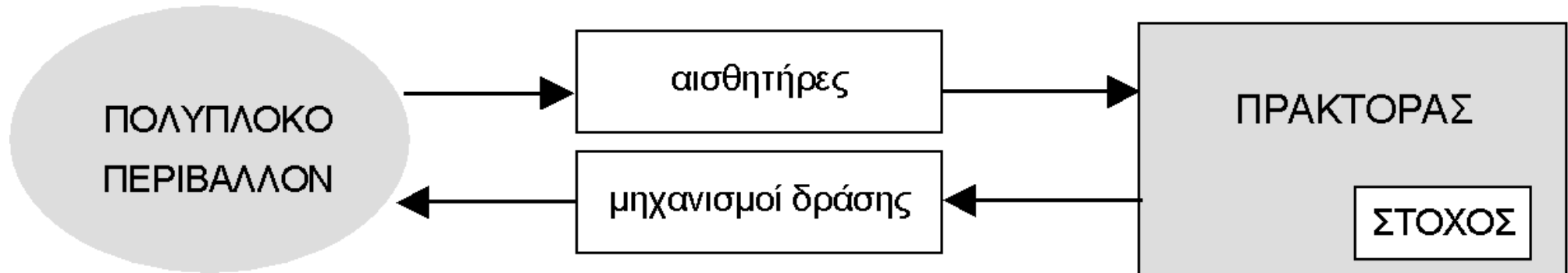




## 2) Βασικός Πράκτορας κατά Maes.

- ❖ 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".

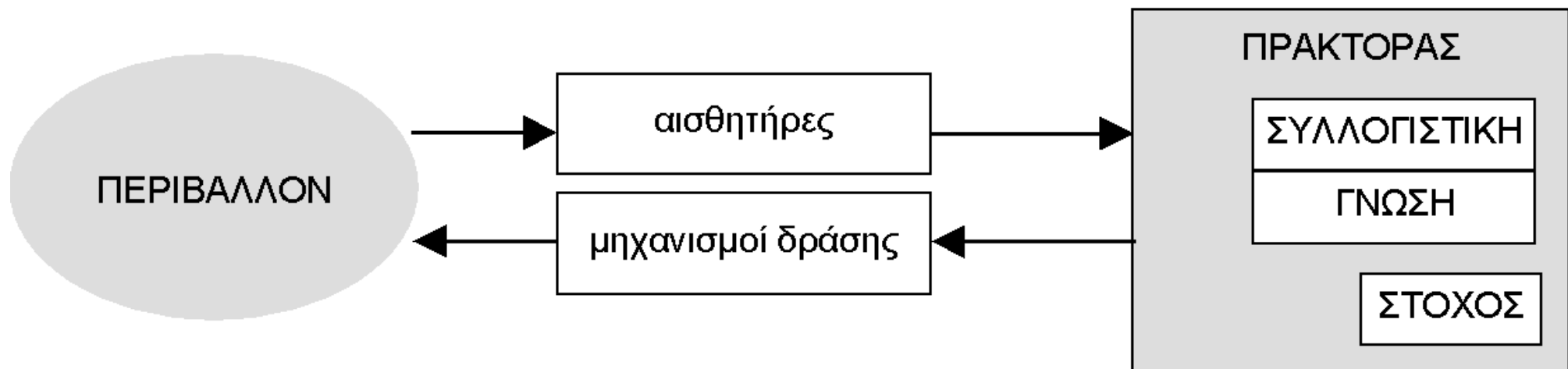




### 3) Βασικός Πράκτορας κατά Hayes-Roth

#### ❖ 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".

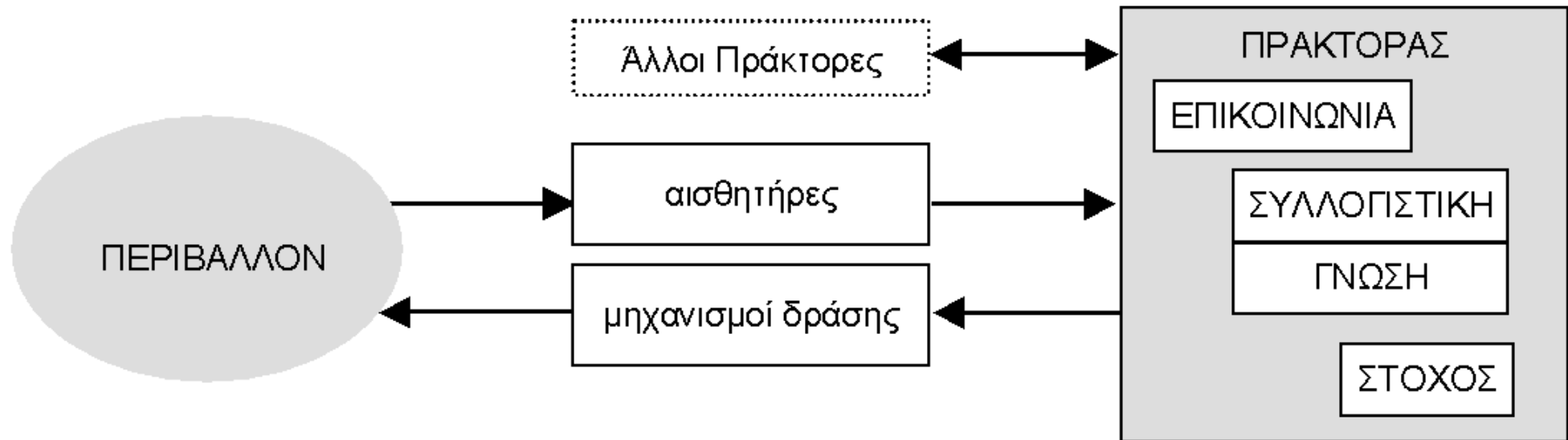




## 4) Βασικός Πράκτορας κατά Coen

### ❖ 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**.
  - ❑ 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 αντανakλαστικά στο περιβάλλον αλλά επιδεικνύουν και **συμπεριφορά που βασίζεται σε στόχους**, ανάλογα με τις συνθήκες του περιβάλλοντος τους.
  - ❑ Κοινωνικότητα (social ability)
    - ✓ Οι πράκτορες επικοινωνούν με άλλους πράκτορες και χρήστες μέσω μιας κοινά κατανοητής **γλώσσας**. (π.χ. KQML, FIPA ACL)
- ❖ 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.

☐ Ορθολογικότητα (rationality)

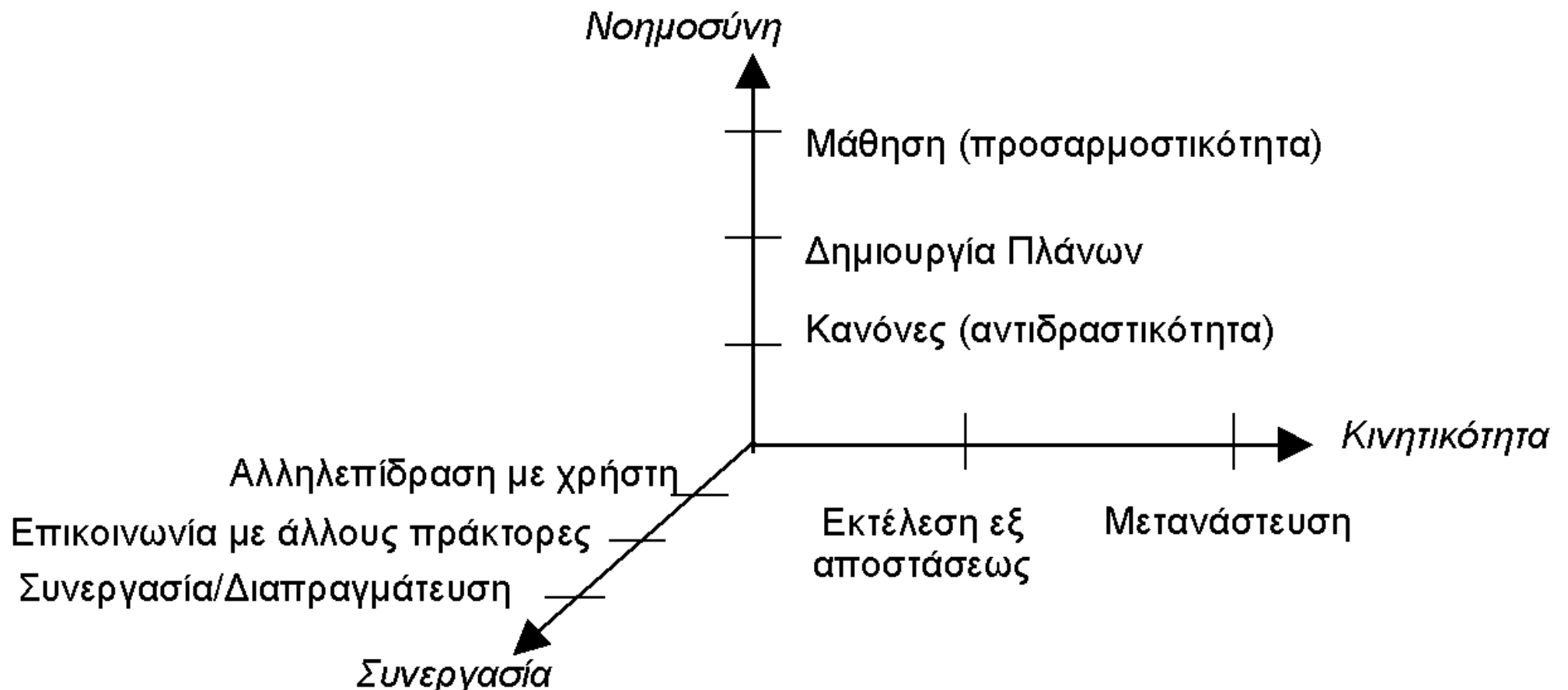
✓ Οι πράκτορες δρουν για να πετύχουν τους στόχους τους, δηλαδή δεν κάνουν αναίτιες ενέργειες και δεν λειτουργούν εναντίον της επίτευξης των στόχων τους.

☐ Συναισθημα (emotion)

✓ Κάποιοι πράκτορες, κάτω από ορισμένες συνθήκες, δρουν βάση (τεχνητών) συναισθημάτων για να πετύχουν τους στόχους τους, όπως για παράδειγμα για να επιδείξουν στοργή (affectiveness).



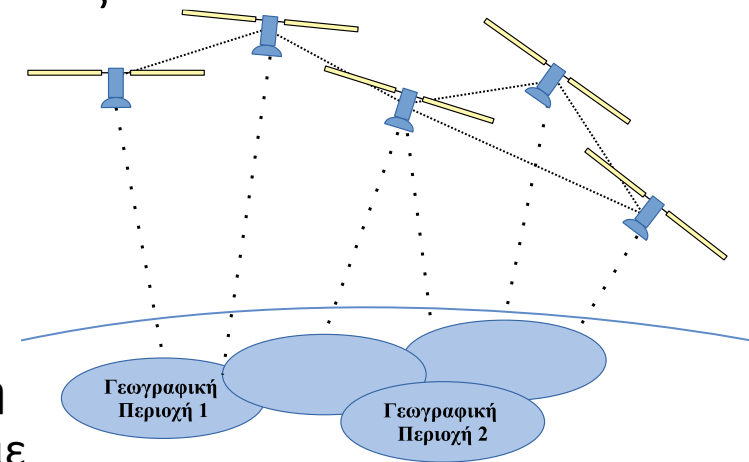
# Άξονες Προσδιορισμού των Χαρακτηριστικών των Πρακτόρων





## Παράδειγμα: Μικροσκοπικοί Δορυφόροι

- ❖ Η σύγχρονη τάση στις επικοινωνίες είναι η δημιουργία ενός δικτύου πάρα πολλών μικρών δορυφόρων οι οποίοι θα προσφέρουν υπηρεσίες διαδικτύου (internet)
- ❖ Η διαχείριση ενός τόσο μεγάλου δικτύου δεν μπορεί να γίνει αποδοτικά χωρίς κάποιας μορφής ευφυές λογισμικό, καθώς ο όγκος της πληροφορίας είναι μεγάλος.
- ❖ Το λογισμικό μπορεί να μοντελοποιηθεί σαν ένα πολυπρακτορικό σύστημα, όπου κάθε δορυφόρος ελέγχεται από δύο είδη πρακτόρων:
- ❖ Τους sat-agent για:
  - ☐ να ελέγχουν πλήρως την λειτουργία του δορυφόρου, την κατάσταση του συστήματος ηλιακής φόρτισης (περιστροφή πτερυγίων προς στην κατεύθυνση του ήλιου), τη σύνδεση με επίγειους σταθμούς (αύξηση/μείωση σήματος, δυναμική διόρθωση τροχιάς),
  - ☐ να αποφεύγουν βροχές "διαστημικών σκουπιδιών" και μετεωριτών
- ❖ και τους manager-agent, οι οποίοι θα:
  - ☐ φροντίζουν το σωστό προγραμματισμό υλοποίησης των αιτημάτων που λαμβάνουν,
  - ☐ συνεργάζονται με άλλους δορυφόρους,
  - ☐ διαπραγματεύονται με γειτονικούς δορυφόρους διαφορετικών εταιρειών την "ενοικίαση" εύρους δικτύου (bandwidth), να γνωρίζουν ποιες (επίγειες) περιοχές μπορούν να καλύψουν,
  - ☐ καταγράφουν την κατάσταση του δορυφόρου (βλάβες, κλπ).







- ❖ Οι πράκτορες του παραπάνω παραδείγματος, εμφανίζουν όλα τα χαρακτηριστικά πρακτόρων που αναφέρθηκαν παραπάνω.
  - ❑ Έχουν αυτονομία και αντιδραστικότητα, καθώς αποφασίζουν για τις ενέργειες τις οποίες πρέπει να κάνουν βάσει της τρέχουσας κατάστασης του περιβάλλοντος,
    - ✓ για παράδειγμα να ανιχνεύουν κοντινές απειλές και να διορθώνουν την τροχιά τους για αποφυγή εμποδίων.
  - ❑ Είναι προνοητικοί καθώς ενεργούν για την επίτευξη των στόχων τους,
    - ✓ για παράδειγμα όταν δέχονται ένα αίτημα φιλοξενίας ιστοσελίδας ενεργούν για την επίτευξη του συγκεκριμένου στόχου,
    - ✓ και τέλος κοινωνικοί καθώς πρέπει να διαπραγματευθούν την ενοικίαση υπολογιστικής ισχύος.
- ❖ Άλλο παράδειγμα
  - ❑ Συστήματα αποφυγής συγκρούσεων αυτοκινήτων,
  - ❑ Οι πράκτορες θα μπορούσαν να προβαίνουν σε ενέργειες οι οποίες είτε να ενημερώνουν τον οδηγό για μια επικείμενη σύγκρουση είτε να μεταβάλλουν την πορεία ή την ταχύτητα του αυτοκινήτου.
    - ✓ Τα συστήματα (ADAS - Advanced Driver Assistance System), μειώνουν απότομα την ταχύτητα του αυτοκινήτου σε περίπτωση που αναγνωριστεί εμπόδιο.
  - ❑ Ο απώτερος στόχος φυσικά είναι η δημιουργία ασφαλών αυτόνομων οχημάτων στα οποία η ανάγκη για συμμετοχή του οδηγού στην πλοήγηση του οχήματος να είναι ολοένα και μικρότερη ή και περιττή.



# 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.
    - ✓ Π.χ. το περιβάλλον στο οποίο κινείται ένα αυτόνομο όχημα θεωρείται μη-προσβάσιμο, καθώς η μειωμένη εμβέλεια των αισθητήρων δεν επιτρέπει την απόκτηση πλήρους πληροφορίας για αυτό.
- ❖ Αιτιοκρατικά ή Μη Αιτιοκρατικά/Στοχαστικά (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.
    - ✓ Η ενέργεια για την εκκίνηση μιας νέας web υπηρεσίας είναι διακριτή, ενώ η ενεργοποίηση του συστήματος πέδησης ενός αυτόνομου οχήματος συνεχής, καθώς η δύναμη πέδησης που θα εφαρμοστεί μπορεί να πάρει (σχεδόν) συνεχείς τιμές σε ένα εύρος.



# Αναπαράσταση Περιβάλλοντος και Πράκτορα

- ❖ Τα δομικά στοιχεία του πράκτορα αλλά και το περιβάλλον αναπαρίστανται σαν υπολογιστικές συναρτήσεις
  - ❑ 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 (ως συνάρτηση agent)  $\text{agent}: \text{seq } S \rightarrow A$ 
    - ✓ πεδίο ορισμού την ακολουθία  $\text{seq } S$  των στιγμιότυπων του περιβάλλοντος
    - ✓ πεδίο τιμών τις ενέργειες  $A$
  - ❑ Αντίληψη ενός πράκτορα (συνάρτηση see)  $\text{see}: S \rightarrow P$ 
    - ✓ πεδίο ορισμού τις καταστάσεις  $S$
    - ✓ πεδίο τιμών τα δεδομένα αντίληψης  $P$  (percepts)
  - ❑ Εσωτερικές καταστάσεις  $I: I = \{I_1, I_2, \dots, I_n\}$ 
    - ✓  $I$  είναι ένα σύνολο από καταστάσεις  $I_i$  που αντιστοιχούν στις καταστάσεις του πραγματικού κόσμου:
  - ❑ Ορισμός εσωτερικής κατάστασης (συνάρτηση update)  $\text{update}: I \times P \rightarrow I$ 
    - ✓ Η συνάρτηση update έχει σαν πεδίο ορισμού τις εσωτερικές καταστάσεις  $I$  του πράκτορα και τα δεδομένα αντίληψης  $P$  και σαν πεδίο τιμών τις εσωτερικές καταστάσεις του πράκτορα



# Γενικές Αρχιτεκτονικές Πρακτόρων

## ❖ Νοήμονες Πράκτορες με Εσωτερική Κατάσταση (deliberative agents)

- ❑ 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



# Α) Πράκτορες με Εσωτερική Κατάσταση

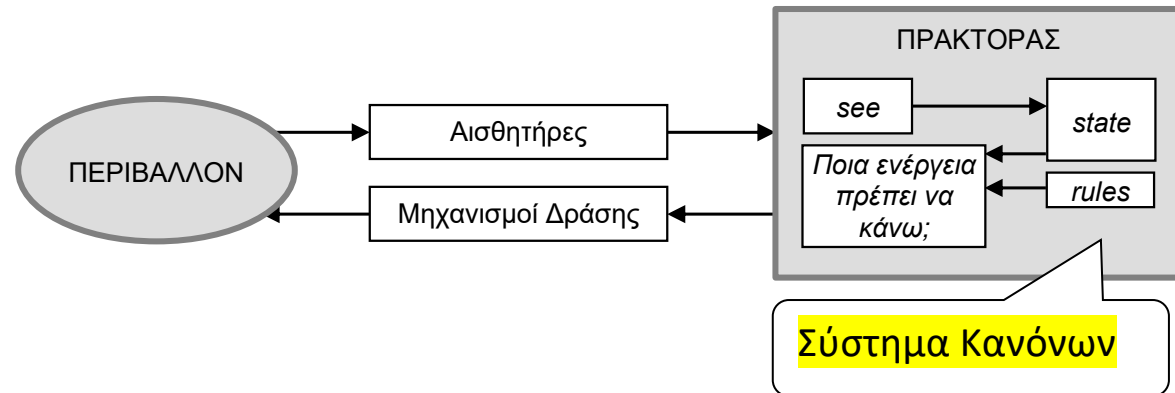
## (α) Γενική Αρχιτεκτονική Πρακτόρων με Εσωτερική Κατάσταση

### ❖ Έχουν:

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

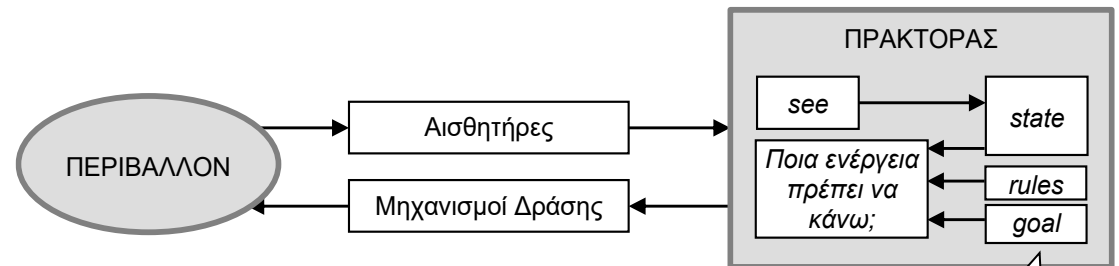
### ❖ Η συνάρτηση η οποία αναπαριστά τη λειτουργία τους:

```
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

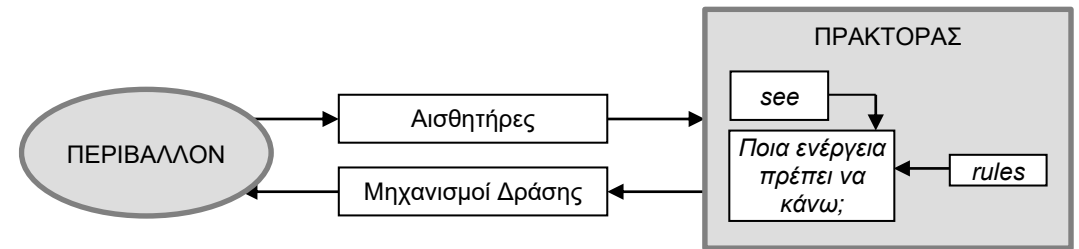
end

Σύστημα σχεδιασμού  
(Planning System)



## Β) 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(environmentstate) 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 ΤΝ:
  - ❑ 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.



# Πράκτορες με Εσωτερική Κατάσταση

## Βασισμένοι στη Λογική

### ❖ 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
  - ✓ το πρόβλημα της επιλογής της επόμενης ενέργειας μετατρέπεται ουσιαστικά σε πρόβλημα απόδειξης στη μαθηματική λογική
- ☐ 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 τις οποίες 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 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
  - ☐ Η 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.



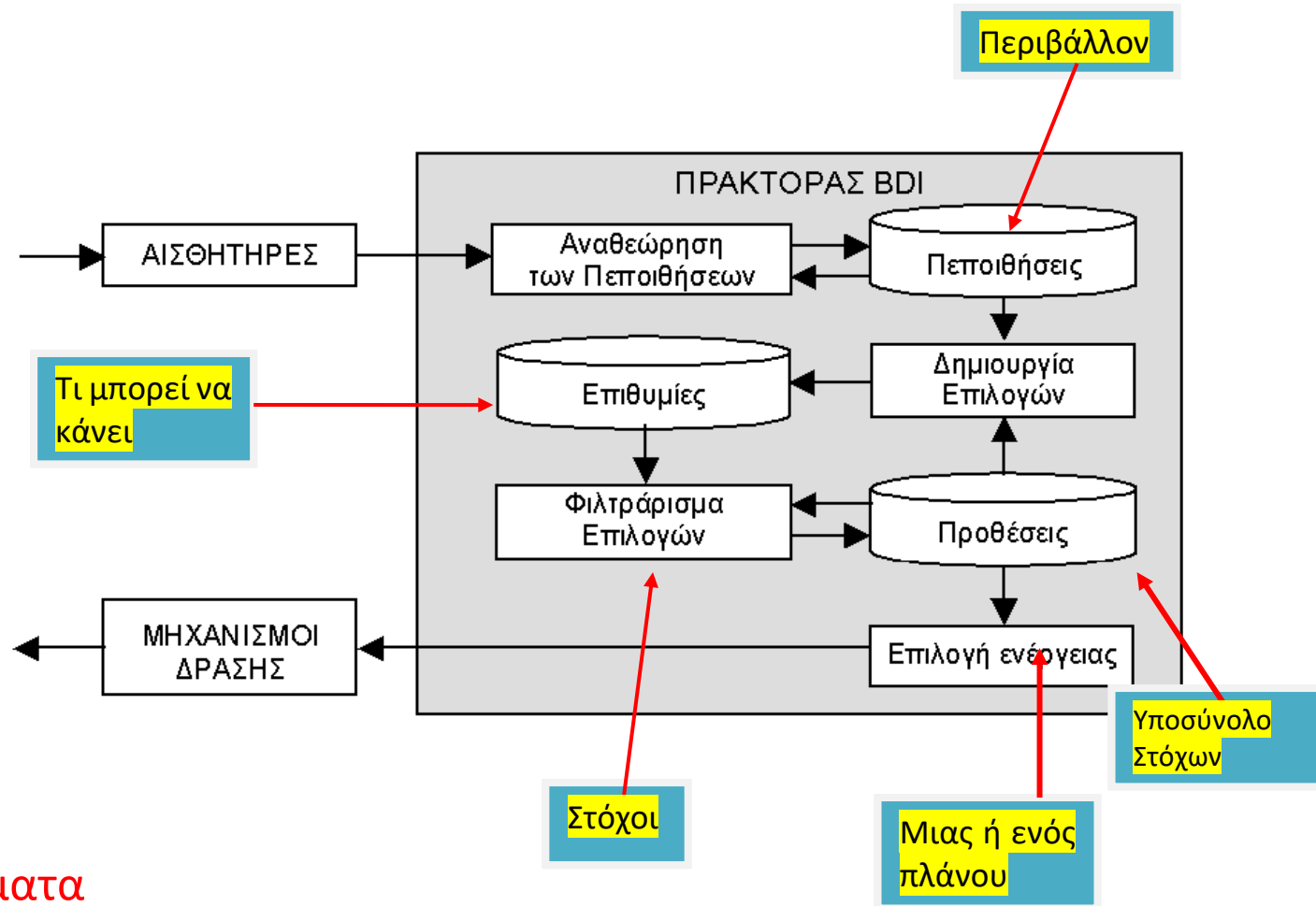
# Αρχιτεκτονική BDI

❖ 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. (δηλαδή επιλογές τις οποίες έχει δεσμευτεί να υλοποιήσει)
- ☐ 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).



# 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.

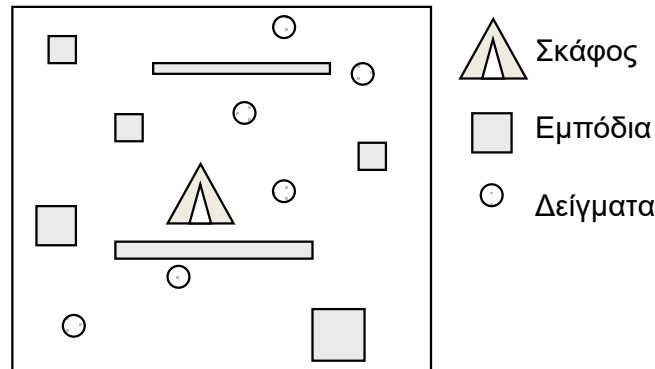






## Παράδειγμα Πράκτορα Υπαγωγής (1/2)

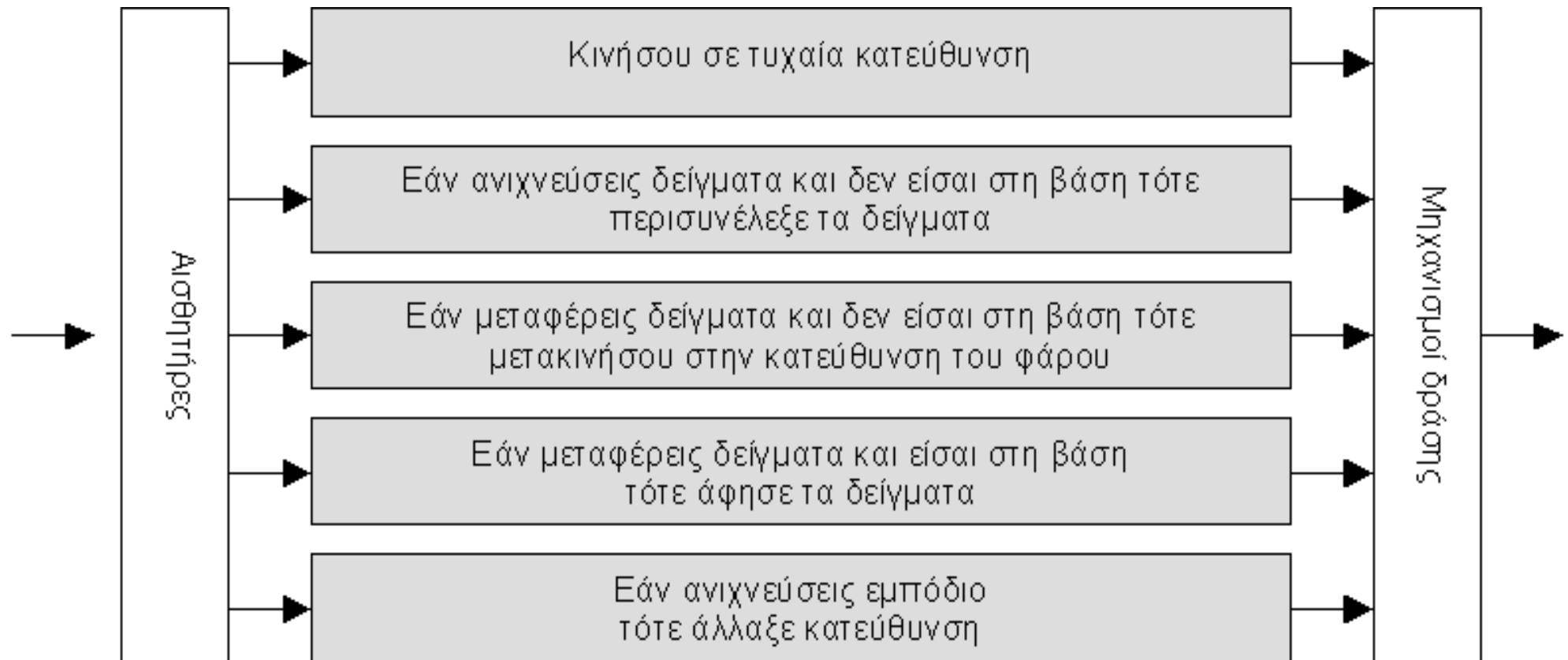
- ❖ Υπάρχει ένα διαστημικό όχημα, το οποίο έχει προσεδαφιστεί σε έναν απομακρυσμένο πλανήτη και ένας αριθμός από πράκτορες, οι οποίοι πρέπει να συλλέξουν δείγματα και να τα μεταφέρουν στο σκάφος, αποφεύγοντας ένα πλήθος εμποδίων.



- ❖ 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).



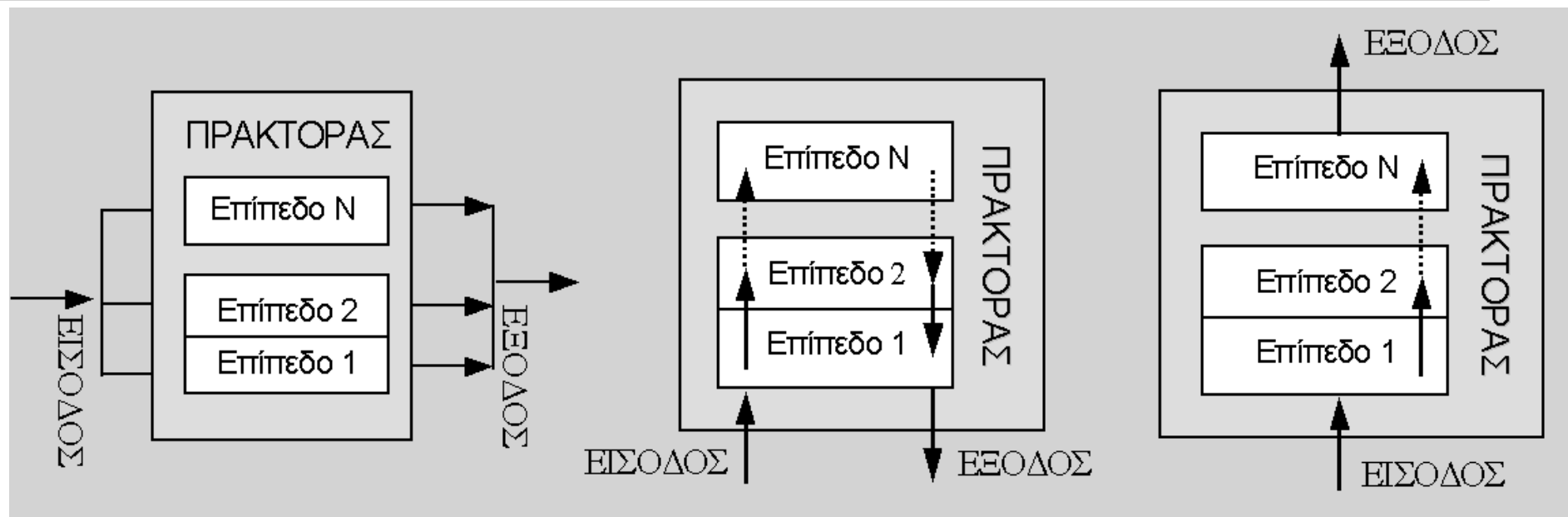
## Παράδειγμα Αντιδραστικών Πρακτόρων

- ❖ Οι πράκτορες sat-agents των δορυφόρων είναι υπεύθυνοι για την αποφυγή βροχών μετεωριτών και άλλων διαστημικών σκουπιδιών και για να διατηρούν επικοινωνία με άλλους δορυφόρους.
  - ❑ Είναι εξοπλισμένοι με διάφορους αισθητήρες, για παράδειγμα έχουν έναν αισθητήρα (radar) ο οποίος ανιχνεύει την ύπαρξη μετεωριτών στη περιοχή τους, έναν ο οποίος ανιχνεύει άλλους δορυφόρους σε κοντινή απόσταση που αποτελούν απειλές και αισθητήρες οι οποίοι ελέγχουν την τηλεπικοινωνιακή σύνδεση του δορυφόρου με γειτονικούς του.
  - ❑ Υποθέτουμε την ύπαρξη ενός σημείου αναφοράς, δηλαδή ότι ο δορυφόρος είναι εξοπλισμένος με ένα όργανο το οποίο του δείχνει την κατεύθυνση σε ένα συγκεκριμένο σημείο στην τροχιά που βρίσκεται.
  - ❑ Η αρχιτεκτονική υπαγωγής, προσφέρεται για τον σχεδιασμό και την υλοποίηση του sat-agent, καθώς με συνδυασμό απλών συμπεριφορών είναι δυνατό να επιτευχθεί η επιθυμητή συμπεριφορά.



## 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.

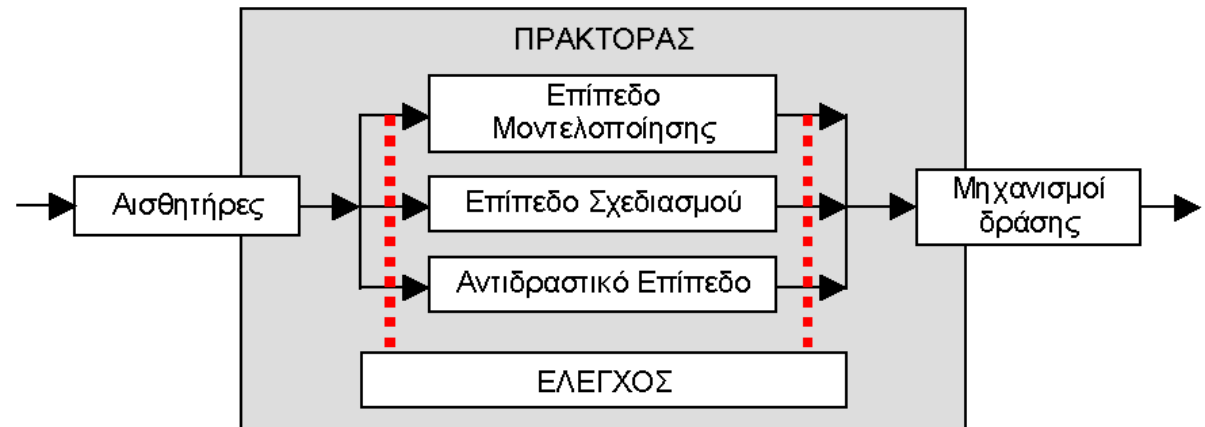
- ❑ Το επίπεδο σχεδιασμού (*planning layer*)

- ✓ 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.



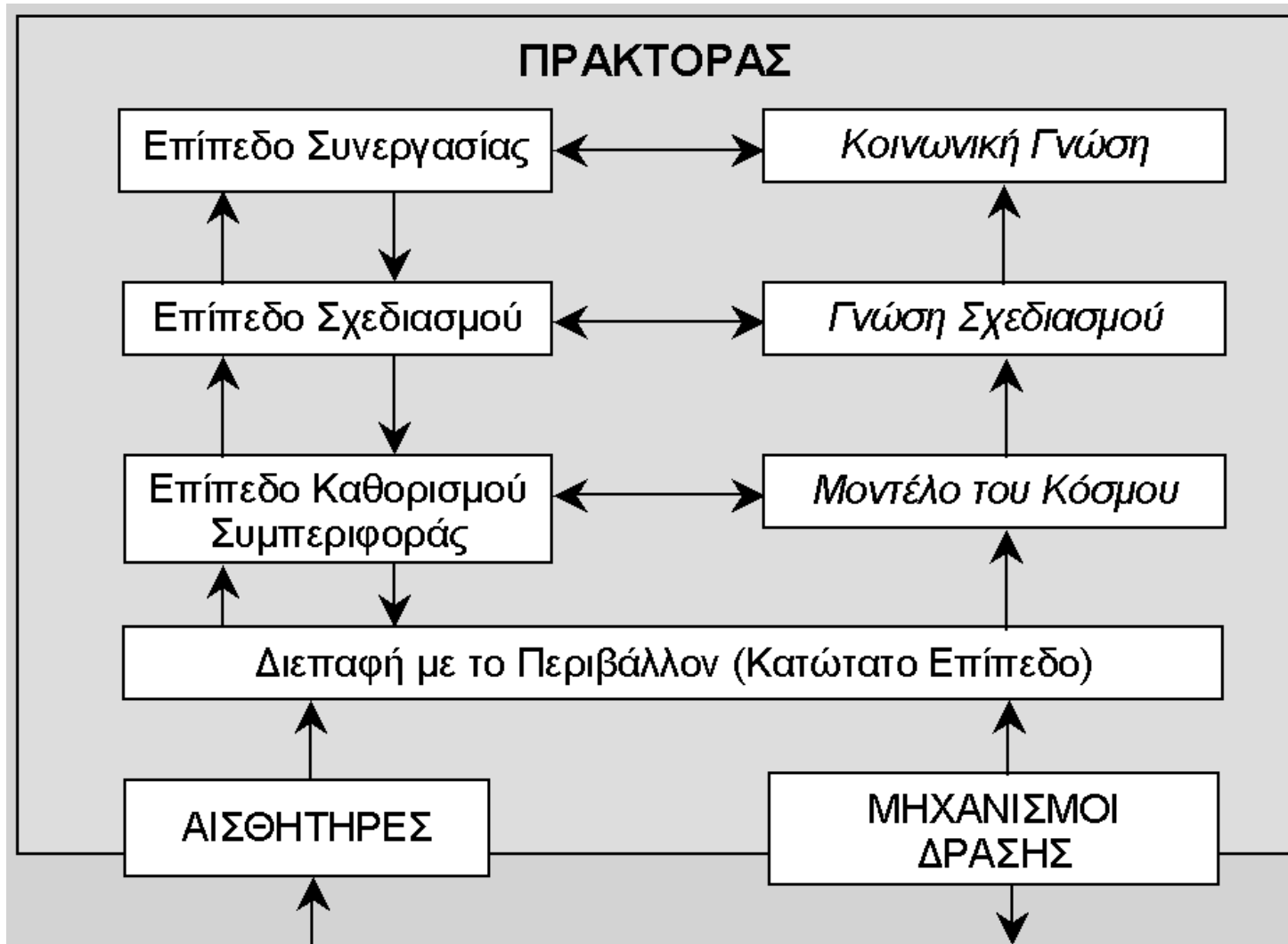


# Αρχιτεκτονική Κάθετης Ροής Ελέγχου (1/2)

- ❖ 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 TELESRIPT, JAVA, C++, κλπ.

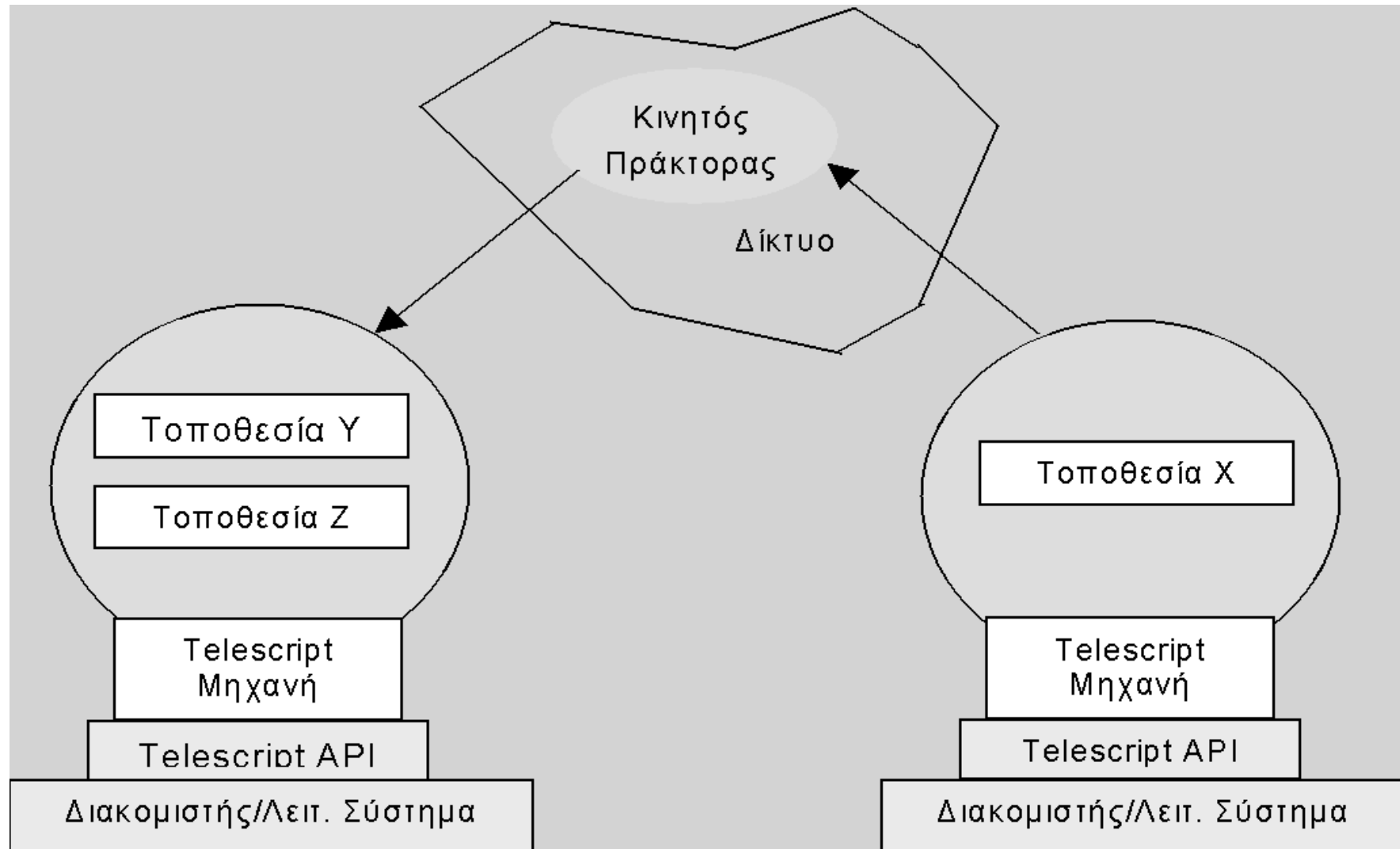


# Η Γλώσσα Προγραμματισμού TELESRIPT (1/2)

- ❖ Αντικειμενοστραφής γλώσσα προγραμματισμού.
- ❖ Σχεδιάστηκε ειδικά για υλοποίηση κινητών πρακτόρων
- ❖ Οι εφαρμογές που αναπτύσσονται στη συγκεκριμένη γλώσσα αποτελούνται από:
  - ☐ TELESRIPT Μηχανές (engines)
    - ✓ Διερμηνευτές (interpreters) που μπορούν να φιλοξενήσουν πολλές διεργασίες.
  - ☐ Τοποθεσίες (places)
    - ✓ Διεργασίες που προσφέρουν υπηρεσίες και μπορούν να περιέχουν και άλλες διεργασίες.
  - ☐ Πράκτορες (TELESRIPT 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 TELESRIPT.
  - ☐ 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++.



# Μια Γλώσσα Προγραμματισμού BDI Πρακτόρων

- ❖ Υπάρχει πλήθος γλωσσών οι οποίες έχουν προταθεί, όπως για παράδειγμα
  - ❑ η 3APL και 2APL, BRAHMS, CONGOLOG, κλπ.
  - ❑ Μια από τις γλώσσες οι οποίες έχουν προταθεί και έχει ενεργές υλοποιήσεις μέχρι σήμερα είναι η AGENTSPEAK(L), ή απλά AGENTSPEAK.
  - ❑ Η τελευταία αποτελεί μια προσπάθεια προς την πρακτική υλοποίηση της αρχιτεκτονικής BDI, η οποία προτάθηκε από τον Rao, με κύριο στόχο την τυπική (formal) περιγραφή ήδη υλοποιημένων BDI συστημάτων, όπως το PRS, dMARS κλπ.
  - ❑ Η γλώσσα αποτελεί επέκταση του Λογικού Προγραμματισμού και προτείνει απλές δομές για την αναπαράσταση της πολύπλοκης συλλογιστικής BDI.
- ❖ Τα βασικά στοιχεία ενός προγράμματος στη γλώσσα AGENTSPEAK(L) είναι οι πεποιθήσεις (beliefs), οι στόχοι (goals) και τα πλάνα (plans).
  - ❑ Οι πεποιθήσεις περιγράφουν την κατάσταση του πράκτορα, όπως ο τελευταίος τη γνωρίζει και είναι εκφρασμένες σε λογική πρώτης τάξης.
  - ❑ Για παράδειγμα, οι πεποιθήσεις που αναφέρθηκαν για το σύστημα των μικροσκοπικών δορυφόρων περιγράφονται απλά με τον ακόλουθο κώδικα:
    - ✓ `cpu_load(cu1,load(20)).`
    - ✓ `disk_utilisation(cu1,40,max(240)).`





# Κεφάλαιο 28

## 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)

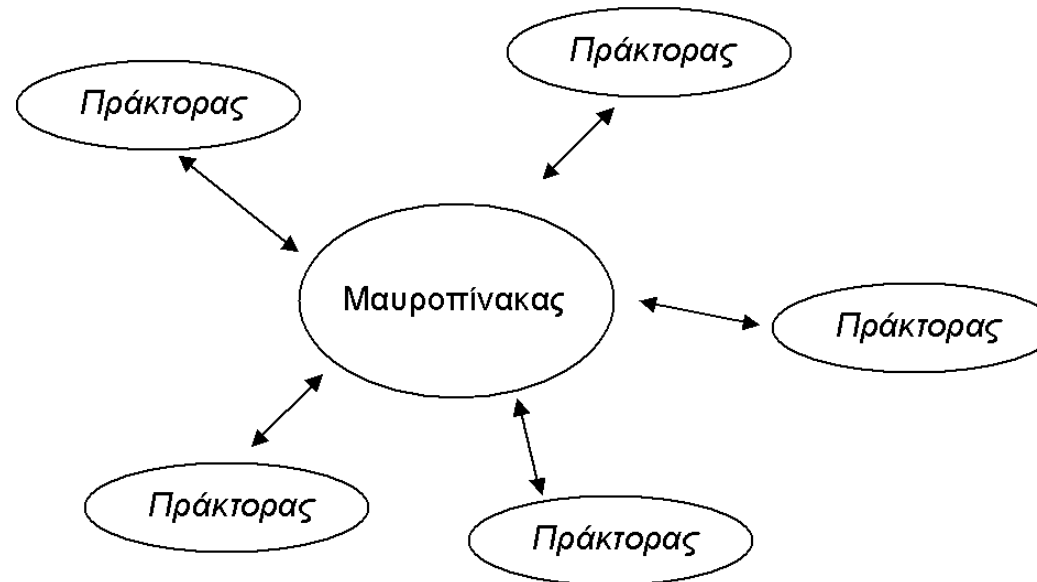
## Μοντέλα Διασύνδεσης

- ❖ Που έχουν προταθεί για τη συνεργασία ανάμεσα στους πράκτορες περιλαμβάνουν δύο βασικές κατηγορίες
  - ☐ 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

- ❖ 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.

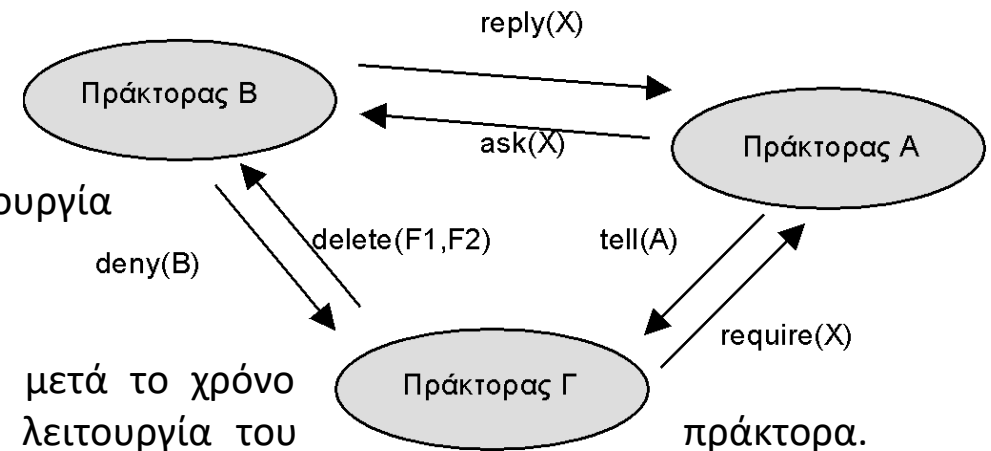
## ❖ Τύπος Επικοινωνίας (Communication Type)

### ❑ 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



## ❖ 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.

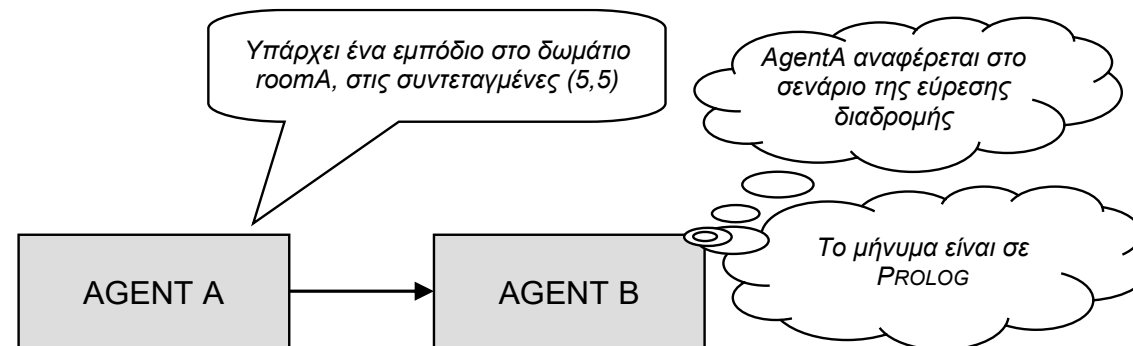




## Παραδείγματα μηνυμάτων σε KQML

- ❖ Έστω ότι ο AgentA δηλώνει στον AgentB το γεγονός ότι υπάρχει ένα εμπόδιο στο δωμάτιο roomA στις συντεταγμένες (5, 5).
- ❖ Το μήνυμα αυτό σε KQML φαίνεται παρακάτω:

```
(tell  
  :sender AgentA  
  :receiver AgentB  
  :language Prolog  
  :ontology path-finding  
  :content obstacle_at(roomA, (5,5) )  
)
```





## Παράδειγμα 2

- ❖ Ένα μήνυμα μπορεί να είναι όμως περισσότερο πολύπλοκο.
- ❖ Για παράδειγμα, έστω ότι 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

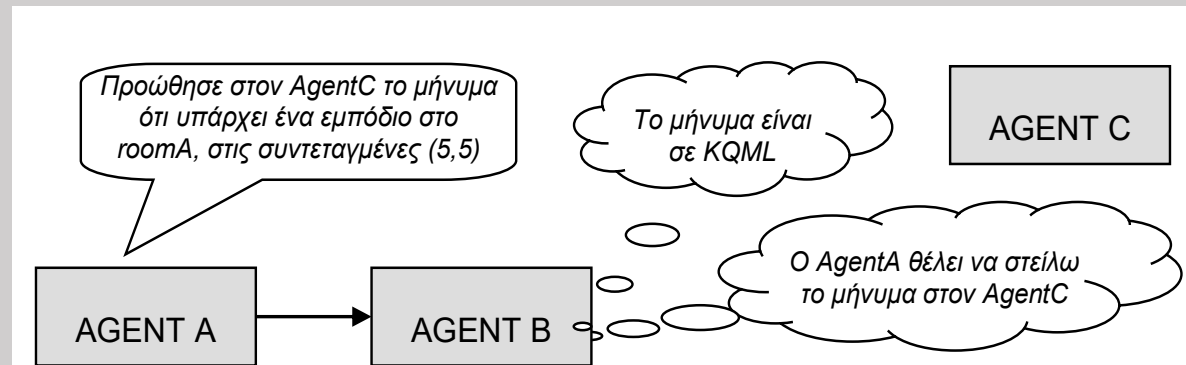
```
:from AgentA  
:to AgentC  
:sender AgentA  
:receiver AgentB  
:language KQML  
:ontology KQML-ontology  
:content
```

(tell

```
:sender AgentA  
:receiver AgentC  
:language Prolog  
:ontology path-finding  
:content obstacle_at(roomA, (5,5))
```

)

)





# 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.



## 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)



## 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.

# 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 the 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 support the debugging and deployment phases. A JADE-based system 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, as and when required. JADE is completely implemented in Java language and the minimal system requirement is the version 5 of JAVA (the run time environment or the JDK).

Besides the **agent abstraction**, JADE provides a simple yet powerful **task** execution and composition model, peer to peer agent **communication** based on the asynchronous message passing paradigm, a **yellow pages** service supporting publish subscribe discovery mechanism and many other advanced features that facilitates the development of a distributed system.

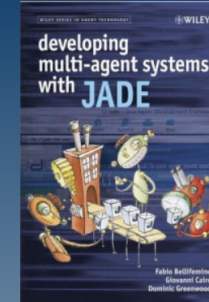
Thanks to the contribution of the LEAP project, ad hoc versions of JADE exist designed to deploy JADE agents transparently on different Java-oriented environments such as **Android** devices and J2ME-CLDC MIDP 1.0 devices.

Furthermore suitable configurations can be specified to run JADE agents in networks characterized by partial connectivity including NAT and firewalls as well as intermittent coverage and IP-address changes.

### Latest news

- 23/12/2015  
JADE 4.4, WADE 3.5 and AMUSE 1.5 have been released
- 11/12/2014  
JADE 4.3.3, WADE 3.4 and AMUSE 1.0 have been released

### Jade book



Developing Multi-Agent Systems with JADE can be ordered from [Wiley](#)

### Download

- [Jade 4.4.0 latest version \(23/12/2015\)](#)
- [White paper Jade](#)



# NETLOGO

## NetLogo



[Home](#)  
[Download](#)  
[Help](#)  
[Resources](#)  
[Extensions](#)  
[FAQ](#)  
[References](#)  
[Contact Us](#)  
[Donate](#)

Models:  
[Library](#)  
[Community](#)  
[Modeling Commons](#)

User Manuals:  
[Web](#)  
[Printable](#)  
[Chinese](#)  
[Czech](#)  
[Japanese](#)

[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](#). Click [here](#) to watch videos.

Join mailing lists [here](#).

**Download NetLogo**



**Go to NetLogo Web**



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



**NetLogo news (via [Twitter](#))**

 NetLogo Retweeted

 **ComplexityExplorer** @ComplexExplorer  
Introducing the Virtual Laboratory! Series of instructional materials and [@netlogo](#) models.



## Ερωτήσεις

- ❖ Αναφέρετε τα 4 κύρια χαρακτηριστικά των Ευφυών Πρακτόρων.
- ❖ Αναφέρετε τα 4 επιθυμητά (ή δευτερεύοντα) χαρακτηριστικά των Ευφυών Πρακτόρων.
- ❖ Αναφέρετε 3 χαρακτηριστικά από τα οποία αποτελείται η εσωτερική κατάσταση των BDI πρακτόρων.
- ❖ Αναφέρετε τους 4 στόχους των πολυπρακτορικών (multiagent) συστημάτων.
- ❖ Αναφέρετε μία γλώσσα επικοινωνίας πρακτόρων.
- ❖ Ποιοι είναι οι τύποι επικοινωνίας μεταξύ πρακτόρων σε ένα πολυπρακτορικό σύστημα; Δώστε μία σύντομη επεξήγηση για τον κάθε τύπο.
- ❖ Αν στον κόσμο του πράκτορα υπάρχει διαθέσιμη πλήρης, ακριβής και ανανεωμένη πληροφορία, ο κόσμος του πράκτορα ονομάζεται:.....
- ❖ Αν μια συγκεκριμένη ενέργεια ενός πράκτορα έχει πάντα συγκεκριμένα αποτελέσματα, ο κόσμος του πράκτορα ονομάζεται:.....
- ❖ Το κύριο χαρακτηριστικό των αντιδραστικών πρακτόρων είναι: .....
  - ☐ Η συμπεριφορά τους βασίζεται σε:.....
  - ☐ Αντιπροσωπευτικό παράδειγμα αρχιτεκτονικής τους είναι: .....
- ❖ Αναφέρετε τα μοντέλα διασύνδεσης πρακτόρων και δώστε μια σύντομη περιγραφή τους.





- ❖ Ποια είναι η διαφορά μεταξύ των πρακτόρων με εσωτερική κατάσταση και των αντιδραστικών πρακτόρων;
- ❖ Αναφέρετε 2 πλεονεκτήματα και 2 μειονεκτήματα των κινητών πρακτόρων.
- ❖ Τι είναι οι αντιδραστικοί πράκτορες. Πλεονεκτήματα, μειονεκτήματα.
- ❖ Ποια χαρακτηριστικά διαφοροποιούν τους πράκτορες από τα συμβατικά προγράμματα (απλή αναφορά).
- ❖ Ποιες γλώσσες χρησιμοποιούνται για την ανάπτυξη κινητών πρακτόρων (απλή αναφορά).

**Κυκλώστε το αντίστοιχο γράμμα Σ (ωστό) – Λ (άθος) στις επόμενες ερωτήσεις:**

Οι ευφυείς πράκτορες προσπαθούν πάντα να πετύχουν τους στόχους που τους έχουν ανατεθεί	Σ	Λ
Στην αρχιτεκτονική του μαυροπίνακα δεν επιτρέπεται η απ' ευθείας επικοινωνία ανάμεσα σε δύο πράκτορες	Σ	Λ
Στα πολυπρακτορικά συστήματα κάθε πράκτορας δεν έχει πλήρη πληροφορία για τον κόσμο	Σ	Λ
Τα πρωτόκολλα επικοινωνίας στα πολυπρακτορικά συστήματα δίνουν δυνατότητες συζητήσεων (ακολουθίες ανταλλαγής μηνυμάτων) στους πράκτορες	Σ	Λ
Οι αποφάσεις των αντιδραστικών πρακτόρων λαμβάνονται μετά από συλλογισμό	Σ	Λ
Η TELESRIPT είναι γλώσσα ανάπτυξης πρακτόρων	Σ	Λ
Η KQML είναι γλώσσα ανάπτυξης πρακτόρων	Σ	Λ
Οι πράκτορες που ανήκουν σε πολυπρακτορικά συστήματα εκτελούν τους υπολογισμούς τους σύγχρονα	Σ	Λ



Η ειλικρίνεια είναι επιθυμητό χαρακτηριστικό των ευφυών πρακτόρων	Σ	Λ
Η κινητικότητα είναι απαραίτητο χαρακτηριστικό των ευφυών πρακτόρων	Σ	Λ
Η αυτονομία είναι απαραίτητο χαρακτηριστικό των ευφυών πρακτόρων	Σ	Λ
Ο κύριος τρόπος ανάγνωσης του κόσμου που χρησιμοποιούν οι ευφυείς πράκτορες είναι μέσω των μηχανισμών ή οργάνων δράσης	Σ	Λ
Η αγαθή προαίρεση είναι απαραίτητο χαρακτηριστικό των ευφυών πρακτόρων	Σ	Λ
Μειονέκτημα των κινητών πρακτόρων είναι η αναγκαιότητα για συνεχή σύνδεση του Η/Υ του χρήστη στο Διαδίκτυο	Σ	Λ
Οι αντιδραστικοί πράκτορες δεν διατηρούν εσωτερική αναπαράσταση του κόσμου	Σ	Λ
Οι στόχοι των πρακτόρων με πεποιθήσεις – επιθυμίες – προθέσεις ταυτίζονται με τις επιθυμίες τους	Σ	Λ
Για να λειτουργήσει ένας ευφυής πράκτορας χρειάζεται εξωτερικό ερέθισμα	Σ	Λ
Ένας ευφυής πράκτορας λογισμικού δεν χρειάζεται παρέμβαση του χρήστη για να λειτουργήσει	Σ	Λ