

ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ

ΣΧΟΛΗ: ΣΔΟ

**ΤΜΗΜΑ: ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΣ ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ
ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ**

**ΘΕΜΑ: ΜΕΘΟΔΟΙ ΑΝΑΠΑΡΑΣΤΑΣΗΣ ΓΝΩΣΗΣ ΣΤΑ
ΕΜΠΕΙΡΑ ΣΥΣΤΗΜΑΤΑ. ΚΕΛΥΦΗ ΕΜΠΕΙΡΩΝ
ΣΥΣΤΗΜΑΤΩΝ ΣΥΓΚΡΙΤΙΚΗ ΜΕΛΕΤΗ ΑΥΤΩΝ**

ΕΙΣΗΓΗΤΗΣ: κα Β. ΣΩΤΗΡΟΠΟΥΛΟΥ

ΕΠΙΜΕΛΕΙΑ: ΜΑΡΙΕΛΛΑ ΠΕΤΡΙΔΟΥ

ΗΜΕΡΟΜΗΝΙΑ: 09/07/2006

ΠΡΟΛΟΓΟΣ

Το θέμα με το οποίο θα πραγματευτεί η παρακάτω εργασία είναι τα έμπειρα συστήματα (expert systems) και τα κέλυφη έμπειρων συστημάτων (expert system shells). Ο σκοπός αυτής της μελέτης είναι να δοθεί μια επισκόπηση στο θέμα και να αναλυθούν τα έμπειρα συστήματα, το κέλυφος έμπειρων συστημάτων, η ασαφής λογική, η τεχνητή νοημοσύνη.

Οι ηλεκτρονικοί υπολογιστές έχουν καθιερωθεί ως ένα ισχυρό εργαλείο για την αντιμετώπιση πολύπλοκων και χρονοβόρων υπολογιστικών διαδικασιών. Πρόσφατα άρχισαν οι προσπάθειες με σκοπό την εκμετάλλευση των νέων δυνατοτήτων της επιστήμης της πληροφορικής, με κυριότερο στόχο την ανάπτυξη πληροφοριακών συστημάτων τα οποία θα έχουν την δυνατότητα να σκέπτονται, προσομοιώνοντας την ανθρώπινη λογική και σκέψη.

Πριν σκιαγραφήσουμε τα χαρακτηριστικά γνωρίσματα των έμπειρων συστημάτων πρέπει να κοιτάξουμε στην έρευνα που ταξινομείται αόριστα ως τεχνητή νοημοσύνη. Η απαρχή της έρευνας στον τομέα των Έμπειρων Συστημάτων συνέπεσε με την ουσιαστική εγκατάλειψη εκ μέρους των ερευνητών της Τεχνητής Νοημοσύνης της αναζήτησης μεθόδων γενικής ισχύος για τη λύση των πραγματικών προβλημάτων (real world problems).

Τα έμπειρα συστήματα έχουν εξελιχθεί από έρευνες μιας μακροχρόνιας παράδοσης τεχνητής νοημοσύνης. Ο τομέας ονομάζεται τεχνητή νοημοσύνη επειδή όλοι οι υπό-τομείς του έχουν ως κοινό στόχο τις διαδικασίες της ανθρώπινης νοημοσύνης μέσω ενός υπολογιστή.

ΙΣΤΟΡΙΚΗ ΑΝΑΣΚΟΠΙΣΗ

Το ενδιαφέρον της ανθρωπότητας για την ανάλυση των σκεπτόμενων διαδικασιών, στην κατανόηση της εργασίας του ανθρώπινου εγκεφάλου, χρονολογείται πίσω αιώνες. Η εποχή της αναζήτησης γενικών μεθόδων υπήρξε και η εποχή των μεγάλων ελπίδων στον τομέα της Τεχνητής Νοημοσύνης και χαρακτηρίστηκε από την προσπάθεια πλήρους μεταφύτευσης της ανθρώπινης νοημοσύνης στους υπολογιστές. Η περίοδος αυτή διάρκεσε περίπου μέχρι τα τέλη της δεκαετίας του 1960. Η έρευνα που έγινε οδήγησε σε πολλές ικανοποιητικές μεθόδους επίλυσης σαφώς καθορισμένων και σαφώς εργαστηριακών προβλημάτων, όχι όμως και προβλημάτων του πραγματικού κόσμου.

Οι σοβαρές απλοποιητικές παραδοχές που γίνονται στα εργαστηριακά προβλήματα, οδηγούν σε μεθόδους, οι οποίες συνήθως δεν αποδίδουν ικανοποιητικά, όταν εφαρμόζονται στα σύνθετα πραγματικά προβλήματα. Επιπλέον οι μέθοδοι που αρμόζει να ακολουθήσουν για την παράσταση και οργάνωση της γνώσης και για την επίλυση των προβλημάτων, συνδέονται αναπόσπαστα με το μέγεθος και την πολυπλοκότητα των προβλημάτων αυτών.

Διαδέχτηκε μια περίοδος της οποίας το κύριο χαρακτηριστικό υπήρξε η αναζήτηση γενικών μεθόδων παράστασης της γνώσης. Αυτή οι μέθοδοι συνδυάζονταν με μηχανισμούς αναζήτησης λύσεων σε ειδικότερα προβλήματα. Η εμπειρία που αποκτήθηκε από όλη την μέχρι τότε έρευνα, οδήγησε στο συμπέρασμα ότι η ικανότητα επίλυσης προβλημάτων πηγάζει από τη μάζα της γνώσης η οποία διατίθεται.

Οι διαπιστώσεις αυτές είχαν σαν αποτέλεσμα την στροφή της έρευνας προς την επίλυση τελείως εξειδικευμένων προβλημάτων, τα οποία απαιτούν την επεξεργασία μεγάλης μάζας ειδικευμένης γνώσης. Τέτοια προβλήματα αναφέρονται μεταξύ των άλλων και σε συγκεκριμένους τομείς επιστημών, όπως η ιατρική, η ηλεκτρονική κ.λ.π. Τα προγράμματα, τα οποία αντιμετωπίζουν τέτοιου είδους προβλήματα ονομάζονται Έμπειρα Συστήματα.

ΕΙΣΑΓΩΓΗ

Το ενδιαφέρον για τη διαμόρφωση της ανθρώπινης νοημοσύνης αναπτύχθηκε πολύ πριν από την εφεύρεση των υπολογιστών. Αλλά από την αυγή της εποχής υπολογιστών, η ερευνητική εστίαση έχει επεκταθεί. Δεδομένου ότι οι γνωστικοί επιστήμονες μαθαίνουν για τις ευφείς διαδικασίες συμπεριφοράς, προσπαθούν να προσδιορίσουν τα συστατικά και τις διαδικασίες, κατόπιν να τις κωδικοποιήσουν στη σειρά συστηματικών βημάτων. Τα βήματα μπορούν έπειτα να προγραμματιστούν σε έναν υπολογιστή και να μιμηθούν. Η ανατροφοδότηση που παρέχεται από τους ανθρώπους που χρησιμοποιούν εκείνα τα συγκροτήματα ηλεκτρονικών υπολογιστών επιτρέπει έπειτα στους ερευνητές να καθορίσουν περαιτέρω τα πρότυπα νοημοσύνης τους.

Η τεχνητή νοημοσύνη εστιάζει στις συμβολικές, μη αλγοριθμικές μεθόδους επίλυσης προβλήματος. Η νοημοσύνη στηρίζεται στη δυνατότητά μας να χειριστεί σύμβολα, που περιλαμβάνουν αλλά δεν περιορίζονται στους αριθμούς. Οι γλώσσες της τεχνητής νοημοσύνης (LIPS και PROLOG είναι οι διασημότερες) επιτρέπουν σε μας να επεξεργαστούμε πιο αποτελεσματικά, χρησιμοποιώντας τον υπολογισμό κατηγορήματος. Αυτά τα συστήματα βασισμένα στη λογική, εμπιστευόμενα στην αφαίρεση, είναι πολύ διαφορετικά από τον παραδοσιακό προγραμματισμό. Δεν επιλύουμε πλέον τις βαθμιαίες διαδικασίες. Άντ' αυτού συλλέγουμε τα γεγονότα και τις σχετικές εμπειροτεχνικές μεθόδους και παράγουμε έπειτα τα συμπεράσματα. Τα συστήματα στα οποία η γνώση ή οι εμπειροτεχνικές μέθοδοι που κωδικοποιούνται προέρχονται από ανθρώπους, είναι τα έμπειρα

συστήματα. Η έρευνα τεχνητής νοημοσύνης εστιάζει στο ταίριασμα σχεδίων. Κατανοούμε τον κόσμο μας με την αναγνώριση των σχεδίων ή/ και των σχέσεων, τα οποία δίνουν το νόημα στα αντικείμενα και τα γεγονότα που αντιμετωπίζουμε.

Η πιο υποσχόμενη τεχνολογία τεχνητής νοημοσύνης είναι τα έμπειρα συστήματα. Τα έμπειρα συστήματα αποτελούν το γνωστότερο πεδίο εφαρμογής της τεχνητής νοημοσύνης. Είναι προγράμματα που συνδυάζουν τη γνώση των ειδικών και προσπαθούν να επιλύσουν πολύπλοκα προβλήματα του πραγματικού κόσμου, κωδικοποιούν τη γνώση και τη συλλογιστική ενός έμπειρου ειδικού σε έναν εξειδικευμένο τομέα της επιστήμης ή της τεχνολογίας. Το κύριο χαρακτηριστικό των έμπειρων συστημάτων σήμερα είναι ότι περιλαμβάνουν τη διαμόρφωση των σκεπτόμενων διαδικασιών των ανθρώπινων εμπειρογνωμόνων που εξοικειώνονται πολύ με τη δεδομένη περιοχή προβλήματος.

Ένα έμπειρο σύστημα πρέπει να διαθέτει και να χειρίζεται μια σημαντική μάζα ειδικής γνώσης. Η γνώση μεταφέρεται από ανθρώπους που είναι ειδικοί σε ένα τομέα και μεταφυτεύεται στο σύστημα με τους κατάλληλους τρόπους. Στη μεταφορά της γνώσης συμμετέχουν οι ειδικοί στο συγκεκριμένο τομέα, οι οποίοι διαθέτουν τη γνώση τους, και ένας εξειδικευμένος επιστήμονας, ο οποίος αποσπά την γνώση από τους ειδικούς τη μεταφέρει κατάλληλα στο σύστημα και ονομάζεται «μηχανικός της γνώσης» (knowledge engineer). Ο μηχανικός της γνώσης πρέπει να έχει πολύ καλό υπόβαθρο στην πληροφορική και ιδιαίτερα στην Τεχνητή Νοημοσύνη.

ΚΕΦΑΛΑΙΟ 1

ΤΕΧΝΙΤΗ ΝΟΗΜΟΣΥΝΗ

1.1 ΦΑΣΕΙΣ ΤΕΧΝΙΤΗΣ ΝΟΗΜΟΣΥΝΗΣ

Από πολύ παλιά (Babbage, 1884) είχαν αρχίσει οι προσπάθειες για τη δημιουργία μηχανών που πίστευαν ότι θα μπορούσε να διέθεται κάποια ευφυΐα. Στα 1950 ο Shannon υποστήριξε ότι θα μπορεί κάποιος να παίζει σκάκι με τη βοήθεια υπολογιστή. Ο Wiener, ο θεμελιωτής της κυβερνητικής, συνεισέφερε και αυτός αναγνωρίζοντας τις ομοιότητες των λειτουργιών μεταξύ ανθρώπων και μηχανών.

Ο όρος "Τεχνητή Νοημοσύνη" (Artificial Intelligence) χρησιμοποιήθηκε για πρώτη φορά από τον John McCarthy το 1956 κατά τη διάρκεια ενός συνεδρίου που διεξήχθη στο κολέγιο του Dartmouth. Οι διάφορες φάσεις της εξέλιξης της τεχνητής νοημοσύνης :

1960: Προϊστορική Φάση

1960-1965: Φάση Χαραυγής

1965- 1970: Μεσαιωνική Φάση

1970-1975: Φάση Αναγέννησης

1975-1980: Φάση Συνεργασίας

1980-1985: Φάση Έρευνας

1985-1990: Φάση Εμπορευματοποίησης

1990-...: Νευρωτική Φάση

Το συνέδριο και τα όσα συζητήθηκαν σε αυτό, θεωρούνται η αρχή της επανάστασης στο χώρο της τεχνητής νοημοσύνης. Μεταξύ αυτών που συμμετείχαν σε αυτό το συνέδριο ήταν οι Minsky, Simon, Newell που θεωρούνται μαζί με το McCarthy σαν οι πρωτοπόροι σε θέματα T.N. Στο συνέδριο αυτό ο μεν McCarthy παρουσίασε τη γλώσσα List Processing Programming Language (Lisp) που είναι ίσως το γνωστότερο εργαλείο με το οποίο μπορεί να γίνεται χειρισμός εννοιών (κανόνων, εντολών, ονομάτων), οι δε Newell και Simon παρουσίασαν ένα πρόγραμμα T.N, το Logic Theoriest, που αποδεικνυε μαθηματικά θεωρήματα.

Ο αρχικός ενθουσιασμός δημιούργησε υπερβολικές προσδοκίες για εύκολη και γρήγορη κατασκευή νοημόνων υπολογιστών. Έτσι άρχισε η φάση της χαραυγής (dawn age) με κύριο χαρακτηριστικό ότι πολλοί ήταν αυτοί που πίστευαν ότι σε δέκα χρόνια οι υπολογιστές θα ήταν το ίδιο έξυπνοι με τους ανθρώπους. Τα αποτελέσματα δεν ήταν τα αναμενόμενα και έτσι προκλήθηκαν απογοητεύσεις. Παρόλο αυτά υπήρξαν μερικά βήματα όπως η κατασκευή δύο προγραμμάτων, οι ιδέες των οποίων χρησιμοποιούνται σήμερα στα έμπειρα συστήματα, ενός για την επίλυση της γεωμετρικής αναλογίας (geometric analogy) και ενός που έκανε συμβολική ολοκλήρωση (symbolic integration).

Μια φωτεινή εποχή ακολουθείται κατά κανόνα από μια σκοτεινή και αντιστροφή. Έτσι κατά τη φάση του μεσαιώνα (dark age) δεν έγινε καμιά ουσιαστική πρόοδος μια και όλοι αναζητούσαν αυτό το κάτι που θα επίλυε όλα τα προβλήματα.

Κάθε μεσαιώνα ακολουθεί η αναγέννηση (renaissance). Τα επόμενα χρόνια (1970-75), κατασκευάστηκαν συστήματα που άρχισαν

να προκαλούν το ενδιαφέρον. Βασικοί εκπρόσωποι αυτής της περιόδου μπορούν να θεωρηθούν τα επιτυχημένα συστήματα: PROSPECTOR (Duda 1979), XCON (Kraft 1984) και MYCIN (Shortliffe, 1984, Buchanan and Shortliffe 1985).

Από το 1975 έως το 1980, οι επιστήμονες της Τ.Ν άρχισαν να συνεργάζονται (age of partnerships) και με επιστήμονες άλλων κλάδων (Cercone and McCalba 1984, Tuhill 1990). Το γεγονός αυτό οδήγησε αρχικά στο να αυξηθεί ο αριθμός των ερευνητών, στη συνέχεια να βγει από τα πανεπιστημιακά εργαστήρια και να μπει στα ερευνητικά εργαστήρια των επιχειρήσεων και σταδιακά να αρχίσει η ανάπτυξη εφαρμογών που αφορούσαν στην επίλυση πραγματικών προβλημάτων. Τα επόμενα χρόνια άρχισαν να αναπτύσσονται μεγάλα ερευνητικά προγράμματα. Κορυφαίο γεγονός αποτελεί το δεκαετές πρόγραμμα δημιουργίας πέμπτης γενιάς υπολογιστών, που έγινε το 1981 στο Τόκιο από την Ιαπωνική κυβέρνηση και στην ανταπόκριση που βρήκε αυτή η πρόταση τόσο από τις κυβερνήσεις των Η.Π.Α και των Ευρωπαϊκών χωρών όσο και από τις επιχειρήσεις. Το πρόγραμμα ESPRIT της Ε.Ο.Κ είναι ένα παρόμοιο πρόγραμμα αλλά περιέχει και άλλα πεδία.

Σαν επόμενη φάση δεν μπορούσε να είναι άλλη από αυτή της εμπορευματοποίησης (commercialization), όπου μεγάλος αριθμός εταιριών και ερευνητών ασχολούνται με την ανάπτυξη συστημάτων τεχνητής νοημοσύνης που επιλύουν πραγματικά προβλήματα. Τέτοιες περιοχές είναι τα έμπειρα συστήματα και τα συστήματα κατανόησης φυσικής γλώσσας.

Η τρέχουσα φάση χαρακτηρίζεται από τις έρευνες που γίνονται για τη ανάπτυξη υπολογιστών έκκτης γενιάς και αναφέρονται στη

συμπεριφορά του εγκεφάλου (νευρωνική νοημοσύνη) και τη μοριακή νοημοσύνη (Winston and Shellard, 1990). Τα πρώτα πρακτικά αποτελέσματα λαμβάνονται από την ανάπτυξη νευρωνικών δικτύων (neural-networks).

1.2 ΝΟΗΜΟΝΑ ΣΥΣΤΗΜΑΤΑ

Οι διαφορετικές οπτικές γωνίες αντιμετώπισης της τεχνητής νοημοσύνης έχουν σαν αποτέλεσμα την ύπαρξη πολλών διαφορετικών ορισμών, οι σημαντικότεροι είναι από τους πιο κάτω:

Ο Minsky θεωρεί τη T.N σαν την επιστήμη που επιτρέπει στις μηχανές να κάνουν πράγματα που αν γίνονταν από ανθρώπους θα απαιτούσαν, νοημοσύνη. Ο Barr and Feigenbaum εκτιμούν ότι η T.N είναι εκείνο το πεδίο της επιστήμης των ηλεκτρονικών υπολογιστών που σχετίζεται με τη σχεδίαση ευφυών υπολογιστικών συστημάτων, τα οποία είναι υπολογιστικά συστήματα που επιδεικνύουν χαρακτηριστικά νοήμονος ανθρώπινης συμπεριφοράς.

Το 1983 η Rich έδωσε τον ορισμό: Τεχνητή νοημοσύνη είναι η μελέτη του πως να κάνουμε τους υπολογιστές να κάνουν πράγματα, τα οποία προς το παρόν οι άνθρωποι τα κάνουν καλύτερα .

Οι Charniak και McDermott, θεωρούν την τεχνητή νοημοσύνη σαν την επιστήμη που ασχολείται με την μελέτη των πνευματικών ικανοτήτων μέσω της χρήσης υπολογιστικών προτύπων.

Τον επόμενο χρόνο ο Bonnet εκτιμούσε ότι μέσω της τεχνητής νοημοσύνης γίνεται προσπάθεια να μελετηθεί η δομή των

προγραμμάτων των ηλεκτρονικών υπολογιστών που αναπαριστούν την ανθρώπινη συμπεριφορά ώστε να κατανοηθεί η φύση της ανθρώπινης νοημοσύνης.

Ο Tanimoto, πιστεύει ότι η τεχνητή νοημοσύνη μελετά τις υπολογιστικές τεχνικές εκτέλεσης διαφόρων εργασιών οι οποίες όταν γίνονται από ανθρώπους απαιτούν τη χρήση νοημοσύνης.

Ο κοινός παρονομαστής όλων αυτών των απόψεων είναι αφ'ενός μεν να γίνουν κατανοητές οι διαδικασίες της ανθρώπινης σκέψης, αφ'ετέρου δε να καταστεί δυνατή η μοντελοποίησή και ο προγραμματισμός τους σε ηλεκτρονικό υπολογιστή, έτσι ώστε ο υπολογιστής να μπορεί να επιλύει τα ίδια προβλήματα με τον ίδιο τρόπο που μπορεί να τα επιλύει και ένας άνθρωπος.

Πεδία εφαρμογών

Η αλληλεπίδραση της Τ.Ν με άλλες επιστήμες όπως ψυχολογία, φιλοσοφία, γλωσσολογία, μηχανική, υλικό (hardware) και λογισμικό (software), λήψη αποφάσεων, στατιστική, επιχειρησιακή έρευνα κ.λ.π οδήγησε στην ανάπτυξη πολλών επιμέρους επιστημονικών πεδίων έρευνας. Μεταξύ αυτών των πεδίων υπάρχουν επικαλύψεις και αλληλεπιδράσεις γεγονός που καθιστά πολύ δύσκολη τη ταξινόμηση τους. Η λύση του προβλήματος είναι να γίνει η ταξινόμηση με βάση τα αποτελέσματα των εφαρμογών τους. Έτσι σαν κύριες περιοχές θεωρούνται τα: έμπειρα συστήματα (expert systems) ή συστήματα βασισμένα στη γνώση (knowledge based systems), επεξεργασία φυσικής γλώσσας (natural language processing), κατανόηση ομιλίας (speech

understanding), έξυπνα συστήματα διδασκαλίας (intelligent computer-based instruction), κ.λ.π.

Επεξεργασία φυσικής γλώσσας (natural language processing)

Η επικοινωνία του χρήστη με τον υπολογιστή γίνεται με διαλογικό τρόπο και με τη βοήθεια της τεχνολογίας επεξεργασίας της φυσικής γλώσσας, στην καθομιλουμένη γλώσσα του χρήστη. Το πρόβλημα, ακόμα και αν το περιορίσουμε μόνο στη κατανόηση της γραπτής γλώσσας, παραμένει εξαιρετικά περίπλοκο. Η δυσκολία γίνεται αντιληπτή αν αναλογισθεί κανείς ότι θα πρέπει να γνωρίζει πολύ καλά όχι μόνο τη συγκεκριμένη γλώσσα, αλλά και τις συντακτικές και γραμματικές της ιδιαιτερότητες.

Η τεχνολογία επεξεργασίας της φυσικής γλώσσας αποτελείται από τα ακόλουθα, υπομμήματα:

Κατανόηση φυσικής γλώσσας: Το υπομμήμα αυτό ασχολείται με τη διερεύνηση εκείνων των μεθόδων που θα επιτρέπουν στα συστήματα (λογισμικού και υλικού) να κατανοούν τις εντολές που δίνει ο χρήστης στο τρέχοντα γραπτό λόγο.

Δημιουργία φυσικής γλώσσας: Εδώ γίνεται προσπάθεια προς την αντίθετη κατεύθυνση, δηλαδή τα συστήματα (λογισμικού και υλικού) να έχουν τη δυνατότητα να συνδιαλέγονται με το χρήστη στο τρέχοντα γραπτό λόγο. Μερικές από τις σημαντικότερες περιοχές εφαρμογής των προγραμμάτων που έχουν αναπτυχθεί μέχρι σήμερα είναι τα συστήματα: επικοινωνίας επεξεργασίας φυσικής γλώσσας, γραμματικής ανάλυσης, μετάφρασης από μια γλώσσα σε μια άλλη, κατανόησης ομιλίας κ.λ.π (Andriole, 1985; Harris, 1984; Frenzel, 1987).

Κατανόηση ομιλίας (speech understanding)

Από την άλλη μεριά η κατανόηση της ομιλίας ασχολείται με τη προσπάθεια να γίνονται κατανοητές, από τα συστήματα (λογισμικού και υλικού), οι προτάσεις του χρήστη που διατυπώνονται στη καθομιλουμένη γλώσσα. Η διαδικασία αυτή αποτελείται από δύο φάσεις:

Αναγνώριση ομιλίας (speech recognition): Είναι η φάση κατά την οποία ο χρήστης ομιλεί προς το σύστημα και αυτό αναγνωρίζει απλώς τις λέξεις χωρίς να κατανοεί το μήνυμά τους.

Κατανόηση ομιλίας (speech understanding): Στη φάση αυτή γίνονται κατανοητά, από τα συστήματα (λογισμικού και υλικού), τα μηνύματα των προτάσεων που έχουν προηγουμένως αναγνωρισθεί.

Έξυπνα συστήματα διδασκαλίας (intelligent tutoring systems)

Για πολλά χρόνια γίνονταν προσπάθειες ένταξης των υπολογιστικών συστημάτων στην εκπαιδευτική διαδικασία, ώστε να δημιουργηθούν προγράμματα που θα διδάσκουν κάποιο συγκεκριμένο αντικείμενο, έχοντας τις ίδιες ικανότητες με έναν εκπαιδευτή (Skinner, 1950). Η τεχνολογία της Τ.Ν έδωσε ώθηση την ανάπτυξη έξυπνων συστημάτων διδασκαλίας. Τα συστήματα αυτά διαθέτουν την απαραίτητη γνώση για να ανταποκρίνονται τόσο στις γνωστικές όσο και στις εκπαιδευτικές απαιτήσεις της διδασκαλίας όπως θα ανταποκρινόταν και ένας κανονικός εκπαιδευτικός. Ορίζεται λοιπόν: Σαν έξυπνο σύστημα διδασκαλίας μπορεί να θεωρηθεί ένα ευφυές σύστημα, που επιχειρεί να μιμηθεί τις ικανότητες ενός ανθρώπου-καθηγητή, που συνεργάζεται σε προσωπικό επίπεδο με το μαθητή, εντοπίζοντας ότι γνωρίζει ο μαθητής

και κατανοώντας τη λογική του σκέψης, καθώς και τις αδυναμίες που υπάρχουν στη χρησιμοποίηση της γνώσης του. Το σύστημα χρησιμοποιεί τη γνώση του μαθητή για να καθορίσει το πως μπορεί να τον κρίνει σωστότερα (Sleeman and Brown 1982, O'Shea and Self 1984, Δουκίδης και Αγγελίδης, 1992).

Μερικά από τα συστήματα αυτά είναι: ALGEBRA, BUGGY, GUIDON, SCHOLAR κ.α.(Sleeman and Brown 1982, Andriole 1984, Expert Systems, Special issue, vol. 5, no. 4, 1988). Τα συστήματα αυτά σήμερα χρησιμοποιούνται σε διάφορες εργασίες όπως στη προσομοίωση, παίγνια, ελέγχους κ.λ.π. (Bork, 1981).

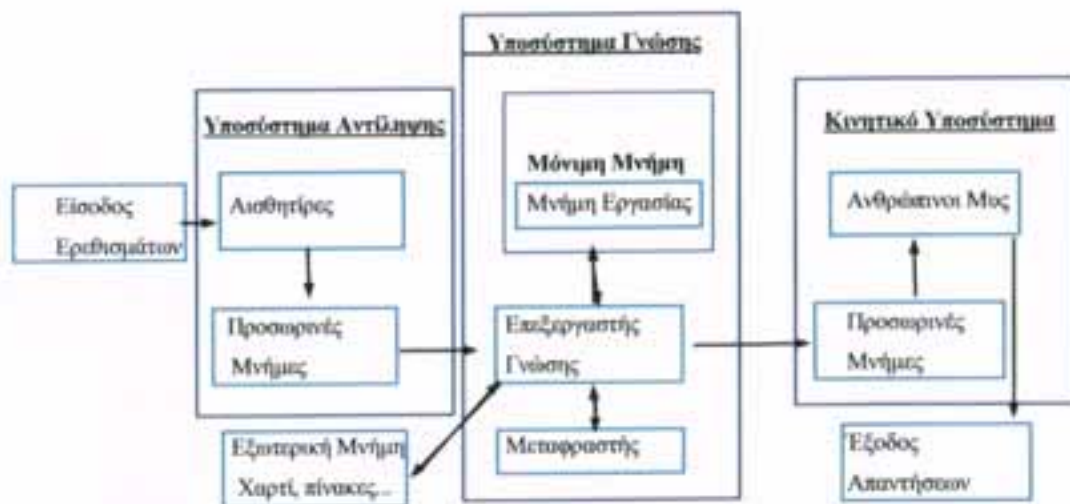
1.3 Η ΤΕΧΝΗΤΗ ΕΝΑΝΤΙ ΤΗΣ ΑΝΘΡΩΠΙΝΗΣ ΝΟΗΜΟΣΥΝΗΣ

Για να κατανοηθεί καλύτερα η τεχνητή νοημοσύνη θα πρέπει να συγκριθεί με την ανθρώπινη ώστε να επισημανθούν τα σημεία στα οποία υπερτερεί η μία της άλλης (Καρίαν, 1984). Στη συνέχεια δίνονται τα αποτελέσματα μερικών τέτοιων συγκρίσεων.

- Σε αντίθεση με τον άνθρωπο που έχει μεταβαλλόμενη νοημοσύνη, είτε γιατί αλλάζει άποψη για κάποιο θέμα είτε γιατί ξεχνά, η γνώση της Τ.Ν παραμένει αναλλοίωτη.
- Η διαδικασία απόσπασης και αποθήκευσης της γνώσης σε ένα σύστημα Τ.Ν γίνεται μόνο μία φορά, ενώ ακολούθως μπορεί να αναπαραχθεί και να εγκατασταθεί σε όσους υπολογιστές χρειάζεται. Αντιθέτως η ανθρώπινη γνώση απαιτεί μακροχρόνια εκπαίδευση κάθε φορά που χρειάζεται να μεταφερθεί από ένα άτομο σε ένα άλλο.

- Η χρήση τεχνικών T.N κοστίζει τις περισσότερες φορές λιγότερο από το κόστος χρησιμοποίησης ανθρώπινου δυναμικού για τη διεξαγωγή των ίδιων εργασιών.
- Η ανθρώπινη συμπεριφορά και νοημοσύνη είναι ασταθής και επηρεάζεται από πολλούς παράγοντες, σε αντίθεση με τη T.N η οποία είναι σταθερή και ανεξάρτητη από εξωτερικές επιδράσεις.
- Η T.N μπορεί ανά πάσα στιγμή να τεκμηριώσει το τρόπο "σκέψης" της. Μπορεί δηλαδή να παραθέσει τα διαδοχικά βήματα που ακολούθησε για να καταλήξει στο συγκεκριμένο συμπέρασμα σε αντίθεση με τον άνθρωπο που πολλές φορές δεν μπορεί να αναπαράγει τα διαδοχικά βήματα που ακολούθησε για να φτάσει σε κάποιο συμπέρασμα.
- Ο άνθρωπος έχει τη δυνατότητα να αποκτά και να δημιουργεί γνώση, ενώ αντιθέτως η γνώση στα συστήματα T.N έχει αυστηρώς καθορισμένη δομή η οποία είναι δύσκολο να μετατραπεί.
- Η ανθρώπινη λογική μπορεί να χρησιμοποιεί ένα πολύ ευρύτερο πεδίο γνώσης από αυτό του συγκεκριμένου θέματος απόφασης. Η T.N, αντιθέτως είναι επικεντρωμένη μόνο σε εξειδικευμένα θέματα, διαθέτοντας γνώση μόνο για την επίλυση συγκεκριμένων προβλημάτων.

Με στόχο να κατανοηθεί καλύτερα ο τρόπος εργασίας της T.N καθώς και οι περιορισμοί της, προτάθηκε από τους Newell και Simon (1972) ένα μοντέλο ανθρώπινης επεξεργασίας των πληροφοριών (σχήμα 1.2).



Σχήμα 1.2 Ανθρώπινη επεξεργασία πληροφορίας κατά Newell-Simon

Ένας εξωτερικός ερεθισμός ανιχνεύεται μέσω ανθρώπινων αισθητήρων (μάτια, αυτιά), οι οποίοι αποθηκεύουν προσωρινά τη πληροφορία αυτή στις προσωρινές μνήμες, μέχρι ότου κληθεί από το γνωστικό υποσύστημα όταν δημιουργηθεί η ανάγκη χρησιμοποίησής της για τη λήψη μιας απόφασης. Η πληροφορία αυτή μεταφέρεται στη μνήμη εργασίας. Αν απαιτείται περισσότερη πληροφορία τότε ο επεξεργαστής γνώσης απευθύνεται, για την απόκτησή της, στη μόνιμη μνήμη. Ο μεταφραστής, μεταφράζει τις εντολές για την επίλυση του συγκεκριμένου προβλήματος και τη λήψη της ζητούμενης απόφασης. Η μόνιμη μνήμη αποτελείται από "κομμάτια" (chunks) που περιλαμβάνουν ομάδες συμβόλων καθώς και τις σχέσεις που επικρατούν μεταξύ των. Ένα "κομμάτι" είναι μια μονάδα αποθηκευμένης πληροφορίας και μπορεί να αποτελείται από οργανωμένα μικρότερα «κομμάτια». Η εξωτερική μνήμη, υποστηρίζει τη διαδικασία λήψης απόφασης από τον άνθρωπο, και αποτελείται από εξωτερικά μέσα όπως χαρτιά, πίνακες κ.λ.π. Η

μόνιμη μνήμη έχει απεριόριστες ικανότητες. Στη μνήμη εργασίας μπορούν να κρατούνται προσωρινά μόνο πέντε έως επτά "κομμάτια" ενώ όταν εκτελείται μια άλλη εργασία μπορούν να παραμένουν μέχρι δύο. Αυτό σημαίνει ότι η χρήση της μνήμης εργασίας είναι για επεξεργασίες εισόδου -εξόδου. Σύμφωνα με αυτό το μοντέλο ο άνθρωπος μπορεί να επεξεργάζεται μία πληροφορία κάθε χρονική περίοδο. Μετά την επεξεργασία ο επεξεργαστής στέλνει εντολές προς το μυϊκό υποσύστημα το οποίο προκαλεί τις απαραίτητες δραστηριότητες όπως κίνηση, ομιλία κ.λ.π.

Αντίθετα με τους περιορισμούς που υπάρχουν στον άνθρωπο, ο υπολογιστής μπορεί να:

- Επεξεργάζεται, αποθηκεύει και ανακτά γρηγορότερα τα δεδομένα που χειρίζεται.
- Έχει περισσότερες δυνατότητες χειρισμού δεδομένων που λειτουργούν με πιθανότητες.
- Διαθέτει απεριόριστο αριθμό προσωρινών μνημών.
- Μπορεί να λειτουργεί είτε με σειριακή είτε με παράλληλη επεξεργασία.

ΚΕΦΑΛΑΙΟ 2

ΕΜΠΕΙΡΑ ΣΥΣΤΗΜΑΤΑ

2.1 ΑΝΑΤΟΜΙΑ ΕΝΟΣ ΕΜΠΕΙΡΟΥ ΣΥΣΤΗΜΑΤΟΣ

Οι στόχοι τεχνητής νοημοσύνης είναι να αναπαραχθεί η ανθρώπινη επεξεργασία πληροφοριών με μια μηχανή. Μεταξύ των σημαντικών προκλήσεων που ωθούν, η τρέχουσα έρευνα είναι: όραμα, επεξεργασία φυσικής γλώσσας, μηχανές που μπορούν να μάθουν από τα λάθη τους, ευφυή συστήματα παράδοσης ιδιαίτερων μαθημάτων που μπορούν να προσαρμοστούν στο ύψος εκμάθησης του χρήστη και να προσδιορίσουν τους τύπους της παραγωγής των λαθών, και νευρικά δίκτυα που μιμούνται τα σχέδια επεξεργασίας της ανθρώπινης ροής εγκεφάλου. Πολύ εμπορικότερα επιτυχής γι'αυτό είναι τα έμπειρα συστήματα. Τα έμπειρα συστήματα στοχεύουν να κωδικοποιήσουν την γνώση των ανθρώπινων εμπειρογνομώνων στις συγκεκριμένες περιοχές προβλήματος, καθιστώντας κατά συνέπεια εκείνη την γνώση διαθέσιμη για να την χρησιμοποιήσουν άλλοι.

Η τεχνολογία έμπειρων συστημάτων έχει ενηλικιωθεί σαφώς. Ένα έμπειρο σύστημα είναι ένα πρόγραμμα υπολογιστών που αντιπροσωπεύει την λογική με τη γνώση κάποιου ειδικού υπαγόμενου, με σκοπό την επίλυση των προβλημάτων ή την παροχή των συμβουλών.

Η ανάπτυξή τους άρχισε από τα τέλη της δεκαετίας του 1960. Αυτή τη περίοδο οι προσπάθειες ήταν στραμμένες προς την επίλυση προβλημάτων γενικού σκοπού (generalpurpose problem solver) που είχε αναπτυχθεί από τους Newell και Simon (1972) στη προσπάθειά

τους για δημιουργία ενός "νοήμονος υπολογιστή". Σε ένα πρωτοπόρο άρθρο τους, οι Newell and Simon (1958), ανέπτυξαν την ιδέα ότι ένα σύστημα, που θα χρησιμοποιεί κανόνες παραγωγής (production rules), θα μπορεί να μοντελοποιήσει μερικές μορφές ανθρώπινης διαδικασίας επίλυσης προβλημάτων. Το πέρασμα από τα προγράμματα γενικού-σκοπού στα προγράμματα ειδικού-σκοπού έγινε στα μέσα της δεκαετίας του 1960 με την ανάπτυξη του DENDRAL από τον Feigenbaum στο Πανεπιστήμιο του Stanford. Η παρατήρηση, ότι η γνώση, που απαιτείται για την διεξαγωγή ειδικών εργασιών, ήταν υψηλότερου επιπέδου από την γνώση της κοινής λογικής, που χρησιμοποιείτο μέχρι τότε στην ανάπτυξη συστημάτων T.N., οδήγησε στη ραγδαία ανάπτυξη των Ε.Σ. (Goodman 1991). Από τα μέσα του '70 άρχισε η συνεχής ανάπτυξη τέτοιων συστημάτων, τα οποία απαιτούσαν την επεξεργασία μεγάλης μάζας εξειδικευμένης γνώσης. Η ισχύ ενός Ε.Σ. εξαρτάται κυρίως από τη γνώση που διαθέτει και όχι από τους χρησιμοποιούμενους φορμαλισμούς αναπαράστασης της γνώσης ή τους μηχανισμούς εξαγωγής συμπερασμάτων.

Από τις αρχές του '80 οι τεχνολογίες ανάπτυξης Ε.Σ., που περιορίζονταν σε πανεπιστημιακούς χώρους άρχισαν να εφαρμόζονται για την ανάπτυξη εμπορικών εφαρμογών (R-1, XCON, XSEL, CATS-1 κ.α.). Ταυτόχρονα άρχισαν οι προσπάθειες ανάπτυξης ειδικών εργαλείων, που θα συνέβαλαν στην ταχύτερη ανάπτυξη Ε.Σ. (EMYCIN, AGE, KAS, EURISKO, M.1, EXSYS κ.α.) και τα οποία άρχισαν να εμφανίζονται στην αγορά από το 1983.

Στην ραγδαία ανάπτυξη των Ε.Σ. οδήγησε η όλο και αυξανόμενη ανάγκη για όσο το δυνατόν πιο εξειδικευμένες συμβουλές, σε όλα τα επίπεδα η έλλειψη και το υψηλό κόστος εξειδικευμένου προσωπικού

ώθησε τις επιχειρήσεις να επενδύσουν τεράστια ποσά για την ανάπτυξη εμπειρών συστημάτων.

2.2 ΟΡΙΣΜΟΙ ΕΜΠΕΙΡΩΝ ΣΥΣΤΗΜΑΤΩΝ

Όπως και στη T.N. έτσι και εδώ υπάρχει πληθώρα ορισμών, που εξαρτώνται από την γωνία, που ο κάθε ένας προσεγγίζει τα Ε.Σ. Γενικά οι ορισμοί μπορούν να χωρισθούν σε αυτούς που στηρίζονται στο τι κάνει ένα Ε.Σ. και σε αυτούς, που ασχολούνται με το πώς το κάνει. Ακολούθως παρατίθενται μερικοί από τους ορισμούς :

Ο Feigenbaum (1982) θεωρεί ότι ένα έμπειρο σύστημα είναι ένα έξυπνο πρόγραμμα το οποίο για να λύσει κάποιο πρόβλημα, που είναι αρκετά δύσκολο ώστε να απαιτεί σημαντική εμπειρία για την επίλυσή του, χρησιμοποιεί ειδική γνώση και διαδικασίες εξαγωγής συμπερασμάτων. Η γνώση που είναι απαραίτητη για τη λειτουργία σε αυτό το επίπεδο, μαζί με τις διαδικασίες εξαγωγής συμπερασμάτων, μπορεί να θεωρηθούν σαν ένα μοντέλο εμπειρίας των καλύτερων ειδικών του χώρου.

Ο Hayes-Roth (1984), αναφέρει ότι τα βασιζόμενα στη γνώση έμπειρα συστήματα ή όπως για συντομία τα αποκαλεί, τα συστήματα γνώσης χρησιμοποιούν ανθρώπινη γνώση για να επιλύσουν προβλήματα, που κανονικά για να επιλυθούν χρειάζονται ανθρώπινη νοημοσύνη.

Έμπειρο σύστημα είναι εκείνο που χειρίζεται πραγματικά σύνθετα προβλήματα, που απαιτούν την παρουσία ειδικού, και τα επιλύει χρησιμοποιώντας ένα ειδικό υπολογιστικό μοντέλο λογικής,

καταλήγοντας στα ίδια συμπεράσματα που ένας ειδικός θα έφτανε αν αντιμετώπιζε ένα ανάλογο πρόβλημα. Ένα Ε.Σ. προσπαθεί να συλλάβει αρκετή από τη γνώση ενός ειδικού έτσι, ώστε να επιλύει με έμπειρο τρόπο τα προβλήματα (Weiss and Kulikowski 1984).

Ο Sell δίνει τον ορισμό, ότι ένα Ε.Σ. είναι ένα σύστημα βασισμένο στη γνώση που προσομοιάζει τη σκέψη του ειδικού προκειμένου να επιλύσει σημαντικά προβλήματα σε ένα ξεχωριστό πεδίο.

Ο Hart (1986) εκτιμά ότι τα Ε.Σ. είναι προγράμματα, τα οποία εκτελούν εργασίες τις οποίες συνήθως τις κάνουν ειδικοί. Αυτά ενσωματώνουν τη γνώση ειδικών και την ικανότητά τους να χρησιμοποιούν αυτή τη γνώση για να επιλύουν προβλήματα. Τα προγράμματα αυτά περιορίζονται από τον τύπο των εργασιών που μπορούν να εκτελέσουν, αλλά παρουσιάζουν την εξειδίκευσή τους κατά την προσέγγιση κατάλληλων προβλημάτων. Σε τέτοια προγράμματα η γνώση κωδικοποιείται δημιουργώντας έτσι ισχυρά εργαλεία.

Ο Kumara (1986) θεωρεί ότι ένα Ε.Σ. είναι ένα εργαλείο που έχει την ικανότητα να κατανοεί την ιδιαίτερη γνώση ενός προβλήματος και χρησιμοποιώντας ευφυώς τη γνώση, του πεδίου αυτού, να προτείνει εναλλακτικές ενέργειες.

Έμπειρα Συστήματα είναι προγράμματα με τα οποία γίνεται προσπάθεια να αναπαρασταθεί η συμπεριφορά ενός ειδικού κάποιου τομέα. Τα συστήματα αυτά διαθέτουν γνώση και εμπειρία και είναι ικανά με τη χρήση λογικής να παρέχουν συμβουλές ή να παίρνουν αποφάσεις έχοντας την ικανότητα να αιτιολογούν το πώς οδηγούνται σε αυτές.

2.3 ΛΕΙΤΟΥΡΓΙΑ ΚΑΙ ΣΤΟΧΟΙ ΕΜΠΕΙΡΩΝ ΣΥΣΤΗΜΑΤΩΝ

Ένα έμπειρο σύστημα μπορεί να εκπληρώνει την ανθρώπινη εμπειρία που απαιτείται, ή μπορεί να διαδραματίζει το ρόλο ενός βοηθού σε έναν ανθρώπινο υπεύθυνο. Με άλλα λόγια, ο πελάτης μπορεί να αλληλεπιδράσει με το πρόγραμμα άμεσα, ή να αλληλεπιδράσει με έναν ανθρώπινο εμπειρογνώμονα που αλληλεπιδρά με το πρόβλημα. Ο ευθυνόν μπορεί να είναι εμπειρογνώμονας στο δικαίωμά του, οπότε σ'αυτή την περίπτωση το πρόγραμμα μπορεί να δικαιολογήσει την ύπαρξή του με τη βελτίωση της παραγωγικότητάς του. Εναλλακτικά, ο ανθρώπινος συνεργάτης μπορεί να είναι κάποιος που να είναι σε θέση να παρέχει κάποια τεχνική βοήθεια στα ειδικά επίπεδα απόδοσης του προγράμματος. Μπορεί να πάρει τη σωστή κατανομή της λειτουργίας μεταξύ του προσώπου και της μηχανής που είναι συχνά το κλειδί στην επιτυχή ειδική επέκταση.

Οι χαρακτηριστικοί στόχοι για τα έμπειρα συστήματα περιλαμβάνουν:

- ερμηνεία των στοιχείων (όπως τα σήματα sonar),
- διάγνωση των δυσλειτουργιών (όπως τα ελαττώματα εξοπλισμού ή οι ανθρώπινες ασθένειες),
- δομική ανάλυση των σύνθετων αντικειμένων (όπως οι χημικές ενώσεις),
- διαμόρφωση των σύνθετων αντικειμένων (όπως τα συγκροτήματα ηλεκτρονικών υπολογιστών), και
- ακολουθίες προγραμματισμού ενεργειών (όπως να εκτελεσθεί από τα ρομπότ).

2.4 ΤΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΕΝΟΣ ΕΜΠΕΙΡΟΥ ΣΥΣΤΗΜΑΤΟΣ

Επιδίωξη των Έμπειρων Συστημάτων αποτελεί η μοντελοποίηση της ανθρώπινης σκέψης, με την παράλληλη εκμετάλλευση των δυνατοτήτων που προσφέρουν οι ηλεκτρονικοί υπολογιστές. Όπως η ταχύτητα και ακρίβεια υπολογισμών και η μεγάλη χωρητικότητα μνήμης είναι απαραίτητα για τη φύλαξη και το χειρισμό μεγάλων ποσοτήτων εξειδικευμένης γνώσης, η οποία άλλωστε αποτελεί την καρδιά του Έμπειρου Συστήματος.

Στα πραγματικά προβλήματα η γνώση δεν είναι στατιστική, αλλά δυναμική, υποκειμενική σε μεταβολές, προσθήκες, ανακατατάξεις και αναθεωρήσεις. Τα Έμπειρα Συστήματα χαρακτηρίζονται από σημαντική προσαρμοστικότητα και προσφέρουν εξαιρετική ευελιξία, λόγω της ευκολίας που παρέχουν για προσθήκη νέας γνώσης και αναθεώρηση της ήδη υπάρχουσας και της ευχέρειας που προσφέρουν για σταδιακή ανάπτυξη.

Η υφή του πραγματικού κόσμου επιβάλλει την εξαγωγή συμπερασμάτων με χειρισμό ελλιπούς γνώσης, ασαφών και αβέβαιων γεγονότων, καθώς επίσης και μη αντικειμενικά ορισμένων εννοιών. Η τεχνολογία των Ε.Σ. προσφέρει την δυνατότητα έκφρασης τέτοιων χαρακτηριστικών του πραγματικού κόσμου με τη χρήση διαφόρων μεθόδων και τεχνικών, οι οποίες έχουν ήδη αποδώσει ικανοποιητικά.

Ιδιαίτερη έμφαση δίνεται στην ικανότητα του συστήματος να πλησιάζει το χρήστη. Αυτό εκφράζεται με την ανάπτυξη του συστήματος ιδιαίτερα εξελιγμένων και φιλικών συνιστωσών διασύνδεσης με τον

άνθρωπο (human interfaces) με την παροχή ευκολιών, όπως η επεξήγηση και η αιτιολόγηση της πορείας συλλογισμού.

Το Ε.Σ. έχει δύο κύρια στοιχεία, το πρώτο στοιχείο αναφέρεται στην ιδιότητα των έμπειρων συστημάτων να χειρίζονται την γνώση σε συμβολική μορφή δηλαδή να περιέχει όλη τη γνώση του ειδικού προβλήματος (βάση γνώσης) και το δεύτερο στοιχείο αναφέρεται στη δομή τους. Το οποίο ασχολείται με το χειρισμό της γνώσης με σκοπό την εξαγωγή συμπερασμάτων (μηχανισμός εξαγωγής συμπερασμάτων).

Ένα έμπειρο σύστημα μπορεί να διακριθεί από ένα συμβατικότερο πρόγραμμα εφαρμογών σε αυτό:

➤ **Μιμείται τον ανθρώπινο συλλογισμό** για μια περιοχή προβλήματος, παρά τη μίμηση της ίδιας περιοχής. Αυτό διακρίνει τα έμπειρα συστήματα από τα πιο γνωστά προγράμματα που περιλαμβάνουν τη μαθηματική διαμόρφωση ή τη ζωτικότητα υπολογιστών. Αυτό δεν πρόκειται να πει ότι το πρόγραμμα είναι ένα πιστό ψυχολογικό πρότυπο του εμπειρογνώμονα, αλλά μόνο ότι η εστίαση είναι επάνω στη μίμηση των δυνατοτήτων επίλυσης προβλήματος εμπειρογνομώνων.

➤ Εκτελεί το συλλογισμό πέρα από **την ανθρώπινη γνώση αντιπροσωπεύσεων**, εκτός από το να κάνει τους αριθμητικούς υπολογισμούς ή την ανάκτηση στοιχείων. Η γνώση στο πρόγραμμα που εκφράζεται κανονικά σε κάποια ειδική χρήση γλώσσας και που κρατιέται χωριστό από τον κώδικα που εκτελεί το συλλογισμό. Αυτές οι ευδιάκριτες ενότητες προγράμματος αναφέρονται ως **βάση γνώσης** και **μηχανή συμπεράσματος**, αντίστοιχα.

➤ **Λύνει τα προβλήματα με τις ευρετικές ή κατά προσέγγιση μεθόδους** που, αντίθετα από τις αλγοριθμικές λύσεις, δεν είναι εγγυημένες για να πετύχουν. Ένας ευρετικός είναι ουσιαστικά μια εμπειροτεχνική μέθοδος που κωδικοποιεί ένα κομμάτι της γνώσης για το πώς να λύσει τα προβλήματα κάποιας περιοχής. Τέτοιες μέθοδοι είναι κατά προσέγγιση υπό την έννοια ότι (1) δεν απαιτούν τα τέλεια στοιχεία και (2) οι λύσεις που παράγονται από το σύστημα μπορούν να προταθούν με τους ποικίλους βαθμούς βεβαιότητας.

Ένα έμπειρο σύστημα διαφέρει από άλλα είδη προγράμματος τεχνητής νοημοσύνης σε αυτό:

➤ Εξετάζει το περιεχόμενο **της ρεαλιστικής πολυπλοκότητας** που απαιτεί κανονικά ένα μη αμελητέο ποσό ανθρώπινης εμπειρίας. Πολλά προγράμματα T.N. είναι ερευνητικοί φορείς, και μπορούν να στραφούν προς τα μαθηματικά προβλήματα ή τις απλουστευμένες εκδόσεις των πραγματικών προβλημάτων προκειμένου να αποκτηθούν οι επίγνωσεις ή να καθαριστούν οι τεχνικές. Ενώ τα έμπειρα συστήματα, λύνουν τα προβλήματα γνήσιου επιστημονικού ή εμπορικού ενδιαφέροντος.

➤ Πρέπει να εκθέσει **την υψηλή απόδοση** από την άποψη της ταχύτητας και της αξιοπιστίας προκειμένου να είναι χρήσιμο εργαλείο. Οι ερευνητικές μέθοδοι T.N. μπορούν να μην τρέξουν πολύ γρήγορα, και μπορεί να μην περιέχουν ένα καλό λογισμικό πρόγραμμα. Αλλά ένα έμπειρο σύστημα προτείνει τις λύσεις σε έναν λογικό χρόνο και να είναι σωστό τις περισσότερες φορές, δηλαδή τουλάχιστον τόσο συχνά όσο ένας ανθρώπινος εμπειρογνώμονας.

➤ Πρέπει να είναι ικανό **στις λύσεις** ή τις συστάσεις προκειμένου να πειστεί ο χρήστης ότι ο συλλογισμός του είναι στην πραγματικότητα

σωστός. Τα ερευνητικά προγράμματα χαρακτηριστικά μόνο οργανώνονται από τους δημιουργούς τους, ή από άλλο προσωπικό σε παρόμοια εργαστήρια. Ένα έμπειρο σύστημα θα οργανωθεί από ένα ευρύτερο φάσμα των χρηστών, και πρέπει επομένως να σχεδιαστεί κατά τέτοιο τρόπο ώστε τα έργα του να είναι διαφανέστερα.

Το knowledge based system χρησιμοποιείται μερικές φορές ως συνώνυμο για το έμπειρο σύστημα. Ένα βασισμένο στη γνώση σύστημα είναι οποιοδήποτε σύστημα που εκτελεί έναν στόχο με την εφαρμογή των εμπειροτεχνικών μεθόδων σε μια συμβολική αντιπροσώπευση της γνώσης, αντί της υιοθέτησης των συνήθως αλγοριθμικών ή στατιστικών μεθόδων.

2.5 ΣΧΕΔΙΑΣΜΟΣ ΕΜΠΕΙΡΩΝ ΣΥΣΤΗΜΑΤΩΝ

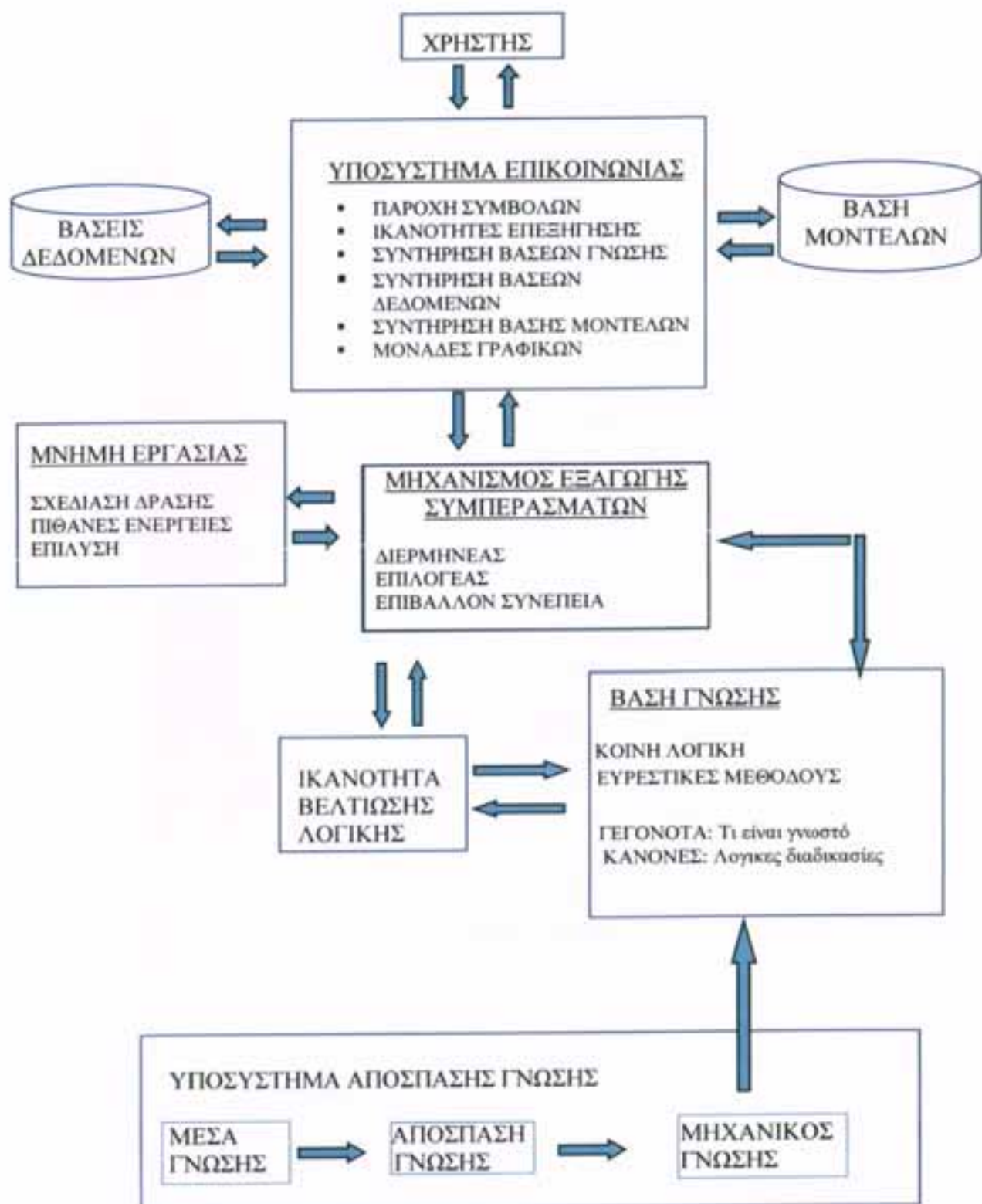
ΔΟΜΗ ΕΜΠΕΙΡΩΝ ΣΥΣΤΗΜΑΤΩΝ

Κάθε έμπειρο σύστημα έχει δύο όψεις. Η μία είναι αυτή που βλέπει ο κατασκευαστής του, κατά τη φάση ανάπτυξης και εισαγωγής της γνώσης σε αυτό, ενώ η άλλη είναι αυτή που αντιλαμβάνεται ο χρήστης του συστήματος κατά τη διάρκεια της λειτουργίας του όταν δέχεται τις συμβουλές του.

Μια τυπική διαδικασία χρήσης ενός έμπειρου συστήματος είναι αρχικά να διατυπωθεί ένα αίτημα, από το χρήστη, για παροχή μιας συμβουλής. Στη συνέχεια το Ε.Σ. χρησιμοποιεί το μηχανισμό εξαγωγής συμπερασμάτων για να χειρισθεί τους κανόνες της βάσης γνώσης. Αν το σύστημα έχει κάποιες απορίες σχετικά με κάποια

γεγονότα (facts) τότε απευθύνεται στο χρήστη για πρόσθετες πληροφορίες. Τα γεγονότα αυτά καθώς και όσα προκύπτουν στα διάφορα ενδιάμεσα στάδια λειτουργίας του Ε.Σ. αποθηκεύονται στη μνήμη εργασίας. Η διαδικασία συνεχίζεται μέχρι το σύστημα να καταλήξει σε ένα τελικό συμπέρασμα. Το συμπέρασμα αυτό μπορεί να είναι είτε η παροχή συμβουλής στο αρχικό ερώτημα, είτε να δηλώσει αδυναμία παροχής συμβουλής επειδή δεν υπάρχει επαρκής γνώση για αυτό.

Στη συνέχεια θα εξετασθούν αναλυτικότερα τα διάφορα τμήματα ενός αντιπροσωπευτικού έμπειρου συστήματος όπως αυτού που παρουσιάζεται στο σχήμα 2.1.



Σχήμα 2.1 Δομή ενός Έμπειρου Συστήματος

2.5.1 ΧΡΗΣΤΗΣ

Ο χρήστης ενός έμπειρου συστήματος μπορεί να είναι ένας από τους παρακάτω:

Πελάτης: Είναι κάποιος μη ειδικός ο οποίος χρησιμοποιεί το Ε.Σ. γιατί χρειάζεται τη συμβουλή του. Με τη χρήση του συστήματος μπορεί να βελτιώσει τη ποιότητα των αποφάσεων που παίρνει.

Φοιτητής-σπουδαστής: Εδώ ο χρήστης χρησιμοποιεί το Ε.Σ. για να αυξήσει τη γνώση του πάνω σε ένα ειδικό θέμα. Το σύστημα στη περίπτωση αυτή παίζει το ρόλο του εκπαιδευτή.

Ειδικός: Για να έχει μια δεύτερη γνώμη σε κάποια απόφαση που θέλει να πάρει ή για να τον βοηθά σε κάποιες δευτερεύουσες εργασίες ή τέλος γιατί θέλει να παρακολουθεί τη λογική, μέσω των σταδιακών βημάτων που ακολουθεί το σύστημα, για να οδηγηθεί σε κάποιο συμπέρασμα.

Κατασκευαστής: Ο κατασκευαστής χρησιμοποιεί το σύστημα για να ελέγχει, να βελτιώνει και να αυξάνει την γνώση που αυτό παρέχει.

2.5.2 ΣΥΣΤΗΜΑ ΕΠΙΚΟΙΝΩΝΙΑΣ

Μέσω του υποσυστήματος αυτού και με τη βοήθεια διαφόρων γραφικών οθονών για καλύτερα αποτελέσματα εξασφαλίζεται η επικοινωνία του χρήστη με το σύστημα σε φυσική γλώσσα.

Ο χρήστης μπορεί να επικοινωνεί με το σύστημα και να συντηρεί τις διάφορες βάσεις (γνώσης, δεδομένων, μοντέλων) που είναι απαραίτητες για τη σωστή λειτουργία του.

Οι συμβουλές που δέχεται, λαμβάνονται μέσω του υποσυστήματος αυτού. Ο χρήστης μπορεί να εκμεταλλευτεί τις ικανότητες επεξήγησης και να λάβει απαντήσεις στα ερωτήματα:

- Πώς οδηγήθηκε σε κάποιο συμπέρασμα;
- Γιατί απορρίφθηκαν κάποιες εναλλακτικές απαντήσεις;
- Ποια είναι η διαδικασία για να καταλήξει το σύστημα στο τελικό συμπέρασμα;
- Γιατί το σύστημα υποβάλλει κάποιες ερωτήσεις στη προσπάθειά του να οδηγηθεί στη τελική απάντηση;

2.5.3 ΜΗΧΑΝΙΣΜΟΣ ΕΞΑΓΩΓΗΣ ΣΥΜΠΕΡΑΣΜΑΤΩΝ

Η καρδιά ενός Ε.Σ. είναι το τμήμα εκείνο του προγράμματος που είναι γνωστό σαν μηχανισμός εξαγωγής συμπερασμάτων. Ο μηχανισμός αυτός είναι υπεύθυνος για τη διαχείριση και τον έλεγχο της γνώσης που βρίσκεται αποθηκευμένη τόσο στη βάση γνώσης όσο και στη μνήμη εργασίας, με στόχο τη διαμόρφωση συμπερασμάτων. Τα κύρια μέρη ενός τέτοιου μηχανισμού είναι ο διερμηνέας (interpreter), ο επιλογέας ή χρονοπρογραμματιστής (scheduler) και το τμήμα επιβολής συνέπειας (consistency enforcer).

Ο διερμηνέας, στα περισσότερα συστήματα είναι γνωστός σαν διερμηνέας κανόνων, εφαρμόζοντας στη βάση γνώσης τους αντίστοιχους κανόνες με σκοπό την παραγωγή νέας γνώσης. Για το

σκοπό αυτό εφαρμόζει διάφορες τεχνικές όπως στατιστικές μεθόδους ή μεθόδους ταυτοποίησης προτύπων (pattern matching). Με βάση αυτές τις μεθόδους εκτιμά τις πληροφορίες που βρίσκονται αποθηκευμένες στη βάση γνώσης, ώστε να ανακαλύψει τους ενδεδειγμένους κανόνες.

Ο επιλογέας είναι υπεύθυνος για τη στρατηγική ελέγχου του συστήματος, έτσι ασχολείται με τη τήρηση της σειράς εκτέλεσης των διαφόρων ενεργειών και υπολογίζει τα αποτελέσματα εφαρμογής των κανόνων. Αποφασίζει στην ουσία πότε και με ποια σειρά θα χρησιμοποιηθούν τα δεδομένα της βάσης γνώσης. Περιέχει τη γνώση εκτίμησης (assessment knowledge), ώστε να μπορεί να αξιολογεί τις εναλλακτικές διαδρομές έρευνας, και τη γνώση ελέγχου (control knowledge) για να μπορεί να συντονίζει τις διάφορες λειτουργίες.

Συχνά είναι απαραίτητο να περιέχεται στο μηχανισμό και γνώση, που αναφέρεται σε κάποιο συγκεκριμένο πρόβλημα. Στις περιπτώσεις αυτές, όπου δηλαδή ο μηχανισμός εξαγωγής συμπερασμάτων εμπεριέχει γνώση, θα θεωρείται ότι είναι εξαρτώμενος από τη συγκεκριμένη ειδική γνώση.

Η μνήμη εργασίας είναι ένας χώρος τον οποίο χρησιμοποιεί ο μηχανισμός εξαγωγής συμπερασμάτων, είτε για να τοποθετεί τις απαραίτητες πληροφορίες του τρέχοντος προβλήματος, είτε για να αποθηκεύει ενδιάμεσα αποτελέσματα στη πορεία του προς αναζήτηση της τελικής λύσης. Όπως φαίνεται και στο πάρα πάνω σχήμα, στη μνήμη εργασίας καταχωρούνται τα ακόλουθα τρία είδη αποφάσεων:

Σχεδίαση δράσης: Αναφέρεται στο τρόπο επίλυσης του τρέχοντος προβλήματος.

Πιθανές ενέργειες: Είναι ένας κατάλογος των ενδεχόμενων επόμενων ενεργειών.

Επίλυση: Εδώ καταχωρούνται οι εναλλακτικές υποθέσεις και οι ενέργειες που το σύστημα έχει επιτελέσει.

2.5.4 ΒΑΣΕΙΣ ΓΝΩΣΗΣ

Στη βάση γνώσης περιέχονται δύο βασικά στοιχεία, τα γεγονότα (facts) και οι κανόνες (rules). Τα γεγονότα αναφέρονται τόσο στη κατάσταση του προβλήματος, όσο και στην αντίστοιχη θεωρία. Οι κανόνες κατευθύνουν την εκμετάλλευση της υπάρχουσας γνώσης για την επίλυση του τρέχοντος προβλήματος. Συχνά αντί για κανόνες χρησιμοποιούνται κάποιοι ειδικοί κανόνες, που ονομάζονται ευρετικοί (heuristics).

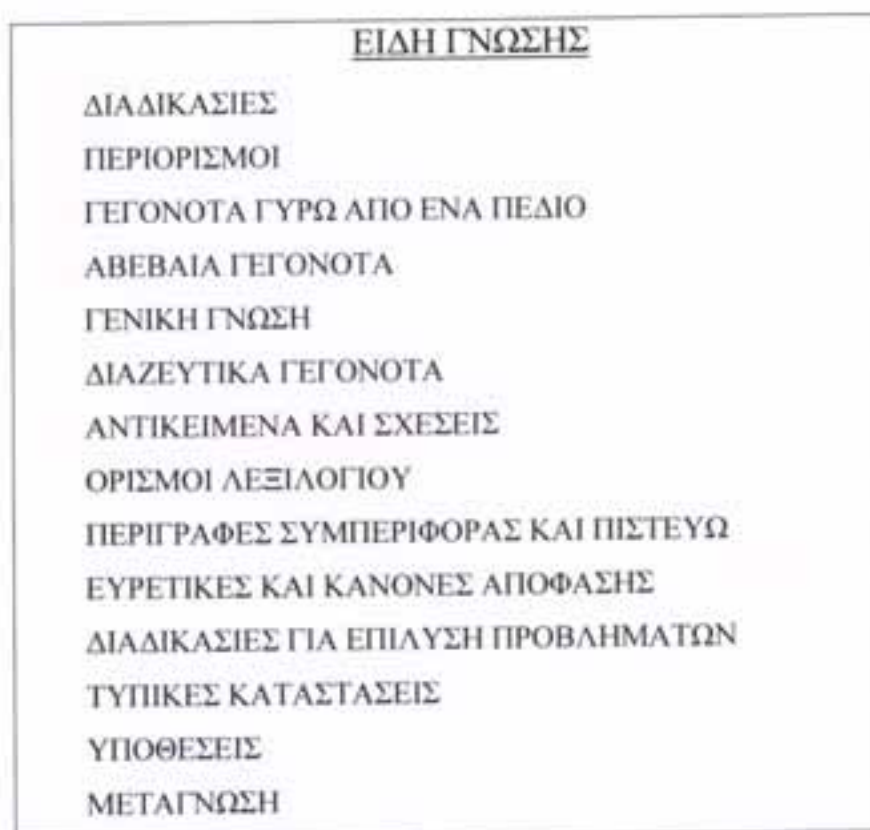
2.5.4.1 ΥΠΟΣΥΣΤΗΜΑ ΑΠΟΣΠΑΣΗΣ ΓΝΩΣΗΣ

Το υποσύστημα αποτελείται από τα μέσα γνώσης του πεδίου και τον μηχανικό γνώσης ο οποίος εργάζεται με στόχο την απόσπαση, συγκέντρωση, κωδικοποίηση και μεταφορά της γνώσης-εμπειρίας κάποιου πεδίου, σε μια βάση γνώσης ενός Ε.Σ. Σαν πηγές γνώσης, εκτός από έναν ή περισσότερους ειδικούς, μπορούν να χρησιμοποιηθούν οι γνώσεις που συγκεντρώνονται από τα αποτελέσματα ειδικών ερευνών, τις βάσεις δεδομένων.

2.5.5 Μηχανική της γνώσης (Knowledge Engineering)

Σαν γνώση μπορεί να θεωρηθεί μια συλλογή ειδικών γεγονότων (facts), διαδικασιών (procedures) και κανόνων (rules). Μερικά από τα

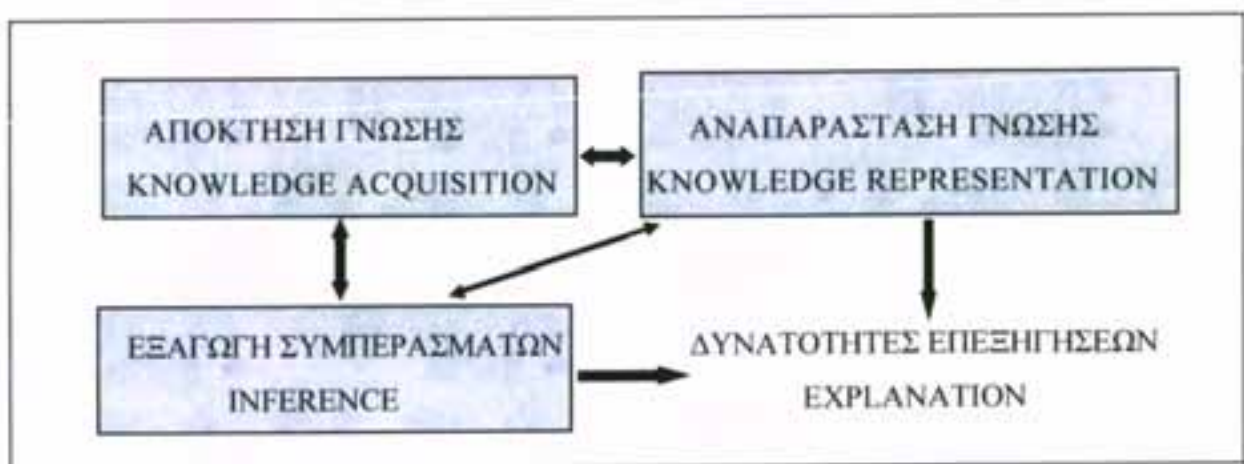
είδη, της χρησιμοποιούμενης γνώσης στα Ε.Σ. παρουσιάζονται στο σχήμα 2.2 (Fikes and Kehler, 1985). Όλες οι ενέργειες που αναφέρονται στη γνώση αποτελούν το αντικείμενο, της μηχανικής της γνώσης (Feigenbaum and McCorduck, 1983) ενώ οι ειδικοί που ασχολούνται με αυτή ονομάζονται μηχανικοί γνώσης. Η μηχανική της γνώσης εμπεριέχει και τη συνεργασία των ειδικών (experts), ενός τομέα, με το μηχανικό γνώσης με στόχο τη κωδικοποίηση της γνώσης και το καθορισμό με σαφήνεια των λογικών διαδικασιών που ο ειδικός χρησιμοποιεί για να επιλύσει ένα πραγματικό πρόβλημα (σχήμα 2.3). Η διαδικασία ανάπτυξης ενός έμπειρου συστήματος και η συνεργασία του ειδικού με το μηχανικό γνώσης τον οδηγεί (εξαναγκάζει) να βρει τρόπους να εκφράσει τη γνώση και το τρόπο σκέψης του με μεγάλη σαφήνεια (Goodman et al., 1991).



Σχήμα 2.2 Είδη Γνώσης

Ο μηχανικός της γνώσης πρέπει να δώσει μεγάλη προσοχή στο διαχωρισμό των δομών της γνώσης από τους μηχανισμούς ελέγχου. Επίσης το Ε.Σ. θα πρέπει να έχει την ικανότητα να εξηγεί τι έκανε (what), γιατί το έκανε (why) και πως το έκανε (how).

Ένα άλλο σημαντικό πρόβλημα εκτός από την απόσπαση της γνώσης είναι και ο τρόπος αναπαράστασής της ούτως ώστε να είναι δυνατή η λειτουργία του μηχανισμού εξαγωγής συμπερασμάτων του Ε.Σ.



Σχήμα 2.3 Δραστηριότητες

2.5.5.1 ΑΠΟΚΤΗΣΗ ΓΝΩΣΗΣ (Knowledge Acquisition)

Η πολυπλοκότητα της διαδικασίας απόκτησης γνώσης οφείλεται, κατά κύριο λόγο, στην ύπαρξη πολλών μέσων πληροφόρησης και στα διαφορετικά είδη γνώσης (Boose 1985, Shaw and Gaines 1986, Wright and Ayton 1987, Johnson and Johnson 1987, McGraw and Harbison 1989, Brule and Blount 1989, McGovern 1991). Σημαντικός παράγων

της διαδικασίας απόσπασης της γνώσης είναι και το κόστος απόκτησής της (Dos Santos and Mookerjee, 1993).

Διαδικασία απόκτησης γνώσης

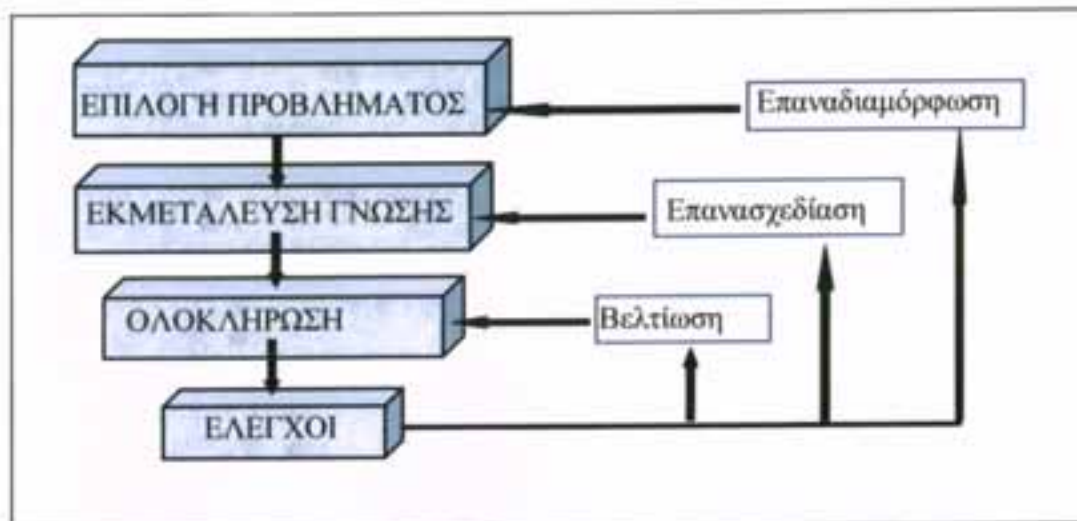
Στο σχήμα 2.4 παρουσιάζονται τα τέσσερα κύρια στάδια απόκτησης της γνώσης (Doukidis and Whitley, 1988). Κατά τη διάρκεια του πρώτου σταδίου γίνεται αρχικά η επιλογή του προς επίλυση προβλήματος, στη συνέχεια, αναγνωρίζονται. Κατά το στάδιο εκμείευσης της γνώσης από έναν ή πολλούς ειδικούς, αρχικά γίνεται η απόσπαση της γνώσης (knowledge elicitation) και ακολουθεί η διαδικασία απόκτησής της (knowledge acquisition), κατά την οποία η αποσπασθείσα γνώση οργανώνεται και αναπαρίσταται, από το μηχανικό γνώσης, σε μορφή κατάλληλη για χρήση από το προς ανάπτυξη Ε.Σ.

Κατά το στάδιο της ολοκλήρωσης η αποκτηθείσα γνώση ενσωματώνεται στο Ε.Σ. αφού προηγουμένως διαμορφωθεί καταλλήλως ούτως ώστε να είναι δυνατή η χρήση της από το Ε.Σ.

Στη διαδικασία απόκτησης της γνώσης περιέχονται και οι έλεγχοι που αφορούν την εκτίμηση (evaluation), την εγκυρότητα (validation) και την επαλήθευση (verification) της αποκτηθείσης γνώσης (O'Keefe 1987, Marcot, 1987, Nguyen 1987, Parsaye 1988).

Τα κύρια χαρακτηριστικά του διευκρινίζονται και καθορίζονται με σαφήνεια. Τα διάφορα στάδια αλληλεπιδρούν μεταξύ τους μέσω των εργασιών επαναδιαμόρφωσης, επανασχεδίασης και βελτίωσης του αναπτυσσόμενου συστήματος.

Το σύστημα βρίσκεται κάτω από τη συνεχή παρακολούθηση και εκτίμηση του ή των ειδικών, η γνώση των οποίων χρησιμοποιήθηκε για την ανάπτυξή του.



Σχήμα 2.4 Στάδια Απόκτησης Γνώσης

Σημαντικός παράγοντας στην ανάπτυξη ενός Ε.Σ. είναι ο αριθμός των συμμετεχόντων ειδικών (multiple experts). Η χρησιμοποίηση πολλών ειδικών επιβάλλεται κυρίως είτε για λόγους διεύρυνσης των προτεινόμενων λύσεων ενός προβλήματος, είτε για να αυξηθούν οι δυνατότητες διαφορετικών λογικών προσεγγίσεων. Αυτή η συνεργασία παρουσιάζει και μειονεκτήματα όπως αυτά που οφείλονται στις αλληλοσυγκρουόμενες απόψεις-εκτιμήσεις των ειδικών, τις οποίες πρέπει να επιλύσει ο μηχανικός της γνώσης (Boose 1985, LeClair 1985, Nii 1986, Mittal 1985, Shaw and Gaines 1986).

Μέθοδοι απόσπασης γνώσης

Η απόσπαση της γνώσης μπορεί να γίνει με ένα από τους παρακάτω τρόπους (Wielinga 1990):

Χειρογραφικά (manual knowledge elicitation)

Μη καταγραμμένη γνώση (Waterman and Newell 1976, Ericsson and Simon 1984, Boose 1986b, Hart 1986, Abdul-Gader and Kozar 1990)

- Προσωπικές συνεντεύξεις (Hart 1986, Wolf gram 1987, Hoffman 1987, McGraw and Harbison-Briggs 1989, Agarwal and Tanniru 1990).
- Παρατήρηση του τρόπου εργασίας του ειδικού κατά την επίλυση του προβλήματος (Newall and Simon 1972, Ericsson and Simon 1984, Wolfgram 1987).
- Συμπλήρωση ειδικών ερωτηματολογίων ή περιγραφών από τους ειδικούς (Wolfgram et al., 1987).

Καταγραμμένη γνώση

- Γνώση από βάσεις δεδομένων, βιβλία, επιστημονικά περιοδικά κ.λ.π. (Rothman, 1988).

Με τη βοήθεια Η/Υ (computer-aided knowledge acquisition)

Η απόσπαση γνώσης γίνεται με τη βοήθεια ειδικών προγραμμάτων Η/Υ και μπορεί να γίνει είτε με την παντελή απουσία του ειδικού ή/και του μηχανικού γνώσης, είτε με τη μειωμένη συμμετοχή τους (Quinlan 1983, 1984, 1986, Gale 1986, Cronan 1991, Turban 1993, Deng, 1993).

Αυτόματη εκμάθηση (auto-intelligence)

Τα συστήματα αυτά προσπαθούν, μέσω της αλληλεπίδρασής τους με ένα ειδικό, αρχικά να αποκτήσουν την απαραίτητη γνώση για την επίλυση ενός συγκεκριμένου προβλήματος, στη συνέχεια επεξεργάζονται καταλλήλως αυτή τη γνώση και τελικά δημιουργούν την αντίστοιχη βάση γνώσης (Parsaye 1988).

Αλληλεπιδραστικότητα απόκτηση γνώσης (interactive knowledge acquisition)

Με τη μέθοδο αυτή ο μηχανικός γνώσης στη προσπάθειά του για απόσπαση γνώσης χρησιμοποιεί ένα συνδυασμό χειρογραφικών μεθόδων και ειδικών εργαλείων πληροφορικής (Gaines, 1988). Μερικά από τα εργαλεία που μπορεί να χρησιμοποιήσει είναι τα συστήματα επικοινωνίας, παροχής επεξηγήσεων, μεταφοράς εμπειρίας, ελέγχου των βάσεων γνώσης κ.α (Freiling 1985, Boose 1986, Diederich 1987, Gaines 1988).

Προβλήματα απόσπασης γνώσης

Κατά τη διαδικασία απόσπασης της γνώσης από τον ειδικό και της μεταφοράς της στο Ε.Σ., με τη βοήθεια του μηχανικού γνώσης, παρουσιάζονται διάφορες δυσκολίες και προβλήματα (Welbank, 1983; Hart, 1986; Doukidis and Whitley, 1988; Poulymenakou and Doukidis, 1988; Gaines, 1988; Turban, 1993), είναι:

- Ένα αρχικό πρόβλημα είναι το διαφορετικό γνωστικό υπόβαθρο των συμμετεχόντων στη διαδικασία μεταφοράς γνώσης. Στην ανάπτυξη ενός έμπειρου συστήματος λαμβάνουν μέρος τουλάχιστον τέσσερα άτομα, ο ειδικός (expert), ο μηχανικός γνώσης (knowledge engineer), ο κατασκευαστής του συστήματος (system analyst and programmer) και ο χρήστης (user). Ο κάθε ένας από αυτούς μπορεί να έχει διαφορετική εμπειρία, να χρησιμοποιεί διαφορετική ορολογία και να έχει εντελώς διαφορετικό γνωστικό αντικείμενο από τους υπόλοιπους.
- Δεν υπάρχει μια γενικά αποδεκτή μεθοδολογία για τη σχεδίαση και το προγραμματισμό της απόσπασης γνώσης καθώς και για την εκτίμηση των αποτελεσμάτων.
- Η αποσπώμενη γνώση συχνά περιέχει αντιφάσεις ενώ άλλοτε παρουσιάζει ελλείψεις. Για το λόγο αυτό πρέπει να υφίσταται βελτιώσεις πριν χρησιμοποιηθεί.
- Συχνά ο ειδικός αγνοεί το τρόπο με τον οποίο επιλύει ένα πρόβλημα. Υπάρχει επομένως ο κίνδυνος να μεταφέρει στο μηχανικό γνώσης διαφορετικό τρόπο σκέψης από αυτόν που χρησιμοποιεί στη πραγματικότητα.
- Δυσκολίες προκύπτουν επίσης και όταν ο ειδικός δεν έχει τη ικανότητα να εκφράζει με σαφήνεια το τρόπο σκέψης του ή τον εκφράζει εσφαλμένα ή εκφράζει μόνο μέρος της.
- Μερικοί ειδικοί είναι άτομα απρόσιτα, αγενή ή γενικότερα άτομα μη συνεργάσιμα. Προβλήματα παρατηρούνται στη συνεργασία με τον ειδικό όταν αυτός δεν πιστεύει στην όλη διαδικασία απόσπασης και εκμετάλλευσης της γνώσης του ή όταν είναι υποχρεωμένος να μετέχει στη διαδικασία ενώ ο ίδιος δεν το επιθυμεί ή όταν η διαδικασία συνεχίζεται για μεγάλο χρονικό διάστημα.
- Σχεδόν πάντοτε ο ειδικός είναι άτομο πολυάσχολο και ως εκ τούτου δεν διαθέτει επαρκή χρόνο για το μηχανικό γνώσης.

- Η απασχόληση του ειδικού για μεγάλο χρονικό διάστημα είναι, οικονομικά ασύμφορη.
- Δυσκολίες, μπορεί να προκύψουν τόσο στη μεταφορά της γνώσης όσο και στην αναπαράστασή της, λόγω της πολύ συμπυκνωμένης μορφής της.
- Όταν δεν βρεθεί μια κοινή γλώσσα συνεννόησης μεταξύ ειδικού και μηχανικού γνώσης.
- Όταν δεν υπάρχουν καλές σχέσεις μεταξύ των δύο, που είναι απαραίτητο αφού λόγω της εργασίας τους θα πρέπει να εργάζονται πολλές ώρες μαζί.

ΚΕΦΑΛΑΙΟ 3

3.1 ΑΝΑΠΑΡΑΣΤΑΣΗ ΤΗΣ ΓΝΩΣΗΣ (Knowledge Representation)

Η αποσπασθείσα γνώση του ειδικού πρέπει να οργανωθεί και να αναπαρασταθεί κατά τέτοιο τρόπο ώστε να είναι όσο το δυνατόν πιο κατανοητή τόσο από το Ε.Σ. όσο και από τον ίδιο τον ειδικό. Η γνώση που αναφέρεται στη διαδικασία επίλυσης του συγκεκριμένου προβλήματος είναι περισσότερο γενική (αφορά το τρόπο επίλυσης περισσότερων προβλημάτων) και τοποθετείται στο μηχανισμό εξαγωγής συμπερασμάτων του Ε.Σ., ενώ η γνώση που αναφέρεται στο συγκεκριμένο πρόβλημα είναι ειδική και αποθηκεύεται στη βάση γνώσης του.

Μέθοδοι αναπαράστασης

Η επιλογή ενός τρόπου για την αναπαράσταση της γνώσης ενός συγκεκριμένου προβλήματος είναι σημαντικότερη εργασία, δεδομένου ότι προσδιορίζει το αν θα προσεγγισθεί ικανοποιητικά η συλλογιστική του ειδικού. Πρέπει λοιπόν να επιλεγεί η κατάλληλη μέθοδος για τη συγκεκριμένη γνώση και για το συγκεκριμένο πρόβλημα.

Οι μέθοδοι αναπαράστασης της γνώσης, αν και έχουν μεγάλες διαφορές μεταξύ τους εν τούτοις, έχουν δύο κοινά χαρακτηριστικά:

- Υλοποιούνται με τη βοήθεια μιας από τις υπάρχουσες γλώσσες προγραμματισμού και
- Χρησιμοποιούν τα γεγονότα και τη γνώση που περιέχουν, στη προσπάθειά τους για αναζήτηση λογικών συμπερασμάτων με τη βοήθεια των ειδικών τεχνικών αναζήτησης (search) και ταυτοποίησης (pattern-matching) του μηχανισμού εξαγωγής συμπερασμάτων στη βάση γνώσης του Ε.Σ.

Οι διάφορες μέθοδοι αναπαράστασης διαχωρίζονται σε περιγραφικές ή δηλωτικές (declarative representation), που χρησιμοποιούνται για αναπαράσταση γεγονότων και ισχυρισμών, και διαδικαστικές (procedural representation), που χρησιμοποιούνται για ενέργειες και διαδικασίες. Γνωστές δηλωτικές μέθοδοι αναπαράστασης είναι: λογική (logic), πλαίσια (frames), σημασιολογικά δίκτυα (semantic networks), κανόνες (rules) και σενάρια (scripts), ενώ διαδικαστικές είναι: διαδικασίες (procedures), υπορουτίνες (subroutines) και κανόνες (rules). Οι δηλωτικές μέθοδοι περιγράφουν τη γνώση ανεξάρτητα από το τρόπο χειρισμού της και την απεικονίζουν σαν δεδομένο, σε αντίθεση με τις διαδικαστικές που εκτός από την ειδική γνώση περιέχουν και γνώση σχετικά με το τρόπο χειρισμού της και την απεικονίζουν σαν πρόγραμμα. Στα Ε.Σ. που χρησιμοποιούν δηλωτική μέθοδο αναπαράστασης υπάρχει διαχωρισμός της ειδικής γνώσης από τη γνώση ελέγχου και ως εκ τούτου έχουν μεγαλύτερη διαφάνεια.

Στη διαδικαστική αναπαράσταση, το μεγαλύτερο μέρος της γνώσης αναπαρίσταται με τη μορφή ειδικών διαδικασιών που αντιμετωπίζουν συγκεκριμένα προβλήματα και οι οποίες ενεργοποιούνται μέσα από άλλες διαδικασίες. Μεταξύ αυτών υπάρχει μια ειδική κατηγορία διαδικασιών που ενεργοποιούνται όταν ισχύσει μία συνθήκη (demons). Η υλοποίηση της διαδικαστικής μορφής αναπαράστασης της γνώσης είναι εύκολη, παρέχει μεγάλες εκφραστικές δυνατότητες αλλά έχει μικρή διαφάνεια. Χρησιμοποιεί απλές μορφές μηχανισμών για την εξαγωγή συμπερασμάτων. Λόγω της μεγάλης αδιαφάνειας και της αλληλεπίδρασης που υπάρχει ανάμεσα στις διαδικασίες είναι δύσκολες οι εργασίες συντήρησης καθώς και οι εργασίες αναθεώρησης της γνώσης.

Η διαδικαστική μορφή αναπαράστασης της γνώσης χρησιμοποιείται κυρίως σε συνδυασμό με τη δηλωτική μορφή για την αντιμετώπιση ειδικών προβλημάτων.

Κάθε μία από τις υπάρχουσες μεθόδους αναπαριστά καλύτερα κάποιες ειδικές μορφές γνώσης. Έτσι, για να μπορούμε να έχουμε την καλύτερη δυνατή αναπαράσταση της γνώσης ενός ειδικού προβλήματος, δεν θα πρέπει να μιλάμε πάντοτε για την αποκλειστική εφαρμογή μιας συγκεκριμένης μεθόδου αλλά για την εύρεση του αποτελεσματικότερου συνδυασμού μεθόδων αναπαράστασης.

Κατά τη διαδικασία επιλογής των μεθόδων αυτών θα πρέπει να συνεκτιμηθεί το γεγονός των διαφορετικών μορφών μηχανισμών εξαγωγής συμπερασμάτων που χρησιμοποιείται σε κάθε μορφή αναπαράστασης της γνώσης.

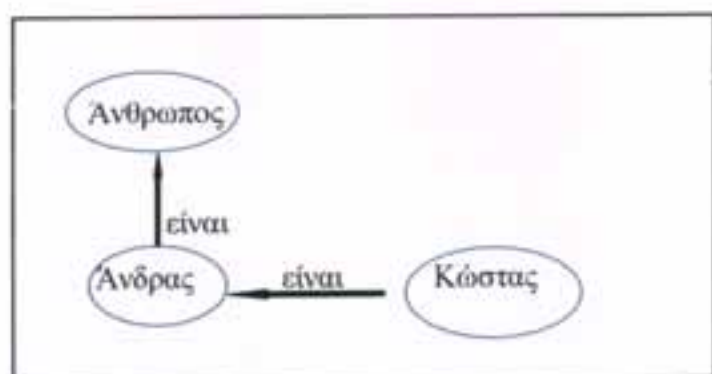
Ακολούθως θα περιγραφούν μερικές από τις ευρύτερα χρησιμοποιούμενες μεθόδους αναπαράστασης της γνώσης.

Σημαντικά -Σημασιολογικά δίκτυα (Semantic Networks)

Με τα σημαντικά δίκτυα γίνεται προσπάθεια να οργανωθεί και να αναπαρασταθεί η αποσπασθείσα γνώση του ειδικού με τρόπο ανάλογο με αυτόν που χρησιμοποιούν οι άνθρωποι για να αναπαραστούν την γνώση τους. Το σημαντικό δίκτυο αποτελείται από ένα δίκτυο κόμβων (nodes) και συνδέσμων τόξων (links). Οι κόμβοι αναπαριστούν διάφορα γεγονότα που μπορεί να είναι φυσικά αντικείμενα (Κώστας, γραφείο, σκύλος), έννοιες (αριθμοί, ποτάμι), καταστάσεις και περιγραφές (είναι, έχει). Από την άλλη μεριά οι συνδέσμοι περιγράφουν τις σχέσεις που υφίστανται μεταξύ των γεγονότων.

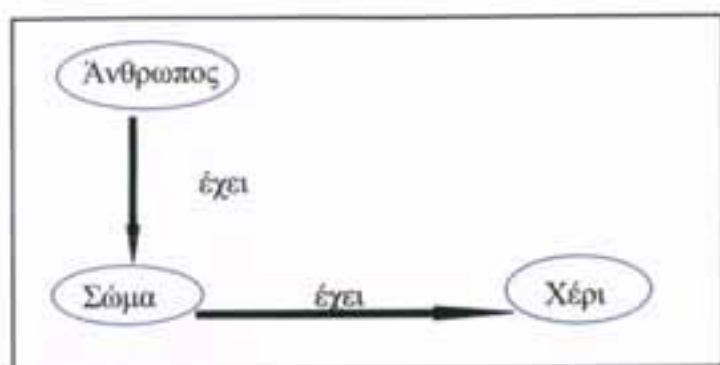
Η σημαντικότερη μορφή σύνδεσης είναι η: *είναι* (is a). Με αυτή απεικονίζεται μια υποπερίπτωση ενός γεγονότος ανώτερης τάξης (class). Στο σχήμα 3.1 απεικονίζεται η γνώση ότι ο "Κώστας" *είναι* μια υποπερίπτωση του ανώτερου γεγονότος "Άνδρας", το οποίο με τη σειρά του *είναι* υποπερίπτωση του γεγονότος "Άνθρωπος". Ο "Άνδρας" είναι μέλος

της τάξης "Άνθρωπος" και ο "Κώστας" είναι μέλος της τάξης "Άνδρας" και κατ'επέκταση και της τάξης "Άνθρωπος". Παρατηρείται ότι οι ιδιότητες και τα χαρακτηριστικά του γεγονότος "Άνθρωπος" κληρονομούνται στα ιεραρχικά κατώτερα γεγονότα που σχετίζονται με αυτό. Έτσι ο "Άνδρας" κληρονομεί τις ιδιότητες και τα χαρακτηριστικά του "Άνθρωπος", ενώ στη συνέχεια ο "Κώστας" κληρονομεί τις ιδιότητες και τα χαρακτηριστικά του "Άνδρας". Κληρονομικότητα (inheritance) είναι η δυνατότητα που έχουν τα γεγονότα μιας τάξης να κληρονομήσουν τις ιδιότητες και τα χαρακτηριστικά εκείνων που βρίσκονται σε ιεραρχικά ανώτερο από αυτά επίπεδο μέσα στο δίκτυο.



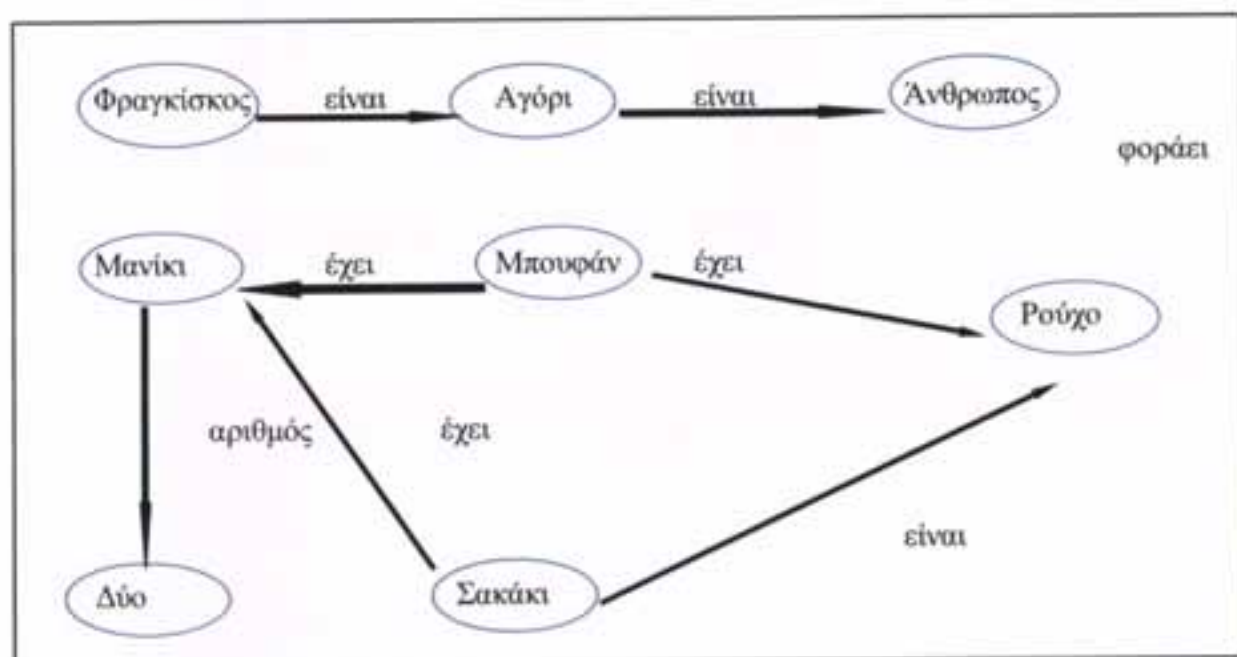
Σχήμα 3.1 Σύνδεση «Είναι»

Μια άλλη μορφή σύνδεσης είναι η: *έχει* (has a). Με αυτή τη σύνδεση περιγράφονται οι σχέσεις ενός γεγονότος με τα μέρη από τα οποία αποτελείται. Έτσι στο σχήμα 3.2. Ο "Άνθρωπος" *έχει* "Σώμα" και αυτό με τη σειρά του "Χέρι". Το "Χέρι" είναι ένα τμήμα από το "Σώμα" αλλά δεν έχει τις ιδιότητες, και τα χαρακτηριστικά του.



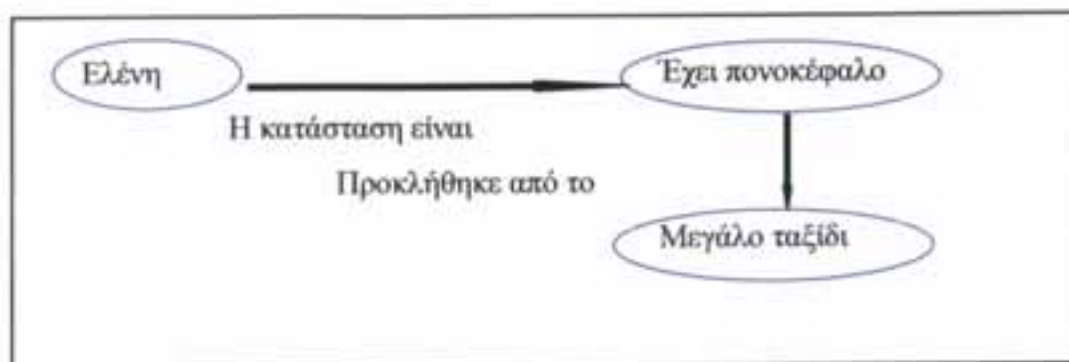
Σχήμα 3.2 Σύνδεση «Έχει»

Ο ορισμός μιας σχέσης μεταξύ δύο ανεξάρτητων γεγονότων είναι μία άλλη μορφή συνδέσμου. Στο σχήμα 3.3 παρουσιάζονται δύο τέτοια είδη συνδέσμων, έτσι ο "Άνθρωπος" *φοράει* "Ρούχο" και το "Μπουφάν" έχει "Δύο" "Μανίκι".



Σχήμα 3.3 Σύνδεση «Ορισμού»

Η σύνδεση μεταξύ δύο γεγονότων, που οφείλεται σε εμπειρική ευρετική γνώση (heuristic) είναι ένα διαφορετικό είδος σύνδεσης. Η «Ελένη» έχει πονοκέφαλο επειδή έκανε «Μεγάλο ταξίδι» (σχήμα 3.4).

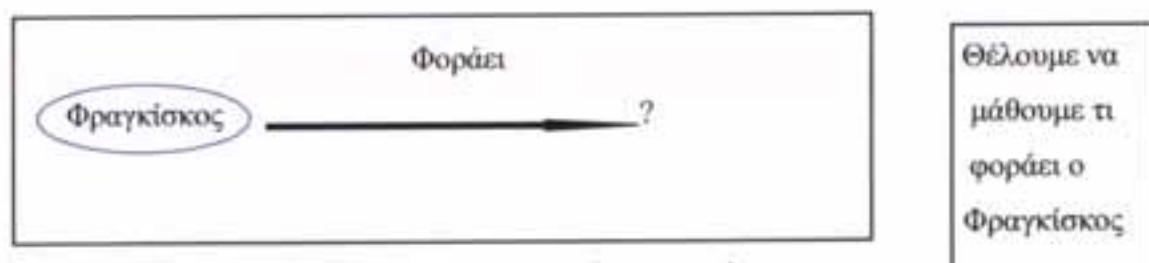


Σχήμα 3.4 Σύνδεση «ευρετική»

Τα δίκτυα αυτού του τύπου είναι ευέλικτα αφού μπάρουν να επεκταθούν με ευκολία αρκεί να προστεθούν οι απαραίτητοι κόμβοι και οι συνδέσεις τους. Ένα άλλο πλεονέκτημα που έχουν είναι ότι επιβάλουν την ιεραρχία στις σχέσεις.

Ο μηχανισμός έκφρασης της λογικής στα δίκτυα γίνεται με τη προσπάθεια ταυτοποίησης δικτυωτών δομών (σχήμα 3.5).

Το σύστημα αναζητά ανάμεσα στους συνδέσμους του γεγονότος «Φραγκίσκος», αν υπάρχει



Σχήμα 3.5 Ταυτοποίηση δικτυωτών δεσμών

κάποιος ή κάποιοι με το όνομα «Φοράει». Η αναζήτηση στα σημαντικά δίκτυα είναι δύσκολη διαδικασία. Ένα άλλο πρόβλημα, που σχετίζεται όμως με τη κληρονομικότητα, είναι η δυσκολία χειρισμού των εξαιρέσεων, οι

οποίες αντιμετωπίζονται με τη προσθήκη νέων γεγονότων. Η ύπαρξη πολλών εξαιρέσεων αυξάνει τη πολυπλοκότητα του δικτύου.

Πλαίσια (frames)

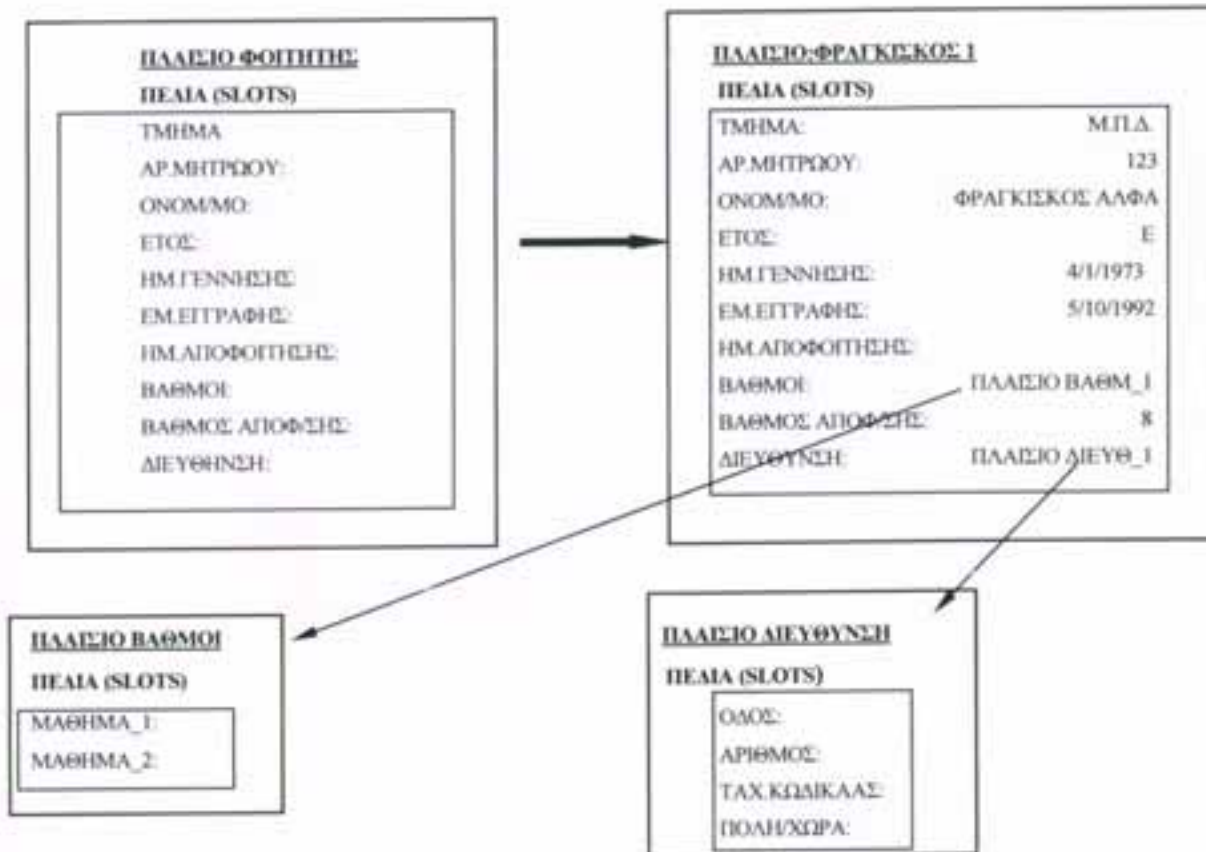
Το πλαίσιο είναι μια δομή δεδομένων που αναπαριστά ένα συγκεκριμένο αντικείμενο ή γεγονός (Minsky 1975, Doukidis 1988, Arcidiacono 1988, Blanning 1988, Pracht 1990, Δουκίδης και Αγγελίδης 1992, Turban 1993). Σε κάθε αντικείμενο υπάρχουν πεδία (σχισμές - slots) -στα οποία αποθηκεύονται είτε οι πληροφορίες που το αφορούν είτε τα χαρακτηριστικά που διαθέτει. Τα πεδία (σχισμές) αποτελούνται από υποπεδία (subslots - facets). Σε αυτά μπορεί να αποθηκεύονται άλλα πλαίσια, διαδικασίες, κανόνες, προκαθορισμένες τιμές ή οποιασδήποτε μορφής πληροφορία (σχήμα 3.6). Ο χειρισμός των πλαισίων καθώς και η οργάνωση συστημάτων βασιζόμενων σε πλαίσια είναι παρόμοιες με αυτών των σημαντικών δικτύων. Μια τέτοια θεώρηση μπορεί να γίνει αν στη θέση των κόμβων των σημαντικών δικτύων τοποθετηθούν πλαίσια. Τα πλαίσια είναι ένα αντιπροσωπευτικό δείγμα αντικειμενοστραφούς προγραμματισμού (object-oriented programming) (Stefik and Bobrow, 1986).

Η δυνατότητα των πλαισίων να καλούν διαδικασίες, τους δίνει το πλεονέκτημα της ταυτόχρονης χρήσης τόσο της διαδικαστικής όσο και της δηλωτικής, αναπαράστασης. Η κλήση ενός πλαισίου από ένα άλλο, εισάγει τη έννοια της ιεραρχίας.

Ένα άλλο χαρακτηριστικό των πλαισίων είναι η κληρονομικότητα, η οποία είναι δυνατή μέσω της ιδιότητας να διασυνδέονται μεταξύ τους. Ένα πλαίσιο μπορεί να κληρονομεί χαρακτηριστικά από περισσότερα του ενός πλαίσια.

Τα πλαίσια αποτελούν μια αρκετά περίπλοκη μέθοδο αναπαράστασης της γνώσης. Η αναζήτηση, στη προσπάθεια ανεύρεσης του κατάλληλου

πλαίσιου μίας συγκεκριμένης περίπτωσης, ξεκινά κατά κανόνα από τα πιο ειδικά και επεκτείνεται στις γενικότερες περιπτώσεις.



Σχήμα 3.6. Ιεραρχική διασύνδεση πλαισίων

Κανόνες παραγωγής (Production rules)

Τα πρώτα συστήματα αναπτύχθηκαν από τους Newell και Simon το 1972 για να περιγράψουν την ανθρώπινη γνώση. Στα συστήματα αυτά η γνώση αναπαρίσταται με τη βοήθεια κανόνων που έχουν τη μορφή:

IF(Εαν) συνθήκες (υποθέσεις)
THEN(Τότε) ενέργειες (συμπεράσματα)

Αυτό σημαίνει ότι όταν οι συνθήκες είναι αληθείς τότε συνεπάγεται ότι αληθεύουν οι ενέργειες και αντιστρόφως. Παρομοίως όταν οι συνθήκες

είναι ψευδείς συνεπάγεται ότι και οι ενέργειες είναι ψευδείς και αντιστρόφως. Η λογική αυτή είναι γνωστή σαν *modus ponens* (Harmon and King, 1985).

Τα χαρακτηριστικά των κανόνων είναι (Curry and Moutinho, 1991):

1. Η σύνταξη των κανόνων πρέπει να περιέχει περιορισμένο κείμενο με σαφή έννοια και να είναι κατανοητό σε κάθε στάδιο εργασιών (π.χ. στάδια παροχής συμβουλών και αιτιολόγησης της συλλογιστικής του).

2. Οι κανόνες μπορούν να παρίστανται με σύνθετη δομή ή με διασυνδεόμενες ομάδες κανόνων.

3. Για το χειρισμό τους και την κατάλληλη επεξεργασία τους, απαιτείται η χρήση ειδικών γλωσσών, όπως Lisp, Prolog κ.λ.π., διαφόρων shells, ή ειδικά κατασκευασμένου λογισμικού για το χειρισμό των διαδικασιών "λογικής", του συστήματος.

4. Η γνώση που χρησιμοποιείται έχει ποιοτικές διαστάσεις.

5. Παρέχει τη δυνατότητα στα συστήματα να αποκτούν (αντλούν) γνώση (*automated knowledge acquisition*) από παραδείγματα ή εφαρμογές αλγορίθμων όπως ο ID3 του Quinlan (1986)).

Οι κανόνες χρησιμοποιούνται για τη περιγραφή ευρετικών διαδικασιών επίλυσης προβλημάτων. Ο ορισμός των συνθηκών των κανόνων γίνεται με μεγάλη σαφήνεια με αποτέλεσμα να μειώνονται οι αλληλεπιδράσεις μεταξύ τους. Κάθε κανόνας περιγράφει μία συγκεκριμένη περίπτωση γνώσης με αποτέλεσμα την ανεξαρτησία και τη καθαρότητα της αναπαριστώμενης γνώσης. Οι κανόνες χρησιμοποιούνται τόσο για την αναπαράσταση της δηλωτικής (*declarative*) όσο και της διαδικαστικής (*procedural*) γνώσης.

Κατά την ανάπτυξη ενός Ε.Σ. δημιουργούνται δύο είδη κανόνων ανάλογα με το είδος της γνώσης που αναπαριστούν. Οι δηλωτικοί κανόνες γνώσης είναι αυτοί που περιέχουν τη γνώση για το πρόβλημα, ενώ οι διαδικαστικοί κανόνες περιέχουν τη γνώση για το τρόπο επίλυσής του. Οι

πρώτοι αποτελούν τη βάση γνώσης ενώ οι δεύτεροι περιέχονται στο μηχανισμό εξαγωγής συμπερασμάτων. Υπάρχει άλλο ένα είδος κανόνων, οι μετα-κανόνες (metarules) οι οποίοι περιέχουν γνώση για το χειρισμό τόσο των άλλων δύο τύπων κανόνων όσο και για το χειρισμό των ίδιων. Οι μετά-κανόνες περιέχονται και αυτοί στο μηχανισμό εξαγωγής συμπερασμάτων.

Στη συνέχεια δίνονται τα σημαντικότερα πλεονεκτήματα και μειονεκτήματα των κανόνων παραγωγής.

Πλεονεκτήματα:

- Έχουν "φυσική" μορφή απεικόνισης της γνώσης, απλή σύνταξη και είναι εύκολη η κατανόηση της γνώσης που αναπαριστούν.
- Ανεξαρτησία κανόνων.
- Παρέχουν ευκολίες μεταβολών.
- Η ανάπτυξη και η συμπλήρωση της βάσης γνώσης μπορεί να γίνει σταδιακά.
- Η διαδικασία εξαγωγής συμπερασμάτων και η παροχή επεξηγήσεων είναι σχετικά εύκολη.
- Δυνατότητα να αναπαριστούν αβέβαιες καταστάσεις.

Μειονεκτήματα:

- Δυσκολίες στη αναπαράσταση γενικής γνώσης όσον αφορά έννοιες και αντικείμενα.
- Δυσκολία στην αναπαράσταση σύνθετης γνώσης κυρίως λόγω του μεγάλου αριθμού κανόνων που απαιτείται για αυτό.
- Δυσκολία κατανόησης του καθολικού αλγορίθμου.
- Περιορισμοί στην αναζήτηση λύσης λόγω του μεγάλου αριθμού κανόνων.

Λογική (formal logic)

Με τη λογική μέθοδο αναπαράστασης της γνώσης, γίνεται έλεγχος μόνο των σχέσεων μεταξύ των προτάσεων (Goodman et all., 1991). Δεν την ενδιαφέρει το περιεχόμενο των προτάσεων αλλά μόνο το αν είναι γραμμένες συντακτικά ορθά. Χρησιμοποιεί απλές μορφές σύνταξης, είναι εκφραστικά σαφής και ακριβής μέθοδος αλλά παρουσιάζει αδυναμίες στη κατανόηση των εννοιών που περιέχονται στις προτάσεις, ενώ απαιτεί πολύπλοκες ευρετικές μεθόδους για να αντιμετωπίσει τα προβλήματα χειρισμού αβέβαιης συλλογιστικής. Η μέθοδος αυτή χρησιμοποιείται κυρίως σε θεωρητικά θέματα τεχνητής νοημοσύνης ενώ αποτελεί τη βάση λειτουργίας της Prolog. Οι πιο γνωστές μέθοδοι γραφής της είναι η προτασιακή ή δηλωτική και η κατηγορική λογική (propositional and predicate logic), που αποτελεί επέκταση της πρώτης.

Σενάρια (scripts)

Τα σενάρια είναι περιγραφές συγκεκριμένων αλυσιδωτών γεγονότων. Λόγω της ιεραρχημένης διαδοχής και της αλληλοσύνδεσης των γεγονότων είναι δυνατόν να προβλεφθούν γεγονότα ή να υποθεθούν ότι έχουν γίνει (εύλογη υπόθεση) αν στο σενάριο που έχει επιλεγεί έχουν συμβεί μετέπειτα γεγονότα. Από τη φύση τους τα σενάρια μπορούν να παρέχουν επεξηγήσεις για τα διάφορα γεγονότα. Τα σενάρια μπορούν να αναπαριστούν μόνο γεγονότα.

Πολλαπλή αναπαράσταση

Στη πραγματικότητα, σε πολλές περιπτώσεις δεν είναι συνιστώμενη ενέργεια η χρήση μίας μόνο τεχνικής αναπαράστασης αλλά η παράλληλη χρησιμοποίηση κατάλληλων τεχνικών. Κάθε τεχνική χρησιμοποιείται για την αναπαράσταση της γνώσης ενός τμήματος της ολικής γνώσης του

συστήματος. Στη περίπτωση της ταυτόχρονης χρήσης δύο ή περισσότερων μορφών αναπαράστασης παρουσιάζεται το πρόβλημα της μεταξύ τους συνεργασίας, καθώς και του χειρισμού της γνώσης του ενός από τα άλλα (Kunz 1984, Drake and Hess 1990).

Μια τέτοια υβριδική μορφή αναπαράστασης (hybrid representation) είναι η αναπαράσταση γνώσης με τη συνεργασία πλαισίων και κανόνων παραγωγής. Με τη συνεργασία αυτή γίνεται προσπάθεια εκμετάλλευσης των πλεονεκτημάτων των δύο αυτών μεθόδων (Thuraisingham, 1989).

Μη αθροιστική συλλογιστική (non monotonic reasoning)

Τα συστήματα που χρησιμοποιούν μη αθροιστική συλλογιστική μπορούν να αναθεωρούν την υπάρχουσα γνώση τους υπό το φως νέων δεδομένων (Reiter, 1978; 1980; Doyle 1979, 1979b McDermott and Doyle 1980, McCarthy 1980, Rich 1983, Reinfrank 1989, Rich and Knight 1991). Αντιθέτως τα συστήματα που χρησιμοποιούν αθροιστική συλλογιστική αποδέχονται τα νέα γεγονότα χωρίς να εξετάζουν την επίδρασή τους στην ήδη υπάρχουσα γνώση. Άλλου είδους μη αθροιστικά συστήματα είναι αυτά που στη προσπάθειά τους για επίλυση ενός προβλήματος προβαίνουν σε εύλογες υποθέσεις τις οποίες αναθεωρούν σε περίπτωση αποτυχίας.

3.2 Συλλογιστική υπό αβεβαιότητα

Στη πραγματικότητα, τις περισσότερες φορές, οι άνθρωποι χειρίζονται γεγονότα που χαρακτηρίζονται από αβεβαιότητα, σχετικότητα ή ελλιπή πληροφόρηση. Έτσι δημιουργήθηκε η ανάγκη για χρήση τεχνικών εξαγωγής συμπερασμάτων που θα επεξεργάζονται αβέβαια δεδομένα και γνώση. Η αβεβαιότητα στα συστήματα που βασίζονται στη χρήση κανόνων αναπαρίσταται με τη βοήθεια παραγόντων βεβαιότητας (certainty factors - CF).

IF	«ΜΑΡΤΥΡΙΑ»	
THEN	«ΥΠΟΘΕΣΗ»	CF (E,H)

όπου: E είναι η μαρτυρία (γεγονός, συνθήκη),
H είναι η υπόθεση (συμπέρασμα), και
CF(E, H) είναι ο συντελεστής βεβαιότητας του συγκεκριμένου κανόνα.

Δύο προβλήματα που πρέπει να επιλυθούν προκύπτουν (Cortes-Rello and Golshani, 1990):

- Πως θα μετρήσουμε το βαθμό ανακρίβειας και πως θα υπολογίσουμε τους συντελεστές βεβαιότητας για ανακριβείς καταστάσεις.
- Πως θα διαδώσουμε την αβεβαιότητα στη κατασκευή συμπερασμάτων ούτως ώστε να καταλήξουμε σε καλύτερα συμπεράσματα.

Αρκετές μέθοδοι έχουν προταθεί για τη διαχείριση της αβεβαιότητας, μεταξύ αυτών μερικές από τις περισσότερο χρησιμοποιούμενες τεχνικές είναι η ασαφής λογική με τη χρήση ασαφών συνόλων (Zadeh 1965), η Bayesian (Άγγλος στατιστικός του 18ου αιώνα) συλλογιστική με πιθανότητες (Duda 1979b, Andersen and Hooker 1994), η θεωρία της μαρτυρίας (Dempster 1967, Shafer 1976, 1981, 1986, 1987, Voorbraak 1991) και η τεχνική αξίας των κανόνων (Naylor 1983).

3.2.1 ΑΣΑΦΗΣ ΛΟΓΙΚΗ (FUZZY LOGIC)

Ένα ασαφής έμπειρο σύστημα είναι ένα έμπειρο σύστημα που χρησιμοποιεί μια συλλογή των ασαφής λειτουργιών και των κανόνων ιδιότητας μέλους, αντί της Boolean λογικής.

Η ασαφής λογική αποτελεί προσπάθεια αντιμετώπισης των δυσχερειών που είχε η δίτιμη λογική Boolean στην έκφραση των αβέβαιων γεγονότων και εννοιών. Έτσι, ενώ η λογική Boolean δέχεται μόνο δύο δυνατές τιμές για μια πρόταση (αληθής=1, μη αληθής=0), στην ασαφή λογική επιτρέπονται όλες οι τιμές μεταξύ 0 και 1. γίνεται, λοιπόν δυνατή η παράσταση της μερικής αλήθειας, όπως για παράδειγμα στην πρόταση:

«Πιστεύεται ότι κατά 70% αύριο θα βρέξει»

η οποία μπορεί να παρασταθεί σαν:

$$P(\text{Βροχή(αύριο)})=0.7$$

Η πρόταση αυτή, δηλαδή είναι αληθής κατά 70% και ψευδής κατά 30%. Έτσι σε ένα κανόνα της μορφής:

«IF (A AND B AND C) THEN D»

το συμπέρασμα εξάγεται με βεβαιότητα ίση με τη μικρότερη από τις βεβαιότητες των τριών γεγονότων A,B,C ($P_D = \min(P_A, P_B, P_C)$). Αντίστοιχα σε ένα κανόνα της μορφής:

«IF (A OR B OR C) THEN D»

το συμπέρασμα εξάγεται με βεβαιότητα ίση με τη μέγιστη των βεβαιοτήτων που αντιστοιχούν στα μεμονομένα γεγονότα A,B,C ($P_D = \max(P_A, P_B, P_C)$).

Το βασικό μειονέκτημα της θεωρίας αυτής συνιστάται στο γεγονός ότι αντιμετωπίζει καταρχήν όλα τα γεγονότα A,B,C σαν γεγονότα ίσης σπουδαιότητας. Κατά την εφαρμογή ενός κανόνα, η βεβαιότητα του συμπεράσματος εξαρτάται από το γεγονός, το οποίο έχει τη μικρότερη (περίπτωση(1)) ή τη μεγαλύτερη (περίπτωση(2)) βεβαιότητα, χωρίς να λαμβάνονται καθόλου υπόψη τα υπόλοιπα γεγονότα να είναι πιο σημαντικά

από άλλα. Έτσι, στον υπολογισμό της βεβαιότητας του συμπεράσματος, πρέπει να λαμβάνεται αναλογικά υπόψη η βαρύτητα του κάθε γεγονότος.

Η έκφραση της ασάφειας επιτυγχάνεται και με τη χρήση των ασαφών συνόλων (fuzzy sets), που εισήχθησαν από τον Zadeh (1965). Ένα ασαφές σύνολο A συμβολίζεται ως:

$$A = \{(a_1, P_{a1}), (a_2, P_{a2}), \dots\}$$

όπου a_1 είναι ένα γεγονός και P_{a1} είναι πιθανότητα να ανήκει το a_1 στο σύνολο A. Αν θεωρηθεί και ένα δεύτερο ασαφές σύνολο B:

$$B = \{(a_1, P_{b1}), (a_2, P_{b2}), \dots\}$$

ορίζονται οι πράξεις μεταξύ αυτών, ως εξής:

$$AB = \{(a_1, \min(P_{a1}, P_{b1})), (a_2, \min(P_{a2}, P_{b2})), \dots\}$$

$$AB = \{(a_1, \max(P_{a1}, P_{b1})), (a_2, \max(P_{a2}, P_{b2})), \dots\}$$

$$A^c = \{(a_1, (1 - P_{a1})), (a_2, (1 - P_{a2})), \dots\}$$

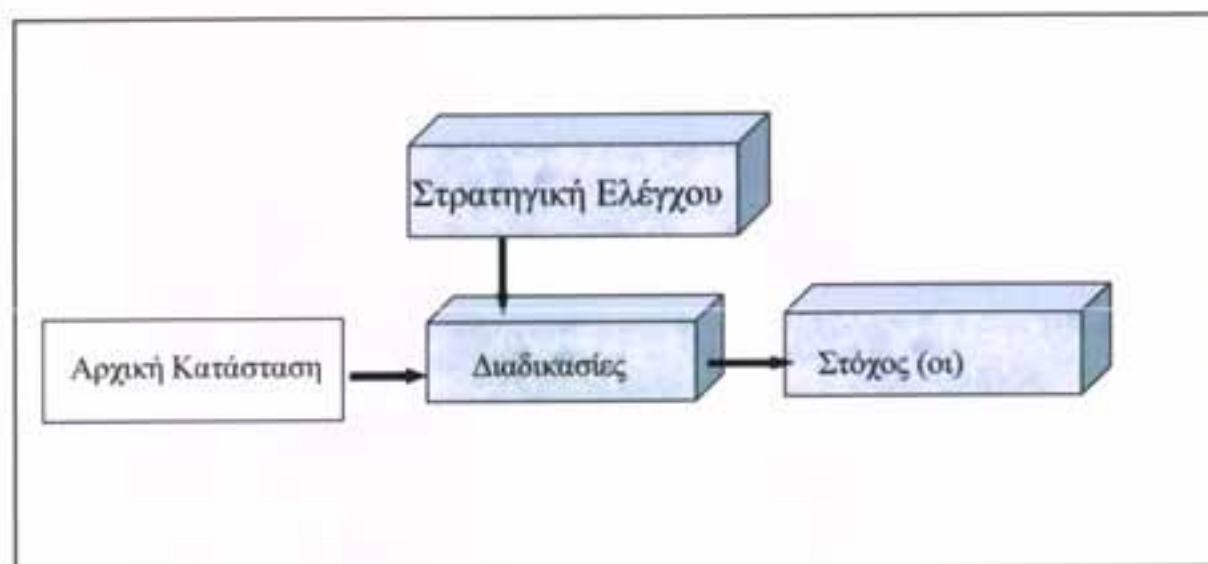
Μέχρι σήμερα, τα ασαφής έμπειρα συστήματα είναι η πιο κοινή χρήση της ασαφής λογικής.

Χρησιμοποιούνται σε διάφορους εκτεταμένους τομείς, που περιλαμβάνουν:

- Γραμμικός και μη γραμμικός έλεγχος
- Αναγνώριση σχεδίων
- Οικονομικά συστήματα
- Έρευνα λειτουργίας
- Ανάλυση στοιχείων

3.3 ΕΞΑΓΩΓΗ ΣΥΜΠΕΡΑΣΜΑΤΩΝ (inferencing)

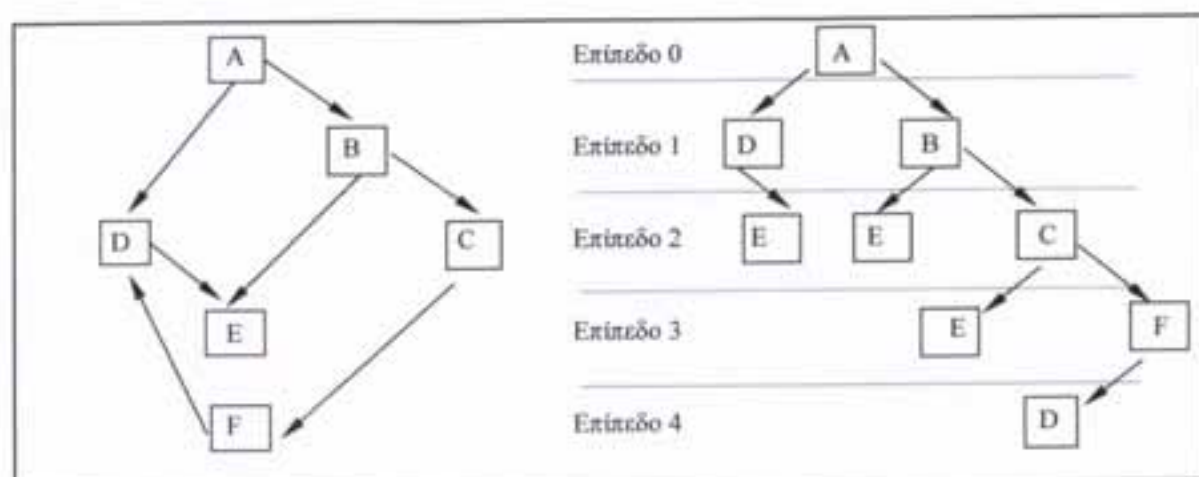
Η επίλυση ενός προβλήματος είναι στην πραγματικότητα η αναζήτηση μίας λύσης μέσα σε ένα πεδίο δυνατών καταστάσεων. Η διαδικασία επίλυσης ενός προβλήματος δίνεται στο σχήμα 3.7 (Frenzel 1987, Turban 1993).



Σχήμα 3.7 Διαδικασία επίλυσης

Στη προσπάθεια αναζήτησης ενός στόχου και σύμφωνα με τη στρατηγική ελέγχου, οι διαδικασίες επιδρούν στα δεδομένα της αρχικής κατάστασης του προβλήματος. Τόσο η στρατηγική ελέγχου όσο και η μέθοδος αναζήτησης εξαρτώνται σε μεγάλο βαθμό από τη χρησιμοποιούμενη μέθοδο αναπαράστασης της γνώσης.

Η γραφική αναπαράσταση του συνόλου των δυνατών λύσεων καθώς και των διασυνδέσεών τους μπορεί να γίνει είτε με τη βοήθεια γραφήματος (search graph) είτε με δένδρο αναζήτησης (search tree) (σχήμα 3.8) (Benchimol 1987, Turban 1993).



Σχήμα 3.8 Γράφημα και δένδρο αναζήτησης

Γράφημα: Στα γραφήματα υπάρχουν κόμβοι (nodes) και σύνδεσμοι (links) ή τόξα (arcs). Οι κόμβοι παριστάνουν τα διάφορα γεγονότα, τα οποία συνδέονται μεταξύ τους με συνδέσμους ή τόξα, τα οποία παριστάνουν ενέργειες.

Δένδρο: Τα δένδρα είναι πιο απλές απεικονίσεις από ότι τα γραφήματα τα οποία μπορούν να θεωρηθούν ότι έχουν πιο συμπυκνωμένη απεικόνιση. Στα δένδρα ένας κόμβος μπορεί να εμφανίζεται πολλές φορές σε αντίθεση με τα γραφήματα όπου εμφανίζεται μόνο μία φορά. Η ιεραρχική δομή του δένδρου υποδιαιρείται σε διάφορα επίπεδα. Το πλήθος των επιπέδων προσδιορίζουν το βάθος του δένδρου. Κάθε μία από τις διαδρομές ενός δένδρου παριστάνει μία διαφορετική λύση.

Ο εντοπισμός μιας λύσης μέσα από το σύνολο των δυνατών λύσεων είναι δύσκολη ενέργεια και γίνεται ακόμη δυσκολότερη όσο αυξάνει το βάθος του δένδρου. Προσοχή πρέπει να δίνεται στην αποφυγή δημιουργίας του φαινομένου της συνδυαστικής έκρηξης (combinatorial explosion), που δημιουργείται όταν προκύπτει ένα τεράστιο πλήθος εναλλακτικών διαδρομών. Το γεγονός αυτό καθιστά αδύνατη τη διερεύνηση όλων των δυνατών λύσεων για την ανεύρεση της βέλτιστης λύσης, οπότε η επίλυση

του προβλήματος καθίσταται αδύνατη. Για την ταχύτερη ανεύρεση της λύσης έχουν αναπτυχθεί διάφορες τεχνικές αναζήτησης, εκ των οποίων δύο είναι οι βασικές. Η τυφλή αναζήτηση (blind search) και η ευρετική (heuristic search). Οι μέθοδοι τυφλής αναζήτησης εξετάζουν τους κόμβους με τη σειρά, χωρίς να παίρνουν υπόψιν τους κάποια ειδική γνώση, όσον αφορά τη σειρά εξέτασης των κόμβων που θα έπρεπε να ακολουθήσουν ώστε να οδηγηθούν γρηγορότερα στη λύση. Αντίθετα κατά την ευρετική αναζήτηση χρησιμοποιείται εμπειρική γνώση για να επιλεγούν οι δρόμοι που πρέπει να ακολουθηθούν. Οι ευρετικές μέθοδοι οδηγούν την έρευνα από διαδρομές που είναι πιθανότερο να βρεθεί μία λύση. Οι μέθοδοι τυφλής έρευνας βρίσκουν τη καλύτερη λύση. Με τη χρήση ευρετικών μεθόδων, υπάρχει η περίπτωση να μη βρεθεί καμιά λύση ενώ αντιθέτως με τις μεθόδους τυφλής αναζήτησης πάντα βρίσκεται λύση.

3.4 MYCIN

Το MYCIN είναι ένα πρόγραμμα που αποτελείται από πολλές διαφορετικές ενότητες. Εντούτοις, υπάρχει ένα μέρος της δομής ελέγχου MYCIN που είναι παρόμοιο με ένα άλλο πρόγραμμα το STRIPS, δηλαδή το μέρος που εκτελεί μια σχεδόν-διαγνωστική λειτουργία. Αλλά οι στόχοι που επιτυγχάνονται δεν είναι φυσικοί στόχοι, που περιλαμβάνουν τη μετακίνηση των αντικειμένων στο διάστημα, αλλά τους στόχους συλλογισμού που περιλαμβάνουν την καθιέρωση των διαγνωστικών υποθέσεων. Αυτό το τμήμα συγκεντρώνεται επάνω στη διαγνωστική ενότητα MYCIN, που δίνει μια απλουστευμένη περιγραφή της συμπεριφοράς λειτουργίας, δομών και χρόνου εκτέλεσης.

Το MYCIN περιγράφεται συχνά ως διαγνωστικό πρόγραμμα, αλλά αυτό δεν είναι έτσι. Ο σκοπός του είναι να βοηθήσει έναν παθολόγο που δεν είναι εμπειρογνώμονας στον τομέα των αντιβιοτικών με τη θεραπεία των μολύνσεων αίματος. Με αυτές τις ενέργειες αναπτύσσει τις διαγνωστικές

υποθέσεις και τις ζυγίζει, αλλά δεν χρειάζεται να επιλέξει απαραίτητως μεταξύ τους. Η εργασία για το MYCIN άρχισε το 1972 ως συνεργασία μεταξύ των ιατρικών και κοινοτήτων T.N. στο πανεπιστήμιο του Στάνφορντ.

Έχουν υπάρξει διάφορες επεκτάσεις, αναθεωρήσεις και αφαιρέσεις του MYCIN από το 1976, αλλά η βασική έκδοση έχει πέντε συστατικά, όπως φαίνεται στο σχήμα 3.9, όπου τα βέλη παρουσιάζουν βασικό σχέδιο της ροής πληροφοριών μεταξύ των ενοτήτων.

(1) Μια **βάση γνώσεων** που περιέχει την πραγματική γνώση για την περιοχή.

(2) **Μια δυναμική βάση δεδομένων** που περιέχει τις πληροφορίες για μια ιδιαίτερη περίπτωση.

(3) Ένα **πρόγραμμα διαβουλεύσεων** που υποβάλλει τις ερωτήσεις, συνάγει τα συμπεράσματα, και δίνει τις συμβουλές για μια ιδιαίτερη περίπτωση βασισμένη στα στοιχεία ασθενών και τη στατική γνώση.

4) Ένα **πρόγραμμα εξήγησης** που απαντά στις ερωτήσεις και δικαιολογεί αυτές τις συμβουλές, χρησιμοποιώντας τη στατική γνώση και ένα ίχνος της εκτέλεσης του προγράμματος.

(5) Ένα **πρόγραμμα αποκτήσεων γνώσης** για τους νέους κανόνες.

Το σύστημα που αποτελείται από τα συστατικά (1), (3) είναι το μέρος επίλυσης προβλήματος MYCIN, που παράγει τις υποθέσεις όσον αφορά τους προσβάλλοντες οργανισμούς, και καθιστά τις συστάσεις θεραπείας βασισμένες στις υποθέσεις διατριβών.



Σχήμα 3.9 MYCIN

3.4.1 ΒΑΣΗ ΓΝΩΣΕΩΝ MYCIN

Η βάση γνώσεων **MYCIN** οργανώνεται γύρω από ένα σύνολο κανόνων της γενικής μορφής
 εάν ο όρος₁, και... και ο όρος_n κρατά
 κατόπιν συνάγετε το συμπέρασμα₁,
 και... και το συμπέρασμα_n
 κωδικοποιημένο ως δομές δεδομένων της γλώσσας προγραμματισμού LISP.

Το σχήμα 3.10 παρουσιάζει την αγγλική μετάφραση ενός χαρακτηριστικού κανόνα του **MYCIN** για την κατηγορία ενός οργανισμού. Αυτή η μετάφραση παρέχεται από το ίδιο το πρόγραμμα. Τέτοιοι κανόνες καλούνται ORG RULES και προσπαθούν να καλύψουν τέτοιους οργανισμούς όπως το στρεπτόκοκκο και τα εντεροβακτηρίδια.

Ο κανόνας λέει ότι εάν ένας απομονωμένος οργανισμός εμφανίζει ράβδο-διαμορφωμένο, μολύνει με έναν ορισμένο τρόπο, και αυξάνεται η παρουσία του οξυγόνου, κατόπιν είναι ιδιαίτερα πιθανό να είναι στην κατηγορία εντεροβακτηρίδια. Ο αριθμός 0,8 καλείται **έλεγχος** του κανόνα, ο οποίος λέει πόσο απαραίτητο το συμπέρασμα είναι, δεδομένου ότι οι όροι είναι ικανοποιημένοι. Η χρήση του ελέγχου εξηγείται κατωτέρω. Κάθε κανόνας αυτού του είδους μπορεί να θεωρηθεί ως κωδικοποίηση, ένα κομμάτι της ανθρώπινης γνώσης της οποίας η δυνατότητα εφαρμογής εξαρτάται μόνο από το πλαίσιο που καθιερώνεται από τους όρους του κανόνα.

Οι όροι ενός κανόνα μπορούν επίσης να ικανοποιηθούν με τους ποικίλους βαθμούς βεβαιότητας, έτσι η εισαγωγή τέτοιων κανόνων είναι κατά προσέγγιση η ακόλουθη:

*If condition₁ holds with certainty X₁ ... and condition_m holds with certainty X_m
then draw conclusion₁ with certainty Y₁ and ... and conclusion_n with certainty Y_n*

όπου η βεβαιότητα που συνδέεται με κάθε συμπέρασμα είναι μια λειτουργία των συνδυασμένων βεβαιοτήτων των όρων και του ελέγχου, ο οποίος προορίζεται να απεικονίσει το βαθμό εμπιστοσύνης μας στην εφαρμογή του κανόνα.

Εν περιλήψει, ένας κανόνας είναι ένα ζευγάρι εγκατάσταση-δράσης. Τέτοιοι κανόνες καλούνται μερικές φορές "παραγωγές". Είναι κλίσεις των όρων, και η βεβαιότητά τους είναι μια λειτουργία της βεβαιότητας αυτού του όρου.

Οι όροι είναι είτε προτάσεις που αξιολογούν την αλήθεια ή την αναλήθεια με κάποιο βαθμό βεβαιότητας, παραδείγματος χάριν:

«the organism is rod-shaped»

ή αποσυνδέσεις τέτοιων όρων. **Οι ενέργειες** είναι **συμπεράσματα** που συνάγονται με κάποιο κατάλληλο βαθμό βεβαιότητας, παραδείγματος χάριν η ταυτότητα κάποιου οργανισμού, **οδηγίες** που πραγματοποιούνται, παραδείγματος χάριν συντάσσοντας έναν κατάλογο θεραπείας.

IF	1) The stain of organism is gramneg, and 2) The morphology of the organism is rod, and 3) The aerobicity of the organism is aerobic
THEN	There is strongly suggestive evidence (0.8) that the class of the organism is enterobacteriaceae

Σχήμα 3.10

3.4.2 ΔΟΜΗ ΕΛΕΓΧΟΥ MYCIN

Η MYCIN έχει έναν κορυφαίο κανόνα στόχου που καθορίζει ολόκληρο το στόχο του συστήματος διαβουλεύσεων, το οποίο παραφράζεται πιο κάτω:

- EAN 1) υπάρχει ένας οργανισμός που απαιτεί τη θεραπεία, και
2) η προσοχή έχει δοθεί σε οποιουδήποτε άλλους οργανισμούς που απαιτούν τη θεραπεία

ΚΑΤΟΠΙΝ συντάξτε έναν κατάλογο πιθανής θεραπείας, και καθορίστε καλύτερη σε αυτόν τον κατάλογο.

Μια σύνοδος διαβουλεύσεων ακολουθεί μια απλή σε δύο στάδια διαδικασία:

- δημιουργήστε το υπομονετικό πλαίσιο ως κορυφαίο κόμβο στο δέντρο πλαισίου
- προσπάθεια να εφαρμοστεί ο κανόνας στόχου σε αυτό το υπομονετικό πλαίσιο.

Η εφαρμογή του κανόνα περιλαμβάνει την αξιολόγηση της προϋπόθεσής της, η οποία περιλαμβάνει να ανακαλύψει εάν υπάρχει πράγματι ένας οργανισμός που απαιτεί τη θεραπεία. Προκειμένου να ανακαλυφθεί αυτό, πρέπει πρώτα να ανακαλύψει εάν υπάρχει πράγματι ένας παρόν οργανισμός που συνδέεται με μια σημαντική ασθένεια. Αυτές οι πληροφορίες μπορούν είτε να ληφθούν από το χρήστη άμεσα, είτε μέσω κάποιας αλυσίδας του συμπεράσματος βασισμένης στα συμπτώματα και των εργαστηριακών στοιχείων που παρέχονται από το χρήστη.

Οι διαβουλεύσεις είναι ουσιαστικά μια αναζήτηση μέσω ενός **δέντρου των στόχων**. Ο κορυφαίος στόχος στη ρίζα του δέντρου είναι το μέρος δράσης του κανόνα στόχου, δηλαδή η σύσταση μιας θεραπείας φαρμάκων. Οι υπό-στόχοι πιο κάτω στο δέντρο περιλαμβάνουν τον καθορισμό του σχετικού οργανισμού και βλέπουμε εάν είναι σημαντικό. Πολλοί από αυτούς τους υπο-στόχους έχουν δικούς τους υπό-στόχους, όπως να ανακαλύψουν τις ιδιότητες μόλυνσεων και τη μορφολογία ενός οργανισμού. Τα φύλλα του δέντρου είναι οι στόχοι γεγονότος, όπως τα εργαστηριακά στοιχεία, τα οποία δεν μπορούν να συναχθούν.

Ένα ειδικό είδος δομής, αποκαλούμενο **AND/OR TREE**, είναι πολύ χρήσιμο για τον τρόπο με τον οποίο οι στόχοι μπορούν να επεκταθούν στους υπό-στόχους από ένα πρόγραμμα. Η βασική ιδέα είναι ότι ο κόμβος ρίζας του δέντρου αντιπροσωπεύει τον κύριο στόχο, οι τελικοί κόμβοι αντιπροσωπεύουν τις πρωτόγονες ενέργειες που μπορούν να ληφθούν, ενώ οι τερματικοί κόμβοι αντιπροσωπεύουν τους υπό-στόχους που είναι ευαίσθητοι στην περαιτέρω ανάλυση. Υπάρχει μια απλή αλληλογραφία μεταξύ αυτού του είδους ανάλυσης και της ανάλυσης των συνόλων κανόνα.

Εξετάστε το ακόλουθο σύνολο κανόνων όρος-δράσης:

If X has BADGE and X has GUN, then X is POLICE

If X has REVOLVER or X has PISTOL or X has RIFLE, then X has

GUN

If X has SHIELD, then X has BADGE

Μπορούμε να αντιπροσωπεύσουμε αυτόν τον κανόνα που τίθεται από την άποψη ενός δέντρου των στόχων, εφ' όσον διατηρούμε τη διάκριση μεταξύ των κλίσεων και των αποσυνδέσεων των υπό-στόχων. Κατά συνέπεια, στο σχήμα 3.11, σύρουμε ένα τόξο μεταξύ των συνδέσεων που συνδέουν τους κόμβους ΔΙΑΚΡΙΤΙΚΟ και ΠΥΡΟΒΟΛΟ ΌΠΛΟ με τον κόμβο ΑΣΤΥΝΟΜΙΑ, για να δηλώσουμε ότι και το ΔΙΑΚΡΙΤΙΚΟ και το ΠΥΡΟΒΟΛΟ ΌΠΛΟ πρέπει να είναι ικανοποιημένα για να ικανοποιήσουν τον στόχο ΑΣΤΥΝΟΜΙΑ. Εντούτοις, δεν υπάρχει κανένα τόξο μεταξύ των συνδέσεων που συνδέουν το ΠΕΡΙΣΤΡΟΦΟ και το ΠΙΣΤΟΛΙ και του ΤΟΥΦΕΚΙΟΥ με το ΠΥΡΟΒΟΛΟ ΌΠΛΟ, επειδή η ικανοποίηση καθενός από αυτά θα ικανοποιήσει το ΠΥΡΟΒΟΛΟ ΌΠΛΟ. Ένας κόμβος όπως το ΔΙΑΚΡΙΤΙΚΟ μπορεί να έχει ένα ενιαίο παιδί, ΑΣΠΙΔΑ, που δηλώνει ότι μια ασπίδα μετρά ως διακριτικό.

Το AND/OR tree στο σχήμα 3.11 μπορεί να θεωρηθεί ως τρόπος για να παρουσιάσουμε το **διάστημα αναζήτησης** για την ΑΣΤΥΝΟΜΙΑ, με την απαρίθμηση των τρόπων με τους οποίους οι διαφορετικοί χειριστές μπορούν να εφαρμοστούν προκειμένου να καθιερωθεί η ΑΣΤΥΝΟΜΙΑ ως αληθής.



Σχήμα 3.11 Αντιπροσωπεύει έναν κανόνα που τίθεται ως OR/AND tree

Η δομή ελέγχου MYCIN χρησιμοποιεί το AND/OR tree, και είναι αρκετά απλή καθώς τα προγράμματα T.N. προχωρούν. **Υπάρχουν** μόνο μερικές αποκλίσεις από τις εξαντλητικές μεθόδους αναζήτησης:

(1) Κάθε ίδρυση υπό-στόχου είναι πάντα μια γενικευμένη μορφή του αρχικού στόχου. Έτσι εάν ο υπό-στόχος πρόκειται να αποδείξει την πρόταση ότι η ταυτότητα του οργανισμού είναι E - **COLI**, **κατόπιν ο υπό-στόχος που ιδρύεται πραγματικά πρόκειται** να καθορίσει την ταυτότητα του οργανισμού. Αυτό αρχίζει μια εξαντλητική αναζήτηση σε ένα δεδομένο θέμα που συλλέγει όλα τα διαθέσιμα στοιχεία για τους οργανισμούς.

(2) Κάθε κανόνας σχετικός με το στόχο χρησιμοποιείται, εκτός αν ένας από αυτούς επιτευχθεί με βεβαιότητα. Εάν περισσότεροι από ένας κανόνες προτείνουν ένα συμπέρασμα για μια παράμετρο, όπως η φύση του οργανισμού, κατόπιν τα αποτελέσματά τους συνδυάζονται. Εάν τα στοιχεία για μια υπόθεση μειώνονται μεταξύ $-0,2$ και $+ 0,2$, θεωρούνται αναποτελεσματικά, και η απάντηση αντιμετωπίζεται ως άγνωστη.

(3) Εάν ο τρέχων υπό-στόχος είναι ένας κόμβος φύλλων, κατόπιν προσπαθήστε να ικανοποιήσετε το στόχο με την ερώτηση του χρήστη για τα στοιχεία. Αλλιώς ιδρύστε τον υπό-στόχο για το περαιτέρω συμπέρασμα, και πηγαίνετε στο (1).

Η επιλογή της θεραπείας πραγματοποιείται αφού έχει συνεχίσει αυτή η διαγνωστική διαδικασία την πορεία της. Αποτελείται από δύο φάσεις: επιλογή των φαρμάκων υποψηφίων, και έπειτα επιλογή προτεινόμενου φαρμάκου, ή του συνδυασμού φαρμάκων, από αυτόν τον κατάλογο.

Συνδυασμός στοιχείων

Στο MYCIN, δύο ή περισσότεροι κανόνες πρέπει να συναγάγουν τα συμπεράσματα για μια παράμετρο με τα διαφορετικά βάρη των στοιχείων. Κατά συνέπεια πρέπει ένας κανόνας να καταλήξει στο συμπέρασμα ότι ο οργανισμός είναι E. COLI με βεβαιότητα 0,8, ενώ άλλος να καταλήξει στο συμπέρασμα από άλλα στοιχεία ότι είναι E. COLI με μια βεβαιότητα του 0,5 ή 0,8. Στην περίπτωση μιας βεβαιότητας λιγότερο από μηδέν, τα στοιχεία είναι πραγματικά, ενάντια στην υπόθεση.

Αφήστε το X και το Y να είναι βάρη που προέρχονται από την εφαρμογή των διαφορετικών κανόνων. Το MYCIN συνδυάζει αυτά τα βάρη χρησιμοποιώντας τον ακόλουθο τύπο για να παραγάγει τον ενιαίο παράγοντα βεβαιότητας,

$$CF(X, Y) = \begin{cases} X+Y-XY & X, Y > 0 \\ X + Y + XY & X, Y < 0 \\ (X + Y) / (1 - \min(|X|, |Y|)) & \text{διαφορετικά} \end{cases}$$

όπου $|X|$ δείχνει την απόλυτη αξία του X.

Κάποιος μπορεί να δει τι συμβαίνει σε διαισθητική βάση. Εάν και τα δύο κομμάτια των στοιχείων επιβεβαιώνουν (ή ακυρώνουν) την υπόθεση, κατόπιν η εμπιστοσύνη στην υπόθεση ανεβαίνει (ή κατεβαίνει). Εάν τα δύο κομμάτια των στοιχείων είναι σε σύγκρουση, κατόπιν ο παρονομαστής επηρεάζει την επίδραση.

Αυτός ο τύπος μπορεί να εφαρμοστεί περισσότερο από μία φορά, εάν διάφοροι κανόνες σύρουν τη σχεδόν ίδια παράμετρο συμπερασμάτων. Είναι μεταλλακτικό, έτσι δεν πειράζει σε ποια διαταγή τα βάρη συνδυάζονται.

ΕΑΝ η ταυτότητα του οργανισμού είναι pseudomonas

ΚΑΤΟΠΙΝ συστήνω τη θεραπεία μεταξύ των ακόλουθων φαρμάκων:

- 1- (.98)
- 2- polymyxin (.96)
- 3- gentamicin (.96)
- 4- carbenicillin (.65)
- 5- sulfisoxazole (.64)

Σχήμα 3.12 MYCIN Κανόνας θεραπείας

Ο ειδικός κανόνας στόχου στην κορυφή AND/OR tree δεν οδηγεί σε ένα συμπέρασμα, αλλά υποκινεί τις ενέργειες, υποθέτοντας ότι οι όροι στην προϋπόθεση είναι ικανοποιημένοι. Σε αυτό το σημείο, οι κανόνες θεραπείας MYCIN για τις θεραπείες φαρμάκων που μπαίνουν παίζουν, περιέχουν τις πληροφορίες ευαισθησιών για τους διάφορους οργανισμούς που είναι γνωστοί στο σύστημα. Ένας κανόνας θεραπείας δειγμάτων δίνεται στο σχήμα 3.12

Οι αριθμοί που συνδέονται με κάθε φάρμακο είναι οι πιθανότητες ότι τα pseudomonas θα είναι ευαίσθητα στο υποδειγμένο φάρμακο σύμφωνα με τις ιατρικές στατιστικές. Το προτιμώμενο φάρμακο επιλέγεται από τον κατάλογο σύμφωνα με τα κριτήρια που προσπαθούν να καλύπτουν τις αντενδείξεις του φαρμάκου και ελαχιστοποιούν τον αριθμό φαρμάκων που αντιμετωπίζονται, εκτός από τη μεγιστοποίηση της ευαισθησίας. Ο χρήστης μπορεί να συνεχίσει την εναλλακτική θεραπεία έως ότου τρέχει το MYCIN από τις επιλογές, έτσι οι δηλώσεις του προγράμματος δεν είναι οριστικές.

3.5 ΑΞΙΟΛΟΓΗΣΗ ΚΑΙ ΣΥΓΚΡΙΣΗ ΤΩΝ ΕΜΠΕΙΡΩΝ ΣΥΣΤΗΜΑΤΩΝ

Υπάρχουν πολλοί τρόποι με τους οποίους κάποιος μπορεί να αξιολογήσει ή να συγκρίνει τα έμπειρα συστήματα, αλλά ο προφανέστερος είναι σε σύγκριση με έναν ανθρώπινο εμπειρογνώμονα. Αναπτύσσοντας το σύστημα, ο εμπειρογνώμονας και η γνώση μηχανικής, χαρακτηριστικά εργάζονται μαζί σε ένα σύνολο από κρίσιμα παράδειγματα έως ότου το πρόγραμμα να μπορεί να τα λύσει όλα. Η αξιολόγηση περιλαμβάνει έπειτα το δόσιμο του συστήματος 'απαρατήρητα' παραδείγματα και βλέποντας εάν οι κρίσεις της συμφωνούν με εκείνες του εμπειρογνώμονα.

3.5.1 ΑΞΙΟΛΟΓΗΣΗ - ΣΥΜΠΕΡΑΣΜΑΤΑ MYCIN

Από το 1974, μια αρχική μελέτη που χρησιμοποιεί την τρέχουσα έκδοση MYCIN παρήγαγε την ενθάρρυνση των αποτελεσμάτων. Μια επιτροπή πέντε εμπειρογνομώνων στη διάγνωση των μολυσματικών ασθενειών ενέκρινε 72% των συστάσεων MYCIN σχετικά με 15 πραγματικές περιπτώσεις της βακτηριακής μόλυνσης. Το **κύριο πρόβλημα δεν ήταν η ακρίβεια της διάγνωσης, αλλά μια έλλειψη κανόνων για τη δριμύτητα της ασθένειας.**

Το 1979, οι πιο επίσημες μελέτες ενός βελτιωμένου συστήματος έδειξαν ότι η απόδοση του MYCIN που συγκρίνεται ευνοϊκά με αυτήν των εμπειρογνώμωνων στους ασθενείς με τη βακτηραιμία και τη μηνιγγίτιδα. Τα τελικά συμπεράσματα του προγράμματος σχετικά με 10 πραγματικές περιπτώσεις συγκρίθηκαν με εκείνα των παθολόγων του Στάνφορντ, συμπεριλαμβανομένης της πραγματικής θεραπείας. Οκτώ άλλοι εμπειρογνώμονες κλήθηκαν έπειτα να εκτιμήσουν τις 10 συστάσεις θεραπείας σχετικά με κάθε μια από τις περιπτώσεις και να απονεύμουν ένα σημάδι από τα 80 για κάθε σύνολο συστάσεων, χωρίς γνώση ο οποίος προήλθε, ενδεχομένως, από έναν υπολογιστή. Τα αποτελέσματα παρουσιάζονται στο σχήμα 3.13.

Εκτιμήσεις από 8 εμπειρογνώμονες σε 10 περιπτώσεις Τέλαιο αποτέλεσμα = 80			
MYCIN	52	πραγματική θεραπεία	46
Ικανότητα-1	50	ικανότητα-4	44
Ικανότητα-2	48	κάτοικος	36
Συνεργάτης INF DIS	48	ικανότητα	34
Ικανότητα-3	46	σπουδαστής	24
Απαράδεκτη θεραπεία = 0 Ισοδύναμη ή αποδεκτή θεραπεία=1			

Σχήμα 3.13 Απόδοση MYCIN έναντι των ανθρώπινων εμπειρογνώμωνων

Οι διαφορές μεταξύ του αποτελέσματος MYCIN και εκείνων των εμπειρογνώμωνων του Στάνφορντ δεν ήταν σημαντικές, αλλά του αποτελέσμάτος της είναι τόσο καλές όσο οι εμπειρογνώμονες και καλύτερα από τους μη ειδικούς παθολόγους.

Εντούτοις, το MYCIN δεν χρησιμοποιήθηκε ποτέ στους θαλάμους νοσοκομείων για αριθμούς λόγων, που περιλαμβάνουν:

1. Η βάση γνώσεών της είναι ελλιπής, δεδομένου ότι είναι 400 ή οι κανόνες δεν καλύπτουν πολύ μεγάλη περιοχή των μολυσματικών ασθενειών

2. Λειτουργώντας θα είχε απαιτήσει περισσότερη δύναμη υπολογισμού από ότι τα περισσότερα νοσοκομεία θα μπορούσαν να αντέξουν οικονομικά εκείνη την περίοδο
3. Οι γιατροί δεν απολαμβάνουν τη δακτυλογράφηση στο τερματικό και απαιτούν ένα πολύ καλύτερο user interface από αυτό που παρέχεται από το MYCIN του 1976.

Το MYCIN ήταν ένας ερευνητικός φορέας, και επομένως δεν καθόρισε να επιτύχει την πρακτική εφαρμογή ή την εμπορική επιτυχία. Εντούτοις, οι απόγονοι MYCIN έχουν δει την εφαρμογή στα νοσοκομεία. Επίσης, η προσέγγιση επιτυχίας των υπό-στόχων δεν έχει περιοριστεί στις ιατρικές εφαρμογές.

Τα έμπειρα συστήματα, και οι σχετικές περιοχές και οι στόχοι τους, είναι αρκετά διαφορετικοί για να αντισταθούν στη γενίκευση σχετικά με τα κριτήρια για την επιτυχία. Εντούτοις, μπορούμε να προσδιορίσουμε διάφορες προϋποθέσεις που είναι απαραίτητες για την αξιολόγηση για να είναι σημαντικές (Hayes- Roth 1983).

- I. Πρέπει να υπάρξουν μερικά **αντικειμενικά κριτήρια** για την επιτυχία, διαφορετικά **δεν ξέρουμε ποτέ** εάν η απάντηση που επιστρέφεται από ένα σύστημα είναι ή όχι σωστή. **Σε κάποια περιοχή**, παραδείγματος χάριν οικονομική επένδυση, δεν μπορεί να υπάρξει κανένα κριτήριο εκτός από (i) τη συναινετική άποψη μιας ομάδας των ειδικών, ή (ii) πραγματοποιώντας τις συμβουλές του συστήματος και βλέποντας εάν αποδεικνύεται δεξιά από τα γεγονότα. Το πρόβλημα με τα πρώτα είναι ότι οι εμπειρογνώμονες μπορούν να διαφωνήσουν ακριβώς σε εκείνες τις περιπτώσεις όπου η άποψή τους θα ήταν η χρησιμότερη (δηλαδή οι δύσκολες περιπτώσεις) το πρόβλημα με τα τελευταία είναι ότι ο πειραματισμός στον πραγματικό κόσμο μπορεί να είναι ακριβής ή επιβλαβής στην υγεία και την ασφάλεια.

- Οι κατάλληλες **πειραματικές διαδικασίες** πρέπει να ακολουθηθούν. Από το να ζητήσει από τους εμπειρογνώμονες για να εκτιμηθεί μια εκτέλεση του προγράμματος, είναι καλύτερο (όπως στη μελέτη MYCIN που αναφέρεται ανωτέρω) να τρεχτεί ένα τυφλό «πείραμα», έτσι ώστε τα raters δεν ξέρουν ποιες λύσεις παράγονται από τον υπολογιστή και ποιες από τους ανθρώπους. Αυτό το σχέδιο έχει την ευεργετική επίδραση και από να αφαιρέσει διαγωνίως και τη μετάδοση μιας πραγματικής αίσθησης πόσο καλά ένα πρόγραμμα αποδίδει όσον αφορά τον ανθρώπινο ανταγωνισμό.

Η αξιολόγηση πρέπει να γίνει προσεκτικά ή καθόλου. Δεν υπάρχει κανένα σημείο στην προσπάθεια του με τον ανεπαρκή χρόνο ή τους πόρους. Μπορεί να πάρει περισσότερο για να εξετάσει λεπτομερώς ένα βασισμένο στη γνώση πρόγραμμα απ' το να το σχεδιάσει και να το εφαρμόσει, έτσι ένα ορισμένο επίπεδο υποχρέωσης είναι ουσιαστικό.

ΚΕΦΑΛΑΙΟ 4

ΑΝΑΠΤΥΞΗ ΕΜΠΕΙΡΩΝ ΣΥΣΤΗΜΑΤΩΝ

Η ανάπτυξη έμπειρων συστημάτων δεν είναι μια σταθερή και απόλυτη ακολουθία διαδοχικών ενεργειών που τηρούν όλοι οι κατασκευαστές. Κάθε έμπειρο σύστημα έχει τις δικές του ιδιομορφίες και κάθε ομάδα κατασκευαστών έχει τις δικές της πρακτικές που ακολουθεί. Πολλοί ερευνητές έχουν προτείνει κατά καιρούς διάφορες μεθοδολογίες ανάλυσης των συστημάτων (Davis 1974, Mandell 1979, Liebowitz 1984, Silverman 1984, 1985, 1987, DeSalvo 1987; Turban, 1993 κ.α). Στο σχήμα 4.1 δίνονται τα διαδοχικά στάδια ανάπτυξης ενός τυπικού Ε.Σ.

4.1 ΠΡΟΒΛΗΜΑΤΑ ΕΦΑΡΜΟΓΗΣ ΤΩΝ ΕΜΠΕΙΡΩΝ ΣΥΣΤΗΜΑΤΩΝ

Το πιο σημαντικό πρόβλημα στην ανάπτυξη ενός έμπειρου συστήματος είναι η απόκτηση της κατάλληλης γνώσης και η αναπαράστασή της με τη βοήθεια μιας γλώσσας κατά τέτοιο τρόπο ώστε να είναι δυνατή η χρήση της (Gale 1986, Curry and Moutinho 1991).

Η υποκειμενική φύση της λογικής και της εμπειρίας δημιουργούν σοβαρά εμπόδια στην ανάγκη τυποποιημένης αναπαράστασης της γνώσης. Αυτό είναι η αιτία της μικρής εφαρμογής των τεχνικών των Ε.Σ. στο μάρκετινγκ και τη στρατηγική. Το γεγονός αυτό αποτέλεσε μια νοητική πρόκληση για τους ερευνητές (Curry and Moutinho 1991). Σε όλα αυτά δεν θα πρέπει να λησμονιέται ότι ο χρήστης πρέπει να αντιλαμβάνεται τη λογική που χρησιμοποιεί ένα έμπειρο σύστημα για να καταλήξει σε κάποιο συμπέρασμα.

4.1.1 ΔΙΑΔΙΚΑΣΙΑ ΑΝΑΠΤΥΞΗΣ

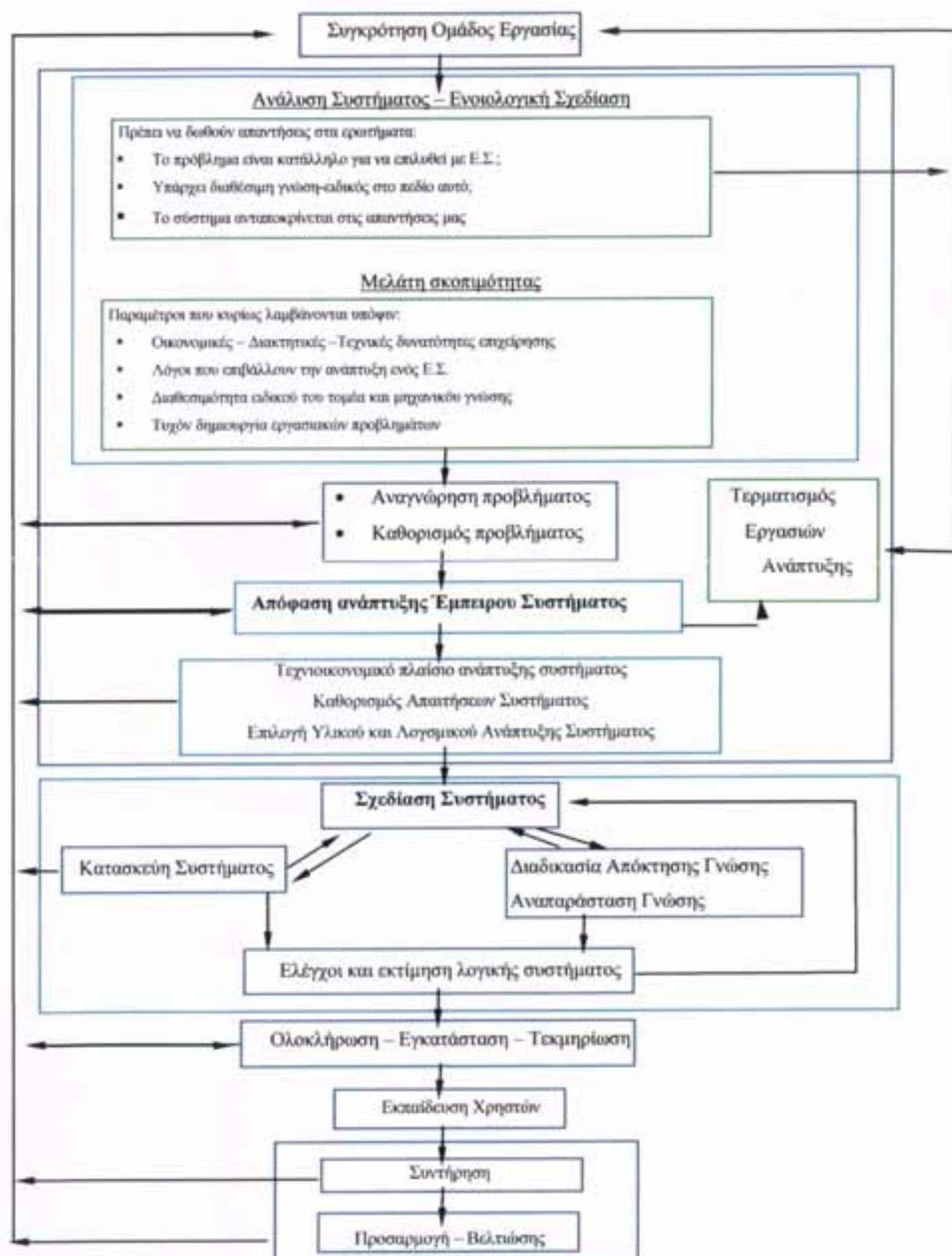
Στη περίπλοκη διαδικασία ανάπτυξης ενός έμπειρου συστήματος εμπλέκονται με τον ένα ή τον άλλο τρόπο πολλοί άνθρωποι διαφόρων ειδικοτήτων, απαιτείται η λήψη αποφάσεων τόσο σε θέματα οικονομικά, κοινωνικά όσο και τεχνικά.

Το πρώτο θέμα που πρέπει να αντιμετωπισθεί είναι η επιλογή, μέσα από το σύνολο των προβλημάτων της επιχείρησης, εκείνου του προβλήματος που θα έχει τη μέγιστη προτεραιότητα για την επίλυσή του. Τα στάδια με τα οποία θα γίνει η μελέτη για τη επιλογή του κατάλληλου προβλήματος είναι τα εξής:

- Απαραίτητες απαιτήσεις (necessary requirements): Συνολικά τίθενται επτά απαιτήσεις, οι οποίες θα πρέπει να πληρούνται όλες για να είναι δυνατή η ανάπτυξη ενός Ε.Σ.
- Αιτιολόγηση (justification): Αποτελείται από τέσσερις προϋποθέσεις, εκ των οποίων θα πρέπει τουλάχιστον η μία να ισχύει, για να μπορεί να αιτιολογηθεί η ανάπτυξη του Ε.Σ.
- Κατάλληλότητας (appropriateness): Κατά πόσο το πρόβλημα είναι κατάλληλο για να αναπτυχθεί σε Ε.Σ.

Στην όλη προσπάθεια να διαπιστωθεί αν ένα πρόβλημα είναι κατάλληλο να επιλυθεί μέσω της ανάπτυξής του σε Ε.Σ., έχουν αναπτυχθεί διάφορες τυποποιημένες ακολουθίες ερωτήσεων.

Στη συνέχεια παρατηρούμε τη διαδικασία ανάπτυξης του έμπειρου συστήματος από μια συγκεκριμένη ομάδα (στο σχήμα 4.1). Η ομάδα αυτή αποτελείται από άτομα της διοίκησης και από ειδικούς επιστήμονες (επιστήμονες πληροφορικής, οικονομολόγους, μηχανικούς γνώσης κ.α).



Σχήμα 4.1 Στάδια Ανάπτυξης Έμπειρων Συστημάτων

Στη συνέχεια, η ομάδα ανάπτυξης του συστήματος, προχωρεί στις εργασίες ανάλυσής του, εξετάζοντας τα ακόλουθα θέματα:

- Κατά πόσο το συγκεκριμένο πρόβλημα είναι κατάλληλο για να μπορεί να επιλυθεί ικανοποιητικά με τη χρήση έμπειρων συστημάτων.
- Αν υπάρχει διαθέσιμη γνώση στο πεδίο εφαρμογής του (ειδικός, βιβλιογραφία κ.λ.π).
- Αν το σύστημα ανταποκρίνεται και ικανοποιεί τις προδιαγραφές που έχουν τεθεί από την επιχείρηση.

Αν κάτι από τα παραπάνω δεν μπορεί να ικανοποιηθεί, τότε οι εργασίες ανάπτυξης του συστήματος διακόπτονται, είτε οριστικά, είτε γίνεται επανακαθορισμός του προβλήματος, είτε επιλέγεται άλλο πρόβλημα για επίλυση.

Στην όλη διαδικασία ανάπτυξης ενός Ε.Σ., τους κορυφαίους ρόλους παίζουν δύο παράγοντες, ο ειδικός του τομέα και ο μηχανικός γνώσης. Η έννοια της εργασίας του μηχανικού γνώσης μπορεί να οριστεί σαν η τέχνη της σύνδεσης των αρχών και των εργαλείων της έρευνας της τεχνητής νοημοσύνης, ώστε να προσφέρουν λύσεις σε προβλήματα δύσκολων εφαρμογών, για την επίλυση των οποίων απαιτούνται ειδικές γνώσεις (Feigenbaum and McCorduck, 1983). Επίσης τα τεχνικά θέματα της απόσπασης γνώσης, της αναπαράστασής της και της κατάλληλης χρήσης της στη κατασκευή και επεξήγηση της λογικής, είναι σημαντικά προβλήματα στη σχεδίαση συστημάτων βασιζόμενων στη γνώση, αποτελούν μέρος των ευθυνών του μηχανικού γνώσης. Για τη δουλειά του ο μηχανικός γνώσης χρειάζεται να έχει στη διάθεσή του γνώση, την οποία μπορεί να αποκτήσει από: ειδικούς, βιβλία, βάσεις δεδομένων, αναφορές και από διάφορες άλλες πηγές (Klein, 1990). Μερικά από τα προβλήματα, αν τυχόν προκύψουν, που θα πρέπει να επιλυθούν στο σημείο αυτό είναι:

- Ποιος επιλέγει τον ειδικό ή ακόμη πιο γενικά ποιος επιλέγει το ποιες πηγές γνώσης είναι κατάλληλες για τη συγκεκριμένη εφαρμογή.
- Τι κίνητρα θα δοθούν στον ειδικό για να επιτευχθεί η συνεργασία του.

- Ποια είναι τα χαρακτηριστικά που πρέπει να διαθέτει ο ειδικός (Goyal et al., 1985; Shanteau, 1986)
- Τι ενέργειες γίνονται όταν σε ένα πρόβλημα είναι απαραίτητη η παρουσία πολλών ειδικών (Alexander and Evans, 1988).
- Πώς εκτιμάται η ποιότητα της κρίσης του ειδικού και κάτω από ποιές συνθήκες αξιολογείται αυτή (Bolger and Wright, 1994)

Όταν έχει πλέον διευκρινισθεί η δυνατότητα κατασκευής του έμπειρου συστήματος για την επίλυση του προβλήματος, τότε γίνεται η εννοιολογική σχεδίαση και διενεργείται η μελέτη σκοπιμότητας της κατασκευής του.

Με την εννοιολογική σχεδίαση του συστήματος δίνεται μια γενική ιδέα για το σύστημα, για τους σκοπούς του και για τις αναμενόμενες ικανότητές του.

Μερικές από τις παραμέτρους που λαμβάνονται υπόψη κατά τη διαδικασία της μελέτης σκοπιμότητας δίνονται ακολούθως:

- Οι οικονομικές, διοικητικές και τεχνικές δυνατότητες της επιχείρησης, όπως:
 - i. Κόστος ανάπτυξης συστήματος.
 - ii. Αναμενόμενο όφελος της εταιρείας από την ανάπτυξη του Ε.Σ.
 - iii. Κόστος συντήρησης συστήματος
 - iv. Οργανωτικό σχήμα της επιχείρησης
 - v. Λειτουργία συστήματος σε δίκτυο
 - vi. Ασφάλεια λειτουργίας συστήματος
 - vii. Διαθεσιμότητα υλικού (hardware) και λογισμικού (software)
 - viii. Συμβατότητα διαθέσιμων συστημάτων
 - Η διαθεσιμότητα τόσο ειδικού του τομέα όσο και μηχανικού γνώσης.
 - Οι λόγοι που επιβάλλουν την ανάπτυξη ενός Ε.Σ. έναντι ενός άλλου.
 - Οι τυχόν επιπτώσεις, της λειτουργίας του Ε.Σ., στις εργασιακές σχέσεις.

- Ο χρονικός προγραμματισμός του έργου.
- Διάφοροι περιορισμοί (π.χ. νομικοί).

Οι εργασίες αυτές οδηγούν, αρχικά στην αναγνώριση και τον καθορισμό του προβλήματος, ενώ ακολούθως, είτε στην απόφαση κατασκευής του είτε στη διακοπή των εργασιών του.

Στη συνέχεια καθορίζονται τόσο το τεχνικοοικονομικό πλαίσιο της ανάπτυξης του συστήματος όσο και οι απαιτήσεις, που θέτει η επιχείρηση για τη λειτουργία του.

Οι αποφάσεις που απομένει να ληφθούν αφορούν:

- Την επιλογή και τη προμήθεια του υλικού (hardware) στο οποίο θα εγκατασταθεί και θα λειτουργεί τελικά το Ε.Σ.
- Την επιλογή και την προμήθεια των κατάλληλων εργαλείων λογισμικού (software tools) για την ανάπτυξη του Ε.Σ.

Ακολουθεί η αναλυτική σχεδίαση του συστήματος στην οποία εκτός από τους ειδικούς επιστήμονες της πληροφορικής συμμετέχει και ο μηχανικός της γνώσης ο οποίος συνεργάζεται με τον ειδικό του γνωστικού θέματος του Ε.Σ. Αρχικά αναπτύσσεται ένα μικρής κλίμακας σύστημα (prototype) στο οποίο περιέχονται τα βασικά μέρη του Ε.Σ., και ένα μικρό αλλά αντιπροσωπευτικό τμήμα της αποσπασθείσης γνώσης (Redin, 1987; Cholawsky, 1988). Μερικά από τα οφέλη της εργασίας αυτής αφορούν:

- τον έλεγχο της ορθής λειτουργίας των διαφόρων τμημάτων του,
- τον έλεγχο και την εκτίμηση των βάσεων γνώσης του, και
- την επίδειξή του στους ενδιαφερόμενους ώστε να κάνουν τις παρατηρήσεις τους.

Σε περίπτωση παρουσίασης κάποιου προβλήματος, επαναλαμβάνονται κάποιες από τις προηγούμενες εργασίες, αφού

προηγουμένως γίνουν οι απαραίτητες διορθώσεις. Όταν οι έλεγχοι έχουν θετική κατάληξη, τότε ολοκληρώνεται η ανάπτυξη του συστήματος.

Η ολοκλήρωση μπορεί να θεωρηθεί σαν μια διαδικασία, που εξελίσσεται από τα αρχικά στάδια ανάπτυξης του Ε.Σ. μέχρι την εγκατάστασή του. Το ολοκληρωμένο σύστημα ελέγχεται και εκτιμάται η επιτυχία του τόσο στο χώρο ανάπτυξής του, όσο και στο χώρο εργασίας του. Κατά την ολοκλήρωση πληροφοριακών συστημάτων μπορεί να προκύψουν πολλά προβλήματα, η γνώση των οποίων βοηθά στην αποφυγή τους. Στη τελική του μορφή, γίνεται η εγκατάστασή του στο χώρο εργασίας του ενώ παράλληλα ολοκληρώνεται η τεκμηρίωσή του (documentation).

Μεγάλη βαρύτητα πρέπει να δοθεί στη σωστή και πλήρη εκπαίδευση των χρηστών. Χρήστες που δεν γνωρίζουν καλά τις δυνατότητες και τη λειτουργία του συστήματος, τοποθετούνται αρνητικά απέναντί του, γεγονός που μπορεί να οδηγήσει στη σταδιακή αδρανοποίησή του. Οι επιπτώσεις μιας τέτοιας ενέργειας είναι καταστροφικές για τη περαιτέρω ανάπτυξη νέων έμπειρων συστημάτων, τόσο στην ίδια επιχείρηση, όσο και σε άλλες που έχουν σχέσεις με αυτήν. Αυτό που μένει από μια τέτοια ενέργεια δεν είναι ότι φταίνε οι χρήστες αλλά ότι φταίει το σύστημα.

Κάθε σύστημα πρέπει να διαθέτει ευελιξία (flexibility) και προσαρμοστικότητα (adaptability) σε τυχόν νέες απαιτήσεις που θα παρουσιαστούν στο μέλλον. Ένα σύστημα είναι ένας ζωντανός οργανισμός, που πρέπει να έχει την ικανότητα να προσαρμόζεται κατά τον καλύτερο δυνατό τρόπο στις ανάγκες των χρηστών του. Στην ουσία τα τελευταία στάδια της συντήρησης, προσαρμογής και βελτίωσης δεν τελειώνουν παρά μόνο με το τερματισμό της λειτουργίας του Ε.Σ.

4.1.2 ΕΠΙΣΚΟΠΗΣΗ ΤΩΝ ΕΡΓΑΛΕΙΩΝ ΕΜΠΕΙΡΩΝ ΣΥΣΤΗΜΑΤΩΝ

Όλα τα εργαλεία έμπειρων συστημάτων σχεδιάζονται για να υποστηρίξουν τη διαμόρφωση πρωτοτύπου. Στο ιδίωμα της τεχνολογίας λογισμικού (Schach, 1993), ένα πρωτότυπο είναι:

‘ένα λειτουργώντας πρότυπο που είναι λειτουργικά ισοδύναμο με ένα υποσύνολο του προϊόντος’

Η ιδέα είναι να αναπτυχθεί, νωρίς στο πρόγραμμα, μια απόδειξη της έννοιας που μπορεί να είναι από τους εμπειρογνώμονες ή τους χρήστες και που λύνει κάποιο μη ουσιώδη μέρος του προβλήματος.

Αυτό γίνεται συνήθως ως μέσο απαιτήσεων συστημάτων καθαρισμού για να πειστεί ότι το πρόβλημα είναι πραγματικό προτού να ξεοδευτούν τα ουσιαστικά χρηματικά ποσά. Το αρχικό πρωτότυπο ακολουθεί μια περίοδο ανάπτυξης, την οποία η βασική προσέγγιση που επικυρώνεται από το πρωτότυπο, και συμπληρώνεται έτσι ώστε να λύνει ολόκληρο το πρόβλημα. Αυτό γίνεται κανονικά σε μια διαδοχή των κατασκευών που ενσωματώνουν την πρόσθετη λειτουργία στο πρόγραμμα, κυκλικά. Αυτές οι διαδοχικές κατασκευές μπορούν ή όχι να αποτελέσουν τη βάση των πραγματικών απελευθερώσεων στον πελάτη, ανάλογα με το πόσο αποδεκτό ένα μερικό σύστημα είναι στους τελικούς χρήστες.

Κατά συνέπεια η ανάπτυξη έμπειρων συστημάτων είναι συχνά ένα μίγμα της γρήγορης διαμόρφωσης πρωτοτύπου και των επαυξητικών προτύπων της τεχνολογίας λογισμικού, παρά των «water fall» πρότυπο, στο οποίο οι απαιτήσεις οδηγούν στις προδιαγραφές, στον προγραμματισμό, στο σχέδιο, την εφαρμογή, και την ολοκλήρωση, με τα συστήματα ανατροφοδότησης πληροφοριών μεταξύ των παρακείμενων σταδίων. Το γεγονός ότι ένα πρωτότυπο έχει χτιστεί και έχει εγκριθεί

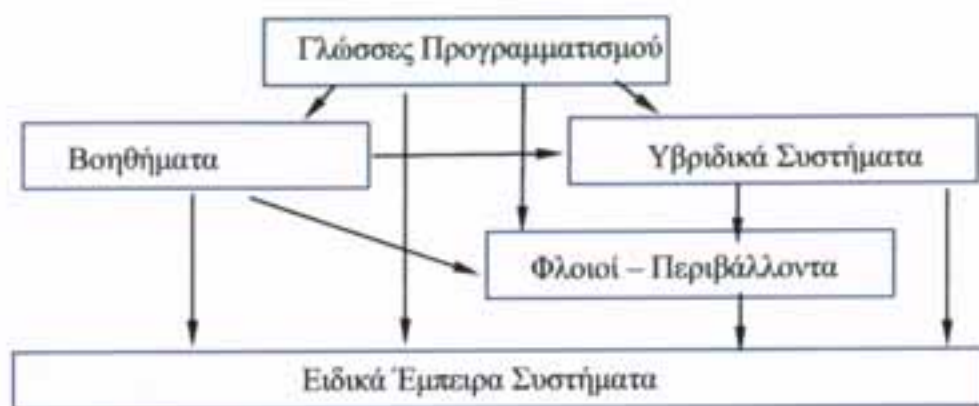
γενικά μειώνει το ποσό επανασχεδιασμού, κ.τ.λ. Ένα μειονέκτημα του επαυξητικού προτύπου στα συμβατικότερα παραδείγματα προγραμματισμού είναι το πρόβλημα της νέας λειτουργίας με την προηγούμενη έκδοση. Τα περιβάλλοντα ανάπτυξης έμπειρων συστημάτων στοχεύουν να λύσουν αυτό το πρόβλημα με τη χρησιμοποίηση των μορφωματικών αντιπροσωπεύσεων της γνώσης του είδους. Τέτοια εργαλεία μπορούν να ταξινομηθούν σύμφωνα με τη φύση και την ποικιλία των αντιπροσωπευτικών σχεδίων που υποστηρίζουν, και των ειδών άλλων δομικών μονάδων που παρέχουν.

4.1.2.1 ΕΡΓΑΛΕΙΑ ΑΝΑΠΤΥΞΗΣ

Το λογισμικό που χρησιμοποιείται για την ανάπτυξη των Ε.Σ. αποτελείται από (Benchimol et al., 1987; Price, 1990; Turban, 1993):

- Συμβατικές γλώσσες (π.χ. C, Fortran, Pascal, Basic) και γλώσσες τεχνητής νοημοσύνης (π.χ. Lisp, Prolog)
- Φλοιούς και περιβάλλοντα (π.χ. 1st CLASS, ESE, Nexpert, Level5, ART, Level5 Object)
- Βοηθήματα με τα οποία κατασκευάζονται τα διάφορα τμήματα ενός ΕΣ, εκτός από τη βάση γνώσης, που μπορεί να κατασκευασθεί και με τη βοήθεια φλοιών.
- Υβριδικά συστήματα, που αποτελούνται από βοηθήματα και γλώσσες προγραμματισμού. Με αυτά παρέχεται η δυνατότητα ταχύτερης ανάπτυξης φλοιών ή ειδικών Ε.Σ. (π.χ. ART, KEE, Nexpert Object).
- Ειδικά έμπειρα συστήματα που απευθύνονται σε εξειδικευμένα θέματα.

Με τη βοήθεια των εργαλείων μπορούν να αναπτυχθούν σκελετοί (φλοιοί) και ειδικά Ε.Σ. Για την ανάπτυξη των ειδικών Ε.Σ. μπορούν να χρησιμοποιηθούν είτε εργαλεία είτε σκελετοί, για την ανάπτυξη των οποίων χρησιμοποιούνται εργαλεία (σχήμα 4.2).



Σχήμα 4.2 Εργαλεία ανάπτυξης Έμπειρων Συστημάτων

4.2 ΚΕΛΥΦΟΣ ΕΜΠΕΙΡΩΝ ΣΥΣΤΗΜΑΤΩΝ (EXPERT SYSTEM SHELL)

Δεδομένου ότι τα έμπειρα συστήματα τρέχουν στους υπολογιστές με την ικανότητα της διαδικαστικής βαθμιαίας επεξεργασίας, χρειαζόμαστε κάποιο είδος προγράμματος για να τρέξει το έμπειρο σύστημά. Αυτά τα προγράμματα καλούνται κέλυφος. Τα κέλυφη έμπειρων συστημάτων παρέχουν ένα στρώμα της αφαίρεσης μεταξύ του ίδιου υπολογιστή και του έμπειρου συστήματος. Μεταξύ των πρώτων ήταν το EMYCIN, το οποίο παρείχε τον κανόνα παραγωγής διερμηνέα του MYCIN, μαζί με όλες τις δομές βοηθητικών δεδομένων, όπως οι πίνακες γνώσης, και συντάσσοντας ευρετήριο με τους σχετικούς μηχανισμούς τους.

Ενώ οποιαδήποτε συμβατική γλώσσα προγραμματισμού μπορεί να χρησιμοποιηθεί για να χτίσει μια βάση γνώσεων, το κέλυφος έμπειρων συστημάτων απλοποιεί τη διαδικασία της βάσης γνώσεων. Το κέλυφος επεξεργάζεται πραγματικά τις πληροφορίες που εισάγονται από έναν χρήστη όσον αφορά τις έννοιες που περιλαμβάνονται στη βάση γνώσεων και παρέχει μια αξιολόγηση ή μια λύση για ένα ιδιαίτερο πρόβλημα. Κατά συνέπεια ένα κέλυφος έμπειρων συστημάτων παρέχει ένα στρώμα μεταξύ του user interface και του operating system (λειτουργικού συστήματος) υπολογιστών για να διαχειριστεί την εισαγωγή και την παραγωγή των

στοιχείων. Χειρίζεται επίσης τις πληροφορίες που παρέχονται από το χρήστη από κοινού με τη βάση γνώσεων για να φθάσουν σε ένα ιδιαίτερο συμπέρασμα. Το κέλυφος διαχειρίζεται επίσης το user interface, εκτελώντας τις λειτουργίες που κυμαίνονται με σειρά.

4.2.1 ΤΑΙΡΙΑΖΟΝΤΑΣ ΤΑ ΚΕΛΥΦΗ ΣΤΟΥ ΣΤΟΧΟΥΣ

Είναι σαφές ότι όλα τα κέλυφα δεν ταιριάζουν σε όλους τους στόχους. Ο Van Melle ήταν μεταξύ των πρώτων που επισήμαναν ότι το EMYCIN δεν ήταν μια γενικής χρήσης αρχιτεκτονική επίλυσης προβλήματος. Πρότεινε ότι το EMYCIN ήταν κατάλληλο για παραγωγικές προσεγγίσεις στα διαγνωστικά προβλήματα, όπου τα μεγάλα ποσά στοιχείων είναι διαθέσιμα και είναι δυνατόν να απαριθμηθεί το διάστημα λύσης των διαγνωστικών κατηγοριών. Ο Clancey έχει μεταγλωττίσει από τότε αυτό το πρόβλημα της «ευρετικής ταξινόμησης». Η προσέγγιση εμφανίστηκε να ταιριάζει λιγότερο σε αυτό του Van Melle που το κάλεσε «προβλήματα σχηματισμού» που περιλαμβάνουν τη συναρμολόγηση ενός σύνθετου συνόλου που πρέπει να ικανοποιήσει κάποιο σύνολο περιορισμών.

Είναι δύσκολο να ξεχωρίσεις ποιο κέλυφος πρέπει να χρησιμοποιηθεί για ποιο πρόβλημα. Αυτό συμβαίνει επειδή δεν υπάρχουν σαφείς ιδέες σχετικά με το πώς η ευρεία σειρά των στόχων των έμπειρων συστημάτων πρέπει να ταξινομηθούν. Μπορείς να θεωρήσεις ότι οι διακρίσεις όπως αυτή μεταξύ των διαγνωστικών και των προβλημάτων σχηματισμού δεν είναι αρκετά σωστές. Πολλά προβλήματα μπορούν να λυθούν με διαφορετικούς τρόπους, η προσέγγιση στη διάγνωση μοιράζεται σε πολλά χαρακτηριστικά γνωρίσματα της εποικοδομητικής επίλυσης προβλήματος, και τα δυσκολότερα προβλήματα απαιτούν συχνά μια μικτή προσέγγιση για τη λύση τους.

Όσον αφορά τα κέλυφη, η πλειοψηφία των εμπορικών προϊόντων παρείχε αρχικά στον χρήστη τις εγκαταστάσεις που είναι μόνο επαρκείς για τα μικρά διαστήματα αναζήτησης. Συγχρόνως τα κέλυφη όπως το M.4 θεωρούνται ότι ισχύουν σε ένα ευρύτερο φάσμα των προβλημάτων, επειδή υποστηρίζουν ένα ευρύτερο φάσμα των χαρακτηριστικώ γνωρισμάτων για την αντιπροσώπευση και τον έλεγχο.

Ένα κέλυφος όπως M.4 έρχεται πλήρης με ένα ιδιαίτερο φορμαλισμό όπως οι παράγοντες βεβαιότητας για την εκτέλεση του ανακριβός συλλογισμού. Εντούτοις οι περισσότεροι, εάν όχι όλοι οι φορμαλισμοί που υιοθετούνται από αυτά τα κέλυφη είναι είτε ασυμβίβαστοι με την θεωρία πιθανότητας είτε έχουν τις ιδιότητες που είναι πιο δύσκολο να αναλυθούν. Τα σύγχρονα κέλυφη είναι πιο εύκαπτα από τις προηγούμενες προσφορές δεδομένου ότι είναι καλύτερα ενσωματωμένα στα εμπορικά υπολογιστικά περιβάλλοντα και έχουν το ανώτερο user interface.

Άλλες γλώσσες προγραμματισμού όπως η Lisp και η Prolog δόθηκαν για να βελτιώσουν την αναγνωσιμότητα. Οι γλώσσες αυτές παρέχουν τη μέγιστη ευελιξία και αποτελούν τη βάση σε πολλά από τα άλλα κελύφη. Τα περιπλοκότερα κελύφη παρέχουν ένα πλαίσιο για το πρόβλημά, που κάνει την μηχανική γνώσης ένα θέμα δημιουργίας των κανόνων, τους όρους, και στις εκβάσεις.

4.3 INTELLIGENT AGENTS

Η συλλογή, η επιλογή, η εκτίμηση και χρήση πληροφοριών για την υποστήριξη των αποφασιζόντων στη λήψη αποφάσεων καθώς και ο συντονισμός της απόκτησης της πληροφόρησης και των προσπαθειών επίλυσης προβλημάτων των πόρων πληροφόρησης και των συστημάτων λήψης αποφάσεων, γίνεται μία όλο και περισσότερο κρίσιμη διαδικασία (Sycara and Zeng, 1996). Στις απαιτήσεις αυτές ανταποκρίνονται οι intelligent software agents (Rao and Georgeff, 1993; Sycara and Zeng, 1994; Wooldridge and Jennings, 1995).

Με την έννοια agent θεωρούμε μια οντότητα που δρα εκ μέρους κάποιου άλλου για να επιτύχει τους στόχους που του έχουν τεθεί. Σαν intelligent software agents θεωρούνται προγράμματα που δρουν εκ μέρους των χρηστών τους, συγκεντρώνοντας και φιλτράροντας πληροφορίες online από διάφορες πηγές, με στόχο την ικανοποίηση των αναγκών του. Μερικοί έχουν χαρακτηρίσει τους intelligent agents σαν «Objects that Think». Αυτά τα σκεπτόμενα «objects» στη προσπάθειά τους να ικανοποιήσουν τις απαιτήσεις του χρήστη τους αλληλεπιδρούν με αυτόν, με άλλα «objects» και με τους διαθέσιμους πόρους. Οι agents όμως διαθέτουν επιπλέον από τα 'objects' beliefs, commitments and goals. Οι agents μπορούν να εργάζονται είτε μόνοι τους είτε σε συνεργασία με άλλους σε μια cooperative society, επικοινωνώντας μεταξύ τους με ανταλλαγή μηνυμάτων.

Μερικοί από τους ορισμούς που κατά καιρούς έχουν δοθεί είναι οι παρακάτω. Ένας agent είναι ένα λογισμικό που μπορεί να κάνει πράγματα που πιθανότατα θα έκανε ο κάθε χρήστης, αν είχε το χρόνο (Janca 1995).

Οι intelligent agents ορίζονται σαν οι λειτουργικές μονάδες ενός multiagent system, το οποίο ορίζεται σαν ένα χαλαρά συνδεδεμένο σύστημα λυτών προβλημάτων, οι οποίοι συνεργάζονται για να λύσουν

προβλήματα τα οποία είναι πέρα από τις δυνατότητες ενός μόνο λύτη (Durfee 1989). Αυτοί οι λύτες είναι οι intelligent agents.

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

Σε ένα ελάχιστο επίπεδο ευφυΐας μπορεί να υπάρχουν μερικές εκφράσεις προτιμήσεων πιθανόν με τη μορφή κανόνων, εφοδιασμένο με ένα μηχανισμό εξαγωγής συμπερασμάτων ή κάποιο άλλο μηχανισμό συλλογιστικής που θα επιδρά σε αυτές. Σε ένα επόμενο επίπεδο περιλαμβάνεται ένα μοντέλο user ή κάποιας άλλης μορφής συλλογιστικής, γύρω από το τι ένας χρήστης επιθυμεί να γίνει και προγραμματισμού των μέσων έτσι ώστε να επιτευχθούν οι στόχοι. Επίσης, υπάρχουν συστήματα με δυνατότητες μάθησης, προσαρμοζόμενα στο περιβάλλον λειτουργίας τους ανάλογα με τους διαθέσιμους πόρους και τους τιθέμενους στόχους των χρηστών. Τα συστήματα αυτά προσομοιάζουν με βοηθούς του ανθρώπου χρήστη μια και μπορούν ανεξάρτητα από αυτόν να διερευνούν νέες σχέσεις, συνδέσμους ή έννοιες με στόχο να καλύψουν τις δικές του ανάγκες (Björn 1996).

Οι Intelligent agent, οι οποίοι χαρακτηρίζονται σαν αυτόνομες τεχνητές οντότητες (λογισμικό-προγράμματα) που χρησιμοποιούνται για την επίλυση συγκεκριμένων προβλημάτων, παρουσιάζουν τις παρακάτω ιδιότητες:

- *Αυτονομία (Autonomy)*: Επιλύουν το πρόβλημα που τους ανατίθεται χωρίς να είναι αναγκαία η παρέμβαση του χρήστη ή άλλων

προγραμμάτων και έχουν τον αποκλειστικό έλεγχο τόσο των ενεργειών που επιτελούν όσο και της εσωτερικής τους κατάστασης. Οι ευφυείς πράκτορες αναλαμβάνουν δράση όποτε το κρίνουν αυτοί σκόπιμο κι όχι απαραίτητα όποτε έχουν άμεση εντολή εκτέλεσης. Έχουν την δυνατότητα να ελέγχουν την κατάσταση του περιβάλλοντος τους, καθώς επίσης και να παρατηρούν την αλληλεπίδραση του χρήστη με το σύστημα, προκειμένου να κρίνουν αν υπάρχει λόγος να ξεκινήσουν κάποια διεργασία.

- *Κοινωνικότητα (Social ability)*: Αλληλεπιδρούν μεταξύ τους ή με ανθρώπους, όταν το θεωρούν απαραίτητο, με σκοπό να ολοκληρώσουν την επίλυση του προβλήματος ή να βοηθήσουν άλλες οντότητες (προγράμματα ή χρήστες) με τις ενέργειές τους. Η επικοινωνία αυτή γίνεται κυρίως μέσω ειδικών γλωσσών, όπως η KQML. Η κοινωνικότητά τους είναι πολύ σημαντική γιατί αφενός επιτρέπει την μεταξύ πρακτόρων υποστήριξη και βοήθεια και αφετέρου τους επιτρέπει να μαθαίνουν από την εμπειρία των άλλων.
- *Προσαρμοστικότητα (adaptivity)*: Πολύ σημαντικό είναι για ένα πράκτορα να μπορεί να προσαρμόζει τον τρόπο δράσης, μάθησης και συμπεριφοράς του τόσο στις διάφορες πλατφόρμες υλικού και λογισμικού, στις οποίες θα κληθεί να δουλέψει, όσο και στις απαιτήσεις και τα ιδιαίτερα χαρακτηριστικά του χρήστη του.
- *Πρωτοβουλία (proactiveness)*: Παίρνουν πρωτοβουλίες όταν αυτό κρίνεται απαραίτητο. Πέρα από την συμπεριφορά ανταπόκρισης σε εξωτερικά ερεθίσματα, οι πράκτορες είναι ικανοί να προσανατολίζονται στην επίτευξη στόχων και να προετοιμάζουν το έδαφος προς αυτή την κατεύθυνση, κρίνοντας και με βάση το περιβάλλον τους. Έτσι για παράδειγμα όταν ένας πράκτορας εντοπίσει τον χρήστη του να ψάχνει για κάποιο λογισμικό στο Δίκτυο, το οποίο προβλέπεται να χρειαστεί αρκετό χώρο στο δίσκο, φροντίζει από νωρίς για την απομνηματοποίηση του δίσκου.

- *Αντιδραστικότητα (reactivity)*: Οι πράκτορες αντιλαμβάνονται το περιβάλλον τους (το οποίο μπορεί να είναι ο φυσικός κόσμος, ένας χρήστης χρήσης, μιας ομάδος άλλων πρακτόρων, το internet καθώς και διάφοροι συνδυασμοί τους) και ανταποκρίνονται στις αλλαγές του μέσα σε λογικό χρόνο. Αυτό μπορεί να σημαίνει ότι οι πράκτορες βρίσκονται σε κατάσταση αναμονής περιμένοντας κάποιο κίνητρο δραστηριοποίησης (άφιξη ενός ηλεκτρονικού μηνύματος για παράδειγμα).
- *Μεταφερσιμότητα (mobility)*: Οι πράκτορες έχουν την ικανότητα μετακίνησης και δράσης και σε άλλα συστήματα πέρα από αυτό στο οποίο είναι εγκατεστημένοι. Αυτή τους την ικανότητα την βοηθά ιδιαίτερα η μεταξύ τους συνεργασία και η κοινωνική συμπεριφορά που οφείλουν να δείχνουν. Η εκτέλεση διεργασιών σε απομακρυσμένο υπολογιστικό περιβάλλον χρειάζεται την παροχή προηγουμένως άδεια, για την οποία θα χρειαστεί να επικοινωνήσουν και να συνεργαστούν με τους αντίστοιχους πράκτορες (White 1994).
- *Λογική (rationality)*: Ένας πράκτορας οφείλει να χαρακτηρίζεται από λογική και ευφυΐα όσον αφορά τις αποφάσεις που παίρνει και τον τρόπο προγραμματισμού επίτευξης στόχου. Η ευφυΐα και η λογική αφενός βασίζονται σε δυναμικούς κανόνες αφετέρου καλλιεργούνται με την εμπειρία. Μια βασική δυνατότητα για έναν πράκτορα είναι να μπορεί να μαθαίνει είτε από τον χρήστη του είτε από άλλους πράκτορες (Galliers 1994).
- *Συνεργιστικότητα (collaboration)*: Ένας πράκτορας δεν πρέπει να δέχεται και να εκτελεί εντολές, χωρίς προηγούμενο έλεγχο, αλλά θα πρέπει να λαμβάνει υπόψη του τη πιθανότητα οι χρήστες του να κάνουν λάθη όπως να ξεχνούν πληροφορίες ή ακόμα να δίνουν ασαφή πληροφόρηση. Ένας πράκτορας πρέπει να ελέγχει τέτοιες καταστάσεις είτε ρωτώντας τον χρήστη είτε δομώντας μοντέλα χρηστών (Eichmann 1994).
- *Προσανατολισμό στο στόχο (goal orientedness)*: Ένας πράκτορας είναι ικανός να διαχειρίζεται πολύπλοκες και υψηλού επιπέδου

εργασίες. Αποφάσεις για το πια εργασία θα διασπασθεί σε υποεργασίες και ποιές, ή με πια σειρά θα εκτελεστούν κάποιες από αυτές προκειμένου να κινηθούμε προς την επίτευξη των στόχων, τις παίρνουν οι υπεύθυνοι πράκτορες.

- *Αντίληψη (Responsiveness)*: Αντιλαμβάνονται το περιβάλλον τους και αποκρίνονται στις διάφορες αλλαγές του.

Όπως είναι προφανές από τις ιδιότητες τους η προσπάθεια επικεντρώνεται στη δημιουργία προγραμμάτων που έχουν χαρακτηριστικά οντοτήτων με νοημοσύνη, όπως ο άνθρωπος. Οι Intelligent agents εκτελούν εργασίες για λογαριασμό κάποιου χρήστη ή κάποιου άλλου προγράμματος. Για παράδειγμα τέτοιες οντότητες έχουν αναπτυχθεί για να ψάχνουν μέσα σε μηνύματα ηλεκτρονικού ταχυδρομείου για συγκεκριμένες φράσεις ή μέσα σε σελίδες WWW για συγκεκριμένα θέματα.

Ένα βασικό θέμα που αφορά την συνεργασία των agents είναι ο συντονισμός τους. Συντονισμός (coordination) των ευφυϊών πρακτόρων σε ένα σύστημα πολλαπλών πρακτόρων είναι η διαδικασία με την οποία ένας πράκτορας αντιλαμβάνεται τις ενέργειές του και τις (αναμενόμενες) ενέργειες των άλλων στοχεύοντας στην συμπαγή λειτουργία της κοινότητας σύμφωνα με τον Gerson (1976).

Οι Lesser και Corkill (1987) μίλησαν για την συντονιστική διεργασία, της οποίας στόχος είναι η διασφάλιση του ότι όλα τα επιμέρους κομμάτια του ολικού προβλήματος περιλαμβάνονται στις δραστηριότητες τουλάχιστον ενός ευφυούς πράκτορα, του ότι οι ευφυείς πράκτορες αλληλεπιδρούν με τέτοιο τρόπο ώστε οι δραστηριότητές τους να ενσωματώνονται σε μια ολοκληρωμένη λύση και πως όλα τα παραπάνω μπορούν να κατακτηθούν με την υπολογιστική ισχύ και τους πόρους που είναι διαθέσιμοι.

Ο Jennings (1996) βλέποντας το θέμα από τη σκοπιά του αρχιτέκτονα συστήματος πολλαπλών πρακτόρων συμπληρώνει τους

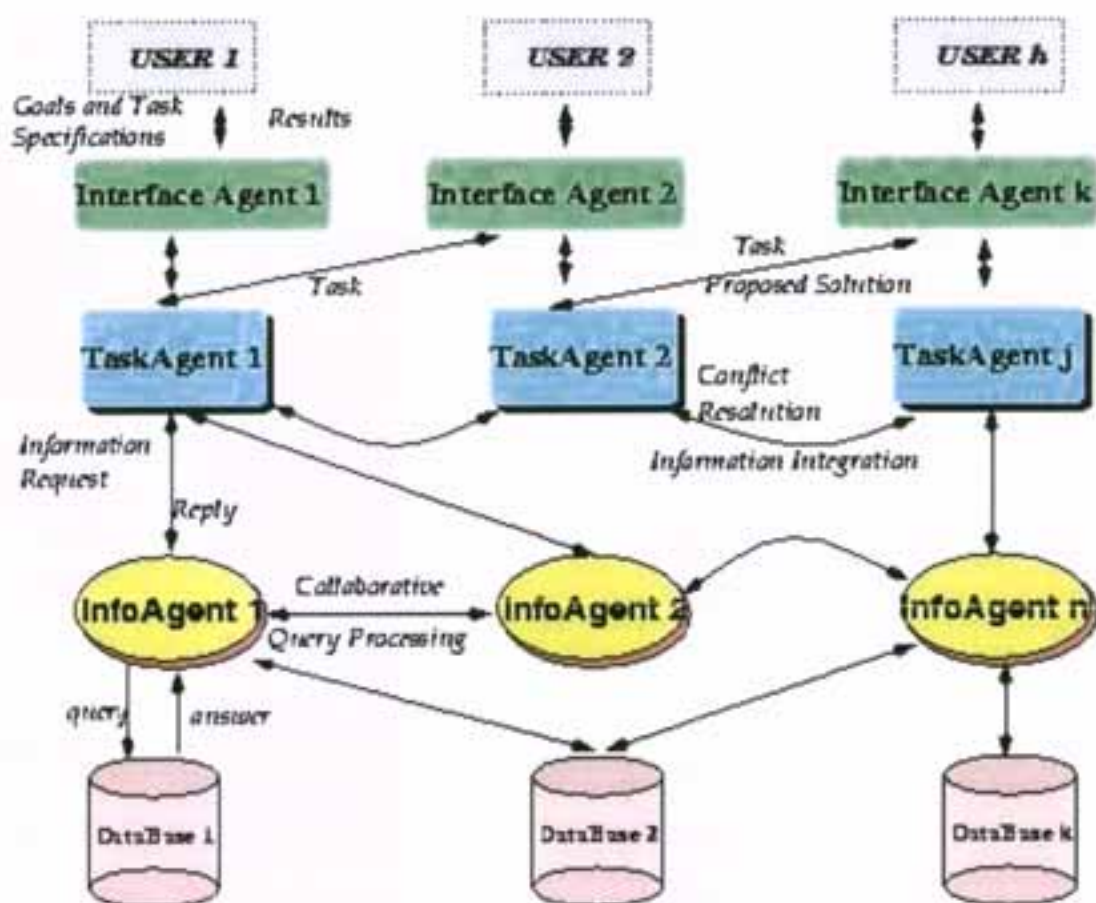
παραπάνω ερευνητές διατυπώνοντας τους λόγους για τους οποίους οι έξυπνοι πράκτορες πρέπει να συντονιστούν και οι οποίοι είναι οι εξής:

- Υπάρχουν εξαρτήσεις μεταξύ των ενεργειών των ευφυϊών πρακτόρων. Αυτό σημαίνει πως η δουλειά ενός ευφυούς πράκτορα μπορεί να προϋποθέτει την ολοκλήρωση της δουλειάς κάποιου άλλου. Έτσι για να φορτώσει ένας φορτωτής το φορτηγό πρέπει πρώτα ο εκκαφέας να έχει σκάψει το χώμα
- Υπάρχουν σφαιρικοί περιορισμοί που πρέπει να ικανοποιηθούν. Δηλαδή η εργασία των πρακτόρων είναι δυνατόν να έχει χρονικά περιθώρια, ή οι ευφυείς πράκτορες να έχουν περιορισμένο χρόνο αντίδρασης σε εξωτερικά ερεθίσματα, ή ακόμα να έχουν περιορισμένο εύρος επικοινωνιών
- Κανένα άτομο δεν έχει την απαραίτητη ικανότητα, πόρους ή πληροφορία για να λύσει ολόκληρο το πρόβλημα. Αυτό είναι αυτονόητο γιατί αλλιώς δεν θα είχαμε συστήματα πολλαπλών πρακτόρων αλλά ένα μονολιθικό λύτη προβλημάτων

Η συντονισμένη δράση των πρακτόρων σε ένα συστήματα πολλαπλών πρακτόρων είναι συνήθως δύσκολο να παρατηρηθεί από μια εξωτερική οπτική γωνία. Για παράδειγμα, το να πάνε όλοι οι πράκτορες που είναι κατανεμημένοι σε ένα χωράφι να κάτσουν κάτω από το μοναδικό δέντρο που υπάρχει σε μια δεδομένη χρονική στιγμή μπορεί να είναι αποτέλεσμα συντονισμού, όπως και τυχαίο αποτέλεσμα. Έτσι αν οι πράκτορες πήγαν κάτω από το δέντρο όλοι μαζί επειδή το απαιτούσε κάποια χορευτική φιγούρα, τότε αυτή τους η πράξη είναι αποτέλεσμα συντονισμού. Αν όμως η κίνηση αυτή των πρακτόρων έγινε επειδή έπιασε βροχή τότε δεν έχουμε συντονισμό αλλά την ατομική αντίδραση του κάθε πράκτορα σε ένα εξωτερικό ερέθισμα.

Στο σχήμα 4.3 δίνεται η κατανεμημένη αρχιτεκτονική του προτεινόμενου συστήματος RETSINA από τους Sycara and Zeng (1996) η οποία ακολουθεί τη γενική φιλοσοφία BDI των Rao and Georgeff (1993),

και στο οποίο φαίνονται οι τρεις τύποι agent: interface, task και information agents.



Σχήμα 4.3 Αρχιτεκτονική Συστήματος RETSINA

Μερικές από τις περιοχές εφαρμογής τις τεχνολογίας των agents δίνονται επιγραμματικά στη συνέχεια (Gilbert 1995):

Systems and Network

- Management
 - Mobile Access / Management
 - Mail and Messaging
- Information Access and Management _ Collaboration _ Workflow and Administrative Management _ Electronic Commerce _ Adaptive User Interfaces

Λαμβάνοντας υπόψη τα παραπάνω οι ευφυείς πράκτορες θα μπορούσαν να μοντελοποιηθούν στις εξής κύριες κατηγορίες:

- Δικτυακοί πράκτορες: Το κύριο επιπλέον χαρακτηριστικό αυτής της κατηγορίας των πρακτόρων είναι ο χώρος δράσης τους.
- Διεπιφανειακοί πράκτορες: Είναι οι πράκτορες λογισμικού που σαν πρώτους στόχους έχουν την διευκόλυνση της διεπικοινωνίας των χρηστών με τις διάφορες διεπιφάνειες χρήσης.
- Πράκτορες γενικής χρήσης: Πρόκειται για ευφυείς πράκτορες που αναλαμβάνουν επιλύσεις προβλημάτων σε δυναμικά και πολύπλοκα περιβάλλοντα και ψυχαγωγικούς πράκτορες.
- Οι intelligent agents μπορούν να χωριστούν σε τρεις κατηγορίες ανάλογα με την ικανότητά τους να λύνουν προβλήματα (O'Hare 1996, O'Hare και Jennings 1996)
- Reactive agents: Αυτό το είδος του πράκτορα μπορεί να στέλνει ή να δέχεται μηνύματα αντιδρώντας σε αλλαγές στο περιβάλλον του ή σε μηνύματα άλλων πρακτόρων.
- Intentional agents: Ο πράκτορας αυτός συγχρονίζεται με τους άλλους πράκτορες ανταλλάσσοντας πληροφορίες σχετικές με τις πεποιθήσεις τους, τους στόχους τους ή τις δράσεις τους ενώ είναι ικανός να κρίνει τις προθέσεις του και τις πεποιθήσεις του, να σχεδιάσει τις δράσεις του και να πραγματοποιήσει τα σχέδιά του.
- Social agents: Ο κοινωνικός πράκτορας πρέπει να διατηρεί τα μοντέλα και να κρίνει τη γνώση που αυτά περιέχουν (προθέσεις, υποχρεώσεις, προσδοκίες, αναμενόμενες αντιδράσεις και υποθετικές συμπεριφορές), ώστε να μπορεί να πάρει τις αποφάσεις του και να σχεδιάσει τις δράσεις του ανάλογα με τα μοντέλα αυτά.

ΚΕΦΑΛΑΙΟ 5

ΣΥΜΒΟΛΙΚΗ ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

5.1 LISP

Μια από τις πρώτες γλώσσες επεξεργασίας ήταν ο LISP (McCarthy, 1960). Αυτή η γλώσσα έχει υποβληθεί σε καλή ανάπτυξη κατά τη διάρκεια των προηγούμενων τεσσάρων δεκαετιών, αλλά το βασικό σχέδιο έχει παραμείνει το ίδιο. Όπως ο McCarthy (1965) επισημαίνει, ο LISP διαφέρει από τις περισσότερες γλώσσες προγραμματισμού από τρεις σημαντικές απόψεις:

- η κύρια δομή δεδομένων της είναι ο κατάλογος
- τα προγράμμάτα του αντιπροσωπεύονται επίσης από τις δομές καταλόγων και
- οι πρωτόγονες διαδικασίες της είναι διαδικασίες στους καταλόγους.

Η αρχική επιλογή του καταλόγου ως βασική δομή δεδομένων για τον LISP ήταν κάπως επαναστατική το 1960. Σήμερα, οι περισσότερες γενικής χρήσης γλώσσες υποστηρίζουν τους καταλόγους με κάποιο τρόπο, αν και ο προγραμματιστής πρέπει συνήθως να διαθέσει και να ελευθερώσει τη μνήμη που χρησιμοποιείται για να χτίσει έναν κατάλογο από τα συνδεδεμένα κύτταρα μνήμης. Το LISP ήταν επίσης επαναστατικό για την παροχή του μηχανισμού της αυτόματης 'συλλογής απορριμμάτων', οι οποίοι περνούν στη μνήμη αυτόματα.

Η βασική δομική μονάδα των δομών δεδομένων LISP είναι η συμβολική έκφραση, ή το **S - έκφραση**. Οι απλές S-εκφράσεις είναι ακριβώς ατομικά σύμβολα, ή *άτομο*: σειρές των αλφαριθμητικών χαρακτήρων αρχίζοντας με ένα γράμμα, παραδείγματος χάριν WOMBAT. Ένα άτομο αντιπροσωπεύεται εσωτερικά ως κύτταρο στην καλλιεργήσιμη

μνήμη. Το T είναι επίσης ένα ειδικό άτομο που αντιπροσωπεύει «αληθινό», ενώ ένα άλλο ειδικό άτομο, ΜΗΔΕΝ, αντιπροσωπεύει «ψεύτικο». Το ΜΗΔΕΝ χρησιμεύει επίσης ως ο κενός κατάλογος.

Οι σύνθετες εκφράσεις εφαρμόζονται ως κύτταρα μνήμης που συνδυάζονται για να διαμορφώσουν τις αυθαίρετες πεπερασμένες δομές δέντρων, που εκμεταλλεύονται μια απλή αλληλογραφία μεταξύ των πεπερασμένων δέντρων και τις S-εκφράσεις.

Οι κατάλογοι είναι πολύ εύκαμπτοι στις δομές δεδομένων, που είναι μη δακτυλογραφημένοι και του αυθαίρετου μήκους. Κατά συνέπεια δεν υπάρχει τίποτα λανθασμένο με τον κατάλογο, ("a" (9) () T (? (WOMBAT)) (A . B) ΜΗΔΕΝ 0.9) ακόμα κι αν περιλαμβάνει τα στοιχεία των μάλλον διαφορετικών τύπων.

Εντούτοις, υπάρχουν μερικά προφανή μειονεκτήματα με τους καταλόγους που έχουν οδηγήσει σε άλλες δομές δεδομένων που προστίθενται στον LISP. Οι κατάλογοι του LISP είναι πραγματικά πολλοί. Δεν υπάρχει καμία έννοια της πρόσβασης από τη θέση, όπως σε μια σειρά, ή από ένα κλειδί, στους πίνακες. Έτσι, αν και οι κατάλογοι είναι ακόμα χρήσιμοι ως αποθήκες για τα προσωρινά τοπικά στοιχεία στις ιδιαίτερες λειτουργίες, τα μεγάλα ποσά μόνιμων στοιχείων αντιπροσωπεύονται καλύτερα σε άλλα είδη δομής. Το σύγχρονο LISP παρέχει τις διάφορες εναλλακτικές λύσεις - συμπεριλαμβανομένων των σειρών και των αρχείων-όπως δομών - που είναι αποδοτικότεροι από την άποψη του διαστήματος αποθήκευσης και της ταχύτητας της πρόσβασης.

Κατάλογος και διάστικτα ζευγάρια

Λαμβάνοντας υπόψη έναν χειριστή για το συνδυασμό των κυττάρων στις δομές δέντρων, ο καθορισμός μιας S-έκφρασης στον LISP μπορεί να παραχθεί ως εξής:

οποιοδήποτε άτομο είναι μια S-έκφραση.

εάν το A1 και A2 είναι S-εκφράσεις, έπειτα (A1 . A2) είναι μια S-έκφραση.

Εάν $S=(A_1 . (A_2 . (A_{n-1} . A_n)\dots))$ είναι μια S-έκφραση για κάποιο $n>0$, κατόπιν το S είναι ένας κατάλογος εάν $A_n = \text{NLL}$.

Από τη σύμβαση, εάν $n=0$, έπειτα S είναι ο κενός κατάλογος, NLL. Αυτός ο καθορισμός επιτρέπει τον κατάλογο καταλόγων, καθώς επίσης και τους καταλόγους ατόμων. Εάν S_1, S_2, \dots, S_n είναι S-εκφράσεις, κατόπιν θα αντιπροσωπεύσουμε τον κατάλογο

$$(S_1 . (S_2 . (\dots (S_n . \text{NLL})\dots)))$$

όπως απλά

$$(S_1 S_2 \dots S_n)$$

κατά συνέπεια (A . (B . NLL)) είναι ένας κατάλογος. Αντιπροσωπεύει τον κατάλογο (A B), αλλά (A . (B .C)) δεν είναι ένας κατάλογος, επειδή $C \neq \text{NLL}$.

Οι S-εκφράσεις που δεν είναι ούτε άτομα ούτε κατάλογοι καλούνται διάστικτα ζευγάρια. Εάν (A . B) είναι ένα διάστικτο ζευγάρι, κατόπιν το A είναι το κεφάλι του, και B η ουρά του. Τα διάστικτα ζευγάρια μπορούν να είναι σύνθετα. Κατά συνέπεια ((A . B) . C) είναι ένα διάστικτο ζευγάρι,

όπως είναι $((A \ . \ B) \ . \ (C \ . \ D))$. Χάρης στην αλληλογραφία μεταξύ των διάστικτων ζευγαριών και των καταλόγων, οι έννοιες του κεφαλιού και της ουράς καθορίζονται για αυτούς. Λαμβάνοντας υπόψη ότι ο κατάλογος $(A \ B)$ είναι απλώς $(A \ . \ (B \ . \ NLL))$, πρέπει να είναι προφανές ότι αν και το A είναι το κεφάλι $(A \ B)$, η ουρά του είναι (B) , όχι το B . τελικά, η ουρά (B) είναι NLL .

5.1.1 ΠΡΟΓΡΑΜΜΑΤΑ LISP

Μπορούμε εύκολα να συνηθίσουμε την ιδέα των καταλόγων ως δομές δεδομένων. Λιγότερο γνωστή είναι η ιδέα ότι ένας κατάλογος μπορεί να είναι ένα πρόγραμμα, ή δηλώσεις ενός προγράμματος παραδείγματος χάριν, ο κατάλογος

$$(+ \ X \ Y)$$

είναι μια μαθηματική έκφραση της μορφής

$$(< \text{λειτουργία} > \ < 1^\circ \ \text{επιχείρημα} > \ < 2^\circ \ \text{επιχείρημα} >)$$

δείξιμο της προσθήκης δύο αριθμών. Αυτή η σημείωση, κάπως διαφορετική

$$F (X_1, \dots, X_n)$$

η σημείωση ότι χρησιμοποιούμαστε για τις λειτουργίες των επιχειρημάτων n , υιοθετείται ακριβώς, επειδή θέλουμε την επακόλουθη έκφραση για να είναι ένας κατάλογος.

Αλλά πώς θα μπορούσε αυτή η εργασία, χωρίς να προκαλέσει σύγχυση; Πώς θα μπορούσε ένας μεταγλωττιστής ή ένας διερμηνέας για τη γλώσσα να ξέρει πότε εξετάζει τα στοιχεία και πότε εξετάζει τις

εκτελέσιμες δηλώσεις; Τουλάχιστον τα ακόλουθα μηχανήματα απαιτούνται.

Πρέπει να είμαστε σε θέση **να αξιολογήσουμε** τις εκφράσεις όπως (+ X Y). Για να το κάνουμε αυτό, πρέπει να είμαστε σε θέση να ανατρέξουμε σε κάποιο καθορισμό λειτουργίας για +, ο οποίος μας λέει ποια ακολουθία πρωτόγονων διαδικασιών για να ισχύσει τα επιχειρήματα προκειμένου να υπολογισθεί η αξία.

Πρέπει να είμαστε σε θέση **να καθορίσουμε** τις λειτουργίες αρχικά, και να εφαρμόσουμε τους ορισμούς τους στα επιχειρήματα (δηλαδή μη επίσημος, παράμετροι). Οι μηχανισμοί του καθορισμού λειτουργίας και της εφαρμογής λειτουργίας είναι βασισμένοι σε ένα λογιστικό σύστημα, αποκαλούμενο **υπολογισμό λάμδα** (Church, 1941). Αντίθετα από την πρώτη - διαταγή υπολογισμού κατηγορήματος, ο υπολογισμός λάμδα είναι λειτουργικός παρά συγγενικός, δηλαδή η πρωτόγονη έννοια είναι αυτή των σχέσεων πολλοί - πολλές σχέσεις. Κατά συνέπεια ο πατέρας είναι μια λειτουργική σχέση, δεδομένου ότι για κάθε πρόσωπο υπάρχει μόνο ένας πατέρας, ενώ ο αδελφός είναι μια γενικότερη σχέση, δεδομένου ότι κάθε πρόσωπο μπορεί να έχει περισσότερους από έναν αδελφούς. Θα αγγίξουμε τη σχέση μεταξύ του LISP και του υπολογισμού λάμδα αργότερα σε αυτό το τμήμα.

Πρέπει να είμαστε σε θέση **να έχουμε πρόσβαση** στις τρέχουσες τιμές των μεταβλητών (ή των επίσημων παραμέτρων) όπως το X και το Y, που η αξιολόγηση κάθε S-έκφρασης εκτελείται σχετικά με ένα περιβάλλον, δεδομένου ότι υπολογίζουμε την αξία μιας σύνθετης S-έκφρασης με τον υπολογισμό και το συνδυασμό των τιμών οποιωνδήποτε υπό-εκφράσεων που περιλαμβάνονται εκεί μέσα.

Στην αξιολόγηση των σύνθετων S-εκφράσεων, που μπορούν να συμπεριλάβουν και την εφαρμογή περισσότερων από μιας λειτουργιών

και τη σύνθεση των λειτουργιών, πρέπει να είμαστε σε θέση να αποθηκεύσουμε την τρέχουσα S - έκφραση, δηλαδή αυτή που αξιολογούμε σε μια δεδομένη στιγμή. Πρέπει να είμαστε σε θέση να αντιγράψουμε τις S - εκφράσεις εν αναμονή της αξιολόγησης ώστε να μπορούμε να τους σώσουμε κατά τη διάρκεια των υπό-υπολογισμών. Πρέπει επίσης να αποθηκεύσουμε τα ενδιάμεσα αποτελέσματα.

Τέλος, πρέπει να είμαστε σε θέση να εμποδίσουμε την αξιολόγηση των καταλόγων που δεν είναι δηλώσεις προγράμματος αλλά μόνο δομές δεδομένων. Κατά συνέπεια, δεν θέλουμε να προσπαθήσουμε να αξιολογήσουμε μια έκφραση όπως

((1 2 3) (4 5 6) (7 8 9))

για να δούμε εάν έχουμε έναν καθορισμό για την ανύπαρκτη λειτουργία

(1 2 3).

Αυτό επιτυγχάνεται από το ΑΠΟΣΠΑΣΜΑ ειδικής μορφής: για όλο το X,

(ΑΠΟΣΠΑΣΜΑ X)

επιστροφή στο X.

Εάν αφήνουμε ((λ X) M) (α) στάση για την εφαρμογή του καθορισμού λειτουργίας (λ X) μ στο επιχείρημα α, έπειτα

((λX)(X * X)) (3) = (3 * 3) = 9.

(καθορίστε το ΤΕΤΡΑΓΩΝΟ (λ(X) (* X X))).

Οι ακριβείς λεπτομέρειες εξαρτώνται από τη διάλεκτο και την εφαρμογή της γλώσσας, αλλά η πρόθεση είναι σαφής. Στην πραγματικότητα, στον ΚΟΙΝΟ LISP, κάποιος θα χρησιμοποιούσε απλά τη στενογραφία

(defun TETΡΑΓΩΝΙΚΟ (X) (* X X)).

Καθένας τρόπος, η πρόθεση του καθορισμού είναι να συνδεθεί ένα όνομα λειτουργίας, όπως το ΤΕΤΡΑΓΩΝΟ, με μια έκφραση λάμδα που αποτελεί τον καθορισμό της. $(\lambda (X) (* X X))$ είναι ακριβώς μια συντακτική παραλλαγή $(\lambda X) (X * X)$. Στην πραγματικότητα

$(\text{TETΡΑΓΩΝΟ } 3) = ((\lambda(X) (* X X)) 3) = ((\lambda X)(X * X)) (3) = 9$

σε οποιοδήποτε LISP, μια φορά το ΤΕΤΡΑΓΩΝΟ έχει καθοριστεί. Αξιολογώντας κατά συνέπεια

$((\lambda (X) (* X X)) 3)$

δεσμεύει το X σε 3 από τη μετατροπή λάμδα, έτσι ώστε $(* X X)$ τώρα δείχνει τον αριθμό 9.

Η βασική σύνταξη μιας κλήσης λειτουργίας LISP είναι ακριβώς ένας κατάλογος της μορφής

$(\langle \text{Λειτουργία} \rangle \langle \text{επιχείρημα} \rangle \dots \langle \text{επιχείρημα} \rangle)$.

5.1.2 ΕΠΕΞΕΡΓΑΣΙΑ ΚΑΤΑΛΟΓΩΝ

Το LISP είναι μια επίσημη γλώσσα που μπορεί να δοθεί μια πολύ συνοπτική περιγραφή. Τα περισσότερα προγράμματα LISP μπορούν να διευκρινιστούν από την άποψη μόνο πέντε πρωτόγονων διαδικασιών στο s - εκφράσεις και μια ειδική μορφή (μια υπό όρους έκφραση). Είναι εύκολο να αγνοηθεί η ομορφιά και η κομψότητα του καθαρού LISP, επειδή οι περισσότερες εφαρμογές παρέχουν έναν πλήθος των πρόσθετων διαδικασιών. Ο σύγχρονος LISP πνίγεται επίσης στα

FORTTRAN-χαρακτηριστικά γνωρίσματα, μερικά από τα οποία είναι χρήσιμα, και μερικά από τα οποία είναι μια αποστροφή.

Βγάζει ότι οι κατάλογοι είναι ένα εξαιρετικά καλό μέσο για τους μη αριθμητικούς υπολογισμούς οι οποίοι απαιτούν ότι ένα πρόγραμμα τοποθετεί την κάποια αναζήτηση μιας λύσης σε ένα διάστημα των εναλλακτικών λύσεων. Οι κατάλογοι μπορούν να παρακολουθήσουν αυτές τις εναλλακτικές λύσεις, εκείνοι που έχουν επισκεφτεί ήδη, εκείνοι που πρέπει να επισκεφτούν ακόμα, και ούτω καθ' εξής. Λαμβάνοντας υπόψη τον ισομορφισμό μεταξύ των καταλόγων, τα διαταγμένα δέντρα, αυτό είναι φυσικά να αντιπροσωπεύσουν ένα ξετυλίγοντας διάστημα αναζήτησης που χρησιμοποιεί έναν ή περισσότερους καταλόγους.

5.2 ΣΥΜΠΕΡΑΣΜΑΤΑ

Η εξειδικευμένη γλώσσα LIPS έχει να κάνει με τον προγραμματισμό των εννοιών και των δομών που είναι ουσιαστικές στο σχέδιο των έμπειρων συστημάτων. Η συγκεκριμένη γλώσσα όπως έχει αναφερθεί παρα πάνω είναι συμβολική. Η λέξη συμβολική πηγάζει από τη λέξη **σύμβολο** που σε αυτή την περίπτωση η έννοια ενός συμβόλου είναι κυρίαρχη στη τρέχουσα θεωρία και πρακτική της τεχνητής νοημοσύνης. Είναι αυτή η έννοια που διαμορφώνει την κρίσιμη σύνδεση μεταξύ της τεχνητής νοημοσύνης και των επίσημων συστημάτων της λογικής και των μαθηματικών.

Η συγκεκριμένη γλώσσα προγραμματισμού παρέχει διάφορες πρωτόγονες δομές δεδομένων για τα σύμβολα ένωσης με άλλα σύμβολα, καθώς επίσης και διαδικασίες χειρισμού των συμβόλων και των σχετικών δομών τους. Ο κατάλογος που χρησιμοποιείται από τον LISP είναι μια δομή συμβόλων ο οποίος είναι μια φυσική οντότητα.

Χαρακτηριστικά ένα συμβολικό πρόγραμμα παίρνει ως εισαγωγή του μια ή περισσότερες δομές συμβόλων, που αντιπροσωπεύουν την αρχική κατάσταση κάποιου προβλήματος, και επιστρέφει ως παραγωγή μιας δομής συμβόλων, που αντιπροσωπεύει λύση.

ΕΠΙΛΟΓΟΣ

Είναι πραγματικά δύσκολο να πει κανείς υπεύθυνα και αντικειμενικά σε ποίο στάδιο έχει φτάσει η έρευνα στον τομέα της τεχνητής νοημοσύνης. Τα πρακτικά αποτελέσματα των προσπαθειών που γίνονται στον τομέα αυτό ίσως φανούν καθαρά μερικά χρόνια αργότερα. Προς το παρόν, πολλές καινούργιες σκέψεις εμφανίζονται και μελετώνται και η μαγευτική ιδέα της προσωμοίωσης του ανθρώπου ερεθίζει, προκαλεί και παροτρύνει τους ανθρώπους της έρευνας σε συνεχή προσπάθεια, για την ανάπτυξη μεθόδων που στοχεύουν στην προσομοίωση της κατασκευής του ανθρώπινου νου στα προγράμματα.

Εως ότου ολοκληρωθούν περισσότερο επιτυχημένες προσπάθειες επίτευξης των πιο πάνω στόχων, κανείς δεν μπορεί να πεί με σιγουρία αν τα σημερινά επιτεύγματα στο τομέα της τεχνητής νοημοσύνης στοιχειοθετούν την αρχή ή το τέλος της προσπάθειας του άνθρωπου στον τομέα αυτό.

Η περιπλάνηση όμως του ερευνητή στο συναρπαστικό αυτό κόσμο της αναζήτησης έχει προσφέρει στην επιστήμη την ικανότητα και πείρα καθώς επίσης μια ωριμότητα και μια εξελιγμένη άποψη, όσον αφορά την αντιμετώπιση των πραγματικών προβλημάτων.

Τα έμπειρα συστήματα τα οποία είναι ένα προϊόν της έρευνας που γίνεται στον τομέα της τεχνητής νοημοσύνης δεν αποτελούν πανάκεια, ούτε οδηγούν σε αντικατάσταση του ανθρώπου. Εκφράζουν απλώς το συμπλήκνωμα της πείρας που αποκτήθηκε από μια ενασχόληση και συνιστούν μια νέα οπτική, η οποία μεταβάλλει και εμπλουτίζει την άποψη και την πρακτική που έχουμε για τον προγραμματισμό και προτείνει μια νέα φιλοσοφία και μέθοδο για την ανάπτυξη προγραμμάτων.

Ένα έμπειρο σύστημα είναι ένα πρόγραμμα που βοηθά τον χρήστη, παρέχοντας πληροφορίες σε κάποιο πεδίο. Αυτό το κάνει χειριζόμενο πληροφορίες σχετικές με ένα χώρο. Ένα άλλο σημαντικό χαρακτηριστικό ενός έμπειρου συστήματος είναι ότι αυτό έχει την δυνατότητα να εξηγεί, να αιτιολογεί τις χρησιμοποιηθείς μεθόδους για την παροχή πληροφοριών.

ΠΕΡΙΕΧΟΜΕΝΑ

	Σελ.
Πρόλογος	1
Ιστορική Ανασκόπηση	2
Εισαγωγή	4
Κεφάλαιο 1: Τεχνητή Νοημοσύνη	5
1.1 Φάσεις τεχνητής νοημοσύνης	5
1.2 Νοήμονα συστήματα	9
1.3 Η τεχνητή νοημοσύνη έναντι της ανθρώπινης νοημοσύνης	13
Κεφάλαιο 2: Έμπειρα Συστήματα	17
2.1 Ανατομία ενός έμπειρου συστήματος	17
2.2 Ορισμοί έμπειρων συστημάτων	19
2.3 Λειτουργία και στόχοι έμπειρων συστημάτων	21
2.4 Χαρακτηριστικά ενός έμπειρου συστήματος	22
2.5 Σχεδιασμός έμπειρων συστημάτων	25
2.5.1 Χρήστης	28
2.5.2 Συστήματα επικοινωνίας	28
2.5.3 Μηχανισμός εξαγωγής συμπερασμάτων	29
2.5.4 Βάσεις Γνώσης	31
2.5.4.1 Υποσυστήματα απόσπασης γνώσης	31
2.5.5 Μηχανική Γνώσης	31
2.5.5.1 Απόκτηση γνώσης	33
Κεφάλαιο 3: Μέθοδοι Αναπαράστασης Γνώσης	40
3.1 Αναπαράσταση της γνώσης	40
3.2 Συλλογιστική υπό αβεβαιότητα	51
3.2.1 Ασαφής λογική	52
3.3 Εξαγωγή συμπερασμάτων	55
3.4 MYCIN	57
3.4.1 Βάση γνώσεων MYCIN	61
3.4.2 Δομή Ελέγχου MYCIN	66
3.5 Αξιολόγηση και σύγκριση των έμπειρων συστημάτων	66
3.5.1 Αξιολόγηση – Συμπεράσματα MYCIN	66
Κεφάλαιο 4: Ανάπτυξη έμπειρων συστημάτων	70
4.1 Προβλήματα εφαρμογής έμπειρων συστημάτων	70
4.1.1 Διαδικασία ανάπτυξης	71
4.1.2 Επισκόπηση των εργαλείων έμπειρων συστημάτων	77
4.1.2.1 Εργαλεία ανάπτυξης	78
4.2 Κέλυφος έμπειρων συστημάτων	79
4.2.1 Ταιριάζοντας τα κελύφη στους στόχους	80
4.3 Intelligent Agent	82
Κεφάλαιο 5: Συμβολική Γλώσσα Προγραμματισμού	90
5.1 LISP	90
5.1.1 Προγράμματα LISP	93
5.1.2 Επεξεργασία καταλόγων	96
5.3 Συμπεράσματα	97
Επίλογος	99

Βιβλιογραφία

Edmund C Payne, Robert C. McAthur 1990, Developing Expert Systems, A knowledge engineer' s handbook for Rules & Objects

Peter Lucas Linda Van Der Gaag 1990, Principles of Expert System

James Ignizio, Introduction to Expert Systems 1991

D.A. Waterman, *A Guide to Expert Systems*. Addison-Wesley, 1985

N. Nilsson, *Artificial Intelligence: A New Synthesis*. Morgan Kaufmann, 1998

http://en.wikiped.org/wiki/Expert_system

<http://ketnet.cs.unipi.gr/education>

www.cs.cmu.edu/Groups

www.2.cs.cmu