

# Έμπειρα Συστήματα (Expert Systems)

## ❖ Προγράμματα τα οποία:

- ❑ Επιδεικνύουν **νοήμονα συμπεριφορά** σε συγκεκριμένους τομείς και διαδικασίες, ανάλογη ενός ανθρώπου εμπειρογνώμονα με ειδικότητα στον ίδιο τομέα.
- ❑ Κωδικοποιούν και χειρίζονται τη **γνώση** και τη **συλλογιστική** ενός ανθρώπου-ειδικού σε έναν εξειδικευμένο τομέα, με σκοπό την επίλυση προβλημάτων ή την παροχή συμβουλών.

## ❖ Απαιτούν **εμπειρική γνώση**.

- ❑ Δεν είναι μόνο γνώση ή εκπαίδευση πάνω σε ένα συγκεκριμένο τομέα.
- ❑ Περιλαμβάνει εξειδικευμένες ικανότητες που έχουν αποκτηθεί με κόπο και χρόνο.

## ❖ Χρησιμοποιούνται με δύο τρόπους:

- ❑ Από κάποιον άνθρωπο μη-ειδικό, για να παρέχει λύσεις σε συγκεκριμένα προβλήματα.
- ❑ Συμβουλευτικά, από έναν άνθρωπο-ειδικό ο οποίος καλείται να πάρει κάποια απόφαση.

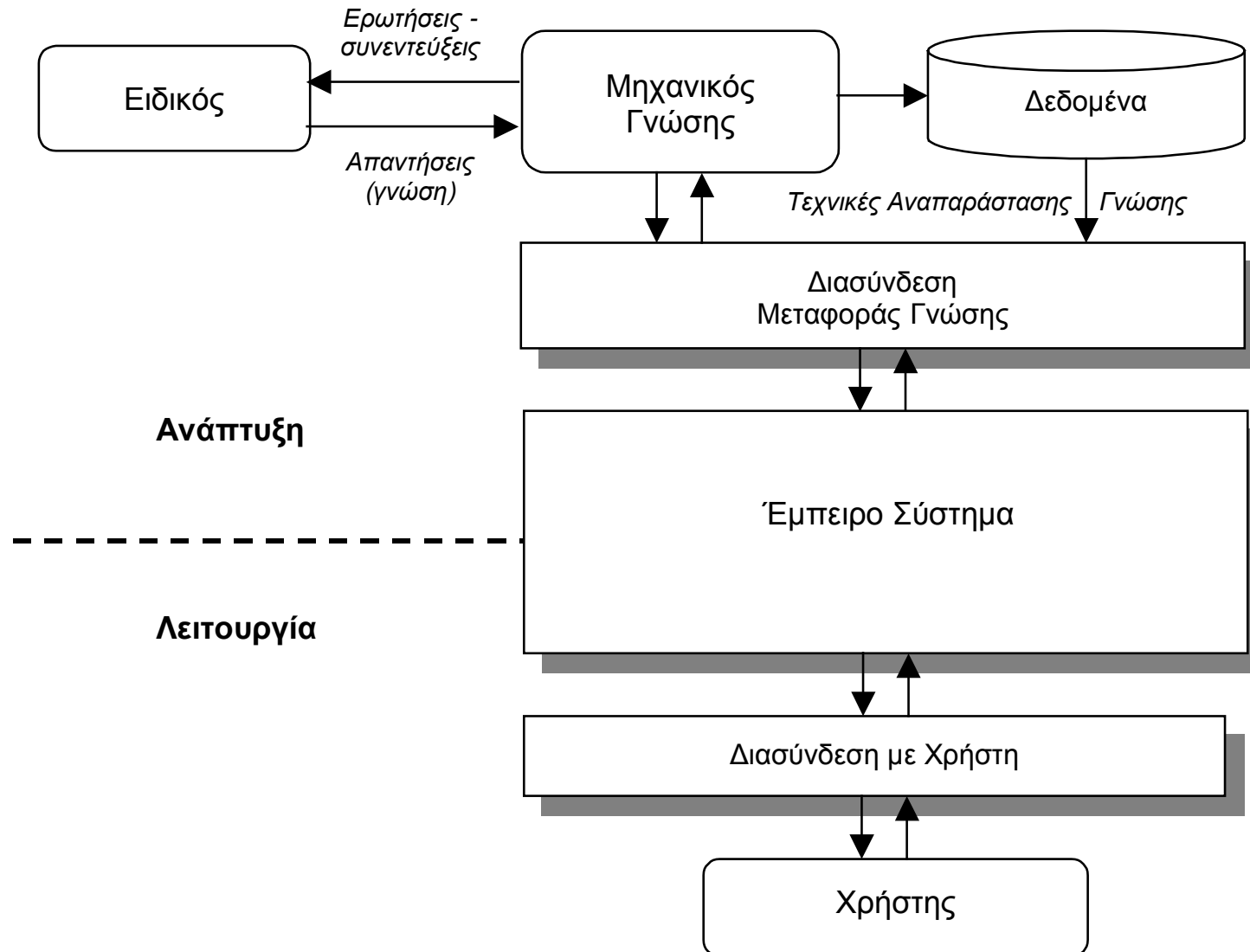
## ❖ Τυπικές κατηγορίες εφαρμογών:

- ❑ Ερμηνεία δεδομένων (π.χ. ηχητικών ή ηλεκτρομαγνητικών σημάτων)
- ❑ Διάγνωση δυσλειτουργιών (π.χ. βλαβών σε μηχανήματα ή ασθενειών σε ανθρώπους)
- ❑ Δομική ανάλυση σύνθετων αντικειμένων (π.χ. χημικών ενώσεων)
- ❑ Διαμόρφωση σύνθετων αντικειμένων (π.χ. πολύπλοκων υπολογιστικών συστημάτων)

# Ανάπτυξη Εμπείρων Συστημάτων

- ❖ Για την ανάπτυξη ενός έμπειρου συστήματος πρέπει να συνεργαστούν:
  - ❑ Ένας ειδικός του τομέα (**domain expert**).
    - Είναι κάποιος άνθρωπος εξειδικευμένος σε έναν τομέα της ανθρώπινης δραστηριότητας.
    - Η γνώση του για τον τομέα αυτό θα μεταφερθεί στο σύστημα.
  - ❑ Ένας μηχανικός γνώσης (**knowledge engineer**) με σκοπό τη λήψη της εμπειρίας (γνώσης) του πρώτου.
    - Είναι ένας επιστήμονας της πληροφορικής, ειδικευμένος σε θέματα ΤΝ και εμπείρων συστημάτων.
    - Με βάση τα αποτελέσματα της συνεργασίας σχεδιάζει το σύστημα και τη δομή της γνώσης και στη συνέχεια το αναπτύσσει.
- ❖ Ο τομέας της ΤΝ που ασχολείται με την ανάπτυξη εμπείρων συστημάτων ονομάζεται **τεχνολογία της γνώσης (knowledge engineering)**.

# Ανάπτυξη και Λειτουργία Έμπειρου Συστήματος



# Χαρακτηριστικά Εμπείρων Συστημάτων

ΕΜΠΕΙΡΑ ΣΥΣΤΗΜΑΤΑ	ΣΥΜΒΑΤΙΚΑ ΠΡΟΓΡΑΜΜΑΤΑ
Προσομοιώνουν τον τρόπο επίλυσης ενός προβλήματος	Προσομοιώνουν το ίδιο το πρόβλημα
Παράσταση και χειρισμός γνώσης σε επίπεδο συμβόλων	Παράσταση και χειρισμός δεδομένων σε επίπεδο αριθμητικών υπολογισμών
Χρήση ευριστικών μεθόδων για περιορισμό του χώρου αναζήτησης	Χρήση αλγορίθμων
Χρήση γλωσσών που πλησιάζουν την ανθρώπινη	Χρήση γλωσσών που βρίσκονται πλησιέστερα στον τρόπο λειτουργίας του Η/Υ
Βάση γνώσης (δεδομένα και εξαγωγή συμπερασμάτων)	Βάση δεδομένων - η γνώση ενσωματώνεται στο πρόγραμμα
Ευχέρεια στην επέκταση και αναθεώρηση της γνώσης	Η αναθεώρηση της γνώσης επιβάλλει ευρείας κλίμακας μεταβολές στο πρόγραμμα
Χειρισμός ασαφούς, αβέβαιης και μη-πλήρους γνώσης	Δυσχέρεια στο χειρισμό ασαφούς, αβέβαιης και μη-πλήρους γνώσης
Δυνατότητα μη μονότονης συλλογιστικής	Δυσχέρεια στη χρήση μη μονότονης λογικής
Επεξήγηση του δρόμου συλλογισμού	Ανυπαρξία επεξήγησης

# Πλεονεκτήματα/Μειονεκτήματα Έμπειρου Συστήματος Σε Σχέση Με Άνθρωπο-Ειδικό

ΑΝΘΡΩΠΟΣ ΕΙΔΙΚΟΣ		ΕΜΠΕΙΡΟ ΣΥΣΤΗΜΑ	
<b>Μειονεκτήματα</b>	Γνώση διαθέσιμη όταν ο ίδιος είναι παρών	<b>Πλεονεκτήματα</b>	Γνώση πάντα διαθέσιμη.
	Δυσκολία μεταφοράς-αποτύπωσης γνώσης		Ευκολία μεταφοράς-αποτύπωσης γνώσης
	Συναισθηματικές παρορμήσεις		Εργάζεται με συνέπεια
	Η απόδοσή του επηρεάζεται από εξωγενείς παράγοντες		Εργάζεται οπουδήποτε
	Υψηλό κόστος		Χαμηλό κόστος λειτουργίας / Υψηλό κόστος ανάπτυξης
	Υποκειμενικότητα		Αντικειμενικότητα αν η γνώση προέρχεται από πολλούς ειδικούς
<b>Πλεονεκτήματα</b>	Δημιουργικότητα, Ευρύνοια	<b>Μειονεκτήματα</b>	Απουσία έμπνευσης, Περιορισμένο πεδίο σκέψης
	Κοινή λογική		Δυσχέρεια στη μεταφύτευση της κοινής λογικής
	Γνώση των ορίων και δυνατοτήτων τους (μετα-γνώση)		Έλλειψη μετα-γνώσης
	Εκφραστική και λειτουργική επεξήγηση του τρόπου σκέψης τους		Μηχανική επεξήγηση του τρόπου λήψης απόφασης
	Ο έλεγχος της γνώσης γίνεται υποσυνείδητα		Πρέπει η γνώση να ελέγχεται για ορθότητα, πληρότητα και συνέπεια
	Αυτονομία στη μάθηση		Πρέπει να προγραμματιστούν για να μαθαίνουν αυτόματα
	Απόκριση σε πραγματικό χρόνο		Δυσκολία απόκρισης σε πραγματικό χρόνο

# Εφαρμογές των Εμπείρων Συστημάτων

## ❖ Διάγνωση (diagnosis).

- ❑ Διάγνωση βλαβών ενός συστήματος βάσει παρατηρήσεων και μετρήσεων.

## ❖ Πρόγνωση (prognosis-prediction).

- ❑ Πρόβλεψη πιθανών μελλοντικών επιπτώσεων με βάση δεδομένες καταστάσεις.

## ❖ Εκπαίδευση (instruction).

- ❑ Κατανόηση, αξιολόγηση και διόρθωση απάντησης μαθητών σε εκπαιδευτικά προβλήματα.

## ❖ Παρακολούθηση καταστάσεων (monitoring).

- ❑ Σύγκριση παρατηρούμενων παραμέτρων με αναμενόμενες καταστάσεις.

## ❖ Επιδιόρθωση λαθών (repair-remedy).

- ❑ Ανάπτυξη και εκτέλεση σχεδίων (πλάνων) για τη διαχείριση βλαβών.

## ❖ Ερμηνεία (interpretation).

- ❑ Περιγραφή αντικειμένων και καταστάσεων βάσει δεδομένων από παρατηρήσεις.

## ❖ Διαμόρφωση (configuration).

- ❑ Ικανοποίηση απαιτήσεων και περιορισμών για τη συναρμολόγηση εξαρτημάτων.

## ❖ Έλεγχος (control).

- ❑ Έλεγχος της συμπεριφοράς ενός συστήματος. Περιλαμβάνει πολλά από τα παραπάνω.

# Γνωστά Έμπειρα Συστήματα

## ❖ DENDRAL

- ❑ Ταυτοποίηση χημικών ενώσεων μέσω φασματικής ανάλυσης.
- ❑ Χρήση ευριστικών κανόνων για περιορισμό του χώρου αναζήτησης.

## ❖ MYCIN

- ❑ Διάγνωση και θεραπεία της μηνιγγίτιδας και της βακτηριαιμίας.
- ❑ Χρήση συντελεστή βεβαιότητας για τις λύσεις, λόγω αβεβαιότητας απαντήσεων χρήστη.

## ❖ PROSPECTOR

- ❑ Πρόβλεψη της ακριβούς θέσης ορυκτών κοιτασμάτων αξιοποιώντας γεωλογικά δεδομένα.
- ❑ Χρήση σημασιολογικών δικτύων και δικτύων πιθανοτήτων.

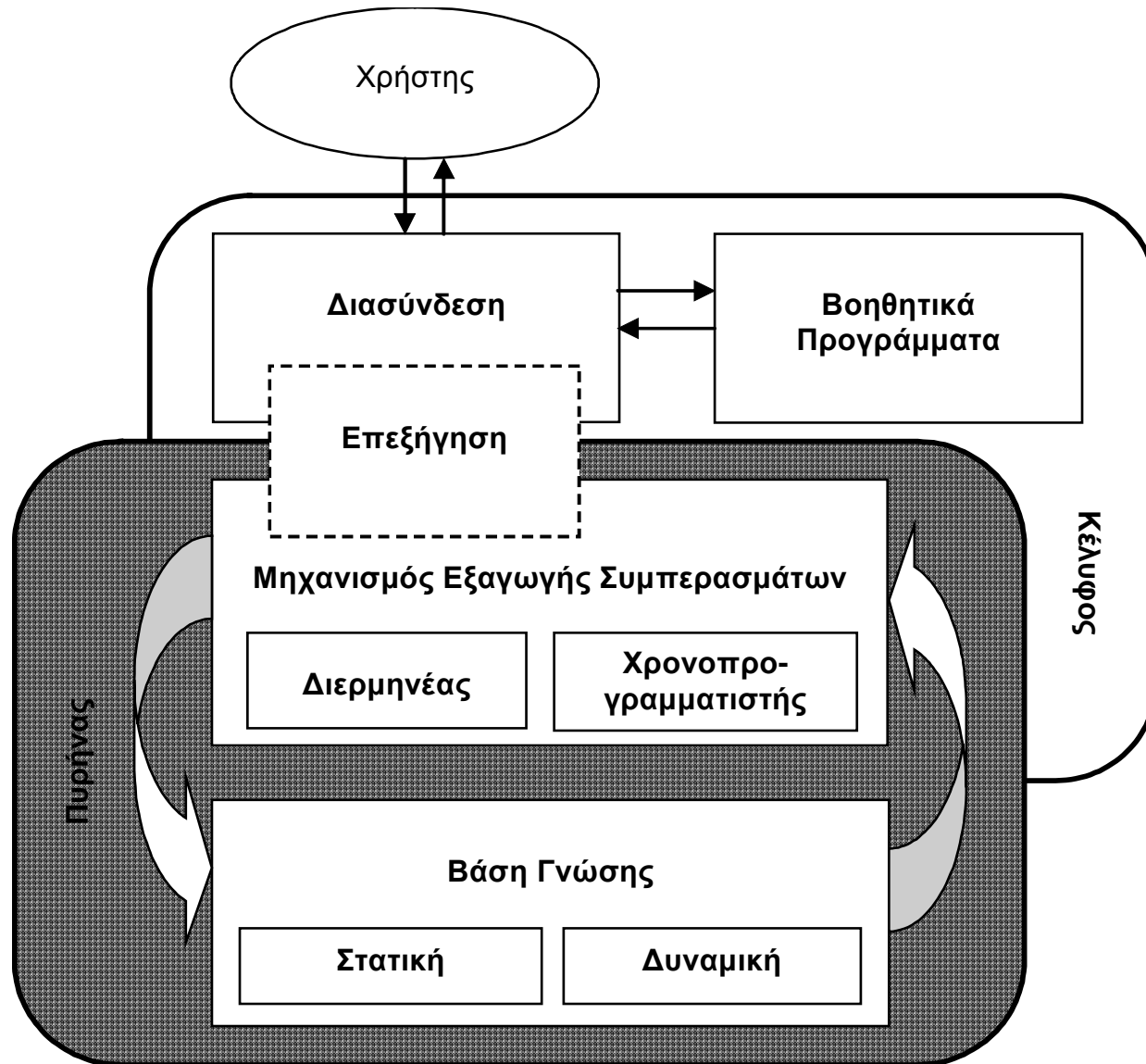
## ❖ INTERNIST

- ❑ Διάγνωση παθολογικών περιπτώσεων με πολύ μεγάλο αριθμό εναλλακτικών διαγνώσεων.
- ❑ Χρήση ευριστικής συλλογιστικής (απαγωγική) για την πιθανότερη διάγνωση.

## ❖ XCON

- ❑ Διαμόρφωση υπολογιστών DEC, για να ανταποκρίνονται στις προδιαγραφές του πελάτη.
- ❑ Αναζήτηση κατάλληλου συνδυασμού και χωρικής διάταξη των εξαρτημάτων, με αποφυγή των ασυμβατοτήτων λειτουργίας και διασύνδεσης μεταξύ τους.

# Αρχιτεκτονική Εμπείρων Συστημάτων (1/2)





## Αρχιτεκτονική Εμπείρων Συστημάτων (2/2)

- ❖ Ο πυρήνας του έμπειρου συστήματος αποτελείται από δύο μέρη:
  - ❑ Τη βάση γνώσης.
  - ❑ Το μηχανισμό εξαγωγής συμπερασμάτων.
- ❖ Ο διαχωρισμός της γνώσης από το μηχανισμό χειρισμού προσφέρει **διαφάνεια**.
  - ❑ Με αλλαγή της γνώσης το έμπειρο σύστημα μπορεί να εκτελεί διαφορετικές λειτουργίες.
- ❖ **Κέλυφος εμπείρων συστημάτων (expert system shell):**
  - ❑ Ο συνδυασμός των βοηθητικών προγραμμάτων με το μηχανισμό εξαγωγής συμπερασμάτων.

# Βάση Γνώσης

## Knowledge Base

- ❖ Περιέχει την εμπειρογνωμοσύνη του συστήματος, όπως την εκμαίευσε ο μηχανικός γνώσης από τον άνθρωπο-ειδικό κατά την ανάπτυξη του έμπειρου συστήματος.
  - ❑ Υπάρχουν διάφορες μορφές αναπαράστασης γνώσης (π.χ. κανόνες, πλαίσια).
- ❖ Αποτελείται από δύο μέρη:
  - ❑ **Στατική:** Περιέχει διαδικασίες, κανόνες, πλαίσια που περιγράφουν το πρόβλημα και τις γνωσιολογικές διαδικασίες επίλυσής τους (αρχικά δεδομένα).
    - Δε μεταβάλλεται κατά τη διάρκεια εκτέλεσης του προγράμματος.
  - ❑ **Δυναμική:** Περιέχει μερικά συμπεράσματα και δημιουργούνται κατά την εκτέλεση του προγράμματος, καθώς και την τελική λύση του προβλήματος.
    - Χώρος εργασίας (**working memory**).

# Μηχανισμός Εξαγωγής Συμπερασμάτων

## Inference Engine

- ❖ Υπεύθυνος για το χειρισμό της βάσης γνώσης και την εξαγωγή συμπερασμάτων.
- ❖ Χωρίζεται σε δύο μέρη:
  - ❑ **Διερμηνέας (interpreter)**: Χειρισμός της υπάρχουσας γνώσης και παραγωγή νέας.
  - ❑ **Χρονοπρογραμματιστής (scheduler)**: Αποφασίζει πότε και με ποια σειρά θα χρησιμοποιηθούν οι κανόνες, επιλύοντας το πρόβλημα της **συγκρούσεως (conflict)**.
    - Στρατηγικές επίλυσης συγκρούσεων.
    - Μετα-κανόνες.

# Μηχανισμός Επεξήγησης

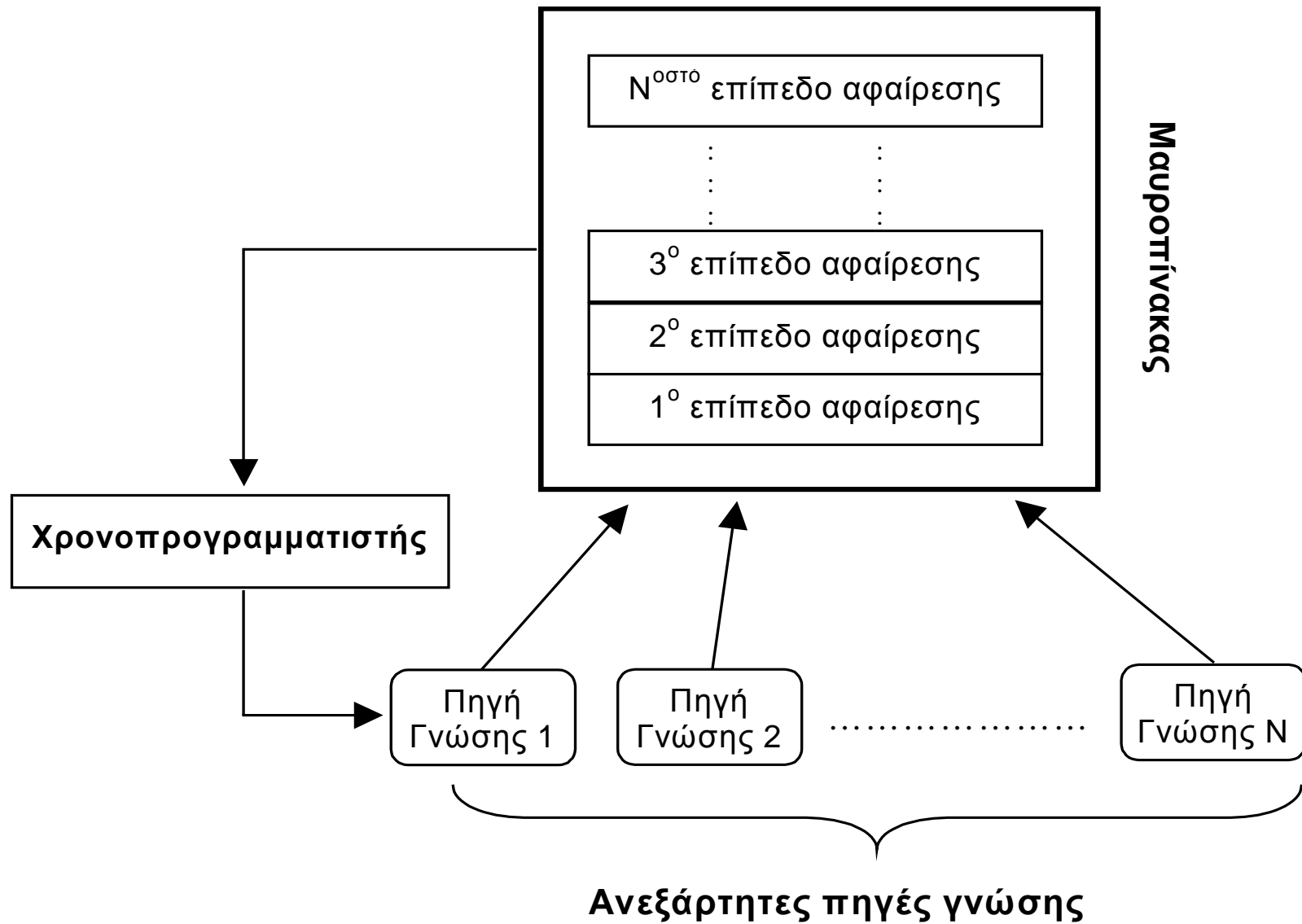
- ❖ Ο χρήστης μπορεί να κάνει ερωτήσεις στο έμπειρο σύστημα σχετικά με:
  - ❑ Τους σκοπούς των ερωτήσεων.
  - ❑ Την πορεία του συλλογισμού.
- ❖ Ο **μηχανισμός επεξήγησης** αλληλεπιδρά με το μηχανισμό εξαγωγής συμπερασμάτων
  - ❑ Η πορεία της συλλογιστικής συνδέεται άμεσα με τον τρόπο εκτέλεσης των κανόνων.
- ❖ Ο **μηχανισμός επεξήγησης** πρέπει να απαντά σε 2 ερωτήσεις:
  - ❑ **Πώς (how)** κατέληξε σε ένα συμπέρασμα
    - Κρατάει πληροφορίες σχετικά με την αποδεικτική διαδικασία και παραθέτει τους κανόνες που ενεργοποιήθηκαν σε κάθε κύκλο λειτουργίας και οδήγησαν στην απόδειξη της τρέχουσας απάντησης.
  - ❑ **Γιατί (why)** ζητά κάποια πληροφορία από το χρήστη.
    - Ψάχνει να βρει τους κανόνες που έχουν στην υπόθεσή τους την τρέχουσα πληροφορία και μπορεί να επιστρέψει όλη την κατοπινή αλυσίδα συλλογισμών που θα προκαλέσει η ενεργοποίηση αυτών των κανόνων.

# Αρχιτεκτονική Μαυροπίνακα

## Blackboard Architecture

- ❖ Η επίλυση δύσκολων προβλημάτων απαιτεί κατακερματισμό του προβλήματος σε μικρότερα και απλούστερα υποπροβλήματα, τα οποία επιλύονται ανεξάρτητα.
  - ❑ Η λύση του συνολικού προβλήματος συνδυάζει τις λύσεις των επιμέρους προβλημάτων.
  - ❑ Κάθε επιμέρους πρόβλημα ανατίθεται σε μια **πηγή γνώσης (knowledge source)**.
    - Ημιαυτόνομο έμπειρο σύστημα με τη δική του βάση γνώσης και πιθανόν διαφορετικές μορφές αναπαράστασης γνώσης και συλλογιστικές από τις υπόλοιπες πηγές.
- ❖ Ο χώρος αναζήτησης διαιρείται σε ιεραρχία επιμέρους **συνόλων μερικών λύσεων**.
  - ❑ Κάθε σύνολο χαρακτηρίζεται από διαφορετικό επίπεδο **αφαίρεσης**.
  - ❑ Π.χ., στο HEARSAY τα επιμέρους σύνολα είναι ήχοι, φθόγγοι, συλλαβές, λέξεις, ομάδες λέξεων και πραγματικές προτάσεις.
- ❖ **Μαυροπίνακας**: Κοινόχρηστη περιοχή μνήμης, που περιέχει τις μερικές λύσεις των διαφόρων επιπέδων αφαίρεσης.
  - ❑ Κάθε πηγή γνώσης παρατηρεί και τροποποιεί το περιεχόμενο του μαυροπίνακα.
  - ❑ Οι πηγές γνώσης δεν μπορούν να επικοινωνούν απευθείας μεταξύ τους.

# Μοντέλο Αρχιτεκτονικής Μαυροπίνακα

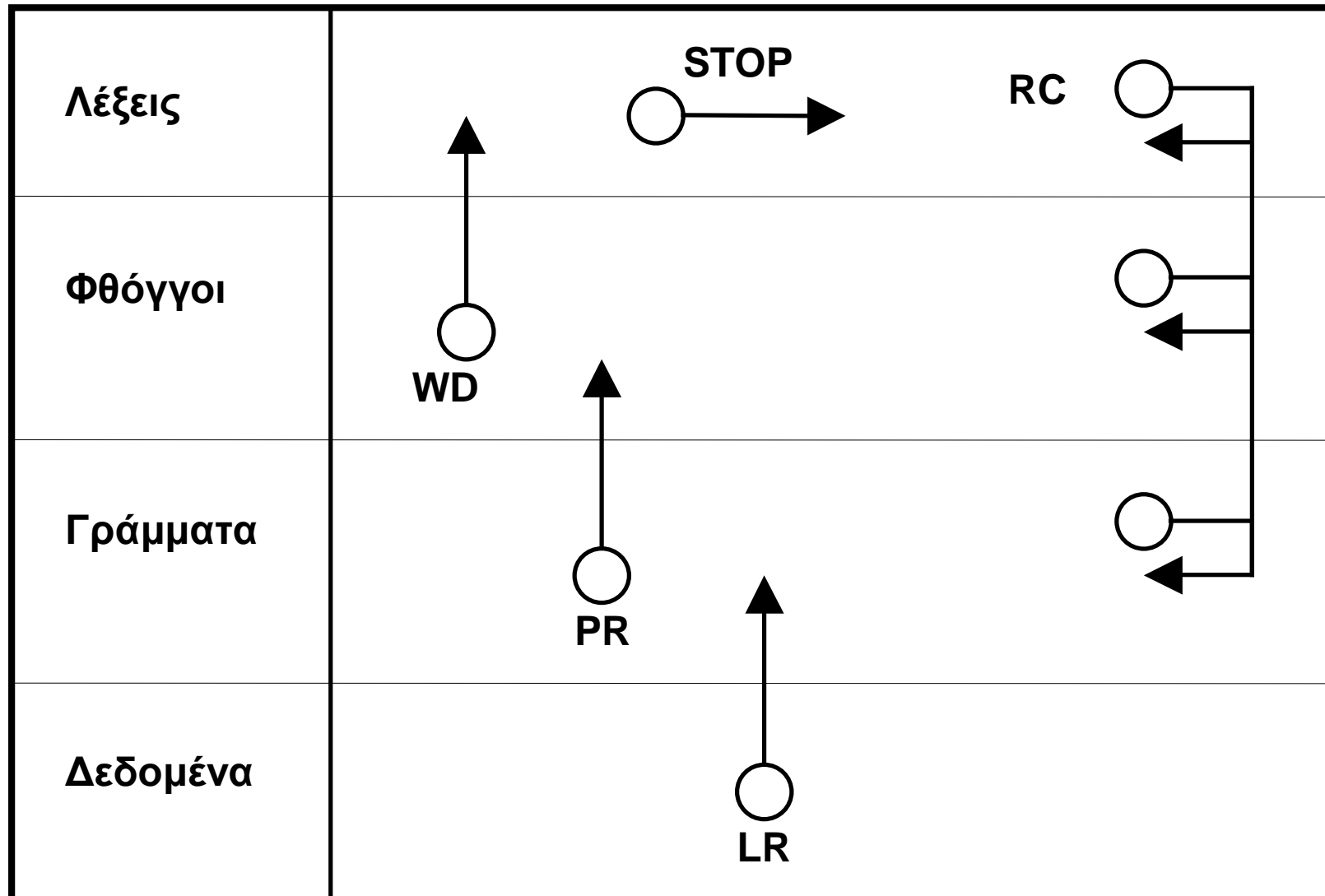


# Αρχιτεκτονική Μαυροπίνακα

## Λειτουργία Συστήματος

- ❖ Οι πηγές γνώσης λειτουργούν ταυτόχρονα.
  - ❑ Παρατηρούν τις μερικές λύσεις που υπάρχουν στο μαυροπίνακα.
  - ❑ Δημιουργούν νέες μερικές λύσεις, σε μεγαλύτερο επίπεδο λεπτομέρειας.
  - ❑ Τροποποιούν ή διαγράφουν μια υπάρχουσα μερική λύση.
    - Νέα δεδομένα που ήρθαν στο μαυροπίνακα αναιρούν τα δεδομένα που ήδη υπήρχαν σε αυτόν.
- ❖ Ο **χρονοπρογραμματιστής**:
  - ❑ Ελέγχει τα δεδομένα που υπάρχουν στο μαυροπίνακα και κρίνει σε ποια πηγή γνώσης πρέπει να επιτραπεί η πρόσβαση.
  - ❑ Διατηρεί μια **ατζέντα** με τις αιτήσεις των υπολοίπων πηγών γνώσης που ζήτησαν πρόσβαση στο μαυροπίνακα.
  - ❑ Επιτρέπει σε μία μόνο από τις πηγές γνώσης να έχει πρόσβαση στο μαυροπίνακα, σε κάθε κύκλο εκτέλεσης του συστήματος.
  - ❑ Εάν κάποια στιγμή δεν υπάρχει καμία αίτηση για πρόσβαση στο μαυροπίνακα, η λειτουργία του συνολικού συστήματος τερματίζεται.

# Παράδειγμα Επιπέδων Αφαίρεσης σε ένα Σύστημα Αναγνώρισης Ομιλίας

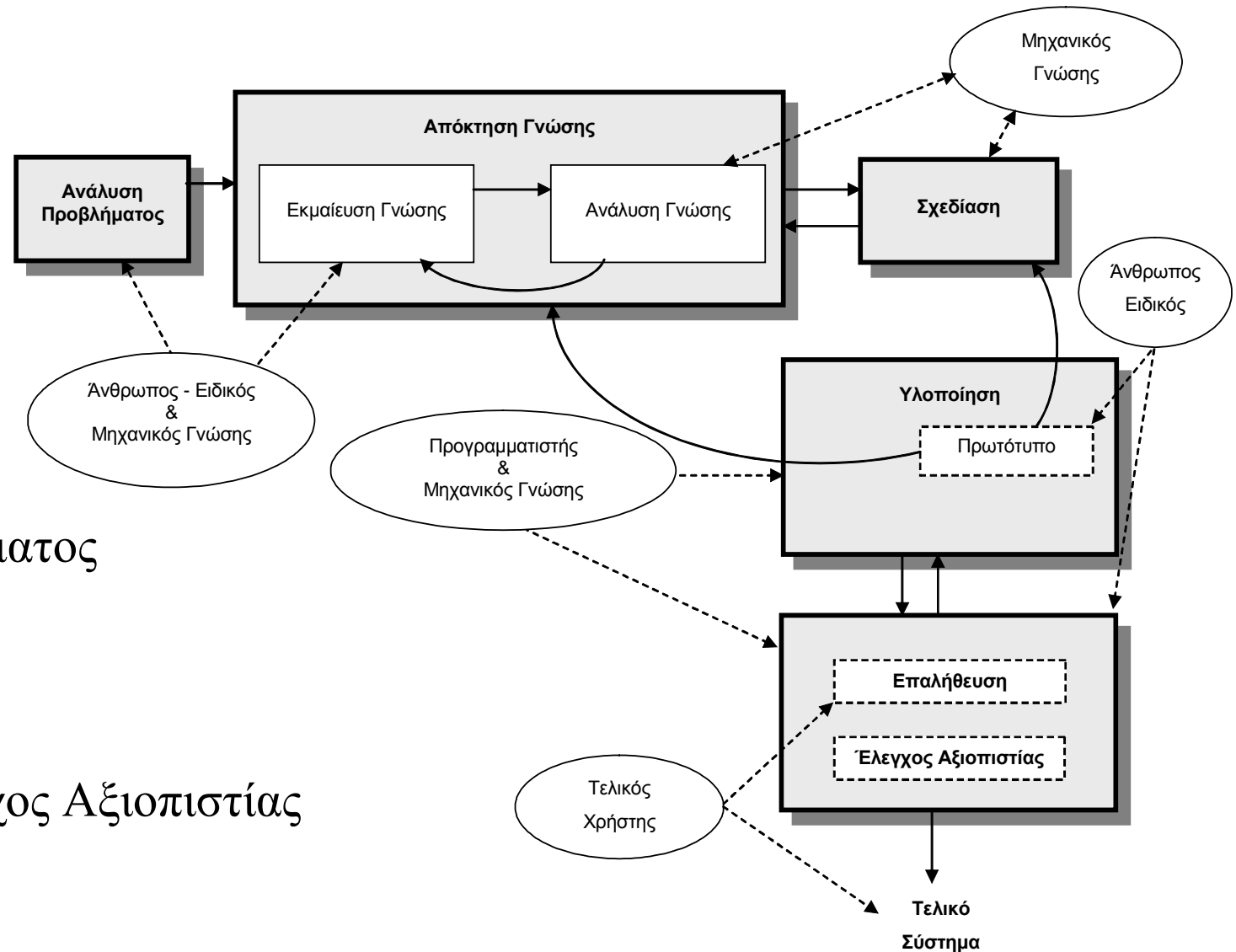




# Παράδειγμα Αναγνώρισης Λέξης

<p><b>Λέξεις</b></p>	<p>ale<sup>9</sup>      all<sup>8</sup>      ape<sup>6</sup></p> <p>del<sup>3</sup></p>
<p><b>Φθόγγοι</b></p>	<p>...al<sup>9</sup> ae<sup>8</sup> .....dp<sup>0</sup>..... le<sup>6</sup> ..... la<sup>7</sup> ..... ll<sup>7</sup>.....</p>
<p><b>Γράμματα</b></p>	<p>a<sup>9</sup> d<sup>9</sup> q<sup>7</sup>      e<sup>8</sup> l<sup>9</sup> p<sup>2</sup>      e<sup>8</sup> a<sup>2</sup> l<sup>6</sup> o<sup>3</sup> p<sup>5</sup></p>
<p><b>Δεδομένα</b></p>	<p>ale</p>

# Διαδικασία Ανάπτυξης Εμπείρων Συστημάτων



- ❖ Ανάλυση του Προβλήματος
- ❖ Απόκτηση της Γνώσης
- ❖ Σχεδίαση
- ❖ Υλοποίηση
- ❖ Επαλήθευση και Έλεγχος Αξιοπιστίας

# Ανάλυση Προβλήματος

- ❖ Προσδιορίζεται η μορφή της επιθυμητής λύσης του προβλήματος.
- ❖ Κυριότερα ζητήματα:
  - ❑ Είναι το πρόβλημα **κατάλληλο** για επίλυση από έμπειρο σύστημα ή συμβατικό πρόγραμμα;
  - ❑ Υπάρχουν έτοιμες μελέτες περιπτώσεων επίλυσης του προβλήματος (**case-studies**);
  - ❑ Ποια είναι τα **οφέλη** από την κατασκευή του έμπειρου συστήματος;

# Απόκτηση της Γνώσης

## Knowledge Acquisition

### ❖ Εκμαίευση γνώσης (**knowledge elicitation**)

- ❑ Απαιτεί συνεχή επικοινωνία ανάμεσα στο μηχανικό γνώσης και τον ειδικό.
- ❑ Εκτός από τις κλασσικές μεθόδους, υπάρχουν:
  - **Ημι-αυτόματες μέθοδοι:** Ο ειδικός εισάγει απευθείας τη γνώση στο σύστημα χρησιμοποιώντας ειδικό λογισμικό (π.χ. TEIRESIAS, OPAL, κλπ)
  - **Αυτόματες μέθοδοι:** Χρησιμοποιούνται τεχνικές μηχανικής μάθησης.

### ❖ Μοντελοποίηση γνώσης (**knowledge analysis & modeling**)

- ❑ **Ανάλυση** της γνώσης από το μηχανικό με σκοπό τη δημιουργία ενός **μοντέλου** της γνώσης.
  - Η αναπαράσταση της γνώσης γίνεται με διάφορες ημιδομημένες μορφές αναπαράστασης.
- ❑ Υπάρχουν μεθοδολογίες που τυποποιούν τη μοντελοποίηση της γνώσης, όπως η KADS.

# Σχεδίαση

- ❖ Προσδιορίζονται:
  - ❑ Η μορφή της αναπαράστασης της γνώσης.
  - ❑ Η συλλογιστική που θα χρησιμοποιηθεί για την εξαγωγή συμπερασμάτων.
  - ❑ Το εργαλείο για την ανάπτυξη του έμπειρου συστήματος.
- ❖ Παράγεται η αρχιτεκτονική του συστήματος
  - ❑ Δεσμεύσεις που λαμβάνονται υπόψη:
    - Απαιτήσεις των χρηστών.
    - Τεχνολογία που θα χρησιμοποιηθεί.
    - Το μοντέλο της γνώσης.

# Υλοποίηση

- ❖ Κωδικοποιείται το μοντέλο της γνώσης χρησιμοποιώντας εργαλεία ανάπτυξης εμπείρων συστημάτων.
- ❖ Αρχικά αναπτύσσεται ένα πρωτότυπο σύστημα επίδειξης:
  - ❑ Καθοδηγεί στη συνέχεια την ανάπτυξη, ή
  - ❑ Οδηγεί σε επανασχεδιασμό όταν δεν ικανοποιεί τις απαιτήσεις που τέθηκαν στην αρχή.
  - ❑ Επαληθεύει τη γνώση που αποκτήθηκε από τον ειδικό και μοντελοποιήθηκε από το μηχανικό γνώσης.

# Επαλήθευση και Έλεγχος Αξιοπιστίας

- ❖ Έλεγχος της συμβατότητας του συστήματος με τις αρχικές προδιαγραφές.
- ❖ Επιβεβαίωση της συνέπειας και πληρότητας της κωδικοποίησης της γνώσης που περιέχεται στο έμπειρο σύστημα.
  - ❑ Ο έλεγχος πραγματοποιείται από το μηχανικό της γνώσης με τη βοήθεια εργαλείων (π.χ. CHECK, TEIRESIAS)
- ❖ Έλεγχος αξιοπιστίας (**validation**): Συνίσταται στην επιβεβαίωση της ορθότητας και γενικότητας της γνώσης που περιέχει το έμπειρο σύστημα.
  - ❑ Το σύστημα επιλύει ένα σύνολο από υποδειγματικές περιπτώσεις (test cases).
  - ❑ Οι λύσεις συγκρίνονται με λύσεις που δόθηκαν από διάφορους ειδικούς του τομέα.
  - ❑ Οι υποδειγματικές περιπτώσεις πρέπει να είναι διαφορετικές από αυτές που χρησιμοποιήθηκαν στις προηγούμενες φάσεις ανάπτυξης του συστήματος.
    - Εξασφαλίζεται η **ευρωστία (robustness)** σε μη-προσδοκώμενα δεδομένα.

# Εκμείευση Γνώσης

## Knowledge Elicitation

- ❖ Διαδικασία απόκτησης (εξαγωγής) της γνώσης από άτομα που θεωρούνται "ειδικοί" ή "εμπειρογνώμονες".
- ❖ Θεωρείται ως το πιο δύσκολο και αμφίβολο βήμα στην ανάπτυξη ενός εμπείρου συστήματος.
- ❖ **Ειδικός** είναι το άτομο που έχει ειδική γνώση ή ικανότητα πάνω σε ένα θέμα.
  - ❑ Δεν περιορίζεται μόνο στην κατοχή πληροφοριών ή δεδομένων πάνω σε ένα θέμα.
  - ❑ Κατέχει ειδική γνώση σε βαθμό που τον κάνει να ξεχωρίζει από τους υπόλοιπους απλούς "γνώστες".
- ❖ **Γνώση**: Πλήρης επίγνωση (awareness) ή οικειότητα (familiarity) για ένα γεγονός, μια θεωρητική ή πρακτική κατανόηση ενός θέματος, η οποία αποκτάται από την εμπειρία ενός ατόμου.
  - ❑ Η εμπειρία ενός ατόμου θεωρείται αναπόσπαστο μέρος της γνώσης που κατέχει.
  - ❑ Διαφοροποίηση της σχεδίασης εμπειρων συστημάτων από την παραδοσιακή τεχνολογία λογισμικού.

# Προβλήματα στην Εκπαίδευση της Γνώσης

- ❖ **Παράδοξο της ειδίκευσης:**
  - ❑ Όσο πιο πολύ ισχυρίζεται κάποιος ότι είναι ειδικός σε κάποιο θέμα, τόσο πιο δύσκολη είναι η ανταλλαγή πληροφοριών μαζί του.
- ❖ **Ευσεβής πόθος (wishful thinking):**
  - ❑ Όταν ο ειδικός περιγράφει ένα παράδειγμα, εκφράζει ουσιαστικά το **τι θα έπρεπε να γίνεται** και όχι το **τι πραγματικά γίνεται**.
- ❖ **Κατάλληλο υπόβαθρο γνώσης του μηχανικού γνώσης.**
- ❖ **Έλλειψη χρόνου:**
  - ❑ Ο ειδικός καταλήγει σε βιαστικές απαντήσεις, όταν αυτές απαιτούν αρκετό χρόνο.
- ❖ Ο μηχανικός γνώσης πρέπει να διασφαλίσει ότι ικανοποιούνται οι **στόχοι** της συνέντευξης.
- ❖ **Αμεροληψία** του μηχανικού γνώσης.
- ❖ **Απροθυμία** του ειδικού να μεταδώσει γνώση.
- ❖ **Ανεπιτήδειος έμπειρος (inexpert expert).**



# Μεθοδολογίες Εκμαίευσης Γνώσης

- ❖ Ο πιο διαδεδομένος και αποδοτικός τρόπος εκμαίευσης γνώσης είναι η συνέντευξη
- ❖ **Μη-δομημένες** συνεντεύξεις.
  - ❑ Αποτελούνται από γενικές ερωτήσεις που υποβάλλονται με την ελπίδα της καταγραφής όσο περισσότερων πληροφοριών γίνεται.
- ❖ **Ημιδομημένες** συνεντεύξεις.
  - ❑ Περιέχουν μια σειρά ανοιχτών ερωτήσεων και θεμάτων που πρέπει να καλυφθούν.
- ❖ **Δομημένες** συνεντεύξεις.
  - ❑ Περιέχουν ένα ερωτηματολόγιο με αυστηρά καθορισμένη δομή που περιλαμβάνει συγκεκριμένες ερωτήσεις σχετικές με τα χαρακτηριστικά του προβλήματος.

# Τεχνικές Συνέντευξης (1/2)

## ❖ Επαναδιδασκαλία:

- ❑ Ο μηχανικός γνώσης προσπαθεί να επαναδημιουργήσει και να συνοψίσει ότι έχει ειπωθεί από τον ειδικό και να το διδάξει σε αυτόν.

## ❖ Διδακτική συνέντευξη (tutorial interview):

- ❑ Ο ειδικός δίνει μια διάλεξη πάνω στην περιοχή του θέματος.

## ❖ Βαθμωτά πλέγματα (laddered grids):

- ❑ Εξετάζεται ο χώρος του γνωστικού πεδίου, βάσει ερωτήσεων οι οποίες κινούνται στον άξονα γενικού-ειδικού (ή κάθετα), πάνω στον οποίο θεωρείται ότι ανήκουν οι έννοιες.

## ❖ Ταξινόμηση καρτών (card sorting):

- ❑ Ανακάλυψη κατάλληλων ιδιοτήτων των στοιχείων που απαρτίζουν την περιοχή του πεδίου, για την ταξινόμηση των εννοιών.

# Τεχνικές Συνέντευξης (2/2)

## Πλέγματα Ρεπερτορίων

### ❖ Πλέγματα ρεπερτορίων (repertory grid):

- ❑ Κάθε στοιχείο της περιοχής κατηγοριοποιείται σύμφωνα με ένα σύνολο από έννοιες ή χαρακτηρισμούς, οι οποίες εφαρμόζονται σε όλα τα στοιχεία σε κάποιο βαθμό.
- ❑ Κάθε έννοια εκφράζεται σε μια γραμμική, αριθμητική κλίμακα.
  - Η κλίμακα είναι ίδια κάθε φορά.
  - Τυπικά οι τιμές κυμαίνονται 1-5 ή 1-10
  - Υπάρχουν δύο ακραίες τιμές, π.χ. βαρύς/ελαφρύς, φτηνός/ακριβός, κ.α.
  - Η μέση τιμή (π.χ. 3 στα 5) αντιπροσωπεύει μια ενδιάμεση τιμή της έννοιας.
- ❑ Ζητείται από τον ειδικό να αποδώσει μια τιμή σε κάθε έννοια για όλα τα στοιχεία της περιοχής, στο πλέγμα που δημιουργείται.
- ❑ Εξετάζεται αν κάποιο ζευγάρι εννοιών είναι παρόμοιο κατά τη σύγκριση των οριζοντίων γραμμών του πλέγματος, ώστε να παραλειφθούν κάποιες παραπλήσιες έννοιες.
- ❑ Υπολογίζεται, σε ένα νέο πλέγμα, πόσο όμοια ή ανόμοια είναι τα στοιχεία της περιοχής μεταξύ τους.

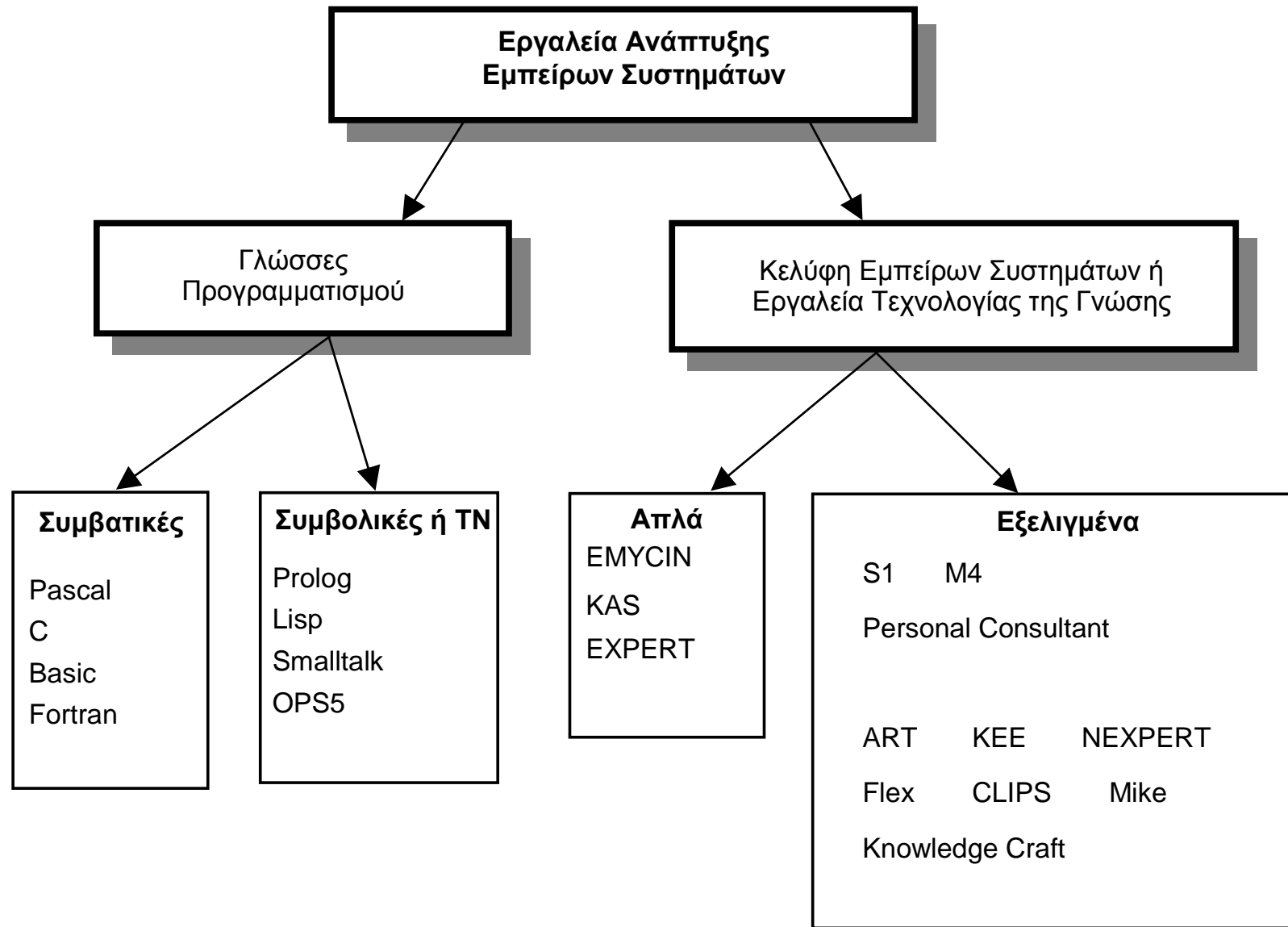
# Παράδειγμα Πλέγματος Ρεπερτορίων

	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>μικροκλοπή</p> <p>διάρρηξη</p> <p>ναρκωτικά</p> <p>δολοφονία</p> <p>ληστεία</p> <p>βιασμός</p> </div> </div>						
οποιονδήποτε	2	1	1	1	1	5	μόνο γυναίκες
μεγάλη καταδίκη	2	1	1	2	3	5	μικρή καταδίκη
ειδική τοποθεσία	2	5	1	1	4	5	οποιαδήποτε τοποθεσία
προσχεδιασμένο	5	3	1	2	5	4	αφθόρμητα
μη-απειλητικός	3	2	2	5	5	5	απειλητικός
απρόσωπο	2	2	1	5	4	5	προσωπικό
ασήμαντο	1	3	1	5	4	5	σημαντικό
μη-βίαιος	1	1	2	5	5	5	βίαιος

μικροκλοπή  
 διάρρηξη  
 ναρκωτικά  
 δολοφονία  
 ληστεία  
 βιασμός

	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>μικροκλοπή</p> <p>-</p> <p>10</p> <p>10</p> <p>18</p> <p>15</p> <p>23</p> </div> </div>						
	-	-	-	-	-	-	μικροκλοπή
	10	-	-	-	-	-	διάρρηξη
	10	10	-	-	-	-	ναρκωτικά
	18	18	16	-	-	-	δολοφονία
	15	15	21	9	-	-	ληστεία
	23	21	29	13	10	-	βιασμός

# Εργαλεία Ανάπτυξης Εμπείρων Συστημάτων



# Γλώσσες Προγραμματισμού TN

- ❖ Αποτελούν εργαλείο για γρήγορη κατασκευή πρωτοτύπου του έμπειρου συστήματος.
  - ❑ Ο κώδικας μπορεί να εκτελεστεί και να ελεγχθεί την ώρα που δημιουργείται.
- ❖ Η διασύνδεση αυτών των γλωσσών με το χρήστη δεν είναι αρκετά εξελιγμένη.
- ❖ Οι γλώσσες προγραμματισμού διαθέτουν συνήθως έναν απλό μηχανισμό ελέγχου.
  - ❑ Δίνουν στον προγραμματιστή τη δυνατότητα να "δημιουργήσει":
    - Μηχανισμό ελέγχου για το έμπειρο σύστημα (συλλογιστική).
    - Τρόπο αναπαράστασης της γνώσης, με τις δομές δεδομένων.
- ❖ Κατηγορίες γλωσσών προγραμματισμού TN:
  - ❑ Συναρτησιακός προγραμματισμός (π.χ. LISP)
  - ❑ Λογικός προγραμματισμός (π.χ. PROLOG)
  - ❑ Αντικειμενοστραφής προγραμματισμός (π.χ. SMALLTALK)
  - ❑ Προγραμματισμός με κανόνες παραγωγής (π.χ. OPS5)

# Συναρτησιακός Προγραμματισμός (LISP)

- ❖ Είναι προσανατολισμένη στο χειρισμό συμβόλων και λιστών.
- ❖ **Διαδικαστική θεώρηση:**
  - ❑ Η γνώση του προβλήματος αναμειγνύεται με τη γνώση του τρόπου επίλυσης.
- ❖ Έχει μόνο 2 τύπους δεδομένων (άτομο-σύμβολο και **λίστα**).
- ❖ Το λεξιλόγιο αποτελείται από 6 θεμελιώδεις συναρτήσεις.
  - ❑ Με βάση αυτές τις στοιχειώδεις συναρτήσεις ο χρήστης ορίζει πιο σύνθετες.
- ❖ **Πλεονεκτήματα:**
  - ❑ Δυναμικότητα, αυτόματη διαχείριση της μνήμης, εύκολη διόρθωση λαθών.
  - ❑ Δυνατότητα για ορισμό αναδρομικών συναρτήσεων.
  - ❑ **Τμηματοποιημένη (modular)** ανάπτυξη προγραμμάτων με τη χρήση πολλών συναρτήσεων.
- ❖ **Μειονέκτημα:**
  - ❑ Δεν αποδίδει κάποια ιδιαίτερη σημασία στα σύμβολα που χειρίζεται.
    - Μέσο "κατασκευής" εργαλείων ανάπτυξης εμπείρων συστημάτων (π.χ. OPS5).
    - Όχι εργαλείο άμεσης κωδικοποίησης και εκτέλεσης της γνώσης.

# Λογικός Προγραμματισμός (PROLOG)

- ❖ Συμβολική γλώσσα που βασίζεται στην κατηγορηματική λογική πρώτης τάξης.
- ❖ Ένα πρόβλημα:
  - ❑ Περιγράφεται με τη μορφή γεγονότων (αξιώματα) και κανόνων (θεωρήματα)
  - ❑ Δεν περιέχει τον ακριβή αλγόριθμο επίλυσης του προβλήματος.
- ❖ Υπάρχει σαφής διαχωρισμός της γνώσης από το μηχανισμό ελέγχου.
  - ❑ Η εξαγωγή συμπερασμάτων γίνεται με τη διαδικασία της **εις άτοπο απαγωγής**.
  - ❑ Οι εναλλακτικές λύσεις ενός προβλήματος ερευνώνται πρώτα σε βάθος (DFS).
- ❖ **Υπόθεση κλειστού κόσμου (closed-world assumption):**
  - ❑ Όσες πληροφορίες δεν αναφέρονται ρητά μέσα στη βάση γνώσης θεωρείται ότι δεν αληθεύουν (ψευδείς).
- ❖ Κατάλληλη για την ανάπτυξη εμπείρων συστημάτων:
  - ❑ Οι κανόνες περιγράφουν έναν κανόνα εμπείρου συστήματος.
  - ❑ **Μηχανισμός ελέγχου:** Ανάστροφη ακολουθία εκτέλεσης κανόνων (**backward chaining**).
  - ❑ **Μηχανισμός επίλυσης συγκρούσεων:** Επιλέγεται πάντα ο πρώτος κανόνας ή γεγονός.
  - ❑ Στις μοντέρνες εκδόσεις της PROLOG υπάρχουν αρκετές επεκτάσεις γραφικής διασύνδεσης με το χρήστη.



# Αντικειμενοστραφής Προγραμματισμός (1/2)

## Object-Oriented Programming

### ❖ Αντικείμενο (object):

- ❑ Συλλογή από συσχετιζόμενα δεδομένα (**χαρακτηριστικά - attributes**) με συγκεκριμένη δομή, που αντιπροσωπεύει συνήθως μία οντότητα του φυσικού κόσμου.
- ❑ Π.χ., ένα αυτοκίνητο μπορεί να αναπαρασταθεί ως αντικείμενο με χαρακτηριστικά όπως ο κατασκευαστής του, ο αριθμός θυρών και θέσεων, η μέγιστη ταχύτητα, το βάρος του, κλπ.

### ❖ Εγκλεισμός (encapsulation):

- ❑ Η εσωτερική κατάσταση του αντικειμένου αποκρύπτεται από τον υπόλοιπο κόσμο.
- ❑ Τα χαρακτηριστικά του αντικειμένου προσπελούνται μόνο μέσω **μήνυμάτων**.
- ❑ Για κάθε μήνυμα υπάρχει μία διαδικασία εξυπηρέτησής του, η οποία ονομάζεται **μέθοδος**.

### ❖ Κλάσεις (classes):

- ❑ Αντικείμενα-πρότυπα που περιγράφουν αφηρημένα τη δομή και τη συμπεριφορά των αντικειμένων ομαδοποιούν τους

### ❖ Στιγμιότυπα (instances):

- ❑ Το σύνολο των αντικειμένων που ανήκουν σε μία κλάση.

### ❖ Οι κλάσεις είναι συνήθως οργανωμένες σε ιεραρχίες γενίκευσης/ειδίκευσης.

- ❑ Π.χ., η κλάση "όχημα" είναι πιο γενική από την κλάση "αυτοκίνητο".

# Αντικειμενοστραφής Προγραμματισμός (2/2)

## ❖ Κληρονομικότητα (inheritance):

- ❑ Η δομή και συμπεριφορά μιας γενικότερης κλάσης κληρονομείται στις περισσότερο συγκεκριμένες.

## ❖ Πλεονέκτημα:

- ❑ Προσφέρει εκφραστικές δομές για συμπαγή και συνεπή αναπαράσταση των:
  - Των αντικειμένων του φυσικού κόσμου.
  - Των συσχετίσεων μεταξύ τους,
  - Των διεργασιών που λαμβάνουν χώρα.

## ❖ Μειονέκτημα:

- ❑ Η επιλογή των αντικειμένων και μηνυμάτων γίνεται τεχνητά/αφύσικα για τα περισσότερα προβλήματα.
  - Τα αντικείμενα δε διευκολύνουν τον προγραμματιστή, αλλά αποτελούν απλώς το μέσο υλοποίησης

## ❖ Πιο γνωστές γλώσσες:

- ❑ SMALLTALK (προσεγγίζει το συμβολικό προγραμματισμό)
- ❑ C++/Java (προσεγγίζει το συμβατικό προγραμματισμό)
- ❑ LOOPS, FLAVORS και CLOS (αντικειμενοστραφείς επεκτάσεις της γλώσσας LISP)
- ❑ COOL (συνοδεύει τη γλώσσα παραγωγής CLIPS - πάρα πολλά κοινά σημεία με την CLOS)

# Παράδειγμα Αντικειμενοστραφούς Προγραμματισμού

## COOL (1/2)

### ❖ Ορισμός κλάσεων:

```
(defclass vehicle
  (is-a USER)
  (slot fuel-type (create-accessor read-write))
  (slot tank-capacity (create-accessor read-write))
  (slot fuel-loaded (create-accessor read-write)))
(defclass car
  (is-a vehicle)
  (slot consumption-rate (create-accessor read-write))
  (slot reset-counter (create-accessor read-write)))
```

### ❖ Ορισμός στιγμιότυπου:

```
(definstances car
  (fiat_brava of car
    (fuel-type 'benzine)
    (tank-capacity 45)
    (fuel-loaded 30)
    (consumption-rate 10)
    (reset-counter 250)))
```

# Παράδειγμα Αντικειμενοστραφούς Προγραμματισμού

## COOL (1/2)

### ❖ Ορισμός μεθόδου:

```
(defmessage-handler car remaining-distance ()  
  (- (* (/ ?self:fuel-loaded  
          ?self:consumption-rate)  
       100)  
     ?self:reset-counter))
```

### ❖ Αποστολή μηνύματος:

```
(send [fiat_brava] remaining-distance)
```

# Διαφορές Αντικειμένων - Πλαισίων

## ❖ Ρόλος των κλάσεων:

### ❑ Αντικείμενα: Τύποι δεδομένων.

- Τα αντικείμενα έχουν αυστηρά καθορισμένη δομή, ώστε να διευκολύνεται ο έλεγχος της ορθότητας των προγραμμάτων και η παραγωγή αποδοτικότερου κώδικα.

### ❑ Πλαίσια: Πρότυπα με προκαθορισμένα χαρακτηριστικά και τιμές.

- Τα στιγμιότυπα των πλαισίων δεν είναι υποχρεωμένα να ακολουθήσουν αυστηρά τα προτεινόμενα χαρακτηριστικά και τιμές (μεγαλύτερη ευελιξία).

## ❖ Πρόσβαση στις τιμές των ιδιοτήτων:

### ❑ Αντικείμενα: Μόνο μέσω μηνυμάτων (εγκλεισμός).

### ❑ Πλαίσια: Άμεση πρόσβαση για ανάγνωση και εγγραφή από τους κανόνες.

## ❖ Ενσωμάτωση διαδικασιών:

### ❑ Αντικείμενα: Υπάρχουν ενσωματωμένα ολόκληρα προγράμματα υπό τη μορφή μεθόδων.

- Ενεργοποιούνται με την εκούσια αποστολή μηνυμάτων.

### ❑ Πλαίσια: Προσκολλώνται μικρές διαδικασίες (**δαίμονες**) σε κάποιες ιδιότητες.

- Ενεργοποιούνται αυτόματα όταν γίνει πρόσβαση στις ιδιότητες.
- Το υπόλοιπο πρόγραμμα είναι αποθηκευμένο εκτός των πλαισίων

# Προγραμματισμός με Κανόνες Παραγωγής (OPS5)

- ❖ Αποτελεί φυσικό τρόπο υλοποίησης της γνώσης.
- ❖ Είναι εύκολος στην εκμάθηση, λόγω της απλότητας των δομών και της γλώσσας.
- ❖ Η απλότητα είναι αρκετές φορές περιοριστική, αφού δεν επιτρέπει τη δημιουργία σύνθετων αναπαραστάσεων της γνώσης ή απλών διαδικαστικών αλγορίθμων.
  - ❑ Π.χ., στην **Ops5** υπάρχουν μόνο γραμμικές εγγραφές (διανύσματα συμβόλων).
    - Είναι δύσκολο να δημιουργηθούν σύνθετες (αναδρομικές) δομές, όπως δένδρα και γράφοι.
    - Η δημιουργία απλών βρόχων επανάληψης ή αναδρομής είναι πολύπλοκη διαδικασία.
- ❖ Χρησιμοποιείται ο **αλγόριθμος Rete**
  - ❑ Επιταχύνει την **ταυτοποίηση (pattern matching)** των κανόνων με τα δεδομένα στη μνήμη εργασίας
- ❖ Μηχανισμός ελέγχου:
  - ❑ Βασίζεται στη συγκέντρωση των ενεργοποιημένων κανόνων στο **σύνολο συγκρούσεων**.
  - ❑ Εφαρμόζονται ευριστικοί αλγόριθμοι **επίλυσης συγκρούσεων** για την επιλογή του κανόνα που θα εκτελεστεί.
    - Βασίζονται σε ψυχολογικές μελέτες (Carnegie-Mellon) πάνω στον τρόπο με τον οποίο σκέφτονται οι άνθρωποι (**cognitive psychology**).

# Αλγόριθμοι Επίλυσης Συγκρούσεων της OPS5

- ❖ Ο κανόνας που αναφέρεται στα πιο πρόσφατα δεδομένα εκτελείται πρώτος (**recency**).
  - ❑ Ακολουθείται μία χρονικά συνεπής πορεία σκέψης.
- ❖ Ο πιο συγκεκριμένος κανόνας εκτελείται πρώτος (**specificity**).
  - ❑ Εξετάζονται πρώτα τα πιο συγκεκριμένα θέματα τα οποία οδηγούν πιθανότατα σε λύση πιο γρήγορα.
- ❖ Οι κανόνες που αναφέρονται στον τρέχοντα **στόχο** έχουν προτεραιότητα (**means-ends analysis**).
  - ❑ Το συνολικό πρόβλημα επιμερίζεται σε απλούστερες **διεργασίες (tasks)**.
  - ❑ Κάθε διεργασία υλοποιείται από μία ομάδα κανόνων (**cluster**), υποσύνολο της συνολικής βάσης γνώσης.
  - ❑ Η αποδεικτική διαδικασία είναι επικεντρωμένη στους τρέχοντες στόχους της.
- ❖ Οι κανόνες που έχουν εκτελεστεί για κάποιο συνδυασμό αντικειμένων δεν ξαναεκτελούνται (**refractoriness**).
  - ❑ Αποφεύγονται άσκοπες ή ατέρμονες επαναλήψεις.

# Παράδειγμα Προγράμματος στην OPS5 (1/2)

❖ Κίνηση ρομπότ σε χώρο με εμπόδια και αντικείμενα-στόχους.

```
(strategy lex)
```

```
(literalize robot_at x_pos y_pos)
```

```
(literalize obstacle_at x_pos y_pos)
```

```
(literalize direction name)
```

```
(literalize object_at x_pos y_pos)
```

```
(literalize test_obstacle x_pos y_pos)
```

```
(literalize choice name)
```

```
(p begin
```

```
  (start)
```

```
  →
```

```
  (make robot_at ^x_pos 6 ^y_pos 4) (make direction ^name e) (make choice ^name w)
```

```
  (make obstacle_at ^x_pos 7 ^y_pos 4)
```

```
  ...
```

```
  (make object_at ^x_pos 4 ^y_pos 7)
```

```
  ...)
```

```
(p move_west
```

```
  (robot_at ^x_pos <x> ^y_pos <y>)
```

```
  (direction ^name w)
```

```
  →
```

```
  (modify 1 ^x_pos (compute <x> - 1)))
```

```
  ...
```



# Παράδειγμα Προγράμματος στην OPS5 (2/2)

```
(p avoid_obstacle_south1
```

```
  (robot_at ^x_pos <x> ^y_pos <y>)
```

```
→
```

```
  (make_test_obstacle ^x_pos <x> ^y_pos (compute <y> - 1)))
```

```
(p avoid_obstacle_south1
```

```
  (test_obstacle ^x_pos <x> ^y_pos <y>)
```

```
  (obstacle_at ^x_pos <x> ^y_pos <y>)
```

```
  (direction ^name s)
```

```
  (choice ^name <ND>)
```

```
→
```

```
  (modify 3 ^name <ND>))
```

```
...
```

```
(p detect_object
```

```
  (robot_at ^x_pos <x> ^y_pos <y>)
```

```
  (object_at ^x_pos <x> ^y_pos <y>)
```

```
→
```

```
  (write 'object is found' (crlf))
```

```
  (halt))
```

# Κελύφη Εμπείρων Συστημάτων

- ❖ Εξειδικευμένα εργαλεία για την ανάπτυξη εμπείρων συστημάτων.
- ❖ Αποτελούν έμπειρα συστήματα χωρίς τη βάση γνώσης.
  - ❑ Μπορούν να παρομοιαστούν ως περιβλήματα (κελύφη) μιας βάση γνώσης.
- ❖ Προέρχονται από:
  - ❑ Υπάρχοντα έμπειρα συστήματα με αφαίρεση της αρχικής βάσης γνώσης τους.
  - ❑ Αναπτύσσονται από την αρχή και δεν προσανατολίζονται σε συγκεκριμένη εφαρμογή.
- ❖ Διαχωρίζονται σε απλά ή εξελιγμένα βάσει σημαντικών χαρακτηριστικών τους:
  - ❑ Μορφές αναπαράστασης γνώσης και συλλογιστικής.
  - ❑ Ευκολίες διασύνδεσης.

# Απλά Κελύφη (Emycin)

- ❖ Προήλθε από το έμπειρο σύστημα MYCIN.
  - ❑ Χρησιμοποιήθηκε για την κατασκευή διαφόρων εμπείρων συστημάτων:  
LITHIO (γεωλογία)      CLOT (ασθένειες πήξης του αίματος)      PUFF (πνευμονικές ασθένειες)  
TAX-ADVISOR (νομική)      SACON (ανάλυση μηχανολογικών σχεδίων)      HEAD MED (ψυχιατρικές διαγνώσεις)
- ❖ Τα γεγονότα παριστάνονται σαν τριάδες: "**έννοια-παράμετρος-τιμή**".
  - ❑ Κάθε τριάδα συνοδεύεται και από ένα **συντελεστή βεβαιότητας**.
- ❖ Οι κανόνες εκτελούνται **ανάστροφα**:
  - ❑ Όταν επαληθεύεται η **συνθήκη** τότε προστίθενται στη μνήμη οι τριάδες της **ενέργειας**.
    - Η ενέργεια συνοδεύεται με συντελεστές βεβαιότητας.
- ❖ Υποστηρίζονται **μετα-κανόνες**:
  - ❑ Εξετάζουν τις συνθήκες των κανόνων που μπορούν να εκτελεστούν.
  - ❑ Καθορίζουν τη σειρά εκτέλεσης, ή αποτρέπουν την εκτέλεση κάποιων κανόνων.
- ❖ **Μειονέκτημα**: Δεν είναι κατάλληλα για την επίλυση όλων των προβλημάτων, αφού δημιουργήθηκαν από έμπειρο σύστημα που επιλύει συγκεκριμένο πρόβλημα.
  - ❑ Π.χ. το EMYCIN είναι κατάλληλο κυρίως για προβλήματα διάγνωσης (όπως το MYCIN).
  - ❑ Οφείλεται στην υποστήριξη ενός είδους αναπαράστασης γνώσης, συλλογιστικής και αβεβαιότητας.

## Εξελιγμένα Κελύφη (1/2)

- ❖ Εργαλεία της τεχνολογίας της γνώσης τα οποία υποστηρίζουν:
  - ❑ Πολλαπλούς τρόπους αναπαράστασης γνώσης και συλλογιστικής.
  - ❑ Εξελιγμένες δυνατότητες διασύνδεσης.
  - ❑ Μεγαλύτερη γενικότητα εφαρμογών.
- ❖ Κατηγορίες εξελιγμένων κελυφών, ανάλογα με τον τρόπο που αναπτύχθηκαν:
  - ❑ Αρχικά από έμπειρα συστήματα και στη συνέχεια μετατράπηκαν και επεκτάθηκαν ώστε να υποστηρίζουν περισσότερα είδη αναπαράστασης γνώσης και συλλογιστικής.
    - Π.χ. PERSONAL CONSULTANT, S1 και M4, κ.ά.
  - ❑ Από την αρχή σε κάποια γλώσσα προγραμματισμού και δεν προήλθαν από κάποιο έμπειρο σύστημα.
    - ART, KEE, και KNOWLEDGE CRAFT (αναπτύχθηκαν σε LISP)
    - FLEX και MIKE (αναπτύχθηκαν σε PROLOG),
    - NEXPERT και CLIPS (αναπτύχθηκαν σε C/C++)

## Εξελιγμένα Κελύφη (2/2)

- ❖ Διαφορές με τα απλά κελύφη
  - ❑ Δε δεσμεύονται από τη δόμηση και τους περιορισμούς του αρχικού έμπειρου συστήματος.
  - ❑ Δίνουν περισσότερες δυνατότητες στην κατασκευή και τη συντήρηση των εμπείρων συστημάτων.
  - ❑ Είναι συνήθως δυσκολότερα στην εκμάθηση γιατί περιέχουν πολλές και ετερογενείς μεταξύ τους προγραμματιστικές έννοιες.
    - Η ανάμειξη διαφόρων προγραμματιστικών μοντέλων γίνεται πειραματικά από τους προγραμματιστές, γιατί δεν υπάρχουν θεωρίες υβριδικών τρόπων αναπαράστασης της γνώσης.
- ❖ Οι κυριότεροι μέθοδοι αναπαράστασης γνώσης:
  - ❑ **Πλαίσια:** Πλεονεκτούν στη δομημένη αναπαράσταση σύνθετων φυσικών αντικειμένων και στο συμπαγή τρόπο χειρισμού τους.
  - ❑ **Κανόνες:** Πλεονεκτούν στο δηλωτικό τρόπο αναπαράστασης εμπειρικών συσχετίσεων μεταξύ παρατηρηθέντων δεδομένων και επαρκούντων δράσεων για την αντιμετώπιση των περιπτώσεων.
- ❖ Ο συνδυασμός των δύο μεθόδων αναπαράστασης γνώσης γίνεται ως εξής:
  - ❑ Οι συνθήκες των κανόνων μπορούν να αναφέρονται σε τιμές των ιδιοτήτων των πλαισίων.
  - ❑ Οι ενέργειες μπορούν να αλλάζουν τις τιμές των ιδιοτήτων ή να δημιουργούν και να διαγράφουν πλαίσια.

# PERSONAL CONSULTANT

- ❖ Είναι εξέλιξη του απλού κελύφους EMYCIN.
  - ❑ Η αναπαράσταση γνώσης γίνεται με κανόνες και γεγονότα.
    - Τα γεγονότα παριστάνονται με τριάδες όπως και στο EMYCIN.
  - ❑ Υποστηρίζει αβεβαιότητα
- ❖ Υποστηρίζει τόσο ανάστροφη όσο και ορθή συλλογιστική
- ❖ Εξελιγμένες δυνατότητες:
  - ❑ Εξελιγμένος επεξεργαστής κειμένου για τη βάση γνώσης.
  - ❑ Δυνατότητα επεξήγησης της πορείας συλλογισμού.
  - ❑ Πρόγραμμα επεξεργασίας φυσικής γλώσσας.
  - ❑ Εργαλείο για τον έλεγχο της συνέπειας της βάσης γνώσης.

# ART

## Automated Reasoning Tool

- ❖ Προσφέρει διάφορους τρόπους αναπαράστασης της γνώσης.
  - Κανόνες.
    - Μπορούν να εκτελεστούν με ορθή ή ανάστροφη συλλογιστική.
  - Σενάρια.
  - κλπ.
- ❖ Προσφέρει εξελιγμένους μηχανισμούς ελέγχου,
  - Αρχιτεκτονική **μαυροπίνακα**
  - Μηχανισμός συντήρησης της αλήθειας (truth maintenance)**
    - Καταγράφεται ο δρόμος συλλογισμού που ακολουθεί η ικανοποίηση της υπόθεσης ενός κανόνα, έτσι ώστε αν καταρριφθεί η τελική υπόθεση, διαγράφονται όλες οι σχετιζόμενες υποθέσεις.
- ❖ Εξελιγμένες δυνατότητες:
  - Γραφικό εργαλείο ART Studio για τη σταδιακή ανάπτυξη της βάσης γνώσης μέσω παραθύρων, μενού, κλπ.
  - Δυνατότητα κλήσης προγραμμάτων που γράφηκαν σε άλλες γλώσσες προγραμματισμού, π.χ. LISP, C, PASCAL.

# ΚΕΕ

## Knowledge Engineering Environment

- ❖ Η αναπαράσταση γνώσης γίνεται με **μονάδες (units)** ή **πλαίσια**.
- ❖ Η εκτέλεση κανόνων μπορεί να είναι ορθή ή ανάστροφη, με μηχανισμό οπισθοδρόμησης.
- ❖ Εξελιγμένες δυνατότητες:
  - ❑ Εργαλείο σύνταξης της βάσης γνώσης.
  - ❑ Εργαλείο ΚΕΕ Worlds που δίνει τη δυνατότητα για αναπαράσταση και σύγκριση εναλλακτικών σεναρίων.
  - ❑ Εργαλείο TMS ως μηχανισμός **συντήρησης** της αλήθειας.
  - ❑ Εργαλείο ΚΕΕ Connection, για διασύνδεση με σχεσιακές βάσεις δεδομένων.