



Τεχνολογικό Εκπαιδευτικό Ίδρυμα Πάτρας

Σχολή Διοίκησης Και Οικονομίας

Τμήμα Επιχειρηματικού Σχεδιασμού Και Πληροφοριακών Συστημάτων

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΚΑΙ ΕΠΙΛΥΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ
ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΠΡΟΣΩΠΙΚΟΥ
ΤΗΛΕΦΩΝΙΚΩΝ ΚΕΝΤΡΩΝ**

ΛΙΝΑΡΔΟΣ Π. ΓΕΩΡΓΙΟΣ

ΝΥΧΤΕΡΙΔΑΣ Ν. ΚΩΝΣΤΑΝΤΙΝΟΣ

ΕΠΟΠΤΕΥΩΝ: ΠΡΑΓΙΑΤΗ ΑΓΓΕΛΙΚΗ



Copyright © Λινάρδος Π. Γεώργιος, Νυχτερίδας Ν. Κωνσταντίνος 2006

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τους συγγραφείς.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τους συγγραφείς και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις το Τεχνολογικό Εκπαιδευτικό Ίδρυμα Πάτρας.

Ευχαριστίες

Η πτυχιακή εργασία εκπονήθηκε με πολύ κόπο και χρόνο. Την βιώσαμε σαν ένα απόσταγμα, μία κατακλείδα των σπουδών μας, κάτι σαν την κατάληξη ενός εποικοδομητικού ταξιδιού με ανατροπές.

Θα θέλαμε να ευχαριστήσουμε θερμά την καθηγήτρια κ. Αγγελική Πραγιάτη που μας έδωσε την ευκαιρία να ασχοληθούμε με το θέμα των αλγορίθμων και του τηλεφωνικού κέντρου και να μάθουμε πράγματα για την λειτουργία τους και την δομή τους. Το ενδιαφέρον και η καθοδήγησή της κατά τη διάρκεια των συναντήσεών μας υπήρξαν πολύ σημαντικά και εποικοδομητικά.

Πίνακας περιεχομένων

ΕΙΣΑΓΩΓΗ	9
ΔΙΑΡΘΡΩΣΗ ΤΗΣ ΕΡΓΑΣΙΑΣ	111
ΚΕΦΑΛΑΙΟ 1	12
1.1 ΤΙ ΕΙΝΑΙ ΠΛΗΡΟΦΟΡΙΑΚΟ ΣΥΣΤΗΜΑ	12
1.2 ΤΥΠΟΙ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ	13
1.3 ΜΕΘΟΔΟΛΟΓΙΕΣ ΚΑΙ ΤΕΧΝΙΚΕΣ ΣΧΕΔΙΑΣΗΣ ΠΛΗΡΟΦΟΡΙΑΚΟΥ ΣΥΣΤΗΜΑΤΟΣ	16
1.4 ΒΑΣΙΚΕΣ ΑΡΧΕΣ ΤΗΣ ΔΟΜΗΜΕΝΗΣ ΠΡΟΣΕΓΓΙΣΗΣ	17
1.5 ΜΕΘΟΔΟΛΟΓΙΑ STRADIS	18
1.6 ΜΕΘΟΔΟΣ ΤΟΥ YOURDON (YOURDON SYSTEMS METHOD)	19
1.6.1 ΦΑΣΗ ΠΡΩΤΗ: ΜΕΛΕΤΗ ΣΚΟΠΙΜΟΤΗΤΑΣ (FEASIBILITY STUDY	19
1.6.2 ΦΑΣΗ ΔΕΥΤΕΡΗ: ΒΑΣΙΚΗ ΜΟΝΤΕΛΟΠΟΙΗΣΗ (ESSENTIAL MODELING)	19
1.6.3 ΦΑΣΗ ΤΡΙΤΗ : ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΥΛΟΠΟΙΗΣΗΣ	20
1.7 ΜΕΘΟΔΟΛΟΓΙΑ INFORMATION ENGINEERING (IE)	21
1.8 ΜΕΘΟΔΟΛΟΓΙΑ MESIRE	26
1.9 ΜΕΘΟΔΟΛΟΓΙΑ SSADM	29
1.10 ΜΕΘΟΔΟΛΟΓΙΑ ΠΟΛΛΑΠΛΗΣ ΘΕΩΡΗΣΗΣ MULTIVIEW METHODOLOGY	30
1.11 ΜΕΘΟΔΟΛΟΓΙΑ JACKSON SYSTEM DEVELOPMENT (JSD)	31
1.11.1 Η ΦΑΣΗ ΤΗΣ ΜΟΝΤΕΛΟΠΟΙΗΣΗΣ	32
1.11.2 Η ΦΑΣΗ ΤΟΥ ΔΙΚΤΥΟΥ	33

1.11.3 ΦΑΣΗ ΤΗΣ ΥΛΟΠΟΙΗΣΗΣ	34
1.12 ΑΔΥΝΑΜΙΕΣ ΤΗΣ ΔΟΜΗΜΕΝΗΣ ΑΝΑΛΥΣΗΣ ΚΑΙ ΣΧΕΔΙΑΣΗΣ	35
1.13 ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ ΤΗΣ ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΟΥΣ ΤΕΧΝΟΛΟΓΙΑΣ	37
1.13.1 ΚΛΑΣΗ	38
1.13.2 ΣΤΙΓΜΙΟΤΥΠΟ – ΕΚΔΟΧΗ.....	38
1.13.3 ΠΕΔΙΟ	39
1.13.4 ΜΕΘΟΔΟΣ	39
1.13.5 ΚΕΛΥΦΟΠΟΙΗΣΗ ΑΠΟΚΡΥΨΗ ΠΛΗΡΟΦΟΡΙΩΝ.....	39
1.14 ΣΧΕΣΕΙΣ ΜΕΤΑΞΥ ΚΛΑΣΕΩΝ	39
1.15 ΓΙΑΤΙ Η ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗ ΑΝΑΠΤΥΞΗ ΕΙΝΑΙ ΚΑΛΥΤΕΡΗ ΑΠΟ ΤΗΝ ΔΟΜΗΜΕΝΗ;.....	42
1.16 ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΑΝΑΛΥΣΗ	44
1.17 ΤΙ ΕΙΝΑΙ ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ	47
1.18 ΑΠΟ ΤΙΣ ΠΕΡΙΠΤΩΣΕΙΣ ΧΡΗΣΗΣ ΣΤΟ ΜΟΝΤΕΛΟ ΑΝΑΛΥΣΗΣ.....	49
1.19 ΤΟ ΜΟΝΤΕΛΟ ΑΝΑΛΥΣΗΣ	50
1.20 ΤΙ ΕΙΝΑΙ ΤΑ ΠΑΚΕΤΑ ΑΝΑΛΥΣΗΣ	51
1.21 ΒΗΜΑΤΑ ΣΤΗΝ ΑΝΑΛΥΣΗ	52
1.22 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΑΝΑΛΥΣΗ	53
1.23 ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ UML	54
1.23.1 ΣΤΟΧΟΣ ΤΗΣ UML.....	55
1.23.2 Η ΧΡΗΣΗ ΤΗΣ UML	55
1.23.3 ΔΙΑΦΟΡΕΤΙΚΟΙ ΤΥΠΟΙ ΣΥΣΤΗΜΑΤΩΝ	56
1.24 ΦΑΣΗΣ ΑΝΑΠΤΥΞΗΣ ΕΝΟΣ ΣΥΣΤΗΜΑΤΟΣ.....	57

1.24.1	Η ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ	57
1.24.2	Η ΑΝΑΛΥΣΗ ΤΟΥ ΜΟΝΤΕΛΟΥ	58
1.24.3	Η ΣΧΕΔΙΑΣΗ	58
1.24.4	Ο ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ.....	58
1.24.5	ΈΛΕΓΧΟΣ.....	59
1.25	ΔΟΜΙΚΑ ΣΤΟΙΧΕΙΑ UML.....	59
1.25.1	ΟΙ ΟΨΕΙΣ ΣΤΗ UML.....	59
1.25.2	ΔΙΑΓΡΑΜΜΑΤΑ ΤΗΣ UML.....	62
1.26	ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ – CPU SCHEDULING	66
1.26.1	ΟΥΡΕΣ ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΔΙΕΡΓΑΣΙΩΝ (PROCESS SCHEDULING QUEUES).....	66
1.26.2	ΤΙ ΕΙΝΑΙ ΟΙ ΔΙΑΔΙΚΑΣΙΕΣ-ΔΙΕΡΓΑΣΙΕΣ.....	67
1.26.3	ΠΩΣ ΔΗΜΙΟΥΡΓΕΙΤΑΙ ΜΙΑ ΔΙΑΔΙΚΑΣΙΑ-ΔΙΕΡΓΑΣΙΑ.....	67
1.26.4	ΠΟΙΕΣ ΟΙ ΠΙΘΑΝΕΣ ΚΑΤΑΣΤΑΣΕΙΣ ΜΙΑΣ ΔΙΑΔΙΚΑΣΙΑΣ-ΔΙΕΡΓΑΣΙΑΣ.....	67
1.26.5	ΤΙ ΕΙΝΑΙ Ο ΦΡΑΓΜΟΣ ΕΛΕΧΓΟΥ ΔΙΕΡΓΑΣΙΑΣ (Process Control Block).....	68
1.26.6	ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΤΕΣ (SCHEDULERS).....	68
1.26.7	ΕΝΑΛΛΑΓΗ ΠΕΡΙΒΑΛΛΟΝΤΟΣ ΛΕΙΤΟΥΡΓΙΑΣ (CONTEXT SWITCH).....	69
1.26.8	ΚΡΙΤΗΡΙΑ ΤΟΥ ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ.....	70
1.27	ΕΝΝΟΙΕΣ ΣΧΕΔΙΑΣΜΟΥ ΚΕΝΤΡΙΚΗΣ ΜΟΝΑΔΑΣ ΕΠΕΞΕΡΓΑΣΙΑΣ.....	71
1.27.1	BURSTS (ΕΚΡΗΞΕΙΣ).....	71
1.27.2	PREEMPTIVE SCHEDULING.....	71
1.27.3	DISPATCHER (ΑΠΟΣΤΟΛΕΑΣ).....	71
1.28	ΒΑΣΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ.....	72
1.28.1	ΑΛΓΟΡΙΘΜΟΣ FIRST COME, FIRST SERVED.....	72
1.28.2	ΑΛΓΟΡΙΘΜΟΣ SHORTEST JOB FIRST (SJF).....	73

1.28.3 ΑΛΓΟΡΙΘΜΟΣ ΠΡΟΤΕΡΑΙΟΤΗΤΑΣ.....	74
1.28.4 ΑΛΓΟΡΙΘΜΟΣ ROUND - ROBIN.....	74
1.28.5 ΑΛΓΟΡΙΘΜΟΣ MULTILEVEL QUEUE(ΠΟΛΛΑΠΛΗΣ ΣΤΑΘΜΗΣ ΣΕΙΡΑΣ ΑΝΑΜΟΝΗΣ).....	75
1.28.6 ΠΟΛΥΕΠΙΠΕΔΕΣ ΟΥΡΕΣ ΜΕ ΑΝΑΤΡΟΦΟΔΟΤΗΣΗ (MULTILEVEL FEEDBACK QUEUES).....	76
1.29 ΚΡΙΤΗΡΙΑ ΕΠΙΛΟΓΗΣ ΕΝΟΣ ΑΛΓΟΡΙΘΜΟΥ ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ	76
1.30 ΠΡΟΣΘΕΤΟΙ ΑΛΓΟΡΙΘΜΟΙ ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ	78
1.30.1 ΑΛΓΟΡΙΘΜΟΣ ΣΤΑΘΕΡΗΣ ΠΡΟΤΕΡΑΙΟΤΗΤΑΣ FIXED PRIORITY.....	78
1.30.2 ΑΛΓΟΡΙΘΜΟΣ ΜΟΝΟΤΟΝΙΚΟΥ ΠΟΣΟΣΤΟΥ RATE MONOTONIC.....	78
1.30.3 RATE MONOTONIC WITH DELAYED PREEMPTION (ΑΛΓΟΡΙΘΜΟΣ ΜΟΝΟΤΟΝΙΚΟΥ ΠΟΣΟΣΤΟΥ ΜΕ ΚΑΘΥΣΤΕΡΗΜΕΝΟ ΔΙΚΑΙΩΜΑ ΠΡΟΤΙΜΗΣΗΣ)	79
1.30.4 DEADLINE MONOTONIC (ΑΛΓΟΡΙΘΜΟΣ ΜΟΝΟΤΟΝΙΚΗΣ ΠΡΟΘΕΣΜΙΑΣ).....	79
1.30.5 ΑΛΓΟΡΙΘΜΟΣ DYNAMIC PRIORITY (ΔΥΝΑΜΙΚΗΣ ΠΡΟΤΕΡΑΙΟΤΗΤΑΣ).....	79
1.30.6 ΑΛΓΟΡΙΘΜΟΣ EARLIEST DEADLINE FIRST (ΠΡΟΩΡΗΣ ΠΡΟΘΕΣΜΙΑΣ)	80
1.31 REAL-TIME SCHEDULING (ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΠΡΑΓΜΑΤΙΚΟΥ ΧΡΟΝΟΥ) ...	80
1.31.1 HARD REAL TIME (ΔΥΣΜΕΤΑΒΛΗΤΟΣ ΠΡΑΓΜΑΤΙΚΟΣ ΧΡΟΝΟΣ).....	80
1.31.2 SOFT REAL TIME (ΕΥΜΕΤΑΒΛΗΤΟΣ ΠΡΑΓΜΑΤΙΚΟΣ ΧΡΟΝΟΣ).....	81
1.32 ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΠΡΑΓΜΑΤΙΚΟΥ ΧΡΟΝΟΥ (REAL-TIME SCHEDULING).....	83

ΚΕΦΑΛΑΙΟ 2

ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ Κ ΣΧΕΔΙΑΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

2.1 ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

2.1.1 ΤΙ ΕΙΝΑΙ ΤΟ ΤΗΛΕΦΩΝΙΚΟ ΚΕΝΤΡΟ.....

2.1.2	ΑΝΤΑΓΩΝΙΣΜΟΣ Η ΑΙΤΙΑ ΔΗΜΙΟΥΡΓΙΑΣ ΚΑΛΥΤΕΡΩΝ ΤΗΛΕΦΩΝΙΚΩΝ ΚΕΝΤΡΩΝ	85
2.1.3	ΑΛΛΑΓΕΣ ΣΤΗΝ ΔΙΟΙΚΗΣΗ ΣΕ ΠΡΟΣΩΠΙΚΟ – ΔΙΑΔΙΚΑΣΙΕΣ – ΤΕΧΝΟΛΟΓΙΑ (ΜΟΝΤΕΛΟ RE-ENGINEERING)	86
2.1.4	ΠΟΥ ΟΦΕΙΛΕΤΑΙ Η ΑΝΑΓΚΗ ΥΠΑΡΞΗΣ ΤΗΛΕΦΩΝΙΚΩΝ ΚΕΝΤΡΩΝ ΑΠΟ ΤΙΣ ΕΤΑΙΡΙΕΣ	87
2.1.5	RE-ENGINEERING ΤΟ ΤΗΛΕΦΩΝΙΚΟ ΚΕΝΤΡΟ	88
2.1.6	Η ΑΝΑΠΤΥΞΗ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΗΣΗΣ ΘΑ ΟΔΗΓΗΣΕΙ ΚΑΙ ΣΕ ΝΕΕΣ ΕΥΚΑΙΡΙΕΣ	90
2.1.7	ΤΡΟΠΟΙ ΕΚΜΕΤΑΛΛΕΥΣΗΣ ΤΩΝ ΝΕΩΝ ΕΥΚΑΙΡΙΩΝ	91
2.1.8	ΑΝΑΓΚΑΙΟΤΗΤΑ ΑΝΑΛΥΣΗΣ ΚΟΣΤΟΥΣ – ΩΦΕΛΕΙΑΣ	92
2.1.9	ΑΝΑΛΥΣΗ ROI (RETURN ON INVESTMENT)–ΑΠΟΔΟΣΗ ΤΗΣ ΕΠΕΝΔΥΣΗΣ.....	92
2.2	ΔΟΜΗ ΤΟΥ ΤΗΛΕΦΩΝΙΚΟΥ ΚΕΝΤΡΟΥ	93
2.2.1.	ΕΙΣΑΓΩΓΗ	93
2.2.2	ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΛΕΙΤΟΥΡΓΙΑΣ ΤΟΥ ΤΗΛΕΦΩΝΙΚΟΥ ΚΕΝΤΡΟΥ	94
2.2.3	ΤΟ ΠΡΟΣΩΠΙΚΟ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ	95
2.2.4	ΑΝΑΛΥΣΗ ΚΑΙ ΣΧΕΔΙΑΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ	96
2.2.5	ΑΝΑΛΥΣΗ ΤΩΝ ΑΠΑΙΤΗΣΕΩΝ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ	96
2.2.6	ΑΝΑΛΥΣΗ ΚΑ ΣΧΕΔΙΑΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ	97
2.2.7	ΟΡΙΣΜΟΣ ΚΛΑΣΕΩΝ	98
2.2.8	ΟΙ ΦΑΣΕΙΣ ΛΕΙΤΟΥΡΓΙΑΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ.....	101
	ΚΕΦΑΛΑΙΟ 3	
3	ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΤΟΥ ΤΗΛΕΦΩΝΙΚΟΥ ΚΕΝΤΡΟΥ	105

3.1 ΤΙ ΕΙΝΑΙ ΕΝΑ ΣΥΣΤΗΜΑ ΑΝΑΜΟΝΗΣ	105
3.1.1 ΒΑΣΙΚΗ ΔΟΜΗ ΤΩΝ ΣΥΣΤΗΜΑΤΩΝ ΑΝΑΜΟΝΗΣ	106
3.1.2 ΟΙΚΟΝΟΜΙΚΗ ΠΛΕΥΡΑ ΤΩΝ ΣΥΣΤΗΜΑΤΩΝ ΟΥΡΩΝ ΑΝΑΜΟΝΗΣ	108
3.1.3 ΣΥΜΒΟΛΙΣΜΟΣ ΚΑΙ ΟΡΟΛΟΓΙΑ	109
3.1.4 ΣΧΕΣΕΙΣ ΚΑΤΑΣΤΑΣΗΣ ΙΣΟΡΡΟΠΙΑΣ ΚΑΤΑ LITTLE	110
3.2 ΣΥΝΗΘΗ ΠΡΟΤΥΠΑ ΜΟΝΤΕΛΩΝ ΑΝΑΜΟΝΗΣ ΤΑ ΟΠΟΙΑ ΣΥΝΑΝΤΑΜΕ ΣΤΗΝ ΤΗΛΕΠΙΚΟΙΝΩΝΙΑ	111
3.2.1 ΜΟΝΤΕΛΟ M/M/1	111
3.2.2 ΤΟ ΜΟΝΤΕΛΟ M(N)/M/S(0)	112
3.2.3 ΤΟ ΜΟΝΤΕΛΟ M/M/S	112
3.3 ΕΠΙΛΟΓΗ ΜΟΝΤΕΛΟΥ ΓΙΑ ΤΟ ΤΗΛΕΦΩΝΙΚΟ ΚΕΝΤΡΟ	115
3.4 ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΤΟΥ ΤΗΛΕΦΩΝΙΚΟΥ ΚΕΝΤΡΟΥ	119
3.5 ΕΠΙΛΟΓΗ ΑΛΓΟΡΙΘΜΟΥ	120
ΚΕΦΑΛΑΙΟ 4	
ΣΥΜΠΕΡΑΣΜΑΤΑ	126
ΚΕΦΑΛΑΙΟ 5	
ΒΙΒΛΙΟΓΡΑΦΙΑ	131

Εισαγωγή

Η εποχή μας είναι μια εποχή ανακαλύψεων σε πολλούς τομείς της ανθρώπινης δραστηριότητας. Ειδικά στον τομέα των τηλεπικοινωνιών και των δικτύων υπολογιστών έχουν συντελεστεί μεγάλα άλματα που άλλαξαν ριζικά την ζωή μας. Έννοιες όπως το διαδίκτυο και το ηλεκτρονικό ταχυδρομείο έχουν γίνει κομμάτι της καθημερινής ζωής των ανθρώπων, ακόμα και αυτών που το επάγγελμά τους δεν έχει άμεση σχέση με υπολογιστές ή τηλεπικοινωνίες. Οι υπηρεσίες που παρέχουν οι νέες αυτές τεχνολογίες άνοιξαν νέους ορίζοντες στα παλιά επαγγέλματα και δημιούργησαν πολλά καινούργια. Επιπρόσθετα, παρατηρείται το φαινόμενο της δημιουργίας νέων κλάδων απασχόλησης με κάθε νέα ανακάλυψη. Γίνεται έτσι ιδιαίτερα ενδιαφέρον θέμα η εξέλιξη της τεχνολογίας σε ότι αφορά τις τηλεπικοινωνίες.

Μια από τις σημαντικότερες ερευνητικές προσπάθειες στον τομέα των τηλεπικοινωνιών αφορά την ανάπτυξη των τηλεφωνικών κέντρων. Οι επιχειρήσεις για να μπορέσουν να κερδίσουν την βιωσιμότητα τους θα πρέπει να επικοινωνούν διαρκώς με το αγοραστικό τους κοινό. Η επικοινωνία αυτή δεν μπορεί να επιτευχθεί από τις εταιρίες λειτουργώντας μόνες τους παρόλο που με την ανάπτυξη της τεχνολογίας θα μπορούσε κάποιος να υποθέσει ότι κάτι τέτοιο είναι και υλοποιήσιμο και εύκολο. Η δυσκολία αυτή έγκειται, πρώτον στην δυσκολία συλλογής της πληροφόρησης για τις απαιτήσεις των πελατών και δεύτερον στο ότι τα δεδομένα τα που προκύπτουν από την επαφή με το αγοραστικό κοινό είναι διάσπαρτα πράγμα που δυσχεραίνεται η χρήση τους.

Το πρόβλημα αυτό έρχεται να λύσει τα τηλεφωνικά κέντρα. Ένα τηλεφωνικό κέντρο έχει την δυνατότητα χρησιμοποιώντας τις τελευταίες τεχνολογικές ανακαλύψεις στον τομέα των τηλεπικοινωνιών να επικοινωνεί άμεσα με τους πελάτες και να συλλέγει πολύτιμα στοιχεία για τα χαρακτηριστικά της αγοράς. Έπειτα με αρκετή ευκολία τα δίνει στο προσωπικό της εταιρίας, οι οποίοι μαθαίνουν καλύτερα τους πελάτες της και παίρνουν καλύτερες και ποιοτικότερες αποφάσεις για την παροχή υπηρεσιών.

Βέβαια η ανάπτυξη ενός συστήματος τηλεφωνικού κέντρου κάθε άλλο παρά εύκολο εγχείρημα είναι. Η οργάνωση της επικοινωνίας της πληροφόρησης δεν είναι εύκολο ζήτημα για αυτό τον λόγο παρουσιάζονται και αρκετά προβλήματα κατά την υλοποίησή του. Σκοπός της εργασίας αυτής είναι η παρουσίαση και υλοποίηση ενός

τέτοιου συστήματος. Παρουσιάζεται η οργάνωση του λογισμικού, η ροή της πληροφορίας από την επικοινωνία με τον πελάτη καθώς παρουσιάζονται και σχεδιαστικές λύσεις για την δημιουργία του συστήματος.

Διάρθρωση της εργασίας

Η πτυχιακή εργασία είναι δομημένη σε κεφάλαια ως εξής:

Το πρώτο κεφάλαιο αποτελεί στην ουσία ένα εισαγωγικό κομμάτι στο οποίο παρουσιάζεται το υπόβαθρο στο οποίο θα στηριχτεί η ανάπτυξη του συστήματος. Ειδικότερα γίνεται αναφορά στο τι είναι πληροφοριακό σύστημα, ποιες κατηγορίες υπάρχουν και στις μεθοδολογίες ανάπτυξης και σχεδίασης (δομημένες και αντικειμενοστρεφής). Γίνεται επίσης και μια παρουσίαση της σχεδιαστικής γλώσσας UML βάση της οποίας θα γίνει σχεδιασμός των διεργασιών του συστήματος μας με την χρήση των διαγραμμάτων της. Τέλος, παρουσιάζεται και η έννοια του χρονοπρογραμματισμού που θα μας βοηθήσει να κατανεύσουμε τις διεργασίες του συστήματος του τηλεφωνικού κέντρου στην ΚΜΕ (κεντρική μονάδα επεξεργασίας).

Στο δεύτερο κεφάλαιο γίνεται μια αναφορά για την σπουδαιότητα των τηλεφωνικών κέντρων. Παρουσιάζεται η τωρινή κατάσταση λειτουργίας των τηλεφωνικών κέντρων και ποιες μελλοντικές βελτιώσεις θα πρέπει να γίνουν. Επίσης, περιγράφεται και ο τρόπος με τον οποίο μια επιχείρηση εξαρτάται από τα τις υπηρεσίες των τηλεφωνικών κέντρων. Τέλος, αναφέρεται στην δημιουργία ενός τηλεφωνικού κέντρου καθώς και στην ανάλυση και σχεδίαση του, με την βοήθεια των διαγραμμάτων της UML.

Στο τρίτο κεφάλαιο έχουμε πρώτον την μοντελοποίηση του συστήματος μας. Δηλαδή περιγράφουμε σε ποια κατηγορία μοντέλων ανήκει το σύστημα μας και δεύτερον παρουσιάζεται ποιος από τα είδη των αλγορίθμων χρονοπρογραμματισμού έχει εφαρμογή στο τηλεφωνικό κέντρο

Στο τέταρτο κεφάλαιο παραθέτουμε τα συμπεράσματα που αποκομίσαμε από την μελέτη μας.

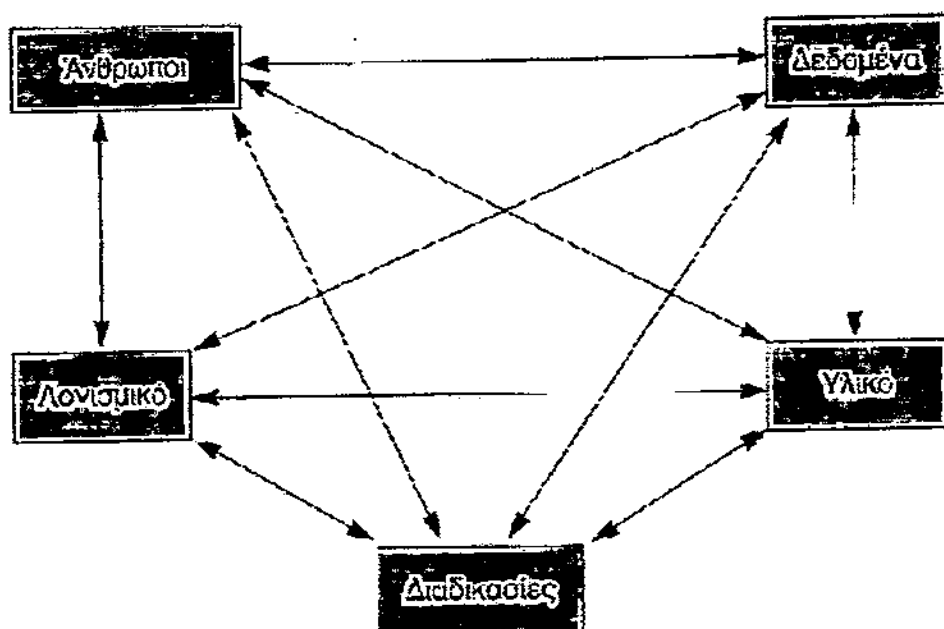
Στο τέλος της εργασίας παρατίθεται η βιβλιογραφία με την βοήθεια της οποίας εκπονήθηκε η πτυχιακή εργασία.

1

1.1 Τι είναι Πληροφοριακό σύστημα

Ως Πληροφοριακό σύστημα ορίζεται ένα σύστημα το οποίο δέχεται πληροφορίες , τις αποθηκεύει , ανακτά , μετασχηματίζει , επεξεργάζεται και διανέμει στους διάφορους χρήστες του οργανισμού , χρησιμοποιώντας υπολογιστές η άλλα μέσα. Το Πληροφοριακό σύστημα είναι ένα οργανωμένο σύνολο πέντε αλληλοεπιδρώντων στοιχείων. Οι 5 συνιστώσες του Πληροφοριακού συστήματος είναι

- άνθρωποι
- διαδικασίες (procedures , methods)
- δεδομένα (data)
- λογισμικό (software)
- υλικός εξοπλισμός (hardware)



Σχήμα 1: Συνιστώσες και αλληλεπιδράσεις του πληροφοριακού συστήματος

Το σχήμα παρουσιάζει τις συνιστώσες του Π.Σ. καθώς και τις αλληλοεπιδράσεις τους . (1)

1.2 Τύποι πληροφοριακών συστημάτων

Οι τύποι των πληροφοριακών συστημάτων είναι οι εξής:

1) **τα πληροφοριακά συστήματα επεξεργασίας συναλλαγών (transaction processing systems)** : το σύστημα αυτής της μορφής παρέχει διαδικασίες για καταγραφή και παραγωγή πληροφοριών σχετικών με κάποιες δοσοληψίες. Τυπικά συστήματα αυτής της κατηγορίας είναι : διαχείριση αποθήκης , επεξεργασία λογιστικών εφαρμογών κτλ. Κοινό χαρακτηριστικό των συστημάτων αυτών είναι ότι συμβάλλουν ουσιαστικά στην βελτίωση της λειτουργίας και της απόδοσης του Οργανισμού η της επιχείρησης με το να παρέχουν γρήγορα και έγκαιρα ακριβείς πληροφορίες .

2) μια δεύτερη κατηγορία είναι **τα πληροφοριακά συστήματα διοίκησης (management information systems)**: είναι συστήματα τα οποία εκτός από την υποστήριξη δοσοληψιών παρέχουν επιπλέον υποστήριξη στις δραστηριότητες της διαχείρισης , ανάληψης και λήψης αποφάσεων από την διοίκηση του Οργανισμού ή της επιχείρησης. Η υποστήριξη παρέχεται με την προετοιμασία σε μορφή αναφορών ή και διαγραμμάτων στις οθόνες τερματικών ή σε εκτυπωμένη μορφή , κατά τακτά χρονικά διαστήματα για χρήση από στελέχη σε διάφορα ιεραρχικά επίπεδα. Κοινό χαρακτηριστικό των πληροφοριών αυτών είναι η αυστηρή δομή.

3) μια τρίτη κατηγορία είναι **τα πληροφοριακά συστήματα υποστήριξης αποφάσεων (decision support systems)**: τα οποία αξιοποιούν δεδομένα και συνήθως μαθηματικά μοντέλα, βοηθούν στην επίλυση των μη δομημένων ή ημιδομημένων προβλημάτων που εμφανίζονται στην κορυφή της διοικητικής πυραμίδας. Είναι ευκολονόητα από ανθρώπους που δεν γνωρίζουν την χρήση Η/Υ.

4) υπάρχουν επίσης **τα εμπειρικά συστήματα ή συστήματα εμπειρογνώμονες (expert systems)**: τα οποία είναι συστήματα που δε στηρίζονται στην συναλλαγή αν και χρησιμοποιούν τη συναλλαγή με δεδομένα. Στηρίζονται στην

προσπάθεια άντλησης γνώσης από την εμπειρία και την δεξιοτεχνία ενός ή περισσότερων εμπειρογνώμων ενός γνωστικού χώρου, με τέτοιο τρόπο, ώστε να είναι δυνατή η εξαγωγή λογικών συμπερασμάτων, όπως ακριβώς κάνει ο ειδικός όταν έχει να αντιμετωπίσει αδόμητα προβλήματα. Ένα ξεχωριστό χαρακτηριστικό των συστημάτων αυτών είναι η ικανότητα τους να αιτιολογήσουν , αν τους ζητηθεί, την πορεία που ακολούθησαν προκειμένου να φτάσουν στην απάντηση.

5) Τα τελευταία χρόνια έχει κάνει την εμφάνιση του και μια άλλη κατηγορία Π.Σ., **αυτή που επεξεργάζεται δεδομένα αποθηκευμένα σε διάφορα μέσα (ήχος, κείμενο, φωνή, video). Πρόκειται για τα πληροφοριακά συστήματα πολυμέσων ΠΣΠ (multimedia information systems):** για παράδειγμα τα ιατρικά δεδομένα ενός ασθενούς μπορεί να είναι αλφαριθμητικά, εικόνες , ομιλία, γραφήματα, video. Όλα αυτά μπορούν να υποστούν επεξεργασία τόσο τοπικά όσο και σε ένα ευρύτερο δίκτυο. Δυστυχώς η ανάπτυξη των πληροφοριακών συστημάτων πολυμέσων αντί να είναι ωθούμενη από τις εφαρμογές , ωθείται από την τεχνολογία. Άμεσο αποτέλεσμα είναι να μην υπάρχει ξεχωριστή μεθοδολογία ανάπτυξης ΠΣΠ ανεξάρτητων από τεχνολογία.

6) Τέλος μια ξεχωριστή κατηγορία Π.Σ είναι **τα συστήματα πραγματικού χρόνου (real time systems):** πρόκειται για συστήματα τα οποία είναι ικανά να λαμβάνουν συνεχώς μεταβαλλόμενα δεδομένα από εξωτερικές πηγές και να επεξεργάζονται τα δεδομένα τόσο γρήγορα ώστε να μπορούν να επηρεάσουν τις πηγές των δεδομένων.

Στα συστήματα αυτά ο παράγοντας χρόνος παίζει μεγάλη σημασία, μάλιστα κατά την ανάπτυξη τους τίθεται ως απαίτηση όπως οι επεξεργασίες του συστήματος γίνονται στον ελάχιστο δυνατό χρόνο. Τόσο το υλικό όσο και το λογισμικό πρέπει να σχεδιαστούν με τέτοιο τρόπο ώστε να ικανοποιούν την απαίτηση αυτή. Είναι γεγονός πως η σχεδίαση ενός πληροφοριακού συστήματος έχει γίνει μια πολύπλοκη διαδικασία και αυτό γιατί έχουν αυξηθεί στο μέγιστο οι απαιτήσεις για μεγαλύτερη αποδοτικότητα και καλύτερη λειτουργικότητα. Η αύξηση των απαιτήσεων αυτών οφείλεται στο γεγονός ο ρόλος των πληροφοριακών συστημάτων έχει αλλάξει. Την περασμένη δεκαετία δινόταν έμφαση στην υποστήριξη των βασικών λειτουργιών της επιχείρησης , κυρίως με την επεξεργασία αριθμητικών δεδομένων (data processing

system). Η τάση σήμερα είναι για υποστήριξη ολόκληρης της επιχείρησης, όχι μόνο in-house αλλά και πέραν των φυσικών γεωγραφικών ορίων. Οι απομακρυσμένοι πελάτες και προμηθευτές θα επικοινωνούν με οποιονδήποτε στην ιεραρχική πυραμίδα του οργανισμού. Ήδη ο Παγκόσμιος Οργανισμός Εμπορίου αποφάσισε τη δημιουργία μέσω του internet, μιας ελεύθερης ζώνης ηλεκτρονικού εμπορίου. Αναμένεται ότι οι συναλλαγές μέσω Internet θα αυξηθούν με ταχύ ρυθμό, όπως αυξήθηκε ραγδαία η αυτόματη τηλεπικοινωνία. Παράλληλα βέβαια με τα διεθνή δίκτυα έχουμε και τα τοπικά στις διαφορές μορφές τους. Έχουμε μια πρώτη σημαντική αλλαγή: τα μελλοντικά Π.Σ θα απαιτούν πολλά από τις επικοινωνίες πράγμα το οποίο απαιτεί αρκετή υπολογιστική ισχύ. Μια δεύτερη σημαντική αλλαγή είναι στη φύση των δεδομένων που επεξεργάζεται το πληροφοριακό σύστημα. Μέχρι σήμερα τα Π.Σ επεξεργάζονταν δεδομένα ενός είδους (αριθμητικά) ενώ ήδη αρχίζουν να δημιουργούνται πληροφοριακά συστήματα πολλαπλών μέσων όπου τα δεδομένα είναι διαφόρων τύπων: κείμενο(text), γραφήματα(graphics), ήχος(sound), εικόνα(image) ,videos και φυσικά αριθμοί. Τέλος μια τρίτη αλλαγή είναι στην υποστήριξη διαφόρων επιπέδων στην επιχείρηση. Οι διάφορες αρχιτεκτονικές και κυρίως η ανάπτυξη των τοπικών δικτύων έχουν ως αποτέλεσμα να υπάρχει τάση για δημιουργία κατακεντρωμένων Π.Σ.

Με γνώμονα τις παραπάνω αλλαγές στις απαιτήσεις τις οποίες θα πρέπει να εκπληρώνει το πληροφοριακό σύστημα ο χρήστης για την ανάπτυξη του Π.Σ εκτός από τα προσόντα του , κρίνεται αναγκαίο η χρήση μιας μεθοδολογίας , δηλαδή ενός οργανωμένου συνόλου από διαδικασίες , τεχνικές , εργαλεία και μέσα τεκμηρίωσης.

Με τη βοήθεια της μεθοδολογίας είναι δυνατή η αντιμετώπιση των προβλημάτων και η επίτευξη των εξής στόχων :

- 1) αναγνώριση , αξιολόγηση και καταγραφή με ακρίβεια , τόσο των τρεχουσών , όσο και των λανθανουσών (potential) απαιτήσεων των χρηστών.
- 2) Παροχή μιας συστηματικής μεθόδου ανάπτυξης ώστε ο έλεγχος της προόδου να είναι απρόσκοπτος.
- 3) Να εξασφαλίζεται ότι το σύστημα τελικά θα ικανοποιεί όλους όσους συναλλάσσονται με αυτό.

4) Ολοκλήρωση του έργου μέσα στα χρονικά περιθώρια που έχουν εκτιμηθεί, χωρίς υπερβάσεις στο κόστος του έργου.

5) Έγκαιρος εντοπισμός των αλλαγών που απαιτούνται να γίνουν. Η προσθήκη των αλλαγών να γίνεται όσο γίνεται πιο απρόσκοπτα και με το μικρότερο κόστος.

6) Ύπαρξη καλής τεκμηρίωσης ώστε να είναι εύκολη η συντήρηση του συστήματος που θα προκύψει.

7) Να λαμβάνει υπόψη τις κοινωνικές, οργανωτικές και τεχνολογικές παραμέτρους της ανάπτυξης.

1.3 Μεθοδολογίες και τεχνικές σχεδίασης Πληροφοριακού Συστήματος

Υπάρχουν οι εξής μεγάλες κατηγορίες μεθοδολογιών οι οποίες χρησιμοποιούνται για τον σχεδιασμό και την ανάπτυξη ενός πληροφοριακού συστήματος

A) οι δομημένες μεθοδολογίες

B) οι αντικειμενοστραφείς μεθοδολογίες

1.4 Βασικές αρχές της δομημένης προσέγγισης

Τέσσερις είναι οι βασικές αρχές πάνω στις οποίες στηρίζονται οι δομημένες μεθοδολογίες.

A) αρχή της αφάιρεσης (abstraction)

Σύμφωνα με την αρχή αυτή γίνεται απλοποίηση των πραγμάτων , διατηρώντας αυτά που θεωρούμε περισσότερο σημαντικά. Με τη βοήθεια της αφάιρεσης μπορούμε να βλέπουμε πράγματα σε διαφορετικά επίπεδα. Κάθε επίπεδο έχει το δικό του βαθμό αφάιρεσης και αποτελείται από λειτουργίες. Λειτουργίες που ανήκουν στο ίδιο επίπεδο ή σε διαφορετικά επίπεδα μπορούν να επικοινωνούν.

Υπάρχουν δυο τρόποι επικοινωνίας :

1) τα κατώτερα επίπεδα δεν γνωρίζουν και δεν έχουν προσπέλαση στα ανώτερα. Αντίθετα τα ανώτερα επίπεδα μπορούν να προσκαλέσουν τα κατώτερα να εκτελέσουν κάποιο έργο.

2) Κάθε επίπεδο έχει τα δικά του δεδομένα, τα οποία χρησιμοποιούνται αποκλειστικά από τις λειτουργίες του επιπέδου αυτού και δεν μπορούν να χρησιμοποιηθούν από λειτουργίες άλλου επιπέδου. Όμως συγκεκριμένα δεδομένα μπορούν να περνούν από το ένα επίπεδο στο άλλο. (κάθε επίπεδο μπορεί να έχει πρόσβαση σε κοινή βάση δεδομένο)

B) Η αρχή της αυστηρής τυπικότητας (formality):

Η μεθοδολογία πρέπει να είναι αυστηρή, συστηματική, βήμα προς βήμα διαδικασία. Έτσι διευκολύνεται η παρουσίαση των ιδεών σε υπολογιστική (αλγοριθμική μορφή), άρα και η απρόσκοπτη αυτοματοποίηση.

Γ) **Η αρχή του διαιρεί και βασίλευε (divide and conquer):** Η κλασσική πρακτική του τεμαχισμού σε μικρότερα, πιο προσιτά προβλήματα.

Δ) **Η δημιουργία ιεραρχικής δομής (hierachical ordering):** Η οργάνωση των επιμέρους τμημάτων να σχηματίζει ένα δένδρο με ιεραρχική δομή.

1.5 Μεθοδολογία STRADIS

Η μεθοδολογία Stradis (structure analysis design and implementation of information systems) στηρίζεται στη δομημένη προσέγγιση και χρησιμοποιεί δομημένες διαγραμματικές τεχνικές. Αποτελείται από διάφορες δραστηριότητες που μπορούν να ταξινομηθούν στις εξής κατηγορίες :

1. δημιουργία προκαταρκτικής μελέτης (*initial study*)
2. δημιουργία λεπτομερειακής μελέτης (*detail study*)
3. εκτίμηση απαιτήσεων χρηστών
4. εκτίμηση φυσικής σχεδίασης
5. σταθεροποίηση απαιτήσεων
6. σχεδιασμός αυτοματοποιημένων δραστηριοτήτων, σχεδιασμός των δραστηριοτήτων που θα εκτελούν οι άνθρωποι, εκτίμηση σχεδίου υλοποίησης και σχεδιασμός δεδομένων.
7. δημιουργία σχεδίων ελέγχου συστήματος
8. δημιουργία σχεδίων εγκατάστασης και έλεγχοι αυτής

Η μεθοδολογία αυτή σε σχέση με το κύκλο ζωής του Π.Σ καλύπτει τις εξής φάσεις :

- 1) ανάλυση

- 2) σχεδιασμό
- 3) έλεγχο
- 4) υλοποίηση
- 5) συντήρηση

Η ανάθεση των εργασιών σε κάθε δραστηριότητα στηρίζεται περισσότερο στην έννοια των ρόλων, παρά στην αυστηρή περιγραφή των προσόντων των εμπλεκόμενων προσώπων. Αυτό σε συνδυασμό με την αυστηρή περιγραφή των επιμέρους δραστηριοτήτων, διευκολύνει το διοικητικό προγραμματισμό του έργου της ανάπτυξης.

1.6 Μέθοδος του Yourdon (Yourdon Systems Method)

Η μέθοδος του Yourdon είναι πανόμοια με την STRADIS. Στηρίζεται στη δομημένη μεθοδολογία και η πρόσφατη έκδοση της χρησιμοποιεί το διαμερισμό των γεγονότων (event partitioning). Περιλαμβάνει τρεις μεγάλες φάσεις :

1.6.1 Φάση πρώτη: μελέτη σκοπιμότητας (Feasibility Study)

Εστιάζει στο υπάρχον σύστημα, το περιβάλλον του και τα προβλήματα που απασχολούν το σύστημα. Έμφαση δίνεται στο τι κάνει το σύστημα και όχι στο πως λειτουργεί. Χρησιμοποιούνται δομημένες διαγραμματικές τεχνικές και κυρίως διαγράμματα ροής δεδομένων και διαγράμματα οντοτήτων- συσχετίσεων.

1.6.2 Φάση δεύτερη: βασική μοντελοποίηση (Essential Modeling)

Είναι η σπουδαιότερη από τις τρεις φάσεις και αφορά την μοντελοποίηση τόσο του ευρύτερου συστήματος όσο και του κυρίως συστήματος. Το βασικό μοντέλο του συστήματος είναι ένα μοντέλο που περιγράφει τι πρέπει να κάνει το σύστημα για να ικανοποιεί τις απαιτήσεις του χρήστη.

Για το λόγο αυτό δημιουργεί διαδοχικά τα εξής μοντέλα:

1. τρέχον φυσικό μοντέλο

2. τρέχον λογικό μοντέλο
3. νέο λογικό μοντέλο
4. νέο φυσικό μοντέλο

Τα παραπάνω μοντέλα περιέχονται στις παρακάτω δυο ευρύτερες κατηγορίες μοντέλων , που μαζί αποτελούν το βασικό μοντέλο.

A) Το μοντέλο περιβάλλοντος το οποίο οριοθετεί το σύστημα σε σχέση με το περιβάλλον και το ευρύτερο σύστημα στο οποίο ανήκει. Το μοντέλο αυτό αποτελείται από τους σκοπούς και στόχους του συστήματος, το διαγράμματα πλαίσιο και των πίνακα γεγονότων.

Το διάγραμμα πλαίσιο απεικονίζει τις βασικές εισροές και εκροές του συστήματος, τις πηγές των εισροών και τις καταλήξεις των εκροών. Συνοδεύεται από ένα λεξικό δεδομένων στο οποίο αναλύονται οι διάφοροι όροι του διαγράμματος. Ο πίνακας γεγονότων περιέχει όλα τα γεγονότα στα οποία αντιδρά το σύστημα.

B) Το μοντέλο συμπεριφοράς το οποίο παρουσιάζει την εσωτερική συμπεριφορά του συστήματος από τη στιγμή που δέχεται τις εισροές μέχρι την παραγωγή των εκροών. Αποτελείται από διαγράμματα ροής δεδομένων, διαγράμματα οντοτήτων-συσχετίσεων και αν χρειαστεί διάγραμμα μετάβασης – κατάσταση. Τα διαγράμματα ροής δεδομένων είναι συνήθως πολλαπλών επιπέδων και συνοδεύονται από πίνακες αποφάσεων δομημένες περιγραφές των επεξεργασιών, λεξικά δεδομένων και περιγραφή των περιεχομένων των αρχείων.

1.6.3 Φάση τρίτη : μοντελοποίηση υλοποίησης

Η φάση αυτή ασχολείται με το σχεδιασμό του συστήματος. Το νέο φυσικό μοντέλο συμπληρώνεται, βελτιώνεται και επικυρώνεται. Δημιουργούνται προδιαγραφές σχεδιασμού λογισμικού, βάσεως δεδομένων, απαιτήσεων για δίκτυα και γενικά ότι απαιτείται για την απρόσκοπτη υλοποίηση.

1.7 Μυθολογία Information engineering (IE)

«Μηχανική της πληροφορίας (Information engineering (IE)) είναι η εφαρμογή ολοκληρωμένου συνόλου αλληλοεξαρτημένων, αυστηρών τεχνικών για το προγραμματισμό (planning) , την ανάλυση, το σχεδιασμό (desing) και την κατασκευή των πληροφοριακών συστημάτων που καλύπτουν είτε ολόκληρη την επιχείρηση είτε σημαντικούς τομείς αυτής».

Όπως προκύπτει από τα παραπάνω η IE αντιμετωπίζεται όχι ως μια μεθοδολογία αλλά ως ένα ευρύ, δομημένο, πλαίσιο μεθοδολογιών, στα πλαίσια του οποίου μπορούν να χρησιμοποιηθούν και συνδυαστούν πλήθος από τεχνικές και μεθοδολογίες.

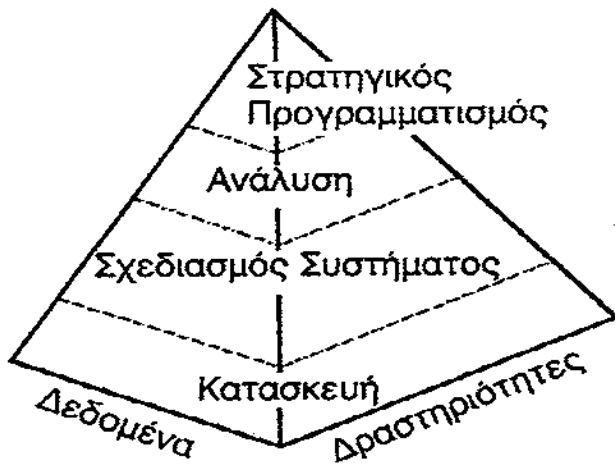
Η φιλοσοφική θέση της IE μπορεί να συνοψιστεί στα εξής:

1. τα δεδομένα αποτελούν το πυρήνα ενός Π.Σ.

2. τα δεδομένα και οι μεταξύ τους σχέσεις είναι σταθερές στο επίπεδο του οργανισμού.

3. οι επεξεργασίες πάνω στα δεδομένα αλλάζουν ανάλογα με το είδος των πληροφοριών που θέλουμε.

Στο παρακάτω σχήμα παριστάνεται η πυραμίδα ανάπτυξης ενός Π.Σ κατά την IE. Παρατηρούμε ότι αποτελείται από τέσσερα επίπεδα. Στην κορυφή είναι ο σχεδιασμός της στρατηγικής πληροφοριακής ολόκληρης της επιχείρησης, ακολουθεί προς τα κάτω το επίπεδο της ανάλυσης όπου δημιουργούνται τα μοντέλα δεδομένων και επεξεργασιών που απαιτούνται για να λειτουργεί η επιχείρηση. Η ανάλυση προσδιορίζει τις απαιτήσεις. Στο επόμενο επίπεδο γίνεται ο σχεδιασμός του συστήματος, το οποίο κατασκευάζεται στο τελευταίο επίπεδο, τη βάση. Η αριστερή βάση εκπροσωπεί τα δεδομένα ενώ η δεξιά τις δραστηριότητες. Και τα δυο υφίστανται καθοδική επεξεργασία, δηλαδή από τη θεώρηση της διοίκησης του οργανισμού στη βάση, που γίνεται η υλοποίηση.(2)



Σχήμα 2: Πυραμίδα ανάπτυξης πληροφοριακού συστήματος κατά την ΙΕ.

Τα στάδια της ΙΕ είναι τα ακόλουθα:

Στάδιο 1: σχεδιασμός πληροφοριακής στρατηγικής

- ✓ επισκόπηση της επιχείρησης
- ✓ περιγραφή των :α) αντικειμενικών στόχων, β) πληροφοριακών αναγκών

και γ)

δραστηριοτήτων και τρεχόντων συστημάτων.

- ✓ προσχεδίασμα των οντοτήτων (data modeling) και των λειτουργιών (function modeling)

α) Ορισμός της Αρχιτεκτονικής Πληροφοριακής Στρατηγικής (Α.Σ.Π)

β) καθορισμός τεχνικών προδιαγραφών

γ) καθορισμός της οργάνωσης και της λειτουργίας των πληροφοριακών συστημάτων

- ✓ αποσύνδεση του ΑΣΠ σε επιχειρησιακή περιοχή και παραγωγή του σχεδίου πληροφοριακής στρατηγικής

Τεχνικές επίτευξης : κυρίως διαγράμματα ροής δεδομένων.

Στάδιο 2: Ανάλυση επιχειρησιακής Περιοχής

- ✓ Ανάλυση οντοτήτων και λειτουργιών. Παραγωγή αναλυτικών μοντέλων δεδομένων και επεξεργασιών.
- ✓ Εξέταση των δεδομένων που χρησιμοποιούνται από κάθε επεξεργασία και αντίστροφα. Γίνεται έλεγχος συσχέτισης των δυο μοντέλων.
- ✓ Δημιουργία λογικού μοντέλου επεξεργασιών
- ✓ Επικύρωση μοντέλων
- ✓ Επιλογή των επεξεργασιών που θα αυτοματοποιηθούν
- ✓ Σύνταξη σχεδίου εργασιών επόμενου σταδίου

Τεχνικές : Διαγράμματα ροής δεδομένων

Διαγράμματα οντοτήτων - συσχετίσεων

Διαγράμματα ενεργειών

Διαγράμματα μετάβασης – κατάστασης

Στάδιο 3 : σχεδιασμός επιχειρησιακού συστήματος

- ✓ Σχεδιασμός της βασικής μορφής βάσεων δεδομένων του συστήματος
- ✓ Δημιουργία διαγραμμάτων ροής δεδομένων με τις διαδικασίες που αντιστοιχούν στις λειτουργίες της επιχείρησης
- ✓ Για κάθε διαδικασία προσδιορίζονται οι:
 - Είσοδοι
 - Έξοδοι
 - Διάλογοι με τις υπόλοιπες διαδικασίες
 - Διαγράμματα ενεργειών
- ✓ Έλεγχος των αποτελεσμάτων – Επικύρωση
- ✓ Σύνταξη σχεδίου για το επόμενο στάδιο συμπεριλαμβανομένου του πιθανού διαχωρισμού σε μικρότερες μονάδες προς υλοποίηση
- ✓ Εκτίμηση των απαιτούμενων πόρων
- ✓ Το τελικό προϊόν του σταδίου είναι οι προδιαγραφές σχεδιασμού επιχειρησιακού συστήματος στο οποίο
 - I. Για κάθε επεξεργασία του συστήματος έχουμε τεκμηρίωση παγιωμένη για τη ροή δεδομένων και ενέργειες χρηστών
 - II. Για κάθε διαδικασία που θα αυτοματοποιηθεί έχουμε:

- Μια παγιωμένη και επιβεβαιωμένη έκδοση των αποτελεσμάτων της ανάλυσης επιχειρησιακής περιοχής
- Σχέδιο διαλόγου, οθόνες, αναφορές και διεπαφές των χρηστών
- Παραγωγή των διαγραμμάτων ενεργειών

Τεχνικές : Διαγράμματα ροής δεδομένων

Διαγράμματα οντοτήτων - συσχετίσεων

Διαγράμματα μετάβασης – κατάστασης

Διαγράμματα ενεργειών

Διαγράμματα δομής δεδομένων

Στάδιο 4: τεχνικός σχεδιασμός συστήματος

Οι στόχοι του σταδίου είναι:

- Ο ορισμός απόδοσης των υπολογιστικών συστημάτων που θα υποστηρίξουν τις επεξεργασίες που θα αυτοματοποιηθούν, και
- η εκτίμηση χρόνου και κόστους του συστήματος για κατασκευή νέου και μετάβασης από το παλαιό στο νέο

✓ Σχεδιασμός των τμημάτων του συστήματος που εξαρτώνται από υπολογιστικά περιβάλλοντα.

✓ Προσδιορισμός λογικών δομών δεδομένων, αρχείων και βάσης δεδομένων, προγραμμάτων, λειτουργικών διαδικασιών και διεπαφών

- ✓ Παραγωγή των προδιαγραφών τεχνικού σχεδιασμού που περιέχει:
 - τα σχέδια των βάσεων δεδομένων
 - λοιπών αρχείων
 - σχέδια λογισμικού εφαρμογών
 - τεχνικά πρότυπα λογισμικού – υλικού
 - τους διάφορους ελέγχους
 - το περιεχόμενο των δυο επόμενων σταδίων και εκτίμηση των πόρων, του κόστους και του χρόνου που απαιτούν

Τεχνικές : Διαγράμματα ροής δεδομένων

Διαγράμματα οντοτήτων - συσχετίσεων

Διαγράμματα μετάβασης – κατάστασης



- Διαγράμματα ενεργειών
- Διαγράμματα δομής δεδομένων
- Διαγράμματα δομής προγράμματος

Στάδιο 5: κατασκευή

- ✓ Για κάθε μονάδα γίνεται
 - προμήθεια και εγκατάσταση εξοπλισμού
 - δημιουργία αρχείων, βάσης δεδομένων
 - κωδικοποίηση και έλεγχος προγραμμάτων
- ✓ Σκοπός είναι η ανάπτυξη του εφαρμογών που θα ανταποκρίνεται στις

προδιαγραφές του σταδίου τεχνικός σχεδιασμός

- ✓ Κριτήρια αποδοχής είναι τα εξής :
 - λειτουργικότητα των συστημάτων
 - έλεγχος λογισμικού
 - ελέγχων λειτουργικότητας
 - ελέγχων διεπαφών
- ✓ επικύρωση συστήματος από χρήστη

Στάδιο 6: μετάβαση

- ✓ Σταδιακή αντικατάσταση του παλιού με το νέο
- ✓ Το σύστημα γίνεται αποδεκτό όταν λειτουργήσει σωστά κάποιο χρονικό

διάστημα έχοντας ως κριτήρια :

- την αποδοτικότητα
- το βαθμό λαθών και
- την ευχρηστία

Συνοψίζοντας τα χαρακτηριστικά της ΙΕ είναι :

- ✓ βασίζεται τόσο στα δεδομένα όσο και στις δραστηριότητες
- ✓ καταρτίζει στρατηγικό πλάνο της εταιρίας για την εξυπηρέτηση των

επιχειρηματικών αναγκών

- ✓ παρέχει δυνατότητα αυτοματοποίησης των διαδικασιών
- ✓ ενισχύει τη συμμετοχή του χρήστη και της διοίκησης
- ✓ παράγει καλή τεκμηρίωση



- ✓ ενισχύει την επαλήθευση
 - ✓ εξασφαλίζει το συντονισμό των projects που πρόκειται να αναπτυχθούν
 - ✓ στηρίζεται στη χρήση δοκιμασμένων τεχνικών (διαγράμματα)
- a) καθοδική ανάλυση (top – down)
 - b) ανάλυση ιεραρχική λειτουργιών
 - c) ανάλυση οντοτήτων
 - d) ανάλυση ενεργειών
 - e) Διαγράμματα δομής δεδομένων
 - f) Διαγράμματα δομής προγράμματος
 - g) Διαγράμματα μετάβασης – κατάστασης
 - h) Διαγράμματα ροής δεδομένων
 - i) Διαγραμματική ανάλυση δεδομένων

1.8 Μεθοδολογία MESIRE

Είναι η πιο ευρέως διαδεδομένη από τις υπόλοιπες μεθοδολογίες. Η διάφορα της από τις υπόλοιπες έγκειται στο ότι αντιμετωπίζει τα δεδομένα και την επεξεργασία τους με τον ίδιο λεπτομερειακό τρόπο. Έτσι όπως ακριβώς η μελέτη των δεδομένων ακολουθεί τρία στάδια (νοητικό, λογικό, φυσικό) τα ίδια στάδια ακολουθεί και η μελέτη της επεξεργασίας. Επίσης, σε αντιδιαστολή με την κλασσική μεθοδολογία που αναπτύσσει τις εφαρμογές με βάση τις ανάγκες των χρηστών, η μεθοδολογία MESIRE βασίζεται στη διαπίστωση ότι τα δεδομένα που τυγχάνουν επεξεργασίας από την επιχείρηση είναι ως ένα βαθμό σταθερά στο βαθμό που ανταποκρίνονται στο αντικείμενο ασχολίας του οργανισμού.

Η MESIRE θεωρεί ότι μια επιχείρηση αποτελείται από τρία υποσυστήματα: το φυσικό σύστημα παραγωγής, το σύστημα διοίκησης - λήψης αποφάσεων και το πληροφοριακό σύστημα.

Η MESIRE στην ουσία αποτελείται από τρεις κύκλους σε αλληλοεπίδραση τον κύκλο ζωής, που καλύπτει χρονολογικά όλο το έργο της ανάπτυξης, τον κύκλο λήψης απόφασης, που σχετίζεται με τους μηχανισμούς λήψης απόφασης και τέλος τον αφαιρετικό κύκλο, ο οποίος εστιάζει στα μοντέλα δεδομένων και επεξεργασιών.

Ο κύκλος ζωής αποτελείται από τα ακόλουθα στάδια:

- ✓ μακροχρόνιος προγραμματισμός
- ✓ προκαταρκτική μελέτη
- ✓ αναλυτική μελέτη
- ✓ υλοποίηση και εγκατάσταση
- ✓ παραγωγή και συντήρηση

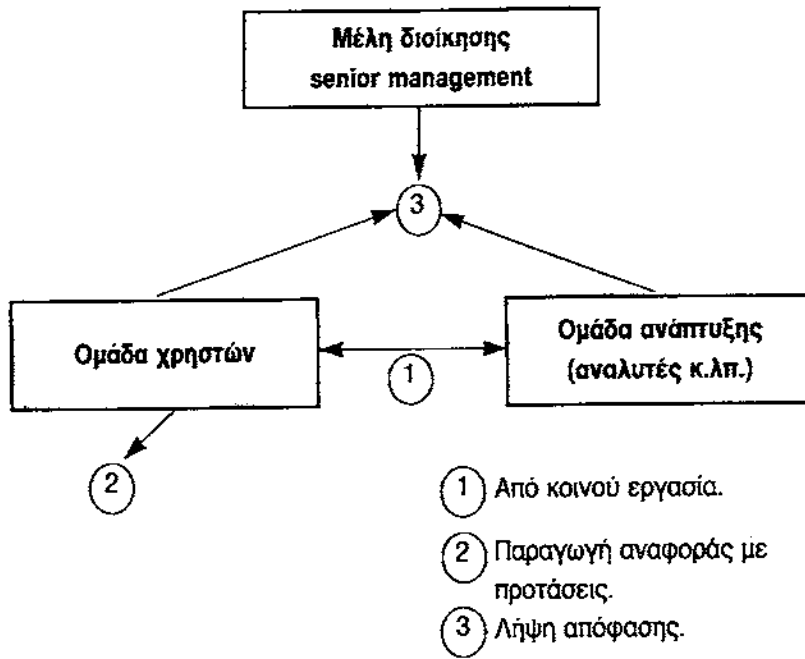
Κάθε στάδιο αναλύεται παραπέρα σε φάσεις. Στη συνέχεια κάθε φάση διασπάται σε εργασίες (tasks). Κάθε στάδιο, φάση και εργασία παράγει συγκεκριμένα παραδοτέα προϊόντα τα οποία θα πρέπει να γίνουν επισήμως αποδεκτά από τους χρήστες.

Στον κύκλο λήψης απόφασης τις αποφάσεις τις λαμβάνουν από κοινού η διοίκηση, οι χρήστες και οι τεχνικοί. Είναι ενδιαφέρον ότι στον κύκλο αυτό προσδιορίζεται μια ιεραρχία για τις αποφάσεις που θα ληφθούν και εντοπίζονται οι «actors» οι οποίοι μπορεί να είναι άνθρωποι αλλά είναι δυνατόν να είναι και υπολογιστικά συστήματα όταν το πρόβλημα είναι τεχνικό.

Οι αποφάσεις αφορούν κυρίως θέματα όπως:

- a) καθορισμό στόχων
- b) διοικητικής μορφής
- c) οικονομικών προβλημάτων
- d) τεχνικών επιλογών (λογισμικού, υλικού, δικτύων κ λ π)
- e) οργανωτικών επιλογών
- f) επιλογή μορφής επεξεργασίας

Το παρακάτω σχήμα δείχνει τον τρόπο τον οποίο συνιστά η μεθοδολογία για την λήψη αποφάσεων.(2)



Σχήμα 3:Μεθοδολογία λήψης αποφάσεων (Μεθοδολογία MESIRE)

Ο αφαιρετικός κύκλος είναι πολύ σημαντικός. Στηρίζεται στην αφαίρεση, δηλαδή, στην παράλειψη δευτερευόντων κυρίως γνωρισμάτων και την εστίαση σε αυτά που θεωρούμε ουσιώδη.

Ο κύκλος αποτελείται από τρία επίπεδα:

- A) νοητικό επίπεδο
- B) λογικό / οργανωτικό επίπεδο και
- Γ) φυσικό επίπεδο

Στο νοητικό επίπεδο η MESIRE προτείνει δυο μοντέλα: A) το νοητικό μοντέλο δεδομένων ή ΝΜΔ και B) το νοητικό μοντέλο επεξεργασίας ή ΝΜΕ

Στο λειτουργικό / οργανωτικό επίπεδο η MESIRE τα εξής δυο μοντέλα: A) το λογικό μοντέλο δεδομένων ή ΛΜΔ και B) το λογικό μοντέλο επεξεργασίας ή ΛΜΕ.

Ομοίως με τα παραπάνω έχουμε και το φυσικό μοντέλο δεδομένων (ΦΜΔ) και το φυσικό μοντέλο επεξεργασιών (ΦΜΕ)

Συνοπτικά η MESIRE χρησιμοποιεί τα εξής μοντέλα όπως δείχνει και ο παρακάτω πίνακας.(2)

	Δεδομένα	Επεξεργασία
Νοητικό	ΝΜΔ	ΝΜΕ
Λογικό / Οργανωτικό	ΛΜΔ	ΛΜΕ
Φυσικό / Λειτουργικό	ΦΜΔ	ΦΜΕ

Σχήμα 4: Τα μοντέλα της MESIRE

Γενικότερες παρατηρήσεις για την MESIRE

1. Η MESIRE κάνει αυστηρή διάκριση ανάμεσα στα δεδομένα και στην επεξεργασία τους.
2. Διακρίνει δυο όψεις ενός συστήματος: τη στατική και τη δυναμική
3. Μπορούν να χρησιμοποιηθούν διαγραμματικές τεχνικές σε όλα τα παραπάνω, εκτός από τα φυσικά μοντέλα δεδομένων και επεξεργασιών.

1.9 Μεθοδολογία SSADM

Η μεθοδολογία Structure System Analysis and Desing Method δημιουργήθηκε από την LBMS (Learmonth and Burchett Management Systems) και την CCTA (Central Computer and Telecommunication).

Τα βασικά της χαρακτηριστικά είναι :

- 1) Αποτελείται από συγκεκριμένα βήματα
- 2) Σπάζει το έργο σε επιμέρους τμήματα φάσεις – στάδια – βήματα – εργασίες
- 3) Ο αναλυτής εκτελεί μια εργασία κάθε φορά
- 4) Εύκολα προσαρμόζεται στα μέτρα κάθε περίπτωσης
- 5) Καθοδηγείται από την ανάλυση των δεδομένων
- 6) Γίνεται επαλήθευση με συχνές επαναλήψεις, συγκριτικούς ελέγχους και συμμετοχή των χρηστών
- 7) Επιτρέπει να γίνουν τροποποιήσεις παντού

8) Μπορεί να αναλυθεί το ίδιο σημείο από διαφορετικές ομάδες και από διαφορετικές οπτικές γωνίες

Περιλαμβάνει τρεις φάσεις:

Φάση 1: σκοπιμότητα στην οποία γίνεται ορισμός του προβλήματος και καθορισμός του έργου

Φάση 2 : ανάλυση του συστήματος στην οποία γίνεται η ανάλυση των λειτουργιών του συστήματος και των υπάρχοντων προβλημάτων. Γίνεται λεπτομερής καθορισμός των απαιτήσεων και τέλος η επιλογή τεχνικών εναλλακτικών λύσεων.

Φάση 3: σχεδιασμός του συστήματος στον οποίο γίνεται ο σχεδιασμός των δεδομένων, των επεξεργασιών και ο φυσικός σχεδιασμός.

Η SSADM χρησιμοποιεί διάφορες διαγραμματικές τεχνικές. Οι τεχνικές παράγουν ενότητες και αλληλοσχετίζονται. Μερικές από τις τεχνικές αυτές είναι:

- Διαγράμματα ροής δεδομένων
- Πίνακες και δένδρα αποφάσεων
- Τεχνικές λογικής δόμησης δεδομένων
- Εναλλακτικές λύσεις χρήστη
- Ιστορία ζωής οντότητας
- Σχεδιασμός διαλόγων
- Σχεσιακή ανάλυση δεδομένων
- Διαγράμματα δομής προγράμματος

1.10 Μεθοδολογία πολλαπλής θεώρησης Multiview methodology

Η Multiview methodology είναι μια πολλαπλής θεώρησης μεθοδολογία αφού πρακτικά αποτελεί ένα συνδυασμό της μεθοδολογίας SSM του P. Checkland για τα ευμετάβλητα συστήματα, της συμμετοχικής μεθοδολογίας ETHICS και της δομημένης προσέγγισης.

Αποτελείται από πέντε στάδια με τα εξής προϊόντα:

Στάδιο 1. ανάλυση ανθρώπινης δραστηριότητας

Προϊόν : μοντέλο πρωταρχικών εργασιών

Στάδιο 2. ανάλυση πληροφοριών

Προϊόν : μοντέλο λειτουργιών και μοντέλο οντοτήτων

Στάδιο 3. ανάλυση και σχεδιασμός κοινωνικό – τεχνικών πλευρών

Προϊόντα : απαιτήσεις υπολογιστικών εργασιών

Σύνολο ρόλων και

Εργασίες ανθρώπων

Στάδιο 4. σχεδιασμός επαφής ανθρώπου – υπολογιστή

Προϊόν : τεχνικές απαιτήσεις διεπαφών

Στάδιο 5. σχεδιασμός τεχνικών πλευρών

Προϊόντα : Σχεδιασμός βάσης δεδομένων,

Λειτουργίες και ανάκτηση δεδομένων και

Απαιτούμενοι έλεγχοι

1.11 Μεθοδολογία Jackson System Development (JSD)

Η JSD είναι κύρια τεχνική μέθοδος. Έχει εφαρμοστεί σε μεγάλο εύρος έργων, από εμπορικές εφαρμογές με εκατοντάδες προγράμματα και πολλαπλές βάσεις δεδομένων έως συστήματα έλεγχου σε μικροϋπολογιστές με τεχνικές παράλληλης επεξεργασίας.

Στην πράξη είναι μια μέθοδος καθορισμού – προδιαγραφής, σχεδιασμού και υλοποίησης συστημάτων λογισμικού. Η φιλοσοφία της είναι ότι «το σύστημα είναι ένα μεγάλο πρόγραμμα λογισμικού». Για το λόγο αυτό δεν κατευθύνει τις «soft» περιοχές που εμπλέκονται στην ανάπτυξη του συστήματος, όπως τεχνικές συνέντευξης χρήστη, αναλύσεις κόστους - οφέλους, διεπαφές χρηστών, διασφάλισης ποιότητας και έλεγχος πορείας του έργου.

Η JSD δίνει έμφαση στο διαχωρισμό του κόσμου σε περιβάλλον οντοτήτων και περιβάλλον υλοποίησης. Το δε σύστημα παίρνει τη μορφή ενός δικτύου ακολουθιακών διεργασιών που επικοινωνούν μεταξύ τους. Υπάρχουν τρεις βασικές φάσεις για την ανάπτυξη αυτού του είδους συστημάτων:

- a) Η φάση της μοντελοποίησης
- b) Η φάση του δικτύου
- c) Η φάση της υλοποίησης

Τα μοντέλα που βασίζονται στην JSD μεθοδολογία είναι προσανατολισμένα σε γεγονότα (events / actions), στις οντότητες και τις ιδιότητες (attributes) και σένα σύνολο διεργασιών, που περιγράφουν τη χρονική ακολουθία των γεγονότων, με δυνατότητα παραλληλισμού κατά την εκτέλεση τους.

1.11.1 Η φάση της μοντελοποίησης

Ο κύριος σκοπός της φάσης αυτής είναι ο προσδιορισμός των οντοτήτων, των συσχετισμένων μ' αυτές διεργασιών, καθώς και των δεδομένων που χρειάζονται.

- Το υπό μελέτη σύστημα διαχωρίζεται στο περιβάλλον οντοτήτων και στο περιβάλλον υλοποίησης, με ξεχωριστή δομή το καθένα.

- Η δομή του πρώτου, περιλαμβάνει τις οντότητες του συστήματος και τον τρόπο που αλληλεπιδρούν μεταξύ τους.

- Η δομή του δεύτερου, περιλαμβάνει το υλικό που θα χρησιμοποιηθεί και τα εργαλεία λογισμικού που θα βοηθήσουν στην ανάπτυξη της συγκεκριμένης εφαρμογής.

- Καθορισμός των οντοτήτων και των ενεργειών

- Γενικευμένη περιγραφή των οντοτήτων και των ενεργειών τους στο υπο-εξέταση σύστημα.

- Καταγραφή των χρονικών περιορισμών στην ακολουθία των ενεργειών.

- Απεικόνιση των παραπάνω με τα διαγράμματα δομής, τα οποία μπορούν να παραστήσουν τρεις διαφορετικές δομές ενεργειών: ακολουθία, επανάληψη και επιλογή. Τα διαγράμματα αυτά είναι <<δένδρα>> , τα φύλλα των οποίων αντιστοιχίζονται στα γεγονότα.

- Αρχικοποίηση μοντέλου
 - Προσδιορισμός συνόλου ακολουθιακών διεργασιών, για την απεικόνιση του πραγματικού συστήματος.
 - Μετατροπή του προηγούμενου διαγράμματος σε μορφή ψευδοκώδικα.
 - Οι παραπάνω ενέργειες αποτελούν το πλαίσιο για τον προσδιορισμό των δεδομένων που θα χρησιμοποιηθούν.

- Καθορισμός των δεδομένων
 - Τα δεδομένα είναι τοπικά για κάθε διεργασία.
 - Προσδιορίζονται για κάθε οντότητα και ορίζονται ως μεταβλητές σε κάθε διεργασία.
 - Τροφοδοτούν τις διεργασίες και τις πληροφορούν για τις αλλαγές που έχουν συμβεί στις οντότητες του συστήματος.
 - Ύπαρξη μηχανισμού ενημέρωσης των δεδομένων κατά τη λειτουργία και εξέλιξη του συστήματος.
 - Επιπλέον δεδομένα (συγκεντρωτικά) που αφορούν μια οντότητα μπορούν να δηλωθούν σε νέες διεργασίες ειδικές για αυτό το σκοπό.

1.11.2 Η φάση του δικτύου

Στη φάση αυτή έχουμε ένα δίκτυο ακολουθιών διεργασιών όπου κάθε διεργασία έχει την δική της εσωτερική δομή. Οι διεργασίες του δικτύου επικοινωνούν μεταξύ τους μέσω μηνυμάτων γραψίματος και διαβάσματος, καθώς και μέσω φορμών. Ειδικότερα στο συγκεκριμένο στάδιο της μεθοδολογίας επικεντρωνόμαστε στα ακόλουθα σημεία:

- ✓ Το αρχικό δίκτυο αποτελείται από «model processs» οι οποίες κρατάνε τα κύρια δεδομένα για το σύστημα καθ' όλο το διάστημα της ενημέρωσης του.

- ✓ Γίνεται πρόσθεση νέων διεργασιών, λαμβάνοντας υπόψη του εξής παράγοντες:
 1. Πως η νέα διεργασία συνδέεται με το υπόλοιπο δίκτυο διεργασιών.

2. τι επεξεργασία χρειάζεται στις ήδη υπάρχουσες διεργασίες ,ε τις οποίες θα πραγματοποιηθεί η σύνδεση.

3. προσδιορισμός των εσωτερικών διαδικασιών της νέας διεργασίας.

✓ Βασικοί λόγοι που οδηγούν στην πρόσθεση νέων διεργασιών :

1. συλλογή δεδομένων και ρύθμιση περιπτώσεων σφαλμάτων.

2. συλλογή πληροφοριών σχετικών με τις ενέργειες των οντοτήτων και έλεγχος της ορθότητας των δεδομένων που εισάγονται στις διεργασίες.

3. εξαγωγή extra πληροφοριών υπολογισμών και παρουσίαση των αποτελεσμάτων που παράγονται κατά την λειτουργία του μοντέλου.

✓ Διαχωρισμός της φάσης σε τρεις παράλληλες υποφάσεις σε αντίστοιχα με τα τρία νέα είδη διεργασιών που προστέθηκαν στο δίκτυο κατά την παρούσα φάση.

✓ Η επικοινωνία μεταξύ των διεργασιών στο δίκτυο γίνεται με τους ακόλουθους τρόπους :

a) Μέσω data stream (επικοινωνία μέσω μηνυμάτων)

b) Μέσω state vector (επικοινωνία share variable)

c)

1.11.3 Φάση της υλοποίησης

Η JSD δεν είναι τόσο αναλυτική στο τρόπο υλοποίησης. Μάλιστα στη φάση αυτή περιλαμβάνει ενέργειες που στις περισσότερες μεθοδολογίες περιλαμβάνονται στη φάση του σχεδιασμού, όπως πχ σχεδιασμός αρχείων και βάσεων δεδομένων.

Στη φάση της υλοποίησης τα δυο βασικά ζητήματα που μας απασχολούν είναι:

a) Πως να τρέξουν οι διεργασίες

b) Πως να φορτωθούν τα δεδομένα που περιέχουν

1.12 Αδυναμίες της δομημένης ανάλυσης και σχεδίασης

Ποιό από τα χαρακτηριστικά των εφαρμογών λογισμικού καθιστά ανεπαρκή την προσέγγιση τους με την δομημένη ανάλυση και σχεδίαση; Η απάντηση στο ερώτημα αυτό αποτελεί κλειδί για την αναζήτηση μιας νέας φιλοσοφίας προσέγγισης του λογισμικού. Η απάντηση στο παραπάνω ερώτημα είναι "η πολυπλοκότητα". Η πολυπλοκότητα αφορά όχι μόνο την καθαρά υπολογιστική εργασία του λογισμικού, αλλά και την ανάγκη διαχείρισης μεγάλου όγκου δεδομένων. Σημεία στα οποία η πολυπλοκότητα του λογισμικού κάνει φανερές τις αδυναμίες της δομημένης ανάλυσης και σχεδίασης, μπορούν να αναζητηθούν σε δυο κατευθύνσεις σε αυτή που αφορά το θεωρητικό και σε αυτή που αφορά το πρακτικό μέρος αυτής.

Σε θεωρητικό επίπεδο, η δομημένη ανάλυση και σχεδίαση παρουσιάζει μια εγγενή αδυναμία στην απεικόνιση των οντοτήτων του πραγματικού κόσμου σε συστατικά λογισμικού, καθένα εκ των οποίων επιτελεί ένα μικρό μέρος της λύσης του. Αυτή η εφαρμογή της αρχαίας ρήσης "διαίρει και βασίλευε", έχει νόημα όταν η λύση του προβλήματος είναι μόνο υπόθεση αριθμητικών υπολογισμών. Στον πραγματικό κόσμο, όμως και ιδιαίτερα στο πεδίο εφαρμογών λογισμικού που σχετίζονται με την επιχειρηματική δραστηριότητα, αυτή η προσέγγιση υστερεί για δυο λόγους:

- Πρώτον, δε λαμβάνει υπόψη τα δεδομένα, τα οποία έχουν την δική τους πολύπλοκη δομή και εξαρτήσεις. Τα δεδομένα στη δομημένη ανάλυση και σχεδίαση είναι ανεξάρτητα από τις λειτουργικές μονάδες που επιδρούν σε αυτά, πράγμα που δεν ισχύει στο επίπεδο του πραγματικού κόσμου όπου η διαχείριση δεδομένων δεν είναι ανεξάρτητη από αυτά.

- Δεύτερον, το υπολογιστικό μοντέλο που αποτελείται από το δίδυμο "συστατικά «λογισμικού» και "ανεξάρτητα δεδομένα", δεν αντιστοιχεί σε οντότητες του πραγματικού κόσμου, δηλαδή δεν παριστά οντότητες που είναι αντιληπτές στον πραγματικό κόσμο. Με μια άλλη διατύπωση, δεν μοντελοποιεί εύκολα και φυσικά την επιχειρησιακή λογική. Ως εκ τούτου, είναι ανεπαρκές ως μεθοδολογικό εργαλείο για την δόμηση του λογισμικού.

Σε πρακτικό επίπεδο μπορούμε να αντιμετωπίσουμε τα ακόλουθα προβλήματα:

- Ο προσδιορισμός των απαιτήσεων είναι το δυσκολότερο από τα βήματα της ανάπτυξης του λογισμικού. Η ακριβής περιγραφή των πολλών, πολύπλοκων και αλληλοσχετιζόμενων απαιτήσεων με χρήση μετασχηματισμών και ανεξάρτητων δεδομένων, καθώς και η αντιμετώπιση των μεταβολών ακόμα και κατά τη διάρκεια της ανάπτυξης του λογισμικού, φέρουν τη δομημένη προσέγγιση στα όρια της.

- Η διαχείριση των μοντέλων παράστασης λογισμικού είναι μια δύσκολη και επιρρεπής σε σφάλματα εργασία. Το πλήθος, η πολυπλοκότητα και οι συσχετίσεις των συστατικών αυτών, η μεταβολή τους με το χρόνο, αλλά και οι παρενέργειες κατά την πραγματοποίηση μεταβολών καθιστούν την εργασία αυτή ακόμα δυσκολότερη.

- Η συντήρηση του λογισμικού έχει εξελιχθεί σε μια πολύ δύσκολη διαδικασία που δυσκολεύει περισσότερο καθώς αυξάνεται το μέγεθος του λογισμικού. Έχει αναφερθεί ότι η συνεισφορά της συντήρησης στο συνολικό κόστος κατά το κύκλο ζωής λογισμικού, μπορεί να ξεπεράσει το 50 τοις εκατό.

- Μολονότι κατά την ανάπτυξη μια νέας εφαρμογής ενδέχεται να αναγνωριστούν ομοιότητες με τα χαρακτηριστικά μιας υπάρχουσας, η επαναχρησιμοποίηση συστατικών λογισμικού που έχουν κατασκευαστεί με την δομημένη ανάλυση και σχεδίαση δεν ενθαρρύνεται. Αυτό ισχύει στη γενική περίπτωση, διότι η πολυπλοκότητα και η φύση των συσχετίσεων- ενός συστατικού λογισμικού- μέρους μιας εφαρμογής που κατασκευάστηκε με τη δομημένη φιλοσοφία, δεν επιτρέπουν τη γενίκευση και εύκολη επαναχρησιμοποίηση του. Ο κανόνας αυτός ισχύει ιδιαίτερα για λογισμικό που χρησιμοποιείται σε επιχειρηματικές εφαρμογές και γενικά σε εφαρμογές που σχετίζονται με τη διαχείριση δεδομένων, ενώ εξαίρεση αποτελούν οι βιβλιοθήκες μαθηματικών συναρτήσεων που αφορούν καθαρά λογιστικές εργασίες.

Όσα αναφέρθηκαν, με κανένα τρόπο δεν σημαίνουν δυο πράγματα: πρώτον, ότι η δομημένη ανάλυση και σχεδίαση είναι «άχρηστη» και «εσφαλμένη», ή κάτι τέτοιο

και δεύτερο, ότι η αντικειμενοστρεφής φιλοσοφία έχει τη λύση σε όλα τα προβλήματα της ανάπτυξης του λογισμικού. Όπως εξάλλου αναφέρθηκε, η αναφορά αυτή έχει το νόημα της αντίληψης των εξελίξεων που εισήγαγαν μια νέα φιλοσοφία προσέγγισης του λογισμικού. Πολλά από τα προβλήματα που υπήρχαν στην ανάπτυξη του λογισμικού εξακολουθούν να υπάρχουν με την μια ή άλλη έννοια, ενδεχομένως σε μικρότερο βαθμό. Η αντικειμενοστρεφής τεχνολογία είναι μόνο ένα καλύτερο εργαλείο στα χέρια του ανθρώπου για την λύση των προβλημάτων, δεν είναι ίδια η λύση των προβλημάτων.

1.13 Βασικές έννοιες της αντικειμενοστραφούς τεχνολογίας

Ορισμοί

Αντικειμενοστρεφής (Object Oriented)

Αντικειμενοστρεφής είναι χαρακτηρισμός που σημαίνει στραμμένος (προσανατολισμένος) σε αντικείμενα και αποδίδεται σε εκείνο τον τρόπο σκέψης κατά την ανάπτυξη του λογισμικού, στον οποίο τα «αντικείμενα» είναι βασικές δομικές μονάδες.(3)

Αντικείμενο

Ένα αντικείμενο είναι ένα δομικό συστατικό του λογισμικού. Κάθε αντικείμενο έχει κατάσταση , συμπεριφορά και ταυτότητα. Η κατάσταση περιγράφει όλες τις στατικές ιδιότητες του αντικειμένου, όπως τιμές σε μεταβλητές μνήμης. Οι ιδιότητες αυτές είναι αποτέλεσμα της συμπεριφοράς του αντικειμένου, δηλαδή του τρόπου με τον οποίο αυτό ανταποκρίνεται σε κλήσεις από το περιβάλλον του. Η ταυτότητα είναι μοναδική διάκριση του αντικειμένου από τα ομοειδή του.

Από τεχνικής πλευράς το αντικείμενο μπορεί να θεωρηθεί ως η συγχώνευση δυο εννοιών που κατέχουν δεσπόζουσα θέση στη δομημένη ανάλυση, σχεδίαση και προγραμματισμό: της εγγραφής και της συνάρτησης ή διαδικασίας. Η εγγραφή είναι μια συλλογή μεταβλητών μνήμης, οι οποίες αποτελούν τη δομή της και κάθε στιγμή έχουν κάποια συγκεκριμένη τιμή. Το σύνολο των τιμών των μεταβλητών μιας εγγραφής αποτελεί την κατάσταση αυτής. Αν στην εγγραφή, όπως την ορίσαμε μέχρι τώρα, προσθέσουμε ενεργά συστατικά λογισμικού, δηλαδή μονάδες προγράμματος

που εκτελούν υπολογισμούς, τότε μιλάμε για αντικείμενο και όχι για εγγραφή. Δηλαδή, ένα αντικείμενο περικλείει ένα σύνολο δεδομένων και ένα σύνολο συναρτήσεων που χειρίζονται τα δεδομένα αυτά και επιτελούν τις λειτουργίες του πεδίου ευθύνης αυτού.

1.13.1 Κλάση

Το σύνολο των αντικειμένων που έχουν την ίδια δομή και την ίδια συμπεριφορά, ονομάζεται κλάση. (3)

Θα πρέπει να διακρίνουμε μεταξύ της έννοιας δομή και της έννοιας κατάσταση. Η δομή χαρακτηρίζεται από το ποιες και τι τύπου μεταβλητές περιγράφουν τις ιδιότητες του αντικειμένου, ενώ η κατάσταση είναι ένα σύνολο συγκεκριμένων τιμών στις μεταβλητές αυτές. Η διατύπωση ίδια συμπεριφορά σημαίνει ίδια απόκριση στο ίδιο εξωτερικό ερέθισμα. Η κλάση είναι μια αφηρημένη έννοια, όπως αφηρημένη είναι η έννοια του τύπου(type) στις γλώσσες προγραμματισμού. Ο τύπος δεν είναι κάτι που υπάρχει την ώρα της εκτέλεσης ενός προγράμματος. Αυτό που υπάρχει είναι οι μεταβλητές μνήμης, καθεμία εκ των οποίων είναι κάποιου τύπου.

Μπορούμε λοιπόν, να αναγνωρίσουμε αντιστοιχία μεταξύ των εννοιών τύπος-μεταβλητή και κλάση-αντικείμενο. Όπως ισχύει ότι κάθε μεταβλητή μνήμης είναι κάποιου τύπου έτσι ισχύει ότι κάθε αντικείμενο ανήκει σε μια κλάση, η οποία καθορίζει πλήρως τη δομή και τη συμπεριφορά του. Με βάση τα παραπάνω, μια κλάση μπορεί να θεωρηθεί ως η περιγραφή της δομής και της συμπεριφοράς των αντικειμένων που ανήκουν σε αυτή, τα οποία είναι τα συστατικά στοιχεία λογισμικού που έχουν τη δομή και εκδηλώνουν τη συμπεριφορά.

1.13.2 Στιγμιότυπο – Εκδοχή

— Κάθε αντικείμενο αποτελεί ένα μοναδικό και συγκεκριμένο στιγμιότυπο ή εκδοχή της κλάσης στην οποία ανήκει.(3)

Η κατάσταση ενός αντικειμένου καθορίζεται από τις μεταβλητές κατάστασης, οι οποίες ανήκουν στα ιδιώματα του.

1.13.3 Πεδίο

Ένα πεδίο (field) είναι μια μεταβλητή η οποία παριστάνει ένα ιδίωμα του αντικείμενου. Το σύνολο των τιμών όλων των πεδίων ενός αντικείμενου αποτελεί την κατάσταση αυτού. Ισοδύναμα χρησιμοποιείται και ο όρος ιδίωμα (attribute).(3)

Η συμπεριφορά ενός αντικείμενου καθορίζεται από τον τρόπο με τον οποίο αυτό αντιδρά σε εξωτερικά ερεθίσματα, δηλαδή σε κλήσεις από τα αντικείμενα που αποτελούν το περιβάλλον του.

1.13.4 Μέθοδος

Μια μέθοδος είναι ένα ενεργό συστατικό λογισμικού (συνάρτηση, διαδικασία), η οποία υλοποιεί ένα στοιχείο συμπεριφοράς ενός αντικείμενου. Το σύνολο όλων των μεθόδων ενός αντικείμενου καθορίζει την συμπεριφορά του.(3)

Μια χαρακτηριστική ιδιότητα των αντικειμένων, βασικό στοιχείο της αντικειμενοστραφούς φιλοσοφίας, είναι η κελυφοποίηση (encapsulation) ή , όπως ισοδύναμα αναφέρεται , η απόκρυψη πληροφοριών (information hiding).

1.13.5 Κελυφοποίηση απόκρυψη πληροφοριών

Η απόκρυψη των λεπτομερών υλοποίησης ενός αντικείμενου από το περιβάλλον του, διάμεσο της ελεγχόμενης ορατότητας των πεδίων και των μεθόδων αυτού από άλλα αντικείμενα.(3)

1.14 Σχέσεις μεταξύ κλάσεων

Είναι λογικό αφού υπάρχουν σχέσεις μεταξύ δεδομένων και ενεργών συστατικών λογισμικού να υπάρχουν σχέσεις και μεταξύ των κλάσεων. Όταν λοιπόν μεταξύ δυο κλάσεων αναγνωρίσουμε μια σχέση, τότε η σχέση αυτή υλοποιείται στο επίπεδο των αντικειμένων των κλάσεων αυτών. Μπορούμε να προσπαθήσουμε να αντέχουμε το είδος των σχέσεων που υπάρχουν μεταξύ των κλάσεων, κάνοντας τις ακόλουθες δυο παρατηρήσεις:

- Πρώτον, επειδή οι κλάσεις όπως οι εγγραφές (records) και οι πίνακες (tables) περιέχουν δεδομένα, θα πρέπει με κάποιο τρόπο να παριστάνονται οι σχέσεις μεταξύ των δεδομένων.

- Δεύτερον, εισάγαμε την έννοια της κλάσης αναφερόμενοι στην ανεπάρκεια της δομημένης φιλοσοφίας να μοντελοποιήσει τον πραγματικό κόσμο. Φυσικό είναι λοιπόν, να περιμένουμε οι κλάσεις να προσφέρουν δυνατότητες καλύτερης μοντελοποίησης του κόσμου, μέσα από την εσωτερική δομή αλλά και από τις δυνατές συσχετίσεις μεταξύ τους

Πράγματι, οι σχέσεις μεταξύ των κλάσεων επαληθεύουν τις δυο αυτές παρατηρήσεις. Τρεις είναι οι κύριοι τύποι σχέσεων μεταξύ των κλάσεων. Η συσχέτιση (association), η κληρονομικότητα (inheritance) και η συναρμολόγηση (aggregation).

➤ **Συσχέτιση**

Η συσχέτιση (association) είναι η πιο γενική από τις σχέσεις που μπορούν να συνδέουν δυο κλάσεις. Μπορούμε με αρκετή ασφάλεια να αντιληφτούμε τη συσχέτιση ως μια περισσότερο γενική απ' ότι στο σχεσιακό μοντέλο, σχέση των δεδομένων. Λέμε περισσότερο γενική , διότι δεν περιορίζεται σε τρεις δυνατούς τύπους, όπως στο σχεσιακό μοντέλο αλλά έχει οποιαδήποτε χαρακτηριστικά απαιτεί η φύση του προβλήματος

Μια συσχέτιση χαρακτηρίζεται από:

- Ένα όνομα, ενδεχομένως διαφορετικό ανάλογα με τη φορά ανάγνωσης
- Την πολυπλοκότητα, δηλαδή τον επιτρεπόμενο αριθμό των μελών κάθε μέρους της συσχέτισης.
- Τα ονόματα του ρόλου (role) που παίζει κάθε μέρος στη συσχέτιση.

➤ Κληρονομικότητα ή γενίκευση

Η κληρονομικότητα (inheritance) είναι, ίσως, η πιο πολυδιαφημισμένη σχέση στην αντικειμενοστρεφή φιλοσοφία. Δυο κλάσεις συνδέονται με κληρονομικότητα όταν η μια αποδίδει στην άλλη τα χαρακτηριστικά της, δηλαδή, κατά κάποιο τρόπο, της τα κληροδοτεί. Η πρώτη κλάση λέγεται κλάση-πατέρας, ενώ η δεύτερη κλάση-παιδί και μπορεί να προσθέτει στα χαρακτηριστικά που κληρονομεί (πεδία και μεθόδους) και δικά της. Μια κλάση-πατέρας μπορεί να αποδίδει τα χαρακτηριστικά της σε πολλές κλάσεις-παιδιά.(3)

Στην περίπτωση που κάθε κλάση που συμμετέχει σε σχέση κληρονομικότητας έχει μόνο ένα πατέρα, τότε μιλάμε για απλή κληρονομικότητα (single inheritance). Όταν μια κλάση κληρονομεί χαρακτηριστικά από περισσότερες της μιας κλάσεις-πατέρα, τότε μιλάμε για πολλαπλή κληρονομικότητα (multiple inheritance). Υπάρχουν πολλές πρακτικές πλευρές στην υλοποίηση της κληρονομικότητας μεταξύ των κλάσεων, οι οποίες αφορούν την κάθε συγκεκριμένη γλώσσα προγραμματισμού που υποστηρίζει κληρονομικότητα και δεν θα μας απασχολήσει εδώ.

Κοιτώντας «από πάνω προς τα κάτω» δηλαδή από την κλάση-πατέρα στην κλάση-παιδί, μιλάμε για κληρονομικότητα, διότι η κλάση-παιδί έχει τουλάχιστον ότι χαρακτηριστικό έχει η κλάση-πατέρας. Όπως είπαμε έχει κάποια επιπλέον τα οποία δεν έχει ο πατέρας. Τα επιπλέον χαρακτηριστικά εξειδικεύουν την κληρονομούμενη κλάση, δηλαδή η κλάση-παιδί είναι μια εξειδίκευση της κλάσης-πατέρα.

Κοιτώντας , τώρα, τη σχέση κληρονομικότητας από <<κάτω προς τα πάνω>>, παρατηρούμε ότι η κλάση-πατέρας έχει λιγότερα χαρακτηριστικά από τη κλάση-παιδί, δηλαδή αποτελεί τη γενίκευση της. Με τη έννοια αυτή, μιλάμε ισοδύναμα για γενίκευση. Και στις δυο περιπτώσεις, έχουμε στη διάθεση μας ένα μηχανισμό ταξινόμησης των οντοτήτων που αντιστοιχούν σε κλάσεις, από το γενικό στο ειδικό ή ,ισοδύναμα, από το ειδικό στο γενικό(γενίκευση)

Στην αντικειμενοστρεφή τεχνολογία, κληρονομικότητα και γενίκευση αποτελούν τις δυο όψεις ενός μηχανισμού ταξινόμησης (classification) των οντοτήτων του πραγματικού κόσμου.

Αυτός ο μηχανισμός ταξινόμησης αποτελεί κεντρικό στοιχείο της αντικειμενοστραφούς φιλοσοφίας. Το τελευταίο διάστημα γίνεται περισσότερος λόγος για γενίκευση και λιγότερος για κληρονομικότητα, διότι η γενίκευση μπορεί να γίνει ευκολότερα αντιληπτή ως εργαλείο μοντελοποίησης.

➤ **Συναρμολόγηση**

Η συναρμολόγηση (aggregation) είναι εκείνη η σχέση που εκφράζει ακριβώς αυτό που λέει το όνομα της: την σύνθεση συνόλων από απλούστερα μέρη. Η έννοια της συναρμολόγησης δεν μπορεί να θεωρηθεί ως σχέση κληρονομικότητας διότι δεν αναγνωρίζουμε χαρακτηριστικά που αποδίδονται από μια κλάση σε μια άλλη. Ενδεχομένως η συναρμολόγηση να μπορεί να θεωρηθεί ως μια ειδική συσχέτιση την οποία, όμως, έχει νόημα να εξετάσουμε αυτοτελώς διότι είναι ένα χρήσιμο εργαλείο για την μοντελοποίηση του πραγματικού κόσμου.

Έχοντας δει μια αρχική περιγραφή των χαρακτηριστικών της αντικειμενοστραφούς προσέγγισης, μπορούμε τουλάχιστον να υποψιαστούμε ότι έχουμε στη διάθεση μας όχι μόνο κάποιους νέους όρους που αφορούν τα συστατικά του λογισμικού και τις σχέσεις μεταξύ τους, αλλά και κάποια εργαλεία που μας επιτρέπουν να μοντελοποιήσουμε ένα πρόβλημα με διαφορετικό τρόπο απ' ό τι η δομημένη ανάλυση και σχεδίαση. Πριν αναφερθούμε σ' αυτό, είναι χρήσιμο να εισάγουμε την έννοια της αφαίρεσης (abstraction).

➤ **Αφαίρεση**

Αφαίρεση (abstraction) είναι η νοητική εκείνη λειτουργία όπου από το σύνολο των λεπτομερειών μιας οντότητας ή ενός γεγονότος, επιλεγούμε να επικεντρώσουμε την προσοχή μας μόνο σε κάποιες αφαιρώντας τις υπόλοιπες.

Η αφαίρεση είναι χρήσιμο εργαλείο για τη δημιουργία μοντέλων της πραγματικότητας, τα οποία έχουν μόνο την λεπτομέρεια που μας απασχολεί και χρησιμοποιείται ευρύτατα στην αντικειμενοστρεφή προσέγγιση.

1.15 Γιατί η αντικειμενοστρεφή ανάπτυξη είναι καλύτερη από την δομημένη;

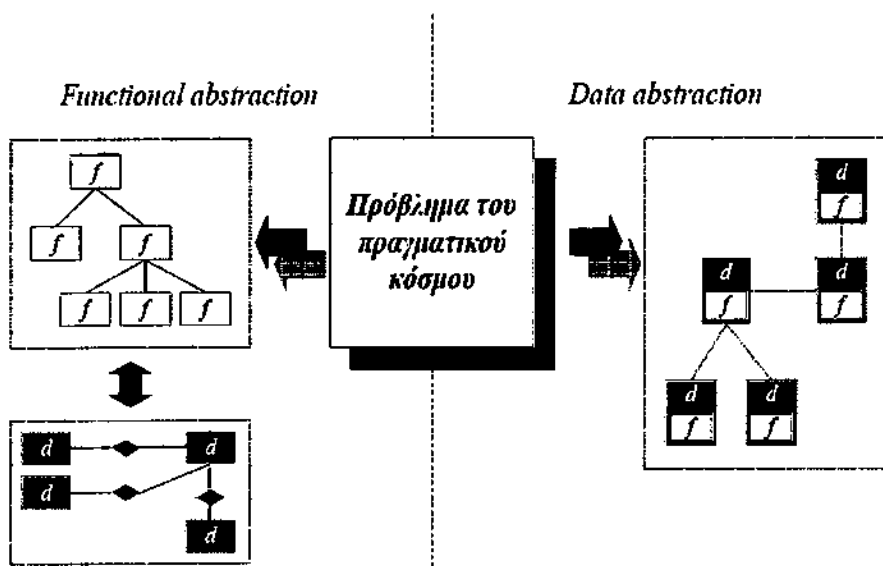
Εδώ γίνεται αναφορά στα χαρακτηριστικά της αντικειμενοστραφούς τεχνολογίας που συνηγορούν στο ότι αυτή αντιμετωπίζει κάποιες από τις αδυναμίες της δομημένης ανάλυσης και σχεδίασης.

Το πρώτο που μπορούμε να υποστηρίξουμε, είναι ότι η ομαδοποίηση δεδομένων και λειτουργιών μέσα στις κλάσεις, τις καθιστούν περισσότερο

αντιστοιχίσαμε με οντότητες του πραγματικού κόσμου. Κάθε οντότητα του πραγματικού κόσμου έχει και κατάσταση και συμπεριφορά και είναι περισσότερο κατανοητό να την παραστήσουμε με ένα σύνθετο αλλά ενιαίο συστατικό λογισμικού που ενσωματώνει και τα δυο, την κλάση, παρά με πολλά ανεξάρτητα μεταξύ τους.

Δεύτερο σημείο, είναι η δυνατότητα αντίληψης της ομαδοποίησης των δεδομένων και λειτουργιών σε μια κλάση, ως τη ευθύνη της κλάσης να χειρίζεται δια των λειτουργιών της τα δεδομένα που περικλείει. Η έννοια αυτής της ευθύνης αποτελεί κεντρικό σημείο σε αρκετές μεθοδολογίες αντικειμενοστραφούς ανάλυσης.

Τρίτο σημείο είναι ο πλούτος των σχέσεων στην αντικειμενοστρεφή φιλοσοφία. Οι σχέσεις της κληρονομικότητας / γενίκευσης και της συναρμολόγησης, δεν υπάρχουν καν ως έννοιες στην δομημένη ανάλυση. Μπορεί να τις συναντήσει κανείς να εφαρμόζονται ως πρακτικές στην κατασκευή του λογισμικού, μόνο που αυτό συμβαίνει χωρίς να αποτελούν εργαλείο μοντελοποίησης αλλά κατασκευαστικό εύρημα. Κοντολογίς με την αντικειμενοστρεφή ανάλυση κάνουμε ανάλυση του εκάστοτε προβλήματος σε διαφορετικά συστατικά από ότι με την δομημένη. Η τοποθέτηση των δεδομένων (πεδία) μαζί με τις λειτουργίες που επιδρούν σε αυτά (μεθόδους) αναδεικνύει τα δεδομένα σε κυρίαρχα στοιχεία της αναλυτικής σκέψης στην αντικειμενοστρεφή φιλοσοφία.



Σχήμα 5:αφαίρεση ως προς τους μετασχηματισμούς και ως προς τα δεδομένα

Στην αντικειμενοστρεφή φιλοσοφία ακολουθούμε αφαίρεση ως προς τα δεδομένα (data abstraction) και μαζί με αυτά τοποθετούμε τις λειτουργίες, πράγμα που δεν ισχύει στη δομημένη ανάλυση, όπου ακολουθούμε αφαίρεση ως προς τής μετασχηματισμούς (functional abstraction) οι οποίοι τελικά είναι μονάδες προγράμματος που επιδρούν σε εντελώς ανεξάρτητα δεδομένα.

1.16 Αντικειμενοστρεφής ανάλυση

Σκοπός των παρακάτω είναι να εισάγει τον χρήστη στην αντικειμενοστρεφή ανάλυση σύμφωνα με την Ενοποιημένη Προσέγγιση ανάπτυξης λογισμικού, η οποία αποτελεί το προϊόν σύγκλισης τριών από τις επικρατέστερες προσεγγίσεις ανάπτυξης λογισμικού με την αντικειμενοστρεφή φιλοσοφία.

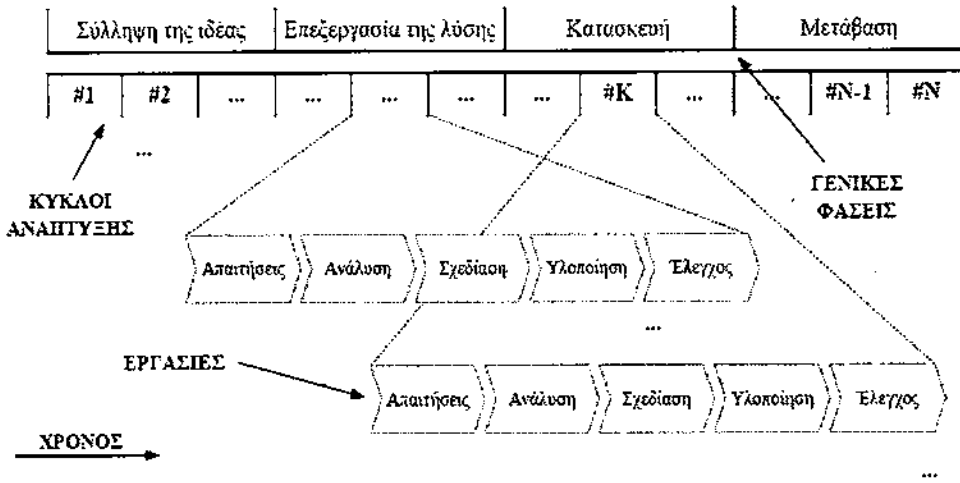
Γενικό πλαίσιο αντικειμενοστρεφή ανάπτυξη λογισμικού

Η αυξανόμενη ζήτηση για ολοένα και περισσότερο πολύπλοκες εφαρμογές λογισμικού έχει δημιουργήσει την απαίτηση ευέλικτων και αποτελεσματικών προσεγγίσεων στην ανάπτυξη του. Η Ενοποιημένη Προσέγγιση υποστηρίζει την ανάπτυξη λογισμικού σύμφωνα με την αντικειμενοστρεφή φιλοσοφία. Τα βασικά χαρακτηριστικά της είναι τα ακόλουθα:

- Χρησιμοποιεί την UML για την παράσταση μοντέλων λογισμικού που κατασκευάζονται κατά την ανάπτυξη.
- Αντιμετωπίζει το λογισμικό ως ένα σύνολο συστατικών που ικανοποιούν απαιτήσεις των χρηστών με αναφορά στις οποίες πραγματοποιεί όλες τις δραστηριότητες ανάπτυξης.
- Αντιμετωπίζει την αρχιτεκτονική του λογισμικού ως κεντρική έννοια στην ανάπτυξη, η οποία είναι δυναμικά αλληλοεξαρτώμενη με τις απαιτήσεις των χρηστών, δηλαδή καθορίζεται από αυτές, αλλά τις επηρεάζει.

- Είναι μια επαναληπτική και αυξητική προσέγγιση, δηλαδή «χτίζει» το τελικό προϊόν ως συσσωρευτικό αποτέλεσμα επαναλήψεων δραστηριοτήτων ανάπτυξης λογισμικού.

Η ενοποιημένη προσέγγιση ακολουθεί το γενικό μοντέλο ζωής λογισμικού όπως αυτό εξειδικεύεται στο παρακάτω σχήμα.(3)



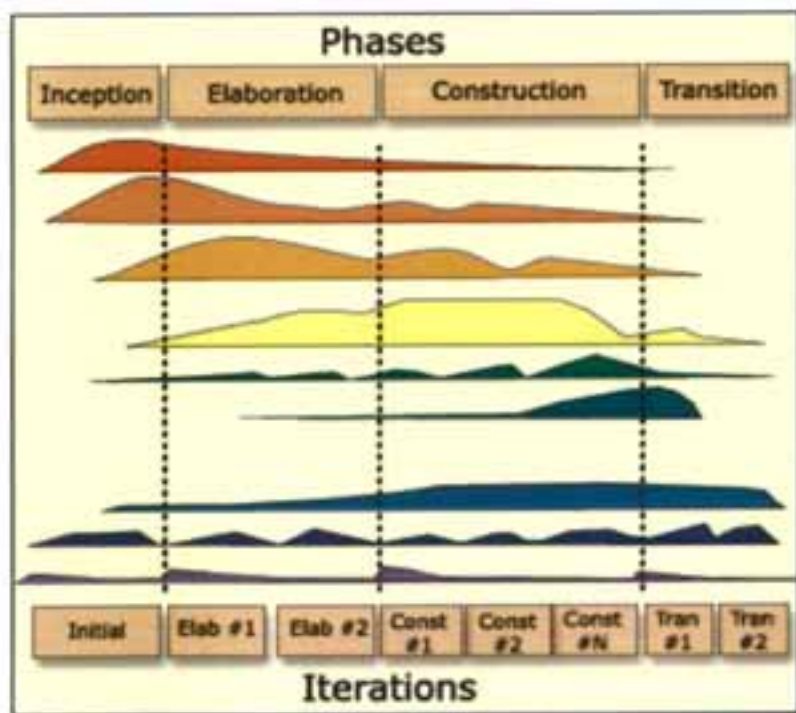
Σχήμα 6:Γενικό μοντέλο ζωής Ενοποιημένης Προσέγγισης

Οι γενικές φάσεις που αναγνωρίζονται, είναι αυτές της σύλληψης της ιδέας, της επεξεργασίας της λύσης, της κατασκευής και της μετάβασης. Κάθε γενική φάση αναλύεται σε κύκλους ανάπτυξης και σε κάθε κύκλο ανάπτυξης λαμβάνουν χώρα οι εργασίες της προδιαγραφής των απαιτήσεων, της ανάλυσης, της σχεδίασης, της υλοποίησης και του ελέγχου.

Κάθε κύκλος ανάπτυξης αφορά ένα υποσύνολο του λογισμικού υπό ανάπτυξη, το οποίο οριοθετείται είτε από το πλήθος των λειτουργιών που περιλαμβάνει, είτε από το βαθμό λεπτομέρειας με τον οποίο τις αντιμετωπίζει. Στο τέλος κάθε κύκλου ανάπτυξης έχουμε μια εκδοχή (release) του λογισμικού. Κατά τις πρώτες (σύλληψη της ιδέας και επεξεργασία της λύσης) η εκδοχή αυτή είναι ένα σύνολο από κείμενα και μοντέλα παράστασης λογισμικού, ενώ κατά τη φάση της κατασκευής η εκδοχή αυτή περιλαμβάνει πηγαίο και εκτελέσιμο κώδικα με ολοένα και περισσότερα από τα απαιτούμενα λειτουργικά χαρακτηριστικά του λογισμικού. Τέλος, κατά τη φάση της μετάβασης το λογισμικό τοποθετείται σε δοκιμαστική λειτουργία, όπου επαληθεύεται

η ικανοποίηση των απαιτήσεων των χρηστών και γίνονται οι απαραίτητες διορθώσεις.

Είναι πιθανόν αντιληπτό ότι δεν εκτελούνται όλες οι εργασίες (προδιαγραφή, ανάλυση, σχεδίαση, υλοποίηση, έλεγχος) σε όλες τις γενικές φάσεις. Για παράδειγμα με όσα γνωρίζουμε από τη δομημένη ανάλυση και σχεδίαση, δεν είναι δυνατό κατά τη φάση της σύλληψης της ιδέας να εκτελεστεί οποιαδήποτε εργασία σχεδίασης λογισμικού. Αυτό είναι εν μέρει μόνο αληθές στην περίπτωση της αντικειμενοστραφούς τεχνολογίας, όπου θα δούμε ότι σχετικά νωρίς στην ανάπτυξη εισάγονται έννοιες όπως «κλάση» και «συνεργασία», οι οποίες τελικά καταλήγουν να είναι πραγματικά κλάσεις και στο πεδίο υλοποίησης. Ενδέχεται, ασφαλώς, οι οντότητες που εντοπίζονται στις πρώτες φάσεις ανάπτυξης του λογισμικού να μην απεικονιστούν αργότερα αυτούσιες κλάσεις. Αυτό, ωστόσο, δεν αναιρεί το γεγονός ότι από πολύ νωρίς στην αντικειμενοστρεφή ανάπτυξη εισάγονται όροι συστατικών στοιχείων υλοποίησης του λογισμικού, οπότε μπορούμε να μιλάμε ταυτόχρονα για ανάλυση και σχεδίαση, πράγμα που δεν ισχύει στην περίπτωση της δομημένης προσέγγισης.



Σχήμα 7: Αναλογία των εργασιών προδιαγραφής των απαιτήσεων, ανάλυσης, σχεδίασης, υλοποίησης και έλεγχου στους κύκλους ανάπτυξης των γενικών φάσεων της Ενοποιημένης Προσέγγισης(3)

Στο παραπάνω σχήμα φαίνεται η αναλογία των εργασιών προδιαγραφής των απαιτήσεων, ανάλυσης, σχεδίασης, υλοποίησης και ελέγχου στους κύκλους ανάπτυξης των γενικών φάσεων της Ενοποιημένης Προσέγγισης ανάπτυξης λογισμικού. Το εμβαδόν μεταξύ του άξονα του χρόνου και κάθε καμπύλης υποδηλώνει την ενέργεια που αποδίδεται σε κάθε εργασία. Παρατηρούμε ότι στις αρχικές φάσεις εκτελούνται περισσότερο εργασίες προδιαγραφής και λιγότερο ανάλυσης, σχεδίασης ή υλοποίησης. Όσο περνάμε σε κύκλους ανάπτυξης που ανήκουν στις επόμενες φάσεις, εκτελούνται περισσότερο κατασκευαστικές εργασίες ή έλεγχος και λιγότερο εργασίες προσδιορισμού των απαιτήσεων, οπότε μεταβάλλεται και το αντίστοιχο εμβαδόν.

Η Ενοποιημένη Προσέγγιση ανάπτυξης λογισμικού παρουσιάζει το πλεονέκτημα ότι σε περίπτωση που κατά την ανάπτυξη εντοπιστεί κάποιο αδιέξοδο, το κόστος περιορίζεται σε αυτό της επανάληψης του τελευταίου κύκλου και όχι ολόκληρης της ανάπτυξης μέχρι το σημείο εκείνο.

1.17 Τι είναι Περίπτωση χρήσης

Μια περίπτωση Χρήσης (Use Case) είναι μια αλληλουχία ενεργειών που εκτελεί το λογισμικό αλληλεπιδρώντας με τον χρήστη ή με εξωτερικά συστήματα, προκειμένου να ικανοποιήσει μία λειτουργική απαίτηση.

Κάθε περίπτωση χρήσης μπορεί να περιγράφεται με μεγαλύτερη ή μικρότερη λεπτομέρεια, όπως άλλωστε και κάθε απαίτηση από το λογισμικό. Επίσης, μια περίπτωση χρήσης χαρακτηρίζεται τόσο από την αλληλουχία των ενεργειών που εκτελεί, όσο και από το μέρος εκείνο με το οποίο αλληλεπιδρά, δηλαδή έναν χρήστη-φυσικό πρόσωπο ή ένα εξωτερικό σύστημα. Το μέρος αυτό ονομάζεται Χειριστής. Ένας Χειριστής (Actor) είναι μια κατηγορία χρηστών ή μια εξωτερική οντότητα με την οποία αλληλεπιδρά το λογισμικό κατά την εκτέλεση των ενεργειών μιας Περίπτωσης Χρήσης.

Στην περίπτωση που ένας Χειριστής αντιστοιχεί σε χρήστη-φυσικό πρόσωπο, κάνουμε λόγο για μια κατηγορία προσώπων και όχι για κάποιο συγκεκριμένο φυσικό πρόσωπο. Αυτό συμβαίνει διότι μας απασχολεί ο καθορισμός της αλληλεπίδρασης ενός χρήστη με το λογισμικό και όχι η ταυτότητα του χρήστη αυτού ως φυσικό

πρόσωπο. Αυτή η αλληλεπίδραση μπορεί να χαρακτηριστεί ως ρόλος που παίζει ένα φυσικό πρόσωπο όταν χρησιμοποιεί το λογισμικό με την έννοια αυτή.

Όταν ένας **Χειριστής** αντιστοιχεί σε κατηγορία χρηστών λογισμικού-φυσικών προσώπων, τότε η έννοια του Χειριστή είναι ισοδύναμη με την έννοια ενός Ρόλου (role) των χρηστών του λογισμικού.

Στην περίπτωση που ο Χειριστής αντιστοιχεί σε εξωτερικό σύστημα (λογισμικό, συσκευή), τότε συνήθως, το σύστημα αυτό είναι συνήθως συγκεκριμένο και πρέπει, σε επόμενη φάση της ανάπτυξης, να προδιαγραφεί πλήρως η διαπροσωπεία (interface) του λογισμικού με αυτό.

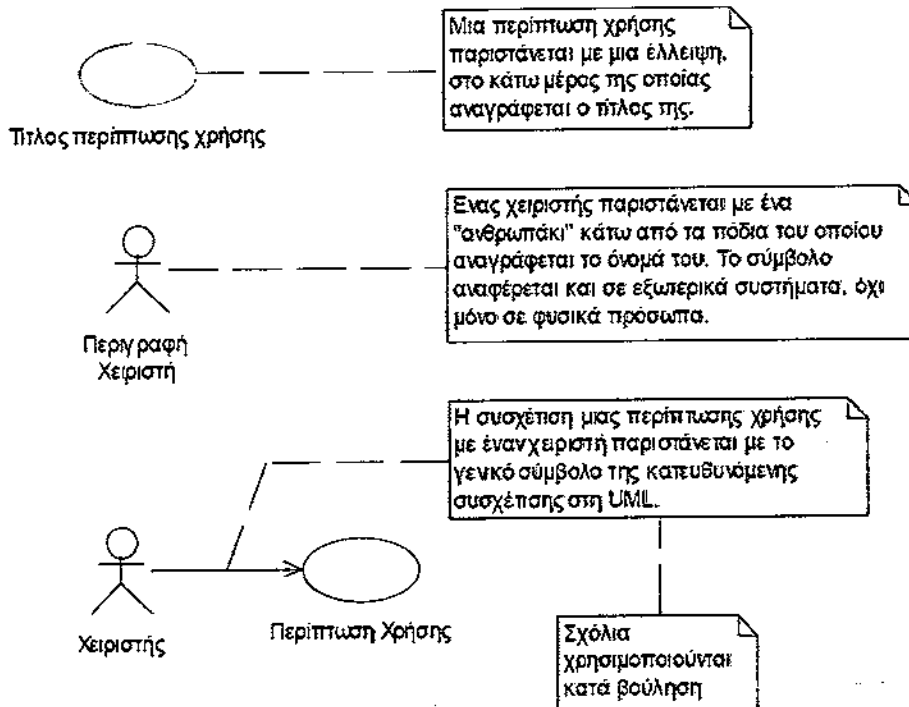
Εξαίρεση σε αυτό αποτελεί η περίπτωση όπου το λογισμικό που κατασκευάζεται προορίζεται να παρέχει υπηρεσίες σε άλλα συστήματα λογισμικού, όποτε ο Χειριστής δεν αντιστοιχεί σε γνωστό εκ των προτέρων εξωτερικό σύστημα. Λαμβάνοντας υπόψη τις δυο τελευταίες παρατηρήσεις μπορούμε να διατυπώσουμε τη θέση ότι στην Ενοποιημένη Προσέγγιση:

Το σύνολο των Χειριστών μιας εφαρμογής λογισμικού αποτελεί το περιβάλλον λειτουργίας της. Κάθε περίπτωση χρήσης ενεργοποιείται από έναν Χειριστή. Όταν εκτελούνται οι εργασίες που περιλαμβάνονται στην περίπτωση χρήσης, τότε μπορούμε να λέμε ότι «εκτελείται η περίπτωση χρήσης».

Η σαφής διάκριση των δυο αλληλεπιδρώντων μερών (περίπτωση χρήσης και χειριστής) εμπεριέχει μια ουσιαστική διαφορά της Ενοποιημένης Προσέγγισης ανάπτυξης λογισμικού από άλλες προσεγγίσεις, στον τομέα του καθορισμού των λειτουργικών αιτήσεων από το λογισμικό. Συνήθως ο καθορισμός των λειτουργικών απαιτήσεων γίνεται απαντώντας στο ερώτημα : τι πρέπει να κάνει το λογισμικό; Στην ενοποιημένη Προσέγγιση το ερώτημα αυτό διαμορφώνεται ως εξής : τι πρέπει να κάνει το λογισμικό για κάθε Χειριστή αυτού;

Μολονότι εκ πρώτης όψεως τα δυο ερωτήματα φαίνονται ισοδύναμα, η εισαγωγή της έννοιας του Χειριστή μας οδηγεί να σκεπτόμαστε με όρους αποτελεσματικότητας για τους χρήστες και το περιβάλλον της εφαρμογής. Αυτό σημαίνει ότι αναζητούμε τις λειτουργικές απαιτήσεις ως εργασίες συνυφασμένες με τους ωφελουμένους από αυτές και όχι αόριστα ως εργασίες που καλό θα ήταν να κάνει το λογισμικό.

Συμβολισμοί UML



Σχήμα 8: Συμβολισμοί UML

Στο παραπάνω σχήμα φαίνονται οι συμβολισμοί που χρησιμοποιεί η UML για την παράσταση των περιπτώσεων χρήσης. Μια απεικόνιση περιπτώσεων χρήσης με τους συμβολισμούς αυτούς αποτελεί ένα διάγραμμα περιπτώσεων χρήσης (Use Case Diagram). Το σύνολο των διαγραμμάτων περιπτώσεων χρήσης αποτελεί το μοντέλο περιπτώσεων χρήσης της εφαρμογής (Use Case Model), το οποίο είναι ένα μοντέλο παράστασης λογισμικού.(3)

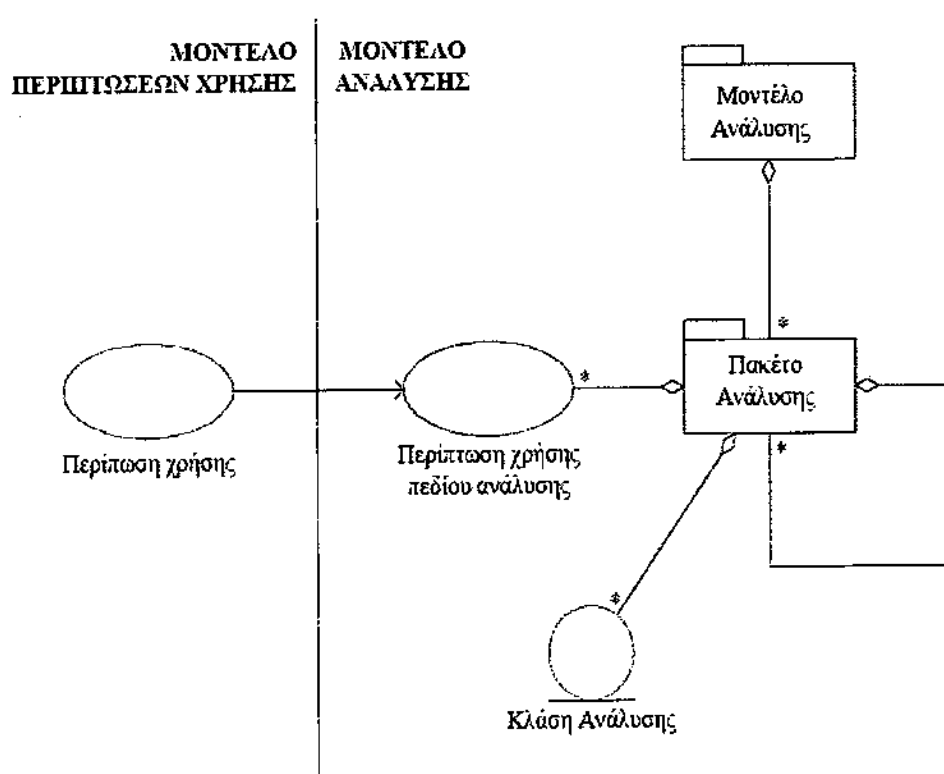
1.18 Από τις περιπτώσεις χρήσης στο Μοντέλο Ανάλυσης

Στην Ενοποιημένη Προσέγγιση, οι περιπτώσεις χρήσης είναι το μέσο για την αποτύπωση των απαιτήσεων του πελάτη κατά τρόπο ώστε αυτές να αποτελούν σημείο αναφοράς για τη συνέχεια της ανάπτυξης. Όπως αναφέραμε οφείλουν να είναι κατανοητές και από τον πελάτη, γεγονός που αναγκαστικά τους αφαιρεί την λεπτομέρεια που απαιτείται από την πλευρά του κατασκευαστή, ο οποίος δεν σκέφτεται μόνο σε όρους λειτουργικότητας, όπως ο πελάτης, αλλά και σε όρους

υλοποίησης. Κατά την ανάλυση, με εναρκτήριο σημείο το μοντέλο περιπτώσεων χρήσης, κατασκευάζεται το μοντέλο ανάλυσης.

1.19 Το μοντέλο ανάλυσης

Ένα μοντέλο ανάλυσης είναι ένα μοντέλο παράστασης λογισμικού, το οποίο αποτελείται από τουλάχιστον ένα πακέτο ανάλυσης (analysis package) το οποίο μπορεί να περιλαμβάνει και άλλα πακέτα τα οποία με τη σειρά τους περιλαμβάνουν κλάσεις ανάλυσης ή και άλλα πακέτα. Κάθε περίπτωση χρήσης του μοντέλου περιπτώσεων χρήσης αντιστοιχίζεται σε μια περίπτωση χρήσης πεδίου ανάλυσης του μοντέλου ανάλυσης, η οποία με τη σειρά της αντιστοιχίζεται σε κάποιο πακέτο ανάλυσης (παρακάτω σχήμα) (3)



Σχήμα 9: από τις περιπτώσεις χρήσης στο μοντέλο ανάλυσης

Το μοντέλο ανάλυσης κάνει πιο απλές τις απαιτήσεις του μοντέλου περιπτώσεων χρήσης στη γλώσσα των κατασκευαστών, έτσι ώστε να μπορούν να εντοπιστούν αρχιτεκτονικά στοιχεία του λογισμικού που απαιτούνται για την συνέχεια της ανάπτυξης, δηλαδή να καθοριστούν οι κλάσεις που θα αποτελέσουν την

εφαρμογή του λογισμικού, η ομαδοποίηση τους και οι συσχετίσεις μεταξύ αυτών. Αυτά, ωστόσο, τα χαρακτηριστικά, είναι μια πρώτη εκδοχή της κατασκευαστικής δομής του λογισμικού, η οποία αναμένεται να μεταβληθεί κατά τη σχεδίαση. Εκεί, προκειμένου να ικανοποιηθούν και οι μη λειτουργικές απαιτήσεις, θα προστεθούν κατασκευαστικές λεπτομέρειες και χαρακτηριστικά του περιβάλλοντος ανάπτυξης και λειτουργίας, τα οποία με κανένα τρόπο δεν απασχολούν την εργασία ανάλυσης.

Κάθε πακέτο ανάλυσης περιλαμβάνει κλάσεις οι οποίες υλοποιούν τη λειτουργική συμπεριφορά μιας ή περισσότερων περιπτώσεων χρήσης. Με τον τρόπο αυτό μεταβαίνουμε από την περιγραφή της απαίτησης σε μια πρώτη περιγραφή υλοποίησης με όρους δομικών μονάδων λογισμικού, δηλαδή των κλάσεων ανάλυσης. Επίσης, κάθε πακέτο ανάλυσης μπορεί να περιλαμβάνει και άλλα πακέτα ανάλυσης και με το τρόπο αυτό μπορεί να γίνει περισσότερο αναγνώσιμο και εύχρηστο το μοντέλο ανάλυσης.

1.20 Τι είναι τα Πακέτα Ανάλυσης

Τα πακέτα ανάλυσης είναι ένα χρήσιμο εργαλείο για την ομαδοποίηση των συστατικών του μοντέλου ανάλυσης. Ένα πακέτο περιέχει κλάσεις ανάλυσης και περιπτώσεις χρήσης πεδίου ανάλυσης. Με την βοήθεια των πακέτων ανάλυσης εφαρμόζεται στην ανάπτυξη του λογισμικού το αρχαίο ρητό "διαίρει και βασίλευε". Αν όλες οι κλάσεις που προέκυπταν από την ανάλυση των περιπτώσεων χρήσης εμφανίζονταν μαζί, τότε το πρόβλημα της παράστασης του μοντέλου ανάλυσης που θα προέκυπτε θα ήταν πολύ μεγάλο.

Με την βοήθεια των πακέτων ανάλυσης, οι κλάσεις που προκύπτουν από την ανάλυση των περιπτώσεων χρήσης ομαδοποιούνται. Δεν υπάρχουν αδιαμφισβήτητα και μοναδικά κριτήρια ομαδοποίησης των κλάσεων ανάλυσης σε πακέτα. Η γενική αρχή είναι ότι σε ένα πακέτο ανήκουν σημασιολογικά συναφείς κλάσεις, όπως για παράδειγμα κλάσεις που υλοποιούν περιπτώσεις χρήσης που ανήκουν σε κάποια κατηγορία που αυθαίρετα την οποία αυθαίρετα ή κλάσεις που σχετίζονται με κάποιο συγκεκριμένο Χειριστή.

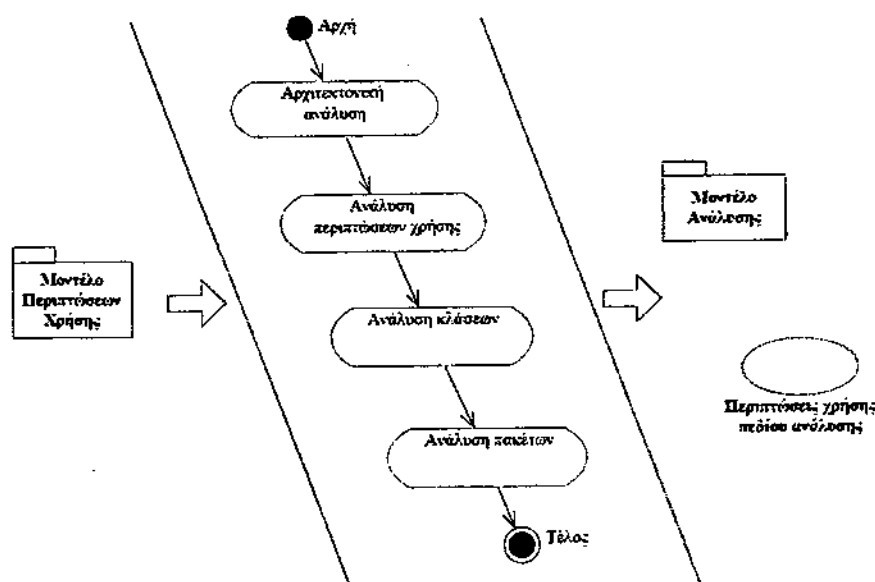
Ένα άλλο κριτήριο διάκρισης των κλάσεων ανάλυσης σε πακέτα αποτελεί η δυνατότητα επαναχρησιμοποίησης αυτών. Αν κατά την ανάλυση ενός προβλήματος διαπιστώσουμε ότι υπάρχουν σε αυτό χαρακτηριστικά που είναι γενικά και απαντώνται και σε άλλα προβλήματα, είναι θεμιτό να θέλουμε να

επαναχρησιμοποιήσουμε τα αποτελέσματα αυτής και για τα άλλα προβλήματα. Ένα υποσύνολο μοντέλου για το οποίο ισχύει το ενδεχόμενο επαναχρησιμοποίησης μπορεί να ορίσει ένα πακέτο ανάλυσης.

1.21 Βήματα στην ανάλυση

Στην αντικειμενοστρεφή ανάπτυξη με την Ενοποιημένη Προσέγγιση, αναγνωρίζονται τέσσερα επιμέρους βήματα στην ανάλυση τα οποία φαίνονται στο παρακάτω σχήμα.

Κατά την αρχιτεκτονική ανάλυση, διαγράφεται η δομή του μοντέλου ανάλυσης με τον καθορισμό των πακέτων ανάλυσης. Κατά την ανάλυση των περιπτώσεων χρήσης καθορίζονται οι κλάσεις ανάλυσης που είναι απαραίτητες έτσι ώστε να επιτευχτεί η απαιτούμενη λειτουργική συμπεριφορά κάθε περίπτωσης χρήσης. Κατά την ανάλυση κλάσεων καθορίζονται τα πεδία και οι μέθοδοι κάθε κλάσης, καθώς και οι συσχετίσεις της με τις άλλες κλάσεις. Τέλος, κατά την ανάλυση πακέτων επιβεβαιώνεται η δόμηση του μοντέλου ανάλυσης και η ικανοποίηση των περιπτώσεων χρήσης που έχουν αντιστοιχηθεί στα πακέτα ανάλυσης.



Σχήμα 10: Βήματα στην ανάλυση της αντικειμενοστραφούς ανάπτυξη με την Ενοποιημένη Προσέγγιση

Στην αντικειμενοστρεφή ανάπτυξη με την Ενοποιημένη Προσέγγιση αναγνωρίζονται τέσσερα επιμέρους βήματα στην ανάλυση τα οποία βλέπουμε στο παραπάνω σχήμα.

1.22 Αρχιτεκτονική ανάλυση

Σκοπός της αρχιτεκτονικής ανάλυσης είναι ο καθορισμός των πακέτων του μοντέλου ανάλυσης. Με τη δόμηση του μοντέλου ανάλυσης σε πακέτα, επιτυγχάνεται με κατάτμηση του προβλήματος η οποία είναι επιθυμητή για την αντιμετώπιση του σε μικρά τμήματα. Δεν είναι απαραίτητο και συνήθως δεν είναι δυνατό να καθοριστούν με την πρώτη όλα τα πακέτα ανάλυσης. Συνήθως, εντοπίζεται μια πρώτη δομή η οποία διαμορφώνεται με των ορισμό νέων πακέτων σε πολλά επίπεδα, σε κάθε κύκλο ανάπτυξης.

Σε πρώτο επίπεδο, ο καθορισμός των πακέτων μπορεί να πραγματοποιηθεί με τα ακόλουθα κριτήρια:

- Ένα πακέτο αντιστοιχεί σε περιπτώσεις χρήσης που είναι σημασιολογικά συναφείς στο πεδίο του προβλήματος.
- Ένα πακέτο αντιστοιχεί σε περιπτώσεις χρήσης που σχετίζονται με έναν Χειριστή.
- Ένα πακέτο είναι όσο το δυνατόν περισσότερο ανεξάρτητο από τα υπόλοιπα

Η εφαρμογή των κριτηρίων αυτών στην πράξη μπορεί, ασφαλώς, να γίνει με πολλούς τρόπους. Η σημασιολογική συναφείς μπορεί να γίνεται με πολλούς τρόπους αντιληπτή και να προκύπτουν περισσότερα του ενός σύνολα πακέτων ανάλυσης για το ίδιο πρόβλημα. Ο ορισμός των πακέτων ανάλυσης με κριτήριο τη σημασιολογική σύναφεια μπορεί να είναι διαφορετικός από αυτόν που θα προκύψει με κριτήριο την αντιστοίχιση στον ίδιο χειριστή. Συνήθως, όταν ο χειριστής περιγράφει εξωτερικό σύστημα κα όχι χρήστη- φυσικό πρόσωπο, η αντιστοίχιση πακέτου ανάλυσης στις περιπτώσεις χρήσης με τις οποίες αυτός σχετίζεται ικανοποιεί και το τρίτο κριτήριο, δηλαδή την ανεξαρτησία του πακέτου ανάλυσης που προκύπτει από τα υπόλοιπα.

Γενικά, η ανεξαρτησία του αναφέρεται ως κριτήριο, έχει το νόημα της ελαχιστοποίησης των επιπτώσεων μιας μεταβολής σε κάποιο από τα συστατικά ενός πακέτου ανάλυσης, σε άλλα πακέτα.

Στο σημείο αυτό, εκτός από τον ορισμό των πακέτων ανάλυσης, μπορεί να γίνει και ένας πρώτος εντοπισμός κάποιων κλάσεων οντοτήτων, συνήθως των περισσότερο φανερών οι οποίες προκύπτουν αβίαστα από την αντίληψη για το πρόβλημα. Η κύρια δουλειά του εντοπισμού των κλάσεων ανάλυσης θα γίνει, πάντως, κατά το επόμενο βήμα, αυτό της ανάλυσης περιπτώσεων χρήσης.

Στην παραπάνω ανάλυση η αντικειμενοστρεφής μέθοδος ανάπτυξης πληροφοριακών συστημάτων χρησιμοποιεί την σχεδιαστική γλώσσα UML. Για το λόγο αυτό θα κρίνεται αναγκαίο να γίνει μια παρουσίαση της.

1.23 Παρουσίαση της UML

Η UML δημιουργήθηκε από την ανάγκη να λυθούν τα προβλήματα τα οποία εμφανίστηκαν κατά τον σχεδιασμό και τη μοντελοποίηση κάποιων συστημάτων. Η UML είναι μία γλώσσα η οποία αναπαριστά όλη την πληροφορία που περιέχει μία μεγάλη επιχείρηση ή ένα μεγάλο σύστημα. Σκοπός της είναι να διευκολύνει την δουλειά αυτών που ασχολούνται με την ανάπτυξη μεγάλων συστημάτων και επιχειρήσεων, και να βοηθά στην δημιουργία «καλών» συστημάτων, ώστε η δουλειά του προγραμματιστή να είναι εύκολη αργότερα όταν αυτός θα περάσει από το σχεδιαστικό στάδιο στο προγραμματιστικό. Η έννοια του «καλού» συστήματος είναι κάπως γενική. Ένα «καλό» σύστημα πρέπει να είναι:

- **Εύχρηστο και χρήσιμο.** Η ευχρηστία είναι ένα πολύ σημαντικό θέμα και είναι ουσιαστικά αυτό που θα καθορίσει το μέλλον του συστήματος ή της επιχείρησης.
- **Αξιόπιστο.** Εξίσου σημαντικό θέμα είναι η αξιοπιστία, διότι η ασφάλεια ενός συστήματος είναι ένα από τα σημαντικότερα θέματα στην μοντελοποίηση.
- **Ευέλικτο.** Το μοντέλο θα πρέπει να προβλέπει όλες τις δυνατές περιπτώσεις στις οποίες θα μεταβεί το σύστημα σε κάθε ενέργεια του χρήστη είτε τυχαία, είτε σκόπιμη.

- **Οικονομικά προσιτό.** Για την ανάπτυξη μίας επιχείρησης το κόστος οφείλει να είναι σε λογικά πλαίσια ώστε το μοντέλο να είναι προσιτό στο αγοραστικό κοινό.
- **Διαθέσιμο και συμβατό** με τις πιο συχνά χρησιμοποιούμενες πλατφόρμες.

Ένας τυπικός ορισμός της UML είναι:

Η ενοποιημένη γλώσσα σχεδιασμού (unified modeling language) (UML) είναι μια γραφική γλώσσα για την οπτική παράσταση, τη διαμόρφωση προδιαγραφών και την τεκμηρίωση συστημάτων που βασίζονται σε λογισμικό. Πρακτικά η UML είναι μια γλώσσα μοντελοποίησης δηλαδή ένα σύνολο από διαγράμματα. Δεν είναι γλώσσα προγραμματισμού(4).

1.23.1 Στόχος της UML

Η UML στοχεύει στο σχεδιασμό αντικειμενοστραφών συστημάτων. Το σχέδιο είναι μια απλοποιημένη παράσταση της πραγματικότητας. Με την χρήση ενός σχεδίου επιτυγχάνεται:

1. παριστάνεται οπτικά το σύστημα το οποίο επρόκειτο να κατασκευαστεί,
2. προσδιορίζεται η δομή και η συμπεριφορά του συστήματος,
3. δημιουργείται ένα πρότυπο στο οποίο θα βασιστεί ην κατασκευή του συστήματος,

1.23.2 Η χρήση της UML

Η UML χρησιμοποιείται για την αναπαράσταση της πληροφορίας που περιέχει μία επιχείρηση. Πολλών διαφορετικών τύπων επιχειρήσεις μπορούν να χρησιμοποιηθούν σε διαφορετικές φάσεις ανάπτυξης του μοντέλου, από τις απαιτήσεις προδιαγραφών ως και τον έλεγχο ενός τελικού συστήματος. Το γεγονός ότι χρησιμοποιούνται πολλά διαγράμματα είναι αυτό που την κάνει πιο προσιτή από τους ανθρώπους που σχεδιάζουν συστήματα λογισμικού και επιχειρήσεις. Σήμερα, η γλώσσα UML είναι ευρύτατα χρησιμοποιούμενη, εξαιτίας των μεγάλων αναγκών για την δημιουργία όλο και περισσότερων μοντέλων συστημάτων ή επιχειρήσεων.

Με τη γλώσσα UML ουσιαστικά προβλέπονται οι τυχόν δυσκολίες που θα εμφανιστούν στο μέλλον, κατά την υλοποίηση του συστήματος. Μπορεί να ακούγεται ασήμαντο όμως η σημασία της είναι τεράστια στο τομέα ανάπτυξης λογισμικού. Επίσης, η UML χρησιμοποιείται κυρίως για την μοντελοποίηση συστημάτων τα οποία στη συνέχεια θα περιγραφούν σε μία αντικειμενοστρεφής γλώσσα.

1.23.3 Διαφορετικοί τύποι συστημάτων

Σκοπός της UML είναι η περιγραφή οποιουδήποτε τύπου συστήματος στα πλαίσια των αντικειμενοστραφών διαγραμμάτων. Λογικά, η πιο κοινή χρήση είναι η δημιουργία μοντέλων ανάπτυξης λογισμικού, αλλά η UML επίσης χρησιμοποιείται και για την περιγραφή μηχανικών συστημάτων ή για την οργάνωση μίας εργασίας.. Παρακάτω παρουσιάζονται κάποιοι διαφορετικοί τύποι συστημάτων και τα πιο κοινά τους χαρακτηριστικά:

- **Σύστημα πληροφοριών.** Η αποθήκευση, η ανάκτηση, η μεταφορά και οι πληροφορίες που υπάρχουν στο σύστημα εκείνη τη στιγμή αποτελούν το σύστημα πληροφοριών.
- **Σύστημα τεχνικών.** Μερικά παραδείγματα ελέγχου τεχνικού εξοπλισμού είναι οι τηλεπικοινωνίες, τα στρατιωτικά συστήματα, ή οι βιομηχανικές διαδικασίες. Πρέπει να χειριζόμαστε ειδική διεπαφή για τον εξοπλισμό και να έχουμε λιγότερο standard λογισμικό ως προς τις πληροφορίες του συστήματος. Τα τεχνικά συστήματα είναι συνήθως συστήματα πραγματικού χρόνου.
- **Εμπεδωμένα (embedded) συστήματα πραγματικού χρόνου.** Εκτελείται σε ένα απλό σύστημα hardware βασισμένο σε κάποιο άλλο εξοπλισμό όπως είναι η κινητή τηλεφωνία, τα αυτοκίνητα, οι οικιακές συσκευές, κ.ά.
- **Κατανεμημένα συστήματα.** Διαχωρίζονται σε ένα πλήθος μηχανών όπου τα δεδομένα μεταφέρονται εύκολα από μηχανή σε μηχανή. Απαιτούν μηχανισμό συγχρονισμένης επικοινωνίας για να εξασφαλίσουν ότι τα δεδομένα είναι ακέραια και

συνήθως σε αντικειμενοστραφείς μηχανισμούς όπως είναι η CORBA, COM/DCOM ή JAVA Beans/RMI.

- **Σύστημα λογισμικού.** Καθορίζει την τεχνική υποδομή η οποία συνήθως χρησιμοποιεί το λογισμικό. Τα λειτουργικά συστήματα, οι βάσεις δεδομένων και οι διεπαφές χρηστών πληρούν χαμηλού επιπέδου λειτουργίες hardware, καθώς παρουσιάζουν μία γενική διεπαφή για τη χρήση από άλλα λογισμικά.
- **Συστήματα επιχειρήσεων.** Περιγράφει τους σκοπούς, τις πηγές πληροφοριών (όπως το ανθρώπινο δυναμικό, τα μηχανήματα κ.ά.), τους κανόνες δηλαδή τους νόμους, τις στρατηγικές των επιχειρήσεων και τις πολιτικές που ακολουθούνται σε μία επιχείρηση.

1.24 Φάσης ανάπτυξης ενός συστήματος

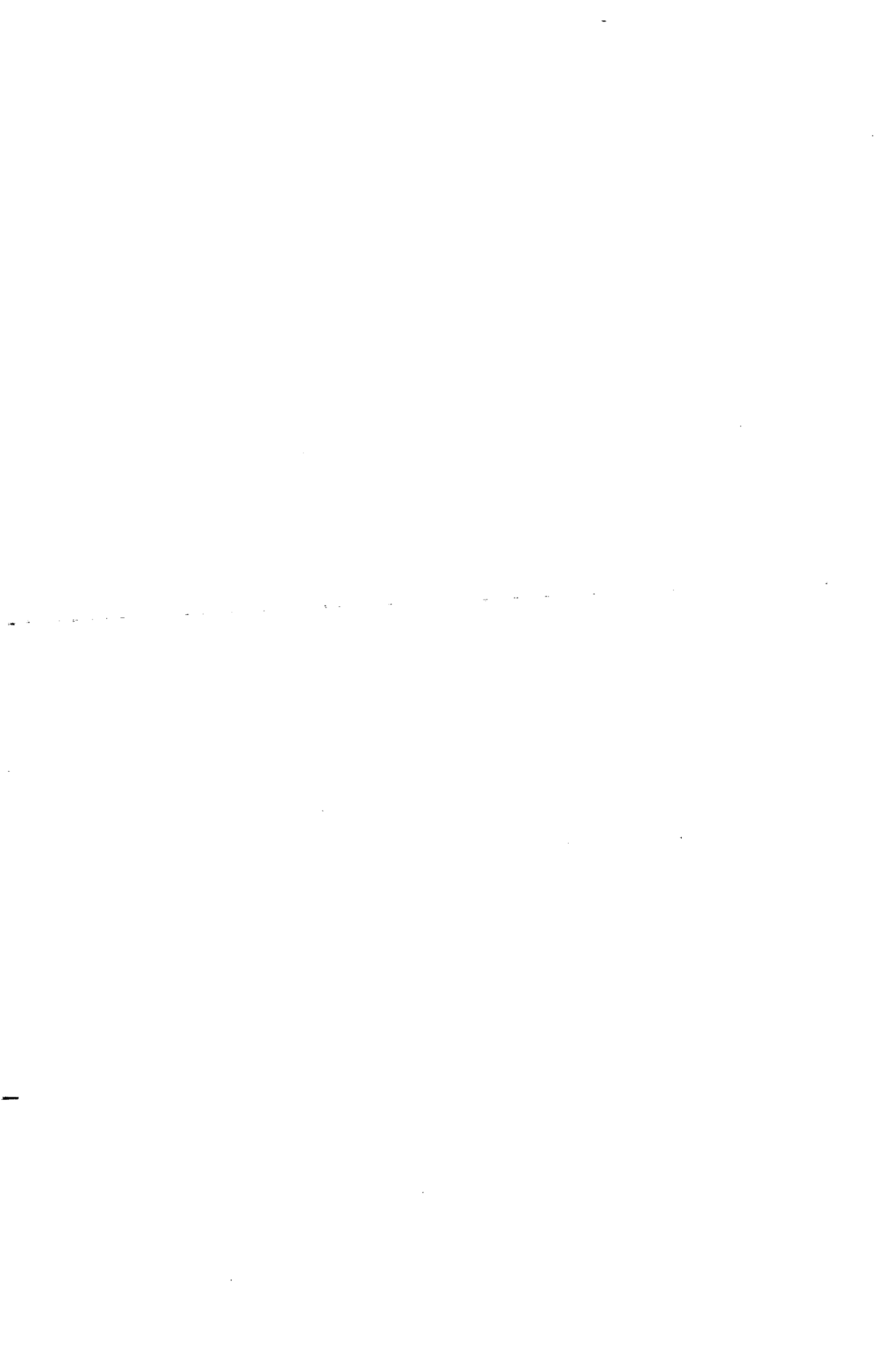
Υπάρχουν πέντε φάσεις ανάπτυξης ενός συστήματος:

- η ανάλυση απαιτήσεων
- η ανάλυση του μοντέλου
- ο σχεδιασμός
- ο προγραμματισμός
- ο έλεγχος

1.24.1 Η ανάλυση απαιτήσεων

Η UML έχει use-cases τα οποία συλλαμβάνουν τις απαιτήσεις του πελάτη. Διαμέσου του σχεδιασμού με την χρήση του use-case οι εξωτερικοί αλληλεπιδρώντες με το σύστημα μοντελοποιούνται μαζί με την λειτουργικότητα που απαιτούν από το σύστημα. Τα άτομα τα οποία αλληλεπιδρούν με το σύστημα και τα use-cases μοντελοποιούνται με σχέσεις, και επικοινωνούν μεταξύ τους συνδεδειμένα ή είναι χωρισμένα σε ιεραρχίες. Τα άτομα αυτά, ονομάζονται actors και οι σχέσεις μεταξύ τους περιγράφονται με τα διαγράμματα use-cases της UML. Κάθε use-case περιγράφεται με κείμενο και αναλύει τις απαιτήσεις του πελάτη, δηλαδή τι ακριβώς περιμένει από το σύστημα χωρίς να λάβει υπόψη του πως θα υλοποιηθεί η





λειτουργικότητά του. Η ανάλυση απαιτήσεων μπορεί επίσης να γίνει με επιχειρησιακές διαδικασίες, όχι μόνο για τα συστήματα λογισμικού.

1.24.2 Η ανάλυση του μοντέλου

Η φάση της ανάλυσης απαιτήσεων αφορά την κύρια αφηρημένη έννοια (κλάσεων και αντικειμένων) και μηχανισμούς οι οποίοι είναι παρόντες στην περιοχή του προβλήματος. Οι κλάσεις οι οποίες μοντελοποιούνται είναι παρόμοιες, μαζί με τις σχέσεις μεταξύ τους και περιγράφονται με ένα διάγραμμα κλάσεων σε UML. Στην ανάλυση, μόνο οι κλάσεις οι οποίες ανήκουν στον τομέα του προβλήματος μοντελοποιούνται με τη χρήση μη τεχνικών κλάσεων οι οποίες καθορίζουν τις λεπτομέρειες και τις λύσεις στα συστήματα λογισμικού, όπως είναι οι κλάσεις για διεπαφές, βάσεις δεδομένων, τηλεπικοινωνίες κ.τ.λ.

1.24.3 Η σχεδίαση

Στη φάση της σχεδίασης τα αποτελέσματα της ανάλυσης μπορούν να δοθούν με τεχνικό τρόπο. Νέες κλάσεις προστίθενται για να βελτιώσουν την τεχνική υποδομή, την διεπαφή με το χρήστη, τη βάση χειρισμού για την αποθήκευση αντικειμένων σε αυτή, τις επικοινωνίες με άλλα συστήματα, την αλληλεπίδραση με τις συσκευές του συστήματος και άλλα. Η ανάλυση του προβλήματος των κλάσεων βασίζεται σε αυτή την τεχνική υποδομή, κάνοντας δυνατή την αλλαγή και την τροποποίηση του συστήματος. Τα αποτελέσματα σχεδιασμού τα παίρνουμε με περισσότερες λεπτομέρειες κατά τη φάση του σχεδιασμού.

1.24.4 Ο προγραμματισμός

Στη φάση του προγραμματισμού ή της κατασκευής καλύτερα, οι κλάσεις από την σχεδιαστική φάση μετατρέπονται σε πραγματικό κώδικα στην αντικειμενοστρεφή γλώσσα προγραμματισμού (η χρήση μη διαδικαστικών γλωσσών δεν προτείνεται). Εξαρτάται από το εύρος της γλώσσας που χρησιμοποιούμε και αυτό μπορεί να είναι δύσκολη ή εύκολη δουλειά. Όταν κάνουμε ανάλυση απαιτήσεων ή σχεδιασμό μοντέλων με τη χρήση UML, είναι καλύτερα να αποφεύγουμε να ερμηνεύουμε νοηματικά τα μοντέλα σε κώδικα. Στις αρχικές φάσεις, η σημασία των μοντέλων είναι η κατανόηση και η δόμηση του συστήματος. Επιπλέον βγάζοντας εύκολα

συμπεράσματα όσον αφορά τον κώδικα μπορεί να είναι αντιπαραγωγικό στη δημιουργία απλού και σωστού κώδικα. Ο προγραμματισμός είναι μία ξεχωριστή φάση κατά την διάρκεια του οποίου τα μοντέλα μετατρέπονται σε πηγαίο κώδικα.

1.24.5 Έλεγχος

Ένα σύστημα είναι λογικά δοκιμασμένο και διαχωρίσιμο σε μονάδες ελέγχου, σε μονάδες ελέγχου ενοποίησης και μονάδες ελέγχου αποδοχής. Οι μονάδες ελέγχου είναι ατομικές κλάσεις και είναι τυπικά εκτελούμενες από τον προγραμματιστή. Οι μονάδες ελέγχου ενοποίησης συνδέουν τις κλάσεις και τα συστατικά σε διάταξη η οποία επαληθεύει ότι επικοινωνούν όπως ήταν αναμενόμενο. Οι μονάδες ελέγχου αποδοχής του συστήματος βλέπουν το σύστημα ως ένα μαύρο κουτί και ελέγχουν αν το σύστημα έχει την αξιοπιστία που αναμένεται από έναν τελικό χρήστη. Ο έλεγχος αποδοχής γίνεται από τον πελάτη για να επιβεβαιώσει ότι το σύστημα ικανοποιεί τις απαιτήσεις οι οποίες είναι παραπλήσιες με τους ελέγχους του συστήματος. Διαφορετικές ομάδες ελέγχου, βασίζονται σε διαφορετικά διαγράμματα UML. Οι μονάδες ελέγχου χρησιμοποιούν διαγράμματα κλάσεων και λεπτομερείς κλάσεις, οι μονάδες ελέγχου ενοποίησης χρησιμοποιούν τυπικά διαγράμματα συνιστωσών και διαγράμματα συνεργασίας, και οι μονάδες ελέγχου αποδοχής του συστήματος εφαρμόζουν τα διαγράμματα use-case για να επικυρώσουν ότι το σύστημα συμπεριφέρεται όπως αρχικά είχε καθοριστεί.

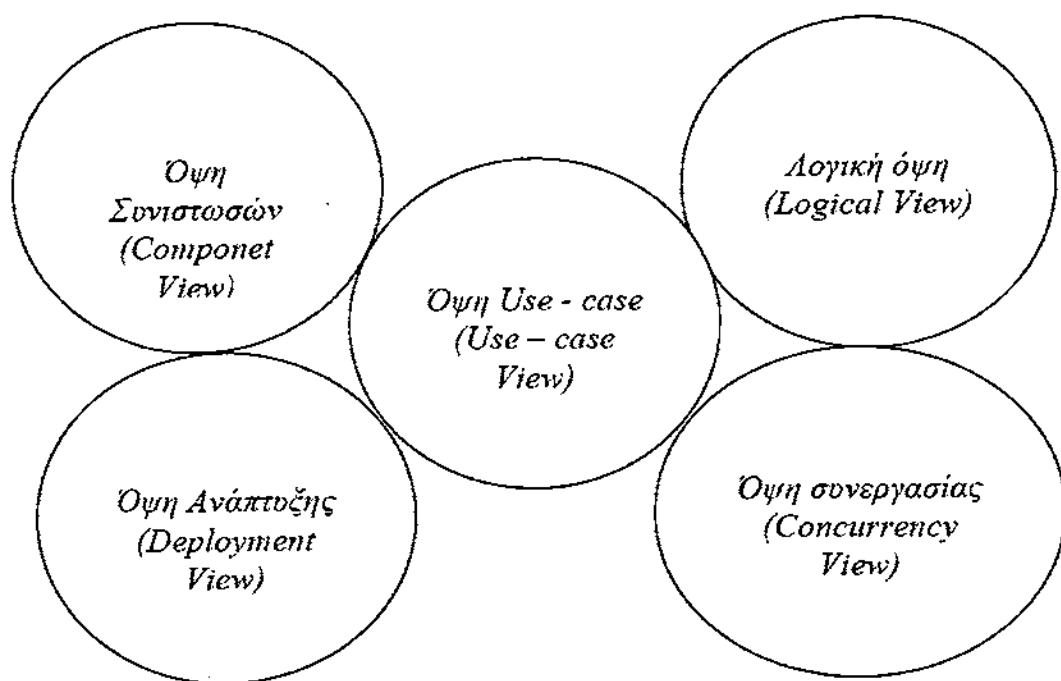
1.25 Δομικά στοιχεία UML

1.25.1 Οι όψεις στη UML

Ο σχεδιασμός μιας επιχείρησης όσο απλός και αν φαίνεται, είναι αρκετά πολύπλοκος και χρονοβόρος. Για καλύτερα αποτελέσματα, όλη η επιχείρηση περιγράφεται από ένα απλό διάγραμμα το οποίο την περιγράφει αναμφίβολα, και είναι εύκολο στην κατανόηση και στην επικοινωνία ανάμεσα στα επιμέρους τμήματα. Ωστόσο, αυτό είναι συνήθως αδύνατο. Ένα απλό γράφημα δεν μπορεί να συλλάβει όλη την πληροφορία η οποία απαιτείται για την περιγραφή μιας επιχείρησης. Ένα σύστημα περιγράφεται από πολλές διαφορετικές απόψεις όπως είναι: η λειτουργικότητα (είναι

στατική δομή και δυναμική αλληλεπίδραση), μη λειτουργικότητα (χρονικές απαιτήσεις, αξιοπιστία, δυνατότητα ανάπτυξης, κ.ά.) και οργανωτικά θέματα (οργάνωση της δουλειάς, χαρτογράφηση σε κώδικα με τη χρήση υποσυστημάτων, κ.ά.). Επιπλέον, μία επιχείρηση περιγράφεται από ένα αριθμό όψεων αναπαριστώντας τη σχεδιαστική περιγραφή μίας ολοκληρωμένης επιχείρησης.

Κάθε όψη περιγράφεται από έναν αριθμό διαγραμμάτων τα οποία περιέχουν την πληροφορία η οποία επικεντρώνεται σε ένα συγκεκριμένο θέμα της επιχείρησης. Υπάρχει μία ασθενής επικάλυψη, έτσι ώστε το διάγραμμα να μπορεί να είναι ένα κομμάτι ενός ή περισσότερων όψεων. Κοιτώντας την επιχείρηση από διαφορετικές όψεις, είναι δυνατόν να επικεντρωθούμε σε ένα συγκεκριμένο θέμα αυτής κάθε φορά. Ένα διάγραμμα σε μία ειδική όψη πρέπει να είναι αρκετά απλό για να επικοινωνεί εύκολα με άλλα επιμέρους τμήματα της επιχείρησης και να είναι κατανοητό. Ένα διάγραμμα περιέχει γραφικούς συμβολισμούς οι οποίοι αναπαριστούν τα στοιχεία των μοντέλων της επιχείρησης. Το παρακάτω σχήμα δείχνει τις όψεις της UML. Αυτές είναι (5)



➤ Η όψη use – case (use – case view)

Η όψη use – case περιγράφει την λειτουργικότητα της επιχείρησης, η οποία πρέπει να παραδοθεί, όπως την αντιλαμβάνονται οι εξωτερικοί αλληλεπιδρώντες με την επιχείρηση που μοντελοποιείται. Ένας χρήστης αλληλεπιδρά με το σύστημα, το οποίο μπορεί να είναι ένας άλλος χρήστης ή ένα άλλο σύστημα. Η όψη use – case προορίζεται για τους πελάτες, τους σχεδιαστές, τα άτομα που αναπτύσσουν το μοντέλο, και τα άτομα που δοκιμάζουν και ελέγχουν την λειτουργικότητα του μοντέλου. Η όψη use – case είναι κεντρική, από τότε που τα περιεχόμενα είναι καθοδηγούμενα από άλλες όψεις. Ο τελικός στόχος του μοντέλου είναι να εξασφαλίσει την λειτουργικότητα η οποία περιγράφεται σε αυτή την όψη για αυτό το λόγο αυτή επηρεάζει όλες τις άλλες. Η όψη αυτή χρησιμοποιείται επίσης για να επικυρώσει και να επαληθεύσει το μοντέλο ελέγχοντας τις όψεις use – case μπροστά στους πελάτες και απέναντι στο ολοκληρωμένο σύστημα.

➤ Logical View (λογική όψη)

Η όψη αυτή περιγράφει πως εξασφαλίζεται η λειτουργικότητα του μοντέλου. Απευθύνεται κυρίως σε σχεδιαστές συστημάτων και στα άτομα που αναπτύσσουν συστήματα. Σε αντίθεση με το διάγραμμα use – case, το λογικό διάγραμμα επικεντρώνεται στο εσωτερικό της επιχείρησης. Περιγράφει και τα δύο και την στατική δομή και την δυναμική συνεργασία η οποία λαμβάνει χώρα όταν τα αντικείμενα στέλνουν μηνύματα το ένα στο άλλο χωρίς να εξασφαλίσουν κάποια συγκεκριμένη λειτουργία. Η στατική δομή περιγράφεται με διαγράμματα κλάσεων και αντικειμένων. Ο δυναμικός σχεδιασμός περιγράφεται από δηλώσεις, ακολουθίες, συνεργασίες και διαγράμματα δραστηριοτήτων.

➤ Component View (όψη συνιστωσών)

Η όψη αυτή είναι μια περιγραφή της ενεργοποίησης κάθε υποομάδας και εξαρτήσεων της. Είναι κυρίως για τα άτομα που αναπτύσσουν το σύστημα, και συμφωνεί με το διάγραμμα συνιστωσών. Οι συνιστώσες, οι οποίες είναι διαφορετικών τύπων κώδικας των υποομάδων, προσδίδοντας τη δομή και τις εξαρτήσεις τους.

➤ **Concurrency View (η όψη συνεργασίας)**

Η όψη της συνεργασίας ασχολείται με την διαίρεση του συστήματος σε επιμέρους διαδικασίες. Αυτή η όψη επιτρέπει την αποτελεσματική χρήση των πόρων της επιχείρησης την παράλληλη εκτέλεση και τον χειρισμό γεγονότων που δεν γίνονται παράλληλα από το περιβάλλον. Ουσιαστικά, η όψη αυτή μας επιτρέπει να χωρίσουμε το σύστημα σε επιμέρους τμήματα τα οποία θα πρέπει να επικοινωνούν σωστά και συγχρονισμένα.

➤ **Deployment View (η όψη ανάπτυξης)**

Τέλος το διάγραμμα deployment δείχνει την φυσική ανάπτυξη του συστήματος ως προς τα υλικά που απαιτούνται, όπως το είδος των υπολογιστών και των συσκευών (κόμβων) και ο τρόπος με τον οποίο συνδέονται μεταξύ τους. Αυτή η όψη είναι ένας χάρτης που αναπαριστά πως τα συστατικά αναπτύσσονται στο φυσικό επίπεδο, όπως για παράδειγμα τα προγράμματα ή τα αντικείμενα που εκτελούνται σε έναν υπολογιστή.

1.25.2 Διαγράμματα της UML

Τα διαγράμματα δείχνουν όλα τα στοιχεία σχεδιασμού τοποθετημένα κατάλληλα για να επεξηγήσουν ένα συγκεκριμένο τμήμα της επιχείρησης. Ένα μοντέλο έχει τυπικά διάφορα διαγράμματα κάθε τύπου και είναι ένα κομμάτι μίας συγκεκριμένης όψης η οποία όταν απεικονίζεται, προσδιορίζεται από μία όψη. Μερικοί τύποι διαγραμμάτων μπορεί να είναι τμήματα διαφόρων όψεων, ανάλογα με τα περιεχόμενα του διαγράμματος.

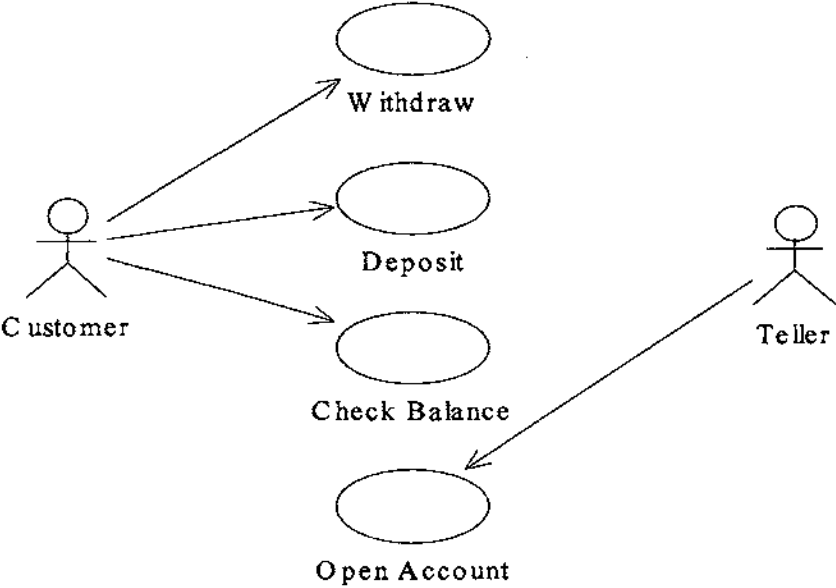
Η UML ορίζει τα παρακάτω διαγράμματα:

- Διάγραμμα περιπτώσεων χρήσης (use case diagram)
- Διαγράμματα δομής
 - Διάγραμμα κλάσεων (class diagram)
 - Διάγραμμα αντικειμένων (object diagram)
- Διαγράμματα συμπεριφοράς

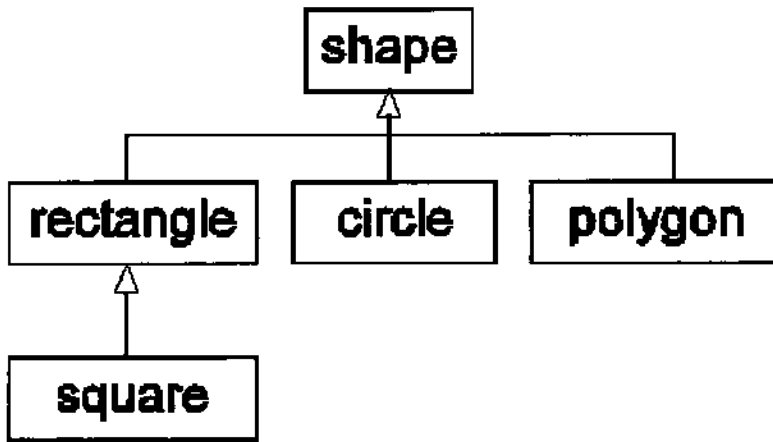
- Διάγραμμα καταστάσεων (statechart diagram)
- Διάγραμμα δραστηριοτήτων (activity diagram)
- Διαγράμματα αλληλεπίδρασης
 - Διάγραμμα ακολουθίας (sequence diagram)
 - Διάγραμμα συνεργασίας (collaboration diagram)
- Διαγράμματα δομής υλοποίησης
 - Διάγραμμα εξαρτημάτων (component diagram)
 - Διάγραμμα ανάπτυξης (deployment diagram)

Στη συνέχεια δίνονται οι λεπτομέρειες, το συντακτικό τους και την ακριβή τους έννοια στα κυριότερα διαγράμματα της UML τα οποία τα συναντάμε πολύ συχνά κατά την ανάπτυξη και σχεδιασμό ενός συστήματος.

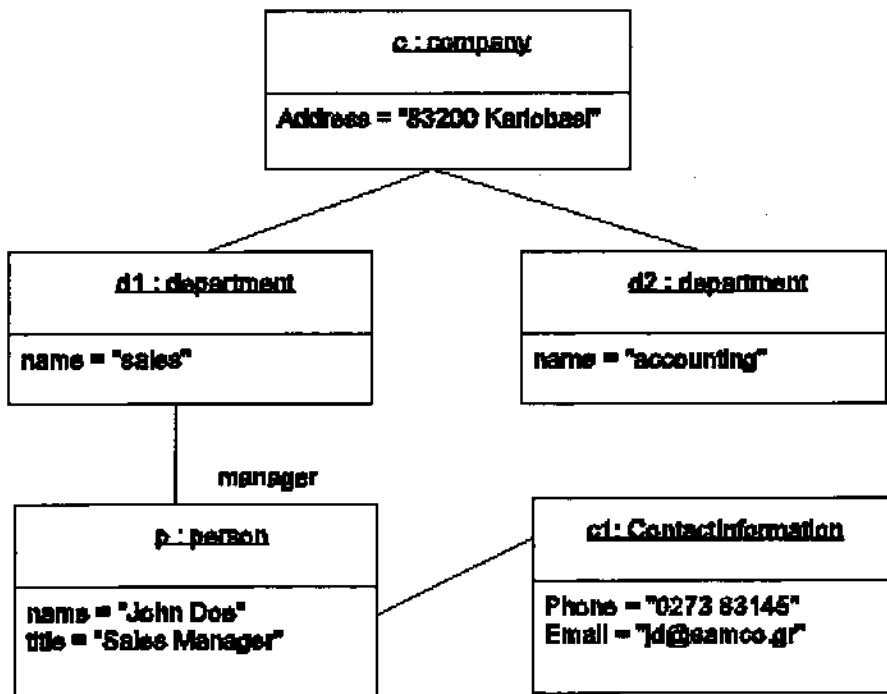
- **Διαγράμματα περίπτωσης χρήσης (Use – Case)** τα οποία χρησιμοποιούνται για την μοντελοποίηση της συμπεριφοράς ενός συστήματος, υποσυστήματος ή κλάσης, όπως αυτή γίνεται αντιληπτή από τον εξωτερικό χρήστη. (4)



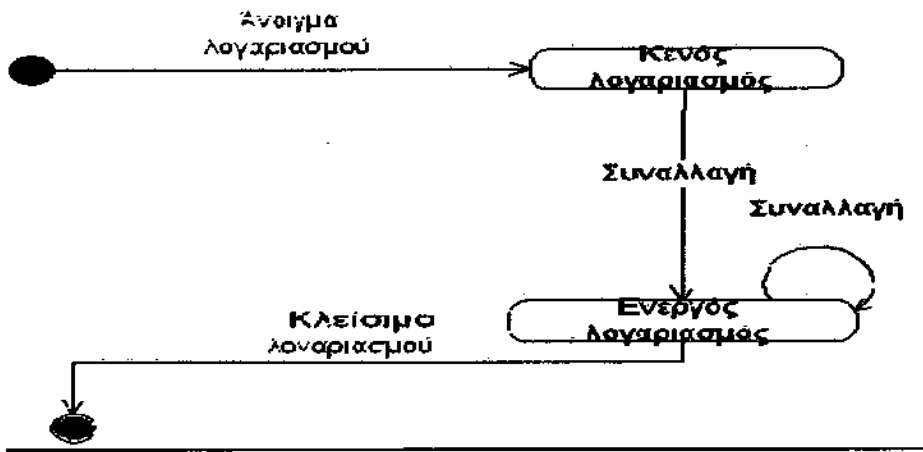
- Διαγράμματα κλάσεων (class diagram) τα οποία δείχνουν τις κλάσεις του συστήματος και τους αντίστοιχους δεσμούς.



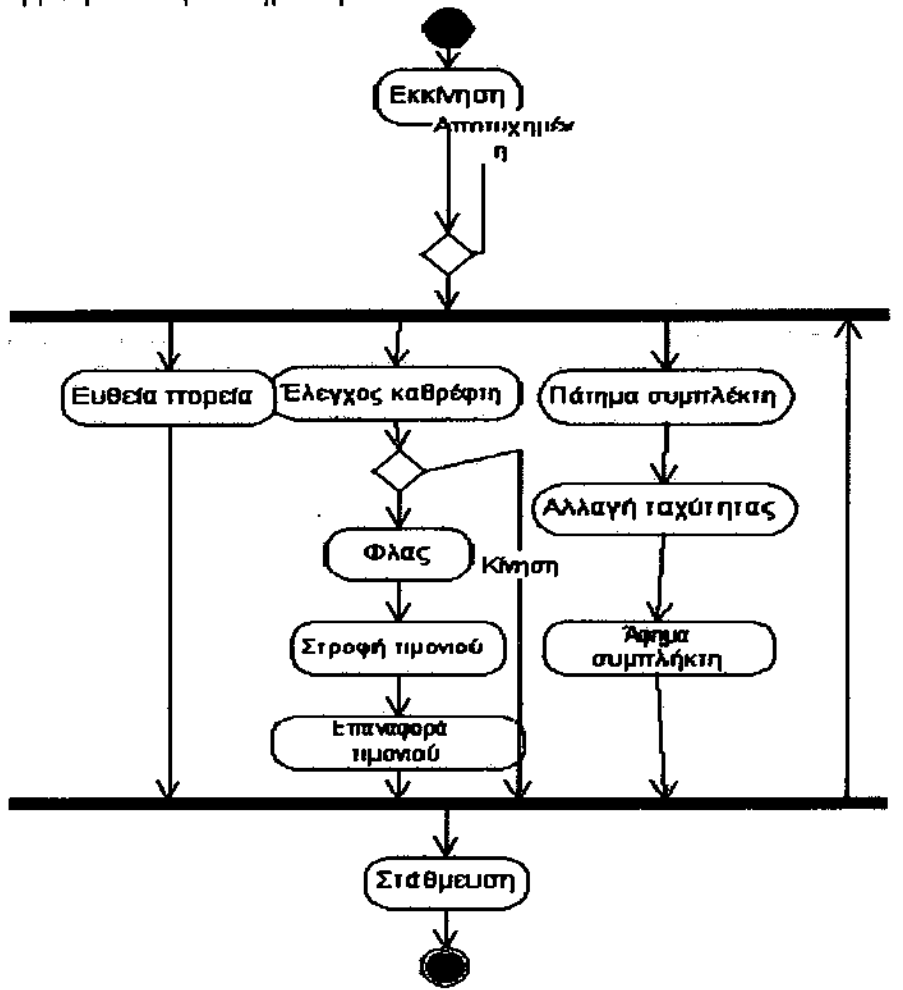
- Διαγράμματα αντικειμένων (object diagram) τα οποία χρησιμοποιούνται για το σχεδιασμό της στατικής κατάστασης του συστήματος κατά μια συγκεκριμένη χρονική στιγμή. (6)



- Διαγράμματα καταστάσεων (state diagram) τα οποία παρουσιάζουν τις μεταπτώσεις μεταξύ καταστάσεων από διάφορα γεγονότα. (6)



- Διαγράμματα δραστηριοτήτων (activity diagram) τα οποία δείχνουν την σειριακή ροή των δραστηριοτήτων.



51

1.26 Χρονοπρογραμματισμός – CPU SCHEDULING

Ο χρονοπρογραμματισμός ο οποίος είναι ευρέως γνωστός ως CPU Scheduling κατέχει εξέχουσα σημασία στην δημιουργία ενός πληροφοριακού συστήματος . ο λόγος για τον οποίο είναι σημαντικός είναι ότι διαιρεί μια διεργασία της ΚΜΕ σε περισσότερες. Με αυτό επιτυγχάνεται:

- (α) να μην χάνεται χρόνος της ΚΜΕ εν αναμονή ολοκλήρωσης των λειτουργιών I/O με περιφερειακά, και
- (β) να διαμοιράζεται ο χρόνος της ΚΜΕ στις διάφορες διεργασίες του συστήματος έτσι ώστε η απόκριση του συστήματος να είναι η «επιθυμητή»

1.26.1 Ουρές Χρονοπρογραμματισμού Διεργασιών (Process Scheduling Queues)

Για την επίτευξη της παραπάνω λειτουργίας έχουν δημιουργηθεί κάποιες ουρές αυτές είναι οι ακόλουθες:

- Ουρά Εισόδου (ή Ουρά Εργασιών, Job queue) – η ουρά στην οποία τοποθετούνται οι διεργασίες που εισέρχονται στο σύστημα
- Ουρά Έτοιμων Διεργασιών (Ready queue) – το σύνολο των διεργασιών που έχουν αποκτήσει χώρο στην κύρια μνήμη, είναι δηλαδή έτοιμες για εκτέλεση και αναμένουν τη σειρά τους
- Ουρές Συσκευών (Device queues) – οι διεργασίες που περιμένουν εξυπηρέτηση από κάποια συσκευή I/O
- Οι διεργασίες μετακινούνται μεταξύ των ουρών κατά τη διάρκεια της «ζωής τους» σε ένα ΥΣ

1.26.2 Τι είναι οι Διαδικασίες- Διεργασίες

Μια διαδικασία- διεργασία είναι η εκτέλεση ενός προγράμματος. Θεωρείται συχνά ως μονάδα της εργασίας σε ένα συγκρότημα ηλεκτρονικών υπολογιστών, επειδή το λειτουργικό σύστημα τρέχει συνεχώς τις διαδικασίες.(8)

1.26.3 Πως δημιουργείται μια διαδικασία- Διεργασία

Οι διαδικασίες μπορούν να δημιουργηθούν από τον πυρήνα του λειτουργικού συστήματος ή από το χρήστη. Ένα παράδειγμα μιας διαδικασίας που δημιουργείται από το χρήστη είναι το Microsoft Word. Αυτή η διαδικασία δημιουργείται μόνο μόλις αποφασίσει να προωθήσει (να χρησιμοποιήσει) ο χρήστης το Word. Οι διαδικασίες που δημιουργούνται από τον πυρήνα φροντίζουν να υπακούν στους διάφορους στόχους των κάθε συστημάτων. Μια υπάρχουσα διαδικασία μπορεί επίσης να δημιουργήσει μια διαδικασία, η οποία αναφέρεται έπειτα ως διαδικασία "παιδιών". Η διαδικασία που την δημιούργησε ονομάζεται γονική διαδικασία

1.26.4 Ποιες οι πιθανές καταστάσεις μιας διαδικασίας- Διεργασίας

Υπάρχουν πέντε πιθανές καταστάσεις στις οποίες μια διαδικασία μπορεί να βρίσκεται νέα-καινούργια, σε λειτουργία, σε αναμονή, σε ετοιμότητα, ή ολοκληρωμένη. Αυτά τα στάδια περιγράφονται παρακάτω:

- Νέα - Η διαδικασία διαμορφώνεται.
- Σε λειτουργία - Οι οδηγίες από τη διαδικασία εκτελούνται.
- Σε αναμονή - Η διαδικασία περιμένει ένα γεγονός να συμβεί προτού να μπορέσει να επαναλάβει – να συνεχίσει την εκτέλεσή της.
- Σε ετοιμότητα - Η διαδικασία είναι έτοιμη να οργανωθεί, αλλά περιμένει τον επεξεργαστή

- Ολοκληρωμένη - Η διαδικασία έχει ολοκληρώσει την εκτέλεσή της.

1.26.5 Τι είναι ο φραγμός ελέγχου διεργασίας (Process Control Block)

Ο φραγμός ελέγχου διεργασίας (PCB) αποθηκεύει τις πληροφορίες σχετικά με μια συγκεκριμένη διαδικασία. Αποθηκεύει τα στάδια - βήματα διαδικασίας, το πρόγραμμα , τις εγγραφές, τις πληροφορίες του σχεδιασμού της Κεντρικής Μονάδας Επεξεργασίας, πληροφορίες διαχείρισης της μνήμης, τις λογιστικές πληροφορίες, και τις I/O πληροφορίες θέσης. Ο φραγμός ελέγχου διεργασίας (PCB) λέει στην ΚΜΕ ποιες τιμές να "φορτώσει" στους καταλόγους του όποτε δίνεται η διαδικασία, καθώς επίσης και από πού να αρχίσει τη διαδικασία. Συνολικά, ο φραγμός ελέγχου διεργασίας (PCB) χρησιμεύει ως τράπεζα στοιχείων που αποθηκεύει τις σημαντικές πληροφορίες για κάθε διαδικασία.

1.26.6 Χρονοπρογραμματιστές (Schedulers)

Ο Χρονοπρογραμματιστής ΚΜΕ επιλέγει μεταξύ των διεργασιών που έχουν είναι έτοιμες για εκτέλεση, και αναθέτει την ΚΜΕ σε μια από αυτές

Υπάρχουν δυο κατηγορίες χρονοπρογραμματισμών :

Μακροπρόθεσμος Χρονοπρογραμματιστής (Long-term scheduler ή job scheduler) – επιλέγει (από την ουρά εισόδου) ποιες διεργασίες θα εισέλθουν στην ουρά έτοιμων διεργασιών

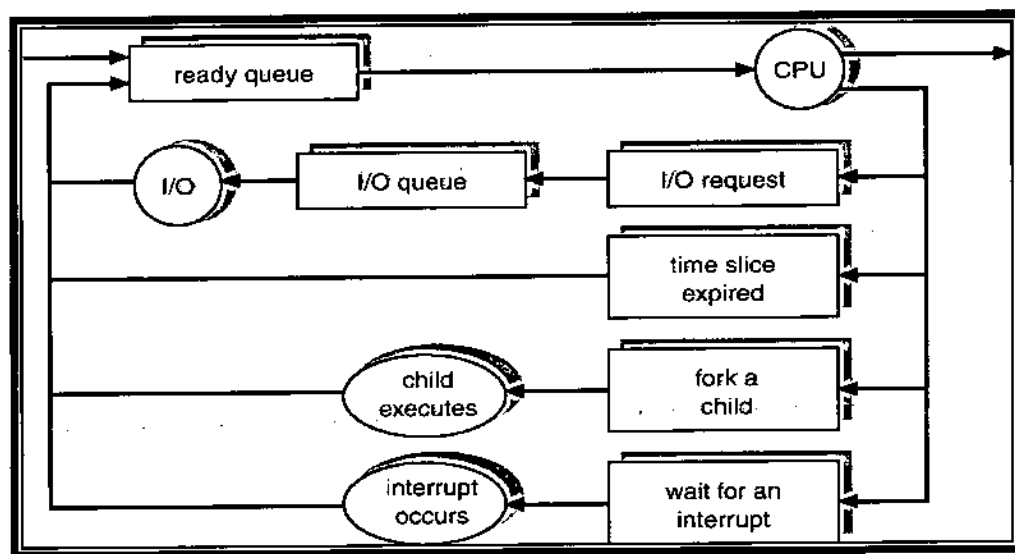
Βραχυπρόθεσμος Χρονοπρογραμματιστής (Short-term scheduler ή CPU scheduler) – επιλέγει τη διεργασία η οποία θα εκτελεστεί και την «αναθέτει» (allocates) την ΚΜΕ

Ο βραχυπρόθεσμος χρονοπρογραμματιστής καλείται πολύ συχνά (milliseconds) (πρέπει να είναι γρήγορος) ενώ ο μακροπρόθεσμος χρονοπρογραμματιστής καλείται αραιά. Ο μακροπρόθεσμος χρονοπρογραμματιστής ελέγχει το βαθμό του πολυπρογραμματισμού

Οι διεργασίες χαρακτηρίζονται ως:

Διεργασίες I/O-bound – ξοδεύουν περισσότερο χρόνο κάνοντας I/O, με λίγα και μικρής διάρκειας ξεσπάσματα ΚΜΕ (CPU bursts)

Διεργασίες CPU-bound – ξοδεύουν περισσότερο χρόνο σε υπολογισμούς, με λίγα και πολύ μεγάλα ξεσπάσματα ΚΜΕ



Σχήμα 11: Αναπαράσταση του Χρονοπρογραμματισμού των διεργασιών (7)

1.26.7 Εναλλαγή Περιβάλλοντος Λειτουργίας (Context Switch)

Όταν η ΚΜΕ «δίνεται» σε μια άλλη διεργασία, το Λ.Σ. πρέπει να σώσει την κατάσταση της διεργασίας που αντικαθίσταται και να φορτώσει την κατάσταση της διεργασίας που την αντικαθιστά

Η διαδικασία αυτή ονομάζεται εναλλαγή περιβάλλοντος λειτουργίας (context switch). Ο χρόνος που απαιτείται για την εκτέλεση της λειτουργίας αυτής είναι καθαρός φόρτος (overhead) για το σύστημα (το οποίο όσο ασχολείται με την εναλλαγή δεν κάνει χρήσιμη δουλειά για τις διεργασίες των χρηστών του)

Ο χρόνος αυτός εξαρτάται από την πολυπλοκότητα του ΛΣ και την υποστήριξη από το υλικό.

1.26.8 Κριτήρια του χρονοπρογραμματισμού

- Χρησιμοποίηση της ΚΜΕ (CPU utilization): Η ΚΜΕ πρέπει να είναι όσο το δυνατόν περισσότερο απασχολημένη
- Ρυθμός διεκπεραίωσης (throughput): Πλήθος διεργασιών που ολοκληρώνουν την εκτέλεσή τους στη μονάδα του χρόνου
- Χρόνος ολοκλήρωσης (turnaround time): Ο χρόνος που μεσολαβεί από την υποβολή ως την ολοκλήρωση της διεργασίας
- Χρόνος αναμονής (waiting time): Ο χρόνος που μια διεργασία περιμένει στην ουρά έτοιμων διεργασιών
- Χρόνος απόκρισης (response time): Ο χρόνος που απαιτείται από την υποβολή μιας αίτησης που αφορά μια διεργασία μέχρι να δρομολογηθεί η διεργασία στην ΚΜΕ (όχι όμως και να παράγει κάποιο «ορατό» στο χρήστη αποτέλεσμα)

Τα παραπάνω κριτήρια του χρονοπρογραμματισμού έχουν σαν σκοπό προγραμματίσουν με τέτοιο τρόπο το λογισμικό σύστημα του συστήματος ώστε να μπορούν να επιτευχθούν τα ακόλουθα:

- Μεγιστοποίηση της χρησιμοποίησης της ΚΜΕ
- Μεγιστοποίηση του ρυθμού διεκπεραίωσης
- Ελαχιστοποίηση του χρόνου ολοκλήρωσης
- Ελαχιστοποίηση του χρόνου αναμονής
- Ελαχιστοποίηση του χρόνου απόκρισης
- Η βελτιστοποίηση όλων των παραμέτρων απαιτεί συμβιβασμούς

1.27 Έννοιες του σχεδιασμού Κεντρικής Μονάδας Επεξεργασίας

1.27.1 Bursts (Εκρήξεις)

Μια διαδικασία εκτέλεσης εναλλάσσεται μεταξύ δύο κύκλων: Της εκτέλεσης της ΚΜΕ και της αναμονή κατάστασης I/O. Το χρονικό διάστημα αυτό που χρειάζεται για κάθε εκτέλεση ΚΜΕ που ολοκληρώνεται καλείται έκρηξη ΚΜΕ, και ο χρόνος που η διαδικασία ξοδεύει στην αναμονή κατάστασης I/O για το αίτημα που εκπληρώνεται λέγεται I/O έκρηξη. Τα ιστορικά τα στοιχεία για τους χρόνους έκρηξης μιας διαδικασίας λαμβάνονται υπόψη κατά επιλογή μιας ΚΜΕ κατά τον σχεδιασμό του αλγόριθμου. Η σημασία αυτού είναι να αποτραπεί μια διαδικασία με έναν πολύ μακροχρόνιο χρόνο έκρηξης από τη λήψη του ελέγχου της ΚΜΕ προτού να τρέξει μια διαδικασία με σύντομο χρόνο έκρηξης .(8)

1.27.2 Preemptive Scheduling

Ένας **Preemptive Scheduling** αλγόριθμος σχεδιασμού έχει τη δύναμη να πάρει τον έλεγχο της ΚΜΕ μακριά από μια διαδικασία. Αυτό γίνεται όταν υπερβεί η διαδικασία που τρέχει αυτήν την περίοδο είτε υπερβεί το διανεμημένο χρόνο χρήση της ΚΜΕ, ή εάν σε μια άλλη διαδικασία έχει δοθεί πιο υψηλή προτεραιότητα από την τρέχουσα διαδικασία. Ο **Preemptive Scheduling** σχεδιασμός είναι μια μέθοδος που μπορεί να ενσωματωθεί σε διάφορους διαφορετικούς αλγορίθμους σχεδιασμού και μπορεί να αποτρέψει μια ενιαία διαδικασία από το να εμπλέξει την ΚΜΕ.

1.27.3 Dispatcher (αποστολέας)

Ο **Dispatcher** είναι αρμόδιος για την εκτέλεση των διακοπών πλαισίου. Αυτό σημαίνει ότι δίνει τον έλεγχο της ΚΜΕ στη διαδικασία που πρόκειται να οργανωθεί έπειτα. Κατ' αρχάς, σώζει την κατάσταση της τρέχουσας διαδικασίας. Έπειτα, η κατάσταση της επιλεγμένης διαδικασίας φορτώνεται στους καταλόγους. Τέλος, η διαδικασία αρχίζει στη θέση που διευκρινίζεται από το μετρητή προγράμματος.

— Το τμήμα του ΛΣ που δίνει τον έλεγχο της ΚΜΕ στη διεργασία που έχει επιλεγεί από τον χρονοπρογραμματιστή.

- Αυτή η διαδικασία περιλαμβάνει:
 - εναλλαγή περιβάλλοντος λειτουργίας
 - εναλλαγή σε τρόπο λειτουργίας χρήστη
 - μετάβαση στην κατάλληλη θέση του κώδικα χρήστη έτσι ώστε να γίνει επανεκκίνηση του προγράμματος

— *Καθυστέρηση του αποστολέα (dispatch latency)*: Ο χρόνος που απαιτείται για τη απενεργοποίηση της παλιάς διεργασίας και την ενεργοποίηση της νέας.(8)

1.28 Βασικοί αλγόριθμοι Χρονοπρογραμματισμού

1.28.1 Αλγόριθμος First-Come, First-Served

Αυτός ο αλγόριθμος (FCFS) είναι ο πιο δημοφιλής αλγόριθμος προγραμματισμού. Διαθέτει απλά την ΚΜΕ στη διαδικασία που το ζήτησε πρώτα. Όλες οι άλλες διαδικασίες που ζητούν την ΚΜΕ τοποθετούνται στην έτοιμη σειρά αναμονής και εξυπηρετούνται ανάλογα με την σειρά προτεραιότητας της διαταγής εκτέλεσης που στέλνουν. Αυτός ο αλγόριθμος είναι non-preemptive μόλις δοθεί σε μια διαδικασία ο έλεγχος της ΚΜΕ θα τον κρατήσει έως ότου τον απελευθερώνει η διαδικασία. Μια διαδικασία θα απελευθερώσει την ΚΜΕ είτε όταν τελειώσει την εκτέλεση της διεργασίας , ή εάν πρέπει να περιμένει μια διαδικασία I/O να εμφανιστεί προτού να μπορέσει να επαναλάβει την εκτέλεσή της.

1.28.2 Αλγόριθμος Shortest Job First – SJF

Ο αλγόριθμος αυτός (SJF), εξετάζει το χρονικό διάστημα που κάθε διαδικασία αναμένεται να χρησιμοποιήσει κατά τη διάρκεια της επόμενης έκρηξης της ΚΜΕ. Χρησιμοποιεί αυτές τις πληροφορίες για να διαμορφώσει μια σειρά αναμονής, και δίνει έπειτα τον έλεγχο της ΚΜΕ στη διαδικασία που αναμένεται να έχει στην πιο σύντομη επόμενη έκρηξη ΚΜΕ. Θεωρητικά, αυτός ο αλγόριθμος είναι πολύ αποδοτικός, επειδή ελαχιστοποιεί το μέσο χρόνο αναμονής για κάθε διαδικασία. Εντούτοις, δεδομένου ότι η επόμενη περίοδος έκρηξης ΚΜΕ μπορεί να είναι πολύ δύσκολο να καθοριστεί ακριβώς, αυτός ο αλγόριθμος, δεν δίνει πάντα, τα αναμενόμενα αποτελέσματα.

Χαρακτηριστικά του αλγορίθμου Shortest Job First – SJF είναι:

- Συσχετίζουμε με κάθε διεργασία το μήκος του επόμενου ξεσπάσματος στην ΚΜΕ, και επιλέγεται πάντα η διεργασία με το μικρότερο χρόνο
- Non-preemptive: Από τη στιγμή που θα δοθεί η ΚΜΕ σε μια διεργασία, αυτή η διεργασία δεν μπορεί να διακοπεί η επιλογή της επόμενης διεργασίας γίνεται όταν τελειώσει η τρέχουσα διεργασία
- Preemptive: Αν μια διεργασία εισέλθει στο σύστημα με μήκος ξεσπάσματος μικρότερο από αυτό που απομένει στην τρέχουσα διεργασία, τότε γίνεται διακοπή. Το σχήμα αυτό είναι γνωστό ως «Πρώτα η Εργασία με το Λιγότερο Υπολειπόμενο Χρόνο» (Shortest-Remaining-Time-First – SRTF)
- SJF είναι βέλτιστος: Επιτυγχάνει ελάχιστο μέσο χρόνο αναμονής
- Το κλειδί της επιτυχίας είναι να γνωρίζει κανείς το μήκος του (επόμενου) ξεσπάσματος κάθε διεργασίας στο σύστημα
- Αυτός ο χρόνος μπορεί μόνο να προσεγγιστεί

1.28.3 Αλγόριθμος Προτεραιότητας

Ο αλγόριθμος προτεραιότητας ορίζει σε κάθε διαδικασία ένα επίπεδο προτεραιότητας, το οποίο αντιπροσωπεύεται από έναν αριθμό. Μερικά συστήματα χρησιμοποιούν χαμηλούς αριθμούς για να έχουν υψηλές προτεραιότητες, και άλλοι υποδείχνουν τους υψηλούς αριθμούς για να κερδίσουν μια πιο υψηλή προτεραιότητα. Αυτές οι προτεραιότητες μπορούν να οριστούν εσωτερικά (από το λειτουργικό σύστημα) ή εξωτερικά (από το χρήστη). Η εξέταση κάποιου είδους μετρήσιμης ποσότητας, όπως πόσοι πόροι κάθε διαδικασία απαιτεί, θέτει συνήθως τα εσωτερικά επίπεδα προτεραιότητας. Τα εξωτερικά επίπεδα προτεραιότητας απεικονίζουν μια καθορισμένη ως προς τον χρήστη διαταγή σπουδαιότητας για κάθε διαδικασία. Η διαδικασία με την πιο υψηλή προτεραιότητα παίρνει αρχικά τον έλεγχο της ΚΜΕ.

Εάν ο αλγόριθμος είναι preemptive (προαγοραστικός), η τρέχουσα λειτουργία της διαδικασίας θα πραγματοποιηθεί αμέσως μόλις ολοκληρωθεί η διαδικασία στην έτοιμη σειρά αναμονής με πιο υψηλή προτεραιότητα. Εντούτοις, εάν ο αλγόριθμος είναι non-preemptive, μόλις αρχίσει η διαδικασία το "τρέξιμο" όλων των άλλων διαδικασιών θα πρέπει να περιμένει έως ότου απελευθερωθεί η ΚΜΕ, ανεξάρτητα από αυτό που τα επίπεδα προτεραιότητάς τους μπορούν να είναι. Η σημαντικότερη ανησυχία με το σχεδιασμό προτεραιότητας είναι η έννοια της "πείνας-τροφοδοσίας". Αυτό εμφανίζεται όταν μια διαδικασία έχει τόσο χαμηλή προτεραιότητα ώστε δεν δίνεται ποτέ ο έλεγχος της ΚΜΕ. Μια λύση σε αυτό είναι η βαθμιαία αύξηση της προτεραιότητας μιας διαδικασίας κατά τη διάρκεια του χρόνου.

1.28.4 Αλγόριθμος Round-Robin

Ο Round-Robin αλγόριθμος (RR) σχεδιασμού καθορίζει ένα χρονικό κβάντο, το οποίο είναι το χρονικό διάστημα που κάθε διαδικασία στη σειρά αναμονής χρησιμοποιεί τη ΚΜΕ. Αυτός ο αλγόριθμος είναι preemptive (προαγοραστικός) και γεμίζει τη σειρά αναμονής που χρησιμοποιεί με τη μέθοδο FCFS. Αφότου έχει υπερβεί μια διαδικασία το χρονικό όριό της, κινείται προς το τέλος της σειράς αναμονής και στην επόμενη διαδικασία στη σειρά αναμονής δίνεται ο έλεγχος της ΚΜΕ. Ο καθορισμός ενός σωστού χρονικού κβάντου είναι κρίσιμος για την απόδοση του αλγορίθμου RR.

Εάν το χρονικό κβάντο είναι πάρα πολύ μικρό, τότε λίγες διαδικασίες θα ολοκληρωθούν τον πρώτο χρόνο που οργανώνονται και πολύς χρόνος θα ξοδευτεί στην συνέχεια εκτελώντας τους διακόπτες πλαισίου. Εναλλακτικά, εάν το χρονικό κβάντο είναι πάρα πολύ μεγάλο, οι περισσότερες διαδικασίες θα τελειώσουν και η ΚΜΕ θα σταματήσει να απασχολείται για το υπόλοιπο του κβάντου. Και στις δύο περιπτώσεις, η ΚΜΕ δεν χρησιμοποιείται αποτελεσματικά. Γενικά, είναι επιθυμητό περίπου το 80 τοις εκατό των διαδικασιών να ολοκληρωθεί προτού να επιτευχθεί το χρονικό κβάντο.

Συνοπτικά με τον αλγόριθμο **Round-Robin** έχουμε τα εξής:

- Κάθε διεργασία λαμβάνει ένα μικρό κλάσμα του χρόνου της ΚΜΕ (μονάδα χρόνου - *time quantum*), συνήθως 10-100 milliseconds
- Μόλις παρέλθει αυτό το χρονικό διάστημα, η διεργασία διακόπτεται και προστίθεται στο τέλος της ουράς έτοιμων διεργασιών
- Αν υπάρχουν n διεργασίες στην ουρά έτοιμων διεργασιών και η μονάδα χρόνου είναι q , τότε κάθε διεργασία θα εκτελείται για το $1/n$ του χρόνου της ΚΜΕ σε τεμάχια μονάδων χρόνου διάρκειας q . Καμιά διεργασία δεν περιμένει περισσότερο από χρόνο $(n-1)q$
- Απόδοση:
 - πολύ μεγάλο $q \Rightarrow$ ο αλγόριθμος μετατρέπεται σε FCFS
 - πολύ μικρό $q \Rightarrow$ το σύστημα αναλώνεται σε context switches

1.28.5 Αλγόριθμος Multilevel Queue (Πολλαπλής στάθμης σειρά αναμονής)

- Η ουρά έτοιμων διεργασιών χωρίζεται σε δύο διαφορετικές ουρές: διεργασίες που τρέχουν στο προσκήνιο (interactive) ή στο παρασκήνιο (batch)
- Κάθε ουρά έχει το δικό της αλγόριθμο χρονοπρογραμματισμού, foreground – RR, background – FCFS

- Χρονοπρογραμματισμός πρέπει να γίνεται και μεταξύ των ουρών
 - Καθορισμένος χρονοπρογραμματισμός με προτεραιότητες (π.χ., πρώτα εξυπηρέτηση του προσκήνιου). Πιθανότητα λιμοκτονίας
 - Μονάδα χρόνου – κάθε ουρά λαμβάνει ένα συγκεκριμένο ποσοστό του χρόνου της ΚΜΕ, το οποίο μπορεί να διατεθεί με όποιο τρόπο μεταξύ των διεργασιών που υπάρχουν στην ουρά, π.χ., 80% στο προσκήνιο με RR 20% στο παρασκήνιο με FCF

1.28.6 Πολυεπίπεδες Ουρές με Ανατροφοδότηση (Multilevel Feedback Queues)

- Μια διεργασία μπορεί να μετακινείται μεταξύ των διάφορων ουρών. Τρόπος υλοποίησης της γήρανσης (aging)
- Ένας χρονοπρογραμματιστής πολυεπίπεδων ουρών προσδιορίζεται από τις ακόλουθες παραμέτρους:
 - πλήθος ουρών
 - αλγόριθμο χρονοπρογραμματισμού για την κάθε ουρά
 - μέθοδος αναβάθμισης διεργασιών
 - μέθοδος υποβάθμισης διεργασιών
 - μέθοδος προσδιορισμού της ουράς στην οποία πρέπει να εισέλθει μια διεργασία όταν χρειάζεται εξυπηρέτηση.

1.29 Κριτήρια επιλογής ενός αλγορίθμου χρονοπρογραμματισμού

Πολλοί διαφορετικοί αλγόριθμοι υπάρχουν, και μπορούν να αναμιχθούν για να δημιουργήσουν και άλλους αλγορίθμους. Κάθε ένας από αυτούς έχει τα πλεονεκτήματα και τα μειονεκτήματα του. Επομένως, οι αλγόριθμοι πρέπει να επιλεχτούν και να είναι βασισμένοι σε ορισμένες επιθυμητές ιδιότητες. Οι ιδιότητες οι οποίες παρατηρούνται συνηθέστερα παρατίθενται παρακάτω:

- Χρήση της ΚΜΕ - αυτό μπορεί να κυμανθεί από 0 έως 100 τοις εκατό. Είναι επιθυμητό να υπάρξει ένα υψηλότερο ποσοστό της χρησιμοποίησης της ΚΜΕ, επειδή δείχνει ότι το υλικό δεν σπαταλιέται.
- Ρυθμός απόδοσης - ο αριθμός διαδικασιών που ολοκληρώνονται ανά μονάδα του χρόνου είναι η ρυθμός απόδοσης. Αυτό είναι μια μέτρηση της εργασίας που γίνεται από την ΚΜΕ. Είναι όχι πάντα ένας καλός δείκτης, εντούτοις, επειδή εξαρτάται άμεσα από το μήκος των διαδικασιών που εκτελούνται.
- Χρόνος ανακύκλωσης - αυτό μετρά το χρονικό διάστημα που χρησιμοποιείται μεταξύ της υποβολής μιας διαδικασίας και του χρόνου μέσα στον που χρειάζεται η διαδικασία για να ολοκληρώνει την εκτέλεσή της. Μετριέται για κάθε διαδικασία και λαμβάνει υπόψη όλη την ώρα που η διαδικασία ξοδεύει στην αναμονή και στην εκτέλεση.
- Χρόνος αναμονής - αυτό είναι ένα καλό μέτρο για το πόσο αποτελεσματικά ο αλγόριθμος σχεδιασμού διαθέτει την ΚΜΕ στις έτοιμες διαδικασίες. Μετρά το ποσό των χρονικών διαστημάτων που μια διαδικασία ξοδεύει περιμένοντας στην έτοιμη σειρά αναμονής.
- Χρόνος απόκρισης - αυτό μετριέται μόνο στα διαλογικά συστήματα, και είναι το χρονικό διάστημα που χρειάζεται στην έναρξη απόκρισης ενός αιτήματος. Η ταχύτητα της συσκευής παραγωγής περιορίζει συνήθως το χρόνο απόκρισης.

1.30 Πρόσθετοι αλγόριθμοι χρονοπρογραμματισμού

Εκτός από του βασικούς αλγορίθμους χρονοπρογραμματισμού που αναφέρθηκαν παραπάνω έχουμε και κάποιους άλλους οι οποίοι δεν είναι τόσο γνωστοί. Μερικοί από αυτούς αναφέρονται παρακάτω.

1.30.1 Αλγόριθμος σταθερής προτεραιότητας Fixed Priority

Σε έναν αλγόριθμο σταθερής προτεραιότητας, σε κάθε διαδικασία ορίζεται αρχικά ένα επίπεδο προτεραιότητας. Αυτά τα επίπεδα είναι μόνιμα και δεν μπορούν ποτέ να αλλάξουν. Οι σταθεροί αλγόριθμοι προτεραιότητας είναι εύκολοι στην χρήση και την κατανόηση εντούτοις παρουσιάζουν το εξής πρόβλημα. Μια διαδικασία με χαμηλή προτεραιότητα πάντα θα έχει πάντα χαμηλή προτεραιότητα και δεν θα τρέξει ποτέ μέχρι όλες οι διαδικασίες με την πιο υψηλή προτεραιότητα να ολοκληρωθούν.

1.30.2 Αλγόριθμος μονοτονικού ποσοστού Rate Monotonic

Η προτεραιότητα ορίζεται σε διαδικασίες βασισμένες στα μήκη των περιόδων εκτέλεσής τους. Στις διαδικασίες με τις μικρότερες χρονικές περιόδους δίνεται πιο υψηλή προτεραιότητα από εκείνες με τις πιο μεγάλες περιόδους. Ένα σύστημα που χρησιμοποιεί το μονοτονικό αλγόριθμο ποσοστού θα δεχτεί μόνο μια διαδικασία εάν ο χρόνος μέσα στον οποίο πρέπει να εκτελεστεί και η περίοδος της διαδικασίας είναι ο ίδιος. Η χειρότερη περίπτωση σχεδιασμού που δεσμεύεται από τον μονοτονικό αλγόριθμο ποσοστού αντιπροσωπεύεται από τον τύπο:

$$\underline{Wn = n*(2(1/n) - 1)}$$

όπου το N είναι ο αριθμός στόχων στο σύστημα. Οποτεδήποτε η συνολική ζητούμενη χρησιμοποίηση είναι λιγότερο ή ίσο προς Wn , το σύστημα θα είναι σε θέση να ολοκληρώσει όλες τις υποχρεώσεις σύμφωνα με τις αντίστοιχες προθεσμίες.

Το μεγαλύτερο πρόβλημα με το Rate Monotonic algorithm είναι η έλλειψη χρησιμοποίησης της ΚΜΕ. Οποτε περισσότεροι από ένας στόχοι είναι παρόντες στο σύστημα, ο συνδεδεμένος σχεδιασμένος χειρότερης περίπτωσης είναι πάντα

λιγότερο από 100 τοις εκατό. Αυτό σημαίνει ότι η ΚΜΕ δεν μπορεί ποτέ να χρησιμοποιηθεί εκατό τοις εκατό για να ολοκληρώσει όλες τις υποχρεώσεις μέχρι τις προθεσμίες τους. με . Παρά αυτά τα μειονεκτήματα, ο Rate Monotonic algorithm είναι βέλτιστος μεταξύ των αλγορίθμων σταθερής προτεραιότητας σχεδιασμού.

1.30.3 Rate Monotonic with Delayed Preemption (Αλγόριθμος μονοτονικού ποσοστού με καθυστερημένο δικαίωμα προτίμησης)

Ένα σημαντικό πρόβλημα με το Rate Monotonic algorithm είναι ότι δεν είναι αρκετά δυνατό να αντιμετωπιστεί ένας μεγάλος αριθμός αλλαγών πλαισίου, ο οποίος μπορεί να είναι πολύ ακριβός. Το σημείο του **Rate Monotonic with Delayed Preemption** είναι να επιτραπεί μια τρέχουσα διαδικασία για να καθυστερήσει την χρησιμοποίηση της ΚΜΕ σε μια διαδικασία με μια πιο υψηλή προτεραιότητα, μειώνοντας κατά συνέπεια τον αριθμό αλλαγών πλαισίου.

1.30.4 Deadline Monotonic (Αλγόριθμος μονοτονικής προθεσμίας)

Το χρονικό διάστημα μεταξύ της έναρξης της περιόδου και της προθεσμίας της διαδικασίας καθορίζει την προτεραιότητα στο **Deadline Monotonic** αλγόριθμο (DM) προθεσμίας. Στις διαδικασίες με τα πιο σύντομα χρονικά διαστήματα δίνονται πιο υψηλές προτεραιότητες. Τα συστήματα που χρησιμοποιούν τον αλγόριθμο αυτό επιβάλλουν τον περιορισμό ότι κάθε προθεσμία διαδικασιών πρέπει να είναι λιγότερη ή ίση προς την περίοδό της.

1.30.5 Αλγόριθμος Dynamic Priority (δυναμικής προτεραιότητας)

Στους **Dynamic Priority** αλγορίθμους η προτεραιότητα μιας διαδικασίας μπορεί να αλλάξει. Αυτό καθιστά τον αλγόριθμο πιο σύνθετο, αλλά και σημαντικά ισχυρότερο. Μπορεί εάν μια διαδικασία είναι στη σειρά αναμονής για ένα καθορισμένο χρονικό διάστημα και δεν έχει λάβει τον έλεγχο της ΚΜΕ, κατόπιν η προτεραιότητά της αυξάνεται. Οι **Dynamic Priority** αλγόριθμοι είναι πολύ χρήσιμοι στις σε πραγματικό χρόνο εφαρμογές. Εάν μια διαδικασία χρειάζεται να τελειώσει και η προθεσμία της πλησιάζει, η προτεραιότητά της μπορεί να αυξηθεί έτσι ώστε να είναι σε θέση να ολοκληρωθεί εγκαίρως. Το κόστος που αναλαμβάνεται με τη

χρησιμοποίηση ενός αλγόριθμου παρά ενός **Dynamic Priority** είναι μια αύξηση στο ποσό χρόνου εκτέλεσης.

Ένας δυναμικός αλγόριθμος θα είναι σε θέση να τελειώσει τη συγκεκριμένη διαδικασία στον καθορισμένο χρόνο εάν οι απαιτήσεις για συνολική χρησιμοποίηση της ΚΜΕ είναι λιγότερο από 100 τοις εκατό. Οι δυναμικοί αλγόριθμοι μπορούν πολύ δύσκολα να προβλέψουν τότε το σύστημα δοκιμάζει έναν πολύ βαρύ φόρτο εργασίας.

1.30.6 Αλγόριθμος Earliest Deadline First (πρόωρης προθεσμίας)

Σύμφωνα με τον **Earliest Deadline First** η προτεραιότητα ορίζεται με βάση ποια διαδικασία βρίσκεται κοντά στην λήξη της προθεσμίας της για την εκτέλεσή της. Η διαδικασία που έχει το πιο μικρό χρονικό διάστημα που μέχρι να λήξει η προθεσμία της παίρνει και την πιο υψηλή προτεραιότητα. Ο αλγόριθμος αυτός είναι preemptive algorithm και δίνει την δυνατότητα στην διεργασία με την υψηλότερη προτεραιότητα να ελέγχεται από την ΚΜΕ. Από την στιγμή που οι προτεραιότητες αλλάζουν, η τρέχουσα διαδικασία μεταβάλλεται σε νέα με υψηλότερη προτεραιότητα. Αυτός ο αλγόριθμος θα ολοκληρώσει όλες τις διεργασίες εγκαίρως όταν οι προθεσμίες αυτών είναι κοντινές μεταξύ τους , και η συνολική χρησιμοποίηση της ΚΜΕ δεν είναι μεγαλύτερη από 100 τοις εκατό.

1.31 Real-Time Scheduling (Προγραμματισμός πραγματικού χρόνου)

1.31.1 Hard Real Time (Δυσμετάβλητος πραγματικός χρόνος)

Ένα **Hard Real Time** σύστημα εγγυάται ότι μια κρίσιμη υποχρέωση θα ολοκληρωθεί μέσα σε καθορισμένο χρονικό διάστημα "A". Κάθε διαδικασία και ο τρέχοντας χρόνος της παρουσιάζονται το σύστημα, και αυτό καθορίζει εάν είναι ή όχι δυνατό να ολοκληρωθεί η διαδικασία στο αναμενόμενο χρονικό διάστημα. Για να γίνει αυτό, το σύστημα εξετάζει τον χρόνο που θα χρειαστεί για να τρέξει κάθε μιας από τις διάφορες λειτουργίες που χρησιμοποιεί η διαδικασία και το συγκρίνει με το χρόνο τρεξίματος της διαδικασίας. Εάν είναι δυνατό να ολοκληρωθεί η διεργασία, το σύστημα το δέχεται. Διαφορετικά, κρίνεται αδύνατο και απορρίπτεται από το σύστημα. Η μέθοδος που περιγράφεται πιο πάνω προορίζεται για ένα **Hard Real**

Time σύστημα που αποτελείται από ειδικής χρήσης λογισμικό που τρέχει στο υλικό που αφιερώνεται σε μια κρίσιμη διαδικασία. Αυτά τα ειδικευμένα συστήματα στερούνται την πλήρη λειτουργία που προσφέρεται από τους σύγχρονους υπολογιστές και τα λειτουργικά συστήματα.

Οι **Hard Real Time** εγγυήσεις δεν μπορούν να γίνουν σε συστήματα με δευτερεύουσα αποθήκευση ή εικονική μνήμη. Υπάρχει μια απρόβλεπτη διαφορά στο χρονικό διάστημα που θα χρειαστεί για να εκτελεστεί μια ιδιαίτερη διαδικασία σε ένα από αυτά τα συστήματα. Μπορεί να χρειαστεί περισσότερος χρόνος για να εκτελεστεί μια διαδικασία σε μια μηχανή με την εικονική μνήμη εάν ο αποτελεσματικός χρόνος πρόσβασης είναι υψηλός. Όποτε όταν ένα ελάττωμα εμφανίζεται στις σελίδες, ο χρόνος πρόσβασης αυξάνεται. Αυτό παρουσιάζεται ως εξής:

Αποτελεσματικός χρόνος πρόσβασης = $(1 - \pi) * \mu_a + \pi (1 - \pi) * (\text{χρόνος ελαττωμάτων σελίδων}),$

Effective access time = $(1 - p) * m_a + p * (\text{page fault time}),$

όπου το π είναι η πιθανότητα ενός ελαττώματος σελίδων και ενός μ_a είναι ο χρόνος πρόσβασης μνήμης.

1.31.2 Soft Real Time (Ευμετάβλητος πραγματικός χρόνος)

Ένα **Soft Real Time** σύστημα προσπαθεί να ολοκληρώσει τις υποχρεώσεις του μέσα στις αντίστοιχες προθεσμίες, αλλά δεν εγγυάται ότι αυτό θα συμβεί. Οι κρίσιμες διαδικασίες λαμβάνουν υψηλότερη προτεραιότητα σε σχέση με εκείνες τις διαδικασίες που δεν είναι κρίσιμες. Σε ένα time-sharing σύστημα (σύστημα χρονικής κατανομής) με **Soft Real Time** ικανότητες, εκείνες οι διαδικασίες που είναι σε πραγματικό χρόνο πρέπει να δοθεί πιο υψηλή προτεραιότητα και αυτή η προτεραιότητα δεν πρέπει να υποβιβαστεί κατά τη διάρκεια του χρόνου. Εντούτοις, η προτεραιότητα των διαδικασιών που δεν είναι σε πραγματικό χρόνο στο σύστημα μπορεί να μειωθεί κατά τη διάρκεια του χρόνου. Αυτό μπορεί να προκαλέσει μακροχρόνιες καθυστερήσεις για αυτές τις διαδικασίες. Η λανθάνουσα κατάσταση αποστολών σε ένα **Soft Real Time** σύστημα πρέπει να είναι όσο το δυνατόν μικρότερη. Ένας μικρός χρόνος διακοπής θα παράσχει περισσότερο χρόνο για κάθε

μία διαδικασία που τρέχει. Οι χρόνοι λανθάνουσας κατάστασης αποστολών μπορούν να γίνουν πολύ μακροχρόνιοι σε ένα λειτουργικό σύστημα που απαιτεί μια κλήση συστημάτων για να ολοκληρωθεί ή ένας I/O φραγμός για να εμφανιστεί πριν εκτελέσει έναν διακόπτη πλαισίου. Ένας τρόπος να λυθεί αυτό το ζήτημα είναι καταστήσουμε τις κλήσεις συστημάτων preemptible, με την παρεμβολή σημείων προτίμησης στις κλήσεις συστημάτων. Ένα σημείο δικαιώματος προτίμησης είναι μια θέση στον κώδικα που ακόμα και αν υπάρξει διακοπή σε αυτόδιακοπή της διαδικασίας αυτό το σημείο δεν θα βλάψει τα στοιχεία του. Οποτε ένα σημείο δικαιώματος προτίμησης επιτυγχάνεται, το σύστημα ελέγχει για να δει εάν μια διαδικασία υψηλής προτεραιότητας πρέπει να οργανωθεί. Εάν ισχύει, ο έλεγχος της ΚΜΕ δίνεται στη διαδικασία. Μόλις τελειώσει, η ΚΜΕ επιστρέφει στο αρχικό δικαίωμα προτίμησης και συνεχίζει την εκτέλεση. Η αντιστροφή προτεραιότητας εμφανίζεται σε ένα "μαλακό" (ευμετάβλητο) σε πραγματικό χρόνο σύστημα όταν μια ή περισσότερες διαδικασίες χρησιμοποιεί τους πόρους που απαιτούνται για μια διαδικασία υψηλής προτεραιότητας. Σε αυτήν την κατάσταση, στις χαμηλής προτεραιότητας διαδικασίες δίνεται προτεραιότητα ίση με τη διαδικασία προτεραιότητας έως ότου χρησιμοποιήσουν τους επιθυμητούς πόρους. Μόλις απελευθερωθεί ο πόρος, σε κάθε διαδικασία δίνεται το προηγούμενο επίπεδο προτεραιότητάς. Αυτή η τεχνική είναι γνωστή ως πρωτόκολλο προτεραιότητας-κληρονομιάς, και αυτό αποτρέπει τις διαδικασίες υψηλής προτεραιότητας να "ξοδεύουν" πολύ χρόνο να περιμένουν τους διαθέσιμους πόρους.

1.32 Χρονοπρογραμματισμός Πραγματικού Χρόνου (Real-Time Scheduling)

— Αυστηρά συστήματα πραγματικού χρόνου (hard real-time): Πρέπει να είναι σε θέση να ολοκληρώσουν μια εργασία μέσα σε αυστηρά καθορισμένα χρονικά περιθώρια (δέσμευση πόρων - resource reservation).

➤ Δύσκολα υλοποιήσιμα. Συνήθως οι ακριβής φύση των διεργασιών πρέπει να είναι γνωστή εκ των προτέρων

— Χαλαρά συστήματα πραγματικού χρόνου (soft real-time): Πρέπει να δίνεται προτεραιότητα στις κρίσιμες διεργασίες έναντι όλων των υπολοίπων.

➤ Δημιουργεί άδικη ανάθεση πόρων

➤ Είναι υλοποιήσιμα και χρησιμοποιούνται για την υποστήριξη πολυμέσων και γραφικών σε συστήματα γενικού σκοπού

— Δύο βασικές απαιτήσεις:

➤ Σύστημα με προτεραιότητες, διακοπές και χωρίς γήρανση.

➤ Μικρή καθυστέρηση αποστολέα

➤ Η δεύτερη απαίτηση υλοποιείται δύσκολα. Απαιτείται είτε υλοποίηση μηχανισμών διακοπών σε συγκεκριμένα σημεία κλήσεων συστήματος (preemption points in long system calls) ή δυνατότητα διακοπής του πυρήνα του ΛΣ. Απαιτείται προστασία των δομών δεδομένων του πυρήνα

➤ Priority inversion (μια διεργασία χαμηλής προτεραιότητας κατέχει πόρο που χρειάζεται η διεργασία υψηλής προτεραιότητας) και priority inheritance (η διεργασία αυτή κληρονομεί για όσο διάστημα κρατάει τον πόρο την υψηλή προτεραιότητα)

2

Περιγραφή του προβλήματος κ σχεδίαση του συστήματος

Σε αυτό το κεφάλαιο παρουσιάζεται το τηλεφωνικό κέντρο και γίνεται μια εκτενή αναφορά στο γενικότερο πλαίσιο το οποίο θα λειτουργήσει το σύστημα. Περιγράφεται η σημερινή κατάσταση σχετικά με την λειτουργία των τηλεφωνικών κέντρων, τα διάφορα προβλήματα και οι περιορισμοί, οι ανάγκες που οδήγησαν στην επιτακτική ανάπτυξη και εξέλιξη τους καθώς και σε άλλα παρεμφερή ζητήματα. Επίσης, στο κεφαλαίο αυτό περιγράφεται με λεπτομέρειες η ανάλυση και η σχεδίαση του συστήματος.

2.1 Περιγραφή του προβλήματος

2.1.1 Τι είναι το τηλεφωνικό κέντρο

Τα τηλεφωνικά κέντρα χρησιμοποιούνται από οργανισμούς σαν ένα σημαντικό κανάλι επικοινωνίας και διεξαγωγής δοσοληψιών με τους πελάτες τους. Το πιο διαδομένο μέσο επικοινωνίας είναι το τηλέφωνο. Όταν λοιπόν η επικοινωνία γίνεται μόνο μέσω τηλεφώνου τότε αυτό καλείται τηλεφωνικό κέντρο.

Λίγα τμήματα στο εσωτερικό των εταιριών έχουν αναπτυχθεί μέσα σε λίγα χρόνια σαν την ανάπτυξη που παρατηρείται στην λειτουργία των τηλεφωνικών κέντρων. Μερικές μελέτες έχουν δείξει ότι ο αριθμός των εισερχόμενων τηλεφωνημάτων από πελάτες σε εταιρίες έχουν αυξηθεί από 11 εκατομμύρια στα μέσα της δεκαετίας του 80 σε πάνω από 15 εκατομμύρια το 2000 ενώ το ποσό αυτό έχει σχεδόν διπλασιαστεί στα μέσα του 2005. Εξαιτίας αυτής της πυρετώδους αύξησης, οι περισσότερες εταιρίες έχουν επικεντρώσει όλες τις δυνάμεις τους στο πως θα ανταποκριθούν καλύτερα και αποδοτικότερα στις απαιτήσεις που δημιουργούνται από το φαινόμενο της ραγδαίας αυτής αύξησης. Απόρροια αυτού, είναι οι εταιρίες να χρησιμοποιούν περισσότερο προσωπικό, καινούργιες τηλεφωνικές γραμμές και καλύτερη τεχνολογία για να μπορέσουν να απαντούν

στους πελάτες με τρόπο τέτοιο ώστε να ελαχιστοποιείται ο χρόνος αναμονής και τα αποτυχημένα τηλεφωνήματα.

2.1.2 Ανταγωνισμός η αίτια δημιουργίας καλύτερων τηλεφωνικών κέντρων

Ο ανταγωνισμός ο οποίος υφίσταται στην αγορά είναι κατά βάση ο κύριος λόγος ο οποίος έχει οδηγήσει στην αύξηση των απαιτήσεων για καλύτερα τηλεφωνικά κέντρα, τα οποία θα μπορούν να ανταποκρίνονται ποιοτικότερα στα τηλεφωνήματα τα οποία δέχονται. Στην πραγματικότητα όμως, ο ανταγωνισμός ο οποίος υπάρχει και καθορίζει σε μεγάλο βαθμό την αγορά, είναι δυναμικός. Αυτό σημαίνει ότι μεταβάλλεται διαρκώς. Έτσι για παράδειγμα, μπορεί μέχρι πρότινος ο ανταγωνισμός των μεγάλων επιχειρήσεων να ήταν για το μερίδιο της αγοράς που αφορούσε κάποιες βασικές βιομηχανίες, ενώ μετέπειτα μετακινήθηκε στα ηλεκτρονικά, στα αυτοκίνητα και στις τραπεζικές εργασίες κτλ. Ο αριθμός των νέων επιχειρήσεων που ξεκινούν τις δραστηριότητες τους και οι απαιτήσεις υπηρεσιών από αυτές προβάλλουν την ανάγκη για δημιουργία πολλαπλών καναλιών επαφής όπως για παράδειγμα μέσω του Web, του chat και των e-mail.

Σαν αποτέλεσμα αυτού, οι επιχειρήσεις δεν μπορούν πλέον να αντέξουν να μην επικοινωνούν και με συνέπεια να μην απαντούν στις απαιτήσεις των πελατών τους. Αν μια επιχείρηση δεν μπορεί να γνωρίσει τις ανάγκες των πελατών της τότε θα το κάνει κάποια άλλη. Αυτό βεβαίως λειτουργεί υπέρ των καταναλωτών, οι οποίοι αποτελούν τα πραγματικά αφεντικά στην αγορά και μάλιστα χρησιμοποιούν αυτήν την δύναμη που διαθέτουν ώστε να ισχυροποιήσουν τις απαιτήσεις τους για το προϊόν το οποίο επιθυμούν, τον τρόπο με τον οποίο θέλουν το προϊόν καθώς και όποτε αυτοί το θέλουν. Ο μόνος τρόπος λοιπόν επιβίωσης μιας επιχείρησης στην νέα τάξη πραγμάτων είναι το να ακούσουν τον καταναλωτή και τις ανάγκες του.

Σύμφωνα με τα παραπάνω, μια επιχείρηση προσφέρει αποτελεσματική εργασία όταν αυτή εναρμονίζεται με τις προτιμήσεις του πελάτη. Για την απόκτηση ενός πλεονεκτήματος, κάποιες έξυπνες εταιρίες, προχώρησαν στην χρήση τηλεφωνικών κέντρων για να παρακολουθούν τις απαιτήσεις των πελατών και να αναπτύσσουν συνεχώς τα νέα προϊόντα, τις υπηρεσίες τους και την διανομή αυτών βασισμένα στις ανεξάντλητες ανάγκες των πελατών.

2.1.3 Αλλαγές στην διοίκηση σε προσωπικό – διαδικασίες – τεχνολογία (μοντέλο re-engineering)

Η χρήση των νέων τεχνολογιών οι οποίες υπάρχουν, μπορούν να επιταχύνουν και τις ανάγκες αλλά και τις απαιτήσεις για αλλαγή, δίνοντας παράλληλα την δυνατότητα για ταχύτερη επικοινωνία, περισσότερη και λεπτομερέστερη πληροφόρηση για τους καταναλωτές καθώς και ακριβέστερα κριτήρια για την μέτρηση του κόστους και της ποιότητας. Παρόλα αυτά όμως, η ανάγκη για βελτιστοποίηση του τριπτύχου άνθρωποι / διαδικασίες / τεχνολογία είναι απαραίτητη για να επιτευχθεί η μεγιστοποίηση της επιστροφής ενός μέρους της επένδυσης από την εφαρμογή του τύπου της τεχνολογίας που χρησιμοποιήθηκε και της διαδικασίας που έγινε. Μερικές φορές οι εταιρίες προσπαθούν να εξασφαλίσουν την καλύτερη διαθέσιμη τεχνολογία. Με τον καιρό όμως συνειδητοποιούν ότι αυτή δεν είναι αρκετή για την πλήρη λειτουργία του συστήματος. Για το λόγο αυτό θα πρέπει να γίνουν και ριζικές αλλαγές στην διοίκηση η οποία θα πρέπει να κατευθύνει την δουλειά. Σκοπός της νέας διοίκησης είναι να εφαρμόσει ένα νέο πρόγραμμα το οποίο θα της δίνει την δυνατότητα να κάνει τα εξής:

- Να εξακριβώνει την ετοιμότητα – ευστροφία και την ικανότητα των υπαλλήλων στην αλλαγή του τρόπου εργασίας τους.
- Το κέρδος από την διοικητική υποστήριξη στην προσπάθεια να παρέχει την απαραίτητη εκπαίδευση εναρμονισμένη με το budget και τον σχεδιασμό της εταιρίας.
- Δημιουργώντας ένα πλάνο επικοινωνίας για το χρονικό περιθώριο μέσα στο πού θα γίνουν οι αλλαγές στην τεχνολογία καθώς και στις διαδικασίες εργασίας.
- Σκιαγραφώντας ποιες από τις τωρινές διαδικασίες της εταιρίας μπορούν να υποστηριχτούν από την νέα τεχνολογία και σχεδιάζοντας τις διαδικασίες εκείνες οι οποίες θα δημιουργηθούν από την εισαγωγή της νέας τεχνολογίας.
- Δημιουργώντας και βάζοντας σε εφαρμογή ένα πλάνο πάνω στο οποίο θα γίνει η μετάβαση από το παλιό στο καινούργιο σύστημα και παράλληλα

χρησιμοποίηση του παλιού και του καινούργιου συστήματος για ένα χρονικό διάστημα για να εξασφαλιστεί η καλύτερη μετάβαση.

- Δημιουργώντας μια διαδικασία η οποία θα μπορεί καθώς το καινούργιο σύστημα είναι σε λειτουργία να προβαίνει σε βελτιώσεις και ενημερώσεις οι οποίες είναι απαραίτητες να γίνουν και είναι πολύ εύκολες να κατανοηθούν από τους υπαλλήλους.

2.1.4 Που οφείλεται η ανάγκη ύπαρξης τηλεφωνικών κέντρων από τις εταιρίες

Οι περισσότερες βιομηχανίες βρίσκονται σε μια περίοδο στην οποία παρατηρείται μια γενική περατωμένη κατάσταση ικανοτήτων. Αυτό έχει σαν αποτέλεσμα πολλές εταιρίες να ωθούνται στο σημείο επιβίωσης τους από την στιγμή που δεν μπορούν να παρακολουθήσουν και να ανταποκριθούν άμεσα με τις απαιτήσεις των καταναλωτών. Αυτό παρατηρείται σε εκείνες, οι οποίες λειτουργούν με βάση τους παλιότερους κανόνες. Οι εταιρίες αυτές, έχουν επικεντρωμένη την προσοχή τους στην κουλτούρα αλλά και σε διαδικασίες οι οποίες βασίζονται σε εκτός εποχής μοντέλα. Για να μπορέσουν όμως να ευημερήσουν, θα πρέπει να αναπροσαρμόσουν και να αλλάξουν την λειτουργία τους και τον τρόπο δουλειάς τους (re-engineering), ώστε να μην έχουν σαν στόχο μόνο την επίτευξη ενός κόστους στο οποίο μπορούν να ανταποκριθούν, αλλά η εργασία τους να είναι αποδοτική και να κατευθύνεται από τις ανάγκες της εκάστοτε αγοράς.

Με αυτόν το τρόπο, οι ευκίνητες εταιρίες οι οποίες ακούνε και προσαρμόζονται στις ανάγκες και τις υπηρεσίες που επιζητούν οι πελάτες αναπτύσσονται ραγδαία. Αναπτύσσονται γιατί επιτρέπουν στους υπάλληλους τους τα εξής : α) είναι ελεύθεροι να επικοινωνούν άμεσα με τους καταναλωτές β) τους δίνεται η δυνατότητα να αντιδρούν και να ενεργούν ανάλογα με τις ανάγκες των πελατών. Το κλειδί της επιτυχίας είναι η αμεσότητα.

Επίσης, οι γρήγορα αναπτυσσόμενες εταιρίες είναι αυτές οι οποίες έχουν απαλλαγεί από το παρελθόν τους. Είναι αυτές οι οποίες έχουν ασπαστεί τις νέες απαιτήσεις των πελατών τους. Τις νέες αυτές απαιτήσεις τις πληροφορούνται από τα τηλεφωνικά κέντρα μέσω των οποίων έρχονται σε επαφή με τους καταναλωτές. Για το λόγο αυτό γίνεται αντιληπτό για την ανάγκη ύπαρξης τηλεφωνικών κέντρων τα οποία αποτελούν καθοριστικό ρόλο στην επιβίωση των εταιριών.

2.1.5 Re-engineering το τηλεφωνικό κέντρο

Η χρησιμοποίηση του μοντέλου re-engineering σε ένα τηλεφωνικό κέντρο για την επίτευξη της βελτιστοποίησης έχει γίνει για την ικανοποίηση ενός από τους δυο αντικειμενικούς στόχους, τα οποία αποτελούν και το κλειδί της επιτυχίας και αυτά είναι 1) να τραβήξουν την προσοχή του καταναλωτή και 2) την χρήση της πληροφορίας της τεχνολογίας.

Μέχρι πρόσφατα , όμως , το να γνωρίζεις τους δικούς σου καταναλωτές δεν ήταν εύκολο αντιθέτως η ευκολία αυτή έγκειται μόνο στα λόγια. Η τεχνολογία των υπολογιστών δεν ήταν έτοιμη να φέρει εις πέρας την δουλειά : τα δεδομένα τα οποία υπήρχαν ήταν ογκώδη , πάρα πολύ διάσπαρτα μέσα στον οργανισμό καθώς και πάρα πολύ ευμετάβλητα καταχωρημένα ώστε να δυσχεραίνεται η αποτελεσματική χρήση τους. Τώρα όμως με την δημιουργία δυνατών εργασιακών κέντρων , εκτεταμένων δικτύων, ειδικών πακέτων λογισμικού καθώς και δυνατών μηχανών βάσεων δεδομένων, η τεχνολογία δεν αποτελεί πλέον πρόβλημα.

Το πιο σημαντικό πράγμα που μια εταιρία θα πρέπει να κάνει είναι το να κατανοήσει τις ανάγκες των τωρινών και μελλοντικών της πελατών χρησιμοποιώντας άμεσα την πληροφόρηση που έχει από τα τηλεφωνικά κέντρα.

Ένα καλά οργανωμένο τηλεφωνικό κέντρο βοηθά στο να :

- Διατηρεί τους τωρινούς πελάτες και να προσελκύει καινούργιους.
- Καθορίζει και γνωστοποιεί τις παρατηρήσεις των πελατών για ποιοτικά σφάλματα στα προϊόντα – υπηρεσίες που επιζητούν.
- Ακούει πρώτα τα <<θέλω>> των καταναλωτών και έπειτα προσαρμόζουν τα προϊόντα και τις υπηρεσίες
- Δίνει τις προσδοκίες που υπάρχουν για το καινούργιο προϊόν όπως έχει διατυπωθεί από τους πελάτες που τηλεφωνούν.

- Γνωστοποιεί τις προσδοκίες των καταναλωτών παρέχοντας γρήγορη, ακριβής απάντηση όταν απαιτείται από τον καταναλωτή.

- Μεγιστοποιεί τα κέρδη – οφέλη από την διαδικασία.

Ο αντικειμενικός στόχος για βελτίωση των τηλεφωνικών κέντρων κρίνεται επιτακτικός και ζωτικής σημασίας για τις εταιρίες. Σε κάθε άλλη περίπτωση η εταιρία θα είναι επικεντρωμένη σε διαδικασίες οι οποίες λαμβάνουν χώρα στο εσωτερικό της μόνο. Η ύπαρξη του επιπλέον κόστους οδηγεί ολοένα και περισσότερους καταναλωτές να έρχονται σε επικοινωνία με τις εταιρίες μέσω τηλεφωνικών κέντρων και άλλων ηλεκτρονικών μέσων επικοινωνίας. Οι τεχνολογίες οι οποίες χρησιμοποιούνται για συλλογή της πληροφόρησης, η οποία προέρχεται από την επικοινωνία με το αγοραστικό κοινό, διαδραματίζει ένα σημαντικό ρόλο στον καθορισμό των διαδικασιών ενός τηλεφωνικού κέντρου. Όπως για παράδειγμα:

- ✓ Το περιορισμένο κόστος και πολλαπλή χρήση της τεχνολογικής πληροφόρησης.

- ✓ Η χρήση της τεχνολογίας της τηλεπικοινωνίας κάνει πιθανό τον εντοπισμό των προτιμήσεων των καταναλωτών κάθε φορά.

- ✓ Μέσω του Web και του Internet έχουν αυξηθεί και διαμορφωθεί τα κανάλια πληροφόρησης των καταναλωτών.

- ✓ Η ικανότητα ένταξης – εισαγωγής του τηλεφώνου και της τεχνολογικής πληροφόρησης παρέχει νέους τρόπους εκμετάλλευσης από την διοίκηση της γνώσης ώστε να μπορεί να απαντήσει άμεσα.

2.1.6 Η ανάπτυξη της τεχνολογίας πληροφόρησης θα οδηγήσει και σε νέες ευκαιρίες

Την στιγμή που θα υπάρξει βελτίωση στην τεχνολογία της τηλεπικοινωνίας τότε θα επιταχυνθεί και η αλλαγή. Η αλλαγή αυτή θα δημιουργήσει και ευκαιρίες. Οι νέοι κανόνες που θα οδηγήσουν την επιχείρηση στην επιτυχία θα επιτρέπουν και την μίμηση, δηλαδή οι οργανισμοί που επικεντρώνονται στον καταναλωτή θα μπορέσουν να κλέβουν και να κάνουν δικούς τους πελάτες οι οποίοι μέχρι τότε προϋπήρχαν σε άλλες εταιρίες. Αυτό δημιουργεί την ευκαιρία για αυτούς που μπορούν να βοηθήσουν τις εταιρίες να βελτιώσουν το τηλεφωνικό τους κέντρο μέσω ειδικής ακρόασης ειδικών συμβούλων και μέσω αποδοτικής χρήσης της τεχνολογίας.

Η αποτίμηση αυτού, είναι η είσοδος ενός φορέα στο περιβάλλον του τηλεφωνικού κέντρου για τον πελάτη. Αυτό έχει σχεδιαστεί να τεκμηριώνει – αποθηκεύει τον τωρινό τρόπο χρήσης της τεχνολογικής πληροφόρησης και των ανθρώπων και μπορεί να εντοπίζει με ακρίβεια τις ευκαιρίες για βελτίωση ή re-engineering. Επειδή η λειτουργία πολλών τηλεφωνικών κέντρων γίνεται χωρίς να βασίζεται σε μια σχεδιασμένη μέθοδο, για αυτό πολλές φορές δυσκολεύονται στο να απαντήσουν σε απελαθεί τηλεφωνήματα. Δεν έχουν ούτε μια τυπική διαδικασία η οποία θα βελτιώσει τις δεσμεύσεις, αλλά ούτε την προοπτική στο να μετατρέψουν το τηλεφωνικό κέντρο σε ένα υψηλής απόδοσης και υψηλής τεχνολογίας πλεονέκτημα για αυτές.

Η αποτίμηση αυτή αποκαλύπτει την ανάγκη για την ύπαρξη των παρακάτω:

- Επέκταση και αναβάθμιση των ήδη υπάρχοντος εξοπλισμού.
- Βελτίωση των ήδη υπαρχόντων στάνταρ, διαδικασιών και ασφάλειας.
- Προσδιορισμός της ποσότητας παροχής υπηρεσιών σύμφωνα με τα ζητούμενα επίπεδα των πελατών.
- Προσέλκυση ,διαλέγοντας, και εκπαιδεύοντας ποιοτικά το προσωπικό.
- Διοίκηση εκπαίδευση των ικανοτήτων τους.

- Αλλαγή της συλλογικής κουλτούρας.

Όταν τα θέματα της επιχείρησης έχουν καθοριστεί, εμφανίζονται ορισμένες ευκαιρίες για την εφαρμογή και υποστήριξη της τεχνολογίας της πληροφόρησης, οι οποίες περιλαμβάνουν:

- Ενσωμάτωση του ηλεκτρονικού τηλεφώνου.
- Η εφαρμογή μιας σχετικής βάσης δεδομένων.
- Η διανομή μέσω ηλεκτρονικών καναλιών επικοινωνίας.
- Κατάλογοι βασισμένοι στο διαδίκτυο.
- Μια πιθανή εικονική του προϊόντος.

2.1.7 Τρόποι εκμετάλλευσης των νέων ευκαιριών

Είναι πάρα πολύ εύκολο να πέσουμε σε σφάλματα χρησιμοποιώντας τους παλιούς τρόπους σκέψης όσον αφορά την επικοινωνία με τον καταναλωτή. Οι επιχειρήσεις το κάνουν καθώς και οι μικροπωλητές. Σε μια εποχή εξαιρετικού ενδιαφέροντος όσον αφορά την τεχνολογία, οι επιχειρήσεις που έχουν ζήσει με την τεχνολογία θα να αποφύγουν τον πειρασμό να καθοδηγηθούν από την τεχνολογία. Μπορεί να είναι της μόδας και θελκτικό να γίνονται συζητήσεις για το τελευταίο τσιπ που δημιουργήθηκε, για την ψηφιακή τεχνολογία και αλλά, αλλά δεν αποτελεί το βέλτιστο παράγοντα στην επιλογή μιας ενδεικτικής επιχειρησιακής πρωτοβουλίας.

Για να μπορέσουμε να καταπολεμήσουμε το αυξημένο κόστος και την φτωχή απόδοση θα πρέπει να δημιουργήσουμε ένα μέτρο σύγκρισης το οποίο θα ελέγχει και θα δίνει την προτεραιότητα στις ευκαιρίες που παρουσιάζονται. Θα πρέπει να διαλέξουμε έναν παράγοντα, ο οποίος θα εξυπηρετεί στο να μετράει την απόδοση της εκτέλεσης της λειτουργίας, της τεχνολογίας, του ανθρώπινου δυναμικού, του κόστους, των εγκαταστάσεων, της διοικητικής γνώσης και την παροχή υπηρεσιών.

Θα πρέπει να εξετάζει όχι μόνο πως θα βελτιώσει την εσωτερική διαδικασία, αλλά να μπορεί να παρατηρήσει και να την συγκρίνει τουλάχιστον με άλλες 20 με 30 ανταγωνιστές στην αγορά.

2.1.8 Αναγκαιότητα ανάλυσης κόστους – ωφέλειας

Η Βελτίωση των τηλεφωνικών κέντρων θα πρέπει να παρέχει ωφέλεια στους καταναλωτές, τους και τους μετόχους της εταιρίας. Αυτό αποδεικνύεται μέσω της αύξησης των οικονομικών της. Συζητήσεις για την δημιουργία σχετικών οικονομικών προτύπων δεν αποτελεί συχνά αντικείμενο συζήτησης της διοίκησης των τηλεφωνικών κέντρων. Από την άλλη όμως, μαθαίνοντας από τα αποτελέσματα μιας τέτοιας ανάλυσης μπορεί να κερδίσει το σεβασμό και να ανάδειξη την ανάγκη των εταιριών να στραφούν περισσότερο στα τηλεφωνικά κέντρα, μέσω των οποίων θα πετύχουν θετικό πρόσημο στην ανάλυση κόστους – ωφέλειας

2.1.9 Ανάλυση ROI (return on investment) –απόδοση της επένδυσης

Το κάθε στέλεχος της εταιρίας έχει σαν εργασία να μπορέσει να μεγιστοποιήσει την επιστροφή όσων περισσότερων χρημάτων τα οποία έχουν διατεθεί από αυτήν. Επομένως, θα πρέπει με ένα περιορισμένο κεφάλαιο για επένδυση που θα χρησιμοποιηθεί στην ενίσχυση της διαδικασίας κάθε πρόταση για μεγάλα έξοδα θα πρέπει να συνοδεύονται από μια οικονομική ανάλυση η οποία θα εξασφαλίζει την επιστροφή της δαπάνης στην επιχείρηση.

Υπάρχουν δυο τρόποι για να προβούμε σε μια ROI ανάλυση :

➤ Η ελαχιστοποίηση ή η αποφυγή του κόστους

Το άμεσο κόστος, είναι η δαπάνη για τα αντικείμενα τα οποία συνδέονται απευθείας με το προσφερόμενο από την επιχείρηση προϊόν ή υπηρεσία, και το οποίο μπορεί πολύ εύκολα να εντοπίσει από το υπολογιστικό σύστημα της επιχείρησης. Το έμμεσο κόστος, είναι το λιγότερο αντιληπτό, δεν μπορεί εύκολα να εντοπίσει .Η ROI ανάλυση επικεντρώνεται κυρίως στο άμεσο κόστος και για αυτό χρησιμοποιείται στις επενδύσεις για τεχνολογική και τηλεπικοινωνιακή πληροφόρηση

των τηλεφωνικών κέντρων. Μερικά από τα πλεονεκτήματα του άμεσου κόστους είναι τα εξής:

- Αυξάνει την παραγωγικότητα το οποίο επιτρέπει την χρήση λιγότερων πρακτόρων για να κάνουν περισσότερη δουλειά σε λιγότερο χρόνο.

- Εφαρμόζεται η τεχνολογική πληροφόρηση η οποία αντικαθιστά την λειτουργία όλων των agent.

- Ελαττώνεται ο χρόνος αναμονής οπότε και το τηλεφωνικό κόστος

➤ **Οφέλη από τους νέους παραγόμενους πόρους**

Αυτή η προσέγγιση βρίσκει εφαρμογή σε μερικές επενδύσεις των τηλεφωνικών κέντρων και επικεντρώνεται στην παρακάτω αντίληψη: α) σίγουρα η βελτίωση στην εξυπηρέτηση των πελατών θα έχει σαν αποτέλεσμα την διατήρηση των περισσότερων από αυτούς και β) οι καταναλωτές οι οποίοι έχουν διατηρηθεί θα συνεχίζουν να ψωνίζουν από την ίδια εταιρία και θα παράγουν κέρδος

2.2 Δομή του τηλεφωνικού κέντρου

2.2.1. Εισαγωγή

Στη σύγχρονη κοινωνία των πληροφοριών, πολλές οργανώσεις εμφανίζουν το πρόβλημα της συνεχόμενης αναπτυσσόμενης ροής των τηλεφωνημάτων. Οι τράπεζες για παράδειγμα δεν θέλουν τις οικονομικά ασύμφωρες λύσεις των εμπειρογνωμόνων τους στους long-winded πελάτες. Ένα τηλεφωνικό κέντρο θα μπορούσε να είναι λύση σε αυτά τα προβλήματα. Οι πελάτες μπορούν να καλέσουν έναν κεντρικό αριθμό για όλες τις ερωτήσεις τους, και να κατευθυνθούν στο τηλεφωνικό κέντρο, όπου διάφοροι τηλεφωνικοί εμπειρογνώμονες είναι έτοιμοι να τις βοηθήσουν.

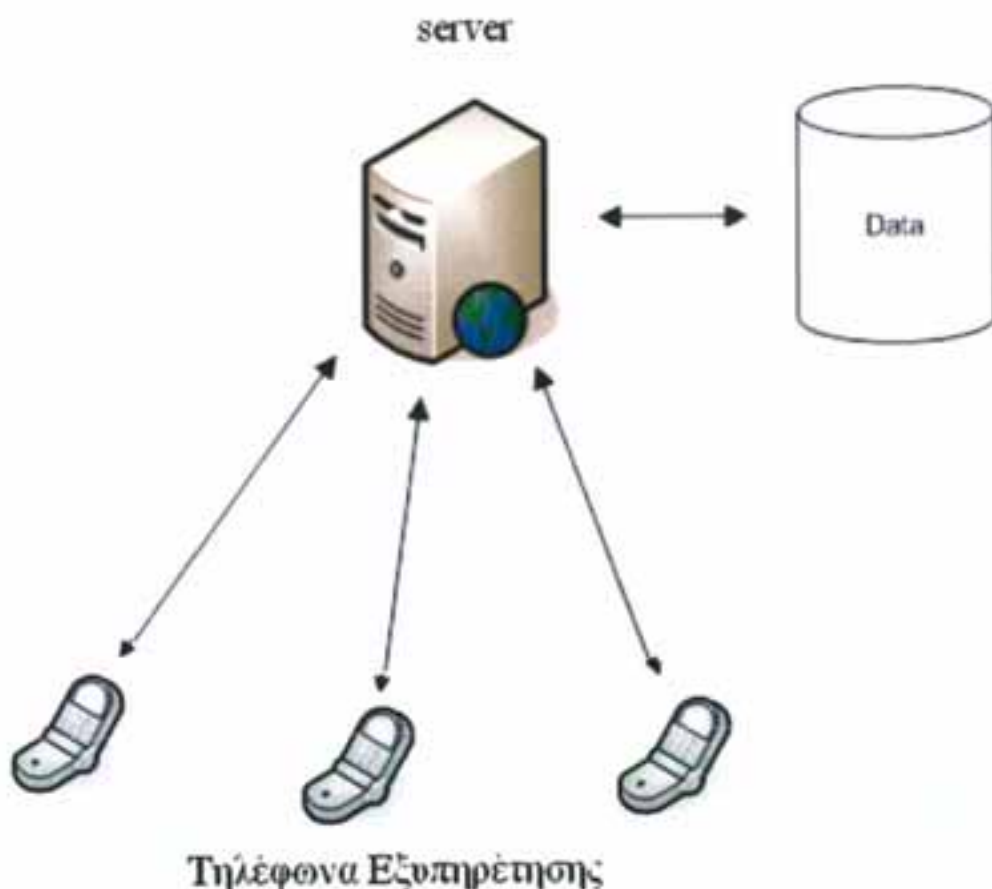
Κατ' αυτό τον τρόπο, μόνο οι agents στο τηλεφωνικό κέντρο, εξετάζουν τους πελάτες άμεσα, ελευθερώνοντας περισσότερο χρόνο στους υπαλλήλους της επιχείρησης, οι οποίοι μπορούν να εστιάσουν σε άλλες συγκεκριμένες εργασίες ανάλογα με την πείρα τους. Οι agents θα πρέπει να έχουν την κατάλληλη εκπαίδευση, η οποία θα τους μετατρέψει σε τηλεφωνικούς επαγγελματίες εξυπηρετήσεων πελατών.

Σε αυτήν την ενότητα, σκιαγραφείται το τηλεφωνικό κέντρο το οποίο θα υλοποιηθεί. Με άλλα λόγια παρουσιάζονται τα στοιχεία εκείνα, οι οντότητες, οι οποίες απαρτίζουν το σύστημα και διαδραματίζουν πρωταγωνιστικό ρόλο για την άρτια λειτουργία του, καθώς και οι ροές διεργασιών μεταξύ των οντοτήτων. Επίσης αναφέρονται και κάποια στοιχεία – πληροφορίες όσον αφορά τους όρους λειτουργίας του συστήματος δηλαδή του τηλεφωνικού κέντρου.

2.2.2 Χαρακτηριστικά λειτουργίας του τηλεφωνικού κέντρου

Το κυρίαρχο χαρακτηριστικό του συστήματος είναι ότι αποτελεί ένα inbound σύστημα. Αυτό σημαίνει ότι το σύστημα απασχολεί μόνο τα τηλεφωνήματα – κλήσεις, οι οποίες γίνονται μόνο προς το τηλεφωνικό κέντρο. Δεν λαμβάνονται καθόλου υπόψη οι εξερχόμενες κλήσεις, δηλαδή οι κλήσεις οι οποίες γίνονται από το σύστημα προς τους πελάτες (outbound σύστημα). Αυτό γίνεται με οδηγό το σύστημα να είναι πιο απλό με διεργασίες οι οποίες δεν θα χαρακτηρίζονται από πολυπλοκότητα και δεν θα υπάρχει σύνθετη δρομολόγηση. Απόρροια αυτού είναι να είναι πιο κατανοητή η λειτουργία του από τρίτους.

Το τηλεφωνικό κέντρο θα λειτουργεί πέντε μέρες την εβδομάδα και το ωράριο το οποίο θα μπορεί να εξυπηρετείται το κοινό είναι από τις 8 το πρωί μέχρι της 12 το βραδύ. Επίσης θα περιλαμβάνει τρία τηλέφωνα τα οποία θα λειτουργούν παράλληλα για την εξυπηρέτηση των πελατών κατά την διάρκεια του ωραρίου του συστήματος. Τα τρία τηλέφωνα αυτά θα είναι συνδεδεμένα με έναν κεντρικό server μέσω ενός τοπικού δικτύου LAN. Ο server θα κατ'αγράφει τα χρήσιμα στοιχεία των τηλεφωνημάτων τα οποία θα αποθηκεύονται σε μια βάση δεδομένων και τα οποία θα μπορούν πολύ εύκολα να ανακληθούν από το προσωπικό του συστήματος.



Σχήμα 12 :Δικτύου LAN σύνδεσης των τηλεφώνων του συστήματος με τον server

2.2.3 Το προσωπικό του συστήματος

Το προσωπικό το οποίο θα επανδρώσει το τηλεφωνικό κέντρο αποτελεί και την καρδιά του συστήματος. Επειδή πρόκειται για ένα απλό, βασικό τηλεφωνικό κέντρο θα χρησιμοποιηθούν δέκα άτομα τα οποία θα είναι το προσωπικό και θα καλούνται agents. Οι agents θα επωμιστούν την ευθύνη της άρτιας λειτουργίας του συστήματος. Αυτοί θα είναι οι οποίοι θα επικοινωνούν καθημερινά με τους πελάτες και θα είναι υπεύθυνοι να φέρουν εις πέρας τις απαιτήσεις τους.

Ο κάθε agent από το προσωπικό, θα εργάζεται 8 ώρες ημερησίως σε μεταβαλλόμενες βάρδιες. Μέσα στο μήνα (δηλαδή σε 20 εργάσιμες μέρες) θα έχουν 2 ρεπό. Πράγμα που σημαίνει ότι κάθε ένας από αυτούς δεν θα ξεπερνά μηνιαίως τις 144 ώρες υπηρεσίας.

Όπως αντιλαμβάνεται κανείς, η λειτουργία του συστήματος εξαρτάται αποκλειστικά από το προσωπικό που θα το επανδρώσει. Οπότε μια από τις πρωταρχικές φροντίδες είναι η τακτική και συστηματική εκπαίδευση του προσωπικού. Η εκπαίδευση αποτελεί πρωταρχικό στόχο, γιατί με αυτό το τρόπο το προσωπικό θα είναι εφοδιασμένο με όλες τις απαραίτητες εκείνες γνώσεις, ώστε να μπορούν να αντιμετωπίζουν με επιτυχία όλα εκείνα τα τυχόν προβλήματα που θα παρουσιαστούν, και επομένως θα διατηρούν σε υψηλή επίπεδα την δική τους απόδοση και κατά συνέπεια και του συστήματος. Από τη πρώτη στιγμή κιόλας και σε τακτά χρονικά διαστήματα, οι agents θα εκπαιδεύονται σε θέματα τηλεφωνικής επικοινωνίας εν γένει, εξυπηρέτησης πελατών, πωλήσεων μέσω τηλεφώνου, αντιμετώπισης αντιρρήσεων και διαχείρισης παραπόνων.

Τέλος χώρια από το προσωπικό θα υπάρχει και ένας κεντρικός διευθυντής (manager) ο οποίος θα είναι επωμισμένος με την ευθύνη για την σωστή λειτουργία του συστήματος. Οι αρμοδιότητες του, έχουν περισσότερο χαρακτήρα ελέγχου, αφού θα είναι ο αρμόδιος να ελέγχει το προσωπικό, να αντιμετωπίζει τυχόν απορροίες αυτών και τέλος αυτός θα είναι υπεύθυνος για την εκπαίδευση του προσωπικού.

2.2.4 Ανάλυση και σχεδίαση του συστήματος

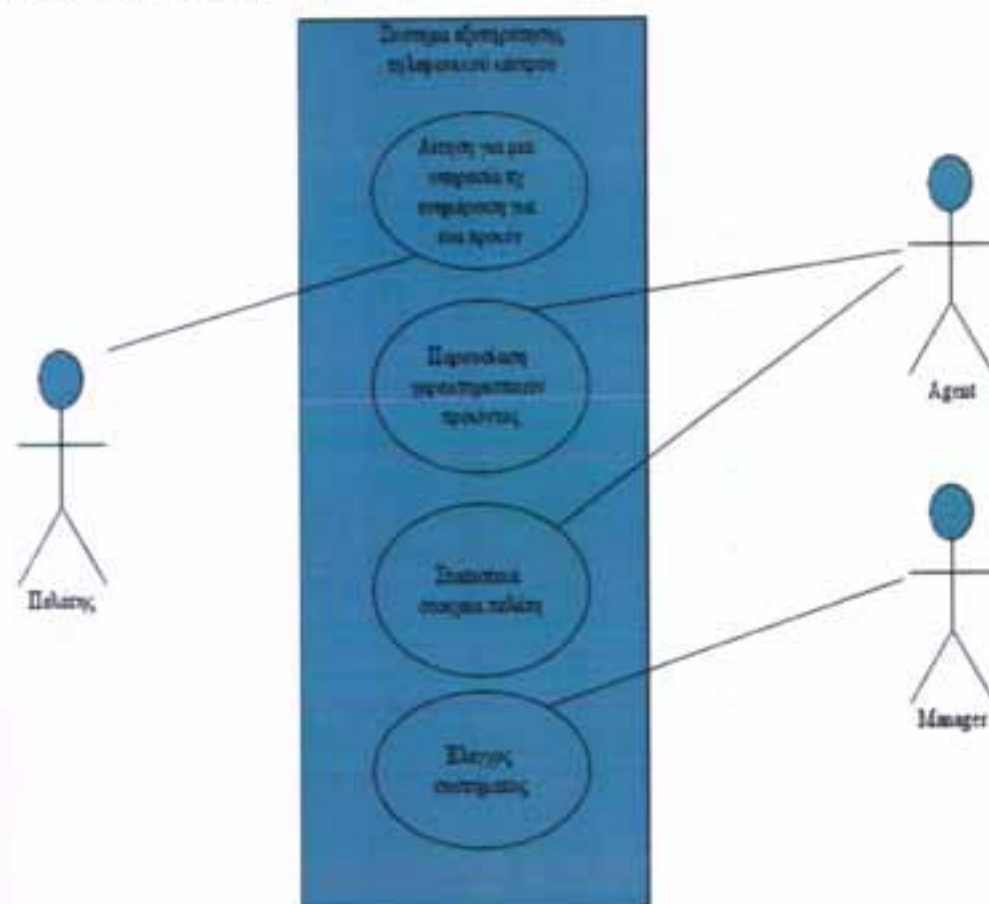
Στο κεφάλαιο αυτό περιγράφεται με λεπτομέρεια η ανάλυση και η σχεδίαση του συστήματος. Συγκεκριμένα, παρουσιάζεται η αρχιτεκτονική που χρησιμοποιείται και οι λειτουργίες που υλοποιήθηκαν ως ροές διεργασιών. Για την ανάλυση και σχεδίαση θα χρησιμοποιήσουμε την UML και τα διαγράμματα της για τα οποία έχει γίνει αναφορά σε προηγούμενο κεφάλαιο.

Για την υλοποίηση των διαγραμμάτων της UML θα χρησιμοποιηθεί το πρόγραμμα Visual Paradigm For Uml 5.3 Enterprise Edition. Το πρόγραμμα αυτό είναι ένας οδηγός σχεδιασμού διαγραμμάτων της Uml.

2.2.5 Ανάλυση των απαιτήσεων του συστήματος

Σε αυτήν την φάση οι εξωτερικοί αλληλεπιδρώντες με το σύστημα μοντελοποιούνται μαζί με την λειτουργικότητα που απαιτούν από το σύστημα. Οι εξωτερικοί αλληλεπιδρώντες στην περίπτωση μας είναι ο πελάτης που τηλεφωνεί στο

σύστημα και επιθυμεί μια υπηρεσία όπως πχ ενημέρωση για ένα προϊόν που τον ενδιαφέρει. Ο εξυπηρετητής ικανοποιεί το αίτημα του πληροφορώντας τον για το προϊόν αφού πρωτίστως, έχει καταχωρήσει τα στοιχεία του πελάτη. Τέλος ο διευθυντής του τηλεφωνικού κέντρου παρακολουθεί και ελέγχει την διαδικασία για τυχόν ύπαρξη προβλημάτων. Η παραπάνω διαδικασία απαίτησης του πελάτη μιας υπηρεσίας από το σύστημα του τηλεφωνικού κέντρου δίνεται στο ακόλουθο σχήμα με την χρήση του διαγράμματος use-case της UML.



Σχήμα 13: Απαιτήσεις πελατών

2.2.6 Ανάλυση και σχεδίαση του συστήματος

Εδώ παρουσιάζονται οι κλάσεις οι οποίες δημιουργούνται. Δίνεται σχηματικά με το διάγραμμα κλάσεων της UML (class diagram) το οποίο περιγράφει την αλληλεπίδραση μεταξύ τους.

Στο σύστημα μας εμφανίζονται τέσσερις κλάσεις οι οποίες είναι:

- ✓ Κλάση πελάτη

- ✓ Κλάση υπηρεσίες
- ✓ Κλάση Agent
- ✓ Κλάση Manager

2.2.7 Ορισμός κλάσεων

Στην κλάση του πελάτη αναγράφονται τα δεδομένα καθώς και οι operations.

Οι ιδιότητες και οι τύποι δεδομένων είναι όπως φαίνεται από το ακόλουθο σχήμα της κλάσης είναι

- Ο αριθμός ταυτότητας του πελάτη που έχει τύπο char. Ο αριθμός ταυτότητας είναι unique για κάθε πελάτη δηλαδή είναι μοναδικό στοιχείο για κάθε πελάτη.

- Το όνομα του πελάτη με τύπο char
- Το επώνυμο του πελάτη με τύπο char
- Το τηλέφωνο του πελάτη με τύπο char

Οι operations της κλάσης πελάτη είναι οι εξής:

- Προσθήκη νέου πελάτη
- Διαγραφή πελάτη
- Μεταβολή στοιχείων πελάτη

Πελάτης
-αριθμός ταυτότητα : char
-ονομα : char
-επώνυμο : char
-πολη : char
-τηλέφωνο : int
+προσθήκη πελατη()
+διαγραφή πελατη()
+μεταβολή στοιχείων πελατη()

Στην κλάση υπηρεσίες τα δεδομένα καθώς και οι operations είναι οι εξής :

Ιδιότητες κλάσης:

- Κωδικός προϊόντος με τύπο integer. Αυτή η ιδιότητα είναι και unique για να γίνεται ο διαχωρισμός ενός προϊόντος από τα υπόλοιπα.
- Το όνομα του προϊόντος με τύπο char.

- Τιμή του προϊόντος με τύπο integer.
- Οι operations είναι :
- Προσθήκη υπηρεσίας
 - Διαγραφή υπηρεσίας
 - Μεταβολή στοιχείων υπηρεσίας
 - Παρουσίαση δεδομένων υπηρεσίας
 - Αντιστοίχιση κωδικού πελάτη με κωδικού υπηρεσίας.

κλάση
κωδ. προϊόντος : int
ονομα : char
μισθ : int
+ προσθήκη υπηρεσίας()
+ διαγραφή υπηρεσίας()
+ μεταβολή στοιχείων υπηρεσίας()
+ παρουσίαση δεδομένων υπηρεσίας()
+ αντιστοίχιση κωδ. πελάτη με κωδ. υπηρεσίας()

Για την κλάση agent διακρίνονται τα ακόλουθα:

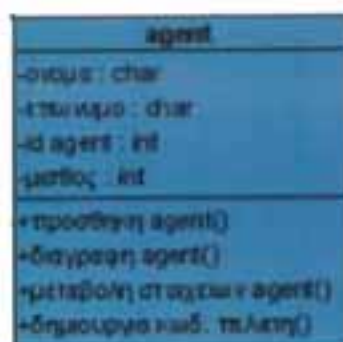
Τα δεδομένα της ακόλουθης κλάσης είναι:

• Το id agent με τύπο integer. Είναι unique δηλαδή είναι το χαρακτηριστικό στοιχείο του που τον διαφοροποιεί από τους υπόλοιπους agents.

- Το όνομα του agent με τύπο char.
- Το επώνυμο του agent με τύπο char.
- Ο μισθός του agent με τύπο integer

Οι operations είναι:

- Προσθήκη agent
- Διαγραφή agent
- Μεταβολή στοιχείων agent
- Δημιουργία κωδικού πελάτη



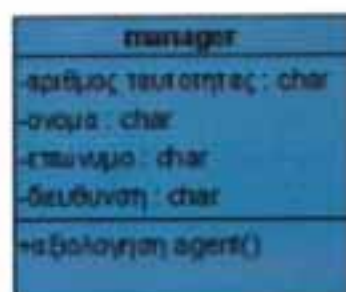
Για την κλάση manager ισχύουν τα ακόλουθα:

Δεδομένα κλάσης:

- Αριθμός ταυτότητας manager με τύπο char. Είναι unique δεδομένο.
- Το όνομα του manager με τύπο char.
- Το επώνυμο του manager με τύπο char
- Η διεύθυνση κατοικίας του manager με τύπο char.

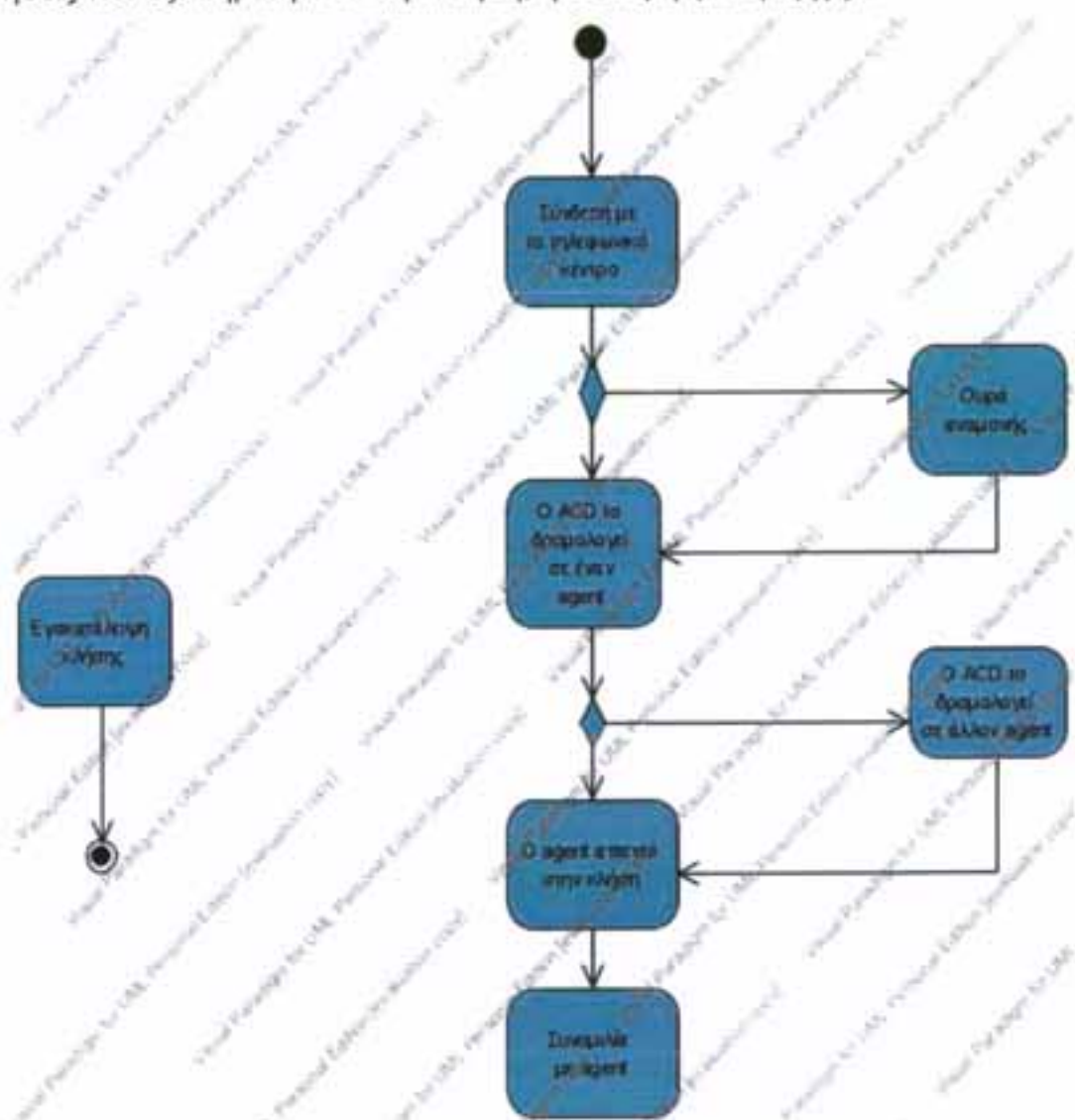
Οι operation της κλάσης είναι μια:

- Αξιολόγηση του agent



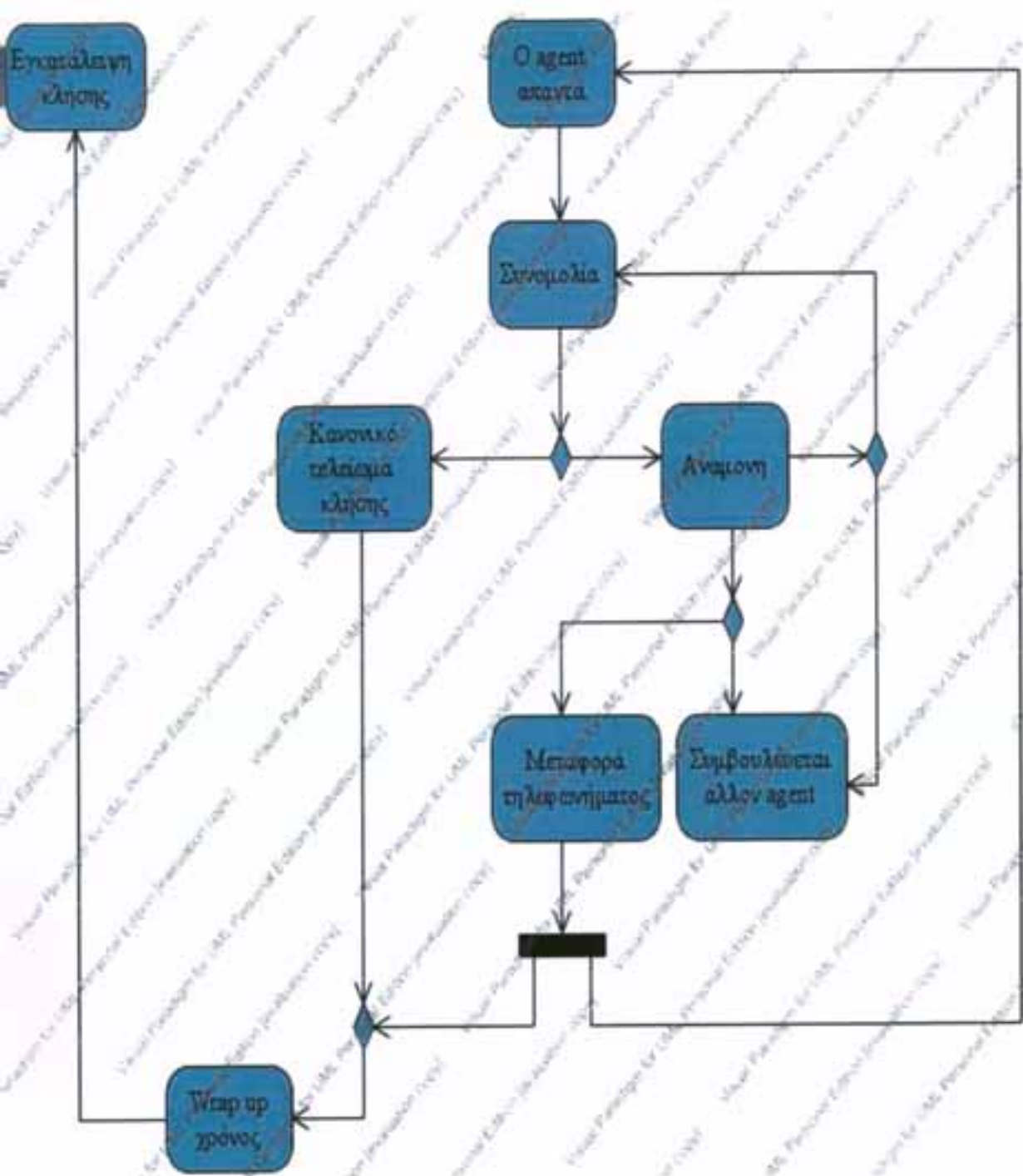
Το παρακάτω διάγραμμα κλάσεων της UML δείχνει την αλληλεπίδραση των κλάσεων του συστήματος του τηλεφωνικού κέντρου. Σύμφωνα με αυτό ο πελάτης επικοινωνεί με τον agent του τηλεφωνικού κέντρου ζητώντας του μια υπηρεσία. Ο agent δημιουργεί για τον πελάτη έναν κωδικό. Έπειτα βρίσκει τον κωδικό της ζητούμενης από αυτόν υπηρεσίας και αυτή διατίθεται στον πελάτη. Την διαδικασία που κάνει ο agent για να παρέχει μια υπηρεσία σε έναν πελάτη ελέγχεται από τον manager του τηλεφωνικού κέντρου ο οποίος και τον αξιολογεί.

το τηλεφωνικό κέντρο. Από αυτό το σημείο και μετά, ο πελάτης μπορεί να εγκαταλείψει οποιαδήποτε στιγμή την διαδικασία. Από την στιγμή που θα συνεχιστεί το τηλεφώνημα, εάν όλο το προσωπικό είναι απασχολημένο (agents), τότε το τηλεφώνημα τοποθετείται στην σειρά, δηλαδή σε μια ουρά αναμονής. Όταν ένας πράκτορας είναι διαθέσιμος, ένας διανομέας κλήσης ο οποίος είναι τοποθετημένος καθοδηγεί το τηλεφώνημα σε εκείνον τον εξυπηρετητή. Ο εξυπηρετητής πρέπει να απαντήσει στο τηλεφώνημα, αλλά εάν όχι, ο αυτόματος διανομέας κλήσης καθοδηγεί το τηλεφώνημα σε έναν άλλο εξυπηρετητή διαθέσιμο να ανταποκριθεί. Μετά από τις απαντήσεις των εξυπηρετητών στην κλήση, η δεύτερη φάση αρχίζει.



Σχήμα 15: Περίπτωση κλήσης με ACD router (Φάση 1)

Στην δεύτερη φάση, πραγματοποιείται η πραγματική ομιλία. Όταν το τηλεφώνημα τελειώνει σε αυτήν την φάση, ακολουθεί ένας wrap-up χρόνος εγκατάλειψης ή της λήξης της κλήσης, που επιτρέπει στον εξυπηρετητή να κάνει τις σημειώσεις του για την κλήση. Μετά από την συζήτηση δραστηριότητας, δύο δραστηριότητες μπορούν να ακολουθήσουν: λήξη, εάν ο πελάτης είναι ικανοποιημένος με την υπηρεσία, ή αναμονή. Κατά τη διάρκεια της αναμονής, ο πράκτορας μπορεί, παραδείγματος χάριν, να ανατρέξει γρήγορα για κάποιες πληροφορίες που θα του έχουν ζητηθεί. Σε εκείνη την περίπτωση, μετά την αναμονή ο πράκτορας ανακτά την κλήση. Ο πράκτορας μπορεί επίσης να επιλέξει να συμβουλευθεί έναν άλλο πράκτορα με την έναρξη μιας δεύτερης, εσωτερικής κλήσης, η οποία είναι συνήθως σύντομη. Όταν ο νέος αυτός πράκτορας ξέρει αρκετά για να βοηθήσει τον πελάτη, ρίχνει την εσωτερική κλήση και ανακτά την κλήση πελατών. Όταν το πρόβλημα είναι πάρα πολύ δύσκολο για να το χειριστεί ο πράκτορας, μπορεί να επιλέξει να μεταφέρει την κλήση πελάτη σε έναν άλλο πράκτορα. Ο αρχικός πράκτορας εκτελεί έπειτα wrap-up χρόνο για την κλήση, η οποία τελειώνει την κλήση για εκείνο τον πράκτορα. Ο νέος πλέον πράκτορας απαντά στην κλήση πελατών. Η κλήση ακολουθεί το αρχικό διάγραμμα δραστηριότητας από εκείνο το σημείο και μετά, αλλά τώρα για το δεύτερο πράκτορα.



Σχήμα 16:Περίπτωση κλήσης με ACD router (Φάση 2)

3

Μοντελοποίηση του τηλεφωνικού κέντρου

Επόμενο σημαντικό βήμα στην ανάπτυξη του συστήματος του τηλεφωνικού κέντρου είναι να προσδιοριστεί αρχικά σε ποια κατηγορία μοντέλων ανήκει και έπειτα να παρουσιαστεί το ακριβές μοντέλο του. Με την μοντελοποίηση αυτή, θα δοθεί η δυνατότητα να αξιολογηθεί η λειτουργία και η απόδοση του και έτσι θα μπορέσουμε να διεξάγουμε χρήσιμα συμπεράσματα όσον αφορά το αν λειτουργεί σωστά και αποδοτικά, ποια προβλήματα παρουσιάζονται και πως θα αντιμετωπιστούν καθώς και άλλα παρόμοια ζωτικής σημασίας ερωτήματα.

Γενικά ένα τηλεφωνικό κέντρο μπορεί να χαρακτηριστεί σαν ένα σύστημα αναμονής (queueing systems). Αυτό συμβαίνει γιατί τα συστήματα αναμονής, βρίσκονται πίσω από τα περισσότερα μοντέλα μελέτης της απόδοσης υπολογιστικών συστημάτων, αφού φαινόμενα καθυστερήσεων λόγω της απαίτησης χρήσης περιορισμένων πόρων από πολλούς "πελάτες", μπορούν να εντοπίσουν τόσο σε απλούς υπολογιστές (πχ CPU, I/O, μνήμη), όσο και σε πολύπλοκα συστήματα όπως τα δίκτυα υπολογιστών.

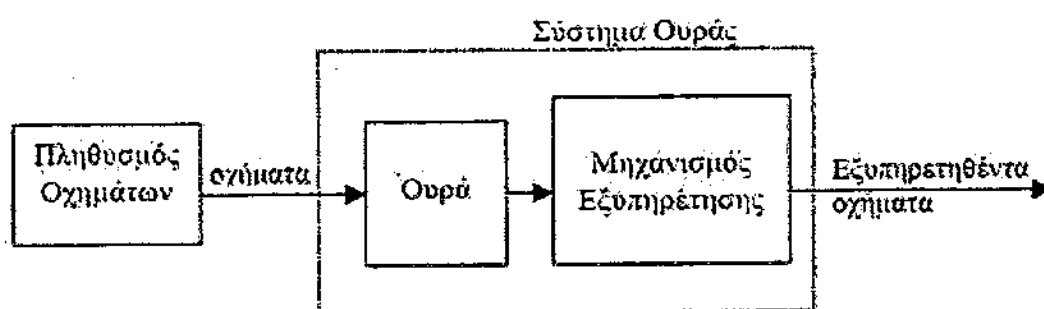
3.1 Τι είναι ένα σύστημα αναμονής

Σύστημα αναμονής είναι ένα σύστημα στο οποίο η τρέχουσα ζήτηση για εξυπηρέτηση είναι μεγαλύτερη από την τρέχουσα δυναμικότητα του συστήματος, που παρέχει την εξυπηρέτηση. Αυτό έχει σαν αποτέλεσμα να δημιουργούνται ουρές αναμονής στις οποίες παραμένουν αυτοί που θέλουν να εξυπηρετηθούν μέχρι το σύστημα να μπορέσει να τους εξυπηρετήσει.(9)

3.1.1 Βασική δομή των συστημάτων αναμονής

➤ Βασική διαδικασία ουράς

Η βασική διαδικασία που υποτίθεται στα περισσότερα πρότυπα ουράς είναι η ακόλουθη: οι μονάδες (οχήματα, αεροπλάνα, πλοία, πελάτες, κλπ.) που χρειάζονται εξυπηρέτηση δημιουργούνται από έναν *πληθυσμό μονάδων*. Οι μονάδες αυτές εισέρχονται σ' ένα *σύστημα ουράς* και ακολουθούν μια *ουρά*. Σε ορισμένες χρονικές στιγμές επιλέγεται μια μονάδα από την ουρά, για να εξυπηρετηθεί σύμφωνα με κάποιο κανόνα γνωστό ως *τρόπο εξυπηρέτησης* ή *πειθαρχία ουράς*. Η μονάδα εξυπηρετείται από ένα *μηχανισμό εξυπηρέτησης* και μετά φεύγει από το σύστημα ουράς. Η διαδικασία αυτή παρουσιάζεται στο σχήμα 1.(9)



Σχήμα 17: Βασική διαδικασία ουράς

➤ Πληθυσμός Μονάδων

Ένα από τα χαρακτηριστικά του πληθυσμού μονάδων είναι το μέγεθος του, δηλαδή ο συνολικός αριθμός μονάδων που χρειάζονται εξυπηρέτηση και που μπορεί να είναι ή περιορισμένος ή άπειρος. Επειδή οι υπολογισμοί είναι πιο εύκολοι για την περίπτωση του άπειρου αριθμού οχημάτων, η υπόθεση αυτή γίνεται συχνά και όταν το πραγματικό μέγεθος του πληθυσμού είναι ένας σχετικά μεγάλος πεπερασμένος αριθμός. Η περίπτωση του περιορισμένου αριθμού οχημάτων είναι πιο δύσκολη. Αυτό συμβαίνει επειδή ο αριθμός των οχημάτων που βρίσκεται ήδη στο σύστημα επηρεάζει τον αριθμό των πιθανών οχημάτων που είναι εκτός συστήματος και που θα προσέλθουν στο σύστημα. Παρόλα αυτά, η υπόθεση του περιορισμένου αριθμού

οχημάτων πρέπει να γίνεται όταν ο ρυθμός με τον οποίο προσέρχονται νέα οχήματα επηρεάζεται σημαντικά από τον αριθμό των οχημάτων που βρίσκονται ήδη στο σύστημα.

Ένα ακόμα χαρακτηριστικό που πρέπει να προσδιοριστεί είναι η στατική κατανομή από την οποία δημιουργούνται οι μονάδες διαχρονικά. Η υπόθεση που συνήθως γίνεται είναι ότι δημιουργούνται σύμφωνα με μια διαδικασία Poisson, δηλαδή, ο αριθμός των μονάδων που δημιουργούνται μέχρι μία συγκεκριμένη στιγμή ακολουθεί την κατανομή Poisson.

➤ Ουρά

Η ουρά χαρακτηρίζεται από το μέγιστο επιτρεπτό αριθμό μονάδων που μπορεί να συμπεριλάβει. Οι ουρές είναι *άπειρου* ή *περιορισμένου μήκους* σύμφωνα με το αν ο αριθμός αυτός είναι άπειρος ή πεπερασμένος. Η υπόθεση της ουράς απείρου μήκους γίνεται στα περισσότερα πρότυπα ουράς, ακόμα και στις περιπτώσεις όπου υπάρχει ένα σχετικά μεγάλο πεπερασμένο ανώτερο φράγμα στον επιτρεπτό αριθμό μονάδων.

➤ Τρόπος Εξυπηρέτησης

Ο τρόπος εξυπηρέτησης της ουράς αφορά στην προτεραιότητα με την οποία «επιλέγονται» οι μονάδες στην ουρά, για να εξυπηρετηθούν. Οι πιο συνηθισμένοι τρόποι εξυπηρέτησης είναι :

- Εξυπηρέτηση κατά προτεραιότητα αφίξεως, (First In- First Out, FIFO). Οι μονάδες εξυπηρετούνται κατά τη σειρά αφίξεως. Αποτελεί την πιο κοινή περίπτωση εξυπηρέτησεως για εφαρμογή στην κυκλοφοριακή τεχνική.

- Εξυπηρέτηση με αντίστροφη προτεραιότητα. (Last In- First Out, LIFO). Η μονάδα που φτάνει τελευταία εξυπηρετείται πρώτη.

- Τυχαία εξυπηρέτηση. (Service In Random Order, SIRO). Η εξυπηρέτηση γίνεται τυχαία.

- Ουρές με αποχωρήσεις (queue with balking or renegeing). Σ' αυτές μερικές μονάδες δεν θα παραμείνουν στην ουρά αν αυτή είναι μεγάλη αλλά θα αποχωρήσουν αμέσως μετά την άφιξή τους. Συμβαίνει σε ώρες αιχμής όταν το μήκος της ουράς και η διάρκεια αναμονής είναι αυξημένα.

- Περικαμμένες ουρές. (Truncated queues). Όταν υπάρχει περιορισμός στο χώρο αναμονής, οπότε όλες οι μονάδες που φθάνουν όταν η ουρά φθάσει στο μέγιστο δυνατό, αποχωρούν αμέσως μετά την άφιξή τους.

- Ουρές με προτεραιότητες (priority queues). Οι μονάδες που φθάνουν τοποθετούνται σε διάφορες ομάδες προτεραιοτήτων και η επιλογή γίνεται από την ομάδα με τον υψηλότερο βαθμό προτεραιότητας, όπου υπάρχουν μονάδες που αναμένουν.

➤ Σύστημα Εξυπηρέτησης

Το σύστημα εξυπηρέτησης αποτελείται από μία ή περισσότερες παράλληλες δίοδους εξυπηρέτησης. Σε ένα σύστημα εξυπηρέτησης κάθε μονάδα πηγαίνει σε μια δίοδο εξυπηρέτησης και εξυπηρετείται από τη θέση αυτή. Ο χρόνος που παραμένει η μονάδα στο σύστημα εξυπηρέτησης ονομάζεται χρόνος εξυπηρέτησης. Σε ένα συγκεκριμένο σύστημα ουράς πρέπει να προσδιοριστεί η κατανομή πιθανότητας των χρόνων εξυπηρέτησης για κάθε δίοδο. Η κατανομή του χρόνου εξυπηρέτησης, που συνήθως χρησιμοποιείται, είναι η εκθετική κατανομή.

3.1.2 Η οικονομική πλευρά των συστημάτων ουρών αναμονής

Σε γενικές γραμμές, οι ουρές είναι ένα πολύ συνηθισμένο φαινόμενο, που δημιουργείται όταν πχ οι πελάτες ενός συστήματος είναι περισσότεροι από τους εξυπηρετητές του συστήματος. Παρατηρείται, δηλαδή μια χαμηλή δυναμικότητα εξυπηρέτησης του συστήματος. Οι αποφάσεις που έχουν σχέση με τον προσδιορισμό της δυναμικότητας είναι πολλές φορές δύσκολες, επειδή είναι συχνά αδύνατο να προβλεφθεί επακριβώς πότε θα αφιχθούν οι μονάδες στο σύστημα για να εξυπηρετηθούν και πόσος χρόνος θα χρειαστεί για την εξυπηρέτησή τους. Η

ύπαρξη στο σύστημα μεγαλύτερης δυναμικότητας από αυτή που χρειάζεται, θα απαιτήσει μεγαλύτερο κόστος. Από την άλλη μεριά, αν είναι μικρότερη, θα δημιουργηθεί συνωστισμός και η ουρά θα μεγαλώνει με την πάροδο του χρόνου. Η ύπαρξη ουράς δημιουργεί και αυτή κόστος, όπως π.χ. κοινωνικό κόστος, κόστος διαρροής πελατών, κόστος αχρησιμοποίητων πόρων κ.α. Έτσι, αντικειμενικός σκοπός είναι να βρεθεί μία οικονομική ισορροπία μεταξύ του κόστους εξυπηρέτησης και του κόστους που συνδέεται με την αναμονή γι' αυτή την εξυπηρέτηση. Η θεωρία ουράς δε λύνει το πρόβλημα αυτό άμεσα. όμως δίνει την πληροφόρηση που χρειάζεται για μια τέτοια απόφαση, με το να προσδιορίζει τα διάφορα χαρακτηριστικά του συστήματος.

3.1.3 Συμβολισμός και Ορολογία

Εδώ παραπέιθονται τα βασικά σύμβολα και οι ορισμοί ενός συστήματος αναμονής οι οποίοι καλύπτουν όλο το φάσμα των διαφορετικών τύπων συστημάτων αναμονής.(10)

Σύμβολα:

- n : Πλήθος πελατών στο σύστημα κάποια χρονική στιγμή ($n(t)$).
- s : Πλήθος των θέσεων εξυπηρέτησης του συστήματος.
- λ : Μέσος ρυθμός αφίξεων στη μονάδα του χρόνου (λn).
- μ : Μέσος ρυθμός εξυπηρέτησης στη μονάδα του χρόνου (μn).
- ρ : Βαθμός απασχόλησης θέσεων εξυπηρέτησης.

Ορισμοί :

- P_n : Πιθανότητα να υπάρχουν n πελάτες στο σύστημα ($P_n(t)$).
- L_s : Μέσο πλήθος πελατών που εξυπηρετούνται.
- L : Μέσο πλήθος πελατών στο σύστημα.
- L_q : Μέσο πλήθος πελατών στην ουρά.
- W : Μέσος χρόνος παραμονής πελάτη στο σύστημα.
- W_q : Μέσος χρόνος αναμονής πελάτη στην ουρά.

3.1.4 Σχέσεις κατάστασης ισορροπίας κατά Little (10)

➤ $L = \lambda * W$ (το μέσο πλήθος πελατών στο σύστημα ισούται με τον μέσο ρυθμό αφίξεων στη μονάδα του χρόνου επί τον μέσο χρόνο παραμονής πελάτη στο σύστημα).

➤ Αντίστοιχα $L_q = \lambda * W_q$ (το μέσο πλήθος πελατών στην ουρά ισούται με τον μέσο ρυθμό αφίξεων στη μονάδα του χρόνου επί τον μέσο χρόνο αναμονής πελάτη στην ουρά).

➤ $L = L_q + L_s = L_q + \lambda / \mu$ (το μέσο πλήθος πελατών στο σύστημα ισούται με το μέσο πλήθος πελατών στην ουρά συν το πηλίκο του μέσου ρυθμού αφίξεων προς τον μέσο χρόνο εξυπηρέτησης. Στην ουσία αυτός ο τύπος δίνει τον αριθμό των πελατών στο σύστημα).

➤ $W = W_q + 1 / \mu$ (ο μέσος χρόνος παραμονής του πελάτη στο σύστημα ισούται με το μέσο πλήθος πελατών στην ουρά συν το πηλίκο του μέσου ρυθμού αφίξεων προς τον μέσο χρόνο εξυπηρέτησης. Στην ουσία αυτός ο τύπος δίνει τον χρόνο αναμονής στην ουρά).

➤ $\rho = \lambda / (s * \mu)$ (ο βαθμός απασχόλησης = ποσοστό χρόνου που είναι απασχολημένες οι θέσεις απασχόλησης).

3.2 Συνήθη πρότυπα μοντέλων αναμονής τα οποία συναντάμε στην τηλεπικοινωνία

3.2.1 Μοντέλο M/M/1

Το μοντέλο M/M1 είναι το πιο απλό σύστημα αναμονής. Το πρώτο M δείχνει τις αφίξεις των κλήσεων στο σύστημα. Το δεύτερο M δείχνει τον ρυθμό εξυπηρέτησης των κλήσεων και το 1 δείχνει ότι για την εξυπηρέτηση υπάρχει και ένας εξυπηρετητής. Οι γενικές παραδοχές του συστήματος αυτού είναι οι εξής:

1. Η πηγή των πελατών (κλήσεων) περιέχει άπειρους πελάτες (κλήσεις) ($N=\infty$).
2. Η διαδικασία αφίξεων των κλήσεων ακολουθεί την κατανομή Poisson με γνωστό μέσο ρυθμό αφίξεων στη μονάδα του χρόνου λ , ο οποίος παραμένει σταθερός κατά τη διάρκεια παρατήρησης. Ισοδύναμα, ο χρόνος που μεσολαβεί ανάμεσα σε διαδοχικές αφίξεις ακολουθεί εκθετική κατανομή με μέση τιμή $1/\lambda$.
3. Υπάρχει μία μόνο ουρά με πειθαρχία FIFO.
4. Οι πελάτες εισέρχονται πάντα στην ουρά και περιμένουν να εξυπηρετηθούν, χωρίς να αποχωρούν από αυτή.
5. Χωρητικότητα του συστήματος απεριόριστη ($k = \infty$).
6. Υπάρχει μία θέση εξυπηρέτησης ($s = 1$).
7. Η εξυπηρέτηση γίνεται σε μία φάση.

8. Ο χρόνος εξυπηρέτησης ακολουθεί την εκθετική κατανομή με γνωστή μέση τιμή $1/\mu$, που παραμένει σταθερή κατά η διάρκεια παρατήρησης. Ισοδύναμα, η διαδικασία (ρυθμός) εξυπηρέτησης ακολουθεί την κατανομή Poisson με μέση τιμή μ .

9. Ο μέσος ρυθμός εξυπηρέτησης είναι μεγαλύτερος από το μέσο ρυθμό αφίξεων ($\lambda < \mu$ & $\rho < 1$), οπότε το σύστημα οδηγείται σε κατάσταση ισορροπίας

3.2.2 Το μοντέλο $M(n)/M/s(0)$

Το σύστημα $M(n)/M/s(0)$ (σύστημα με εκθετική κατανομή άφιξης και εξυπηρέτησης των κλήσεων και s αριθμό εξερχόμενων γραμμών με μηδενική ουρά αλλά με πεπερασμένο αριθμό εισόδων n). Στο σύστημα αυτό θεωρούμε τα εξής:

- αριθμός των εισόδων του συστήματος έχει πεπερασμένη τιμή n .
- Οι αφίξεις των κλήσεων ακολουθούν εκθετική κατανομή με μέση τιμή $1/\lambda$, όπου λ είναι ο ρυθμός αφίξεων των κλήσεων στο σύστημα.
- Θεωρούμε ότι κάθε μία από τις εισερχόμενες n πεπερασμένες γραμμές είναι, όσον αφορά την κίνησή της, ανεξάρτητη από τις άλλες γραμμές. Επίσης θεωρούμε ότι κάθε μία από τις n γραμμές έχει την ίδια κίνηση.

3.2.3 Το μοντέλο $M/M/s$

Το επικρατέστερο μοντέλο αναμονής το οποίο χρησιμοποιείται για παρουσιάσει και να αξιολογήσει την απόδοση ενός συστήματος τηλεφωνικού κέντρου είναι το $M/M/s$ με $s > 1$. Το σύστημα αυτό είναι ένα σύστημα αναμονής στο οποίο χρησιμοποιούνται περισσότεροι από έναν εξυπηρετητές, οι οποίοι εργάζονται παράλληλα και ανεξάρτητα. Αυτό σημαίνει ότι οι πελάτες (τηλεφωνήματα) που φτάνουν στο σύστημα μας σχηματίζουν μια ουρά με την σειρά που καταφθάνουν. Όταν καταφθάνει ένας πελάτης στο σύστημα μας, και υπάρχουν περισσότερες από μια ελεύθερες διόδους, διαλέγει κάποια στην τύχη.

Παραδοχές – χαρακτηριστικά τα οποία ισχύουν σε ένα $M/M/s$, $s > 1$ σύστημα.

1. Η πηγή των πελατών περιέχει άπειρους πελάτες ($N=\infty$).
2. Η διαδικασία σφίξεων ακολουθεί κατανομή Poisson με γνωστό μέσο ρυθμό αφίξεων στην μονάδα του χρόνου λ , ο οποίος παραμένει σταθερός κατά τη διάρκεια της παρατήρησης. Ισοδύναμα, ο χρόνος που μεσολαβεί ανάμεσα σε δυο διαδοχικές αφίξεις ακολουθεί εκθετική κατανομή με μέση τιμή $1/\lambda$.
2. Υπάρχει μια ουρά με πειθαρχία FIFO (First Come First Out), σύμφωνα με την οποία οι πελάτες εξυπηρετούνται σύμφωνα με την σειρά άφιξης τους.
3. Οι πελάτες μπαίνουν πάντα στην ουρά και περιμένουν να εξυπηρετηθούν, χωρίς να αποχωρούν από αυτήν.
4. Η χωρητικότητα του συστήματος είναι απεριόριστη ($K=\infty$).
5. Υπάρχουν περισσότερες από μια παράλληλες θέσεις εξυπηρέτησης.
6. Η εξυπηρέτηση γίνεται σε μια φάση σε μια από τις παράλληλες θέσεις εξυπηρέτησης.
7. Ο χρόνος εξυπηρέτησης σε κάθε θέση ακολουθεί την ίδια εκθετική κατανομή με γνωστή μέση τιμή $1/\mu$, που παραμένει σταθερή κατά τη διάρκεια της παρατήρησης. Άρα και ο ρυθμός εξυπηρέτησης κάθε θέσης ακολουθεί την ίδια κατανομή Poisson με μέση τιμή μ .
8. Ο συνολικός μέσος ρυθμός εξυπηρέτησης όλων των θέσεων μαζί είναι μεγαλύτερος από το μέσο ρυθμό αφίξεων ($\lambda < s \times \mu$! $\rho < 1$) οπότε το σύστημα οδηγείται σε κατάσταση ισορροπίας.

Δείκτες απόδοσης του μοντέλου M/M/s, s>1

$$\rho = \lambda / \mu$$

$$L_x = \lambda / \mu$$

$$L = L_q + L_s = L_q + \lambda / \mu$$

$$W = W_q + 1 / \mu = L / \lambda$$

$$W_q = L_q / \lambda$$

$$P_0 = \frac{1}{\sum_{n=0}^{s-1} \left[\frac{(\lambda / \mu)^n}{n!} \right] + \frac{(\lambda / \mu)^s}{s!} \left(\frac{s\mu}{s\mu - \lambda} \right)}$$

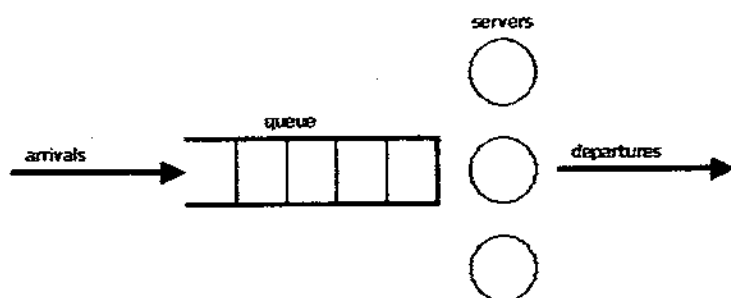
$$P_n = \begin{cases} \frac{1}{n!} \left(\frac{\lambda}{\mu} \right)^n P_0 & n \leq s \\ \frac{1}{s! s^{n-s}} \left(\frac{\lambda}{\mu} \right)^n P_0 & n > s \end{cases}$$

$$L_q = \frac{\left(\frac{\lambda}{\mu} \right)^s \lambda \mu}{(s-1)! (s\mu - \lambda)^2} P_0$$

$$P_w = 1 - \sum_{n=0}^{s-1} P_n$$

3.3 Επιλογή μοντέλου για το τηλεφωνικό κέντρο

Το σύστημα του τηλεφωνικού κέντρου που παρουσιάστηκε και σχεδιάστηκε σε προηγούμενο κεφάλαιο συμφωνεί απόλυτα με το $M/M/s$, $s>1$ μοντέλο αναμονής. Το s είναι γνωστό και είναι όσο με 3 αφού στο σύστημα χρησιμοποιούνται τρία τηλέφωνα, πράγμα που σημαίνει τρεις εξυπηρετητές. Συνεπώς οι κλήσεις οι οποίες εισέρχονται στο σύστημα θα εξυπηρετούνται από έναν από τους τρεις εξυπηρετητές εφόσον αυτοί δεν είναι ήδη απασχολημένοι. Αλλιώς θα εισέρχονται στην ουρά αναμονής και θα εξυπηρετείται η πρώτη κλήση που ήρθε στο σύστημα. Το παρακάτω σχήμα δείχνει την μορφή του.



Σχήμα 18: Περίπτωση ουράς αναμονής

Σε αυτό το σημείο δίνεται ένα παράδειγμα λειτουργίας και μέτρησης της απόδοσης του συστήματος μέσω των δεικτών απόδοσης.

Παράδειγμα :

Έστω ότι στο τηλεφωνικό μας κέντρο τις πρωινές ώρες οι οποίες είναι και ώρες αιχμής, δέχεται με ρυθμό $\lambda=72$ τηλεφωνήματα (πελάτες) την ώρα (Poisson) και σχηματίζουν ουρά αναμονής. Οι τρεις εξυπηρετητές που έχουμε στο σύστημα μας έχουν μέσο χρόνο εξυπηρέτησης 1 λεπτό ανά πελάτη (εκθετική κατανομή). Να μελετήσουμε την απόδοση του συστήματος με βάση τους δείκτες απόδοσης.

Λύση:

Πρώτον βλέπουμε πως το $\lambda=72$ άτομα ανά ώρα, το οποίο μας δείχνει το μέσο ρυθμό αφίξεων των κλήσεων στο σύστημα μας ανά ώρα.

Δεύτερον βλέπουμε ότι το $\mu=1$ ανά λεπτό, το οποίο μας δείχνει ότι ο μέσος ρυθμός εξυπηρέτησης από τους εξυπηρετητές είναι ένα άτομο ανά λεπτό.

Θα πρέπει οι δυο παράμετροι μ και λ να αναφέρονται στην ίδια μονάδα χρόνου, για αυτό θα μετατρέψουμε το $\mu=1$ άτομο ανά λεπτό σε $\mu=60$ άτομα ανά ώρα. Επομένως, αν παραμείνει το $\mu=1$ άτομα το λεπτό ισοδύναμα θα πρέπει να θέσουμε $\lambda=1.2$ άτομα ανά λεπτό. Έτσι έχουμε:

$\rho = \lambda / s * \mu \Leftrightarrow \rho = 72 / 180 \Leftrightarrow \rho = 0,40$ που είναι και ο βαθμός απασχόλησης του συστήματος

Έστω P_0 η πιθανότητα να μην υπάρχει κανείς πελάτης στο σύστημα. Έχουμε λοιπόν

$$P_0 = \frac{1}{\sum_{n=0}^2 \left[\frac{(72/60)^n}{n!} \right] + \frac{(72/60)^3}{3!} \left(\frac{3 \cdot 60}{3 \cdot 60 - 72} \right)} = \frac{1}{\frac{(72/60)^0}{0!} + \frac{(72/60)^1}{1!} + \frac{(72/60)^2}{2!} + \frac{(72/60)^3}{3!} \times \frac{180}{108}} = 0.294118$$

Επομένως βλέπουμε ότι: $P_0 = 29.41\%$

Στην συνέχεια ας βρούμε το μέσο πλήθος πελατών στην ουρά αναμονής του συστήματος. Έστω λοιπόν L_q το μέσο πλήθος πελατών στην ουρά αναμονής του συστήματος. Είναι:

$$L_q = \frac{\left(\frac{\lambda}{\mu}\right)^s \lambda \mu}{(s-1)!(s\mu - \lambda)^2} P_0 = \frac{1}{2!} \left(\frac{72}{60}\right)^3 \frac{72 \cdot 60}{(3 \cdot 60 - 72)^2} \cdot P_0 = 0.0941$$

$$L = L_q + \lambda / \mu = 0,0941 + 72 / 60 = 1,2941 \text{ πελάτες}$$

Επομένως, ο μέσος αριθμός πελατών στο σύστημα είναι $L = 1.2941 \approx 3$ πελάτες

Έστω W_q ο μέσος χρόνος αναμονής του πελάτη στην ουρά. Είναι:

$$W_q = L_q / \lambda = 0,0941 / 72 = 0,0013 \text{ ώρες (= 0,078 λεπτά)}$$

$$W = L / \lambda = 1,2941 / 72 = 0,01797 \text{ ώρες (1,078 λεπτά)}$$

Παρατηρούμε ότι ο μέσος χρόνος παραμονής του κάθε πελάτη στο σύστημα είναι

$$W = 1.078 \text{ λεπτά} \approx 64.68 \text{ δευτερόλεπτα}$$

Ερώτημα: Ποια όμως είναι η πιθανότητα ένας πελάτης να εξυπηρετηθεί αμέσως;

Έχουμε: $P_0 + P_1 + P_2$

Είναι:

$$P_1 = \frac{(72/60)^1}{1!} \cdot P_0 = 0.3529$$

$$P_2 = \frac{(72/60)^2}{2!} \cdot P_0 = 0.2118$$

Οπότε έχουμε:

$$P_0 + P_1 + P_2 = 0.2941 + 0.3529 + 0.2118 = 0.8588 = 85.88\% \Rightarrow$$

$$1 - (P_0 + P_1 + P_2) = 1 - 0.8588 = 0.1412 = 14.12\%$$

Επομένως η πιθανότητα ένας πελάτης να εξυπηρετηθεί αμέσως από το σύστημα είναι 14.12%

3.4 Χρονοπρογραμματισμός του τηλεφωνικού κέντρου

Η ανάλυση που θα ακολουθεί έχει σαν στόχο να απαντηθεί το εξής ερώτημα: με ποιον τρόπο μια διεργασία επιλέγεται από την ΚΜΕ του συστήματος ώστε να γίνει η εκτέλεση της;

Το τηλεφωνικό κέντρο το οποίο παρουσιάστηκε παραπάνω δέχεται διαρκώς κλήσεις οι οποίες αποτελούν τις διεργασίες και οι οποίες θα πρέπει να ανατεθούν στην ΚΜΕ του συστήματος για την εκτέλεση τους. Αρχικά οι διεργασίες αυτές που πρέπει να εκτελεστούν βρίσκονται αποθηκευμένες όλες μαζί σε κάποια μονάδα μαζικής αποθήκευσης (π.χ. το σκληρό δίσκο). Από αυτές επιλέγεται μια ομάδα διεργασιών, οι οποίες θα εναλλάσσονται εκτελούμενες στην ΚΜΕ. Έπειτα οι διεργασίες που έχουν επιλεγεί βρίσκονται πλέον στην κύρια μνήμη και είναι έτοιμες για εκτέλεση.

Στην ουσία, το πρόβλημα και η απάντηση στο ερώτημα που τέθηκε είναι με ποιον τρόπο θα γίνει ο χρονοπρογραμματισμός του συστήματος δηλαδή με ποιον τρόπο θα γίνει η χρονοδρομολόγηση των διεργασιών στην ΚΜΕ. Το πρόβλημα αυτό θα λυθεί με την χρησιμοποίηση ενός από των υπάρχοντων αλγορίθμων χρονοπρογραμματισμού. Ένας τέτοιος περιγράφει με σαφήνεια τον τρόπο που επιλέγεται μια διεργασία για να την προωθηθεί στην ΚΜΕ ή να την απομακρυνθεί από αυτή, πόσο χρόνο θα απασχολεί κάθε διεργασία την ΚΜΕ κλπ.

Βασικός στόχος του αλγορίθμου χρονοπρογραμματισμού που θα χρησιμοποιηθεί είναι:

- ✓ Το υπολογιστικό σύστημα πρέπει να έχει όσο το δυνατόν καλύτερη απόδοση σύμφωνα με κάποια κριτήρια αξιολόγησης.
- ✓ Ο ίδιος ο αλγόριθμος πρέπει να λαμβάνει τις αποφάσεις του όσο πιο γρήγορα γίνεται, ιδιαίτερα όταν το πλήθος των διεργασιών είναι μεγάλο, για να προσθέτει την ελάχιστη επιβάρυνση στο σύστημα.

3.5 Επιλογή αλγορίθμου

Καταρχάς θα πρέπει να τονιστεί ότι ο αλγόριθμος που θα χρησιμοποιηθεί θα πρέπει να είναι ένας **non preemptive** αλγόριθμος.

Το χαρακτηριστικό ενός **non preemptive** αλγορίθμου είναι ότι διεργασία που επιλέγεται να πάρει τον έλεγχο της ΚΜΕ, τον διατηρεί έως ότου αυτή ολοκληρωθεί, ή απαιτήσει κάποια άλλη λειτουργία. Ο αλγόριθμος αυτός δεν έχει την δύναμη να διακόψει την εκτέλεση της διεργασίας και να αναθέσει την ΚΜΕ σε μια άλλη η οποία μπορεί να έχει πιο υψηλό δείκτη προτεραιότητας από την τρέχουσα.

Αυτό συμβαίνει για τον εξής λόγο: το σύστημα του τηλεφωνικού κέντρου δέχεται διαρκώς κλήσεις από πελάτες που επιθυμούν να εξυπηρετηθούν. Όταν ένας πελάτης διατεθεί σε έναν agent θα συνεχίσει να εξυπηρετείται μέχρι αυτός να ικανοποιήσει όλες τις απαιτήσεις του. Ο μόνος τρόπος διακοπής είναι ο ίδιος ο πελάτης να διακόψει από μόνος του την διεργασία κλείνοντας το τηλέφωνο. Δεν μπορεί δηλαδή η ΚΜΕ να διακόψει την εξυπηρέτηση ενός πελάτη από έναν εξυπηρετητή και να διαθέσει έναν νέο πελάτη που εισήλθε στο σύστημα.

Οι κυριότεροι **non preemptive** αλγόριθμοι είναι δυο και είναι οι εξής :

1. First Come First Served αλγόριθμος. Ο αλγόριθμος ονομάζεται και **αλγόριθμος με βάση την σειρά άφιξης** των διεργασιών στο σύστημα. Οι βασικές αρχές που διέπουν τον αλγόριθμο παρουσιάζονται παρακάτω και είναι οι εξής:

➤ Οι διεργασίες εξυπηρετούνται με τη σειρά που φθάνουν. Κάθε νέα διεργασία που ετοιμάζεται να εκτελέσει την επόμενη έκρηξή της στην ΚΜΕ εισάγεται στο τέλος της λίστας έτοιμων διεργασιών. Όταν έρθει η ώρα να εισαχθεί στην ΚΜΕ μια νέα διεργασία, επιλέγεται αυτή που είναι στην αρχή της λίστας, προωθείται στην ΚΜΕ και εκεί εκτελεί ολόκληρη την έκρηξή της στην ΚΜΕ.

➤ Ακόμη και αν οι διεργασίες φθάνουν την ίδια χρονική στιγμή, η σειρά άφιξης είναι διακριτή, αλλιώς η διεργασία που θα εξυπηρετηθεί κατά προτεραιότητα επιλέγεται τυχαία

➤ Κατάσταση απόφασης : χωρίς προεκχώρηση



- Η διεργασία εκτελείται μέχρι να ανασταλεί από μόνη της
- Είναι πολύ απλός αλγόριθμος και υλοποιείται εύκολα αλλά είναι ακατάλληλος για συστήματα πολλών χρηστών.

Οι ιδιότητες του First Come First Served αλγορίθμου είναι οι ακόλουθες:

- Χρησιμοποιεί μια ουρά FIFO. Δηλαδή η πρώτη διεργασία που θα εισέρθει στο σύστημα θα είναι και η πρώτη που θα εξυπηρετηθεί.
- Η επιλογή της επόμενης διεργασίας είναι ταχύτατη και ανεξάρτητη από το πλήθος των διεργασιών στην ουρά των έτοιμων διεργασιών
- Συχνά προκύπτουν μεγάλοι χρόνοι αναμονής και απόκρισης

Τα μειονεκτήματα τα οποία παρουσιάζει ο αλγόριθμος First Come First Served είναι τα εξής:

- Μια διεργασία που δεν χρησιμοποιεί I/O θα μονοπωλεί τη χρήση του επεξεργαστή
- Ευνοούνται οι προοριζόμενες για τη CPU διεργασίες.
- Οι προοριζόμενες για I/O διεργασίες θα περιμένουν μέχρι να ολοκληρωθούν οι προοριζόμενες για τη CPU διεργασίες. Θα περιμένουν ακόμη και αν οι I/O λειτουργίες τους έχουν ολοκληρωθεί
- Υπάρχουν μεγάλες διακυμάνσεις στον μέσο χρόνο επιστροφής
- Είναι ακατάλληλος αλγόριθμος για αλληλεπιδραστικά συστήματα.
- Η επίδοση του αλγορίθμου αυτού αν κριθεί με βάση το χρόνο ανακύκλωσης (turnaround time) είναι χαμηλή, γιατί εξαναγκάζει ακόμα και διεργασίες με πολύ σύντομες εκρήξεις ΚΜΕ να αναμένουν για την εκτέλεση άλλων. Παρόμοιες επιδόσεις πετυχαίνει και με βάση τα υπόλοιπα κριτήρια αξιολόγησης που

αναφέρθηκαν στο μάθημα αυτό· η χρησιμοποίηση π.χ. της ΚΜΕ μπορεί να είναι χαμηλή.(11)

Ο αλγόριθμος εξυπηρέτησης με βάση τη διάρκεια **Shortest Job First - SJF**, κατατάσσει τις διεργασίες κατά αύξουσα σειρά διάρκειας. Πρώτα εκτελείται η συντομότερη διεργασία, μετά αυτή με την αμέσως μεγαλύτερη διάρκεια. Τελευταία εκτελείται αυτή με τη μεγαλύτερη διάρκεια. Οι βασικές αρχές του είναι οι εξής:

- Επιλέγεται η διεργασία με το μικρότερο χρόνο καταιγισμού στη CPU
- Μια διεργασία που εισάγεται στη λίστα έτοιμων διεργασιών, όταν μια άλλη ήδη εκτελείται, θα τοποθετηθεί στο κατάλληλο σημείο ανάλογα με τη διάρκεια της έκρηξης της στην ΚΜΕ, και θα εκτελεστεί πριν από κάποια άλλη που είχε εισαχθεί στη λίστα νωρίτερα αλλά είχε μεγαλύτερη διάρκεια.

- Ενσωματώνει αναμφίβολα προτεραιότητες στις διεργασίες: οι συντομότερες

- Οι διεργασίες έχουν δεδομένη προτεραιότητα.
- Αποτελεί μια βελτίωση του αλγορίθμου F.C.F.S.
- Απαιτείται ο υπολογισμός του χρόνου καταιγισμού (CPU burst time) για κάθε διεργασία

Οι ιδιότητες του **Shortest Job First – SJF** αλγορίθμου

- Δίνει πολύ καλύτερο μέσο χρόνο αναμονής σε σχέση με τον αλγόριθμο FCFS

- Χωρίς προεκχώρηση ωστόσο η έλλειψη προεκχώρησης δεν είναι κατάλληλη σε ένα περιβάλλον καταμερισμού χρόνου

- Απαιτείται η γνώση των χρόνων καταιγισμού στη CPU που γενικά είναι δύσκολο και συνήθως ακατόρθωτο

- Η επιλογή είναι περισσότερο σύνθετη από αυτήν του FCFS

- Είναι πιθανή η παρατεταμένη στέρηση

- Αν νέες μικρής διάρκειας διεργασίας φθάνουν στο σύστημα οι προγενέστερες, μεγάλης διάρκειας διεργασίες, δεν θα εξυπηρετηθούν ποτέ.

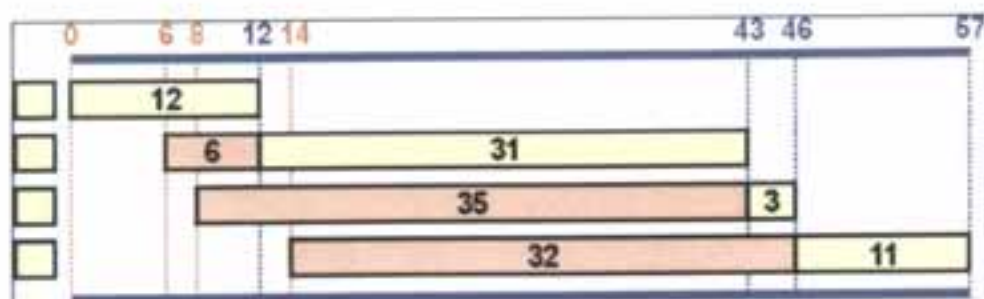
Στην συνέχεια θα δοθεί ένα παράδειγμα το οποίο θα γίνει σύγκριση των επιδόσεων των δυο αλγορίθμων. (11)

Ας υποθέσουμε λοιπόν ότι οι τέσσερις διεργασίες προς εξυπηρέτηση είναι οι δ1, δ2, δ3 και δ4. Η δ1 φθάνει στη λίστα έτοιμων διεργασιών τη χρονική στιγμή 0 και έχει διάρκεια 12 χρονικές μονάδες, η δ2 φθάνει τη στιγμή 6 και έχει διάρκεια 31, η δ3 φθάνει τη στιγμή 8 και έχει διάρκεια 3 και η δ4 φθάνει τη στιγμή 14 και έχει διάρκεια 11. Επιπλέον γίνεται η παραδοχή ότι οι διεργασίες απασχολούν μόνο την ΚΜΕ και δεν εκτελούν λειτουργίες Ε/Ε.

Διεργασία	Αφιξη	Χρόνος ξεσπάσματος στην ΚΜΕ
Δ1	0	12
Δ2	6	31
Δ3	8	3
Δ4	14	11

Υπολογισμοί απόδοσης με βάση τον First Come First Served αλγόριθμο.

Στο σχήμα 19 φαίνεται το σχεδιάγραμμα εκτέλεσης των τεσσάρων διεργασιών του παραδείγματος. Οι χρόνοι αναμονής είναι 0 για την πρώτη διεργασία, $12 - 6 = 6$ για τη δεύτερη, $12 + 31 - 8 = 35$ για την τρίτη και $12 + 31 + 3 - 14 = 32$ για την τέταρτη. Ο μέσος χρόνος αναμονής είναι λοιπόν $(0 + 6 + 35 + 32) / 4 = 18.25$



Σχήμα 19: Σχεδιάγραμμα εκτέλεσης των τεσσάρων διεργασιών του παραδείγματος.

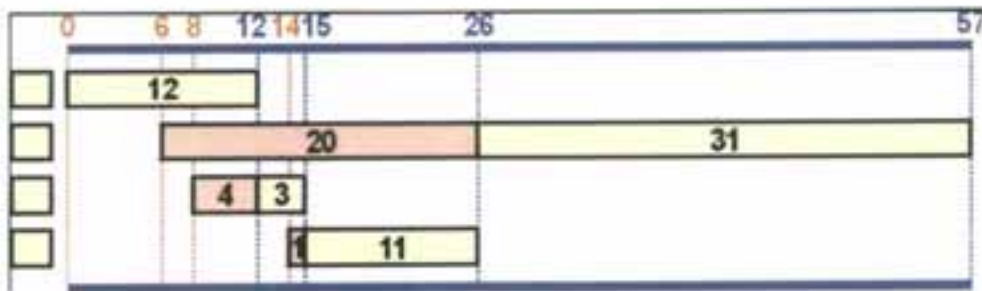
Παρατηρούμε επίσης ότι η διεργασία Δ3, η οποία είχε διάρκεια μόνο 3, εξαναγκάστηκε να περιμένει για 35 χρονικές μονάδες για να εκτελεστεί, είχε δηλαδή τεράστια αναμονή σε σχέση με τη διάρκειά της.

Οι χρόνοι απόκρισης (χρόνος αναμονής + χρόνος εκτέλεσης) είναι: 12 για τη δ1, 37 για τη δ2, 38 για τη δ3 και 43 για τη δ4. Έτσι ο μέσος χρόνος απόκρισης είναι $(12 + 37 + 38 + 43) / 4 = 32.5$

Υπολογισμοί απόδοσης με βάση τον Shortest Job First – SJF αλγόριθμο.

Η εκτέλεση των διεργασιών του παραδείγματος με τον αλγόριθμο εξυπηρέτησης με βάση τη διάρκεια, Shortest Job First – SJF φαίνεται στο παρακάτω σχήμα 20. Η σειρά εκτέλεσης των διεργασιών είναι Δ1, Δ3, Δ4, Δ2. Η Δ1 επειδή είναι μόνη της πρώτη δεν περιμένει καθόλου, η Δ2 περιμένει για $(12 + 3 + 11) - 6 = 20$, η Δ3 για $(12 - 8) = 4$ και η Δ4 για $(12 + 3) - 14 = 1$. Ο μέσος χρόνος αναμονής είναι λοιπόν $(0 + 20 + 4 + 1) / 4 = 6.25$ πολύ μικρότερος από τον αντίστοιχο για την εξυπηρέτηση με βάση τη σειρά άφιξης, και ο μικρότερος δυνατός.

Οι χρόνοι απόκρισης είναι: 12 για τη δ1, 51 για τη δ2, 7 για τη δ3 και 12 για τη δ4. Ο μέσος χρόνος απόκρισης είναι $(12 + 51 + 7 + 12) / 4 = 20.5$, πολύ καλύτερος επίσης από την εξυπηρέτηση με βάση τη σειρά άφιξης.



Σχήμα 20:Χρόνοι απόκρισης

Επιλογή του αλγορίθμου χρονοπρογραμματισμού

Παρόλο που ο αλγόριθμος Shortest Job First – SJF έχει καλύτερα αποτελέσματα από τον First Come First Served αλγόριθμο αφού και μικρότερο μέσο χρόνο αναμονής έχει αλλά και μεγαλύτερο μέσο χρόνο απόκρισης ο αλγόριθμος που επιλέγεται για τον χρονοπρογραμματισμό του συστήματος του τηλεφωνικού κέντρου είναι ο δεύτερος(First Come First Served). Αυτό γίνεται γιατί ο αλγόριθμος Shortest Job First εμφανίζει μια απροσπέλαστη δυσκολία.

Η δυσκολία στην υλοποίηση του αλγορίθμου αυτού είναι η εκτίμηση της διάρκειας για την έκρηξη ΚΜΕ κάθε διεργασίας. Δεν είναι εύκολο να υπολογιστεί εκ των προτέρων η επόμενη έκρηξη ΚΜΕ για μια διεργασία μπορεί βέβαια να εκτιμηθεί στατιστικά με βάση τις προηγούμενες εκρήξεις ΚΜΕ, αλλά η εκτίμηση αυτή δεν είναι ακριβής και επιπλέον απαιτεί υπολογιστικό χρόνο. Ο λόγος ο οποίος δεν μπορεί να υπολογιστεί η διάρκεια έκρηξης της ΚΜΕ για μια διεργασία είναι γιατί το σύστημα δεν μπορεί να γνωρίζει πόσο κάθε πελάτης ο οποίος εισέρχεται στο σύστημα μέσω μιας κλήσης θα απασχολήσει την ΚΜΕ. Για το λόγο αυτό το λειτουργικό σύστημα του τηλεφωνικού κέντρου έχει σαν βάση προώθησης των διεργασιών στην ΚΜΕ την σειρά με την οποία οι διεργασίες (κλήσεις) φθίνουν στο σύστημα.

Συνεπώς, παρόλο που ο First Come First Served αλγόριθμος εμφανίζει μεγάλους μέσους χρόνους αναμονής με αποτέλεσμα να μένουν για μεγάλο χρονικό διάστημα στην ουρά και οι διεργασίες εκείνες οι οποίες απαιτούν μικρές εκρήξεις της ΚΜΕ, είναι ο αλγόριθμος με τον οποίο θα γίνει ο χρονοπρογραμματισμός του συστήματος.

Συμπεράσματα

Στη παράγραφο αυτή παρουσιάζονται τα συμπεράσματα τα οποία αφορούν την μοντελοποίηση και τον χρονοπρογραμματισμό του συστήματος του τηλεφωνικού κέντρου, τόσο από την μελέτη της σχετικής βιβλιογραφίας όσο και από τη διαδικασία σχεδιασμού και υλοποίησης του συστήματος.

Ένα πρώτο γενικό συμπέρασμα αφορά την σπουδαιότητα και την χρησιμότητα ενός συστήματος τηλεφωνικού κέντρου. Η σπουδαιότητα του έγκειται στο ότι θα δώσει στην επιχείρηση που διαθέτει στις τάξεις της τηλεφωνικό κέντρο, ένα προβάδισμα έναντι των ανταγωνιστών της για την κατάκτηση της αγοράς. Αυτό οφείλεται στο ότι τώρα η επιχείρηση θα είναι σε θέση να γνωρίζει καλύτερα τις απαιτήσεις των πελατών της ώστε να μπορεί να προσαρμόσει και τα προϊόντα της όσο περισσότερο πιο κοντά γίνεται στις προτιμήσεις – απαιτήσεις των πελατών. Αντίθετα οι επιχειρήσεις που δεν θα προβούν σε ένα τέτοιο εγχείρημα θα αντιμετωπίσουν προβλήματα τα οποία με τον καιρό μπορεί να αφορούν και την βιωσιμότητα της ίδιας της επιχείρησης.

Βέβαια η ανάπτυξη και η υλοποίηση ενός τηλεφωνικού κέντρου δεν αποτελεί ένα εύκολο εγχείρημα. Υπάρχουν πολλά ερωτήματα τα οποία θα πρέπει να απαντηθούν. Ένα από αυτά είναι ποια μεθοδολογία θα χρησιμοποιηθεί για την ανάπτυξη του .

Για την ανάπτυξη ενός συστήματος τηλεφωνικού κέντρου χρησιμοποιείται η αντικειμενοστρεφή ανάλυση και σχεδίαση. Οι λόγοι που προτιμάμε από την δομημένη ανάλυση είναι οι εξής :

- Γίνεται καλύτερα ο προσδιορισμός των απαιτήσεων από το σύστημα.

- Τα δεδομένα και οι λειτουργίες του συστήματος ομαδοποιούνται σε κλάσεις ανάλογα με τα χαρακτηριστικά τους
- Τα δεδομένα αυτά αποτελούν κυρίαρχα στοιχεία και δεν είναι ανεξάρτητα από τις λειτουργίες όπως γίνεται στην δομημένη ανάλυση και
- Στην αντικειμενοστρεφή ανάλυση υπάρχει πληθώρα στην περιγραφή των σχέσεων των οντοτήτων του συστήματος καθώς και των δραστηριοτήτων που λαμβάνουν χώρα μεταξύ αυτών.

Για την αποτύπωση αυτών των σχέσεων και των αλληλεπιδράσεων μεταξύ τους χρησιμοποιείται από την αντικειμενοστρεφή ανάλυση η UML. Η UML είναι ένα πολύτιμο εργαλείο για την σχεδίαση συστημάτων. Με αυτήν μπορούν να σχεδιαστούν και να αποτυπωθούν όλες οι σχέσεις, οι ροές δεδομένων και οι λειτουργίες γενικότερα του συστήματος. Αυτό γίνεται με την χρήση των ποικίλων διαγραμμάτων που έχει και καλύπτουν ολόκληρο το σχεδιαστικό φάσμα του συστήματος αφού μπορούν με την χρήση του κατάλληλου διαγράμματος να αποτυπωθούν λεπτομερώς όλες οι σχέσεις και λειτουργίες των οντοτήτων που αποτελούν το σύστημα. Απόρροια αυτού είναι να απλουστεύεται περισσότερο η υλοποίηση του συστήματος.

Ένα δεύτερο συμπέρασμα αφορά το είδος του συστήματος το οποίο είναι το τηλεφωνικό κέντρο. Κατά την υλοποίηση του συστήματος μπορεί να προκύψουν προβλήματα λειτουργίας τα οποία θα πρέπει να αντιμετωπιστούν. Για να γίνει ακριβέστερη η αντιμετώπιση των προβλημάτων αυτών και να μπορεί να γίνεται και πρόβλεψη για μελλοντικά προβλήματα το σύστημα του τηλεφωνικού κέντρου θα πρέπει να αντιμετωπίζεται σαν ένα μοντέλο που ανήκει σε μια ευρύτερη κατηγορία μοντέλων. Για το λόγο αυτό το τηλεφωνικό κέντρο αντιμετωπίζεται σαν ένα σύστημα αναμονής. Έτσι, γνωρίζοντας σε γενικά πλαίσια ποια είναι τα θετικά και τα αρνητικά στη λειτουργία ενός συστήματος αναμονής μπορεί να προβλεφθεί και τι προβλήματα θα παρουσιάσει η λειτουργία του συστήματος του τηλεφωνικού κέντρου. Με αυτό δίνεται η δυνατότητα εύρεσης τρόπων και μεθόδων για εξάλειψη αυτών των προβλημάτων ή περιορισμού τους.

Ένα τελευταίο συμπέρασμα αφορά τον τρόπο με τον οποίο οι διεργασίες δηλαδή τα τηλεφωνήματα διανέμονται στην κεντρική μονάδα επεξεργασίας του συστήματος. Αυτό είναι ένα θέμα που αφορά τον τρόπο με τον οποίο θα γίνει ο χρονοπρογραμματισμός του συστήματος του τηλεφωνικού κέντρου. Το συμπέρασμα που διεξάγεται είναι η χρησιμοποίηση ενός αλγορίθμου. Ο αλγόριθμος αυτός θα δίνει τις διεργασίες στην ΚΜΕ για την εκτέλεση τους.

Ειδικότερα ο καταλληλότερος αλγόριθμος είναι ο γνωστός First Come First Served αλγόριθμος χρονοπρογραμματισμού. Η επιλογή αυτού οφείλεται στα εξής :

- είναι απλός και εύκολος στην υλοποίηση
- δίνει έμφαση στην σειρά που καταφθάνουν οι διεργασίες στο σύστημα και
- είναι Non-preemptive δηλαδή οι διεργασίες δεν διακόπτονται δηλαδή το κάθε τηλεφώνημα που έρχεται στο τηλεφώνημα δεν διακόπτεται πρόωρα.

Παραπομπές

- 1) Κιουντούζης Ευάγγελος, « Μεθοδολογίες Ανάλυσης κ Σχεδιασμού Πληροφοριακών Συστημάτων (Α΄ Έκδοση 1997)», Εκδοτικός Οίκος Μπένου, κεφάλαιο 1
- 2) Κιουντούζης Ευάγγελος, « Μεθοδολογίες Ανάλυσης κ Σχεδιασμού Πληροφοριακών Συστημάτων (Α΄ Έκδοση 1997)», Εκδοτικός Οίκος Μπένου, κεφάλαιο 1
- 3) Ηλεκτρονική διεύθυνση σελίδα
http://www.softlab.ntua.gr/~bxb/courses/unipi2001_te/00-CourseNotes/031-OO&Java/TE031-1.pdf
- 4) Ηλεκτρονική διεύθυνση σελίδα
eos.uom.gr/~achat/softfiles/UML_Review.ppt
- 5) Ηλεκτρονική διεύθυνση σελίδα
<http://cqi.di.uoa.gr/~afrodite/Dynamic-Diagrams.ppt>
- 6) Ηλεκτρονική διεύθυνση σελίδα
<http://www.dmst.aueb.gr/dds/ism/oo/indexw.htm>
- 7) Ηλεκτρονική διεύθυνση σελίδα
<http://support.inf.uth.gr/courses/CE222/scheduling-280306-pl.pdf#search=%22cpu%20scheduling%22>
- 8) Ηλεκτρονική διεύθυνση σελίδα
<http://people.msoe.edu/~taylor/cs384/mellottk.pdf>

9) Ηλεκτρονική διεύθυνση σελίδα

<http://www.it.civil.ntua.gr/kyklotex/queues%202001.pdf>

10) Ηλεκτρονική διεύθυνση σελίδα

www.enman.uowm.gr/dep/ts/lect12-13.ppt

11) Ηλεκτρονική διεύθυνση σελίδα

www.it.teithe.gr/~stefanos/OPS-10.pdf

ΒιβλιογραφίαΕλληνική βιβλιογραφία

- Υψηλάντης Παντελής Γ. ,Επιχειρησιακή Έρευνα, Λήψη Επιχειρηματικών Αποφάσεων, Εκδόσεις «ΕΛΛΗΝ», 1998
- Κοίλιας Χρήστος, Δομές Δεδομένων και Οργανώσεις Αρχείων, Εκδόσεις Νέων Τεχνολογιών, 1993 (Ανατυπώσεις 1995,1998,2000)
- Σημειώσεις του μαθήματος «Σχεδιασμός και Χρονοπρογραμματισμός Ενεργειών» του χειμερινού εξαμήνου 2005-2006, του Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης, του Τμήματος Πληροφορικής
- Δημητριάδης Αντώνης, Διοίκηση – Διαχείριση Πληροφοριακών Συστημάτων, Εκδόσεις Νέων Τεχνολογιών, 1998 (Ανατύπωση 2000)
- Κιουντούζης Ευάγγελος, Μεθοδολογίες ανάλυσης και Σχεδιασμού Πληροφοριακών Συστημάτων, Εκδοτικός οίκος : Μπένου, Α έκδοση 1997

Ηλεκτρονική βιβλιογραφία

- eos.uom.gr/~achat/softfiles/UML_Review.ppt
- www.it.teithe.gr/~stefanos/OPS-10.pdf
- www.it.civil.ntua.gr/kyklotex/queues%202001.pdf
- <http://searchcrm.techtarget.com/searchCRM/downloads/HowToConducChapter2and9.pdf>
- http://www.softlab.ntua.gr/~bxb/courses/unipi2001_te/00-CourseNotes/031-OO&Java/TE031-1.pdf
- <http://people.msoe.edu/~taylor/cs384/mellotk.pdf>
- www.enman.uowm.gr/dep/ts/lect12-13.ppt
- anamorfosi.teiser.gr/paradotea/d_3_1/telecommunication/pps/lecture_8.ppt
- netmode.ntua.gr/courses/undergraduate/queues/2006/presentations/queueing_5_7_06.ppt
- <http://codex.cs.yale.edu/avi/os-book/os7/slide-dir/ch5.ppt>
- <http://support.inf.uth.gr/courses/CE222/scheduling-280306-pl.pdf#search=%22cpu%20scheduling%22>
- http://mathcs.holycross.edu/~croyden/os/notes/Lec15_scheduling.ppt
- http://cs.clarku.edu/~jbreecher/os/lectures/Section05-Scheduling.ppt#266,20,Slide_20
- <http://cgi.di.uoa.gr/~afrodite/Dynamic-Diagrams.ppt>

- <http://courses.softlab.ntua.gr/softeng/2004b/slides/uml-handout.pdf>
- <http://cqi.di.uoa.gr/~ys04/Use-Cases.ppt>
- http://150.140.9.29/simulation/Project_2001-2002.doc