



ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΠΑΤΡΩΝ

ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ

ΤΜΗΜΑ ΛΟΓΙΣΤΙΚΗΣ



«**ΟΡΓΑΝΩΣΗ ΕΡΓΑΣΙΑΣ ΜΕ ΤΗ ΧΡΗΣΗ
ΕΡΓΑΛΕΙΩΝ ΥΠΟΣΤΗΡΙΞΗΣ
ΕΠΙΧΕΙΡΗΜΑΤΙΚΩΝ ΔΙΑΔΙΚΑΣΙΩΝ**»

ΕΠΙΜΕΛΕΙΑ:

ΚΑΨΗ ΠΑΡΑΣΚΕΥΗ

ΒΟΜΒΥΛΑ ΙΟΥΛΙΑ

ΕΙΣΗΓΗΤΗΣ:

ΖΑΧΑΡΑΚΗΣ ΙΩΑΝΝΗΣ

ΠΑΤΡΑ 2005

ΠΡΟΛΟΓΟΣ

Η ανάγκη για εσωτερική ολοκλήρωση προήλθε όταν οι αρχές της επιστήμης εισήχθησαν στην πρακτική των επιχειρήσεων. Οι αρχές προήλθαν κατά την βιομηχανική επανάσταση και από τους κανόνες που διατυπώθηκαν από τον Adam Smith τον 18 αιώνα. Στις αρχές του 20ου αιώνα το σκεπτικό του Smith αναπτύχθηκε από τον F.W.Taylor. Η συνεχιζόμενη πολυπλοκότητα των παραγωγικών διαδικασιών και το αυξανόμενο μέγεθος των παραγωγικών συστημάτων εμφάνισε δυσκολίες στον συντονισμό μεγάλων οργανισμών. Τα συμπτώματα από αυτές τις δυσκολίες περιλάμβαναν:

- Έμμεσα κόστη
- Μειωμένη ευελιξία
- Μεγάλους χρόνους παράδοσης
- Καθυστερημένη υλοποίηση των καινοτομιών
- Προβλήματα στην διασφάλιση ποιότητας

Σαν συνέπεια να εμφανίζεται και μειωμένη ανταγωνιστικότητα. Η επίλυση τέτοιων προβλημάτων απαιτεί αποτελεσματικό management και όχι μεμονωμένες διαδικασίες ή λειτουργίες αλλά αλυσίδες δραστηριοτήτων και διαδικασιών μέσα σε μία επιχείρηση.

Οι Βασικές ιδέες που σχετίζονται με την θεωρία Enterprise Modeling (EM) εμφανίστηκαν για πρώτη φορά στις αρχές της δεκαετίας του 1980 από την Plandata στην Σουηδία και καθορίστηκαν από SISU (The Swedish Institute for System Development) στα τέλη της δεκαετίας του 80. Η σημαντική διαφοροποίηση σε σχέση με τα μοντέλα του παρελθόντος είναι ότι πλέον λαμβάνονται σοβαρά υπόψη τα μέρη μίας προδιαγραφής όπως για παράδειγμα οι προθέσεις και οι στόχοι μιας επιχείρησης, σε αντίθεση με τα παραδοσιακά

μοντέλα όπου μελετούσαν τις σχέσεις και τις διαδικασίες των επιχειρήσεων. Η συγκεκριμένη αυτή προσέγγιση εφαρμόστηκε σε πολλές διαφορετικές εφαρμογές τα περασμένα χρόνια και εμφάνισε σημαντικά οφέλη τα οποία δεν οφειλόntonτουσαν μόνο στο πλαίσιο εργασίας του EM αλλά και εξαιτίας της κατάλληλης διαχείρισης της διαδικασίας των απαιτήσεων του engineering.

Το EM χρησιμοποιείται για να περιγράψουμε και να αναλύσουμε την παρούσα κατάσταση μίας επιχείρησης ή για να περιγράψουμε και να αναλύσουμε πιθανές μελλοντικές καταστάσεις μίας επιχείρησης. Επιπλέον, αναλύονται οι υπάρχουσες επιχειρηματικές διαδικασίες με σεβασμό στους στόχους του μέλλοντος με σκοπό να διαπιστωθούν οι ανάγκες για βελτίωση αλλαγές στην δομή της επιχείρησης. Το EM είναι μία δομημένη τεχνική που περιγράφει τις διαφορετικές απόψεις μέσα σε μία επιχείρηση. Αποτελεί ένα νέο τρόπο ο οποίος παρέχει ένα πλαίσιο εργασίας στο management των επιχειρήσεων για να διασφαλίσει ότι οι εφαρμογές συστήματος αλλάζουν με την συχνότητα που αλλάζουν οι επιχειρηματικές διαδικασίες και τα επιχειρησιακά μοντέλα .

Τα μοντέλα επιχειρήσεων (EM) περιγράφουν τις επιχειρήσεις και χρησιμοποιούνται για να αναλύσουν και να εξομοιώσουν τις επιχειρήσεις. Αποτελούν τον πυρήνα της μεθοδολογίας που πρέπει να ακολουθήσει η επιχείρηση. Οι μέθοδοι EM είναι ευρέως αναγνωρισμένοι για την αξία τους και την πρακτική τους. Χαρακτηρίζονται από τους εξής τύπους :

1. Business modeling method

Επιχειρηματικά μοντέλα της IBM BSDM (Business System Development)

2. Process modeling method

IDEF0, IDEF3, PSL, RAD, RACD

3. Organizational Modeling method

Ulrich Franks MPM προσέγγιση

4. Capability Modeling method

5. Ontology

ΒΗΜΑΤΑ ΣΥΜΒΟΥΛΕΣ

Ο ρόλος του επιχειρηματικού μοντέλου (EM) είναι να επιτευχθεί μία ανάλυση και να λειτουργήσει ο σχεδιασμός μίας επιχείρησης. Από μία άποψη ένα EM θα πρέπει να παρέχει την «γλώσσα» που χρησιμοποιείται για να οριστεί ρητά μία επιχείρηση. Θα πρέπει επίσης να υπάρχει η δυνατότητα να μελετηθούν διαφορετικά μοντέλα στον σχεδιασμό της δομής και συμπεριφοράς ενός οργανισμού. Για να δικαιολογηθεί ο σχεδιασμός διαφορετικών σχεδίων για έναν οργανισμό πρέπει να δικαιολογηθούν διαφορετικές ομάδες περιορισμών μέσα σε κάθε μοντέλο. Θα πρέπει τεθούν ερωτήσεις όπως:

✓ Μπορεί μία διαδικασία να εκτελεσθεί με διαφορετικό τρόπο;

✓ Μπορεί να εκπληρωθούν οι στόχοι μίας επιχείρησης με διαφορετικό τρόπο;

Θα πρέπει να είμαστε σε θέση να καθορίσουμε την επίδραση των αλλαγών σε όλα τα μέρη της επιχείρησης. Για παράδειγμα αν ελαττώσουμε την ένταση που ασκούμε μία πολιτική, πώς θα επηρεαστεί η ποιότητα των προϊόντων και υπηρεσιών που παρέχονται από την επιχείρηση; Αν προμηθευτούμε νέες μηχανές πώς θα επηρεαστούν οι δραστηριότητες που ήδη εκτελούνται; Θα χρειαστεί να επανεκπαιδεύσουμε το προσωπικό μέσα στην επιχείρηση για να τους δώσουμε τις γνώσεις να χρησιμοποιούν τις νέες μηχανές; Αν αλλάξουμε τις δραστηριότητες τις οποίες εκτελούμε πώς θα επηρεαστεί η κατανάλωση;

Από μία άποψη το επιχειρηματικό μοντέλο θα πρέπει να αντιπροσωπεύει ότι σχεδιάζεται, ότι είναι πιθανόν να συμβεί και ότι έχει γίνει. Θα πρέπει να παρέχει τις πληροφορίες και την τεχνολογία που είναι απαραίτητα να υποστηρίζουν τις λειτουργίες της επιχείρησης ανεξαρτήτως αν εκτελούνται χειρονακτικά ή από μηχανές, θα πρέπει να είναι σε θέση να παρέχουν απαντήσεις στις ερωτήσεις που κοινά τίθενται κατά την εκτέλεση μίας αποστολής.

Είναι ξεκάθαρο ότι από αυτές τις απαιτήσεις χρειαζόμαστε πολλά περισσότερα από ένα απλό μοντέλο δεδομένων. Αυτό που χρειάζεται είναι η ικανότητα να συμπεραίνουμε τι συνεπάγεται από αυτό το μοντέλο.

Ευχαριστίες

Η πτυχιακή που έχετε στα χέρια σας αποτελεί μια πρώτη προσπάθεια για να κατανοήσουμε καλύτερα τη σημαντικότητα της οργάνωσης μιας διαδικασίας με τη χρήση εργαλείων υποστήριξης των επιχειρηματικών διαδικασιών. Συγκεκριμένα, με τις διάφορες τεχνικές μοντελοποίησης και φυσικά τη γλώσσα σχεδιασμού τους, UML.

Το θεωρητικό κομμάτι της εργασίας έχει ως γενικό σκοπό την παρουσίαση και ανάλυση των όρων επιχειρηματική διαδικασία, μοντελοποίηση, τεχνικές μοντελοποίησης και ενοποιημένη γλώσσα σχεδιασμού UML και ως ειδικό σκοπό την όσο το δυνατόν καλύτερη κατανόησή τους από τους αναγνώστες.

Στο εμπειρικό μέρος παρουσιάζονται πέντε εφαρμογές των παραπάνω διαγραμματικών τεχνικών έτσι ώστε να γίνει πλήρως κατανοητός ο ρόλος τους και ο τρόπος εφαρμογής τους.

Θέλουμε να ευχαριστήσουμε θερμά τον επιβλέποντα καθηγητή κ. Ζαχαράκη, που συνέβαλε με τις παρατηρήσεις του και τις γνώσεις του, στην ολοκλήρωση αυτής της πτυχιακής εργασίας αλλά και τους υπόλοιπους εκπαιδευτικούς που ανέλαβαν το δύσκολο έργο της διδασκαλίας μας.

ΠΕΡΙΕΧΟΜΕΝΑ

1	ΕΠΙΧΕΙΡΗΜΑΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ	9
1.1	ΕΙΣΑΓΩΓΗ.....	9
1.2	ΕΝΝΟΙΑ ΕΠΙΧΕΙΡΗΜΑΤΙΚΗΣ ΔΙΑΔΙΚΑΣΙΑΣ.....	10
	1.2.1 Χαρακτηριστικά επιχειρηματικής διαδικασίας.....	12
	1.2.2 Επανασχεδιασμός των επιχειρηματικών διαδικασιών.....	14
	1.2.3 Ροή εργασίας.....	15
	1.2.3.1. Συστήματα διαχείρισης της ροής εργασίας.....	16
	1.2.3.2. Συμμετέχων σε ροή εργασίας.....	17
1.3	ΜΕΘΟΔΟΙ ΟΡΙΣΜΟΥ ΕΠΙΧΕΙΡΗΜΑΤΙΚΗΣ ΔΙΑΔΙΚΑΣΙΑΣ.....	18
	Α. Χαρτογράφηση της διαδικασίας.....	20
	Β. Χάρτης διαδικασίας για δραστηριότητες με προετοιμασμένη την πρόταση της διαδικασίας	20
1.4	ΣΤΟΧΟΙ ΚΑΙ ΔΟΜΗ ΕΡΓΑΣΙΑΣ.....	21
2	ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΚΑΙ ΤΕΧΝΙΚΕΣ ΜΟΝΤΕΛΟΠΟΙΗΣΗΣ	23
2.1	ΕΙΣΑΓΩΓΗ.....	23
2.2	ΕΝΝΟΙΑ ΤΟΥ ΜΟΝΤΕΛΟΥ.....	26
	2.2.1 <i>Business Model</i>	26
	2.2.2 ΕΝΝΟΙΑ ΜΟΝΤΕΛΟΠΟΙΗΣΗΣ.....	28
	2.2.3 Μοντελοποίηση επιχειρηματικών διαδικασιών.....	29
	2.2.3.1 Οι ρόλοι κατά τη διαδικασία της μοντελοποίησης μιας επιχειρηματικής διαδικασίας.....	30
	2.2.3.2 Παραδείγματα.....	31
	Α. Παράδειγμα μοντελοποίησης μιας επιχειρηματικής διαδικασίας – Περιγραφή.....	31
	Β. Παράδειγμα μοντελοποίησης μιας επιχειρηματικής διαδικασίας – Εντοπίζοντας τις οντότητες.....	32

	Γ. Μοντελοποιώντας τις σχέσεις μεταξύ των οντοτήτων μιας επιχειρηματικής διαδικασίας.....	31
2.3	ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΑΝΑΛΥΣΗ ΤΩΝ ΔΙΑΓΡΑΜΜΑΤΙΚΩΝ ΤΕΧΝΙΚΩΝ.....	33
	2.3.1 Διαγράμματα ροής.....	33
	2.3.1.1 Σύμβολα ροής.....	34
	2.3.1.2 Είδη διαγραμμάτων ροής.....	38
	2.3.1.3 Παράδειγμα διαγράμματος ροής.....	39
	2.3.2 Διαγράμματα ροής δεδομένων.....	40
	2.3.2.1 Βασικές αρχές κατασκευής ενός ΔΡΔ.....	43
	2.3.2.2 Παράδειγμα ΔΡΔ.....	45
	2.3.3 Διαγράμματα Δομής.....	46
	2.3.4 Διαγράμματα ΗΙΡΟ.....	49
	2.3.4.1 Δομικά στοιχεία διαγραμμάτων ΗΙΡΟ.....	49
	2.3.4.2 Πλεονεκτήματα διαγραμμάτων ΗΙΡΟ.....	50
	2.3.4.3 Παράδειγμα διαγράμματος ΗΙΡΟ.....	51
	2.3.5 Διαγράμματα ΗΟΣ.....	52
	2.3.6 Διαγράμματα Warnier-orr.....	52
	2.3.7 Διαγράμματα ενεργειών.....	54
	2.3.8 Διαγράμματα Michael Jackson.....	55
	2.3.9 Δομημένη περιγραφή και ψευδοκώδικας.....	56
2.4	ΕΠΙΛΟΓΟΣ – ΣΥΜΠΕΡΑΣΜΑΤΑ.....	58
3	ΕΝΙΑΙΟΣ ΤΡΟΠΟΣ ΜΟΝΤΕΛΟΠΟΙΗΣΗΣ (UML)	
3.1	ΕΙΣΑΓΩΓΗ.....	59
3.2	ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΗΣ UML.....	61
	3.2.2 Βασικά στοιχεία της uml.....	62
	α) τι είναι η οντότητα.....	62
	β) τι είναι οι χαρακτηριστικές ιδιότητες.....	64
	γ) τι είναι οι σχέσεις.....	65
	3.2.3 Διαγράμματα της uml	68

3.2.3.1	Χρήση Διαγραμμάτων UML στις Δραστηριότητες Ανάπτυξης Λογισμικού.....	69
3.2.3.2	Διαγράμματα Περιπτώσεων Χρήσεων (use case diagrams)...	72
3.2.3.3	Διαγράμματα κλάσεων.....	75
3.2.3.4	Διαγράμματα αντικειμένων.....	75
3.2.3.5	Διαγράμματα ακολουθίας.....	77
3.2.3.6	Διαγράμματα συνεργασίας.....	78
3.2.3.7	Διαγράμματα καταστάσεων.....	79
3.2.3.8	Διαγράμματα δραστηριοτήτων.....	81
3.3	ΕΠΙΛΟΓΟΣ – ΣΥΜΠΕΡΑΣΜΑΤΑ.....	82
4	ΠΕΡΙΠΤΩΣΕΙΣ ΧΡΗΣΗΣ (use case scenario).....	83
4.1	ΕΙΣΑΓΩΓΗ.....	83
4.2	ΤΙ ΕΙΝΑΙ ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΜΕ ΠΕΡΙΠΤΩΣΕΙΣ ΧΡΗΣΗΣ	83
4.2.1	1 ^ο παράδειγμα διαγραμματικής τεχνικής.....	86
4.2.2	2 ^ο παράδειγμα με κείμενο.....	89
4.2.3	3 ^ο παράδειγμα.....	91
4.2.4	4 ^ο παράδειγμα.....	98
4.2.5	5 ^ο παράδειγμα.....	99
4.3	ΣΥΜΠΕΡΑΣΜΑΤΑ.....	107
	ΕΠΙΛΟΓΟΣ	108

ΚΕΦΑΛΑΙΟ 1ο

ΕΠΙΧΕΙΡΗΜΑΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ

1.1 ΕΙΣΑΓΩΓΗ

Πριν από μερικά χρόνια, οι επιχειρήσεις οργάνωναν τις επιχειρησιακές τους διαδικασίες με τρόπο που καθηστερούσε την παραγωγή αλλά και αύξανε το κόστος αυτής. Η αυτοματοποίηση της διαχείρισης ορισμένων διαδικασιών και η ανάπτυξη των ηλεκτρονικών υπολογιστών σε μεγάλο βαθμό, επέφερε σιγά σιγά την αλλαγή του τρόπου με τον οποίο διεξάγονται σήμερα οι διαδικασίες των επιχειρήσεων από την πιο μικρή, ως και την μεγαλύτερη πολυεθνική εταιρία σε όλο τον κόσμο. Έτσι, αντί οι επιχειρήσεις να διαχειρίζονται τα διάφορα έγγραφα τους και τις διαδικασίες τους χρησιμοποιώντας μια δομή εργασίας που ήταν αργή ως προς τον χρόνο διεκπαιρέωσης, τα τελευταία χρόνια άρχισαν να σκέφτονται πιο επιχειρηματικά.

Αυτό είχε ως αποτέλεσμα τη διάσπαση των παλαιών πολύπλοκων εργασιακών δομών σε ευέλικτες και όσο το δυνατόν απλές λειτουργίες-κύτταρα και την οργάνωση των εργαζομένων σε ευέλικτες ομάδες εργασίας που καλύπτουν διάφορες ειδικότητες και μπορούν να ανταποκριθούν γρηγορότερα στις ανάγκες των πελατών και στις απαιτήσεις του ανταγωνισμού. Για το σκοπό αυτό οι ομάδες χρειάζονταν την υποστήριξη νέων συστημάτων λογισμικού για τις ανάγκες της συνεργασίας τους.

Η επιχειρηματική διαδικασία αρχίζει να χρησιμοποιείται όλο και περισσότερο ως το βασικό κύτταρο οργάνωσης, σχεδιασμού και υλοποίησης των επιχειρηματικών στόχων. Ο οργανισμός καθορίζει τον τρόπο με τον οποίο θα εκπληρωθούν οι επιχειρηματικές διαδικασίες ανζητώντας τα συστήματα λογισμικού που θα υποστηρίξουν, θα συντονίσουν και θα ελέγξουν αυτές τις

διαδικασίες. Οι επιχειρηματικές διαδικασίες είναι σημαντικοί πόροι πληροφορίας, των οποίων η ποιότητα και η απόδοση αντανακλούν τη βιωσιμότητα της επιχείρησης.

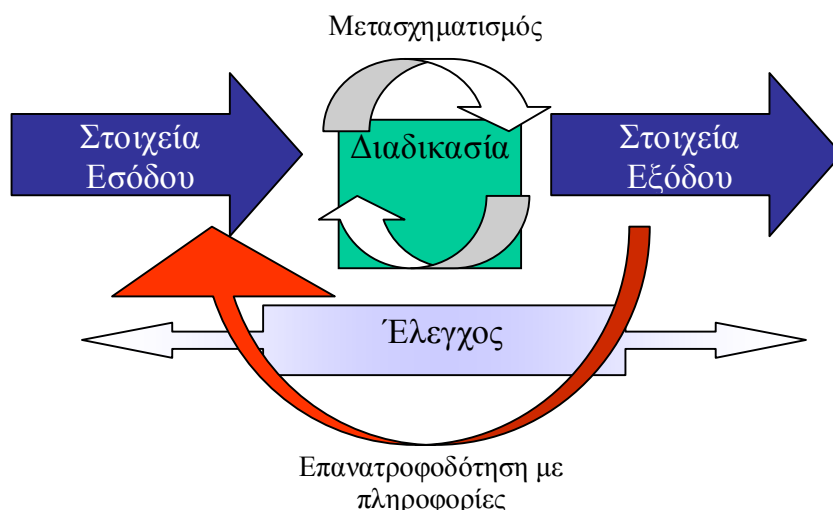
Τα Συστήματα Διαχείρισης Ροής Εργασιών (Workflow), βοηθούν στην απλούστευση της διαχείρισης, στη διακίνηση και στην αποθήκευση των διαφόρων εγγράφων ζωτικής και μη σημασίας των επιχειρήσεων. Η σκέψη να μοντελοποιήσουν την οργανωτική δομή της επιχείρησης, τους φαινόταν πολύ αποτελεσματική. Έτσι, οργανώνοντας την επιχειρηματική διαδικασία, από τον προϊστάμενο μέχρι και τον τελευταίο υφιστάμενο, σε όλο το μήκος της πυραμίδας της επιχείρησης, θα πετύχαιναν μείωση του χρόνου διεξαγωγής των διαδικασιών, αύξηση της παραγωγικής διαδικασίας και φυσικά μείωση του κόστους παραγωγής.

Το κεφάλαιο αυτό αναλύει το σύστημα διαχείρισης της ροής εργασίας καθώς και την έννοια της μοντελοποίησης και της επιχειρηματικής διαδικασίας ως ένα τρόπο οργάνωσης της κάθε επιχείρησης.

1.2. ΕΝΝΟΙΑ ΤΗΣ ΕΠΙΧΕΙΡΗΜΑΤΙΚΗΣ ΔΙΑΔΙΚΑΣΙΑΣ

Ο ορισμός της διαδικασίας αποτελείται από ένα δίκτυο δραστηριοτήτων και τους μεταξύ τους συσχετισμούς, από κριτήρια που σηματοδοτούν την έναρξη και τερματισμό της διαδικασίας και πληροφορίες για τις ανεξάρτητες δραστηριότητες, όπως συμμετέχοντες, σχετικές εφαρμογές και δεδομένα IT, κλπ. Γενικά μπορούμε να θεωρήσουμε πως μια διαδικασία αποτελείται από συγκεκριμένο τύπο ενεργειών που στοχεύουν σε ένα συγκεκριμένο αποτέλεσμα: την αύξηση της απόδοσης της εργασίας. Ο ορισμός της διαδικασίας προκύπτει από την εργασία κατά τη διάρκεια της κατάστασης ορισμού διαδικασίας και μπορεί να περιλαμβάνει τόσο χειρονακτικές δραστηριότητες, όσο και δραστηριότητες ροής εργασίας (αυτοματοποιημένες), αναφορές σε υποδιαδικασίες και τέλος μπορεί να

αναφέρεται σε έναν ξεχωριστό Οργανισμό ή Μοντέλο Πληροφόρησης, για να δώσει τη δυνατότητα στους συμμετέχοντες να προσδιοριστούν εμμέσως, κάνοντας αναφορά σε χαρακτηριστικές ιδιότητες, όπως ο ρόλος ή η οργανική θέση μέσα στην επιχείρηση.



Σε έναν οργανισμό διακρίνουμε τις υλικές διαδικασίες, τις διαδικασίες επεξεργασίας της πληροφορίας και τις επιχειρηματικές διαδικασίες.

Οι υλικές (material) διαδικασίες έχουν σαν σκοπό τη συναρμολόγηση υλικών στοιχείων και τη διανομή φυσικών προϊόντων. Αυτές οι διαδικασίες περιλαμβάνουν τη μεταφορά, αποθήκευση, μέτρηση και συναρμολόγηση φυσικών αντικειμένων.

Οι διαδικασίες επεξεργασίας της πληροφορίας έχουν σχέση με τις αυτοματοποιημένες ή μερικά αυτοματοποιημένες εργασίες που δημιουργούν, διαχειρίζονται και προσφέρουν πληροφορία (πχ. Οι εργασίες που εκτελούνται από προγράμματα ή ανθρώπους που αλληλεπιδρούν με υπολογιστές). Οι βασικές

που τις υποστηρίζουν είναι οι βάσεις δεδομένων και τα καταναμημένα συστήματα.

Οι επιχειρηματικές διαδικασίες (business processes) είναι μια ομάδα συσχετιζόμενων διαδικασιών ή δραστηριοτήτων, οι οποίες υλοποιούν (επιτυγχάνουν) συλλογικά έναν επιχειρηματικό αντικειμενικό στόχο ή σκοπό τακτικής, συνήθως μέσα σε μια οργανωτική δομή, προσδιορίζοντας λειτουργικούς ρόλους και σχέσεις.

1.2.1 Τα χαρακτηριστικά μιας επιχειρηματικής διαδικασίας

Μια επιχειρηματική διαδικασία είναι τυπικά συνδεδεμένη με λειτουργικούς στόχους και επιχειρηματικές σχέσεις, για παράδειγμα μια Διαδικασία Διεκδίκησης Ασφάλειας (Insurance Claims Process). Μια διαδικασία μπορεί να εμπεριέχεται ολόκληρη σε μια οργανωτική μονάδα, ή να συνδέει αρκετούς διαφορετικούς οργανισμούς, όπως σε μια σχέση πελάτη-προμηθευτή. Έχει καθορισμένες συνθήκες (όρους) που πυροδοτούν την έναρξη της, σε κάθε νέα περίπτωση [π.χ. η παραλαβή μιας αίτησης (αξίωσης)] και καθορισμένα αποτελέσματα, με τη συμπλήρωση της. Επίσης μπορεί να περιλαμβάνει επίσημες ή σχετικά ανεπίσημες μορφές αλληλεπίδρασης, μεταξύ των συμμετεχόντων και η χρονική διάρκεια της μπορεί επίσης να ποικίλει αρκετά. Τέλος, μια επιχειρηματική διαδικασία είναι πιθανό να αποτελείται από αυτοματοποιημένες δραστηριότητες, ικανές για διαχείριση ροής εργασίας και/ ή χειρονακτικές δραστηριότητες, οι οποίες δεν βρίσκονται στις προθέσεις/ επιδιώξεις της διαχείρισης της ροής εργασίας.

Κάποιες γενικές κατηγορίες επιχειρηματικών διαδικασιών είναι:

- ∅ Η επιμελητεία των αγορών (απογραφή, έλεγχος, επιστροφές στους προμηθευτές κλπ.)
- ∅ Η επιμελητεία των πωλήσεων (επεξεργασία δελτίων πωλήσεων, εφοδιασμός κλπ.)
- ∅ Το μάρκετινγκ (διαφήμιση, προώθηση κλπ)
- ∅ Οι υπηρεσίες (εγκατάσταση, συντήρηση κλπ.)
- ∅ Οι προμήθειες (αγορά, παραγγελία κλπ.)
- ∅ Η ανάπτυξη της τεχνολογίας (έρευνα και ανάπτυξη)
- ∅ Η διαχείριση του οργανισμού και του ανθρώπινου δυναμικού (εκμίσθωση, εκπαίδευση)
- ∅ Η υποδομή της επιχείρησης (λογιστική, επιχειρηματικός σχεδιασμός, διαχείριση κεφαλαίων)

Οι πρωτεύουσες επιχειρηματικές διαδικασίες περιλαμβάνουν δραστηριότητες που προσθέτουν άμεση αξία στα προϊόντα της επιχείρησης (πχ. Οι πρώτες τέσσερις από τις παραπάνω). Οι υποστηρικτικές επιχειρηματικές διαδικασίες περιλαμβάνουν δραστηριότητες που προσθέτουν έμμεσα αξία και υποστηρίζουν τις πρωτεύουσες (πχ. Οι τέσσερις τελευταίες). Συνολικά οι πρωτεύουσες και οι υποστηρικτικές επιχειρηματικές διαδικασίες συνθέτουν τη γραμμή/αλυσίδα των δραστηριοτήτων που προσδιορίζουν την αξία της επιχείρησης.

Η ανάλυση της λειτουργίας της επιχείρησης μέσω της έννοιας της επιχειρηματικής διαδικασίας είναι ένας πολύ καλός τρόπος για την καταμέτρηση της ανταγωνιστικότητας της επιχείρησης. Η προσεκτική ανάλυση μπορεί να αναδείξει σημεία τα οποία μπορούν να βελτιωθούν από άποψη οργάνωσης ή και να υποστηριχθούν από τα αντίστοιχα πληροφοριακά συστήματα. Η προσέγγιση αυτή είναι εξαιρετικά χρήσιμη όχι μόνο για τον λογιστή, αλλά και για τους περισσότερους υπαλλήλους γιατί τους επιτρέπει να εστιάζουν την προσοχή τους και να μελετούν συγκεκριμένες κάθε φορά περιοχές από το μεγάλο σύνολο των

δραστηριοτήτων μιας επιχείρησης.

Αν μια επιχειρηματική διαδικασία χρειάζεται βελτίωση τότε μπορεί να γίνει ο επανασχεδιασμός της, δηλαδή θα πρέπει να επαναπροσδιοριστεί το τι κάνουν και το πώς το κάνουν οι εργαζόμενοι που συμμετέχουν σε αυτή. Συχνά είναι απαραίτητο να επανασχεδιάσουμε μια χειρονακτική διαδικασία που θέλουμε να την υποστηρίξουμε από κάποιο πληροφοριακό σύστημα. Η μεθοδολογία του ελέγχου ροής εργασίας (workflow) επιτρέπει την αναπαράσταση, την κατανόηση, την αξιολόγηση και τον επανασχεδιασμό μιας επιχειρηματικής διαδικασίας.

1.2.2 Ο επανασχεδιασμός των επιχειρηματικών διαδικασιών

Όταν ένας οργανισμός έχει προδιαγραφές και έχει υλοποιήσει τις δραστηριότητες του με τη μορφή επιχειρηματικών διαδικασιών, μπορεί να τις επανασχεδιάσει ή να τις προσαρμόσει στις νέες απαιτήσεις. Σκοπός αυτού είναι η αύξηση της ικανοποίησης του πελάτη, η βελτίωση της αποδοτικότητας, η αύξηση της ποιότητας των προϊόντων, η μείωση του κόστους, η διεύρυνση του πελατολογίου κτλ.

Ο επανασχεδιασμός των επιχειρηματικών διαδικασιών (BPR-Business Process Reengineering) περιλαμβάνει την αναθεώρηση και τον επανασχεδιασμό της επιχ. διαδικασίας και πραγματοποιείται πριν χρησιμοποιηθούν τα πληροφοριακά συστήματα για την αυτοματοποίηση της διαδικασίας. Ουσιαστικά υλοποιεί τις επανασχεδιασμένες πληροφοριακές διαδικασίες προσφέροντας τα πλεονεκτήματα της τεχνολογίας και την αποδοτικότητα των Π.Σ.

1.2.3 Ροή εργασίας

Ονομάζεται η αυτοματοποίηση μιας επιχειρησιακής διαδικασίας, συνολικά ή τμηματικά, κατά την οποία έγγραφα, πληροφορίες ή καθήκοντα μεταβιβάζονται από έναν συνεργάτη/ συμμετέχοντα σε κάποιον άλλο προς διεκπεραίωση, σύμφωνα με συγκεκριμένους διαδικαστικούς κανόνες. Η αυτοματοποίηση καθορίζεται στα πλαίσια ενός Καθορισμού Διεργασίας, η οποία προσδιορίζει τις διάφορες δραστηριότητες διαδικασίας, διαδικαστικούς κανόνες και σχετικά δεδομένα ελέγχου, που χρησιμοποιούνται στη διαχείριση της ροής εργασίας κατά τη διάρκεια της εφαρμογής της. Πολλές ανεξάρτητες περιπτώσεις διεργασιών μπορούν να είναι λειτουργικές κατά τη διάρκεια εφαρμογής της διαδικασίας, καθεμιά από αυτές συσχετισμένη με ένα συγκεκριμένο σύνολο δεδομένων, που να είναι σχετικά με τη συγκεκριμένη περίπτωση διαδικασίας ή «Υπόθεση» ροής εργασίας. Μερικές φορές γίνεται μια ελαφρά διάκριση, μεταξύ της ροής εργασίας παραγωγής, στην οποία οι περισσότεροι διαδικαστικοί κανόνες αποσαφηνίζονται από πριν και την εξ' ορισμού ροή εργασίας, στην οποία οι διαδικαστικοί κανόνες μπορούν να τροποποιηθούν ή να δημιουργηθούν κατά τη λειτουργία της διαδικασίας.

Η έννοια του ελέγχου εργασίας είναι άμεσα συνηφασμένη με τον ορισμό της επιχειρηματικής διαδικασίας. Ο έλεγχος της ροής εργασίας (workflow), που στο εξής θα αναφέρεται ως ΕΡΕ, περιγράφει τις δραστηριότητες μιας επιχειρηματικής διαδικασίας σε ένα επίπεδο ικανό για την κατανόηση, αξιολόγηση και τον επανασχεδιασμό της επιχειρηματικής διαδικασίας. Επιπλέον ο ΕΡΕ περιλαμβάνει τις δραστηριότητες και της πληροφορικής διαδικασίας σε επίπεδο που περιγράφει ικανοποιητικά τις απαιτούμενες δεξιότητες του προσωπικού. Από αυτά καταλήγουμε στη θεώρηση πως ο ΕΡΕ αποτελεί ένα απαραίτητο εργαλείο στο σχεδιασμό νέων ή και επανασχεδιασμένων πληροφοριακών συστημάτων.

Εκτός από τα παραπάνω η έννοια της ροής εργασίας μπορεί να αναφέρεται και στον προσδιορισμό της επιχειρηματικής διαδικασίας (τις προδιαγραφές της), δηλαδή είτε στο λογισμικό που υλοποιεί και αυτοματοποιεί τη διαδικασία, είτε στο

λογισμικό που απλά υποστηρίζει το συντονισμό και τη συνεργασία των ανθρώπων που πραγματοποιούν τη διαδικασία.

Συνοψίζοντας μπορούμε να πούμε πως **ο ΕΡΕ είναι μια συλλογή δραστηριοτήτων (tasks) που είναι οργανωμένες για ολοκλήρωση μιας επιχειρηματικής διαδικασίας και προσδιορίζει τη σειρά επίκλησης των δραστηριοτήτων, τη ροή δεδομένων (data flow) και τους συμμετέχοντες στη διαδικασία αυτή.**

Κάθε δραστηριότητα ολοκληρώνεται από ένα ή περισσότερα λογισμικά συστήματα, έναν ή περισσότερους εργαζόμενους ή ένα συνδυασμό των παραπάνω. Η ανθρώπινη συμμετοχή συνίσταται στη στενή συνεργασία με τους υπολογιστές (όλες οι διαδικασίες γίνονται μέσω υπολογιστή) ή στη λιγότερο στενή (ο υπολογιστής υποδεικνύει μόνο την πρόοδο της διαδικασίας).

1.2.3.1 Συστήματα διαχείρισης της ροής εργασίας

Τα συστήματα που προσδιορίζουν, δημιουργούν και διαχειρίζονται την εκτέλεση των ροών εργασίας με τη χρήση λογισμικού, που να λειτουργεί σε μια ή περισσότερες μηχανές ροής εργασίας, τα οποία είναι ικανά να ερμηνεύουν τον ορισμό της διαδικασίας, να αλληλεπιδρούν με τους συμμετέχοντες στην ροή εργασίας και όπου απαιτείται και να ζητούν την υποστήριξη της χρήσης των εργαλείων και εφαρμογών ΙΤ. Ένα Σύστημα Διαχείρισης Ροής Εργασίας αποτελείται από συνθετικά στοιχεία λογισμικού που χρησιμοποιούνται για να αποθηκεύουν και να ερμηνεύουν ορισμούς διαδικασιών, να δημιουργούν και να διαχειρίζονται περιπτώσεις ροής εργασίας, καθώς αυτές εκτελούνται και να ελέγχουν την αλληλεπίδραση τους με εφαρμογές και συμμετέχοντες στη ροή εργασίας. Τέτοια συστήματα επίσης τυπικά παρέχουν διοικητικές και εποπτικές λειτουργίες (υπηρεσίες), για παράδειγμα επιτρέπουν εκ νέου ανάθεση εργασίας ή κλιμάκωση, μαζί με πληροφορίες περιοδικού ελέγχου και διαχείρισης, για το

σύστημα συνολικά, ή πληροφορίες που να έχουν σχέση με ανεξάρτητες διαδικαστικές περιπτώσεις.

1.2.3.2 Συμμετέχων σε ροή εργασίας

Ένας πόρος ο οποίος εκτελεί την εργασία που αναπαριστάται από μια χαρακτηριστική περίπτωση δραστηριότητας ροής εργασίας. Η εργασία αυτή εμφανίζεται ως ένα ή περισσότερα κομμάτια εργασίας, που έχουν ανατεθεί στον συμμετέχοντα σε ροή εργασίας, μέσω της λίστας εργασιών.

Ο όρος Συμμετέχων σε Ροή Εργασίας έχει συνήθως εφαρμογή σε ανθρώπινους πόρους, αλλά θα μπορούσε θεωρητικά [conceptually] να συμπεριλάβει μηχανικούς πόρους, όπως έναν intelligent agent. Ένας συμμετέχων σε ροή εργασίας μπορεί να προσδιοριστεί ευθέως μέσα στον ορισμό της επιχειρηματικής διαδικασίας, ή προσδιορίζεται παραπέμποντας στον ορισμό της διαδικασίας, ως ένας ρόλος ή οργανωτική οντότητα, που μπορεί έπειτα να συμπληρωθεί από έναν ή περισσότερους από τους διαθέσιμους πόρους του συστήματος ροής εργασίας, για να λειτουργήσει σε αυτό το ρόλο στην εκτέλεση της διαδικασίας.

Μέσα στο Μετά-Μοντέλο Ορισμού Διεργασίας ΕΔΡΕ, η δήλωση συμμετέχοντος σε ροή εργασίας μπορεί να προσδιορίσει έναν συμμετέχοντα ως έναν από τους τέσσερις τύπους - άνθρωπο, πόρο, ρόλο, ή οργανωτική μονάδα. Σαν εναλλακτική λύση, αναφορά μπορεί να γίνει σε ένα εξωτερικό Οργανωτικό Μοντέλο, το οποίο παρέχει πληροφορίες για συμμετέχοντες που πληρούν τα επιλεγέντα κριτήρια.

1.3 Μέθοδοι ορισμού της επιχειρηματικής διαδικασίας

Παραδοσιακές προσεγγίσεις για τον ορισμό διαδικασίας στον πίνακα δανείζονται από καθιερωμένες αναλύσεις συστημάτων και μέθοδοι σχεδίασης. Αυτή η προσέγγιση χρησιμοποιεί διαγράμματα γνωστά στους αναλυτές συστημάτων όπως τα διαγράμματα ροής δεδομένων (DFD) και τα διαγράμματα μεταβολής. Βασίζεται σε ένα συνδυασμό μεθόδων I διαφορετικής διαγράμμισης που θα τις μελετήσουμε με τη σειρά. Αυτές ορίζουν:

- Τις διαδικασίες και τα συστατικά των υπό-εργασιών τους.
- Τις εξαρτήσεις ανάμεσα στις διαδικασίες.
- Τις εισόδους (πόρους) που χρειάζονται από τις διαδικασίες και τις εξόδους.

Αυτές οι παραδοσιακές μέθοδοι μερικές φορές είναι γνωστές ως δραστηριότητα βασισμένη σε μεθόδους από τότε που επικεντρώθηκαν στον ορισμό της εργασίας που χρειάζεται να εκτελεστεί κατά τη διάρκεια των διεργασιών. Η δραστηριότητα βασίζεται σε μεθόδους που κυριαρχούν σε εμπορικά προϊόντα. Ένας μικρός αριθμός εμπόρων έχει αποδεχτεί μια εναλλακτική προσέγγιση, η οποία βασίζεται σε αποσπάσματα που εγκαθίστανται στο διαχείρισης ροής εργασίας.

Ως μια εισαγωγή για την εξέταση των διαφορετικών μεθόδων της ανάλυσης της διαδικασίας ο παρακάτω πίνακας συνοψίζει τις διαφορές ανάμεσα στις βασικές μεθόδους. Σημειώνουμε ότι για κάθε μια «μέθοδο» υπάρχει μια σειρά από εναλλακτικές λύσεις που συνιστούν έμποροι και αναλυτές διαφορετικών προϊόντων.

Πίνακας: Μια περίληψη των βασικών μεθόδων ανάλυσης της διαδικασίας

Μέθοδος ανάλυσης	Διακεκριμένο χαρακτηριστικό	Πιο καλό για	Χρησιμοποιείται από
Στην δραστηριότητα ή στις παραδοσιακές μεθόδους.	Ορίζει τις διαδικασίες, τις εξαρτήσεις και τους πόρους χρησιμοποιώντας μια σειρά από μεθόδους όπως DFD.	Όλους τους βαθμούς περιπλοκής του διαχείρισης ροής εργασίας, συγκεκριμένα για την παραγωγή του διαχείρισης ροής εργασίας	Την πλειοψηφία των εμπόρων του διαχείρισης ροής εργασίας
Βασίζεται στην επικοινωνία	Βασίζεται στα αποσπάσματα που συντάχθηκαν ανάμεσα στους μετόχους του διαχείρισης ροής εργασίας.	Την ειλικρινή διαχείριση των συστημάτων διαχείρισης ροής εργασίας	Την ενέργεια του διαχείρισης ροής εργασίας
Προσανατολισμένη στο αντικείμενο	Σε κάθε αντικείμενο του διαχείρισης ροής εργασίας ενσωματώνονται και τα δύο, δεδομένα και μέθοδοι για την λειτουργία του.	Όλους τους βαθμούς της πολυσύνθεσης	Περιορισμένο αριθμό εμπόρων

α. Η χαρτογράφηση της διαδικασίας

Στο αρχικό στάδιο πριν οι δραστηριότητες προσδιοριστούν λεπτομερειακός χρειάζεται να προσδιορίσουμε που προκύπτουν οι διαδικασίες του οργανισμού και ο οποίος είναι υπεύθυνος γι' αυτές Αυτή η διαδικασία συχνά είναι γνωστή ως χαρτογράφηση της διαδικασίας Ο παρακάτω πίνακας απεικονίζει τις δραστηριότητες που προκύπτουν από τις δια μέσου λειτουργίες σε μια μικρομεσαία επιχείρηση. Μια τέτοια χαρτογράφηση διαδικασίας είναι σαφώς σημαντική για τον προσδιορισμό των ενδεχομένων χρηστών ενός νέου συνεργασιακού συστήματος και την σειρά που θα εκτελέσουν τις δραστηριότητες τους.

β.Χάρτης διαδικασίας για δραστηριότητες με «προετοιμασμένη την πρόταση» της διαδικασίας

Δραστηριότητες διαδικασίας	Τοποθέτηση ενός προϊόντος στην αγορά	Λειτουργία	Οικονομικά	Ανώτερη διαχείριση
Η εκτίμηση του κόστους		M		
Ο προσδιορισμός του οικονομικού		m	M	
Η παρουσίαση της διαφήμισης	M	m		
απολογισμός	M	m	M	M
Η εξουσιοδότηση			m	M

M=O Ο κύριος ρόλος στη λειτουργία m=ο δευτερεύων ρόλος στη λειτουργία
Η διεργασία αποσύνθεσης του συστήματος διαχείρισης ροής εργασίας για την κωδικοποίηση της εκτέλεσης του έργου κατά στάδια.

1.4 Στόχοι και δομή εργασίας

Σύμφωνα με τα παραπάνω καταλήγουμε στο γεγονός ότι η ανάπτυξη ενός επιχειρηματικού συστήματος περιέχει τρεις φάσεις: την ανάλυση του συστήματος, τον σχεδιασμό του και την υλοποίηση του.

Η ανάλυση του συστήματος αναφέρεται κυρίως στη τυποποίηση και αξιολόγηση των λύσεων στα προβλήματα του χρήστη.Οι βασικοί σκοποί της ανάλυσης είναι:

Να βελτιώσει την ποιότητα της πληροφορίας προς δύο κατευθύνσεις, την πληροφορία που έχει ήδη ληφθεί από τους χρήστες για το πως θέλουν να οργανωθεί το υπό ανάπτυξη σύστημα και την πληροφορία που θα δίνει το ίδιο το σύστημα όταν αναπτυχθεί (τις υπηρεσίες και τις λειτουργίες που θα προσφέρει.Επίσης, να βελτιώσει τον εσωτερικό έλεγχο και να ελαχιστοποιήσει τα κόστη, όσο αυτό είναι δυνατόν.

Ο σχεδιασμός του συστήματος έχει ως σκοπό τον προσδιορισμό των λεπτομερειών των λύσεων που έχει προτείνει η ανάλυση.

Η υλοποίηση του συστήματος είναι η τελική διαδικασία που θα μετατρέψει το σχεδιασμένο σύστημα σε ένα σύστημα έτοιμο για λειτουργία.

Ένα νέο πληροφοριακό σύστημα κατά κανόνα αλλάζει τις εργασιακές σχέσεις του προσωπικού αφού αλλάζει το περιεχόμενο της εργασίας και ίσως και την τυπική οργανωτική δομή της επιχείρησης. Η συνεργασία των χρηστών είναι επίσης πολύ σημαντική γιατί οι χρήστες είναι αυτοί που αρχικά θα προτείνουν λύσεις για την ανάπτυξη του συστήματος και αυτοί που τελικά θα το χρησιμοποιήσουν μετά την ολοκλήρωσή του. Η ανάμιξη των χρηστών και στη

προκειμένη περίπτωση του λογιστή αποτελεί σημαντική στρατηγική που εξασφαλίζει τη μακροβιότητα, τη βελτιστοποίηση και το χαμηλότερο κόστος λειτουργίας του συστήματος.

Δίνοντας λοιπόν ιδιαίτερη έμφαση στα παραπάνω, την υλοποίηση, τον σχεδιασμό και την ανάλυση του συστήματος θα αποδείξουμε τελικά πόσο σημαντικό ρόλο παίζει η ανάλυση δεδομένων ενός μοντέλου δεδομένων κατασκευασμένο με συγκεκριμένες τεχνικές.

2^ο ΚΕΦΑΛΑΙΟ

ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΚΑΙ ΤΕΧΝΙΚΕΣ ΜΟΝΤΕΛΟΠΟΙΗΣΗΣ

2.1 Εισαγωγικά

Για την κατασκευή του μοντέλου ενός ολοκληρωμένου πληροφοριακού συστήματος απαιτείται να πραγματοποιηθούν οι εξής διεργασίες:

1. *Ανάλυση και μοντελοποίηση των επεξεργασιών* που γίνονται στο σύστημα (κατασκευή του μοντέλου επεξεργασιών του συστήματος).
2. *Ανάλυση και μοντελοποίηση των δεδομένων* που διαχειρίζεται το σύστημα (Κατασκευή του μοντέλου δεδομένων του συστήματος).
3. *Δημιουργία του μοντέλου επεξεργασίας δεδομένων* (συγχώνευση των δύο προηγούμενων μοντέλων).

Η κατασκευή των αντίστοιχων μοντέλων και στις τρεις περιπτώσεις γίνεται κυρίως με τη χρήση διαγραμματικών τεχνικών. Οι τεχνικές αυτές βοηθούν ιδιαίτερα στην κατανόηση, την ανάλυση και την παρουσίαση των προβλημάτων και των απαιτήσεων του συστήματος αφού έχουν την δυνατότητα να περιγράφουν τις λειτουργίες του με χρήση εικόνων (διαγραμμάτων, δένδρων, πινάκων). Η χρήση των διαγραμμάτων δίνει σημαντικά πλεονεκτήματα γιατί προσφέρει τον απαραίτητο βαθμό εποπτείας για την αναπαράσταση, τον έλεγχο και την ανάλυση ενός συστήματος.

Χρησιμοποιούνται κατ' αρχας από το προσωπικό που ασχολείται με την ανάλυση και τον σχεδιασμό του συστήματος, είτε είναι εξειδικευμένο σε θέματα πληροφορικής, είτε είναι λογιστές που λαμβάνουν μέρος στον σχεδιασμό του λογιστικού συστήματος και κατά δεύτερον από τους ελεγκτές για τον έλεγχο ενός

συστήματος.

Οι διαγραμματικές τεχνικές που είναι ευρύτερα διαδεδομένες και χρησιμοποιούνται στην ανάπτυξη του μοντέλου ενός Π.Σ. χωρίζονται κυρίως σε δύο κατηγορίες. Σε αυτές που χρησιμοποιούνται για τη γενική περιγραφή μιας λειτουργίας ή ενός προβλήματος και σε αυτές που χρησιμοποιούνται για την λεπτομερή περιγραφή τους. Φυσικά υπάρχουν διαγραμματικές τεχνικές που ανήκουν και στις δύο κατηγορίες, δηλαδή μπορούν να χρησιμοποιηθούν τόσο για γενική όσο και τη λεπτομερή περιγραφή των λειτουργιών ή/και των προβλημάτων ενός Π.Σ..

Οι πιο γνωστές και χρησιμοποιούμενες τεχνικές που ανήκουν στην κατηγορία των τεχνικών γενικής περιγραφής είναι οι εξής:

- Διαγράμματα ροής
- διαγράμματα ροής δεδομένων
- διαγράμματα δομής (structure diagrams)
- διαγράμματα HIPO (Hierarchical Input - Process - Output)
- διαγράμματα Wamier -Orr
- διαγράμματα HOS(Higher Order Software)
- διαγράμματα ενεργειών
- διαγράμματα Jackson

Οι πιο γνωστές και χρησιμοποιούμενες τεχνικές που ανήκουν στην κατηγορία των τεχνικών λεπτομερούς περιγραφής είναι οι εξής:

- δομημένη περιγραφή και ψευδοκώδικας
- δέντρα αποφάσεων
- πίνακες αποφάσεων
- διαγράμματα Nassi-Shneiderman

Οι περισσότερες από τις παραπάνω διαγραμματικές τεχνικές χρησιμοποιούνται καθ' όλη τη διάρκεια της ανάλυσης, του σχεδιασμού, της υλοποίησης και της λειτουργίας ενός Π.Σ.

Γενικά, τα διαγράμματα που έχουν αναπτυχθεί για τη δημιουργία και κατανόηση ενός συστήματος αποτελούν ένα σημαντικό μέρος του και το συνοδεύουν σε όλο τον κύκλο ζωής του. Σε αυτά προτρέπει οποιοσδήποτε θέλει να κατανοήσει ή να βελτιώσει ή να ελέγξει ένα Λογιστικό Πληροφοριακό Σύστημα.

Εμείς θα αχοληθούμε κυρίως με τα διαγράμματα ροής, τα διαγράμματα ροής δεδομένων και τα διαγράμματα HIPO, τα οποία είναι και τα πιο φημισμένα. Στα υπόλοιπα γίνεται απλά μια γενική αναφορά.

2.2 Έννοια του μοντέλου

Ένα **μοντέλο** αποτελεί μια εξιδανίκευση μιας οντότητας ή κατάστασης, και αποτελεί ένα σύστημα *αντικειμένων, ιδιοτήτων, και σχέσεων* τα οποία έχουν σχεδιαστεί για να μιμούνται, σε συγκεκριμένες σχετικές καταστάσεις, τον χαρακτήρα ενός δοσμένου συστήματος του αληθινού κόσμου. Πάντα δίνουν έμφαση σε κάποια από τα στοιχεία της πραγματικότητας και υποβιβάζουν ή αγνοούν τα υπόλοιπα.

Δεν τα κάνουμε με αποκλειστικό σκοπό να βελτιώσουμε μια διαδικασία, αλλά πολλές φορές τα δημιουργούμε για να καταλάβουμε πως αυτή δουλεύει.

2.2.1 Business model

Η διαμόρφωση είναι μια απλουστευμένη αντιπροσώπευση της σύνθετης πραγματικότητας που μας επιτρέπει να αποβάλλουμε τις άσχετες λεπτομέρειες και να στρέψει την προσοχή σε μια ή περισσότερες σημαντικές πτυχές σε έναν χρόνο. Το ιδανικό επιχειρησιακό μοντέλο αποτελείται από ένα διάγραμμα που περιλαμβάνει όλες τις σημαντικές πτυχές μιας επιχείρησης, το οποίο όμως στην πραγματικότητα δεν είναι συνήθως πιθανό. Αντ' αυτού, χρησιμοποιούμε διάφορες διαφορετικές όψεις που συλλαμβάνουν τις συγκεκριμένες πληροφορίες για τη μια πτυχή της επιχείρησης. Κάθε όψη καλείται μοντέλο και περιλαμβάνει μια γραφική αντιπροσώπευση και μια περιγραφή των στοιχείων της. Σύμφωνα με UMIST, το επιχειρησιακό μοντέλο αποτελείται από τα ακόλουθα υποπρότυπα:

- .. μοντέλο λειτουργίας οργάνωσης
- .. μοντέλο ροής πληροφοριών οργάνωσης
- .. μοντέλο επιχειρησιακής διαδικασίας οργάνωσης
- .. και μοντέλο στοιχείων οργάνωσης

Το μετα-μοντέλο εκθέτει πολλές σύνθετες σχέσεις μεταξύ των εννοιών που χρησιμοποιούνται ως πρότυπα στοιχεία στην επιχειρηματική διαμόρφωση. Μια συνοπτική περιγραφή της μετα-πρότυπης σημασιολογίας δίνεται στην επόμενη παράγραφο. Η επιχειρησιακή μονάδα περιγράφει μια οργάνωση, ή το τμήμα που έχει σαν στόχο τις επιχειρησιακές δραστηριότητες.

Οι επιχειρησιακές μονάδες μπορούν να τακτοποιηθούν σε μια ιεραρχία που απεικονίζει τη δομή οργάνωσης. Είναι μια κεντρική οντότητα του μοντέλου και συμφωνεί με τη σχέση με διάφορες άλλες έννοιες:

.. Θέση/Τοποθεσία: η θέση/τοποθεσία προσαρμόζει τις πληροφορίες για τη θέση επιχειρησιακών μονάδων. Σημειώστε ότι η επιχειρησιακή μονάδα μπορεί να διανεμηθεί σε πολλές θέσεις.

.. Όραμα, Αποστολή, Στόχος: το όραμα και η αποστολή είναι σημαντικά στοιχεία για μια επιχείρηση. Το όραμα περιγράφει το σκοπό της ενώ μια αποστολή τεκμηριώνει τα απαραίτητα επιτεύγματα για να εκπληρώσει το όραμα. Η αποστολή αποσυντίθεται περαιτέρω σε έναν ή πολλούς στόχους που αντιπροσωπεύουν τα επιθυμητά κράτη μιας επιχείρησης (κράτη που μια επιχείρηση επιθυμεί να επιτύχει, π.χ. η θέση της στην αγορά).

.. Ρόλος οργάνωσης: ένας οργανωτικός ρόλος αντιπροσωπεύει ένα ή περισσότερα ανθρώπινα στοιχεία συμπεριφοράς διαθέσιμα μέσα σε μια οργάνωση. Κάθε ρόλος οργάνωσης επομένως συνδέεται με ένα πρόσωπο, έναν ανθρώπινο δράστη που συμμετέχει στις επιχειρησιακές διαδικασίες. Όπως εμφανίζεται, ένας ανθρώπινος δράστης μπορεί να έχει διάφορους διαφορετικούς ρόλους μέσα σε μια οργάνωση.

.. Πηγή: η πηγή είναι ένα αφηρημένο στοιχείο. Αντιπροσωπεύει μια οντότητα που συμμετέχει στις στοιχειώδεις εργασίες που αποτελούν ή είναι σχετικές με τις επιχειρησιακές διαδικασίες. Η πηγή μπορεί να είναι ένα φυσικό στοιχείο, μία πηγή πληροφοριών ή μια οργανωτική μονάδα. Οι σημαντικές σχέσεις υπάρχουν επίσης μεταξύ των στοιχείων, όπως η οντότητα, η λειτουργία, ο οργανωτικός ρόλος και

η επιχειρησιακή μονάδα. Κάθε ένα από εκείνα τα στοιχεία συνδέεται με άλλα μέσω των πολλαπλών σχέσεων. Παραδείγματος χάριν, μια λειτουργία χρησιμοποιεί διάφορες οντότητες κατά την εκτέλεση, και μια οντότητα εξυπηρετεί ως στοιχείο συμπεριφοράς πληροφοριών σε διάφορες λειτουργίες.

2.3 Έννοια Μοντελοποίησης

Με τη χρήση κατάλληλου λογισμικού δεν είναι μόνο δυνατό να προσομοιώνουμε διάφορες καταστάσεις και φαινόμενα, χρησιμοποιώντας ήδη υπάρχοντα στο πρόγραμμα μοντέλα, αλλά έχουμε τη δυνατότητα να δημιουργήσουμε δικά μας μοντέλα και στη συνέχεια να τα προσομοιώνουμε. Η έννοια της μοντελοποίησης παρουσιάζεται από τη στιγμή που ο άνθρωπος προσπαθεί να κατανοήσει και να ερμηνεύσει τα διάφορα φαινόμενα που παρατηρεί, καθώς και να κάνει προβλέψεις για το πώς θα συμπεριφερθούν ορισμένα συστήματα γύρω του. Η δημιουργία μοντέλων περιέχει την προοπτική του χειρισμού τους και όχι του χειρισμού των αντικειμένων ή εννοιών που αναπαριστούν (γεγονός που δεν είναι πάντα εφικτό) και επιτρέπει τη δυνατότητα υπολογισμών, την ανακάλυψη νέων σχέσεων και νέων σχημάτων, την κατάκτηση νέων βεβαιοτήτων αλλά και την ανατροπή κάποιων άλλων.

Η επιστημονική δραστηριότητα συνίσταται, κατά κύριο λόγο στη δημιουργία μοντέλων των φαινομένων και των αντικειμένων τα οποία μελετά. Τα μοντέλα αυτά μπορεί να είναι πολύ φορμαλιστικά (όπως για παράδειγμα στη Φυσική με τα μαθηματικά μοντέλα) ή να παρέχουν μια απλή και πραγματική αναπαράσταση (π.χ. στη Βιολογία με τη χρήση απλών εικόνων και μεταφορών από την καθημερινή ζωή).

Η μοντελοποίηση ως διαδικασία είναι η ίδια είτε γίνεται με κλασικά μέσα, είτε βασίζεται σε υπολογιστή. Στον υπολογιστή το μοντέλο που οικοδομούμε είναι *εκτελέσιμο*. Δηλαδή μπορούμε να προσομοιώσουμε τη συμπεριφορά ενός

μοντέλου και να δούμε την εξέλιξή του, κάτι που τις περισσότερες φορές δεν είναι δυνατόν με τα παραδοσιακά μέσα μοντελοποίησης. Συνολικά τα Πληροφοριακά Συστήματα μοντελοποίησης αποτελούν ένα σημαντικό εργαλείο γιατί μπορούν να συνεισφέρουν σε τρία βασικά επίπεδα:

- Στον αρχικό σχεδιασμό του μοντέλου (ποιο είναι το πρόβλημα, ποιος ο σκοπός του μοντέλου, ποιος θα το χρησιμοποιήσει, και ποιοι είναι οι κύριοι άξονές του).
- Στη διαμόρφωσή του (ορισμός των σχέσεων ανάμεσα στα συστατικά του μοντέλου).
- Στον έλεγχο του μοντέλου (αποτίμηση του μοντέλου με έλεγχο και εξέταση της συμπεριφοράς του σε σχέση με τους σκοπούς του).

Η προσομοίωση είναι η τεχνική μίμησης της συμπεριφοράς ενός συστήματος από ένα άλλο σύστημα. Επεκτείνεται σε ιδιαίτερα πολύπλοκους χώρους, όπως η όραση, η ανθρώπινη φωνή και ο ήχος γενικότερα, η συμπεριφορά επιστημονικών μοντέλων, η εξέλιξη κοινωνικο-οικονομικών συστημάτων και βρίσκει έναν κατεξοχήν χώρο εφαρμογής, τα ηλεκτρονικά παιχνίδια. Ο όρος της προσομοίωσης που άλλοτε υποδήλωνε την ιδέα της απάτης και του δόλου γίνεται στις μέρες μας συνώνυμο της εμπιστοσύνης και της κυριαρχίας.

2.3.1 Μοντελοποίηση Επιχειρηματικών Διαδικασιών

Η περιγραφική δύναμη ενός μοντέλου το καθιστά ως μέσο επικοινωνίας ανάμεσα σε ανθρώπους. Χρειαζόμαστε μια ευρέως αποδεκτή μέθοδο μοντελοποίησης και μια περιεκτική και απλή διαγραμματική τεχνική, ικανή να μπορεί να αναπαραστήσει την πραγματικότητα μιας Επιχειρηματικής Διαδικασίας,

αποβάλλοντας την περιπλοκότητά της και αποσαφηνίζοντας τα σημαντικά αρχιτεκτονικά χαρακτηριστικά της. Οι λόγοι που οδήγησαν στην μοντελοποίηση είναι:

- Βελτιστοποίηση οργανωσιακών αλλαγών
- Αποθήκευση επιχειρηματικής γνώσης σε μοντέλα αναφοράς
- Χρήση σε διαδικασίες τεκμηρίωσης (πχ ISO 9000)
- Υπολογισμός χρόνου και κόστους εκτέλεσης των διαδικασιών
- Υλοποίηση διαδικασιών ροής πληροφοριών για την κατασκευή λογισμικού και συστημάτων ροής εργασίας.

2.3.1.1 Οι ρόλοι κατά την διαδικασία της μοντελοποίησης μιας επιχειρηματικής διαδικασίας

Κατά την διάρκεια του προγράμματος της μοντελοποίησης μιας ΕΔ, καθορίζονται οι παρακάτω ρόλοι:

- Ο **Manager/Owner**. Είναι κάποιος από τον οργανισμό με σκοπό να:
 - Ενθαρρύνει την υπόλοιπη ομάδα
 - Κατανοεί τα μοντέλα
- Ο **Αναλυτής του Συστήματος**
 - Αυτός που δημιουργεί τα μοντέλα
- Ο **ειδικός στο πεδίο**
 - Αυτός που επικυρώνει τα μοντέλα



2.3.1.2 Παραδείγματα

α. Παράδειγμα μοντελοποίησης μιας ΕΔ - Περιγραφή

Ένας πελάτης θέλει να παραγγείλει διάφορα από τα παραγόμενα προϊόντα ενός οργανισμού.

Ο οργανισμός, βασιζόμενος στις πληροφορίες για τον πελάτη και των προϊόντων που παρήγγειλε εξετάζει την δυνατότητα παραγωγής τους. Μόλις δημιουργηθεί η παραγγελία, επεξεργάζεται από τα Τμήματα Πωλήσεων, Προμηθειών και Παραγωγής, και οι απαραίτητες πρώτες ύλες παραγγέλλονται από τους προμηθευτές.

Όταν έρθουν οι πρώτες ύλες και γίνει το πλάνο παραγωγής, το προϊόντα κατασκευάζονται, ενώ τέλος το Τμήμα Μεταφορών τα μεταφέρει μαζί με τα απαραίτητα έγγραφα (δελτίο αποστολής, τιμολόγιο) στον πελάτη.

β. Παράδειγμα μοντελοποίησης μιας ΕΔ – Εντοπίζοντας τους εμπλεκόμενους (οντότητες).

Οι οργανωσιακές μονάδες (οντότητες) που εντοπίζονται στην ΕΔ είναι οι ακόλουθες:

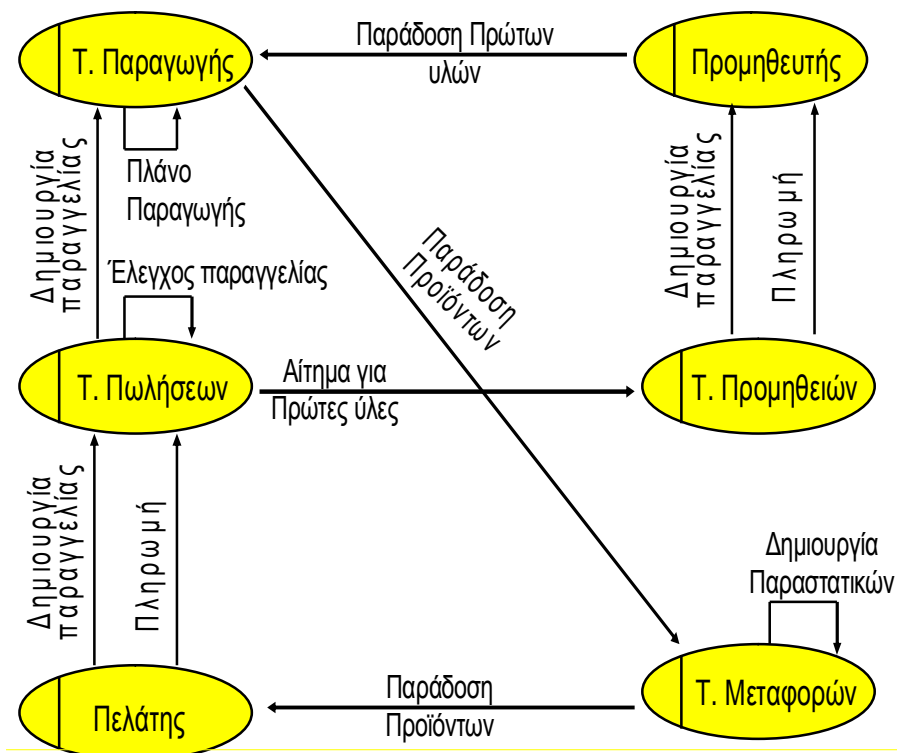
- § Πελάτης
- § Τμήμα Πωλήσεων
- § Τμήμα Παραγωγής
- § Τμήμα Προμηθειών
- § Τμήμα Μεταφορών
- § Προμηθευτής

γ. Μοντελοποιώντας τις σχέσεις μεταξύ των οντοτήτων μιας ΕΔ.

Τις σχέσεις μεταξύ των οντοτήτων μιας ΕΔ μπορούμε να τις αποδώσουμε με την χρήση ενός Διαγράμματος Αλληλεπιδράσεων (Interaction Diagram).

Ένα Διάγραμμα Αλληλεπιδράσεων είναι ένα στατικό διάγραμμα στο οποίο αναπαριστούνται οι οργανωσιακές μονάδες (organizational unit) που εμπλέκονται στην ΕΔ, οι οποίες συνδέονται μεταξύ τους με βέλη τα οποία παρουσιάζουν τις αλληλεπιδράσεις οι οποίες λαμβάνουν χώρα μεταξύ τους. Η χρονική ακολουθία των διαδικασιών (processes) που εκτελούνται δεν εμφανίζεται σε αυτό το διάγραμμα. Παρόλα αυτά εξασφαλίζει μία αρχική εικόνα της δομής της ΕΔ.

Παράδειγμα μοντελοποίησης μιας ΕΔ – Εντοπίζοντας τις σχέσεις μεταξύ των οντοτήτων.



2.4 Περιγραφή και ανάλυση των διαγραμματικών τεχνικών

2.4.1 Διαγράμματα ροής

Ένα διάγραμμα ροής (flowchart) είναι ένα συμβολικό διάγραμμα που δείχνει τη ροή δεδομένων και την αλληλουχία των λειτουργιών σε ένα σύστημα.

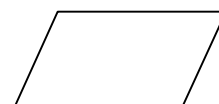
2.4.1.1 Σύμβολα διαγραμμάτων ροής

Τα σύμβολα χρησιμοποιούνται σε ένα διάγραμμα ροής για να αναπαραστήσουν τις λειτουργίες ενός συστήματος. Η σειρά εκτέλεσης των λειτουργιών προσδιορίζεται από τις γραμμές ροής.

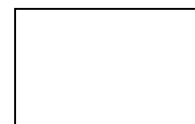
Υπάρχουν τέσσερις ομάδες συμβόλων διαγραμμάτων ροής εργασίας: τα βασικά σύμβολα, τα εξειδικευμένα σύμβολα εισόδου/εξόδου, τα εξειδικευμένα σύμβολα επεξεργασίας και τα επιπρόσθετα σύμβολα.

- **Βασικά σύμβολα**

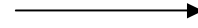
το σύμβολο **εισόδου/εξόδου** αναπαριστά μια λειτουργία εισόδου εξόδου.



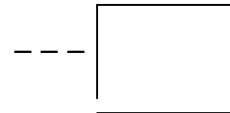
το σύμβολο **επεξεργασίας** αναπαριστά κάθε είδος επεξεργασίας π.χ. εκτέλεση κάποιων λειτουργιών που έχουν ως αποτέλεσμα την αλλαγή μιας τιμής ή την λήψη μιας απόφασης σχετικά με ποια κατεύθυνση ροής θα ακολουθήσει στη συνέχεια.



το σύμβολο της **γραμμής ροής** χρησιμοποιείται για να ενώσει δύο άλλα σύμβολα. Υποδεικνύει την αλληλουχία της εκτέλεσης των λειτουργιών. Σε μια διασταύρωση, οι γραμμές δεν έχουν κάποια λογική συσχέτιση. Κάθε γραμμή ροής που εισέρχεται ή εξέρχεται από/προς το σημείο συνένωσης θα πρέπει να έχει κεφαλή βέλους κοντά στο σημείο συνένωσης.



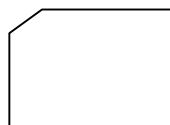
το σύμβολο **σχεδιασμού** χρησιμοποιείται όταν θέλουμε να σχολιάσουμε, να επεξηγήσουμε ή να διευρυνήσουμε ένα άλλο σύμβολο. Η διακεκομμένη γραμμή ενώνεται με το σύμβολο που σχολιάζεται.



- **Τα εξειδικευμένα σύμβολα εισόδου/εξόδου (I/O)**

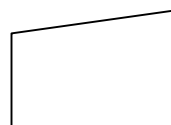
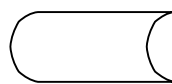
Τα σύμβολα αυτά μπορούν να αναπαραστήσουν τη λειτουργία I/O και να δηλώσουν το μέσο στο οποίο οι πληροφορίες καταγράφονται ή τον τρόπο διαχείρισης της πληροφορίας ή και τα δύο.

Διάτρητη
κάρτα



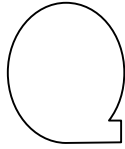
έγγραφο

on-line
αποθήκευση

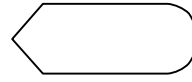


χειροκίνητη
καταχώρηση

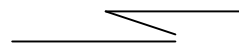
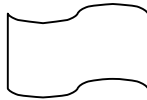
Μαγνητική ταινία



απεικόνιση

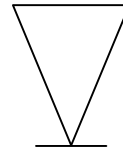
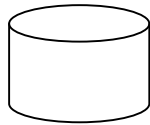


Διάτρητη ταινία



σύνδεση
επικοινωνία

μαγνητικός
αποθήκευση
δίσκος

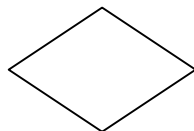


off-line

- **Τα εξειδικευμένα σύμβολα επεξεργασίας**

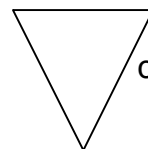
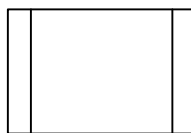
Τα εξειδικευμένα σύμβολα επεξεργασίας αναπαριστούν τη λειτουργία της επεξεργασίας και επιπλέον προσδιορίζουν ένα συγκεκριμένο τύπο επεξεργασίας της πληροφορίας.

απόφαση



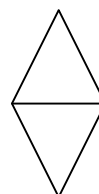
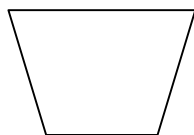
Βοηθητική
λειτουργία

προκαθορισμένη
διαδικασία



συγχώνευση

Χειροκίνητη
λειτουργία



ταξινόμηση

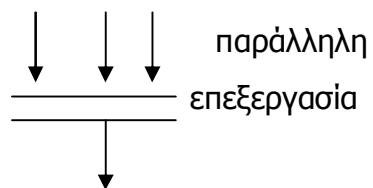
σύγκριση



- **Τα επιπρόσθετα σύμβολα**

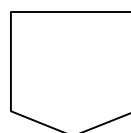
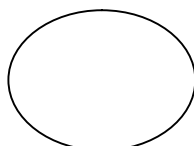
Τα επιπρόσθετα σύμβολα μπορούν να χρησιμοποιηθούν για να διευκρινίσουν ένα διάγραμμα ροής ή για να το κάνουν πιο αντιπροσωπευτικό.

τερματικό



παράλληλη
επεξεργασία

σύνδεση



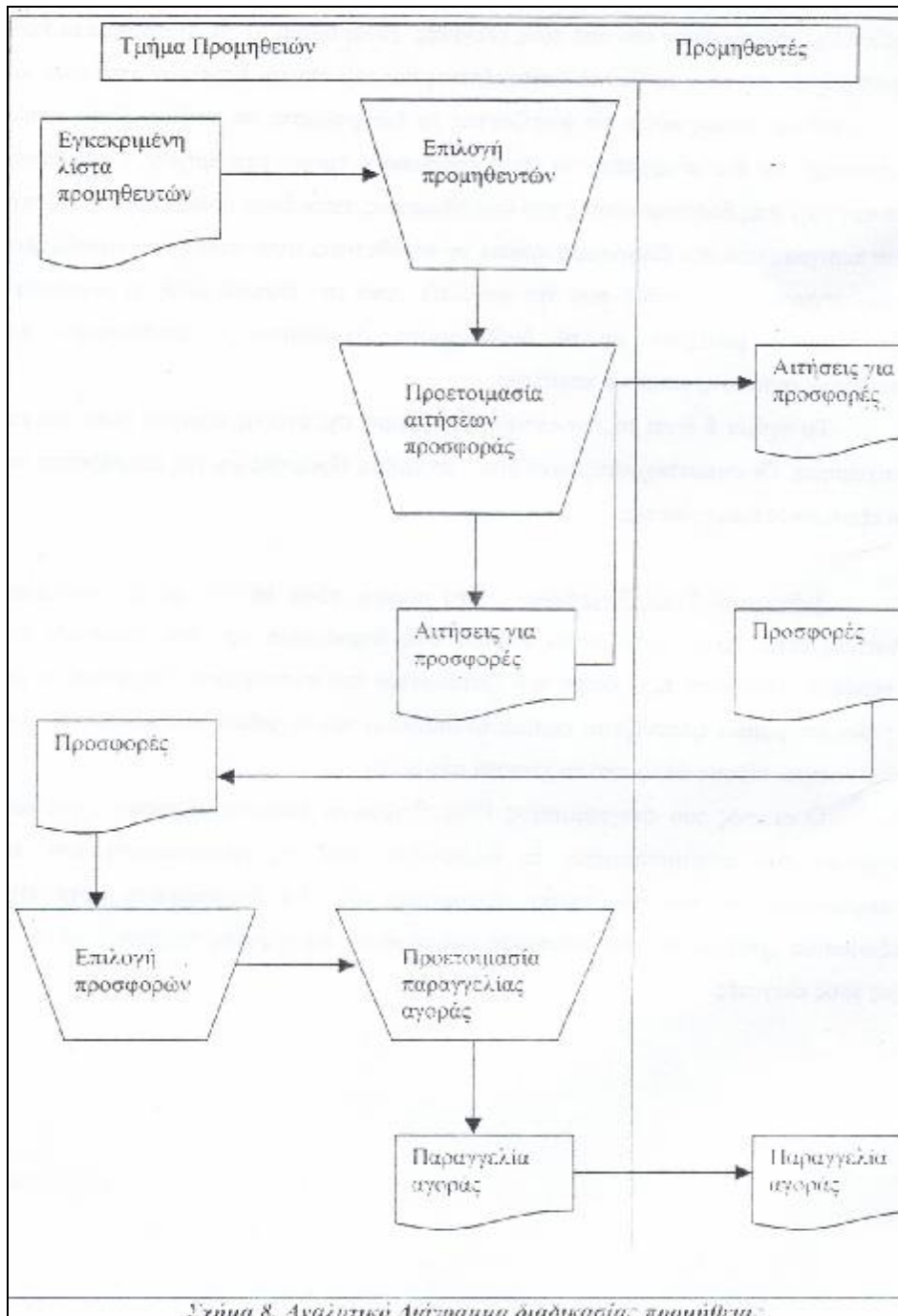
σύνδεση
με άλλη σελίδα

2.4.1.2 Είδη διαγραμμάτων ροής

Τα διαγράμματα ροής χρησιμοποιούνται με διάφορες μορφές που η καθεμία τους εξυπηρετεί και διαφορετικό σκοπό. Σε όλες αυτές τις διαφορετικές μορφές ο συμβολισμός παραμένει ο ίδιος.

- Διαγράμματα ροής συστημάτων: δίνουν έμφαση στην παράθεση των διαφόρων λειτουργικών μονάδων παρά στη λεπτομερή περιγραφή της καθεμίας και έχουν σαν σκοπό να περιγράψουν τις λειτουργίες ολόκληρου του συστήματος. χρησιμοποιούνται κυρίως από το προσωπικό ανάπτυξης συστημάτων και από τους ελεγκτές.
- Διαγράμματα ροής προγράμματος: επεξηγούν με λεπτομέρεια τις διάφορες λειτουργικές μονάδες του διαγράμματος ροής συστήματος και χρησιμοποιούνται κατά τον σχεδιασμό ενός Λογιστικού Πληροφοριακού Συστήματος συνήθως από το προσωπικό ανάπτυξης συστημάτων.
- Αναλυτικά διαγράμματα ροής: είναι όμοια με τα Διαγράμματα ροής συστημάτων ως προς το βαθμό λεπτομέρειας και τον σκοπό, απαντούν στο ερώτημα <ποιος κάνει τι> χωρίζοντας τα διαγράμματα σε στήλες και χρησιμοποιούνται και από το προσωπικό ανάπτυξης και απ' τους ελεγκτές.
- Διαγράμματα ροής εγγράφων: εστιάζονται κυρίως στη δημιουργία και στη διακίνηση των εγγράφων παρά στην περιγραφή των λειτουργιών του συστήματος. Επομένως, σε τέτοια διαγράμματα εμφανίζεται κυρίως το σύμβολο του εγγράφου και μόνο όταν είναι απαραίτητο κάποιο άλλο σύμβολο. Σκοπός του είναι η περιγραφή της δημιουργίας, της δρομολόγησης, της επεξεργασίας και του προορισμού κάθε εγγράφου που χρησιμοποιείται.

2.4.1.3 Ένα παράδειγμα διαγράμματος ροής φαίνεται στο παρακάτω σχήμα 8:



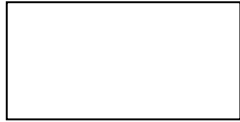
2.4.2 Διαγράμματα ροής δεδομένων

Τα *διαγράμματα ροής δεδομένων - ΔΡΔ (Data Flow Diagrams - DFD)* αποτελούν τη βασική διαγραμματική τεχνική της *δομημένης ανάλυσης και σχεδίασης (Structured Analysis and Design - SADT)*. Χρησιμοποιούνται για να απεικονίσουν τη ροή των *δεδομένων* μέσα σε ένα Π.Σ. και είναι ιδιαίτερα χρήσιμα για την περιγραφή και αναγνώριση των λειτουργιών ενός Π.Σ. και των διαδοχικών μετασχηματισμών που υφίστανται τα δεδομένα που κυκλοφορούν σε αυτό. Τα ΔΡΔ ανήκουν στις διαγραμματικές τεχνικές λογικής - γενικής περιγραφής.

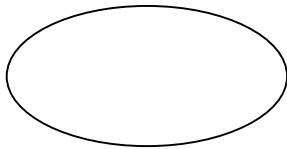
Τα βασικά στοιχεία ενός ΔΡΔ είναι τα εξής:

- Οι *πηγές* παροχής των δεδομένων και οι *προορισμοί* τους (συμβολίζονται με ορθογώνια παραλληλόγραμμα).
- Οι *ροές* δεδομένων που αποτελούν τα κανάλια στα οποία ρέουν τα δεδομένα (συμβολίζονται με βέλη τα οποία δείχνουν την κατεύθυνση της πληροφορίας).
- Η *επεξεργασία* των δεδομένων που αντιστοιχεί σε μετασχηματισμούς δεδομένων (συμβολίζεται με ελλείψεις).
- Οι *αποθήκες* δεδομένων που αποτελούν τους αποθηκευτικούς χώρους του Π.Σ., δηλαδή τα αρχεία αποθήκευσης των δεδομένων (συμβολίζονται με ορθογώνια παραλληλόγραμμα των οποίων η δεξιά κάθετη πλευρά είναι ανοικτή).

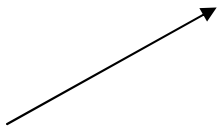
Τα βασικά στοιχεία ενός ΔΡΔ:



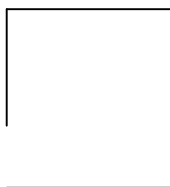
Εξωτερική οντότητα→χρήστης(άνθρωπος), άλλο σύστημα, υλικό



Διαδικασία→μετασχηματισμός, επεξεργασία δεδομένων



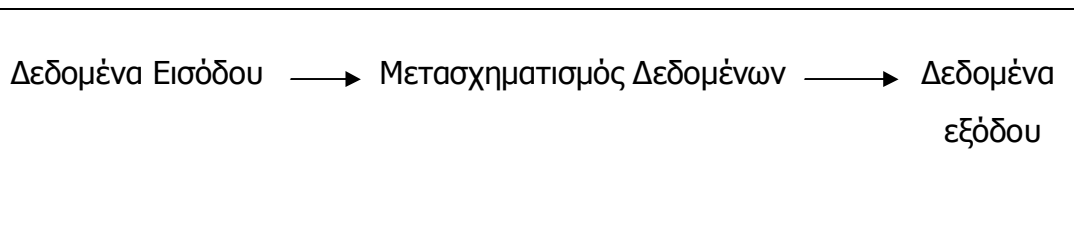
Απλά ή σύνθετα δεδομένα(προσδιορίζονται με ετικέτα που γράφεται πάνω από το βέλος)



Αποθήκη δεδομένων(μέσα στην αποθήκη πρέπει να αναγράφεται το είδος των δεδομένων που είναι αποθηκευμένα)

Πρέπει να τονιστεί ότι ένα ΔΡΔ δεν είναι είναι *διάγραμμα ροής ελέγχου* (*flowchart*). Το ΔΡΔ αναφέρεται και περιγράφει τη ροή των δεδομένων μέσα σε ένα ΠΣ.. Το διάγραμμα ροής ελέγχου ή διάγραμμα ροής αναφέρεται και περιγράφει τη ροή του ελέγχου σε έναν αλγόριθμο επίλυσης ενός προβλήματος. Κατά τη διαδικασία ανάπτυξης του ΔΡΔ ο αναλυτής δεν πρέπει να εμπλέκεται σε λεπτομέρειες που αφορούν την ακολουθία της επεξεργασίας. Αν κάποιος κατά τη σχεδίαση σκέπτεται για βρόχους και αποφάσεις βρίσκεται σίγουρα σε λάθος κατεύθυνση!

Ένα από τα πλεονεκτήματα των ΔΡΔ είναι η ιεραρχική δομή τους. Ξεκινώντας από την αρχική διαδικασία ο αναλυτής προσπαθεί να την διασπάσει (αναλύσει) σε άλλες απλούστερες καθώς κατεβαίνει τα επίπεδα της ανάλυσης. Επομένως, η ανάλυση ενός Π.Σ. με χρήση ΔΡΔ βασίζεται σε διαδοχική εκλέπτυνση των διαδικασιών που εκτελούνται σε αυτό. Για τη διευκόλυνση της κατανόησης ενός ΔΡΔ οι διαδικασίες αριθμούνται. Η βασική αρχή που πρέπει να ακολουθεί κανείς στη σχεδίαση ΔΡΔ είναι *η διατήρηση της συνέχειας στη ροή της πληροφορίας* (η είσοδος και η έξοδος σε κάθε εκλέπτυνση πρέπει να παραμένει η ίδια).



2.4.2.1 Βασικές αρχές της κατασκευής ενός ΔΡΔ

Όταν κατασκευάζεται ένα ΔΡΔ για την ανάλυση και το σχεδιασμό ενός Π.Σ. πρέπει να λαμβάνεται υπόψη ότι αυτό πρέπει να αποτελείται από τα εξής βασικά διαγράμματα:

1. Γενικό διάγραμμα (διάγραμμα επιπέδου 0)

- Περιγράφει τα «όρια» του συστήματος (πηγές και προορισμοί του συστήματος και τα σχετικά εισερχόμενα και εξερχόμενα δεδομένα).

2. Διάγραμμα 1ου επιπέδου

- α Πρώτη γενική θεώρηση του συστήματος.
- α Αναλύεται το όλο σύστημα στις βασικές διαδικασίες του.
- α Περιγράφονται οι βασικές διαδικασίες καθώς και οι ροές και αποθηκεύσεις δεδομένων.

3. Διάγραμμα κατώτερου επιπέδου (2ου . 3ου . κλπ.)

- Αναπαριστά με περισσότερη λεπτομέρεια μια διαδικασία (και τις σχετικές εισερχόμενες ή εξερχόμενες ΡΔ) που εμφανίζεται σε ένα διάγραμμα ανώτερου επιπέδου.

Ένα σωστό ΔΡΔ πρέπει να παρέχει στοιχεία για:

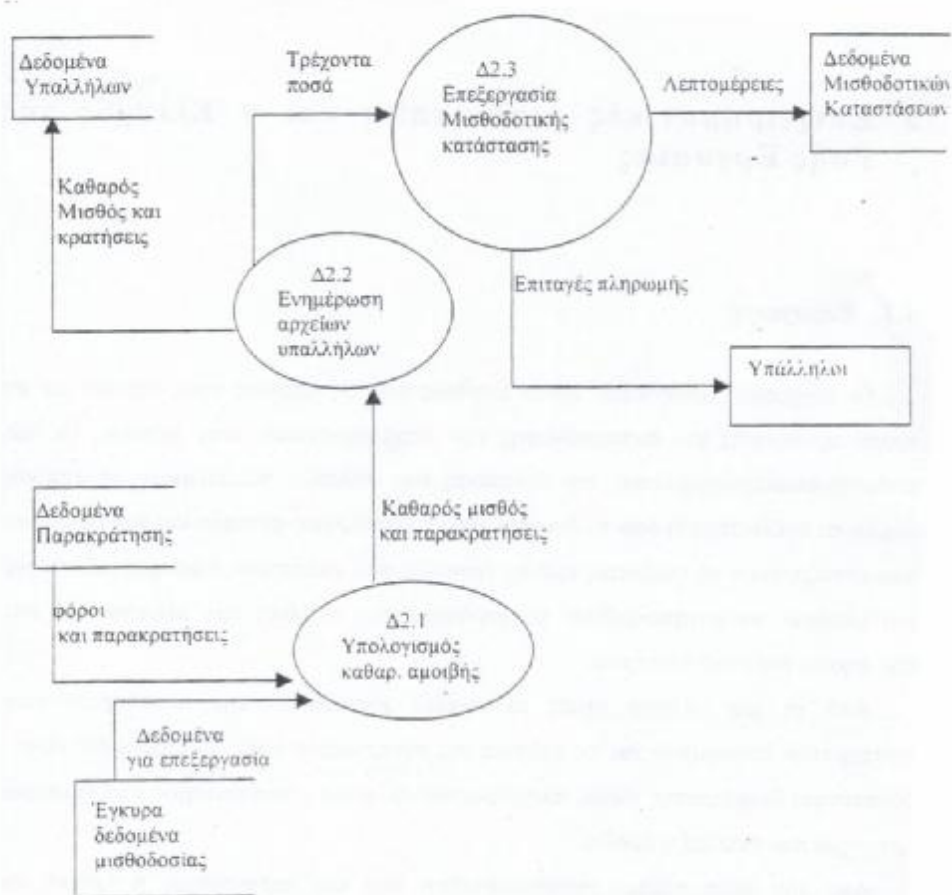
- Ø η διάσπαση ενός συστήματος σε υποσυστήματα.
- Ø Τις ροές δεδομένων στο σύστημα.
- Ø Τα εισερχόμενα και εξερχόμενα δεδομένα καθώς και τις αποθηκεύσεις τους.
- Ø Τις πηγές και τους προορισμούς του συστήματος.

Ένα σωστό ΔΡΔ δεν πρέπει να παρέχει στοιχεία για:

- Ø Τη δομή των ρών δεδομένων στο σύστημα.
- Ø Τις ανάγκες ανάκτησης αποθηκευμένων δεδομένων.
- Ø Τις αποφάσεις που λαμβάνονται στο σύστημα.
- Ø Τις επαναλήψεις των διαδικασιών και τους αριθμητικούς υπολογισμούς.

2.4.2.2 ΠΑΡΑΔΕΙΓΜΑ ΔΙΑΓΡΑΜΜΑΤΟΣ ΡΟΗΣ ΔΕΔΟΜΕΝΩΝ

Στο παρακάτω σχήμα βλέπουμε ένα διάγραμμα ροής δεδομένων για τη διαδικασία <υπολογισμός μισθου>.

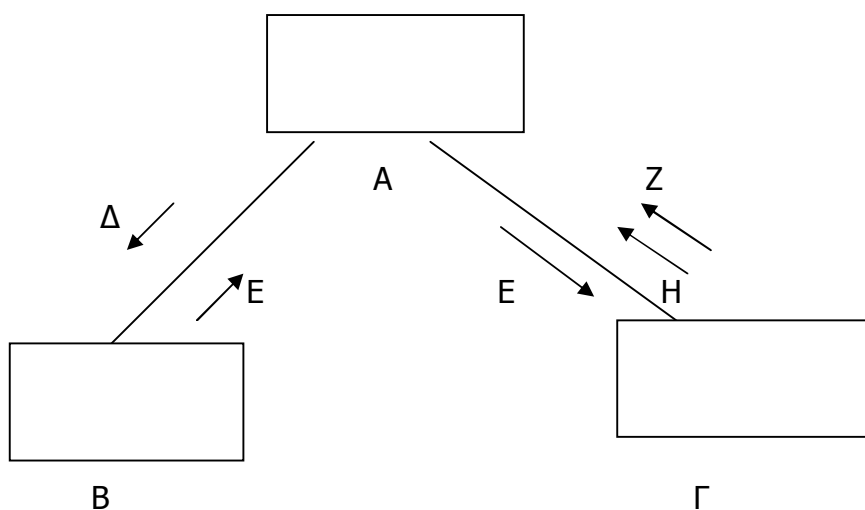


2.4.3 Διαγράμματα δομής

Τα διαγράμματα δομής ανήκουν στις διαγραμματικές τεχνικές γενικής περιγραφής και μαζί με τα διαγράμματα ροής δεδομένων αποτελούν τις πιο διαδεδομένες διαγραμματικές τεχνικές της κατηγορίας αυτής.

Όπως είναι γνωστό, το βασικό δομικό συστατικό ενός συστήματος/προγράμματος είναι η ενότητα (module). Τα δομημένα συστήματα/προγράμματα είναι στην πραγματικότητα οργανωμένα ως μια ιεραρχία ενότητων. Ένα διάγραμμα δομής λοιπόν, που χρησιμοποιείται για την περιγραφή ενός συστήματος/προγράμματος, αποτελεί ένα ιεραρχικό διάγραμμα που καθορίζει τη συνολική αρχιτεκτονική δομή του συστήματος/προγράμματος, παρουσιάζοντας τόσο τις ενότητες από τις οποίες αποτελείται το σύστημα/πρόγραμμα όσο και τις μεταξύ τους σχέσεις.

Τα διαγράμματα δομής αποτελούνται από ορθογώνια παραλληλόγραμμα και γραμμές με βέλη τα οποία ενώνουν τα ορθογώνια αυτά μεταξύ τους. Κάθε ορθογώνιο παραλληλόγραμμο παριστάνει μια ενότητα του συστήματος/προγράμματος το όνομα της οποίας αναγράφεται μέσα στο εσωτερικό του ορθογωνίου. Στο σχήμα παρουσιάζεται η γενική μορφή ενός διαγράμματος δομής.



Η ενότητα A μπορεί να καλέσει την ενότητα B όπως επίσης και την ενότητα Γ. Αυτό υποδεικνύεται από τις γραμμές που ενώνουν τα ορθογώνια που αντιπροσωπεύουν τις ενότητες. Το διάγραμμα δεν παρουσιάζει τη σειρά με την οποία εκτελούνται οι κλήσεις αυτές ούτε καν εάν εκτελούνται στην πραγματικότητα. Όταν ολοκληρωθεί η εκτέλεση μιας υφιστάμενης διαδικασίας ο έλεγχος επιστρέφει στη διαδικασία που την κάλεσε.

Όταν η ενότητα A καλεί την ενότητα B, της στέλνει δεδομένα τύπου Δ. Όταν η B ολοκληρώσει την εκτέλεσή της, επιστρέφει δεδομένα τύπου Ε στην Α. Ομοίως, η Α επικοινωνεί με την Γ χρησιμοποιώντας δεδομένα τύπου Ε και Η.

Όταν η ενότητα Γ ολοκληρώσει τη λειτουργία της, στέλνει μια σημαία τύπου Ζ στην ενότητα Α. Μία σημαία χρησιμοποιείται ως μία ροή δεδομένων ελέγχου.

A. Σχέσεις ελέγχου

Ένα διάγραμμα δομής διατάσσει τις ενότητες του συστήματος/προγράμματος σε επίπεδα και συνδέοντας τα επίπεδα μεταξύ τους με βέλη απεικονίζει τις σχέσεις ελέγχου ανάμεσα στις ενότητες κάθε επιπέδου. Η φορά του βέλους που ενώνει δύο ενότητες καθορίζει τη φορά κατά την οποία περνάει ο έλεγχος κατά την εκτέλεση του προγράμματος. Μια ενότητα μπορεί να καλεί περισσότερες από μια ενότητες καθώς όμως στο διάγραμμα δομής δε φανερώνεται η ακολουθία δεν είναι γνωστή η σειρά με την οποία η ενότητα καλεί τις ενότητες που αποτελούν τα παιδιά της. Μια ενότητα χωρίς παιδιά, όπως συνηθίζεται στην ορολογία των δενδρικών δομών, ονομάζεται φύλλο. Στη συνέχεια παρουσιάζονται οι σχέσεις ελέγχου που εμφανίζονται σε ένα διάγραμμα δομής (Martin and Mc Clure, 1985).

Στο 1 ο επίπεδο ενός διαγράμματος δομής υπάρχει μια και μόνο ενότητα η οποία, όπως συνηθίζεται στις δενδρικές δομές, ονομάζεται ρίζα και από την οποία ξεκινάει ο έλεγχος που ασκείται στις ενότητες των υπόλοιπων επιπέδων.

Αρχίζοντας από το 1 ο επίπεδο του διαγράμματος δομής, ο έλεγχος περνάει σε όλες τις ενότητες από επίπεδο σε επίπεδο. Ο έλεγχος επιστρέφει πάντοτε στην ενότητα που τον ασκεί. Γι' αυτό το λόγο όταν ολοκληρωθεί η εκτέλεση του προγράμματος, ο έλεγχος επιστρέφει στη ρίζα.

Ανάμεσα σε δύο οποιοσδήποτε ενότητες, υπάρχει το πολύ μια σχέση ελέγχου. Έτσι, αν μια ενότητα E1 ασκήσει έλεγχο σε μια ενότητα E2, η ενότητα E2 δεν μπορεί να ασκήσει και αυτή με τη σειρά της έλεγχο στην ενότητα E1. Επίσης μια ενότητα δε μπορεί να ασκήσει έλεγχο στον εαυτό της.

B. Κοινές ενότητες

Επιτρέπεται περισσότερες από μια ενότητες να μεταφέρουν τον έλεγχο στην ίδια ενότητα.

Γ. Ενότητες βιβλιοθήκης

Πολλές φορές χρησιμοποιούνται εκ των προτέρων κάποιες υλοποιημένες ενότητες βιβλιοθήκης οι οποίες συνήθως βρίσκονται στο διάγραμμα δομής με τη μορφή φύλλων.

Δ. Μεταφορά δεδομένων

Όταν μεταφέρεται ο έλεγχος από μια ενότητα σε μια άλλη μεταφέρονται συνήθως και κάποια δεδομένα τα οποία μπορούν να μεταφερθούν προς οποιαδήποτε κατεύθυνση ανάμεσα στις δύο οντότητες. Η κατεύθυνση καθορίζεται από ένα βέλος ενώ τα ονόματα των δεδομένων που μεταφέρονται σημειώνονται δίπλα στο βέλος.

Ε. Κέντρο συναλλαγών

Όταν σε μια ενότητα ενός διαγράμματος δομής γίνεται η επεξεργασία πολλών διαφορετικών τύπων συναλλαγών, μπορεί για κάθε διαφορετικό τύπο συναλλαγής να χρησιμοποιηθεί μια επιμέρους ενότητα.

2.4.4 Διαγράμματα HIPO (Hierarchical Input - Process - Output)

Τα διαγράμματα HIPO (Hierarchical-Input-Process-Output) δηλαδή ιεραρχικής εισόδου επεξεργασίας-εξόδου, λόγω της χρήσης τους και των χαρακτηριστικών τους θεωρούνται περισσότερο τεχνικές λεπτομερούς περιγραφής. Παρ'όλα αυτά, χρησιμοποιούνται και ως τεχνικές γενικής περιγραφής. Αποτελούν μια τεχνική που χρησιμοποιεί ένα σύνολο διαγραμμάτων για να απεικονίσει την είσοδο και την έξοδο ενός προγράμματος καθώς και την επεξεργασία που υφίστανται τα δεδομένα στα πλαίσια του προγράμματος αυτού, δηλαδή τις λειτουργίες που οδηγούν από τα δεδομένα εισόδου στα δεδομένα εξόδου. Στην πραγματικότητα αποτελούν μια διαγραμματική τεχνική που περιγράφει *τι* κάνει ένα σύστημα και *όχι πως* το κάνει.

Χρησιμοποιούνται συχνά και σε συνδυασμό με τα διαγράμματα ροής δεδομένων, ενώ αποτελούν μια αρκετά καλή μέθοδο τεκμηρίωσης, αφού παρέχουν τη δυνατότητα για περιγραφή των συστημάτων και των προγραμμάτων που αναλύουν.

2.4.4.1 Δομικά στοιχεία των διαγραμμάτων HIPO

Το γενικότερο πακέτο των διαγραμμάτων HIPO αποτελείται από δύο μέρη: τον *παραστατικό πίνακα περιεχομένων* (VTOC - Visual Table of Contents) και τα *διαγράμματα εισόδου επεξεργασίας - εξόδου* (γενικά και λεπτομερειακά).

Ο παραστατικός πίνακας περιεχομένων αποτελείται από:

το ιεραρχικό διάγραμμα που αποτελεί το διάγραμμα που περιγράφει τις λειτουργίες του συστήματος/προγράμματος. Κάθε λειτουργία αναλύεται σε υπολειτουργίες. Έτσι, με αυτή τη ταξινόμηση διευκολύνεται το έργο του αναλυτή

διότι προκύπτουν διάφορα επίπεδα ανάλυσης. Το ιεραρχικό διάγραμμα καθορίζει επίσης τα ονόματα και τους αριθμούς των γενικών και των λεπτομερειακών διαγραμμάτων.

τον επεξηγηματικό πίνακα που περιέχει όλα τα σύμβολα που χρησιμοποιούνται στο πακέτο και περιγράφει το συμβολισμό τους.

το τμήμα περιγραφής που δίνει προαιρετικά μια λεπτομερειακή περιγραφή της λειτουργίας του συστήματος/προγράμματος.

Τα διαγράμματα εισόδου - επεξεργασίας - εξόδου διακρίνονται σε:

γενικά διαγράμματα που περιγράφουν γενικά κάθε λειτουργία του συστήματος. Τα δεδομένα εισόδου αναφέρονται στα αριστερά ενώ τα δεδομένα εξόδου (αποτελέσματα) αναγράφονται στα δεξιά. Στο μεσαίο ορθογώνιο του γενικού διαγράμματος υπάρχει συνήθως μια γενική περιγραφή της λειτουργίας που αναλύεται. Το διάγραμμα δεν περιγράφει πουθενά πως και που χρησιμοποιούνται τα δεδομένα εισόδου και εξόδου όπως επίσης δεν περιέχει και λεπτομέρειες υλοποίησης της συγκεκριμένης λειτουργίας.

λεπτομερειακά διαγράμματα όπου τα δεδομένα εισόδου και εξόδου παίρνουν την φυσική τους σημασία, δηλαδή εμφανίζονται με τη μορφή αρχείων ή εγγραφών σε αρχεία. Αυτά τα διαγράμματα περιέχουν και το όνομα της ρουτίνας που υλοποιεί τη συγκεκριμένη λειτουργία.

2.4.4.2 Πλεονεκτήματα των διαγραμμάτων HIPO

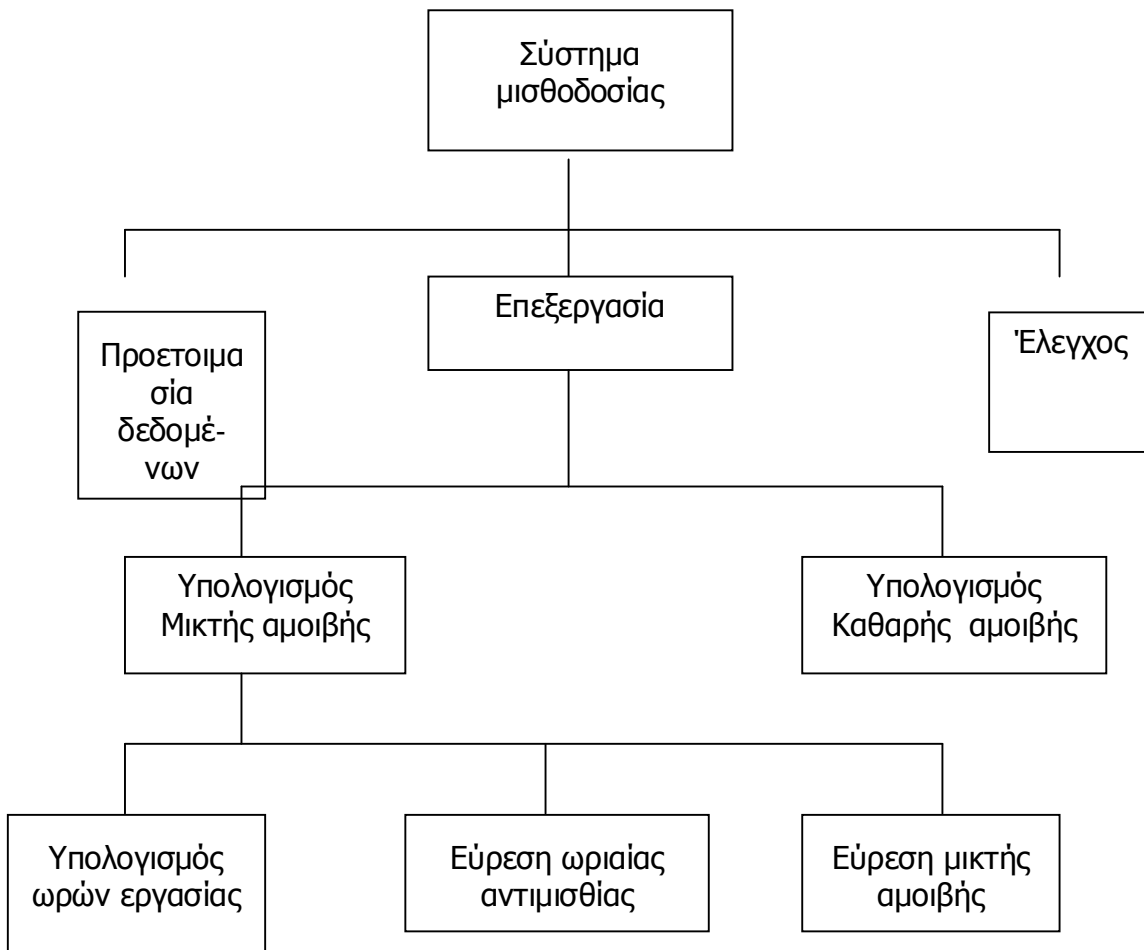
Τα διαγραμμάτων HIPO έχουν βρει μεγάλη απήχηση ανάμεσα στους αναλυτές και σχεδιαστές Π.Σ. διότι εμφανίζουν τα εξής πλεονεκτήματα:

1.χρησιμοποιούνται για τον ορισμό των, συστατικών στοιχείων ενός

συστήματος/προγράμματος με γενικό τρόπο. Ο ορισμός που προκύπτει αποτελεί σημείο αφετηρίας για την φάση της σχεδίασης.

2. χρησιμοποιούνται για την αναπαράσταση κάθε ενός από τα συστατικά συστήματος/προγράμματος με λεπτομερή τρόπο.
3. βοηθούν στην προοδευτική τεκμηρίωση του νέου συστήματος/προγράμματος.
4. υποστηρίζονται από πολλά πακέτα λογισμικού του εμπορίου.

2.4.4.3 ΠΑΡΑΔΕΙΓΜΑ ΔΙΑΓΡΑΜΜΑΤΟΣ ΗΙΡΟ



Σχήμα 1. διαδικασία μισθοδοσίας με διάγραμμα HIPO

2.4.5 Διαγράμματα HOS (Higher Order Software)

Τα διαγράμματα HOS που δημιουργήθηκαν από τους Hamilton και Zeldin (1976) είναι διαγραμματικές τεχνικές και γενικής και λεπτομερούς περιγραφής. Υποστηρίζουν αποδοτικά τη διαδικασία της λειτουργικής αποσύνθεσης και μάλιστα δίνουν τη δυνατότητα στον αναλυτή να φτάσει μέχρι τη δημιουργία απλών προγραμμάτων από τα οποία μπορεί εύκολα να παραχθεί ο αντίστοιχος εκτελέσιμος κώδικας.

Τα διαγράμματα HOS επιτρέπουν μόνο αυστηρά καθορισμένες μορφές αποσύνθεσης, σε αντίθεση με άλλες διαγραμματικές τεχνικές οι οποίες αφήνουν τις λεπτομέρειες της αποσύνθεσης των διεργασιών στην κρίση του αναλυτή/σχεδιαστή. Όμως, επειδή οι μορφές αυτές έχουν ορισθεί με μεγάλη ακρίβεια, χρησιμοποιώντας μαθηματικούς κανόνες, η ορθότητα της διαδικασίας αποσύνθεσης είναι εύκολο να ελεγχθεί.

2.4.6 Διαγράμματα Warnier – Orr

Τα διαγράμματα Warnier – Orr αποτελούν διαγραμματική τεχνική που μπορεί να χρησιμοποιηθεί τόσο για γενική όσο και για λεπτομερή περιγραφή ενός συστήματος. Βασικός τους σκοπός είναι να παραστήσουν διαγραμματικά την ιεραρχική δομή ενός προγράμματος, μιας δομής δεδομένων ή και ολόκληρου του Π.Σ. Έχουν πάρει το όνομα τους από τους Dominique Warnier και Ken Orr που πρώτοι πρότειναν τη χρησιμοποίησή τους. Τα βασικά χαρακτηριστικά των διαγραμμάτων αυτών είναι ότι κατά την σχεδίαση τους χρησιμοποιούν άγκιστρα και εκτείνονται από την αριστερή προς τη δεξιά πλευρά της σελίδας. Κάθε άγκιστρο συμβολίζει μια λειτουργική ανάλυση του στοιχείου που βρίσκεται

δεξιά από το άγκιστρο στα στοιχεία που βρίσκονται αριστερά του. Τα διαγράμματα Warnier - Orr μπορούν να χρησιμοποιηθούν είτε για την απεικόνιση της δομής ενός προγράμματος είτε για την απεικόνιση της διάσπασης των σύνθετων δεδομένων ενός προγράμματος στα στοιχεία από τα οποία αποτελούνται.

Τα διαγράμματα Warnier - Orr μπορούν να χρησιμοποιηθούν τόσο κατά τη διαδικασία κατασκευής νέων συστημάτων όσο και κατά την ανάλυση και περιγραφή συστημάτων που ήδη υπάρχουν. Επίσης, πρέπει να τονιστεί στο σημείο αυτό, ότι είναι ιδιαίτερα εύκολη η διαδικασία μετατροπής ενός διαγράμματος Warnier - Orr σε κώδικα δομημένου προγραμματισμού λόγω της χρήσης της δομής begin -end.

Σε σχέση με τα διαγράμματα HIPO παρατηρεί κανείς ότι και οι δύο διαγραμματικές τεχνικές μπορούν με επιτυχία να χρησιμοποιηθούν για την απεικόνιση ενός συστήματος/προγράμματος, σε διάφορα επίπεδα λεπτομέρειας, εφαρμόζοντας μια διεργασία λειτουργικής αποσύνθεσης. Η μόνη διαφορά ανάμεσά τους είναι ότι τα διαγράμματα HIPO χρησιμοποιούνται μόνο για την απεικόνιση διαδικασιών ενώ τα Warnier - Orr χρησιμοποιούνται τόσο για την απεικόνιση διαδικασιών όσο και για την απεικόνιση δεδομένων. Πάντως, τα διαγράμματα Warnier - Orr έχοντας ανώτερη διαγραμματική τεχνική και όντας κατάλληλα να αντικαταστήσουν άλλους συνδυασμούς διαγραμματικών τεχνικών (όπως για παράδειγμα διαγράμματα δομής και ψευδοκώδικα ή διαγράμματα δομής και διαγράμματα Nassi-Shneiderman) είναι προτιμότερα από τα διαγράμματα HIPO.

Στο σημείο αυτό θα πρέπει να τονιστεί το εξής. Τα διαγράμματα Warnier - Orr (όπως και τα διαγράμματα HIPO) αν χρησιμοποιηθούν για την ανάλυση των υψηλών επιπέδων ενός συστήματος/προγράμματος προσφέρουν ξεκάθαρη απεικόνιση της δομής του. Εάν, όμως χρησιμοποιηθούν για ανάλυση σε χαμηλό επίπεδο μπορεί να αυξηθεί η έκτασή τους και να μειωθεί η αναγνωσιμότητά τους. Συμπερασματικά, τα διαγράμματα Warnier Orr αν και μειονεκτούν στο ότι

μπορούν να απεικονίσουν μόνο ιεραρχικές δομές δεδομένων είναι κατάλληλα για τη σχεδίαση και την τεκμηρίωση μικρών σχετικά συστημάτων/προγραμμάτων που χρησιμοποιούν απλές δομές αρχείων και χαρακτηρίζονται από υψηλό βαθμό αλληλεπίδρασης με το περιβάλλον τους.

2.4.7 Διαγράμματα ενεργειών

Τα διαγράμματα ενεργειών αποτελούν διαγραμματική τεχνική που μπορεί να χρησιμοποιηθεί τόσο για γενική όσο και για λεπτομερή περιγραφή ενός συστήματος. Αυτό έχει ως αποτέλεσμα να είναι ιδιαίτερα κατάλληλα για την ανάπτυξη πολύπλοκων Π.Σ., αφού μπορούν να χρησιμοποιηθούν σε όλα τα στάδια της ανάλυσης και του σχεδιασμού προσφέροντας στο τέλος ένα ενιαίο διαγραμματικό περιβάλλον.

Τα βασικά χαρακτηριστικά των διαγραμμάτων ενεργειών είναι τα εξής:

- Οι αγκύλες
- Οι συνθήκες ελέγχου
- Οι υπορουτίνες
- Οι ανακυκλώσεις
- Τα σημεία τερματισμού
- Οι γλώσσες τέταρτης γενιάς
- Ο παραλληλισμός

Τα πλεονεκτήματα των διαγραμμάτων ενεργειών είναι τα εξής:

1. είναι εύκολη η εκμάθηση του τρόπου χρήσης τους.
2. προσφέρονται και για χειρογραφική χρήση και για χρήση μέσω Η.Υ.
3. παρέχουν τη δυνατότητα για γρήγορη και εύκολη σχεδίαση και εφαρμογή αλλαγών.
4. χαρακτηρίζονται από δυνατότητες αποσύνθεσης, δηλαδή επιτρέπουν εύκολα τη μεταβίβαση από τη γενική περιγραφή σε λεπτομερή περιγραφή.
5. είναι πιο ευανάγνωστα από τον ψευδοκώδικα.
6. μπορούν να χρησιμοποιηθούν σε συνδυασμό με λεξικά δεδομένων.
7. συνδυάζονται με επιτυχία με τις γλώσσες τέταρτης γενιάς.

2.4.8 Διαγράμματα Michael Jackson

Τα διαγράμματα Michael Jackson μπορούν να παραστήσουν γραφικά τόσο δομές προγραμμάτων όσο και δομές δεδομένων, δίνοντας ιδιαίτερη έμφαση στον τρόπο με τον οποίο προκύπτουν οι δομές των προγραμμάτων από τις δομές των δεδομένων. Για το λόγο αυτό, στην ανάλυση ενός προγράμματος χρησιμοποιούνται κυρίως οι δομές των δεδομένων εισόδου και εξόδου.

Τα διαγράμματα Michael Jackson χρησιμοποιούν διαγράμματα δενδρικής μορφής και γι' αυτό σε ένα βαθμό μοιάζουν με τα διαγράμματα δομής. Τα διαγράμματα δενδρικής μορφής που χρησιμοποιούνται αποτελούνται από τα εξής βασικά δομικά στοιχεία:

- Στοιχειώδης δομικό στοιχείο
- Ακολουθία
- Επανάληψη
- Επιλογή

Συμπερασματικά, ένα διάγραμμα Michael Jackson, για κάθε συγκεκριμένο πρόγραμμα που εξετάζεται, αντιστοιχεί τα δομικά στοιχεία των δεδομένων εισόδου σε δομικά στοιχεία δεδομένων εξόδου. Έτσι, διευκολύνεται σημαντικά η διαδικασία σχεδίασης των προγραμμάτων.

2.4.9 Δομημένη περιγραφή και ψευδοκώδικας

Οι περισσότερες από τις τεχνικές μοντελοποίησης που παρουσιάστηκαν στις προηγούμενες ενότητες χρησιμοποιούν ιεραρχικά διαγράμματα δενδρικής μορφής για την περιγραφή του υπό ανάλυση συστήματος ή προγράμματος. Στην ενότητα αυτή, η τεχνικές που παρουσιάζονται (δομημένη περιγραφή και ψευδοκώδικας) δε χρησιμοποιούν διαγράμματα τέτοιας μορφής. Αποτελούν μια διασκευασμένη διάλεκτο της ομιλούμενης γλώσσας που χρησιμοποιείται στη περιγραφή της δομής προγραμμάτων.

Οι διαφορές του ψευδοκώδικα και της δομημένης περιγραφής είναι ελάχιστες. Ο ψευδοκώδικας χρησιμοποιεί περισσότερο τυπικό συμβολισμό και απευθύνεται κυρίως σε ειδικούς ανάπτυξης λογισμικού. Η μορφή του μοιάζει περισσότερο με τη μορφή που θα έχει ο κώδικας που θα υλοποιήσει τη διεργασία που περιγράφεται. Ως αποτέλεσμα πολλές φορές περιλαμβάνει εντολές που αντιστοιχούν σε αντίστοιχες εντολές της γλώσσας προγραμματισμού που θα χρησιμοποιηθεί για την τελική υλοποίηση. Στο σχήμα 2. παρουσιάζεται ένα απλό παράδειγμα ψευδοκώδικα για ένα πρόγραμμα που ζητάει από το χρήστη δύο αριθμούς. Πρώτα ένα μεγάλο και στη συνέχεια ένα μικρότερο.

Ο ψευδοκώδικας παρουσιάζει τα εξής βασικά χαρακτηριστικά:

1. έχει ιεραρχική δομή.
2. έχει δομή παρόμοια με τη δομή του κώδικα προγράμματος που θα χρησιμοποιηθεί για την υλοποίησή του.
3. είναι εύκολη η κατανόησή του.
4. υποστηρίζει τη χρήση επεξηγηματικών σχολίων

Η δομημένη περιγραφή υποστηρίζει τους εξής βασικούς τύπους δομών:

- 1) ακολουθία, όπου οι ενέργειες εκτελούνται με τη σειρά που εμφανίζονται (ακολουθιακά) από πάνω προς τα κάτω.
- 2) επιλογή (if then else), όπου αν μια συνθήκη αληθεύει ,τότε μια συγκεκριμένη ενέργεια εκτελείται. Σε αντίθετη περίπτωση, εκτελείται μια διαφορετική ενέργεια.
- 3) δομή case, όπου υπάρχουν πολλές αμοιβαία αποκλειόμενες συνθήκες. Όταν μια από τις συνθήκες αυτές αληθεύει, εκτελείται μια συγκεκριμένη ενέργεια που αντιστοιχεί στησυγκεκριμένη συνθήκη.
- 4) επανάληψη (repeat while, repeat until), όπου η εκτέλεση ενός καθορισμένου συνόλου ενεργειών επαναλαμβάνεται όσο ισχύει μία συγκεκριμένη συνθήκη.

2.5 Επίλογος-συμπεράσματα

Ένα Π.Σ. διαχειρίζεται δεδομένα που αντιπροσωπεύουν αντικείμενα, γεγονότα, καταστάσεις ή ανθρώπους του φυσικού κόσμου. Όταν κατασκευάζεται ένα μοντέλο Π.Σ. χρησιμοποιώντας τις παραπάνω τεχνικές, το οποίο στη συνέχεια θα χρησιμοποιηθεί από το σχεδιαστή, πρέπει να είναι όσο το δυνατόν πιο ανεξάρτητο από την τεχνολογία υλοποίησης. Θα πρέπει δηλαδή να αντικατοπτρίζει την πραγματική εικόνα των καταστάσεων χωρίς ελλείψεις ή πλεονασμούς, ώστε να μη δημιουργούνται προβλήματα στο έργο του σχεδιαστή. Ένας τρόπος για να κατασκευαστεί ένα μοντέλο με αυτά τα χαρακτηριστικά, είναι να δοθεί περισσότερη έμφαση στη δομή και την οργάνωση των δεδομένων, δηλαδή στη διάταξη και στις σχέσεις που υπάρχουν μεταξύ τους.

Σκοπός είναι η κατασκευή ενός μοντέλου δεδομένων που θα υποστηρίζει τις επεξεργασίες χωρίς να καθοδηγείται από αυτές. Για το λόγο αυτό, προσδιορίζονται αρχικά τα στοιχειώδη δεδομένα που απαιτούνται για την υποστήριξη του συστήματος επεξεργασίας δεδομένων του ΠΣ, στη συνέχεια ομαδοποιούνται κατά ορθολογικό τρόπο και τέλος προσδιορίζονται οι σχέσεις που υπάρχουν μεταξύ των διαφόρων ομάδων που έχουν σχηματιστεί. Η περιγραφή των δεδομένων για την κατασκευή του μοντέλου, πρέπει να είναι γενική ώστε να χρησιμοποιείται από όλο τον οργανισμό καλύπτοντας και μελλοντικές ανάγκες για απρόσκοπτες αλλαγές, αλλά και αναλυτική ώστε να ικανοποιεί τις τρέχουσες απαιτήσεις των επεξεργασιών του συστήματος. Έτσι θα είναι σε θέση να αντιμετωπίσει με επιτυχία το βασικότερο πρόβλημα που εμφανίζεται μετά τη δημιουργία των προγραμμάτων λογισμικού: τις αλλαγές που οι περισσότεροι χρήστες επιθυμούν να συμπεριληφθούν στα ήδη ολοκληρωμένα προγράμματα.

Επειδή η δομή δεδομένων μιας επιχείρησης είναι περισσότερο σταθερή από την οργανωτική δομή της επιχείρησης ή τις απαιτήσεις των χρηστών, ένα μοντέλο το οποίο στηρίζεται σ' αυτή εμφανίζει πολλά πλεονεκτήματα:

- Ø Οδηγεί σε καλά σχεδιασμένα και ευέλικτα συστήματα.
- Ø Μπορεί να ικανοποιήσει τόσο τις παρούσες όσο και τις μελλοντικές ανάγκες

των χρηστών.

- Ø Δημιουργείται από πληροφορίες που δίνουν άμεσα οι ίδιοι οι χρήστες.
- Ø Επιτρέπει τη δημιουργία δεδομένων που μπορούν να τα μοιράζονται πολλοί χρήστες, ενώ παράλληλα υποστηρίζει την ακεραιότητα και την ασφάλειά τους.
- Ø Οι χρήστες μπορούν ευκολότερα να αντιληφθούν, να τροποποιήσουν και, τελικά, να επικυρώσουν ένα μοντέλο που στηρίζεται στη δομή των δεδομένων της επιχείρησης παρά σε άλλης μορφής μοντελοποίηση.
- Ø Εξασφαλίζει ότι δεν θα υπάρχουν πολλαπλά αντίγραφα των δεδομένων σε διαφορετικά σημεία του συστήματος.

Κάποιος θα μπορούσε να ισχυριστεί ότι θα αρκούσε η χρήση των *Διαγραμμάτων Ροής Δεδομένων* (ΔΡΔ) για την ικανοποίηση των παραπάνω ιδιοτήτων. Η απάντηση είναι ότι τα ΔΡΔ δείχνουν μόνο τη ροή δεδομένων χωρίς να περιγράφουν τις σχέσεις μεταξύ των δεδομένων. Επίσης, δεν εξασφαλίζουν ότι για παράδειγμα, δε θα υπάρχουν αντίγραφα δεδομένων σε διαφορετικά σημεία.

Το μοντέλο δεδομένων, που πρέπει να κατασκευαστεί πρέπει είναι ένα ορθολογικό μοντέλο, δηλαδή ένα μοντέλο που να δίνει μια λογική εικόνα της πραγματικότητας και να απεικονίζει την ουσία της. Ένα τέτοιο μοντέλο μπορεί να κατασκευαστεί από τη συγχώνευση δυο μοντέλων, πχ:

- Ø Του μοντέλου *οντοτήτων - σχέσεων*.
- Ø Του μοντέλου που προέρχεται από τη *σχεσιακή ανάλυση δεδομένων*.

Δεν θα αναλύσουμε περισσότερο το παραπάνω απλά το αναφέρουμε ως συμπέρασμα του κεφαλαίου των διαγραμματικών τεχνικών.

3^ο ΚΕΦΑΛΑΙΟ

ΕΝΙΑΙΟΣ ΤΡΟΠΟΣ ΜΟΝΤΕΛΟΠΟΙΗΣΗΣ (UML)

3.1 Εισαγωγή στη UML

Στο κεφάλαιο 3 δίνονται οι βασικές έννοιες της αντικειμενοστρεφούς φιλοσοφίας με έμφαση στην αποκάλυψη των στοιχείων ενός διαφορετικού από τη δομημένη προσέγγιση τρόπου σκέψης. Επίσης, εισάγεται η UML, ένα πρότυπο για την παράσταση συστατικών στοιχείων λογισμικού, η έλλειψη του οποίου ταλαιπώρησε επί μακρόν την κοινότητα των κατασκευαστών λογισμικού και τους αποξένωνε μεταξύ τους. Ο αναγνώστης με βασική εμπειρία στην ανάπτυξη λογισμικού, θα συμφωνήσει με τη θέση ότι, παρά την απλότητα και το «αυτονόητο» της ανάγκης για χρήση κοινών συμβολισμών από όλους τους κατασκευαστές λογισμικού, κάτι τέτοιο δεν επετεύχθει για μεγάλο χρονικό διάστημα, οπότε αποτελεί ουσιώδες βήμα προόδου από μόνο του, η εισαγωγή της UML.

Είναι φυσικό ότι ο τρόπος με τον οποίο σχεδιάζουμε ένα σύστημα, είναι άρρηκτα συνδεδεμένος με την τεχνολογία που θα χρησιμοποιήσουμε για να το αναπτύξουμε. Έτσι από τα διαγράμματα ροής προγράμματος, περάσαμε με την εμφάνιση του δομημένου προγραμματισμού σε πιο προχωρημένες τεχνικές παράστασης και σχεδίασης συστημάτων που περιελάμβαναν πιο σύνθετες διαγραμματικές τεχνικές για τον έλεγχο των πιο πολύπλοκων σύγχρονων συστημάτων. Αυτές οι τεχνικές περιελάμβαναν λεξικά δεδομένων, διαγράμματα ροής δεδομένων, πίνακες αποφάσεων κλπ.

Σήμερα τα συστήματα που αναπτύσσονται είναι σχεδόν στο σύνολό τους αντικειμενοστραφή. Και υπάρχουν καλοί λόγοι γι' αυτό:

- Το αντικειμενοστραφές λογισμικό είναι ευκολότερο στην αρχική του σύλληψη μια και τα αντικείμενα είναι – εν μέρει – οντότητες του υπαρκτού κόσμου του χώρου ανάπτυξης του λογισμικού (π.χ. σε ένα τραπεζικό σύστημα θα υπάρχουν αντικείμενα όπως «Τραπεζικός Λογαριασμός», «Πελάτης» κλπ.).
- Το αντικειμενοστραφές λογισμικό είναι ευκολότερο στην εξέλιξή του. Ιδιαίτερα με την χρήση των διασυνδέσεων (interfaces) δηλαδή με την χρήση τύπων σε αντιδιαστολή με την χρήση τάξεων (classes) δηλαδή υλοποίησης, μπορεί πράγματι να επιτρέψει την εξέλιξη του λογισμικού με πρωτοφανή ευκολία για την βιομηχανία παραγωγής λογισμικού. Φυσικά αυτό δεν επιτυγχάνεται αυτόματα με την απλή χρήση των διασυνδέσεων, αλλά αυτές είναι μια καλή αρχή.

Τα αντικείμενα αποτελούν την βάση για την κατασκευή των συστατικών (components) που αποτελούν τη βάση για το μεγάλο όραμα της βιομηχανίας λογισμικού μια και θεωρητικά στο μέλλον θα μπορούμε να κατασκευάσουμε προγράμματα με την σύνθεση συστατικών προερχόμενα από διάφορες πηγές εκφράζοντας απλά τις επιθυμίες μας. Υπάρχουν μεγάλα ανοικτά θέματα τα οποία όμως χωρίς αμφιβολία θα λυθούν στα επόμενα χρόνια. Τα συστήματα λογισμικού βασισμένα στα συστατικά είναι χωρίς αμφιβολία το μέλλον της πληροφορικής.

Θα μπορούσαμε να συνεχίσουμε την απαρίθμηση των πλεονεκτημάτων των αντικειμενοστραφών συστημάτων για πολύ ακόμα. Προκύπτει λοιπόν το άμεσο ερώτημα πως σχεδιάζουμε αντικειμενοστραφή συστήματα, πως μοντελοποιούμε αντικειμενοστραφή συστήματα και φυσικά πως προγραμματίζουμε αντικειμενοστραφή συστήματα. Στο κεφάλαιο αυτό θα προσπαθήσουμε να απαντήσουμε σε αυτά τα ερωτήματα μέσα από παραδείγματα μοντελοποίησης και προγραμματισμού δίνοντας έμφαση στη γλώσσα σχεδιασμού uml.

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.

3.2 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΗΣ UML

- Είναι ιδιαίτερα εκφραστική.
- Σημασιολογικά, είναι εκτενής.
 - Ø Υποστηρίζει την σημασιολογία τύπων και μοντέλων για όλα τα μοντέλα ενός συστήματος.
- Σε επίπεδο βασικών αρχών είναι “μικρή” και απλή.
 - Ø Διακρίνουμε πέντε βασικούς άξονες.
- Είναι επεκτάσιμη.
 - Ø Υπάρχει η δυνατότητα του εμπλουτισμού του μετά-μοντέλου με κλάσεις, ιδιότητες και σημασιολογία.

- Είναι επακριβώς ορισμένη με βάση τα δομικά συστατικά ενός αντικειμενοστρεφούς συστήματος.
- Ενσωματώνει τις ιδέες “καλής πρακτικής” από τη βιομηχανία Λογισμικού.
- Είναι σήμερα βιομηχανικό πρότυπο και αναπτύχθηκε στο πλαίσιο του οργανισμού OMG.
- Υλοποιεί την ανάγκη της βιομηχανίας Λογισμικού για μια ενιαία γλώσσα μοντελοποίησης.
- Αντιμετώπιση σημερινών και βραχυπρόθεσμων προβλημάτων στην ανάπτυξη Λογισμικού:

- Ø Κλίμακα
- Ø Γλώσσες: Java, C++, Smalltalk, Ada, Visual Basic
- Ø Πολυεπεξεργασία και παραλληλία
- Ø Πρότυπα Λογισμικού (Patterns)
- Ø Ψηφίδες λογισμικού (Componentware)
- Ø Μοντελοποίηση επιχειρησιακής πρακτικής

Η UML περιλαμβάνει τρία βασικά στοιχεία: τις Οντότητες, τις Σχέσεις και τα Διαγράμματα, τα οποία θα αναλύσουμε όλα παρακάτω.

3.2.2 Βασικά στοιχεία της uml

α. Τι είναι η Οντότητα

Μέσα στα πλαίσια μιας επιχείρησης ή οργανισμού - είναι μία κατηγορία ομοειδών αντικειμένων που έχουν κοινές ιδιότητες και διατηρούν ένα μοναδικό και με ακρίβεια προσδιορισμένο ρόλο στην επιχείρηση ή στον οργανισμό. Αυτό έχει ως αποτέλεσμα το σύστημα να ενδιαφέρεται και να πρέπει να κρατάει δεδομένα για αυτά, ώστε να έχει τη δυνατότητα αναζήτησης και εντοπισμού τους. Από τα παραπάνω προκύπτει ότι μία οντότητα αποτελεί ένα είδος «κουτιού», στο οποίο εισάγονται μόνο ομοειδή αντικείμενα και άρα θα πρέπει να έχει τουλάχιστον μια ιδιότητα.

Οντότητες μπορούν να είναι:

Άνθρωποι που εκτελούν κάποια λειτουργία (π.χ. Εργαζόμενος, Πελάτης, Φοιτητής, κλπ)

Πράγματα ή υλικά (π.χ. Αυτοκίνητο, Οικοδομήματα, Εργαλεία, κλπ)

Γεγονότα που γίνονται- κάποια συγκεκριμένη στιγμή (π.χ. Συναλλαγή, Δανεισμός, Πληρωμή, κλπ)

Ομάδα αντικειμένων (π.χ. Εμπόρευμα, Στόλος, κλπ)

Τοποθεσίες, που χρησιμοποιούνται από ανθρώπους ή πράγματα(π.χ. Πόλη, Γραφείο, Αποθήκη, κλπ)

Οργανισμοί που έχουν κάποια αποστολή (π.χ. Διεύθυνση, Τμήμα, Παράρτημα, κλπ)

Έννοιες ή ιδέες για τις οποίες η επιχείρηση κρατάει στοιχεία (π.χ. Αίτημα, Συμφωνία, Λογαριασμός, Δρομολόγιο, κλπ)

Στη συνέχεια θα γίνει μια αναφορά στα βασικά χαρακτηριστικά και τις ιδιότητες που πρέπει να έχει μία οντότητα. Τα σημεία αυτά πρέπει να ακολουθούνται όταν κάποιος προσπαθεί να καθορίσει και να περιγράψει τις οντότητες ενός Π.Σ.

α Επειδή μια οντότητα φανερώνει γεγονότα, πρόσωπα ή πράγματα (συγκεκριμένα ή αφηρημένα), το όνομα της είναι ουσιαστικό (συνήθως στο πρώτο ενικό πρόσωπο).

α Θα πρέπει να γίνεται διάκριση μεταξύ της - στιγμιαίας εμφάνισης ενός συγκεκριμένου αντικειμένου της κατηγορίας (στιγμιότυπο) και του γενικού τύπου της που απλά φέρει το όνομα της οντότητας. Για παράδειγμα, ο Ζαφειρόπουλος Νώντας αποτελεί μία συγκεκριμένη εμφάνιση της οντότητας ΠΕΛΑΤΗΣ, ενώ ο όρος ΠΕΛΑΤΗΣ αντιπροσωπεύει τον τύπο της οντότητας.

α Μια οντότητα ενός συστήματος δεν σημαίνει ότι αναγκαστικά είναι οντότητα και σε κάποιο άλλο. Μια οντότητα σε ένα σύστημα μπορεί να είναι χαρακτηριστική ιδιότητα οντότητας σε ένα άλλο.

α Επίσης σχετίζεται με δεδομένα αλλά η ίδια δεν αποτελεί δεδομένα. Για παράδειγμα στην οντότητα ΦΟΙΤΗΤΗΣ δεν μπορούμε να δώσουμε τιμή. Έχει όμως χαρακτηριστικές ιδιότητες (π.χ. ΟΝΟΜΑ, ΕΠΩΝΥΜΟ) τα οποία μπορούν να πάρουν κάποια τιμή.

α Επειδή η οντότητα αντιπροσωπεύει πολλά ομοειδή αντικείμενα, κάθε συγκεκριμένη εμφάνιση ενός αντικειμένου πρέπει να μπορεί να προσδιοριστεί μονοσήμαντα. π.χ. Ένα συγκεκριμένο ΒΙΒΛΙΟ μπορεί να προσδιοριστεί

μονοσήμαντα με την βοήθεια του Αριθμού ISBN.

α Κάθε οντότητα πρέπει να έχει συγκεκριμένο και μοναδικό ρόλο στο σύστημα.

β. Τι είναι οι χαρακτηριστικές ιδιότητες (γνωρίσματα)

Χαρακτηριστική ιδιότητα είναι ένα γνώρισμα που χαρακτηρίζει μια οντότητα και εμφανίζεται ως στοιχειώδες δεδομένο που τη συνοδεύει (π.χ. για την οντότητα ΦΟΙΤΗΤΗΣ τα Όνομα, Επώνυμο, ΑΜ, Διεύθυνση, Τηλέφωνο αποτελούν χαρακτηριστικές ιδιότητές της). Η οντότητα είναι λοιπόν ένα σύνολο χαρακτηριστικών ιδιοτήτων (γνωρισμάτων). Οι τιμές που παίρνουν κάθε φορά τα γνωρίσματα αυτά επιτρέπουν το διαχωρισμό μιας συγκεκριμένης οντότητας από τις άλλες που έχουν τον ίδιο τύπο με αυτή. Στη συνέχεια θα γίνει μια αναφορά στα βασικά χαρακτηριστικά και τις ιδιότητες που πρέπει να έχει μία χαρακτηριστική ιδιότητα. Τα σημεία αυτά πρέπει να ακολουθούνται όταν κάποιος προσπαθεί να καθορίσει και να περιγράψει τις χαρακτηριστικές ιδιότητες των οντοτήτων ενός Π.Σ.

- α Οι ιδιότητες μιας οντότητας είναι συνήθως επίθετα δηλαδή είναι λέξεις που σημαίνουν ποιότητα, γνώρισμα ή ιδιότητα ενός ουσιαστικού.
- α Το όνομα κάθε γνωρίσματος πρέπει να είναι μοναδικό μέσα στο σύνολο των γνωρισμάτων της οντότητας. Μπορεί όμως να εμφανίζεται ως γνώρισμα σε κάποια άλλη οντότητα του Π.Σ.
- α Ένα γνώρισμα μιας οντότητας δεν μπορεί να πάρει πάνω από μία τιμή για συγκεκριμένη εμφάνιση της οντότητας.
- α Όλα τα γνωρίσματα που είναι σημαντικά για την πλήρη περιγραφή και την κατανόηση κάθε οντότητας θα πρέπει να συμπεριλαμβάνονται σε αυτή.
- α Η διαφορά μεταξύ χαρακτήρων γνωρισμάτων και οντότητας είναι ότι τα γνωρίσματα είναι ατομικά και δεν απαιτούν επιπλέον ιδιότητες για να

περιγραφούν, ενώ οι οντότητες απαιτούν γνωρίσματα για την περιγραφή τους.

- α Οι τιμές των ιδιοτήτων δεν αλλάζουν τον ρόλο της οντότητας στο σύστημα.

γ. Τι είναι οι Σχέσεις

Σχέση είναι η σύνδεση των οντοτήτων και η μέσω αυτής απεικόνιση του ρόλου τους σε μια επιχείρηση / οργανισμό. Κατά συνέπεια η ύπαρξη μιας σχέσης μεταξύ δύο οντοτήτων του συστήματος, θα πρέπει να αποδεικνύεται από την υπάρχουσα ή τη δημιουργούμενη νέα πολιτική της επιχείρησης / οργανισμού.

Στη συνέχεια θα γίνει μια αναφορά στα βασικά χαρακτηριστικά και τις ιδιότητες που πρέπει να έχει μία σχέση. Τα σημεία αυτά πρέπει να ακολουθούνται όταν κάποιος προσπαθεί να καθορίσει και να περιγράψει τις σχέσεις μεταξύ των οντοτήτων ενός Π.

Η συσχέτιση αντιπροσωπεύει κάποια άμεση αλληλοεπίδραση μεταξύ δύο οντοτήτων του φυσικού κόσμου (της επιχείρησης) και μπορεί να περιγραφεί με μια απλή πρόταση :

υποκείμενο (οντότητα) - ρήμα (σχέση) - αντικείμενο (οντότητα)

Π.χ. Ο ΣΠΟΥΔΑΣΤΗΣ παρακολουθεί το ΜΑΘΗΜΑ

Το ΜΑΘΗΜΑ παρακολουθείται από το...

Επειδή οι σχέσεις είναι αμφίδρομες χρησιμοποιούνται δύο προτάσεις με το ρήμα σε ενεργητική και παθητική φωνή αντίστοιχα. Μερικές σχέσεις έχουν τις δικές τους ιδιότητες οι οποίες είναι σημαντικές γιατί χαρακτηρίζουν την κάθε σχέση και πρέπει να αποθηκεύονται. Οι ιδιότητες αυτές χαρακτηρίζουν τη σχέση και όχι τις οντότητες που συμμετέχουν σε αυτή.

Μας ενδιαφέρουν μόνο οι άμεσες και σημαντικές για το σύστημα σχέσεις.

Στη UML ορίζονται τρεις βασικές σχέσεις:

1. εξάρτηση (*dependency*)
2. γενίκευση (*generalisation*)
3. σύνδεση (*association*)

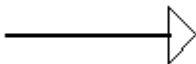
Εξάρτηση

Η εξάρτηση δηλώνει πως μια αλλαγή σε μίαν οντότητα θα επηρεάσει μίαν άλλη αλλά όχι απαραίτητα και το αντίστροφο. Παριστάνεται με μία διακεκομμένη γραμμή με ανοιχτό βέλος που δείχνει προς την οντότητα που υπάρχει εξάρτηση:



Γενίκευση

Η γενίκευση δηλώνει μία σχέση ανάμεσα σε κάτι γενικό (τη βασική κλάση ή αλλιώς γονέα) και κάτι ειδικό (μίαν υποκλάση ή αλλιώς παιδί της). Παριστάνεται με μία συνεχή γραμμή με κλειστό βέλος που δείχνει προς τη βασική κλάση:

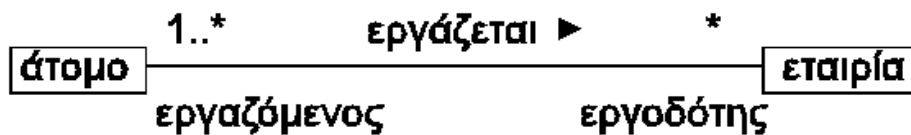


Σύνδεση

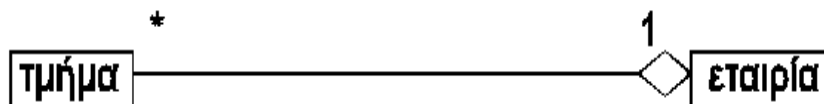
Η σύνδεση αναφέρεται σε αντικείμενα τα οποία συνδέονται με κάποιο τρόπο με άλλα. Όταν δύο κλάσεις είναι συνδεδεμένες μπορεί κανείς να μεταβεί από αντικείμενα της μίας σε αντικείμενα της άλλης. Η σύνδεση παριστάνεται με μία ευθεία γραμμή ανάμεσα στα δύο αντικείμενα.

- Αν η σύνδεση δεν είναι αμφίδρομη τότε η κατεύθυνσή της μπορεί να οριστεί με ένα ανοιχτό βέλος.

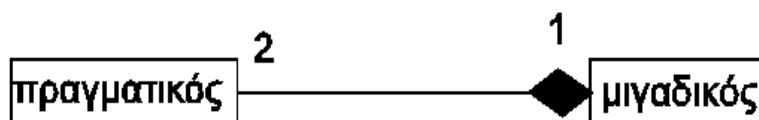
- Το όνομα της σύνδεσης μπορεί να γραφεί πάνω από τη γραμμή, ενώ η κατεύθυνση του ονόματος ορίζεται από ένα βέλος πλάι στο όνομα.
- Ο ρόλος των οντοτήτων που συνδέονται προσδιορίζεται από ένα όνομα στην κάθε άκρη της γραμμής.
- Ο αριθμός που δηλώνει πόσα αντικείμενα αντιστοιχούν σε κάθε αντικείμενο στην άλλη άκρη της σχέσης (πολλαπλότητα (*multiplicity*)) δηλώνεται από έναν αριθμό (π.χ. 3), ή μια περιοχή αριθμών (π.χ. 1..* για ένα έως πολλά) πάνω από την αντίστοιχη άκρη της γραμμής.



Αν σε μια σχέση τα αντικείμενα απαρτίζουν τμήματα ενός όλου, τότε αυτή απεικονίζεται ως συγκρότημα (*aggregation*) με την παράσταση ενός διαμαντιού στην άκρη του "όλου".



Αν σχέση τα αντικείμενα που απαρτίζουν τμήματα ενός όλου έχουν την ίδια διάρκεια ζωής με το όλο, τότε αυτή απεικονίζεται ως σύνθεση (*composition*) με την παράσταση ενός γεμάτου διαμαντιού στην άκρη του "όλου".



3.2.3 ΔΙΑΓΡΑΜΜΑΤΑ UML

Η UML ορίζει τα παρακάτω διαγράμματα:

- Διαγράμματα συμπεριφοράς
 - Διαγράμματα περιπτώσεων χρήσης (*use case diagrams*)
 - Διαγράμματα καταστάσεων (*state machine diagrams*)
 - Διαγράμματα δραστηριοτήτων (*activity diagrams*)
 - Διαγράμματα επικοινωνίας (*communication diagrams*)
 - Διαγράμματα ακολουθίας (*sequence diagrams*)

- Δομικά διαγράμματα
 - Διαγράμματα κλάσεων (*class diagrams*)
 - Διαγράμματα αντικειμένων (*object diagrams*)
 - Διαγράμματα πακέτων (*package diagrams*)

- Αρχιτεκτονικά διαγράμματα
 - Διαγράμματα εξαρτημάτων (*component diagrams*)
 - Διαγράμματα ανάπτυξης (*deployment diagrams*)
 - Διαγράμματα συστατικών (*component diagrams*)
 - Διαγράμματα συνεργασίας (*collaboration diagrams*)

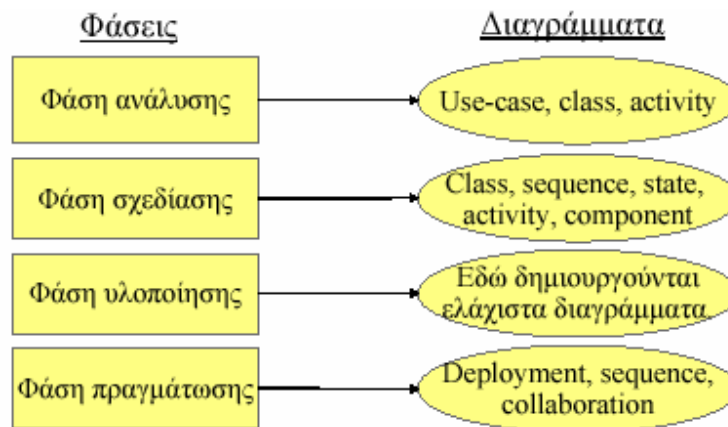
3.2.3.1 Χρήση των Διαγραμμάτων UML στις Δραστηριότητες Ανάπτυξης Λογισμικού

Τα διαγράμματα της UML χρησιμοποιούνται στις δραστηριότητες ανάπτυξης λογισμικού ως εξής:

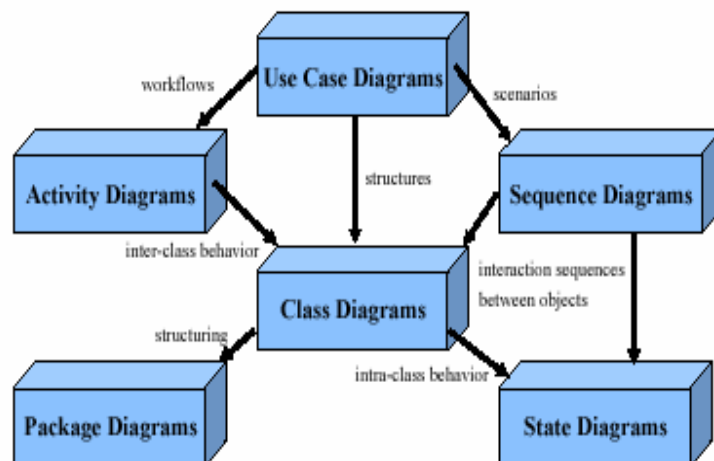
- Για την Ανάλυση Απαιτήσεων χρησιμοποιούνται:
 - Διαγράμματα περιπτώσεων χρήσης για την καταγραφή των απαιτήσεων σε σενάρια και των χρηστών σε ρόλους.
 - Διαγράμματα κλάσεων για την περιγραφή, σε υψηλό επίπεδο αφαίρεσης, των οντοτήτων του συστήματος και των σχέσεων μεταξύ τους.
 - Διαγράμματα δραστηριότητας για την απεικόνιση της ροής εργασίας στον οργανισμό.
 - Διαγράμματα κατάστασης για την καταγραφή των γεγονότων στον οργανισμό.
- Για τον Σχεδιασμό του συστήματος χρησιμοποιούνται
 - Διαγράμματα δραστηριοτήτων για την καταγραφή των επιχειρηματικών δραστηριοτήτων.
 - Διαγράμματα ανάπτυξης για την περιγραφή της φυσικής υποδομής του συστήματος σε υψηλό επίπεδο αφαίρεσης.
 - Διαγράμματα κλάσεων για την καταγραφή των κλάσεων αντικειμένων που απαρτίζουν το σύστημα.
 - Διαγράμματα ακολουθίας για την καταγραφή της συνεργασίας αντικειμένων των κλάσεων για την πραγματοποίηση ενός σεναρίου χρήσης.
 - Διαγράμματα καταστάσεων για την περιγραφή της συμπεριφοράς ενός αντικειμένου κατά την υλοποίηση συγκεκριμένων σεναρίων περιπτώσεων χρήσης.
- Και για την Υλοποίηση

- Διαγράμματα κλάσεων για την περιγραφή των κλάσεων αντικειμένων που απαρτίζουν το σύστημα σε λεπτομέρεια που να καθοδηγεί την κωδικοποίηση.
- Διαγράμματα αντικειμένων για την ανάλυση του συστήματος σε αντικείμενα σε μία δεδομένη χρονική στιγμή.
- Διαγράμματα ακολουθίας για την καταγραφή της υλοποίησης της συνεργασίας αντικειμένων των κλάσεων για την πραγματοποίηση ενός σεναρίου χρήσης.
- Διαγράμματα καταστάσεων για την περιγραφή της συμπεριφοράς ενός αντικειμένου, σε επίπεδο καταστάσεων, γεγονότων και μηνυμάτων, στην υλοποίηση συγκεκριμένων σεναρίων χρήσης.
- Διαγράμματα εξαρτημάτων για την ανάλυση του συστήματος σε εξαρτήματα, διεπαφές, πηγαίο κώδικα, βιβλιοθήκες, αρχεία, και όποιες άλλες κατασκευές απαρτίζουν την υλοποίηση.

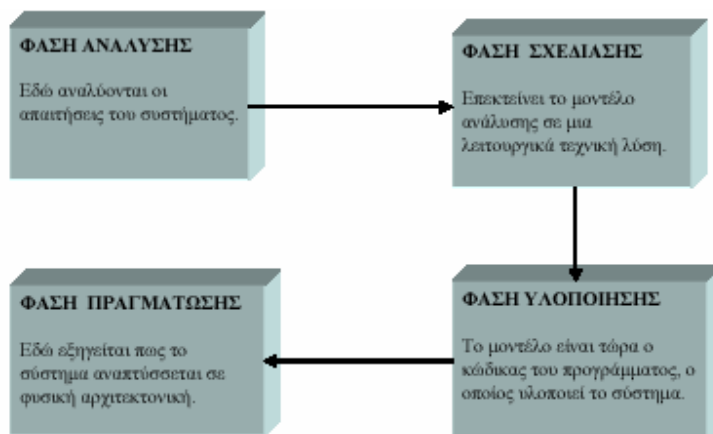
Αντιστοιχία διαγραμμάτων στις διάφορες φάσεις ανάπτυξης του συστήματος



Σχέσεις διαγραμμάτων uml



Μοντελοποιώντας με την uml



Τώρα λοιπόν φτάσαμε στο σημείο στο οποίο θα αναλύσουμε τα διαγράμματα της uml και θα δώσουμε ιδιαίτερη βάση σ' αυτά που μας ενδιαφέρουν περισσότερο.

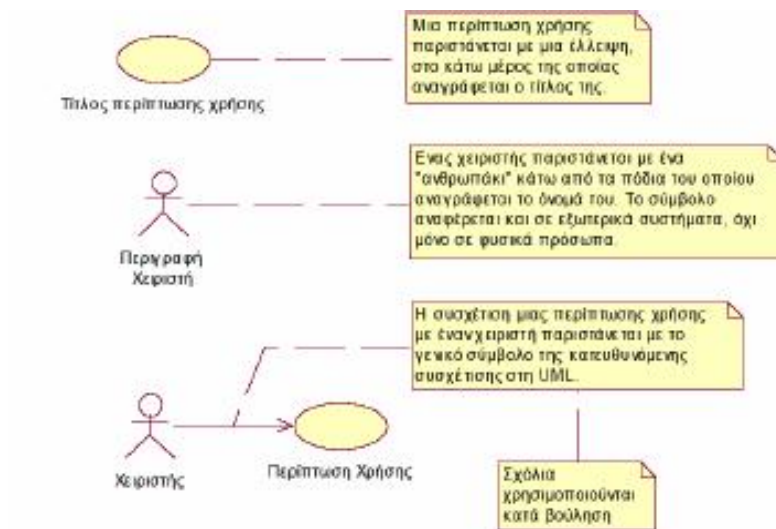
3.2.3.2 Διαγράμματα Περιπτώσεων Χρήσεων (use case diagrams)

Τα διαγράμματα περιπτώσεων χρήσεων περιγράφουν τις λειτουργικές απαιτήσεις του συστήματος.

- Μία περίπτωση χρήσης είναι ένα σύνολο σεναρίων με τον ίδιο στόχο.
- Μία περίπτωση χρήσης είναι ένα σύνολο σεναρίων που αποσκοπούν στην εκπλήρωση ενός στόχου ενός χρήστη.
- Οι χρήστες ονομάζονται ερμηνευτές ή δρώντες (*actors*).
 - Οι χρήστες δεν είναι απαραίτητα άνθρωποι.
 - Ένας άνθρωπος μπορεί να εμφανίζεται ως ερμηνευτής σε διάφορες περιπτώσεις χρήσης με διαφορετικό ρόλο κάθε φορά.

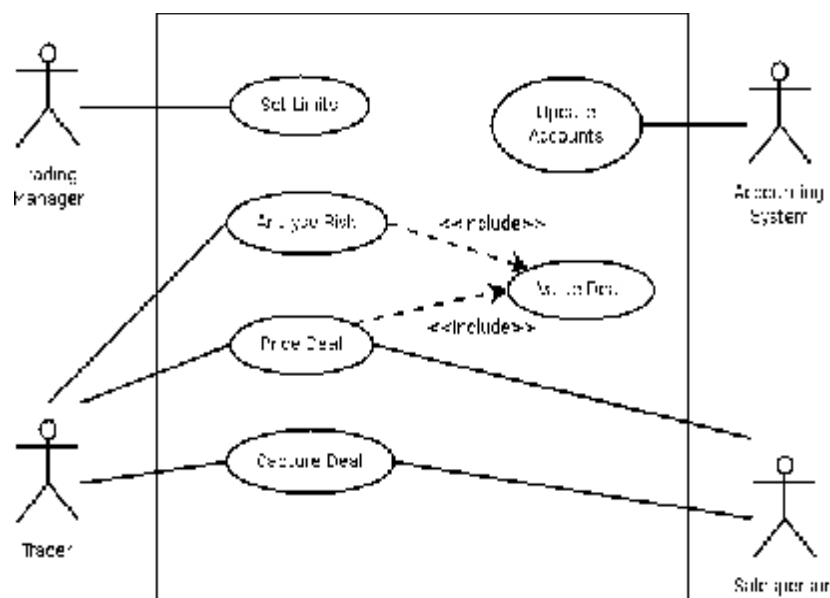
Μία περίπτωση χρήσης μπορεί να περιλαμβάνει μια άλλη (*includes*), ή και να επεκτείνει μια άλλη (*extends*)

Παράδειγμα α.



σχήμα 1:διάγραμμα περίπτωσης χρήσης (use case diagram) που δείχνει απλά τα σύμβολα που χρησιμοποιούνται για την κατασκευή του διαγράμματος.

Παράδειγμα β.



σχήμα 2:Διάγραμμα περιπτώσεων χρήσης

Περιγραφή Περίπτωσης Χρήσης

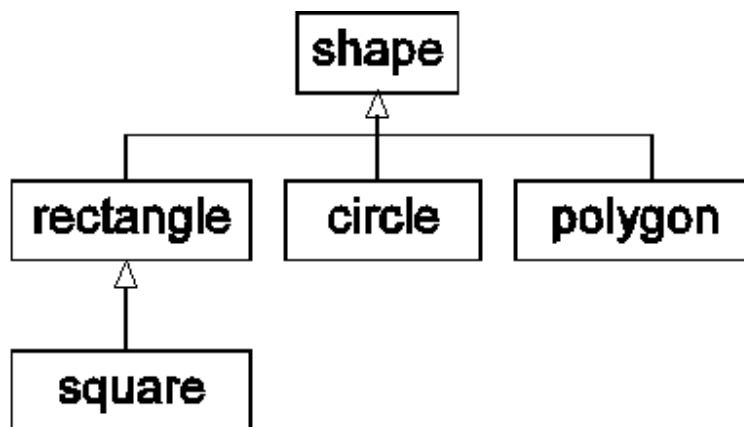
1. Ο ενδιαφερόμενος εισάγει τον δρόμο και τον αριθμό.
2. Το σύστημα παρουσιάζει τις περιοχές στις οποίες υπάρχουν αντίστοιχες διευθύνσεις.
3. Ο ενδιαφερόμενος επιλέγει μια περιοχή.
4. Το σύστημα παρουσιάζει χάρτη της περιοχής με σημειωμένη τη διεύθυνση που ζήτησε ο ενδιαφερόμενος.

Εναλλακτικά σενάρια

- 1.2α Το σύστημα δεν εντόπισε καμία περιοχή με αντίστοιχη διεύθυνση.
 - 1.2α.1 Το σύστημα εμφανίζει άλλες οδούς με παραπλήσια ονόματα.
 - 1.2α.2 Ο ενδιαφερόμενος επιλέγει μία από τις άλλες οδούς ή ακυρώνει τη διαδικασία.
- 1.3α Ο ενδιαφερόμενος δεν επιλέγει καμία από τις περιοχές
 - 1.3α.1 Ο ενδιαφερόμενος επιστρέφει στην αρχή της περίπτωσης χρήσης. Η σχέση *includes* μπορεί να αναπαρασταθεί στην περιγραφή με την κατάλληλη παραπομπή.
 - Η σχέση *extends* μπορεί να αναπαρασταθεί συμπεριλαμβάνοντας στις επικεφαλίδες αναφορά στην περίπτωση χρήσης που επεκτείνεται.

3.2.3.3 ΔΙΑΓΡΑΜΜΑΤΑ ΚΛΑΣΕΩΝ

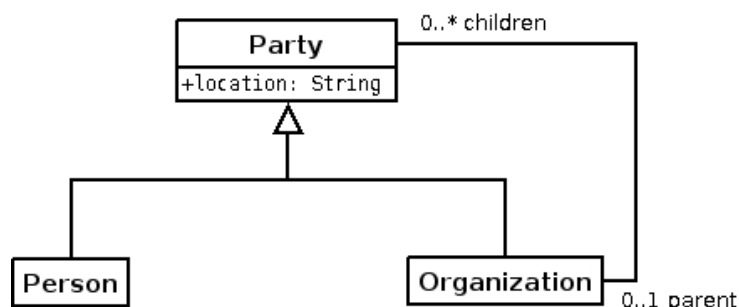
Το διάγραμμα των κλάσεων ενός συστήματος περιέχει τις κλάσεις μαζί με του αντίστοιχους δεσμούς εξάρτησης, γενίκευσης και σύνδεσης. Έτσι ένα διάγραμμα κλάσεων μπορεί να απεικονίσει τη χρήση της κληρονομικότητας στο σχεδιασμό με τη χρήση δεσμών γενίκευσης όπως στο παρακάτω σχήμα:



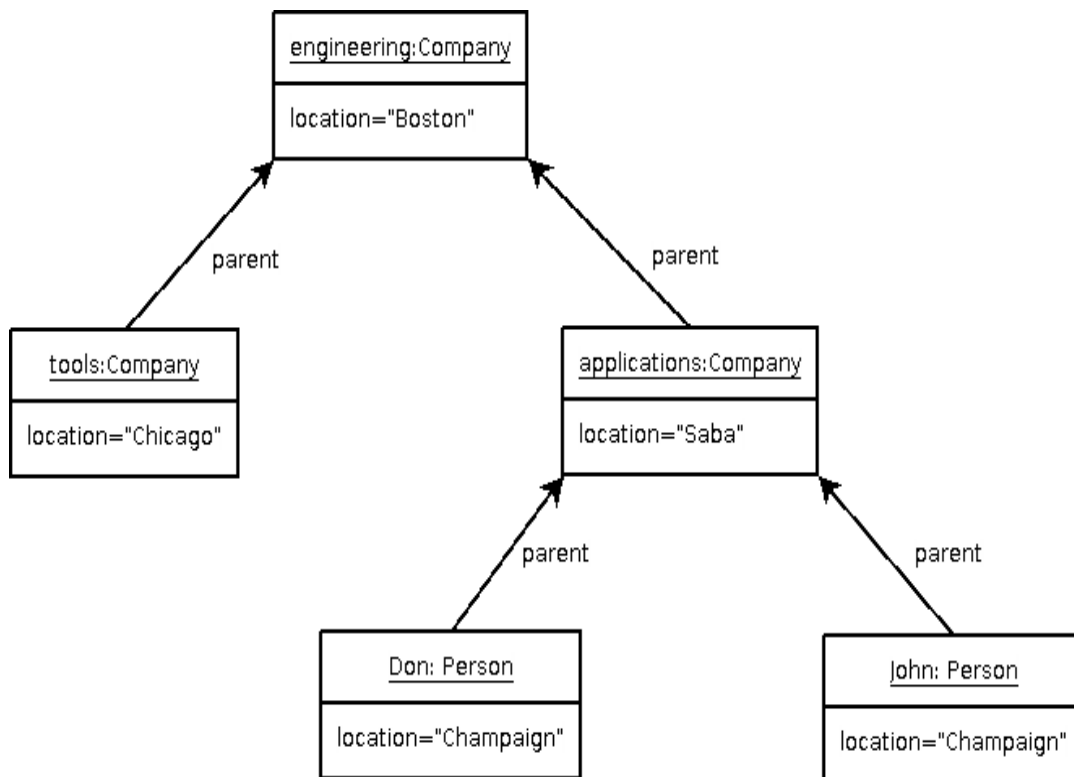
3.2.3.4 ΔΙΑΓΡΑΜΜΑΤΑ ΑΝΤΙΚΕΙΜΕΝΩΝ

Τα διαγράμματα αντικειμένων χρησιμοποιούνται για το σχεδιασμό της στατικής κατάστασης του συστήματος κατά μια συγκεκριμένη χρονική στιγμή. Κάθε αντικείμενο σχεδιάζεται ως ένα ορθογώνιο με την παρακάτω μορφή:

ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ



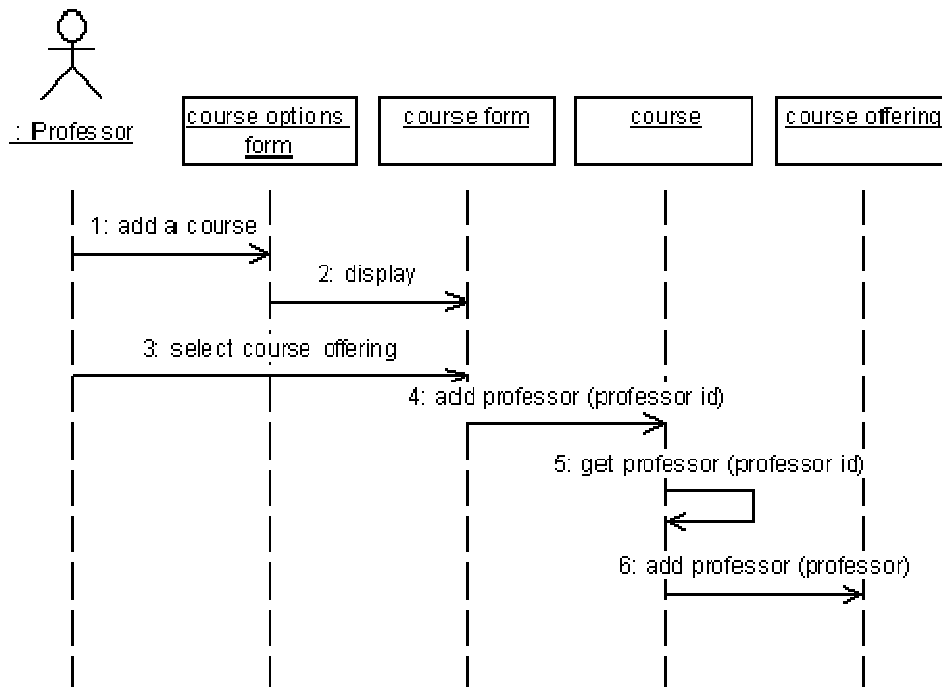
Διάγραμμα Κλάσεων (Fowler, 2004)



Διάγραμμα Αντικειμένων (Fowler, 2004)

3.2.3.5 ΔΙΑΓΡΑΜΜΑΤΑ ΑΚΟΛΟΥΘΙΑΣ

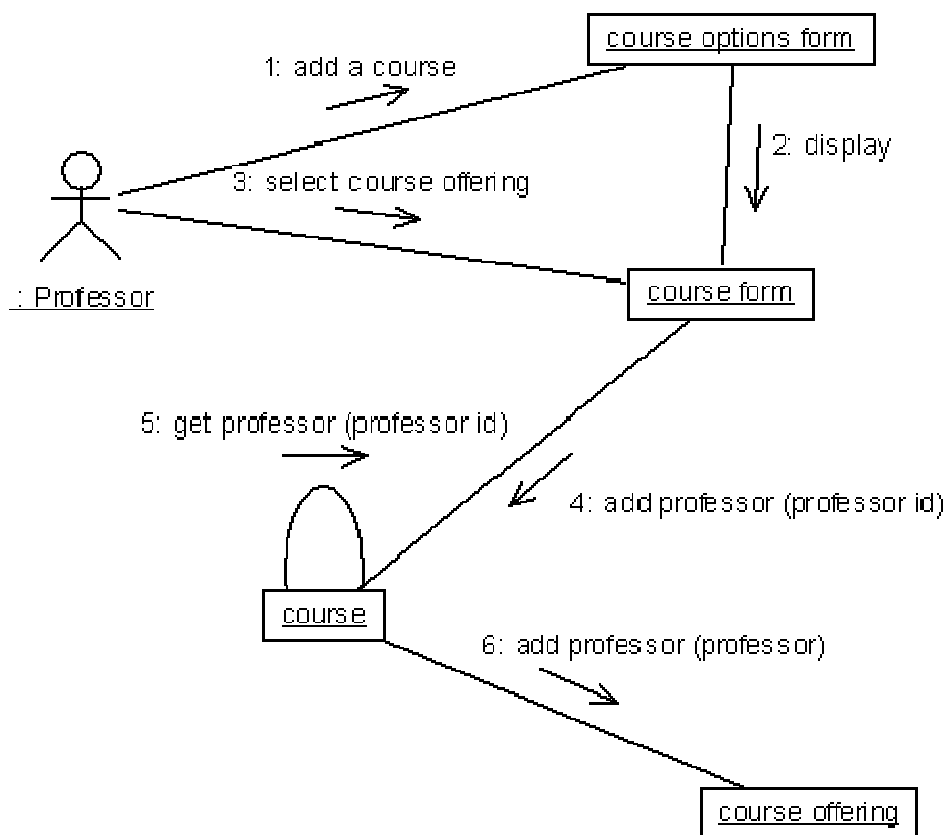
Τα διαγράμματα ακολουθίας δείχνουν πώς συνεργάζεται ένα σύνολο οντοτήτων για την επίτευξη μιας συγκεκριμένης συμπεριφοράς. Η συμπεριφορά που περιγράφεται σε ένα διάγραμμα ακολουθίας αντιστοιχεί συνήθως σε ένα απλό σενάριο χρήσης. Στα **διαγράμματα ακολουθίας** (sequence) τα αντικείμενα παριστάνονται με κουτιά ή/και ρόλους τα οποία παρατάσσονται στο χώρο (συνήθως από αριστερά προς τα δεξιά). Τα αντικείμενα ανταλλάσσουν μηνύματα τα οποία παριστάνονται με κατευθυνόμενα βέλη των οποίων η φορά δείχνει την κατεύθυνση του μηνύματος (ποιο αντικείμενο κάνει την κλήση και σε ποιο αντικείμενο γίνεται η κλήση). Ο χρόνος παριστάνεται από πάνω προς τα κάτω στο διάγραμμα. Δηλαδή τα μηνύματα που βρίσκονται πιο πάνω στο διάγραμμα προηγούνται χρονικά αυτών που βρίσκονται πιο χαμηλά.



3.2.3.6 ΔΙΑΓΡΑΜΜΑΤΑ ΣΥΝΕΡΓΑΣΙΑΣ

Σε ένα **διάγραμμα συνεργασίας** (collaboration diagrams) τα αντικείμενα απεικονίζονται με τις γραμμές συσχετίσεων των τάξεών τους να τα ενώνουν, δηλαδή απεικονίζονται οι στατικές συνδέσεις μεταξύ των αντικειμένων. Δεν υπάρχει συγκεκριμένη μορφή (τα αντικείμενα μπορούν να είναι σε όποιες θέσεις βολεύει) και για απεικονιστεί η ακολουθία των μηνυμάτων που ανταλλάσσονται χρησιμοποιείται αρίθμηση.

Ένα παράδειγμα ενός διαγράμματος συνεργασίας είναι το ακόλουθο:



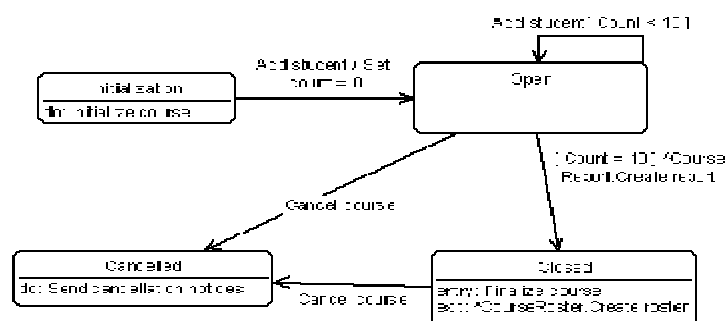
Το διάγραμμα αυτό αντιστοιχεί στο προηγούμενο διάγραμμα ακολουθίας και μπορείτε εύκολα να δείτε τις αντιστοιχίες μεταξύ των συμβόλων.

Τα βασικά διαγραμματικά στοιχεία των διαγραμμάτων συνεργασίας είναι τα ίδια σχεδόν με τα διαγράμματα ακολουθίας. Η μόνη, ίσως, αξιοσημείωτη διαφοροποιήσεις είναι ότι:

- Σε κάθε αντικείμενο απεικονίζονται και οι στατικές του συνδέσεις με τα άλλα αντικείμενα (κάτι που κάνει εμφανής τη στατική δομή και δεν φαίνεται στα διαγράμματα ακολουθίας)
- Σε κάθε μήνυμα υπάρχει και ένας αριθμός μηνύματος. (Το Rational Rose εμφανίζει αριθμούς και στα μηνύματα του διαγράμματος ακολουθίας. Αυτό δεν βλάπτει αλλά δεν είναι και απαραίτητο μια και η χρονική σειρά των μηνυμάτων στο διάγραμμα ακολουθίας είναι από πάνω προς τα κάτω.)

3.2.3.7 ΔΙΑΓΡΑΜΜΑΤΑ ΚΑΤΑΣΤΑΣΕΩΝ

Στην UML τα διαγράμματα κατάστασης χρησιμοποιούνται κυρίως για την μοντελοποίηση λειτουργιών που καθοδηγούνται από συμβάντα (αναδραστικές λειτουργίες). Ουσιαστικά τα διαγράμματα κατάστασης περιγράφουν όλες τις πιθανές **καταστάσεις** (states) στις οποίες μπορεί να βρεθούν τα αντικείμενα μιας τάξης και τον τρόπο που αλλάζει η κατάσταση αυτών των αντικειμένων σαν ανάδραση σε συμβάντα που προκαλούνται από άλλα αντικείμενα και φτάνουν στα αντικείμενα της τάξης την δυναμική συμπεριφορά της οποίας και περιγράφουμε.



Οι καταστάσεις παριστάνονται με οβάλ σχήματα. Για να περάσουμε από μια κατάσταση σε μια άλλη γίνεται μια **μετάβαση (transition)** η οποία συμβολίζεται με ένα κατευθυνόμενο βέλος. Επίσης υπάρχει η δυνατότητα της **αυτό-μετάβασης (self-transition)** δηλαδή μιας μετάβασης πάλι πίσω στην ίδια κατάσταση (όπως είναι η μετάβαση «Add Student» από την κατάσταση «Open»).

Οι μεταβάσεις γίνονται σαν ανάδραση σε συμβάντα που προκαλούν την μετάβαση και αναγράφονται πάνω στα κατευθυνόμενα βέλη. Επίσης κάποιες μεταβάσεις μπορεί να είναι υπό συνθήκη. Οι συνθήκες όπως και στα διαγράμματα αλληλεπίδρασης, αναγράφονται μέσα σε αγκύλες.

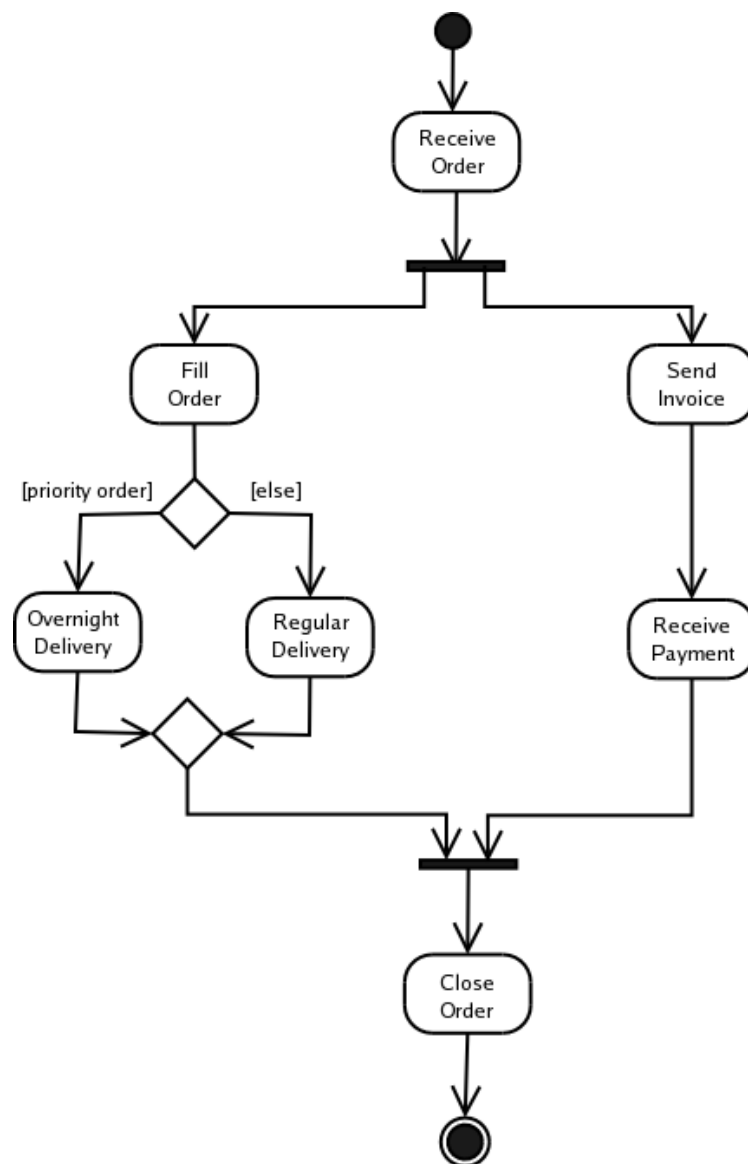
Τα διαγράμματα κατάστασης δείχνουν την ιστορία της ζωής των αντικειμένων μιας δεδομένης τάξης, τα **συμβάντα (events)** που προκαλούν τις μεταβάσεις από μια κατάσταση σε κάποια άλλη κατάσταση, και τις **ενέργειες (actions)** που είναι αποτέλεσμα αυτών των αλλαγών κατάστασης.

Δεν δημιουργούμε διαγράμματα κατάστασης για όλες τις τάξεις ενός συστήματος αλλά μόνο για εκείνες των οποίων τα αντικείμενα επιδεικνύουν μια έντονα δυναμική συμπεριφορά.

3.2.3.8 ΔΙΑΓΡΑΜΜΑΤΑ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ

Τα διαγράμματα δραστηριοτήτων χρησιμοποιούνται για την απεικόνιση διαδικασιών, επιχειρηματικών διεργασιών και ροής εργασίας.

ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ - Διάγραμμα Δραστηριοτήτων (Fowler, 2004)



3.3 Επίλογος - Συμπεράσματα

Η UML στοχεύει στο σχεδιασμό αντικειμενοστραφών συστημάτων. Το σχέδιο είναι μια απλοποιημένη παράσταση της πραγματικότητας.

Σχεδιάζουμε για να μπορέσουμε να καταλάβουμε το σύστημα που αναπτύσσουμε. Έτσι δημιουργώντας ένα σχέδιο επιτυγχάνουμε τέσσερις στόχους:

1. παριστάνουμε οπτικά το σύστημα που έχουμε ή θέλουμε να κατασκευάσουμε,
2. προσδιορίζουμε τη δομή και τη συμπεριφορά του συστήματος,
3. δημιουργούμε ένα πρότυπο για να βασίσουμε την κατασκευή του συστήματος,
4. τεκμηριώνουμε τις αποφάσεις που λάβαμε.

Σε όλους τους τεχνολογικούς τομείς ο σχεδιασμός βασίζεται σε τέσσερις βασικές αρχές:

1. η επιλογή του είδους του σχεδίου έχει επίπτωση στον τρόπο και την μορφή επίλυσης του προβλήματος,
2. όλα τα σχέδια εκφράζονται σε διαφορετικές βαθμίδες ακρίβειας,
3. τα καλύτερα σχέδια σχετίζονται με την πραγματικότητα, ένα είδος σχεδίων δεν είναι ποτέ αρκετό.

4^ο ΚΕΦΑΛΑΙΟ

ΠΕΡΙΠΤΩΣΕΙΣ ΧΡΗΣΗΣ (Use Case Scenario)

4.1 Εισαγωγή

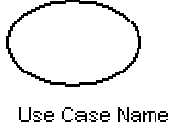




Στο κεφάλαιο 4 υπάρχει μια εισαγωγή στα εργαλεία CASE, ένα σύνολο από εφαρμογές λογισμικού οι οποίες στοχεύουν στην υποστήριξη ή και την αυτοματοποίηση της ίδιας της κατασκευής του λογισμικού. Τα εργαλεία CASE, αν και υποσχέθηκαν πολλά, στην αρχή τουλάχιστον δεν είχαν τα αναμενόμενα αποτελέσματα, θέση η οποία φαίνεται να φθίνει καθώς η κοινότητα του λογισμικού ωριμάζει. Έχοντας κατά νου το πλαίσιο αυτό, επιχειρούμε μια εισαγωγή και μια ταξινόμηση των εργαλείων CASE, καθώς και μια σύντομη αναφορά σε κριτήρια επιλογής τους.

Σε αυτό το κεφάλαιο θα παρουσιάσουμε αναλυτικότερα τα στοιχεία, τους λόγους χρησιμοποίησης διαγραμμάτων χρήσης και τα βήματα που πρέπει να ακολουθήσουμε για να κατασκευάσουμε ένα τέτοιο διάγραμμα.

4.2 Τι είναι η μοντελοποίηση με περιπτώσεις χρήσης;

Κατ' αρχήν θα πρέπει να αναφέρουμε ξανά τί είναι Μοντέλο περιπτώσεων χρήσης: Είναι μία άποψη του συστήματος που αποδίδει έμφαση στην λειτουργικότητα ενός συστήματος, όπως αυτή είναι ορατή από τους εξωτερικούς χρήστες του συστήματος. Μια περίπτωση χρήσης διαμερίζει την λειτουργικότητα ενός συστήματος σε συναλλαγές («περιπτώσεις χρήσης») που έχουν νόημα για τους χρήστες του συστήματος («ρόλους»).

Στον ακόλουθο πίνακα συνοψίζονται οι βασικές έννοιες που συναντάμε σε ένα διάγραμμα περιπτώσεων χρήσης.

ΚΑΤΑΣΚΕΥΗ	ΠΕΡΙΓΡΑΦΗ	ΣΥΝΤΑΚΤΙΚΟ
Περίπτωση χρήσης	Μία ακολουθία ενεργειών, συμπεριλαμβανομένων και των παραλλαγών τους, που μπορεί να επιτελέσει ένα σύστημα σε αλληλεπίδραση με τους ρόλους που υπάρχουν στο σύστημα αυτό	
Ρόλος	Ένας ρόλος είναι ένας ρόλος που μπορεί να παίξει ένας χρήστης του συστήματος όταν αλληλεπιδρά με τις περιπτώσεις χρήσης του συστήματος	
Συσχέτιση	Δηλώνει την συμμετοχή ενός ρόλου σε μια περίπτωση χρήσης	
Επέκταση	Η σχέση μιας επεκταμένης και μιας περίπτωσης χρήσης βάσης, που προσδιορίζει πως θα γίνει αυτή η επέκταση (τα σημεία επέκτασης).	
Γενίκευση	Μια συσχέτιση μιας πιο γενικής περίπτωσης χρήσης με μια πιο ειδική περίπτωση χρήσης.	

Περιεκτικότητα	Μία τέτοια συσχέτιση δείχνει ότι μια περίπτωση χρήσης περιλαμβάνει τις λειτουργίες μιας άλλης περίπτωσης χρήσης. Δηλαδή η λειτουργικότητα της περιλαμβανόμενης περίπτωσης χρήσης εισάγεται στην περίπτωση χρήσης βάσης.	<pre>«include» →</pre>
----------------	---	------------------------

Συνήθως υπάρχουν κάποια βήματα-συμβουλές τις οποίες πρέπει ο κατασκευαστής να ακολουθεί, για την αποτελεσματικότερη επιτέλεση της εργασίας του μοντέλου.

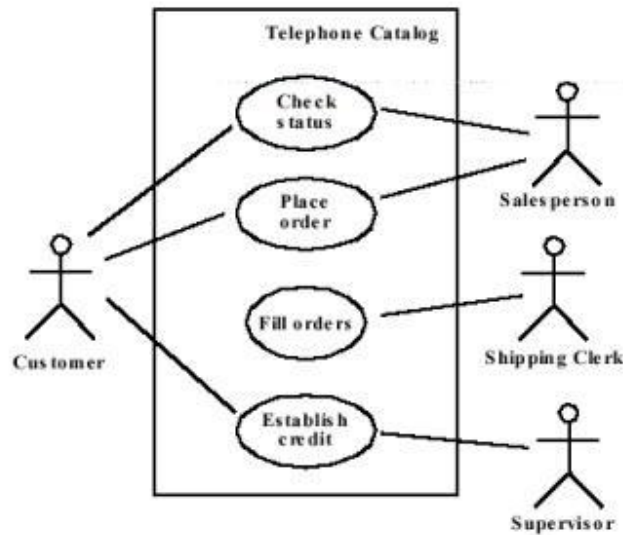
- Εξασφαλίστε ότι κάθε περίπτωση χρήσης περιγράφει ένα σημαντικό τμήμα της λειτουργικότητας του συστήματος που είναι κατανοητό από τους ειδικούς του χώρου ανάπτυξης (domain experts) αλλά και από τους προγραμματιστές.
- Όταν ορίζετε τις περιπτώσεις χρήσης με κείμενο να είστε ακριβείς και συνεπείς με τα ρήματα και τα ουσιαστικά. Αυτό θα βοηθήσει στην παραγωγή των αντικειμένων και των διαγραμμάτων αλληλεπίδρασης αργότερα
- Εξάγετε τις περιπτώσεις χρήσης που απαιτούνται από άλλες περιπτώσεις χρήσης:
 - Αν η περίπτωση χρήσης που περιλαμβάνεται είναι υποχρεωτική χρησιμοποιείστε σχέση περιεκτικότητας («include»).
 - Αν η περίπτωση χρήσης βάσης είναι πλήρης και η χρήση της περιλαμβανόμενης περίπτωσης χρήσης προαιρετική χρησιμοποιείστε τη σχέση της επέκτασης («extend»).

- Ένα διάγραμμα περιπτώσεων χρήσης θα πρέπει:
 - Να περιέχει μόνο περιπτώσεις χρήσης που ανήκουν στο ίδιο επίπεδο αφαίρεσης
 - Να περιλαμβάνει μόνο ρόλους που απαιτούνται
- Αν σε κάποιο έργο υπάρχουν πολλές περιπτώσεις χρήσης τότε αυτές θα πρέπει να οργανωθούν σε πακέτα

Ένα μοντέλο περιπτώσεων χρήσης απεικονίζει τις *περιπτώσεις χρήσης*, τους *ρόλους* και τις *συσχετίσεις* μεταξύ τους. Πρέπει να σημειωθεί ότι η εσωτερική λειτουργία κάθε περίπτωσης χρήσης μπορεί να περιγραφεί με κείμενο ή να απεικονισθεί με διαγράμματα αλληλεπίδρασης. Τα είδη που υπάρχουν δεν περιορίζονται μόνο στα διαγράμματα αλλά μπορούμε να έχουμε και την περιγραφή με κείμενο. Αρχικά θα εξετάσουμε τη διαγραμματική τεχνική και στη συνέχεια την περιγραφή με κείμενο.

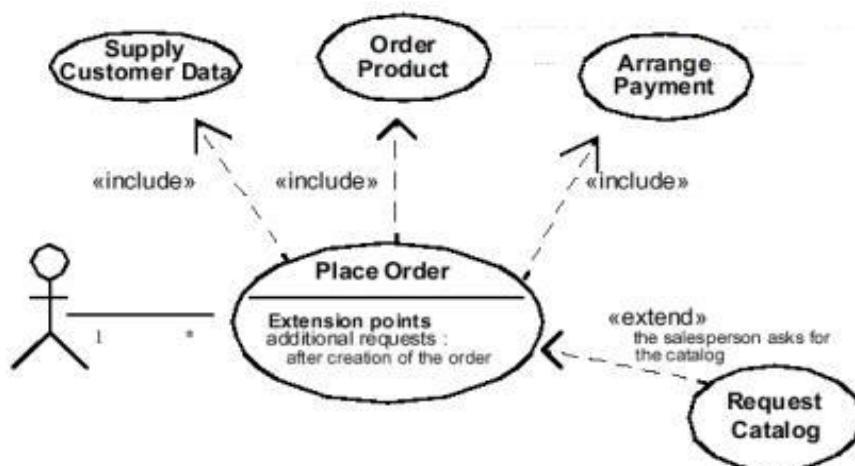
4.2.1 1ο Παράδειγμα διαγραμματικής τεχνικής:

Το ακόλουθο είναι ένα διάγραμμα που απεικονίζει ένα Πελάτη (Customer) που συνδιαλέγεται με ένα σύστημα αγοράς εξ αποστάσεως (μέσω τηλεφώνου ή και μέσω Internet). Ο πελάτης θέλει να μπορεί να ελέγξει την κατάσταση σε ότι αφορά ένα προϊόν (Check Status), να μπορεί να δώσει μια παραγγελία (Place Order) και να μπορεί να δώσει και να επιβεβαιώσει τα στοιχεία της πιστωτικής του κάρτας. Ο πωλητής (Salesperson – υποθέτουμε ότι η εταιρεία πέρα από τηλεφωνικές πωλήσεις κάνει και άλλες πωλήσεις μέσω πωλητών) θέλει να μπορεί να κάνει τα ίδια που κάνει και ένας πελάτης με την διαφορά ότι δεν χρειάζεται να δώσει τις λεπτομέρειες της πιστωτικής του κάρτας. Ο υπάλληλος αποστολής (Shipping Clerk) θέλει να μπορεί να εκδώσει τις παραγγελίες (Fill Orders), και τέλος ο Επόπτης (Supervisor) θέλει να μπορεί να εξετάσει την πιστωτική εικόνα των πελατών (Establish Credit).

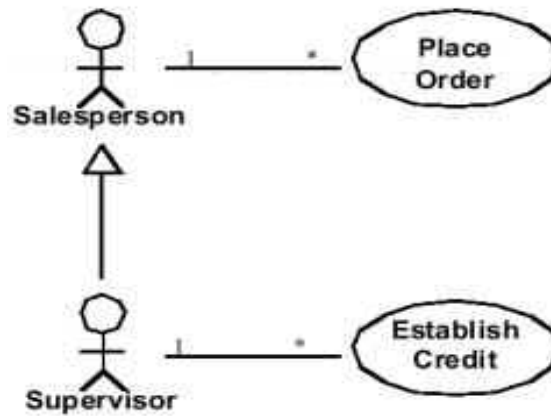


Το ακόλουθο είναι ένα παράδειγμα των σχέσεων περιεκτικότητας («includes») και επέκτασης («extend»).

Προκειμένου ο πελάτης να πραγματοποιήσει μια παραγγελία θα πρέπει αρχικά να του δοθεί κατάλογος προϊόντων από τον πωλητή («extend»). Στη συνέχεια η εκτέλεση της παραγγελίας, που γίνεται από τον πωλητή περιλαμβάνει τα στοιχεία του πελάτη, την παραγγελία του προϊόντος και τον τρόπο πληρωμής («include»).



Επίσης μπορούν να υπάρξουν και σχέσεις μεταξύ των ρόλων ενός διαγράμματος περιπτώσεων χρήσης όπως δείχνει το ακόλουθο διάγραμμα:



Όπως βλέπουμε στο παραπάνω διάγραμμα ο επόπτης (supervisor) εξετάζει την πιστωτική εικόνα των πελατών και εγκρίνει την παραγγελία (γενική περίπτωση χρήσης) συσχετιζόμενος με τον πωλητή που δίνει εντολή για τις παραγγελίες.

4.2.2 2ο Παράδειγμα με κείμενο

Το ακόλουθο παράδειγμα περιγράφει μια περίπτωση χρήσης για την αλλαγή στοιχείων πτήσης ενός ταξιδιώτη:

ΠΕΡΙΠΤΩΣΗ ΧΡΗΣΗΣ: ΑΛΛΑΓΗ ΣΤΟΙΧΕΙΩΝ ΠΤΗΣΗΣ

Ρόλοι: ταξιδιώτης, λογαριασμός ΒΔ πελάτη, σύστημα κράτησης θέσεων

Προ-συνθήκες:

Ο ταξιδιώτης συνδέθηκε με το σύστημα και επέλεξε την επιλογή «αλλαγής πτήσης»

Βασικό σενάριο:

Το σύστημα ανακτά τον λογαριασμό του πελάτη από την ΒΔ

Το σύστημα ρωτά τον ταξιδιώτη να επιλέξει την παράμετρο της πτήσης που θέλει να αλλάξει.

Ο ταξιδιώτης επιλέγει την επιθυμητή παράμετρο.

Το σύστημα ζητά νέα αναχώρηση και προορισμό. Ο ταξιδιώτης δίνει τις απαιτούμενες πληροφορίες

...

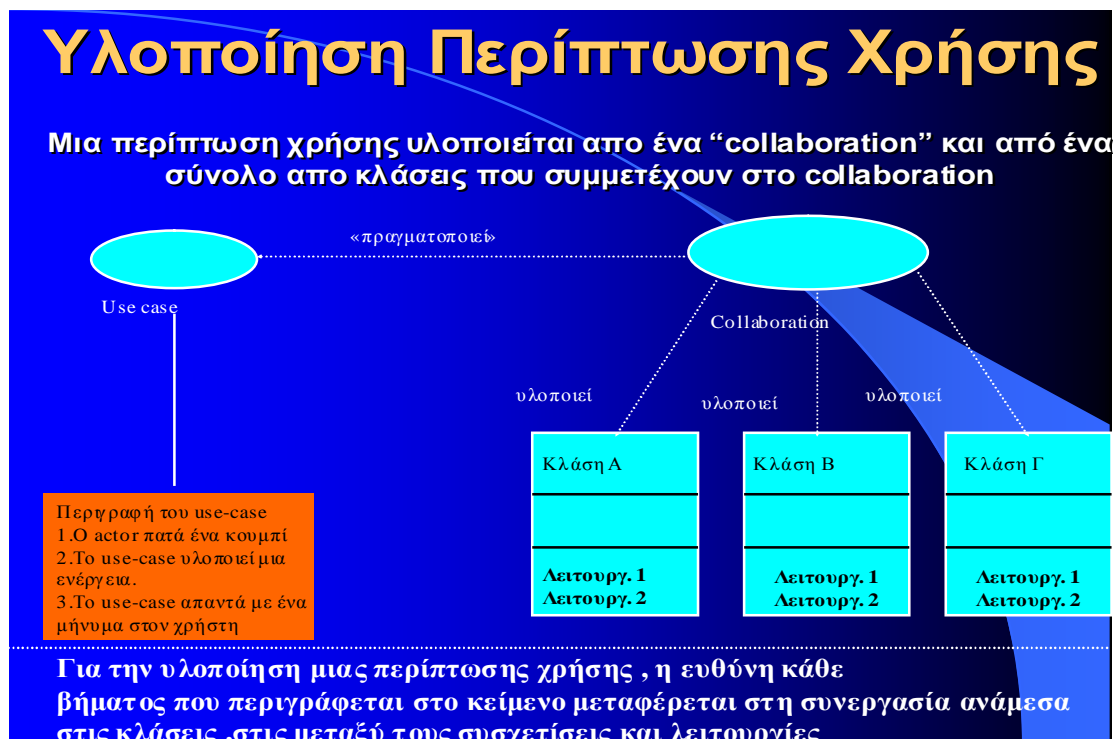
Το σύστημα παρουσιάζει περίληψη της συναλλαγής

Εναλλακτικά σενάρια:

Αν δεν υπάρχουν διαθέσιμες πτήσεις τότε...

Για την υλοποίηση μιας περίπτωσης χρήσης χρησιμοποιούνται κάποιες αρχές της UML, όπως για παράδειγμα, μια συνεργασία κλάσεων και αντικειμένων (collaboration). Το διάγραμμα που θα κατασκευαστεί θα παρουσιάζει την εσωτερική υλοποίηση στο επίπεδο των αντικειμένων και των κλάσεων και τις μεταξύ τους σχέσεις (το πεδίο (context) της συνεργασίας (collaboration) θα φανερώνει δηλαδή τη μεταξύ τους αλληλεπίδραση για την επίτευξη της επιθυμητής λειτουργικότητας (αλληλεπίδραση (interaction) της συνεργασίας των κλάσεων). Επίσης θα δείχνει τόσο το πεδίο όσο και την αλληλεπίδραση μεταξύ των συστατικών.

Το σενάριο που είναι ένα στιγμιότυπο της περίπτωσης χρήσης ή του collaboration, περιγράφει τις εμπλεκόμενες κλάσεις, τις λειτουργίες του και τη μεταξύ τους επικοινωνία όπως θα δούμε και στο παρακάτω διάγραμμα.



Υπάρχουν ορισμένα είδη αντικειμένων τα οποία χρησιμοποιούνται για την περιγραφή του collaboration που υλοποιεί μια περίπτωση χρήσης (μέθοδος Jacobson) και είναι τα εξής:

- **Αντικείμενα ορίων** το είδος των αντικειμένων αυτών βρίσκεται κοντά στα όρια του συστήματος και αλληλεπιδρά με τους χρήστες έξω από τα όρια του συστήματος
- **Αντικείμενα ελέγχου** τα οποία ελέγχουν τις αλληλεπιδράσεις ανάμεσα σε ομάδες αντικειμένων
- **Αντικείμενα οντότητας** του οποίου ο τύπος εκπροσωπεί μια οντότητα στην περιοχή εφαρμογής του συστήματος. Είναι κυρίως παθητικό και δεν αρχικοποιεί αλληλεπιδράσεις από μόνο του.

4.2.3 Παραδειγμα 3ο

Το σύστημα ρωτά το χρήστη για έναν αριθμό παραγγελίας .

Ο χρήστης παρέχει τον αριθμό .

Το σύστημα ζητά τη παραγγελία (από τη χρήση Locate Order Περίπτωση).

Εάν η παραγγελία δεν βρίσκεται, λάθος, στοπ.

Αλλιώς:

Το σύστημα παρέχει τη παραγγελία στο χρήστη.

Ο χρήστης επιλέγει ένα αντικείμενο.

Μέχρι ο χρήστης να δείξει ότι η παραγγελία έχει γίνει ή ότι δεν υπάρχουν ασυμπλήρωτες ποσότητες στοιχείων μεγαλύτερες από 0:

Το σύστημα ζητά τη θέση του αντικειμένου και την ασυμπλήρωτη ποσότητα (από την περίπτωση χρήσης Locate Product).

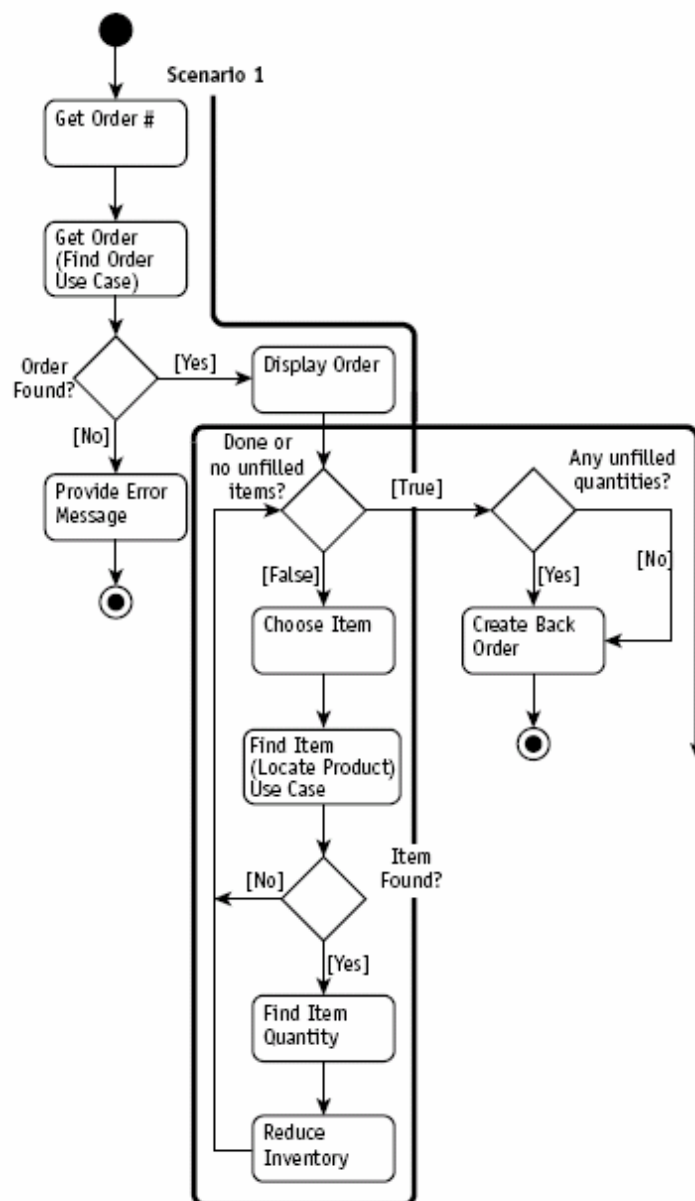
Εάν το αντικείμενο βρίσκεται (διαθέσιμο):

Ο χρήστης δείχνει την ποσότητα του αντικειμένου που θα συμπληρώσει.

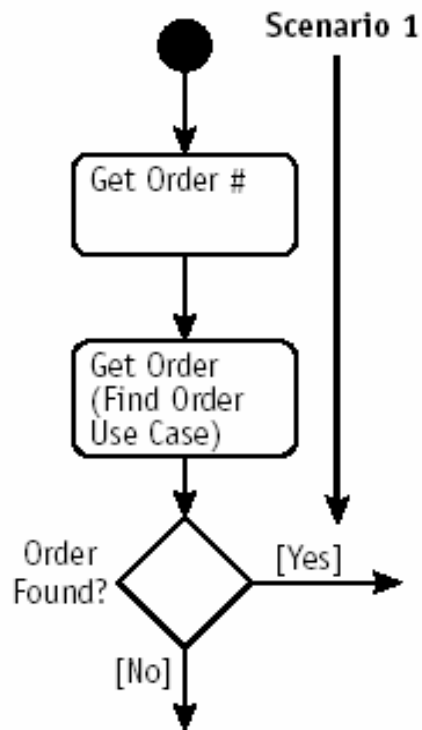
Εάν υπάρχουν οποιεσδήποτε ασυμπλήρωτες ποσότητες αντικειμένων μεγαλύτερες από 0:

Δημιουργει παραγγελία backorder(χρησιμοποιώντας το create backorder use case).

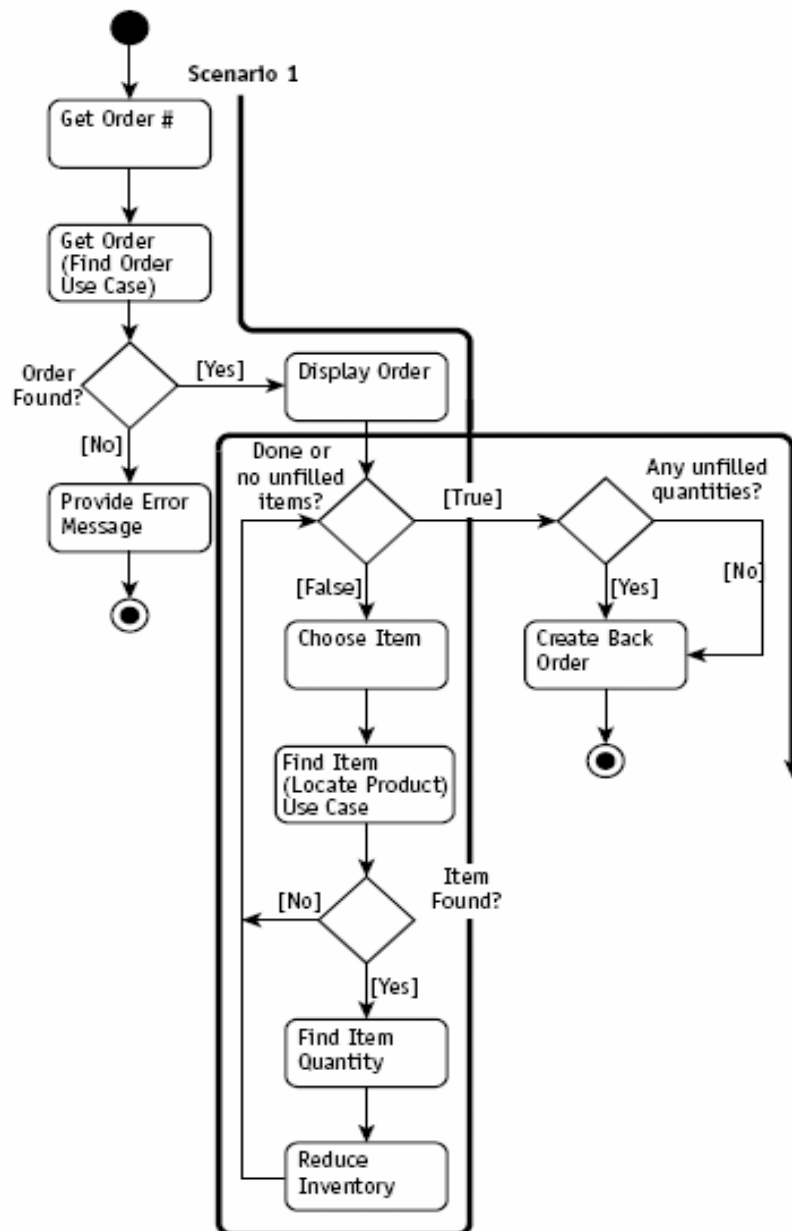
Για το Fillorder use case σενάριο δημιουργήσαμε ένα διάγραμμα δραστηριότητας (activity diagram) για την καλύτερη κατανόηση των βημάτων της διαδικασίας.Σενάριο 1.



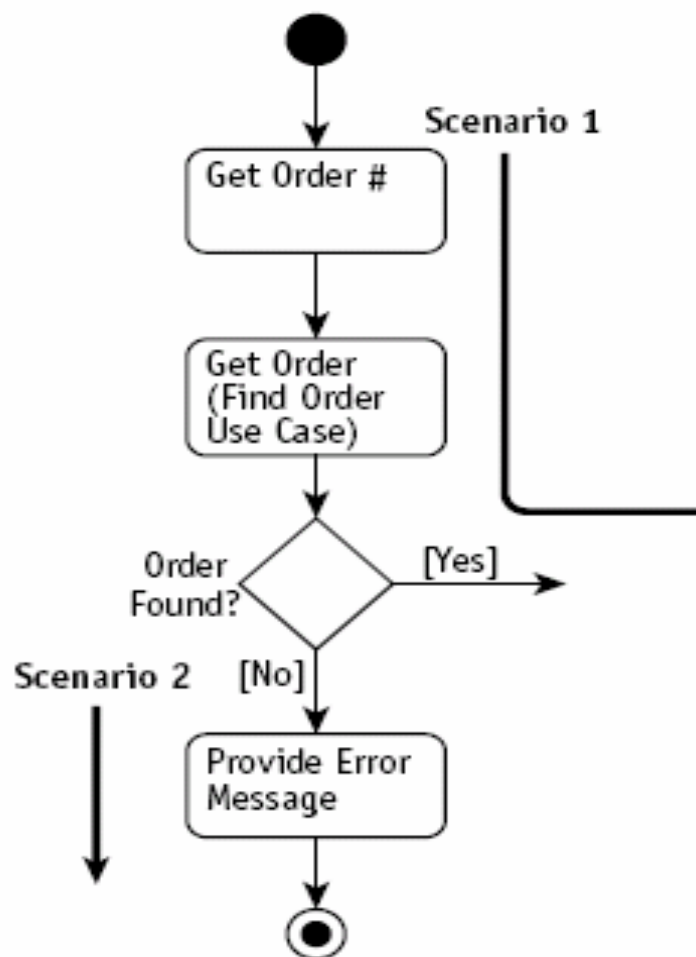
Στο επόμενο σχήμα απεικονίζουμε τη λογική για ένα πετυχημένο σενάριο. Βλέπουμε αναλυτικά τα βήματα που ακολουθούμε στην αρχή μέχρι το σενάριο να φτάσει σε σημείο που αρχίζουν οι εναλλακτικές λύσεις.



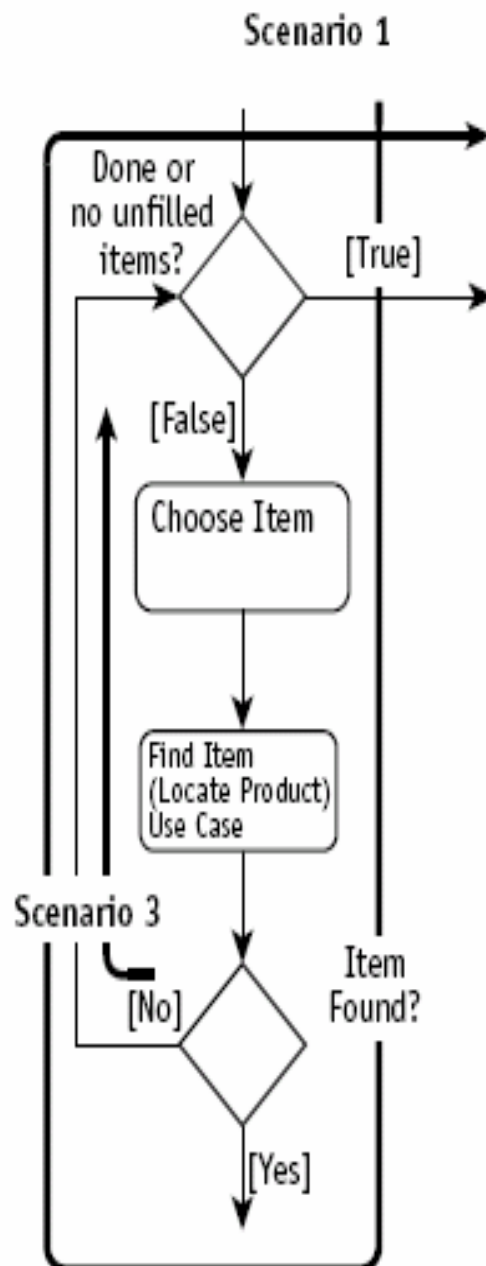
Στο αμέσως επόμενο σχήμα απεικονίζεται η ροή των βημάτων από τη στιγμή που εντοπίστηκε η παραγγελία και επιλέγεις τη yes εντολή. Συνεχίζουμε μέχρι την επόμενη εναλλακτική (έγινε η παραγγελία ή υπάρχουν ασυμπλήρωτες ποσότητες) ακολουθώντας μία απ' τις δύο ροές.



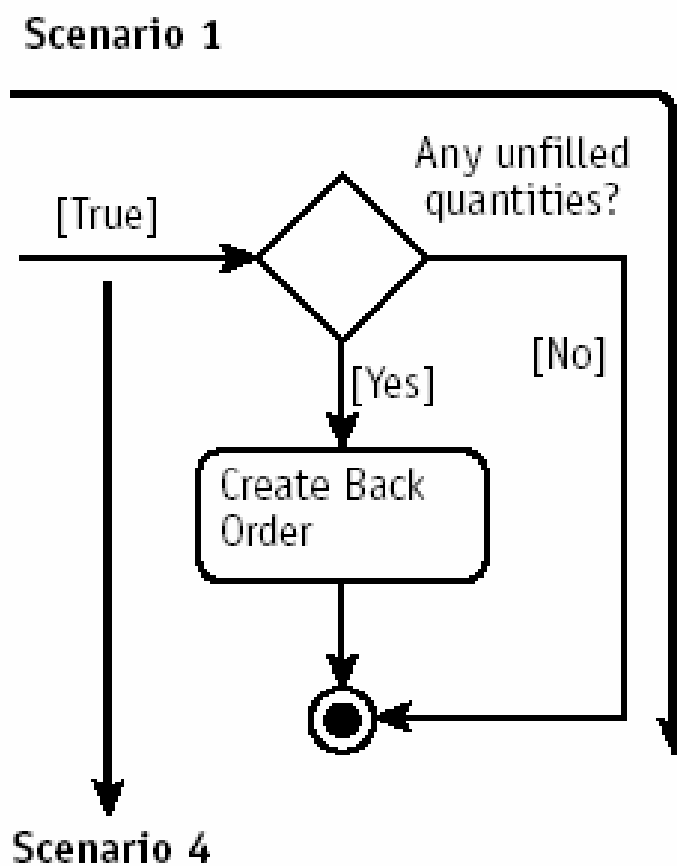
Στο διάγραμμα που ακολουθεί, αρχίζουμε να προσδιορίζουμε τις εναλλακτικές λύσεις. Στην περίπτωση αυτή, παίρνουμε την false εντολή και συνεχίζουμε μέχρι το τέλος της. Σενάριο 2



Στο επόμενο διάγραμμα προσδιορίζεται η ροή του διαγράμματος από το σημείο που ακόμα δεν ξέρουμε για το αν βρέθηκε το αντικείμενο, μόνο που σ' αυτή τη περίπτωση, επιλέγουμε τη διαδικασία κατά την οποία δεν έχει βρεθεί. Αυτό είναι το σενάριο 3.

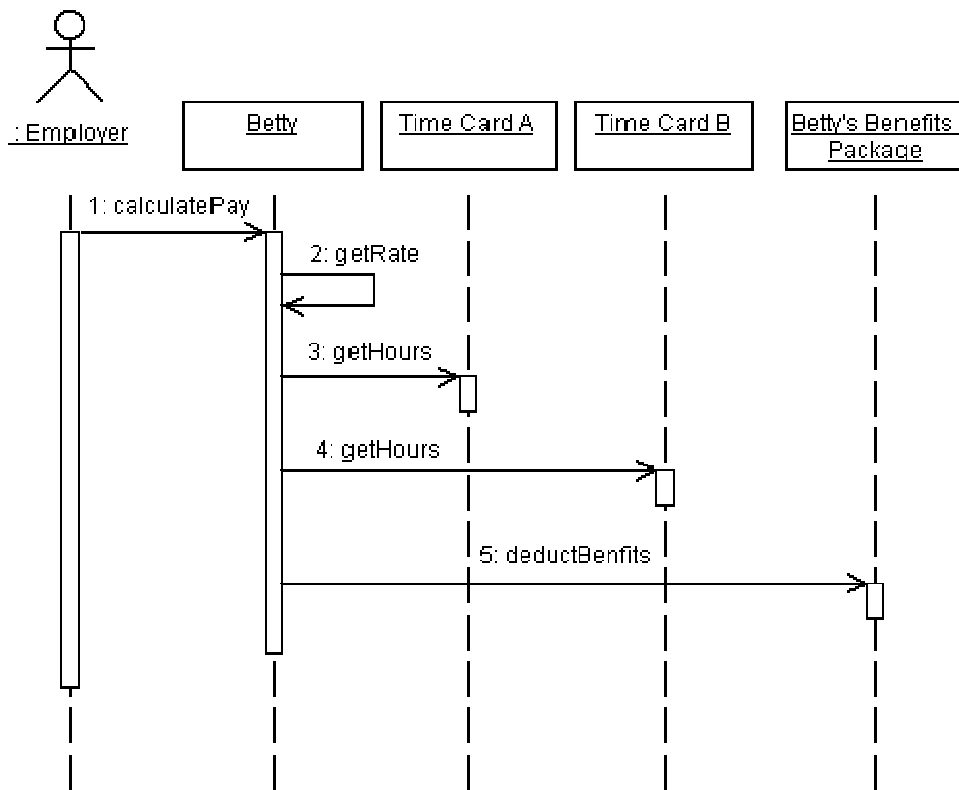


Στο παρακάτω διάγραμμα εμφανίζεται το σενάριο 4, το οποίο συνεχίζεται από την αναζήτηση μη συμπληρωμένων ποσοτήτων (σενάριο 1). Αυτή τη φορά διαλέγουμε την περίπτωση στην οποία υπάρχουν ποσότητες ασυμπλήρωτες και έτσι είναι απαραίτητη η δημιουργία παραγγελίας (backorder).

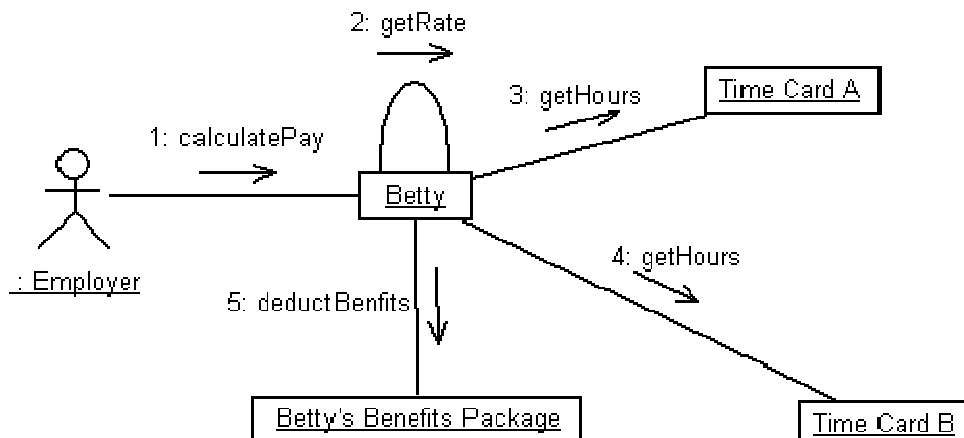


4.2.4 4ο ΠΑΡΑΔΕΙΓΜΑ(χρησιμοποιούμε διάγραμμα ακολουθίας και συνεργασίας)

θα αρχίσουμε με ένα διάγραμμα ακολουθίας που απεικονίζει το σενάριο πληρωμής των ωρομισθίων υπαλλήλων μιας εταιρίας. Τα διαγράμματα ακολουθίας χρησιμοποιούνται (κυρίως) για την απεικόνιση των εναλλακτικών σεναρίων μιας περίπτωσης χρήσης.



το διάγραμμα συνεργασίας που θα προκύψει από αυτό το σενάριο είναι το παρακάτω:



Το πρόγραμμα rational rose σου δίνει τη δυνατότητα να μετατρέψεις αυτόματα το διάγραμμα ακολουθίας σε διάγραμμα συνεργασίας.

4.2.5 5^ο ΠΑΡΑΔΕΙΓΜΑ

Θα επανέλθουμε αντιμετωπίζοντας αυτή τη φορά την ανάπτυξη της με την αντικειμενοστρεφή τεχνολογία σύμφωνα με την Ενοποιημένη Προσέγγιση. Στη συνέχεια παραθέτουμε τον ορισμό του προβλήματος και τις απαιτήσεις από το λογισμικό που εντοπίσαμε κατά την αναφορά μας στον πρώτο τόμο.

<<ΓΑΛΗΝΟΣ>> Μια εφαρμογή λογισμικού για την υποστήριξη μιας φαρμακαποθήκης.

Ο πελάτης μας είναι υπεύθυνος για τη λειτουργία της φαρμακαποθήκης.

Λόγω του πλήθους των προμηθευτών (φαρμακευτικές εταιρίες)

των πελατών (φαρμακεία) και των φαρμάκων, του όγκου και της πολυπλοκότητας

των εργασιών υποστήριξης (αρχείου, παραγγελιών και αποδοχών κ.ά.), είναι αναγκαία η

χρήση μιας εφαρμογής λογισμικού, την οποία ο πελάτης αποφασίζει να ονομάσει «ΓΑΛΗΝΟΣ» και να μας αναθέσει την ανάπτυξή της.

Η εφαρμογή θα πρέπει να τηρεί αρχεία προμηθευτών, πελατών, φαρμάκων, καθώς και παραγγελιών.

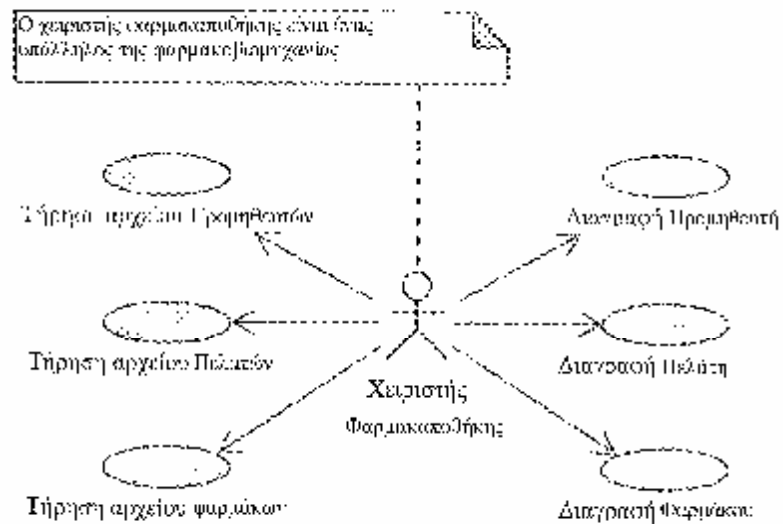
Η εφαρμογή θα πρέπει να δίνει πληροφορίες για τους παραπάνω με κριτήρια που θα δίνει ο χρήστης. Η

εφαρμογή δεν θα πρέπει να επιτρέπει τη διαγραφή ενός προμηθευτή ή πελάτη από το αρχείο αν αυτός έχει οποιοσδήποτε συναλλαγές. Το περιβάλλον λειτουργίας θα είναι ένας αυτόνομος ηλεκτρονικός υπολογιστής με Windows 9x.

Απαιτήσεις από το λογισμικό:

1. Ο «ΓΑΛΗΝΟΣ» θα τρέχει σε αυτόνομο υπολογιστή κάτω από το λειτουργικό σύστημα Windows 9x – 32 bit (95, 98, NT, 2000). Δεν απαιτείται σύνδεση σε δίκτυο.
2. Ζητείται η τήρηση αρχείων προμηθευτών, πελατών, φαρμάκων
3. Κάθε προμηθευτής / πελάτης εγγράφεται κάθε φορά που εκτελεί κάποια συναλλαγή (αγορά/πώληση)
4. Δεν πρέπει να επιτρέπεται η καταχώρηση παραγγελίας φαρμάκων σε προμηθευτή /πελάτη ο οποίος δεν έχει εγγραφεί.
5. Επιτρέπεται η διαγραφή προμηθευτή /πελάτη μόνο αν δεν έχει πραγματοποιήσει οποιαδήποτε συναλλαγή σε διάστημα 5ετίας.
6. Επιτρέπεται η διαγραφή φαρμάκου μόνο αν έχει αποσυρθεί από την αγορά (κυκλοφορία).
7. Ζητείται αλφαβητική εκτύπωση ολόκληρου του αρχείου των προμηθευτών, πελατών και φαρμάκων για διευκόλυνση σε στατιστικές έρευνες.

Θα κατασκευάσουμε ένα διάγραμμα περιπτώσεων χρήσης που αντιστοιχεί στις απαιτήσεις 2, 5 και 6. Προς το παρόν, θα θεωρήσουμε ότι έχουμε έναν χειριστή ο οποίος αντιστοιχεί στους συμμετέχοντες των δοσοληψιών της φαρμακαποθήκης. Το διάγραμμα αυτό φαίνεται παρακάτω:



Το διάγραμμα περιπτώσεων χρήσης για μερικές από τις λειτουργίες απαιτήσεις του λογισμικού "Υάληνός"

Στη συνέχεια θα δώσουμε την αναλυτική περιγραφή των περιπτώσεων χρήσης «Τήρηση αρχείου προμηθευτών» και «διαγραφή προμηθευτή» σύμφωνα με το πρότυπο που δώσαμε στο παραπάνω σχήμα.

Προδιαγραφή περίπτωσης χρήσης

1. Τίτλος περίπτωσης χρήσης

Τήρηση αρχείου προμηθευτών

2. Σύντομη περιγραφή

Η εφαρμογή εμφανίζει μια φόρμα διαλόγου μέσω της οποίας ο χρήστης ενημερώνει το αρχείο προμηθευτών

3. Ροή γεγονότων

3.1 Βασική ροή

- 1. Ο Χειριστής «χειριστήςφαρμακαποθήκης» επιλέγει από το μενού της εφαρμογής την «τήρηση αρχείου προμηθευτών».*
- 2. Ο «ΓΑΛΗΝΟΣ» ανοίγει το αρχείο προμηθευτών.*
- 3. Ο «ΓΑΛΗΝΟΣ» εμφανίζει φόρμα διαλόγου με τα πεδία που περιλαμβάνονται στο αρχείο, καθώς και δύο κουμπιά (buttons) με χαρακτηρισμούς «αποδοχή» και «ακύρωση».*
- 4. Ο Χειριστής δίνει τα στοιχεία προμηθευτή που περιέχονται στη φόρμα και πατάει το κουμπί «αποδοχή».*
- 5. Ο «ΓΑΛΗΝΟΣ» ελέγχει την εγκυρότητα των στοιχείων.*
- 6. Ο «ΓΑΛΗΝΟΣ» εισάγει μια νέα εγγραφή στο αρχείο προμηθευτών.*
- 7. Ο έλεγχος επανέρχεται στο βήμα 2.*

3.2 Εναλλακτικές ροές

3.2.1 Εναλλακτική ροή 1

- 4α. Ο χειριστής επιλέγει «ακύρωση».*
- 5α. Ο «ΓΑΛΗΝΟΣ» κλείνει τη φόρμα και τερματίζει την εργασία.*

3.2.2 Εναλλακτική ροή 2

- 6α. Τα στοιχεία που δόθηκαν είναι ελλιπή.*
- 7α. Ο «ΓΑΛΗΝΟΣ» εμφανίζει μήνυμα στο χρήστη και επανέρχεται στο βήμα 2.*

4. Μη λειτουργικές απαιτήσεις

Δεν υπάρχουν για αυτή την περίπτωση χρήσης.

5. Κατάσταση εισόδου

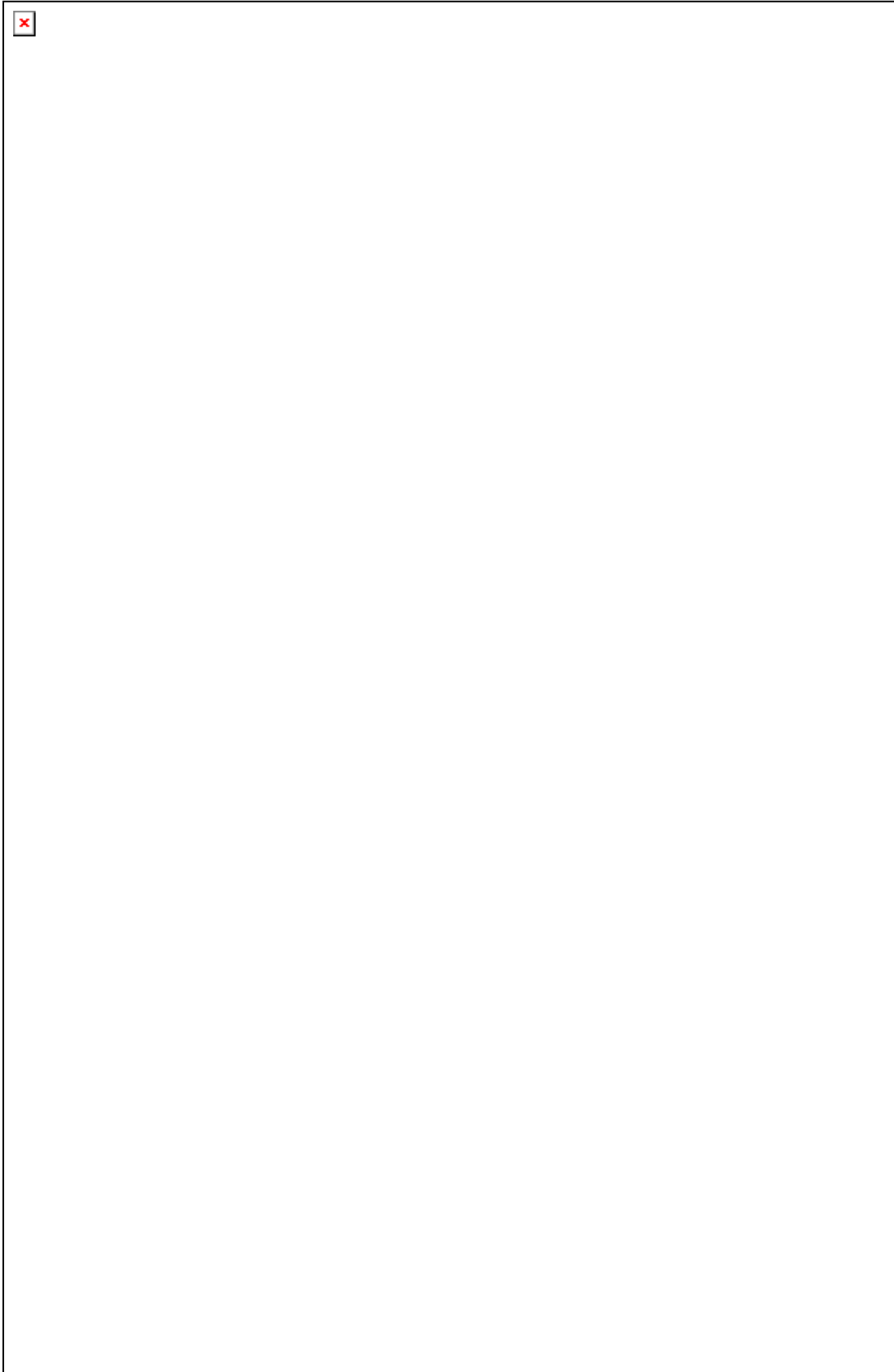
Δεν υπάρχουν ιδιαίτερες απαιτήσεις εισόδου στην περίπτωση χρήσης.

6. Κατάσταση εξόδου

Έχουν προστεθεί 0–N νέες εγγραφές στο αρχείο προμηθευτών.

Διάγραμμα δραστηριότητας περίπτωσης χρήσης

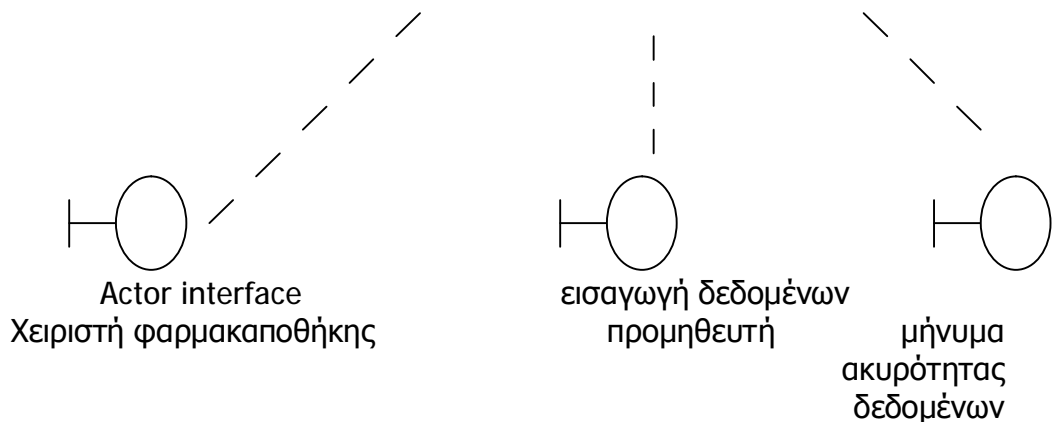
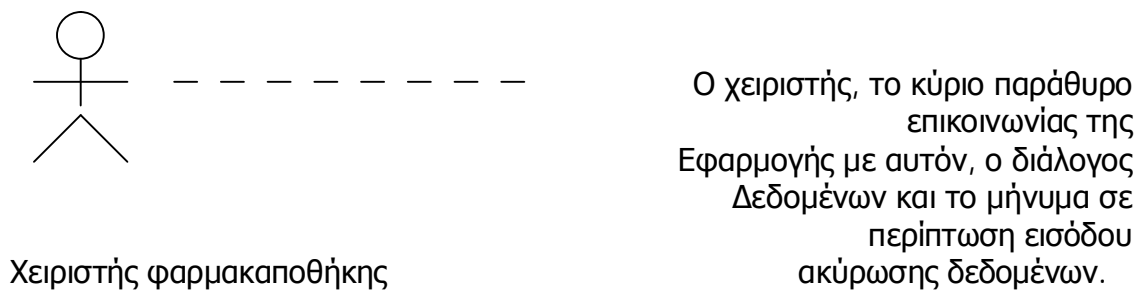
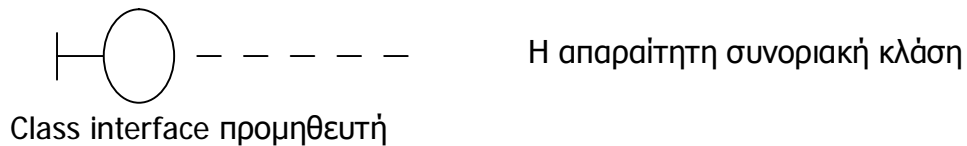
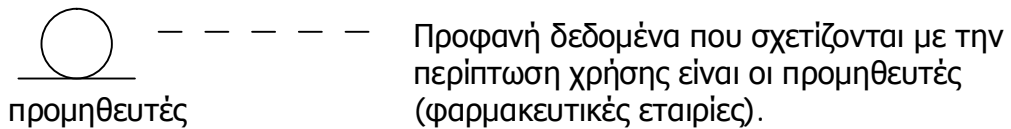
Στο Σχήμα που ακολουθεί δίνεται το διάγραμμα δραστηριότητας για την παραπάνω περιγραφή της ροής εργασιών της περίπτωσης χρήσης. Παρατηρήστε ότι, λόγω σχετικής απλότητας, τόσο η κύρια, όσο και οι εναλλακτικές ροές παριστάνονται με τη βοήθεια ενός και μόνο διαγράμματος.



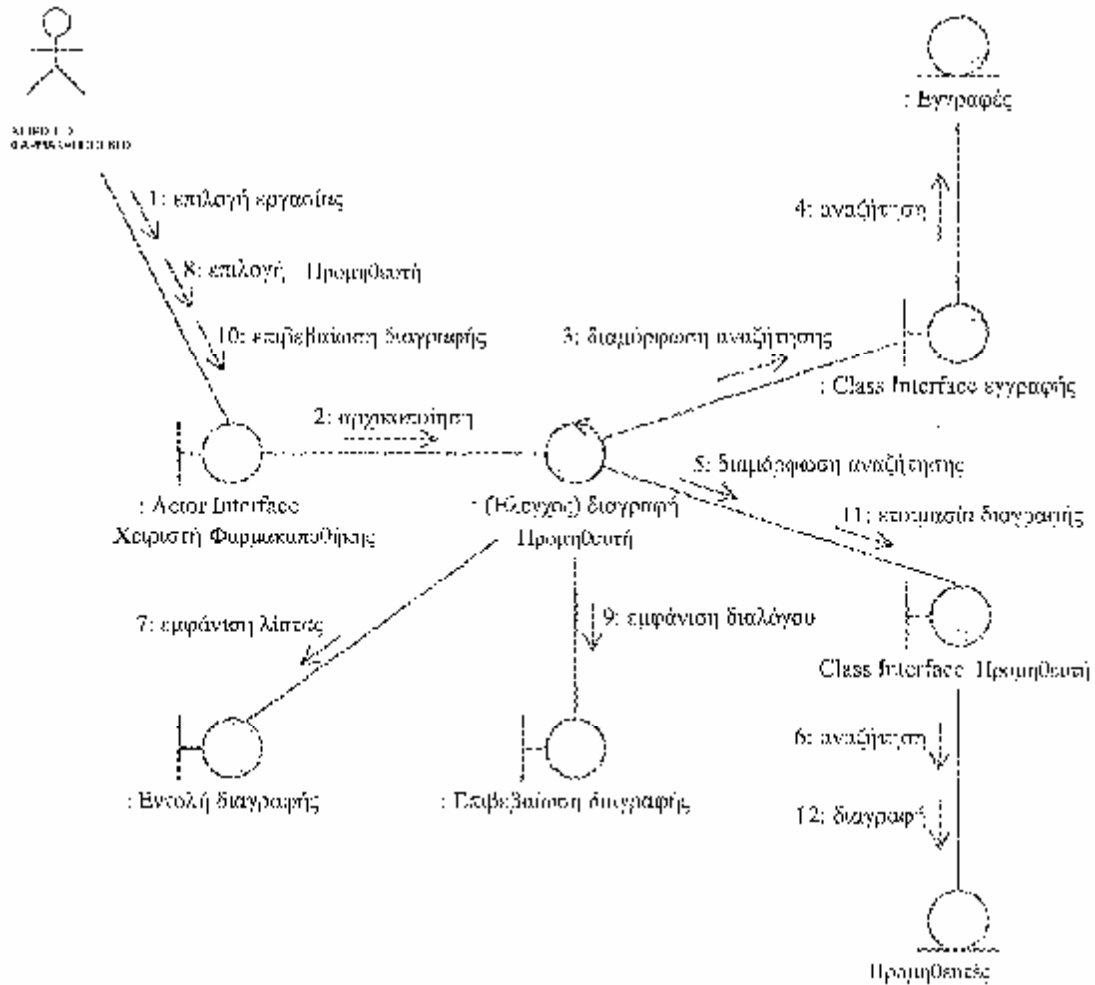
Θα κατασκευάσουμε τα διαγράμματα συνεργασίας για την περίπτωση χρήσης πεδίου ανάλυσης «διαγραφή προμηθευτή». Θα πρέπει να χρησιμοποιήσουμε όλες τις κλάσεις που θα προσδιορίσουμε και να ορίσουμε συνδέσμους συνεργασίας μεταξύ τους, καθώς και ανταλλασσόμενα μηνύματα, ώστε να ικανοποιείται η ροή εργασιών της προδιαγραφής της περίπτωσης χρήσης.

Στο Σχήμα 4 φαίνεται το διάγραμμα συνεργασίας που σχετίζεται με την κύρια ροή ενώ στο Σχήμα 3 οι κλάσεις.

Προσδιορισμός Κλάσεων:



σχήμα 3. ανάλυση της περίπτωσης χρήσης <τήρηση του αρχείου προμηθευτών> και εμφάνιση κλάσεων της εφαρμογής ΓΑΛΗΝΟΣ.



Λαίψηγραμμα συνεργασίας για την κύρια ροή της περίπτωσης χρήσης «διαγραφή προμηθευτή»

4.3 Συμπεράσματα

Μιά καλοσχεδιασμένη περίπτωση χρήσης...

- Περιγράφει μιά συγκεκριμένη και λογικά ατομική (ανεξάρτητη) συμπεριφορά του συστήματος (ή μέρους του συστήματος)
- Περιγράφει τη ροή των γεγονότων με σαφή και εύληπτο τρόπο για κάποιον μη ειδικό πληροφορικής
- Περιγράφεται με ένα ελάχιστο σύνολο σεναρίων που περιγράφουν την κανονική συμπεριφορά και εναλλακτικές συμπεριφορές της περίπτωσης χρήσης
- Έχει χρησιμοποιήσει με επιτυχία τις σχέσεις γενίκευσης/εξειδίκευσης, «περιλαμβάνει» και «επεκτείνει».

ΕΠΙΛΟΓΟΣ

Η Ενοποιημένη Προσέγγιση ακολουθεί το γενικό μοντέλο κύκλου ζωής λογισμικού, και η ανάπτυξη λογισμικού σύμφωνα με αυτή, χαρακτηρίζεται ως επαναληπτική και επαυξητική. Στην αντικειμενοστρεφή ανάλυση σύμφωνα με την Ενοποιημένη Προσέγγιση, οι λειτουργικές απαιτήσεις παριστάνονται ως περιπτώσεις χρήσης. Οι περιπτώσεις χρήσης είναι μια κεντρική έννοια στην Ενοποιημένη Προσέγγιση και όλα τα προϊόντα και οι εργασίες της ανάπτυξης λογισμικού αναφέρονται σε αυτές. Κάθε περίπτωση χρήσης τεκμηριώνεται με τη βοήθεια δομημένου κειμένου και διαγραμμάτων. Η τεκμηρίωση αυτή αποτελεί την είσοδο στην εργασία της ανάλυσης, όπου το λογισμικό διασπάται σε τμήματα (πακέτα), στα οποία αντιστοιχίζονται οι περιπτώσεις χρήσης. Για κάθε περίπτωση χρήσης, εντοπίζονται οι κλάσεις και η απαιτούμενη συνεργασία αυτών, ώστε να παραχθεί το επιθυμητό αποτέλεσμα. Στην ανάλυση διακρίνονται τρεις τύποι κλάσεων, οι κλάσεις οντοτήτων, οι συννοριακές και οι κλάσεις ελέγχου. Τέλος, προσδιορίζονται τα πεδία και οι συσχετίσεις μεταξύ των κλάσεων. Οι εργασίες της ανάλυσης δεν αφορούν ολόκληρη την εφαρμογή λογισμικού, αλλά μόνο το τμήμα αυτής με το οποίο ο κατασκευαστής επιλέγει να ασχοληθεί σε έναν κύκλο ανάπτυξης. Με αυτό τον τρόπο γίνεται ευκολότερη η διαχείριση της ανάπτυξης και ελαχιστοποιείται το ρίσκο αλλά και οι επιπτώσεις εσφαλμένων επιλογών. Η προσοχή κατά την ανάλυση εστιάζεται στα ποιοτικά χαρακτηριστικά της υλοποίησης των ροών των περιπτώσεων χρήσης και όχι σε κατασκευαστικές λεπτομέρειες, μη λειτουργικές απαιτήσεις και βελτιστοποιήσεις, προβλήματα με τα οποία καταπιάνεται η σχεδίαση. Σε πολλές περιπτώσεις είναι αναπόφευκτη η σύγκριση της δομημένης με την αντικειμενοστρεφή φιλοσοφία, όχι μόνο για να τεκμηριωθεί ποια είναι «καλύτερη», αλλά και για να γίνει περισσότερο συνειδητή η δεύτερη, ως νεότερη. Σε τελική ανάλυση, δηλαδή στο επίπεδο αυτού καθαυτού του πηγαίου κώδικα όπου καταλήγει οποιαδήποτε προσέγγιση

ανάλυσης και σχεδίασης, είναι ο δομημένος προγραμματισμός που χρησιμοποιείται. Η δομημένη φιλοσοφία χρησιμοποιεί την έννοια της ιεραρχικής δομής τόσο στο μακρο-επίπεδο (αρχιτεκτονική) όσο και στο μικρο-επίπεδο (πηγαίος κώδικας) του λογισμικού. Η αντικειμενοστρεφής φιλοσοφία αντιμετωπίζει με άλλο τρόπο το μακρο-επίπεδο, χρησιμοποιεί όμως (καλά κελυφοποιημένο) το δομημένο πηγαίο κώδικα. Μπορείτε να περιπλανηθείτε στη βιβλιογραφία όπου θα βρείτε εκτενείς συγκριτικές αναφορές, όχι απαραίτητα όλες συγκλίνουσες. Ένα είναι το σίγουρο, η αντικειμενοστρεφής τεχνολογία δεν είναι πανάκεια, όπως αποδείχθηκε ότι δεν ήταν και η δομημένη στην εποχή της.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- Ø « Uml toolkit » by Hans – Erik Eiksson and Magnus Penker, Wiley Computer Publishing
- Ø « Uml Distilled : Applying the Standard Object Modeling Language » by Martin Fowler, Kendall Scott and Ivan Jacobson, Addison – Wesley Object Technology Series
- Ø Uml Weekend Crash Course, Thomas A. Pender, Wiley Publishing Inc., Indiana
- Ø Β. Τ. Ταμπακάς , « Εισαγωγή στις Βάσεις Δεδομένων» , Διδακτικές Σημειώσεις για το τμήμα Λογιστικής, ΑΤΕΙ Πάτρας, 2002
- Ø Γιακουμάκης Ε., Τεχνολογία Λογισμικού, Τόμος Α, Εκδόσεις Σταμούλη, Αθήνα, 1993
- Ø Γιακουμάκης Ε., «Τεχνολογία Λογισμικού» , Τόμος Β, Εκδόσεις Σταμούλη, Αθήνα, 1994
- Ø Κάβουρας Ι.Κ., « Οργάνωση συστημάτων ενός χρήστη », Εκδόσεις Κλειδάριθμος, Αθήνα, 1991
- Ø Γιαλούρης Κ.- Γκιμπερίτης Ε.- Κόμης Β.- Σιδερίδης Α.- Σταθόπουλος Κ., « Εφαρμογές Πληροφορικής – Υπολογιστών Α' , Β' , Γ' ενιαίου λυκείου », ΟΕΔΒ, Αθήνα , 2000
- Ø Βλαχοπούλου Μ., « e-marketing », Εκδόσεις Rosili, Αθήνα, 1999

INTERNET

www.jeckle.de/umltool.htm

www.cs.teilar.gr

www.uom.gr

www.dmst.aueb.gr

www.cs.ua.gr

www.rational.com

www.unipi.gr