



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ  
ΕΡΓΑΣΤΗΡΙΟ ΣΧΕΔΙΑΣΜΟΥ ΚΑΙ ΑΝΑΠΤΥΞΗΣ  
ΣΥΣΤΗΜΑΤΩΝ ΥΠΟΣΤΗΡΙΞΗΣ ΑΠΟΦΑΣΕΩΝ

*Διπλωματική Εργασία*

ΑΝΑΠΤΥΞΗ ΕΝΟΣ ΒΑΣΙΣΜΕΝΟΥ ΣΤΗΝ ΤΕΧΝΟΛΟΓΙΑ ΤΩΝ  
ΠΟΛΛΑΠΛΩΝ ΠΡΑΚΤΟΡΩΝ ΣΥΣΤΗΜΑΤΟΣ ΓΙΑ ΤΗΝ ΠΡΟΣΟΜΟΙΩΣΗ  
ΤΗΣ ΓΡΑΜΜΗΣ ΠΑΡΑΓΩΓΗΣ ΤΥΠΟΓΡΑΦΕΙΟΥ

Τσουκάκης Δημήτριος

Τριμελής Εξεταστική Επιτροπή

Αναπληρωτής Καθηγητής Ν.Φ. Ματσατσίνης (Επιβλέπων)

Καθηγητής Κ. Ζοπουνίδης

Καθηγητής Α.Μυδαλάς

ΧΑΝΙΑ 2005

*Στους γονείς μου  
Γιώργο και Αθηνά  
για την αγάπη τους*

## ΕΥΧΑΡΙΣΤΙΕΣ

Θέλω να ευχαριστήσω θερμά τον επιβλέπων καθηγητή κ. Νικόλαο Ματσατσίνη για τον πολύτιμο χρόνο που αφιέρωσε, τη συνεχή συνεργασία, την καθοδήγηση και την πολύτιμη βοήθεια που μου προσέφερε καθ' όλη τη διάρκεια της εκπόνησης της παρούσας διπλωματικής εργασίας.

Σημαντική και πολύτιμη ήταν και η συνεργασία μου, με τον κ. Δημήτρη Δετοράκη, διευθύνων σύμβουλο της Γ. Δετοράκης ΑΕΒΕ, τον οποίο ευγνωμονώ για την υπομονή, την καλή διάθεση και την προσπάθεια που κατέβαλε για την πραγματοποίηση αυτής της προσπάθειας.

Επίσης, θέλω να ευχαριστήσω τον εργοδότη μου κ. Άγγελο Καραγκουνάκη, ο οποίος μου έδωσε το έναυσμα για να ενταχθώ στον κόσμο του προγραμματισμού υπολογιστών και της πληροφορίας καθώς και τον φίλο και συνεργάτη μου κ. Παύλο Δελιά για την πολύτιμη βοήθεια που μου προσέφερε.

Τέλος θα ήθελα να ευχαριστήσω θερμά, όλους τους φίλους, τους συμφοιτητές, τους καθηγητές μου καθώς και την οικογένειά μου για την πολύτιμη βοήθεια και την ηθική υποστήριξη που μου προσέφεραν καθ' όλη τη διάρκεια της φοίτησής μου στο τμήμα Μηχανικών Παραγωγής και Διοίκησης του Πολυτεχνείου Κρήτης.

## Περιεχόμενα

Περιεχόμενα.....	- 1 -
1. Εισαγωγή.....	- 6 -
1.1    Επισκόπηση των προσεγγίσεων του προβλήματος Ανάθεσης Εργασιών .....	- 8 -
1.2    Οργάνωση της εργασίας .....	- 10 -
2. Θεωρητικό Υπόβαθρο.....	- 11 -
2.1    Συστήματα Υποστήριξης Αποφάσεων (Σ.Υ.Α.).....	- 11 -
2.1.1    Ορισμοί – χαρακτηριστικά .....	- 11 -
2.1.2    Δομή Σ.Υ.Α.....	- 12 -
2.1.3    Ευφυή Συστήματα Υποστήριξης Αποφάσεων (Intelligent DSS) .....	- 14 -
2.2    Πράκτορες και συστήματα πολλαπλών πρακτόρων (Agents & Multi-Agent Systems).....	- 14 -
2.2.1    Πράκτορες και ευφυείς πράκτορες (Agents & Intelligent Agents) .....	- 14 -
2.2.2    Αρχιτεκτονικές Ευφυνών Πρακτόρων.....	- 18 -
2.2.3    Βασικοί Τύποι Πρακτόρων .....	- 20 -
2.2.4    Επιπλέον Ορισμοί .....	- 20 -
2.3    Συστήματα πολλαπλών πρακτόρων (multi-agent systems) .....	- 21 -
2.3.1    Συντονισμός, Συνεργασία και Διαπραγματεύσεις .....	- 22 -
2.3.2    Επικοινωνία και Αλληλεπιδράσεις των Ευφυνών Πρακτόρων .....	- 27 -
2.3.3    Μεθοδολογίες Ανάπτυξης Συστημάτων Πολλαπλών Πρακτόρων .....	- 30 -
2.4    Έμπειρα Συστήματα (Expert Systems) .....	- 34 -
2.4.1    Η φύση της ειδίκευσης.....	- 34 -
2.4.2    Άλλοι Ορισμοί .....	- 35 -
2.4.3    Τα χαρακτηριστικά ενός έμπειρου συστήματος .....	- 36 -
2.4.3    Θεμελιώδη ζητήματα των έμπειρων συστημάτων.....	- 37 -
2.4.4    Έλεγχος της λειτουργίας του διερμηνέα .....	- 40 -
2.4.5    Σύνοψη Έμπειρων Συστημάτων .....	- 43 -
3. Περιγραφή του προβλήματος της Προσομοίωσης και Κοστολόγησης Γραμμών Παραγωγής.....	45
3.1    Το πρόβλημα.....	45
3.2    Παρούσα Κατάσταση.....	45
3.3    Εκμείευση Γνώσης .....	48
3.4    Βασικοί μέθοδοι εκτύπωσης .....	49
3.4.1    Ποίες είναι οι μέθοδοι εκτύπωσης .....	49

3.4.2 Τι είναι η Ταμπονογραφία.....	49
3.4.3 Τι είναι η Τυπογραφία ή αλλιώς Υψιτυπία.....	49
3.4.4 Τι είναι η Φλεξογραφία.....	49
3.4.5 Τι είναι η Offset (Λιθογραφία ή Επιπεδοτυπία) .....	49
3.4.6 Τι είναι η Μεταξοτυπία.....	50
3.4.7 Τι είναι η Ψηφιακή Εκτύπωση.....	50
3.4.8 Τι είναι το halftone ή ράστερ, ή κουκίδα ή ημιτονικό.....	50
4. Σχεδίαση Συστήματος.....	52
4.1 Μεθοδολογία σχεδίασης .....	52
4.2 Καθορισμός των στόχων.....	54
4.3 Προσδιορισμός των πρακτόρων και των ρόλων τους.....	55
4.4 Αρχιτεκτονική του συστήματος και των πρακτόρων.....	56
4.5 Τάξεις πρακτόρων.....	57
4.5.1 Ο πράκτορας διεπαφής τελικού χρήστη .....	57
4.5.2 Ο πράκτορας ανάθεσης εργασιών.....	61
4.5.3 Οι πράκτορες εργασιών .....	62
4.5.4 Ο πράκτορας διαχείρισης.....	65
4.6 Τα αντικείμενα του προγράμματος.....	68
5. Παράδειγμα εκτέλεσης του προγράμματος .....	80
6. Συμπεράσματα – Επεκτάσεις.....	84
7. Βιβλιογραφία .....	85
Ευρετήριο Διαγραμμάτων.....	87

## 1. Εισαγωγή

Η λήψη μιας απόφασης είναι μία διαδικασία την οποία βιώνουμε όλοι στην καθημερινότητα μας. Η επιλογή της διαδρομής που θα ακολουθήσουμε για να φτάσουμε στον προορισμό μας, το τι φαγητό θα φάμε μέχρι και το πως θα μοιράσουμε το εισόδημά μας, είναι όλα συνήθη προβλήματα απόφασης. Αυτή ωστόσο η καθημερινή μας τριβή με τη διαδικασία απόφασης, αν και αυξάνει την προσωπική μας γνώση και εμπειρία, δεν είναι ικανή για να υποστηρίξει την άποψη πως είμαστε σε θέση να πετυχαίνουμε τη βέλτιστη απόφαση και λύση σε όλα τα προβλήματα που μας απασχολούν. Όταν μάλιστα ο όγκος των πληροφοριών αυξάνεται και η διαδικασία της απόφασης γίνεται ολοένα και πιο πολύπλοκη, η προηγούμενη άποψη δεν ευσταθεί.

Στη διαδικασία απόφασης λαμβάνουμε υπόψη μας, λιγότερο ή περισσότερο συνειδητά, ορισμένα κριτήρια. Μάλιστα τις περισσότερες φορές τα κριτήρια αυτά είναι περισσότερα από ένα. Η πολυπλοκότητα της λήψης απόφασης αυξάνεται ανάλογα με το πλήθος των εναλλακτικών, τον αριθμό των κριτηρίων και τον όγκο των πληροφοριών, που έχουμε. Όταν γνωρίζουμε τον τρόπο με τον οποίο θα εκτιμήσουμε όλα αυτά τα δεδομένα και ακολουθούμε μία συγκεκριμένη και σαφώς καθορισμένη διαδικασία λήψης απόφασης, τότε η επιλογή μας θα είναι ικανοποιητική αν όχι βέλτιστη. Η εμπειρία μας όμως, δείχνει πως κάτι τέτοιο σπάνια συμβαίνει. Η διαδικασία λήψης απόφασης είναι αόριστα δομημένη. Για τέτοια ακριβώς χαμηλής δόμησης προβλήματα, αναπτύχθηκαν τα Συστήματα Υποστήριξης Αποφάσεων. Τα συστήματα αυτά έχουν σα σκοπό να βοηθήσουν τον αποφασίζοντα να καταλήξει σε μία ικανοποιητική ή ακόμα και βέλτιστη απόφαση για τα προβλήματα που τον απασχολούν.

Το πρόβλημα που απασχολεί την εργασία αυτή είναι η ανάθεση εργασιών σε μηχανές ή εργαζόμενους παραγωγικών μονάδων με απώτερο σκοπό την προσομοίωση της γραμμής παραγωγής εργοστασίου και εντέλει την εξαγωγή λεπτομερών κοστολογίων των παραγόμενων προϊόντων. Η ανάλυση του προβλήματος θα γίνει πάνω στην γραμμή παραγωγής ενός τυπογραφείου για τον λόγο ότι τα παραγόμενα προϊόντα χαρακτηρίζονται από υψηλή ανομοιομορφία, γεγονός που καθιστά αδύνατη την υιοθέτηση συγκεκριμένης γραμμής και σε επέκταση την εξαγωγή κοστολογίων χρήση στατιστικών στοιχείων.

Η **πληροφορίες που απαιτούνται** για να εξαχθεί ένα κοστολόγιο είναι οι προδιαγραφές των προϊόντων, η γραμμή παραγωγής που πρέπει να ακολουθηθεί, οι παραγωγικές παραμέτρους για την παραγωγή του συγκεκριμένου προϊόντος (ρυθμοί παραγωγής, χρόνοι δέσμευσης εργαζομένων, νεκροί χρόνοι μηχανών κ.ο.κ.) και τα τεχνικά χαρακτηριστικά των μηχανών του εργοστασίου (μέγιστες και ελάχιστες διαστάσεις εισερχόμενων υποπροϊόντων, ποιότητα - είδος εισερχόμενων πρώτων υλών κ.ο.κ.).

Στο πρόβλημα που επιλύει αυτή η εργασία, τα **δεδομένα που δίδονται** είναι οι προδιαγραφές του προϊόντος, τα τεχνικά χαρακτηριστικά των μηχανών της παραγωγικής μονάδας, οδηγοί χρόνου και κόστους εκτέλεσης εργασιών ανά μηχανή (ή εργαζόμενο αν πρόκειται για καθαρά χειρωνακτική εργασία) και ανά διαδικασία, πρότυπα (templates) αλληλουχίας διαδικασιών που πρέπει να εκτελεστούν για την παραγωγή προϊόντων αντιπροσωπευτικών των διαφόρων ομάδων προϊόντων που παράγει το εργοστάσιο και τέλος κανόνες (συνήθως εμπειρικοί) στους οποίους υπακούουν συγκεκριμένα προϊόντα, διαδικασίες και μηχανές.

Έτσι το **έργο που πρέπει να επιτελεστεί** είναι:

- ⇒ Ένταξη του προϊόντος σε κάποια ευρύτερη ομάδα προϊόντων σύμφωνα με τις προδιαγραφές του.
- ⇒ Εξαγωγή πιθανών γραμμών παραγωγής σε επίπεδο διαδικασιών συνδυάζοντας τα πρότυπα γραμμών παραγωγής, τις προδιαγραφές του προϊόντος και τους εμπειρικούς κανόνες.
- ⇒ Εξαγωγή πιθανών γραμμών παραγωγής σε επίπεδο μηχανών βάσει των τεχνικών χαρακτηριστικών της κάθε μηχανής, την κατάσταση των υποπροϊόντων την στιγμή που πρέπει να εισέλθουν στην εκάστοτε μηχανή και εμπειρικών κανόνων. Τελικά δημιουργείται ένα δέντρο όπου τα διάφορα μονοπάτια αντιστοιχούν σε εναλλακτικές γραμμές παραγωγής.
- ⇒ Υπολογισμός των παραγωγικών παραμέτρων σε κάθε κόμβο της γραμμής παραγωγής.
- ⇒ Υπολογισμός κόστους και χρόνου σε κάθε κόμβο της γραμμής παραγωγής βάσει των παραμέτρων που υπολογίστηκαν παραπάνω και των δοθέντων αντίστοιχων οδηγιών.
- ⇒ Ανίχνευση του δέντρου που δημιουργήθηκε και κατάταξη των μονοπατιών που οδηγούν σε τελικό προϊόν βάσει κόστους.

Καθώς η διαδικασία που περιγράφεται παραπάνω χαρακτηρίζεται από υψηλή πολυπλοκότητα και εφαρμογή εξειδικευμένων επιχειρησιακών μοντέλων, αποφασίστηκε η ανάπτυξη ενός Συστήματος Υποστήριξης Αποφάσεων.

Τα Συστήματα Υποστήριξης Αποφάσεων περιγράφονται αργότερα σε αυτήν την εργασία. Εδώ θα αναφέρουμε απλά πως για την προσέγγιση του προβλήματος βασιστήκαμε στην κατανομημένη τεχνητή νοημοσύνη και την τεχνολογία πρακτόρων που αυτή υποστηρίζει. Η διαδικασία απόφασης που θα υποστηρίξει το σύστημα βασίζεται στην μίμηση της αντίστοιχης διαδικασίας που ακολουθείται από τους εκάστοτε αποφασίζοντες.

Αναλυτικότερα, η εργασία αυτή έχει σα στόχο να σχεδιάσει ένα Σύστημα Υποστήριξης Αποφάσεων που θα υποβοηθά τον χρήστη του να επιλέγει με βέλτιστο τρόπο την γραμμή παραγωγής που πρέπει να ακολουθήσει το εκάστοτε προϊόν και που θα υποδεικνύει το αντίστοιχο κοστολόγιο της παραγωγικής διαδικασίας. Για τη σχεδίαση του συστήματος αυτού προτιμήθηκε η σύγχρονη τεχνολογία πρακτόρων και πιο συγκεκριμένα μία διασκευή της μεθοδολογίας MaSE των συστημάτων πολλαπλών πρακτόρων που έχει αναπτύξει το Ινστιτούτο Τεχνολογίας Πολεμικής Αεροπορίας των Ηνωμένων Πολιτειών (AFIT). Η εργασία προσδιορίζει την αρχιτεκτονική του συστήματος, τους πράκτορες που πρέπει αυτό να περιέχει και τέλος πως αυτοί θα πρέπει να κατασκευαστούν ώστε να μπορούν να αποδίδουν αρμονικά μέσα σε ένα πληροφοριακό σύστημα.

Επίσης γίνεται αναφορά στην τεχνολογία των Έμπειρων Συστημάτων, μια τεχνολογία η οποία αποτέλεσε την αρχική προσέγγισή μας για την δόμηση και επίλυση του ΣΥΑ που απασχολεί αυτή την εργασία, που ωστόσο εγκαταλήφθηκε λόγω τεχνικών και οικονομικών περιορισμών που περιγράφονται σε επόμενο κεφάλαιο.

Σε αυτό το σημείο αξίζει να σημειώσουμε τις προσπάθειες που γίνονται από ιδιωτικούς και ακαδημαϊκούς φορείς για την ενσωμάτωση της Τεχνολογίας Πρακτόρων στην νοητική λειτουργία των πρακτόρων. Χαρακτηριστικότερο παράδειγμα αποτελεί η δυνατότητα ενσωμάτωσης πρακτόρων γραμμένων στην γλώσσα Έμπειρων Συστημάτων JESS στην πλατφόρμα Πολυπρακτορικών Συστημάτων JADE που αναπτύσσεται από την Ιταλική εταιρία TILAB.

## **1.1 Επισκόπηση των προσεγγίσεων του προβλήματος Ανάθεσης Εργασιών**

Η ανάθεση εργασιών στα συστήματα πολλαπλών πρακτόρων έχει απασχολήσει σε μεγάλο βαθμό τους επιστήμονες της Τεχνητής Νοημοσύνης.

Σύμφωνα με τους Zhang και Li, 1999, υπάρχουν τρία ξεχωριστά πεδία (εκτός από τους ίδιους τους πράκτορες) στα οποία έγκειται το πρόβλημα της ανάθεσης :

⇒ Το σύνολο των εργασιών,

⇒ οι πληροφορίες και

⇒ οι πόροι.

Οι εργασίες είναι το βασικό πρόβλημα που πρέπει να λυθεί. Μπορούμε να τις διακρίνουμε σε δύο κυρίως κατηγορίες : Σε ερωτήματα που πρέπει να απαντηθούν και σε ενέργειες που πρέπει να εκτελεστούν. Οι πληροφορίες είναι τα δεδομένα (είτε ντετερμινιστικά είτε δυναμικά) τόσο για τις εργασίες όσο και για το περιβάλλον που δρουν οι πράκτορες και οι πόροι είναι οτιδήποτε μπορεί να χρησιμοποιηθεί από τους πράκτορες για την εκπλήρωση των εργασιών.

Έτσι, σύμφωνα πάντα με τους Zhang και Li, μπορούμε να αντιμετωπίσουμε το πρόβλημα της ανάθεσης είτε με αποσύνθεση των εργασιών σε υπό-εργασίες, είτε με μοναδική ανάθεση εργασίας σε πράκτορα είτε ακόμα με πολλαπλή ανάθεση (μία εργασία σε περισσότερους από έναν πράκτορες ή περισσότερες από μία εργασίες σε έναν πράκτορα) πάντα βέβαια λαμβάνοντας υπόψιν τις απαιτήσεις των εργασιών και τις ικανότητες των πρακτόρων. Σε όλες αυτές τις περιπτώσεις είναι απαραίτητη η συνεργασία των πρακτόρων τόσο σε επίπεδο εργασιών όσο και στο επίπεδο ανταλλαγής πληροφοριών. Με δεδομένη βέβαια τη αυτονομία των πρακτόρων, είμαστε σχεδόν σίγουροι ότι οι πράκτορες θα δρουν βάση των προσωπικών τους συμφερόντων.

Ο Smith (1980) πρότεινε ένα πρωτόκολλο δικτύου συμβάσεων “Contract Net Protocol” που θα μπορούσε να υιοθετηθεί από τους πράκτορες. Σύμφωνα με αυτήν την προσέγγιση, οι πράκτορες διαπραγματεύονται σε πραγματικό χρόνο και αποφασίζουν για την ανάθεση των εργασιών. Όμως, εφόσον οι πράκτορες είναι αυτόνομοι, οι διαπραγματεύσεις μπορούν να πραγματοποιούνται σε πολύ ευρύ πεδίο με αποτέλεσμα να χρειάζεται αρκετός χρόνος για να έρθουν οι πράκτορες σε συμφωνία. Μία τέτοια εξέλιξη βέβαια, θα μπορούσε να ήταν καταστροφική καθώς ο χρόνος είναι από τα κρίσιμότερα χαρακτηριστικά στο πρόβλημα. Επιπλέον, δε πρέπει να παραβλέψουμε το γεγονός ότι αυτές οι διαπραγματεύσεις απαιτούν συνεχή επικοινωνία ανάμεσα στους πράκτορες που αυξάνει σημαντικά το κόστος.

Επιθυμώντας να μειωθεί το εύρος των διαπραγματεύσεων που προβλέπεται από το πρωτόκολλο δικτύου συμβάσεων οι Tidhar et al (1996) πρότειναν μία περισσότερο καθοδηγημένη διαδικασία επιλογής μιας ομάδας πρακτόρων με συμπληρωματικές ικανότητες για την επίτευξη ενός στόχου. Ο «εκπαιδευτής» είναι και ο δημιουργός του συστήματος πολλαπλών πρακτόρων γιατί αυτός γνωρίζει εκ των προτέρων όλους τους πράκτορες και τις ικανότητες που αυτοί διαθέτουν. Έτσι, κάθε φορά που πρέπει να πραγματοποιηθεί ένας στόχος, ο εκπαιδευτής τον διαιρεί σε μικρότερα τμήματα (υπό-στόχους) και για κάθε τέτοιο τμήμα καθορίζει τα στοιχεία που πρέπει να διαθέτει ένας πράκτορας (ή μια ομάδα πρακτόρων) ώστε να τον φέρει εις πέρας. Ο εκπαιδευτής καθορίζει δηλαδή τον τύπο των υποψηφίων για την ανάθεση της εργασίας πρακτόρων.



Αν κάποιος πράκτορας δεν ανήκει σε αυτόν τον τύπο, τότε δε παίρνει μέρος στη διαπραγμάτευση για την συγκεκριμένη αποστολή. Συνεπώς ο αριθμός των υποψηφίων για τη ανάθεση της κάθε εργασίας μειώνεται και η όλη διαδικασία γίνεται σημαντικά λιγότερο καταναλωτική σε χρόνο. Βεβαίως, οι πράκτορες από τη στιγμή που θα χριστούν «υποψήφιοι» αποκτούν και πάλι την αυτονομία τους.

Αυτή ακριβώς η αυτονομία που αποδίδεται στους πράκτορες από το Smith (1980) όσο και από τους Tidhar et al (1996) προκαλεί σε αρκετές των περιπτώσεων σοβαρά προβλήματα. Μια τέτοια περίπτωση περιγράφεται και αντιμετωπίζεται από τους Walsh και Wellman (1998). Σε αυτό το άρθρο, οι διάφοροι πράκτορες δρουν είτε ως προμηθευτές πρώτων υλών είτε ως παραγωγοί προϊόντων και ο σκοπός είναι κάποιος πράκτορας-καταναλωτής να αποκτήσει τα αγαθά που επιθυμεί. Κάθε πράκτορας στέλνει τακτικά μηνύματα προσφορών για τα αγαθά που επιθυμεί να αγοράσει ή να πουλήσει, διευκρινίζοντας επίσης και την τιμή κάτω / πάνω από την οποία προτίθεται να αγοράσει / πουλήσει. Κάθε φορά που φτάνει μία τέτοια προσφορά οι πράκτορες επιλέγουν αν θα αναθεωρήσουν τις δικές τους προσφορές. Οι πράκτορες μπορούν να απαντήσουν μόνο στην πλέον πρόσφατη προσφορά που υπάρχει στο σύστημα. Οι προσφορές συνεχίζονται μέχρι να ληφθούν όλα τα μηνύματα και κανένας πράκτορας να επιθυμεί περαιτέρω αναθεώρηση της προσφοράς του. Το πρόβλημα που προκύπτει, και που προφανώς ούτε η προσέγγιση του Smith μπορεί να αντιμετωπίσει, είναι ότι τα μηνύματα που αποστέλλουν οι πράκτορες πρέπει να είναι έγκυρα. Για παράδειγμα, δε πρέπει να επιτρέπεται σε κάποιον παραγωγό να διεκδικεί την πώληση των προϊόντων του, εάν δεν έχει πρώτα εξασφαλίσει την αγορά των απαραίτητων πρώτων υλών. Επιπλέον, επιβάλλεται οι προσφορές να ακολουθούν μία συγκεκριμένη οικονομική πολιτική. Για παράδειγμα, δε θα πρέπει ένας καταναλωτής να απαιτεί τη αγορά ενός αγαθού προσφέροντας λιγότερα χρήματα από την προκαθορισμένη τιμή του και φυσικά αυτή η τιμή πρέπει να επιτρέπει περιθώρια κέρδους σε προμηθευτές και παραγωγούς. Για να διασφαλισθούν λοιπόν όλα αυτά, είναι απαραίτητη η ύπαρξη πολιτικών προσφοράς που θα υιοθετούνται από κάθε πράκτορα του συστήματος. Αυτό σημαίνει ότι οι πράκτορες χάνουν μέρος της αυτονομίας τους και ακολουθούν αρκετούς αυστηρούς κανόνες.

Σε όλα τα παραπάνω παραδείγματα η ανάθεση των εργασιών πραγματοποιείται μετά από διαπραγμάτευση των ίδιων των υποψηφίων πρακτόρων. Μία διαφορετική προσέγγιση υποδεικνύει την ύπαρξη ενός πράκτορα-ηγέτη που θα αναλάβει τις ευθύνες για την ανάθεση των εργασιών στους υπόλοιπους πράκτορες-υπαλλήλους. Μία τέτοια προσέγγιση παρουσιάζεται από τους Fujita και Lesser (1996). Σ' αυτή την περίπτωση υπάρχει ένας στόχος που πρέπει να εκπληρωθεί πριν από μία ορισμένη προθεσμία και με ένα συγκεκριμένο επίπεδο ποιότητας. Αρχικά ο ηγέτης πράκτορας διαιρεί το στόχο σε μικρότερους υπό-στόχους. Εδώ βέβαια υπάρχει το ενδεχόμενο οι στόχοι να είναι απαραίτητο να εκτελεστούν διαδοχικά, με την έννοια πως τα αποτελέσματα του ενός να είναι αναγκαία για την εκπλήρωση του άλλου. Χρησιμοποιώντας ευρετικούς αλγορίθμους, ο συγγραφέας καθορίζει τη σειρά ανάθεσης και εκτέλεσης των υπό-στόχων.

Η ανάθεση των υποστόχων αυτών γίνεται με βάση ένα και μοναδικό κριτήριο, την ελαχιστοποίηση του χρόνου εκτέλεσης του κεντρικού στόχου. Πιο συγκεκριμένα, ένας υποστόχος θα ανατεθεί σε κάποιον πράκτορα αν και μόνο αν ο πράκτορας αυτός είναι ο πιο άμεσα διαθέσιμος από το σύνολο των πρακτόρων. Στην περίπτωση που η εκτέλεση του υποστόχου αυτού απαιτεί την ολοκλήρωση της εκτέλεσης κάποιου άλλου

υποστόχου, τότε και οι δύο αναθέτονται στον ίδιο πράκτορα. Για την περίπτωση που ο υποστόχος Y1 προαπαιτεί την εκπλήρωση περισσοτέρων του ενός άλλων υποστόχων Y2,Y3,... τότε ο πράκτορας που ανέλαβε το Y1 θα αναλάβει και έναν εκ των Y2,Y3,... και οι υπόλοιποι υποστόχοι θα ανατεθούν στους υπόλοιπους διαθέσιμους πράκτορες. Τέλος, οι Fujita και Lesser δίνουν τη δυνατότητα στον ηγέτη πράκτορα να ελέγχει και να εκτιμά περιοδικά το αρχικό πλάνο της ανάθεσης των υποστόχων-εργασιών ώστε να διασφαλίζονται οι χρονικές και ποιοτικές προδιαγραφές που προβλέπει ο κεντρικός στόχος.

## **1.2 Οργάνωση της εργασίας**

Η εργασία αυτή χωρίζεται σε πέντε κεφάλαια. Στο δεύτερο κεφάλαιο, δίνεται το θεωρητικό υπόβαθρο της εργασίας αυτής. Εκεί περιγράφονται τα Συστήματα Υποστήριξης Αποφάσεων τόσο ως προς τη δομή τους όσο και προς τη λειτουργία τους. Στη συνέχεια και πάντα στο ίδιο κεφάλαιο προσπαθούμε να αποσαφηνίσουμε τους όρους «πράκτορες» και «ευφυείς πράκτορες» παραθέτοντας ορισμούς, κατηγορίες και τις αρχιτεκτονικές τους. Το επόμενο βήμα είναι να εισχωρήσουμε στο πεδίο των Συστημάτων Πολλαπλών Πρακτόρων (Multi-Agent Systems) μελετώντας τη δομή τους και δύο μεθοδολογίες κατασκευής τους. Στο τέλος του δεύτερου κεφαλαίου περιγράφεται η τεχνολογία των Έμπειρων Συστημάτων, μια τεχνολογία η οποία όπως και η τεχνολογία των πρακτόρων προέρχεται από τον χώρο της Τεχνητής Νοημοσύνης. Η τεχνολογία των έμπειρων συστημάτων αποτέλεσε την πρώτη μας προσέγγιση για την δόμηση του ΣΥΑ, ωστόσο για λόγους που θα περιγράψουμε στη συνέχεια, αποδείχθηκε ακατάλληλη η υιοθέτησή της.

Στο τρίτο κεφάλαιο γίνεται παρουσίαση μιας πραγματικής εφαρμογής προσομοίωσης και κοστολόγησης της γραμμής παραγωγής ενός τυπογραφείου. Σε αυτό το στάδιο περιγράφεται η υφιστάμενη κατάσταση στην επιχείρηση, αναγνωρίζονται τα προβλήματα που καλείται να επιλύσει το ΣΥΑ και γίνεται μια αναφορά σε τεχνικά χαρακτηριστικά της παραγωγικής διαδικασίας των τυπογραφείων. Κλείνοντας το τρίτο κεφάλαιο, αναλύουμε την διαδικασία εκμαίευσης της απαιτούμενης γνώσης από τον αποφασίζοντα και γίνεται η μοντελοποίηση του προβλήματος.

Η σχεδίαση του συστήματος περιέχεται στο τέταρτο κεφάλαιο. Σε αυτό προτείνουμε τους πράκτορες που πρέπει να δημιουργηθούν ώστε να υλοποιούν το σύστημα υποστήριξης, τον τρόπο κατασκευής τους και προτείνουμε μία αρχιτεκτονική του συστήματος για τη σωστή και αποτελεσματική λειτουργία του.

Στο πέμπτο και τελευταίο κεφάλαιο παραθέτουμε τα συμπεράσματά μας και κάποιες σκέψεις για μελλοντικές βελτιώσεις του συστήματος.

Στο τέλος της εργασίας βρίσκονται οι βιβλιογραφικές αναφορές και το ευρετήριο σχημάτων και πινάκων.

## 2. Θεωρητικό Υπόβαθρο

### 2.1 Συστήματα Υποστήριξης Αποφάσεων (Σ.Υ.Α.)

Ο όρος «Συστήματα Υποστήριξης Αποφάσεων (Decision Support System) » εμφανίζεται σχεδόν ταυτόχρονα με τη δημιουργία της Επιστήμης των Αποφάσεων (Decision Science) ως ξεχωριστού επιστημονικού κλάδου στις αρχές του '70. Στην παράγραφο αυτή προσπαθούμε να περιγράψουμε τα Σ.Υ.Α. παραθέτοντας ορισμούς, συστατικά στοιχεία και λειτουργίες που χαρακτηρίζουν τα συστήματα αυτά.

#### 2.1.1 Ορισμοί – χαρακτηριστικά

Τα συστήματα υποστήριξης αποφάσεων (Σ.Υ.Α.) αποτελούν την εφαρμογή της επιστήμης των ηλεκτρονικών υπολογιστών ώστε να γίνει δυνατή η διευκόλυνση του αποφασίζοντα στη διαδικασία λήψης αποφάσεων. Τα Σ.Υ.Α. καλούνται να βοηθήσουν τον αποφασίζοντα να αντιμετωπίσει κακώς δομημένα προβλήματα και να τον οδηγήσουν στη βέλτιστη δυνατή λύση. Πρέπει να γίνει σαφές ότι τα Σ.Υ.Α. υποστηρίζουν τον αποφασίζοντα και δεν τον αντικαθιστούν.

Για τα Σ.Υ.Α. έχουν δοθεί πολλοί ορισμοί (Ματσατσίνης, 2000) βασισμένοι στη λειτουργία του συστήματος, στα χαρακτηριστικά επικοινωνίας με το χρήστη, στη διαδικασία ανάπτυξης, τα συστατικά και τις δυνατότητες του συστήματος. Το 1978 οι Keen και Scott-Morton ορίζουν εύστοχα τα Σ.Υ.Α. ως συστήματα που « συνδυάζουν τις διανοητικές ικανότητες των ανθρώπων με αυτές των ηλεκτρονικών υπολογιστών με αντικειμενικό στόχο τη βελτίωση της ποιότητας των λαμβανομένων αποφάσεων. Είναι (τα Σ.Υ.Α.) βασιζόμενα στους ηλεκτρονικούς υπολογιστές συστήματα υποστήριξης των αποφασιζόντων σε θέματα, που αφορούν την αντιμετώπιση ημιδομημένων προβλημάτων.» ενώ λίγο παλιότερα (1970) ο Little περιέγραφε περιεκτικά πως « ένα Σ.Υ.Α. είναι ένα σύνολο διαδικασιών για την επεξεργασία των δεδομένων και κρίσεων με στόχο να υποβοηθήσουν τους μάντζερ στη διαδικασία λήψης μιας απόφασης. Ένα τέτοιο σύστημα θεωρείται επιτυχημένο αν είναι : απλό, εύρωστο, προσαρμόσιμο, πλήρες, εύκολο στη χρήση και με καλό σύστημα επικοινωνίας με το χρήστη.»

Οι παραπάνω ορισμοί φαίνεται να είναι πλήρεις, όμως το πραγματικά μεγάλο εύρος χρήσης των Σ.Υ.Α. και ο συνδυασμός διαφορετικών εννοιών (σύστημα, υποστήριξη, απόφαση) που ενσωματώνουν δεν τους επιτρέπουν να είναι και καθολικά αποδεκτοί. Μπορούμε όμως να αποσαφηνίσουμε περαιτέρω την έννοια των Σ.Υ.Α. αναφέροντας τα χαρακτηριστικά τους. Σύμφωνα λοιπόν με τους Sprague και Carlson (1982) τα σημαντικότερα χαρακτηριστικά ενός αποτελεσματικού Σ.Υ.Α. είναι οι δυνατότητες που αυτά έχουν για να :

- ⇒ Υποστηρίζουν τις διαδικασίες λήψης ημιδομημένων ή αδόμητων αποφάσεων.
- ⇒ Υποστηρίζουν και τις τέσσερις φάσεις λήψης αποφάσεων (νοητική, σχεδίασης, επιλογής και ολοκλήρωσης).

- ⇒ Συνδυάζουν τη συνεργασία μοντέλων, βάσεων δεδομένων και τεχνικών παρουσίασης των αποτελεσμάτων.
- ⇒ Δίνουν έμφαση στην ευκολία χρήσης, την ευελιξία και την προσαρμοστικότητα τους.
- ⇒ Αλληλεπιδρούν με άλλα πληροφοριακά συστήματα που ήδη λειτουργούν.

Πολλές φορές τα Σ.Υ.Α συγχέονται με τα πληροφοριακά συστήματα διοίκησης (M.I.S.). Η λειτουργία και ο σκοπός των Σ.Υ.Α. όμως είναι σαφώς διαφορετική από αυτόν των M.I.S. . Συνοπτικά, τα χαρακτηριστικά που τα διαφοροποιούν είναι :

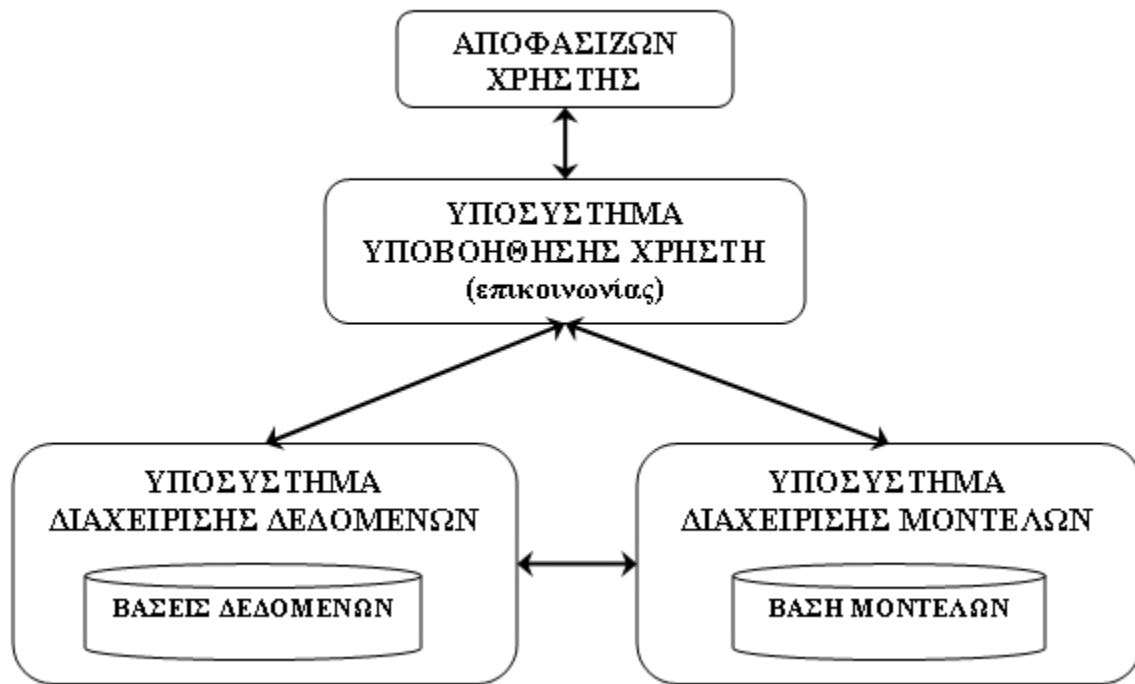
- ⇒ Η χρήση τους βοηθά στην αύξηση της αποτελεσματικότητας και όχι της αποδοτικότητας ( περίπτωση M.I.S.)
- ⇒ Κατασκευάζονται για να παρέχουν υποστήριξη σε όλα τα επίπεδα διοίκησης βοηθώντας και όχι υποκαθιστώντας τον αποφασίζοντα ο οποίος διατηρεί τον πλήρη έλεγχο του συστήματος καθόλη τη διάρκεια λήψης μιας απόφασης.
- ⇒ Η υποστήριξη παρέχεται είτε σε ομάδες ατόμων ( Group Decision Support Systems) είτε σε ξεχωριστά άτομα (Σ.Υ.Α.)
- ⇒ Παρέχουν υποστήριξη σε όλες τις φάσεις μιας διαδικασίας λήψης απόφασης. Διαθέτουν ευελιξία και προσαρμοστικότητα.
- ⇒ Χρησιμοποιούνται κυρίως για να παρέχουν υποστήριξη σε ημιδομημένες ή αδόμητες αποφάσεις.
- ⇒ Είναι φιλικά προς το χρήστη.
- ⇒ Είναι ευέλικτα και ευπροσάρμοστα σε τυχόν μεταβολές
- ⇒ Η χρήση ενός Σ.Υ.Α. βελτιώνει και το ίδιο το Σ.Υ.Α. καθώς προσαρμόζεται στις απαιτήσεις του χρήστη και αναπτύσσεται ανάλογα με τη χρήση του.

### 2.1.2 Δομή Σ.Υ.Α.

Ένα σύστημα υποστήριξης αποφάσεων έχει τη δομή που περιγράφεται στο παρακάτω σχήμα (Sprague και Carlson,1982) :

Υποσύστημα διαχείρισης δεδομένων : Σε αυτό το τμήμα συγκεντρώνονται και αποθηκεύονται τα δεδομένα από εξωτερικές και εσωτερικές πηγές . Όλες οι απαραίτητες για τη λειτουργία του συστήματος πληροφορίες βρίσκονται εδώ, και αυτό το υποσύστημα προσφέρει τη δυνατότητα εισαγωγής, αποθήκευσης, πρόσβασης, ενημέρωσης και γενικότερα διαχείρισης των δεδομένων. Για την αναπαράσταση, την οργάνωση και την αποθήκευση των δεδομένων χρησιμοποιούνται διάφορες τεχνικές :

- ⇒ Αναπαράσταση δεδομένων με τη μορφή εγγραφών (records). Τα δεδομένα αποτελούν ξεχωριστές εγγραφές, κάθε μια από τις οποίες αποτελείται από ένα σύνολο πεδίων.
- ⇒ Σχεσιακή αναπαράσταση των δεδομένων . Τα δεδομένα καταχωρούνται σε πεδία που συνδέονται μεταξύ τους.
- ⇒ Ιεραρχική αναπαράσταση των δεδομένων . Τα δεδομένα δομούνται ιεραρχικά από τα πιο γενικά προς τα πιο ειδικά.



Εικόνα 1: Η πρότυπη δομή ενός ΣΥΑ

⇒ Αναπαράσταση μέσω δικτύου . Αποτελεί γενίκευση της ιεραρχικής δομής . Το δίκτυο των δεδομένων αποτελείται από ένα σύνολο εγγραφών οι οποίες συνδέονται μεταξύ τους μέσω συνδέσμων (links).

Υποσύστημα διαχείρισης μοντέλων : Εδώ πέρα υπάρχουν ή δημιουργούνται όλες οι μέθοδοι, οι τεχνικές και τα μοντέλα ανάλυσης του προβλήματος που διαπραγματεύεται το Σ.Υ.Α. Επικοινωνεί με το υποσύστημα διαχείρισης δεδομένων ώστε να αντλεί τα απαραίτητα για την ανάλυση δεδομένα. Οι βασικές λειτουργίες που επιτελεί το υποσύστημα αυτό είναι :

- ⇒ Ανάπτυξη μοντέλων.
- ⇒ Επανακαθορισμός των υπαρχόντων μοντέλων
- ⇒ Ενημέρωση των μοντέλων.

Υποσύστημα υποβοήθησης χρήστη : Πραγματοποιεί την επικοινωνία με το χρήστη. Περιέχει το interface του συστήματος και η ευχρηστία και η εργονομία του επηρεάζουν την απόδοση και τη λειτουργία του συστήματος. Οι τεχνικές που χρησιμοποιούνται περισσότερο είναι :

- ⇒ Επικοινωνία με τη μορφή ερωτήσεων –απαντήσεων
- ⇒ Επικοινωνία με τη μορφή εντολών.
- ⇒ Επικοινωνία μέσω μενού επιλογών.

### **2.1.3 Ευφυή Συστήματα Υποστήριξης Αποφάσεων (Intelligent DSS)**

Τα ευφυή συστήματα υποστήριξης αποφάσεων ( ή αλλιώς βασιζόμενα στη γνώση συστήματα υποστήριξης αποφάσεων) προκύπτουν από τα απλά Σ.Υ.Α. αν σε αυτά ενσωματώσουμε κάποιο μηχανισμό γνώσης. Τα Σ.Υ.Α. αποκτούν με αυτόν τον τρόπο ικανότητα λογικής, έτσι μπορούν εκτός από την πληροφόρηση να παρέχουν και μεθοδολογική γνώση. Το ευφυές πια Σ.Υ.Α. είναι σε θέση να χρησιμοποιήσει τα μοντέλα απόφασης και τους μηχανισμούς γνώσης ώστε να προτείνει λύσεις υποστηρίζοντας πάντα ( και όχι αντικαθιστώντας !) τον αποφασίζοντα. Ο μηχανισμός γνώσης που θα ενσωματωθεί στο Σ.Υ.Α. αλληλεπιδρά με όλα τα υποσυστήματα του, έχοντας πάντα σαν αντικειμενικό στόχο τη λήψη βέλτιστης απόφασης από τον αποφασίζοντα .

Ένα τέτοιο ευφυές σύστημα υποστήριξης αποφάσεων φιλοδοξεί να υλοποιηθεί και ο γράφοντας στη συγκεκριμένη διπλωματική εργασία . Βασισμένο λοιπόν στην τεχνολογία πρακτόρων και αναθέτοντας το ρόλο των υποσυστημάτων του Σ.Υ.Α. στους πράκτορες το σύστημα υποστήριξης απόφασης για την ανάθεση εργασιών παρουσιάζεται στα επόμενα κεφάλαια.

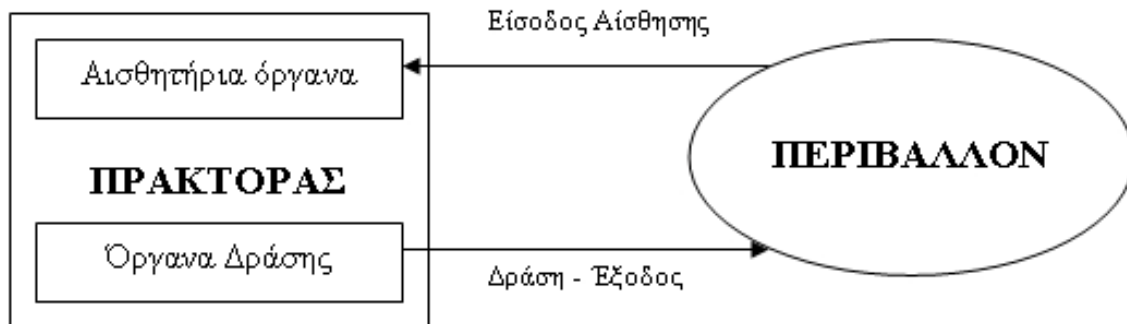
## **2.2 Πράκτορες και συστήματα πολλαπλών πρακτόρων (Agents & Multi-Agent Systems)**

Η έννοια του «πράκτορα» έχει αρχίσει και γίνεται ιδιαίτερα σημαντική στον τομέα της τεχνητής νοημοσύνης και ιδίως στην κατανεμημένη τεχνητή νοημοσύνη. Η παράγραφος αυτή προσπαθεί να προσδιορίσει τον όρο «πράκτορας» (agent) επιχειρώντας να δώσει απάντηση στα ερωτήματα «τί είναι πράκτορας ;» και «ποία χαρακτηριστικά πρέπει να διαθέτει ένας πράκτορας;». Στη συνέχεια αναφέρονται οι αρχιτεκτονικές σχεδίασης πρακτόρων . Όταν οι πράκτορες συνεργάζονται σε ένα σύστημα τότε αυτό το σύστημα χαρακτηρίζεται πολυπρακτορικό (Multiagent). Σε ένα τέτοιο σύστημα θα προσπαθήσουμε να αποδώσουμε ιδιότητες και θα αναφέρουμε τις μεθοδολογίες που έχουν αναπτυχθεί για τη σχεδίαση του.

### **2.2.1 Πράκτορες και ευφυείς πράκτορες (Agents & Intelligent Agents)**

Η έννοια του «πράκτορα» χρησιμοποιείται σε αυτή την εργασία με την έννοια που έχει δημιουργήσει η αντίληψη για τον όρο στα επιστημονικά πεδία της τεχνητής νοημοσύνης και των συστημάτων υποστήριξης αποφάσεων. Ωστόσο, όσο και αν ο όρος «πράκτορας» εδραιώνεται ως γνωστικό αντικείμενο στα παραπάνω πεδία και χρησιμοποιείται ευρέως πλέον, δεν υπάρχει κάποιος συγκεκριμένος και καθολικά αποδεκτός ορισμός της έννοιας.

Περιγράφοντας όμως τα χαρακτηριστικά που θα πρέπει να διαθέτει ένας πράκτορας, είναι δυνατόν να σκιαγραφήσουμε την έννοια που αυτός αντιπροσωπεύει :



**Εικόνα 2:** Ένας πράκτορας σε αλληλεπίδραση με το περιβάλλον του. Ο πράκτορας παίρνει αισθητήρια είσοδο από το περιβάλλον του και παράγει σαν έξοδο δράσεις που το επηρεάζουν (Πηγή : Multiagent Systems, MIT Press, 1999).

Στο παραπάνω σχήμα παρουσιάζεται η πλέον γενική άποψη για τον «πράκτορα»: Αυτός αποτελείται από ένα σύνολο αισθητηρίων οργάνων με τα οποία αντιλαμβάνεται το περιβάλλον του και από ένα σύνολο οργάνων δράσης με των οποίων τη χρήση προσπαθεί να πραγματοποιήσει τις λειτουργίες που ορίζει ο σκοπός της δημιουργίας του. Αναφερόμαστε όμως στο «σκοπό δημιουργίας» του πράκτορα, χωρίς να έχουμε ξεκαθαρίσει αν η ύπαρξη ενός πράκτορα συνεπάγεται και κάποιον συγκεκριμένο σκοπό-στόχο. Αντιλαμβανόμαστε όμως, πως ο στόχος, σκοπός σχεδίασης ενός πράκτορα προηγείται της δημιουργίας του, γεγονός που οδηγεί τους M.Wooldridge και N.Jennings στον εξής ορισμό :

*«Ένας πράκτορας είναι ένα υπολογιστικό σύστημα που είναι εγκατεστημένο σε κάποιο περιβάλλον και έχει την ικανότητα αυτόνομης δράσης στο περιβάλλον προκειμένου να ανταποκριθεί στους σκοπούς σχεδίασης του.»*

Ο παραπάνω ορισμός εισάγει δύο νέες έννοιες για τη φύση του πράκτορα, την αυτονομία και το περιβάλλον, χωρίς όμως να προσδιορίζει ο ίδιος τις έννοιες αυτές. Για την έννοια της «αυτονομίας» είναι εξαιρετικά δύσκολο να δώσουμε κάποιον ακριβή ορισμό, μπορούμε μόνο να πούμε με γενικό τρόπο ότι στην αυτονομία του πράκτορα έγκειται η ικανότητα του να δρα χωρίς την παρέμβαση ανθρώπων ή άλλων συστημάτων, να δέχεται εξωτερικά ερεθίσματα, να διαμορφώνει την κατάσταση του και να προσδιορίζει τη συμπεριφορά του. Το περιβάλλον είναι και αυτό εξαιρετικά πολύπλοκη έννοια και μπορεί να διακρίνεται από τα εξής χαρακτηριστικά :

⇒ Προσβάσιμο ή Μη Προσβάσιμο.

Ένα περιβάλλον χαρακτηρίζεται προσβάσιμο όταν οι πράκτορες που δραστηριοποιούνται σε αυτό διαθέτουν απόλυτη και ακριβή πρόσβαση σε ενημερωμένες πληροφορίες σχετικά με την κατάσταση του περιβάλλοντος. Τα παραδείγματα από την καθημερινή μας ζωή (φυσικό – κοινωνικό περιβάλλον, Internet) αποτελούν μη προσβάσιμα περιβάλλοντα. Οι πράκτορες αναπτύσσονται τόσο πιο εύκολα σε ένα περιβάλλον όσο περισσότερο προσβάσιμο είναι αυτό.

- ⇒ Ντετερμινιστικό ή Μη Ντετερμινιστικό.  
Ντετερμινιστικό θεωρείται το περιβάλλον στο οποίο μπορούμε να προβλέψουμε τις επιδράσεις και τις συνέπειες που θα προκληθούν από μια δράση μας. Η κατάσταση του περιβάλλοντος προσδιορίζεται με σαφήνεια τόσο πριν όσο και μετά τη δράση μας. Εύκολα καταλαβαίνουμε πως τα μη ντετερμινιστικά περιβάλλοντα παρουσιάζουν και τις μεγαλύτερες δυσκολίες για τους σχεδιαστές πρακτόρων. Ο φυσικός κόσμος αποτελεί ίσως το πιο χαρακτηριστικό παράδειγμα μη ντετερμινιστικού περιβάλλοντος.
- ⇒ Επεισοδιακό ή Μη Επεισοδιακό.  
Σε ένα επεισοδιακό περιβάλλον η επίδοση-δράση του πράκτορα εξαρτάται από μια σειρά διακριτών επεισοδίων (events), χωρίς η επίδοση αυτή να συνδέεται με κάποια από τις επιδόσεις ενός πράκτορα σε διαφορετικά σενάρια. Ένα σύστημα ταξινόμησης ηλεκτρονικής αλληλογραφίας θα μπορούσε να είναι ένα παράδειγμα επεισοδιακού περιβάλλοντος. Τα επεισοδιακά περιβάλλοντα διευκολύνουν την ανάπτυξη πρακτόρων γιατί ο κάθε πράκτορας αναπτύσσει συγκεκριμένη δράση με βάση το τρέχων επεισόδιο.
- ⇒ Στατικό ή Δυναμικό.  
Στατικό περιβάλλον είναι αυτό που παραμένει σταθερό (χωρίς αλλαγές της κατάστασης του) όσο κανένας πράκτορας δεν δραστηριοποιείται. Ένα δυναμικό περιβάλλον περιέχει και άλλες διαδικασίες που λειτουργούν σε αυτό και είναι δυνατόν να το μεταβάλλουν ανεξάρτητα από τις διαδικασίες των πρακτόρων. Και πάλι ο φυσικός κόσμος είναι το πλέον χαρακτηριστικό παράδειγμα.
- ⇒ Διακριτό ή Συνεχές  
Ένα περιβάλλον είναι διακριτό όταν οι αντιλήψεις και οι κινήσεις δράσης των πρακτόρων είναι πεπερασμένες. Μια παρτίδα σκάκι είναι παράδειγμα διακριτού περιβάλλοντος ενώ η οδήγηση ενός αυτοκινήτου συμβαίνει σε συνεχές περιβάλλον.

Σύμφωνα με όσα περιγράψαμε παραπάνω ως πράκτορας μπορεί να θεωρηθεί κάθε σύστημα ελέγχου! Το πλέον κλασσικό παράδειγμα της βιβλιογραφίας είναι ο θερμοστάτης-πράκτορας. Ο θερμοστάτης λοιπόν, χρησιμοποιεί έναν αισθητήρα για να αντιλαμβάνεται τη θερμοκρασία του περιβάλλοντος (δωματίου). Έτσι ο θερμοστάτης μπορεί να δράσει ανάλογα με τα ερεθίσματα που δέχεται από το περιβάλλον να ανοίξει δηλαδή τη θέρμανση αν αντιληφθεί ότι η θερμοκρασία του δωματίου είναι χαμηλή ή αντίστροφα να κλείσει τη θέρμανση αν θεωρεί ικανοποιητική τη θερμοκρασία του δωματίου. Όμως ο συγκεκριμένος πράκτορας δε μπορεί να διακρίνει αν η πόρτα του δωματίου είναι ανοιχτή ώστε να πετύχει καλύτερα το στόχο του, τη θέρμανση του δωματίου, δε διακρίνεται δηλαδή για την ευφυΐα του! Εδώ προκύπτει ένα νέο ερώτημα : *«Τί είναι ευφυής πράκτορας και πότε ένας πράκτορας χαρακτηρίζεται ευφυής ; »*. Όπως και προηγουμένως, διαπιστώνουμε πως ένας αυστηρός, ακριβής και καθολικά αποδεκτός ορισμός δεν έχει δοθεί ακόμη. Άλλωστε, και το ερώτημα *«τί είναι ευφυΐα ;»* περιμένει ακόμα να βρει τη δική του απάντηση...

Εμείς θα περιμέναμε από μία ευφυή οντότητα, όπως οι πράκτορες εν προκειμένω, να έχει τη δυνατότητα να αντιλαμβάνεται το περιβάλλον του, να κρίνει τα στοιχεία που συλλέγει, να αποφασίζει το βέλτιστο σχέδιο δράσης και να το πραγματοποιεί, ωστόσο επειδή οι απαιτήσεις μας προσεγγίζουν μία ιδανική οντότητα, θα εξετάσουμε κάποιες πιο



ρεαλιστικές διατυπώσεις. Συγκεκριμένα, οι M. Wooldridge και N.R. Jennings (1995) προτείνουν ως ευφυή πράκτορα αυτόν που έχει την ικανότητα ελαστικής αυτονομίας δράσης προκειμένου να διεκπεραιώσει τους σκοπούς σχεδίασης, ορίζοντας σαν στοιχεία της ελαστικής αυτονομίας δράσης, της ευελιξίας δηλαδή που διαθέτουν οι πράκτορες, τα παρακάτω:

- ⇒ Αντιδραστικότητα (reactivity) : Οι ευφυείς πράκτορες (ΕΠ) είναι ικανοί να αντιλαμβάνονται το περιβάλλον και να ανταποκρίνονται με έγκαιρη διαμόρφωση στις αλλαγές που προκύπτουν ώστε να αντεπεξέλθουν στις απαιτήσεις της σχεδίασης τους
- ⇒ Προνοητικότητα (pro-activeness) : Οι ΕΠ είναι ικανοί να επιδεικνύουν συμπεριφορά προσανατολισμένη στον σκοπό τους με δικές τους πρωτοβουλίες.
- ⇒ Κοινωνική Συμπεριφορά (social ability) : Οι ΕΠ είναι ικανοί να διεπικοινωνούν με άλλους πράκτορες (και πιθανόν ανθρώπους), προκειμένου να φτάσουν στους στόχους τους.

Με αυτά τα χαρακτηριστικά να φαίνονται ως τα κυρίαρχα γνωρίσματα ενός πράκτορα ο Μωραΐτης (2001) ορίζει τους πράκτορες «ως δραστήριες, επίμονες (persistent) (λογισμικές ή φυσικές) οντότητες, οι οποίες αντιλαμβάνονται, συλλογίζονται, δρουν και επικοινωνούν ενώ από μία άλλη άποψη μπορούν να θεωρηθούν σαν συστήματα H/Y, τοποθετημένα σε ένα περιβάλλον εκτελώντας δράσεις που μπορούν να αλλάξουν αυτό το περιβάλλον ». Θεωρεί επίσης ότι είναι δυνατόν να αποδώσουμε στους πράκτορες ανθρωποκεντρικές έννοιες όπως ευφυΐα (intelligence), πιστεύω (beliefs), επιθυμίες (desires), προθέσεις (intentions), γνώσεις, δεσμεύσεις (commitments), συγκίνηση (emotion), αλτρουισμό, σεβασμό της αλήθειας, ανιδιοτέλεια (benevolence), λογικότητα (rationality).

Βέβαια, τα παραπάνω χαρακτηριστικά, είναι δυνατό, κάποιοι πράκτορες να τα επαληθεύουν ή όχι, να είναι κρίσιμα για τη λειτουργία τους ή όχι, να αποτελούν ή όχι βασικά στοιχεία της σχεδίασης τους, καταλήγοντας έτσι σε τρεις βασικές κατηγορίες πρακτόρων (Μωραΐτης, 2001) ανάλογα με τις αντιλήψεις τους για τον κόσμο, τους άλλους πράκτορες και τις δράσεις τους :

- ⇒ Διαβουλευτικοί (Deliberative) ή Λογικοί (Rational) Πράκτορες.  
Ως Διαβουλευτικοί χαρακτηρίζονται οι πράκτορες που σχεδιαστήκανε για να δραστηριοποιούνται μέσω νοητικών λειτουργιών ( π.χ. συλλογισμού, λήψη απόφασης, επικοινωνίας, οργάνωσης, συνεργασίας) βασισμένων (όχι απαραίτητα) σε διανοητικές καταστάσεις όπως πιστεύω (beliefs), επιθυμίες (desire), προθέσεις (intentions)(BDI πράκτορες). Έτσι ένας διαβουλευτικός πράκτορας έχει τη δυνατότητα να επιλέξει με ορθολογικό τρόπο την εκάστοτε καταλληλότερη λύση.
- ⇒ Βασισμένοι στο Περιβάλλον ή Αντιδραστικοί (Reactive) Πράκτορες.  
Οι αντιδραστικοί πράκτορες αποφασίζουν βασιζόμενοι απευθείας σε αισθητήρια δεδομένα, στα ερεθίσματα που δέχονται από το περιβάλλον, ουσιαστικά δηλαδή στις αλλαγές στο περιβάλλον ή σε μηνύματα άλλων πρακτόρων. Οι πληροφορίες που δέχονται είναι περιορισμένες και δεν φιλτράρονται από κάποια νοητική διεργασία ενώ οι αποφάσεις και οι δράσεις συμβαίνουν σε τρέχοντα χρόνο (runtime).

⇒ Υβριδικοί (Hybrid) Πράκτορες.

Είναι πράκτορες που προκύπτουν από τους συνδυασμούς των παραπάνω κατηγοριών και έχουν τη δυνατότητα να ενεργούν ενίοτε ως διαβουλευτικοί και ενίοτε ως αντιδραστικοί πράκτορες.

## 2.2.2 Αρχιτεκτονικές Ευφυών Πρακτόρων

Η αρχιτεκτονική ενός πράκτορα καθορίζει τη σχεδίαση του. Από αυτήν θα προκύψουν τα δομικά του στοιχεία και πώς αυτά θα αλληλεπιδρούν. Η αρχιτεκτονική ενός πράκτορα πρέπει να καθορίζει τον τρόπο με τον οποίο αυτός θα επικοινωνεί με το περιβάλλον, πώς θα επεξεργάζεται τα στοιχεία που συλλέγει, πώς θα δρα και τι αλλαγές θα επιφέρει στο περιβάλλον του. Οι κυριότερες και πλέον διαδεδομένες αρχιτεκτονικές προσεγγίσεις είναι (αναφέρονται χωρίς ιδιαίτερη ανάλυση) :

⇒ Διαβουλευτικοί ή Λογικοί Πράκτορες.

Η λήψη αποφάσεων δράσης γίνεται μέσω λογικής επαγωγής. Στο σχήμα φαίνεται πως η δράση του πράκτορα είναι μία συνάρτηση που παίρνει δεδομένα από την κατάσταση του πράκτορα και πως οι νοητικές διεργασίες του πράκτορα (Συλλογισμός) ενημερώνουν και ενημερώνονται από την κατάστασή του.

⇒ Αντιδραστικοί (Reactive) Πράκτορες.

Οι αποφάσεις – δράσεις αποτελούν αντίδραση στα εξωτερικά ερεθίσματα. Το περιβάλλον περιγράφεται μέσω της αντίληψης και οδηγεί τον πράκτορα στην επιλογή μιας δράσης που επενεργεί απευθείας στο περιβάλλον.

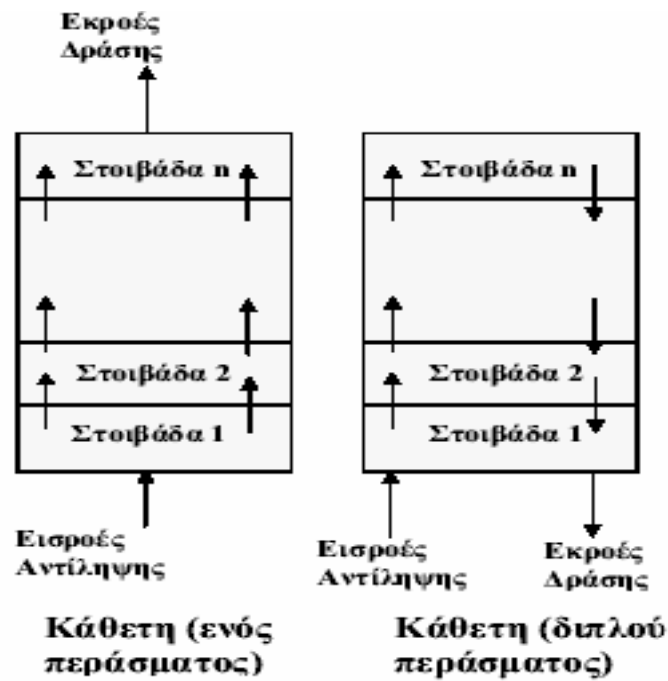
⇒ Πράκτορες που βασίζονται στο τρίπτυχο Πιστεύω – Επιθυμίες – Προθέσεις (BDI Agents)

Η λήψη των αποφάσεων – επιλογή δράσεων γίνεται με τη διαχείριση ειδικών δομών που αντιπροσωπεύουν τα πιστεύω, τις επιθυμίες και τις προθέσεις του πράκτορα.

⇒ Πράκτορες σε Διαστρωματοποιημένη Αρχιτεκτονική (Layered Agents)

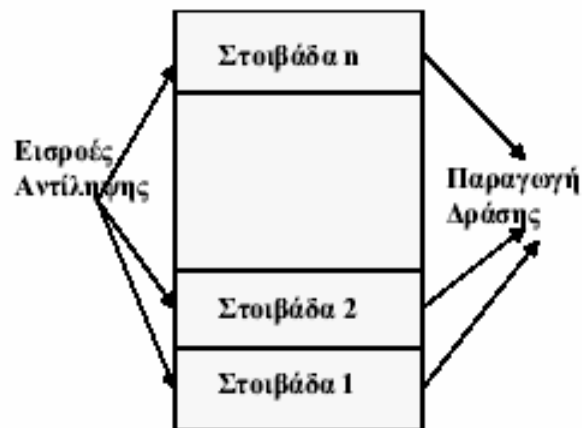
Η λήψη απόφασης πραγματοποιείται μέσω διαμελισμού του πράκτορα σε διάφορες στοιβάδες, έτσι κάθε στοιβάδα μπορεί και λειτουργεί σε ένα διαφορετικό επίπεδο αφαίρεσης. Οι στοιβάδες επικοινωνούν είτε με το περιβάλλον είτε μεταξύ τους με αλληλεπιδραστικό τρόπο. Μπορούμε να κατατάξουμε τις στρωματοποιημένες αρχιτεκτονικές σε δύο βασικούς τύπους :

- Κάθετες στοιβάδες : Μόνο μία στοιβάδα διαθέτει τα αισθητήρια όργανα και αντιλαμβάνεται το περιβάλλον και μόνο μία (η ίδια ή διαφορετική) διοχετεύει τις εκροές της δράσης του πράκτορα προς το περιβάλλον.



Εικόνα 3: Κάθετα στρωματοποιημένη αρχιτεκτονική.

- Οριζόντιες στοιβάδες : Σε αυτήν την περίπτωση κάθε στοιβάδα συνδέεται με το περιβάλλον, τόσο σε επίπεδο αντίληψης του όσο και με τις εκροές της δράσης .



Εικόνα 4: Οριζόντια Στρωματοποιημένη Αρχιτεκτονική πράκτορα

### 2.2.3 Βασικοί Τύποι Πρακτόρων

Καταλήγοντας, παραθέτουμε τους βασικούς τύπους πρακτόρων ανάλογα με τη χρησιμότητα και τη λειτουργία τους (Μωραΐτης, 2001).

- ⇒ Πράκτορες λογισμικού (software agents)
- ⇒ Προσωπικοί βοηθοί (personal assistants) : Μαθαίνουν από τον χρήστη, μέσω της συνεργασίας τους, να χρησιμοποιούν συγκεκριμένες εφαρμογές (π.χ. ένα λειτουργικό σύστημα ή ένα λογισμικό πίνακα) προς όφελος του χρήστη.
- ⇒ Πράκτορες εργασιών (task agents) : Αναλαμβάνουν την υλοποίηση συγκεκριμένων εργασιών που τους αναθέτονται είτε από άλλους πράκτορες είτε από τον χρήστη.
- ⇒ Πράκτορες διεπαφής (interface agents) : Συνήθως είναι οι διαμεσολαβητές μεταξύ του χρήστη και των άλλων πρακτόρων σε ένα πολυπρακτορικό σύστημα (multiagent) .
- ⇒ Πληροφοριακοί πράκτορες (information agents): Διαχειρίζονται ή απλά συλλέγουν πληροφορίες από πολλές κατανεμημένες πηγές (π.χ. βάσεις δεδομένων, διαδίκτυο)
- ⇒ Κινητοί πράκτορες (mobile agents): Κινούνται μέσα στο διαδίκτυο ή κάποιο δίκτυο υπολογιστών, αλληλεπιδρούν με υπολογιστικά συστήματα σε απόσταση, συλλέγουν πληροφορίες ή εκτελούν λειτουργίες και επιστρέφουν στη βάση τους.
- ⇒ Φυσικοί πράκτορες (physical agents): Ρομπότ

### 2.2.4 Επιπλέον Ορισμοί

Παραπάνω παρατηρήσαμε ότι η επιστημονική κοινότητα δεν έχει καταλήξει σε ένα ακριβή ορισμό για την έννοια του ευφυούς πράκτορα. Το ευρύ πεδίο εφαρμογής της τεχνολογίας των πρακτόρων καθώς και ο τρόπος χρήσης της από τα διάφορα επιστημονικά πεδία καθιστά ακόμα πιο δύσκολη την προσπάθεια αποσαφήνισης του όρου. Παρακάτω παραθέτουμε τους ορισμούς που έχουν προσδώσει στον όρο ευφυείς πράκτορες, γνωστοί ερευνητές του χώρου.

«Ένας ευφυής πράκτορας είναι λογισμικό που μπορεί να κάνει πράγματα που πιθανότητα θα έκανε ο χρήστης, αν είχε το χρόνο.» (Ted Selker, IBM Almaden Research Centre)

«Ευφυής Πράκτορας είναι ένα τμήμα λογισμικού, το οποίο προκειμένου να κινηθεί κατάλληλα για την επιτυχή διεκπεραίωση μιας δοσμένης εργασίας, χρησιμοποιεί πληροφορίες που έχει αποκτήσει από το περιβάλλον του.» (Lecky – Thompson)

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

Παρατηρώντας τους παραπάνω ορισμούς αντιλαμβανόμαστε το ευρύ πεδίο χαρακτηριστικών που μπορεί να φέρει ένας πράκτορας. Ωστόσο το κύριο συμπέρασμα το

οποίο εξάγεται από την παραπάνω μελέτη, είναι ότι οι πράκτορες διαφέρουν από το κοινό λογισμικό, κυρίως στην αντίληψη του ίδιου του εαυτού τους σαν ανεξάρτητες οντότητες. Ένας ιδανικός πράκτορας γνωρίζει ποιος είναι ο σκοπός του και αγωνίζεται για την επίτευξη του. Επίσης αποτελεί εύρωστη και προσαρμόσιμη οντότητα, ικανή να μαθαίνει από την εμπειρία που αποκτά και να ανταποκρίνεται σε απρόβλεπτες καταστάσεις, βάσει διαφόρων δυναμικών μεθόδων. Τέλος ένας πράκτορας θα πρέπει να είναι αυτόνομος, έτσι ώστε να μπορεί να αντιλαμβάνεται την τρέχουσα κατάσταση του περιβάλλοντός του και να δρα ανεξάρτητα, προκειμένου να σημειώσει πρόοδο στην επίτευξη του στόχου του. (N. Ματσατσίνης, N. Σπανουδάκης, Α. Σαμαράς –2005)

### **2.3 Συστήματα πολλαπλών πρακτόρων (multi-agent systems)**

Σε πολλές περιπτώσεις σε ένα σύστημα συνυπάρχουν περισσότεροι του ενός πράκτορες . Ένα τέτοιο σύστημα το ονομάζουμε σύστημα πολλαπλών πρακτόρων (ΣΠΠ). Σε ένα ΣΠΠ οι πράκτορες μπορούν να είναι ομογενείς ή και ετερογενείς, να συνεργάζονται ή να ανταγωνίζονται, να επικοινωνούν με διάφορους τρόπους και να δρουν σε οποιοδήποτε περιβάλλον. Για να λειτουργεί ένα τέτοιο σύστημα σωστά και αποτελεσματικά πρέπει σύμφωνα με τους Corkill και Lesser (1983) να διαθέτει τις παρακάτω ιδιότητες :

- ⇒ Κάλυψη : Κάθε κομμάτι του συνολικού προβλήματος πρέπει να είναι μέσα στις ικανότητες τουλάχιστον ενός πράκτορα.
- ⇒ Συνδεσιμότητα : Οι πράκτορες πρέπει να επικοινωνούν και να αλληλεπιδρούν με τέτοιο τρόπο ώστε οι δράσεις τους να μπορούν να συντεθούν σε μία ολοκληρωμένη λύση.
- ⇒ Ικανότητα : Η λύση πρέπει να είναι εφικτή μέσα στους περιορισμούς που ορίζει το περιβάλλον και οι πόροι του συστήματος . Η κάλυψη και η συνδεσιμότητα πρέπει να είναι δυνατές στο σύνολο που ορίζουν τα υπολογιστικά και επικοινωνιακά περιθώρια και η προδιαγραφές των πρακτόρων.

Ένα ΣΠΠ για να λειτουργήσει σωστά πρέπει να έχει σαφώς καθορισμένη τη δομή του. Δηλαδή πρέπει να δηλώνεται με σαφή τρόπο η κατανομή των δραστηριοτήτων και των ρόλων των πρακτόρων, η σχέση επικοινωνίας και ελέγχου μεταξύ τους.

Γενικά, η δομή πρέπει να ορίζει ρόλους και σχέσεις για να πετύχει τις παραπάνω προϋποθέσεις. Για παράδειγμα η κάλυψη πετυχαίνεται όταν η κατάλληλη δομή μοιράσει ρόλους στους πράκτορες ανάλογα με την ανταγωνιστικότητά τους και την γνώση τους για ένα δεδομένο υποπρόβλημα. Περισσότερα για τους ρόλους αναφέρονται στο επόμενο κεφάλαιο .

Η οργάνωση σε ένα ΣΠΠ είναι λιγότερο δομημένη στην προοπτική της και περισσότερο σχετίζεται με τη σύγχρονη οργανωτική θεωρία (Maines, 1984, Strauss, 1978). Ο Gasser (1986) όρισε την οργάνωση σαν «ένα ιδιαίτερο σύνολο από ορισμένα και αόριστα ζητήματα τα οποία αφορούν τις πεποιθήσεις και τις δράσεις μέσα από τις οποίες οι πράκτορες έχουν άποψη για τους άλλους πράκτορες». Σύμφωνα με τον παραπάνω ορισμό, οργανωτική αλλαγή σημαίνει άνοιγμα ή/και ορισμός κάποιου διαφορετικού συνόλου ζητημάτων με κάποιο διαφορετικό τρόπο, δίνοντας σε συγκεκριμένους

πράκτορες νέα προβλήματα προς λύση όπως και μια διαφορετική βάση υποθέσεων για τις πεποιθήσεις και τις δράσεις των άλλων πρακτόρων. Έτσι, σε ένα πρόβλημα ένας πράκτορας μπορεί να συνεργάζεται με κάποιον άλλο πράκτορα, σε κάποιο επόμενο πρόβλημα όμως να μην είναι καν στην ίδια ομάδα.

Ο Malone (1990) ασχολείται με την οργάνωση ομάδας. “Μια ομάδα πρακτόρων είναι οργανωμένη αν οι τελευταίοι είναι συνδεδεμένοι με κάποιο τρόπο όπου οι συνδυασμένες ενέργειές τους αποδίδουν καλύτερα από το να μην υπήρχε αυτή η σύνδεση. Η οργανωμένη ομάδα αποτελείται από: μια ομάδα πρακτόρων, ένα σύνολο ενεργειών των πρακτόρων, ένα σύνολο συνδέσεων μεταξύ των πρακτόρων, και ένα σύνολο από στόχους ή κριτήρια εκτίμησης με τα οποία οι συνδυασμένες ενέργειες των πρακτόρων εκτιμούνται”.

Οι ΕΠ μπορούν να είναι γεωγραφικά κατανεμημένοι επιτρέποντας την αλληλεπίδραση με το σύστημα χρηστών με διαφορετικά επίπεδα ευθύνης-πρόσβασης. Ακόμα, διαβαθμισμένες βάσεις γνώσεις που χρησιμοποιούνται από συγκεκριμένους τύπους πρακτόρων μπορούν να φυλάσσονται σε συγκεκριμένους υπολογιστές.

### **2.3.1 Συντονισμός, Συνεργασία και Διαπραγματεύσεις**

Οι τρεις αυτές έννοιες βρίσκονται πολύ κοντά εννοιολογικά για αυτό και περικλείονται στην ίδια παράγραφο. Με τον συντονισμό οι πράκτορες συγχρονίζουν τις πράξεις τους έτσι ώστε να είναι καλή η συνεργασία τους και να επιτευχθούν οι στόχοι που συμφώνησαν στις διαπραγματεύσεις τους.

Οι Jennings, Sycara και Wooldridge (1998) βλέπουν τις τρεις αυτές έννοιες ως τύπους αλληλεπίδρασης των ΕΠ. Οι ΕΠ συνεργάζονται για να κατακτήσουν ένα κοινό στόχο, συντονίζονται ώστε να οργανώσουν τις δραστηριότητές τους αποφεύγοντας τις συγκρούσεις και επιτυγχάνοντας βέλτιστο αποτέλεσμα και τέλος διαπραγματεύονται ώστε να συμφωνήσουν όλοι μαζί στην πορεία προς το συλλογικό στόχο.

### **Συντονισμός**

Ο Malone, 1990 αναγνωρίζει δύο τρόπους συντονισμού: ιεραρχίες και αγορές. Οι ιεραρχίες βασίζονται στην άμεση επίβλεψη εργασιών. Οι αγορές από την άλλη βασίζονται στην επίτευξη συμφωνίας μεταξύ πρακτόρων οι οποίοι έχουν πρόσβαση σε διαφορετικούς πόρους, λειτουργίες. Και οι δύο αυτοί τρόποι συντονισμού γίνονται απαιτητικοί σε πόρους και επικοινωνία όσο αυξάνεται ο πληθυσμός της ομάδας.

Ένας άλλος τρόπος συντονισμού είναι μέσω ενός κεντρικού ελεγκτή ο οποίος θα συντονίζει τις πράξεις όλων των ΕΠ και ο οποίος θα μπορεί να έχει μια ευρύτερη άποψη για την κοινωνία των πρακτόρων (Moulin και Chaib-draa, 1996, Artificial Intelligence Center, 2000). Αυτός ο τρόπος συντονισμού αναιρεί το πλεονέκτημα της DAI όπου η αστοχία ενός λυτή δεν επηρεάζει την απόδοση των υπολοίπων. Αν αστοχήσει ο κεντρικός ελεγκτής όλο το σύστημα αδρανεί.

Το 1993 ο Jennings μίλησε για τις συμβάσεις και δεσμεύσεις. Σύμφωνα με αυτή τη θεωρία οι πράκτορες συντονίζονται μέσω συμβάσεων που στέλνουν ο ένας στον άλλο και δεσμεύσεων (όταν κάποιος «υπογράψει» μια σύμβαση δεσμεύεται ως προς το περιεχόμενό της). Αυτή η θεωρία έχει πρακτική εφαρμογή σε συστήματα όπου οι πράκτορες είναι «ειλικρινείς» και αυτόνομοι.

Τις περισσότερες φορές ο συντονισμός γίνεται πριν οι πράκτορες αναλάβουν δράση – στη φάση σχεδιασμού της δράσης τους – σε ορισμένες περιπτώσεις όμως είναι δυνατή και αργότερα (π.χ. αλλάζουν οι συνθήκες στις οποίες εκτελείται ένα έργο). Ο επιμέρους σφαιρικός σχεδιασμός (Partial Global Planning) των Durfee και Lesser (1987) προβλέπει την αναδιαπραγμάτευση μεταξύ των πρακτόρων αφού έχει αρχίσει να εκτελείται μια εργασία.

Το contract-net είναι ένα πολύ σημαντικό πρωτόκολλο συντονισμού για τα ΣΠΠ, προτάθηκε το 1981 από τους Smith και Davis και χρησιμοποιείται ακόμα και στα πιο σύγχρονα ΣΠΠ, το ZEUS (Collis και Ndumu, 1999b) επί παραδείγματι. Σύμφωνα με αυτό το πρωτόκολλο διαπραγμάτευσης κάθε πράκτορας συντονίζεται με τους άλλους μέσω συμβολαίων. Κάθε πράκτορας που αναλαμβάνει ένα συμβόλαιο το σπάει σε μικρότερα και τα αναθέτει σε πράκτορες που γνωρίζει ότι μπορούν να τα φέρουν σε πέρας. Στέλνει λοιπόν σε όλους το συμβόλαιο με τα υποέργα αναλαμβάνοντας το ρόλο του μανάτζερ. Οι πράκτορες εκτιμούν αν μπορούν να το φέρουν σε πέρας και αν ναι αποστέλλουν τις προσφορές τους στο μανάτζερ. Ο τελευταίος αξιολογεί τις προσφορές και επιλέγει σε ποιον θα κάνει την ανάθεση. Από κει και πέρα ο μανάτζερ ολοκληρώνει το συμβόλαιό του επικοινωνώντας με τους πράκτορες στους οποίους ανέθεσε τα υποέργα. Το πρωτόκολλο αυτό είναι και μέρος του κεφαλαίου της διαπραγμάτευσης (σε κάποιες εργασίες αναφέρεται ως πρωτόκολλο διαπραγμάτευσης, π.χ. Moulin και Chaib-Draa, 1996).

## Συνεργασία

Σύμφωνα με τους Durfee et al. (1989) η συνεργασία των πρακτόρων που ανήκουν σε ένα ΣΠΠ πρέπει να έχει τέσσερις στόχους:

- ⇒ Αύξηση του ρυθμού ολοκλήρωσης των εργασιών μέσω του παραλληλισμού εκτέλεσης υπο-εργασιών
- ⇒ Αύξηση του συνόλου ή του εύρους των εργασιών που μπορούν να εκτελεστούν μέσω του διαμοιρασμού πόρων.
- ⇒ Αύξηση της πιθανότητας να ολοκληρωθούν οι εργασίες μέσω της διπλής εκτέλεσής τους.
- ⇒ Μείωση της αλληλεπίδρασης εργασιών

Τέτοιοι πράκτορες είναι συνήθως οι συνεργατικοί κατανεμημένοι λυτές προβλημάτων (cooperative distributed problem-solving, CDPS). Αυτοί λύνουν προβλήματα τα οποία δεν είναι ποτέ μέσα στις δυνατότητες ενός μόνο ΕΠ. Έτσι, το πρόβλημα χωρίζεται σε υποπροβλήματα/ υποέργα των οποίων η σύνθεση δίνει τη λύση.

Οι Cammarata et al. (1983) πρότειναν μια στρατηγική συνεργασίας για αποφυγή συγκρούσεων μεταξύ πρακτόρων οι οποίοι εκτελούν τα σχέδιά τους. Σύμφωνα με αυτή τη στρατηγική, στην περίπτωση που οι πράκτορες δουν μια πιθανή σύγκρουση συμφερόντων πρέπει να επιλέξουν κάποιον ικανότερο από τους άλλους για να δημιουργήσει ένα κοινό σχέδιο δράσης. Η απόφαση στη δουλειά των Cammarata et al. αφορούσε το ποιος από τους πράκτορες που κανονίζουν τις πορείες αεροσκαφών που κινδυνεύουν να συγκρουστούν θα σχεδιάσει τις νέες πορείες των αεροσκαφών. Για να αποφασιστεί από την ομάδα ποιος είναι ο ικανότερος λαμβάνονταν υπόψη συγκεκριμένα

κριτήρια όπως επάρκεια πληροφόρησης, πόσο περιορισμένος στη δράση του (ποιος έχει περισσότερους περιορισμούς, π.χ. ποιος έχει τα λιγότερα καύσιμα) είναι κάποιος πράκτορας, κλπ.

Ο Decker (1987) θέτει το θέμα της συνεργασίας στο επίπεδο του βαθμού στον οποίο είναι συνεργάσιμοι οι ΕΠ από πλήρως συνεργάσιμοι μέχρι ανταγωνιστικοί. Οι πλήρως συνεργάσιμοι πράκτορες εκτελούν τις εργασίες τους συνήθως με μεγάλο επικοινωνιακό κόστος και με καθυστέρηση καθώς είναι πάντα έτοιμοι να αλλάξουν τους στόχους τους για να ικανοποιήσουν τις ανάγκες άλλων πρακτόρων λειτουργώντας με συμπαγή τρόπο και συντονιζόμενοι μαζί τους. Οι ανταγωνιστικοί πράκτορες από την άλλη μπορεί να μην είναι συνεργάσιμοι αν δεν τους συμφέρει, ακόμα και μπλοκάρουν ο ένας τους στόχους των άλλων.

Τέλος αξίζει να αναφερθεί η δουλειά των Levesque et al. (1990) πάνω στη δημιουργία ομάδων. Σύμφωνα με αυτή τη δουλειά μια ομάδα χαρακτηρίζεται από μια κοινή πρόθεση (joint intention). Η κοινή πρόθεση αυτή γίνεται πράξη όταν υπάρχει μια κοινή δέσμευση (joint commitment) από όλα τα μέλη της ομάδας για την ανάληψη κοινής δράσης. Η κοινή δέσμευση αυτή ορίζεται ως ένας κοινός επίμονος στόχος (joint persistent goal). Για να γίνει κάποιος κοινωνός μιας κοινής δέσμευσης πρέπει να ενστερνιστεί τις κοινές πεποιθήσεις (beliefs) και δεσμεύσεις (commitments) της ομάδας.

### **Διαπραγμάτευση**

Η διαπραγμάτευση είναι ένας μηχανισμός που χρησιμεύει συνήθως στον συντονισμό μιας ομάδας ΕΠ. Οι Durfee, Lesser και Corkill (1989) περιγράφουν τη διαπραγμάτευση ως ένα τρόπο βελτίωσης μιας συμφωνίας (μειώνοντας την ασάφεια και την αβεβαιότητα). Οι Jennings, Sycara και Wooldridge (1998) ορίζουν την διαπραγμάτευση σαν μια μέθοδο συντονισμού και επίλυσης συγκρούσεων. Σύμφωνα με τους ίδιους τα χαρακτηριστικά ενός ΣΠΠ τα οποία κάνουν αναγκαία τη χρήση διαπραγμάτευσης είναι:

- ⇒ Η ύπαρξη κάποιας μορφής σύγκρουσης η οποία πρέπει να επιλυθεί με έναν αποκεντρωμένο τρόπο.
- ⇒ Η λειτουργία εγωκεντρικών πρακτόρων οι οποίοι έχουν πεπερασμένη λογική και τοπική-ελλιπή πληροφόρηση
- Όταν η διαπραγμάτευση χρειάζεται να αυτοματοποιείται (για χρήση σε πληροφοριακά συστήματα όπως τα ΣΠΠ) τρία είναι τα βασικά θέματα που πρέπει να οριστούν καλά:
  - ⇒ Πρωτόκολλα διαπραγμάτευσης (ένα σύνολο κανόνων σύμφωνα με τους οποίους θα αλληλεπιδράσουν οι διαπραγματευόμενοι ΕΠ, π.χ. ποιοι θα διαπραγματευτούν, ποιες είναι οι πιθανές καταστάσεις της διαπραγμάτευσης, κλπ),
  - ⇒ Αντικείμενα διαπραγμάτευσης (τα θέματα για τα οποία θα γίνει διαπραγμάτευση, π.χ. τιμή αγοράς προϊόντος) και
  - ⇒ μοντέλα λήψης αποφάσεων των πρακτόρων (τα μοντέλα αυτά χρησιμοποιούνται παράλληλα με το πρωτόκολλο από τους ΕΠ ώστε καλύτερα να υπερασπιστούν τα συμφέροντά τους)



Η διαδικασία της διαπραγμάτευσης συνήθως προβλέπει την αποστολή από έναν πράκτορα σε έναν άλλο μιας επιχειρηματικής πρότασης-προσφοράς (proposal). Δύο είναι συνήθως οι απαντήσεις του πράκτορα που δέχεται την προσφορά:

- ⇒ Η διατύπωση μιας κριτικής (critique) για την προσφορά ή/και
- ⇒ η διατύπωση μιας νέας προσφοράς (counter-proposal)

Οι συνηθισμένοι τρόποι διαπραγμάτευσης είναι οι (Jennings et al., 2001):

- ⇒ Πλειστηριασμός (Αγγλικού τύπου, Ολλανδικού τύπου κλπ). Σε αυτή την περίπτωση κάποιος πράκτορας κάνει προσφορές σε μια ομάδα πρακτόρων ή δέχεται προτάσεις πάνω σε ένα συγκεκριμένο αντικείμενο διαπραγμάτευσης από αυτούς.
- ⇒ Προσέγγιση με τη θεωρία παιγνίων (game theory). Σύμφωνα με αυτή την προσέγγιση οι πράκτορες κάνουν την κίνησή τους προσπαθώντας να απαντήσουν στην ερώτηση: «Ποιο είναι η καλύτερη-περισσότερο λογική κίνηση που μπορεί να κάνει ο πράκτορας;». Η θεωρία παιγνίων προκύπτει από τη μελέτη παιχνιδιών όπως το σκάκι. Στην περίπτωση της διαπραγμάτευσης πρακτόρων χρησιμοποιείται ώστε οι πράκτορες να επιλέξουν την καλύτερη στρατηγική από τον χώρο όλων των πιθανών στρατηγικών, αναλογιζόμενοι όλες τις πιθανές αλληλεπιδράσεις. Σημαντική δουλειά στο χώρο αυτό έκαναν οι Rosenschein και Zlotkin (1994). Αυτοί χρησιμοποίησαν ένα μονότονο πρωτόκολλο συμβιβασμού με στρατηγική Zeuthen (μοντέλο λήψης απόφασης). Σύμφωνα με αυτό η διαπραγμάτευση προχωρά με γύρους διαπραγματεύσεων. Σε κάθε γύρο κάθε εμπλεκόμενος πράκτορας κάνει μια προσφορά. Αν οι προσφορές είναι αλληλοκαλυπτόμενες η διαπραγμάτευση σταματά. Αν όχι ξεκινά νέος γύρος διαπραγματεύσεων, στον οποίο οι πράκτορες μπορούν είτε να κάνουν μια παραχώρηση (concession) είτε να προτείνουν την ίδια πρόταση που είχαν πριν. Αν κανείς δεν κάνει παραχώρηση τότε η διαπραγμάτευση σταματά χωρίς αποτέλεσμα. Σύμφωνα με την στρατηγική Zeuthen ο πράκτορας που θα κάνει παραχώρηση είναι αυτός που έχει να χάσει τα περισσότερα σε περίπτωση μη κατάληξης σε συμφωνία αλλά η παραχώρηση πρέπει να είναι η ελάχιστη που απαιτείται ώστε κάποιος άλλος να έχει πιο πολλά να χάσει σε επόμενο γύρο διαπραγμάτευσης.
- ⇒ Ευρετική προσέγγιση (Heuristic approach). Εδώ, ευριστικά πρωτόκολλα και μηχανισμοί λήψης απόφασης στοχεύουν στο να βρεθεί σε βέλτιστο χρόνο μια καλή (όχι βέλτιστη) απόφαση συμβιβασμού.
- ⇒ Προσέγγιση βασισμένη σε επιχειρήματα (argumentation-based approach). Σε αντίθεση με τους προηγούμενους τρόπους διαπραγμάτευσης εδώ δίνεται έμφαση στην άσκηση κριτικής σε μία προσφορά αντί για να κατατεθεί μια αντι-προσφορά. Ακόμα, μπορεί να περιλαμβάνονται απειλές, μόνους, κλπ σύμφωνα με την ανθρώπινη επιχειρηματολογία. Οι Parsons et al (1998) δείχνουν πως ένα μοντέλο επιχειρηματολογίας μπορεί να λειτουργήσει σε BDI πράκτορες (μια έρευνα για BDI αρχιτεκτονικές έκαναν οι Haddadi και Sundermeyer το 1996) επηρεάζοντας, με επιχειρηματολογία, ο ένας τις πεποιθήσεις του άλλου.

Όλες οι μέθοδοι αυτές έχουν πλεονεκτήματα και μειονεκτήματα. Οι πλειστηριασμοί περιλαμβάνουν μεγάλο όγκο ανταλλαγής μηνυμάτων και απευθύνονται σε ΣΠΠ στα οποία υπάρχουν πολλοί πράκτορες. Ακόμα ο πράκτορας που κάνει προσφορές δεν

μπορεί να ξέρει γιατί δεν γίνονται αποδεκτές (η μόνη πληροφορία που δέχεται από τους άλλους πράκτορες είναι η αποδοχή ή όχι της προσφοράς του). Η θεωρία παιγνίων είναι τεκμηριωμένη αλλά είναι ιδιαίτερα απαιτητική σε υπολογιστικό χρόνο και αφορά διαπραγματεύσεις σε συγκεκριμένα θέματα (ένα μοντέλο δεν μπορεί να γενικευτεί) και επίσης να γνωρίζουν οι πράκτορες όλο τον χώρο των λύσεων. Οι ευρετικοί μέθοδοι επειδή δεν αποδεικνύουν την επίτευξη ή όχι συμφωνίας θέλουν συνήθως εξαντλητικό έλεγχο για τη σωστή τους λειτουργία. Τέλος, η επιχειρηματολογία απευθύνεται σε πολύπλοκες αρχιτεκτονικές πρακτόρων και είναι ακόμα νέα ως κλάδος της διαπραγμάτευσης.

Οι παραπάνω περιπτώσεις αφορούν κυρίως ΣΠΠ όπου οι πράκτορες είναι πρόθυμοι να συνεργαστούν. Σε αντίθετη περίπτωση η διαπραγμάτευση πολύ συχνά φτάνει σε αδιέξοδο. Η Sycara (1990) πρότείνει την ύπαρξη τρίτων πρακτόρων οι οποίοι θα μεσολαβούν σε περίπτωση σύγκρουσης απόψεων. Η συγκεκριμένη εφαρμογή (PERSUADER) προέβλεπε ΕΠ που εκπροσωπούσαν τον εργοδότη και το σωματείο των εργαζομένων οι οποίοι πολύ συχνά είχαν αντίθετα συμφέροντα. Σε αυτή την περίπτωση ένας τρίτος πράκτορας – ο διαμεσολαβητής – αναλάμβανε να κάνει νέες προτάσεις για να μικρύνει τις διαφορές στις προτάσεις των προηγούμενων δύο.

### 2.3.2 Επικοινωνία και Αλληλεπιδράσεις των Ευφυών Πρακτόρων

Σε αυτή την παράγραφο εξετάζονται τα είδη επικοινωνίας μεταξύ πρακτόρων. Η επικοινωνία επιτρέπει στους πράκτορες να ανταλλάζουν πληροφορίες, να συντονίσουν τη δράση τους. Δύο είδη επικοινωνίας κυριαρχούν μεταξύ των πρακτόρων και αυτά είναι η κατευθείαν ανταλλαγή μηνυμάτων και η χρησιμοποίηση ενός κοινού σημείου συγκέντρωσης δεδομένων (μαυροπίνακας).

Υπάρχουν μέθοδοι συντονισμού οι οποίες έχουν χρησιμοποιηθεί και προβλέπουν ελάχιστη ή καθόλου επικοινωνία μεταξύ των ΕΠ. Έτσι, αν οι ΕΠ δεν χρειάζονται να συνεργαστούν για να πετύχουν τους στόχους τους αλλά ούτε συγκρούονται τα συμφέροντά τους τότε η επικοινωνία μεταξύ τους δεν είναι απαραίτητη για την εύρυθμη λειτουργία τους. Ακόμα υπάρχουν περιπτώσεις όπου ελάχιστη επικοινωνία με τη μορφή ενός πεπερασμένου συνόλων σημάτων αρκεί για τον συντονισμό των πρακτόρων (Georgeff, 1983). Τέλος, οι πράκτορες σε κάποια άλλα μοντέλα μπορούν να συντονιστούν παρατηρώντας τις αλλαγές που επιφέρουν άλλοι πράκτορες στο περιβάλλον (παρατηρούν τις αλλαγές γύρω τους). Σε αυτή την περίπτωση οι πράκτορες ενδέχεται να προσπαθήσουν να «μαντέψουν» ποιοι κινήθηκαν και τι κίνηση έκαναν παρατηρώντας τις επιδράσεις τους στο περιβάλλον (Rosenschein και Breese, 1989). Τέτοιες μεθόδους επικοινωνίας-συντονισμού χρησιμοποιούν συνήθως οι πράκτορες που αντιδρούν σε ερεθίσματα.

Ο μαυροπίνακας είναι μια μορφή κοινής μνήμης (Nii, 1986) όπου όλοι οι πράκτορες μπορούν να γράψουν κάτι ή να τον διαβάσουν. Συνήθως ο μαυροπίνακας χωρίζεται σε περιοχές και οι πράκτορες αποκτούν προνόμια με την ικανότητα να διαβάζουν περιοχές. Έτσι, άλλοι μπορούν να διαβάσουν και να γράψουν σε περισσότερες περιοχές από κάποιους άλλους.

Ο πιο διαδεδομένος τρόπος επικοινωνίας στα σύγχρονα συστήματα πολλαπλών πρακτόρων είναι η ανταλλαγή μηνυμάτων. Το περιεχόμενο του μηνύματος διαφοροποιείται σημαντικά από σύστημα σε σύστημα. Αυτό που χρειάζεται στην περίπτωση των μηνυμάτων είναι η διαμόρφωση ενός πρωτοκόλλου - γλώσσας σύνταξης και μεταφοράς του μηνύματος όπως και ο προσδιορισμός των διαφορετικών τύπων μηνυμάτων (π.χ. το πρωτόκολλο contract-net των Smith και Davis, 1981). Κάποιοι μελετητές πρότειναν την ανταλλαγή των σχεδίων δράσης των πρακτόρων μέσω των μηνυμάτων για τον καλύτερο συντονισμό τους. Ο Rosenschein το 1986 περιέγραψε τους κινδύνους του να μεταφέρεται μέσω μηνυμάτων το σχέδιο δράσης ενός πράκτορα. Οι κίνδυνοι αυτοί υπάρχουν επειδή σε αυτή την περίπτωση ανταλλάσσονται μεγάλα μηνύματα πράγμα που κάνει χρονοβόρα την μετάδοσή τους και που αυξάνει τις ανάγκες σε ευρύς διαύλους επικοινωνίας. Ακόμα, σε ένα δυναμικό περιβάλλον όταν το σχέδιο φτάσει μπορεί πια να μην εκφράζει τις προθέσεις του αποστολέα. Τέλος, ειδικά στην περίπτωση των αυτόνομων πρακτόρων αλλά και σε κάθε άλλη περίπτωση εκτός του πράκτορα αφέντη και του πράκτορα σκλάβου, είναι αμφίβολο το αν οι αποδέκτες ενός σχεδίου δράσης θα το υιοθετήσουν.

Οι δυνατότητας επικοινωνίας μεταξύ πρακτόρων μπορούν να επεκταθούν και πέραν της ανταλλαγής ενός απλού μηνύματος εισάγοντας σε ένα μήνυμα τη δυνατότητα να επηρεάζει τις γνώσεις και πεποιθήσεις του πράκτορα που λαμβάνει το μήνυμα όπως και να περιέχει στοιχεία της θεωρίας λόγου-δράσης (speech-act, προτάθηκε από τον Searle

το 1969). Έτσι όταν οι πράκτορες ακολουθούν την BDI λογική οι πεποιθήσεις τους μπορεί να αλλάζουν αντίστοιχα με αλλαγές στο περιβάλλον ή στη συμπεριφορά άλλων πρακτόρων. Με χρήση σύγχρονων γλωσσών όπως η γλώσσα χειρισμού ερωταπαντήσεων γνώσης (Knowledge Query Manipulation Language, KQML) ή η γλώσσα επικοινωνίας πρακτόρων (Agent Communication Language, συντομογραφία: ACL) είναι δυνατή αυτή η λειτουργία όπως και η δυνατότητα speech-act δηλαδή δράσης μέσω του λόγου (κάποιος πράκτορας μπορεί να ζητήσει κάτι λέγοντας «παρακαλώ θέλω...» ενώ κάποιος άλλος-πιθανόν σε ιεραρχικές δομές- να το ζητήσει λέγοντας «θέλω αμέσως...»). Αυτές οι δυνατότητες επικοινωνίας συνήθως αξιοποιούνται από κοινωνικούς πράκτορες και περιγράφονται αναλυτικά στο επόμενο κεφάλαιο.

Οι Jennings, Sycara και Wooldridge (1998) υποστηρίζουν πως σε ένα ΣΠΠ οι πράκτορες που το αποτελούν «συνεργάζονται για να κατακτήσουν ένα κοινό στόχο, συντονίζονται ώστε να οργανώσουν τις δραστηριότητες τους αποφεύγοντας τις συγκρούσεις και επιτυγχάνοντας βέλτιστο αποτέλεσμα και τέλος διαπραγματεύονται ώστε να συμφωνήσουν όλοι μαζί στην πορεία προς το συλλογικό στόχο».

Ο συντονισμός των πρακτόρων μπορεί σύμφωνα με τον Malone (1990) να γίνεται είτε με ιεραρχίες είτε με αγορές. Στην περίπτωση των ιεραρχιών έχουμε άμεση επίβλεψη των εργασιών. Υπάρχει μία καθορισμένη ιεραρχική δομή των πρακτόρων έτσι που κάθε εργασία να επιβλέπεται από τον προϊστάμενο πράκτορα. Κάποια ΣΠΠ υποστηρίζουν ένα κεντρικό ελεγκτή ο οποίος συντονίζει όλους τους υπόλοιπους πράκτορες, με τον κίνδυνο να καταρρεύσει όλο το σύστημα αν αστοχήσει ο ηγέτης πράκτορας. Στην περίπτωση των αγορών οι πράκτορες (ανταγωνιστικοί ή συνεργατικοί) προσπαθούν να έρθουν σε συμφωνία για τη βέλτιστη κατανομή των πόρων και των λειτουργιών που πρέπει να εκτελέσει το σύστημα. Χαρακτηριστικό σε αυτή την περίπτωση είναι το «πρωτόκολλο διαπραγμάτευσης» (Contract Net Protocol) που προτείνανε οι Smith και Davis (1981) και περιγράψαμε στο προηγούμενο κεφάλαιο.

Ίσως όμως το πιο σημαντικό στοιχείο σε ένα ΣΠΠ είναι η επικοινωνία μεταξύ των πρακτόρων. Η σωστή και αποτελεσματική επικοινωνία είναι το χαρακτηριστικό που θα κρίνει τη συνολική λειτουργία του ΣΠΠ. Η επικοινωνία επιτρέπει στους πράκτορες να ανταλλάξουν πληροφορίες, να αποκτήσουν τα δεδομένα που χρειάζονται για τη λειτουργία τους και να συντονίσουν τη δράση τους. Δύο είναι τα κυρίαρχα είδη επικοινωνίας, ο μαυροπίνακας και η ανταλλαγή μηνυμάτων. Την έννοια του μαυροπίνακα εισήγαγαν για πρώτη φορά οι Erman, Hayes-Roth, Lesser, και Reddy το 1980. Πρόκειται για ένα μηχανισμό κοινής μνήμης. Κάθε πράκτορας πραγματοποιώντας τις λειτουργίες που του έχουν αναθέσει, κοιτά στον μαυροπίνακα για να συγκεντρώσει νέες πληροφορίες και αποστέλλει εκεί τα αποτελέσματα της δουλειάς του. Οι πράκτορες αποκτούν διάφορα προνόμια σε σχέση με την πρόσβαση στον μαυροπίνακα. Έτσι, καθώς αυτός χωρίζεται σε περιοχές, κάποιοι πράκτορες έχουν πρόσβαση σε ορισμένες από αυτές, κάποιοι σε διαφορετικές περιοχές και κάποιοι σε όλες. Η πρόσβαση αυτή μπορεί να επιτρέπει την απλή ανάγνωση των πληροφοριών ή και την επεξεργασία τους (σβήσιμο δεδομένων, εισαγωγή στοιχείων, τροποποίηση δεδομένων). Ο δεύτερος τρόπος και αρκετά διαδεδομένος στα σύγχρονα ΣΠΠ είναι η ανταλλαγή μηνυμάτων. Το ΣΠΠ που υποστηρίζει αυτού του είδους την επικοινωνία καθορίζει ένα πρωτόκολλο κατασκευής και διάδοσης μηνυμάτων. Έτσι τα μηνύματα περιέχουν τυποποιημένες πληροφορίες και ακολουθούν ένα συγκεκριμένο τρόπο διάδοσης. Υπάρχουν πολλές μέθοδοι ανταλλαγής

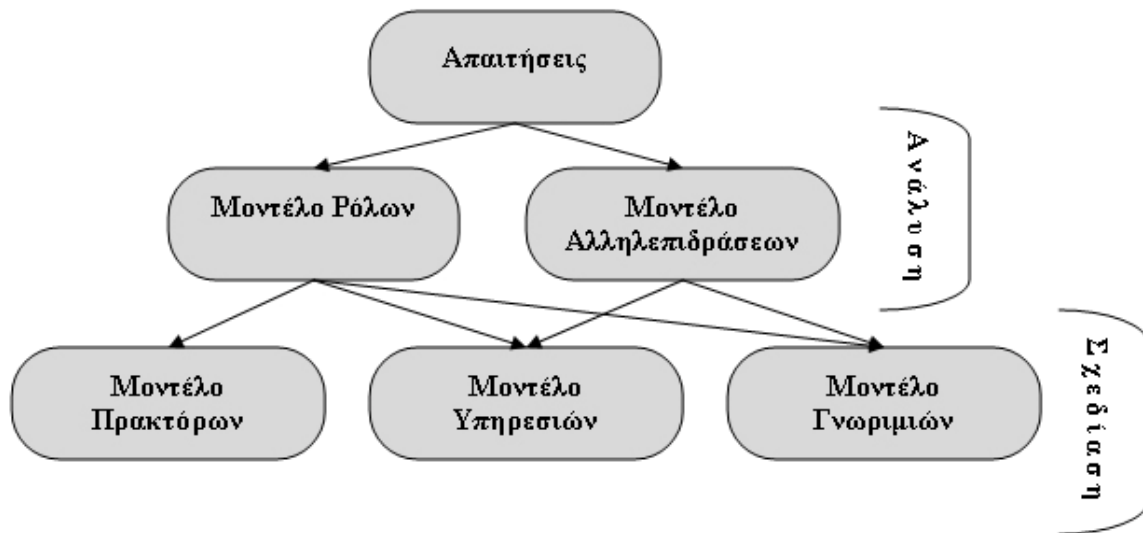
μηνυμάτων ώστε η απογραφή τους να είναι ένα αρκετά ενδιαφέρον θέμα που ξεφεύγει όμως από το σκοπό της εργασίας αυτής.

Ενδεικτικά αναφέρουμε μερικές υπάρχουσες πλατφόρμες ανάπτυξης ΣΠΠ : RETSINA (Sycara et al 1996), ARCHON ( Jennings et al. 1996b), Open Agent Architecture (SRI International ), AgentBuilder (Reticular Systems Inc., 1999), GrassHopper (IKV θυγατρική GMD FOCUS 1998), ZEUS (British Telecommunications).

### 2.3.3 Μεθοδολογίες Ανάπτυξης Συστημάτων Πολλαπλών Πρακτόρων

Για την ανάπτυξη ενός ΣΠΠ έχουν αναπτυχθεί πολλές διαφορετικές μεθοδολογίες (Iglesias C., Garijo M., Gonzalez J.) ωστόσο φαίνεται ο κάθε κατασκευαστής ενός ΣΠΠ να προτείνει και μία νέα μεθοδολογία που υλοποιεί με τον καταλληλότερο τρόπο τους σκοπούς του. Οι πιο γενικές και αυτές που καλύπτουν ολόκληρη τη διαδικασία ανάπτυξης ενός ΣΠΠ είναι η Gaia και η Multiagent Systems Engineering – MaSE (Wood and DeLoach, 2000), που κατασκευάστηκε από το Ινστιτούτο τεχνολογίας της Πολεμικής Αεροπορίας των Η.Π.Α. την AFIT, και περιγράφονται παρακάτω.

**Gaia μεθοδολογία:** Η μεθοδολογία της Gaia δουλεύει κάτω από ορισμένες προϋποθέσεις που ορίζουν τους πράκτορες του ΣΠΠ να χρησιμοποιούν υπολογιστικούς πόρους, να συνεργάζονται και όχι να ανταγωνίζονται ενώ οι σχέσεις τους, οι ικανότητες τους και οι υπηρεσίες που προσφέρουν είναι στατικές, δε μεταβάλλονται δηλαδή σε τρέχον χρόνο (run-time). Υπάρχει επίσης περιορισμός για τον αριθμό των τύπων των πρακτόρων, ο οποίος πρέπει να είναι σχετικά μικρός. Η ανάπτυξη λοιπόν ενός ΣΠΠ σύμφωνα με τη Gaia χωρίζεται σε δύο κυρίως φάσεις : τη φάση της ανάλυσης (όπου σχηματίζουμε μια πρώτη εικόνα του συστήματος και της δομής του) και τη φάση της σχεδίασης (όπου υλοποιείται με αντικειμενοστραφή σχεδιασμό το σύστημα).



Εικόνα 5: Σχέσεις μεταξύ των μοντέλων της μεθοδολογίας Gaia

Ακολουθώντας τη μεθοδολογία της Gaia προκύπτουν τα μοντέλα που φαίνονται στο σχήμα στις φάσεις που αναφέραμε.

#### Φάση Ανάλυσης

⇒ Μοντέλο Ρόλων : Αναγνωρίζονται οι ρόλοι που απαιτούνται για τη σωστή και αποτελεσματική λειτουργία του συστήματος και καθορίζονται πλήρως οι πόροι που

θα χρησιμοποιεί, οι λειτουργίες που θα εκτελεί οι συνεργασίες και η επικοινωνία που θα έχει ο κάθε ρόλος

- ⇒ Μοντέλο Αλληλεπιδράσεων : Εδώ ορίζεται το πρωτόκολλο επικοινωνίας και αλληλεπίδρασης μεταξύ των πρακτόρων.

### **Φάση Σχεδίασης**

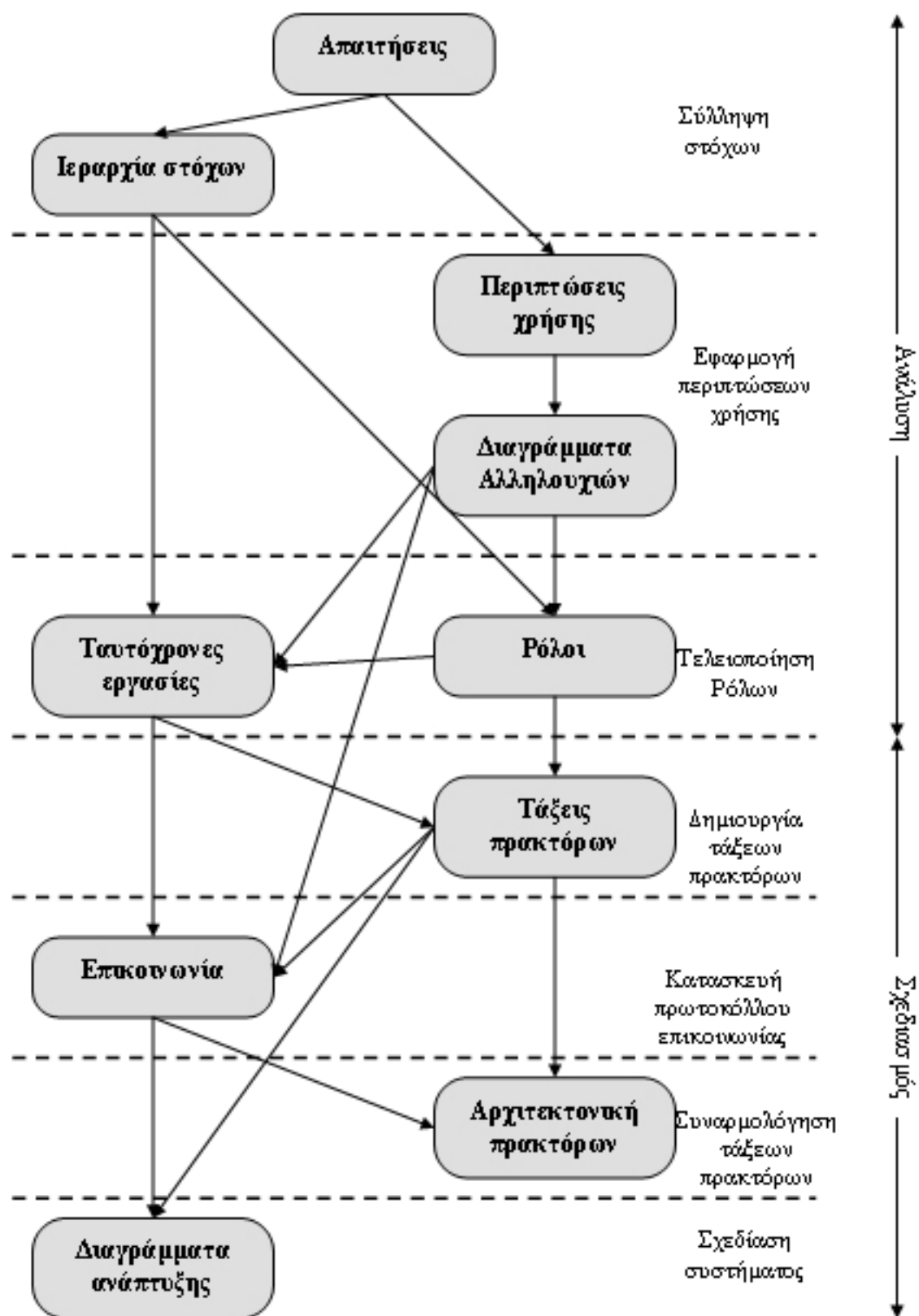
- ⇒ Μοντέλο Πρακτόρων: Ορίζονται οι τύποι των πρακτόρων και αναθέτονται οι ρόλοι
- ⇒ Μοντέλο Υπηρεσιών: Εδώ αποσαφηνίζονται οι λειτουργίες κάθε ρόλου . Καθορίζεται ξεχωριστά η κάθε λειτουργία, σε ποιόν ρόλο ανήκει, τι απαιτήσεις έχει και πότε είναι εφικτή.
- ⇒ Μοντέλο Γνωριμιών: Αυτό το μοντέλο καθορίζει ποίος αλληλεπιδρά (συνεργάζεται ) με ποιόν χωρίς όμως να καθορίζει τον τρόπο επικοινωνίας ή τα αποτελέσματα της αλληλεπίδρασης .

**Μεθοδολογία MaSE:** Η MaSE αποτελεί μία ολοκληρωμένη μεθοδολογία ανάπτυξης ΣΠΠ και καθοδηγεί τον κατασκευαστή ενός ΣΠΠ με προσεγμένο τρόπο. Πρέπει να σημειώσουμε πως στη MaSE στηρίχτηκε και ο γράφοντας για το ΣΠΠ που προτείνει. Είναι ανεξάρτητη αρχιτεκτονικής πρακτόρων, συστήματος μεταβίβασης μηνυμάτων και γλώσσας προγραμματισμού. Η MaSE προσφέρει επίσης τη δυνατότητα να πραγματοποιούνται και να ανιχνεύονται οι αλλαγές κατά τη διάρκεια της διαδικασίας επιτρέποντας την ανάδραση μεταξύ των αντικειμένων σχεδιασμού σε κάθε φάση της μεθοδολογίας . Παρόλα αυτά, περιορίζει τη σχεδίαση ενός ΣΠΠ προϋποθέτοντας (Wood M. and DeLoach A., AFIT 2001)

- ⇒ Το περιβάλλον λειτουργίας των πρακτόρων να είναι κλειστό και η επαφή με το περιβάλλον γίνεται με τη χρήση ενός πράκτορα.
- ⇒ Η MaSE δεν επιτρέπει τη δημιουργία, καταστροφή ή μετακίνηση πρακτόρων κατά τη διάρκεια εκτέλεσης (run-time).
- ⇒ Οι αλληλεπιδράσεις μεταξύ των πρακτόρων είναι ένας-προς-έναν, δεν επιτρέπεται δηλαδή η αποστολή μηνυμάτων σε περισσότερους από έναν αποδέκτες.
- ⇒ Το ΣΠΠ δεν μπορεί να περιέχει πάνω από 10 τύπους (τάξεις ) πρακτόρων.

Τα βασικά βήματα της μεθόδου είναι αυτά που περιγράφονται στο παραπάνω σχήμα και αναλύονται παρακάτω :

- ⇒ Σύλληψη στόχων: Σε αυτό το πρώτο βήμα έχοντας σα δεδομένες τις απαιτήσεις της σχεδίασης, το βασικό σκοπό δηλαδή δημιουργίας του ΣΠΠ προσπαθούμε να καταλήξουμε σε στοιχειώδεις στόχους που θα συνθέτουν τον κεντρικό σκοπό (είναι πολύ πιθανό αυτοί οι στοιχειώδεις στόχοι να είναι και οι εργασίες των πρακτόρων)
- ⇒ Εφαρμογή περιπτώσεων χρήσης: Στο βήμα αυτό κατασκευάζονται τα διαγράμματα αλληλουχιών, τα σενάρια λειτουργίας του συστήματος. Μελετούνται οι περιπτώσεις χρήσης (uses cases) του συστήματος και προσομοιώνεται σε θεωρητική βάση η λειτουργία του.
- ⇒ Τελειοποίηση ρόλων: Αποσαφηνίζεται ο τρόπος με τον οποίο θα εκτελούνται παράλληλα εργασίες διαφόρων πρακτόρων και ορίζονται οι ρόλοι και οι αλληλεπιδράσεις τους σύμφωνα με τους στόχους που καθορίστηκαν στο πρώτο βήμα.



Εικόνα 6: Η μεθοδολογία MaSE. (Πηγή Wood and DeLoach, 2001).



- ⇒ Δημιουργία τάξεων πρακτόρων: Υλοποιούνται οι τύποι (τάξεις ) των πρακτόρων ανάλογα με τους ρόλους που θα αναλάβει ο καθένας . Να μη ξεχνάμε ότι ένας πράκτορας είναι δυνατόν να αναλάβει έναν ή και περισσότερους ρόλους.
- ⇒ Κατασκευή πρωτοκόλλου επικοινωνίας: Εδώ αποφασίζεται και αναπτύσσεται ο τρόπος με τον οποίο θα επικοινωνούν οι πράκτορες.
- ⇒ Συναρμολόγηση τάξεων πρακτόρων: Ορίζεται η αρχιτεκτονική του κάθε πράκτορα του ΣΠΠ
- ⇒ Σχεδίαση συστήματος: Κατασκευάζονται σε κάποια αντικειμενοστραφή γλώσσα (π.χ. UML Unified Modeling Language) τα διαγράμματα ανάπτυξης (deployment diagrams) που παρουσιάζουν πως θα λειτουργεί το ΣΠΠ

Η MaSE μεθοδολογία προσπαθεί να καλύψει πλήρως την διαδικασία ανάλυσης και σψεδιασμού ενός ΣΠΠ. Η βασική της διαφορά με την Gaia είναι η πρόβλεψη για σενάρια και περιπτώσεις χρήσης (use-cases) των σχεδιαζόμενων ΣΠΠ. Ακόμα οι δημιουργοί της μεθοδολογίας αυτής δείχνουν ότι έχουν ένα πολύ καλό υπόβαθρο στη σχεδίαση συστημάτων και ιδιαίτερα στη σχεδίαση αντικειμενοστραφών συστημάτων και στη χρήση της γλώσσας μοντελοποίησης UML, καθώς η μεθοδολογία καλύπτει σχεδόν εξολοκλήρου τη διαδικασία σχεδίασης συστημάτων πολλαπλών πρακτόρων με μοντέρνες τεχνικές και διαγράμματα.

Παρόλα αυτά, τα μειονεκτήματα της είναι ότι θέτει μεγαλύτερο περιορισμό στον αριθμό των τύπων των πρακτόρων που θα συμμετέχουν στο ΣΠΠ και επίσης απαγορεύει τη δυναμική δημιουργία – καταστροφή πρακτόρων κατά τη διάρκεια λειτουργίας του ΣΠΠ.

## 2.4 Έμπειρα Συστήματα (Expert Systems)

### 2.4.1 Η φύση της ειδικεύσης

Κάτω από ποιες προϋποθέσεις θα μπορούσε ένα πρόγραμμα υπολογιστή να θεωρηθεί ειδικό, για κάποιο γνωστικό πεδίο;

Καταρχάς το πρόγραμμα οφείλει να κατέχει γνώση. Με αυτή την έννοια δεν εννοούμε την επεξεργασία κάποιου αλγορίθμου, παραδείγματος χάριν την χρήση μιας λίστας καταστάσεων προς έλεγχο, καθώς είναι σίγουρο ότι αργά ή γρήγορα το πρόγραμμα θα οδηγηθεί σε αδιέξοδο.

Η γνώση πρέπει να εστιάζεται σε ένα συγκεκριμένο πεδίο. Μια τυχαία συλλογή από ονόματα, ημερομηνίες, τοποθεσίες και γνωμικά δεν είναι η μορφή της γνώσης που προσφέρει την βάση για την εξειδίκευση. Η γνώση υποδηλώνει οργάνωση και ενσωμάτωση, από την άποψη ότι διαφορετικά κομμάτια γνώσης, σχετίζονται μεταξύ τους και το ένα φέρνει το άλλο στην επιφάνεια. Τέλος, αυτή η γνώση πρέπει να είναι ικανή για την απευθείας επίλυση προβλημάτων.

Λαμβάνοντας, τα παραπάνω υπόψιν δημιουργούμε τον ακόλουθο ορισμό για το τι είναι Έμπειρα Συστήματα.

***Ένα Έμπειρο Σύστημα είναι ένα προγράμμα υπολογιστή το οποίο αναπαριστά (την γνώση κάποιου ειδικού) και συλλογίζεται με την γνώση κάποιου ειδικού, με σκοπό την επίλυση προβλημάτων και την παροχή συμβουλών.***

Ένα έμπειρο σύστημα μπορεί να ικανοποιεί πλήρως μια λειτουργία που απαιτεί ανθρώπινη εξειδίκευση ή μπορεί να λειτουργεί υποστηρικτικά στην λήψη αποφάσεων ενός αποφασίζοντα. Ο αποφασίζοντας, μπορεί να είναι και αυτός ένας ειδικός. Σε αυτή την περίπτωση το σύστημα οφείλει να λειτουργεί έτσι ώστε να αυξάνει την παραγωγικότητά του. Σε άλλες περιπτώσεις ο χρήστης μπορεί να είναι ικανός να παραγεί σημαντική εργασία, ανάλογη κάποιου ειδικού εάν του παρασχεί η κατάλληλη τεχνική υποστήριξη από το πρόγραμμα. Συνοψίζοντας, ο κατάλληλος καταμερισμός των λειτουργιών μεταξύ του υπολογιστή και του ανθρώπου είναι το κλειδί στην επιτυχημένη εφαρμογή των έμπειρων συστημάτων.

Όπως προαναφέρθηκε, η τεχνολογία των Έμπειρων Συστημάτων, προέρχεται από την έρευνα της Τεχνητής Νοημοσύνης: ένα κλάδο της επιστήμης υπολογιστών που ασχολείται με τον σχεδιασμό και την υλοποίηση προγραμμάτων ικανών να προσομοιώνουν τις νοητικές ικανότητες των ανθρώπων όπως την επίλυση προβλημάτων, την οπτική αντίληψη και την κατανόηση γλώσσας. Αυτή η τεχνολογία εφαρμόστηκε επιτυχημένα σε διάφορα πεδία, συμπεριλαμβανομένων της οργανικής χημείας, της εξερεύνησης κοιτασμάτων και της φαρμακευτικής. Τυπικές ασχολίες των Έμπειρων Συστημάτων είναι:

- ⇒ Ερμηνεία ψηφιακών δεδομένων (π.χ. σήματα ραντάρ),
- ⇒ Διάγνωση δυσλειτουργιών (π.χ. ασθένειες, βλάβες),
- ⇒ Δομική ανάλυση σύνθετων αντικειμένων (π.χ. χημικές ενώσεις),
- ⇒ Configuration σύνθετων αντικειμένων (π.χ. υπολογιστές), και
- ⇒ Σχεδιασμός σειράς δράσεων (π.χ. στα ρομπότ)

Παρά το γεγονός ότι συμβατικά προγράμματα, διεκπεραιώνουν παρόμοιες λειτουργίες σε παρόμοια πεδία, διαφέρει σημαντικά ο τρόπος προσέγγισης της επίλυσης των προβλημάτων. Πραγματικά, ο ορισμός που δόθηκε παραπάνω αφήνει πολλά ερωτηματικά για το τι είναι και τι κάνει ένα Έμπειρο Σύστημα. Ωστόσο, υπάρχει ένα σημαντικό πλήθος χαρακτηριστικών που ένα Έμπειρο Σύστημα οφείλει να ικανοποιεί σε κάποιο βαθμό.

## 2.4.2 Άλλοι Ορισμοί

Σε αυτή την παράγραφο παραθέτουμε μερικούς από τους ορισμούς των Έμπειρων Συστημάτων, που έχουν δοθεί κατά καιρούς, οι οποίοι διακρίνονται σε αυτούς που στηρίζονται στο **τι κάνει** ένα Έμπειρο Σύστημα και σε αυτούς που ασχολούνται στο **πώς το κάνει**. (N. Ματσατσίνης, N. Σπανουδάκης, Α. Σαμαράς – 2005)

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 αφενός μεν χειρίζεται πραγματικά σύνθετα προβλήματα, που απαιτούν την παρουσία ειδικού, αφετέρου δε επιλύει τέτοια προβλήματα χρησιμοποιώντας ένα υπολογιστικό μοντέλο λογικής ειδικού, καταλήγοντας στα ίδια συμπεράσματα που ένας ειδικός θα έφτανε αν αντιμετώπιζε ένα ανάλογο πρόβλημα. Ένα Έμπειρο Σύστημα προσπαθεί να συλλάβει αρκετή από τη γνώση ενός ειδικού έτσι, ώστε να επιλύει με έμπειρο τρόπο τα προβλήματα (Weiss and Kulikowski, 1984)

Τον επόμενο χρόνο ο Sell δίνει τον ορισμό, ότι ένα Έμπειρο Σύστημα είναι ένα σύστημα βασιζόμενο στη γνώση που προσομοιάζει τη σκέψη του ειδικού προκειμένου να επιλύσει σημαντικά προβλήματα σε ένα ξεχωριστό πεδίο. (Sell, 1985)

Ο Hart (1986), εκτιμά ότι τα Έμπειρα Συστήματα 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.

Κατά μια άλλη άποψη ένα Έμπειρο Σύστημα, είναι ένα εργαλείο που έχει την ικανότητα να κατανοεί την ιδιαίτερη γνώση ενός προβλήματος και χρησιμοποιώντας ευφυώς την γνώση, του πεδίου αυτού, να προτείνει εναλλακτικές ενέργειες. (Kumara et al., 1986)

Έμπειρα Συστήματα είναι προγράμματα με τα οποία γίνεται προσπάθεια να αναπαρασταθεί η συμπεριφορά ενός ειδικού κάποιου τομέα (Curry and Moutinho, 1991). 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 πως οδηγούνται σε αυτές.

Οι Doukidis (1998) και Soukidis and Paul (1992), συνδυάζοντας το τι και το πώς κάνει ένα Έμπειρο Σύστημα ορίζουν ότι Έμπειρο Σύστημα είναι ένα πρόγραμμα που βοηθά το χρήστη, παρέχοντας πληροφορίες σε κάποιο ιδιαίτερο πεδίο. Αυτό το κάνει χειριζόμενο πληροφορίες σχετικές με ένα χώρο, που παρέχονται από μια σειρά από «ειδικούς» του χώρου αυτού.

### 2.4.3 Τα χαρακτηριστικά ενός έμπειρου συστήματος

Ένα Έμπειρο Σύστημα μπορεί να διακριθεί από τα συμβατικά προγράμματα στο ότι:

- ⇒ Μιμείται τον ανθρώπινο συλλογισμό για το γνωστικό πεδίο ενός προβλήματος, παρά να μιμείται το ίδιο το πεδίο. Αυτό διακρίνει τα έμπειρα συστήματα από τα συμβατικά προγράμματα τα οποία χρησιμοποιούν μαθηματικά μοντέλα. Αυτό δεν σημαίνει ότι το έμπειρο σύστημα αποτελεί ένα πιστό ψυχολογικό μοντέλο του ειδικού, αλλά στόχος είναι η μίμηση των ικανοτήτων του ειδικού στην επίλυση προβλημάτων του πεδίου, ώστε να εργάζεται κατά τον ίδιο ή καλύτερο τρόπο από τον ειδικό.
- ⇒ Πέρα από αριθμητικούς υπολογισμούς και ανάκτηση δεδομένων, εκτελεί συλλογισμούς πάνω σε αναπαραστάσεις της ανθρώπινης γνώσης. Η γνώση στο σύστημα εκφράζεται με από μια ειδική γλώσσα και είναι διαχωρισμένη από τον κώδικα που εκτελεί τους συλλογισμούς. Αυτά τα στελέχη του προγράμματος χαρακτηρίζονται ως βάση γνώσης (knowledge-base) και μηχανισμός εξαγωγής συμπερασμάτων (inference engine), αντίστοιχα.
- ⇒ Επιλύει προβλήματα με ευρετικές ή προσεγγιστικές μεθόδους οι οποίες σε αντίθεση με τις αλγοριθμικές λύσεις δεν εγγυώνται επιτυχία. Οί ευρετικές μέθοδοι βασίζονται κυρίως σε εμπειρικούς κανόνες. Ο λόγος που καλούνται προσεγγιστικές, είναι ότι αφενός δεν απαιτούν τέλεια δεδομένα εισόδου και αφετέρου τα αποτελέσματά τους γίνονται αποδεκτά με κάποια αβεβαιότητα.

Ένα Έμπειρο Σύστημα διαφέρει από άλλα προγράμματα Τεχνητής Νοημοσύνης στο ότι:

- ⇒ Ασχολείται με προβλήματα ρεαλιστικής σύνθεσης που κανονικά απαιτούν την επέμβαση κάποιου ειδικού. Πολλά προγράμματα Τεχνητής Νοημοσύνης έχουν κυρίως ερευνητικούς σκοπούς και ασχολούνται με αφηρημένα μαθηματικά προβλήματα ή απλοποιημένα πραγματικά προβλήματα ('toy' problems). Αντίθετα τα Έμπειρα

Συστήματα ασχολούνται με πραγματικά προβλήματα επιστημονικής ή εμπορικής φύσεως.

- ⇒ Ένα Έμπειρο Σύστημα για να είναι χρήσιμο οφείλει να παρουσιάζει υψηλές αποδόσεις όσον αφορά την ταχύτητα και την αξιοπιστία. Αντίθετα τα προγράμματα Τεχνητής Νοημοσύνης μπορούν και να είναι αργά και να περιέχουν προγραμματιστικά σφάλματα αφού δεν αποτελούν υποστηρικτικό λογισμικό. Ωστόσο ένα Έμπειρο Σύστημα οφείλει να παράγει λύσεις γρήγορα και με ποσοστό επιτυχίας ανάλογο αυτού των ειδικών.
- ⇒ Οφείλει να εξηγεί και να αιτιολογεί τις λύσεις ή τις προτάσεις που παράγει, ώστε να πείθει για την ορθότητα τους. Σε αντίθεση τα προγράμματα Τεχνητής Νοημοσύνης εκτελούνται κυρίως από τους δημιουργούς τους ή άλλους από το εργαστηριακό προσωπικό.

Ο όρος “knowledge-based systems” συχνά χρησιμοποιείται ως συνώνυμο του όρου “expert systems” αν και είναι πιο γενικός. Knowledge-based system είναι οποιοδήποτε πρόγραμμα εκτελεί εμπειρικούς κανόνες πάνω σε μια συμβολική αναπαράσταση γνώσης, αντί να εκτελεί αλγοριθμικές ή στατιστικές μεθόδους επίλυσης. Αντίθετα ένα έμπειρο σύστημα συνδυάζει όλα τα παραπάνω. Για παράδειγμα ένα Knowledge-based system είναι ικανό να εκφέρει γνώμη γύρω από τον καιρό αν και δεν κατέχει κάποια εξειδίκευση, ενώ ένα “expert system” είναι ικανό να παράγει προβλέψεις για τον καιρό συνδυάζοντας την γνώση των εμπειρικών κανόνων με στατιστικά στοιχεία που παράγει. Η διαδικασία ανάπτυξης ενός έμπειρου συστήματος καλείται “knowledge engineering” και θεωρείται ως «Εφαρμοσμένη Τεχνητή Νοημοσύνη».

### 2.4.3 Θεμελιώδη ζητήματα των έμπειρων συστημάτων

Παρακάτω παρατίθενται τα τέσσερα πιο θεμελιώδη ζητήματα στην ανάπτυξη των Έμπειρων Συστημάτων, που όπως είναι αναμενόμενο απαντούνται σε μεγάλο βαθμό και στην Τεχνητή Νοημοσύνη.

#### Εκμείευση Γνώσης

Είναι η μεταφορά και ο μετασχηματισμός της ικανότητας επίλυσης προβλημάτων από κάποια πηγή γνώσης σε ένα πρόγραμμα. Αυτή η μεταφορά συνήθως επιτυγχάνεται από μια σειρά εκτενών συνεντεύξεων μεταξύ ενός μηχανικού γνώσης, ο οποίος κατά βάση είναι ένας επιστήμων υπολογιστών και του ειδικού ο οποίος καλείται να προσφέρει με τρόπο κατανοητό την γνώση του. Υπολογίζεται ότι με αυτή τη διαδικασία παράγονται δύο έως πέντε μονάδες γνώσης (π.χ. εμπειρικοί κανόνες) ημερησίως. Αυτός ο μικρός ρυθμός μεταφοράς της γνώσης, καταστεί την εκμείευση της γνώσης, ένα από τα μεγαλύτερα προβλήματα στην ανάπτυξη Έμπειρων Συστημάτων.

Υπάρχει πλήθος λόγων, για τους οποίους η παραγωγικότητα είναι τόσο μικρή. Παρακάτω, δίδονται μερικοί από αυτούς.

- ⇒ Ανάλογα με το πεδίο της μελέτης, υπάρχει μια τεχνική διάλεκτος που δεν είναι εύκολο να γίνει κατανοητή από τους μηχανικούς γνώσης.
- ⇒ Κάποια γεγονότα και αρχές που υπησέρχονται σε πολλά πεδία γνώσης δεν είναι εύκολο και ίσως δυνατόν να χαρακτηριστούν με μαθηματικά ή ντετερμινιστικά μοντέλα. Για παράδειγμα ένας οικονομολόγος σπεσιαλίστας μπορεί να γνωρίζει την

αντίδραση της αγοράς σε συγκεκριμένα γεγονότα, ωστόσο δεν μπορεί να γνωρίζει τους ακριβείς μηχανισμούς που κινούνται για να προκληθούν αυτά τα αποτελέσματα (αντίδραση αγοράς, άνοδος - κάθοδος) καθώς και τον αντίκτυπο αυτών των αποτελεσμάτων. Με την βοήθεια στατιστικών στοιχείων έχουμε επιτύχει σημαντικές μακροχρόνιες προβλέψεις, πράγμα που δεν συμβαίνει όσο ο χρονικός ορίζοντας συρρικνώνεται.

- ⇒ Οι ειδικοί χρειάζεται να γνωρίζουν κάτι παραπάνω από τα δεδομένα και τις αρχές του πεδίου τους. Για παράδειγμα, ξέρουν τι είδους πληροφόρηση είναι κατάλληλη για τι είδους κρίσεις, πόσο αξιόπιστη είναι μια πηγή, πώς να απλοποιήσουν ένα πρόβλημα ώστε να είναι ευκολότερη η επίλυσή του. Η απόκτηση τέτοιου είδους γνώσης που συχνά προέρχεται από την εμπειρία των ειδικών και όχι από εκπαίδευση, είναι πολύ δυσκολότερο να αποκτηθεί.
- ⇒ Η εμπλοκή της κοινής λογικής με την εξειδικευμένη γνώση.
- ⇒ Η απογοήτευση από την μέθοδο της συνέντευξης οδήγησε πολλούς επιστήμονες στην προσπάθεια αυτοματοποίησης της εκμάτευσης της γνώσης μέσω ειδικών προγραμμάτων και τεχνικών μηχανικής μάθησης χωρίς σημαντικά αποτελέσματα έως σήμερα.

### **Αναπαράσταση της γνώσης**

Η αναπαράσταση της γνώσης αποτελεί ένα σημαντικό πεδίο έρευνας το οποίο συνδέεται άρρηκτα τόσο με την επίσημη όσο και με την διανοητική φιλοσοφία. Ασχολείται με τους τρόπους με τους οποίους η πληροφορία αποθηκεύεται και αλληλοσχετίζεται στον ανθρώπινο εγκέφαλο, συνήθως μέσα από ένα λογικό παρά βιολογικό πρίσμα. Με άλλα λόγια δεν ασχολείται με τις φυσικές λεπτομέρειες της αποκωδικοποίησης της γνώσης, αλλά για την μορφή που μπορεί να έχει το όλο νοητικό σχήμα.

Στη δεκαετία του 70' η έρευνα της αναπαράστασης της γνώσης, προσπάθησε να απαντήσει ερωτήσεις όπως πώς λειτουργεί η ανθρώπινη μνήμη, προτείνοντας θεωρίες για την υπενθύμιση, την αναγνώριση και την ανάκληση. Μερικές από τις προκύπτουσες θεωρίες οδήγησαν σε προγράμματα υπολογιστών τα οποία προσπαθούσαν να προσομοιώσουν διαφορετικούς τρόπους με τους οποίους οι διάφορες έννοιες μπορεί να συσχετίζονται, έτσι ώστε οι εφαρμογές υπολογιστών να είναι ικανές να βρουν το κατάλληλο κομμάτι γνώσης, την κατάλληλη στιγμή όταν επιλύουν ένα πρόβλημα. Με τον καιρό, η ψυχολογική ακριβολογία αυτών των θεωριών έγινε λιγότερο σημαντική από την χρησιμότητά τους ως μέσα πειραματισμού με νέα δεδομένα και νέες δομές ελέγχου, τουλάχιστον από το πλευρά της Τεχνητής Νοημοσύνης.

Η Αναπαράσταση της Γνώσης πάντα αποτελούσε και είναι πιθανό να εξακολουθήσει να αποτελεί ένα ιδιαίτερα αμφιλεγόμενο πεδίο. Σύχνά οι φιλόσοφοι και οι ψυχολόγοι εκπλήσσονταν αρνητικά από την ευχέρεια που οι ερευνητές Τεχνητής Νοημοσύνης χρησιμοποιούσαν λέξεις από την φιλοσοφία, την γλωσσολογία, την ψυχολογία, την λογική και την επιστήμη υπολογιστών σε μια τεχνική διάλεκτο που είχαν δημιουργήσει. Ωστόσο, στην αντίπερα όχθη, η μεταφορά του υπολογιστή, παρείχε ένα πρωτότυπο τρόπο ερώτησης και περιστασιακά απάντησης, δύσκολων ερωτήσεων οι οποίες μέχρι τότε άνηκαν στην σφαίρα του μεταφυσικού.

Στον κόσμο των Έμπειρων Συστημάτων, η Αναπαράσταση της Γνώσης έχει να κάνει με την εύρεση τρόπων με τους οποίους μεγάλο μέγεθος πληροφόρησης, μπορεί να τυποποιηθεί ώστε να είναι κατάλληλο για συμβολικούς υπολογισμούς. Με τον όρο

τυποποίηση, εννοούμε την έκφραση της πληροφόρησης σε μια απερίφραστη γλώσσα (ή συμβολισμό), η οποία διαθέτει ένα καλά ορισμένο συντακτικό, το οποίο διαχειρίζεται την μορφή των εκφράσεων στην γλώσσα, και μια καλά ορισμένη σημαντική η οποία αποκαλύπτει το νόημα των εκφράσεων.

Με την φράση συμβολικοί υπολογισμοί εννοούμε μη αριθμητικούς υπολογισμούς, στους οποίους τα σύμβολα και οι δομές τους μπορούν να ερμηνευθούν ως αναπαραστάσεις των μεταξύ των σχέσεων.

Οι ερευνητές της Τεχνητής Νοημοσύνης, έχουν κάνει σημαντικές προσπάθειες για την κατασκευή Γλωσσών Αναπαράστασης της Γνώσης, δηλαδή, γλωσσών υπολογιστών οι οποίες στοχεύουν στην οργάνωση των περιγραφών αντικειμένων και ιδεών, παρά στην δήλωση της αλληλουχίας ενεργειών ή στην αποθήκευση δεδομένων. Τα βασικά κριτήρια για τον ορισμό μιας αναπαράστασης της γνώσης είναι λογική επάρκεια, ευρετική δύναμη και παραστατική ευκολία. Παρακάτω αναλύουμε αυτούς τους όρους.

Λογική επάρκεια σημαίνει ότι η αναπαράσταση πρέπει να είναι ικανή να κάνει όλες τις διακρίσεις που επιθυμούμε. Για παράδειγμα, δεν είναι δυνατόν να αναπαραστήσουμε την ιδέα ότι όλα τα φάρμακα έχουν κάποια παρενέργεια πριν συνδέσουμε συγκεκριμένα φάρμακα με συγκεκριμένες παρενέργειες (π.χ. η ασπιρίνη επιβαρύνει το έλκος).

Ευρετική δύναμη σημαίνει ότι η αναπαράσταση πρέπει να κατάλληλη για την επίλυση προβλημάτων. Συνήθως, όσο πιο εκφραστική είναι μια αναπαράσταση, δηλαδή όσο πιο πολλές σημαντικές διακρίσεις είναι ικανή να διεκπεραιώσει, τόσο πιο δύσκολη είναι η εξαγωγή συμπερασμάτων κατά την επίλυση προβλημάτων. Οι περισσότερες από τις αναπαραστάσεις που έχουν υιοθετηθεί έως σήμερα, συνήθως υπολείπονται σε εκφραστικότητα σε σύγκριση με γλώσσες, όπως η Αγγλική, ωστόσο κερδίζουν σε ευρετική δύναμη. Δηλαδή, είναι ευκολότερο να ανακληθεί η κατάλληλη γνώση την κατάλληλη στιγμή. Η γνώση, πια γνωστική περιοχή είναι πιο σχετική με το εκάστοτε πρόβλημα, είναι αυτή που ξεχωρίζει έναν ειδικό από έναν αρχάριο ή έναν απλά καλοδιαβασμένο.

Η Παραστατική Ευκολία αποτελεί μεγάλη αρετή για μια αναπαράσταση καθώς στα περισσότερα Έμπειρα Συστήματα απαιτείται η αποκωδικοποίηση σημαντικού μεγέθους γνώσης και κάτι τέτοιο δεν θα ήταν δυνατό εάν οι συμβάσεις της αναπαράστασης ήταν ιδιαίτερα μπερδεμένες. Οι προκύπτουσες εκφράσεις οφείλουν να είναι ευκατανόητες ακόμα και όταν δεν γνωρίζουμε πως πρόκειται ο υπολογιστής να τις συνδυάσει και να τις ερμηνεύσει.

Εώς σήμερα οι σημαντικότερες συμβάσεις που έχουν προταθεί είναι:

⇒ Κανόνες παραγωγής - Production rules (Davis and King, 1977)

⇒ Δομημένα αντικείμενα - Structured objects (Findler, 1979)

⇒ Λογικά προγράμματα - Logic programs (Kowalski, 1979)

Τα περισσότερα Έμπειρα Συστήματα χρησιμοποιούν μία ή περισσότερες από τις παραπάνω συμβάσεις, των οποίων τα πλεονεκτήματα και μειονεκτήματα αποτελούν μέχρι και σήμερα αντικείμενο διαμάχης.

### **Έλεγχος του συλλογισμού**

Ο σχεδιασμός ενός έμπειρου Συστήματος απαιτεί μεγάλη προσοχή στο πως προσεγγίζεται και εφαρμόζεται η γνώση, κατά την διαδικασία εύρεσης μιας λύσης. Η

γνώση του τι ακριβώς γνωρίζει κάποιος και πώς και πότε να κάνει χρήση της γνώσης του, είναι μια σημαντική έκφανση της ειδίκευσης. Αυτή η γνώση, συχνά χαρακτηρίζεται ως μέτα-γνώση (meta-knowledge), δηλαδή γνώση για την γνώση. Για την επίλυση μη συνηθισμένων προβλημάτων απαιτείται σημαντικός σχεδιασμός και έλεγχος κατά την επιλογή των ερωτήσεων ή των τέστ που είναι να πραγματοποιηθούν. Οι διαφορές στρατηγικές που χρησιμοποιούνται για την ανάδυση της κατάλληλης γνώσης παίζουν ρόλο στην απόδοση των προγραμμάτων. Αυτές ορίζουν τον τρόπο με τον οποίο ένα πρόγραμμα αναζητά την λύση σε ένα σύνολο εναλλακτικών.

### **Επεξήγηση των λύσεων**

Η ερώτηση για το πώς να βοηθηθεί ένας χρήστης να καταλάβει την δομή και την λειτουργία ενός πολύπλοκου λογισμικού σχετίζεται με το σχετικά νέο ερευνητικό πεδίο της αλληλεπίδρασης ανθρώπων - υπολογιστών, το οποίο αναδύθηκε από την τομή της Τεχνητής Νοημοσύνης, της Ψυχολογίας, της Μηχανικής και της Εργονομίας. Η συνεισφορά των ερευνητών των Έμπειρων Συστημάτων, έχει να κάνει με την απαίτηση τους, τα συμβουλευτικά προγράμματα, να «λογοδοτούν», και με την υπόδειξή τους, πώς η επεξήγηση της συμπεριφοράς των προγραμμάτων μπορεί συστηματικά να συσχετιστεί με την αλληλουχία του συλλογισμού που εκπονείται από αυτά τα συστήματα. Η ανάλυση της συμπεριφοράς των Έμπειρων συστημάτων, είναι σημαντική για πολλούς λόγους:

- ⇒ Οι χρήστες των προγραμμάτων επιθυμούν να γνωρίζουν ότι όντως τα συμπεράσματα των προγραμμάτων είναι σωστά.
- ⇒ Οι μηχανικοί γνώσης επιθυμούν να γνωρίζουν ότι η γνώση χρησιμοποιείται κατάλληλα και σύμφωνα με τα πρωτότυπα που έχουν δημιουργήσει.
- ⇒ Οι ειδικοί επιθυμούν να γνωρίζουν πώς χρησιμοποιείται η γνώση τους, ώστε να κρίνουν εάν η εκμείυσή της προχωρά επιτυχημένα.
- ⇒ Οι προγραμματιστές οι οποίοι συντηρούν, αποσφαλματώνουν και επεκτείνουν τα προγράμματα, χρειάζονται ένα παράθυρο στην συμπεριφορά του προγράμματος σε ανώτερο επίπεδο από αυτό του κώδικα.
- ⇒ Τα υπεύθυνα στελέχη μιας επιχείρησης επιθυμούν να γνωρίζουν ότι ο συλλογισμός του προγράμματος ταιριάζει στο γνωστικό πεδίο της επιχείρησης.

Συχνά αντί της φράσης «Επεξήγηση των λύσεων» χρησιμοποιείται ο όρος «διαφάνεια», με σκοπό να τονιστεί η ευκολία με την οποία μπορεί κάποιος να καταλαβεί τι κάνει ένα πρόγραμμα και γιατί.

Εάν το εγχείρημα απαιτεί, αισθητήρια ή μηχανικά προσώντα πέρα από τις δυνατότητες της διαθέσιμης τεχνολογίας στην ρομποτική και στην αίσθηση των υπολογιστών.

### **2.4.4 Έλεγχος της λειτουργίας του διερμηνέα**

Ο διερμηνέας ενός συνόλου κανόνων μπορεί να περιγραφεί μέσω του κύκλου «αναγνώρισης - δράσης», ο οποίος αποτελείται από τα παρακάτω βήματα.

1. Ταίριασμα των μελών των υποθέσεων των κανόνων με στοιχεία της μνήμης εργασίας.



2. Εάν ικανοποιούνται οι συνθήκες πολλαπλών κανόνων, τότε διάλεξε ποιός θα ενεργοποιηθεί. Αυτό το βήμα ονομάζεται ανάλυση συγκρούσεων (conflict resolution).
3. Εφαρμογή του κανόνα, ίσως γράφοντας ή σβήνοντας ένα στοιχείο από την μνήμη εργασίας. Πήγαινε στο βήμα 1.

Συχνά, εισάγουμε ένα στοιχείο εκκίνησης στην μνήμη εργασίας, το οποίο αντιπροσωπεύει την αρχική κατάσταση του συστήματος. Στην CLIPS, αυτό το στοιχείο εκκίνησης είναι το διάνυσμα (initial - fact).

Ο υπολογισμός τερματίζεται εάν σε έναν κύκλο δεν ενεργοποιηθεί κανένας κανόνας ή όταν ένας κανόνας που ενεργοποιείται περιέχει ρητή εντολή τερματισμού.

Στο βήμα 2, το σύστημα έχει ένα σύνολο ζευγαριών αποτελούμενα από κανόνες και τιμές μεταβλητών, που προκύπτουν από το ταίριασμα των συνθηκών των κανόνων. Αυτά τα ζευγάρια αποκαλούνται στιγμιότυπα. Η ανάλυση συγκρούσεων απαντά στο ερώτημα ποιός κανόνας να ενεργοποιηθεί. Βέβαια, είναι δυνατόν η κατασκευή συνόλων κανόνων που μόνο οι συνθήκες ενός κανόνα είναι δυνατόν να ικανοποιούνται. Τέτοια σύνολα κανόνων καλούνται αιτιοκρατικά (deterministic), αλλά είναι μάλλον σπανιότερες οι περιπτώσεις που θα μας πασχολήσουν σε ένα έμπειρο σύστημα.

Ο έλεγχος της συμπεριφοράς των συστημάτων που βασίζονται σε κανόνες, συχνά θέτει σημαντικά προβλήματα. Υπάρχουν δύο προσεγγίσεις ελέγχου: Ο ολικός και ο τοπικός έλεγχος. Ο ολικός έλεγχος, χρησιμοποιεί κυρίως γενικές τεχνικές ελέγχου παρά εξειδικευμένες τεχνικές, ειδικά για το πεδίο. Συνήθως, τέτοιες στρατηγικές είναι ενσωματωμένες στον κώδικα του διερμηνέα (όπως στην CLIPS) και είναι δύσκολο να παρακαμθούν από τους προγραμματιστές. Αντίθετα οι τεχνικές τοπικού ελέγχου, εξαρτώνται από το εκάστοτε πεδίο και εκφράζονται κυρίως με την μορφή μετα-κανόνων, δηλαδή κανόνων που επιδρούν στους κανόνες. Οι μετα-κανόνες συνήθως εισάγονται από τον προγραμματιστή.

### Ανάλυση Συγκρούσεων (Conflict resolution)

Μετά τον έλεγχο των συνθηκών όλων των κανόνων μπορούμε να δημιουργήσουμε μια λίστα με τους κανόνες που δύνανται να ενεργοποιηθούν. Αυτό το σύνολο συνήθως καλείται σύνολο συγκρούσεων (conflict set), ενώ στην CLIPS ονομάζεται agenda.

Η ανάλυση συγκρούσεων, κανονικά επιλέγει έναν από τους κανόνες του συνόλου. Γίνεται εύκολα κατανοητό ότι οι αποφάσεις που παίρνει η ανάλυση συγκρούσεων επηρεάζουν την έκβαση των αποτελεσμάτων που θα πάρουμε. Για αυτό τον λόγο, σε συστήματα όπως η CLIPS, που έχουν ενσωματωμένες τις τεχνικές ελέγχου του διερμηνέα, δίνεται η δυνατότητα παραμετροποίησης των τεχνικών αυτών ώστε να προσεγγίζουν όσο το δυνατόν περισσότερο τις ανάγκες του εκάστοτε προβλήματος.

Οι τεχνικές της ανάλυσης συγκρούσεων, είναι συνήθως συνδυασμός διαφορετικών βασικών μηχανισμών, ο καθένας εκ των οποίων έχει διαφορετικές ιδιότητες. Οι τρεις γνωστότεροι μηχανισμοί, είναι:

⇒ **Refractoriness:** Ένας κανόνας δεν πρέπει να επενεργεί πάνω από μια φορά στα ίδια δεδομένα. Ένας προφανής τρόπος για την ικανοποίηση αυτής της αρχής είναι η διαγραφή από το σύνολο συγκρούσεων, στιγμιότυπων προηγούμενων εκτελέσεων.

Εάν απλά επιθυμούμε την αποτροπή ανακυκλώσεων, μπορούμε απλά να διαγράψουμε τους κανόνες, που προκύπτουν από το πιο πρόσφατο στιγμιότυπο, από το σύνολο συγκρούσεων. Εάν τώρα επιθυμούμε μόνο για συγκεκριμένες περιπτώσεις να έχουμε ανακυκλώσεις, υπάρχει η εντολή “refresh” η οποία παρακάμπτει τον μηχανισμό.

**Χρονική Εγγύτητα (Recency):** Σε πολλά συστήματα όπως το CLIPS, τα στοιχεία της μνήμης εργασίας φέρουν χρονικές ενδείξεις, δηλαδή ενδείξεις για το πότε εισήχθησαν στην μνήμη εργασίας. Σύμφωνα με τον εν λόγω μηχανισμό, οι κανόνες που προτιμούνται είναι αυτοί των οποίων οι συνθήκες επαληθεύθηκαν από τα πιο πρόσφατα στιγμιότυπα.

⇒ **Ιδιαιτερότητα (Specificity):** Στιγμιότυπα τα οποία προήλθαν από ειδικότερους κανόνες, δηλαδή από κανόνες με περισσότερα ορίσματα οι οποίοι ενεργοποιούνται δυσκολότερα, προτιμούνται. Αυτός ο μηχανισμός είναι ιδιαίτερα χρήσιμος στην αντιμετώπιση εξαιρέσεων σε γενικούς κανόνες.

⇒ Η ανάλυση συγκρούσεων είναι αυτή που εξασφαλίζει την ευρετική δύναμη και τον έλεγχο της συμπεριφοράς των έμπειρων συστημάτων. Δίχως την ανάλυση συγκρούσεων θα ήταν δύσκολη, ίσως και αδύνατη η αντιμετώπιση των μη ντετερμινιστικών προβλημάτων και των εξαιρέσεων των κανόνων.

Στην CLIPS, ο προγραμματιστής έχει την δυνατότητα να συνδέσει με κάθε κανόνα μια ιδιότητα προτίμησης. Έτσι το σύνολο συγκρούσεων, ταξινομείται βάσει της τιμής προτίμησης των κανόνων και κατόπιν επενεργούν οι υπόλοιποι τρεις μηχανισμοί για την περαιτέρω ταξινόμηση των υποσυνόλων ίδιας τιμής προτίμησης.

### **Ορθή και Ανάστροφη συλλογιστική (Forward and backward chaining)**

Υπάρχουν δύο χαρακτηριστικές μέθοδοι που εφαρμόζονται κατά την έρευνα στο πεδίο καταστάσεων για τον εντοπισμό της λύσης ενός προβλήματος, και συγκεκριμένα οι μέθοδοι της ορθής και της ανάστροφης συλλογιστικής.

Η μέθοδος της ορθής συλλογιστικής ξεκινά την έρευνα για την ανεύρεση νέας πληροφορίας από την ήδη υπάρχουσα γνώση. Για τον λόγο αυτό η μέθοδος αυτή χαρακτηρίζεται ως καθοδηγούμενη από τα δεδομένα (data-driven) ή σαν πορεία που ξεκινά από την βάση και προχωρά προς το αποδεικτέο (bottom-up).

Περίπτωση ορθής συλλογιστικής, είναι η δημιουργία καρκινικών προτάσεων, με την χρήση κανόνων της μορφής

(P1)\$ → a\$a

(P2)\$ → b\$b

(P3)\$ → c\$c,

ξεκινώντας από ένα αρχικό σύμβολο. Για παράδειγμα ξεκινώντας από το σύμβολο  $c$ , εφαρμόζουμε τον κανόνα P2, παίρνουμε  $bc b$ . Στο νέο δεδομένο εφαρμόζουμε τον κανόνα P1 και παίρνουμε  $abcba$ , κ.ο.κ.

Μειονέκτημα της ορθής συλλογιστικής είναι η μεγάλη πιθανότητα αποπροσανατολισμού του συστήματος, αφού δεν γίνεται έλεγχος σύγκλισης προς κάποια αποδεδειγμένη κατάσταση.

Η ανάστροφη συλλογιστική ξεκινά από το αποδεδειγμένο και προσπαθεί να καταλήξει σε ήδη γνωστά δεδομένα. Για αυτό το λόγο ονομάζεται επίσης μέθοδος προσανατολισμένη στο αποδεδειγμένο (goal oriented) ή εκ των άνω προς τα κάτω (top-down).

Περίπτωση ανάστροφης συλλογιστικής είναι η αναγνώριση καρκινικών προτάσεων, σύμφωνα με τους παραπάνω κανόνες. Για παράδειγμα, δεδομένης της πρότασης, προς απόδειξη,  $bacab$  εξάγουμε το συμπέρασμα ότι είναι προϊόν εφαρμογής του κανόνα P2, πάνω στην πρόταση  $aca$ , η οποία με τη σειρά της είναι αποτέλεσμα εφαρμογής του κανόνα P1 πάνω στην πρόταση  $c$ . Το σύμβολο  $c$ , αποτελεί αξίωμα (δεδομένο) και άρα η διαδικασία τερματίζεται.

Κλείνοντας σημειώνουμε, ότι είναι δυνατόν να συνδυάσουμε τις δύο μεθόδους συλλογισμού. Σε αυτές τις περιπτώσεις, σαν κύρια μέθοδος χρησιμοποιείται η ανάστροφη συλλογιστική και καλείται περιστασιακά η ορθή συλλογιστική.

### **Κανόνες και μέτα-κανόνες (rules and meta-rules)**

Ο κώδικας, του κάθε κανόνα παραγωγής υποτίθεται ότι είναι ανεξάρτητος, από την άποψη ότι όλη η απαραίτητη πληροφορία για την ενεργοποίηση ενός κανόνα εμπεριέχεται στις συνθήκες του κανόνα. Δεν υπάρχει δυνατότητα κλήσης του ενός κανόνα από έναν άλλο, όπως συμβαίνει με τις διαδικασίες. Έτσι ο μόνος τρόπος δρομολόγησης της ενεργοποίησης ενός κανόνα  $R$ , από ένα κανόνα  $R'$ , είναι μέσω κατάλληλων αλλαγών στην μνήμη εργασίας από τον κανόνα  $R'$  σε κάποια ανακύκλωση του συστήματος, ώστε στη συνέχεια να ικανοποιηθούν οι συνθήκες του  $R$ .

Μερικές φορές θα θέλαμε, να χρησιμοποιήσουμε γνώση για την πυροδότηση των κατάλληλων κανόνων, παρά να ακολουθήσουμε την στρατηγική της ανάλυσης συγκρούσεων. Έτσι μερικοί διερνηνείς συστημάτων παραγωγής, επιτρέπουν στον προγραμματιστή την εισαγωγή μέτα-κανόνων, οι οποίοι αποφασίζουν για το ποιός κανόνες από το σύνολο συγκρούσεων πρέπει να τεθούν στην μελέτη της ανάλυσης συγκρούσεων ή ακόμη και ποιός συγκεκριμένος κανόνας πρέπει να ενεργοποιηθεί, παρακάμπτοντας εντελώς την ανάλυση συγκρούσεων.

Οι μέτα-κανόνες διακρίνονται από τους συνηθισμένους κανόνες στο ότι καθοδηγούν τον συλλογισμό επίλυσης του προβλήματος παρά να εκτελούν αυτό τον συλλογισμό. Αυτή η διάκριση μεταξύ των διαφορετικών επιπέδων συλλογισμού, συχνά χαρακτηρίζεται ως διάκριση μεταξύ μέτα-επιπέδου (meta-level) και επιπέδου αντικειμένων (object-level). Έτσι, μπορούμε να έχουμε μέτα-κανόνες για την διενέργεια ανάλυσης συγκρούσεων ή για την ενίσχυση μιας ενσωματωμένης στρατηγικής ανάλυσης συγκρούσεων.

### **2.4.5 Σύνοψη Έμπειρων Συστημάτων**

Σε αυτό το κεφάλαιο τονίσαμε τις δυνατότητες της τεχνολογίας των Έμπειρων Συστημάτων και είδαμε σε ποιές περιπτώσεις είναι δυνατή η εφαρμογή της. Κλείνοντας,

απαριθμούμε τους αποφασιστικούς παράγοντες για το αν ένα Έμπειρο Σύστημα είναι κατάλληλο για κάποιο πρόβλημα, σημειώνουμε τα κριτήρια που πρέπει να ικανοποιεί ο ειδικός σύμβουλος και περιγράφουμε πιθανούς περιορισμούς στη αφομοίωση των ικανοτήτων του.

Οι τρεις πιο αποφασιστικοί παράγοντες για την καταλληλότητα ενός Ε.Σ. είναι:

1. Η φύση του προβλήματος,
2. η ύπαρξη κάποιου ειδικού και
3. η ικανότητα μεταφοράς της γνώσης και της στρατηγικής επίλυσης του προβλήματος στον υπολογιστή.

Ένας ειδικός πρέπει:

1. να είναι διαθέσιμος να εκτελέσει το εγχείρημα,
2. να ξέρει πως να το εκτελέσει,
3. να μπορεί να εξηγήσει πως το εκτελεί,
4. να έχει τον χρόνο να εξηγήσει πως το εκτελεί και
5. να έχει κίνητρα να συνεργαστεί.

Πιθανοί περιορισμοί στη αφομοίωση των ικανοτήτων του ειδικού, είναι:

1. Εάν το εγχείρημα απαιτεί, σε σημαντικό βαθμό, γνώση υποκρυπτόμενη ή κοινή λογική.
2. Εάν το εγχείρημα απαιτεί, αισθητήρια ή μηχανικά προσώνα πέρα από τις δυνατότητες της διαθέσιμης τεχνολογίας στην ρομποτική και στην αίσθηση των υπολογιστών.

### **3. Περιγραφή του προβλήματος της Προσομοίωσης και Κοστολόγησης Γραμμών Παραγωγής**

#### **3.1 Το πρόβλημα**

Η εταιρία «ΓΡΑΦΙΚΕΣ ΤΕΧΝΕΣ – Γ. ΔΕΤΟΡΑΚΗΣ ΑΕΒΕ» με έδρα το Ηράκλειο Κρήτης από το 1970, παρουσιάζει ένα αξιόλογο έργο στον χώρο της έντυπης διαφήμισης και των εκδόσεων. Με γνώμονα την διατήρηση αλλά και ενίσχυση της θέσης της στην αγορά, τα τελευταία δέκα χρόνια έχει ξεκινήσει μια προσπάθεια μηχανογράφησης και υπολογιστικού ελέγχου των λειτουργιών της.

Πρώτο βήμα σε αυτήν την προσπάθεια ήταν η ανάπτυξη μιας ERP (Enterprise Resource Planning) εφαρμογής, υλοποιημένης σε Cobol και η οποία λειτουργεί σε UNIX περιβάλλον. Οι λειτουργίες του προγράμματος περιλαμβάνουν διαχείριση αποθεμάτων και ανθρώπινου δυναμικού, έλεγχο λογιστηρίου και διακίνηση τελικών προϊόντων. Η εφαρμογή αυτή, πραγματικά απέδωσε στην βελτιστοποίηση της παραγωγικής διαδικασίας και του ελέγχου των αποθεμάτων και ήταν αρκετή για την πιστοποίηση της εταιρίας με ISO 9001.

Αφού ο έλεγχος της παραγωγικής διαδικασίας, έχει επιτυχώς αντιμετωπιστεί, το ενδιαφέρον της εταιρίας πλέον στρέφεται στην αντιμετώπιση του μεγάλου προβλήματος της σωστής και γρήγορης κοστολόγησης προϊόντων και υπηρεσιών κυρίως στο στάδιο της σύνταξης προσφορών. Η παρούσα αδυναμία γρήγορης και σωστής σύνταξης προσφορών, αποτελεί ίσως το μεγαλύτερο εμπόδιο στην αποτελεσματικότητα των πωλητών της εταιρίας.

#### **3.2 Παρούσα Κατάσταση**

Η γκάμα προϊόντων της εταιρίας χαρακτηρίζεται από υψηλή ποικιλότητα τόσο όσον αφορά τον τύπο των προϊόντων που παράγονται όσο και τις ποσότητες (τιράζ) των προϊόντων που παράγονται. Αυτό το γεγονός έχει σαν αποτέλεσμα διάφορους περιορισμούς κατά την παραγωγική διαδικασία όπως μεγάλοι χρόνοι προετοιμασίας των μηχανών ώστε να επεξεργαστούν διαφορετικά προϊόντα (καθάρισμα μηχανών, μετρήματα, προθέρμανση μηχανών, φόρτωση και εκφόρτωση πρώτων υλών). Επίσης συχνά απαιτείται και σημαντικός χρόνος για τον προσδιορισμό των προδιαγραφών ενός ιδιαίτερου προϊόντος μέσω της διεπαφής με τον πελάτη. Το κόστος της πρώτης ύλης, η αναλογία κόστους προς τιράζ, το περιβάλλον χρήσης του εκάστοτε εντύπου και ο σκοπός χρήσης του κάθε εντύπου είναι μερικές μόνο από τις παραμέτρους που αναλύει το ειδικό υποστηρικτικό προσωπικό της εταιρίας στους υποψήφιους πελάτες.

Για τους παραπάνω λόγους, η παραγωγή του προϊόντος χωρίζεται σε επί μέρους διαδικασίες συνοδευόμενες από οδηγούς κόστους. Οι διαδικασίες είναι επιλεγμένες με τρόπο τέτοιο ώστε αφενός να είναι δυνατός ο ορισμός λεπτομερών οδηγιών κόστους και αφετέρου το πλήθος τους να είναι όσο το δυνατόν μικρότερο προσβλέποντας στην ευχρηστία και την οικονομία του συστήματος κοστολόγησης. Χαρακτηριστικοί παράμετροι των οδηγιών κόστους είναι:

- ⇒ Χρόνος δέσμευσης εργαζομένου. Μέσω του ERP, γνωρίζουμε ποιος εργαζόμενος δούλεψε σε ποία μηχανή, πότε και για πόσο χρόνο και πόσο αμείβεται ο συγκεκριμένος εργαζόμενος.
- ⇒ Εργασία μηχανής. Μέσω του ERP, γνωρίζουμε πόσο κοστίζει η εργασία κάθε μηχανής βάσει του χρόνου εργασίας της, των αναλώσιμων, των ρυθμών

παραγωγής, της ποιότητας χαρτιού που της διοχετεύουμε και του σταθερού κόστους λειτουργίας της.

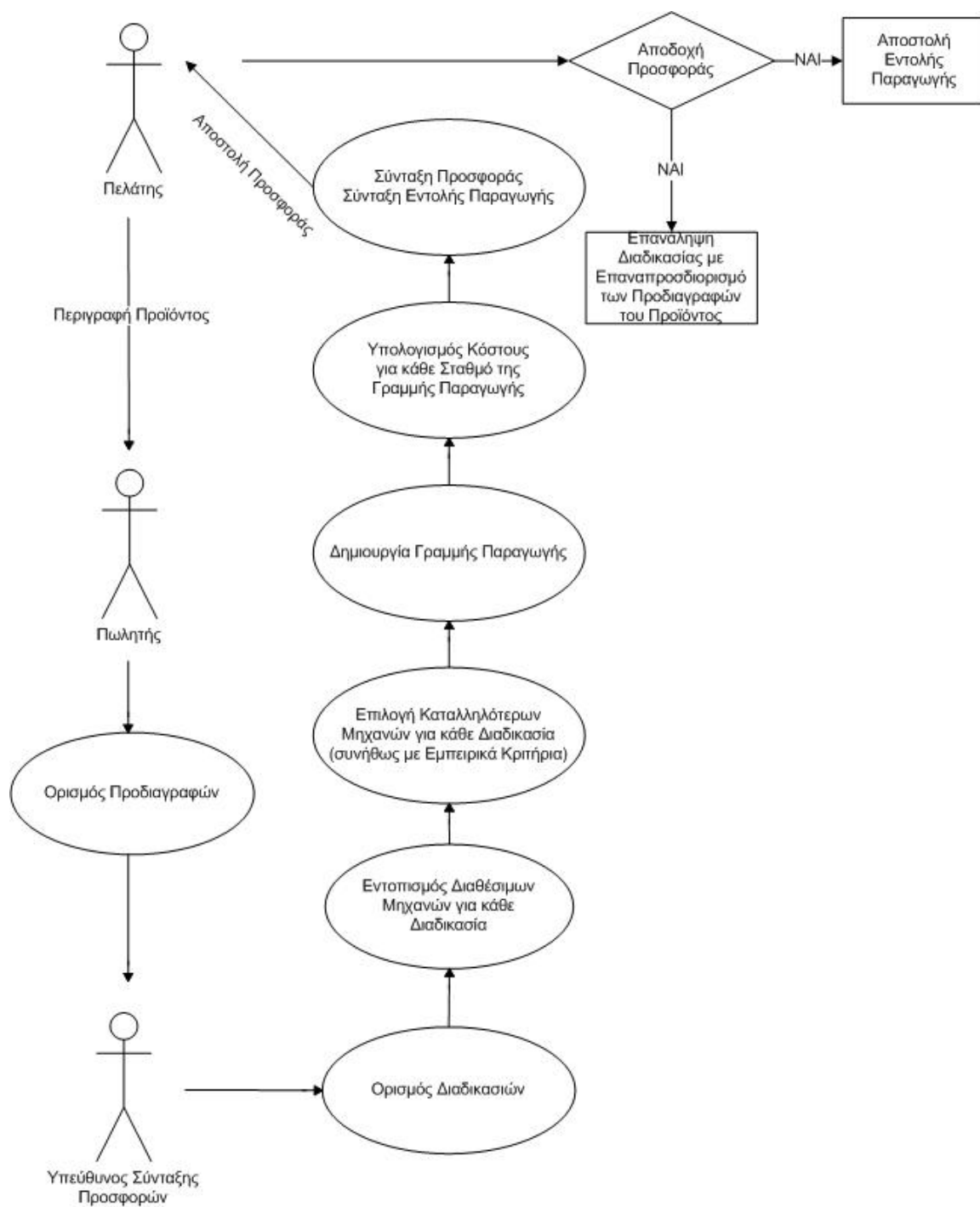
- ⇒ Τρέχουσες τιμές προμήθειας πρώτων υλών. Η παραγωγική διαδικασία χαρακτηρίζεται από μεγάλη ποικιλότητα πρώτων υλών. Ιδιαίτερα, όσον αφορά τα χαρτιά που χρησιμοποιούνται στις εκτυπώσεις μπορούμε να διακρίνουμε πάνω από χίλιους διαφορετικούς τύπους αφού τα χαρτιά ταξινομούνται ανάλογα το μέγεθός τους, το βάρος τους, την ποιότητά τους και τον αν έρχονται σε φύλλα ή ρολά. Το γεγονός αυτό, συνεπάγεται ότι κατά την σύνταξη μιας προσφοράς είναι απαραίτητο να υπάρχει συνεχής ενημέρωση του στελέχους που συντάσσει την προσφορά, για τις πιο πρόσφατες τιμές των πρώτων υλών. Καθημερινά παραγγέλνονται πάνω από 10 διαφορετικά χαρτιά.
- ⇒ Κόστη που προκύπτουν από πρακτική εμπειρία. Για παράδειγμα κατά το στήσιμο των offset εκτυπωτικών μηχανών καταστρέφονται δέκα (κατά μέσο όρο) εκτυπώσεις για τη ρύθμιση της βαρύτητας των μελανιών. Αυτό το κόστος αν και υπολογίζεται προσεγγιστικά συχνά είναι αμελητέο για μεγάλα τιράζ. Η συμμετοχή του στο ολικό κόστος παρτίδας κυμαίνεται από 0 (μεγάλες παρτίδες) έως 1 (μικρές παρτίδες) τοις εκατό.

Σημείωση: Στην σύνταξη προσφορών οι χρόνοι εργασίας των μηχανών ορίζονται ως ο λόγος τιράζ προς ρυθμός παραγωγής, ενώ ο χρόνος εργασίας εργαζομένου ισούται με το άθροισμα χρόνος εργασίας μηχανής συν χρόνος προετοιμασίας μηχανής συν χρόνος καθαρίσματος μηχανής οι οποίοι είναι σταθεροί και δεδομένοι για κάθε μηχανή. Τέλος σε κάθε μηχανή εργάζονται συγκεκριμένοι τεχνίτες με γνωστές απολαβές.

Το πρόβλημα που αντιμετωπίζεται είναι ότι αυτός ο τρόπος κοστολόγησης που βασίζεται σε δραστηριότητες, απαιτεί σημαντικό χρόνο σύνταξης μιας προσφοράς, αφού για κάθε τύπο προϊόντος απαιτούνται πολλαπλές βασικές διαδικασίες, ενώ αν ένα προϊόν απαιτεί περαιτέρω φινιρίσμα ο αριθμός των διαδικασιών μπορεί να αυξηθεί σημαντικά. Εάν τώρα πρέπει να εξετάσουμε διάφορες εναλλακτικές περιπτώσεις ο χρόνος σύνταξης μιας προσφοράς εκτοξεύεται σε υψηλότατα επίπεδα, εάν βέβαια αναλογιστούμε ότι ο ορισμός των διαδικασιών πρέπει να γίνει από υψηλόμισθους ειδικούς τεχνικούς. Για παράδειγμα ένας πελάτης είναι δυνατόν να ζητήσει τιμές για ένα έντυπο παραλλάσσοντας παραμέτρους όπως:

- ⇒ Μέγεθος παρτίδας. Αλλάζει η αναλογία κόστος προς παρτίδα καθώς μειώνεται η συμμετοχή του κόστους διαδικασιών όπως τα μεταφορικά, ο σχεδιασμός του εντύπου, η έκδοση παραστατικών κ.ο.κ. Επίσης μεγάλα τιράζ διοχετεύονται σε μεγαλύτερες – οικονομικότερες μηχανές.
- ⇒ Ποιότητα και βάρος χαρτιού. Η ποιότητα του χαρτιού καθορίζει νεκρούς χρόνους μηχανής (π.χ. ο χρόνος στεγνώματος του μελανιού σε χαρτί Velvet είναι 8 ώρες, ενώ στις υπόλοιπες ποιότητες κυμαίνεται από μερικά δεύτερα έως 2 ώρες) ενώ το βάρος του αποκλείει κάποιες μηχανές.
- ⇒ Πρόσθετο φινιρίσμα. Τα πρόσθετα φινιρίσματα απαιτούν αφενός την χρήση πρόσθετων μηχανών και αφετέρου συχνά υπαγορεύουν τις κατάντες μηχανές. Για παράδειγμα τα πιεστήρια κοπτικών δεν μπορούν να δεχθούν φύλλα μεγέθους 70 x 100, οπότε πρέπει είτε να παρεμβάλουμε μια κοπτική μηχανή μετά την εκτύπωση, είτε να πάμε με μικρότερα φύλλα από την κοπή αποθήκης σε μικρότερη εκτυπωτική.

Τελικά, γίνεται αντιληπτό ότι η αυτοματοποίηση της παραπάνω διαδικασίας, αποτελεί πρωτεύουσα ανάγκη για την βελτίωση της αποτελεσματικότητας των στελεχών της επιχείρησης, τόσο στον τομέα της διοίκησης, όσο και στον τομέα του marketing.



Εικόνα 7: Παρούσα Διαδικασία Σύνταξης Προσφορών

### 3.3 Εκμαίευση Γνώσης

Ίσως η σημαντικότερη δυσκολία που αντιμετωπίστηκε στην ανάπτυξη του συστήματος ήταν η εκμαίευση της γνώσης των ειδικών της Γ. Δετοράκης ΑΕΒΕ. Παρακάτω παρατίθενται τα σημαντικότερα προβλήματα που συναντήσαμε:

Πρώτο κύριο πρόβλημα ήταν και εξακολουθεί να είναι τώρα που βρισκόμαστε στο στάδιο της εκσφαλμάτωσης, η διάθεση χρόνου από τους ειδικούς σε συνδυασμό πάντα με το γεγονός ότι η έδρα της επιχείρησης είναι στο Ηράκλειο.

Η επικοινωνία, ιδιαίτερα στις πρώτες τηλεφωνικές συνεντεύξεις ήταν σχεδόν ανύπαρκτη λόγω των τεχνικών όρων που χρησιμοποιούσαν οι ειδικοί και λεπτομερειών, που θεωρούσαν αυτονόητες ή δεδομένες, οπότε παρέλειπαν να τις αναφέρουν.

Τέλος, πριν προχωρήσουμε βαθύτερα στην εκμαίευση της γνώσης έπρεπε να καταλήξουμε στην μοντελοποίηση της επίλυσης του προβλήματος και στον τρόπο που θα αναπαριστούσαμε και θα επεξεργαζόμασταν την εισερχόμενη γνώση.

Στην αρχική προσέγγισή μας η γνώση θα ήταν συγκεντρωμένη στην βάση γνώσης και η επεξεργασία αυτής θα γινόταν από τον μηχανισμό συλλογισμού (inference engine) ενός Έμπειρου Συστήματος.

Αντίθετα στην λύση που τελικά καταλήξαμε, αυτή της χρήσης ενός πολυπρακτορικού συστήματος, δημιουργήθηκαν πράκτορες οι οποίοι αντιστοιχούσαν στις μηχανές του εργοστασίου, ο καθένας εκ των οποίων έφερε τη δικιά του βάση γνώσης την οποία ανακτούσε από μια βάση δεδομένων και η οποία είχε την μορφή κανόνων παραγωγής.

Τα μέσα που χρησιμοποιήθηκαν για την εκμαίευση της γνώσης ήταν:

- ⇒ Τηλεφωνικές συνεντεύξεις
- ⇒ Συνεντεύξεις πρόσωπο με πρόσωπο σε Χανιά και Ηράκλειο
- ⇒ Ξενάγηση στις εγκαταστάσεις παραγωγής του εργοστασίου στο Ηράκλειο και ανάλυση της παραγωγικής διαδικασίας.
- ⇒ Πίνακες Ενδεικτικών Προϊόντων - Μηχανών, Μηχανών – Διαδικασιών, Προδιαγραφών Μηχανών, Προδιαγραφών Χαρτιών
- ⇒ Μελέτη Δελτίων Εντολής Παραγωγής, Δελτίων Προσφοράς και Δελτίων Παραγγελίας

Στην συνέχεια αναφέρουμε κάποιες τεχνικές ορολογίες που πιστεύουμε ότι θα βοηθήσουν στην κατανόηση του προβλήματος που καλούμαστε να επιλύσουμε.



### **3.4 Βασικοί μέθοδοι εκτύπωσης**

#### **3.4.1 Ποίες είναι οι μέθοδοι εκτύπωσης**

Οι βασικές παραδοσιακές μέθοδοι εκτύπωσης είναι τέσσερις: Η Τυπογραφία, η Λιθογραφία ή Offset, η Βαθυτυπία, και η Μεταξοτυπία. Αυτές χωρίζονται σε επιμέρους ειδικές εφαρμογές όπως η φλεξογραφία. Υπάρχουν ακόμα νέες μέθοδοι εκτύπωσης (Ψηφιακή Εκτύπωση) μέσω ηλεκτρονικών υπολογιστών.

#### **3.4.2 Τι είναι η Ταμπονογραφία.**

Μια εξειδικευμένη μέθοδος εκτύπωσης πάνω σε μη επίπεδες επιφάνειες μικρών διαστάσεων είναι αυτή της Ταμπονογραφίας. Η εικόνα εκτυπώνεται αρχικά πάνω σε μια επίπεδη λεία μεταλλική επιφάνεια είτε με μεταξοτυπία είτε με άλλη μέθοδο (βαθυτυπία). Πριν το μελάνι στεγνώσει, ένα μαλακό "ταμπόν" από σιλικόνη έρχεται σε επαφή και παίρνει την εικόνα για να την μεταφέρει αμέσως πάνω στο αντικείμενο το οποίο μπορεί να έχει ανώμαλη ή κυρτή επιφάνεια.

#### **3.4.3 Τι είναι η Τυπογραφία ή αλλιώς Υψιτυπία.**

Η εκτύπωση γίνεται με ανάγλυφη εκτυπωτική πλάκα. Οι προς εκτύπωση περιοχές της πλάκας βρίσκονται στο ψηλότερο επίπεδο. Η πλάκα δημιουργείται από τον ειδικό τεχνίτη μέσα σε ένα μεταλλικό τελάρο χρησιμοποιώντας κινητά τυπογραφικά στοιχεία για τα γράμματα και τις γραμμές και «κλισέ» για τις εικόνες. Η πλάκα προσαρμόζεται στο τυπογραφικό πιεστήριο και για κάθε αντίτυπο που θα παράγει μελανώνεται με ελαστικούς κυλίνδρους και πιέζεται πάνω της το χαρτί. Στα παλαιότερα τυπογραφεία συναντάμε ακόμα τα Όρθια πιεστήρια που περιορίζονται σε έντυπα διάστασης έως 25 X 35 εκ. και τα Επίπεδα πιεστήρια για τις μεγαλύτερες διαστάσεις. Αν και στο παρελθόν ήταν η κυριότερη μέθοδος για την εκτύπωση των περισσότερων εντύπων, σήμερα ο ρόλος της μεθόδου είναι βοηθητικός και περιορίζεται σε συμπληρωματικές εργασίες όπως αρίθμηση, διάτρηση, κοπή με καλούπι, χρυσοτυπία, γκοφράρισμα κ.α

#### **3.4.4 Τι είναι η Φλεξογραφία.**

Σήμερα η κυριότερη βιομηχανική εφαρμογή της υψιτυπικής (τυπογραφικής) μεθόδου είναι η Φλεξογραφία με την οποία τυπώνονται κυρίως πολλά είδη συσκευασίας όπως σακούλες συνθετικές, τσάντες και κούτες. Βασίζεται σε ελαστικά κλισέ (λάστιχα, όπως συνηθίζεται να αποκαλούνται) που προσαρμόζονται σε κυλίνδρους, μελανώνονται και τυπώνουν.

#### **3.4.5 Τι είναι η Offset (Λιθογραφία ή Επιπεδοτυπία)**

Είναι η πλέον διαδεδομένη μέθοδος εκτύπωσης για χαρτί. Βασίζεται σε μια επίπεδη εκτυπωτική πλάκα από αλουμίνιο στην μια επιφάνεια της οποίας βρίσκεται σχηματισμένη η προς εκτύπωση εικόνα με μια ουσία ελαιόφιλη ή καλύτερα θα λέγαμε μελανόφιλη. Οι υπόλοιπες περιοχές της επιφάνειας που δεν προορίζονται να τυπώσουν είναι το καθαρό αλουμίνιο επεξεργασμένο ώστε κατά την διαδικασία της εκτύπωσης να συγκρατεί πάνω του μια λεπτή μεμβράνη νερού. Η εκτυπωτική πλάκα

προσαρμόζεται σε κύλινδρο και διαδοχικά διαβρέχεται με νερό και μελάνι. Μόνο οι μελανόφιλες περιοχές συγκρατούν μελάνι το οποίο μεταδίδουν σε έναν δεύτερο κύλινδρο με επιφάνεια από καουτσούκ. Μεταξύ αυτού του κυλίνδρου καουτσούκ και ενός τρίτου κυλίνδρου πίεσης, πιέζεται το χαρτί το οποίο δέχεται την εικόνα χωρίς να επηρεάζεται από το νερό το οποίο παραμένει στον πρώτο κύλινδρο. Η μέθοδος αυτή ονομάζεται λιθογραφία καθώς οι πρώτες εκτυπωτικές πλάκες ήταν κατασκευασμένες από επίπεδο λειασμένο ασβεστόλιθο. Η βιομηχανική εφαρμογή της μεθόδου με την μεταφορά της εικόνας από τον κύλινδρο της εκτυπωτικής πλάκας στον κύλινδρο καουτσούκ και από κει στο χαρτί, έδωσε το όνομα "offset" στις εκτυπωτικές αυτές μηχανές.

### **3.4.6 Τι είναι η Μεταξοτυπία**

Είναι η μέθοδος εκτύπωσης που βασίζεται σε ένα οριζόντιο πλαίσιο (τελάρο) όπου στην κάτω του πλευρά είναι τεντωμένη μια γάζα από καθαρές ίνες (παλιότερα ήταν μεταξωτή). Η εικόνα που πρόκειται να τυπωθεί βρίσκεται σχηματισμένη πάνω στη γάζα αφήνοντας ανοικτούς τους πόρους, ενώ οι μη εκτυπούμενες περιοχές είναι κλειστές. Το μελάνι εκχύνεται πάνω στην γάζα και οδηγείται, με την πίεση μιας λαστιχένιας σπάτουλας, να περάσει από τους ανοικτούς πόρους και να σχηματίσει την εικόνα στην επιφάνεια που βρίσκεται μόλις 2-3 χιλ κάτω και παράλληλα με το τελάρο.

### **3.4.7 Τι είναι η Ψηφιακή Εκτύπωση**

Άλλες μέθοδοι εκτύπωσης είναι η φωτοαντιγραφή, οι διάφοροι επιτραπέζιοι ηλεκτρονικοί εκτυπωτές inkjet, laser, due sublimation κ.α. οι οποίοι καλύπτουν ένα πολύ μεγάλο φάσμα τεχνολογιών που ονομάζουμε γενικά Ψηφιακή Εκτύπωση.

### **3.4.8 Τι είναι το halftone ή ράστερ, ή κουκίδα ή ημιτονικό.**

Η εκτύπωση με μελάνες, ανεξάρτητα από την μέθοδο εκτύπωσης, προϋποθέτει την επεξεργασία της εικόνας έτσι ώστε να διαχωρίζονται σαφώς οι περιοχές που θα τυπωθούν από αυτές που θα μείνουν ατύπωτες. Αυτό συμβαίνει διότι δεν είναι δυνατόν με μελάνια να επιτύχουμε την εκτύπωση συνεχούς τόνου όπως συμβαίνει π.χ. στις εικόνες που παραδίδουν τα φωτογραφεία που είναι τονικές εικόνες. Γι αυτό πρέπει να μετατρέψουμε την εικόνα αν χρειάζεται από τονική σε γραμμική, δηλαδή σε απόλυτους τόνους του μαύρου και του λευκού, χωρίς ενδιάμεσες διαβαθμίσεις. Είναι δυνατόν να δώσουμε την ψευδαίσθηση των τόνων με διάφορες μεθόδους, με το χέρι, φωτογραφικά ή μέσω προγραμμάτων στον ηλεκτρονικό υπολογιστή. Για τον σκοπό αυτό χρησιμοποιούνται διαφόρων ειδών "κουκίδες" (ή raster ή halftone ή screen) Η τονική εικόνα που επεξεργάζεται έτσι λέγεται ημιτονική. Η κουκίδα που παράγεται φωτογραφικά ή στον H/Y μπορεί να έχει διάφορα σχήματα και μπορεί να μετρηθεί ως προς την φωτεινότητα σε ποσοστό κάλυψης με μαύρο, ως προς την συχνότητα σε γραμμές ανά εκατοστό ή ανά ίντσα (LPI: Lines Per Inch) και ως προς την γωνία σε μοίρες που σχηματίζει με τον οριζόντιο άξονα. Είναι απαραίτητο να γνωρίζουμε τα χαρακτηριστικά αυτά γιατί επιδρούν αποφασιστικά στην ποιότητα της εικόνας.

### 3.4.9 Τι είναι το φιλμ

Στις περισσότερες περιπτώσεις το προς εκτύπωση θέμα (μακέτα, φωτογραφία, κείμενο κλπ) πρέπει να καταλήξει σε ένα ειδικό εργαστήριο προεκτύπωσης για να παραχθεί το φιλμ που είναι απαραίτητο για την κατασκευή των κατάλληλων για κάθε μέθοδο εκτύπωσης εκτυπωτικών πλακών. Το φιλμ που ονομάζεται λιθογραφικό ή γραμμικό φιλμ, είναι μια μεμβράνη διαφανής που έχει πάνω στην μια της πλευρά την εικόνα που προορίζεται για εκτύπωση, σχηματισμένη με μαύρη αδιαπέραστη από το φως, πάστα. Το φιλμ, ανάλογα με την μέθοδο εκτύπωσης για την οποία προορίζεται πρέπει να έχει τα κατάλληλα χαρακτηριστικά.

## 4. Σχεδίαση Συστήματος

Η σχεδίαση αυτού του συστήματος υποστήριξης αποφάσεων έγινε βασισμένη στη μεθοδολογία MaSE που περιγράψαμε σε προηγούμενο κεφάλαιο, μιας και το Σ.Υ.Α ενσωματώνει σαφώς την τεχνολογία πρακτόρων. Πρέπει να σημειώσουμε παρόλα αυτά πως ο γράφοντας δεν ακολούθησε πιστά τη MaSE καθώς ο σκοπός του Σ.Υ.Α. (η προσομοίωση της διαδικασίας παραγωγής τυπογραφικών προϊόντων με απότερο σκοπό της εξαγωγή λεπτομερών κοστολογίων) παρουσιάζει την ιδιαιτερότητα ο αριθμός και η δομή των πρακτόρων εργασιών να προσδιορίζεται δυναμικά κατά τη διάρκεια της εφαρμογής.

Σε προηγούμενο κεφάλαιο αναφέραμε ότι για την σχεδίαση του συστήματος, αρχικά χρησιμοποιήθηκε η τεχνολογία των Έμπειρων Συστημάτων. Οι λόγοι που εγκαταλήφθηκε αυτή η προσέγγιση και αντικαταστάθηκε από την χρήση ενός πολυπρακτορικού συστήματος το οποίο χρησιμοποιεί μερικώς της τεχνολογία των έμπειρων συστημάτων, είναι τρία.

- ⇒ Πρώτο σημαντικό πρόβλημα αποτέλεσε η απουσία κάποιου κατάλληλου κελύφους για την σχεδίαση της βάσης γνώσης. Τα κέλυφοι που χρησιμοποιήθηκαν παρουσίαζαν ένα ή περισσότερα από τα παρακάτω προβλήματα.
  - Ασυμβατότητα με τις σύγχρονες αντικειμενοστρεφείς γλώσσες προγραμματισμού.
  - Υψηλό κόστος απόκτησης.
  - Υψηλή πολυπλοκότητα σε συνδυασμό με ανύπαρκτη ή περιορισμένη τεκμηρίωση.
- ⇒ Το δεύτερο πρόβλημα που παρουσιάστηκε, ήταν το σημαντικό μέγεθος μαθηματικών υπολογισμών που πρέπει να εκτελεί το σύστημα, γεγονός που συνεπάγεται πλήθος παύσεων στην λειτουργία των διεργασιών των εκάστοτε κελύφων και αφετέρου υψηλές απώλειες όσον αφορά την ακρίβεια των αριθμητικών μεταβλητών λόγω των διαφορετικών τύπων δεδομένων που χρησιμοποιούσαν τα κέλυφοι σε σχέση με την γλώσσα προγραμματισμού που χρησιμοποιούσαμε (casting και boxing - unboxing).
- ⇒ Τέλος, η συγκέντρωση όλη της λογικής του συστήματος σε ένα μοναδικό αρχείο καθιστούσε δύσκολη έως και αδύνατη την εκσφαλμάτωση, συντήρηση και επεκτασιμότητα του συστήματος, ιδιαίτερα στην περίπτωση που οι εργασίες αυτές αναθέτονταν σε ομάδα εργασίας.

### 4.1 Μεθοδολογία σχεδίασης

Η μεθοδολογία σχεδίασης του Σ.Υ.Α. αυτού εντάσσεται σε μία γενικότερη μεθοδολογία ανάπτυξης του, η οποία ξεκινώντας από τη σχεδίαση του προχωρά στην ανάπτυξη του και τέλος συντηρεί και βελτιώνει το Σ.Υ.Α. έχοντας σαν σκοπό το σύστημα να γίνεται συνεχώς πιο λειτουργικό και αποδοτικό.

Η σχεδίαση λοιπόν του συστήματος πραγματοποιήθηκε βαδίζοντας στα παρακάτω βήματα:

- ⇒ Καθορισμός των στόχων του συστήματος με τη μορφή αναλυτικών λειτουργιών, ώστε να ανταποκρίνονται στο βασικό σκοπό του προβλήματος.

- ⇒ Προσδιορισμός των πρακτόρων και των ρόλων τους σύμφωνα με το προηγούμενο βήμα.
- ⇒ Αρχιτεκτονική του συστήματος και των πρακτόρων με διαγράμματα που περιγράφουν τη δομή και την επικοινωνία του συστήματος
- ⇒ Τάξεις πρακτόρων. Ουσιαστικά κατασκευάζονται οι πράκτορες, προσδιορίζοντας τις μεθόδους που πρέπει αυτοί να υποστηρίζουν για να εκπληρώνουν τις λειτουργίες που τους έχουν ανατεθεί από το δεύτερο βήμα. Σε αυτό το βήμα σχεδιάζεται η βάση γνώσης του κάθε πράκτορα.

Η μεθοδολογία σχεδίασης έγινε με προσανατολισμό την ανάπτυξη του συστήματος. Αυτό σημαίνει πως προτιμήθηκε μια αντικειμενοστραφής προσέγγιση για τη σχεδίαση ώστε η κωδικοποίηση να πραγματοποιηθεί πιο εύκολα με κάποια από τις αντικειμενοστραφείς γλώσσες προγραμματισμού.

Σε προηγούμενη παράγραφο, αναφέραμε ότι κάθε πράκτορας του συστήματος εκτελεί τους συλλογισμούς και τις εργασίες που απαιτούνται ώστε να φέρει εις πέρας το έργο για το οποίο έχει κατασκευαστεί, χρησιμοποιώντας μια ιδιωτική βάση γνώσης, δανειζόμενος τεχνολογίες από την τεχνολογία των Έμπειρων Συστημάτων. Αυτό επιτεύχθηκε με την μεταγλώτιση και αξιολόγηση των κανόνων παραγωγής της βάσης γνώσης του κάθε πράκτορα κατόπιν της ενεργοποίησής του. Με αυτό το τρόπο επιτεύχθηκε και η συντηρησιμότητα των πρακτόρων εφόσον η λογική τους δεν είναι ενσωματωμένη στον κώδικα του προγράμματος.

Οι οντολογίες που χρησιμοποιούν οι βάσεις γνώσης των πρακτόρων είναι επίσης δυναμικές και αποθηκεύονται με τη σειρά τους στην βάση δεδομένων. Με αυτό τον τρόπο είναι δυνατή η περαιτέρω ανάπτυξη της λογικής των πρακτόρων αλλά και η χρήση των σε εντελώς διαφορετικά προβλήματα από αυτό που μελετάμε στην παρούσα εργασία.

Αυτή η τεχνική πρωτοεμφανίστηκε στην κονσόλα Πολυπρακτορικών Συστημάτων Jade η οποία εισήγαγε την έννοια του «Έμπειρου Πράκτορα» χρησιμοποιώντας πράκτορες των οποίων ο συλλογισμός ήταν αποτέλεσμα του διερμηνέα ενός Έμπειρου Συστήματος γραμμένου σε JESS (Java Expert System Shell).

Η γλώσσα προγραμματισμού που επιλέχθηκε είναι η Visual Basic .NET σε συνδυασμό με το αντικείμενο επικοινωνίας με βάσεις δεδομένων ADO .NET (ActiveX Data Objects) της Microsoft. Ο λόγος που προτιμήθηκε η εν λόγω γλώσσα είναι:

- ⇒ Οι δυνατότητες που παρέχει για την δυναμική μεταγλώτιση και εκσφαλμάτωση αντικειμένων προγράμματος
- ⇒ Η παρουσία δομών δεδομένων εφάμιλλων αυτών που χρησιμοποιούν γνώστες γλώσσες για την ανάπτυξη Έμπειρων Συστημάτων. Συγκεκριμένα αναφέρουμε την δομή δεδομένων hashtable, η οποία χρησιμοποιήθηκε ως υποκατάστατο των predicates της Prolog, για την έκφραση των οντολογιών του συστήματος και η οποία μπορεί:
  - αφενός μεν να εκφράσει δεδομένα τύπου κειμένου όπως και τα predicates επί των οποίων μπορεί να εφαρμόσει τεχνικές pattern matching και
  - αφετέρου να εκφράσει οποιοδήποτε άλλο τύπο δεδομένων (αριθμοί, λίστες, σύνθετα αντικείμενα)

Η αξιολόγηση του τύπου δεδομένων του κάθε στοιχείου γίνεται δυναμικά και εξαρτάται από τον εκάστοτε τελεστή που χρησιμοποιούμε (late binding).

## 4.2 Καθορισμός των στόχων

Προκύπτουν λοιπόν οι παρακάτω ανάγκες :

- ⇒ Το σύστημα πρέπει να συλλέγει τις προδιαγραφές του εντύπου από τον χρήστη. Τα δεδομένα αυτά αφορούν τον τύπο του εντύπου που θέλουμε να παράγουμε, τις προαιρετικές εργασίες που συνοδεύουν το συγκεκριμένο έντυπο, τις προδιαγραφές εκτύπωσης (χρώματα εκτύπωσης, ποιότητα χαρτιού, διαστάσεις εντύπου, πλήθος αντίτυπων) κ.ο.κ.
- ⇒ Το σύστημα πρέπει να είναι είναι συντηρήσιμο και προσαρμόσιμο σε απρόβλεπτες καταστάσεις όπως η βλάβη μιας μηχανής ή την διαπίστωση κάποιας νέας παραμέτρου η οποία επηρεάζει την παραγωγική διαδικασία.
- ⇒ Το σύστημα πρέπει να μελετάει όλες τις πιθανές γραμμές παραγωγής για κάποιο προϊόν ώστε να είναι δυνατός ο έλεγχος του συλλογισμού του συστήματος από τον αποφασίζοντα, τον μηχανικό γνώσης και τον προγραμματιστή
- ⇒ Το σύστημα πρέπει να παρέχει ένα γραφικό περιβάλλον συντήρησης και ανανέωσης των οντολογιών και των κανόνων των πρακτόρων .
- ⇒ Πρέπει να υπάρχει ένα γραφικό περιβάλλον όπου θα παρουσιάζονται τα αποτελέσματα της εκτέλεσης του συστήματος. Σε αυτό θα παρουσιάζονται σε μορφή δέντρου οι πιθανές γραμμές παραγωγής του εντύπου και θα τονίζονται αυτές που παρουσιάζουν το μικρότερο κόστος. Σε κάθε κόμβο του δέντρου θα δίδονται οι μεταβλητές της διαδικασίας (μηχανή, διαδικασία, τρέχον κόστος, κανόνες που αξιολογήθηκαν κ.ο.κ.). Η απευθείας υπόδειξη μόνο της βέλτιστης λύσης δεν ενδείκνυται καθώς είναι πολύ πιθανό, κάποια μηχανή που χρησιμοποιείται σε αυτήν τη λύση να είναι είτε χαλασμένη, είτε δεσμευμένη. Ένας άλλος λόγος που αποφεύγεται η απευθείας υπόδειξη μόνο της βέλτιστης λύσης, έχει να κάνει με τον εντοπισμό πιθανών αστοχιών του συστήματος κυρίως στα πρώτα στάδια λειτουργίας του όπου θα επιλεχτούν οι οδηγοί κόστους και θα γίνεται αποσφαλμάτωση τόσο του κώδικα του προγράμματος όσο και της λογικής του.
- ⇒ Τέλος πρέπει τα συστατικά του συστήματος να είναι διακριτά και αυτόνομα. Ο διαχωρισμός αυτός πρέπει να υπακούει στις αρχές τόσο του αντικειμενοστραφούς προγραμματισμού όσο και των κατανεμημένων συστημάτων. Η χρήση αντικειμενοστραφούς προγραμματισμού παρέχει συντηρησιμότητα και επεκτασιμότητα στο σύστημα ενώ η κατανεμημένη αρχιτεκτονική μας παρέχει ευελιξία τόσο στην επιλογή του Συστήματος Διαχείρισης Σχεσιακών Βάσεων Δεδομένων (RDBMS – Relational Database Management System) όσο και στην επιλογή του περιβάλλοντος διεπαφής το οποίο μπορεί να είναι είτε Desktop Client, είτε Web Client. Στην δεύτερη περίπτωση είναι απαραίτητη η παρεμβολή μιας εφαρμογής Απομακρυσμένης Κλήσης Διαδικασίας (Remote Procedure Call) μέσω SOAP – XML μηνυμάτων.

### 4.3 Προσδιορισμός των πρακτόρων και των ρόλων τους

Έχοντας καθορίσει του στόχους του συστήματος, μπορούμε να αποδώσουμε σε πράκτορες τις ευθύνες για την πραγματοποίηση τους. Η ιδιαιτερότητα του συστήματος, ο δυναμικός δηλαδή αριθμός των πρακτόρων εργασιών, δε μας εμποδίζει να περιγράψουμε όλους τους πράκτορες του συστήματος, αφού οι πράκτορες εργασιών θα ενσωματώνουν όλοι τους ίδιους μηχανισμούς και λειτουργίες.

Βασικό συστατικό λοιπόν του συστήματος αποτελεί ο **πράκτορας διεπαφής τελικού χρήστη (end user interface agent)**. Ο πράκτορας διεπαφής τελικού χρήστη, πρέπει αφενός να παρέχει ένα εργονομικό και εύχρηστο περιβάλλον επικοινωνίας με τον χρήστη για την εισαγωγή των προδιαγραφών των προϊόντων και αφετέρου να παρουσιάζει τα αποτελέσματα της διαδικασίας επίλυσης στο χρήστη.

Ο **πράκτορας ανάθεσης εργασιών** συντονίζει την διαδικασία της προσομοίωσης. Αποτελεί τον μοναδικό πράκτορα ο οποίος είναι υλοποιημένος στον κώδικα του προγράμματος. Για την εφαρμογή της τεχνικής που παρουσιάζεται σε αυτή την εργασία σε άλλα προβλήματα, αυτός ο πράκτορας πρέπει είτε να επαναπρογραμματιστεί από την αρχή, είτε να χρησιμοποιηθεί ως βάση (base class) για την δημιουργία ενός νέου παραμετροποιημένου πράκτορα ανάθεσης εργασιών. Εκτελεί τις ακόλουθες ενέργειες.

- ⇒ Επικοινωνία με τον πράκτορα διεπαφής τελικού χρήστη.
- ⇒ Έναρξη και λήξη διαδικασίας προσομοίωσης.
- ⇒ Δημιουργία αρχείου προσομοίωσης. (log file σε μορφή XML)
- ⇒ Παίξει ρόλο πομπού και δέκτη στην μεταφορά μηνυμάτων (proxy).

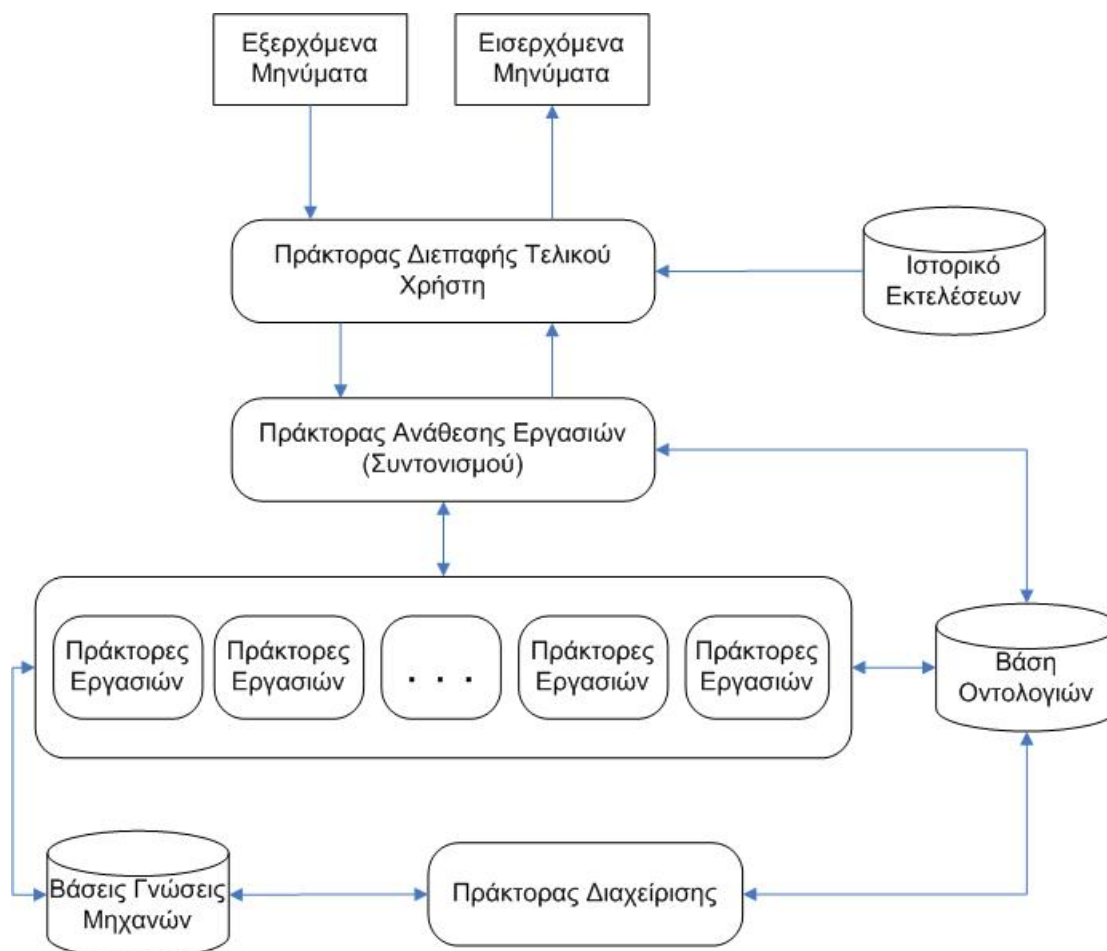
Οι **πράκτορες εργασίας** αντιστοιχούν στις μηχανές του εργοστασίου. Ο καθένας από αυτούς φέρει τη δικιά του βάση γνώσης την οποία ανακτά από μια βάση δεδομένων.

Οι πράκτορες εργασίας από την στιγμή που θα ενεργοποιηθούν αρχικοποιούν την μνήμη εργασίας τους με τιμές που ανακτούν από τον πράκτορα ανάθεσης εργασιών. Δύνανται να αρνηθούν μια εργασία που θα τους ανατεθεί ενώ με το πέρας της εργασίας τους (εφόσον την έχουν αποδεχθεί) επιστρέφουν τα αποτελέσματα της δουλειάς τους (μετάλλαξη της τρέχουσας κατάστασης προϊόντος) στον πράκτορα ανάθεσης εργασιών. Επίσης επιστρέφουν μια λίστα επόμενων μηχανών. Σε περίπτωση που η λίστα επόμενων μηχανών είναι κενή σημαίνει ότι φτάσαμε σε ένα φύλλο του δέντρου γραμμών παραγωγής. Αν η κατάσταση που επέστρεψε ο πράκτορας εργασίας διαφέρει από το επιθυμητό προϊόν, έχουμε αποτυχημένη γραμμή παραγωγής.

Τέλος συναντάμε τον **πράκτορα διαχείρισης**. Φέρει και αυτός γραφικό περιβάλλον. Προορισμός του πράκτορα αυτού είναι η ενημέρωση, συντήρηση και εκσφαλμάτωση των οντολογιών και των κανόνων των βάσεων γνώσεων των πρακτόρων εργασίας. Αποτελείται από τρεις οθόνες προγράμματος. Η πρώτη διαχειρίζεται τις διαδικασίες της παραγωγής, η δεύτερη τις μηχανές του εργοστασίου τις οποίες συσχετίζει με διαδικασίες και τέλος η τρίτη παρέχει πολλαπλά παράθυρα όπου παρουσιάζονται οι οντολογίες του συστήματος και έναν επεξεργαστή κειμένου για την επεξεργασία των κανόνων των μηχανών. Ο επεξεργαστής κειμένου εκτός από την μορφοποίηση του κειμένου, παίζει και το ρόλο εκσφαλματωτή, καθώς παρέχει μηχανισμούς ελέγχου της εγκυρότητας του εισερχόμενου κειμένου ενώ αποτρέπει την αποθήκευση μιας βάσης γνώσης εφόσον περιέχει συντακτικά λάθη.

#### 4.4 Αρχιτεκτονική του συστήματος και των πρακτόρων

Στο διάγραμμα παρακάτω εμφανίζεται η αρχιτεκτονική του συστήματος.



Εικόνα 8: Αρχιτεκτονική του Συστήματος

Όπως παρατηρούμε στο παραπάνω διάγραμμα δεν υπάρχει επικοινωνία μεταξύ των πρακτόρων εργασίας. Η ενημέρωση τους ωστόσο επιτυγχάνεται μέσω του πράκτορα ανάθεσης εργασιών ο οποίος φέρει τον πολλαπλό ρόλο, να αναθέτει τις εργασίες στους πράκτορες εργασιών και να λειτουργεί ως ενδιάμεσος σταθμός για την μεταφορά μηνυμάτων μεταξύ των πρακτόρων εργασιών και μεταξύ του πράκτορα διεπαφής τελικού χρήστη και των πρακτόρων εργασίας.

Σημειώνουμε σε αυτό το σημείο ότι ο πράκτορας διαχείρισης δεν παίρνει μέρος στην διαδικασία προσομοίωσης. Ο ρόλος του είναι υποστηρικτικός της συντήρησης και ενημέρωσης του συστήματος, αφού είναι υπεύθυνος για την συντήρηση και εκσφαλμάτωση των δεδομένων του συστήματος.

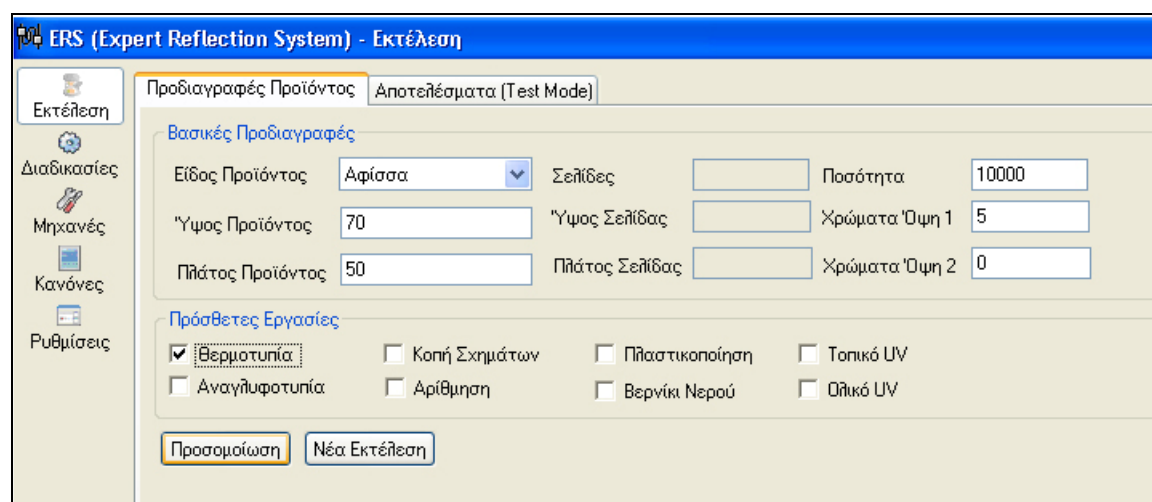
Στην επόμενη παράγραφο θα εξετάσουμε την ακριβή λειτουργία και δομή, καθενός από τους πράκτορες που αποτελούν το σύστημα.



## 4.5 Τάξεις πρακτόρων

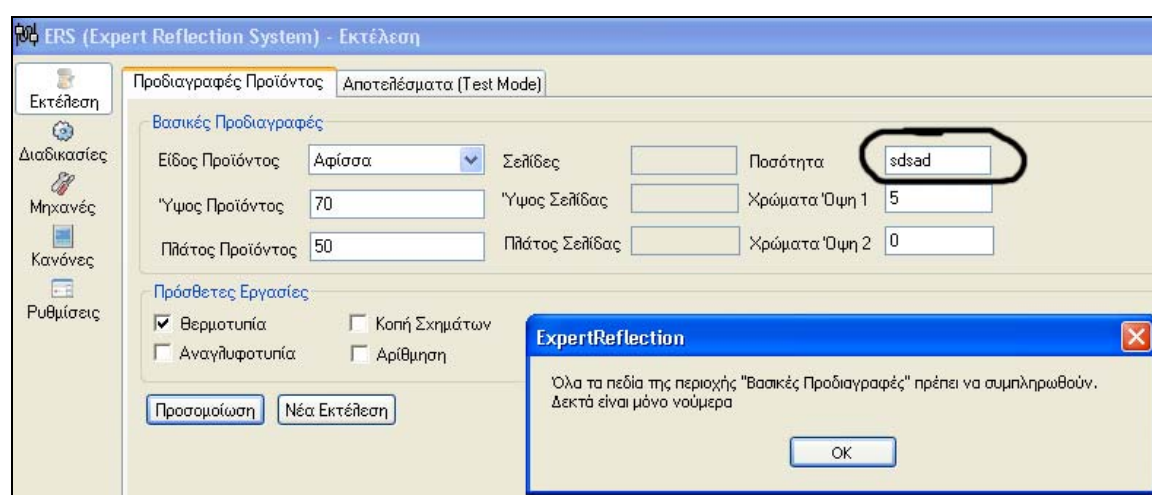
### 4.5.1 Ο πράκτορας διεπαφής τελικού χρήστη

Ο πράκτορας διεπαφής τελικού χρήστη αποτελεί το μέσο επικοινωνίας των τελικών χρηστών με το σύστημα. Στην οθόνη προγράμματος που εργάζεται ο χρήστης, εμφανίζονται πεδία για τον καθορισμό των προδιαγραφών του προϊόντος του οποίου θέλουμε να προσομοιώσουμε την παραγωγική διαδικασία και κουμπιά εντολών για την έναρξη της προσομοίωσης ή την εισαγωγή ενός νέου προϊόντος.



Εικόνα 9: Πράκτορας διεπαφής τελικού χρήστη. Οθόνη προσδιορισμού των προδιαγραφών του προϊόντος

Ωστόσο, συχνό φαινόμενο αποτελεί η λανθασμένη εισαγωγή δεδομένων σε μια εφαρμογή. Για αυτό το λόγο κατά την εντολή έναρξης της προσομοίωσης, όλα τα πεδία εισαγωγής δεδομένων ελέγχονται για την εγκυρότητά τους.



Εικόνα 10: Πράκτορας διεπαφής τελικού χρήστη. Έλεγχος εγκυρότητας δεδομένων εισόδου.

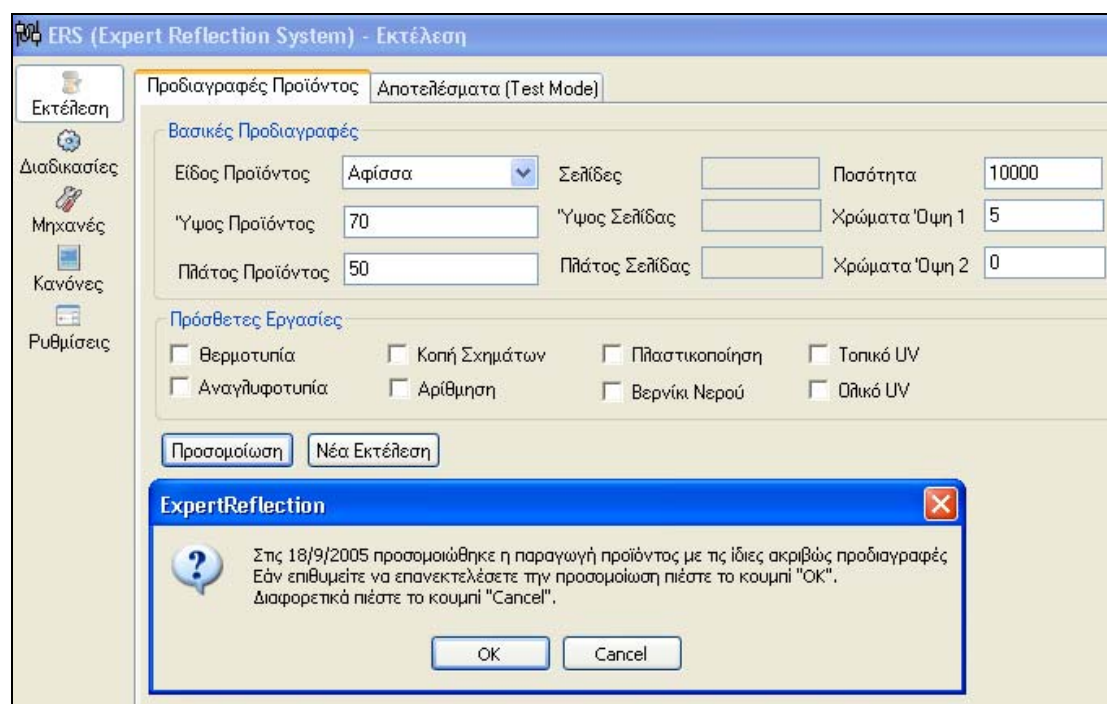
Η εκτέλεση της προσομοίωσης, ανάλογα και με το πλήθος των πρόσθετων εργασιών που θα φέρει το εκάστοτε προϊόν, αποτελεί μια αρκετά απαιτητική, σε χρόνο και πόρους συστήματος, διαδικασία. Για το λόγο αυτό η κάθε προσομοίωση που

εκτελείται αποθηκεύεται σε μορφή XML σε ένα αρχείο στον σκληρό δίσκο, ενώ ταυτόχρονα καταχωρείται στην βάση δεδομένων μια εγγραφή με το όνομα του XML αρχείου, τις προδιαγραφές του προϊόντος (οι οποίες αποτελούν και το κλειδί ανάκλησης του αρχείου) και την ημερομηνία δημιουργίας του αρχείου.

Κάθε φορά που ο χρήστης ξεκινάει μια προσομοίωση, ο πράκτορας διεπαφής τελικού χρήστη διασταυρώνει τις προδιαγραφές του προϊόντος προς προσομοίωση με τις εγγραφές που υπάρχουν στην βάση δεδομένων. Εφόσον βρεθεί κάποια εγγραφή που φέρει, πανομοιότυπες προδιαγραφές (συμπεριλαμβανόμενης και της ποσότητας των αντιτύπων) ένα μήνυμα εμφανίζεται στον χρήστη που τον ενημερώνει ανάλογα. Στη συνέχεια ο χρήστης έχει την δυνατότητα είτε να ανακτήσει τα αποτελέσματα της παλαιότερης προσομοίωσης, είτε να επανεκτελέσει την διαδικασία.

Κάθε φορά που ανανεώνεται η βάση γνώσης κάποιας μηχανής ή προστίθεται μια νέα μηχανή στο σύστημα, τα αποτελέσματα προσομοιώσεων που επηρεάζονται από αυτή την αλλαγή χαρακτηρίζονται ως ανενεργά και παύουν να προτείνονται στους χρήστες. Την ευθύνη για την απενεργοποίηση παλαιότερων προσομοιώσεων, αναλαμβάνουν triggers στο σύστημα διαχείρισης της βάσης δεδομένων που χρησιμοποιούμε (Microsoft SQL Server 2000).

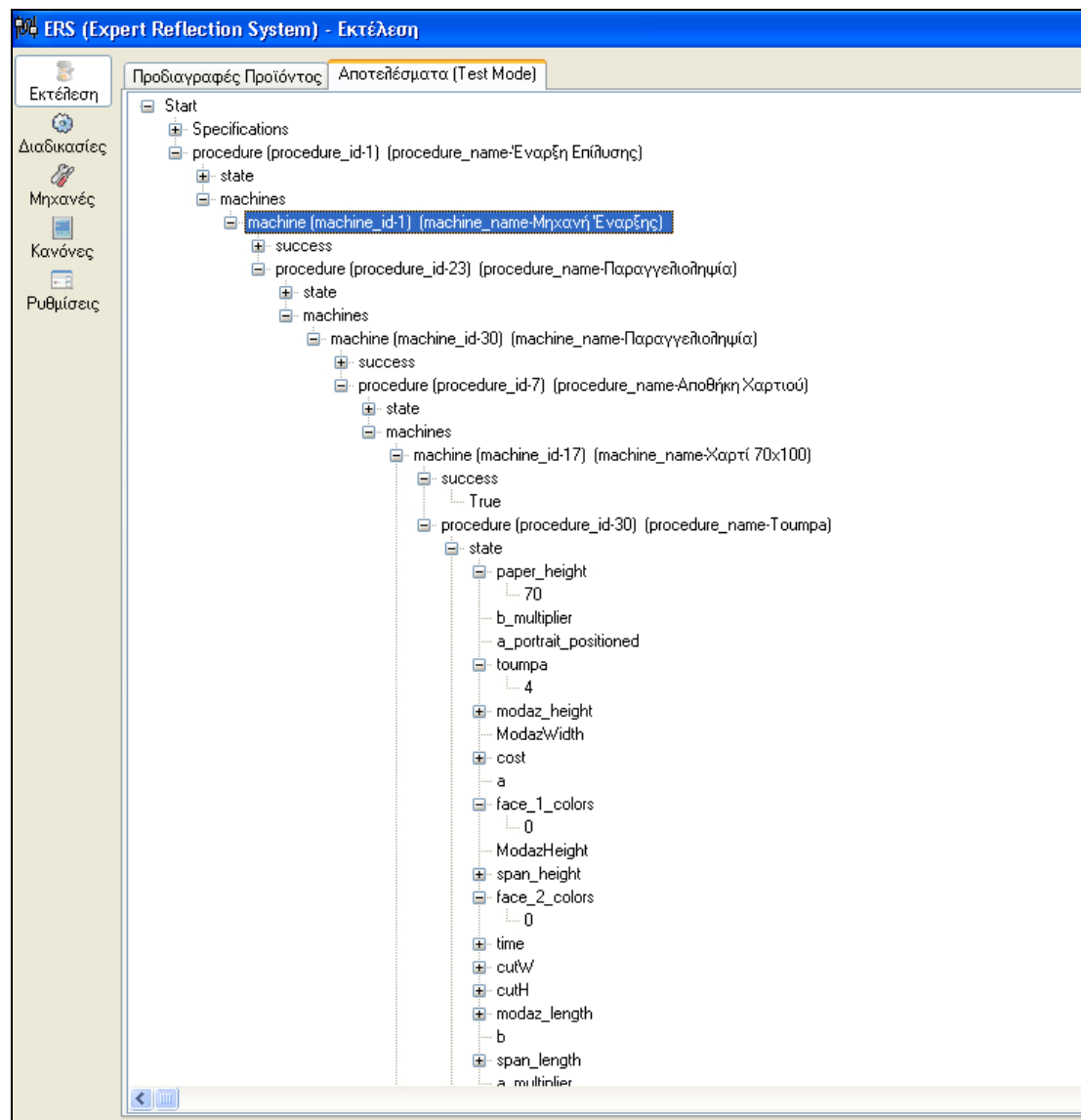
Με αυτό τον τρόπο, επιτυγχάνουμε αφενός μεν σημαντική οικονομία χρόνου και πόρων και αφετέρου δημιουργούμε μια τράπεζα δεδομένων, χρήσιμη τόσο για λόγους εκσφαλμάτωσης των υπάρχοντων κανόνων και δημιουργίας νέων όσο και για την εξαγωγή συμπερασμάτων και την αναθεώρηση εδραιωμένων διαδικασιών στη λειτουργία του εργοστασίου.



**Εικόνα 11: Πράκτορας διεπαφής τελικού χρήστη. Ανάκτηση αποτελεσμάτων προσομοίωσης ίδιων προϊόντων για την αποφυγή πολλαπλών εκτελέσεων προσομοιώσεων των οποίων διαθέτουμε ήδη τα αποτελέσματα.**

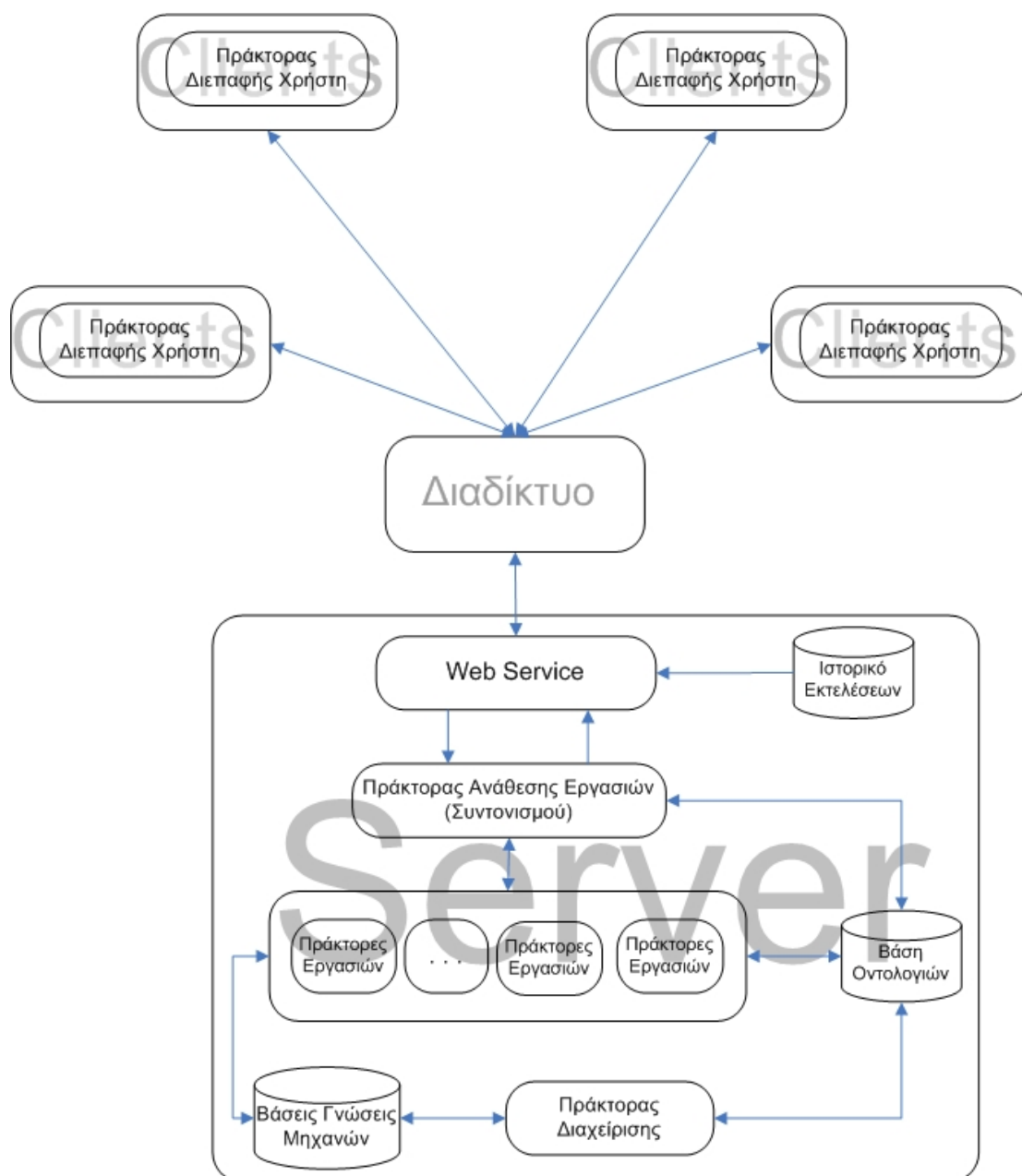
Αφού τα δεδομένα, που εισάγει ο χρήστης, περάσουν από τους κατάλληλους ελέγχους, ξεκινάει η προσομοίωση η οποία, όπως αναφέρθηκε παραπάνω, δημιουργεί ένα XML αρχείο που περιέχει όλες τις πιθανές γραμμές παραγωγής καθώς και όλες

τις ενδιαμέσες καταστάσεις από τις οποίες περνάει το προϊόν. Για την παρουσίαση των αποτελεσμάτων, αρκεί να επεξεργαστούμε αυτό το αρχείο και να εξάγουμε τις πληροφορίες που περιέχει με έναν τρόπο κατανοητό στον εκάστοτε χρήστη. Στο σχήμα παρακάτω παρουσιάζεται η απεικόνιση των αποτελεσμάτων, που χρησιμοποιεί ο γράφοντας για τον έλεγχο και εκσφαλμάτωση του συστήματος.



**Εικόνα 12: Πράκτορας διεπαφής τελικού χρήστη. Παρουσίαση αποτελεσμάτων. Στην συγκεκριμένη εικόνα παρουσιάζεται ένα δέντρο του οποίου κάθε κόμβος, αντιστοιχεί σε αντίστοιχο στοιχείο (element) του αρχείου αποτελεσμάτων.**

Η εξαγωγή των αποτελεσμάτων σε μορφή XML, παρέχει ακόμα ένα σημαντικό χαρακτηριστικό στον πράκτορα διεπαφής χρήστη, που δεν είναι άλλο από την δυνατότητα του να έχει την μορφή διαδικτυακής εφαρμογής. Συγκεκριμένα είναι δυνατόν ο πράκτορας διεπαφής τελικού χρήστη να αποτελεί μια συνηθισμένη ιστοσελίδα υποστηριζόμενη από μια διαδικτυακή εφαρμογή η οποία θα παίζει το ρόλο του ενδιάμεσου ανάμεσα στην ιστοσελίδα και το εξυπηρετητή που φιλοξενεί τους υπόλοιπους πράκτορες.



**Εικόνα 13:** Γεωγραφική αποδέσμευση του πράκτορα διεπαφής τελικού χρήστη. Οι προδιαγραφές του προϊόντος αποστέλνονται μέσω κοινών φορμών, εκτελείται η προσομοίωση στον εξυπηρετητή και επιστρέφεται το όνομα του XML αρχείου αποτελεσμάτων.

#### 4.5.2 Ο πράκτορας ανάθεσης εργασιών

Ο πράκτορας ανάθεσης εργασιών αποτελεί τον πυρήνα της εφαρμογής καθώς διαχειρίζεται την μεταφορά μηνυμάτων και αναθέτει εργασίες στους πράκτορες εργασιών. Είναι ο περισσότερο απλός στην λειτουργία πράκτορας αφού δεν παρουσιάζει κάποια νοημοσύνη.

Ο πράκτορας ανάθεσης εργασιών αφού λάβει την εντολή έναρξης της προσομοίωσης και τις προδιαγραφές του προϊόντος, από τον πράκτορα διεπαφής τελικού χρήστη, δημιουργεί μια κενή (null) αρχική κατάσταση και ενεργοποιεί έναν δεδομένο αρχικό πράκτορα εργασίας. Από αυτό το σημείο και έπειτα η εργασία που εκπληρώνει είναι να δέχεται μηνύματα από τους ενεργοποιημένους πράκτορες, να τα επεξεργάζεται και επαναληπτικά να ενεργοποιεί καινούργιους πράκτορες.

Ο πράκτορας ανάθεσης εργασιών, δεν επιλέγει με κάποιο νοήμων τρόπο ποιούς πράκτορες να ενεργοποιήσει. Η απόφαση για το ποιούς πράκτορες πρέπει να ενεργοποιηθούν στην επόμενη επανάληψη, λαμβάνεται από τους πράκτορες εργασίας και μεταβιβάζεται στον πράκτορα ανάθεσης εργασιών, ο οποίος απλά εκτελεί την εντολή.

Με την ολοκλήρωση της προσομοίωσης το XML αρχείο που παράγεται έχει την ακόλουθη μορφή.

```
<?xml version="1.0" standalone="no"?>
<!--Tsoukakis Dimitrios - Politehneio Kritis-->
<Root>
  <Start>
    <Specifications>
      <paraggeliolipsia />
      <scanning />
      <fwtoorafisi />
      <quantity>10000</quantity>
      <face 1 colors>4</face 1 colors>
      <verniki_nerou>False</verniki_nerou>
      <thermotipia_kordelas />
      <apostoli />
      <prod height>70</prod height>
      <face 2 colors>0</face 2 colors>
      <thermal>False</thermal>
      <subitem_length />
      <prod length>50</prod length>
      <numbering>False</numbering>
      <diplwtiki parathirou />
      <subitem height />
      <sirtarokollitiki />
      <span_count />
      <global uv>False</global uv>
      <pikmansia />
      <sinthetiki 16selidwn />
      <product type id>9</product type id>
      <anaglyph>False</anaglyph>
      <pastikopoiisi>False</pastikopoiisi>
      <kaloupi kopis />
      <siskevasia />
      <kaloupi thermotipias />
      <kopi sximatwn>False</kopi sximatwn>
      <local_uv>False</local_uv>
      <kaloupi_anaglifotipias />
    </Specifications>
    <procedure procedure id="1" procedure name="Έναρξη Επίλυσης">
      <state>
        <paper_height>0</paper_height>
        <b_multiplier />
        <a_multiplier />
        <a portrait positioned />
        <modaz height>0</modaz height>
        <ModazWidth />
        <cutW />
      </state>
    </procedure>
  </Start>
</Root>
```

```

        <cost>0</cost>
        <products per sheet>0</products per sheet>
        <face 1 colors>0</face 1 colors>
        <ModazHeight />
        <span height>0</span height>
        <face_2_colors>0</face_2_colors>
        <time>0</time>
        <modaz length>0</modaz length>
        <cutH />
        <b />
        <span_length>0</span_length>
        <toumpa>4</toumpa>
        <a />
        <paper length>0</paper length>
        <ScrapArea />
    </state>
    <machines>
        <machine machine_id="1" machine_name="Μηχανή Έναρξης">
            <success>True</success>
            <procedure procedure_id="23" procedure_name="Παραγγελιοληψία">
                <state>
                    <paper height>0</paper height>
                    <b_multiplier />
                    <a_portrait_positioned />
                    <toumpa>4</toumpa>
                    <modaz height>0</modaz height>
                    <ModazWidth />
                    <cost>0</cost>
                    <a />
                    <face_1_colors>0</face_1_colors>
                    <ModazHeight />
                    <span height>0</span height>
                    <face 2 colors>0</face 2 colors>
                    <time>0</time>
                    <cutW>0</cutW>
                    <cutH>0</cutH>
                    <modaz length>0</modaz length>
                    <b />
                    <span length>0</span length>
                    <a_multiplier />
                    <products_per_sheet>0</products_per_sheet>
                    <paper length>0</paper length>
                    <ScrapArea />
                </state>
                . . .
                . . .
                . . .
                . . .
                . . .
            </procedure>
        </machine>
    </machines>

```

#### 4.5.3 Οι πράκτορες εργασιών

Οι πράκτορες εργασιών όπως προδίδει και το όνομα τους εκτελούν μία ή περισσότερες εργασίες για τις οποίες έχουν κατασκευαστεί.

Συγκεκριμένα από την στιγμή που θα ενεργοποιηθούν εκτελούν τα παρακάτω βήματα:

1. Ελέγχουν την εργασία μνήμης για την ύπαρξη των απαραίτητων πληροφοριών για την εκτέλεση της εργασίας τους. Εφόσον η μνήμη εργασίας είναι κενή, λαμβάνουν όλες τις απαραίτητες πληροφορίες από τον πράκτορα ανάθεσης εργασιών. Θεωρούμε ότι ο πράκτορας ανάθεσης εργασιών είναι πάντα σε θέση να ικανοποιεί τις απαιτήσεις των πρακτόρων εργασιών.
2. Κάνοντας χρήση της βάσης γνώσης τους και ορθής συλλογιστικής εξετάζουν κατά πόσο είναι ικανοί να εκτελέσουν την εργασία που τους έχει ανατεθεί και ενημερώνουν τον πράκτορα ανάθεσης εργασιών ο οποίος με τη σειρά του

ενημερώνει το XML αρχείο αποθήκευσης των αποτελεσμάτων. Σε περίπτωση που κρίνουν ότι δεν μπορούν να εκτελέσουν την εργασία, καταστρέφονται.

3. Στο βήμα αυτό και εφόσον δεν έχουν καταστραφεί στο προηγούμενο βήμα, εκτελούν την εκάστοτε εργασία. Το αποτέλεσμα της εργασίας του κάθε πράκτορα δεν είναι άλλο από την μετάλλαξη της κατάστασης του παραγώμενου προϊόντος. Για παράδειγμα, ένας πράκτορας που αντιστοιχεί σε μια εκτυπωτική μηχανή αυξάνει το πλήθος των εκτυπωμένων χρωμάτων πάνω στο προϊόν. Με το πέρας της βήματος, ενημερώνουν τον πράκτορα ανάθεσης εργασιών ο οποίος με τη σειρά του ενημερώνει το XML αρχείο αποθήκευσης των αποτελεσμάτων.
4. Στο τέταρτο και τελευταίο στάδιο της λειτουργίας τους οι πράκτορες εργασιών επιλέγουν ποιό θα είναι το επόμενο στάδιο επεξεργασίας του προϊόντος. Με το πέρας της βήματος, ενημερώνουν τον πράκτορα ανάθεσης εργασιών ο οποίος με τη σειρά του ενημερώνει το XML αρχείο αποθήκευσης των αποτελεσμάτων.

Οι πράκτορες εργασιών είναι χωρισμένοι σε ομάδες ανάλογα με τον τύπο των εργασιών που μπορούν να εκτελέσουν για λόγους οικονομίας πόρων. Δηλαδή, κάθε φορά που ένας πράκτορας εργασίας επιλέγει την επόμενη εργασία που πρέπει να εκτελεστεί, ουσιαστικά επιλέγει μια ομάδα άλλων πρακτόρων εργασιών οι οποίοι είναι προορισμένοι για εργασίες αυτού του είδους.

Η μορφή των κανόνων που περιέχονται στην βάση γνώσης των πρακτόρων δίνεται παρακάτω, σε μορφή ψευδικώδικα. Το παράδειγμα προέρχεται από την βάση γνώσης μιας εκτυπωτικής μηχανής.

```

1: Αν [Προδιαγραφές].[Χρώματα στην Α Όψη] <= 2 και
    [Προδιαγραφές].[Χρώματα στην Β Όψη] <= 2 Τότε
    [Μηχανή Μη Συμβατή]
Τέλος Αν

2: Αν [Ύψος Τυπογραφικού] <= 100 και
    [Πλάτος Τυπογραφικού] <= 70 Τότε
    [Μηχανή Συμβατή]
Διαφορετικά
    [Μηχανή Μη Συμβατή]
Τέλος Αν

3: Αν [Προδιαγραφές].[Χρώματα στην Α Όψη] = 1 Τότε
    [Νέα Κατάσταση].[Εκτυπωμένα Χρώματα στην Α Όψη] = 1
    [Νέα Κατάσταση].[Κόστος] += [xxxxxx]
Διαφορετικά Αν [Προδιαγραφές].[Χρώματα στην Α Όψη] = 2 Τότε
    [Νέα Κατάσταση].[Εκτυπωμένα Χρώματα στην Α Όψη] = 2
    [Νέα Κατάσταση].[Κόστος] += [xxxxxx]
Διαφορετικά Αν [Προδιαγραφές].[Χρώματα στην Α Όψη] = 3 Τότε
    [Νέα Κατάσταση].[Εκτυπωμένα Χρώματα στην Α Όψη] = 3
    [Νέα Κατάσταση].[Κόστος] += [xxxxxx]
Διαφορετικά Αν [Προδιαγραφές].[Χρώματα στην Α Όψη] = 4 Τότε
    [Νέα Κατάσταση].[Εκτυπωμένα Χρώματα στην Α Όψη] = 4
    [Νέα Κατάσταση].[Κόστος] += [xxxxxx]
Τέλος Αν

4: Αν [Προδιαγραφές].[Χρώματα στην Β Όψη] = 1 Τότε
    [Νέα Κατάσταση].[Εκτυπωμένα Χρώματα στην Β Όψη] = 1
    [Νέα Κατάσταση].[Κόστος] += [xxxxxx]
Διαφορετικά Αν [Προδιαγραφές].[Χρώματα στην Β Όψη] = 2 Τότε
    [Νέα Κατάσταση].[Εκτυπωμένα Χρώματα στην Β Όψη] = 2

```



```
[Νέα Κατάσταση].[Κόστος] += [xxxxxx]
Διαφορετικά Αν [Προδιαγραφές].[Χρώματα στην Β Όψη] = 3 Τότε
    [Νέα Κατάσταση].[Εκτυπωμένα Χρώματα στην Β Όψη] = 3
    [Νέα Κατάσταση].[Κόστος] += [xxxxxx]
Διαφορετικά Αν [Προδιαγραφές].[Χρώματα στην Β Όψη] = 4 Τότε
    [Νέα Κατάσταση].[Εκτυπωμένα Χρώματα στην Β Όψη] = 4
    [Νέα Κατάσταση].[Κόστος] += [xxxxxx]
Τέλος Αν
```

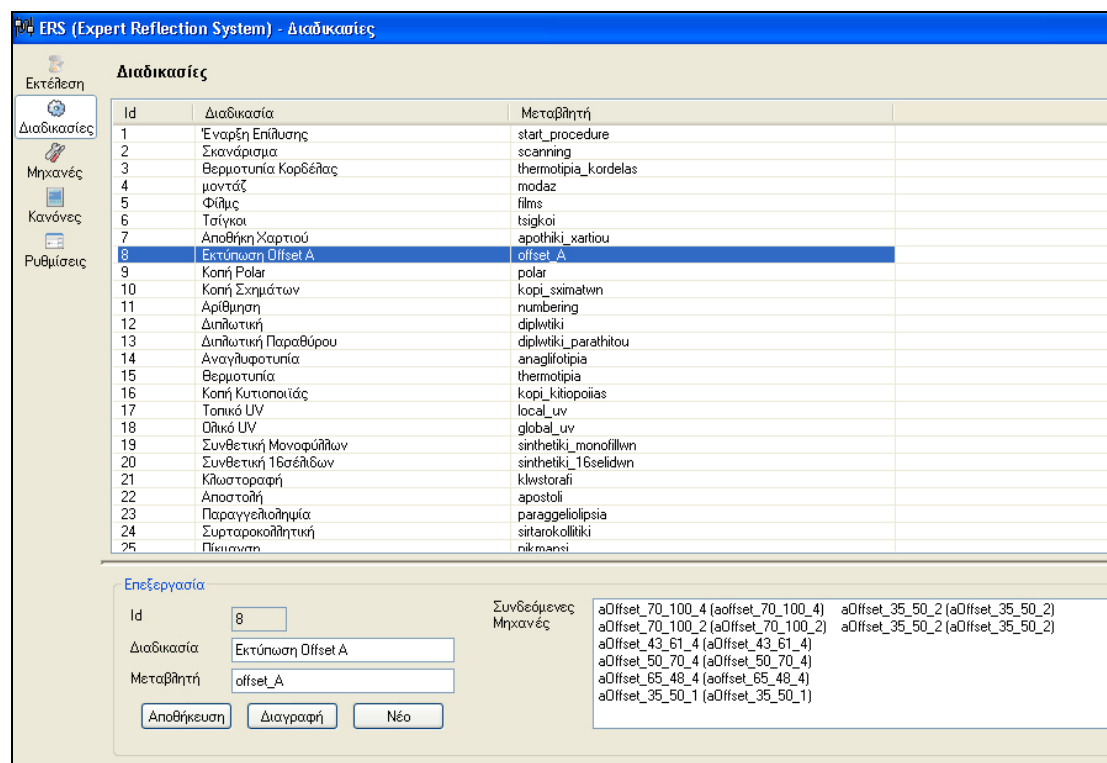
```
5: Αν [Προδιαγραφές].[Βερνίκι Νερού] Τότε
    [Επόμενη Διαδικασία] = [Βερνίκι Νερού]
Διαφορετικά Αν [Προδιαγραφές].[Αρίθμηση] Τότε
    [Επόμενη Διαδικασία] = [Αρίθμηση]
Διαφορετικά Αν [Προδιαγραφές].[Πλαστικοποίηση] Τότε
    [Επόμενη Διαδικασία] = [Πλαστικοποίηση]
Διαφορετικά Αν [Προδιαγραφές].[Ολικό UV] Τότε
    [Επόμενη Διαδικασία] = [Ολικό UV]
Διαφορετικά Αν [Προδιαγραφές].[Τοπικό UV] Τότε
    [Επόμενη Διαδικασία] = [Τοπικό UV]
Διαφορετικά Αν [Προδιαγραφές].[Θερμοτυπία] Τότε
    [Επόμενη Διαδικασία] = [Θερμοτυπία]
Διαφορετικά Αν [Προδιαγραφές].[Αναγλυφοτυπία] Τότε
    [Επόμενη Διαδικασία] = [Αναγλυφοτυπία]
Διαφορετικά Αν [Προδιαγραφές].[Δημιουργία Πτυχών] Τότε
    [Επόμενη Διαδικασία] = [Δίπλωμα]
Διαφορετικά
    [Επόμενη Διαδικασία] = [Συσκευασία]
Τέλος Αν
```



#### 4.5.4 Ο πράκτορας διαχείρισης

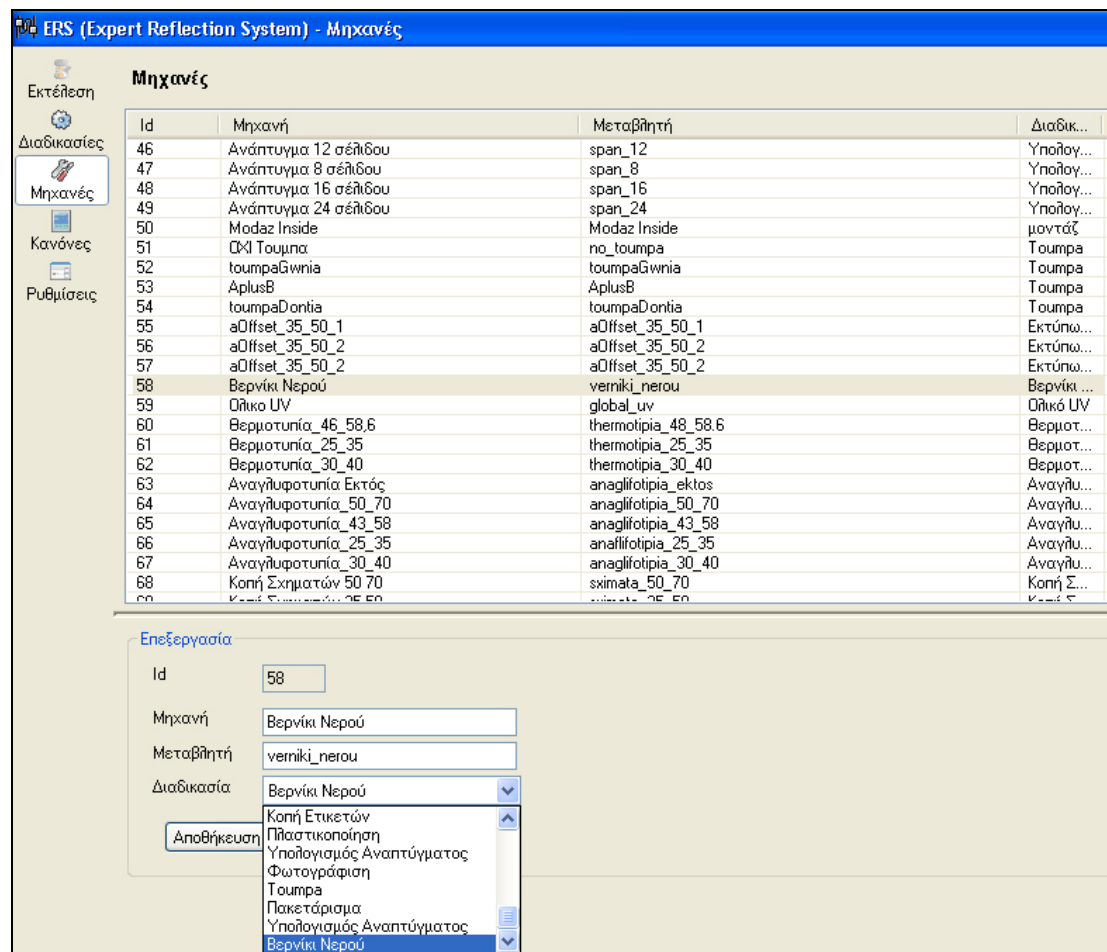
Ο πράκτορας διαχείρισης αναλαμβάνει την συντήρηση, ενημέρωση και εκφαλμάτωση των βάσεων γνώσεων των πρακτόρων και των οντολογιών που αυτοί υποστηρίζουν. Έχει υλοποιηθεί με την χρήση τριών οθονών, οι οποίες είναι διαθέσιμες μόνο σε χρήστες με δικαιώματα διαχειριστή.

⇒ Στην πρώτη οθόνη καθορίζονται οι ομάδες (διαδικασίες) στις οποίες υπάρχουν οι πράκτορες. Σε αυτήν την οθόνη ο χρήστης μπορεί να εισάγει, να επεξεργαστεί ή και να διαγράψει διαδικασίες.



Εικόνα 14: Οθονή επεξεργασίας διαδικασιών.

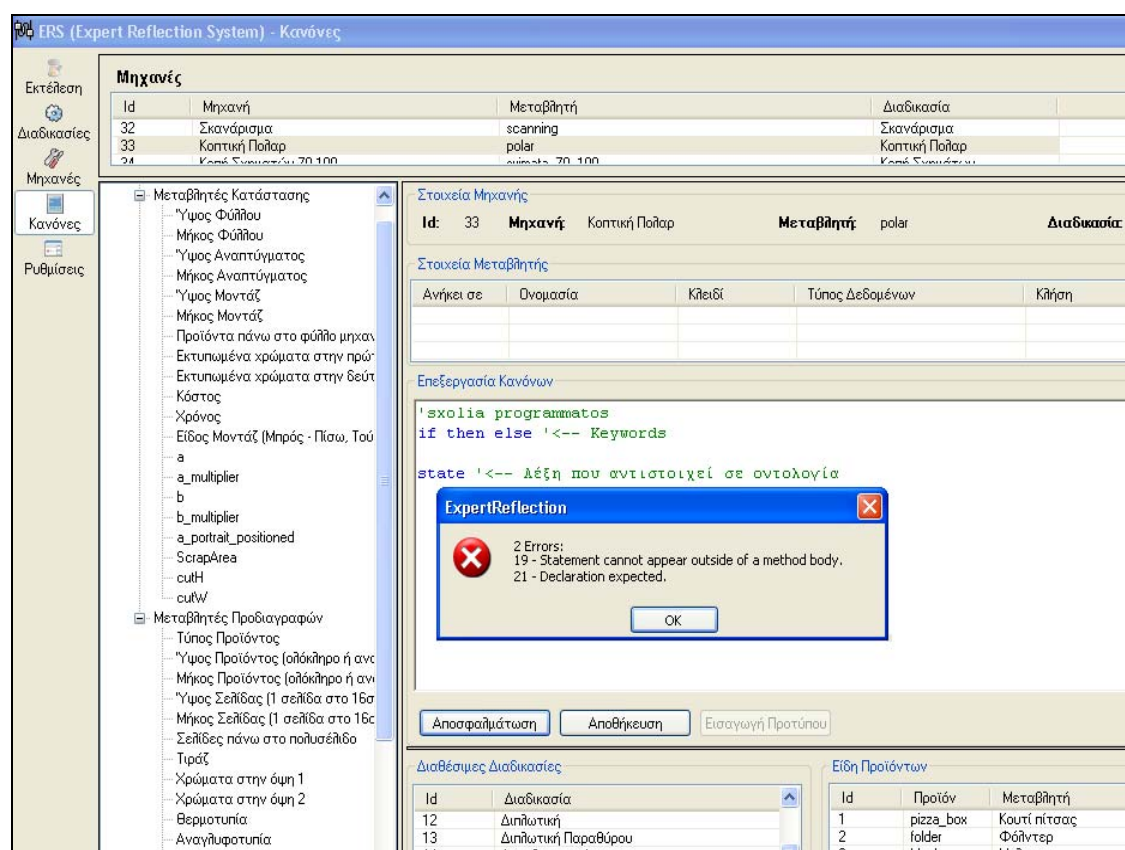
⇒ Στην δεύτερη οθόνη γίνεται διαχείριση των πρακτόρων (μηχανές). Σε αυτή την οθόνη ο χρήστης έχει την δυνατότητα να εισάγει, να επεξεργαστεί ή και να διαγράψει έναν πράκτορα. Στην επεξεργασία ενός πράκτορα συμπεριλαμβάνεται και η συσχέτιση του με μια διαδικασία (η ομάδα στην οποία ανήκει ο πράκτορας εργασίας).



Εικόνα 15: Οθόνη επεξεργασίας μηχανών

⇒ Τέλος η τρίτη οθόνη προορίζεται στην διαχείριση των βάσεων γνώσεων των πρακτόρων. Φέρει πολλαπλές λίστες στις οποίες φαίνονται οι οντολογίες με τις οποίες είναι δυνατόν να εργαστούμε στην δημιουργία των κανόνων. Για την εισαγωγή των κανόνων έχει δημιουργηθεί ένας επεξεργαστής κειμένου με τα ακόλουθα χαρακτηριστικά:

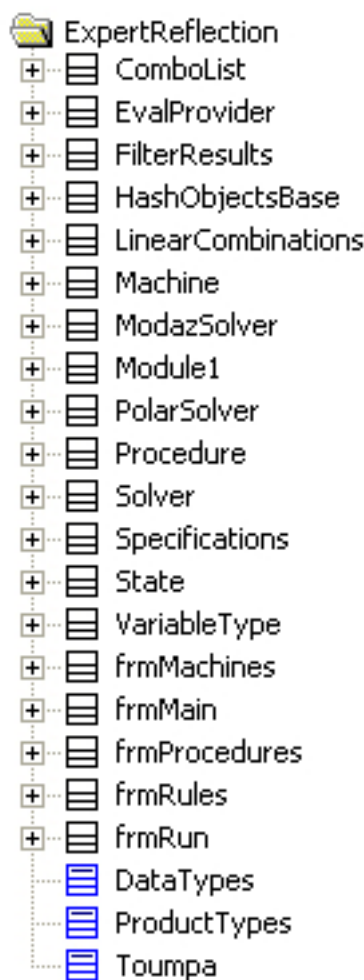
- Χρωματισμός των λέξεων που αποτελούν μέρος του λεξιλογίου της γλώσσας συγγραφής των κανόνων.
- Χρωματισμός των λέξεων που αποτελούν οντολογίες.
- Χρωματισμός των σχολίων της βάσης γνώσης
- Ένδειξη και υπόδειξη των συντακτικών λαθών της βάσης γνώσης.



Εικόνα 16: Οθόνη επεξεργασίας κανόνων

## 4.6 Τα αντικείμενα του προγράμματος

Στην παράγραφο αυτή μελετάμε την δομή των αντικειμένων της εφαρμογής που αναπτύχθηκε μέσω UML διαγραμμάτων που παρουσιάζουν την δομή και την συσχέτιση των αντικειμένων. Αρχικά θα παρουσιάσουμε το δέντρο αντικειμένων του προγράμματος και κατόπιν θα προχωρήσουμε στην παρουσίαση του κάθε αντικειμένου. Σημειώνουμε ότι με τον όρο αντικείμενα δεν εννοούμε στιγμιότυπα κλάσεων της εφαρμογής, αλλά τα συστατικά εκείνα από τα οποία αποτελείται η εφαρμογή. Παρακάτω φίνεται το δέντρο αντικειμένων της εφαρμογής.



Εικόνα 17: Δέντρο αντικειμένων της εφαρμογής

Κατόπιν θα μελετήσουμε τα σημαντικότερα από τα παραπάνω αντικείμενα.

## 1. Αντικείμενο ComboBox

ComboBox
-sName : String -iID : Long
+New() +New(in Name : String, in ID : Long) +Name() : String +ItemData() : Long +ToString() : String

Στο παραπάνω διάγραμμα παρουσιάζεται η κλάση ComboBox. Χρησιμοποιείται στην κατασκευή του γραφικού περιβάλλοντος, υποβοηθητικά σε controls τύπου Combo Box. Παρέχει την ικανότητα δημιουργίας λιστών όπου κάθε στοιχείο φέρει μια κρυφή στον χρήστη τιμή. Συνήθως αυτή η τιμή αφορά κάποιο κλειδί για την ανάκτηση πληροφοριών από την βάση δεδομένων.

## 2. Αντικείμενο EvalProvider

EvalProvider
-m_vbCode : String -specs : Hashtable -state : Hashtable -o : Object
+New(in vbCode : String, in ProdState : Hashtable, in Specifications : Hashtable) +CheckSyntax() : Boolean -BuildCode() : String +CreateNewInstance() +ExecuteCode(in methodName : String) : Object

Η κλάση Eval Provider χρησιμοποιείται για την μεταγλώτιση των κανόνων των μηχανών σε πραγματικό χρόνο. Συγκεκριμένα κάθε φορά που καλείται να εργαστεί ένας πράκτορας, δημιουργείται ένα στιγμιότυπο αυτής της κλάσης η οποία αρχικοποιείται με τους κανόνες του πράκτορα. Το στιγμιότυπο που δημιουργείται, καλεί το Common Language Runtime του .Net Framework και παίρνει πληροφορίες για τον compiler της Visual Basic .NET. Αφού ολοκληρωθεί αυτή η διαδικασία χτίζεται μια εντελώς νέα πλήρης εφαρμογή στην μνήμη του υπολογιστή που περιέχει τους κανόνες του πράκτορα. Με τον όρο πλήρης εννοούμε ότι οι κανόνες περικλείονται σε μία κλάση, η οποία με την σειρά της περιέχεται σε ένα namespace. Ο ακριβής όρος που χρησιμοποιεί η Microsoft είναι Fully Qualified Assembly. Πλέον αυτό το νέο assembly είναι διαθέσιμο στον πράκτορα, χωρίς ωστόσο να μπορεί ο πράκτορας να δημιουργήσει στιγμιότυπα της κλάσης του νεογέννητου assembly, καθώς της στιγμή μεταγλώττισης της κλάσης του πράκτορα απλά δεν υπήρχε αυτό το assembly. Για αυτό το λόγο υπάρχει η μέθοδος CreateNewInstance της EvalProvider που αναλαμβάνει κάνοντας χρήση του System.Reflection Namespace του CLR, να πάρει πληροφορίες για το νέο assembly και να δημιουργήσει ένα πλήρως λειτουργικό

στιγμιότυπο της νέας κλάσης. Πλέον ο πράκτορας μπορεί μέσω της EvalProvider να κάνει κλήσεις στο νέο στιγμιότυπο κάνοντας χρήση της συνάρτησης ExecuteCode περνώντας σαν όρισμα το όνομα της εκάστοτε συνάρτησης. Τα ονόματα των συναρτήσεων που μπορούν να κλειθούν είναι προκαθορισμένα και τυποποιημένα μέσω ενός interface που ικανοποιούν όλες οι βάσεις γνώσεις των πρακτόρων.

Η κλάση EvalProvider χρησιμοποιείται επίσης από τον επεξεργαστή κειμένου που συναντάμε στον πράκτορα διεπαφής διαχειριστή, για τον έλεγχο του συντακτικού των εισηγμένων κανόνων.

### 3. Αντικείμενο FilterResults

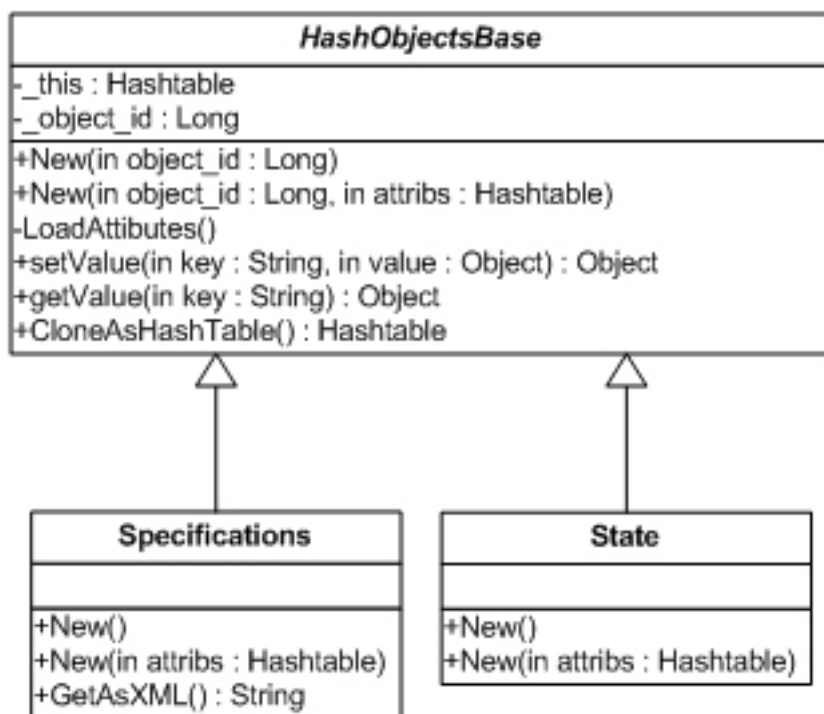
«struct»FilterResults
+l : LinearCombinations
+success : Boolean
+notes : String
+use_a : ArrayList
+use_b : ArrayList
+use_a_multiplier : ArrayList
+use_b_multiplier : ArrayList

Το αντικείμενο FilterResults αποτελεί μια δομή δεδομένων που χρησιμοποιείται στην κατάσταση του Μοντάζ του τυπογραφικού. Εκφράζει τον τρόπο τοποθέτησης πολλαπλών αντιτύπων του προϊόντος πάνω στο τυπογραφικό σαν γραμμικό συνδυασμό portrait και landscape τοποθετημένων προϊόντων καθώς και τους πιθανούς άξονες κοπής του τυπογραφικού.

### 4. Αντικείμενο HashObjectsBase

HashObjectsBase
-_this : Hashtable
-_object_id : Long
+New(in object_id : Long)
+New(in object_id : Long, in attribs : Hashtable)
-LoadAttributes()
+setValue(in key : String, in value : Object) : Object
+getValue(in key : String) : Object
+CloneAsHashTable() : Hashtable

Το αντικείμενο HashObjectsBase αποτελεί μια αφηρημένη κλάση η οποία χρησιμοποιείται για την έκφραση των οντολογιών που υποστηρίζουν οι κανόνες των πρακτόρων εργασίας. Όπως παρουσιάζεται στο παρακάτω διάγραμμα, από την HashObjectsBase κληρονομούν μερικά ή όλα τα χαρακτηριστικά τους, οι κλάσεις Specifications και State που θα δούμε παρακάτω.



## 5. Αντικείμενο Specifications

Το αντικείμενο Specifications, όπως φαίνεται και στο παραπάνω σχήμα κληρονομεί τα χαρακτηριστικά της από την αφηρημένη κλάση HashObjectsBase. Το μοναδικό πρόσθετο χαρακτηριστικό της είναι η μέθοδος GetAsXML η οποία χρησιμοποιείται από τον πράκτορα διεπαφής χρήστη για την αναζήτηση στο ιστορικό των προσομοιώσεων για προσομοίωση προϊόντων που φέρουν τις ίδιες προδιαγραφές. Εκφράζει τις προδιαγραφές του προϊόντος και τα στοιχεία της ανήκουν στο σύνολο των οντολογιών που υποστηρίζουν οι βάσεις γνώσης των πρακτόρων.

## 6. Αντικείμενο State

Το αντικείμενο State, όπως φαίνεται και στο παραπάνω σχήμα κληρονομεί τα χαρακτηριστικά της από την αφηρημένη κλάση HashObjectsBase. Εκφράζει την κατάσταση ενός προϊόντος στα διάφορα στάδια της παραγωγικής του διαδικασίας και τα στοιχεία της ανήκουν στο σύνολο των οντολογιών που υποστηρίζουν οι βάσεις γνώσης των πρακτόρων.

## 7. Αντικείμενο LinearCombinations

«struct»LinearCombinations
+a : Integer
+b : Integer
+ScrapArea : Integer
+a_multiplier : Integer
+b_multiplier : Integer
+A_PortraitPositioned : Boolean
+modaz_dimension1 : Double
+modaz_dimension2 : Double
+ModazHeight() : Double
+ModazWidth() : Double

Το αντικείμενο LinearCombinations αποτελεί μια δομή δεδομένων η οποία χρησιμοποιείται στην κατάστρωση του Μοντάζ του τυπογραφικού. Εκφράζει τον τρόπο τοποθέτησης πολλαπλών αντιτύπων του προϊόντος πάνω στο τυπογραφικό σαν γραμμικό συνδυασμό portrait και landscape τοποθετημένων προϊόντων, την επιφάνεια του τυπογραφικού που δεν καλύπτεται από προϊόν και τις καθαρές διαστάσεις του Μοντάζ.

## 8. Αντικείμενο Machine

Machine
-m_machine_id : Long
-m_machine_name : String
-m_machine_ui_name : String
-m_machine_rules : String
-m_working_state : Hashtable
-m_product_specifications : Hashtable
-eval : EvalProvider
+MachineId() : Long
+VariableName() : String
+MachineName() : String
+New(in machine_id : String, in specifications : Hashtable, in working_state : Hashtable)
-Initialize()
+Work() : MachineResults
-Compatible() : Boolean

Το αντικείμενο Machine, αποτελεί αντιπροσωπεύει τους πράκτορες εργασίας. Σε αυτό το σημείο αξίζει να παρατηρήσουμε την ύπαρξη της eval μεταβλητής η οποία είναι τύπου EvalProvider και η οποία παρέχει τη δυνατότητα δυναμικής εκτέλεσης των κανόνων.

## 9. Αντικείμενο ModazSolver



ModazSolver
-m_p1 : Double -m_p2 : Double -m_s1 : Double -m_s2 : Double -m_Toumpa : Toumpa -m_Thermotipia : Boolean = False -m_LocalUV_Plastic : Boolean = False -m_VariousExtras : Boolean = False -m_PrintingTolerance : Integer = 0
+New(in p1 : Double, in p2 : Double, in s1 : Double, in s2 : Double) +Toumpa() : Toumpa +Thermotipia() : Boolean +VariousExtras() : Boolean +LocalUV_Plastic() : Boolean -ProductHeight() : Double -ProductWidth() : Double -PaperHeight() : Double -PaperWidth() : Double -MinimumHeight() : Double -MinimumWidth() : Double -GetPossibleArrangements() : ArrayList +GetBestArrangement() : FilterResults -Max(in val1 : Double, in val2 : Double) : Double

Το αντικείμενο ModazSolver αποτελεί μια κλάση η οποία χρησιμοποιείται για την κατάστρωση του Μοντάζ του τυπογραφικού. Τοποθετεί τα προϊόντα πάνω στο τυπογραφικό λαμβάνοντας υπόψιν τόσο πιθανές συμμετρίες που απαιτούνται για την εκτύπωση του προϊόντος, όσο και τους άξονες κοπής του τυπογραφικού ώστε να είναι δυνατή η διαίρεση του για την περαιτέρω επεξεργασία σε κατάντις μηχανές.

## 10. Αντικείμενο Module1

«utility»Module1
+ConnectionString : String = "Data source=tsoukihome;User Id=sa;Password= ...,Initial Catalog=detorakis;"
+OpenConnection(inout cn : SqlConnection) +CloseConnection(inout cn : SqlConnection) +GetSafeField(in RowItem : Object) : Object

Το αντικείμενο module1 αποτελεί μια βιβλιοθήκη μεταβλητών και μεθόδων συχνής χρήσης, προσβάσιμων από όλα τα αντικείμενα της εφαρμογής. Συγκεκριμένα παρέχει μια μεταβλητή απαραίτητη για την σύνδεση με την βάση δεδομένων και τρεις μεθόδους οι οποίες παρουσιάζονται ακολούθως:

- ⇒ OpenConnection: Η μέθοδος αυτή αφού πρώτα ελέγχει αν το αντικείμενο τύπου SqlConnection είναι αρχικοποιημένο, φροντίζει για όλες τις λειτουργίες που πρέπει να γίνουν και επιστρέφει στον χρήστη της, μια πλήρως λειτουργική σύνδεση με την βάση δεδομένων.
- ⇒ CloseConnection: Η μέθοδος αυτή φροντίζει για την ασφαλή αποσύνδεση από την βάση δεδομένων.

⇒ GetSafeField: Η μέθοδος αυτή παρέχει μια επαναχρησιμοποιούμενη λύση για τον έλεγχο null ανακτώμενων πεδίων από την βάση δεδομένων.

Σε αυτό το σημείο αξίζει να σημειώσουμε το σπουδαίο όφελος που αποκομίζουμε από την χρήση μιας κοινής μεταβλητής κειμένου για την σύνδεση με την βάση δεδομένων, επειδή με αυτό το τρόπο ενεργοποιούμε μια σημαντική λειτουργία του SQLServer που δεν είναι άλλη από την δημιουργία cache συνδέσεων, γεγονός που συνεπάγεται μικρότερους χρόνους αρχικοποίησης μιας σύνδεσης με την βάση δεδομένων και αφετέρου την διατήρηση μικρού αριθμού ταυτόχρονων συνδέσεων, ενός πόρου υπερπολύτιμου ιδιαίτερα όταν ο SqlServer εξυπηρετεί συγχρόνως δυναμικές ιστοσελίδες ή αλλές δικτυακές ή διαδικτυακές εφαρμογές.

## 11. Αντικείμενο PolarSolver

PolarSolver
-state : Hashtable
-specs : Hashtable
-minHeight : Double
-minWidth : Double
+New(in st : Hashtable, in sp : Hashtable)
+Polar() : Hashtable

Το αντικείμενο PolarSolver αποτελεί μια κλάση η οποία αναλαμβάνει την διαίρεση του τυπογραφικού σε συμμετρικά κομμάτια, μεγέθους μικρότερου ή ίσου από αυτό που καθορίζει ο πράκτορας εργασίας που την χρησιμοποιεί.

## 12. Αντικείμενο Procedure

Procedure
-m_ProcedureName : String
-m_ProcedureUiName : String
-m_ProcedureId : Long
-m_Machines : ArrayList
+ProcedureId() : Long
+VariableName() : String
+ProcedureName() : String
+New(in procedure_id : Long)
-Initialize()
+GetMachines() : ArrayList

Το αντικείμενο Procedure αποτελεί μια κλάση η οποία χρησιμοποιείται για την ομαδοποίηση των πρακτόρων εργασίας βάση των εργασιών που αυτοί εκπονούν. Παρατηρώντας τα χαρακτηριστικά της κλάσης, σημειώνουμε την μέθοδο GetMachines, η οποία επιστρέφει στον πράκτορα ανάθεσης εργασιών που την καλεί, όλους τους κατάλληλους πράκτορες εργασιών. Σε προηγούμενη παράγραφο αναφέραμε ότι υπεύθυνοι για την επιλογή των πρακτόρων είναι εργαζόμενοι πράκτορες εργασίας. Το μήνυμα που στέλνει κάθε εργαζόμενος πράκτορας δεν είναι ένα σύνολο επόμενων πρακτόρων, αλλά ένας δείκτης που υποδεικνύει αυτό το σύνολο και δεν είναι άλλος από το procedureid της κλάσης που μελετάμε.

### 13. Αντικείμενο frmMain

frmMain
+m_machines : frmMachines +m_procedures : frmProcedures +m_run : frmRun +m_settings : frmSettings +m_rules : frmRules -components : IContainer +tlbMain : ToolBar +ToolBarButton1 : ToolBarButton +ToolBarButton2 : ToolBarButton +ToolBarButton3 : ToolBarButton +ToolBarButton4 : ToolBarButton +ImageList1 : ImageList +ToolBarButton5 : ToolBarButton +pnlMain : Panel
+New() #Dispose(in disposing : Boolean) -InitializeComponent() -CloseAllForms() -ShowRules() -ShowSettings() -ShowProcedures() -ShowMachines() -ShowRun() -tlbMain_ButtonClick(in sender : Object, in e : ToolBarButtonClickEventArgs) -frmMain_Closed(in sender : Object, in e : EventArgs)

Το αντικείμενο frmMain αποτελεί το στατικό μέρος του γραφικού περιβάλλοντος της εφαρμογής και χρησιμεύει για την πλοήγηση μεταξύ των υπολοίπων φορμών που παρουσιάστηκαν στην παράγραφο παρουσίασης των πρακτόρων.

### 14. Αντικείμενο frmMachines

Το αντικείμενο frmMachines αποτελεί την κλάση από την οποία προέρχεται η φόρμα διαχείρισης των μηχανών που παρουσιάστηκε σε προηγούμενη παράγραφο. Το διάγραμμα της φαίνεται ακολούθως.

frmMachines
-components : IContainer +Panel1 : Panel +Label1 : Label +ColumnHeader1 : ColumnHeader +ColumnHeader2 : ColumnHeader +ColumnHeader3 : ColumnHeader +Splitter1 : Splitter +Panel2 : Panel +GroupBox1 : GroupBox +Label4 : Label +Label3 : Label +Label2 : Label +lvMachines : ListView +btnDeleteMachine : Button +btnSaveMachine : Button +txtMachineName : TextBox +txtMachineVariable : TextBox +txtMachineId : TextBox +cbProcedure : ComboBox +Label5 : Label +btnNewMachine : Button +ColumnHeader4 : ColumnHeader
+New() #Dispose(in disposing : Boolean) -InitializeComponent() -frmMachines_Load(in sender : Object, in e : EventArgs) -LoadMachines() -LoadProcedures() -lvMachines_SelectedIndexChanged(in sender : Object, in e : EventArgs) -SaveMachine() : <unspecified> -btnSaveMachine_Click(in sender : Object, in e : EventArgs) -GetProcedureName(in procedure_id : String) : String -btnDeleteMachine_Click(in sender : Object, in e : EventArgs) -btnNewMachine_Click(in sender : Object, in e : EventArgs)

### 15. Αντικείμενο frmRun

Το αντικείμενο frmRun αποτελεί την φόρμα εισαγωγής των προδιαγραφών του προς προσομοίωση προϊόντος και παρουσίασης των αποτελεσμάτων. Από το παρακάτω διάγραμμα σημειώνουμε τις μεθόδους ExistsInHistory, η οποία ελέγχει για την ύπαρξη παλιότερης προσομοίωσης του προϊόντος, και την Work η οποία καλεί τον πράκτορα ανάθεσης εργασιών να ξεκινήσει η διαδικασία προσομοίωσης.

frmRun
-m_specs : Specifications -components : IContainer +TabControl1 : TabControl +TabPage1 : TabPage +TabPage2 : TabPage +GroupBox1 : GroupBox +Label1 : Label +cbProductType : ComboBox +Label2 : Label +Label3 : Label +txtProdHeight : TextBox +txtProdLength : TextBox +Label4 : Label +Label5 : Label +txtSpanCount : TextBox +Label6 : Label +txtPageLength : TextBox +txtPageHeight : TextBox +txtQuantity : TextBox +Label7 : Label +txtFace2 : TextBox +txtFace1 : TextBox +Label8 : Label +Label9 : Label +GroupBox2 : GroupBox +cPackaging : CheckBox +cKopi : CheckBox +cParaggeliolipsia : CheckBox +cSinthetiki16Selidwn : CheckBox +cScanning : CheckBox +cPikmansi : CheckBox +cPhotography : CheckBox +cPlastikopolish : CheckBox +cSend : CheckBox +cGlobalUV : CheckBox +cLocalUV : CheckBox +cNumbering : CheckBox +cAnaglyph : CheckBox +cThermal : CheckBox +CheckBox1 : CheckBox +btnExecute : Button +btnNew : Button +tv : TreeView +lblTime : Label
+New() #Dispose(in disposing : Boolean) -InitializeComponent() -frmRun_Load(in sender : Object, in e : EventArgs) -LoadProductTypes() -cbProductType_SelectedIndexChanged(in sender : Object, in e : EventArgs) -btnNew_Click(in sender : Object, in e : EventArgs) -btnExecute_Click(in sender : Object, in e : EventArgs) -ExistsInHistory() : Long -CreateSpecifications() -Work() -LoadTree(in xml_file : String) -PopulateTreeControl(in document : XmlNode, in nodes : TreeNodeCollection)

## 16. Αντικείμενο frmRules

frmRules
<pre> -components : IContainer +pnlMachines : Panel +Splitter1 : Splitter +pnlObjects : Panel +Splitter2 : Splitter +pnlEditor : Panel +tvObjects : TreeViewEX +Label1 : Label +lvMachines : ListView +ColumnHeader6 : ColumnHeader +ColumnHeader7 : ColumnHeader +ColumnHeader8 : ColumnHeader +ColumnHeader9 : ColumnHeader +Panel1 : Panel +Panel2 : Panel +Panel3 : Panel +Splitter3 : Splitter +GroupBox1 : GroupBox +lblProcedure : Label +lblVariable : Label +lblName : Label +lblId : Label +Label5 : Label +Label4 : Label +Label3 : Label +Label2 : Label +GroupBox2 : GroupBox +lvVariables : ListView +ColumnHeader4 : ColumnHeader +ColumnHeader1 : ColumnHeader +ColumnHeader2 : ColumnHeader +ColumnHeader3 : ColumnHeader +ColumnHeader5 : ColumnHeader +GroupBox3 : GroupBox +btnInterface : Button +btnSave : Button +btnCheckSyntax : Button +txtRules : RichTextBox +Panel4 : Panel +GroupBox4 : GroupBox +lvProcedures : ListView +ColumnHeader10 : ColumnHeader +ColumnHeader11 : ColumnHeader +ColumnHeader12 : ColumnHeader +Splitter4 : Splitter +Panel5 : Panel +GroupBox5 : GroupBox +lvProductTypes : ListView +ColumnHeader13 : ColumnHeader +ColumnHeader14 : ColumnHeader +ColumnHeader15 : ColumnHeader -WM_SETREDRAW : Integer = 11 -WM_USER : Integer = 1024 -EM_GETEVENTMASK : Integer = (ExpertReflection.frmRules.WM_USER + 59) -EM_SETEVENTMASK : Integer = (ExpertReflection.frmRules.WM_USER + 69) -pat : String  +New() #Dispose(in disposing : Boolean) -InitializeComponent() -frmRules_Load(in sender : Object, in e : EventArgs) -SendMessage(in hWnd : IntPtr, in msg : Integer, in wParam : Integer, in lParam : IntPtr) : IntPtr -CofigureEditor() -Highlight() -txtRules_TextChanged(in sender : Object, in e : EventArgs) -LoadTree() -LoadVariables(in object_id : Long) -tvObjects_AfterSelect(in sender : Object, in e : TreeViewEventArgs) -LoadProductTypes() -LoadProcedures() -LoadMachines() -lvMachines_SelectedIndexChanged(in sender : Object, in e : EventArgs) -GetRule(in machine_id : Long) : String -btnCheckSyntax_Click(in sender : Object, in e : EventArgs) -btnSave_Click(in sender : Object, in e : EventArgs) -btnInterface_Click(in sender : Object, in e : EventArgs) </pre>

Το αντικείμενο frmRules αποτελεί την φόρμα επεξεργασίας των κανόνων των πρακτόρων, που συναντήσαμε σε προηγούμενη. Από το παραπάνω διάγραμμα αξίζει να επισημάνουμε τις btnCheckSyntax\_Click η οποία ελέγχει το συντακτικό των κανόνων της εκάστοτε μηχανής και Highlight η οποία αναλαμβάνει τον χρωματισμό των λέξεων, πάνω στον επεξεργαστή κειμένου. Η εύρεση και επιλογή των λέξεων προς χρωματισμό γίνεται με την χρήση ευφών μεθόδων pattern matching που προσφέρουν οι σύγχρονες γλώσσες προγραμματισμού μέσω της χρήσης Regular Expressions.

## 17. Αντικείμενο frmProcedures

frmProcedures
-components : IContainer +Panel1 : Panel +Splitter1 : Splitter +Panel2 : Panel +ColumnHeader1 : ColumnHeader +ColumnHeader2 : ColumnHeader +ColumnHeader3 : ColumnHeader +Label1 : Label +lvProcedures : ListView +GroupBox1 : GroupBox +Label2 : Label +Label3 : Label +Label4 : Label +txtProcedureId : TextBox +txtProcedureVariable : TextBox +txtProcedureName : TextBox +btnSaveProcedure : Button +btnDeleteProcedure : Button +btnNewProcedure : Button +Label5 : Label +lvMachines : ListView
+New() #Dispose(in disposing : Boolean) -InitializeComponent() -frmProcedures_Load(in sender : Object, in e : EventArgs) -LoadProcedures() -lvProcedures_SelectedIndexChanged(in sender : Object, in e : EventArgs) -btnSaveProcedure_Click(in sender : Object, in e : EventArgs) -SaveProcedure() : Long -btnNewProcedure_Click(in sender : Object, in e : EventArgs) -btnDeleteProcedure_Click(in sender : Object, in e : EventArgs)

Το αντικείμενο frmProcedures αποτελεί την κλάση από την οποία προέρχεται η φόρμα διαχείρισης των διαδικασιών που παρουσιάστηκε σε προηγούμενη παράγραφο.

## 5. Παράδειγμα εκτέλεσης του προγράμματος

Στην παράγραφο αυτή θα δούμε ένα παράδειγμα εκτέλεσης της εφαρμογής του προγράμματος. Για την παρουσίαση αυτή θα χρησιμοποιήσουμε την περίπτωση παραγωγής αφίσσας.

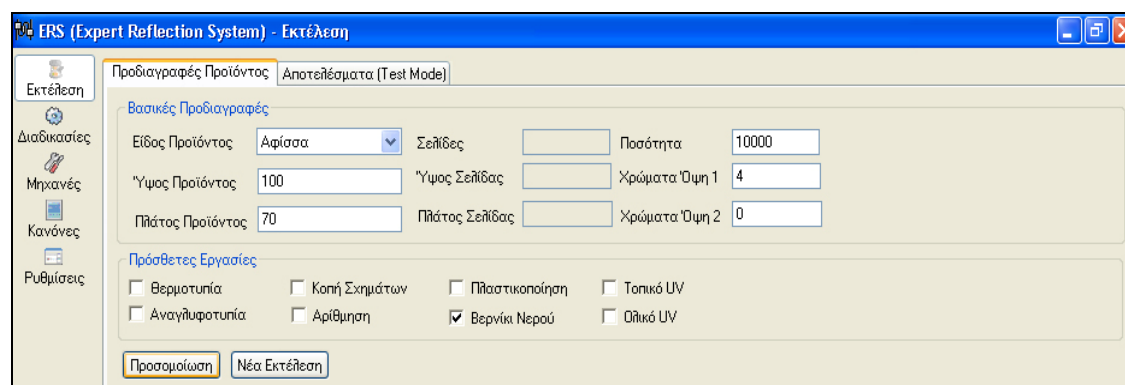
Τα δεδομένα που είναι απαραίτητο να εισαχθούν στο σύστημα, ώστε να ξεκινήσει η διαδικασία προσομοίωσης καθώς και οι τιμές που δώσαμε στο παράδειγμά μας, είναι:

- ⇒ Είδος προϊόντος (Αφίσσα).
- ⇒ Ύψος προϊόντος (100 εκ.)
- ⇒ Πλάτος προϊόντος (70 εκ.)
- ⇒ Ποσότητα (10.000 τεμάχια)
- ⇒ Χρώματα στην πρώτη όψη (4 χρώματα)
- ⇒ Χρώματα στην δεύτερη όψη (0 χρώματα)

Τα παραπάνω δεδομένα είναι αρκετά για τον προσδιορισμό των προδιαγραφών ενός απλού προϊόντος, ενώ για τον προσδιορισμό περισσότερο σύνθετων προϊόντων ο χρήστης δεν έχει παρά να επιλέξει κάποιες από τις εργασίες που παραθέτονται στην δεύτερη περιοχή της οθόνης του προγράμματος. Προς το παρόν οι προαιρετικές εργασίες που αντιμετωπίζει το πρόγραμμα, είναι:

- ⇒ Θερμοτυπία
- ⇒ Αναγλυφοτυπία
- ⇒ Κοπή Σχημάτων
- ⇒ Αρίθμηση
- ⇒ Πλαστικοποίηση
- ⇒ Βερνίκι Νερού (Επιλεγμένη στο παράδειγμα)
- ⇒ Τοπικό UV
- ⇒ Ολικό UV

Στην παρακάτω εικόνα παρουσιάζεται το περιβάλλον εργασίας του χρήστη και παρουσιάζονται οι διάλογοι εισαγωγής των προδιαγραφών του προϊόντος με τα παραπάνω δεδομένα εισηγμένα.

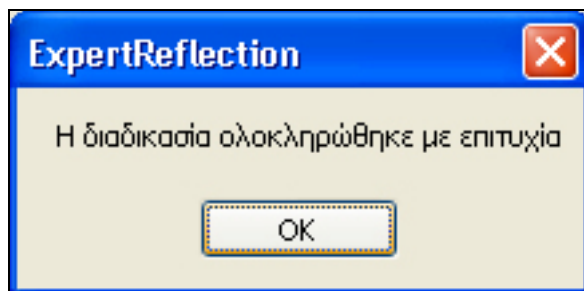


Εικόνα 18: Φόρμα εισαγωγής προδιαγραφών προϊόντος



Αφού ολοκληρωθεί η διακασία εισαγωγής δεδομένων από τον χρήστη, πιέζοντας το κουμπί «Προσομοίωση», γίνεται έλεγχος της εγκυρότητας των εισηγμένων δεδομένων και ξεκινάει η διαδικασία προσομοίωσης. Αν για οποιοδήποτε λόγο παρουσιαστεί κάποιο σφάλμα στην εγκυρότητα των εισηγμένων δεδομένων, εμφανίζεται κατάλληλο μήνυμα στον χρήστη που τον ενημερώνει για την φύση του σφάλματος.

Με το που ολοκληρωθεί η διαδικασία της προσομοίωσης της παραγωγής του προϊόντος κατάλληλο μήνυμα εμφανίζεται στον χρήστη που τον ενημερώνει για την έκβαση της προσομοίωσης .



Εάν η εκτέλεση είναι επιτυχής ο χρήστης μεταφέρεται αυτόματα στην οθόνη αποτελεσμάτων, όπου έχει την δυνατότητα να μελετήσει το δέντρο πιθανών γραμμών παραγωγής που προτείνει το σύστημα. Στιγμιότυπο του δέντρου που δημιουργείται φαίνεται παρακάτω.

Στην εικόνα 19, παρατηρούμε τον τέταρτο κόμβο «procedure (procedure\_id=7) (procedure\_name=Αποθήκη Χαρτιού)» ο οποίος αναπαριστά την έναρξη της αντίστοιχης διαδικασίας. Ο κόμβος αυτός περιέχει δύο παιδιά, τους κόμβους state και machines.

Ο κόμβος state αναπαριστά την κατάσταση του προϊόντος κατά την έναρξη της διαδικασίας. Τα παιδιά του εν λόγω κόμβου αντιστοιχούν στις μεταβλητές που καθορίζουν την κατάσταση του προϊόντος, όπως για παράδειγμα το ύψος (κόμβος paper\_height) και το πλάτος (κόμβος paper\_length) του χαρτιού.

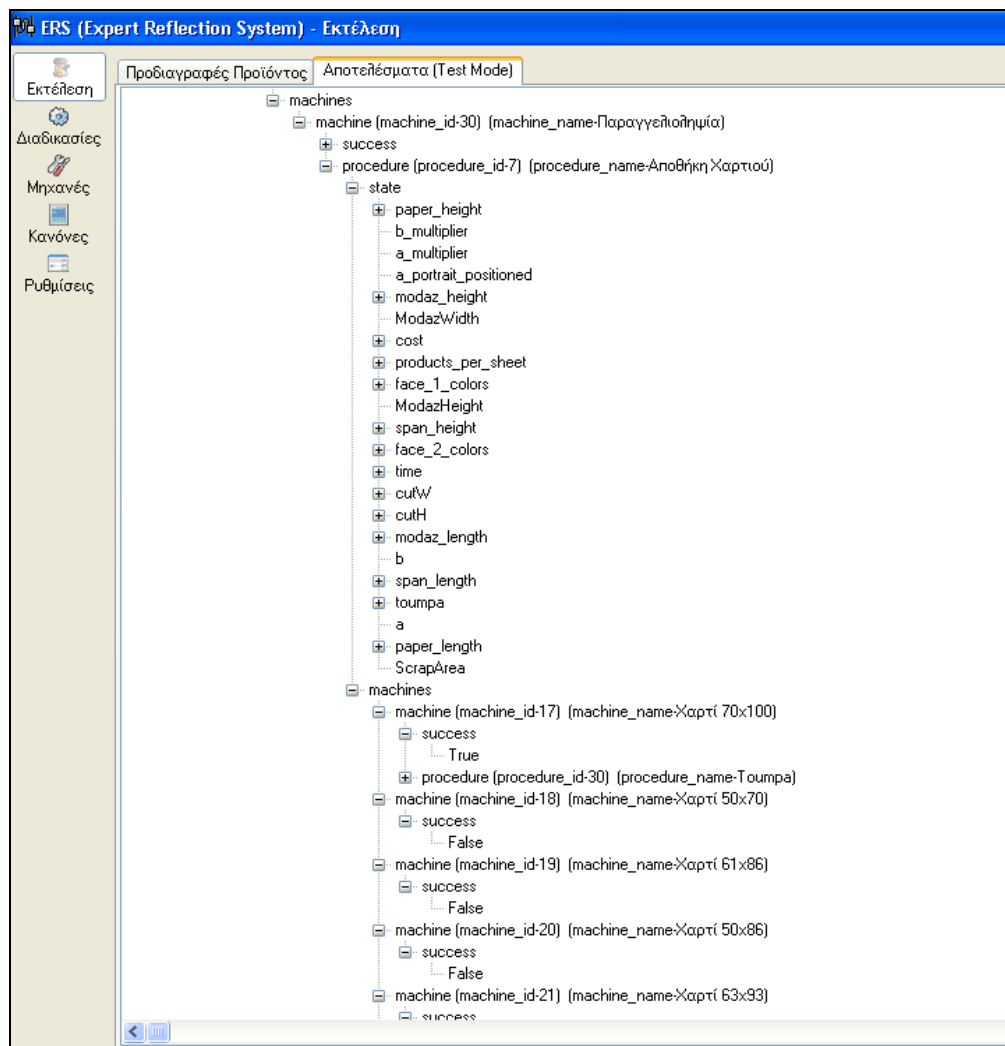
Ο κόμβος machines αναπαριστά το σύνολο των μηχανών που εξετάζουμε για την εκτέλεση της διαδικασίας. Κάθε παιδί του κόμβου αποτελεί μια υποψήφια μηχανή.

Στη συνέχεια κάθε κόμβος που αντιστοιχεί σε υποψήφια μηχανή (κόμβοι machine), περιέχει ένα υποχρεωτικό παιδί, τον κόμβο success και ένα προαιρετικό παιδί τον κόμβο procedure.

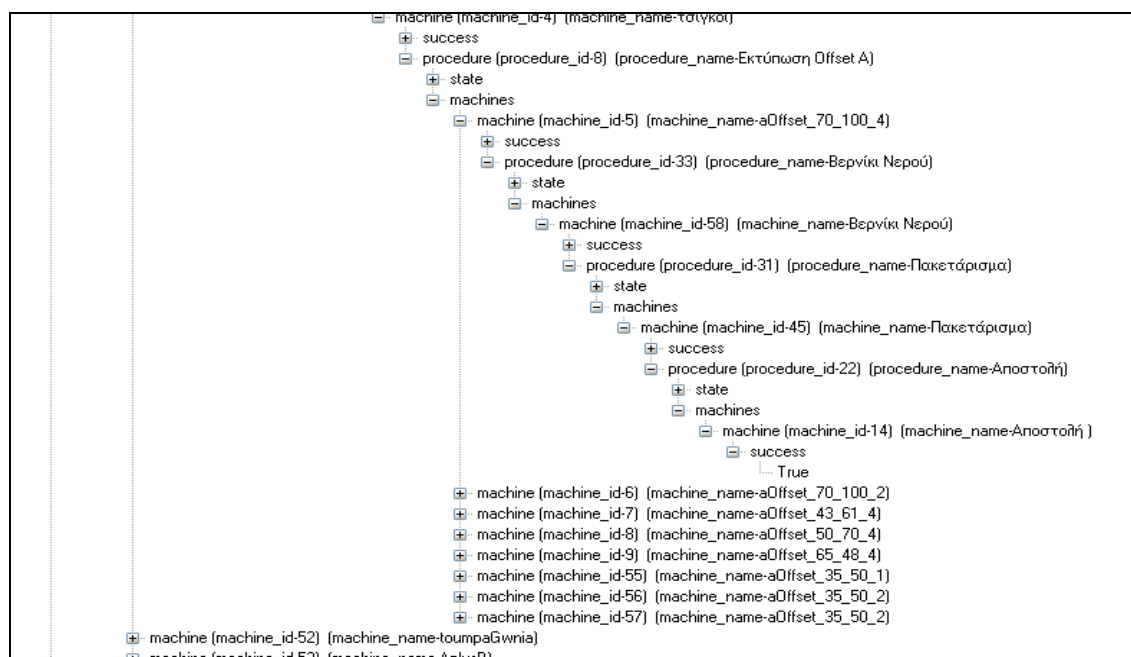
Ο κόμβος success δηλώνει εάν χρησιμοποιήθηκε αυτή η μηχανή. Εάν η μηχανή χρησιμοποιήθηκε λαμβάνει την τιμή true. Διαφορετικά την τιμή false.

Ο κόμβος procedure εμφανίζεται εάν και εφόσον ο κόμβος success, πήρε την τιμή true. Είναι ανάλογος του πρώτου κόμβου που μελετήσαμε και δηλώνει την επόμενη διαδικασία που θα εκτελεστεί. Ο κόμβος state που θα περιέχει ο κόμβος procedure, θα αναπαριστά την κατάσταση στην οποία περιήλθε το προϊόν κατόπιν της επίδρασης της μηχανής που αντιστοιχεί στον κόμβο machine.

Στη συνέχεια το δέντρο δομείται κατά τον ίδιο ακριβώς τρόπο. Η μόνη ιδιαιτερότητα που συναντάμε, είναι στα φύλλα επιτυχημένων διαδρομών όπου παρατηρούμε ένα κόμβο τύπου machine (εικόνα 20), ο οποίος παρά το γεγονός ότι περιέχει κόμβο τύπου success με τιμή true, δεν περιέχει κόμβο procedure (αφού έχει ολοκληρωθεί η διαδικασία!).



Εικόνα 19: Φόρμα παρουσίασης αποτελεσμάτων.



Εικόνα 20: Τα φύλλα του δέντρου των γραμμών παραγωγής

Σημειώνουμε, ότι ο τρόπος παρουσίασης των αποτελεσμάτων που περιγράψαμε παραπάνω και που βλέπουμε στην επόμενη εικόνα είναι προσωρινός. Στην τελική έκδοση της εφαρμογής τα αποτελέσματα θα παρουσιάζονται με την μορφή πολλαπλών λιστών. Κάθε λίστα θα αναπαριστά μία επιτυχήμενη διαδρομή του δέντρου. Θα παρουσιάζονται οι 5 ή 10 βέλτιστες διαδρομές .

Ο λόγος που δεν έχει υλοποιηθεί έως τώρα αυτή η λειτουργικότητα, οφείλεται στο γεγονός ότι δεν έχουν ορισθεί οι παράμετροι των οδηγών κόστους των μηχανών του εργοστασίου, οι οποίοι είναι απαραίτητοι για τον προσδιορισμό των βέλτιστων διαδρομών.

## 6. Συμπεράσματα – Επεκτάσεις

Η εργασία αυτή πρότεινε τη σχεδίαση ενός Συστήματος Υποστήριξης Αποφάσεων βασισμένου στην τεχνολογία πρακτόρων και Έμπειρων Συστημάτων, με τρόπο λειτουργικό και κατανοητό από το χρήστη του συστήματος.

Η επιλογή αυτής της σύγχρονης τεχνολογίας του γνωστικού πεδίου της Κατανεμημένης Τεχνητής Νοημοσύνης έγινε εξαιτίας των δυνατοτήτων για περαιτέρω ανάπτυξη του συστήματος που προσφέρει η τεχνολογία αυτή. Το σύστημα βέβαια είναι ικανό όπως προτάθηκε, να βοηθήσει το χρήστη στη λήψη απόφασης για ανάθεση εργασιών, όμως τα περιθώρια βελτίωσης του είναι πολλά και εμφανή.

Κύρια φιλοδοξία του γράφοντα και του επιβλέποντα καθηγητή της εργασίας αυτής κ. Ματσατσίνη είναι η ανάπτυξη του συστήματος σε μία ολοκληρωμένη εφαρμογή. Για να θεωρηθεί επιτυχημένη η εφαρμογή αυτή, θα πρέπει το Σ.Υ.Α. που θα προκύψει να είναι φιλικό προς το χρήστη, να συνεργάζεται με άλλα πληροφοριακά συστήματα, να διαθέτει ευελιξία ώστε να προσαρμόζεται στις απαιτήσεις των χρηστών του, να διασφαλίζεται η αποτελεσματικότητα του και να συνδυάζει τη συνεργασία μεθοδολογιών και μοντέλων με την παρουσίαση των αποτελεσμάτων (Sprague και Carlson (1982), Ματσατσίνης (2001)).

Ο σχεδιασμός των επόμενων μηνών περιλαμβάνει την τελειοποίηση της εφαρμογής αλλά και τη περαιτέρω γενίκευση των μεθόδων, τεχνολογιών και τεχνικών που χρησιμοποιήσαμε ώστε να είναι δυνατή η μοντελοποίηση και άλλων προβλημάτων.

Οι τεχνικές λεπτομέρειες που θα αναδειχθούν κατά την λειτουργία της εφαρμογής και η αξιολόγηση της αποδοτικότητάς της, θα αποτελέσουν το πρώτο βήμα για την αντιμετώπιση αυτής της νέας πρόκλησης

Κάνοντας μια επισκόπηση των προσπαθειών που καταβλήθηκαν το τελευταίο έτος, τις γνώσεις που αποκομίσαμε και τις εμπειρίες που βιώσαμε αξίζει να σημειώσουμε τα παρακάτω.

Πρώτον, τη τεράστια κινητικότητα που παρατηρείται στην ακαδημαϊκή, επιστημονική και επιχειρηματική κοινότητα σχετικά με την τεχνολογία των πρακτόρων και γενικότερα της Κατανεμημένης Τεχνητής Νοημοσύνης. Χαρακτηριστικά αναφέρουμε την συνεχή εξέλιξη της πλατφόρμας για πολυπρακτορικά συστήματα Jade που πλέον φέρει και ειδικά χαρακτηριστικά που την κάνει συμβατή με τη νέα γενιά Έξυπνων Συσκευών (Jade/Leap) και την ανάπτυξη νέων κελύφων για την ανάπτυξη Έμπειρων Συστημάτων όπως το NxBre.

Δεύτερον, τη διαπίστωση καταστάσεων που αν και αναφέρονται σε πλήθος επιστημονικών συγγραμμάτων δεν είχαν γίνει πλήρως κατανοητές ή δεν τους είχε αποδοθεί η κατάλληλη σημασία από τον γράφοντα. Χαρακτηριστικό παράδειγμα είναι η διαδικασία εκμαίευσης της γνώσης και γενικότερα η επικοινωνία του μηχανικού γνώσης (που στην συγκεκριμένη περίπτωση ήταν και ο προγραμματιστής) και του ειδικού. Μια διαδικασία που αποδείχθηκε πραγματικά επίπονη, χρονοβόρα και ακριβή.

Το τελευταίο σημείο στο οποίο θα θέλαμε να σταθούμε, είναι η σημαντική εξάρτηση του επιστήμονα από τεχνικές γνώσεις όπως ο προγραμματισμός που του επιτρέπουν να υλοποιήσει, να δοκιμάσει, να υιοθετήσει ή και να απορρίψει θεωρητικά μοντέλα.

## 7. Βιβλιογραφία

Ματσατσίνης Νικόλαος, Σπανουδάκης Νικόλαος, Σαμαράς Ανδρέας, 2005, Εισαγωγή στην Τεχνητή Νοημοσύνη και στα Συστήματα Πολλαπλών Πρακτόρων, Εκδόσεις Νέων Τεχνολογιών

Corkill, D.D. and Lesser, V.R., 1983. The use of meta-level control for coordination in a distributed problem solving network. Proc. Int. Jt. Conf. Artif. Intell, 8<sup>th</sup>, Karlsruhe, Germany: 784-756.

Erman, L.D., Hayes-Roth, F., Lesser, V.R., and Reddy, D.R..1980. The HEARSAY-II speech-understanding system: Integrating knowledge to resolve uncertainty. Computing Surveys, 12(2): 213-253.

Fujita, S. and V. Lesser, (1996), Centralized Task Distribution in the Presence of Uncertainty and Time Deadlines, ICMAS-96.

Iglesias, C., Garijo, M., Gonzalez, J., A Survey of Agent-Oriented Methodologies. In: Muller, J.P., Singh, M.P., Rao, A.S., (Eds.): Intelligent agents V Agent Theories, Architectures and Languages. Lecture Notes in Computer Science, Vol. 1555. Springer-Verlag, Berlin Heidelberg (1998) 185-198.

Jennings, N.R., Sycara, K., Wooldridge, M.. 1998 A Roadmap of Agent Research and Development. Int Journal of Autonomous Agents and Multi-Agent Systems 1 (1):7-38.

Keen, P.G.W. and M.S. Scott-Morton, 1978, Decision support systems: An organizational perspective, Reading, MA: Addison-Wesley.

Li, Y. and Zhang C., 1999, Information Fusion and Decision Making for Utility-based Agents, School of Computing and Mathematics, Deakin University.

Little, J.D.C. (1970), Models and managers: The concept of decision calculus, Management Science, vol. 16, no.8, pp 466-489.

Malone, T.W.. 1990. Organizing information processing systems: Parallels between human organizations and computer systems. Cognition, Computation and Cooperation. Eds. Zachary, W.W. and Robertson, S.P.. Ablex, Norwood, NJ: 56-83.

Mark F. Wood and Scott A. DeLoach, An Overview of the Multiagent Systems Engineering Methodology, Department Of Electrical and Computer Engineering, Air Force Institute of Technology, 2001.

Multiagent Systems MIT Press, 1999.

Smith, R., (1980), The Contract Net Protocol: High Level communication and control in a distributed problem-solver, IEEE Transaction on Computers C-29(12):1104-1113.

Sprague, R.H. and E.D. Carlson (1982), Building effective Decision support systems, MIS Quarterly, vol. 4, no. 4.

Tidhar, G., S. Rao and E. Sonnenberg, (1996), Guided Team Selection, ICMAS-96.

Walsh, W. and Wellman, (1998), A Market Protocol for Decentralized Task Allocation, IEEE 1998.

Wooldridge M. and Jennings N.R., “Intelligent agents. Theory and Practice. The Knowledge Engineering Review”, 1995.

Hayes-Roth, Frederick. Waterman, D. A., (Donald Arthur) -Lenat, Douglas B. “Building expert systems”, 1983

Kumara, Soundar T., Soyster, Allen L. Kashyap, Rangasami L., (Rangasami Laksminarayana), “Artificial intelligence: manufacturing theory and practice”, 1989

Ματσατσίνης, Ν.Φ., Συστήματα Υποστήριξης Αποφάσεων, Σημειώσεις μαθήματος Συστήματα Υποστήριξης Αποφάσεων του τμήματος Μηχανικών Παραγωγής & Διοίκησης του Πολυτεχνείου Κρήτης, Χανιά, 2000.

Μωραΐτης Π. , 2001 . Μηχανική Γνώσεων και αποφάσεων. Σημειώσεις μαθήματος Μηχανική Γνώσεων και Αποφάσεων του τμήματος Μηχανικών Παραγωγής & Διοίκησης του Πολυτεχνείου Κρήτης , κεφάλαιο 7 .

Σπανουδάκης, Ν.Ι., Αρχιτεκτονική και επικοινωνία ενός βασισμένου στην τεχνολογία πρακτόρων Συστήματος Υποστήριξης Αποφάσεων, Ερευνητική Μεταπτυχιακή Διατριβή, Πολυτεχνείο Κρήτης τμήμα Μηχανικών Παραγωγής & Διοίκησης, Χανιά, 2001.

Ψωματάκης, Ε.Μ., Ανάλυση και μοντελοποίηση σχεδίασης δράσης και λογικής κρίσης πρακτόρων σε πρότυπο σύστημα πολλαπλών πρακτόρων για την υλοποίηση Συστήματος Υποστήριξης Αποφάσεων, Ερευνητική Μεταπτυχιακή Διατριβή, Πολυτεχνείο Κρήτης τμήμα Μηχανικών Παραγωγής & Διοίκησης, Χανιά, 2001.

## Ευρετήριο Διαγραμμάτων

Εικόνα 1: Η πρότυπη δομή ενός ΣΥΑ .....	13 -
Εικόνα 2: Ένας πράκτορας σε αλληλεπίδραση με το περιβάλλον του. Ο πράκτορας παίρνει αισθητήρια είσοδο από το περιβάλλον του και παράγει σαν έξοδο δράσεις που το επηρεάζουν (Πηγή : Multiagent Systems, MIT Press, 1999). -	15 -
Εικόνα 3: Κάθετα στρωματοποιημένη αρχιτεκτονική.....	19 -
Εικόνα 4: Οριζόντια Στρωματοποιημένη Αρχιτεκτονική πράκτορα.....	19 -
Εικόνα 5: Σχέσεις μεταξύ των μοντέλων της μεθοδολογίας Gaia.....	30 -
Εικόνα 6: Η μεθοδολογία MaSE. (Πηγή Wood and DeLoach, 2001).....	32 -
Εικόνα 7: Παρούσα Διαδικασία Σύνταξης Προσφορών.....	47
Εικόνα 8: Αρχιτεκτονική του Συστήματος.....	56
Εικόνα 9: Πράκτορας διεπαφής τελικού χρήστη. Οθόνη προσδιορισμού των προδιαγραφών του προϊόντος.....	57
Εικόνα 10: Πράκτορας διεπαφής τελικού χρήστη. Έλεγχος εγκυρότητας δεδομένων εισόδου.....	57
Εικόνα 11: Πράκτορας διεπαφής τελικού χρήστη. Ανάκτηση αποτελεσμάτων προσομοίωσης ίδιων προϊόντων για την αποφυγή πολλαπλών εκτελέσεων προσομοιώσεων των οποίων διαθέτουμε ήδη τα αποτελέσματα.....	58
Εικόνα 12: Πράκτορας διεπαφής τελικού χρήστη. Παρουσίαση αποτελεσμάτων. Στην συγκεκριμένη εικόνα παρουσιάζεται ένα δέντρο του οποίου κάθε κόμβος, αντιστοιχεί σε αντίστοιχο στοιχείο (element) του αρχείου αποτελεσμάτων. ....	59
Εικόνα 13: Γεωγραφική αποδέσμευση του πράκτορα διεπαφής τελικού χρήστη. Οι προδιαγραφές του προϊόντος αποστέλονται μέσω κοινών φορμών, εκτελείται η προσομοίωση στον εξυπηρετητή και επιστρέφεται το όνομα του XML αρχείου αποτελεσμάτων. ....	60
Εικόνα 14: Οθονή επεξεργασίας διαδικασιών.....	65
Εικόνα 15: Οθονή επεξεργασίας μηχανών.....	66
Εικόνα 16: Οθονή επεξεργασίας κανόνων.....	67
Εικόνα 17: Δέντρο αντικειμένων της εφαρμογής.....	68
Εικόνα 18: Φόρμα εισαγωγής προδιαγραφών προϊόντος.....	80
Εικόνα 19: Φόρμα παρουσίασης αποτελεσμάτων.....	82
Εικόνα 20: Τα φύλλα του δέντρου των γραμμών παραγωγής.....	82