



ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΠΑΤΡΑΣ

ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ

ΤΜΗΜΑ: ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΥ ΣΧΕΔΙΑΣΜΟΥ & ΠΛΗΡΟΦΟΡΙΑΚΩΝ
ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΘΕΜΑ:

“ΕΠΙΣΚΟΠΗΣΗ ΜΕΘΟΔΟΛΟΓΙΩΝ ΑΝΑΠΤΥΞΗΣ ΠΛΗΡΟΦΟΡΙΑΚΩΝ
ΣΥΣΤΗΜΑΤΩΝ”

ΣΠΟΥΔΑΣΤΕΣ:
ΓΚΟΥΒΕΡΗΣ ΔΗΜΗΤΡΙΟΣ
ΖΙΩΛΙΟΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

ΕΙΣΗΓΗΤΗΣ: ΠΑΠΑΪΩΑΝΝΟΥ ΒΑΪΟΣ

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΕΧΟΜΕΝΑ	1
ΚΕΦΑΛΑΙΟ 1 ΕΙΣΑΓΩΓΗ	5
1.1 ΠΡΟΛΟΓΟΣ.....	5
1.2 ΑΝΑΠΤΥΞΗ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ.....	5
1.3 Έναρξη συστήματος	6
1.4 Ανάλυση Συστήματος.....	7
1.5 Σχεδιασμός συστήματος.....	7
1.6 Εφαρμογή συστήματος.....	8
1.7 Ο ρόλος του αναλυτή συστήματος.....	9
ΚΕΦΑΛΑΙΟ 2 ΑΝΑΛΥΣΗ ΣΥΣΤΗΜΑΤΩΝ	11
2.1 Εισαγωγή.....	11
2.2 Μελέτη Σκοπιμότητας	12
2.2.1 Προσδιορισμός βασικών προβλημάτων και ευκαιριών	13
2.2.2 Διαπραγμάτευση βασικού μεγέθους του έργου.....	13
2.2.3 Σχέση ωφέλειας κόστους έργου.....	14
2.2.4 Ανάπτυξη βασικού προγράμματος και προϋπολογισμού	15
2.2.5 Κοινοποίηση του έργου.....	16
2.3 Εντοπισμός Προβλήματος	17
2.3.1 Κατανόηση της περιοχής του προβλήματος	18
2.3.2 Ανάλυση του προβλήματος και των ευκαιριών.....	18
2.3.3 Ανάλυση επιχειρηματικών διαδικασιών	19
2.3.4 Κατοχύρωση στόχων βελτίωσης συστημάτων	20
2.3.5 Αναβάθμιση ή αναπροσαρμογή του έργου.....	20
2.3.6 Κοινοποίηση συμπερασμάτων και προτεινόμενων λύσεων	21
2.4 Ανάλυση απαιτήσεων	22
2.4.1 Προσδιορισμός των απαιτήσεων του συστήματος	23
2.4.2 Προτεραιότητα στις απαιτήσεις των συστημάτων	24
2.4.3 Αναβάθμιση ή αναπροσαρμογή του αρχικού σχεδίου.....	25
2.4.4 Κοινοποίηση απαιτήσεων	26
2.4.5 Διαρκείς διαχείριση απαιτήσεων.....	26
2.5 ΛΟΓΙΚΟΣ ΣΧΕΔΙΑΣΜΟΣ.....	27
2.5.1 Δομή λειτουργικών απαιτήσεων	27
2.5.2 Διαμόρφωση τρότυπων λειτουργικών απαιτήσεων.....	28
2.5.3 Επικύρωση λειτουργικών απαιτήσεων	29
2.5.4 Καθορισμός αποδοχής δοκιμαστικών λογισμικών	29

2.6 ΑΝΑΛΥΣΗ ΑΠΟΦΑΣΕΩΝ	29
2.6.1 Προσδιορισμός υποψηφίων λύσεων.....	30
2.6.2 Ανάλυση υποψηφίων λύσεων.....	31
2.6.3 Σύγκριση υποψηφίων λύσεων.....	32
2.6.4 Αναβάθμιση επιχειρησιακού σχεδίου.....	32
2.6.5 Υπόδειξη λύσης συστήματος.....	33
2.7 Μοντελοποίηση Δεδομένων και Διαδικασιών	33
2.7.1 Οντότητες (Entities).....	35
2.7.2 Ιδιότητες (Attributes).....	36
2.7.3 Συσχετίσεις (Relationships).....	39
2.7.4 Σχεδιασμός της διαδικασίας μοντελοποίησης.....	46
2.7.5 Στρατηγικός σχεδιασμός συστημάτων.....	47
2.7.6 Process Modeling For business Process Redesign.....	47
2.7.7 Διαδικασία Μοντελοποίησης κατά τη διάρκεια της ανάλυσης συστημάτων.....	48
2.7.8 Μια μελλοντική ματιά στο σχεδιασμό συστήματος.....	48
2.7.9 Διερεύνηση και συλλογή πληροφοριών για τη διαδικασία Μοντελοποίησης.....	48
2.7.10 Computer-Aided Systems Engineering For Process Modeling.....	49
2.8 Business Drivers	49
ΚΕΦΑΛΑΙΟ 3 ΣΧΕΔΙΑΣΜΟΣ ΣΥΣΤΗΜΑΤΩΝ	51
3.1 Εισαγωγή	51
3.1 Προσεγγίσεις Σχεδιασμού Συστημάτων	51
3.1.1 Πρότυπες – Οδηγημένες Προσεγγίσεις.....	52
3.1.2 Σύγχρονος δομημένος σχεδιασμός.....	53
3.1.3 Αντικειμενοστραφείς σχεδιασμός.....	54
3.2 Στόχοι του σχεδιασμού συστημάτων	54
3.2.1 Design the Application Architecture.....	56
3.2.2 Σχεδιασμός της βάσης δεδομένων.....	58
3.2.3 Σχεδιασμός του περιβάλλοντος διεπαφής του συστήματος.....	58
3.2.4 Διαχωρισμός προδιαγραφών σχεδιασμού σε πακέτα.....	60
3.2.5 Αναβάθμιση του σχεδίου του έργου.....	61
3.2.6 Αγορά λογισμικού.....	61
3.3 Αρχιτεκτονική και μοντελοποίηση εφαρμογών	62
3.3.1 Διαγράμματα ροής φυσικών δεδομένων.....	63
3.3.1.1 Φυσικές διαδικασίες.....	63
3.3.1.2 Εξωτερικοί παράγοντες.....	64
3.3.1.3 Αποθήκευση δεδομένων.....	64
3.3.1.4 Ροή δεδομένων.....	65
3.3.2 Αρχιτεκτονική κατανεμημένων συστημάτων.....	66
3.3.2.1 Σύστημα διαχείρισης κατανεμημένων σχεσιακών βάσεων.....	66
3.3.2.2 Αρχιτεκτονικές επιφάνειες διεπαφής – Δεδομένα εισόδου, εξόδου και ενδιάμεσο λογισμικό.....	67
3.3.2.3 Αρχιτεκτονική διαδικασιών – Το περιβάλλον ανάπτυξης λογισμικού.....	67
3.3.3 Στρατηγικές αρχιτεκτονικής εφαρμογών για το σχεδιασμό των συστημάτων.....	67
3.3.3.1 Αρχιτεκτονική επιχειρηματικών εφαρμογών.....	68
3.3.3.2 Αρχιτεκτονική τακτικών εφαρμογών.....	68
3.3.4 Μοντελοποίηση της αρχιτεκτονικής των εφαρμογών πληροφοριακού συστήματος.....	69
3.3.4.1 Σχεδιασμός διαγραμμάτων ροής φυσικών δεδομένων.....	69
3.3.4.2 Αρχιτεκτονική δικτύων.....	70

3.3.4.3 Κατανομή δεδομένων και τεχνολογία	71
3.3.4.4 Κατανομή διαδικασιών και τεχνολογία	72
3.3.4.5 Ο διαχωρισμός ανθρώπου/μηχανής	72
3.4 Σχεδιασμός βάσεων δεδομένων	73
3.4.1 Οι έννοιες των βάσεων δεδομένων για τους αναλυτές συστημάτων	73
3.4.1.1 Πεδία	74
3.4.1.2 Εγγραφές	74
3.4.1.3 Αρχεία και πίνακες	75
3.4.1.4 Βάσεις δεδομένων	76
3.4.2 Προϋποθέσεις για τον σχεδιασμό των βάσεων δεδομένων – Κανονικοποίηση	80
3.4.3 Συμβατικός σχεδιασμός αρχείων	81
3.4.4 Σύγχρονος σχεδιασμός βάσεων δεδομένων	82
3.4.4.1 Στόχοι και προϋποθέσεις του σχεδιασμού βάσεων δεδομένων	82
3.4.4.2 Το σχήμα της βάσης δεδομένων	83
3.4.4.3 Ακεραιότητα δεδομένων και σχέσεων	86
3.4.4.4 Ρόλοι	88
3.4.4.5 Κατανομή και αναπαραγωγή βάσεων δεδομένων	88
3.4.4.6 Πρωτότυπα βάσεων δεδομένων	90
3.4.4.7 Προγραμματισμός χωρητικότητας βάσεων δεδομένων	90
3.4.4.8 Δημιουργία δομής βάσης δεδομένων	91
ΚΕΦΑΛΑΙΟ 4 ΜΕΘΟΔΟΛΟΓΙΕΣ ΑΝΑΠΤΥΞΗΣ (ΠΑΡΑΔΕΙΓΜΑ OWL)	92
4.1 Δομημένη προσέγγιση	92
4.1.1 Μεθοδολογία STRADIS	93
4.1.2 Μεθοδολογία Yourdon	93
4.1.3 Information Engineering	94
4.1.4 Μεθοδολογία MERISE	95
4.1.5 Μεθοδολογία SSADM	95
4.1.6 Μεθοδολογία πολλαπλής θεώρησης	96
4.1.7 Κατασκευή προτύπου (prototyping)	96
4.1.8 μειονεκτήματα δομημένης προσέγγισης	97
4.2 Αντικειμενοστρεφής προσέγγιση (Object Oriented Methodology)	98
4.2.1 Coad & Yourdon	99
4.2.2 OMT (Object Modelling Technique)	99
4.2.3 OOAD (Object Oriented Analysis & Design)	100
4.2.4 OBA (Object Behaviour Analysis)	101
4.2.5 Rapid Application Development	101
4.2.6 UML	102
4.3 Οντολογίες	103
4.3.1 Ταξινόμηση Οντολογιών	104
4.3.2 Εργαλεία Ανάπτυξης Οντολογιών	106
4.3.3 Γλώσσες Οντολογιών	108
4.3.3.1 SHOE	110
4.3.3.2 Ontoligua	110
4.3.3.3 OCML	111
4.4 OWL	111
4.4.1 Μελέτη περίπτωσης : Ανάπτυξη Συστήματος ΙΚΑ	116
4.4.2 Ανάπτυξη οντολογίας για το διάγραμμα Οντοτήτων Σχέσεων (ER)	119
4.4.3 Το διάγραμμα της γλώσσας OWL	122
4.4.4 Εισαγωγή στοιχείων στο Protégé	123

ΚΕΦΑΛΑΙΟ 5 ΣΥΜΠΕΡΑΣΜΑΤΑ 128

ΒΙΒΛΙΟΓΡΑΦΙΑ..... 130

Κεφάλαιο 1 ΕΙΣΑΓΩΓΗ

1.1 ΠΡΟΛΟΓΟΣ

Θέμα της πτυχιακής εργασίας είναι η ανάπτυξη πληροφοριακών συστημάτων και σκοπός της είναι να γίνουν κατανοητές οι διάφορες μεθοδολογίες που υπάρχουν, οι πιο διαδεδομένοι τρόποι ανάλυσης και σχεδιασμού των πληροφοριακών συστημάτων που εφαρμόστηκαν παλαιότερα αλλά και αυτές που επικρατούν σήμερα. Επίσης γίνεται μια προσπάθεια να γίνει κατανοητός και ευδιάκριτος ο ρόλος αλλά και το στάδιο στο οποίο ανήκουν οι χρήστες, οι αναλυτές, οι σχεδιαστές των συστημάτων και τα μέλη της διοίκησης που εμπλέκονται στο σχεδιαζόμενο εγχείρημα. Παράλληλα γίνεται αναφορά στις μεθοδολογίες ανάλυσης και σχεδιασμού με βάση τόσο την κατηγορία που ανήκει η κάθε μεθοδολογία όσο και την χρονολογική σειρά με την οποία αναπτύχθηκε. Τέλος αναπτύσσεται παράδειγμα για την περαιτέρω κατανόηση της ανάλυσης των πληροφοριακών συστημάτων με διαγράμματα και σχεδιαστικά βοηθήματα ανάλυσης.

1.2 ΑΝΑΠΤΥΞΗ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Για την καλύτερη κατανόηση του όρου αυτού δίνεται η περιγραφή μιας απλής διαδικασίας ανάπτυξης συστήματος η οποία αποτελείται από τέσσερις φάσεις: Οι περισσότεροι οργανισμοί έχουν μία επίσημη διαδικασία ανάπτυξης συστήματος συνθέτοντας ένα τυποποιημένο σύνολο διαδικασιών ή βημάτων τα οποία ακολουθούνται σε κάθε αναπτυξιακό έργο συστημάτων (*System Development project*). Ενώ αυτές οι διαδικασίες μπορεί να ποικίλουν από οργανισμό σε οργανισμό υπάρχει ένα κοινό χαρακτηριστικό : οι περισσότεροι

οργανισμοί ακολουθούν τον ίδιο τρόπο λύσης του προβλήματος. Αυτή η προσέγγιση ενσωματώνει τα ακόλουθα τέσσερα βήματα όπως φαίνονται στο σχήμα 1.1

Απλή διαδικασία ανάπτυξης συστήματος	Γενικά βήματα επίλυσης προβλημάτων
1. Έναρξη συστήματος	<ul style="list-style-type: none"> • Προσδιορισμός προβλήματος (σχεδιασμός της λύσης του προβλήματος)
2. Ανάλυση συστήματος	<ul style="list-style-type: none"> • Ανάλυση και κατανόηση του προβλήματος • Προσδιορισμός απαιτήσεων και προσδοκιών
3. Σχεδιασμός συστήματος	<ul style="list-style-type: none"> • Προσδιορισμός εναλλακτικής λύσης και επιλογή της καλύτερης
4. Εφαρμογή συστήματος	<ul style="list-style-type: none"> • Σχεδιασμός της επιλεγμένης λύσης • Εφαρμογή της επιλεγμένης λύσης • Αξιολόγηση των αποτελεσμάτων (εάν το πρόβλημα δεν έχει λυθεί επιστρέφουμε στις φάσεις 1 ή 2 αναλόγως)

Σχήμα 1.1

Ας δούμε εν συντομία πως περιγράφετε η απλή αυτή διαδικασία ανάπτυξης συστήματος έτσι ώστε να γίνει καλύτερα κατανοητό το κάθε βήμα στη διαδικασία αυτή. Πλαμβάνοντας υπόψη ένα πρόβλημα το οποίο πρέπει να λυθεί, τι θα γίνει κατά τη διάρκεια της έναρξης συστήματος, ανάλυσης, σχεδιασμού και εφαρμογής; Επίσης ποιος θα αναμιχθεί σε κάθε φάση; Είναι ερωτήματα που απαντώνται πιο κάτω.

1.3 Έναρξη συστήματος

Η ανάπτυξη πληροφοριακών συστημάτων είναι συνήθως πολύπλοκη. Απαιτεί σημαντικό χρόνο, προσπάθεια και οικονομικούς πόρους. Τα προβλήματα

που πρέπει να λυθούν είναι συχνά πολύ αόριστα, που σημαίνει ότι η αρχική προβλεπόμενη λύση μπορεί να είναι πρόωρη. Κατά την εισαγωγή του συστήματος καθιερώνεται το πεδίο, το σχέδιο της λύσης του προβλήματος και ο προϋπολογισμός.

1.4 Ανάλυση Συστήματος

Το επόμενο βήμα στην ανάπτυξη ενός συστήματος είναι η ανάλυση του. Έχει ως σκοπό να δώσει στο προσωπικό μια ποιο κατανοητή "πλευρά" του προβλήματος και των αναγκών. Απαιτείτε δουλεία από τους χρήστες του συστήματος για να καθοριστούν σαφέστερα οι απαιτήσεις και οι προσδοκίες για κάθε νέο σύστημα που επρόκειτο να αναπτυχθεί. Η ανάλυση μπορεί να αποκαλύψει την ανάγκη για αναθεώρηση του επιχειρηματικού σχεδίου η και των στόχων. Αυτό έχει ως σκοπό και την αλλαγή του προϋπολογισμού. Τέλος από όλα αυτά θα συμπεράνουμε αν πρέπει να συνεχίσουμε στην επόμενη φάση ή όχι.

1.5 Σχεδιασμός συστήματος

Εφόσον στην ανάλυση συστήματος έγινε κατανοητή το πρόβλημα αλλά και οι απαιτήσεις για το νέο πληροφοριακό σύστημα μπορούμε να προχωρήσουμε στην σχεδίαση του. Κατά την διάρκεια του σχεδιασμού ωστόσο θα χρειαστεί να αναζητηθούν εναλλακτικές λύσεις σε τεχνικό επίπεδο. Σπάνια υπάρχει και εδώ μόνο μία λύση για κάθε πρόβλημα. Εφόσον επιλεγεί το καλύτερο δυνατό τότε αναπτύσσονται τα τεχνικά σχεδιαγράμματα και οι προδιαγραφές που χρειάζονται για την τελική εφαρμογή. Τα τεχνικά σχεδιαγράμματα και οι προδιαγραφές αυτές θα χρησιμοποιηθούν στην εφαρμογή της απαραίτητης βάσης δεδομένων, των προγραμμάτων, (*user interfaces) και των δικτύων των πληροφοριακών συστημάτων.

που πρέπει να λυθούν είναι συχνά πολύ αόριστα, που σημαίνει ότι η αρχική προβλεπόμενη λύση μπορεί να είναι πρόωρη. Κατά την εισαγωγή του συστήματος καθιερώνεται το πεδίο, το σχέδιο της λύσης του προβλήματος και ο προϋπολογισμός.

1.4 Ανάλυση Συστήματος

Το επόμενο βήμα στην ανάπτυξη ενός συστήματος είναι η ανάλυση του. Έχει ως σκοπό να δώσει στο προσωπικό μια ποιο κατανοητή “πλευρά” του προβλήματος και των αναγκών. Απαιτείτε δουλεία από τους χρήστες του συστήματος για να καθοριστούν σαφέστερα οι απαιτήσεις και οι προσδοκίες για κάθε νέο σύστημα που επρόκειτο να αναπτυχθεί. Η ανάλυση μπορεί να αποκαλύψει την ανάγκη για αναθεώρηση του επιχειρηματικού σχεδίου η και των στόχων. Αυτό έχει ως σκοπό και την αλλαγή του προϋπολογισμού. Τέλος από όλα αυτά θα συμπεράνουμε αν πρέπει να συνεχίσουμε στην επόμενη φάση ή όχι.

1.5 Σχεδιασμός συστήματος

Εφόσον στην ανάλυση συστήματος έγινε κατανοητή το πρόβλημα αλλά και οι απαιτήσεις για το νέο πληροφοριακό σύστημα μπορούμε να προχωρήσουμε στην σχεδίαση του. Κατά την διάρκεια του σχεδιασμού ωστόσο θα χρειαστεί να αναζητηθούν εναλλακτικές λύσεις σε τεχνικό επίπεδο. Σπάνια υπάρχει και εδώ μόνο μία λύση για κάθε πρόβλημα. Εφόσον επιλεγεί το καλύτερο δυνατό τότε αναπτύσσονται τα τεχνικά σχεδιαγράμματα και οι προδιαγραφές που χρειάζονται για την τελική εφαρμογή. Τα τεχνικά σχεδιαγράμματα και οι προδιαγραφές αυτές θα χρησιμοποιηθούν στην εφαρμογή της απαραίτητης βάσης δεδομένων, των προγραμμάτων, (*user interfaces) και των δικτύων των πληροφοριακών συστημάτων.

1.6 Εφαρμογή συστήματος

Το τελικό βήμα στην απλή διαδικασία ανάπτυξης συστήματος είναι η εφαρμογή του. Στην εφαρμογή συστήματος κατασκευάζεται το νέο πληροφοριακό σύστημα και το τίθεται σε λειτουργία. Κατά τη διάρκεια αυτής της φάσης δοκιμάζεται το νέο λειτουργικό και το λογισμικό του συστήματος ξεχωριστά. Όταν το σύστημα δοκιμαστεί και λειτουργεί κανονικά μετά μπαίνει σε λειτουργία.

Μετά την ανάλυση των παραπάνω φάσεων, το ερώτημα που είναι χρήσιμο να απαντηθεί είναι ποιοι συμμετέχουν σε αυτό το σύστημα; Αυτοί που συμμετέχουν μπορούν να ταξινομηθούν σε τέσσερις κατηγορίες (όπως φαίνεται ποιο κάτω). Αξίζει να σημειωθεί ότι ο καθένας από αυτούς που συμμετέχουν στην ανάπτυξη των πληροφοριακών συστημάτων βλέπει διαφορετικά το ίδιο πληροφοριακό σύστημα.

A) Ιδιοκτήτες Συστήματος: Πληρώνουν για να σχεδιαστεί το σύστημα και να λειτουργήσει. Θέτουν το σκοπό και τις προτεραιότητες για το σύστημα. Το βλέπουν από την πλευρά του κόστους και των κερδών. Τους ενδιαφέρει να λυθούν τυχόν προβλήματα και να ανακαλύψουν ευκαιρίες.

B) Χρήστες Συστήματος: Καθορίζουν τις ανάγκες και τις προσδοκίες τις επιχείρησης για το νέο σύστημα. Ως εκ τούτου βλέπουν το πληροφοριακό σύστημα από την πρακτική του πλευρά, κατά πόσο είναι χρήσιμο στη δουλειά τους, αν είναι εύκολο στη χρήση και εύκολο στο να το μάθει κανείς.

Γ) Σχεδιαστές Συστήματος: Μεταφράζουν τις απαιτήσεις τις επιχείρησης σε ψηφιακή τεχνική λύση. Κατασκευάζουν ένα σχεδιάγραμμα που θα καθοδηγήσει στη κατασκευή του τελικού συστήματος.

Δ) Κατασκευαστές Συστήματος: Κατασκευάζουν, επεκτείνουν και διατηρούν το πληροφοριακό σύστημα. Τους ενδιαφέρει να λειτουργεί το λειτουργικό και λογισμικό κομμάτι κατά την εφαρμογή του συστήματος.

1.7 Ο ρόλος του αναλυτή συστήματος

Όπως είδαμε και πιο πάνω οι ιδιοκτήτες συστημάτων, οι σχεδιαστές και οι κατασκευαστές έχουν διαφορετικές προοπτικές αλλά και γνώσεις για την κατασκευή και χρήση του κάθε πληροφοριακού συστήματος. Μερικοί ενδιαφέρονται για γενικότητες, ενώ άλλοι επικεντρώνουν σε λεπτομέρειες. Μερικοί είναι τεχνικοί ενώ άλλοι δεν έχουν ιδέα και άποψη πάνω σε αυτό το κομμάτι του πληροφοριακού συστήματος. Έτσι εμφανίζεται ένα κενό επικοινωνίας που υπήρχε πάντα ανάμεσα σε αυτούς που αναζητούν επιχειρησιακές λύσεις βασισμένες σε υπολογιστή και σε αυτούς που κατανοούν την τεχνολογία πληροφοριών. Οι αναλυτές συστημάτων έχουν ως σκοπό να γεφυρώσουν το κενό αυτό.

Για τους ιδιοκτήτες συστημάτων και τους χρήστες οι αναλυτές συστημάτων προσδιορίζουν και επικυρώνουν τα επιχειρησιακά προβλήματα και τις ανάγκες. Για τους σχεδιαστές και τους κατασκευαστές συστημάτων οι αναλυτές σιγουρεύονται ότι οι τεχνική λύση εκπληρώνει της ανάγκες τις επιχείρησης και ενσωματώνετε σωστά μέσα σε αυτή. Με άλλα λόγια, οι αναλυτές συστημάτων διευκολύνουν την ανάπτυξη πληροφοριακών συστημάτων μέσω της αλληλεπίδρασης με όλους τους συμμετέχοντες.

Οι αναλυτές συστημάτων κατέχουν επιχειρησιακές γνώσεις και τεχνικές. Μελετούν επιχειρησιακά προβλήματα και ευκαιρίες και μετά τα μετατρέπουν σε απαιτήσεις στην προδιαγραφή των πληροφοριακών συστημάτων όπου και θα εφαρμοστούν τεχνικούς ή ακόμα και από προγραμματιστές. Οι υπολογιστές και τα πληροφοριακά συστήματα έχουν αξία στην επιχείρηση μόνο εάν βοηθούν στην λύση προβλημάτων ή επιφέρουν βελτιώσεις,

Οι αναλυτές συστημάτων ξεκινούν τις αλλαγές μέσα στην επιχείρηση. Κάθε καινούργιο σύστημα αλλάζει την λειτουργία της επιχείρησης. Οι καλύτεροι αναλυτές συστημάτων αλλάζουν ριζικά την επιχείρηση που εργάζονται, παρέχοντας πληροφορίες που μπορούν να χρησιμοποιηθούν ως ανταγωνιστικό πλεονέκτημα , βρίσκοντας νέες αγορές και υπηρεσίες, και ακόμα αλλάζοντας δραματικά και βελτιώνοντας τον τρόπο με τον οποίο η επιχείρηση δουλεύει.

Έχοντας μια καλύτερη εικόνα για τους αναλυτές συστημάτων μπορούμε να πούμε πως βασικά προτείνουν λύσεις για τυχόν προβλήματα που υπάρχουν ή θα παρουσιαστούν. Ο όρος *προβλήματα* περιγράφει πολλές καταστάσεις οι ποιο συνηθισμένες όμως είναι προβλήματα είτε υπαρκτά είτε προσδοκώμενα που απαιτούν διορθωτικές ενέργειες. Ευκαιρίες να βελτιωθεί μια κατάσταση ακόμα και χωρίς να υπάρχει παράπονο για αυτήν και οδηγίες για να αλλάξει μια κατάσταση άσχετα από το αν κάποιος έχει παράπονα από την μέχρι τώρα λειτουργία ή όχι.

Η αναφορά των συμμετεχόντων στην ανάπτυξη ενός πληροφοριακού συστήματος γίνεται με σκοπό, να γίνουν ευδιάκριτοι οι ρόλοι του καθενός στην κάθε φάση ανάπτυξης πληροφοριακού συστήματος. Παρότι στην πραγματικότητα μπορεί να συμμετέχουν κάποιοι σε περισσότερες φάσεις, έχουν κυρίαρχο ρόλο σε μια από αυτές.

Κεφάλαιο 2 ΑΝΑΛΥΣΗ ΣΥΣΤΗΜΑΤΩΝ

2.1 Εισαγωγή

Τι είναι η ανάλυση συστημάτων:

Ανάλυση συστημάτων είναι η μελέτη ενός συστήματος και των στοιχείων που το συνθέτουν και έχει σαν προϋπόθεση τον σχεδιασμό νέου (θέτει τις προδιαγραφές για ένα νέο και βελτιωμένο σύστημα). Γενικότερα είναι μία τεχνική που λύνει προβλήματα αποσυνθέτοντας τα στοιχεία του συστήματος και αναλύοντας τα λεπτομερώς έχοντας ως στόχο την μελέτη του κάθε στοιχείου ξεχωριστά. Έτσι μπορούμε να δούμε πόσο αποδίδει το καθένα στοιχείο και το αν είναι μέσα στους στόχους μας ή όχι.

Ξεφεύγοντας από αυτόν τον κλασικό ορισμό της ανάλυσης συστημάτων σε κάτι πιο σύγχρονο, η ανάλυση συστημάτων είναι ένας όρος που επιλεκτικά (συλλογικά) περιγράφει τις πρώτες φάσεις της ανάπτυξης του συστήματος. Πρέπει να αναφερθεί ότι δεν υπάρχει ένας αποδεκτός όρος για την ανάλυση συστήματος παγκοσμίως. Στην πραγματικότητα δεν υπάρχει ακόμα παγκόσμια συμφωνία για το πότε τελειώνει η ανάλυση ενός πληροφοριακού συστήματος και το πότε αρχίζει η σχεδίαση του.

Έχοντας ξεκαθαρίσει την διαδικασία ανάπτυξης ενός πληροφοριακού συστήματος και την γενική έννοια της ανάλυσης, σε αυτό το κεφάλαιο γίνεται η περιγραφή των πέντε βασικών βημάτων της ανάλυσης συστημάτων, τα οποία είναι τα εξής:

- Μελέτη Σκοπιμότητας
- Ανάλυση Προβλήματος
- Ανάλυση Απαιτήσεων
- Λογικό Μοντέλο
- Ανάλυση Αποφάσεων

2.2 Μελέτη Σκοπιμότητας

Η φάση της μελέτης σκοπιμότητας είναι η πρώτη φάση μιας κλασσικής διαδικασίας ανάπτυξης συστήματος. Εδώ γίνεται η πρώτη επαφή του χρήστη με έναν ειδικό αναλυτή, για να γίνει η πρώτη περιγραφή του προβλήματος, που συνήθως είναι ελλιπής ή λανθασμένη. Αυτό έχει ως αποτέλεσμα την παρερμηνεία σημαντικών παραμέτρων, αφού και ο ίδιος ο χρήστης δεν γνωρίζει ακριβώς το πρόβλημα.

Ο σκοπός της μελέτης σκοπιμότητας, είναι ο ακριβής προσδιορισμός του μεγέθους του προβλήματος, η στρατηγική ανάπτυξης, το πρόγραμμα, οι πόροι και ο προϋπολογισμός κόστους. Εφόσον έχει γίνει ο προσδιορισμός του προβλήματος πρέπει να εξεταστεί αν η μελέτη σκοπιμότητας είναι απαραίτητη. Ο πρώτος παράγοντας που εξετάζεται είναι το μέγεθος του έργου, αφού η μελέτη σκοπιμότητας χρησιμοποιείται μόνο σε μεσαία και μεγάλα έργα, με υψηλά κόστη και μεγάλα περιθώρια χρόνου. Ο δεύτερος παράγοντας είναι ο τεχνολογικός και η μελέτη αυτή γίνεται σε έργα που απαιτούν εξειδικευμένη τεχνολογία. Τέλος αυτή η μελέτη είναι απαραίτητη για έργα αμφίβολης αναγκαιότητας και έργα οποιουδήποτε μεγέθους που δεν έχουν οριστεί με σαφήνεια, με σκοπό τον καλύτερο προσδιορισμό του έργου άρα και την καλύτερη λήψη αποφάσεων.

Για να αντιληφθούμε όμως καλύτερα σε τι πραγματικά χρησιμεύει η μελέτη σκοπιμότητας, καλό είναι να περιγράψουμε τα περιεχόμενα της, που περιγράφουν αναλυτικότερα την διαδικασία σαν μια σειρά από βήματα:

- Προσδιορισμός βασικών προβλημάτων και ευκαιριών
- Διαπραγμάτευση βασικού μεγέθους του έργου
- Σχέση ωφέλειας κόστους έργου
- Ανάπτυξη βασικού προγράμματος και προϋπολογισμού
- Κοινοποίηση του έργου

2.2.1 Προσδιορισμός βασικών προβλημάτων και ευκαιριών

Ένα από τα βασικότερα βήματα αυτής της φάσης είναι να κατοχυρωθούν αρχικά τα βασικά προβλήματα, οι ευκαιρίες και οι κατευθύνσεις που απασχολούν το έργο. Το κάθε πρόβλημα, η κάθε ευκαιρία και οι κατευθύνσεις εξετάζονται βάση της σημαντικότητας του έργου, του ρεαλισμού, των ευκαιριών που παρουσιάζονται και των προτεραιοτήτων. Οποιαδήποτε λεπτομερή ανάλυση δεν σχετίζεται με αυτή την φάση, θα ήταν όμως χρήσιμο να καταγραφούν όλοι οι περιορισμοί του έργου όπως για παράδειγμα χρονικά όρια, μέγιστη χρηματοδότηση, τεχνολογία κ.α.

Σε αυτό το βήμα συμμετέχουν ο αναλυτής συστημάτων ή ο διαχειριστής έργου. Οι υπόλοιποι συμμετέχοντες είναι οι ιδιοκτήτες συστημάτων συμπεριλαμβανομένων και των αποκλειστικών χρηματοδοτών και του υψηλά ιστάμενου προσωπικού που θα χρηματοδοτούν και θα στηρίζουν το έργο. Το αποτέλεσμα αυτού του βήματος είναι η μια αρχική δήλωση των προβλημάτων, των ευκαιριών και των κατευθύνσεων που έχουν εντοπιστεί. Οι καταθέσεις αυτές των προβλημάτων αρχειοθετούνται για μετέπειτα χρήση στο έργο.

2.2.2 Διαπραγμάτευση βασικού μεγέθους του έργου

Αναλόγως το μέγεθος κάθε επιχείρησης καθορίζονται και τα όρια σε κάθε έργο. Το εύρος του έργου μπορεί να αλλάξει κατά την διάρκεια της μελέτης του, παρόλα αυτά ο αρχικός σχεδιασμός του έργου είναι αυτός που θα θέσει και τα όρια του. Όπως στο προηγούμενο βήμα έτσι και σε αυτό, συμμετέχει ο αναλυτής συστημάτων ή ο διαχειριστής έργου ενώ συμμετέχοντες είναι και οι ιδιοκτήτες του συστήματος.

Σε αυτό το βήμα χρησιμοποιούνται οι δηλώσεις προβλημάτων, οι οποίες αντλούνται από το αρχείο όπως είδαμε παραπάνω. Πρέπει να γίνει κατανοητό πως αυτά τα προβλήματα, οι ευκαιρίες και οι κατευθύνσεις, δίνουν τις βάσεις για να καθοριστεί το μέγεθος του έργου. Οι προτάσεις που γίνονται για το μέγεθος του έργου αρχειοθετούνται και αυτές για μελλοντική χρήση.

Το μέγεθος μπορεί να καθοριστεί ευκολότερα με την διαδικασία της τμηματοποίησης του πληροφοριακού συστήματος. Για παράδειγμα το μέγεθος του έργου μπορεί να καθοριστεί από τα παρακάτω :

- Οι τύποι δεδομένων που περιγράφουν το σύστημα εξαρτώνται από το είδος του συστήματος. Για παράδειγμα, σε ένα πληροφοριακό σύστημα πωλήσεων μπορεί να απαιτούνται στοιχεία όπως πελάτες, οδηγίες, προϊόντα, πωλήσεις κ.α.
- Όπως οι τύποι δεδομένων έτσι και οι επιχειρηματικές διαδικασίες εξαρτώνται από το είδος του συστήματος. Για παράδειγμα, σε ένα πληροφοριακό σύστημα πωλήσεων μπορεί να περιλαμβάνονται επιχειρηματικές διαδικασίες για διαχείριση πελατών, διαχείριση καταλόγου, εισαγωγή παραγγελίας, διαχείριση σχέσης πελατών κ.α.
- Το περιβάλλον διεπαφής του συστήματος με τον χρήστη και η τοποθεσία πρέπει να βρίσκεται, αλλάζουν ανάλογα με την ομάδα (αρμοδιότητα) που ανήκει. Για παράδειγμα, σε ένα πληροφοριακό σύστημα πωλήσεων μπορεί να περιλαμβάνονται πελάτες, αντιπροσώπους πωλήσεων, clerk και υπεύθυνους πωλήσεων, τοπικά γραφεία πωλήσεων, μισθοδοσία και περιοχή ελέγχου πληροφοριακού συστήματος.

Το βήμα αυτό ολοκληρώνεται με την συλλογή στοιχείων και τις συνελεύσεις. Είναι σύνηθες φαινόμενο, οι αναλυτές να συνδυάζουν το προηγούμενο βήμα με αυτό και το επόμενο, να τα αναλύουν σε μια συνέλευση καθώς το θεωρούν αρκετό για την συγκεκριμένη διαδικασία.

2.2.3 Σχέση ωφέλειας κόστους έργου

Σε αυτό το βήμα καλείται να απαντηθεί το ερώτημα αν αξίζει να προχωρήσουμε αυτό το έργο. Λόγω του αρχικού σταδίου που βρισκόμαστε δεν μπορούμε να γνωρίζουμε οικονομικά μεγέθη και κατά πόσο αυτά θα ικανοποιήσουν την επιχείρηση.

Ο αναλυτής συστημάτων ή ο διαχειριστής έργου που ηγείται αυτού του βήματος, βάση των γνώσεων και των εμπειριών του, αναλύει ζητήματα όπως: η

επίλυση προβλημάτων, η παρουσίαση ευκαιριών και η εκπλήρωση των κατευθύνσεων που είχαν τεθεί πάντα σε συνάρτηση με το αν το έργο πρέπει να συνεχιστεί. Η τελική απόφαση για την συνέχιση του έργου λαμβάνεται από κοινού, από τον ιδιοκτήτη του συστήματος, τους χρηματοδότες, τους υπεύθυνους επιχειρηματικών μονάδων και τους υπεύθυνους πληροφοριακών συστημάτων. Σε αυτό το στάδιο θα έχουμε την θετική ή την αρνητική εισήγηση για το εάν θα συνεχίσουμε το έργο, εκτός εάν τεθούν και άλλες εναλλακτικές λύσεις όπως η επαναδιαπραγμάτευση (αύξηση ή μείωση του μεγέθους του έργου).

2.2.4 Ανάπτυξη βασικού προγράμματος και προϋπολογισμού

Εφόσον το έργο εκτιμήθηκε πως αξίζει να συνεχιστεί μπορούμε να σχεδιάσουμε το έργο σε βάθος. Το αρχικό σχέδιο του έργου πρέπει να περιλαμβάνει τουλάχιστον τις παρακάτω προϋποθέσεις :

- Αρχικό πλάνο (οδηγός) το οποίο περιλαμβάνει το πρόγραμμα και τους διαθέσιμους πόρους του συγκεκριμένου έργου. Το πλάνο αυτό ανανεώνετε στο τέλος της κάθε φάσης του έργου και πολλές φορές λέγεται βασικό κατευθυντήριο πλάνο.
- Ένα λεπτομερές πλάνο και πρόγραμμα για την ολοκλήρωση της επόμενης φάσης (φάση ανάλυσης προβλήματος)

Σε αυτό το βήμα αρμόδιος είναι ο υπεύθυνος έργου. Οι περισσότεροι υπεύθυνοι έργου, για να έχουν μια ποιο εμπειριστατωμένη άποψη, προσπαθούν να συμπεριλάβουν όσους περισσότερους μπορούν από την ομάδα του έργου (ιδιοκτήτες συστημάτων, χρήστες, σχεδιαστές, προγραμματιστές). Όπως και πριν έτσι και τώρα θα πρέπει να παρθεί μια απόφαση για το αν θα συνεχίσουμε το έργο. Το ζητούμενο αυτού του βήματος είναι η βασική κατεύθυνση του έργου και του προγραμματισμού.

2.2.5 Κοινοποίηση του έργου

Έχοντας φτάσει στο τελευταίο βήμα, το έργο θα πρέπει να παρουσιαστεί και να υποστηριχθεί στην ομάδα καθοδήγησης (αρμόδια επιτροπή η οποία αποτελείται από επιχειρηματίες και υπεύθυνους συστημάτων οι οποίοι μελετούν και θέτουν προτεραιότητες στις προτάσεις έργων) έτσι ώστε να εγκριθεί. Στην ομάδα υποστήριξης γίνονται πολλές προτάσεις έργων και δίνεται έγκριση μία ή περισσότερων αναλόγως των προτεραιοτήτων που έχουν τεθεί. Οι περισσότερες επιχειρήσεις χρησιμοποιούν αυτές τις ομάδες καθοδήγησης για να εγκρίνουν και να παρακολουθούν την πρόοδο των έργων.

Η εναρκτήρια συνέλευση θα είναι ανοικτή σε όλο το προσωπικό και όχι μόνο στους άμεσα ενδιαφερόμενους ή τους υψηλά ιστάμενους έτσι ώστε να αποτρέπεται η παραπληροφόρηση. Ανεξάρτητα από το αν το έργο θέλει την έγκριση της επιτροπής, εξίσου σημαντικό είναι να γνωστοποιηθεί το έργο (στόχοι και προγραμματισμός) σε όλο το προσωπικό της επιχείρησης. Για να επιτευχθεί αυτό δημιουργείται μία δικτυακή εφαρμογή και για αυτό τον λόγο στην ομάδα έργου συμπεριλαμβάνεται ένας δικτυακός διαχειριστής. Η δικτυακή εφαρμογή ανοίγει μια πόρτα επικοινωνίας σε όλα τα νέα και τις καταγραφές (documentation) που γίνονται στο έργο. Σε αυτό το βήμα συμμετέχουν και οι χρηματοδότες μαζί με τον υπεύθυνο έργου.

Διαπροσωπικές σχέσεις και ικανότητες επικοινωνίας είναι τα απαραίτητα στοιχεία αυτού του βήματος. Επίσης και βασικές αρχές όπως πειθώ, πειστικότητα, επιχειρηματικό γράψιμο και δημόσιες ομιλίες είναι απαραίτητες.

Τέλος σε αυτή τη φάση οι συμμετέχοντες μπορεί να αποφασίσουν ότι το έργο δεν αξίζει να προταθεί, είναι επίσης πιθανό οι αρμόδια επιτροπή να αποφασίσει πως δεν είναι αρκετά σημαντικό (άλλες προτάσεις έργων πιο σημαντικές). Ακόμα μπορεί ο χρηματοδότης να μην υποστηρίξει το έργο. Σε κάθε μία από αυτές τις περιπτώσεις το έργο τερματίζεται εκεί, και έτσι έχει αναλωθεί λίγος χρόνος και προσπάθεια. Από την άλλη εάν οι ιδιοκτήτες συστημάτων και η αρμόδια επιτροπή συμφωνήσουν το έργο μπορεί να προχωρήσει στην φάση ανάλυσης προβλήματος.

2.3 Εντοπισμός Προβλήματος

Η φάση του εντοπισμού του προβλήματος δίνει την δυνατότητα στον αναλυτή να έχει μια καλύτερη εικόνα των προβλημάτων και των ευκαιριών. Σε αυτή την φάση απαντώνται τα εξής ερωτήματα : Αξίζει να λυθούν τα προβλήματα; Αξίζει να δημιουργηθεί νέο σύστημα; Σε άλλες μεθοδολογίες η φάση αυτή είναι γνωστή ως φάση μελέτης υπάρχοντος συστήματος, φάση λεπτομερείς έρευνας ή φάση εφικτής ανάλυσης.

Σπανίως μπορεί κάποιος να παρακάμψει την φάση ανάλυσης προβλήματος. Πρέπει πάντα να υπάρχει γνώση του υπάρχοντος συστήματος. Ο μόνος λόγος που μπορεί να επιταχυνθεί η διαδικασία αυτής της φάσης, είναι να έχει εφαρμοστεί κάποιο στρατηγικό πλάνο όπου δεν χρειάζεται η ανάλυση του τωρινού συστήματος παρά μόνο η κατανόηση του.

Ο σκοπός της φάσης της ανάλυσης προβλήματος είναι η μελέτη και η κατανόηση της περιοχής του προβλήματος. Για να γίνει η ανάλυση αυτών, πρέπει να βρεθούν οι ευκαιρίες και οι περιορισμοί. Μερικές μεθοδολογίες ενθαρρύνουν μια ποιο λεπτομερή κατανόηση του συστήματος καθώς επίσης και την προσεκτική καταγραφή των δεδομένων, χρησιμοποιώντας διαγράμματα ροής. Σε αυτή την φάση αρχικά εμπλέκονται οι ιδιοκτήτες συστημάτων και οι χρήστες συστημάτων.

Οι στόχοι της φάσης του εντοπισμού προβλήματος είναι:

- Κατανόηση της περιοχής του προβλήματος
- Ανάλυση του προβλήματος και των ευκαιριών
- Ανάλυση επιχειρησιακών διαδικασιών
- Κατοχύρωση στόχων βελτίωσης συστημάτων
- Αναβάθμιση ή αναπροσαρμογή του έργου
- Κοινοποίηση συμπερασμάτων και προτεινόμενων λύσεων

2.3.1 Κατανόηση της περιοχής του προβλήματος

Κατά την διάρκεια αυτής της φάσης, η ομάδα αρχικά προσπαθεί να μάθει το υπάρχον σύστημα. Ο καθένας από τους ιδιοκτήτες, χρήστες και αναλυτές του συστήματος αντιλαμβάνεται διαφορετικά το σύστημα, εντοπίζει διαφορετικές λεπτομέρειες, χρησιμοποιεί διαφορετικό λεξιλόγιο, έχει διαφορετικές αντιλήψεις και εκφράζει διαφορετικές γνώμες. Μια καλά οργανωμένη μελέτη μπορεί να αποδειχθεί αποκαλυπτική σε όλους τους τομείς. Είναι σημαντική η μελέτη και η κατανόηση του προβλήματος στην περιοχή όπου υπάρχουν τα επιχειρησιακά προβλήματα, οι ευκαιρίες, οι οδηγίες και οι περιορισμοί.

Αυτός ο στόχος θα οδηγηθεί από τον διευθυντή του προγράμματος, αλλά θα διευκολυνθεί από τον αναλυτή συστημάτων. Σε πολλές περιπτώσεις ένα άτομο και μόνο διαδραματίζει και τους δύο ρόλους, αλλά αν κριθεί απαραίτητο μπορούν να χρησιμοποιηθούν παραπάνω αναλυτές συστημάτων. Μια περιεκτική μελέτη για να είναι αντιπροσωπευτική θα πρέπει να περιλαμβάνει ιδιοκτήτες και χρήστες συστημάτων από όλες τις επιχειρησιακές μονάδες.

Είναι σπάνιο ένας χρήστης να μπορεί να αντιπροσωπεύει τα συμφέροντα όλων των χρηστών, για αυτό είναι εξαιρετικά σημαντικό να συμπεριλαμβάνονται αρκετοί χρήστες για την κάλυψη του πλήρους πεδίου της μελέτης του συστήματος.

Τα ζητούμενα κατά την φάση αυτή είναι η κατανόηση της περιοχής προβλήματος και του λεξιλογίου. Εφόσον έχει κατανοηθεί πλήρως η περιοχή προβλήματος πρέπει να καταγραφεί είτε σχεδιάζοντας το μοντέλο του συστήματος (μπορεί όμως να οδηγήσει σε ένα φαινόμενο που ονομάζεται «ανάλυση παράλυσης»), είτε χρησιμοποιώντας Building Blocks (ανάλυση του έργου σε υπομονάδες).

2.3.2 Ανάλυση του προβλήματος και των ευκαιριών

Πέρα από την κατανόηση του τωρινού συστήματος, η ομάδα του προγράμματος πρέπει να συνεργαστεί με του ιδιοκτήτες και τους χρήστες συστημάτων για να αναλύσει τα προβλήματα και τις ευκαιρίες.

Η ανάλυση του προβλήματος σε βάθος, παρουσιάζει δυσκολίες ειδικά για τους άπειρους αναλυτές συστημάτων. Εμπειρικά η πλειοψηφία των νέων αναλυτών προσπαθούν να λύσουν τα προβλήματα χωρίς εμπειριστωμένη ανάλυση. Συχνά θέτουν το κάθε πρόβλημα σε συνάρτηση με την λύση του, δηλαδή λένε «χρειαζόμαστε να...» ή «επιθυμούμε να...». Με αυτόν τον τρόπο δίνουν στο πρόβλημα μια και μόνο λύση. Από την άλλη για να είναι κάποιος αποτελεσματικός θα πρέπει να αναλύει σε βάθος το πρόβλημα πριν προχωρήσει στην λύση του, αναλύοντας τα αίτια κάθε αντιληπτού προβλήματος.

Αναλύεται το κάθε πιθανό πρόβλημα με την τεχνική *causes and effects* (αίτιου αιτιατού). Πρακτικά το αποτέλεσμα μπορεί να είναι σύμπτωση διαφορετικού, ποιο ριζικού ή ποιο βασικού προβλήματος. Αυτό το πρόβλημα μπορεί επίσης να αναπτυχθεί με την τεχνική του αίτιου αιτιατού, και να συνεχίζεται μέχρι την στιγμή που το αίτιο αιτιατό δεν θα αποτελεί σύμπτωση άλλων προβλημάτων. Η ανάλυση του αίτιου αιτιατού οδηγεί σε πραγματική κατανόηση του προβλήματος και δεν δίνει τόσο εμφανείς, αλλά χρήσιμες και δημιουργικές λύσεις.

Σε αυτόν το στόχο συμμετέχουν οι αναλυτές συστημάτων, παρόλα αυτά οι ιδιοκτήτες και οι χρήστες συστημάτων πρέπει να έχουν ενεργή συμμετοχή στη διαδικασία ανάλυσης αίτιου αιτιατού, αφού αυτοί είναι οι ειδικοί του προβλήματος. Οι σχεδιαστές και προγραμματιστές συνήθως δεν εμπλέκονται σε αυτή την διαδικασία, εκτός και αν κληθούν να αναλύσουν τεχνικά προβλήματα που μπορεί να υπάρχουν στο τωρινό σύστημα.

2.3.3 Ανάλυση επιχειρηματικών διαδικασιών

Αυτό το στάδιο είναι κατάλληλο μόνο για προγράμματα επανασχεδιασμού επιχειρησιακής διαδικασίας (*business process redesign*) ή αναπτυξιακά έργα συστημάτων. Η ανάλυση επιχειρηματικής διαδικασίας έχει άμεση εξάρτηση και από την πολιτική. Οι αναλυτές που εμπλέκονται πρέπει να επικεντρώνουν στις διαδικασίες και όχι στους ανθρώπους που τις εκτελούν.

Στην ανάλυση επιχειρηματικών διαδικασιών εμπλέκονται ένας ή περισσότεροι αναλυτές συστημάτων ή επιχειρηματικοί αναλυτές. Το ιδανικό θα ήταν οι αναλυτές να είναι έμπειροι, εκπαιδευμένοι ή να έχουν πιστοποίηση σε προγράμματα όπως το business process redesign. Οι υπόλοιποι συμμετέχοντες θα είναι μόνο οι ιδιοκτήτες συστημάτων και οι χρήστες. Οι αναλυτές επιχειρησιακών διαδικασιών θα πρέπει να αποφύγουν οποιοδήποτε πειρασμό να εστιάσουν σε λύσεις πληροφοριών τεχνολογίας μέχρι να επανασχεδιαστεί η επιχειρηματική διαδικασία για την μέγιστη αποδοτικότητα.

2.3.4 Κατοχύρωση στόχων βελτίωσης συστημάτων

Εφόσον έχουμε κατανοήσει τον σκοπό, τα προβλήματα και τις ευκαιρίες του συστήματος, τώρα μπορούμε να προσδιορίσουμε τους «στόχους» ανάπτυξης του συστήματος. Σκοπός αυτού του σταδίου είναι η καταχώρηση κριτηρίων βελτίωσης του συστήματος. Επίσης και η ανεύρεση περιορισμών που μπορεί να αποτρέπουν από την επίτευξη των βελτιώσεων που έχουμε θέσει ως στόχο.

Σε αυτή την φάση συμμετέχουν οι αναλυτές συστημάτων, καθώς επίσης οι ιδιοκτήτες και οι χρήστες που συμμετείχαν σε προηγούμενες φάσεις. Επειδή ακόμα δεν μας ενδιαφέρει το τεχνολογικό μέρος οι σχεδιαστές και οι προγραμματιστές (designers and builders) δεν εμπλέκονται. Οι αναλυτές και οι χρήστες θα πρέπει να προσδιορίσουν συγκεκριμένα τους στόχους ανάπτυξης του συστήματος.

2.3.5 Αναβάθμιση ή αναπροσαρμογή του έργου

Βάση του έργου και του αρχικού προϋπολογισμού από τη φάση προσδιορισμού του σκοπού, το πεδίο μπορεί να έχει αυξηθεί ή μειωθεί στο μέγεθος και την πολυπλοκότητα (συνήθως αυξάνεται). Για αυτό τον λόγο τώρα που πλησιάζουμε στην ολοκλήρωση της «φάσης ανάλυσης προβλήματος», θα πρέπει να γίνει επανεκτίμηση και αναπροσαρμογή στο έργο αναλόγως.

Σε αυτή την φάση συμμετέχει ο διευθυντής προγράμματος από κοινού με τους ιδιοκτήτες συστημάτων και ολόκληρη την ομάδα έργου. Οι αναλυτές και οι ιδιοκτήτες είναι τα άτομα «κλειδιά» για το στάδιο αυτό. Αυτοί θα είναι που θα εξετάσουν αν μπορούν να καλυφθούν όλοι οι στόχοι από το σύστημα. Αυτό συμβαίνει επειδή το νέο σύστημα μπορεί να είναι μεγαλύτερο από το αναμενόμενο και ως εκ τούτου θα πρέπει να μειωθεί το πεδίο για να τηρηθεί η προθεσμία. Σε αυτήν την περίπτωση οι ιδιοκτήτες συστημάτων αξιολογούν τους στόχους και τους βάζουν σε σειρά προτεραιότητας. Έτσι αν πρέπει να μειωθεί το πεδίο, οι αναλυτές θα ξέρουν ποιος είναι ο πιο σημαντικός στόχος και ποιος όχι.

2.3.6 Κοινοποίηση συμπερασμάτων και προτεινόμενων λύσεων

Όπως η φάση της μελέτης σκοπιμότητας έτσι και η φάση της ανάλυσης προβλήματος ολοκληρώνεται με το βήμα της κοινοποίησης. Πρέπει να γίνει κοινοποίηση των συμπερασμάτων και των προτάσεων στο προσωπικό της εταιρίας. Συμμετέχοντες σε αυτό το βήμα είναι ο υπεύθυνος έργου ο αποκλειστικός χρηματοδότης. Στην συνέλευση πρέπει επίσης να είναι παρόντες όλοι η ομάδα σχεδίου, συμπεριλαμβανομένων και των ιδιοκτητών, χρηστών, αναλυτών, σχεδιαστών και προγραμματιστών. Ως συνήθως η συνέλευση είναι ανοικτή σε κάθε ενδιαφερόμενο από το προσωπικό της εταιρίας. Επίσης εάν μια δικτυακή εφαρμογή είχε κατοχυρωθεί για το έργο θα πρέπει να διατηρείται κατά την διάρκεια της φάσης της ανάλυσης του προβλήματος για να βεβαιώσουμε την διαρκή κοινοποίηση της προόδου του έργου. Σκοπός σε αυτό το τελευταίο βήμα, είναι να διατηρηθεί μια διαρκής ενημέρωση για το σύστημα, σε όλη την φάση της ανάλυσης του προβλήματος. Αυτό το βήμα ξεκινά από το συμπέρασμα του αναβαθμισμένου έργου. Υπάρχουν αρκετοί τρόποι ενημέρωσης, όπως μια αναφορά, μια έκθεση, μια παρουσίαση κτλ.

Μετά τα συμπεράσματα θα πρέπει να επιλεγεί μία από τις πιο κάτω αποφάσεις :

- Έγκριση του σχεδίου ως έχει για να συνεχίσουμε στην φάση απαιτήσεων

- Προσαρμογή του σκοπού, του κόστους και του σχεδίου και έπειτα συνέχεια στη φάση απαιτήσεων
- Ακύρωση του σχεδίου είτε από έλλειψη αγαθών, είτε τα προβλήματα και οι ευκαιρίες δεν ήταν τόσο σημαντικά όσο αναμένονταν, είτε τα κέρδη από το νέο σύστημα δεν αναμένονται να ξεπεράσουν το κόστος.

Με την έγκριση από τους ιδιοκτήτες συστήματος, μπορούμε να προχωρήσουμε στο επόμενο βήμα που είναι η φάση των απαιτήσεων.

2.4 Ανάλυση απαιτήσεων

Σχεδόν όλοι οι αναλυτές χωρίς πείρα, κάνουνε ένα σημαντικό λάθος μετά την ολοκλήρωση της φάσης ανάλυσης του προβλήματος. Ο πειρασμός σε αυτό το σημείο είναι να αρχίσεις να ψάχνεις για εναλλακτικές λύσεις, ειδικότερα τεχνικές. Ένα από τα ποιο συχνά αναφερόμενα λάθη σε ένα νέο πληροφοριακό σύστημα, διευκρινίζεται στην φράση «σίγουρα το σύστημα λειτουργεί, και είναι τεχνικά εντυπωσιακό, αλλά δεν κάνει αυτό που χρειαζόμαστε να κάνει». Η φάση ανάλυσης απαιτήσεων καθορίζει της επιχειρησιακές απαιτήσεις για το νέο σύστημα.

Η λέξη κλειδί σε αυτήν την φάση είναι το «τι» και όχι το «πως»! Οι αναλυτές είναι συχνά τόσο επιφυλακτικοί, με την τεχνική λύση που προσδιορίζουν ανεπαρκώς τις επιχειρησιακές απαιτήσεις για το νέο σύστημα. Αυτή η φάση απαντά στο ερώτημα «τι χρειάζονται και τι θέλουν από το νέο σύστημα οι χρήστες;» Είναι από τις σημαντικότερες φάσεις για την επιτυχία κάθε νέου πληροφοριακού συστήματος. (Σε άλλες μεθοδολογίες η φάση ανάλυσης απαιτήσεων μπορεί να λέγεται φάση προσδιορισμού ή φάση λογικού σχεδιασμού. Επίσης σε κάποιες μεθοδολογίες μπορεί να δούμε φάση ανάλυσης προβλήματος με την φάση ανάλυσης απαιτήσεων σαν ένα βήμα μαζί.) Αυτή την φάση δεν μπορούμε να την παρακάμψουμε. Σε αυτή τη φάση έχουμε τους ακόλουθους στόχους :

- Προσδιορισμός των απαιτήσεων του συστήματος

- Καθορισμός προτεραιότητας των απαιτήσεων
- Αναβάθμιση ή αναπροσαρμογή του αρχικού σχεδίου
- Κοινοποίηση απαιτήσεων

2.4.1 Προσδιορισμός των απαιτήσεων του συστήματος

Ο αρχικός στόχος της ανάλυσης των απαιτήσεων είναι, να προσδιοριστούν οι απαιτήσεις. Ενώ αυτό μπορεί να φαίνεται εύκολο ή τετριμμένο, είναι συχνά η πηγή αρκετών λαθών, συγκρούσεων και παραλείψεων. Οι βάσεις σε αυτό το στόχο τέθηκαν στην φάση της ανάλυσης προβλήματος όταν προσδιορίστηκαν οι στόχοι ανάπτυξης του συστήματος. Έτσι μπορούμε να πούμε πως γίνεται μια περιληπτική μετάφραση των λειτουργικών και μη λειτουργικών απαιτήσεων, που θα χρειαστούν για να επιτύχουμε τους στόχους μας. Οι λειτουργικές απαιτήσεις είναι συχνά, από την άποψη των διαδικασιών, αποτελεσμάτων, εισαγωγών και των αποθηκευμένων στοιχείων, απαραίτητες για την βελτίωση του συστήματος. Παραδείγματα μη λειτουργικών απαιτήσεων είναι η απόδοση, η ευκολία εκμάθησης και χρήσης, προϋπολογισμοί, κόστος, μείωση κόστους, χρονικά περιθώρια, ανάγκες εκπαίδευσης, ποιοτική διοίκηση, ασφάλεια και εσωτερικοί λογιστικοί έλεγχοι.

Σπάνια θα βρεθούν όλες οι λειτουργικές και μη απαιτήσεις. Η ολοκλήρωση και η τελειότητα δεν μας ενδιαφέρει σε αυτό τον στόχο. Όσο θα προχωράμε στα επόμενα βήματα, θα προσθέτουμε νέες απαιτήσεις και λεπτομέρειες που θα βελτιώνουν το σύστημα.

Οι αναλυτές συστημάτων διευκολύνουν αυτόν τον στόχο και καταγράφουν τα αποτελέσματα. Οι χρήστες του συστήματος είναι η βασική πηγή των επιχειρησιακών απαιτήσεων. Μερικοί ιδιοκτήτες συστημάτων μπορεί να επιλέγουν να συμμετέχουν, αφού και αυτοί έπαιξαν ρόλο στην καταγραφή των στόχων βελτίωσης. Οι Σχεδιαστές και οι προγραμματιστές συστημάτων δεν εμπλέκονται επειδή έχουν την τάση να επικεντρώνουν σε τεχνολογικές λύσεις.

Το μόνο παραδοτέο σε αυτό το βήμα είναι ένα σχέδιο λειτουργικών και μη λειτουργικών απαιτήσεων (μπορεί να είναι και παραπάνω από ένα σχήματα

διαφορετικά). Στην απλούστερη μορφή του , χωρίζεται σε τέσσερα λογικά μέρη : μία λίστα εισαγωγών, διαδικασιών, αποτελεσμάτων και αποθηκευμένων δεδομένων που χρειάζονται για να πετύχουμε τον στόχο μας. Σταδιακά βέβαια οι αναλυτές εκφράζουν της λειτουργικές απαιτήσεις χρησιμοποιώντας ένα εργαλείο που ονομάζεται Use Cases. Το Use Cases είναι ένα επιχειρηματικό σενάριο ή γεγονός, για το οποίο ο αναλυτής πρέπει να παρέχει μία καθορισμένη απάντηση.

Πολλές τεχνικές μπορούν να εφαρμοστούν σε αυτό το στάδιο όπως η Joint Requirements Planning η οποία είναι τέλεια τεχνική για γρήγορη καταγραφή των απαιτήσεων. Εναλλακτική λύση για τους αναλυτές είναι άλλες μέθοδοι όπως οι έρευνες και οι συνεντεύξεις (αυτές οι μέθοδοι ονομάζονται Fact- finding methods).

2.4.2 Προτεραιότητα στις απαιτήσεις των συστημάτων

Νωρίτερα, δηλώσαμε ότι η επιτυχία ενός αναπτυξιακού έργου συστημάτων μπορεί να αξιολογηθεί σε βαθμό στον οποίο καλύπτονται οι επιχειρησιακές απαιτήσεις, αλλά δεν δημιουργούνται όμοια όλες οι απαιτήσεις. Αν μια εργασία μείνει εκτός χρονοδιαγράμματος ή εάν ξεπεράσει τον προϋπολογισμό, το επόμενο βήμα είναι να γίνει ευδιάκριτο ποιες απαιτήσεις είναι περισσότερο σημαντικές από άλλες. Επομένως, λαμβάνοντας υπόψη τις επικυρωμένες απαιτήσεις, οι ιδιοκτήτες συστημάτων και οι χρήστες θα πρέπει να δώσουν προτεραιότητα στις απαιτήσεις των συστημάτων.

Ο καθορισμός προτεραιοτήτων των απαιτήσεων μπορεί να γίνει πιο εύκολος με τη χρήση μιας δημοφιλούς τεχνικής η οποία αποκαλείται Time Boxing. Το Time boxing προσπαθεί να διαιρέσει τις απαιτήσεις σε «χοντρά κομμάτια» τα οποία εντός μιας χρονικής περιόδου έχουν την δυνατότητα να τεθούν σε εφαρμογή. Επίσης ωθεί τις προτεραιότητες να γίνουν ξεκάθαρα αναγνωρίσιμες.

Οι αναλυτές συστημάτων καθιστούν περισσότερο εύκολο τον στόχο του καθορισμού των προτεραιοτήτων. Οι ιδιοκτήτες συστημάτων και οι χρήστες καθιερώνουν τις βασικές προτεραιότητες. Οι σχεδιαστές και οι χρήστες συστημάτων δεν περιλαμβάνονται στον στόχο. Ο στόχος αυτός ΞΕΚΙΝΑ από τις

επικυρωμένες απαιτήσεις. Θα πρέπει να είναι προφανές ότι δεν μπορεί κάποιος να δώσει επαρκώς προτεραιότητα σε ένα ημιτελές σύνολο απαιτήσεων. Παραδοτέα στο τέλος αυτό του στόχου είναι οι απαιτήσεις μαζί με τις προτεραιότητες. Οι προτεραιότητες μπορούν να ταξινομηθούν με βάση την σχετική ονομασία τους :

- Η υποχρεωτική απαίτηση είναι αυτή η οποία πρέπει να εκπληρωθεί από το ελάχιστο σύστημα (έκδοση 1.0). Το σύστημα είναι άχρηστο χωρίς αυτό. Πρέπει όμως να είναι κάποιος προσεχτικός. Υπάρχει η περίπτωση να παραπλανηθεί και να χαρακτηρίσει πάρα πολλές απαιτήσεις ως υποχρεωτικές.
- Η επιθυμητή απαίτηση είναι αυτή η οποία δεν είναι απόλυτα ουσιαστική σύμφωνα με την έκδοση 1.0. Μπορεί και να είναι ουσιαστική όσον αφορά το όραμα κάποιας μελλοντικής έκδοσης. Οι επιθυμητές απαιτήσεις μπορούν και πρέπει να καταταχθούν. Η χρήση των αριθμών έκδοσης ως σχέδιο κατάταξης είναι ένας αποτελεσματικός τρόπος να μεταβιβαστούν και να κατηγοριοποιηθούν οι επιθυμητές απαιτήσεις.

2.4.3 Αναβάθμιση ή αναπροσαρμογή του αρχικού σχεδίου

Τώρα που έχουμε προσδιορίσει τις απαιτήσεις του επιχειρησιακού συστήματος, θα πρέπει να κάνουμε ένα βήμα πίσω, να επαναπροσδιορίσουμε τον τρόπο που εκλαμβάνουμε το πεδίο δράσης της επιχείρησης και να εκσυγχρονίσουμε ανάλογα το σχέδιο της. Η ομάδα πρέπει να εξετάσει την πιθανότητα να είναι μεγαλύτερο από ότι αρχικά αναμενόταν το νέο σύστημα. Αν έτσι έχουν τα πράγματα η ομάδα πρέπει να προσαρμόσει το χρονοδιάγραμμα, τον προϋπολογισμό ή ανάλογα να θέσει κάποια περιθώρια. Επίσης θα πρέπει να εξασφαλίσουμε την έγκριση της επιχείρησης για να συνεχίσουμε στην επόμενη φάση. Η εργασία ίσως και να έχει ήδη ξεκινήσει στις φάσεις σχεδιασμού. Παρόλα αυτά οι αποφάσεις εξακολουθούν να απαιτούν μια αναθεώρηση.

Ο υπεύθυνος έργου διευκολύνει το στόχο από κοινού με τους ιδιοκτήτες συστημάτων και ολοκληρώνει την ομάδα επιχείρησης. Ο υπεύθυνος έργου και οι ιδιοκτήτες συστημάτων ως συνήθως, αποτελούν τα άτομα κλειδιά σε αυτό τον

στόχο. Πρέπει να λάβουν, όμως, υπόψη τους πως υπάρχει η πιθανότητα να υπερβαίνουν οι απαιτήσεις το αρχικό όραμα το οποίο είχε καθιερωθεί στην επιχείρηση και το νέο σύστημα. Ίσως να πρέπει να μειώσουν το περιθώριο πεδίου δράσης να βάλουν ένα χρονικό περιθώριο ή να αυξήσουν το προϋπολογισμό έτσι ώστε να γίνει η εργασία.

2.4.4 Κοινοποίηση απαιτήσεων

Η κοινοποίηση είναι ένας τρέχων στόχος της αναλυτικής φάσης των απαιτήσεων . Εμείς πρέπει να μεταδώσουμε διαμέσων αυτής της φάσης τις απαιτήσεις και τις προτεραιότητες στην επιχειρησιακή κοινότητα. Οι χρήστες και οι υπεύθυνοι θα ενεργούν ομαδικά για την εκτίμηση των απαιτήσεων και των προτεραιοτήτων. Η κοινοποίηση είναι η διαδικασία στη οποία πρέπει να καταγραφούν οι διαφορές των απόψεων . Ο υπεύθυνος έργου και ο αποκλειστικός χρηματοδότης θα πρέπει με ενιαία συμμετοχή να διευκολύνουν το στόχο. Στη σημερινή εποχή, μια δικτυακή εφαρμογή χρησιμοποιείται να μεταδίδει τις απαιτήσεις.

2.4.5 Διαρκείς διαχείριση απαιτήσεων

Κάποτε ήταν δημοφιλές να παγώνεις τις επιχειρησιακές απαιτήσεις πριν ξεκινήσει ο σχεδιασμός του και η φάση κατασκευής του. Αλλά η σημερινή οικονομία αυξάνεται ολοένα και με γρηγορότερο ρυθμό. Οι επιχειρήσεις μετρούν την ικανότητα της γρήγορης προσαρμογής στις διαρκές αλλαγές σε απαιτήσεις και ευκαιρίες. Επομένως η φάση της ανάλυσης των απαιτήσεων δεν τελειώνει ποτέ. Έτσι ενώ προχωράμε σιγά σιγά στις επόμενες φάσεις, θα γίνετε διαρκώς διαχείριση των απαιτήσεων.

Η διαχείριση των απαιτήσεων προσδιορίζει την διαδικασία για τους ιδιοκτήτες, χρήστες, αναλυτές σχεδιαστές και προγραμματιστές συστημάτων να καταθέτουν προτάσεις αλλαγών στις απαιτήσεις του συστήματος. Η διαδικασία διευκρινίζει πως η αλλαγές θα ζητηθούν και τεκμηριωθούν, πως θα καταγραφούν

και θα ακολουθηθούν, τότε και πως θα αξιολογηθούν κατά προτεραιότητα, αν και πως τελικά θα ικανοποιηθούν.

2.5 ΛΟΓΙΚΟΣ ΣΧΕΔΙΑΣΜΟΣ

Μία λογική σχεδίαση καταγράφει τις επιχειρησιακές απαιτήσεις χρησιμοποιώντας μοντέλα συστημάτων όπου επεξηγείτε η δομή δεδομένων, οι επιχειρησιακές διαδικασίες, οι ροές στοιχείων και οι διεπαφές χρηστών. Κατά μία έννοια επικυρώνονται οι απαιτήσεις που καθιερώθηκαν στην προηγούμενη φάση.

Για μία ακόμη φορά η τμηματοποίηση του πληροφοριακού συστήματος εξυπηρετεί στην καταγραφή των απαιτήσεων του. Υπάρχει βέβαια η ανησυχία για τις προοπτικές των χρηστών του συστήματος. Σε αυτή τη φάση σχεδιάζονται διάφορα μοντέλα που καταγράφουν τις απαιτήσεις για ένα νέο και βελτιωμένο σύστημα. Τα μοντέλα φέρνουν στην επιφάνεια διάφορες πτυχές της τμηματοποίησης. Η εναλλακτική λύση είναι ο σχεδιασμός πρωτοτύπων για την «ανακάλυψη των απαιτήσεων». Η φάση λογικού σχεδιασμού περιλαμβάνει τους ακόλουθους στόχους :

- Δομή λειτουργικών απαιτήσεων
- Διαμόρφωση πρότυπων λειτουργικών απαιτήσεων
- Επικύρωση λειτουργικών απαιτήσεων
- Καθορισμός αποδοχής δοκιμαστικά λογισμικά

2.5.1 Δομή λειτουργικών απαιτήσεων

Μια προσέγγιση του λογικού σχεδιασμού είναι η κατασκευή των λειτουργικών απαιτήσεων. Για τον σκοπό αυτό χρησιμοποιούνται «ευέλικτες μέθοδοι» (agile methods), αυτό σημαίνει ότι πρέπει να σχεδιάσουμε ή να αναβαθμίσουμε ένα ή περισσότερα μοντέλα συστημάτων για να απεικονίσουμε τις λειτουργικές απαιτήσεις. Αυτό μπορεί να περιλαμβάνει οποιοδήποτε

συνδυασμό μοντελοποίησης δεδομένων, διαδικασιών και αντικειμένων όπου φέρνουν στην επιφάνεια με ακρίβεια τις απαιτήσεις των χρηστών και της επιχείρησης. Τα συστήματα μοντέλων δεν ολοκληρώνονται μέχρι να μοντελοποιηθούν όλες οι απαραίτητες λειτουργικές απαιτήσεις. Τα μοντέλα αυτά συχνά εμπλουτίζονται με λεπτομερή λογικές υποδείξεις που περιγράφουν τις ιδιότητες δεδομένων καθώς επίσης και τους επιχειρηματικούς κανόνες και τις πολιτικές.

Οι αναλυτές συστημάτων είναι αυτοί που συμμετέχουν σε αυτή την φάση και καταγράφουν τα αποτελέσματα. Οι χρήστες συστημάτων είναι η βασική πηγή πραγματικών λεπτομερειών που χρειάζεται να σχεδιαστούν τα μοντέλα. Αυτή η φάση ενεργοποιείται από τις λειτουργικές απαιτήσεις. Τα αποτελέσματα είναι η δημιουργία των συστημάτων μοντέλων και οι λεπτομερείς υποδείξεις. Τέλος το επίπεδο των απαιτούμενων λεπτομερειών εξαρτάται από την μέθοδο που ακολουθείτε.

2.5.2 Διαμόρφωση πρότυπων λειτουργικών απαιτήσεων

Μερικές φορές οι χρήστες, αντιμετωπίζουν δυσκολία στο να εκφράσουν τα γεγονότα που είναι απαραίτητα για να σχεδιαστούν επαρκής μοντέλα συστημάτων. Σε αυτή την περίπτωση, μία εναλλακτική ή συμπληρωματική προσέγγιση για τις μεθόδους συστημάτων είναι να κατασκευαστούν νέα πρότυπα. Η δημιουργία προτύπων χρησιμοποιείται στην φάση ανάλυσης απαιτήσεων, για την κατασκευή δειγμάτων δεδομένων και αποτελεσμάτων. Τα δεδομένα και τα αποτελέσματα βοηθούν στην κατασκευή της βασικής βάσης δεδομένων και των προγραμμάτων για την εισαγωγή δεδομένων και εξαγωγή αποτελεσμάτων από και προς την βάση δεδομένων. Αν η κατασκευή νέων προτύπων είναι προαιρετική, συχνά εφαρμόζεται στα έργα ανάπτυξης συστημάτων, ειδικά σε περιπτώσεις όπου χρήστες αντιμετωπίζουν δυσκολία στην δήλωση ή στην απεικόνιση των απαιτήσεων της επιχείρησης. Η όλη φιλοσοφία είναι οι χρήστες να αναγνωρίζουν τις απαιτήσεις όταν τις βλέπουν.

Οι σχεδιαστές συστήματος διευκολύνουν αυτό το στόχο. Οι αναλυτές συστημάτων καταγράφουν και αναλύουν τα αποτελέσματα. Ως συνήθως οι χρήστες συστήματος είναι η βασική πηγή συλλογής πραγματικών δεδομένων. Οι προγραμματιστές και αναλυτές συστημάτων ανταποκρίνονται στον συγκεκριμένο στόχο κατασκευάζοντας πρότυπα.

2.5.3 Επικύρωση λειτουργικών απαιτήσεων

Τα μοντέλα συστημάτων και τα πρότυπα αντιπροσωπεύουν τις απαιτήσεις των χρηστών τα οποία πρέπει να είναι επικυρωμένα με την ολοκλήρωσή τους και την ορθότητά τους. Οι αναλυτές συστημάτων διευκολύνουν το στόχο δεσμεύοντας αμφίδρομα και τους χρήστες συστημάτων να αναγνωρίζουν τα λάθη και τις παραλείψεις ή να κάνουν διευκρινήσεις.

2.5.4 Καθορισμός αποδοχής δοκιμαστικών λογισμικών

Ενώ αυτός δεν είναι ένας απαραίτητος στόχος, οι περισσότεροι ειδικοί συμφωνούν πως δεν είναι νωρίς για να αρχίσει ο σχεδιασμός για την δοκιμή του συστήματος. Τα συστήματα μοντέλων και τα πρότυπα προσδιορίζουν πολύ αποδοτικά την διαδικασία απαιτήσεων, τους κανόνες δεδομένων και τους επιχειρησιακούς κανόνες για το νέο σύστημα. Αναλόγως αυτές οι προδιαγραφές μπορούν να χρησιμοποιηθούν για να προσδιοριστούν τα δοκιμαστικά λογισμικά, που μπορούν τελικά να δοκιμάσουν προγράμματα για την ορθότητα τους. Οι αναλυτές συστημάτων και οι προγραμματιστές μπορούν να συμμετέχουν σε αυτή τη φάση επικυρώνοντας τα δοκιμαστικά λογισμικά με τους χρήστες συστημάτων.

2.6 ΑΝΑΛΥΣΗ ΑΠΟΦΑΣΕΩΝ

Με δεδομένες τις επιχειρησιακές απαιτήσεις για ένα βελτιωμένο σύστημα, μπορούμε να διευθετήσουμε πως το νέο σύστημα μπορεί να εφαρμοστεί τεχνολογικά. Ο σκοπός σε αυτή την φάση είναι να αναγνωριστούν οι υποψήφιος

λύσεις, να αναλυθούν και να προταθεί ο στόχος συστήματος που θα σχεδιαστεί, θα κατασκευαστεί και θα εφαρμοστεί. Υπάρχουν πιθανότητες βέβαια να έχει κάποιος ήδη προτείνει ή οραματιστεί την τεχνική λύση. Κατά την διάρκεια της ανάλυσης απαιτήσεων είναι επιτακτικό να αναγνωριστούν οι επιλογές, να αναλυθούν και μετά να πωληθούν οι καλύτερες λύσεις βασισμένες στην ανάλυση.

Για ακόμη μία φορά η τεχνική της τμηματοποίησης μπορεί να είναι χρήσιμη για την φάση ανάλυσης αποφάσεων. Ένα από τα πρώτα πράγματα που παρατηρείτε , είναι πως οι τεχνολογικές πληροφορίες και η αρχιτεκτονική επηρεάζουν στις αποφάσεις που πρέπει να παρθούν. Σε μερικές περιπτώσεις πρέπει να δουλεύουμε μέσα στα πρότυπα, ενώ σε άλλες περιπτώσεις πρέπει να βρίσκουμε διαφορετικές τεχνολογικές λύσεις ή να αναδεικνύουμε καινούργιες. Πρέπει επίσης να αναφερθεί πως οι προοπτικές , είναι στη μετάβαση από τους χρήστες συστημάτων σε αυτούς που σχεδιάζουν το σύστημα.

Σε αυτήν την τελική φάση θα παρουσιαστεί η πρόταση για το νέο σύστημα που θα εκπληρώνει όλες τις επιχειρησιακές απαιτήσεις που είδαμε στο προηγούμενο βήμα. Η φάση ανάλυσης απαιτήσεων περιλαμβάνει τους ακόλουθους στόχους :

- Προσδιορισμός υποψήφιων λύσεων
- Ανάλυση υποψήφιων λύσεων
- Σύγκριση υποψήφιων λύσεων
- Αναβάθμιση επιχειρησιακού σχεδίου
- Υπόδειξη λύσης συστήματος

2.6.1 Προσδιορισμός υποψήφιων λύσεων

Δεδομένου των επιχειρησιακών απαιτήσεων που προσδιορίστηκαν στην φάση προσδιορισμού , πρέπει πρώτα να αναγνωρίσουμε τις εναλλακτικές υποψήφιες λύσεις. Μερικές από τις υποψήφιες λύσεις θα τεθούν από σχεδιασμένες ιδέες και απόψεις από τους ιδιοκτήτες και χρήστες συστημάτων. Άλλες λύσεις μπορεί να προέρθουν από ποικίλες πηγές συμπεριλαμβανομένων

των αναλυτών συστημάτων, των σχεδιαστών, τεχνικών συμβούλων και άλλους ειδικούς πληροφοριακών συστημάτων. Κάποιες από τις τεχνικές λύσεις μπορεί να περιοριστούν λόγω μιας προκαθορισμένης αρχιτεκτονικής. Πρόθεση αυτού του στόχου είναι να αξιολογήσει τις υποψήφιας λύσεις , από το να μας προσδιορίσει αυτές που θα πρέπει να λάβουμε υπόψη μας.

2.6.2 Ανάλυση υποψήφιας λύσεων

Κάθε υποψήφια λύση θα πρέπει να αναλυθεί για την σκοπιμότητά της. Αυτό προκύπτει από την μελέτη όλων των υποψήφιας λύσεων. Η ανάλυση σκοπιμότητας δεν πρέπει να περιορίζεται από κόστη και κέρδη. Οι περισσότεροι αναλυτές αξιολογούν τις λύσεις με τα παρακάτω κριτήρια.

- Τεχνολογική σκοπιμότητα- είναι η λύση τεχνολογικά πρακτική; Έχει το προσωπικό τις απαραίτητες γνώσεις να σχεδιάσει και να υλοποιήσει αυτή την λύση.
- Λειτουργικότητα σκοπιμότητας- θα εκπληρώνει η λύση, τις απαιτήσεις των χρηστών; Σε ποιο βαθμό; Σε ποιο βαθμό η λύση θα αλλάξει το εργασιακό περιβάλλον των χρηστών; Πως νοιώθουν οι χρήστες με αυτή την λύση;
- Οικονομική σκοπιμότητα- Είναι η λύση οικονομικά συμφέρουσα;
- Πρόγραμμα σκοπιμότητας- Μπορεί η λύση να σχεδιαστεί και να εφαρμοσθεί σε ένα αποδεκτό χρονικό διάστημα;

Κατά την ολοκλήρωση αυτού του στόχου, οι αναλυτές και οι χρήστες δεν πρέπει να κάνουν το λάθος της σύγκρισης μεταξύ των υποψήφιας λύσεων. Η ανάλυση σκοπιμότητας αποδίδεται σε κάθε λύση ξεχωριστά, χωρίς να λαμβάνετε υπόψη η σκοπιμότητα των άλλων υποψήφιας λύσεων. Αυτή η προσέγγιση αποθαρρύνει τους αναλυτές και τους χρήστες να πάρουν μια πρόωρη – βιαστική λύση όσον αφορά για το ποια είναι η καλύτερη λύση.

Σε αυτή την φάση συμμετέχουν πάλι οι αναλυτές συστημάτων. Συνήθως οι ιδιοκτήτες συστημάτων και οι χρήστες, αναλύουν το λειτουργικό, οικονομικό και τον προγραμματισμό σκοπιμότητας. Οι σχεδιαστές συστημάτων και οι

προγραμματιστές συνεισφέρουν στις αναλύσεις και παίζουν σημαντικό ρόλο στην ανάλυση τεχνολογικής σκοπιμότητας.

2.6.3 Σύγκριση υποψήφιων λύσεων

Μόλις η ανάλυση σκοπιμότητας έχει ολοκληρωθεί για κάθε υποψήφια λύση, μπορούμε να τις συγκρίνουμε και να διαλέξουμε μια ή περισσότερες προτεινόμενες λύσεις για να τις παρουσιάσουμε στον ιδιοκτήτη και στους χρήστες του συστήματος. Σε αυτό το σημείο, ξεχωρίζονται οι λύσεις που απορρίφθηκαν από αυτές που θα προταθούν. Αναζητώντας την ιδανικότερη λύση θα εντοπισθεί και θα προταθεί αυτή που προσφέρει τον καλύτερο συνολικό συνδυασμό τεχνολογικού, οικονομικού, λειτουργικού και προγραμματισμού σκοπιμότητας.

Οι αναλυτές συστημάτων συμμετέχουν και σε αυτή την φάση. Οι σχεδιαστές συστημάτων και οι προγραμματιστές είναι διαθέσιμοι να απαντήσουν σε κάθε ερώτημα τεχνολογικής σκοπιμότητας. Οι ιδιοκτήτες συστημάτων και οι χρήστες πρέπει να πάρουν την απόφαση για την τελική ανάλυση και πρόταση. Το παραδοτέο αυτής της ενότητας είναι η τελική απόφαση. Αν προταθούν περισσότερες από μια λύσεις πρέπει να ορισθούν προτεραιότητες.

2.6.4 Αναβάθμιση επιχειρησιακού σχεδίου

Όπως έχει παρατηρηθεί σε αυτό το κεφάλαιο ανανεώνουμε διαρκώς το επιχειρησιακό πλάνο ανάλογα με το πρόβλημα, τις απαιτήσεις και τις λύσεις. Βασισμένοι στην προτεινόμενη λύση πρέπει πάλι να επαναπροσδιορίσουμε το σκοπό του έργου και να αναβαθμίσουμε το επιχειρησιακό πλάνο σχετικά.

Ο υπεύθυνος έργου σε συνδυασμό με τους ιδιοκτήτες συστήματος και ολόκληρη την ομάδα έργου συμμετέχουν σε αυτό το στόχο. Οι αναλυτές συστημάτων και ιδιοκτήτες έχουν τον πρώτο λόγο, αλλά επειδή βρισκόμαστε στον τεχνολογικό σχεδιασμό συστήματος, πρέπει να εμπλέξουμε τόσο τους σχεδιαστές όσο και τους προγραμματιστές συστημάτων στην αναβάθμιση επιχειρησιακού σχεδίου.

2.6.5 Υπόδειξη λύσης συστήματος

Όπως με την αρχική έρευνα και την φάση ανάλυσης προβλήματος, έτσι και η φάση απόφασης περιλαμβάνει το κομμάτι της επικοινωνίας. Πρέπει να προτείνουμε μια λύση για το σύστημα στην κοινωνία της επιχείρησης.

Ο υπεύθυνος έργου και ο αποκλειστικός χρηματοδότης συμμετέχουν από κοινού σε αυτό τον στόχο. Όπως συνήθως, μέσω συσκέψεων (όπου παρευρίσκονται οι ιδιοκτήτες, χρήστες, αναλυτές, σχεδιαστές, προγραμματιστές και όλη η ομάδα του έργου) γίνεται η ενημέρωση. Επίσης μπορεί να παρευρεθεί όλο το ενδιαφερόμενο προσωπικό από την κοινωνία της εταιρίας. Ακόμα εάν έχει δημιουργηθεί μια δικτυακή εφαρμογή για το πλάνο, θα πρέπει να ανανεώνεται διαρκώς για την καλύτερα δυνατή ενημέρωση του προσωπικού.

Διαπροσωπικές και επικοινωνιακές ικανότητες είναι απαραίτητες σε αυτήν την ενότητα. Επίσης απαραίτητο είναι να υπάρχει πειθώ όπως σε έναν ικανό πωλητή. Οι αναλυτές συστημάτων πρέπει να μπορούν να γράφουν επίσημες αναφορές και παρουσιάσεις στην επιχείρηση χωρίς να εμπλέκουν τεχνικά θέματα.

2.7 Μοντελοποίηση Δεδομένων και Διαδικασιών

Τα μοντέλα συστημάτων διαδραματίζουν ένα σημαντικό ρόλο στην ανάπτυξη συστημάτων. Ένας αναλυτής ή ένας χρήστης συστημάτων αντιμετωπίζει συχνά μη δομημένα προβλήματα. Ένας τρόπος για να αντιμετωπιστούν αυτά τα προβλήματα είναι ο σχεδιασμός μοντέλων. Τα μοντέλα αντιπροσωπεύουν την πραγματικότητα όπως μία εικόνα χίλιες λέξεις. Τα μοντέλα σχεδιάζονται για ένα ήδη υπάρχον σύστημα έτσι ώστε να γίνει πιο κατανοητό ή υποθετικά για ένα προτεινόμενο σύστημα έτσι ώστε να γίνει πιο εύκολη η καταγραφή των επιχειρηματικών απαιτήσεων και του τεχνικού σχεδιασμού. Μια σημαντική έννοια είναι η διάκριση μεταξύ λογικού και φυσικού μοντέλου.

Το λογικό μοντέλο μας δείχνει τι είναι ένα σύστημα και τι κάνει. Απεικονίζει το σύστημα ανεξάρτητα από οποιαδήποτε τεχνική εφαρμογή. Σαν αποτέλεσμα τα λογικά μοντέλα επεξηγούν σωστά την χρησιμότητα και τον τρόπο λειτουργίας του συστήματος. Τα φυσικά μοντέλα δείχνουν όχι μόνο τι είναι και τι κάνει ένα σύστημα αλλά και πως ένα σύστημα εφαρμόζεται τεχνικά.

Οι αναλυτές συστημάτων έχουν από καιρό αναγνωρίσει την αξία του διαχωρισμού των τεχνικών και των επιχειρησιακών ανησυχιών. Για αυτό το λόγο χρησιμοποιούν λογικά μοντέλα συστημάτων για την απεικόνιση επιχειρησιακών απαιτήσεων και φυσικά μοντέλα συστημάτων για την απεικόνιση τεχνικού σχεδιασμού. Οι δραστηριότητες των αναλυτών συστημάτων κλείνουν στα λογικά μοντέλα συστημάτων για τους ακόλουθους λόγους:

- Ενθαρρύνουν την δημιουργικότητα.
- Μειώνουν το ρίσκο παράβλεψης επιχειρησιακών απαιτήσεων λόγω της απασχόλησης μας με τεχνικές λεπτομέρειες. Αυτά τα λάθη για να διορθωθούν μετά την εφαρμογή του συστήματος κοστίζουν παραπάνω. Διαχωρίζοντας τι πρέπει να κάνει το σύστημα από το πώς θα το κάνει, μπορούμε να αναλύσουμε καλύτερα τις απαιτήσεις όσον αφορά την πληρότητα, την ακρίβεια, και την συνέπεια τους.
- Μας επιτρέπουν να επικοινωνούμε με τους τελικούς χρήστες με μη τεχνικές γλώσσες. Κατά συνέπεια δεν παραλείπονται οι απαιτήσεις λόγω επαγγελματικής γλώσσας.

Η μοντελοποίηση δεδομένων είναι μια τεχνική που οργανώνει και καταγράφει τα δεδομένα του συστήματος. Ορισμένες φορές μπορεί να την συναντήσουμε και σαν μοντελοποίηση βάσης δεδομένων. Επίσης καταγράφει και την λογική, την πολιτική και τις διαδικασίες που εφαρμόζονται.

Υπάρχουν διάφοροι τρόποι μοντελοποίησης δεδομένων όπως program structure charts, logic flowcharts, decision tables, data flow diagram και το Entity Relationship Diagram (E.R.D.) τα δύο τελευταία είναι τα πιο διαδεδομένα και για αυτό το λόγο θα αναλυθούν.

Το διάγραμμα οντοτήτων συσχετίσεων (Entity Relationship Diagram) απεικονίζει δεδομένα βάση των οντοτήτων και των σχέσεων που περιγράφονται

από τα δεδομένα. Υπάρχουν πολλοί τρόποι απεικόνισης των διαγραμμάτων οντοτήτων συσχετίσεων. Οι περισσότεροι έχουν πάρει το όνομα του εφευρέτη τους όπως για παράδειγμα οι Chen, Martin, Bachman, Merise και άλλοι από δημοσιευμένα πρότυπα όπως η IDEF1X. Όλες αυτές οι γλώσσες μοντελοποίησης δεδομένων υποστηρίζουν τις ίδιες θεμελιώδεις αρχές και δομές κατασκευής.

Μερικές από τις βασικές αρχές που αντιπροσωπεύουν όλα τα μοντέλα δεδομένων είναι :

- Οντότητες (Entities)
- Ιδιότητες (Attributes)
- Συσχετίσεις (Relationships)

2.7.1 Οντότητες (Entities)

Η οντότητα είναι η γενική ιδέα που παρουσιάζει περιληπτικά όλες τις περιπτώσεις κάποιων «πραγμάτων» και είναι κάτι για το οποίο οι επιχειρήσεις χρειάζονται να καταχωρούν δεδομένα. Κατά την μοντελοποίηση συστημάτων είναι απαραίτητο σε κάθε ξεχωριστή ιδέα να ορίσουμε ένα σχήμα. Συνήθως το σχήμα που χρησιμοποιείται είναι το ορθογώνιο παραλληλόγραμμο με κυκλικές γωνίες. Για παράδειγμα ένα σύστημα σχολείου, περιλαμβάνει δεδομένα σχετικά με «πράγματα» όπως μαθητές, καθηγητές, μαθήματα και τάξεις. (όπως φαίνεται και στο σχήμα 2.1).



Σχήμα 2.1

Όλες οι οντότητες ανήκουν σε κάποια κατηγορία, οι κατηγορίες αυτές είναι :

Πρόσωπα : Αντιπρόσωποι, κατασκευαστές, πελάτες, υπάλληλοι, προμηθευτές, μαθητές, καθηγητές.

Τοποθεσίες : Περιοχές πωλήσεων, κτίρια, τάξεις, δωμάτια, γραφεία.

Αντικείμενα : Βιβλία, μηχανές, προϊόντα, πρώτες ύλες, άδειες λογισμικού, πακέτα λογισμικού, εργαλεία, όχημα.

Γεγονότα : Εφαρμογές, βραβεία, ακυρώσεις, μαθήματα, κατηγορίες, εντολές, πτήσεις, ανανεώσεις, εγγραφές, πωλήσεις, ταξίδια, κρατήσεις.

Έννοιες : Λογαριασμοί, διορία, δεσμοί, σειρά μαθημάτων, κεφάλαια, προσόντα, αποθέματα.

Είναι σημαντικό να ξεχωρίσουμε την διαφορά μεταξύ μιας οντότητας και των περιπτώσεων της. Για παράδειγμα η οντότητα μαθητές μπορεί να έχει πολλές περιπτώσεις : Μαίρη, Κατερίνα, Αλεξάνδρα, Νότα, Κωνσταντίνος.

¶

2.7.2 Ιδιότητες (Attributes)

Αν μια οντότητα είναι κάτι για το οποίο χρειάζεται να καταχωρήσουμε δεδομένα, τότε πρέπει να αναγνωρίσουμε τα συγκεκριμένα κομμάτια δεδομένων που θέλουμε να αποθηκεύσουμε για κάθε ξεχωριστή περίπτωση οντότητας. Αυτά τα κομμάτια δεδομένων τα ονομάζουμε Attributes (ιδιότητες), δηλαδή είναι μια περιγραφική ιδιότητα ή χαρακτηριστικά μιας οντότητας. Όπως είδαμε και στην αρχή κάθε περίπτωση της οντότητας *μαθητές* μπορεί να περιγράφεται από τις ακόλουθες ιδιότητες : Όνομα, επίθετο, τηλέφωνο, πόλη, χώρα, ταχυδρομικός κώδικας και άλλα.

Μπορούμε τώρα να επεκτείνουμε γραφικά την οντότητα για να περιλάβουμε τις ιδιότητες με την καταγραφή εκείνων των ιδιοτήτων μέσα στη μορφή οντοτήτων μαζί με το όνομα. Μερικές ιδιότητες μπορούν να κατηγοριοποιηθούν λογικά σε υπεριδιότητες οι οποίες ονομάζονται σύνθετες ιδιότητες. Για παράδειγμα το όνομα ενός μαθητή είναι στην πραγματικότητα μια σύνθετη ιδιότητα που περιλαμβάνει επίθετο, όνομα, αριθμός ταυτότητας.

Περιοχές (Domains) κατά την ανάλυση ενός συστήματος πρέπει να προσδιορίσουμε τις τιμές μιας ιδιότητας οι οποίες είναι έγκυρες ή έχουν

επιχειρησιακό νόημα. Οι τιμές για κάθε ιδιότητα καθορίζονται από τρεις κατηγορίες : τύπος δεδομένων, περιοχή και προεπιλογή.

Ο τύπος δεδομένων για κάθε ιδιότητα προσδιορίζει τι είδους δεδομένα μπορούν να καταχωρηθούν σε αυτή την ιδιότητα. Οι τύποι δεδομένων είναι γνωστοί σε αυτούς που έχουν ασχοληθεί με προγραμματισμό. Η δήλωση για του τύπου κάθε μεταβλητής είναι κοινή για τις περισσότερες γλώσσες προγραμματισμού όπως φαίνεται και στο σχήμα 2.2

Τύποι Δεδομένων	Λογική εξήγηση
Αριθμός	οποιοσδήποτε ακέραιος ή δεκαδικός
Κείμενο	Μια σειρά από χαρακτήρες συμπεριλαμβανομένων και αριθμών όμως με τους οποίους δεν γίνονται αριθμητικές πράξεις
Υπόμνημα	Το ίδιο με το κείμενο μόνο που έχει απροσδιόριστο μέγεθος
Ημερομηνία	Οποιαδήποτε μορφή ημερομηνίας
Ώρα	Οποιαδήποτε μορφή ώρας
Ναι/Όχι	Μπορεί να γίνει επιλογή μόνο από αυτές τις δύο τιμές
Καθορισμένη τιμή	Ένα πεπερασμένο σύνολο τιμών (π.χ. με μαθητές)
Εικόνα	οποιαδήποτε μορφή εικόνας

Σχήμα 2.2

Ο τύπος δεδομένων μιας ιδιότητας περιέχει μια περιοχή. Η περιοχή μιας ιδιότητας καθορίζει τις τιμές που μπορεί να πάρει η ιδιότητα. Οι σχεδιαστές του συστήματος είναι αυτοί που επιβάλουν ποιες θα είναι οι περιοχές όλων των ιδιοτήτων όπως φαίνεται και στο σχήμα 2.3

Τύποι Δεδομένων	Λογική επεξήγηση	Παραδείγματα
Αριθμός	Προσδιορισμός εύρους για ακέραιους αριθμούς: Για τους πραγματικούς ο προσδιορισμός γίνεται με ακρίβεια:	(10-99)
Κείμενο	Μέγιστο μέγεθος πεδίου	(1.000-799.999) Κείμενο (30)
Υπόμνημα	Δεν υπάρχουν περιορισμοί στο μέγεθος ή στο περιεχόμενο	Μη εφαρμόσιμο
Ημερομηνία	Διαφορετικές μορφές ημερομηνίας MMDDYYYY, MMDDYYYY.	MMDDYYYY MMYYYY YYYY
Ωρα	Διαφορετικές μορφές ώρας	HHMMT HHMM
Ναι/Όχι	{Ναι,Όχι}	{Ναι,Όχι}, {Ανοικτό, Κλειστό}
Καθορισμένη τιμή	(Τιμή#1, Τιμή#2, ..., Τιμή#n ή Πίνακας κωδικών και εννοιών	(Μαθητές, Καθηγητές, ...) M=Μαθητές K=Καθηγητές
Εικόνα	Εικόνα	Μη εφαρμόσιμο

Σχήμα 2.3

Τέλος κάθε ιδιότητα πρέπει να έχει μία λογική προκαθορισμένη τιμή που θα αντιπροσωπεύει την αξία μιας ιδιότητας εάν η αξία της δεν διευκρινίζεται από το χρήστη. Στον πίνακα που ακολουθεί μπορούμε να δούμε πιθανές προκαθορισμένες τιμές. (Not Null – σε αυτή την περίπτωση πρέπει να έχουμε τιμή, Null – δεν είναι υποχρεωτικό να έχουμε τιμή)

Προσδιορισμός (Identification) Μια οντότητα έχει πολλές περιπτώσεις, ίσως χιλιάδες ή εκατομμύρια. Υπάρχει η ανάγκη να προσδιοριστεί μεμονωμένα κάθε περίπτωση βασισμένη στην τιμή των δεδομένων μιας ή περισσότερων ιδιοτήτων. Επίσης κάθε οντότητα πρέπει να έχει ένα «κλειδί». Για παράδειγμα, κάθε περίπτωση της οντότητας μαθητές μπορεί να αναγνωρίζεται από τον αριθμό μητρώου μαθητή που είναι και μοναδικός για τον καθένα.

Μερικές φορές περισσότερες από μία ιδιότητες απαιτούνται για να προσδιοριστεί μεμονωμένα μία περίπτωση οντότητας. Όταν έχουμε ένα κλειδί το οποίο ανήκει σε μία

ομάδα ιδιοτήτων αυτό ονομάζεται «*συνδεδεμένο κλειδί*». Για παράδειγμα σε ένα κατάστημα βίντεο μία κασέτα μπορεί να αναγνωρίζεται από τον συνδυασμό του κωδικού του τίτλου της ταινίας και του αριθμού του αντίτυπου. Ο κωδικός του τίτλου της ταινίας από μόνος του μπορεί να είναι ανεπαρκής γιατί το κατάστημα μπορεί να έχει πολλές φορές την συγκεκριμένη ταινία.

Συχνά μια οντότητα μπορεί να έχει παραπάνω από ένα κλειδιά. Για παράδειγμα η οντότητα υπάλληλοι μπορεί να προσδιοριστεί μεμονωμένα από τον αριθμό κοινωνικής ασφάλισης, από τον αριθμό που τους έχει δώσει η επιχείρηση, ή την ηλεκτρονική διεύθυνση. Ένα συνδεδεμένο κλειδί ονομάζεται και *βασικό κλειδί*. Η προεπιλογή για τα βασικά κλειδιά είναι πάντοτε Not Null (πρέπει να έχουμε τιμή). Οποιαδήποτε κλειδί δεν έχει επιλεγεί να είναι βασικό τότε αυτό ονομάζεται *εναλλακτικό κλειδί*.

2.7.3 Συσχετίσεις (Relationships)

Εννοιολογικά, οι οντότητες και οι ιδιότητες δεν υπάρχουν μεμονωμένα. Αυτά που αντιπροσωπεύουν αλληλεπιδρούν και έχουν αντίκτυπο το ένα με το άλλο για να υποστηρίξουν την επιχειρησιακή αποστολή. Σχέση είναι μία φυσική επιχειρησιακή ένωση που υπάρχει μεταξύ μιας ή περισσότερων οντοτήτων. Η σχέση μπορεί να αντιπροσωπεύει ένα γεγονός που συνδέει τις οντότητες ή μόνο μια λογική συγγένεια που υπάρχει μεταξύ των οντοτήτων. Για παράδειγμα ας δούμε τις οντότητες μαθητές και πρόγραμμα σπουδών, μπορούμε να κάνουμε τους ακόλουθους επιχειρησιακούς ισχυρισμούς που συνδέουν τους μαθητές και τις σειρές μαθημάτων:

- ένας τρέχων μαθητής εγγράφεται σε ένα ή περισσότερα προγράμματα σπουδών.
- ένα πρόγραμμα σπουδών μελετάται από μηδέν, έναν, ή περισσότερους μαθητές.

Τα υπογραμμισμένα ρήματα μας δείχνουν την σχέση που υπάρχει μεταξύ των δύο οντοτήτων.

Μπορούμε γραφικά να επεξηγήσουμε αυτήν την ένωση μεταξύ του σπουδαστή και του προγράμματος σπουδών όπως φαίνεται στο σχήμα 2.4. Η

γραμμή που συνδέει αντιπροσωπεύει τη σχέση και η φράση ρήματος την περιγράφει. Όλες οι συσχετίσεις μπορούν να ερμηνευτούν και προς τις δύο κατευθύνσεις. Οι μέθοδοι μοντελοποίησης συστημάτων μπορεί να διαφέρουν στην ονομασία των σχέσεων τους, μερικές περιλαμβάνουν και τις δύο φράσεις ρήματος και άλλες μόνο τη μία.








Σχήμα 2.4

Αριθμός στοιχείων συνόλου Το σχήμα 2.4 μας δείχνει επίσης την πολυπλοκότητα ή τον βαθμό της κάθε σχέσης. Για παράδειγμα στον παραπάνω ισχυρισμό θα πρέπει να απαντήσουμε στις ακόλουθες ερωτήσεις:

- Πρέπει για κάθε μαθητή να υπάρχει και ένα πρόγραμμα σπουδών; Όχι!
- Πρέπει να υπάρχει πρόγραμμα σπουδών και κάθε μαθητή; Ναι!
- Πόσες περιπτώσεις προγραμμάτων σπουδών μπορούν να υπάρχουν για κάθε μαθητή χωριστά; Πολλές!
- Πόσοι μαθητές μπορούν να υπάρχουν για κάθε πρόγραμμα σπουδών; Πολλοί!

Μπορούμε να πούμε αυτή την ιδέα αριθμό στοιχείων συνόλου (cardinality). Ο αριθμός στοιχείων συνόλου προσδιορίζει τον ελάχιστο και τον μέγιστο αριθμό περιστατικών μιας οντότητας που μπορεί να σχετίζονται με ένα περιστατικό της άλλης οντότητας. Η ποιο διαδεδομένη γραφική σημείωση για τον αριθμό στοιχείων συνόλου (cardinality) φαίνεται στο σχήμα 2.5

Ερμηνεία αριθμού στοιχείων συνόλου	Ελάχιστες περιπτώσεις	Μέγιστες περιπτώσεις	Γραφική σημείωση
Ακριβώς ένα	1	1	 - 1 -
Μηδέν ή Ένα	0	1	
Ένα ή παραπάνω	1	Περισσότερα (>1)	
Μηδέν, Ένα ή παραπάνω	0	Περισσότερα (>1)	
περισσότερα από ένα	>1	>1	

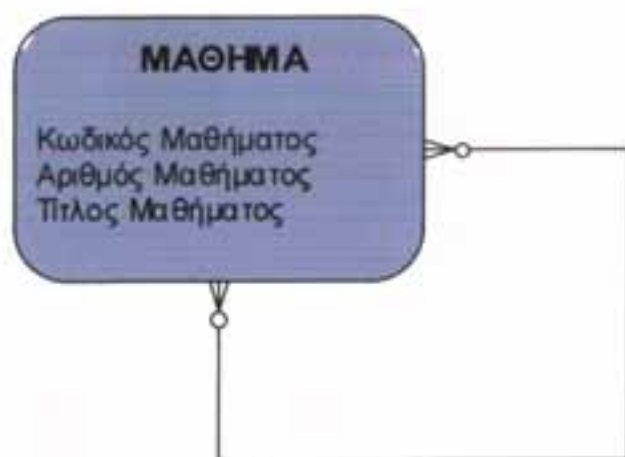
Σχήμα 2.5

Εννοιολογικά, ο αριθμός στοιχείων συνόλου μας λέει τους ακόλουθους κανόνες για τα στοιχεία που θέλουμε να αποθηκεύσουμε:

- Όταν εισάγουμε μια περίπτωση *μαθητή* στη βάση δεδομένων, πρέπει να τον συνδέσουμε τουλάχιστον σε μια περίπτωση *προγράμματος σπουδών*.
- Ένας *μαθητής* μπορεί να παρακολουθήσει περισσότερα από ένα *προγράμματα σπουδών*.
- Πρέπει να εισάγουμε ένα *πρόγραμμα σπουδών* προτού να συνδέσουμε τους *μαθητές* με αυτό. Για αυτό ένα *πρόγραμμα σπουδών* μπορεί να έχει μηδέν *μαθητές* (κανένας *μαθητής*)
- Μόλις εισαχθεί ένα *πρόγραμμα σπουδών* στη βάση δεδομένων, μπορούμε να του συνδέσουμε πολλούς *μαθητές*

Βαθμός: Ένα άλλο μέτρο της πολυπλοκότητας μιας σχέσης είναι ο βαθμός της. Ο βαθμός μιας σχέσης είναι ο αριθμός οντοτήτων που συμμετέχουν στη σχέση. Όλες οι συσχετίσεις που έχουμε εξερευνήσει μέχρι τώρα είναι *δυναδικές* (βαθμός = 2). Με άλλα λόγια, στη σχέση συμμετείχαν δύο διαφορετικές οντότητες.

Οι συσχετίσεις μπορούν επίσης να υπάρξουν μεταξύ διαφορετικών περιπτώσεων της ίδιας οντότητας. Αυτό ονομάζεται *επαναλαμβανόμενη σχέση* (βαθμός = 1). Παραδείγματος χάριν, στο σχολείο μια σειρά μαθημάτων μπορεί να είναι προϋπόθεση για άλλες σειρές μαθημάτων. Ομοίως, μια σειρά μαθημάτων μπορεί να διοργανώσει διάφορες άλλες σειρές μαθημάτων. Το σχήμα 2.6 καταδεικνύει την πολλά προς πολλά επαναλαμβανόμενη σχέση.



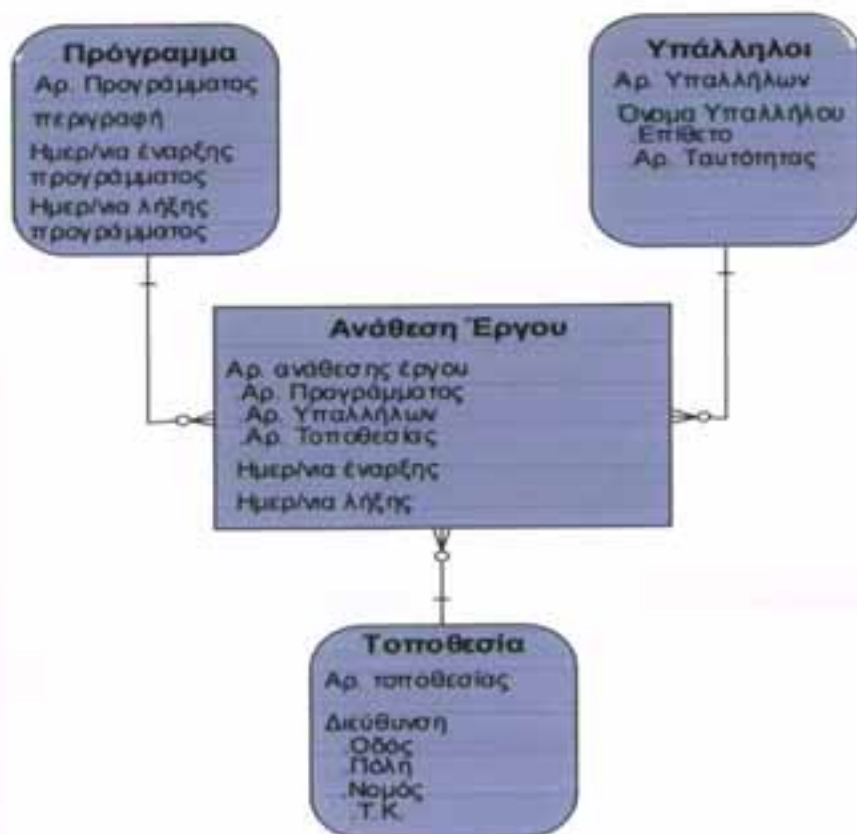
Σχήμα 2.6

Συσχετίσεις μπορούν επίσης να υπάρξουν μεταξύ περισσότερων από δύο διαφορετικών οντοτήτων. Αυτές οι συσχετίσεις μερικές φορές καλούνται *τριαδικές*. Ένα παράδειγμα μιας *τριαδικής σχέσης* παρουσιάζεται στο σχήμα 2.7. Μια τριαδική σχέση διευκρινίζεται με ένα νέο κατασκεύασμα οντοτήτων αποκαλούμενο *συνειρμική οντότητα*. Μια συνειρμική οντότητα είναι μια οντότητα που κληρονομεί το αρχικό κλειδί του από περισσότερες από μια

άλλες οντότητες (αποκαλούμενες *γονέων*). Κάθε μέρος αυτών συνδέει τα βασικά σημεία σε μια και μόνο μια περίπτωση για κάθε μία συνδεδεμένη οντότητα.

Στο σχήμα η οντότητα ανάθεση συνδέεται με έναν υπάλληλο, μια θέση, και ένα πρόγραμμα. Για κάθε περίπτωση ανάθεσης, το κλειδί προσδιορίζει ποια ταυτότητα υπαλλήλων, ποιος αριθμός θέσης, και ποιος αριθμός προγράμματος συνδυάζονται για να διαμορφώσουν αυτή την ανάθεση.

Επίσης όπως φαίνεται στο σχήμα, μια συνειρμική οντότητα μπορεί να περιγραφεί από τις ιδιότητές της που δεν είναι κλειδιά. Εκτός από το αρχικό κλειδί, η οντότητα ανάθεση περιγράφεται από τις ιδιότητες αρχική ημερομηνία και ημερομηνία λήξης. Εάν το σκεφτείτε, καμία από αυτές τις ιδιότητες δεν περιγράφει τις οντότητες υπάλληλος, θέση, ή πρόγραμμα. Περιγράφουν μία μεμονωμένη περίπτωση σχέσης μεταξύ μιας περίπτωσης από την οντότητα.



Σχήμα 2.7

Ξένα κλειδιά : Μια σχέση υπονοεί ότι οι περιπτώσεις μιας οντότητας είναι και περιπτώσεις μιας άλλης οντότητας. Πρέπει να είμαστε σε θέση να προσδιορίσουμε εκείνες τις περιπτώσεις της οντότητας. Η δυνατότητα να προσδιοριστούν οι συγκεκριμένες σχετικές περιπτώσεις οντοτήτων περιλαμβάνει την καθιέρωση του ξένου κλειδιού. Ένα ξένο κλειδί είναι το αρχικό κλειδί μιας οντότητας που συμβάλλει σε μια άλλη οντότητα για να προσδιορίσει τις περιπτώσεις μιας σχέσης.

Ένα ξένο κλειδί (σε μια οντότητα παιδιών) πάντα ταιριάζει με το αρχικό κλειδί (σε μια οντότητα γονέων). Στο σχήμα 8-6 καταδεικνύεται η έννοια των ξένων κλειδιών με την απλή μοντελοποίηση δεδομένων. Παρατηρούμε ότι ο μέγιστος αριθμός στοιχείων συνόλου για το τμήμα είναι "ένα," ενώ ο μέγιστος αριθμός στοιχείων συνόλου για το πρόγραμμα σπουδών είναι "πολλά." Σε αυτήν την περίπτωση, το τμήμα καλείται οντότητα γονέας και το πρόγραμμα σπουδών είναι η οντότητα παιδί. Το αρχικό κλειδί δίνεται πάντα από το γονέα στο παιδί ως ξένο κλειδί. Κατά συνέπεια, μια περίπτωση προγράμματος σπουδών τώρα έχει ένα ξένο κλειδί όνομα τμημάτων του οποίου η αξία δείχνει την περίπτωση τμήματος που προσφέρει εκείνο το πρόγραμμα σπουδών. (Τα ξένα κλειδιά δεν δίνονται ποτέ από το παιδί στο γονέα.)

Το διάγραμμα ροής δεδομένων (Data flow diagram) απεικονίζει την ροή δεδομένων μέσα στο σύστημα και την δουλειά ή διαδικασία που διενεργείται από το σύστημα. Ένα από τα πλεονεκτήματα των data flow diagrams είναι ότι γίνονται εύκολα κατανοητά, υπάρχουν μόνο τρία σύμβολα και μία σύνδεση:

- Τα ορθογώνια παραλληλόγραμμα απεικονίζουν (ή αντιπροσωπεύουν) τις διαδικασίες ή την εργασία που θα πρέπει να γίνει.
- Τα τετράγωνα απεικονίζουν τους εξωτερικούς παράγοντες (ή τους εξωτερικούς συντελεστές) – το όριο του συστήματος.
- Τα open-ended boxes απεικονίζουν τον χώρο φύλαξης δεδομένων, ορισμένες φορές αποκαλούνται αρχεία ή βάσεις δεδομένων. Αυτός ο χώρος φύλαξης δεδομένων αντιστοιχεί σε όλα τα παραδείγματα σε μια απλή ύπαρξη ενός μοντέλου δεδομένων.

χώρος φύλαξης δεδομένων αντιστοιχεί σε όλα τα παραδείγματα σε μια απλή ύπαρξη ενός μοντέλου δεδομένων.

- Τα βέλη απεικονίζουν διαγράμματα ροής (data flows), ή εισαγωγές και εξαγωγές προς και από τις διαδικασίες.

Μερικές φορές υπάρχει μια τάση να συγχύζονται τα διαγράμματα ροής δεδομένων (data flow diagrams) με τα διαγράμματα ροής (flow charts) επειδή το πρόγραμμα σχεδίασης, συχνά, εμπλέκεται με την χρήση των διαγραμμάτων ροής. Παρόλα αυτά τα διαγράμματα ροής δεδομένων είναι πολύ διαφορετικά. Ας συνοψίσουμε τις διαφορές τους.

- Οι διαδικασίες σε ένα διάγραμμα ροής δεδομένων μπορούν να λειτουργήσουν σε μία παράλληλο. Τοιουτοτρόπως, θα μπορούσαν να εκτελούνται ή να εργάζονται ταυτόχρονα αρκετές διαδικασίες. Αυτό είναι η συνέπεια του τρόπου εργασίας μίας επιχείρησης. Από την άλλη πλευρά, οι διαδικασίες στα διαγράμματα ροής είναι δυνατόν να εκτελέσουν μόνο μία κάθε φορά.
- Τα διαγράμματα ροής δεδομένων δείχνουν μόνο την πτώση των δεδομένων διαμέσου του συστήματος. Τα βέλη τους απεικονίζουν τις πορείες τις οποίες μπορούν να ακολουθήσουν τα δεδομένα. Τυπικά δεν εμφανίζονται τα looping (βρόχοι) και branching (διακλαδώσεις). Από την άλλη, τα διαγράμματα ροής δείχνουν την συχνότητα των διαδικασιών ή των χειρισμών σε ένα σε ένα αλγόριθμο ή σε ένα πρόγραμμα. Τα βέλη τους απεικονίζουν τους στόχους προς την επόμενη διαδικασία ή χειρισμό. Αυτό μπορεί να περιλαμβάνει το looping και το branching.
- Τα διαγράμματα ροής δεδομένων μπορούν να δείξουν διαδικασίες οι οποίες έχουν δραματικά διαφορετικό συγχρονισμό (μεταξύ τους). Για παράδειγμα, ένα απλό διάγραμμα ροής δεδομένων μπορεί να περιλαμβάνει διαδικασίες οι οποίες συμβαίνουν ωριαία, ημερησίως, εβδομαδιαία, ετήσια και έπειτα από απαίτηση. Αυτό δεν συμβαίνει στα διαγράμματα ροής.

Τα διαγράμματα ροής δεδομένων υπήρξαν δημοφιλή για περισσότερο από 20 χρόνια, αλλά το ενδιαφέρον για τα διαγράμματα ροής δεδομένων έχει πρόσφατα ανανεωθεί λόγω της ευελιξίας του στην διαδικασία επανασχεδιασμού μίας επιχείρησης (Business Process Redesign). Αφού οι επιχειρήσεις έχουν αρχίσει να συνειδητοποιούν ότι τα περισσότερα συστήματα επεξεργασίας δεδομένων έχουν μόνο απαρχαιωμένες αυτοματοποιημένες, αναποτελεσματικές και παρακρατικές διαδικασίες επιχείρησης, υπάρχει ένα ανανεωτικό ενδιαφέρον για μία οργανωτική απόδοση διαδικασιών των επιχειρήσεων. Αυτό μπορεί να πραγματοποιηθεί, πρώτα από την μοντελοποίηση των επιχειρηματικών διαδικασιών με σκοπό τη ανάλυση, τον επανασχεδιασμό ή βελτιώνοντας αυτά. Επακόλουθα η τεχνολογία της πληροφορίας είναι δυνατόν να εφαρμοστεί στις βελτιωμένες επιχειρηματικές διαδικασίες με δημιουργικούς τρόπους οι οποίοι μεγεθύνουν την αξία που επιστρέφει στην επιχείρηση.

Οι θεωρίες και οι βασικές «ιδέες» του συστήματος μας βοηθούν να καταλάβουμε τον τρόπο με τον οποίο έχει οργανωθεί και πως δουλεύει. Διάφορες τεχνικές μας μαθαίνουν πώς να εφαρμόσουμε τις θεωρίες και τις ιδέες και να τις υλοποιήσουμε. Εάν κατανοηθούν οι βασικές ιδέες μπορούμε να προσαρμόσουμε καλύτερα τις τεχνικές για να προσπεράσουμε τα προβλήματα και τις δύσκολες καταστάσεις. Αφού γίνει η κατανόηση των βασικών ιδεών τότε μπορούμε να προχωρήσουμε στην φάση σχεδίασης της διαδικασίας μοντελοποίησης.

2.7.4 Σχεδιασμός της διαδικασίας μοντελοποίησης

Σε αυτή την φάση ξεκαθαρίζουμε το πότε θα γίνει ο σχεδιασμός της διαδικασίας μοντελοποίησης, πόσα Process Models θα σχεδιαστούν, και ποια τεχνολογία υποστηρίζει αυτό που θέλουμε να αναπτύξουμε. Ποιο συγκεκριμένα η διαδικασία αυτή χωρίζεται σε έξι στάδια:

- Στρατηγικός σχεδιασμός συστημάτων
- Process Modeling For business Process Redesign
- Διαδικασία Μοντελοποίησης κατά τη διάρκεια της ανάλυσης συστημάτων

- Μια μελλοντική ματιά στο σχεδιασμό συστήματος
- Διερεύνηση και συλλογή πληροφοριών για τη διαδικασία Μοντελοποίησης
- Computer-Aided Systems Engineering For Process Modeling

2.7.5 Στρατηγικός σχεδιασμός συστημάτων

Πολλές επιχειρήσεις επιλέγουν application development process βασισμένες σε στρατηγικά πλάνα πληροφοριακών συστημάτων. Ο στρατηγικός σχεδιασμός είναι ένα ξεχωριστό έργο από το οποίο προκύπτει ένα πλάνο πληροφοριακού συστήματος. Αυτό με τη σειρά του καθορίζει μία συνολική εικόνα και την αρχιτεκτονική για το πληροφοριακό σύστημα. Η αρχιτεκτονική αυτή συχνά περιλαμβάνει ένα Enterprise process model το οποίο προσδιορίζει μόνο επιχειρησιακές περιοχές και λειτουργίες.

2.7.6 Process Modeling For business Process Redesign

Κατά την διαδικασία επιχειρησιακού επανασχεδιασμού (Business Process Redesign) αναλύονται οι επιχειρησιακές διαδικασίες κατόπιν επανασχεδιάζονται για να περιοριστεί η οποιαδήποτε ανεπάρκεια και γραφειοκρατία. Για να γίνει ο επανασχεδιασμός της επιχειρησιακής διαδικασίας πρέπει πρώτα να μελετηθεί η υπάρχουσα διαδικασία.

Κάθε μεθοδολογία B.P.R. προτείνει τις δικές τις κατευθύνσεις και καταγραφές της διαδικασίας μοντελοποίησης. Τα περισσότερα μοντέλα είναι μεταξύ διαγραμμάτων ροής δεδομένων και διαγραμμάτων ροής. Τα διαγράμματα τείνουν να είναι φυσικά επειδή η B.P.R. ομάδα προσπαθεί να μειώσει την αναποτελεσματικότητα που μειώνει την αξία. B.P.R. των διαγραμμάτων ροής δεδομένων / διαγραμμάτων ροής που μπορεί να περιλαμβάνουν νέα σύμβολα και πληροφορίες σχετικά με τον χρόνο, την απόδοση, το κόστος και την αξία. Δοσμένων αυτών των δεδομένων η B.P.R. ομάδα προσπαθεί να απλουστεύσει την διαδικασία και τη ροή των δεδομένων ούτως ώστε να επιτευχθεί μέγιστη αποτελεσματικότητα.

2.7.7 Διαδικασία Μοντελοποίησης κατά τη διάρκεια της ανάλυσης συστημάτων

Η πιο μοντέρνα δομημένη στρατηγική ανάλυσης επικεντρώνει αποκλειστικά στην ανάπτυξη του λογικό μοντέλο του target system. Αντί να κατασκευάζονται από πάνω προς τα κάτω όλο και πιο αναλυτικά, οργανώνονται σύμφωνα με μια κοινή στρατηγική που ονομάζεται event partitioning (σχήμα διαμέρισης συμβάντων). Το event partitioning factors ένα σύστημα σε υποσυστήματα βασισμένο σε επιχειρησιακά γεγονότα και ανταποκρίνεται σε αυτά.

2.7.8 Μια μελλοντική ματιά στο σχεδιασμό συστήματος

Κατά την διάρκεια του σχεδιασμού συστήματος, η λογική διαδικασία μοντελοποίησης θα μετατραπεί σε φυσική διαδικασία μοντελοποίησης για την επιλεγμένη τεχνική αρχιτεκτονική. Αυτό το μοντέλο θα απεικονίζει τις τεχνικές δυνατότητες και τους περιορισμούς τις επιλεγμένης τεχνολογίας.

2.7.9 Διερεύνηση και συλλογή πληροφοριών για τη διαδικασία Μοντελοποίησης

Η Διαδικασία μοντελοποίησης δεν μπορεί να ξεκινήσει χωρίς τα απαραίτητα γεγονότα και τις πληροφορίες που θα παρέχουν οι χρήστες. Τα γεγονότα μπορούν να συλλεχθούν με διάφορες τεχνικές όπως για παράδειγμα δείγμα υπάρχοντων φορμών και αρχείων, έρευνα παρόμοιων συστημάτων, έρευνες χρηστών και management και συνεντεύξεις χρηστών και management. Η πιο γρήγορη μέθοδος συλλογής στοιχείων και πληροφοριών και ταυτόχρονη κατασκευή και επαλήθευση της διαδικασίας μοντελοποίησης είναι η Joint Requirement Planning (JRP).

2.7.10 Computer-Aided Systems Engineering For Process Modeling

Όπως σε όλα τα μοντέλα συστημάτων, η διαδικασία μοντελοποίησης αποθηκεύονται στην βάση. Χρησιμοποιώντας Computer-Aided Systems Engineering (C.A.S.E.) προϊόντα, γίνεται εύκολη η δημιουργία επαγγελματικών και ευανάγνωστων μοντέλων διαδικασιών χωρίς τη χρήση χαρτιού. Τα μοντέλα αυτά μπορούν εύκολα τροποποιηθούν και να διορθωθούν ανάλογα με τις προτάσεις των τελικών χρηστών. Επίσης τα περισσότερα CASE προϊόντα παρέχουν δυναμικά αναλυτικά εργαλεία που ελέγχουν το μοντέλο για μηχανικά λάθη, για την πληρότητα τους και για την συνέπεια τους. Μερικά CASE προϊόντα μπορούν να βοηθήσουν στην ανάλυση της διαδικασίας μοντελοποίησης όσον αφορά την συνέπεια, την πληρότητα και την ευελιξία.

Τα CASE προϊόντα έχουν τους περιορισμούς τους, για παράδειγμα δεν υποστηρίζουν όλα τα μοντέλα διαδικασιών. Σαν αποτέλεσμα ένα CASE προϊόν μπορεί να αναγκάσει την εταιρία να προσαρμόσει τις μεθοδολογίες μοντελοποίησης διαδικασιών για σύμβολα ή προσεγγίσεις έτσι ώστε να είναι συμβατό με το συγκεκριμένο CASE προϊόν.

2.8 Business Drivers

Ας υποθέσουμε ότι είμαστε σε θέση να χτίσουμε ένα πληροφορικό σύστημα. Το ερώτημα είναι *ποιοι συμμετέχουν σε αυτό το σύστημα*; Αυτοί που συμμετέχουν μπορούν να ταξινομηθούν σε τέσσερις κατηγορίες (όπως φαίνεται πιο κάτω). Αξίζει να σημειωθεί ότι ο καθένας από αυτούς που συμμετέχουν στην ανάπτυξη των πληροφοριακών συστημάτων βλέπει διαφορετικά το ίδιο πληροφοριακό σύστημα.

Ιδιοκτήτες Συστήματος : Πληρώνουν για να σχεδιαστεί το σύστημα και να λειτουργήσει. Θέτουν το σκοπό και τις προτεραιότητες για το σύστημα. Το

βλέπουν από την πλευρά του κόστους και των κερδών. Τους ενδιαφέρει να λυθούν τυχόν προβλήματα και να ανακαλύψουν ευκαιρίες.

Χρήστες Συστήματος : Καθορίζουν τις ανάγκες και τις προσδοκίες της επιχείρησης για το νέο σύστημα. Ως εκ τούτου βλέπουν το πληροφοριακό σύστημα από την πρακτική του πλευρά, κατά πόσο είναι χρήσιμο στη δουλειά τους, αν είναι εύκολο στη χρήση και εύκολο στο να το μάθει κανείς.

Σχεδιαστές Συστήματος : Μεταφράζουν τις απαιτήσεις της επιχείρησης σε εφικτή τεχνική λύση. Κατασκευάζουν ένα σχεδιάγραμμα που θα καθοδηγήσει στη κατασκευή του τελικού συστήματος.

Κατασκευαστές Συστήματος : Κατασκευάζουν, επεκτείνουν και διατηρούν το πληροφοριακό σύστημα. Τους ενδιαφέρει να λειτουργεί το λειτουργικό και λογισμικό κομμάτι κατά την εφαρμογή του συστήματος.

Κεφάλαιο 3 ΣΧΕΔΙΑΣΜΟΣ ΣΥΣΤΗΜΑΤΩΝ

3.1 Εισαγωγή

Ο σχεδιασμός πληροφοριακών συστημάτων είναι το επόμενο βήμα από την ανάλυση και καθορίζει τους στόχους που εστιάζουν στην προδιαγραφή μιας λεπτομερούς λύσης βασισμένη σε ηλεκτρονικό υπολογιστή. Ακόμα μπορεί να τον συναντήσουμε και ως φυσικό σχεδιασμό. Κατά συνέπεια, ενώ η ανάλυση συστημάτων επικεντρώνεται στο επιχειρηματικό πρόβλημα, ο σχεδιασμός συστημάτων επικεντρώνεται σε τεχνικά ή προβλήματα εφαρμογής των συστημάτων.

Όπως ήδη έγινε αντιληπτό το σημαντικότερο ρόλο σε αυτή τη φάση τον έχουν οι σχεδιαστές συστημάτων. Ως εκ τούτου εξετάζεται και η τμηματοποίηση πληροφοριακών συστημάτων από την σκοπιά των σχεδιαστών συστημάτων. Οι αναλυτές συστημάτων χρησιμεύουν ως βοηθοί στη φάση του σχεδιασμού συστημάτων.

Οι περισσότεροι συμμετέχοντες στην διαδικασία του σχεδιασμού συστημάτων φτιάχνουν τα σχεδιαγράμματα συστημάτων βασισμένα σε υπολογιστή, για να προγραμματιστούν και να αναπτυχθούν με τη βοήθεια των προγραμματιστών. Κατά συνέπεια, σχεδιάζονται οι εισαγωγές δεδομένων, τα αποτελέσματα, τα αρχεία, οι βάσεις δεδομένων και άλλα στοιχεία υπολογιστών. Οι περισσότερες εταιρείες αγοράζουν περισσότερα προγράμματα από αυτά που παράγουν. Οι προμηθευτές λογισμικού έχουν γράψει επαρκή αλλά σπανίως τέλεια λογισμικά πακέτα που μπορούν να αγοραστούν και να τροποποιηθούν σύμφωνα με τις ανάγκες του τελικού χρήστη.

3.1 Προσεγγίσεις Σχεδιασμού Συστημάτων

Υπάρχουν πολλές στρατηγικές ή τεχνικές για τον σχεδιασμό συστημάτων. Περιλαμβάνουν τον σύγχρονο δομημένο σχεδιασμό, information engineering,

prototyping, JAD, RAD και αντικειμενοστραφή σχεδιασμό. Αυτές οι στρατηγικές αντιμετωπίζονται συχνά ως ανταγωνιστικές, εναλλακτικές προσεγγίσεις στον σχεδιασμό συστημάτων αλλά, και στην πραγματικότητα ορισμένοι σχεδιασμοί συμπληρώνουν ο ένας τον άλλον.

Ας εξετάσουμε εν' συντομία αυτές τις στρατηγικές και τον σκοπό ή τους στόχους των έργων στα οποία έχουν ταιριάξει. Η πρόθεση είναι να αναπτυχθεί ένα υψηλό επίπεδο κατανόησης.

3.1.1 Πρότυπες – Οδηγημένες Προσεγγίσεις

Ο δομημένος σχεδιασμός, Information Engineering και ο αντικειμενοστραφής σχεδιασμός είναι παραδείγματα της πρότυπης δομημένης προσέγγισης. Ο πρότυπος οδηγημένος σχεδιασμός υπογραμμίζει τον σχεδιασμό εικονογραφημένων προτύπων συστημάτων για να τεκμηριώσει τις τεχνικές ή τις πτυχές εφαρμογής ενός νέου συστήματος.

Τα πρότυπα σχεδιασμού προέρχονται συχνά από λογικά μοντέλα που είχαν σχεδιαστεί νωρίτερα στην πρότυπη οδηγημένη ανάλυση. Τελικά τα πρότυπα σχεδιασμού συστημάτων γίνονται τα σχεδιαγράμματα για την κατασκευή και την εφαρμογή του νέου συστήματος.

Σήμερα, οι πρότυπες οδηγημένες προσεγγίσεις σχεδόν πάντα ενισχύονται από την χρήση αντικειμενοστραφών εργαλείων. Μερικοί σχεδιαστές σχεδιάζουν πρότυπα συστήματα με γραφιστικά προγράμματα όπως είναι το Visio Professional ή το Corel Flow. Άλλοι σχεδιαστές και οργανισμοί απαιτούν την χρήση εργαλείου βασισμένου σε βάση δεδομένων ή εργαλείων μοντελοποίησης όπως το Architect 2001, Visio Enterprise, Visible Analyst και Rational Rose. Τα εργαλεία CASE προσφέρουν την συνέπεια και την πληρότητα καθώς επίσης και το βασισμένο στους κανόνες έλεγχου λάθους.

Ας εξετάσουμε τις πιο διαδεδομένες πρότυπες οδηγημένες προσεγγίσεις σχεδιασμού.

3.1.2 Σύγχρονος δομημένος σχεδιασμός

Η τεχνική δομημένου σχεδιασμού βοηθά τους σχεδιαστές να εξετάσουν το μέγεθος και την πολυπλοκότητα των προγραμμάτων. Ο σύγχρονος δομημένος σχεδιασμός είναι μία διαδικασία (προσανατολισμένη τεχνική) για ένα μεγάλο πρόγραμμα σε μία ιεραρχία οντοτήτων που οδηγούν σε ένα πρόγραμμα υπολογιστών που είναι ευκολότερο να διατηρηθεί. Τον σύγχρονο δομημένο προγραμματισμό μπορούμε να τον βρούμε και ως από πάνω προς τα κάτω σχεδιασμό προγράμματος.

Ο σχεδιασμός ενός προγράμματος από πάνω προς τα κάτω γίνεται με ιεραρχία οντοτήτων. Η οντότητα μπορεί να αποτελείται από οδηγίες παραγράφου, έναν φραγμό, ένα υποπρόγραμμα ή μία υπορουτίνα. Η από πάνω προς τα κάτω δομή αυτών των οντοτήτων αναπτύσσεται σύμφωνα με τους διάφορους κανόνες και τις οδηγίες σχεδιασμού.

Ο δομημένος σχεδιασμός θεωρείται process oriented τεχνική επειδή δίνει έμφαση στην process building blocks στο πληροφοριακό μας σύστημα, συγκεκριμένα στις διαδικασίες λογισμικού. Ο δομημένος σχεδιασμός επιδιώκει ένα πρόγραμμα στην από πάνω προς τα κάτω διαδικασία των οντοτήτων να έχουν τις ακόλουθες ιδιότητες :

- Οι οντότητες πρέπει να είναι ιδιαίτερα συνεκτικές δηλαδή κάθε οντότητα πρέπει να ολοκληρώσει μια και μόνο λειτουργία. Αυτό καθιστά τις ενότητες επαναχρησιμοποιήσιμες στα μελλοντικά προγράμματα.
- Οι ενότητες πρέπει να συνδεθούν αόριστα, με άλλα λόγια πρέπει να είναι όσο το δυνατόν λιγότερο εξαρτημένες μεταξύ τους. Αυτό ελαχιστοποιεί την πιθανότητα σε μελλοντικές αλλαγές σε μια ενότητα ,να έχουμε αλλαγές και σε άλλες.

Το λογισμικό μοντέλο που προέρχεται από δομημένο σχεδιασμό καλείται διάγραμμα δομών (Structure Charts). Το διάγραμμα δομών παράγεται με την μελέτη της ροής των στοιχείων μέσω του προγράμματος. Ο δομημένος σχεδιασμός εκτελείται κατά την διάρκεια του σχεδιασμού συστημάτων. Δεν

εξετάζει όλες τις πτυχές σχεδιασμού για κάθε περίπτωση, δηλαδή ο δομημένος σχεδιασμός δεν θα μας βοηθήσει να σχεδιάσουμε αποτελέσματα ή βάσεις δεδομένων.

Ο δομημένος σχεδιασμός έχει χάσει κάπως την δημοτικότητα, με πολλές από τις σημερινές εφαρμογές, να απαιτούν νεότερες τεχνικές που εστιάζουν στις Event Driven και αντικειμενοστραφείς τεχνικές προγραμματισμού. Εν τούτοις, είναι ακόμα μια δημοφιλής τεχνική που περιλαμβάνει το σχεδιασμό εφαρμογών των κεντρικών υπολογιστών, βασισμένη σε εφαρμογή που χρησιμοποιείται για την σύζευξη και την συνοχή στο επίπεδο του συστήματος.

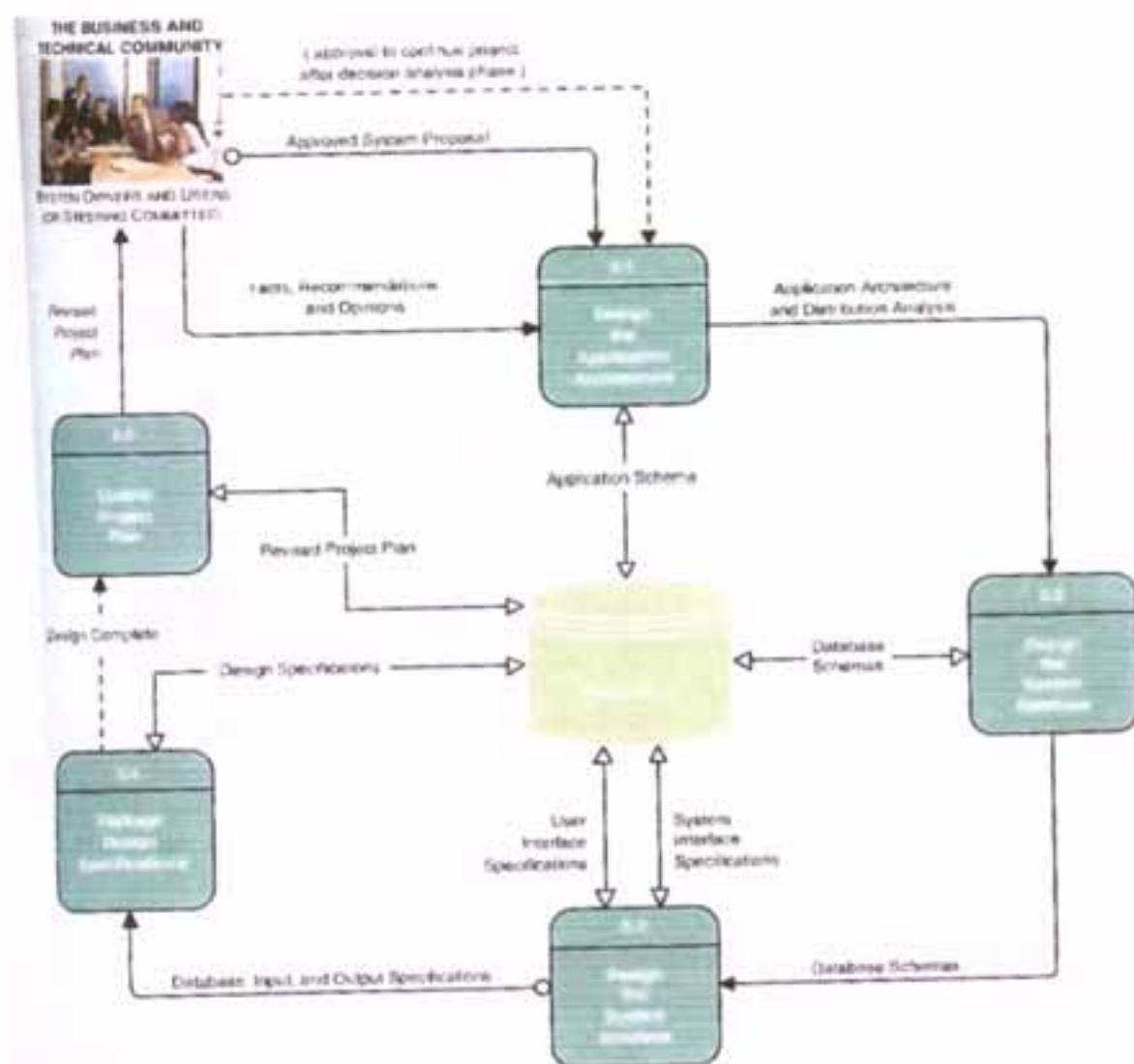
3.1.3 Αντικειμενοστραφείς σχεδιασμός

Είναι η νεότερη στρατηγική σχεδιασμού. Οι αντικειμενοστραφείς τεχνολογίες και τεχνικές προσπαθούν να περιορίσουν τον διαχωρισμό μεταξύ των δεδομένων και των διαδικασιών. Οι τεχνικές αντικειμενοστραφούς σχεδιασμού, χρησιμοποιούνται για τον καθορισμό απαιτήσεων των αντικειμένων, οι οποίες έχουν καθοριστεί κατά τη διάρκεια της ανάλυσης.

Ας ξεκινήσουμε τοποθετώντας τον σχεδιασμό συστημάτων για την ανάπτυξη εσωτερικών έργων στο πλαίσιο του κύκλου ζωής. Το ζητούμενο της φάσης του σχεδιασμού έχει δύο σκέλη. Στο πρώτο ο αναλυτής επιδιώκει ένα σύστημα που εκπληρώνει τόσο τις απαιτήσεις που προκύπτουν από το πρόβλημα όσο και αυτές των τελικών χρηστών για ένα φιλικό περιβάλλον διεπαφής. Τον κεντρικό ρόλο σε αυτό το κομμάτι του σχεδιασμού έχουν οι μηχανικοί συστημάτων. Στο δεύτερο και εξίσου σημαντικό κομμάτι ο αναλυτής επιδιώκει να παρουσιάσει ξεκάθαρα και ολοκληρωμένα τις προδιαγραφές στους προγραμματιστές και τους τεχνικούς.

3.2 Στόχοι του σχεδιασμού συστημάτων

Το παρακάτω διάγραμμα (σχήμα 3.1) απεικονίζει τις εργασίες που πρέπει να διενεργηθούν για την ολοκλήρωση της φάσης του σχεδιασμού. Στο διάγραμμα αυτό δεν χρησιμοποιείται κάποια συγκεκριμένη μεθοδολογία, παρακάτω όμως θα αναλύσουμε τις προσεγγίσεις, τα εργαλεία και τις τεχνικές που μπορεί να χρειαστούν για την εκπλήρωση του συγκεκριμένου στόχου. Το διάγραμμα αυτό αποτελεί ένα πρότυπο.



Σχήμα 3.1

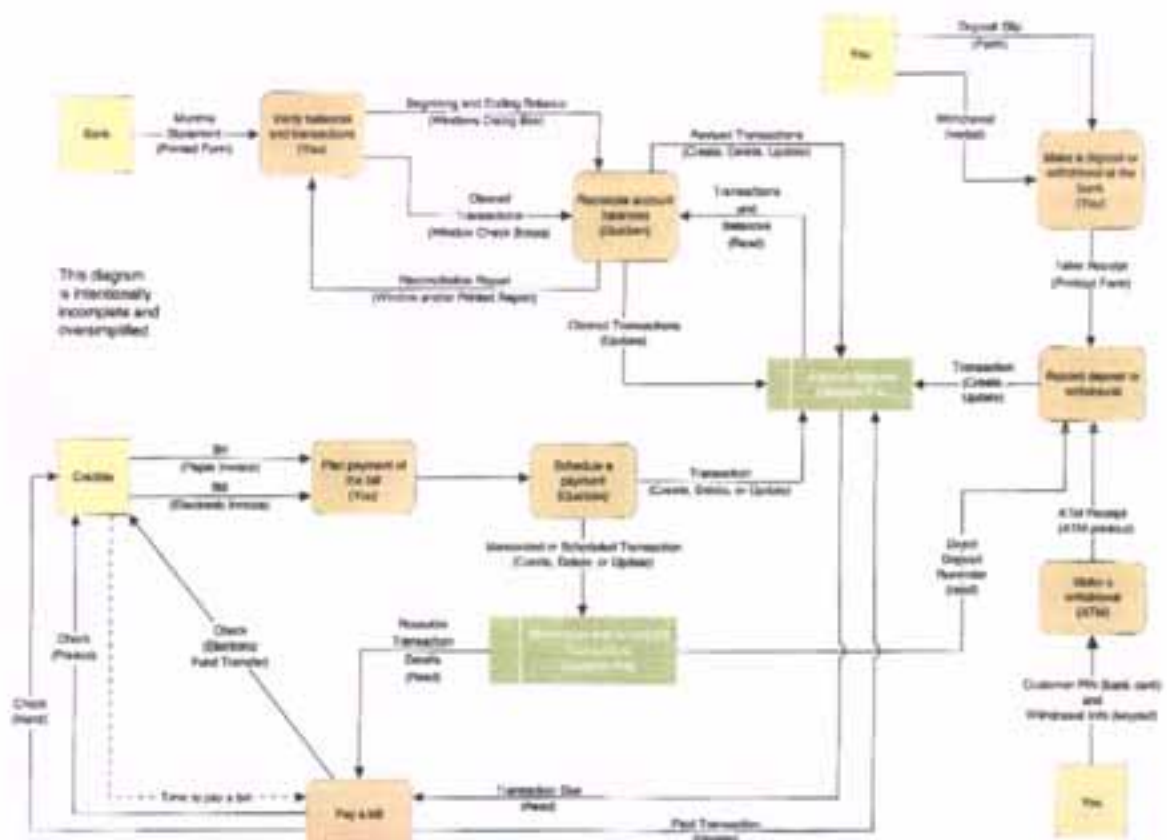
Η ομάδα του έργου και ο υπεύθυνος μπορεί να πληθήνουν επίσης μπορούν να αλλάξουν το πρότυπο για να απεικονίσουν τις μοναδικές ανάγκες οποιουδήποτε προβλήματος. Οι στόχοι του σχεδιασμού συστημάτων είναι οι εξής:

- Design the Application Architecture
- Σχεδιασμός της βάσης δεδομένων
- Σχεδιασμός του περιβάλλοντος διεπαφής του συστήματος
- Package Design Specifications
- Update The Project Plan

3.2.1 Design the Application Architecture

Σκοπός του πρώτου στόχου του σχεδιασμού είναι να διευκρινισθεί μια application architecture για την εφαρμογή. Η application architecture καθορίζει τις τεχνολογίες που θα χρησιμοποιηθούν από ένα, περισσότερα ή και όλα τα πληροφοριακά συστήματα από την σκοπιά των δεδομένων, των διαδικασιών, του περιβάλλοντος διεπαφής με τον χρήστη και των τμημάτων δικτύου. Κατά συνέπεια, ο σχεδιασμός του application architecture περιλαμβάνει την εξέταση τεχνολογιών δικτύου και την λήψη των αποφάσεων σχετικά με τον τρόπο με τον οποίο τα δεδομένα, οι διαδικασίες και το περιβάλλον διεπαφής πρέπει να διανεμηθούν μεταξύ των επιχειρησιακών θέσεων.

Ο στόχος αυτός ολοκληρώνεται με την ανάλυση της μοντελοποίησης των δεδομένων και των διαδικασιών που δημιουργήθηκαν κατά τη διάρκεια της ανάλυσης απαιτήσεων. Λαμβάνοντας υπ' όψη την μοντελοποίηση δεδομένων, διαδικασιών και την ζητούμενη λύση πρέπει να γίνει η λήψη αποφάσεων ξεχωριστά για το καθένα. Δεδομένου ότι οι αποφάσεις σχετικά με όλα τα παραπάνω έχουν παρθεί είναι στοιχειοθετημένες. Ένα παράδειγμα είναι το physical data flow diagram (P.D.F.D.) που χρησιμοποιείται για να καθιερώσει φυσικές διαδικασίες και βάσεις δεδομένων μέσω ενός δικτύου.



Σχήμα 3.2

Για να ολοκληρωθεί αυτή η διαδικασία ο αναλυτής πρέπει να βοηθηθεί από τους σχεδιαστές συστημάτων και τους χρήστες. Οι χρήστες βοηθούν στην κατηγοριοποίηση δεδομένων, διαδικασιών και θέματα τοποθεσίας. Πολλοί διαφορετικοί σχεδιαστές πρέπει να συγχρονιστούν για τη ολοκλήρωση αυτής της διαδικασίας, συμπεριλαμβανομένου ενός administrator της βάσης δεδομένων, ενός administrator δικτύου, μηχανικών, administrator εφαρμογής και πολλούς άλλους ειδικούς.

Οι βασικές εισαγωγές σε αυτό το στόχο είναι τα γεγονότα, οι προτάσεις και οι γνώμες από διάφορες πηγές. Επίσης σημαντική είναι η πρόταση για το

σύστημα από το στάδιο της ανάλυσης αποφάσεων. Το βασικό παραδοτέο του στόχου είναι η application architecture και distribution analysis που χρησιμεύει σαν σχεδιάγραμμα για τις επόμενες λεπτομερείς δραστηριότητες της φάσης του σχεδιασμού.

3.2.2 Σχεδιασμός της βάσης δεδομένων

Τυπικά ο επόμενος στόχος σχεδιασμού συστήματος είναι η ανάπτυξη σχεδιασμού προδιαγραφών βάσεων δεδομένων. Ο σχεδιασμός των βάσεων δεδομένων δεν είναι ένα απλό σχεδιάγραμμα από records. Οι βάσεις δεδομένων είναι κοινόχρηστες και μπορούν να χρησιμοποιηθούν από πολλά συστήματα. Μελλοντικά προγράμματα μπορεί να τις χρησιμοποιήσουν με τρόπους που δεν είχαν προβλεφθεί αρχικά. Συνεπώς οι αναλυτές οι οποίοι συμμετέχουν στην μοντελοποίηση της βάσης δεδομένων διευκολύνουν αυτό τον στόχο, ενώ οι σχεδιαστές είναι υπεύθυνοι για την ολοκλήρωση αυτής της δραστηριότητας. Ο administrator δεδομένων μπορεί να συμμετέχει, ή ακόμα και να ολοκληρώσει, τον σχεδιασμό της βάσης δεδομένων. Αναγνωρίζοντας ότι συνήθως τα νέα συστήματα χρησιμοποιούν κάποια δεδομένα από μια ήδη υπάρχουσα βάση. Αυτό είναι το σημείο όπου η γνώση του administrator είναι σημαντική.

Οι προγραμματιστές του συστήματος αν τους ζητηθεί μπορεί να χτίσουν μια prototype βάση δεδομένων για το έργο. Το παραδοτέο του στόχου συμπεριλαμβάνει τα σχήματα των βάσεων δεδομένων που έχουν προκύψει. Ένα σχήμα βάσης δεδομένων είναι το δομικό μοντέλο για μια βάση δεδομένων. Είναι μια φωτογραφία (εικόνα) ή χάρτης των εγγραφών και σχέσεων που εφαρμόζονται από την βάση δεδομένων.

3.2.3 Σχεδιασμός του περιβάλλοντος διεπαφής του συστήματος

Από την στιγμή που έχει σχεδιαστεί η βάση δεδομένων και πιθανός έχει χτιστεί ένα prototype, ο σχεδιαστής μπορεί να συνεργαστεί με τους χρήστες για να αναπτύξουν τα εισερχόμενα, εξερχόμενα και τις προδιαγραφές του διαλόγου του συστήματος. Επειδή οι τελικοί χρήστες και οι διευθυντές θα πρέπει να

δουλέψουν με εισερχόμενα και εξερχόμενα ο σχεδιαστής πρέπει να είναι προσεκτικός, να ζητήσει τις ιδέες και τις προτάσεις τους ειδικά αυτές που αφορούν το format. Οι ιδέες και οι απόψεις τους πρέπει να είναι εύκολες στην εκμάθηση και στην χρήση έτσι ώστε ο διάλογος του νέου συστήματος να είναι εύκολος. Τα αποτελέσματα συναλλαγής θα σχεδιαστούν συχνά σαν προεκτυπωμένες μορφές πάνω στις οποίες οι λεπτομέρειες τη συναλλαγής θα τυπωθούν. Αναφορές και άλλα εξερχόμενα συνήθως τυπώνονται απευθείας στο χαρτί ή παρουσιάζονται στην τερματική οθόνη. Τα ακριβές format και σχεδιάγραμμα των εξερχόμενων πρέπει να είναι προκαθορισμένο. Τέλος οι εσωτερικοί έλεγχοι πρέπει να είναι συγκεκριμένοι ώστε να εξασφαλισθεί ότι τα εξερχόμενα δεν θα χαθούν ή θα πάνε σε λάθος διεύθυνση ή θα είναι ελλιπή.

Για τον σχεδιασμό των εισερχομένων είναι σημαντικό να χρησιμοποιείται η μέθοδος data capture. Για παράδειγμα μπορεί να σχεδιαστεί μια φόρμα στην οποία για να εισέλθουν τα δεδομένα θα πρέπει πρώτα να έχουν καταγραφεί. Θέλουμε να είναι εύκολη η καταγραφή των δεδομένων στην φόρμα αλλά επίσης να απλοποιήσουμε την είσοδο δεδομένων από την φόρμα στον ηλεκτρονικό υπολογιστή. Αυτό κυρίως γιατί απευθύνεται και σε άτομα που δεν έχουν πολλές γνώσεις business application. Επίσης κάθε φορά που εισάγουμε τα δεδομένα στο σύστημα μπορεί να κάνουμε λάθη, για αυτό πρέπει να έχουμε ελέγχους έκδοσης ώστε να εξασφαλίσουμε την ακρίβεια των δεδομένων.

Για το περιβάλλον διεπαφής, ο σχεδιασμός πρέπει να έχει προβλέψει παράγοντες όπως η συμβατότητα με τα τερματικά, πιθανά λάθη ή μη κατανόηση από τον τελικό χρήστη. Για αυτό και είναι απαραίτητες επιπλέον οδηγίες ή βοήθεια σε συγκεκριμένα σημεία καθώς και screen content and layout. Προσπαθούμε να προβλέψουμε κάθε μικρό λάθος ή λάθος πάτημα κουμπιού που μπορεί να κάνει ο τελικός χρήστης, ακόμα και αν είναι απίθανο. Επιπλέον προσπαθούμε να κάνουμε κατανοητό για τον τελικό χρήστη αυτό που βλέπει στην οθόνη του.

Σε αυτή την δραστηριότητα εμπλέκονται οι χρήστες. Τα εισερχόμενα, τα εξερχόμενα και οι διάλογοι διεπαφής είναι αυτά που θα συναντούν στη δουλειά τους. Κυρίως εμπλέκονται στις προσπάθειες σχεδιασμού που αφορούν

prototyping. Θα τους ζητηθεί να παρέχουν ανατροφοδότηση σχετικά με κάθε εισερχόμενο και εξερχόμενο πρωτότυπο. Οι σχεδιαστές είναι υπεύθυνοι για την ολοκλήρωση αυτής της δραστηριότητας και ιδιαίτερα αυτή με ειδικότητα γραφιστικού σχεδιασμού περιβάλλοντος διεπαφής του χρήστη. Επίσης οι κατασκευαστές συστήματος μπορούν να κατασκευάσουν μία ποικιλία screen designs για τους χρήστες που αναθεωρούν κατά τη διάρκεια του σχεδιασμού από τη διαμόρφωση πρωτοτύπου.

Το παραδοτέο του στόχου του σχεδιασμού είναι η ολοκλήρωση της βάσης δεδομένων και των προδιαγραφών των εισερχομένων και των εξερχομένων.

3.2.4 Διαχωρισμός προδιαγραφών σχεδιασμού σε πακέτα

Ο τελικός στόχος σχεδιασμού διαχωρίζει σε πακέτα όλες τις προδιαγραφές από τους προηγούμενους στόχους σχεδιασμού που θα καθοδηγήσουν τον προγραμματιστή κατά τη διάρκεια της φάσης ανάπτυξης μεθοδολογίας συστήματος.

Σε αυτό τον στόχο εκτός από τον διαχωρισμό των πακέτων μας απασχολεί και κατά πόσο εξαρτάται από δύο παράγοντες. Τον καθορισμό της διαχωριστικής γραμμής ανάμεσα στις αρμοδιότητες του σχεδιαστή και του προγραμματιστή και τον σχεδιασμό της γενικής δομής του προγράμματος εάν η μεθοδολογία και η λύση το απαιτούν. Οι περισσότεροι οργανισμοί έχουν υιοθετήσει επιταχυνόμενες προσεγγίσεις ανάπτυξης που δεν απαιτούν τα τελευταία. Η δομή του προγράμματος εξέτασε τα ποιοτικά ζητήματα που ήταν αρμοδιότητα των σχεδιαστών που χρησιμοποιούσαν παλαιότερες γλώσσες προγραμματισμού και έτειναν να είναι εφαρμογές βασισμένες σε υπολογιστή.

Ο αναλυτής συστημάτων, που μπορεί να βοηθηθεί από το σχεδιαστή, συνήθως ολοκληρώνει το στόχο. Πριν να συνεχίσει στο package των προδιαγραφών του συστήματος και στη φάση κατασκευής, οι σχεδιαστές θα πρέπει να έχουν επανεξετάσει τις προδιαγραφές με το κατάλληλο κοινό. Καθώς οι χρήστες έχουν ήδη δει και δεχθεί τα εισερχόμενα, τα εξερχόμενα και το περιβάλλον διεπαφής για το νέο σύστημα, τις γενικές εργασίες και τη ροή

δεδομένων, πρέπει να πάρουν την τελική έγκριση. Οι ιδιοκτήτες συστημάτων έχουν την τελευταία ευκαιρία να αμφισβητήσουν την δυνατότητα του έργου και να καθορίσουν αν τα πρόγραμμα χρειάζεται να προσαρμοστεί ή να τερματιστεί ή να εγκριθεί για να συνεχιστεί η κατασκευή. Σε αυτό το στάδιο του έργου το audit staff μπορεί να αναμειχθεί σε μεγάλο βαθμό. Το προσωπικό θα περάσει την κρίση για τους εσωτερικού ελέγχους σε ένα νέο σύστημα.

Μόλις τα εισερχόμενα, τα εξερχόμενα και οι προδιαγραφές, που δημιουργήσαμε νωρίτερα, ελεγχθούν, γίνουν αποδεκτά και οργανωθούν όντας πλέον κατάλληλα για την κατασκευή του νέου συστήματος γίνονται διαθέσιμα στην ομάδα των κατασκευαστών μέσω του repository του έργου.

3.2.5 Αναβάθμιση του σχεδίου του έργου

Τώρα που πλησιάζουμε στην ολοκλήρωση της φάσης του σχεδιασμού, πρέπει να επανεξετάσουμε την εφικτότητα του έργου και να αναπροσαρμόσουμε το πλάνο του έργου αναλόγως. Ο υπεύθυνος έργου από κοινού με τους ιδιοκτήτες συστήματος και την υπόλοιπη ομάδα του έργου, βοηθούν σε αυτό τον στόχο. Το σημαντικότερο ρόλο σε αυτό το στόχο έχουν οι αναλυτές και οι ιδιοκτήτες συστήματος. Οι αναλυτές και οι ιδιοκτήτες πρέπει να εξετάσουν τις δυνατότητες, βασισμένοι στην ολοκληρωμένη εργασία σχεδιασμού, του γενικού προγραμματισμού του έργου, τον προϋπολογισμό δαπανών και άλλες εκτιμήσεις που πρέπει να ρυθμισθούν.

Το παραδοτέο αυτού του στόχου είναι το αναπροσαρμοσμένο πλάνο του έργου. Το αναπροσαρμοσμένο πλάνο πρέπει να περιλαμβάνει ένα λεπτομερές πλάνο για την φάση της κατασκευής που πρέπει να ακολουθηθεί.

3.2.6 Αγορά λογισμικού

Εάν κριθεί απαραίτητο από την εταιρία ότι χρειάζεται νέο λογισμικό, γίνεται η επιλογή του κατάλληλου προϊόντος. Η επιλογή είναι περίπλοκη όσον αφορά τα

τεχνικά, τα οικονομικά ζητήματα αλλά και την πολιτική της εταιρίας. Μια λανθασμένη απόφαση μπορεί να καταστρέψει μια σωστή ανάλυση και ένα σωστό σχεδιασμό.

Οι αναλυτές συστημάτων είναι αυτοί που θα κάνουν την επιλογή της κατάλληλης λύσης. Οι περισσότεροι από αυτούς ενημερώνονται από σοβαρά έντυπα και από την αναζήτηση στο διαδίκτυο.

Μετά την έρευνα αγοράς οι αναλυτές ζητούν προτάσεις από τους προμηθευτές οι οποίες ταξινομούνται και αξιολογούνται. Για να συγκεντρωθούν οι προτάσεις σχεδιάζονται δύο έγγραφα : αίτημα για τις αναφορές και αίτημα για τις προτάσεις. Το αίτημα για τις αναφορές χρησιμοποιείται όταν έχει αποφασιστεί ποιο θα είναι το συγκεκριμένο προϊόν αλλά έχουμε την δυνατότητα να επιλέξουμε μεταξύ πολλών προμηθευτών. Το αίτημα για τις προτάσεις χρησιμοποιείται όταν έχουμε να επιλέξουμε μεταξύ προϊόντων και προμηθευτών με σκοπό να δημιουργηθεί ανταγωνισμός μεταξύ των προμηθευτών και να μας γίνει καλύτερη πρόταση.

Μετά την αποστολή των αιτήσεων στους υποψήφιους προμηθευτές οι σχεδιαστές συστημάτων απορρίπτουν αυτές που δεν ικανοποιούν τις απαιτήσεις μας. Οι εναπομείναντες αιτήσεις αξιολογούνται και ταξινομούνται. Αφού οι αιτήσεις έχουν ταξινομηθεί, η καλύτερη θα προταθεί στην διοίκηση για δοθεί η τελική έγκριση. Εφόσον εγκριθεί η εταιρία θα διαπραγματευτεί συμβόλαιο με το προμηθευτή.

3.3 Αρχιτεκτονική και μοντελοποίηση εφαρμογών

Τι είναι η αρχιτεκτονική εφαρμογών:

Η αρχιτεκτονική εφαρμογών είναι ο προσδιορισμός των τεχνολογιών που χρησιμοποιούνται για την εφαρμογή των πληροφοριακών συστημάτων.

Η αρχιτεκτονική πληροφοριακών εφαρμογών και ο μοντελισμός των φυσικών διαδικασιών περιλαμβάνουν τεχνικές για τη διάδοση των γνώσεων, των διαδικασιών και της επικοινωνίας σε δικτυακούς τόπους στο πλαίσιο ενός

διασκορπισμένους υπολογιστικού περιβάλλοντος. Τα διαγράμματα ροή φυσικών δεδομένων χρησιμοποιούνται για την τεκμηρίωση της αρχιτεκτονικής και των σχεδίων σύμφωνα με σχεδιαστικές μονάδες που μπορούν να σχεδιαστούν, να αποτελέσουν πρωτότυπα ή να κατασκευαστούν λεπτομερέστερα και να χρησιμοποιηθούν ως αυτόνομα υποσυστήματα.

3.3.1 Διαγράμματα ροής φυσικών δεδομένων

Τα Διαγράμματα ροής δεδομένων (DFD) χρησιμοποιούνται ως εργαλεία ανάλυσης των συστημάτων για τη μοντελοποίηση των *λογικών* (δηλαδή όχι των τεχνικών) επιχειρηματικών προδιαγραφών ενός πληροφοριακού συστήματος. Επεκτείνοντας λίγο τη γραφική γλώσσα, το DFD. Μπορούν να χρησιμοποιηθούν επίσης ως εργαλεία σχεδιασμού συστημάτων για τη μοντελοποίηση της *φυσικής* (δηλαδή της τεχνικής) αρχιτεκτονικής και σχεδιασμού ενός πληροφοριακού συστήματος.

Τα διαγράμματα ροής φυσικών δεδομένων απεικονίζουν τις τεχνικές και ανθρώπινες σχεδιαστικές αποφάσεις που πρέπει να υλοποιηθούν στο πλαίσιο του πληροφοριακού συστήματος. Αναφέρουν τις τεχνικές επιλογές και άλλες σχεδιαστικές αποφάσεις σε αυτούς που θα κατασκευάσουν και θα υλοποιήσουν το σύστημα. Με άλλα λόγια διαγράμματα ροής φυσικών δεδομένων αποτελούν ένα τεχνικό προσχέδιο για την κατασκευή και την υλοποίηση των συστημάτων. Ποιο κάτω θα εξετάσουμε τις γραφικές συμβάσεις των *διαγραμμάτων ροής φυσικών δεδομένων*, αυτές είναι :

3.3.1.1 Φυσικές διαδικασίες

Θυμηθείτε ότι οι διαδικασίες είναι τα βασικά σχήματα κάθε DFD. Γι' αυτό το λόγο ονομάζονται μοντέλα διαδικασιών. Τα διαγράμματα ροής φυσικών δεδομένων αποτυπώνουν τη σχεδιασμένη, φυσική υλοποίηση κάθε διαδικασίας. Μια *φυσική διαδικασία* είναι είτε ένας *επεξεργαστής*, όπως ένας υπολογιστής ή ένας άνθρωπος, είτε η τεχνική υλοποίηση μιας συγκεκριμένης εργασίας η οποία

πρέπει να πραγματοποιηθεί, όπως ένα πρόγραμμα υπολογιστή ή μια χειροκίνητη διαδικασία.

3.3.1.2 Εξωτερικοί παράγοντες

Οι εξωτερικοί παράγοντες μεταφέρονται από τα διαγράμματα ροής λογικών δεδομένων στα διαγράμματα ροής φυσικών δεδομένων χωρίς αλλαγές. Γιατί; Εξ ορισμού οι εξωτερικοί παράγοντες τοποθετούνται κατά την ανάλυση των συστημάτων εκτός του πλαισίου των συστημάτων και γι' αυτό δεν υπόκεινται σε αλλαγές. Μόνο μια αλλαγή στις προδιαγραφές μπορεί να προκαλέσει αλλαγή ενός εξωτερικού παράγοντα.

3.3.1.3 Αποθήκευση δεδομένων

Η αποθήκευση φυσικών δεδομένων υποδηλώνει ένα από τα παρακάτω

- 1) μια βάση δεδομένων
- 2) έναν πίνακα μιας βάσης δεδομένων
- 3) ένα αρχείο υπολογιστή
- 4) ένα αντίγραφο ταινίας ή μέσου για κάτι σημαντικό
- 5) κάποιο προσωρινό αρχείο ή δεσμίδα που είναι απαραίτητα σε κάποιο πρόγραμμα
- 6) οποιοδήποτε είδος μη μηχανογραφημένου αρχείου.

Όταν οι περισσότεροι άνθρωποι σκέφτονται την αποθήκευση δεδομένων έρχονται στο νου τους αρχεία υπολογιστών και βάσεις δεδομένων. Ωστόσο πολλές αποθήκες δεδομένων δεν είναι μηχανογραφημένες. Αμέσως σκεφτόμαστε τα ερμάρια των αρχείων ή τα έντυπα έγγραφα. Ωστόσο οι περισσότερες επιχειρήσεις ξεχειλίζουν από άλλα είδη χειρόγραφων αποθηκών δεδομένων όπως είναι οι κάρτες διευθύνσεων, οι έντυποι κατάλογοι, οι σημειώσεις διάφορων σημαντικών και επαναχρησιμοποιήσιμων πληροφοριών, τα εγχειρίδια προτύπων, τα εγχειρίδια τεκμηριωμένων διεργασιών, οι τηλεφωνικοί κατάλογοι, κτλ. Παρά τις προβλέψεις για την εξάλειψη των έντυπων αρχείων αυτά θα παραμένουν βασικό τμήμα πολλών συστημάτων στο άμεσο μέλλον -

τουλάχιστον γιατί πρώτον υπάρχει μια φυσική άνεση με το χαρτί και δεύτερον συχνά τα ζητούν οι κυβερνήσεις.

Κάποια προγράμματα απαιτούν τη δημιουργία προσωρινών αρχείων που λειτουργούν σαν ουρά ή περιοχή προσωπικής αποθήκευσης μεταξύ των φυσικών διαδικασιών που εκτελούνται ταυτόχρονα. Τα αρχεία αυτά καταγράφονται με τον ίδιο τρόπο, μόνο που τα ονόματά τους υποδηλώνουν τον προσωρινό χαρακτήρα τους.

Οι φυσικές διεργασίες, οι ροές δεδομένων, οι εξωτερικοί παράγοντες και οι αποθήκευση δεδομένων αποτελούν τα διαγράμματα ροής φυσικών δεδομένων. Αυτά τα διαγράμματα απεικονίζουν την προτεινόμενη ή προγραμματισμένη αρχιτεκτονική ενός πληροφοριακού συστήματος. Στη συνέχεια μπορούμε να χρησιμοποιήσουμε αυτό το φυσικό μοντέλο για να σχεδιάσουμε της εσωτερικές και εξωτερικές λεπτομέρειες κάθε αποθήκης δεδομένων και κάθε ροής δεδομένων.

3.3.1.4 Ροή δεδομένων

Θυμηθείτε ότι όλες οι διαδικασίες ενός DFD πρέπει να έχουν μία ροή δεδομένων εισόδου και εξόδου. Η ροή φυσικών δεδομένων παρουσιάζει ένα από τα παρακάτω:

- 1) τη σχεδιασμένη εφαρμογή ενός στοιχείου εισόδου ή εξόδου μιας φυσικής διεργασίας
- 2) μια εντολή που έχει σχέση με μια βάση δεδομένων ή ενέργειες όπως: δημιουργία, ανάγνωση, ενημέρωση ή κατάργηση
- 3) την εισαγωγή ή την εξαγωγή δεδομένων προς ένα διαφορετικό πληροφοριακό σύστημα ενός δικτύου
- 4) τη ροή δεδομένων ανάμεσα σε δύο διαφορετικά δομήματα ή υπορουτίνες εντός του ίδιου προγράμματος.

Τα διαγράμματα ροής φυσικών δεδομένων πρέπει να υποδεικνύουν επίσης όλες τις ροές δεδομένων που πρέπει να εκτελεστούν στο πλαίσιο των επιχειρηματικών δομών.

3.3.2 Αρχιτεκτονική κατανεμημένων συστημάτων

Τα σύγχρονα πληροφοριακά συστήματα δεν είναι πια μονολιθικά συστήματα βασισμένα σε έναν κεντρικό υπολογιστή. Αντίθετα δημιουργούνται με το συνδυασμό πολλών δικτύων για τη δημιουργία κατανεμημένων συστημάτων. *Κατανεμημένο σύστημα* είναι αυτό του οποίου οι συνιστώσες είναι διασκορπισμένες σε πολλούς τόπους και σε δίκτυα υπολογιστών. Ομοίως το ωφέλιμο φορτίο της επεξεργασίας που απαιτείται για τη στήριξη αυτών των συνιστωσών κατανέμεται σε πολλούς υπολογιστές ενός δικτύου.

3.3.2.1 Σύστημα διαχείρισης κατανεμημένων σχεσιακών βάσεων

Η υποκείμενη τεχνολογία πελάτη/διακομιστή και δικτυακών συστημάτων έδωσε τη δυνατότητα διάδοσης των δεδομένων χωρίς την απώλεια του ελέγχου. Ο έλεγχος αυτός επιτυγχάνεται μέσω των εξελίξεων που παρατηρούνται στην τεχνολογία των κατανεμημένων σχεσιακών βάσεων δεδομένων. Κάθε αρχείο περιγράφεται με τη μορφή ενός πίνακα. Κάθε πεδίο αποτελεί μία στήλη του πίνακα. Κάθε καταγραφή του αρχείου αποτελεί μια σειρά του πίνακα. Οι σχεσιακές καταγραφές μεταξύ δύο πινάκων περιγράφονται με την επανάληψη των στηλών και στους δύο πίνακες. Μια *κατανεμημένη σχεσιακή βάση δεδομένων* κατανέμει ή επαναλαμβάνει πίνακες για να αυξήσει τους διακομιστές βάσεων δεδομένων που υπάρχουν σε σημαντικούς γεωγραφικούς τόπους (όπως διαφορετικά τμήματα πωλήσεων). Το λογισμικό που απαιτείται για τη χρήση των βάσεων αυτών λέγεται *σύστημα διαχείρισης κατανεμημένων σχεσιακών βάσεων δεδομένων (RDBMS)*. Πρόκειται για ένα πρόγραμμα το οποίο ελέγχει την πρόσβαση και τη διατήρηση των αποθηκευμένων δεδομένων στον σχεσιακό μορφότυπο. Επίσης διασφαλίζει τη δημιουργία αντιγράφων ασφαλείας, την ανάκτηση και την ασφάλεια. Μερικές φορές λέγεται *σύστημα διαχείρισης βάσεων δεδομένων πελάτη/διακομιστή*.

3.3.2.2 Αρχιτεκτονικές επιφάνειας διεπαφής – Δεδομένα εισόδου, εξόδου και ενδιάμεσο λογισμικό

Μια άλλη σοβαρή απόφαση που πρέπει να ληφθεί σχετικά με την τεχνολογία αφορά τα δεδομένα εισόδου, εξόδου και τη διασύνδεση του συστήματος. Η απόφαση παλιά ήταν απλή – δεσμίδες δεδομένων εισόδου ή διαδικτυακά δεδομένα εισόδου. Σήμερα πρέπει να λάβουμε υπόψη μας σύγχρονες εναλλακτικές όπως η αυτόματη αναγνώριση, η εισαγωγή δεδομένων με γραφίδα, οι πολλές γραφικές διεπαφές χρήστη, η ηλεκτρονική ανταλλαγή δεδομένων, η δημιουργία εικόνων και η αναγνώριση φωνής μεταξύ άλλων.

3.3.2.3 Αρχιτεκτονική διαδικασιών – Το περιβάλλον ανάπτυξης λογισμικού

Η αρχιτεκτονική διαδικασιών μιας εφαρμογής προσδιορίζεται όσον αφορά τη γλώσσα προγραμματισμού και τα εργαλεία που πρόκειται να χρησιμοποιηθούν για την ανάπτυξη της επιχειρηματικής λογικής και των προγραμμάτων που αφορούν μια διαδικασία. Σε γενικές γραμμές αυτό εκφράζεται σαν ένα μενού επιλογών, γιατί υπάρχουν διάφορα περιβάλλοντα ανάπτυξης λογισμικού που είναι ταιριάζουν σε διάφορες εφαρμογές. Το περιβάλλον ανάπτυξης λογισμικού είναι μια γλώσσα και ένα σύνολο εργαλείων για την κατασκευή εφαρμογών πληροφοριακών συστημάτων. Ένας τρόπος κατάταξης των ΠΑΛ είναι σύμφωνα με την αρχιτεκτονική πελάτη/διακομιστή ή δικτύου που υποστηρίζουν.

3.3.3 Στρατηγικές αρχιτεκτονικής εφαρμογών για το σχεδιασμό των συστημάτων

Οι επιχειρήσεις μπορούν να αναπτύξουν δύο στρατηγικές προκειμένου να έχουν τα αποτελέσματα που θέλουν. Παρακάτω αναλύονται εκτενέστερα οι δύο αυτές αρχιτεκτονικές εφαρμογών.

3.3.3.1 Αρχιτεκτονική επιχειρηματικών εφαρμογών

Σε αυτή τη στρατηγική η επιχείρηση αναπτύσσει μια αρχιτεκτονική για όλα τα συστήματά της η οποία χρησιμοποιείται σε όλα τα μετέπειτα προγράμματα ανάπτυξης πληροφοριακών συστημάτων. Αυτή η αρχιτεκτονική καθορίζει τα παρακάτω:

- Το εγκεκριμένο δίκτυο, τα δεδομένα, τις επιφάνειες διεπαφής, τις τεχνολογίες επεξεργασίας και τα εργαλεία σχεδιασμού (υλικό και λογισμικό, πελάτες και διακομιστές).
- Μια στρατηγική για την εισαγωγή των κληρονομημένων συστημάτων και τεχνολογιών στην αρχιτεκτονική των εφαρμογών.
- Μια συνεχή διαδικασία για τη διαρκή αναθεώρηση της αρχιτεκτονικής ώστε να είναι σύγχρονη και κατάλληλη.
- Μια συνεχή διαδικασία για την αναζήτηση νέων τεχνολογιών και για την διατύπωση προτάσεων ένταξής τους στην αρχιτεκτονική των εφαρμογών.
- Μια διαδικασία για την ανάλυση των αιτήσεων απόκλισης από την εγκεκριμένη αρχιτεκτονική.

3.3.3.2 Αρχιτεκτονική τακτικών εφαρμογών

Όταν δεν υπάρχει μια αρχιτεκτονική για όλη την επιχείρηση κάθε περίπτωση πρέπει να καθορίσει τη δική της αρχιτεκτονική για το πληροφοριακό σύστημα που αναπτύσσεται. Είναι πιθανό να υπάρχει Κάποια ομάδα έρευνας και χρήσης των τεχνολογιών.

Αν και η προτεινόμενη αρχιτεκτονική εφαρμογών των νέων πληροφοριακών συστημάτων είναι πιθανό να επηρεάζεται από τις υπάρχουσες τεχνολογίες, οι προγραμματιστές συνήθως έχουν μεγάλα περιθώρια δράσης στην αναζήτηση νέων τεχνολογιών. Η τελική επιλογή πρέπει να θεωρείται βιώσιμη. Η πληροφοριακή βιωσιμότητα συνήθως περιλαμβάνει τα παρακάτω στοιχεία:

- *Τεχνική βιωσιμότητα* – Αυτό μπορεί να είναι ένα μέτρο της ωριμότητας της τεχνολογίας, της καταλληλότητάς της όσον αφορά την υπό σχεδιασμό εφαρμογή ή την ικανότητα συνεργασίας της με άλλες τεχνολογίες.
- *Λειτουργική βιωσιμότητα* – Πρόκειται για ένα μέτρο που αφορά την εξοικείωση της διοίκησης και των χρηστών με την τεχνολογία, καθώς και την εξοικείωση των τεχνικών και του προσωπικού υποστήριξης με αυτήν.
- *Οικονομική βιωσιμότητα* – Έχει να κάνει με το αν η τεχνολογία είναι εντός του προϋπολογισμού της εταιρείας και αν είναι οικονομική, δηλαδή αν τα οφέλη από αυτή ξεπερνούν το κόστος.

3.3.4 Μοντελοποίηση της αρχιτεκτονικής των εφαρμογών πληροφοριακού συστήματος

Η χρήση διαγραμμάτων ροής λογικών δεδομένων για την μοντελοποίηση των απαιτήσεων των διαδικασιών είναι μια απόλυτα αποδεκτή πρακτική. Ωστόσο, η μετάβαση από τα αναλυτικά διαγράμματα ροής λογικών δεδομένων στα σχεδιαστικά διαγράμματα ροής φυσικών δεδομένων είναι κάτι μυστηριώδες και δυσδιάκριτο. Απαιτείται ένα γενικό σχέδιο υψηλού επιπέδου το οποίο θα λειτουργεί ως η αρχιτεκτονική των εφαρμογών του συστήματος και ένα γενικό σχέδιο των διαδικασιών που αποτελούν το σύστημα

3.3.4.1 Σχεδιασμός διαγραμμάτων ροής φυσικών δεδομένων

Ο τρόπος σχεδιασμού ενός διαγράμματος ροής φυσικών δεδομένων είναι σχεδόν ίδιος με αυτόν που χρησιμοποιείται για τα διαγράμματα ροής λογικών δεδομένων. Ίδιοι είναι και οι κανόνες ορθότητας. Ο σωστός σχεδιασμός έχει τα παρακάτω αποτελέσματα:

- Ένα σύστημα που λειτουργεί.
- Ένα σύστημα που πληροί τις προϋποθέσεις του χρήστη (οι οποίες αναφέρονται στα διαγράμματα ροής λογικών δεδομένων)
- Ένα σύστημα που έχει καλή απόδοση (ρυθμαπόδοση και χρόνος αντίδρασης)

- Ένα σύστημα που περιλαμβάνει επαρκείς εσωτερικούς ελέγχους (για την εξάλειψη των ανθρώπινων και υπολογιστικών λαθών, τη διασφάλιση της αρτιότητας και της ασφάλειας των δεδομένων και την ικανοποίηση των περιορισμών της ελεγκτικής παρακολούθησης).
- Ένα σύστημα που μπορεί να τροποποιείται κάθε φορά που αλλάζουν οι προδιαγραφές ή γίνονται βελτιώσεις.
Θα μπορούσαμε να σχεδιάσουμε ένα διάγραμμα ροής φυσικών δεδομένων για όλο το σύστημα ή μια σειρά διαγραμμάτων για το σύστημα στόχο. Η μέθοδός μας περιλαμβάνει τα παρακάτω:
- Για την αρχιτεκτονική των δικτύων πρέπει να δημιουργείται ένα διάγραμμα ροής φυσικών δεδομένων. Κάθε διαδικασία αυτού του διαγράμματος είναι ένας φυσικός επεξεργαστής (πελάτης ή διακομιστής) του συστήματος. Κάθε διακομιστής εμφανίζεται σαν μεμονωμένος επεξεργαστής. Ωστόσο δεν είναι πρακτικό να δηλώνονται όλοι οι πελάτες. Έτσι κάθε ομάδα πελατών απεικονίζεται από έναν επεξεργαστή. Για κάθε επεξεργαστή του παραπάνω μοντέλου πρέπει να σχεδιαστεί ένα διάγραμμα ροής φυσικών δεδομένων το οποίο να δείχνει τις διαδικασίες που θα ανατεθούν σε αυτόν.
- Ακόμη και οι απλούστερες διαδικασίες πρέπει να εντάσσονται σε σχεδιαστικές μονάδες και να μοντελοποιούνται σαν ένα απλό διάγραμμα ροής φυσικών δεδομένων. Μια σχεδιαστική μονάδα είναι μια αυτοδύναμη ομάδα διαδικασιών, αποθηκών δεδομένων και ροών δεδομένων που έχουν κοινά σχεδιαστικά χαρακτηριστικά. Αποτελεί ένα τμήμα του συνολικού συστήματος τους οποίους τα δεδομένα εισόδου, εξόδους, τα αρχεία, οι βάσεις δεδομένων και τα προγράμματα μπορούν να σχεδιαστούν, να κατασκευαστούν και να δοκιμαστούν σαν ένα μεμονωμένο υποσύστημα.

3.3.4.2 Αρχιτεκτονική δικτύων

Το πρώτο διάγραμμα ροής φυσικών δεδομένων που πρέπει να σχεδιαστεί είναι αυτό της αρχιτεκτονικής δικτύων. Το *DFD της αρχιτεκτονικής δικτύων* είναι

- Ένα σύστημα που περιλαμβάνει επαρκείς εσωτερικούς ελέγχους (για την εξάλειψη των ανθρώπινων και υπολογιστικών λαθών, τη διασφάλιση της αρτιότητας και της ασφάλειας των δεδομένων και την ικανοποίηση των περιορισμών της ελεγκτικής παρακολούθησης).

- Ένα σύστημα που μπορεί να τροποποιείται κάθε φορά που αλλάζουν οι προδιαγραφές ή γίνονται βελτιώσεις.

Θα μπορούσαμε να σχεδιάσουμε ένα διάγραμμα ροής φυσικών δεδομένων για όλο το σύστημα ή μια σειρά διαγραμμάτων για το σύστημα στόχο. Η μέθοδός μας περιλαμβάνει τα παρακάτω:

- Για την αρχιτεκτονική των δικτύων πρέπει να δημιουργείται ένα διάγραμμα ροής φυσικών δεδομένων. Κάθε διαδικασία αυτού του διαγράμματος είναι ένας φυσικός επεξεργαστής (πελάτης ή διακομιστής) του συστήματος. Κάθε διακομιστής εμφανίζεται σαν μεμονωμένος επεξεργαστής. Ωστόσο δεν είναι πρακτικό να δηλώνονται όλοι οι πελάτες. Έτσι κάθε ομάδα πελατών απεικονίζεται από έναν επεξεργαστή. Για κάθε επεξεργαστή του παραπάνω μοντέλου πρέπει να σχεδιαστεί ένα διάγραμμα ροής φυσικών δεδομένων το οποίο να δείχνει τις διαδικασίες που θα ανατεθούν σε αυτόν.
- Ακόμη και οι απλούστερες διαδικασίες πρέπει να εντάσσονται σε σχεδιαστικές μονάδες και να μοντελοποιούνται σαν ένα απλό διάγραμμα ροής φυσικών δεδομένων. Μια σχεδιαστική μονάδα είναι μια αυτοδύναμη ομάδα διαδικασιών, αποθηκών δεδομένων και ροών δεδομένων που έχουν κοινά σχεδιαστικά χαρακτηριστικά. Αποτελεί ένα τμήμα του συνολικού συστήματος τους οποίους τα δεδομένα εισόδου, εξόδου, τα αρχεία, οι βάσεις δεδομένων και τα προγράμματα μπορούν να σχεδιαστούν, να κατασκευαστούν και να δοκιμαστούν σαν ένα μεμονωμένο υποσύστημα.

3.3.4.2 Αρχιτεκτονική δικτύων

Το πρώτο διάγραμμα ροής φυσικών δεδομένων που πρέπει να σχεδιαστεί είναι αυτό της αρχιτεκτονικής δικτύων. Το *DFD της αρχιτεκτονικής δικτύων* είναι

ένα διάγραμμα ροής φυσικών δεδομένων το οποίο προσδιορίζει τους επεξεργαστές (πελάτες και διακομιστές) και τις συσκευές (π.χ. μηχανές και ρομπότ) ενός δικτύου και καθορίζει τη διασύνδεση των πελατών και των διακομιστών καθώς και τα σημεία αλληλεπίδρασης των χρηστών με τους επεξεργαστές (συνήθως μόνο για τους πελάτες).

Για να προσδιορίσει τους επεξεργαστές και τα σημεία τους, ο προγραμματιστής χρησιμοποιεί δύο πηγές:

- Αν υπάρχει μια αρχιτεκτονική επιχειρηματικών εφαρμογών τότε είναι πιθανό να προσδιορίζει εκείνη πώς πρέπει να δημιουργηθούν οι πελάτες/διακομιστές.
- Πρέπει να ζητηθεί η βοήθεια των αρμόδιων διευθυντών ή και ειδικών για να καθοριστεί τι ισχύει, τι είναι πιθανό και τις επιπτώσεις μπορεί να έχει το σύστημα στο δίκτυο των υπολογιστών.

3.3.4.3 Κατανομή δεδομένων και τεχνολογία

Το επόμενο βήμα είναι να κατανεμηθούν οι αποθήκες δεδομένων στους επεξεργαστές του δικτύου. Οι απαιτούμενες αποθήκες δεδομένων μας είναι ήδη γνωστές από την ανάλυση των συστημάτων με τη μορφή των αποθηκών δεδομένων στα διαγράμματα ροής λογικών δεδομένων ή σαν οντότητες στο πλαίσιο των διαγραμμάτων σχέσης οντοτήτων. Το μόνο που πρέπει να γίνει είναι να οριστεί πού θα τοποθετηθεί κάθε αποθήκη και πώς θα υλοποιηθεί.

Για την κατανομή των δεδομένων και τον προσδιορισμό των μεθόδων υλοποίησής τους, οι προγραμματιστές χρησιμοποιούν τρεις πηγές:

- Αν είναι διαθέσιμα τα δίκτυα κατανεμημένων δεδομένων της ανάλυσης του συστήματος υποδεικνύουν τις ανάγκες σε δεδομένα στις επιχειρήσεις χωρίς να εξαρτώνται από κάποια τεχνολογία.
- Αν υπάρχει αρχιτεκτονική επιχειρηματικών εφαρμογών τότε η αρχιτεκτονική αυτή καθορίζει τις βάσεις δεδομένων και την τεχνολογία που πρέπει να χρησιμοποιηθεί.

- Πρέπει να ζητηθεί βοήθεια από διαχειριστές δεδομένων και βάσεων δεδομένων για να προσδιοριστεί τι υπάρχει, τι είναι πιθανό και ποιες μπορεί να είναι οι συνέπειές του σε όλο το σύστημα.

3.3.4.4 Κατανομή διαδικασιών και τεχνολογία

Οι διαδικασίες των πληροφοριακών συστημάτων μπορούν να ανατεθούν στους επεξεργαστές ως εξής:

- Στα διστρωματικά συστήματα πελάτη/διακομιστή όλα τα λογικά γεγονότα αντιστοιχίζονται στον πελάτη.
- Στα τριστρωματικά συστήματα πελάτη/διακομιστή και στα δίκτυα πρέπει να εξεταστεί προσεκτικά το διάγραμμα ροής των αρχέγονων (λεπτομερών) δεδομένων κάθε γεγονότος. Πρέπει να προσδιοριστούν ποιες αρχέγονες διαδικασίες πρέπει να αντιστοιχηθούν στον ελάτη και ποιες στο διακομιστή εφαρμογών. Σε γενικές γραμμές, η καταγραφή και η επεξεργασία των δεδομένων ανατίθεται στους πελάτες ενώ ένα άλλο μέρος της επιχειρηματικής λογικής στους διακομιστές. Σε περίπτωση που έχετε καταφέρει κάποια στοιχεία των διαγραμμάτων ροής λογικών δεδομένων σε διαφορετικούς πελάτες και διακομιστές, πρέπει να σχεδιάσετε διαφορετικά διαγράμματα ροής φυσικών δεδομένων για τα τμήματα κάθε πελάτη και διακομιστή.

3.3.4.5 Ο διαχωρισμός ανθρώπου/μηχανής

Το τελευταίο βήμα του σχεδιασμού των διαδικασιών είναι να προσδιοριστεί οποιοδήποτε τμήμα του διαγράμματος ροής φυσικών δεδομένων που αντιπροσωπεύει χειροκίνητες και όχι μηχανικές διαδικασίες. Αυτό πολλές φορές ονομάζεται «διαχωρισμός ανθρώπου/μηχανής». Η διάκριση αυτή δεν είναι δύσκολη αλλά δεν είναι όσο απλή μπορεί να νομίζετε. οι δυσκολίες εμφανίζονται ότι τα όρια ανθρώπου/μηχανής εντάσσονται σε μια λογική διαδικασία -με άλλα λόγια όταν ένα μέρος της διαδικασίας είναι χειροκίνητο ενώ ένα άλλο είναι μηχανικό. Αυτό παρατηρείται συχνά στα διαγράμματα ροής λογικών δεδομένων γιατί σχεδιάζονται χωρίς να λαμβάνουν υπόψη τους τις εναλλακτικές υλοποίησης.

Τα όρια περιλαμβάνονται σε διάφορες διαδικασίες. Η λύση απαιτεί δύο ενέργειες:

1. Τα τμήματα των χειροκίνητων διαδικασιών εξάγονται με τη μορφή μιας μεμονωμένης σχεδιαστικής μονάδας. Η διεπαφή των χειροκίνητων σχεδιαστικών μονάδων και των μηχανικών διαδικασιών αποτυπώνονται σαν εξωτερικοί παράγοντες. Τέλος οι χειροκίνητες διαδικασίες των σχεδιαστικών μονάδων πρέπει να περιγράφουν με σαφήνεια στα άτομα που θα πρέπει να τις εκτελέσουν.
2. Αν είναι απαραίτητο, οι διαδικασίες του αρχικού διαγράμματος θα πρέπει να μετονομαστούν ώστε να υποδεικνύουν τα μηχανικά τμήματα.

3.4 Σχεδιασμός βάσεων δεδομένων

Η αποθήκευση δεδομένων είναι σημαντικό στοιχείο των περισσότερων πληροφοριακών συστημάτων. Στο κεφάλαιο αυτό θα ασχοληθούμε με τον σχεδιασμό και την κατασκευή των φυσικών βάσεων δεδομένων. Όλα τα πληροφοριακά συστήματα δημιουργούν, διαβάζουν, ενημερώνουν και καταργούν δεδομένα. Αυτά είναι αποθηκευμένα σε αρχεία και βάσεις δεδομένων. Ένα αρχείο είναι μια ομάδα παρόμοιων καταγραφών. Μια βάση δεδομένων είναι μια ομάδα αλληλοσυνδεόμενων αρχείων. Η λέξη κλειδί εδώ είναι το «αλληλοσυνδεόμενος». Μια βάση δεδομένων δεν είναι απλά μια ομάδα αρχείων. Οι καταγραφές κάθε αρχείου πρέπει να έχουν σχέση με τις καταγραφές άλλων αρχείων.

3.4.1 Οι έννοιες των βάσεων δεδομένων για τους αναλυτές συστημάτων

Πολλές από τις έννοιες και τα ζητήματα που έχουν σημασία κατά τον σχεδιασμό των βάσεων δεδομένων διδάσκονται και στα προγράμματα που αφορούν τη διαχείριση των δεδομένων και των βάσεων δεδομένων.

Οι έννοιες και τα ζητήματα που αφορούν τις βάσεις δεδομένων και αποτελούν βασικές αρμοδιότητες των αναλυτών συστημάτων κατά το σχεδιασμό ενός πληροφοριακού συστήματος.

3.4.1.1 Πεδία

Τα πεδία υπάρχουν και στα αρχεία και στις βάσεις δεδομένων. Ένα πεδίο είναι η φυσική εφαρμογή μιας ιδιότητας δεδομένων. Είναι η μικρότερη μονάδα δεδομένων που έχουν κάποιο νόημα και τα οποία αποθηκεύονται σε ένα αρχείο ή σε μια βάση δεδομένων. Υπάρχουν τέσσερις κατηγορίες πεδίων που μπορούν να αποθηκευτούν: τα πρωτεύοντα κλειδιά, τα δευτερεύοντα κλειδιά, τα ξένα κλειδιά και τα περιγραφικά πεδία.

Το πρωτεύον κλειδί είναι ένα πεδίο του οποίου οι τιμές ταυτοποιούν μόνο μία εγγραφή ενός αρχείου. Ένα πρωτεύον κλειδί μπορεί να δημιουργηθεί από το συνδυασμό δύο ή περισσότερων πεδίων (ονομάζεται συνενωμένο κλειδί).

Το δευτερεύον κλειδί είναι ένα εναλλακτικό αναγνωριστικό μιας βάσης δεδομένων. Η τιμή του μπορεί να ταυτοποιεί είτε μία εγγραφή (όπως και το πρωτεύον) ήταν μια υποομάδα όλων των εγγραφών. Ένα αρχείο μιας βάσης δεδομένων μπορεί να έχει μόνο ένα πρωτεύον κλειδί αλλά μπορεί να έχει πολλά δευτερεύοντα κλειδιά. Για να διευκολυνθεί η έρευνα και η ταξινόμηση, συνήθως δημιουργείται ένα ευρετήριο για τα κλειδιά.

Τα ξένα κλειδιά υποδεικνύουν εγγραφές άλλων αρχείων της βάσεις δεδομένων. Αυτά επιτρέπουν στη βάση να διασυνδέσει τις εγγραφές μιας κατηγορίας με αυτές μιας άλλης.

Τα περιγραφικά πεδία είναι όλα τα άλλα πεδία που περιλαμβάνουν δεδομένα της επιχείρησης.

Οι απαιτήσεις της επιχείρησης όσον αφορά τα κλειδιά και τους περιγραφείς προσδιορίζονται κατά τη μοντελοποίηση των δεδομένων στο πλαίσιο της ανάλυσης των συστημάτων.

3.4.1.2 Εγγραφές

Τα πεδία οργανώνονται σε εγγραφές. Αυτές υπάρχουν και στα αρχεία και στις βάσεις δεδομένων. Μια εγγραφή είναι μια ομάδα πεδίων διατεταγμένη σε μια προκαθορισμένη μορφή.

Κατά το σχεδιασμό των συστημάτων οι εγγραφές ταξινομούνται είτε σαν εγγραφές σταθερού μήκους είτε σαν εγγραφές μεταβλητού μήκους. Οι περισσότερες τεχνολογίες βάσεων δεδομένων επιβάλλουν μια δομή εγγραφών σταθερού μήκους, πράγμα που σημαίνει ότι κάθε εγγραφή έχει τα ίδια πεδία, τον ίδιο αριθμό πεδίων και το ίδιο λογικό μέγεθος. Ωστόσο κάποια συστήματα βάσεων δεδομένων συμπίεζουν κάποια μη χρησιμοποιούμενα πεδία και τιμές για να κάνουν οικονομία στο δίσκο. Ο σχεδιαστής της βάσης δεδομένων πρέπει να κατανοεί και να προσδιορίζει το επίπεδο αυτής της συμπίεσης κατά τον σχεδιασμό.

Οι τεχνολογίες των βάσεων δεδομένων συνήθως δεν επιτρέπουν (ή τουλάχιστον δε προωθούν) τις εγγραφές μεταβλητού μήκους.

Όταν ένα πρόγραμμα «διαβάζει» μια εγγραφή από μια βάση δεδομένων συνήθως ανακτά μια σειρά ή μια ομάδα (ή σελίδα) εγγραφών κάθε φορά. Με αυτό τον τρόπο μειώνεται η πρόσβαση στο δίσκο. Ο παράγοντας ομαδοποίησης είναι ο αριθμός των λογικών εγγραφών που περιλαμβάνονται σε μια ενέργεια ανάγνωσης ή καταγραφής (εκ μέρους του υπολογιστή). Η ομάδα αυτή μερικές φορές λέγεται φυσική εγγραφή. Σήμερα ο παράγοντας αυτός συνήθως καθορίζεται και βελτιστοποιείται από την επιλεγμένη τεχνολογία βάσεων δεδομένων. Ωστόσο ένας έμπειρος διαχειριστής βάσεων δεδομένων μπορεί να τον μικροσυντονίσει για λόγους επιδόσεων.

3.4.1.3 Αρχεία και πίνακες

Οι παρόμοιες εγγραφές οργανώνονται σε ομάδες που ονομάζονται αρχεία. Στα συστήματα βάσεων δεδομένων τα αρχεία συχνά ονομάζονται πίνακες. Ένα αρχείο είναι το σύνολο όλων των στοιχείων μιας δομής εγγραφής. Ένας πίνακας είναι το αντίστοιχο του αρχείου στις σχεσιακές βάσεις δεδομένων. Σε λίγο θα αναφερθούμε στην τεχνολογία των σχεσιακών βάσεων δεδομένων. Μερικές κατηγορίες παραδοσιακών αρχείων και πινάκων αναφέρονται παρακάτω:

- Τα κύρια αρχεία ή πίνακες περιλαμβάνουν εγγραφές που είναι σχεδόν μόνιμες. Έτσι όταν μια εγγραφή προστεθεί στα κύρια αρχεία, παραμένει

για πάντα στο σύστημα. Οι τιμές των πεδίων της θα αλλάξουν αλλά οι εγγραφές αυτές θα διατηρηθούν για πάντα.

- Τα *αρχεία* ή πίνακες *συναλλαγών* περιλαμβάνουν εγγραφές που περιγράφουν επιχειρηματικά γεγονότα. Τα δεδομένα που περιγράφουν αυτά τα γεγονότα συνήθως έχουν περιορισμένο χρόνο ζωής. Στα πληροφοριακά συστήματα τα αρχεία αυτά συνήθως διατίθενται στο *διαδίκτυο* για κάποιο χρονικό διάστημα. Μετά το χρήσιμο χρόνο ζωής τους *αρχειοθετούνται* εκτός δικτύου.
- Τα *αρχεία* ή πίνακες *εγγράφων* περιλαμβάνουν αποθηκευμένα αντίγραφα ιστορικών δεδομένων για εύκολη ανάκτηση και αναθεώρηση χωρίς να επιβάλλουν την επανεγγραφή του κειμένου.
- Τα *αρχεία* ή πίνακες *αρχειοθέτησης* περιλαμβάνουν κύρια και αρχεία συναλλαγών που καταργήθηκαν από τη διαδικτυακή αποθήκευση. Έτσι τα αρχεία καταργούνται σπάνια. Απλά μετακινούνται από τη διαδικτυακή στη μη διαδικτυακή αποθήκευση. Οι απαιτήσεις τους υποδεικνύονται από τη νομοθεσία και την ανάγκη συνεχών ελέγχων ή ανάλυσης.
- Τα *αρχεία αναζήτησης σε πίνακα* περιλαμβάνουν σχετικά στατικά δεδομένα τα οποία μπορούν να μοιραστούν οι εφαρμογές για να διατηρηθεί η συνάφεια και να βελτιωθεί η απόδοση.
- Τα *αρχεία επιθεώρησης* είναι ειδικές εγγραφές ενημερώσεων άλλων αρχείων ιδίως των κύριων και των αρχείων συναλλαγών. Χρησιμοποιούνται σε συνδυασμό με αρχεία αρχειοθέτησης για την ανάκτηση «χαμένων» δεδομένων. Τα ίχνη επιθεώρησης συνήθως δημιουργούνται με πολύ καλές τεχνολογίες βάσεων δεδομένων.

3.4.1.4 Βάσεις δεδομένων

Μια βάση δεδομένων μπορεί να θεωρηθεί σαν μια ομάδα συσχετιζόμενων εγγραφών. Με τον όρο συσχετιζόμενος εννοούμε ότι οι εγγραφές του ενός αρχείου μπορεί να έχουν κάποια σχέση ή να συνδέονται με τις εγγραφές κάποιου άλλου αρχείου.

Η βάση δεδομένων αποτελεί την τεχνική υλοποίηση αυτών των οντοτήτων και σχέσεων.

Είναι τόσες πολλές οι εφαρμογές που δημιουργούνται βάσει της τεχνολογίας των βάσεων δεδομένων ώστε ο σχεδιασμός των βάσεων αποτελεί βασικό προσόν για έναν αναλυτή.

Αρχιτεκτονική δεδομένων : Τα δεδομένα μετατρέπονται σε επιχειρηματικό πόρος στο πλαίσιο μιας βάσης δεδομένων. Τα πληροφοριακά συστήματα έχουν δημιουργηθεί γύρω από αυτούς τους πόρους ώστε να παρέχουν στους προγραμματιστές και στους τελικούς χρήστες εύκολη πρόσβαση στα δεδομένα. Η αρχιτεκτονική δεδομένων μιας επιχείρησης προσδιορίζει πώς θα αναπτυχθεί η επιχείρηση, πώς θα χρησιμοποιεί τα αρχεία και τις βάσεις δεδομένων για να αποθηκεύει όλα τα δεδομένα της, την τεχνολογία αρχείων και βάσεων δεδομένων που θα χρησιμοποιηθεί και την διαχειριστική δομή που θα δημιουργηθεί για τη διαχείριση των πόρων.

Οι περισσότερες επιχειρήσεις εξακολουθούν να έχουν πολλές εφαρμογές βασισμένες στα παραδοσιακά αρχεία, πολλές από τις οποίες δημιουργήθηκαν πριν από την εμφάνιση των τεχνολογιών βάσεων δεδομένων μεγάλων επιδόσεων. Σε πολλές περιπτώσεις η αποτελεσματικότητα της επεξεργασίας τέτοιων αρχείων ή το προβλεπόμενο κόστος τους επανασχεδιασμού των αρχείων καθυστέρησε τη μετατροπή των συστημάτων στις βάσεις δεδομένων.

Οι αποθήκες δεδομένων αποθηκεύουν δεδομένα από τις λειτουργικές βάσεις δεδομένων. Τα εργαλεία αναζήτησης και υποστήριξης αποφάσεων χρησιμοποιούνται για τη σύνταξη αναφορών και αναλύσεων αυτών των αποθηκών δεδομένων. Τα εργαλεία αυτά επιτρέπουν στους χρήστες να εξάγουν δεδομένα από τα παραδοσιακά αρχεία και από τις λειτουργικές βάσεις δεδομένων. Αυτό συχνά ονομάζεται εξόρυξη δεδομένων.

Υπάρχουν επίσης οι προσωπικές και ομαδικές (ή τμηματικές) βάσεις δεδομένων. Οι προσωπικοί υπολογιστές και οι τεχνολογία των βάσεων δεδομένων για τοπικά δίκτυα έχει ωριμάσει γρήγορα και επιτρέπει στους χρήστες να δημιουργούν προσωπικές και τμηματικές βάσεις δεδομένων. Αυτές περιλαμβάνουν μοναδικά δεδομένα ή εισάγουν στοιχεία από τα παραδοσιακά

αρχεία, τις λειτουργικές βάσεις δεδομένων ή και τις αποθήκες δεδομένων. Οι προσωπικές βάσεις δεδομένων αναπτύσσονται με τη χρήση προγραμμάτων όπως τα *Access*, *dBase*, *Paradox* και *FoxPro*.

Η σύγχρονη αρχιτεκτονική δεδομένων χρησιμοποιεί και τις τεχνολογίες που βασίζονται στο διαδίκτυο. Για παράδειγμα το *Oracle* και διαθέτει ειδικά εργαλεία και ευκολίες για διαδικτυακές βάσεις δεδομένων.

Για να γίνει διαχείριση των επιχειρησιακών δεδομένων μια ομάδα ειδικών στις βάσεις δεδομένων πρέπει να εργάζεται γύρω από τους παρακάτω διαχειριστές: ένας διαχειριστής δεδομένων ο οποίος είναι υπεύθυνος για τον σχεδιασμό, τον προσδιορισμό, την αρχιτεκτονική και τη διαχείριση των δεδομένων. Ένας ή περισσότεροι διαχειριστές βάσεων δεδομένων (DBA) είναι υπεύθυνοι για την τεχνολογία, το σχεδιασμό, τις συμβουλές κατασκευή, την ασφάλεια, τα αρχεία ασφαλείας, την ανάκτηση και το συντονισμό της απόδοσης των βάσεων δεδομένων. Στις μικρότερες επιχειρήσεις αυτά τα καθήκοντα μπορούν να ανατεθούν σε έναν ή περισσότερους αναλυτές συστημάτων.

Αρχιτεκτονική βάσεων δεδομένων: Έχουμε αναφέρει αρκετές τεχνολογίες βάσεων δεδομένων που καθιστούν εφικτή την αρχιτεκτονική των βάσεων δεδομένων. Η αρχιτεκτονική βάσεων δεδομένων παραπέμπει στην τεχνολογία βάσεων δεδομένων η οποία περιλαμβάνει τη μηχανή, τα βοηθητικά προγράμματα, τα εργαλεία CASE για την ανάλυση και το σχεδιασμό και τα εργαλεία ανάπτυξης εφαρμογών βάσεων δεδομένων. Το κέντρο ελέγχου αυτής της αρχιτεκτονικής είναι το σύστημα διαχείρισης της βάσης δεδομένων.

Το σύστημα διαχείρισης βάσεων δεδομένων (DBMS) είναι ένα εξειδικευμένο λογισμικό που διατίθεται σε καταστήματα υπολογιστών και χρησιμοποιείται για τη δημιουργία, πρόσβαση, έλεγχο και διαχείριση μιας βάσης δεδομένων. ο πυρήνας του ονομάζεται συνήθως μηχανή βάσης δεδομένων. η μηχανή αυτή ανταποκρίνεται σε συγκεκριμένες εντολές για τη δημιουργία δομών βάσεων δεδομένων καθώς και για τη δημιουργία, ανάγνωση, ενημέρωση και κατάργηση εγγραφών στη βάση. Το σύστημα μπορεί να αγοραστεί από μια εταιρεία που διαθέτει τεχνολογία βάσεων δεδομένων όπως οι *Oracle*, *IBM*, *Microsoft* ή *Sybase*.

Συστήματα διαχείρισης σχεσιακών βάσεων δεδομένων Υπάρχουν πολλές κατηγορίες συστημάτων διαχείρισης βάσεων δεδομένων. Κατατάσσονται ανάλογα με τον τρόπο που ταξινομούν τις εγγραφές τους. Τα πρώτα τέτοια συστήματα οργάνωναν τις εγγραφές τους ιεραρχικά ή σε δίκτυα με ευρετήρια και συνδεδεμένους καταλόγους. Σήμερα τα πιο επιτυχημένα συστήματα διαχείρισης βάσεων δεδομένων βασίζονται στη σχεσιακή τεχνολογία. Στις σχεσιακές βάσεις δεδομένων οι εγγραφές οργανώνονται με τη μορφή πινάκων δύο διαστάσεων που συνδέονται μέσω ξένων κλειδιών. Κάθε πίνακας (μερικές φορές ονομάζεται *σχέση*) αποτελείται από συγκεκριμένες στήλες (που αποτελούν πεδία ή ιδιότητες) και από σειρές χωρίς όνομα (που αντιστοιχούν στις εγγραφές).

Στις σχεσιακές βάσεις δεδομένων τα αρχεία έχουν τη μορφή απλών πινάκων δύο διαστάσεων που είναι γνωστοί ως *σχέσεις*. Οι σειρές είναι οι εγγραφές. Οι στήλες αντιστοιχούν στα πεδία.

Για την πρόσβαση στα δεδομένα που βρίσκονται στους πίνακες και στις εγγραφές η SQL διαθέτει τις παρακάτω εντολές:

- **SELECT**: Επιλογή συγκεκριμένων εγγραφών από έναν πίνακα βάσει συγκεκριμένων κριτηρίων.
- **PROJECT**: Προβολή συγκεκριμένων εγγραφών ενός πίνακα.
- **JOIN**: Συγχώνευση δύο ή περισσότερων πινάκων όσον αφορά ένα κοινό πεδίο -ένα πρωτεύον και ένα ξένο κλειδί.

Όταν χρησιμοποιούνται σε συνδυασμό αυτές οι βασικές εντολές μπορούν να εκτελέσουν τις περισσότερες λειτουργίες μιας βάσης δεδομένων. Ένα βασικό χαρακτηριστικό της SQL είναι ότι οι εντολές εμφανίζουν έναν αριθμό εγγραφών και όχι απαραίτητα μία εγγραφή (όπως στις μη σχεσιακές βάσεις δεδομένων και στις τεχνολογίες που βασίζονται στα αρχεία). Οι βάσεις δεδομένων τύπου SQL διαθέτουν επίσης εντολές για τη δημιουργία, την ενημέρωση, την κατάργηση καθώς και την ταξινόμηση των εγγραφών.

Οι σχεσιακές βάσεις δεδομένων υψηλού επιπέδου διευρύνουν την SQL ώστε να υποστηρίζει διεγέρτες και αποθηκευμένες διαδικασίες. Οι διεγέρτες είναι προγράμματα ενσωματωμένα σε έναν πίνακα τα οποία εκκινούνται αυτόματα όταν παρατηρούνται ενημερώσεις σε άλλο πίνακα. Οι αποθηκευμένες

διαδικασίες είναι προγράμματα ενσωματωμένα σε έναν πίνακα τα οποία αποτελούν μέρος μιας άλλης εφαρμογής. Για παράδειγμα ένας αλγόριθμος επικύρωσης δεδομένων μπορεί να ενσωματωθεί σε έναν πίνακα για να διασφαλίσει ότι οι νέες και ενημερωμένες εγγραφές περιέχουν έγκυρα δεδομένα πριν αποθηκευτούν. Οι αποθηκευμένες διαδικασίες συντάσσονται σε μια αποκλειστική επέκταση της SQL, όπως είναι τα: *Transact SQL* της Microsoft ή *PL/SQL* της Oracle.

Οι διεγέρτες και οι αποθηκευμένες διαδικασίες είναι επαναχρησιμοποιήσιμοι για αποθηκεύονται στους ίδιους τους πίνακες (σαν μεταδεδομένα). Αυτό γλιτώνει τους προγραμματιστές από τη δημιουργία της αντίστοιχης λογικής σε κάθε εφαρμογή που χρησιμοποιεί τους πίνακες.

3.4.2 Προϋποθέσεις για τον σχεδιασμό των βάσεων δεδομένων – Κανονικοποίηση

Προηγουμένως αναφέρθηκε πώς μοντελοποιούνται οι προϋποθέσεις των δεδομένων για ένα πληροφοριακό σύστημα. Το μοντέλο είχε τη μορφή ενός πλήρους διαγράμματος σχέσεων οντοτήτων και μιας αποθήκης μεταδεδομένων. Η τεχνική αυτή χρησιμοποιήθηκε για να δημιουργηθεί ένα μοντέλο που συμμορφώνεται με τα παρακάτω κριτήρια ποιότητας:

- Ένα καλό μοντέλο δεδομένων είναι απλό. Γενικά τα δεδομένα που περιγράφουν μια οντότητα πρέπει να περιγράφουν μόνο αυτή την οντότητα.
- Ένα καλό μοντέλο δεδομένων δεν έχει πλεονάζοντα στοιχεία. Αυτό σημαίνει ότι κάθε ιδιότητα δεδομένου, εκτός από τα ξένα κλειδιά, περιγράφει όσο το δυνατό περισσότερο μια οντότητα.
- Ένα καλό μοντέλο δεδομένων πρέπει να είναι ευέλικτο και ευπροσάρμοστο στις μελλοντικές ανάγκες. Όταν δεν πληρείται αυτό το κριτήριο τότε δημιουργούμε βάσεις δεδομένων που ικανοποιούν μόνο τις παροντικές επιχειρησιακές απαιτήσεις.

Πώς επιτυγχάνονται λοιπόν οι παραπάνω στόχοι; Πώς μπορείτε να σχεδιάσετε μια βάση δεδομένων που να προσαρμόζεται σύμφωνα με τις

μελλοντικές ανάγκες τις οποίες δεν μπορείτε να προβλέψετε; Η απάντηση βρίσκεται στην ανάλυση των δεδομένων.

Θυμηθείτε ότι η κανονικοποίηση είναι μια τεχνική τριών σταδίων που περνά το μοντέλο δεδομένο από την πρώτη, τη δεύτερη και την τρίτη κανονική μορφή. Ο σχεδιασμός μιας βάσης δεδομένων μπορεί να προχωρήσει μόνο αν το υποκείμενο μοντέλο δεδομένων είτε τουλάχιστον στην τρίτη κανονική μορφή.

3.4.3 Συμβατικός σχεδιασμός αρχείων

Εδώ θα εστιάσουμε στο σχεδιασμό βάσεων δεδομένων. Ωστόσο δεν πρέπει να παραλείψουμε να πούμε δυο λόγια για τον συμβατικό σχεδιασμό αρχείων. Καταρχήν ο σχεδιασμός τους είναι απλοποιημένος λόγω του ότι απευθύνονται σε μία εφαρμογή. Συνήθως πρώτα δημιουργούνται τα σχέδια εξόδου και εισόδου γιατί ο σχεδιασμός των αρχείων βασίζεται στη στήριξη των απαιτήσεων της εφαρμογής.

Οι θεμελιώδεις οντότητες του μοντέλου δεδομένων θα σχεδιαστούν ως κύριες ή εγγραφές συναλλαγής. Τα κύρια αρχεία είναι εγγραφές σταθερού μήκους. Οι σχετικές οντότητες του μοντέλου δεδομένων συγχωνεύονται συνήθως με τις εγγραφές συναλλαγών και σχηματίζουν εγγραφές μεταβλητού μήκους (βασισμένες σε μία ή πολλές σχέσεις). Τα άλλα είδη αρχείων (που δεν αναπαρίστανται στο μοντέλο δεδομένων) προστίθενται εφόσον είναι αναγκαία.

Δύο απαραίτητοι παράγοντες του συμβατικού σχεδιασμού αρχείων είναι η πρόσβαση και η οργάνωση του αρχείου. Οι αναλυτές συστημάτων συνήθως μελετούν τον τρόπο με τον οποίο θα έχει πρόσβαση κάθε πρόγραμμα στις εγγραφές ενός αρχείου (διαδοχικά ή τυχαία) και στη συνέχεια επιλέγουν την κατάλληλη οργάνωση των αρχείων (π.χ. διαδοχική, με δείκτες, κατατεμαχισμένη, κλπ). Στην πράξη πολλοί αναλυτές συστημάτων επιλέγουν μια σειριακή οργάνωση με δείκτες (ISAM/VSAM) για να υποστηρίξουν την πιθανότητα να υπάρχουν διαφορετικοί τρόποι πρόσβασης στις εγγραφές.

3.4.4 Σύγχρονος σχεδιασμός βάσεων δεδομένων

Ο σχεδιασμός μιας βάσης δεδομένων απαιτεί τη συμμετοχή του DBA και του προσωπικού βάσεων δεδομένων. Αυτοί θα ασχοληθούν με τις τεχνικές λεπτομέρειες και τα ζητήματα πολλαπλών εφαρμογών. Ωστόσο καλό είναι να καταλαβαίνει ο αναλυτής συστήματος τις βασικές αρχές σχεδιασμού των σχεσιακών βάσεων δεδομένων.

Οι κανόνες που παρουσιάζονται εδώ είναι στην ουσία οδηγίες. Δεν μπορούμε να καλύψουμε όλες τις περιπτώσεις. Κάθε DBMS έχει δικές τους ιδιότητες και περιορισμούς. Ευτυχώς οι οδηγίες που παρουσιάζονται είναι γενικές και εφαρμόζονται στα περισσότερα περιβάλλοντα.

Τα Προγράμματα υποβοήθησης για την ανάπτυξη σύνθετων εφαρμογών (CASE) έχουν αναφερθεί αρκετές φορές. Υπάρχουν συγκεκριμένα προϊόντα CASE που αφορούν την ανάλυση και τον σχεδιασμό των βάσεων δεδομένων. Τα περισσότερα εργαλεία CASE μπορούν να αναπαράγουν αυτόματα έναν κωδικό SQL για να δημιουργήσουν δομές βάσεων δεδομένων για τα διασημότερα συστήματα διαχείρισης βάσεων δεδομένων. Αυτή η δυνατότητα δημιουργίας κώδικα εξοικονομεί πολύ χρόνο.

3.4.4.1 Στόχοι και προϋποθέσεις του σχεδιασμού βάσεων δεδομένων

Οι στόχοι του σχεδιασμού βάσεων δεδομένων είναι οι παρακάτω:

- Μια βάση δεδομένων πρέπει να διαθέτει επαρκή αποθήκευση, ενημέρωση και ανάκτηση δεδομένων.
- Μια βάση δεδομένων πρέπει να είναι αξιόπιστη –τα αποθηκευμένα δεδομένα πρέπει να έχουν αριότητα και να προάγουν την εμπιστοσύνη του χρήστη σε αυτά.
- Μια βάση δεδομένων πρέπει να είναι ευπροσάρμοστη και να μεταβάλλεται σύμφωνα με τις νέες και απρόβλεπτες απαιτήσεις και εφαρμογές.

3.4.4.2 Το σχήμα της βάσης δεδομένων

Ο σχεδιασμός της βάσης δεδομένων αποτυπώνεται σαν ένα ιδιαίτερο μοντέλο που ονομάζεται *σχήμα βάσης δεδομένων*. Αυτό είναι το φυσικό μοντέλο ή σχέδιο μιας βάσης δεδομένων. Παρουσιάζει την τεχνική υλοποίηση μιας βάσης δεδομένων.

Το σχήμα των σχεσιακών βάσεων δεδομένων προσδιορίζει τη δομή της βάσης δεδομένων ως προς τους πίνακες, τους δείκτες και τους κανόνες αρτιότητας. Το σχήμα αυτό περιλαμβάνει λεπτομέρειες που αφορούν τις δυνατότητες, την ορολογία και τους περιορισμούς του συγκεκριμένου συστήματος διαχείρισης βάσεων δεδομένων. Κάθε DBMS υποστηρίζει διαφορετικά είδη δεδομένων, κανόνες αρτιότητας κλπ.

Η μεταβολή του μοντέλου λογικών δεδομένων σε μια φυσικό σχήμα σχεσιακής βάσης δεδομένων διέπεται από κάποιους γενικούς κανόνες και προτιμήσεις. Αυτές οι οδηγίες και οι κανόνες συνοψίζονται στα παρακάτω:

1. Κάθε θεμελιώδης, συσχετιστική και ασθενής οντότητα αποτελεί έναν ξεχωριστό πίνακα. Τα ονόματα των πινάκων πρέπει να δημιουργηθούν σύμφωνα με τους κανόνες ονοματολογίας και τους περιορισμούς μεγέθους του DBMS.
 - a) Το πρωτεύον κλειδί προσδιορίζεται και παρουσιάζεται ως δείκτης μέσα στον πίνακα.
 - b) Κάθε δευτερεύον κλειδί παρουσιάζεται με δικό του δείκτη μέσα στον πίνακα.
 - c) Ένας δείκτης πρέπει να τοποθετηθεί και για τα στοιχεία που δεν έχουν κλειδί και τα οποία θεωρούνται δευτερεύουσες προϋποθέσεις.
 - d) Κάθε ξένο κλειδί πρέπει να παρουσιάζεται ως τέτοιο. Η ύπαρξη τέτοιων κλειδιών συμβάλει στην υλοποίηση των σχέσεων που υπήρχαν στο μοντέλο δεδομένων και επιτρέπει στους πίνακες να συγχωνευτούν σε SQL και εφαρμογές.

e) Τα στοιχεία μπορούν να έχουν πεδία. Τα πεδία αυτά αντιστοιχούν σε στήλες του πίνακα. Για κάθε στοιχεία πρέπει να προσδιορίζονται οι παρακάτω τεχνικές λεπτομέρειες.

Τα ονόματα των πεδίων πρέπει να συντηθούν και να μετατραπούν σύμφωνα με τους περιορισμούς και τους εσωτερικούς κανόνες του DBMS.

- i. *Κατηγορίες δεδομένων.* Κάθε DBMS υποστηρίζει διαφορετικούς τύπους και όρους δεδομένων.
- ii. *Μέγεθος πεδίων.* Τα διαφορετικά DBMS εκφράζουν διαφορετικά την ακρίβεια των πραγματικών αριθμών.
- iii. *Κενό ή χωρίς κενό.* Πρέπει το πεδίο να έχει κάποια τιμή πριν αποθηκευτεί; Και πάλι κάθε DBMS μπορεί να απαιτεί διαφορετικές λέξεις για να εκφράσει αυτή την ιδιότητα. Από τον ορισμό τους τα πρωτεύοντα κλειδιά δεν επιτρέπουν ΚΕΝΕΣ τιμές.
- iv. *Πεδίο ορισμού.* Πολλά συστήματα διαχείρισης βάσεων δεδομένων επεξεργάζονται αυτόματα τα δεδομένα για να διασφαλίσουν ότι είναι έγκυρα. Αυτό μπορεί να συμβάλλει στην αρτιότητα των δεδομένων ασχέτως της εφαρμογής. Ωστόσο για τα DBMS που υποστηρίζουν την αρτιότητα των δεδομένων οι κανόνες πρέπει να διατυπωθούν σαφώς στη γλώσσα που κατανοεί το DBMS.
- v. *Προεπιλογή.* Πολλά συστήματα διαχείρισης βάσεων δεδομένων επιτρέπουν να δίνεται μια προεπιλεγμένη τιμή σε περίπτωση που ο χρήστης ή ο προγραμματιστής δημιουργήσει εγγραφές με κενά πεδία. Σε πολλές περιπτώσεις προεπιλογή αποτελεί το ΚΕΝΟ.
- vi. Άλλες φορές πολλά από τα παραπάνω στοιχεία τεκμηριώνονται ως τμήμα του πλήρους μοντέλου λογικών δεδομένων. Αν το μοντέλο αυτό έχει δημιουργηθεί με εργαλεία CASE τότε αυτά θα μπορούν να μετατρέψουν αυτόματα το μοντέλο δεδομένων στη φυσική γλώσσα της συγκεκριμένης τεχνολογίας βάσεων δεδομένων.

2. Οι υπέρτυπες/ υπότυπες οντότητες έχουν επιπλέον επιλογές:
- a) Κάθε υπέρτυπο και υπότυπο πρέπει να τοποθετείται σε διαφορετικό πίνακα (γιατί όλα έχουν το ίδιο πρωτεύον κλειδί).
 - b) Διαφορετικά αν τα υπότυπα έχουν *παρόμοιο μέγεθος* και περιεχόμενο, ο διαχειριστής βάσεων δεδομένων ίσως πρέπει να τα μετατρέψει σε υπέρτυπα για να δημιουργήσει έναν μόνο πίνακα. Αυτό είναι πιθανό να δημιουργήσει προβλήματα κατά τον προσδιορισμό των προεπιλογών και τον έλεγχο των πεδίων ορισμού. Στα υψηλής ποιότητας DBMS αυτά τα προβλήματα αντιμετωπίζονται με τη συγχώνευση της λογικής των προεπιλογών και του πεδίου ορισμού στις αποθηκευμένες *διαδικασίες* του πίνακα.
 - c) Τα στοιχεία των υπέρτυπων μπορούν να επαναληφθούν σε έναν πίνακα για κάθε υπότυπο.
 - d) Μπορεί να χρησιμοποιηθεί συνδυασμός των παραπάνω επιλογών.
3. Αξιολόγηση και προσδιορισμός των αντίστοιχων περιορισμών αρτιότητας.

Στο σχεδιασμό της βάσης δεδομένων μπορούν να προστεθούν επιπλέον δείκτες για μοναδικά δευτερεύοντα κλειδιά ή για οποιοδήποτε στοιχείο χωρίς κλειδί που μπορεί να χρησιμοποιηθεί για να δημιουργήσει μια υποομάδα όλων των εγγραφών ενός πίνακα. Οι δείκτες αυτοί μπορεί να βελτιώσουν την απόδοση της τελικής βάσης δεδομένων.

Ορισμένα εργαλεία CASE δημιουργούν σχήματα βάσεων δεδομένων με πολλές λεπτομέρειες. Για παράδειγμα σε κάποια σχήματα τα πεδία υποδεικνύουν για κάθε πεδίο αν πρέπει να πάρει κάποια τιμή ή όχι:

- KENO σημαίνει ότι το πεδίο δεν χρειάζεται να πάρει κάποια τιμή.
- ΟΧΙ KENO σημαίνει ότι το πεδίο πρέπει να πάρει κάποια τιμή. Επειδή τα πρωτεύοντα κλειδιά χρησιμοποιούνται για να δίνουν μοναδική πρόσβαση στα δεδομένα τα πεδία τους δεν μπορεί να έχουν μηδενική τιμή.

Θα θυσιάζατε ποτέ τις οντότητες τρίτης κανονικής μορφής σε μια βάση δεδομένων; Για παράδειγμα θα θέλατε ποτέ να συγχωνέψετε δύο οντότητες

τρύτης μορφής σε έναν πίνακα (αφαιρώντας τους έτσι την ιδιότητα της τρίτης μορφής); Συνήθως όχι! Αν και ένας διαχειριστής βάσης δεδομένων μπορεί να κάνει κάτι τέτοιο για να βελτιώσει την απόδοση της βάσης δεδομένων πρέπει να ζυγίσει προσεκτικά τα πλεονεκτήματα και τα μειονεκτήματα. Αν και κάτι τέτοιο μπορεί να προσφέρει λιγότερους πίνακες ή καλύτερη γενική απόδοση, τέτοιου είδους συνδυασμοί μπορεί να οδηγήσουν σε πιθανή απώλεια της ανεξαρτησίας των δεδομένων –αν κάποια μελλοντικά νέα πεδία απαιτούν τον διαχωρισμό ενός πίνακα σε δύο, τα προγράμματα θα πρέπει να γραφτούν από την αρχή. Σε γενικές γραμμές δεν συνίσταται η συγχώνευση οντοτήτων στους πίνακες.

3.4.4.3 Ακεραιότητα δεδομένων και σχέσεων

Η ακεραιότητα μιας βάσης δεδομένων είναι θέμα εμπιστοσύνης. Μπορούν να εμπιστευτούν η επιχείρηση και οι χρήστες τα δεδομένα που υπάρχουν στη βάση; Η ακεραιότητα των δεδομένων προϋποθέτει την ύπαρξη ελεγκτικών μηχανισμών. Σε κάθε βάση δεδομένων υπάρχουν τουλάχιστον τρεις κατηγορίες ακεραιότητας που μπορούν να σχεδιαστούν.

Ακεραιότητα κλειδιού : Κάθε πίνακας πρέπει να διαθέτει ένα πρωτεύον κλειδί (το οποίο να μπορεί να συνενωθει). Το πρωτεύον κλειδί πρέπει να ελέγχεται έτσι ώστε να μην αντιστοιχεί σε δύο εγγραφές ενός πίνακα. (Σημειώστε ότι για κάθε συνενωμένο κλειδί μοναδική πρέπει να είναι η συνενωμένη τιμή όχι οι τιμές που αποτελούν τη συνένωση).

Επίσης το πρωτεύον κλειδί μιας εγγραφής δεν πρέπει να έχει μηδενική τιμή. Με αυτό τον τρόπο θα εξαλειφθεί ο ρόλος του, που είναι ο μοναδικός προσδιορισμός των εγγραφών.

Αν το σύστημα διαχείρισης βάσεων δεδομένων δεν επιβάλλει αυτούς του κανόνες, τότε πρέπει να ληφθούν άλλα μέτρα για τη διασφάλισή τους. Ωστόσο τα περισσότερα DBMS επιβάλλουν την ακεραιότητα των κλειδιών.

Ακεραιότητα πεδίου ορισμού: Πρέπει να σχεδιαστούν κατάλληλοι μηχανισμοί ελέγχου ώστε να διασφαλίζεται ότι δεν υπάρχει πεδίο που να λαμβάνει τιμές εκτός από τις επιτρεπόμενες.

Λίγο καιρό πριν οι εφαρμογές αναλάμβαναν και την επεξεργασία των δεδομένων. σήμερα τα περισσότερα συστήματα διαχείρισης βάσεων δεδομένων μπορούν να επιβάλουν κανόνες σχετικά με τα πεδία ορισμού. Στο άμεσο μέλλον η ευθύνη της επεξεργασίας των δεδομένων θα εξακολουθήσει να μοιράζεται μεταξύ των εφαρμογών και των DBMS.

Ακεραιότητα σχέσεων: Η αρχιτεκτονική των βάσεων δεδομένων δημιουργεί σχέσεις μεταξύ των εγγραφών των πινάκων μέσω των ξένων κλειδιών. Η χρήση των ξένων κλειδιών αυξάνει την ευελιξία και τη δυνατότητα τροποποίηση των βάσεων δεδομένων, αλλά αυξάνει τον κίνδυνο εμφάνισης σχεσιακών προβλημάτων. Ένα σφάλμα στην ακεραιότητα των σχέσεων: Παρατηρείται όταν η τιμή ενός ξένου κλειδιού ενός πίνακα δεν ταιριάζει με την τιμή του πρωτεύοντος κλειδιού του άλλου πίνακα.

Η ακεραιότητα των σχέσεων προσδιορίζεται με τη μορφή κανόνων κατάργησης ως εξής:

- *Κανένας περιορισμός* – οι εγγραφές ενός πίνακα μπορούν να καταργηθούν χωρίς να συνυπολογιστούν οι εγγραφές άλλων πινάκων.
- *Κατάργηση: Στοίβαξη* – η κατάργηση μιας εγγραφής ενός πίνακα πρέπει να ακολουθηθεί αυτόματα από την κατάργηση των εγγραφών που συνδέονται με αυτή σε κάποιους άλλους πίνακες. Πολλά συστήματα διαχείρισης βάσεων δεδομένων εκτελούν αυτόματα αυτή την εντολή με τη χρήση διεγερτών.
- *Κατάργηση: περιορισμός* – η κατάργηση μιας εγγραφής ενός πίνακα πρέπει να απορρίπτεται με να καταργηθούν οι σχετικές τις εγγραφές στους άλλους πίνακες. Και πάλι υπάρχουν πολλά συστήματα διαχείρισης βάσεων δεδομένων εκτελούν αυτόματα αυτή την εντολή.
- *Κατάργηση: τοποθέτηση κενού* – η κατάργηση της εγγραφής ενός πίνακα πρέπει να ακολουθείται αυτόματα από την δημιουργία κλειδιών στον σχετικό πίνακα που να έχουν μηδενική τιμή. Πολλά συστήματα διαχείρισης βάσεων δεδομένων εκτελούν αυτόματα αυτή την εντολή με τη χρήση διεγερτών.

3.4.4.4 Ρόλοι

Κάποια πρότυπα βάσεων δεδομένων επιμένουν ότι δεν είναι δυνατόν δύο πεδία να έχουν ακριβώς το ίδιο όνομα. Αυτός ο περιορισμός απλοποιεί την τεκμηρίωση, τα βοηθητικά συστήματα και τους ορισμούς των μεταδεδομένων. Αυτό δημιουργεί πρόβλημα στα ξένα κλειδιά. Εξ ορισμού ένα ξένο κλειδί πρέπει να έχει ένα αντίστοιχο πρωτεύον κλειδί. Κατά τη μοντελοποίηση των λογικών δεδομένων η χρήση ίδιων ονομάτων μας βοήθησε τους χρήστες να καταλάβουν ότι τα ξένα κλειδιά μας επιτρέπουν να αντιστοιχίζουμε τις σχετικές εγγραφές με διαφορετικές οντότητες. Ωστόσο σε μια φυσική βάση δεδομένων δεν είναι πάντα απαραίτητο ή επιθυμητό να υπάρχουν αυτά τα πλεονάζοντα ονόματα πεδίων στη βάση

Για να αντιμετωπιστεί αυτό το πρόβλημα τα ξένα κλειδιά μπορούν να λάβουν ονόματα ρόλων. Ένα όνομα ρόλου είναι το εναλλακτικό όνομα του ξένου κλειδιού που παρουσιάζει εμφανώς την υπηρεσία που επιτελεί το κλειδί στον πίνακα.

Η απόφαση για τη χρήση ή όχι των ονομάτων ρόλων λαμβάνεται από τον διαχειριστή δεδομένων και βάσεων δεδομένων.

3.4.4.5 Κατανομή και αναπαραγωγή βάσεων δεδομένων

Η *ανάλυση της κατανομής των δεδομένων* προσδιορίζει ποιες θέσεις εργασίας χρειάζονται πρόσβαση σε συγκεκριμένες οντότητες και ιδιότητες δεδομένων.

Στον σημερινό πολυστρωματικό, δικτυοκεντρικό κόσμο πελάτη/διακομιστή τα πληροφοριακά συστήματα και οι βάσεις δεδομένων είναι σπάνια συγκεντρωμένα. Αντίθετα κατανέμονται σε ένα δίκτυο που μπορεί να καλύπτει πολλά κτίρια, πόλεις, κράτη ή χώρες. Γι' αυτό ίσως χρειαστεί να τεμαχίσουμε, να κατανείμουμε ή να αναπαράγουμε ολόκληρο ή ένα μέρος του σχεδιασμού των βάσεων δεδομένων για τους διαφορετικούς διακομιστές φυσικών βάσεων δεδομένων που βρίσκονται σε διαφορετικές τοποθεσίες. Βασικά πρέπει να πραγματοποιήσουμε μια ανάλυση κατανομής των φυσικών βάσεων δεδομένων η

οποία να συνυπολογίζει όσα αναφέρθηκαν κατά την ανάλυση κατανομής λογικών βάσεων δεδομένων.

Έχουμε στη διάθεσή μας μια σειρά επιλογών:

- Συγκέντρωση της βάσης δεδομένων. Με άλλα λόγια θα αναπτύξουμε τη βάση δεδομένων σε έναν διακομιστή άσχετα με τον αριθμό των φυσικών τόπων που θα έχουν πρόσβαση σε αυτή. Αυτή η λύση είναι απλή και διατηρείται εύκολα. Παραβιάζει ένα κανόνα της διαχείρισης των δεδομένων που θεωρείται πολύ σημαντικός από διαχειριστές και χρήστες δεδομένων- τα δεδομένα πρέπει να βρίσκονται όσο το δυνατό εγγύτερα στους χρήστες.
- Οριζόντια κατανομή των δεδομένων. Σε αυτή την περίπτωση κάθε πίνακας (ή ολόκληρες γραμμές ενός πίνακα) θα τοποθετούνται σε διαφορετικούς διακομιστές και σημεία. Με αυτό τον τρόπο διασφαλίζεται εύκολη πρόσβαση και ασφάλεια γιατί κάθε περιοχή διαθέτει μόνο τους πίνακες και τις γραμμές που χρειάζεται. Δυστυχώς τα δεδομένα δεν συνδυάζονται εύκολα για να διεξαχθεί διαχειριστική ανάλυση στα διάφορα σημεία.
- Κάθετη κατανομή των δεδομένων. Σε αυτή την περίπτωση συγκεκριμένες στήλες του πίνακα τοποθετούνται σε συγκεκριμένες βάσεις δεδομένων και διακομιστές. Τα πλεονεκτήματα και τα μειονεκτήματα αυτής της εναλλακτικής μοιάζουν με αυτά της οριζόντιας κατανομής.
- Αναπαραγωγή δεδομένων. Η αναπαραγωγή αφορά τη φυσική επανάληψη ολόκληρων πινάκων σε διάφορες τοποθεσίες. Τα περισσότερα επιχειρηματικά συστήματα διαχείρισης βάσεων δεδομένων περιλαμβάνουν τεχνολογίες αναπαραγωγής που συντονίζουν την ενημέρωση των επαναλαμβανόμενων πινάκων και εγγραφών για να διασφαλίζεται η ακεραιότητα των δεδομένων. αυτή η λύση προσφέρει επιδόσεις και διευκολύνει την πρόσβαση, ενώ μειώνει την κίνηση του δικτύου. Ωστόσο η ακεραιότητα των δεδομένων γίνεται πιο πολύπλοκη και απαιτεί περισσότερες αποθηκευτικές ικανότητες.

Οι εναλλακτικές αυτές δεν αποκλείουν η μία την άλλη. Ο προγραμματιστής πρέπει να σχεδιάσει με προσοχή τον βαθμό κατανομής και αναπαραγωγής των δεδομένων.

Σε σχήμα φυσικών βάσεων δεδομένων μπορούμε να προσδιορίσουμε εικόνες που αντιστοιχούν σε συγκεκριμένες γεωγραφικές περιοχές (ή υποεικόνες για τους διαφορετικούς χρήστες και εφαρμογές). Η εικόνα μιας βάσεις δεδομένων μπορεί να είναι πολύ επιλεκτική. Μπορεί να περιλαμβάνει μια συγκεκριμένη υποομάδα πινάκων, μια συγκεκριμένη υποομάδα στηλών πινάκων ή ακόμα και μια συγκεκριμένη υποομάδα εγγραφών πινάκων. Κάθε εικόνα πρέπει να ταιριάζει απόλυτα με το κύριο σχήμα βάσεων δεδομένων έτσι ώστε οι αλλαγές σε αυτό να μπορούν, αν είναι απαραίτητο, να επεκταθούν και στις εικόνες. Τα εργαλεία CASE είναι πολύ χρήσιμα στον προσδιορισμό των εικόνων και στον συγχρονισμό τους.

3.4.4.6 Πρωτότυπα βάσεων δεδομένων

Η κατασκευή πρωτοτύπων δεν αποτελεί εναλλακτική για τον προσεκτικό σχεδιασμό των σχημάτων βάσεων δεδομένων. Αντίθετα, μετά την ολοκλήρωση του σχήματος μια πρωτότυπη βάση δεδομένων μπορεί να δημιουργηθεί πολύ γρήγορα. Τα σύγχρονα DBMS περιλαμβάνουν δυνατούς γεννήτορες βάσεων δεδομένων οδηγούμενους από μενού οι οποίοι δημιουργούν αμέσως μια DDL από την οποία παράγουν τη βάση δεδομένων. στη συνέχεια η βάση αυτή μπορεί να γεμίσει με δοκιμαστικά δεδομένα τα οποία θα αποδειχτούν χρήσιμα για την κατασκευή πρωτοτύπων και για τη δοκιμή των δεδομένων εξόδου και εισόδου, των οθόνων και άλλων στοιχείων του συστήματος.

3.4.4.7 Προγραμματισμός χωρητικότητας βάσεων δεδομένων

Μια βάση δεδομένων αποθηκεύεται στο δίσκο. Κάποια στιγμή ο διαχειριστής της βάσης δεδομένων θα πρέπει να κάνει μια εκτίμηση σχετικά με τον χώρο που θα χρειαστεί η βάση για να διασφαλίσει ότι η χωρητικότητα αυτή είναι διαθέσιμη στο δίσκο. Ο προγραμματισμός της χωρητικότητας μιας βάσης δεδομένων

μπορεί να γίνει με απλή αριθμητική, όπως παρακάτω. Αυτή η απλή μέθοδος δεν λαμβάνει υπόψη της παράγοντες όπως η συσκευασία, η κωδικοποίηση και η συμπίεση. Αφήνοντας στην άκρη αυτές τις δυνατότητες το αποτέλεσμα είναι μια μεγαλύτερη χωρητικότητα.

1. Σε κάθε πίνακα αθροίστε το μέγεθος των πεδίων. Αυτό θα είναι το μέγεθος των εγγραφών του πίνακα. Αποφύγετε τις συνέπειες της συμπίεσης, την κωδικοποίησης και της συσκευασίας—με άλλα λόγια υποθέστε ότι κάθε αποθηκευμένος χαρακτήρας και ψηφίο θα καταναλώσει ένα byte μνήμης. Να θυμάστε ότι τα σημεία στίξης (π.χ. κόμμα, παύλες, κάθετοι, κλπ) δεν αποθηκεύονται σχεδόν ποτέ στις βάσεις δεδομένων.
2. Σε κάθε πίνακα πολλαπλασιάστε το μέγεθος των εγγραφών με τον αριθμό των οντοτήτων που θα περιληφθούν σε αυτόν. Καλό είναι να συνυπολογίσετε και την αύξηση που θα παρατηρηθεί μετά από ένα λογικό χρονικό διάστημα (π.χ. τρία χρόνια). Αυτό είναι το μέγεθος του πίνακα.
3. Προσθέστε τα μεγέθη των πινάκων. Αυτό είναι το μέγεθος της βάσης δεδομένων.
4. Εναλλακτικά αφήστε ένα περιθώριο ασφαλείας (π.χ. 10%) για απρόσμενους παράγοντες ή για ανακριβείς υπολογισμούς. Αυτή είναι η προβλεπόμενη χωρητικότητα της βάσης δεδομένων.

3.4.4.8 Δημιουργία δομής βάσης δεδομένων

Τα εργαλεία CASE συνήθως μπορούν να δημιουργήσουν έναν κώδικα SQL για τη βάση δεδομένων από ένα σχήματα βάσης δεδομένων βασισμένο σε αυτά. Ο κώδικας αυτό μπορεί να εξαχθεί στο DBMS για μεταγλώττιση. Ακόμα και μια μικρή βάση δεδομένων μπορεί να απαιτεί 50 ή περισσότερες σελίδες σε γλώσσα ορισμού δεδομένων SQL για τη δημιουργία πινάκων, δεικτών, κλειδιών, πεδίων και διεγερτών. Προφανώς η ιδιότητα των εργαλείων CASE να παράγουν αυτόματα το σωστό κώδικα αποτελεί σημαντικό παραγωγικό πλεονέκτημα. Επιπλέον συνήθως αποδεικνύεται ευκολότερη η τροποποίηση του σχήματος της βάσης δεδομένων και η εκ νέου δημιουργία του κώδικα από τη διατήρηση του κώδικα.

Κεφάλαιο 4 Μεθοδολογίες Ανάπτυξης Πληροφοριακών Συστημάτων (παράδειγμα OWL)

Το σύστημα αρχών, πρακτικών και διαδικασιών που εφαρμόζονται σε ένα συγκεκριμένο κλάδο γνώσης ονομάζεται μεθοδολογία. Συγκεκριμένα είναι ένα οργανωμένο σύνολο διαδικασιών, τεχνικών, εργαλείων και μέσων τεκμηρίωσης που καθοδηγεί τις ενέργειες σε ένα χώρο γνώσης, περιγράφεται από φάσεις, στάδια, βήματα, δραστηριότητες και βασίζεται σε θεωρητικές αρχές κάτω από μια οπτική γωνία.

Οι Μεθοδολογίες πληροφοριακών συστημάτων χωρίζονται σε δύο κατηγορίες. Αυτές είναι οι Δομημένες μεθοδολογίες (με προσανατολισμό στις λειτουργίες ενός συστήματος, δεκαετία '80) και Αντικειμενοστραφής μεθοδολογίες (με προσανατολισμό στα αντικείμενα ενός συστήματος), μεταξύ των οποίων θα γίνει επιλογή ανάλογα με τις απαιτήσεις και τα ζητούμενα που θα έχει το σύστημα. Ποιο κάτω αναλύονται κάποιες από τις μεθοδολογίες που χρησιμοποιούνται ευρέως.

4.1 Δομημένη προσέγγιση

Η Δομημένη προσέγγιση υιοθετεί κάποιες αρχές:

- Την αρχή της αφαίρεσης (abstraction). Τα πράγματα απλοποιούνται όταν διαιρούμαι αυτά που θεωρούμαι σημαντικά, με τη βοήθεια της αφαίρεσης μπορούμε να βλέπουμε τα πράγματα σε διαφορετικά επίπεδα που έχουν το δικό τους βαθμό αφαίρεσης και αποτελούνται από λειτουργίες.
- Η αρχή της αυστηρής τυπικότητας. Η μεθοδολογία πρέπει να είναι αυστηρή, συστηματική, βήμα προς βήμα διαδικασία, πράγμα που διευκολύνει την παρουσία ιδεών σε υπολογιστική μορφή (αλγοριθμική).
- Η αρχή του διαιρεί και βασίλευε (divide and conquer)
- Η δημιουργία ιεραρχική δομής (hierarchical ordering)

4.1.1 Μεθοδολογία STRADIS

STRADIS (Structured Analysis Design and Implementation of Information Systems) χρησιμοποιεί δομημένες διαγραμματικές τεχνικές. Αποτελείται από δραστηριότητες που μπορούν να ταξινομηθούν στις εξής κατηγορίες:

- Δημιουργία προκαταρκτικής μελέτης (initial study)
- Δημιουργία λεπτομερειακής μελέτης (detailed study)
- Εκτίμηση απαιτήσεων χρηστών
- Εκτίμηση φυσικής σχεδίασης
- Σταθεροποίηση απαιτήσεων
- Σχεδιασμός των αυτοματοποιημένων δραστηριοτήτων, σχεδιασμός των δραστηριοτήτων που θα εκτελούν άνθρωποι, εκτίμηση σχεδίου υλοποίηση και (φυσικός) σχεδιασμός δεδομένων
- Δημιουργία σχεδίων ελέγχου συστήματος
- Δημιουργία σχεδίων εγκατάστασης και ελέγχου αυτής

4.1.2 Μεθοδολογία Yourdon

Στηρίζεται στη δομημένη μεθοδολογία και χρησιμοποιεί τον διαμερισμό γεγονότων (event partitioning). Περιλαμβάνει 3 μεγάλες φάσεις:

Μελέτη σκοπιμότητας (Feasibility Study) εστιάζει στο υπάρχον σύστημα, το περιβάλλον του και τα προβλήματα που απασχολούν το σύστημα

Χρησιμοποιεί δομημένες διαγραμματικές τεχνικές και κυρίως διαγράμματα ροής δεδομένων (data flow diagrams) και διαγράμματα οντοτήτων – συσχετίσεων (entity-relationship diagrams)

Βασική μοντελοποίηση (Essential Modelling) – περιγραφή του μοντέλου του συστήματος που περιγράφει τι κάνει το σύστημα για να ικανοποιεί τις απαιτήσεις του χρήστη (δεν ασχολείται με την υλοποίησή ή την τεχνολογία υλοποίησής του)

Μοντέλο περιβάλλοντος (environmental model) – αποτελείται από τους σκοπούς και στόχους του συστήματος, το διάγραμμα πλαίσιο (context diagram) το οποίο συνοδεύεται από ένα λεξικό δεδομένων και τον πίνακα γεγονότων (event list)

Μοντέλο συμπεριφοράς (behavioural model) – αποτελείται από διαγράμματα ροής δεδομένων, διαγράμματα οντοτήτων και συσχετίσεων και διαγράμματα μετάβασης-κατασκευής (state transition diagram)

Μοντέλο υλοποίησης – δημιουργούνται οι προδιαγραφές σχεδιασμού λογισμικού, βάσεων δεδομένων, απαιτήσεων για δίκτυα κτλ.

4.1.3 Information Engineering

Μεθοδολογία πληροφοριών (IE) είναι η εφαρμογή αλληλοεξαρτώμενων (interlocking) αυστηρών (formal) τεχνικών για τον προγραμματισμό (planning) την ανάλυση το σχεδιασμό (design) και την κατασκευή του ΠΣ που καλύπτουν είτε ολόκληρη την επιχείρηση είτε σημαντικούς τομείς αυτής

Το αρχικό εργαλείο του information engineering είναι το πρότυπο διάγραμμα δεδομένων. Το information engineering περιλαμβάνει την διεύθυνση των επιχειρησιακών απαιτήσεων για την ανάλυση των εφαρμογών του πληροφοριακού συστήματος που θα παροπλισθούν ή θα τεθούν σε προτεραιότητα. Οι εφαρμογές που προσδιορίζονται μέσα στο information engineering γίνονται έργα στα οποία άλλες μέθοδοι ανάλυσης και σχεδιασμού συστημάτων, προορίζονται για να εφαρμοστούν προκειμένου να αναπτυχθούν τα συστήματα παραγωγής. Αυτές οι μέθοδοι περιλαμβάνουν ένα σχεδιασμό από σύγχρονη δομημένη ανάλυση, σύγχρονο δομημένο σχεδιασμό, prototyping και αντικειμενοστραφή ανάλυση και σχεδιασμό.

Τα στάδια της IE είναι επτά:

1. Σχεδιασμός Πληροφοριακής Στρατηγικής
2. Ανάλυση Επιχειρησιακού Συστήματος
3. Σχεδιασμός Επιχειρησιακού Συστήματος
4. Τεχνικός Σχεδιασμός Συστήματος
5. Κατασκευή
6. Μετάπτωση

7. Λειτουργία- Συντήρηση

4.1.4 Μεθοδολογία MERISE

Η MERISE αντιμετωπίζει τα δεδομένα και την επεξεργασία τους με τον ίδιο λεπτομερειακό τρόπο. Διακρίνει τη στατική και τη δυναμική όψη ενός συστήματος. θεωρεί ότι μια επιχείρηση αποτελείται από 3 υποσυστήματα: Το φυσικό σύστημα παραγωγής, το σύστημα διοίκησης/λήψης αποφάσεων και το πληροφοριακό σύστημα. Επίσης αποτελείται από τρεις κύκλους αλληλεπίδρασης:

1. Ο κύκλος ζωής (life cycle) Μακρόχρονος προγραμματισμός (long-range planning), Προκαταρκτική μελέτη (preliminary study), Αναλυτική μελέτη (detailed study), Υλοποίηση και εγκατάσταση (implementation and installation), Παραγωγή και συντήρηση (maintenance)
2. Ο κύκλος απόφασης (decision cycle)
3. Ο αφαιρετικός κύκλος (abstraction) που εστιάζει στα μοντέλα δεδομένων και επεξεργασιών (Νοητικό επίπεδο, Λογικό/οργανωτικό επίπεδο και Φυσικό επίπεδο)

4.1.5 Μεθοδολογία SSADM

Τα βασικά χαρακτηριστικά της Structured System Analysis and Design Method (SSADM) είναι ότι αποτελείται από συγκεκριμένα βήματα και σπάζει το έργο σε επιμέρους φάσεις (phases) – στάδια (stages) – βήματα (steps) – εργασίες (tasks).

Ο αναλυτής εκτελεί μια εργασία κάθε φορά

- Εύκολα προσαρμόζεται στα μέτρα κάθε περίπτωσης
- Καθοδηγείται από την ανάλυση δεδομένων
- Γίνεται επαλήθευση με συχνές επαναλήψεις
- Επιτρέπει να γίνουν τροποποιήσεις παντού
- Μπορεί να αναλυθεί το ίδιο σημείο από διαφορετικές ομάδες
- Εναλλακτικές λύσεις χρήστη

- Ιστορία ζωής οντότητας
- Σχεδιασμός διαλόγων
- Σχεσιακή ανάλυση δεδομένων
- Διαγράμματα δομής προγράμματος

Για την εφαρμογή του SSADM είναι απαραίτητη η δημιουργία ενός περιβάλλοντος υποστήριξης με συνιστώσες:

- Αρχικοποίηση έργου (project initiation)
- Διοίκησης έργου (project management)
- Εξασφάλισης ποιότητας (quality assurance)
- Προετοιμασία για υλοποίηση (preperation for implementation)
- Κατασκευή (construction)
- Υλοποίηση (implementation)
- Λειτουργία (operating)

4.1.6 Μεθοδολογία πολλαπλής θεώρησης

Η mutiview methodology αποτελείται από συνδυασμό της μεθοδολογίας (SSM) Checkland για ευμετάβλητα συστήματα και της συμμετοχικής μεθοδολογίας ETHICS του Mumford. Αποτελείται από 5 στάδια με τα εξής προϊόντα: Ανάλυση ανθρωπίνων δραστηριοτήτων, μοντέλο πρωταρχικών εργασιών, ανάλυση πληροφοριών, μοντέλο λειτουργιών και μοντέλο οντοτήτων, ανάλυση και σχεδιασμός κοινωνικο-τεχνικών πλευρών, απαιτήσεις υπολογιστικών εργασιών, σύνολο ρόλων, εργασίες ανθρώπων, σχεδιασμός διαπεφής ανθρώπου –υπολογιστή, τεχνικές απαιτήσεων διεπαφών, σχεδιασμός τεχνικών πλευρών, σχεδιασμός βάσεων δεδομένων, λειτουργίες για ανάκτηση δεδομένων και απαιτούμενοι έλεγχοι

4.1.7 Κατασκευή προτύπου (prototyping)

Παραδοσιακά ο φυσικός σχεδιασμός ήταν μια διαδικασία που γινόταν στο χαρτί. Οι αναλυτές σχεδίαζαν εικόνες που απεικόνιζαν το σχεδιάγραμμα ή την

δομή των αποτελεσμάτων, των δεδομένων, των βάσεων δεδομένων και την ροή των διαδικασιών. Αυτή η εργασία είναι χρονοβόρα και επιρρεπείς στα λάθη αλλά και στις παραλήψεις που πιθανόν να γίνουν. Συχνά τα αποτελέσματα που προκύπτουν από την καταγραφή σε χαρτί είναι ελλιπή ή ανακριβή.

Σήμερα πολλοί αναλυτές και σχεδιαστές προτιμούν το prototyping, μια μοντέρνα προσέγγιση του σχεδιασμού με βάση την μηχανική. Η προσέγγιση prototyping είναι μια επαναλαμβανόμενη διαδικασία η οποία συνδυάζει μια στενή εργασιακή σχέση μεταξύ του σχεδιαστή και των χρηστών.

Prototypes μπορούν να αναπτυχθούν γρήγορα με την χρήση των 4GIs και των γλωσσών προγραμματισμού που είναι αντικειμενοστραφείς. Prototypes μπορούν να σχεδιαστούν για ένα απλό αποτέλεσμα, computer dialogs, key factions, ολόκληρο υποσύστημα ή και για ολόκληρο το σύστημα. Κάθε σύστημα prototype εκτιμάται και αξιολογείται από τους τελικούς χρήστες και του διαχειριστές οι οποίοι κάνουν τις προτάσεις τους όσον αφορά τις απαιτήσεις, τις μεθόδους και formats. Τα prototypes στη συνέχεια διορθώνονται, ενισχύονται ή επαναπροσδιορίζονται για να απεικονίζουν τις νέες απαιτήσεις. Με τεχνολογία prototyping οι αναθεωρήσεις γίνονται με απλό τρόπο. Η αναθεώρηση, η διαδικασία εκτίμησης και αξιολόγησης συνεχίζεται μέχρι να γίνει αποδεκτό το prototype. Σε αυτό το σημείο οι τελικοί χρήστες αποδέχονται και τις απαιτήσεις και τον σχεδιασμό που εκπληρώνει τις απαιτήσεις.

Ο σχεδιασμός με prototyping δεν εκπληρώνει απαραίτητα όλες τις απαιτήσεις της φάσης σχεδιασμού. Για παράδειγμα, τα prototypes ενσωματώνουν σπανίως τους εσωτερικούς ελέγχους. Επίσης δεν αντιμετωπίζουν ζητήματα απόδοσης και περιορισμού αποθήκευσης. Οι αναλυτές ή οι σχεδιαστές είναι αυτοί που θα πρέπει να τα προσδιορίσουν.

4.1.8 μειονεκτήματα δομημένης προσέγγισης

Η αλλαγή από μια φάση στη άλλη είναι δύσκολη, όπως δύσκολος είναι και ο έλεγχος της ορθότητας κάθε μετασχηματισμού

Ο σωστός μετασχηματισμός ακόμη και αν έχει επιτευχθεί δεν μπορεί να διαρκέσει γιατί όταν παρουσιάζονται νέες απαιτήσεις που συχνά παρουσιάζονται με αλλαγές στον υπάρχοντα κώδικα

4.2 Αντικειμενοστρεφής προσέγγιση (Object Oriented Methodology)

Η αντικειμενοστρεφής προσέγγιση διευκολύνει την μετάβαση από μια φάση σε άλλη παραμερίζοντας το εμπόδιο του ριζικού μετασχηματισμού, χρησιμοποιώντας πάντα την ίδια γλώσσα.

Τα αντικείμενα του πραγματικού περιβάλλοντος εξετάζονται στην ανάλυση απαιτήσεων, μεταφράζονται άμεσα σε αντικείμενα φυσικού σχεδιασμού, υλοποιούνται και ελέγχονται ως αντικείμενα αντικειμενοστραφούς γλώσσας προγραμματισμού (object oriented programming - OPP) ή βάσης δεδομένων (object oriented data base system - OODBS).

Η δημιουργία ΠΣ με αντικειμενοσταφή προσέγγιση γίνεται σε 3 επίπεδα:

1. Μοντέλο
2. Κλάσεις
3. Εφαρμογές

Οι πλέον βασικές έννοιες της αντικειμενοστραφούς προσέγγισης είναι οι εξής:

- Το αντικείμενο (object) – μια λογική οντότητα που συνδέει δεδομένα και επεξεργασίες για να εκπληρώσει το ρόλο της μέσα στο σύστημα
- Κατάσταση (state)
- Συμπεριφορά (behaviour)
- Ταυτότητα (identity)
- Οι μέθοδοι (methods)
- Τα μηνύματα (messages)
- Κλάση
- Οψη
- Κληρονομικότητα

Αυτές οι έννοιες βοηθούν στην κατανόηση των:

Ενσωμάτωση (encapsulation) - Τμηματοποίηση (modularity) - Αφαίρεση (abstraction) - Πολυμορφισμός (polymorphism)

Η οργάνωση των κλάσεων χρησιμοποιείται για να γίνει το μοντέλο του φυσικού κόσμου, σε αυτό βοηθούν οι:

- Ιεραρχίες (hierarchies)
- Κληρονομικότητα (inheritance)

4.2.1 Coad & Yourdon

Αποτελείται από τα ακόλουθα στάδια:

- Προσδιορισμός αντικειμένων (objects)
- Προσδιορισμός δομής (structure)
- Προσδιορισμός θεμάτων (subjects)
- Προσδιορισμός χαρακτηριστικών ιδιοτήτων (attributes)
- Προσδιορισμός υπηρεσιών (services)
- Προσδιορισμός της κατάστασης αντικειμένων
- Προσδιορισμός των απαραίτητων υπηρεσιών
- Προσδιορισμός της σύνδεσης μηνυμάτων

4.2.2 OMT (Object Modelling Technique)

Έχει τις ρίζες της στις δομημένες μεθοδολογίες αλλά χρησιμοποιεί πολύπλοκους συμβολισμούς (CASE). Στην φάση της ανάλυσης δημιουργεί:

- Τον ορισμό του προβλήματος
- Το μοντέλο αντικειμένων
- Το δυναμικό μοντέλο
- Το λειτουργικό μοντέλο

Κατά τη μοντελοποίηση αντικειμένων γίνεται:

- Προσδιορισμός αντικειμένων και κλάσεων
- Προετοιμασία του λεξικού κλάσεων

- Προσδιορισμός των συνδέσεων μεταξύ αντικειμένων
- Προσδιορισμός των ιδιοτήτων των αντικειμένων και των κλάσεων
- Οργάνωση και απλοποίηση των κλάσεων των αντικειμένων κάνοντας χρήση της κληρονομικότητας

Στο δυναμικό μοντέλο γίνεται:

- Προετοιμασία σεναρίων
- Προσδιορισμός καταστάσεων (states) και γεγονότων (events)
- Προετοιμασία ιχνηλασίας γεγονότων για κάθε σενάριο
- Δημιουργία διαγράμματος κατάστασης (state diagram)
- Σύνδεση γεγονότων μεταξύ των αντικειμένων έτσι ώστε να πιστοποιηθεί η συνέπεια

Το λειτουργικό μοντέλο περιγράφει ροές δεδομένων, τα κέντρα λειτουργιών, και λειτουργικές εξαρτήσεις δεδομένων

4.2.3 OOAD (Object Oriented Analysis & Design)

- Οι Marting & O'Dell περιγράφουν την OOAD ως εξής:
- Ορισμός σκοπού της ανάλυσης
- Προσδιορισμός των πεδίων
- Προσδιορισμός του τελικού στόχου
- Διερεύνηση των τύπων των γεγονότων
- Γενίκευση των τύπων των γεγονότων
- Ορισμός των συνθηκών των λειτουργιών
- Προσδιορισμός της λειτουργίας
- Προσδιορισμός πότε η λειτουργία είναι εσωτερική/εξωτερική
- Προσδιορισμός συνθηκών ελέγχου
- Προσδιορισμός της αιτίας της λειτουργίας
- Έλεγχος κυκλικών αποτελεσμάτων

4.2.4 OBA (Object Behaviour Analysis)

Grady Booch (1991) προτείνει το σχεδιασμό συστημάτων που περιλαμβάνει δραστηριότητες που ταξινομούνται σε 4 ομάδες:

- Λογική δομή
- Διαγράμματα κλάσεων
- Διαγράμματα αντικειμένων
- Φυσική δομή
- Διαγράμματα ενοτήτων
- Διαγράμματα επεξεργασιών
- Δυναμική των κλάσεων
- Διαγράμματα μετάβασης κατάστασης
- Δυναμική των όψεων
- Διαγράμματα χρόνου

4.2.5 Rapid Application Development

Μια άλλη δημοφιλής στρατηγική σχεδιασμού που χρησιμοποιείται σήμερα είναι η Rapid Application Development (R.A.D). Είναι ένας συνδυασμός διάφορων δομημένων τεχνικών (ειδικά της data driven information engineering) με τεχνικές prototyping και joint application development για της επιτάχυνση της ανάπτυξης συστημάτων.

Η R.A.D. απαιτεί αμφίδρομη χρησιμοποίηση των δομημένων τεχνικών και του prototyping για να καθοριστούν οι απαιτήσεις των χρηστών και ο σχεδιασμός του τελικού συστήματος. Η χρησιμοποίηση δομημένων τεχνικών, από τον υπεύθυνο ανάπτυξης, πρώτα φτιάχνει τα preliminary data and process models από τις απαιτήσεις της επιχείρησης. Μετά τα prototypes βοηθούν τον αναλυτή και τους χρήστες να ελέγξουν εκείνες τις απαιτήσεις και το ξεκαθάρισμα των μοντέλων δεδομένων και των διαδικασιών. Ο κύκλος των μοντέλων είναι πρωτότυπα κατόπιν μοντέλα και πάλι πρότυπα και ούτω καθ' εξής. Τα τελικά

πρότυπα είναι ένας συνδυασμός των απαιτήσεων της επιχείρησης και του τεχνικού σχεδιασμού για να χρησιμοποιηθούν στην κατασκευή ενός νέου συστήματος.

Η προσπάθεια για τον σχεδιασμό ενισχύεται δίνοντας έμφαση στη συμμετοχή του χρήστη στην Joint Application Development Session (J.A.D.). Η J.A.D. είναι μια τεχνική που συνδυάζει άλλες τεχνικές ανάλυσης και σχεδιασμού συστημάτων δίνοντας έμφαση στην *participative development* των ιδιοκτητών του συστήματος, των χρηστών, των σχεδιαστών και των κατασκευαστών. Κατά την διάρκεια των session της J.A.D. για τον σχεδιασμό του συστήματος οι σχεδιαστές θα πάρουν το ρόλο του βοηθού σε εργαστήρια για την καθοδήγηση διαφορετικών συστημάτων σχεδιασμού και παραδοτέων. Η J.A.D. είναι ουσιαστικά ένα αντικείμενο που συμβάλει στην επιτάχυνση της R.A.D.

4.2.6 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. Η UML στοχεύει στο σχεδιασμό αντικειμενοστρεφών συστημάτων. Το σχέδιο είναι μια απλοποιημένη παράσταση της πραγματικότητας. Προήλθε από την ενοποίηση των συμβολισμών που χρησιμοποιούσαν οι Μεθοδολογίες Booch, OMT, OOSE κ.λπ. Ενσωματώνει τις ιδέες καλής πρακτικής από τη βιομηχανία Λογισμικού και είναι ένα βιομηχανικό πρότυπο που αναπτύχθηκε στο πλαίσιο του οργανισμού OMG. Υλοποιεί την ανάγκη της βιομηχανίας λογισμικού για μια εννοιαία γλώσσα μοντελοποίησης

Διαφορετικές μεθοδολογίες ανάπτυξης είναι δυνατόν να υιοθετηθούν έχοντας τη UML ως γλώσσα μοντελοποίησης (π.χ. USDP, RUP, κλπ) Κάθε μεθοδολογία προτείνει τα δικά της μοντέλα ανάλογα με το μοντέλο που μας ενδιαφέρει (Daniels 2002).

Στη UML ορίζονται τέσσερις βασικές σχέσεις:

1. εξάρτηση (*dependency*)

2. γενίκευση (*generalisation*)
3. σύνδεση (*association*)
4. υλοποίηση (*realisation*)

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

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

3. 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 μιας σε αντικείμενα της άλλης. Η σύνδεση παριστάνεται με μια 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.

4. Υλοποίηση : Η υλοποίηση δηλώνει πως ο εξυπηρετούμενος (αυτός που βρίσκεται στην ουρά του βέλους) υποστηρίζει τη διεπαφή (τουλάχιστον όλες τις πράξεις) που ορίζονται από τον παροχέα (αυτόν που βρίσκεται στην κεφαλή του βέλους)

4.3 Οντολογίες

Μια οντολογία καθορίζει τους βασικούς όρους και τις σχέσεις που συμπεριλαμβάνουν το λεξικό όρων (*dictionary*) ενός θέματος, καθώς επίσης κανόνες για τους όρους και τις σχέσεις που καθορίζουν τις επεκτάσεις στο λεξικό αυτό. Επίσης προβάλλεται η διαμόρφωση των οντολογιών που βοηθά στη

σωστή διαχείριση του μεγάλου όγκου πληροφοριών που περιτριγυρίζουν μια επιχείρηση καθώς και στη σωστή και ακριβή περιγραφή των διαδικασιών – εννοιών – λειτουργιών που διέπουν έναν επιχειρηματικό οργανισμό, έτσι ώστε να προκαθορίζονται οι ενέργειες των χρηστών και των πληροφοριακών συστημάτων για κάθε επιχειρηματική διαδικασία (παραγωγική, διοικητική κ.λ.π.).

Ο κύριος στόχος των οντολογιών είναι η διανομή και επαναχρησιμοποίηση της γνώσης έτσι ώστε να μην χάνονται οι πολύτιμες πληροφορίες μιας επιχείρησης αλλά να μπορούν με κάποιο τρόπο να διαρθρώνονται και να αξιοποιούνται. Με την χρήση των οντολογιών μπορούμε να αποφεύγουμε λάθη που οφείλονται στην εκφραστικότητα και στην ακρίβεια των διαδικασιών και των λειτουργιών ενός οργανισμού/επιχείρησης. Ένας επίσης σημαντικός λόγος που καθιστά αναγκαία την δημιουργία της οντολογίας είναι η επικοινωνία μεταξύ των χρηστών των πληροφοριακών συστημάτων (IS) καθώς η σωστή και ασφαλή ανταλλαγή δεδομένων και πληροφοριών έτσι ώστε να υπάρχει πλήρη και ορθή επικοινωνία μεταξύ τμημάτων, υπαλλήλων, πελατών και επιχειρήσεων.

4.3.1 Ταξινόμηση Οντολογιών

Υπάρχει ένας μεγάλος αριθμός διαφορετικών τύπων οντολογιών. Ο Obitko ορίζει τους παρακάτω τύπους οντολογιών.

Η *οντολογία εργασιακού χώρου* διευκρινίζει τους όρους που χαρακτηρίζουν και δικαιολογούν τη συμπεριφορά επίλυσης προβλημάτων στον εργασιακό χώρο. Μια *οντολογία στόχου* αποτελείται από ένα λεξιλόγιο για την περιγραφή μιας δομής επίλυσης προβλήματος όλων των υπαρχόντων στόχων, ανεξάρτητα από την περιοχή. Η γνώση του στόχου δίνει τους ρόλους σε κάθε αντικείμενο και τις σχέσεις που υπάρχουν μεταξύ τους. Μια *γενική οντολογία* καλύπτει τα γενικά ή κοινά αντικείμενα, διάφορα πράγματα όπως, τα γεγονότα, ο χρόνος, το διάστημα, οι περιγραφικοί όροι κ.λπ.

Ο Mizoguchi διακρίνει μεταξύ της οντολογίας στόχου και την οντολογία περιοχών. Μια οντολογία στόχου χαρακτηρίζει την υπολογιστική αρχιτεκτονική

ενός βασισμένου στη γνώση συστήματος που εκτελεί έναν στόχο. Η οντολογία περιοχών χαρακτηρίζει τη γνώση περιοχών όπου ο στόχος εκτελείται.

Ο Heijst ταξινομεί τις οντολογίες σύμφωνα με δύο διαφορετικές διαστάσεις. Η πρώτη εξετάζει τον τύπο δομής της σύλληψης και η δεύτερη εξετάζει το θέμα της σύλληψης. Στην πρώτη διάσταση υπάρχουν τρεις διαφορετικές κατηγορίες. α) Οι οντολογίες ορολογίας, π.χ. τα λεξικά, που διευκρινίζουν τους όρους που χρησιμοποιούνται για να αντιπροσωπεύσουν τη γνώση σε μια συγκεκριμένη περιοχή. β) Οι οντολογίες πληροφοριών, όπως τα σχήματα βάσεων δεδομένων, που διευκρινίζουν τη δομή αρχείων των βάσεων δεδομένων. γ) Οι οντολογίες διαμόρφωσης γνώσης που διευκρινίζουν τις συλλήψεις της γνώσης και έχουν μια πλουσιότερη εσωτερική δομή από τις οντολογίες πληροφοριών. Είναι συχνά εξειδικευμένες για μια ιδιαίτερη χρήση της γνώσης που περιγράφουν. Στην άλλη διάσταση διακρίνουμε τέσσερις διαφορετικές κατηγορίες. α) Οι οντολογίες εφαρμογής που συσχετίζονται με μια συγκεκριμένη εφαρμογή και διαμορφώνουν τη γνώση που απαιτείται για αυτό. β) Οι οντολογίες περιοχών που είναι συγκεκριμένες για τις ιδιαίτερες περιοχές. γ) Οι γενικές οντολογίες καθορίζουν τις έννοιες που είναι γενικές σε πολλούς τομείς. δ) Τέλος, οι οντολογίες αντιπροσωπευσης που παρέχουν ένα αντιπροσωπευτικό πλαίσιο χωρίς παραγωγή των αξιώσεων για τον κόσμο.

Ακόμα ένας χωρισμός μεταξύ διαφορετικών οντολογιών έγινε από τον McGuinness και αυτός ορίζει τρεις διαφορετικούς τύπους οντολογιών. α) Οι οντολογίες των πόρων καθορίζουν τη σημασιολογία που χρησιμοποιούνται στα συστήματα λογισμικού. β) Οι προσωπικές οντολογίες καθορίζουν τη σημασιολογία ενός χρήστη ή μιας ομάδας χρηστών και γ) οι κοινές οντολογίες καθορίζουν την κοινή σημασιολογία που μοιράζονται μεταξύ των συστημάτων πληροφοριών.

Μία οντολογία εξαρτημένη από εφαρμογή δεν είναι τόσο επαναχρησιμοποιήσιμη, ενώ μια γενική οντολογία μπορεί να επαναχρησιμοποιηθεί εύκολα σε διαφορετικά προγράμματα.

4.3.2 Εργαλεία Ανάπτυξης Οντολογιών

Τα τελευταία χρόνια, ο αριθμός εργαλείων για τις οντολογίες έχει αυξηθεί εκθετικά. Αυτά τα εργαλεία στοχεύουν στην παροχή της υποστήριξης για τη διαδικασία ανάπτυξης οντολογίας. Σε αυτό το τμήμα παρουσιάζονται τα πιο σχετικά από αυτά. Ο στόχος αυτής της ενότητας είναι να παρουσιαστούν οι διάφορες κατηγοριοποιήσεις των διαφορετικών εργαλείων οντολογίας που υπάρχουν σήμερα. Σύμφωνα με την έρευνα του OntoWeb Project και τη συμμετοχή του SIG on Enterprise - Standard Ontology Environments, τα διαφορετικά εργαλεία οντολογίας θα μπορούσαν να ταξινομηθούν, ως εξής:

Εργαλεία ανάπτυξης οντολογίας, τα οποία μπορούν να χρησιμοποιηθούν για την οικοδόμηση μιας νέας οντολογίας από την αρχή ή την επαναχρησιμοποίηση των οντολογιών που ήδη υπάρχουν. Εκτός από την κοινή λειτουργία εκδόσεων, αυτά τα εργαλεία περιλαμβάνουν συνήθως την τεκμηρίωση οντολογίας, την εξαγωγή οντολογίας και την εισαγωγή από τα διαφορετικά σχήματα, τις γραφικές απόψεις των οντολογιών που χτίζονται, τις βιβλιοθήκες οντολογίας, τις συνημμένες μηχανές συμπεράσματος, κ.λπ.

Το ontolingua Server ήταν το εργαλείο οντολογίας που δημιουργήθηκε αρχικά και αναπτύχθηκε στο εργαστήριο συστημάτων γνώσης (KSL) στο πανεπιστήμιο του Στάνφορντ. Το εργαλείο Ontolingua-Server εμφανίστηκε στην αρχή της δεκαετίας του '90 και χτίστηκε για να διευκολύνει την ανάπτυξη των οντολογιών Ontolingua. Αρχικά, ο κύριος ρόλος του εργαλείου Ontolingua-Server ήταν η σύνταξη της οντολογίας.

Συγχρόνως, ένα άλλο εργαλείο το Ontosaurus αναπτύχθηκε από το ίδρυμα επιστημών των πληροφοριών (ISI) στο πανεπιστήμιο της νότιας Καλιφόρνιας. Το συγκεκριμένο εργαλείο αποτελείται από δύο ενότητες: από έναν κεντρικό υπολογιστή οντολογίας, ο οποίος χρησιμοποιεί το μεταφραστή Loom ως σύστημα αντιπροσώπευσης γνώσης και έναν web browser διαθέσιμο για τις οντολογίες.

Το Protege2000 έχει αναπτυχθεί από την ιατρική πληροφορική του Στάνφορντ (SMI) στο πανεπιστήμιο του Στάνφορντ, και είναι η πιο πρόσφατη έκδοση της γραμμής εργαλείων Protege. Ο πυρήνας αυτού του συγκεκριμένου εργαλείου είναι ο συντάκτης οντολογίας και χρησιμοποιεί μια βιβλιοθήκη δεδομένων που προσθέτει περισσότερη και καλύτερη λειτουργία στο περιβάλλον.

Εργαλεία συγχώνευσης(merging) και ολοκλήρωσης(integration) οντολογίας, τα οποία υποστηρίζουν την ανάγκη συγχώνευσης ή ολοκλήρωσης των διαφορετικών οντολογιών.

Κάποια αντιπροσωπευτικά εργαλεία οντολογίας αυτής της κατηγορίας είναι το PROMPT το οποίο κατασκευάστηκε και αυτό από την ιατρική πληροφορική του Στάνφορντ (SMI) στο πανεπιστήμιο του Στάνφορντ καθώς και το ODEMerge που αναπτύχθηκε από το Ontology Group LIA – UPM.

Εργαλεία σχολιασμών(anoatation) οντολογίας, τα οποία επιτρέπουν στους χρήστες να εισάγουν και να διατηρούν (ημι)αυτόματα τα ποσοστά κέρδους στις ιστοσελίδες.

Τα πιο χαρακτηριστικά εργαλεία εδώ είναι το AeroDAML το οποίο αναπτύχθηκε από το UML Based Ontology Toolset (UBOT) Project και το OntoMat – Anotizer που κατασκευάστηκε από το OntoAgent Project.

Εργαλεία Αποθήκευσης και Ερωτημάτων οντολογίας, τα οποία έχουν αναπτυχθεί για να χρησιμοποιούνται οι οντολογίες πιο εύκολα.

Εδώ θα αναφερθούν τρία εργαλεία για την συγκεκριμένη κατηγορία. Το RedLand που αναπτύχθηκε στο πανεπιστήμιο του Bristol, το Jena της Hewlett – Packard και το RdfGateway της Intellidimension Company.

Εργαλεία εκμάθησης οντολογίας, που χρησιμοποιούνται (ημι)αυτόματα για να αντλούν τις οντολογίες από τα κείμενα φυσικής γλώσσας.

Αντιπροσωπευτικά εργαλεία εκμάθησης οντολογιών είναι το ASIUM Tool και το OntoLearn Tool.

Εργαλεία αξιολόγησης οντολογίας, τα οποία εξασφαλίζουν ότι τόσο οι οντολογίες όσο και οι σχετικές τεχνολογίες τους έχουν ένα δεδομένο επίπεδο ποιότητας.

Αυτή η κατηγορία είναι ιδιαίτερα σημαντική γιατί έχει να κάνει με την σωστή λειτουργία των οντολογιών και την ανταπόκρισή τους στο σκοπό για τον οποίο δημιουργήθηκαν. Σε αυτό βοηθά το OntoAnalyser που αναπτύχθηκε από την Ontoprise GmbH και το OntoClean in WebODE που κατασκευάστηκε στη Μαδρίτη από το Ontology Group(UPM).

4.3.3 Γλώσσες Οντολογιών

Αυτό το τμήμα στοχεύει στην παρουσίαση μιας γενικής άποψης του τομέα των γλωσσών αντιπροσώπευσης μιας οντολογίας, καθώς επίσης και στην υπόδειξη ενός ενδεικτικού κατάλογου γλωσσών αντιπροσώπευσης.

Πολλές γλώσσες που μπορούν να χρησιμοποιηθούν για τη διευκρίνιση της οντολογίας υπάρχουν σήμερα. Ανάλογα με τη γλώσσα που χρησιμοποιείται για να διευκρινίσει μια οντολογία, σύμφωνα με το IDEAS Project προσδιορίζονται τα ακόλουθα επίπεδα τυπικότητας:

- *Ιδιαίτερα άτυπες οντολογίες* όταν εκφράζονται οι οντολογίες στη φυσική γλώσσα.
- *Ημι-άτυπες οντολογίες* όταν εκφράζονται σε μια περιορισμένη και δομημένη μορφή φυσικής γλώσσας.
- *Ημι-τυπικές οντολογίες* όταν εκφράζονται σε μια τεχνητή και τυπικά καθορισμένη γλώσσα.
- *Αυστηρά επίσημες οντολογίες* όταν καθορίζονται σε μια γλώσσα με την επίσημη σημασιολογία, τα θεωρήματα και τις αποδείξεις τέτοιων ιδιοτήτων όπως είναι η πληρότητα.

Στις πρώτες τρεις κατηγορίες, η γλώσσα που χρησιμοποιείται είναι είτε φυσική γλώσσα είτε διαγράμματα (π.χ. ER, UML, EPC, κ.λπ...). Στη τέταρτη κατηγορία, οι γλώσσες που χρησιμοποιούνται είναι είτε βασισμένες στη λογική (Description Logic, First Order Logic, Action Logic, Horn Logic, etc.), είτε στην άλγεβρα (Z formalism, VDM ++).

Η επιλογή μιας γλώσσας για τη διευκρίνιση της οντολογίας εξαρτάται από τα χαρακτηριστικά της γλώσσας, καθώς επίσης και από τα εργαλεία που την υποστηρίζουν, τις εφαρμογές στις οποίες η οντολογία θα χρησιμοποιηθεί και τη διαθεσιμότητα των επαναχρησιμοποιήσιμων οντολογιών στην ίδια περιοχή σε μια συγκεκριμένη γλώσσα.

Οι περισσότερες από τις γλώσσες είναι βασισμένες στα πλαίσια ή τη λογική περιγραφής, ενώ μερικές από αυτές παρέχουν τους κατασκευαστές για άλλα παραδείγματα, όπως οι εννοιολογικές γραφικές παραστάσεις, τα σημασιολογικά δίκτυα, οι κανόνες παραγωγής, οι παραγωγικοί κανόνες, κ.λπ. Σε πολλές περιπτώσεις, οι γλώσσες είναι βασισμένες στους συνδυασμούς διαφόρων φορμαλισμών.

Στην επιλογή της σωστής γλώσσας που θα χρησιμοποιεί, τα ακόλουθα δύο κριτήρια μπορούν να χρησιμοποιηθούν για να αναλύσουν τα χαρακτηριστικά των διαφορετικών γλωσσών αντιπροσώπευσης.

- *Η εκφραστική δύναμη* της γλώσσας, η οποία συσχετίζεται με τη σειρά των κατασκευασμάτων που μπορούν να χρησιμοποιηθούν για να περιγράψουν τα συστατικά της οντολογίας.
- *Η σημασιολογική επάρκεια* της γλώσσας, η οποία μετρά την υποστήριξη που η γλώσσα παρέχει στην αντιπροσώπευση της επιλεγμένης περιοχής, καθώς επίσης και την ευκολία στην αντιπροσώπευσή της.

Φαίνεται ότι όσο πιο εκφραστική μια γλώσσα είναι, τόσο λιγότερο σημασιολογική επάρκεια παρέχει.

Υπάρχει επίσης μια δυνατή αλληλεξάρτηση μεταξύ της εκφραστικότητας και του συλλογισμού σε όλες τις γλώσσες, υπό την έννοια ότι η εκφραστική δύναμη μιας γλώσσας πρέπει να περιοριστεί μερικές φορές για να εξασφαλιστεί μια καλά αιτιολογημένη υπηρεσία. Από αυτή την άποψη μερικές γλώσσες έχουν αναπτυχθεί ακολουθώντας κάποια προσέγγιση (OML, OIL), στην οποία τα χαμηλότερα στρώματα (λιγότερο εκφραστικά) έχουν την πλήρη υποστήριξη συλλογισμού και τα υψηλότερα στρώματα (πιο εκφραστικά) έχουν περιορίσει την υποστήριξη συλλογισμού.

Παρακάτω, στον πίνακα 3.1 παρουσιάζεται ένας ενδεικτικός κατάλογος γλωσσών προδιαγραφών. Οι γλώσσες ομαδοποιούνται σε δύο κατηγορίες: Ωριμες Γλώσσες, που είναι σε μια σταθερή φάση ανάπτυξης και Γλώσσες που έχουν αναπτυχθεί στα πλαίσια του World Wide Web.

Πίνακας 3.1

	Language Name
Mature Languages	Cycl
	KIF – Knowledge Interchange Format
	Ontolingua
	OCML
	LOOM
XML based Languages	SHOE
	XOL
	OML - Ontology Markup Language
	RDF(S)
	OIL
	DAML + OIL
	OWL - Ontology Web Language
	Topic Maps

4.3.3.1 SHOE

Η SHOE αναπτύσσεται στο πανεπιστήμιο της Maryland. Αρχικά, η SHOE δεν χρησιμοποίησε XML, αλλά μια επέκταση της HTML, επειδή η αρχική προδιαγραφή της παρουσιάστηκε πολύ νωρίς, το 1996. Εντούτοις, τώρα κινείται προς XML. Η SHOE προσθέτει τις ετικέτες που είναι απαραίτητες για να ενσωματώσουν τα αυθαίρετα σημασιολογικά στοιχεία ιστοσελίδας, οι οποίες διαιρούνται σε δύο κατηγορίες: ετικέτες για την κατασκευή των οντολογιών και ετικέτες για το σχολιασμό των εγγράφων Ιστού.

4.3.3.2 Ontolingua

Η Ontolingua αναπτύχθηκε το 1992 από το KSL (πανεπιστήμιο του Stanford), συνδυάζοντας τα παραδείγματα KR των πλαισίων (frame ontology). Επιτρέπει την αντιπροσώπευση των περιεχομένων, των ταξινομιών των εννοιών,

των n- καταστάσεων σχέσεων, των λειτουργιών, των αξιωμάτων, των περιπτώσεων και των διαδικασιών. Η υψηλή εκφραστικότητα της οδήγησε στις δυσκολίες στην οικοδόμηση των μηχανισμών συλλογισμού για αυτή.

4.3.3.3 OCML

Η γλώσσα μοντελοποίησης OCML αναπτύχθηκε το 1993 στο KMI (Open University) στα πλαίσια του προγράμματος VITAL. Είναι παρόμοια με την Ontolingua, παρέχοντας και πρόσθετα συστατικά, όπως αφαιρετικούς και παραγωγικούς κανόνες. Μπορεί να θεωρηθεί ως είδος "λειτουργικού Ontolingua" (operational Ontolingua). Επιτρέπει την προδιαγραφή και τη λειτουργικότητα των διαδικασιών, των σχέσεων, των κατηγοριών, των περιπτώσεων και των κανόνων. Περιλαμβάνει επίσης έναν πολύ ισχυρό έλεγχο περιορισμών, που μπορεί να ελέγξει και τους περιορισμούς τύπων και αριθμού στοιχείων συνόλου, καθώς επίσης και τους γενικούς περιορισμούς που συνδέονται με τις σχέσεις και τις κατηγορίες. Η OCML έχει χρησιμοποιηθεί σε δεκάδες εφαρμογές, για να υποστηρίξει τις διοικητικές διαδικασίες γνώσης (Knowledge Management processes), την ανάπτυξη οντολογίας, το ηλεκτρονικό εμπόριο και την ανάπτυξη των συστημάτων που βασίζονται στη γνώση. Οι περιοχές εφαρμογής περιλαμβάνουν την ιατρική, τις ηλεκτρονικές εκδόσεις, τις κοινωνικές επιστήμες, το σχέδιο εφαρμοσμένης μηχανικής, τις πύλες Ιστού και διάφορα άλλα. Η γλώσσα OCML υποστηρίζεται επίσης από μια μεγάλη βιβλιοθήκη των επαναχρησιμοποιήσιμων προτύπων, που παρέχουν έναν χρήσιμο πόρο για τη διαμόρφωση της γνώσης.

4.4 OWL

Η OWL, η Γλώσσα Οντολογίας του Παγκοσμίου Ιστού παρέχει μια γλώσσα για τον ορισμό δομημένων οντολογιών που βασίζονται στον Παγκόσμιο Ιστό, η οποία φέρνει πλουσιότερο συνδυασμό και διαλειτουργικότητα δεδομένων μεταξύ περιγραφικών κοινοτήτων. Εκεί όπου προγενέστερες γλώσσες είχαν χρησιμοποιηθεί για την ανάπτυξη εργαλείων και οντολογιών για συγκεκριμένες κοινότητες χρηστών (ειδικά στις επιστήμες και σε ειδικά για κάθε επιχείρηση εφαρμογές ηλεκτρονικού εμπορίου), δεν είχαν οριστεί να είναι συμβατές με την αρχιτεκτονική του Παγκοσμίου Ιστού γενικότερα, καθώς και του Σημασιολογικού Ιστού συγκεκριμένα.

Η OWL χρησιμοποιεί URIs για ονομασία, καθώς και το πλαίσιο περιγραφής για τον Παγκόσμιο Ιστό που παρέχεται από το RDF, προσθέτοντας τις παρακάτω δυνατότητες στις οντολογίες:

- Ικανότητα κατανομής σε εύρος συστημάτων
- Προσαρμογή στις ανάγκες του Παγκοσμίου Ιστού
- Συμβατότητα με τα πρότυπα του Παγκοσμίου Ιστού για προσβασιμότητα και διεθνοποίηση
- Επεκτασιμότητα και δυνατότητα να είναι πιο ανοιχτές

Επίσης χτίζει σε RDF και RDF Schema προσθέτοντας περισσότερα λεξιλόγια για την περιγραφή ιδιοτήτων και τάξεων: μεταξύ άλλων, σχέσεις τάξεων (για παράδειγμα ασυναρτησία), πλήθος στοιχείων συνόλου (για παράδειγμα "ακριβώς ένα"), ισότητα, πλουσιότερη πληκτρολόγηση ιδιοτήτων, χαρακτηριστικά ιδιοτήτων (για παράδειγμα συμμετρία), καθώς και αριθμημένες τάξεις.

"Η OWL κάνει ένα σημαντικό βήμα μπροστά στην αναπαράσταση και οργάνωση της γνώσης στον Παγκόσμιο Ιστό. Επιτυγχάνει ισορροπία ανάμεσα στις ανάγκες της βιομηχανίας για μια γλώσσα που απευθύνει τα σημερινά τους σενάρια χρήσης του Παγκοσμίου Ιστού και τους περιορισμούς στην ανάπτυξη μιας γλώσσας οντολογίας που έχει εναρμονιστεί με καθιερωμένες επιστημονικές αρχές και ερευνητική εμπειρία," εξήγησαν οι Jim Hendler και Guus Schreiber, συν-προεδρεύοντες στην Ομάδα Εργασίας Οντολογίας του Παγκοσμίου Ιστού. "Περισσότερα από πενήντα μέλη της Ομάδας Εργασίας έχουν επιτυχώς

σχεδιάσει μια γλώσσα που απευθύνει και τα δύο σύνολα ανησυχιών και που γίνεται εξίσου αποδεκτή από ακαδημαϊκούς και από όσους την υλοποιούν.”

Ο σημασιολογικός Ιστός είναι ένα όραμα για το μέλλον του Ιστού, στον οποίο δίνεται στις πληροφορίες σαφές νόημα, διευκολύνοντας με αυτόν τον τρόπο τις μηχανές να επεξεργαστούν και να ενσωματώσουν πληροφορίες που είναι διαθέσιμες στον Ιστό αυτόματα. Ο σημασιολογικός Ιστός θα στηριχτεί στη δυνατότητα της XML να καθορίζει εξατομικεύσιμα σχήματα επικόλλησης ετικετών (customized tagging schemes) και στην εύκαμπτη προσέγγιση της RDF για την αναπαράσταση των δεδομένων. Το πρώτο επίπεδο μετά την RDF που απαιτείται για τον σημασιολογικό Ιστό, είναι μια γλώσσα οντολογίας η οποία μπορεί να περιγράψει επίσημα τη σημασία της ορολογίας που χρησιμοποιείται στα έγγραφα Ιστού. Εάν είναι αναμενόμενο από τις μηχανές να εκτελούν χρήσιμους συλλογισμούς χρησιμοποιώντας αυτά τα έγγραφα, η γλώσσα πρέπει να υπερβεί τη βασική σημασιολογία του Σχήματος RDF.

Η OWL είναι μέρος των συστάσεων του W3C σχετικών με το σημασιολογικό Ιστό :

- Η XML παρέχει μια επιφανειακή σύνταξη για δομημένα έγγραφα, αλλά δεν επιβάλλει κανέναν σημασιολογικό περιορισμό στο νόημα αυτών των εγγράφων.

- Το Σχήμα XML (XML Schema) είναι μια γλώσσα για τον περιορισμό της δομής των εγγράφων XML.

- Η RDF είναι ένα μοντέλο δεδομένων για αντικείμενα («πηγές πληροφοριών») και τις σχέσεις μεταξύ τους, παρέχει μια απλή σημασιολογία για αυτό το μοντέλο, και αυτά τα μοντέλα μπορούν να αναπαρασταθούν σε σύνταξη XML.

- Το Σχήμα RDF (RDF Schema) είναι ένα λεξιλόγιο για την περιγραφή των ιδιοτήτων και των κλάσεων των πηγών πληροφοριών RDF, με μια σημασιολογία για ιεράρχιες γενίκευσης τέτοιων ιδιοτήτων και κλάσεων.

- Η OWL προσθέτει περισσότερο λεξιλόγιο για την περιγραφή των ιδιοτήτων και των κλάσεων : μεταξύ άλλων, σχέσεις μεταξύ των κλάσεων (π.χ. ιδιότητα της μη επικάλυψης), αριθμό στοιχείων συνόλου (π.χ. ακριβώς ένα), ισότητα, πλουσιότερη παροχή τύπων για τις ιδιότητες,

χαρακτηριστικά των ιδιοτήτων (π.χ. συμμετρία), και απαριθμημένες κλάσεις.

Η OWL παρέχει τρεις όλο και περισσότερο εκφραστικές υπο-γλώσσες, σχεδιασμένες για χρήση σε ειδικές κοινότητες σχεδιαστών εφαρμογών και τελικών χρηστών. Αυτές είναι :

- **OWL Lite** – Προορίζεται για τους χρήστες που πρωταρχικά χρειάζονται μια ταξινομημένη ιεραρχία και απλούς περιορισμούς. Για παράδειγμα, ενώ υποστηρίζει περιορισμούς προτεραιότητας, επιτρέπει προτεραιότητες τιμών 0 ή 1. Ένα εργαλείο θα μπορούσε να φτιαχτεί πιο εύκολα χρησιμοποιώντας την OWL Lite, παρά τις πιο εκφραστικές OWL DL και OWL Full. Η έκδοση Lite παρέχει γρήγορη ανάπτυξη θησαυρών (thesaurus) και ταξονομιών. Τέλος, έχει μικρότερη τυπική (formal) πολυπλοκότητα από την OWL DL.

- **OWL DL** – Προορίζεται για τους χρήστες που χρειάζονται τη μέγιστη δυνατή εκφραστικότητα, ενώ παράλληλα διατηρεί την υπολογιστική της ικανότητα (είναι εγγυημένο ότι όλα τα συμπεράσματα θα είναι υπολογίσιμα) και αποφασιστικότητα (όλοι οι υπολογισμοί θα τελειώσουν σε ορισμένο χρόνο). Η έκδοση DL περιλαμβάνει όλες τις λειτουργίες της OWL, αλλά αυτές μπορούν να κάνουν αναπαράσταση Ιστού με Οντολογίες & Βάσεις Γνώσης Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών.

Μπορούν χρησιμοποιηθούν μόνο υπό ορισμένους περιορισμούς (π.χ., ενώ μια κλάση μπορεί να είναι υποκλάση πολλών κλάσεων, μια κλάση δε δύναται να είναι στιγμιότυπο μιας άλλης κλάσης). Τα αρχικά αυτής της έκδοσης προκύπτουν σε αντιστοιχία με τα «description logics», τα οποία αποτελούν ένα πεδίο έρευνας που έχει μελετήσει τα «logics» που σχηματίζουν τα τυπικά θεμέλια της OWL.

- **OWL Full** – Προορίζεται για τους χρήστες που χρειάζονται τη μέγιστη δυνατή εκφραστικότητα και τη συντακτική ελευθερία της RDF, αλλά χωρίς να εγγυάται την υπολογισσιμότητα. Για παράδειγμα, μια κλάση μπορεί να είναι ταυτόχρονα μια συλλογή ατομικών στοιχείων ή ένα ατομικό στοιχείο

από μόνη της. Η έκδοση Full επιτρέπει σε μια οντολογία να επαυξάνει το περιεχόμενο και τη σημασία του προκαθορισμένου (RDF ή OWL) λεξιλογίου. Όπως φαίνεται, είναι κάπως απίθανο οποιοδήποτε λογισμικό που παρέχει συλλογισμούς, να είναι ικανό να υποστηρίξει πλήρεις συλλογισμούς για κάθε χαρακτηριστικό της OWL Full.

Κάθε μια από αυτές τις τρεις υπο-γλώσσες είναι προέκταση του πιο απλού προκατόχου : α) Τι είναι νόμιμο να εκφραστεί ; β) Πώς μπορεί να βγει ένα έγκυρο συμπέρασμα ; Ακολουθούν κάποιες σχέσεις μεταξύ των τριών υπο-γλωσσών. Οι αντίστροφες σχέσεις δεν ισχύουν.

- Κάθε νόμιμη οντολογία της OWL Lite είναι και νόμιμη στην OWL DL.
- Κάθε νόμιμη οντολογία της OWL DL είναι και νόμιμη στην OWL Full.
- Κάθε έγκυρο συμπέρασμα της OWL Lite είναι και έγκυρο στην OWL DL.
- Κάθε έγκυρο συμπέρασμα της OWL DL είναι και έγκυρο στην OWL Full.

Οι σχεδιαστές οντολογιών που υιοθετούν την OWL θα πρέπει να λάβουν υπόψιν ποια υπο-γλώσσα ταιριάζει καλύτερα στις ανάγκες τους. Η επιλογή ανάμεσα στις εκδόσεις DL και Lite εξαρτάται από το βαθμό στον οποίο οι χρήστες θα απαιτήσουν μεγαλύτερη εκφραστικότητα την οποία παρέχει η DL. Η επιλογή ανάμεσα στη DL και τη Full εξαρτάται κυρίως, από το βαθμό στον οποίο οι χρήστες θα απαιτήσουν την ευχέρεια των μετα-μοντέλων του RDF Σχήματος (RDF Schema). Συγκρίνοντας τη Full με τη DL, μπορούμε να συμπεράνουμε ότι οι συλλογισμοί της Full είναι λιγότερο προβλέψιμοι, αφού ούτως ή άλλως δεν υπάρχουν πλήρεις υλοποιήσεις της προς το παρόν.

Η OWL Full μπορεί να θεωρηθεί ως μια προέκταση της RDF, ενώ η Lite και η DL ως προεκτάσεις μιας παριορισμένης όψης της RDF. Κάθε κείμενο OWL (οποιασδήποτε έκδοσης) είναι ένα κείμενο RDF, και κάθε κείμενο RDF είναι ένα κείμενο OWL Full. Από την άλλη, μόνο μερικά κείμενα RDF θα είναι έγκυρα κείμενα OWL Lite ή OWL DL. Έτσι, θα πρέπει να δοθεί ιδιαίτερη προσοχή στην περίπτωση που ο χρήστης θέλει να μετατρέψει ένα κείμενο RDF σε OWL. Όταν η εκφραστικότητα της DL και της Lite κριθεί κατάλληλη, πρέπει να ληφθεί πρόνοια για να εξασφαλιστεί ότι το αρχικό κείμενο RDF είναι συμβατό με τους επιπρόσθετους περιορισμούς που υπαγορεύονται από αυτές.

4.4.1 Μελέτη περίπτωσης : Ανάπτυξη Συστήματος ΙΚΑ

Η Οντολογική προσέγγιση είναι απαραίτητη σε πολύπλοκα συστήματα καθώς μας βοηθά να αποφύγουμε παραλείψεις και λάθη στην μετέπειτα πορεία του έργου που είναι ο σχεδιασμός του συστήματος. Ένα ακόμη μεγάλο κίνητρο είναι ότι ενώ

βάζει τις ισχυρότερες βάσεις ανάλυσης, ταυτόχρονα μπορεί να χρησιμοποιηθεί από όλες σχεδόν τις μεθοδολογίες και εργαλεία σχεδίασης συστημάτων. Με βάση την παραπάνω επιλογή μας για την μεθοδολογία ανάλυσης, θα χρησιμοποιήσουμε την γλώσσα OWL για την ανάπτυξη του παραδείγματος μας καθώς είναι η πιο διαδεδομένη σε αυτή την μεθοδολογία.

Το ΙΚΑ θέλοντας να βελτιώσει την λειτουργία του σε θέματα που αφορούν την μείωση της γραφειοκρατίας, την ελαχιστοποίηση της ταλαιπωρίας των ασφαλισμένων, την γρήγορη εύρεση και επεξεργασία των δεδομένων του και την εύκολη διαγραφή των άχρηστων δεδομένων αποφάσισε την εφαρμογή ενός πληροφοριακού συστήματος.

Για την καλύτερη κατανόηση της υπάρχουσας εργασίας, θα δείξουμε με ένα παράδειγμα ανάλυσης και σχεδιασμού πληροφοριακού συστήματος. Το παράδειγμα μας αφορά μέρος του πληροφοριακού συστήματος του ΙΚΑ και ποιο συγκεκριμένα θα σχεδιάσουμε ένα μέρος του υποσυστήματος παροχών.

Το ΙΚΑ είναι ένας τεράστιος οργανισμός με χιλιάδες ασφαλισμένους και ακόμα δεν έχει μηχανογραφηθεί πλήρως. Η διαδικασία αυτή λόγω του όγκου των δεδομένων είναι αρκετά πολύπλοκη και χρονοβόρα, αλλά και άκρως απαραίτητη. Για τον λόγω αυτό θα ασχοληθούμε με ένα επιμέρους τμήμα του συστήματος αυτού και με βάση την πολυπλοκότητα του επιλέγουμε την οντολογική προσέγγιση. Αρχικά γίνεται η Μελέτη σκοπιμότητας όπου ο χρήστης έρχεται σε επαφή με τον ειδικό αναλυτή για να γίνει σωστή καταγραφή και αποτίμηση του μεγέθους του προβλήματος. Συγκεκριμένα υπάλληλος του ΙΚΑ έκρινε πως πρέπει να βοηθήσει στην καλύτερη λειτουργία του συστήματος για αυτό και συνεργάστηκε με τον ειδικό αναλυτή και του είπε σε ποια μέρη του συστήματος

έχει αυτός πρόβλημα. Οι χρήστες εντόπισαν το πρόβλημα στην διαφοροποίηση και ενημέρωση των δεδομένων, στην παρακολούθηση της πορείας μιας νομικής πράξης του ασφαλισμένου και στην ανταλλαγή πληροφοριών και δεδομένων με φορείς του εξωτερικού.

Εφόσον έχει γίνει ο προσδιορισμός του προβλήματος κρίθηκε πως πρέπει να λυθούν τα προβλήματα αυτά και κατόπιν προχωράμε στην ανάλυση απαιτήσεων.

Η φάση ανάλυσης απαιτήσεων καθορίζει της επιχειρησιακές απαιτήσεις για το νέο σύστημα. Συγκεκριμένα στο ΙΚΑ το ζητούμενο είναι να αντιμετωπιστούν τα παραπάνω προβλήματα και όχι να μετατραπεί το σύστημα σε κάτι καλύτερο και πολυπλοκότερο.

Οι επιμέρους λειτουργίες του συστήματος του ΙΚΑ που απαιτούν βελτίωση είναι :

- **Διορθωτικές ενέργειες**

Στο σύστημα παροχών που επιχειρούμε να αναλύσουμε είναι αναγκαίο να έχει τη δυνατότητα διορθωτικών παρεμβάσεων και με τέτοιο τρόπο που δεν θα επηρεάζει κάποια άλλα κομμάτια του συστήματος που δεν θέλουμε να διαφοροποιηθούν.

- **Διαχείριση ενστάσεων προσφυγών και εφέσεων**

Είναι αναμενόμενο σε έναν τόσο μεγάλο οργανισμό με τεράστιο αριθμό εγγραφών και λειτουργιών, να εντοπίζονται σημεία όπου υπάρχουν λάθη σε λειτουργίες όπως οι συστάσεις, οι προσφυγές και οι εφέσεις από τους ασφαλισμένους προς τον οργανισμό μας. Εδώ ο σκοπός μας είναι η οργανωμένη και αποτελεσματική διαχείριση των ενστάσεων, των προσφυγών και των εφέσεων. Πρωταρχικός στόχος μας είναι η άμεση και αναλυτική μελέτη τους. Στη συνέχεια εφόσον διαπιστωθεί ότι βάσει κάποιων κριτηρίων που θα ληφθούν υπόψη είναι έγκυρες, θα πρέπει με τις κατάλληλες διαδικασίες σε ένα αποδεκτό χρονικό διάστημα να διεκπεραιωθούν.

- **Ηλεκτρονική ανταλλαγή εντύπων με ασφαλιστικούς φορείς του εξωτερικού**

Το ήδη υπάρχον σύστημα ανταλλαγής δεδομένων του ΙΚΑ με τους σε σχέση με τους υπόλοιπους Ευρωπαϊκούς ασφαλιστικούς φορείς, δεν εξυπηρετεί άμεσα τις υπάρχουσες ανάγκες. Απαραίτητη διαδικασία είναι να προβούμε σε μία μελέτη σκοπιμότητας έτσι ώστε να διαπιστώσουμε κατά πόσο είναι δυνατό να εκσυγχρονιστεί αυτό το σύστημα. Η μελέτη αυτή θα περιέχει τρόπους επικοινωνίας του τμήματος παροχών με τα υπόλοιπα τμήματα του ΟΠΣ και διασύνδεσης (ηλεκτρονική ανταλλαγή δεδομένων) του ΟΠΣ του ΙΚΑ με τα πληροφοριακά συστήματα των υπολοίπων ευρωπαϊκών ασφαλιστικών φορέων.

Το επόμενο βήμα είναι ο λογικός σχεδιασμός ο οποίος καταγράφει τις επιχειρησιακές απαιτήσεις χρησιμοποιώντας μοντέλα συστημάτων όπου επεξηγείτε η δομή δεδομένων. Κατά μία έννοια επικυρώνονται οι απαιτήσεις που καθιερώθηκαν στην προηγούμενη φάση.

Στο σημείο αυτό αναπτυχθούν οι διαδικασίες και οι ροές στοιχείων μας μέσω ενός διαγράμματος ER. Θα αναλύσουμε τις οντότητες από τις οποίες αποτελείται το διάγραμμα καθώς και τις συσχετίσεις μεταξύ τους.

Οντότητες :

- Ασφαλισμένος (Όνομα, Αρ. Μητρώου)
- Νομικές Πράξεις (Ενστάσεις, Εφέσεις, Προσφυγές)
- Παροχές (Νοσηλεία, Συντάξεις)
- Εξωτερικό (Γοσό, Χώρα)

Υποοντότητες :

- Νοσηλεία (Φάρμακα, Ιδιωτ. Νοσοκ., Δημ. Νοσοκ.)
- Συντάξεις (Αρ. Ενσήμων, Αναπηρία, Κατ. Σύνταξης, Πολύτεκνος)

Συσχετίσεις :

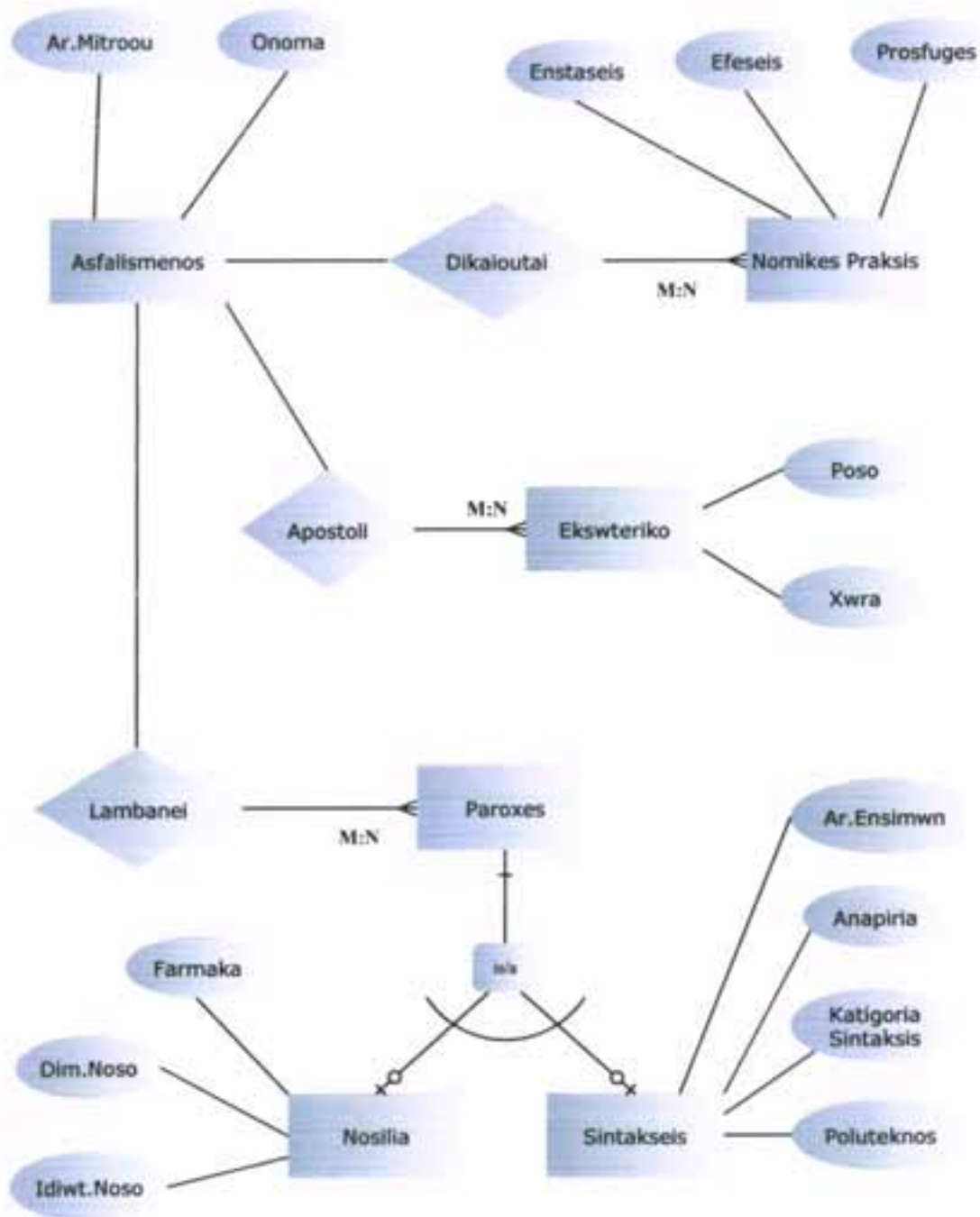
- Δικαιούται (Νομικές Πράξεις)
- Αποστέλλεται (Εξωτερικό)
- Λαμβάνει (Παροχές)

Τύποι Συσχετίσεων :

- Δικαιούται (M:N Πολλοί ασφαλισμένοι δικαιούνται πολλές νομικές πράξεις)

- Αποστέλλεται (M:N Πολλοί ασφαλισμένοι αποστέλλονται σε πολλές χώρες του εξωτερικού)
- Λαμβάνει (M:N Πολλοί ασφαλισμένοι λαμβάνουν πολλές παροχές)

4.4.2 Ανάπτυξη οντολογίας για το διάγραμμα Οντοτήτων Σχέσεων (ER)



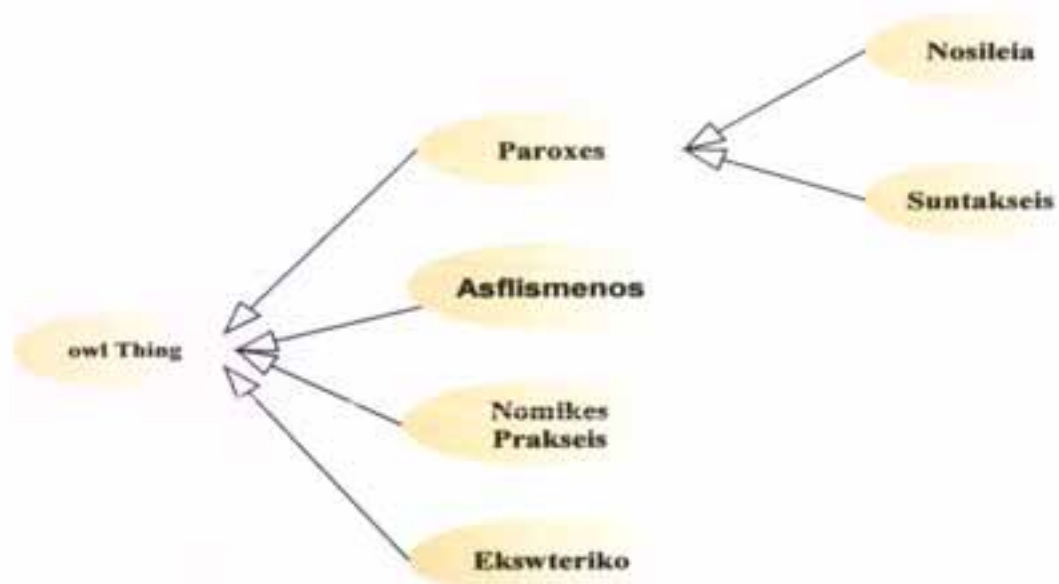
Σχήμα 4.1

Το πρότυπο σχέσης και χρόνου του διαγράμματος ER περιλαμβάνει τις κατηγορίες αντικειμένων και τις κατηγορίες σχέσης. Αυτό το πρότυπο επεκτείνεται έτσι ώστε οι τρέχουσες απαιτήσεις, στην επιχειρησιακή διαμόρφωση, να μπορούν να ενσωματωθούν. Ο βασικός μηχανισμός ER, καθορίζει τους παραγώγους που εμφανίζονται μέσα στην εφαρμογή.

Εφόσον επελέγη η μεθοδολογία της OWL, το case tool που θα χρησιμοποιηθεί για την απεικόνιση της ανάπτυξης οντολογιών είναι το Protégé διότι είναι γρήγορο στην κατανόηση, απλό και με ελεύθερη άδεια προς χρήση. Πιο κάτω βλέπουμε πως συσχετίζονται μεταξύ τους οι κλάσεις και τι δηλώνει η κάθε μια. Οι κλάσεις μας είναι οι εξής:

- Ασφαλισμένος : όπου αναφέρονται στα στοιχεία του ασφαλισμένου
 - Νομικές Πράξεις : όπου αναφέρονται τα νομικά δικαιώματα του ασφαλισμένου.
 - Παροχές : όπου αναφέρει τις δυο κατηγορίες παροχών που δικαιούται ο ασφαλισμένος.
1. Νοσηλεία : όπου περιλαμβάνει την κατηγορία του νοσοκομείου και την φαρμακευτική περίθαλψη.
 2. Συντάξεις : όπου περιλαμβάνει την κατηγορία σύνταξης και τις προϋποθέσεις για τον καθορισμό του ποσού.
Εξωτερικό : όπου αναφέρεται η χώρα και το ποσό που χρειάστηκε για νοσηλεία ο ασφαλισμένος.

4.4.3 Το διάγραμμα της γλώσσας OWL είναι ως εξής :



Σχήμα 4.2

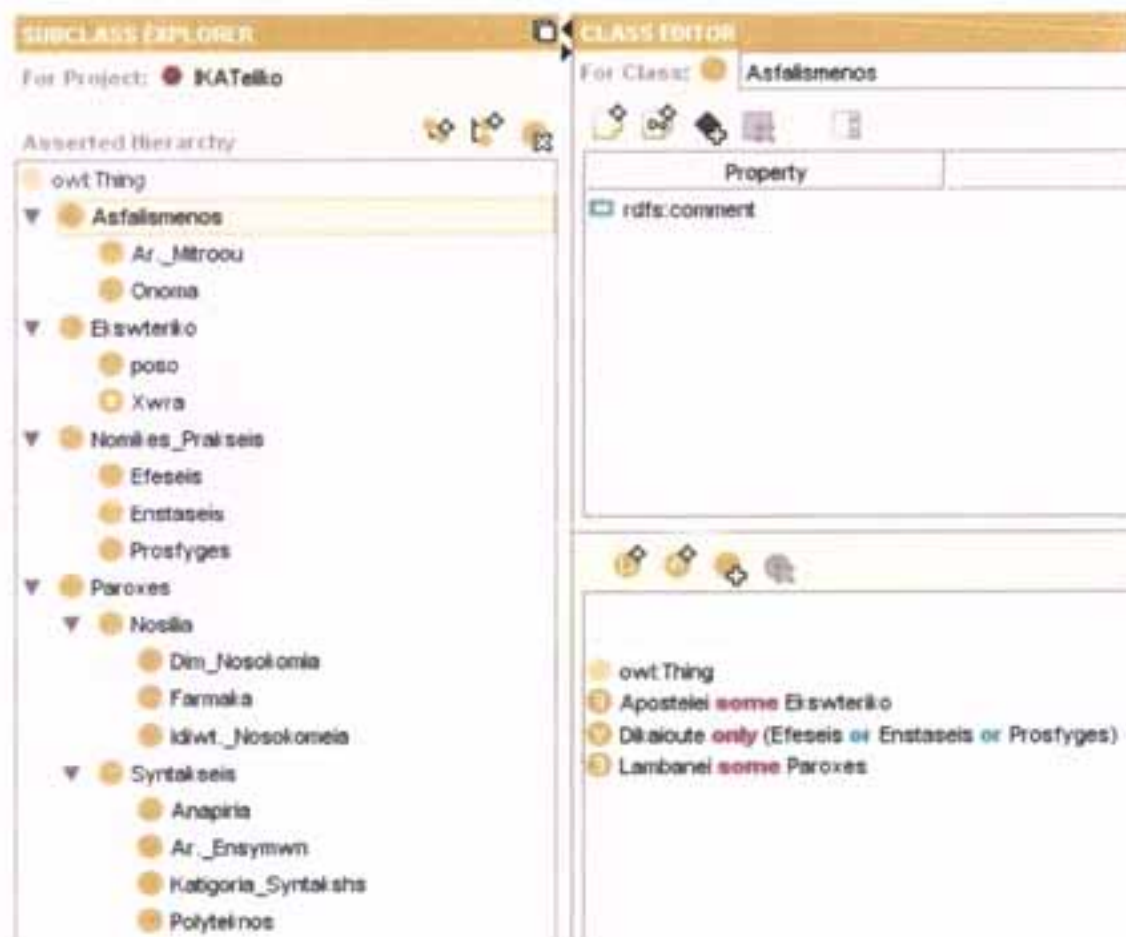
4.4.4 Εισαγωγή στοιχείων στο Protégé

Μετά την εισαγωγή των στοιχείων στο πρόγραμμα Protégé φαίνονται οι κλάσεις.



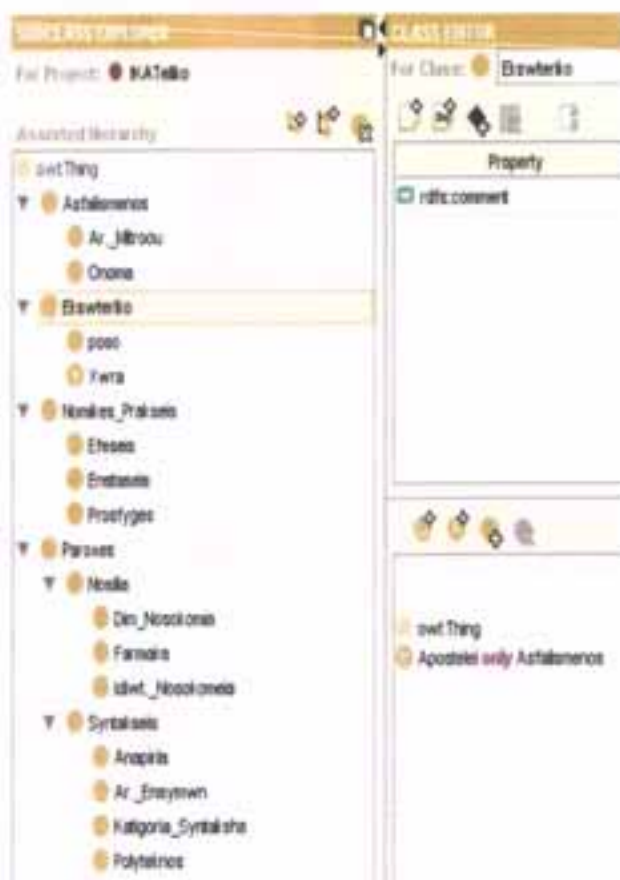
Σε αυτό το σχήμα φαίνεται η ιεραρχία των οντοτήτων. Στην ιεραρχία κάτω από την ρίζα βρίσκονται οι βασικοί τύποι οντοτήτων. Κάτω από τις βασικές οντότητες μπορεί να υπάρχουν υπο-οντότητες και κάτω από αυτές άλλες αναλόγως την πολυπλοκότητα. Για παράδειγμα στην οντότητα Paroxes φαίνεται ότι υπάρχουν δύο υπό οντότητες Nosilia και Syntakseis

Πατώντας στον ασφαλισμένο, εμφανίζονται οι σχέσεις και οι περιορισμοί της συγκεκριμένης οντότητας στο κάτω και δεξιά μέρος της παρακάτω εικόνας :



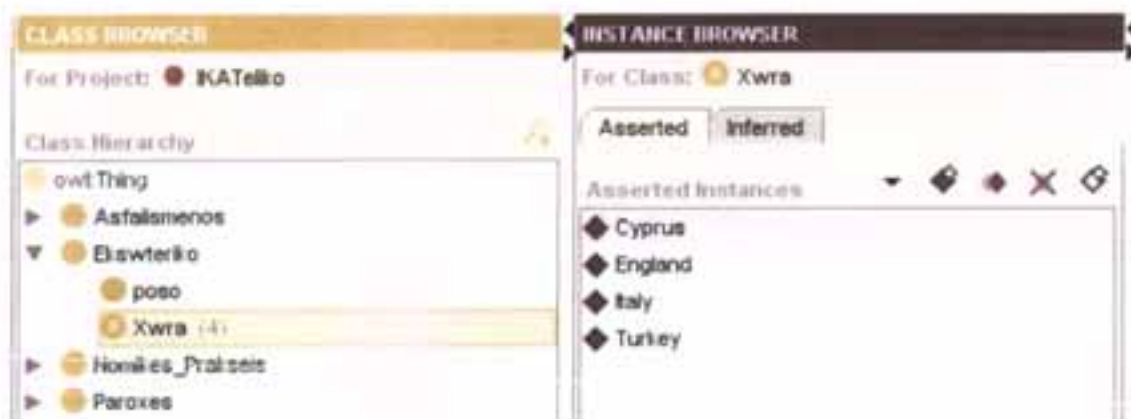
Σύμφωνα με την ανάλυση που έγινε κρίθηκε πως ο ασφαλισμένος μπορεί να έχει ασφάλιση σε ορισμένες μόνο περιπτώσεις (*some*) αν βρίσκεται στο εξωτερικό για παράδειγμα μόνο για σοβαρό και επείγον περιστατικό και μόνο αν πάει σε συγκεκριμένα νοσοκομεία. Δικαιούται να κάνει μόνο (*only*) εφέσεις ενστάσεις ή προσφυγές και καμία άλλη δικαστική ενέργεια. Τέλος λαμβάνει κάποιες (*some*) από τις παροχές αναλόγως την κατάσταση που θα βρίσκεται ο ασφαλισμένος.

Στο σχήμα που ακολουθεί βλέπουμε όλες τις συσχετίσεις όλων των οντοτήτων του I.K.A. Εκεί καταχωρήθηκαν αρχικά στο πρόγραμμα Protégé για τις επιλέξουμε αργότερα ανά οντότητα.

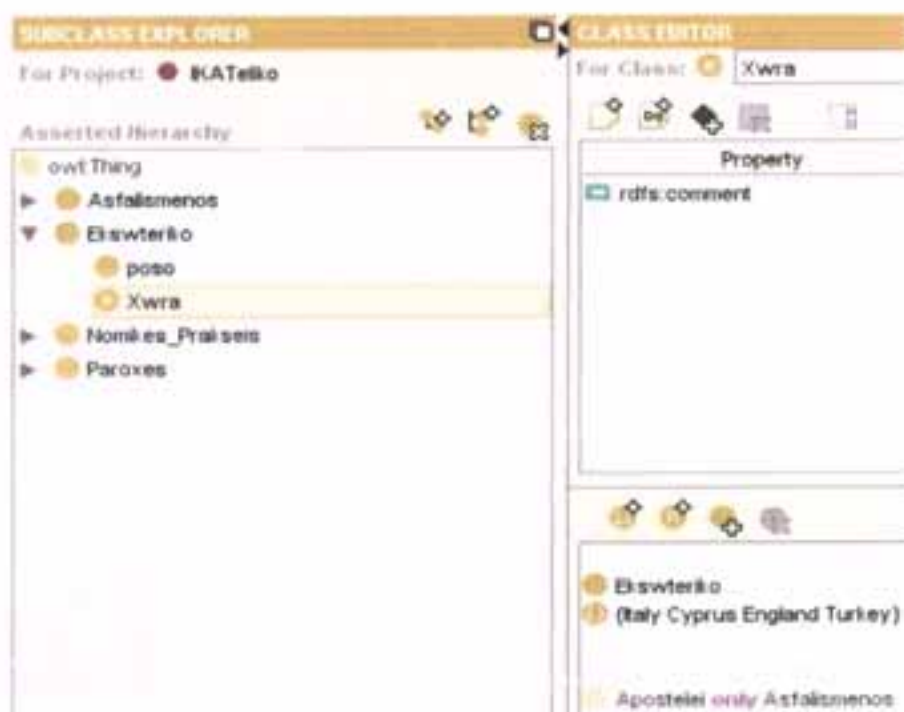


Όπως φαίνεται και στο σχήμα στα αριστερά πατώντας στην οντότητα του εξωτερικού, εμφανίζονται οι περιορισμοί που έχουν και όλες οι υπόλοιπες οντότητες δηλαδή για να αποσταλεί κάποιος στο εξωτερικό πρέπει να είναι ασφαλισμένος

Σε κάθε τύπο οντοτήτων υπάρχει η δυνατότητα να οριστούν οι ιδιότητές του που παριστάνονται με την παρακάτω μορφή όπως βλέπουμε στην εικόνα που ακολουθεί. Ο τύπος οντότητας Χωρα περιλαμβάνει τέσσερις ιδιότητες (τις χώρες σαν πεδία που χαρακτηρίζουν σε ποιο μέρος ο ασφαλισμένος δέχεται περίθαλψη)



Αυτό έγινε γιατί η συγκεκριμένη οντότητα περιλαμβάνει απλώς τις χώρες που πιθανόν να σταλεί κάποιος ασφαλισμένος. Στο σχήμα που ακολουθεί φαίνεται πως παρουσιάζεται η οντότητα χώρα με τις ιδιότητες της στην κεντρική σελίδα του προτέγ



Με την βοήθεια των προγραμματιστικών εργαλείων smart draw και proτέγέ, γίνεται ευκολότερα κατανοητή η έννοια του ER διαγράμματος, της οντολογίας και γενικότερα η βασική απεικόνιση της φάσης της ανάλυσης συστήματος. Μετά την ολοκλήρωση της φάσης αυτής, τα στοιχεία αυτά παραδίδονται στους σχεδιαστές συστήματος και με βάση αυτά θα συνεχίσουν στην σχεδίαση του.

Λαμβάνοντας υπόψη τις επιχειρησιακές απαιτήσεις για ένα βελτιωμένο σύστημα, πρέπει να αποφασιστεί ποιος είναι ο απαραίτητος τεχνολογικός εξοπλισμός για να λειτουργήσει σωστά το νέο σύστημα. Σύμφωνα με τις απαραίτητες επιδιορθώσεις και προσθήκες που πρέπει να γίνουν ώστε το σύστημα να αποδώσει τα απαιτούμενα, θα πρέπει να αναβαθμιστούν οι ηλεκτρονικοί υπολογιστές των χρηστών και να αγοραστούν κάποιοι επιπλέον που θα λειτουργούν ως εξυπηρετητές. Ακόμα θα χρειαστούν δρομολόγητες, καλώδιο δικτύου, καθώς και βελτιώσεις στις τηλεπικοινωνιακές εγκαταστάσεις του οργανισμού. Ο σκοπός σε αυτή την φάση είναι να αναγνωριστούν οι υποψήφιες λύσεις, να μελετηθούν και να προταθεί η λύση για το σύστημα που θα εφαρμοστεί.

Κεφάλαιο 5 ΣΥΜΠΕΡΑΣΜΑΤΑ

Φτάνοντας στο τέλος της εργασίας γίνεται εύκολα αντιληπτή η σημαντικότητα των πληροφοριακών συστημάτων στις επιχειρήσεις και στους οργανισμούς. Κάποια χρόνια πριν ο κόσμος πίστευε ότι αυτού του είδους η τεχνολογία είναι για εξειδικευμένες επιχειρήσεις και ειδικότερα στον τομέα της πληροφορικής. Πλέον είναι τόσο εμφανή τα πλεονεκτήματα που προσφέρουν σε οποιαδήποτε επιχείρηση, με αποτέλεσμα η γνώμη αυτή να έχει αλλάξει. Τα πλεονεκτήματα των πληροφοριακών συστημάτων έχουν να κάνουν με την βελτίωση της λειτουργίας μια επιχείρησης, σε θέματα όπως η ταχύτητα αναζήτησης δεδομένων άρα και εξυπηρέτησης πελατών, στην ελαχιστοποίηση σφαλμάτων, στην ταχύτερη και καλύτερη αντίληψη της κατάστασης και σε επιμέρους τομείς αυτών. Αυτό που έκανε όμως τις επιχειρήσεις να στραφούν προς τα πληροφοριακά συστήματα, είναι το οικονομικό κέρδος που έχουν σε σχέση με το κόστος ενός συστήματος.

Αυτό είναι και το κλειδί της φάσης της ανάλυσης συστημάτων αφού η λεπτομερής αποτύπωση της υπάρχουσας κατάστασης και ο ακριβής εντοπισμός του προβλήματος, με την βοήθεια όλων των προαναφερθέντων γίνεται με έναν και μοναδικό σκοπό. Την λήψη της σωστής απόφασης στο τέλος της φάσης αυτής που δεν έχει να κάνει με τίποτε άλλο από την σχέση κόστους- οφέλους. Για τους αναλυτές η φάση αυτή μπορεί να είναι ένα κομμάτι της ανάπτυξης του πληροφοριακού συστήματος, για τις επιχειρήσεις όμως στην πραγματικότητα είναι μια οικονομοτεχνική μελέτη που θα κρίνει αν το έργο αυτό είναι συμφέρων για την επιχείρηση ή όχι.

Ο σχεδιασμός του συστήματος είναι η φάση που αφορά την επιχείρηση μόνο ως προς το περιβάλλον διαεπαφής, έτσι ώστε να είναι φιλικό και βολικό για τον χρήστη. Μια ακόμα βοήθεια που μπορεί να δώσει ένας χρήστης στους σχεδιαστές είναι η περιγραφή των διαδικασιών και λειτουργιών του συστήματος με σκοπό να υπάρχουν λιγότερες διορθώσεις στο τέλος.

Το υπόλοιπο μέρος του σχεδιασμού και ένα μεγάλο μέρος της ανάλυσης ανήκει στους αναλυτές και στους σχεδιαστές. Ένα βασικό πρόβλημα που

καλούνται να λύσουν αυτοί είναι η επιλογή μεθοδολογίας ανάπτυξης του συστήματος. Αυτό μπορεί να γίνει είτε για κάθε φάση ξεχωριστά, είτε για όλη την ανάπτυξη του συστήματος. Η επιλογή μεθοδολογίας είναι πολύπλοκη και προκύπτει μετά από μελέτη και έρευνα πολλών ατόμων. Δεν υπάρχει μία μεθοδολογία που να είναι η καλύτερη για κάθε περίπτωση, αλλά λαμβάνονται διάφορα κριτήρια για την επιλογή της όπως : Η διαύγεια απαιτήσεων χρήστη, η πολυπλοκότητα του συστήματος, αξιοπιστία του συστήματος, το βραχυπρόθεσμο χρονοδιάγραμμα και η ορατότητα βαθμού προόδου. Η οντολογία είναι ένας απαραίτητος παράγοντας για την μοντελοποίηση και την διαχείριση της γνώσης. Στην περίπτωση των πληροφοριακών συστημάτων, ως οντολογία χαρακτηρίζεται η ενιαία και γενικά αποδεκτή σημασιολογική κωδικοποίηση της πληροφορίας ενός θεματικού χώρου. Η διαδικασία της μετατροπής της πληροφορίας σε σημασιολογική είναι δυνατή μέσω ενός πλαισίου περιγραφής, του RDF (Resource Description Framework). Ο κύριος στόχος των οντολογιών είναι η διανομή και επαναχρησιμοποίηση της γνώσης έτσι ώστε να μην χάνονται οι πολύτιμες πληροφορίες μιας επιχείρησης αλλά να μπορούν με κάποιο τρόπο να διαρθρώνονται και να αξιοποιούνται. Η έρευνα που χρειάστηκε να γίνει προκειμένου να βρεθούν προβλήματα του ΙΚΑ αλλά και οι λύσεις είναι υποθετικά στοιχεία, και μας βοηθούν απλώς στην απεικόνιση της φάσης της ανάλυσης για την καλύτερη κατανόηση της.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- Myrvin,Chester, Avtar,Athwall "Basic Information Systems Analysis and Design"
McGraw-Hill Education - Europe, 2001
- Bennet, Simon "Object-oriented Information Systems Analysis and Design Using
UML",McGraw-Hill Education – Europe, 1999
- Bart,Prakken "Information, Organization and Information Systems Design" , Kluwer
Academic Publishers Group, 2000
- Avison,Fitzgerald "Ανάπτυξη Προηγμένων Πληροφοριακών Συστημάτων" ,Εκδόσεις
Νέων Τεχνολογιών, 2006
- Βασιλακόπουλος Γ., Χρυσικόπουλος Β. " Πληροφοριακά Συστήματα Διοίκησης:
Ανάλυση και Σχεδιασμός", Εκδόσεις Σταμούλη, 1990
- Schlenoff, C., Knutilla, A., Ray, S. "Unified Process Specification Language:
Requirements for Modeling Process", NIST Interagency Report 5910, Gaithersburg, MD,
1996
- Bertolazzi P., C.Krusich, M.Missikoff "An Approach to the Definition of a Core
Enterprise Ontology: CEO", 1996
- Λαοπόδης Βασ.: "Ανάπτυξη Πληροφοριακών Συστημάτων: Ανάλυση και Σχεδιασμός
Συστημάτων", Εκδόσεις Νέων Τεχνολογιών, 1996.

Ηλεκτρονική Βιβλιογραφία

- <http://www.dmst.aueb.gr/dds/ism/oo/index.htm>
- <http://www.spinellis.gr/c3/uml/intro.htm>
- <http://owl.optimism.co.nz/>
- <http://protege.stanford.edu/overview/protege-owl.html>
- http://www.umsl.edu/~sauter/analysis/er/er_intro.html
- <http://www.smartdraw.com/exp/ste/examples/>
- <http://en.wikipedia.org/wiki/>
- <http://www.maes.hw.ac.uk/~lachlan/ism.html>
- <http://sem.ualgary.ca/courses/seng/613/F97/grp4/ssmfinal.html>